

**Δημοκρίτειο Πανεπιστήμιο Θράκης Πολυτεχνική Σχολή**

**Τμήμα Μηχανικών Παραγωγής και Διοίκησης**

Τομέας Συστημάτων Παραγωγής

Επιβλέπων: Αναπληρωτής Καθηγητής Κατσαβούνης Στέφανος

**Διαδικτυακή εφαρμογή παρακολούθησης & διαχείρισης έξυπνων συσκευών με σκοπό την μετατροπή ενός συμβατικού σπιτιού σε έξυπνο**

Διπλωματική Εργασία

**Μησσήν Στέφανος**

Αρ. Μητρώου: 91432



**Ξάνθη, Οκτώβριος 2018**



**Δημοκρίτειο Πανεπιστήμιο Θράκης Πολυτεχνική Σχολή**

**Τμήμα Μηχανικών Παραγωγής και Διοίκησης**

Τομέας Συστημάτων Παραγωγής

Επιβλέπων: Αναπληρωτής Καθηγητής Κατσαβούνης Στέφανος

**Διαδικτυακή εφαρμογή παρακολούθησης & διαχείρισης έξυπνων συσκευών με σκοπό την μετατροπή ενός συμβατικού σπιτιού σε έξυπνο**

Διπλωματική Εργασία

**Μησσήν Στέφανος**

Αρ. Μητρώου: 91432

Διπλωματική εργασία η οποία υποβλήθηκε τον Οκτώβριο 2018

για την απόκτηση του διπλώματος του Μηχανικού Παραγωγής και Διοίκησης

*Αφιερωμένο στους γονείς μου,*

***Κίμωνα*** *και* ***Μαρία***.

# Πρόλογος

Οι έννοιες των έξυπνων συσκευών (Smart Devices), των έξυπνων σπιτιών (Smart Homes) αλλά και των έξυπνων συμπλεγμάτων (Smart Clusters), συνδέονται με την έννοια των Πραγμάτων του Διαδικτύου (Internet of Things – IoT). Όλες οι παραπάνω έννοιες αποτελούν επίκεντρο της τεχνολογικής έρευνας. Έτσι, για την μετατροπή ενός συμβατικού σπιτιού σε έξυπνο, κρίνεται αναγκαία η δημιουργία ενός συστήματος επικοινωνίας, μεταξύ των έξυπνων συσκευών που είναι εγκατεστημένες στο σπίτι, αλλά και του ανθρώπου, έτσι ώστε ο δεύτερος, να διαχειρίζεται τις λειτουργίες που προσφέρουν οι έξυπνες συσκευές ανάλογα με τις απαιτήσεις του.

Η παρούσα διπλωματική εργασία μελετά την δημιουργία ενός συστήματος παρακολούθησης & διαχείρισης έξυπνων συσκευών με σκοπό την μετατροπή ενός συμβατικού σπιτιού σε έξυπνο. Πιο συγκεκριμένα, εξετάζονται οι εσωτερικές & εξωτερικές θερμοκρασίες του χώρου μέσω αισθητήριων οργάνων, γίνεται η αποθήκευση των μετρήσεων σε κατάλληλες βάσεις δεδομένων και έτσι ο χρήστης μπορεί να κρίνει πότε είναι αναγκαία η ενεργοποίηση ή η απενεργοποίηση των έξυπνων συσκευών εντός του χώρου του σπιτιού (στην παρούσα διπλωματική έχουν χρησιμοποιηθεί ως έξυπνες συσκευές, το κλιματιστικό και ο αφυγραντήρας). Η διαχείριση των έξυπνων συσκευών γίνεται μέσω από την διαδικτυακή εφαρμογή (Web Application) Smart Homie που δημιουργήθηκε για τις ανάγκες της παρούσας διπλωματικής εργασίας. Επιπλέον, εξετάζονται τα πρωτόκολλα αποστολής δεδομένων από τα αισθητήρια όργανα προς τις βάσεις δεδομένων, που είναι η ασύρματη μεταφορά δεδομένων (Wi-Fi) καθώς και η ενσύρματη μεταφορά δεδομένων (Ethernet), οι τρόποι αποφυγής απωλειών δεδομένων, καθώς και η εύρυθμη επικοινωνία των έξυπνων συσκευών με την διαδικτυακή εφαρμογή.

Το πρόβλημα προσεγγίστηκε από διαφορετικές διαστάσεις. Αρχικά διαπιστώθηκε το κατά πόσο είναι εφικτό, η δημιουργία μίας γέφυρας επικοινωνίας μεταξύ των έξυπνων συσκευών και του ανθρώπου. Έπειτα αναπτύχθηκε ο κατάλληλος αλγόριθμος για την συλλογή πρωτογενών δεδομένων θερμοκρασίας και υγρασίας εντός και εκτός του σπιτιού που χρησιμοποιήθηκε για την παρούσα εργασία, και τέλος αναπτύχθηκε μία διαδικτυακή εφαρμογή με σκοπό την διεπαφή του χρήστη με τις έξυπνες συσκευές αλλά και την διαχείριση τους.

Ακολούθησε μια εκτενής ανάλυση των αποτελεσμάτων. Διαπιστώθηκαν οι περιορισμοί και οι υποθέσεις της εργασίας που αφορούν το κατά πόσο εύχρηστη θα μπορεί να είναι μία τέτοιου είδους διαδικτυακή εφαρμογή για τον άνθρωπο, καθώς και οι τρόποι βελτίωσης της.

Αναλυτικότερα στο 1ο κεφάλαιο παρουσιάζεται το πρόβλημα, η σημασία του για την ανθρωπότητα, οι στόχοι της διπλωματικής εργασίας καθώς και η δομή αυτής.

Στο 2ο κεφάλαιο εξετάζεται το θεωρητικό υπόβαθρο, αναλύονται οι τεχνολογίες καθώς και ο εξοπλισμός που χρησιμοποιήθηκε για την υλοποίηση της διαδικτυακής εφαρμογής.

Έπειτα αναλύεται το υλικό μέρος, η συνδεσμολογία του συστήματος για την άντληση δεδομένων και τέλος, η σύνδεση αυτών με τον τοπικό διακομιστή και τις βάσεις δεδομένων στο 3ο κεφάλαιο.

Στο κεφάλαιο 4 επεξηγείται λεπτομερώς η σχεδίαση και η υλοποίηση της διαδικτυακής εφαρμογής. Πιο συγκεκριμένα, αναλύονται οι απαιτήσεις του συστήματος, γίνεται λόγος για την ασφάλεια των δεδομένων του χρήστη και πώς προστατεύονται τα παραπάνω. Ακόμη, αναλύονται οι βάσεις δεδομένων και οι λειτουργίες της εφαρμογής.

Στο 5ο κεφάλαιο συνοψίζονται τα συμπεράσματα και τα προβλήματα που αντιμετωπίστηκαν κατά την υλοποίηση της διπλωματικής εργασίας και αναλύονται περαιτέρω, τρόποι βελτίωσης της διαδικτυακής εφαρμογής.

# Ευχαριστίες

Στα πλαίσια της παρούσας διπλωματικής εργασίας, αισθάνομαι την υποχρέωση να ευχαριστήσω ορισμένους ανθρώπους που γνώρισα, όπου χωρίς αυτούς θα ήταν πολύ δυσκολότερος και δύσβατος ο δρόμος που έχω διανύσει για να φτάσω ως εδώ. Συνέβαλλαν όλοι με τον τρόπο τους στην διαμόρφωση του χαρακτήρα και της νοοτροπίας μου. Με βοήθησαν να πετύχω πολλά από τα όνειρα μου αλλά και να δημιουργήσω νέα, βάζοντας έτσι στόχο, στο προσεχές μέλλον να κατορθώσω και να εκπληρώσω όλες τις φιλοδοξίες μου.

Αρχικά οφείλω να ευχαριστήσω τον κ. Κατσαβούνη Στέφανο, Αναπληρωτή Καθηγητή του Τμήματος Μηχανικών Παραγωγής και Διοίκησης της Πολυτεχνικής Σχολής του Δημοκριτείου Πανεπιστημίου Θράκης, επιβλέποντα της παρούσας διπλωματικής εργασίας μου. Η καθοδήγηση, οι προτάσεις για βελτίωση καθώς και η συνεχής διαθεσιμότητα του σε ότι χρειαζόμουν, με βοήθησε να ξεπεράσω τις δυσκολίες που αντιμετώπισα στην διπλωματική μου εργασία και να κάνω τις ιδέες μου πραγματικότητα. Από την πρώτη κιόλας μέρα της γνωριμίας μας, κατάφερε να μου κεντρίσει το ενδιαφέρον, μεταδίδοντας μου έτσι κομμάτι των γνώσεων & και των εμπειριών του όσο αφορά επιστημονικούς κλάδους, που ανταποκρίνονται στα ενδιαφέροντα μου.

Τους Καθηγητές του Τμήματος Μηχανικών Παραγωγής και Διοίκησης, που με την πολυετή πείρα τους κατάφεραν να μεταδώσουν με επιτυχία πολύτιμες γνώσεις και εξελίξεις της επιστήμης μας, χρήσιμες για την περαιτέρω επαγγελματική μου σταδιοδρομία. Τους φίλους και συναδέλφους μου, για την υποστήριξη και τη συνέργεια μας τα τελευταία αυτά 5 χρόνια. Τους ευχαριστώ θερμά για τον ευχάριστο χρόνο, τις χαρές αλλά και τις λύπες που περάσαμε μαζί, και πάνω από όλα που καταφέραμε να είμαστε φίλοι και συνοδοιπόροι όλο αυτό τον καιρό χωρίς διαλλείματα και παρεξηγήσεις.

Τέλος, έχω αφήσει τους σημαντικότερους ανθρώπους όλης μου της ζωής, την οικογένεια μου, χωρίς τους οποίους δεν θα έγραφα αυτές τις σελίδες της παρούσας διπλωματικής εργασίας. Τους ευχαριστώ με όλη μου την καρδιά για την απεριόριστη υποστήριξη, δύναμη αλλά και επιμονή που μου ενέπνευσαν ώστε να φτάσω μέχρι εδώ.

Τον πατέρα μου και την μητέρα μου, Κίμωνα και Μαρία, που με στήριξαν σε κάθε ξεχωριστό βήμα της ζωής μου, που μου έδωσαν την οικονομική αλλά και ψυχολογική υποστήριξη να σπουδάσω και πάνω από όλα στην αγάπη που μου έχουν δείξει καθ’ όλη την διάρκεια της ζωής μου.

Έπειτα τον παππούδες μου, Νικόλαο & Ωριγένη καθώς και τις γιαγιάδες μου, Μαρία & Άννα, με τους οποίους αν και δεν έχω περάσει αρκετές στιγμές, έχουν χαράξει στην μνήμη μου ο καθένας ξεχωριστά την δική του εικόνα & προσωπικότητα.

***Μησσήν Στέφανος,***

***Οκτώβριος 2018***

# Πίνακας περιεχομένων

[Πρόλογος i](#_Toc527386781)

[Ευχαριστίες iii](#_Toc527386782)

[Πίνακας περιεχομένων v](#_Toc527386783)

[Περιεχόμενα Πινάκων και Σχημάτων v](#_Toc527386784)

[Περίληψη viii](#_Toc527386785)

[Abstract ix](#_Toc527386786)

[**1.1** **Το πρόβλημα και η σημασία του** 1](#_Toc527386787)

[**1.2** **Internet of Things (IoT)** 3](#_Toc527386788)

[**1.3** **Στόχοι της διπλωματικής εργασίας** 6](#_Toc527386789)

[**1.4** **Δομή της διπλωματικής εργασίας** 7](#_Toc527386790)

[**2.1** **Γλώσσες Προγραμματισμού Διαδικτυακής Εφαρμογής** 10](#_Toc527386792)

[**2.1.1.** **Γλώσσα Σήμανσης Υπερκειμένου - HTML** 10](#_Toc527386795)

[**2.1.2.** **Γλώσσα Μορφοποίησης – CASCADING STYLES SHEETS (CSS)** 16](#_Toc527386796)

[**2.1.3.** **Γλώσσα Επικοινωνίας Χρήστη – Διακομιστή PHP** 19](#_Toc527386797)

[**2.1.4.** **Γλώσσα Scripting - JavaScript** 23](#_Toc527386798)

[**2.1.5.** **Γλώσσα Σχεσιακών Βάσεων Δεδομένων MySQL** 25](#_Toc527386799)

[**2.2** **Υλικός Εξοπλισμός - Hardware** 27](#_Toc527386800)

[**2.2.1.** **Μικροελεγκτής Arduino uno Rev3** 28](#_Toc527386805)

[**2.2.2.** **Μικροελεγκτής WeMos D1 R2** 33](#_Toc527386806)

[**2.2.3.** **Δρομολογητής - Router** 37](#_Toc527386807)

[**2.3** **Πλατφόρμες Ανάπτυξης Λογισμικού** 38](#_Toc527386808)

[**2.3.1.** **Visual Studio Code** 39](#_Toc527386810)

[**2.3.2.** **Brackets** 41](#_Toc527386811)

[**2.3.3.** **ΧAMPP** 42](#_Toc527386812)

[**2.3.4.** **Πλατφόρμα ανάπτυξης λογισμικού Arduino IDE** 43](#_Toc527386813)

[**3.1** **Hardware Υλοποίησης Συστήματος** 51](#_Toc527386815)

[**3.** 51](#_Toc527386816)

[**3.1.1.** **Arduino Uno Rev3 – Arduino Ethernet Shield** 51](#_Toc527386818)

[**3.1.2.** **WeMos D1 R2** 53](#_Toc527386819)

[**3.1.3.** **Πλακέτα δοκιμών (Breadboard 400)** 54](#_Toc527386820)

[**3.1.4.** **Ακρυλική βάση τοποθέτησης μικροελεγκτή – πλακέτας δοκιμών** 55](#_Toc527386821)

[**3.1.5.** **Αισθητήριο θερμοκρασίας & υγρασίας DHT22/AM2302** 56](#_Toc527386822)

[**3.1.6.** **Καλώδια Jumper** 60](#_Toc527386823)

[**3.1.7.** **Καλώδια σύνδεσης πλακετών με Η/Υ** 61](#_Toc527386824)

[**3.2** **Συνδεσμολογία Συστήματος (Hardware Wiring)** 63](#_Toc527386825)

[**3.3** **Σύνδεση υλικού με τοπικό εξυπηρετητή** 68](#_Toc527386826)

[**3.3.1.** **Προγραμματισμός αισθητηρίων για λήψη δεδομένων** 68](#_Toc527386833)

[**3.3.2.** **Διάγραμμα ροής λειτουργίας αισθητήρα DHT22/Am2302** 73](#_Toc527386834)

[**3.4** **Υπολογισμός κόστους εξαρτημάτων** 75](#_Toc527386835)

[**4.1** **Απαιτήσεις Συστήματος** 77](#_Toc527386838)

[**4.2** **Διαδικτυακή ασφάλεια & Προστασία προσωπικών δεδομένων** 79](#_Toc527386839)

[**4.3** **Δημιουργία και περιγραφή των βάσεων δεδομένων** 80](#_Toc527386840)

[**4.3.1.** **Εργαλείο δημιουργίας βάσεων δεδομένων phpMyAdmin** 80](#_Toc527386845)

[**4.3.2.** **Αρχιτεκτονική βάσης δεδομένων διαδικτυακής εφαρμογής** 82](#_Toc527386846)

[**4.4** **Περιγραφή λειτουργίας διαδικτυακής εφαρμογής Smart Homie** 93](#_Toc527386847)

# Περιεχόμενα Πινάκων και Σχημάτων

Πίνακας 2.1: Βασικές ετικέτες HTML

Πίνακας 2.2: Είδη Arduino

Πίνακας 2.3: Τεχνικά Χαρακτηριστικά WeMos D1 R2

Πίνακας 2.4: Σύνδεση - Λειτουργία των pins του WeMos D1 R2

Πίνακας 2.5: Σύνδεση - Πλακέτες WeMos D1

Πίνακας 2.6: Εργαλεία ανάπτυξης Arduino IDE

Πίνακας 2.7: Βασικές δομές & εντολές - Arduino IDE

Πίνακας 3.1: Τεχνικά Χαρακτηριστικά του DHT22/AM2302

Πίνακας 3.2: Περιγραφή λειτουργίας ακροδεκτών DHT22/AM2302

Σχήμα 1.1 Εκτιμώμενος αριθμός συνδεδεμένων συσκευών σε παγκόσμια κλίμακα έως το 2020

Σχήμα 1.2 Οικοσύστημα του Διαδικτύου των Πραγμάτων (IoT Ecosystem)

Σχήμα 2.1 Η ιστορία εκδόσεων της γλώσσας σήμανσης υπερκειμένου HTML

Σχήμα 2.2 Βασικό κομμάτι κώδικα για την εκκίνηση ενός εγγράφου HTML

Σχήμα 2.3 Κώδικας σύνδεσης ενός αρχείου CSS σε ένα έγγραφο HTML

Σχήμα 2.4 Κομμάτι κώδικα CSS για αλλαγή του χρώματος των παραγράφων σε κόκκινο

Σχήμα 2.5 Ανατομία ενός κώδικα CSS

Σχήμα 2.6 Κώδικας CSS τροποποίησης πολλαπλών ιδιοτήτων ενός στοιχείου

Σχήμα 2.7 Κώδικας CSS τροποποίησης πολλαπλών στοιχείων ταυτόχρονα

Σχήμα 2.8 Τρόπος επιλογής ενός στοιχείου προς μορφοποίηση - CSS

Σχήμα 2.9 Τρόπος σύνταξης της συνάρτησης require\_once στην PHP

Σχήμα 2.10 Τρόπος σύνταξης της PHP

Σχήμα 2.11 Tρόπος σύνταξης της JavaScript

Σχήμα 2.12 Σύνδεση αρχείου JavaScript σε αρχείου HTML/PHP

Σχήμα 2.13 Κομμάτι κώδικα σε JavaScript για την λειτουργικότητα του μενού της διαδικτυακής εφαρμογής Smart Homie

Σχήμα 2.14 Λειτουργία και αλληλεπίδραση της MySQL

Σχήμα 2.15 WeMos D1 R2

Σχήμα 2.16 Δρομολογητής – Μόντεμ TG585 V7

Σχήμα 2.17 Περιβάλλον Προγραμματισμού Visual Studio Code

Σχήμα 2.18 Περιβάλλον Προγραμματισμού Brackets

Σχήμα 2.19 Πίνακας Ελέγχου του XAMPP

Σχήμα 2.20 Λογισμικό Arduino IDE

Σχήμα 3.1 Διάγραμμα ροής συνδεσμολογίας εξαρτημάτων

Σχήμα 3.2 Arduino Uno Rev3 και Ethernet Shield

Σχήμα 3.3 WIZnet Ethernet W5100 Chipset & Schematic Diagram

Σχήμα 3.4 WeMos D1 R2

Σχήμα 3.5 ESP8266 microchip Schematic

Σχήμα 3.6 Breadboard 400 Schematic

Σχήμα 3.7 Ακρυλική Βάση

Σχήμα 3.8 Σχηματική Απεικόνιση & Διαστάσεις του αισθητηρίου DHT22/AM2302

Σχήμα 3.9 Διαδικασία επικοινωνίας DHT22/AM2302 με Arduinο

Σχήμα 3.10 Σχηματική Απεικόνιση ακροδεκτών DHT22/AM2302

Σχήμα 3.11 Καλώδια Jumper

Σχήμα 3.12 Καλώδιo micro USB σε USB

Σχήμα 3.13 Καλώδιo USB Type-B σε USB

Σχήμα 3.14 Καλώδιo Ethernet RJ45

Σχήμα 3.15 Σχηματική απεικόνιση σύνδεσης Arduino Uno Rev3 και DHT22

Σχήμα 3.16 Σύνδεση Arduino Uno Rev3 και Arduino Ethernet Shield

Σχήμα 3.17 Ηλεκτρολογική σχηματική απεικόνιση συνδεσμολογίας

Σχήμα 3.18 Tελική μορφή υποσυστήματος

Σχήμα 3.19 Σχηματική απεικόνιση σύνδεσης WeMos D1 R2 και DHT22

Σχήμα 3.20 Ηλεκτρολογική σχηματική απεικόνιση συνδεσμολογίας δεύτερου υποσυστήματος

Σχήμα 3.21 Tελική μορφή δεύτερου υποσυστήματος

Σχήμα 3.22 Διάγραμμα ροής λειτουργίας αισθητήρα DHT22/AM2302

# Περίληψη

Κείμενο περίληψης

# Abstract

Κείμενο abstract

Κεφάλαιο 1

**Εισαγωγή**

* 1. **Το πρόβλημα και η σημασία του**

Οι έννοιες των έξυπνων συσκευών (Smart Devices), των έξυπνων σπιτιών (Smart Homes), των έξυπνων συμπλεγμάτων (Smart Clusters), αλλά και του Διαδικτύου των πραγμάτων (Internet of Things – IoT), είναι ευρέως γνωστές και αποτελούν σημαντικό τμήμα στο οποίο εστιάζει η τεχνολογική έρευνα. Καθημερινά ολοένα και περισσότερες έξυπνες συσκευές εγκαθίστανται σε σπίτια, σε επιχειρήσεις, σε οργανισμούς ακόμα και σε δημόσια μέρη. Για την εύρυθμη λειτουργία των συσκευών αυτών, απαιτείται σωστή διαχείριση των δεδομένων , η συλλογή των δεδομένων αυτών σε καλά οργανωμένες βάσεις δεδομένων καθώς και η σωστή γραφική απεικόνιση τους είτε μέσω κάποιο προσαρμοσμένο λογισμικό, είτε μέσω κάποιας διαδικτυακής εφαρμογής.

Ως διαδικτυακή εφαρμογή ορίζεται μια εφαρμογή πελάτη/εξυπηρετητή (client/server) που χρησιμοποιεί ένα φυλλομετρητή (browser). Οι εφαρμογές αυτές συνήθως εκτελούνται σε ισχυρές υπολογιστικές μηχανές οι οποίες έχουν τον ρόλο του σταθμού εξυπηρέτησης και παρέχουν τις υπηρεσίες τους σε περισσότερους του ενός χρήστη [1]. Προβάλλουν δυναμικό περιεχόμενο στηριγμένο στις απαιτήσεις του χρήστη. Μια τέτοιου τύπου εφαρμογή συνήθως δεν έχει στατικά στοιχεία προς προβολή, παραδείγματος χάρη μία στατική ιστοσελίδα, αλλά παρέχει δυναμικά στοιχεία προς προβολή και στις περισσότερες περιπτώσεις υπάρχει και η δυνατότητα αλληλεπίδρασης του χρήστη ώστε να επιτυγχάνεται η καλύτερη εμπειρία χρήσης (user experience). Έτσι, οι άνθρωποι για την καθημερινή τους διευκόλυνση καθώς και για την εξοικονόμηση ενέργειας που καταναλώνουν καθημερινά, έχουν αρχίσει να ενσωματώνουν τις έξυπνες συσκευές στους χώρους που περνάνε μεγάλο μέρος της ημέρας όπως είναι τα σπίτια τους. Τα σημαντικότερα ζητήματα ενέργειας που έρχονται να καλύψουν οι έξυπνες συσκευές είναι θέματα που αφορούν την εξοικονόμηση ρεύματος από ηλεκτρικές συσκευές του σπιτιού, που χρησιμοποιούνται για πολλές ώρες ανά την ημέρα όπως τα κλιματιστικά και τα καλοριφέρ, ακόμη για την θέρμανση του νερού όπου χρησιμοποιείται στο μπάνιο, στο πλύσιμο των πιάτων καθώς και στο πότισμα των εξωτερικών χώρων. Επίσης, χρησιμοποιούνται αισθητήρες μέτρησης της ποιότητας του αέρα, έτσι ώστε να μην υπάρχει μεγάλο ποσοστό υγρασίας & μόλυνσης του αέρα που αναπνέουμε στο σπίτι.

Όλες αυτές οι μετρήσεις αλλά και η σωστή απεικόνιση αυτών στον τελικό χρήστη ενθαρρύνουν σημαντικά τη δημιουργία μίας διαδικτυακής εφαρμογής η οποία θα είναι εύχρηστη, θα μπορεί να παρέχει τις απαραίτητες πληροφορίες που ζητάει ο χρήστης, στο σωστό χρόνο και χωρίς να υπάρχουν προβλήματα στην παραμετροποίηση και τροποποίηση αυτών των δεδομένων για την καλύτερη εμπειρία χρήσης.

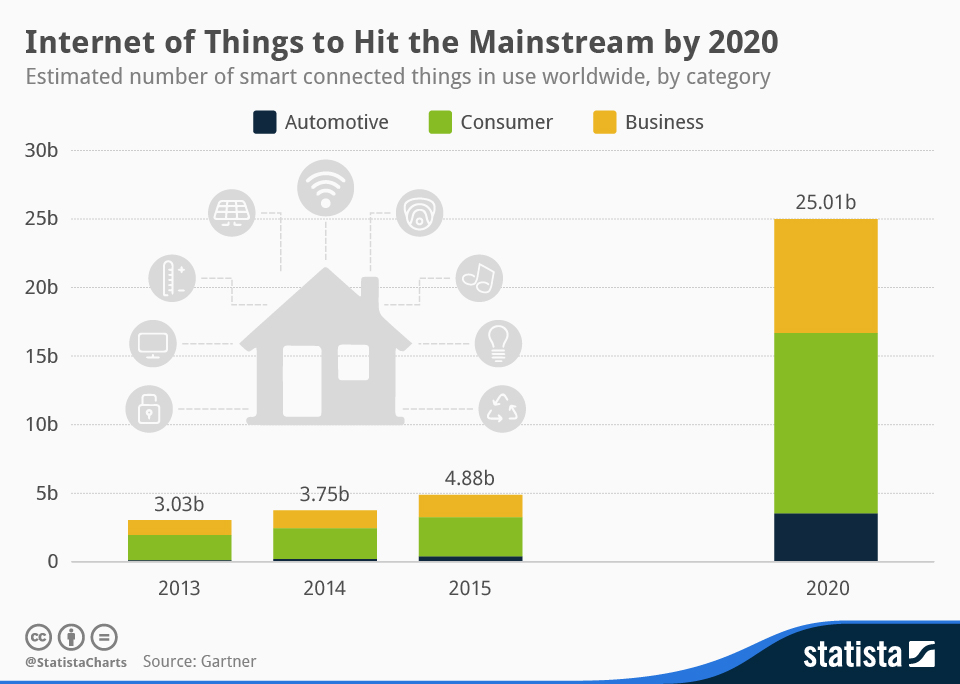
Σκοπός της παρούσας διπλωματικής εργασίας είναι η δημιουργία μιας διαδικτυακής εφαρμογής για την ανάπτυξη ενός συστήματος παρακολούθησης & διαχείρισης έξυπνων συσκευών με σκοπό την μετατροπή ενός συμβατικού σπιτιού σε έξυπνο. Λαμβάνονται ως δεδομένα, πρωτογενείς μετρήσεις θερμοκρασίας και υγρασίας εντός και εκτός του σπιτιού που χρησιμοποιήθηκε για την δημιουργία της παρούσας διπλωματικής εργασίας. Μέσα από την διαδικτυακή εφαρμογή, ο χρήστης μπορεί να παρακολουθεί σε ζωντανό χρόνο τις μετρήσεις που αφορούν το σπίτι του, να λαμβάνει στατιστικά στοιχεία των τελευταίων μηνών καθώς και να παίρνει αποφάσεις για την ενεργοποίηση/απενεργοποίηση έξυπνων συσκευών που έχουν τοποθετηθεί στο σπίτι.

* 1. **Internet of Things (IoT)**

Το διαδίκτυο των πραγμάτων είναι ένα δίκτυο από φυσικές συσκευές, κινούμενα οχήματα, σπίτια και άλλα ενσωματωμένα συστήματα με ηλεκτρονικά, λογισμικό, αισθητήρια, ενεργοποιητές, τα οποία μπορούν να συνδεθούν μεταξύ τους, να συλλέξουν αλλά και να ανταλλάξουν δεδομένα. Με το IoT τα αντικείμενα αυτά μπορούν να ελέγχονται απομακρυσμένα μέσω της υπάρχουσας δικτυακής υποδομής δημιουργώντας ευκαιρίες άμεσης ενσωμάτωσης του φυσικού κόσμου με τα υπολογιστικά συστήματα. Λόγω του εξοπλισμού με αισθητήρες και ενεργοποιητές το IoT είναι μέρος των έξυπνων συστημάτων της καθημερινότητας όπως είναι τα έξυπνα σπίτια και πόλεις. Λόγω του ενσωματωμένου υπολογιστικού συστήματος κάθε αντικείμενο αναγνωρίζεται μοναδικά και μπορεί να λειτουργεί αυτόνομα αλλά και σε συνεργασία με την υπόλοιπη Διαδικτυακή υποδομή [2][3].

Ο όρος “Διαδίκτυο των Πραγμάτων” η αλλιώς Internet of Things (IoT) χρησιμοποιήθηκε πρώτη φορά το 1999 από τον Βρετανό πρωτοπόρο στην τεχνολογία Kevin Ashton, για την περιγραφή ενός συστήματος στο οποίο, τα αντικείμενα του φυσικού κόσμου θα μπορούσαν να συνδεθούν με το Διαδίκτυο μέσω αισθητήρων. Σήμερα το IoT έχει γίνει ένας δημοφιλής όρος για την περιγραφή σεναρίων στα οποία η σύνδεση στο Διαδίκτυο και οι δυνατότητες των υπολογιστών επεκτείνονται σε μια ποικιλία από αντικείμενα όπως συσκευές, αισθητήρες και ήδη καθημερινής χρήσης. Παρά το γεγονός ότι, ο όρος IoT είναι σχετικά νέος, η έννοια του συνδυασμού υπολογιστών και των δικτύων για την παρακολούθηση και των έλεγχο των συσκευών υπάρχει εδώ και δεκαετίες [4].

Υπολογίζεται σύμφωνα με την Statista, η ανάπτυξη του IoT έχει ξεκινήσει αρχίσει να αναπτύσσεται ραγδαία και θα φτάσει σε πολύ μεγάλα χρηματικά ποσά μέχρι το 2020, σύμφωνα με το οποίο και οι επιχειρήσεις αλλά και οι καταναλωτές ενσωματώνουν τις έξυπνες τεχνολογίες σε μεγάλη κλίμακα [5].



Σχήμα 1.1 Εκτιμώμενος αριθμός συνδεδεμένων συσκευών σε παγκόσμια κλίμακα έως το 2020

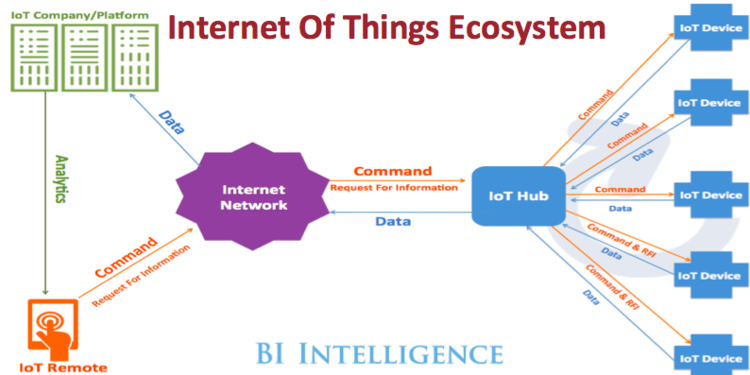
Περαιτέρω, το IoT βοηθάει στην σύνδεση αντικειμένων ή διαδικασιών οι οποίες επιτρέπουν την καλύτερη παρακολούθηση και αλληλεπίδραση. Υπάρχουν τέσσερα στρώματα της IoT αρχιτεκτονικής:

* Το στρώμα ανίχνευσης ή αντίληψης, το οποίο συλλέγει και μεταδίδει δεδομένα με την βοήθεια ιδιοτήτων εντοπισμού αντικειμένων και εργαλείων ανάγνωσης.
* Το στρώμα δικτύου και πύλης, το οποίο συνδέει αντικείμενα ή πράγματα και τους επιτρέπει να κοινοποιούν και να ανταλλάσσουν πληροφορίες με την βοήθεια ειδικών πυλών, όπως ένα εσωτερικό δίκτυο ή ένα τοπικό δίκτυο και ένα εξωτερικό δίκτυο ή ένα δίκτυο ευρείας περιοχής.
* Το στρώμα διαχείρισης υπηρεσιών, το οποίο είναι υπεύθυνο για τις αναλύσεις των πληροφοριών, τον έλεγχο της ασφάλειας, τη μοντελοποίηση της διαδικασίας και τη διαχείριση των συσκευών.
* Το στρώμα εφαρμογών, το οποίο διατηρεί τα δεδομένα που συλλέγονται και μεταφέρονται, επεξεργάζονται μέσω συγκεκριμένων τεχνικών που χρησιμοποιούνται για ειδικά σχέδια. Επιπλέον, αφορά τη διαχείριση αντικειμένων και πραγμάτων [6].

*Παράδειγμα IoT Εφαρμογής: Το έξυπνο σπίτι:*

Τα μελλοντικά έξυπνα σπίτια θα έχουν επίγνωση του τι συμβαίνει μέσα σε ένα κτίριο επηρεάζοντας κυρίως τρεις τομείς:

* Την κατανάλωση πόρων, που αφορούν τη διατήρηση και κατανάλωση ενέργειας
* Την Ασφάλεια
* Την Άνεση

Στόχος αυτών, είναι η επίτευξη των καλύτερων επιπέδων άνεσης μειώνοντας ταυτόχρονα τις συνολικές δαπάνες. Επιπλέον, τα έξυπνα σπίτια αντιμετωπίζουν επίσης ζητήματα ασφάλειας μέσω σύνθετων συστημάτων ασφαλείας για την ανίχνευση κλοπής, πυρκαγιάς ή μη εξουσιοδοτημένης εισόδου. Οι φορείς που εμπλέκονται σε αυτό το σενάριο αποτελούν μια πολύ ετερογενή ομάδα. Διαφορετικοί φορείς θα συνεργαστούν στο σπίτι του χρήστη, όπως εταιρείες του διαδικτύου, κατασκευαστές συσκευών, τηλεπικοινωνιακοί φορείς, υπηρεσίες οπτικοακουστικών μέσων προμηθευτές, εταιρείες ασφάλειας, εταιρείες ηλεκτρισμού [3].

Σχήμα 1.2 Οικοσύστημα του Διαδικτύου των Πραγμάτων (IoT Ecosystem)

* 1. **Στόχοι της διπλωματικής εργασίας**

Ο στόχος της παρούσας διπλωματικής εργασίας είναι η παρακολούθηση των εσωτερικών αλλά και εξωτερικών θερμοκρασιών ενός σπιτιού και η δημιουργία μιας διαδικτυακής εφαρμογής έτσι ώστε ο χρήστης να μπορεί να βγάλει συμπεράσματα ανάλογα με τα δεδομένα που συλλέγονται από τα αισθητήρια όργανα, πότε είναι η κατάλληλη στιγμή να ενεργοποιήσει ή να απενεργοποιήσει τις έξυπνες συσκευές που υπάρχουν στο σπίτι του έτσι ώστε να πετύχει την μέγιστη ωφέλιμη ενεργειακή κατανάλωση. Μέσω της εφαρμογής ο χρήστης θα είναι σε θέση να παρακολουθήσει στατιστικά των μετρήσεων για τον τελευταίο μήνα καθώς και να παρακολουθεί τις ελάχιστες και τις μέγιστες τιμές ανά ημέρα.

Για την εκπλήρωση αυτού του στόχου είναι απαραίτητες δύο κύριες δραστηριότητες.

Η πρώτη είναι η αγορά των αισθητηρίων, των μικροελεγκτών καθώς και ενός Η/Υ ο οποίος θα χρησιμοποιηθεί ως τοπικός εξυπηρετητής (Local Host Server). Για την μείωση του όγκου αλλά και λόγω των περιορισμένων απαιτήσεων που χρειάζεται η διαδικτυακή εφαρμογή μπορεί να γίνει αντικατάσταση του Η/Υ με ένα μικρότερο μονής μονάδας υπολογιστή όπως είναι το Raspberry Pi.

Η δεύτερη δραστηριότητα αφορά την δημιουργία της διαδικτυακής εφαρμογής. Σε αυτή τη φάση, εφόσον έχουν ελεγχθεί όλα τα κομμάτια του υλικού (Hardware) που απαιτούνται και πληρούν τις προτεινόμενες απαιτήσεις συστήματος πρέπει να ξεκινήσει η ανάπτυξη της εφαρμογής. Αναπτύσσονται οι απαραίτητοι αλγόριθμοι συλλογής των δεδομένων από τα αισθητήρια και αποθηκεύονται στις βάσεις δεδομένων. Έπειτα γίνεται η γραφική αναπαράσταση των δεδομένων προς τον χρήστη και έτσι του δίνεται η δυνατότητα επιλογής του τι θέλει να κάνει με τις έξυπνες συσκευές στο σπίτι του.

Τέλος σκοπός είναι να δοθεί μία απάντηση στο ερώτημα εάν μία διαδικτυακή εφαρμογή όπως η Smart Homie κρίνεται αναγκαία για την παρακολούθηση τέτοιων συστημάτων όπως είναι οι έξυπνες συσκευές μέσα στο σπίτι του καθημερινού χρήστη. Ακόμη πρέπει να εξακριβωθούν οι συνθήκες στις οποίες ενδείκνυται μία τέτοια διαδικτυακή εφαρμογή και κατά πόσο εναρμονίζεται με τις υπόλοιπες συσκευές του σπιτιού χωρίς να δημιουργεί πρόβλημα στην χρήση του σπιτιού.

* 1. **Δομή της διπλωματικής εργασίας**

Η δομή της διπλωματικής εργασίας είναι η εξής:

Στο Κεφάλαιο 1 παρουσιάζεται το πρόβλημα που εξετάζει η διπλωματική εργασία καθώς και πληροφορίες για τον τρόπο λειτουργίας της διαδικτυακής εφαρμογής καθώς και για τα ζητήματα που είναι απαραίτητα να λυθούν πριν την εγκατάσταση και χρήση αυτής.

Στο Κεφάλαιο 2 παρουσιάζονται οι βασικές έννοιες καθώς και το υλικό μέρος (Hardware) που θα χρησιμοποιηθεί για την παρούσα διπλωματική εργασία. Πιο συγκεκριμένα, αναλύονται οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για να ανάπτυξη και κωδικοποίηση της διαδικτυακής εφαρμογής, οι μικροελεγκτές και ο δρομολογητής-μόντεμ που χρησιμοποιήθηκαν για την δημιουργία του τοπικού συστήματος & δικτύου και τέλος αναλύονται οι πλατφόρμες πάνω στις οποίες αναπτύχθηκε η διαδικτυακή εφαρμογή.

Στο Κεφάλαιο 3 αναλύονται εκτενώς το υλικό μέρος καθώς και η συνδεσμολογία τους συστήματος. Παραθέτονται τα σχεδιαγράμματα του κάθε μικροελεγκτή, των αισθητηρίων καθώς και οι βάσεις τοποθέτησης των παραπάνω υλικών για την εύρυθμη λειτουργία του συστήματος. Ακόμη, αναλύεται ο τρόπος σύνδεσης όλων των παραπάνω υλικών έτσι ώστε να είναι σε θέση να δεχθούν την κατάλληλη παραμετροποίηση και τους κατάλληλους αλγορίθμους ώστε να λειτουργήσουν, να συλλέξουν αλλά και να αποστείλουν δεδομένα. Έπειτα, γίνεται η σύνδεση με τον τοπικό εξυπηρετητή. Τέλος αναλύεται το κόστος των εξαρτημάτων που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία.

Στο Κεφάλαιο 4 γίνεται ανάλυση της σχεδίασης και της υλοποίησης τη διαδικτυακής εφαρμογής. Αρχικά, εξετάζονται οι απαιτήσεις του συστήματος και γίνεται λόγος για την τοπική λειτουργία της εφαρμογής & προστασία των προσωπικών δεδομένων του χρήστη. Έπειτα γίνεται περιγραφή του τρόπου δημιουργίας της βάσης δεδομένων, της αρχιτεκτονικής της, καθώς και ο τρόπος επεξεργασίας της από την διαδικτυακή εφαρμογή. Επιπλέον, γίνεται μία λεπτομερή ανάλυση για καθεμία από τις καρτέλες που απαρτίζουν την εφαρμογή.

Στο κεφάλαιο 5 γίνεται μία ανασκόπηση της λειτουργικότητας της διαδικτυακής εφαρμογής, εξαγωγή και ανάλυση των συμπερασμάτων που προέκυψαν κατά την διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας. Αναφέρονται τα προβλήματα που αντιμετωπίστηκαν όπως και προτάσεις για μελλοντική έρευνα.

Κεφάλαιο 2

**Θεωρητικό Υπόβαθρο εννοιών και εξοπλισμού**

Η παρούσα διπλωματική εργασία, στην προσπάθεια της να δημιουργήσει μία διαδικτυακή εφαρμογή η οποία θα προσφέρει την καλύτερη δυνατή εμπειρία χρήσης για την διαχείριση έξυπνων συσκευών, είναι απαραίτητος ο διαχωρισμός της σε δύο μέρη καθώς και η εναρμόνιση αυτών.

Αρχικά, το υλικό μέρος (Hardware), το οποίο περιλαμβάνει τα αισθητήρια όργανα καθώς και την συνδεσμολογία αυτών για την λήψη των απαραίτητων δεδομένων. Το δεύτερο μέρος, αποτελείται από την κωδικοποίηση/παραμετροποίηση των αισθητηρίων καθώς και την ανάπτυξη του λογισμικού (Software) με την βοήθεια του οποίου συμπληρώνεται η διαδικτυακή εφαρμογή. Για την ευκολότερη κατανόηση του τρόπου σύνδεσης των δύο αυτών μερών, στο κεφάλαιο αυτό παρουσιάζονται βασικές θεωρητικές γνώσεις και έννοιες που θα χρησιμοποιηθούν στη συνέχεια. Γίνεται εκτενής αναφορά στις γλώσσες προγραμματισμού που θα χρησιμοποιηθούν, στον υλικό εξοπλισμό που είναι απαραίτητος καθώς και στις πλατφόρμες στις οποίες αναπτύχθηκε η εφαρμογή, και στις υπηρεσίες που παρέχουν για την διασφάλιση της ποιότητας της εφαρμογής.

1. 1. **Γλώσσες Προγραμματισμού Διαδικτυακής Εφαρμογής**

Μια γλώσσα προγραμματισμού είναι μια επίσημη γλώσσα, η οποία περιλαμβάνει ένα σύνολο οδηγιών που χρησιμοποιούνται για την παραγωγή διαφόρων ειδών εξόδου. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για τη δημιουργία προγραμμάτων που υλοποιούν συγκεκριμένους αλγόριθμους.

Οι περισσότερες γλώσσες προγραμματισμού αποτελούνται από οδηγίες για υπολογιστές, παρόλο που υπάρχουν προγραμματιζόμενες μηχανές που χρησιμοποιούν περιορισμένο σύνολο συγκεκριμένων οδηγιών αντί για τις γενικές γλώσσες προγραμματισμού των σύγχρονων υπολογιστών [7].

2. * 1. **Γλώσσα Σήμανσης Υπερκειμένου - HTML**

Η HTML (αρχικοποίηση του αγγλικού όρου HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων, τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ. Ο σκοπός ενός φυλλομετρητή είναι να διαβάζει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο φυλλομετρητής δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να παρουσιάσει το περιεχόμενο της σελίδας.

Στις μέρες μας το πρότυπο που χρησιμοποιείται για την HTML είναι το HTML5.2 το οποίο ανακοινώθηκε από την κοινοπραξία World Wide Web (W3C) το 2017. Παρακάτω παρατίθεται η ιστορία της HTML.



Σχήμα 2.1 Η ιστορία εκδόσεων της γλώσσας σήμανσης υπερκειμένου HTML

Η HTML κυρίως παράγεται ως έγγραφο. Μπορεί οι προγραμματιστές να κάνουν λόγο για κομμάτια κώδικα HTML, αλλά στις περισσότερες περιπτώσεις, μία ιστοσελίδα θα απαντήσει σε ένα αίτημα που έγινε από τον διακομιστή ή τον χρήστη μία ένα έγγραφο HTML, το οποίο μετά παραδίδεται στον χρήστη και εμφανίζεται σε μία συσκευή [8].

Παρακάτω εμφανίζεται ένα παράδειγμα κώδικα (HTML snippet) για το βασικό κώδικα εκκίνησης της HTML.



Σχήμα 2.2 Βασικό κομμάτι κώδικα για την εκκίνηση ενός εγγράφου HTML

Στην προσπάθεια που γίνεται για την σωστή ανάπτυξη του κώδικα και την εύρυθμη λειτουργία των HTML εγγράφων, είναι απαραίτητο να χρησιμοποιούνται οι κατάλληλες ετικέτες (tags) ανάλογα με τον τρόπο που θέλει ο προγραμματιστής να επεξεργασθεί τα δεδομένα που είναι προς εμφάνιση στον τελικό χρήστη. Οι σημαντικότερες ετικέτες html που χρησιμοποιούνται για την δημιουργία των HTML εγγράφων την παρούσας διπλωματικής εργασίας, παρουσιάζονται στον παρακάτω πίνακα.

***Πίνακας 2.1: Βασικές ετικέτες HTML***

|  |  |
| --- | --- |
| **Στοιχείο (tag)** | **Λειτουργία** |
| <!--...--> | Η ετικέτα <!--...--> αφορά τα σχόλια τα οποία γράφει ο προγραμματιστής. Με τα σχόλια μπορεί να γίνει ευκολότερη η δόμηση ενός εγγράφου σε μικρότερες υποενότητες καθώς και γίνει επεξήγηση ενός κομμάτι κώδικα ή και να τοποθετηθούν σημειώσεις για την μετέπειτα επεξεργασία του εγγράφου. |
| <!DOCTYPE> | Η δήλωση του DOCTYPE πρέπει να είναι η πρώτη σε ένα έγγραφο HTML. Είναι μια εντολή στο πρόγραμμα περιήγησης για το ποια είναι η έκδοση της HTML στο έγγραφο. |
| <a> | Η ετικέτα <a> καθορίζει έναν υπερσύνδεσμο, ο οποίος χρησιμοποιείται για την σύνδεση μίας σελίδας προς μία άλλη. Τo πιο σημαντικό στοιχείο αυτής της ετικέτας είναι το *href* ,το οποίο υποδεικνύει τον προορισμό του συνδέσμου |
| <article> | Η ετικέτα <article> καθορίζει ένα ανεξάρτητο, αυτό-συντηρούμενο περιεχόμενο. |
| <aside> | Η ετικέτα <aside> καθορίζει περιεχόμενο , πέρα από το ήδη υπάρχων περιεχόμενο των άλλων ετικετών. Το περιεχόμενο αυτό πρέπει να συσχετίζεται με τα υπόλοιπα περιεχόμενα. |
| <blockquote> | Η ετικέτα <blockquote> καθορίζει μία περιοχή η οποία αναφέρθηκε από μία άλλη πηγή. Οι φυλλομετρητές συνήθως δημιουργούν μία εσοχή όταν βρίσκουν την ετικέτα αυτή. Είναι σύνηθες φαινόμενο η συγκεκριμένη ετικέτα να χρησιμοποιείται για την παρουσίαση κομματιών κώδικα. |
| <body> | Η ετικέτα <body> ορίζει το σώμα του εγγράφου. Όλα τα περιεχόμενα ενός εγγράφου HTML, όπως κείμενο, υπερσύνδεσμοι, εικόνες, πίνακες, λίστες, κτλ. Βρίσκονται μέσα σε αυτό. |
| <button> | Η ετικέτα <button> καθορίζει ένα κουμπί το οποίο μπορεί να πατηθεί. Μέσα στην ετικέτα, μπορούμε να προσθέσουμε περιεχόμενο, όπως κείμενο ή φωτογραφίες. Αυτή είναι η διαφορά μεταξύ των κουμπιών που δημιουργούνται με την ετικέτα <button> και με την ετικέτα <input>. |
| <canvas> | Η ετικέτα <canvas> χρησιμοποιείται για την δημιουργία δυναμικών γραφικών (on the fly), μέσω του scripting (συνήθως με την γλώσσα JavaScript) |
| <div> | Η ετικέτα <div> καθορίζει ένα τμήμα σε ένα έγγραφο HTML. Χρησιμοποιείται για την ομαδοποίηση των στοιχείων, ώστε να δοθούν κοινές μορφοποιήσεις με τη χρήση του Id. |
| <footer> | Η ετικέτα <footer> καθορίζει το υποσέλιδο για ένα αντικείμενο ή μία περιοχή. Η ετικέτα, θα πρέπει να περιέχει να περιέχει πληροφορίες για τα στοιχεία που περιβάλει. |
| <form> | Η ετικέτα <form> χρησιμοποιείται για τη δημιουργία φορμών, με τις οποίες αλληλοεπιδρούν οι χρήστες. Χρησιμοποιείται επίσης για να μεταβιβάζει δεδομένα σε ένα διακομιστή. |
| <h1> έως <h6> | Οι ετικέτες <h1> έως <h6> χρησιμοποιούνται για τον καθορισμό του μεγέθους των γραμμάτων για τις επικεφαλίδες. |
| <head> | Η ετικέτα <head> είναι βασική, καθώς στο πεδίο της ορίζονται δεδομένα όπως το είδος των χαρακτήρων που χρησιμοποιούνται (UTF-8, ANSI κτλ), εξωτερικές πηγές αρχείων κτλ. |
| <header> | Η ετικέτα <header> αναπαριστά μία περιοχή για εισαγωγικά περιεχόμενα ή ένα μενού. |
| <input> | Η ετικέτα <input> χρησιμοποιείται για την εισαγωγή πληροφοριών από τον χρήστη. Οι ετικέτες αυτές, περιλαμβάνονται σε μια <form>. Το χαρακτηριστικό type δηλώνει τη μορφή της ετικέτας input (type=text/ password/ button/ radio/ checkbox/ number/ email/ date). |
| <script> | Η ετικέτα script χρησιμοποιείται για να καθορίσει ένα script στην πλευρά του χρήστη. Η ετικέτα περιέχει, είτε περιέχει κομμάτια κώδικα, είτε συνδέει το παρών έγγραφο με ένα εξωτερικό αρχείο script μέσω του στοιχείου *src.* Η βασική γλώσσα scripting είναι ηJavaScript. |
| <table> | Η ετικέτα <table> ορίζει έναν πίνακα HTML. Ο πίνακας αποτελείται από ένα ή περισσότερα <tr>, <td> ή <th>. |

* + 1. **Γλώσσα Μορφοποίησης – CASCADING STYLES SHEETS (CSS)**

Η γλώσσα μορφοποίησης CSS (Cascading Style Sheets - Διαδοχικά Φύλλα Ύφους) ή (αλληλουχία φύλλων ύφους) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης [9]. Σύμφωνα με τα πρότυπα αυτής της γλώσσας γίνονται οι απαραίτητες μορφοποιήσεις στις ετικέτες html του εγγράφου έτσι ώστε να παραχθεί το επιθυμητό για τον χρήστη αποτέλεσμα στον φυλλομετρητή. Στις μέρες μας δεν νοείται η δημιουργία ιστοσελίδων ή διαδικτυακών εφαρμογών χωρίς την γλώσσα CSS καθώς οι δυνατότητες που προσφέρει αυτή η γλώσσα παράγουν ένα πολύ όμορφο αποτέλεσμα.

H CSS είναι προορισμένη να αναπτύσσει τη διαμόρφωση (στυλ) μιας ιστοσελίδας, δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και να δίνει περισσότερες δυνατότητες σε σχέση με την HTML και δημιουργήθηκε για να λύσει ένα μεγάλο πρόβλημα που παρουσίαζε η HTML.

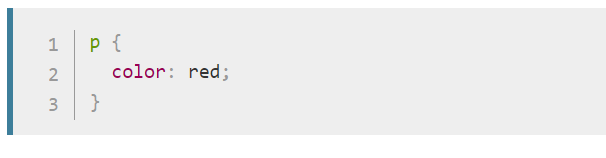
Σύνταξη της CSS

Η σύνταξη της CSS μπορεί να ξεκινήσει από πολύ απλούς κανόνες αλλά και να γίνει τόσο περίπλοκη που μπορεί να μην μπορεί εύκολα να κατανοηθεί από έναν αρχάριο. Η σύνταξη των κανόνων μορφοποίησης των ετικετών μπορεί να διαφέρουν από ένα έγγραφο σε ένα άλλο, για αυτό τον λόγο δημιουργούνται περισσότερα από ένα αρχεία με την κατάληξη .css. Για να το πετύχουμε αυτό, αρκεί να εισάγουμε στο έγγραφο HTML, ανάμεσα στις ετικέτες <head> και </head> την απαραίτητη σύνδεση του αρχείο .css που έχουμε δημιουργήσει. Πρέπει να αναφέρουμε πώς τηρείται η σειρά προτεραιότητας με την οποία συνδέουμε στο έγγραφο HTML μας τα εξωτερικά αρχεία CSS, με αυτό να επιφέρει αλλαγές στον τρόπο με τον οποίο επηρεάζονται οι ετικέτες. Η σύνδεση φαίνεται στην παρακάτω εικόνα.

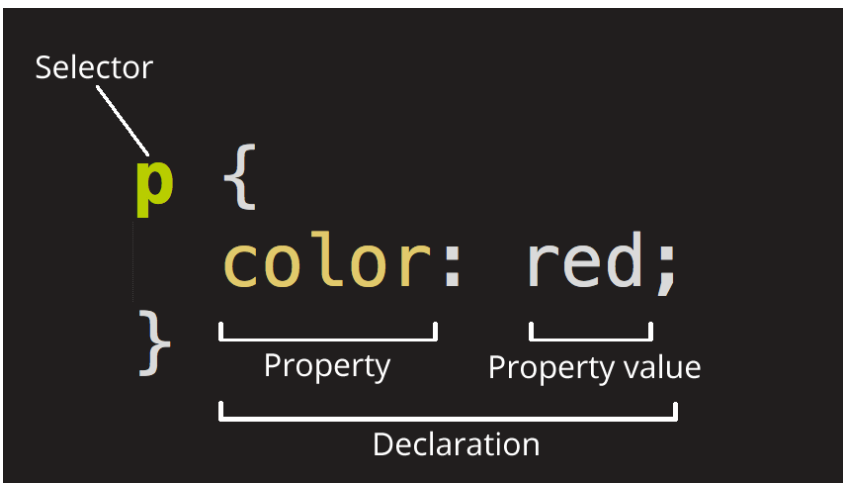


Σχήμα 2.3 Κώδικας σύνδεσης ενός αρχείου CSS σε ένα έγγραφο HTML

Για παράδειγμα, για να επιλέξουμε όλα τα στοιχεία παραγράφου σε μια σελίδα HTML και να μετατρέψουμε το κείμενο μέσα σε αυτά κόκκινο, θα γράψουμε αυτό το κομμάτι κώδικα CSS:



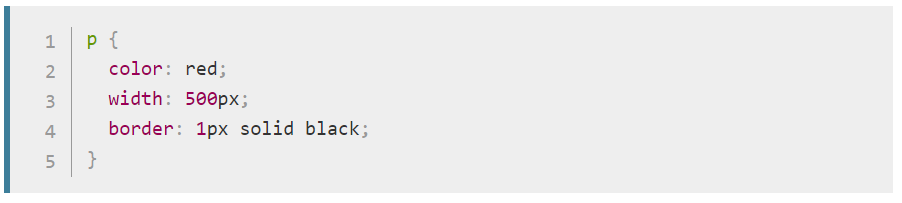
Σχήμα 2.4 Κομμάτι κώδικα CSS για αλλαγή του χρώματος των παραγράφων σε κόκκινο

Κρίνεται αναγκαίο στο συγκεκριμένο σημείο, να κάνουμε λόγο για την ανατομία ενός κανόνα CSS. Κάθε ετικέτα HTML την οποία θέλουμε να επεξεργασθούμε και να διαμορφώσουμε το στιλ της με τέτοιο τρόπο, ώστε να παραχθεί το επιθυμητό αποτέλεσμα πρέπει να γράφει με τέτοιο τρόπο, όπως φαίνεται στην παρακάτω εικόνα.

Σχήμα 2.5 Ανατομία ενός κώδικα CSS

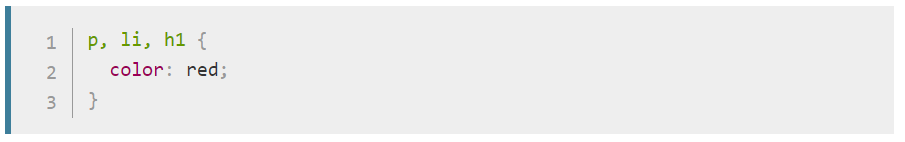
Ο κανόνας χωρίζεται σε κομμάτια για την καλύτερη ανάγνωση του από τους ανθρώπους αλλά και για την εύρυθμη λειτουργία μίας ιστοσελίδας αλλά και μίας διαδικτυακής εφαρμογής. Ο διαχωρισμός γίνεται ως εξής:

* **Selector:** Ο selector είναι το όνομα του στοιχείου HTML και βρίσκεται στην αρχή του σετ κανόνων. Είναι υπεύθυνος για την επιλογή των στοιχείων των οποίων θα διαμορφωθούν με το κατάλληλο στιλ (στην περίπτωση μας, τα στοιχεία <p>). Για να αποδώσουμε στιλ σε άλλο στοιχείο, αρκεί να αλλάξουμε τον selector.
* **Declaration:** Ένας απλός κανόνας όπως color: red; καθορίζει σε ποιες ιδιότητες θα αποδώσουμε στιλ, από το στοιχείο που επιλέξαμε.
* **Properties:** Είναι οι τρόποι με τους οποίους μπορούμε να δώσουμε στιλ στα HTML στοιχεία. Στην CSS, μπορούμε να διαλέξουμε ποιες ιδιότητες θα επηρεαστούν από τον κανόνα που θέτουμε.
* **Property Value:** Στα δεξιά της ιδιότητας και μετά τα εισαγωγικά, έχουμε την τιμή της ιδιότητας, η οποία μας δίνει την επιθυμητή εμφάνιση.

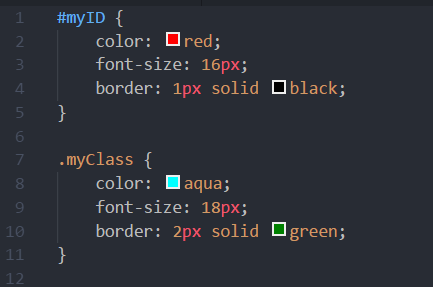
Ακόμη, για να τροποποιήσουμε πολλαπλές τιμές με μία προσπάθεια, απλά χρειάζεται να τις γράψουμε, την μία κάτω από την άλλη και χωρισμένες με ερωτηματικό, όπως φαίνεται παρακάτω.

Σχήμα 2.6 Κώδικας CSS τροποποίησης πολλαπλών ιδιοτήτων ενός στοιχείου

Μπορούμε επίσης να επιλέξουμε πολλαπλούς τύπους στοιχείων και να τους αποδώσουμε έναν κανόνα συνολικά. Μπορούμε να εισάγουμε πολλαπλούς selectors χωρισμένους με κόμμα όπως φαίνεται στο σχήμα 2.7 [10] .



Σχήμα 2.7 Κώδικας CSS τροποποίησης πολλαπλών στοιχείων ταυτόχρονα

Τέλος, για την καλύτερη παραμετροποίηση αλλά και των διαχωρισμό των στοιχείων HTML, η CSS έχει την δυνατότητα να προσδίδει στα στοιχεία ένα διαχωρισμό στον τρόπο που εφαρμόζει τους κανόνες μορφοποίησης. Υπάρχουν δύο τρόποι με τους οποίους μπορούμε να διαλέξουμε ένα στοιχείο, είτε προσδίδοντας του ένα ID είτε μία κλάση. Η επιλογή με ID, χρησιμοποιείται για να μορφοποιηθεί ένα συγκεκριμένο και μοναδικό στοιχείο html. Αυτή η επιλογή χρησιμοποιεί την ιδιότητα id του στοιχείου και ορίζεται με ένα #. Η απόδοση κλάσης, χρησιμοποιείται για να μορφοποιήσει μια ομάδα από στοιχεία της HTML. Αυτό επιτρέπει την χρήση της ίδιας μορφοποίησης σε πολλά στοιχεία ταυτόχρονα που έχουν την ίδια ιδιότητα class. Γι’ αυτή την επιλογή στην CSS χρησιμοποιείται το σύμβολο “.”. Στο παρακάτω σχήμα παρουσιάζεται ο τρόπος επιλογής ενός στοιχείου προς μορφοποίηση.

Σχήμα 2.8 Tρόπος επιλογής ενός στοιχείου προς μορφοποίηση - CSS

* + 1. **Γλώσσα Επικοινωνίας Χρήστη – Διακομιστή PHP**

Η γλώσσα PHP το οποίο είναι ακρολέξιο του Personal Home Page: Hypertext Preprocessor, αποτελεί μία scripting [βασισμένη σε scripts (σενάρια)] γλώσσα υπολογιστή - εξυπηρετητή. Είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργασθεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script [11]. Η γλώσσα αυτή εκτελείται από την πλευρά του διακομιστή, όπως η ASP της Microsoft με τη διαφορά ότι είναι ανοιχτού λογισμικού. Υποστηρίζει πολλές βάσεις δεδομένων όπως η Oracle, η PostgreSQL και η MySQL η οποία χρησιμοποιήθηκε για τη συγκεκριμένη διπλωματική εργασία. Τα αρχεία php έχουν την δυνατότητα να περιλαμβάνουν κείμενο, ετικέτες html και εκτελέσιμο κώδικα. Επιστρέφουν στον φυλλομετρητή (web server), από τον οποίο ερμηνεύονται, σαν απλό HTML.

Η χρήση της PHP ποικίλει ανάλογα με τον τρόπο με τον οποίο θέλουμε να την χρησιμοποιήσουμε. Μπορεί να χρησιμοποιηθεί για την συλλογή δεδομένων μέσω ειδικής φόρμας που συμπληρώνεται από τον χρήστη και με την μέθοδο POST μπορεί να σταλεί στον εξυπηρετητή server και να διαχειριστεί ανάλογα. Μπορεί να χρησιμεύσει στην αποστολή και λήψη των cookies (κομμάτια κώδικα που αφορούν τις προτιμήσεις και άλλες ενέργειες του χρήστη), μπορεί να τροποποιήσει δεδομένα σε βάσεις δεδομένων, για παράδειγμα: 1) προσθήκη, 2) διαγραφή, 3) αλλαγή τιμής, 4) άδειασμα ολόκληρου πίνακα στην βάση δεδομένων. Ακόμη, η PHP χρησιμοποιείται για τον περιορισμό των χρηστών σε σελίδες ανάλογα με τους κωδικούς που έχουν εισάγει στις απαραίτητες φόρμες για την σύνδεση τους με τον εξυπηρετητή, για την κρυπτογράφηση και αποστολή δεδομένων σε εξυπηρετητές καθώς και για προγραμματισμό αυτόματων διαδικασιών στον εξυπηρετητή.

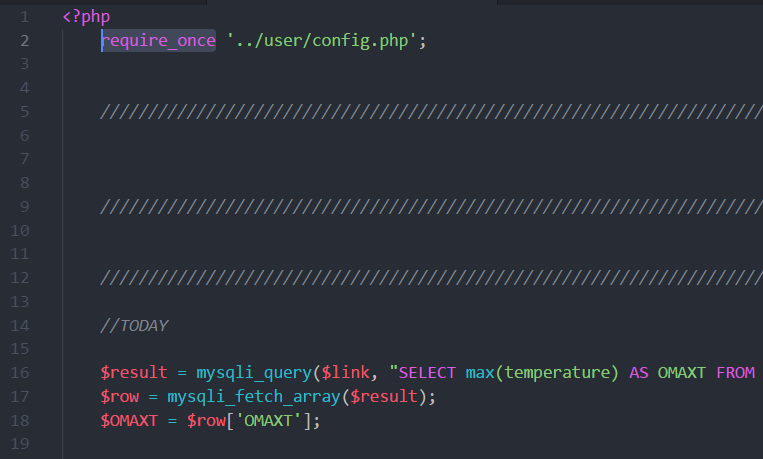
Την παρούσα στιγμή, κατά την οποία συγγράφεται αυτή η διπλωματική εργασία η γλώσσα PHP βρίσκεται στην έκδοση *7.3.0* [12]*.* Είναι συμβατή με τους πιο συχνά χρησιμοποιούμενους διακομιστές παγκόσμιου ιστού όπως ο ISS της Microsoft και ο Apache. Μια σελίδα PHP υπόκειται σε επεξεργασία μέσω του συγκεκριμένου διακομιστή, ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο.

Για την ορθή γραφή του κώδικα σε γλώσσα PHP είναι απαραίτητες οι βασικές μεταβλητές. Η μεταβλητή στην PHP, είναι μία θέση προσωρινής μνήμης στην οποία αποθηκεύονται πληροφορίες οι οποίες διαχειρίζονται από τον εξυπηρετητή και μπορούν να αλλάξουν ριζικά τα παραγόμενα αποτελέσματα. Ακολουθεί μία ανάλυση των βασικών μεταβλητών.

1. **$GLOBALS**: Μια καθολική μεταβλητή που χρησιμοποιείται για πρόσβαση σε καθολικές μεταβλητές από οπουδήποτε στο PHP script. Η PHP αποθηκεύει όλες τις μεταβλητές αυτού του τύπου σε έναν πίνακα που ονομάζεται $GLOBALS[index]. Ο δείκτης index διατηρεί το όνομα της εκάστοτε μεταβλητής.
2. **$\_SERVER**: Μια καθολική μεταβλητή που περιέχει πληροφορίες σχετικά με τις κεφαλίδες, τις διαδρομές των φακέλων που αφορούν τη συγκεκριμένη ιστοσελίδα και τις τοποθεσίες κάθε script.
3. **$\_REQUEST**: Μια καθολική μεταβλητή που περιέχει τις τιμές $\_GET, $\_POST και $\_COOKIE. Χρησιμοποιείται για τη συλλογή των δεδομένων μετά την υποβολή μιας φόρμας HTML. Όταν ένας χρήστης υποβάλλει τα στοιχεία κάνοντας κλικ στο αντίστοιχο κουμπί της φόρμας, τα δεδομένα αυτής στέλνονται στο αρχείο που ορίζεται στην ετικέτα html <form>. Ο τρόπος με τον οποίο χρησιμοποιείται, χρειάζεται ιδιαίτερη προσοχή, λόγω της ευαισθησίας των δεδομένων.
4. **$\_GET**: Μια προκαθορισμένη μεταβλητή που χρησιμοποιείται ευρέως για τη συλλογή δεδομένων σε μια φόρμα html (με τη method=”GET”). Επίσης μπορεί να συλλέξει τα δεδομένα που αποστέλλονται στη διεύθυνση URL. Οι πληροφορίες που αποστέλλονται από μια φόρμα με τη μέθοδο GET, είναι αόρατες σε όλους, και εμφανίζονται στη γραμμή διευθύνσεων του φυλλομετρητή, επίσης υπάρχει όριο στην ποσότητα των πληροφοριών προς αποστολή με αυτήν τη μέθοδο.
5. **$\_POST**: Μια προκαθορισμένη μεταβλητή που χρησιμοποιείται ευρέως για τη συλλογή δεδομένων σε μια φόρμα html (με τη method=”POST”). Οι πληροφορίες που αποστέλλονται από μια φόρμα με τη μέθοδο POST, είναι αόρατες στους άλλους και δεν υπάρχουν όρια για τον όγκο των πληροφοριών κατά την αποστολή.
6. **$\_FILES**: Μια καθολική μεταβλητή που χρησιμοποιείται για να γίνει μεταφόρτωση αρχείων στο διακομιστή (server).
7. **$\_COOKIE**: Ένα μικρό αρχείο που ενσωματώνει ο διακομιστής (server) στον υπολογιστή του χρήστη. Για να οριστεί ένα cookie χρησιμοποιείται η συνάρτηση setcookie().
8. **$\_SESSION**: Μια μεταβλητή συνεδρίας που χρησιμοποιείται για την αποθήκευση πληροφοριών , ή για την αλλαγή των ρυθμίσεων, για μια συνεδρία χρήστη. Οι μεταβλητές συνεδρίας είναι διαθέσιμες σε όλες τις σελίδες σε μια εφαρμογή. Κατά την ενασχόληση με μια εφαρμογή, μπορούν να εκτελεστούν ενέργειες και αλλαγές δεδομένων. Η κατάσταση αυτή είναι μια συνεδρία. Μια συνεδρία PHP, επιτρέπει την αποθήκευση πληροφοριών του χρήστη στο διακομιστή, για μελλοντική χρήση (δηλαδή όνομα, στοιχεία αγορών, κλπ). Ωστόσο, οι πληροφορίες συνεδρίας, είναι προσωρινές και θα διαγράφονται, αφού ο χρήστης έχει αποχωρήσει από το διαδικτυακό τόπο. Για μόνιμη αποθήκευση, μπορεί να χρησιμοποιηθεί μια βάση δεδομένων [13].

Σύνταξη της PHP

Για να ξεκινήσει ένας κώδικας της PHP είναι απαραίτητη η χρήση των ετικετών *<?php* στην εισαγωγή και *?>* και στο τέλος του κώδικα. Μέσα σε αυτές τις ετικέτες περιλαμβάνεται όλος ο κώδικας. Κάθε εντολή της PHP πρέπει να τελειώνει με το σύμβολο “;” το οποίο καλείται ο διαχωριστής και διαχωρίζει την μια εντολή από την επόμενή της. Μέσα σε ένα αρχείο php μπορεί να ανοίξει και να κλείσει όσες φορές είναι απαραίτητο κώδικας. Ακόμη, σε περίπτωση που κομμάτια κώδικα πρέπει να επαναληφθούν πολλές φορές μέσα στο ίδιο αρχείο, μπορεί να γίνει ξεχωριστή συγγραφή ενός php αρχείο που περιέχει τον επαναλαμβανόμενο κώδικα και έπειτα, στα υπόλοιπα αρχεία να εκτελεστεί η συνάρτηση require\_once και έπειτα να μπει ανάμεσα σε μονά εισαγωγικά το μονοπάτι στο οποίο βρίσκεται το αρχείο με τον επαναλαμβανόμενο κώδικα. Στην παρακάτω εικόνα μπορεί να διαπιστωθεί ο τρόπος γραφής της συνάρτησης require\_once.



Σχήμα 2.9 Tρόπος σύνταξης της συνάρτησης require\_once στην PHP

Ο τρόπος σύνταξης είναι πολύ απλός και κατανοητός σε έναν αρχάριο χρήστη. Παρακάτω δίνεται μία εικόνα ενός κομμάτι κώδικα σε για ένα αρχείο PHP.



Σχήμα 2.10 Tρόπος σύνταξης της PHP

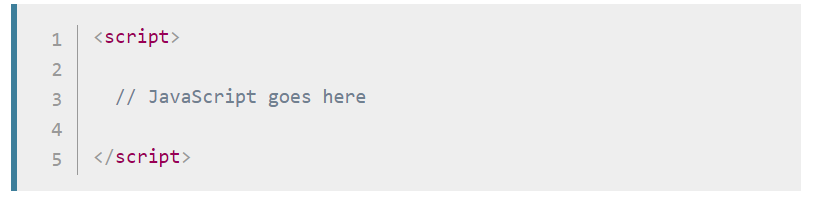
Στην εικόνα διακρίνουμε πώς η PHP γράφεται ανάμεσα στα στοιχεία <body> </body> της HTML. Στην PHP δεν είναι απαραίτητη η δήλωση των μεταβλητών πριν την ανάθεση μιας τιμής σε αυτές, όπως επίσης και η δήλωση του τύπου αυτών. Η PHP αυτόματα αποφασίζει τον τύπο ανάλογα με την τιμή που έχει τεθεί στην εκάστοτε μεταβλητή. Η εντολή echo είναι υπεύθυνη για την εκτύπωση του συνόλου των χαρακτήρων που βρίσκονται μέσα στα εισαγωγικά. Γενικά στην γλώσσα PHP ισχύουν οι τελεστές και οι εντολές εκτέλεσης υπό συνθήκη που υπάρχουν και σε άλλες γλώσσες.

* + 1. **Γλώσσα Scripting - JavaScript**

Η γλώσσα scripting (σεναρίων) που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία είναι η JavaScript. Είναι μία δυναμική γλώσσα προγραμματισμού ηλεκτρονικών υπολογιστών και η πιο δημοφιλής γλώσσα scripting ανά τον κόσμο. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται. Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού. Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side). Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAscript [14].

Χρησιμοποιείται από εκατομμύρια ιστοσελίδες και προσθέτει χρηστικότητα, επικύρωση φορμών και την επικοινωνία με διακομιστές. Γίνεται ευρεία χρήση της ακόμα και για δημιουργία desktop εφαρμογών αλλά και για εφαρμογές κινητής τηλεφωνίας. Υποστηρίζεται από όλους τους σύγχρονους φυλλομετρητές όπως ο Chrome, o Safari και o Firefox. Γνώστες αλλά και χρήσιμες εφαρμογές οι οποίες έχουν δημιουργηθεί με την γλώσσα προγραμματισμού JavaScript είναι το Slack (εφαρμογή η οποία έχει βοηθήσει πολύ στην οργάνωση και επικοινωνία των εργαζομένων – ομάδων των επιχειρήσεων) και το WhatsApp (εφαρμογή μέσα από την οποία οι άνθρωποι μπορούν να επικοινωνήσουν μεταξύ τους, με μηνύματα κειμένου, εικόνας αλλά και να αποστείλουν αρχεία). Και οι δύο αυτές εφαρμογές βασίστηκαν στο framework (βιβλιοθήκη) Electron το οποίο είναι γραμμένο για την JavaScipt [15].

Η υλοποίησης της JavaScript είναι client-side (υπολογιστή πελάτη). Με το χαρακτηριστικό αυτό προσδίδεται ασφάλεια για τα δεδομένα που εισάγονται, καθώς γίνεται έλεγχος για την εγκυρότητά τους πριν αυτά αποσταλούν στο διακομιστή (server). Έτσι επιτυγχάνεται η αποφυγή κακόβουλων επιθέσεων από τους χρήστες.

Για να προσθέσουμε λειτουργικότητα στην σελίδα μας χρησιμοποιούμε την JavaScript. Η γλώσσα αυτή λειτουργεί με παρόμοια λογική όπως και η CSS. Όπως η CSS χρησιμοποιεί τα στοιχεία <link> και <style>, η JavaScript χρειάζεται μόνο ένα στοιχείο, το <script>. Για να ξεκινήσουμε να γράφουμε κώδικα σε JavaScript απλά ανοίγουμε τον επεξεργαστή κειμένου μας και μέσα στο στοιχείο <head> ανοίγουμε το στοιχείο <script> και είμαστε έτοιμοι να γράψουμε τον κώδικα μας, όπως φαίνεται στο παρακάτω παράδειγμα.

Σχήμα 2.11 Tρόπος σύνταξης της JavaScript

Ένας άλλος τρόπος για να προσθέσουμε κώδικα JavaScript στην σελίδα είναι με την σύνδεση ενός εξωτερικού αρχείου. Στην προκειμένη περίπτωση η σύνδεση γίνεται όπως φαίνεται παρακάτω.

Σχήμα 2.12 Σύνδεση αρχείου JavaScript σε αρχείου HTML/PHP

Τα εξωτερικά αυτά αρχεία έχουν την κατάληξη .js και δεν πρέπει να περιέχουν τις ετικέτες <script>…</script>. Για την συγγραφή της παρούσας διπλωματικής εργασίας, δημιουργήθηκαν αρκετά αρχεία σε κώδικα JavaScript καθώς υπάρχει μεγάλη παραμετροποίηση στον τρόπο αλλά και στο είδος εμφάνισης των δεδομένων προς τον χρήστη.

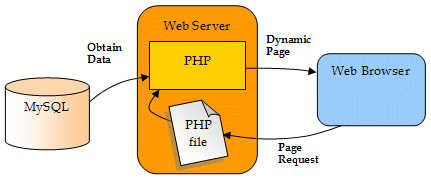


Σχήμα 2.13 Κομμάτι κώδικα σε JavaScript για την λειτουργικότητα του μενού της διαδικτυακής εφαρμογής Smart Homie

* + 1. **Γλώσσα Σχεσιακών Βάσεων Δεδομένων MySQL**

Η MySQL, η οποία είναι ακρολέξιο του (My Structured Query Language), είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων [16]. Η MySQL είναι η πιο δημοφιλής βάση δεδομένων ανοιχτού κώδικα στον κόσμο, επιτρέποντας οικονομικά αποδοτική παράδοση αξιόπιστων, υψηλής απόδοσης και κλιμακούμενων εφαρμογών και ενσωματωμένων βάσεων δεδομένων. Πρόκειται για μια ολοκληρωμένη βάση δεδομένων, συμβατή με το ACID, με πλήρη δέσμευση, επαναφορά, αποκατάσταση σφαλμάτων και δυνατότητες κλειδώματος σε επίπεδο σειράς. Στην πληροφορική, το ACID (ατομικότητα, συνέπεια, απομόνωση, μονιμότητα) είναι ένα σύνολο ιδιοτήτων το οποίο εγγυάται ότι οι συναλλαγές στην βάση δεδομένων (ΒΔ) λειτουργούν αξιόπιστα. Η ιδέα του ACID είναι να αξιολογεί την δομή βάσεων δεδομένων και εφαρμογών. Όσον αφορά τις βάσεις δεδομένων μια λογική ενέργεια που έχει να κάνει με τα δεδομένα της αποκαλείται συναλλαγή [17]. Η MySQL παρέχει την ευκολία χρήσης, την επεκτασιμότητα και την υψηλή απόδοση, καθώς και ένα πλήρες πακέτο προγραμμάτων οδήγησης βάσεων δεδομένων και οπτικά εργαλεία για να βοηθήσουν τους προγραμματιστές και τα DBA να δημιουργήσουν και να διαχειριστούν τις κρίσιμες για την επιχείρησή τους εφαρμογές με την βοήθεια της MySQL. Αναπτύσσεται, διανέμεται και υποστηρίζεται από την Oracle. Παρέχει της ακόλουθες δυνατότητες:

* Υψηλή απόδοση και δυνατότητα κλιμάκωσης για την κάλυψη των απαιτήσεων των εκθετικά αυξανόμενων φορτίων δεδομένων και των χρηστών
* Αυτο-θεραπευόμενες ομάδες ανάπλασης δεδομένων για τη βελτίωση της επεκτασιμότητας, της απόδοσης και της διαθεσιμότητας
* SQL και NoSQL πρόσβαση για την εκτέλεση πολύπλοκων ερωτημάτων και απλών, γρήγορων λειτουργιών για τιμές κλειδιά
* Ανεξαρτησία πλατφόρμας που δίνει την ευελιξία να αναπτύξουμε πολλαπλά λειτουργικά συστήματα.
* Διαλειτουργικότητα μεγάλου όγκου δεδομένων μέσω της MySQL [18].

 Τα δεδομένα στη MySQL αποθηκεύονται σε αντικείμενα βάσης δεδομένων τα οποία ονομάζονται πίνακες. Ένας πίνακας είναι μια συλλογή από σχετικές καταχωρήσεις δεδομένων και αποτελείται από γραμμές και στήλες. Για την αποθήκευση πληροφοριών σε κατηγορίες οι βάσεις δεδομένων είναι πολύ χρήσιμες.

Σχήμα 2.14 Λειτουργία και αλληλεπίδραση της MySQL

Μέσω των SQL εντολών πραγματοποιείται η διαχείριση των δεδομένων στις βάσεις δεδομένων. Η πιο σημαντική ίσως κατηγορία εντολών που χρησιμοποιείται είναι τα queries. Ένα query είναι ένα ερώτημα ή ένα αίτημα. Με τη MySQL, επιτρέπεται η διερεύνηση σε μια βάση δεδομένων και η επιστροφή ζητηθέντων πληροφοριών. Παρακάτω παρουσιάζονται οι βασικότερες εντολές SQL με τη σημασία τους.

1. SELECT: Εξάγει δεδομένα από μια βάση δεδομένων
2. UPDATE: Ενημερώνει δεδομένα σε μια βάση δεδομένων
3. DELETE: Διαγράφει δεδομένα από μια βάση δεδομένων
4. INSERT INTO: Εισάγει δεδομένα σε μια βάση δεδομένων
5. CREATE DATABASE: Δημιουργεί μια βάση δεδομένων
6. ALTER DATABASE: Τροποποιεί μια βάση δεδομένων
7. CREATE TABLE: Δημιουργεί έναν νέο πίνακα
8. ALTER TABLE: Τροποποιεί έναν πίνακα
9. DROP TABLE: Διαγράφει έναν πίνακα
10. CREATE INDEX: Δημιουργεί έναν δείκτη (κλειδί αναζήτησης)
11. DROP INDEX: Διαγράφει έναν δείκτη
    1. **Υλικός Εξοπλισμός - Hardware**

Για την παρούσα διπλωματική εργασία, είναι απαραίτητος ο υλικός εξοπλισμός (Hardware) διότι χωρίς αυτόν είναι αδύνατη η συλλογή δεδομένων, η οπτική απεικόνιση στο χρήστη καθώς και η δυνατότητα που δίνεται σε αυτόν να ελέγχει τις έξυπνες συσκευές του. Παρακάτω γίνεται μία αναφορά στον εξοπλισμό που χρησιμοποιήθηκε και πώς το κάθε κομμάτι συμπληρώνει το εγχείρημα που πραγματοποιήθηκε. Γίνεται εκτενής αναφορά στις δυνατότητες καθώς και στην επεκτασιμότητα που δέχεται το κάθε υλικό στοιχείο.

2. 2. 1. **Μικροελεγκτής Arduino uno Rev3**

*Ιστορία του Arduino*

Το 2005, με σκοπό την ευκολότερη εκμάθηση των ηλεκτρονικών για τους μαθητές, ο καθηγητής Massimo Banzi στο Ινστιστούτο Σχεδιασμού Αλληλεπίδρασης στην Ivrea της Ιταλίας, επιχείρησε τον σχεδιασμό μιας νέας συσκευής για τον έλεγχο προγραμμάτων και διαδραστικών σχεδίων από τους ίδιους τους μαθητές, με χαμηλότερο κόστος. Μαζί με έναν μηχανικό από το Πανεπιστήμιο Malmo, τον David Cuatrielles, αποφάσισαν να φτιάξουν ένα μικροελεγκτή, ο οποίος ήταν πιο προσιτός ως προς τη χρήση του. Ονόμασαν το έργο τους Arduinο of Ivrea «Arduino». Το πρώτο μοντέλο περιλάμβανε ATmega8 με άμεση σύνδεση RS-232 με το μικροελεγκτή και όλα τα στοιχεία του και ονομάστηκε «Serial Arduino» [19].

*Πληροφορίες για το Arduino*

Το Arduino βασίζεται στην αρχή του ελεύθερου λογισμικού και υλικού. Είναι ουσιαστικά ένας μικροελεγκτής μέσω του οποίου μπορεί να πραγματοποιηθεί έλεγχος και τροποποίηση παραμέτρων σε φυσικές συσκευές. Το περιβάλλον ανάπτυξης του λογισμικού είναι βασισμένο στις γλώσσες προγραμματισμού Processing και Wiring, η οποία είναι στην ουσία η C++ και ένα σύνολο από βιβλιοθήκες υλοποιημένες στην C++. Τόσο η Wiring όσο και η Processing είναι ανοιχτού κώδικα (open source) και μπορεί κάποιος να τις “κατεβάσει δωρεάν”. Το Arduino χρησιμοποιεί έναν ειδικά προγραμματισμένο ATmega328P αντί του FTDI chip ώστε να επιτρέπεται τόσο η πιο γρήγορη ταχύτητα μεταφοράς όσο και η γρήγορη σειριακή επικοινωνία.

*Πλεονεκτήματα επιλογής Arduino*

* **Χαμηλό κόστος**: Οι πλακέτες του Arduino είναι εξαιρετικά οικονομικές σε σύγκριση με άλλες πλατφόρμες μικροελεγκτών. Με τις σχηματικές αναπαραστάσεις που κυκλοφορούν στο Διαδίκτυο δεν μπορεί να κατασκευάσει κάποιος την φθηνότερη εκδοχή ενός Arduino. Ωστόσο αν προμηθευτεί την έτοιμη (μονταρισμένη πλακέτα) αυτή θα κοστίσει το μέγιστο 50 €. Το κόστος διαμορφώνεται ανάλογα με την έκδοση.
* **Πληθώρα Λειτουργικών Συστημάτων**: Οι μηχανικοί λογισμικού, ανέπτυξαν το περιβάλλον προγραμματισμού του Arduino όχι μόνο για Windows αλλά και Machinstoh OSX καθώς και λειτουργικά συστήματα Linux.
* **Απλό και ξεκάθαρο περιβάλλον προγραμματισμού**: Το περιβάλλον προγραμματισμού του Arduino είναι ευέλικτο για προχωρημένους χρήστες αλλά ταυτόχρονα ενδείκνυται και για αρχάριους.
* **Ανοιχτού λογισμικού που επεκτείνεται και παραμετροποιείται**: To Software του Arduino διανέμεται με τη μορφή εργαλείων ανοιχτού λογισμικού και μπορεί να επεκταθεί από έμπειρους προγραμματιστές. Η επέκταση της γλώσσας προγραμματισμού του γίνεται μέσω των βιβλιοθηκών της C++ και οι άνθρωποι που ενδιαφέρονται για περαιτέρω ενασχόληση με τους μικροελεγκτές μπορούν να μεταβούν από το Arduino στην AVR C, η γλώσσα στην οποία βασίστηκε το λογισμικό του Arduino, για προγραμματισμό των μικροελεγκτών Atmel. Ομοίως μπορεί κάποιος να προσθέσει κώδικα της AVR C στο πρόγραμμα που έχει γράψει για το Arduino του.
* **Ανοιχτού υλικού το οποίο μπορεί να επεκταθεί**: To Arduino βασίζεται στους μικροελεγκτές της Atmel. Οι σχηματικές αναπαραστάσεις για τα αναπτυξιακά είναι υπό την άδεια της Creative Commons, επιτρέποντας την κατασκευή αναπτυξιακού σε έμπειρους σχεδιαστές, εξελίσσοντας το ήδη υπάρχον χωρίς να δημιουργηθεί κάποιο νομικό πρόβλημα. Ακόμα, επιτρέπεται στους όχι και τόσο έμπειρους χρήστες να επιδιώξουν να αντιγράψουν και να κατασκευάσουν την πλακέτα για να καταλάβουν τη λειτουργία ενός Arduino [20].

*Διάφορες πλακέτες Arduino*

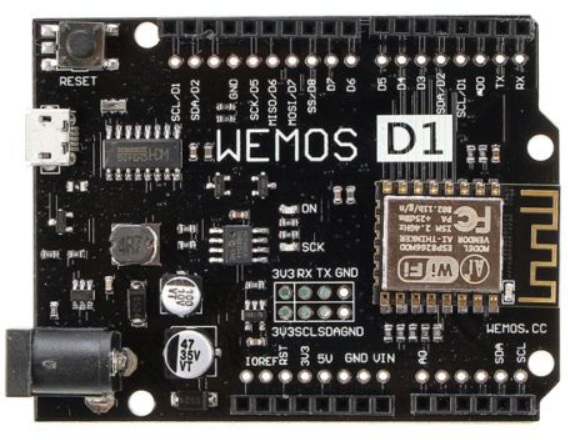
Στην παρούσα διπλωματική εργασία για λόγο ευκολίας αλλά και μεγάλης παραμετροποίησης σύμφωνα με τις ανάγκες της διαδικτυακής εφαρμογής επιλέχθηκε το arduino uno. Αυτό δεν σημαίνει πώς δεν υπάρχουν και άλλες πλακέτες arduino που είναι εξίσου αξιόλογες ανάλογα με την χρήση στην οποία υπόκεινται. Στον παρακάτω πίνακα παρατίθενται διάφορες ακόμα πλακέτες arduino καθώς και τα τεχνικά χαρακτηριστικά τους [21].

***Πίνακας 2.2: Είδη Arduino***

|  |  |
| --- | --- |
| **Πλακέτα Arduino** | **Τεχνικά Χαρακτηριστικά** |
| **Arduino Uno Rev3** | * Μικροελεγκτής : ATMega328P * Τάση λειτουργίας: 5V * Ψηφιακοί είσοδοι/έξοδοι: 14(6 παρέχουν PWM έξοδο) * Ρεύμα Dc ανά είσοδο/ έξοδο: 40mA Αναλογικές είσοδοι: 6 * Μνήμη Flash: 32KB (ATMega328P) * SRAM: 2KB (ATMega328P) * EEPROM: 1KB (ATMega328P) * Ταχύτητα ρολογιού: 16MHz |
| **Arduino Ethernet** | * Μικροελεγκτής: ATMega328 * Τάση λειτουργίας: 5V * Ψηφιακοίείσοδοι/έξοδοι: 14(64παρέχουν PWM έξοδο) * Ρεύμα Dc ανά είσοδο/ έξοδο: 40mA * Αναλογικές είσοδοι: 6 * Μνήμη Flash: 32KB (ATMega328P) * SRAM: 2KB (ATMega328P) * EEPROM: 1KB (ATMega328P) * Ταχύτητα ρολογιού: 16MHz W5100 * TCP/IP Embedded Ethernet Controller |
| **Arduino Fio** | * Μικροελεγκτής: ATMega328P * Τάση λειτουργίας: 3.3V * Ψηφιακοί είσοδοι/έξοδοι: 14(6 παρέχουν PWM έξοδο) * Ρεύμα Dc ανά είσοδο/ έξοδο: 40mA * Αναλογικές είσοδοι: 8 * Μνήμη Flash: 32KB * SRAM: 2KB * EEPROM: 1KB Ταχύτητα ρολογιού: 8MHz |
| **Arduino Mega ADK** | * Μικροελεγκτής: ATMega2560 * Τάση λειτουργίας: 5V * Ψηφιακοί είσοδοι/έξοδοι: 54(6 παρέχουν PWM έξοδο) * Ρεύμα Dc ανά είσοδο/ έξοδο: 40mA * Αναλογικές είσοδοι: 16 * Μνήμη Flash: 256KB * SRAM: 8KB * EEPROM: 4KB * Ταχύτητα ρολογιού: 16MHz |
| **Arduino Due** | * Μικροελεγκτής: AT91SAM3X8E * Τάση λειτουργίας: 3.3V * Συνολικό ρεύμα Dc εξόδου: 130mA * Ψηφιακοί είσοδοι/έξοδοι: 54(12 παρέχουν PWM έξοδο) * Αναλογικές είσοδοι: 12 * Αναλογικές έξοδοι: 2 (DAC) * Μνήμα Flash: 512KB * SRAM: 96KB (64KB και 32ΚΒ) * Ταχύτητα ρολογιού: 84MHz |
| **Arduino Leonardo** | * Μικροελεγκτής: ATMega32U4 * Τάση λειτουργίας: 5V * Ψηφιακοί είσοδοι/έξοδοι:20(7 παρέχουν PWM έξοδο) * Ρεύμα Dc ανά είσοδο/ έξοδο: 40mA * Αναλογικές είσοδοι: 12 * Μνήμη Flash: 32KB (ATMega32U4) * SRAM: 2KB (ATMega32U4) * EEPROM: 1KB (ATMega32U4) * Ταχύτητα ρολογιού: 16MHz |
| **Arduino Yun** | * Μικροελεγκτής: ATMega32U4 * Τάση λειτουργίας: 5V * Ψηφιακοί είσοδοι/έξοδοι: 20(7 παρέχουν PWM έξοδο) * Ρεύμα Dc ανά είσοδο/ έξοδο: 40mA * Αναλογικές είσοδοι: 12 * Μνήμη Flash: 32K * SRAM: 2.5KB EEPROM: 1KB * Ταχύτητα ρολογιού: 16MHz * Μικροεπεξεργαστής: Linux Processorv Arheros AR9331 * Αρχιτεκτονική: MIPS @400MHz * Τάση λειτουργίας: 3.3V * Ethernet: IEEE 802.3 10/100Mbit/s * Wi-Fi: IEE 802.11b/g/n * Card Reader: 64 MB DDR2 * Μνήμη Flash: 16MB |
| **Arduino Micro** | * Μικροελεγκτής: ATMega32U4 * Τάση λειτουργίας: 5V * Ψηφιακοί είσοδοι/έξοδοι: 20(7 παρέχουν PWM έξοδο) * Ρεύμα Dc ανά είσοδο/ έξοδο: 40mA * Αναλογικές είσοδοι: 12 * Μνήμη Flash: 32KB (ATMega32U4) * SRAM: 2.5KB (ATMega32U4) * EEPROM: 1KB (ATMega32U4) * Ταχύτητα ρολογιού:16MHz |

* + 1. **Μικροελεγκτής WeMos D1 R2**

*Πληροφορίες για το WeMos D1*

Ο μικροελεγκτής WeMos D1 είναι ένας μικροελεγκτής ο οποίος είναι συμβατός με το arduino uno, και είναι μία πλακέτα WiFi βασισμένη στο ESP8266EX. Διαθέτει 11 ψηφιακά pins εισόδου/εξόδου, όλα τα pins υποστηρίζουν τις λειτουργίες interrupt/pwm/I2C/one-wire εκτός από pin D0. Διαθέτει μία αναλογική είσοδο όπου η μέγιστη τάση εισόδου ανέρχεται στα 3.2V. Η σύνδεση και ο προγραμματισμός του γίνεται μέσω μίας micro USB θύρας, διαθέτει ένα βύσμα ρεύματος τύπου barrel jack, το οποίο επιτρέπει στο WeMos D1 να περάσει τάση εισόδου από 9 έως 24 volt. Είναι συμβατό με διάφορες πλακέτες arduino και nodemcu.

Σχήμα 2.15 WeMos D1 R2

*Τεχνικά χαρακτηριστικά του WeMos D1*

Στο παρακάτω πίνακα εμφανίζονται τα τεχνικά χαρακτηριστικά για τον μικροελεγκτή WeMos D1 R2 που χρησιμοποιήθηκε για την παρούσα διπλωματική εργασία.

***Πίνακας 2.3: Τεχνικά Χαρακτηριστικά WeMos D1 R2***

|  |  |
| --- | --- |
| Μικροελεγκτής | ESP-8266EX |
| Τάση Λειτουργίας | 3.3V |
| Ψηφιακά Pins I/O | 11 |
| Αναλογικά pins εισόδου | 1 (Μέγιστη τάση εισόδου: 3.2V) |
| Χρονισμός πυρήνα | 80Mhz / 160Mhz |
| Μνήμη Flash | 4MBytes |
| Μήκος | 68.6mm |
| Πλάτος | 53.4mm |
| Βάρος | 25g |

*Pins – Λειτουργίες του WeMos D1*

Κάθε pin στον μικροελεγκτή WeMos έχει σχεδιαστεί έτσι ώστε να συμφωνεί με τους μικροελεγκτές με τους οποίους έχει επίσημη συμβατότητα. Για τον λόγο αυτό παρατίθεται ο πίνακας με τα pins καθώς και τις λειτουργίες του μικροελεγκτή.

***Πίνακας 2.4: Σύνδεση - Λειτουργία των pins του WeMos D1 R2***

|  |  |  |
| --- | --- | --- |
| **Pin** | **Function** | **ESP-8266 Pin** |
| TX | TXD | TXD |
| RX | RXD | RXD |
| A0 | Analog input, max 3.3V input | A0 |
| D0 | IO | GPIO16 |
| D1 | IO, SCL | GPIO5 |
| D2 | IO, SDA | GPIO4 |
| D3 | IO, 10k Pull-up | GPIO0 |
| D4 | IO, 10k Pull-up, BUILTIN\_LED | GPIO2 |
| D5 | IO, SCK | GPIO14 |
| D6 | IO, MISO | GPIO12 |
| D7 | IO, MOSI | GPIO13 |
| D8 | IO, 10k Pull-down, SS | GPIO15 |
| G | Ground | GND |
| 5V | 5V | - |
| 3V3 | 3.3V | 3.3V |
| RST | Reset | RST |

*Διάφορες Πλακέτες WeMos D1*

Η WeMos εκτός από τον μικροελεγκτή WeMos D1 R2 διαθέτει και άλλες πλακέτες WeMos D1. Ανάλογα με την χρήση, την σύνδεση και τον χώρο που έχουμε στην διάθεση μας ανάλογα με το project μπορούμε να επιλέξουμε διαφορετικό μικροελεγκτή. Στο επόμενο πίνακα επεξηγούνται οι υπόλοιπες πλακέτες WeMos D1[22].

***Πίνακας 2.5: Σύνδεση - Πλακέτες WeMos D1***

|  |  |
| --- | --- |
| **ΠΛΑΚΕΤΑ** | **ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ** |
| **WeMos D1 Mini Pro** | * Μικροελεγκτής: ESP-8266EX * Τάση Λειτουργίας: 3.3V * Ψηφιακά Pins I/O: 11 * Αναλογικά pins εισόδου: 1 (Μέγιστη τάση εισόδου: 3.2V) * Χρονισμόςπυρήνα: 80Mhz/160Mhz * Μνήμη Flash: 16MBytes * Μήκος: 48mm * Πλάτος: 25.4mm * Βάρος 4.5g |
| **WeMos D1 Mini** | * Μικροελεγκτής: ESP-8266EX * Τάση Λειτουργίας: 3.3V * Ψηφιακά Pins I/O: 11 * Αναλογικά pins εισόδου: 1 (Μέγιστη τάση εισόδου: 3.2V) * Χρονισμόςπυρήνα: 80Mhz/160Mhz * Μνήμη Flash: 4MBytes * Μήκος: 34.2mm * Πλάτος: 25.6mm * Βάρος 3g |
| **WeMos D1 Mini Lite** | * Μικροελεγκτής: ESP-8266EX * Τάση Λειτουργίας: 3.3V * Ψηφιακά Pins I/O: 11 * Αναλογικά pins εισόδου: 1 (Μέγιστη τάση εισόδου: 3.2V) * Χρονισμόςπυρήνα: 80Mhz/160Mhz * Μνήμη Flash: 1MBytes * Μήκος: 34.2mm * Πλάτος: 25.6mm * Βάρος 10g |

* + 1. **Δρομολογητής - Router**

Για την εκπόνηση της παρούσας διπλωματικής εργασίας καθώς και για την ανάπτυξη της διαδικτυακής εφαρμογής ήταν αναγκαία η τοποθέτηση ενός δρομολογητή (router) στην σύνδεση των μικροελεγκτών με τον εξυπηρετητή. Δρομολογητής (router) είναι μια ηλεκτρονική συσκευή η οποία αναλαμβάνει την αποστολή και λήψη πακέτων δεδομένων μεταξύ ενός ή περισσοτέρων διακομιστών, άλλων δρομολογητών και πελατών, κατά μήκος πολλαπλών δικτύων (δρομολόγηση). Η δρομολόγηση, δηλαδή είναι η διαδικασία μεταφοράς δεδομένων από ένα σημείο σε ένα άλλο και αποτελεί κεντρική λειτουργία του επιπέδου δικτύου, γίνεται με βάση διάφορα κριτήρια και τελικώς επιλέγεται μία ανάμεσα σε διάφορες πιθανές διαδρομές. Κάθε δρομολογητής χρησιμοποιεί ένα ή περισσότερα πρωτόκολλα δρομολόγησης. Με βάση αυτά τα πρωτόκολλα ο δρομολογητής καθορίζει ποιος ή ποιοι διακομιστές ή δρομολογητές είναι οι καταλληλότεροι κάθε χρονική στιγμή και δρομολογεί τα πακέτα δεδομένων προς αυτούς.

*Forwarding – Προώθηση*

Για τις διαδικασίες προώθησης αμιγώς του Internet Protocol (IP), ο δρομολογητής (router) έχει κατασκευαστεί ώστε να ελαχιστοποιεί την κατάσταση της πληροφορίας που σχετίζεται με το κάθε πακέτο. Ο κύριος σκοπός ενός δρομολογητή είναι να συνδέει πολλαπλά δίκτυα και να προωθεί τα πακέτα που προορίζονται είτε στα δίκτυά του, είτε σε άλλα δίκτυα. Η προεπιλεγμένη (στατική) δρομολόγηση (default or static route) είναι αυτή που χρησιμοποιείται όταν η δρομολόγηση όλων των προορισμών δεν διαφοροποιείται στον πίνακα (routing table). Αυτή χρησιμοποιείται στα μικρά δίκτυα όπως είναι π.χ. τα οικιακά η τα δίκτυα μικρών επιχειρήσεων, όπου η προεπιλεγμένη (η στατική) δρομολόγηση απλά στέλνει όλη την κίνηση στον πάροχο - Internet Service Provider (ISP). Η προεπιλεγμένη (στατική) δρομολόγηση μπορεί να ρυθμιστεί “χειροκίνητα” ως static route, η μπορεί να αποκτηθεί από το δυναμικό πρωτόκολλο δρομολόγησης (dynamic routing protocol), η από το DHCP. Ο router μπορεί να χρησιμοποιηθεί τόσο ως DHCP client, όσο και ως DHCP server [23].

Για τις ανάγκες της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε το πρωτόκολλο δρομολόγησης DHCP, με αποτέλεσμα την απόδοση στατικών IP στους μικροελεγκτές, στις έξυπνες συσκευές καθώς και στον εξυπηρετητή δημιουργώντας έτσι ένα μικρό τοπικό δίκτυο για την ταχύτερη, ασφαλέστερη και χωρίς απώλειες αποστολή δεδομένων μεταξύ των μικροελεγκτών και του εξυπηρετητή. Τον ρόλο του δρομολογητή στην παρούσα διπλωματική εργασία κατέχει ένα Thomson TG585 V7.



Σχήμα 2.16 Δρομολογητής – Μόντεμ TG585 V7

* 1. **Πλατφόρμες Ανάπτυξης Λογισμικού**

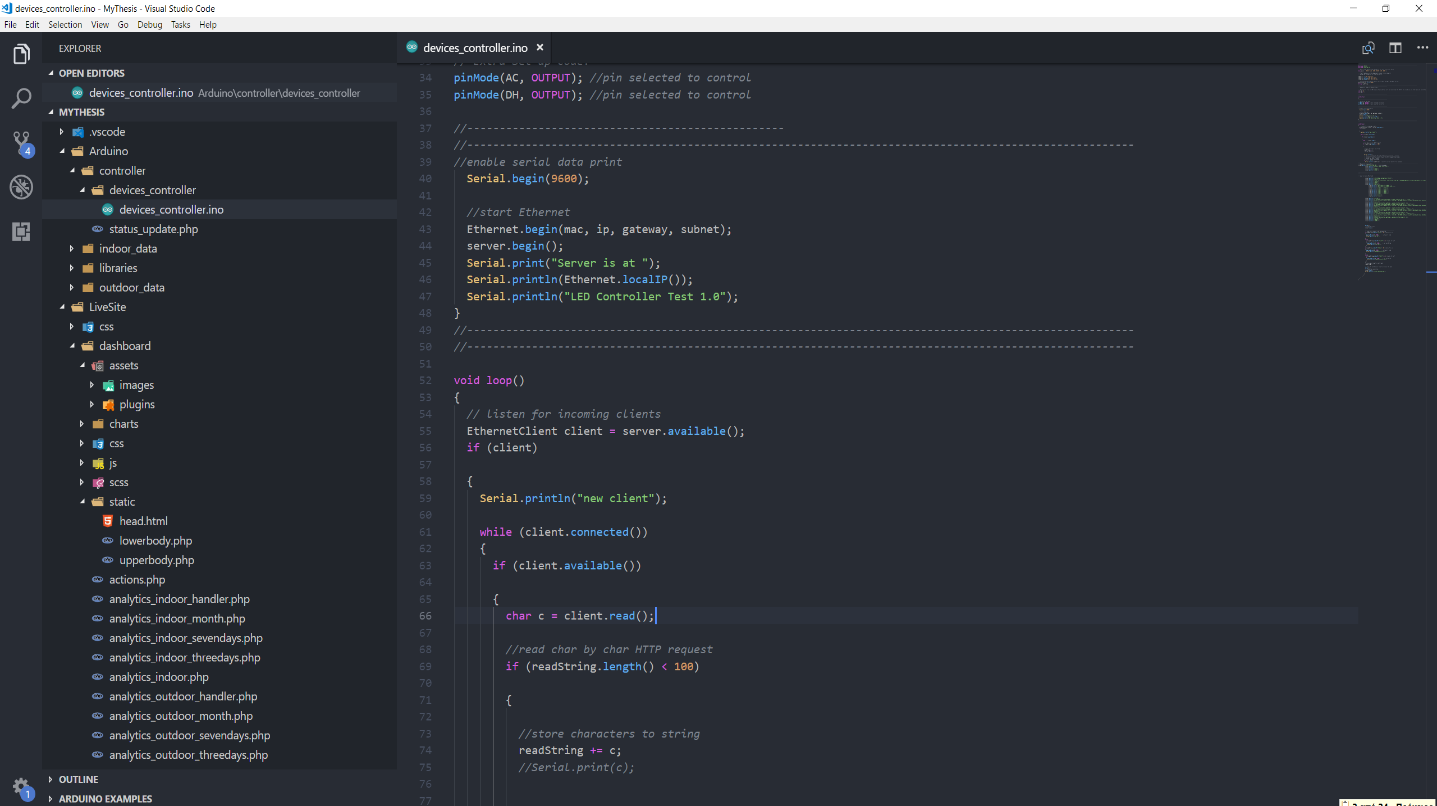
Για την ανάπτυξη λογισμικού, διαδικτυακών εφαρμογών καθώς και για αυτοματισμούς οι προγραμματιστές χρησιμοποιούν διάφορες πλατφόρμες πάνω στις οποίες επεξεργάζονται τον πηγαίο κώδικα ενός project, προσομοιώνουν τοπικούς διακομιστές ή χρήστες αλλά και εξετάζουν τα θέματα ασφαλείας που μπορεί να έχει ο κώδικας που έχουν αναπτύξει. Έτσι λοιπόν, οι πλατφόρμες ανάπτυξης λογισμικού είναι αναπόσπαστο κομμάτι της δουλειάς ενός προγραμματιστή. Αυτό δεν σημαίνει σε καμία περίπτωση πώς οι προγραμματιστές είναι αναγκασμένοι να επιλέξουν μία πλατφόρμα πάνω στην οποία θα βασίσουν το project τους και να μην χρησιμοποιήσουν οποιαδήποτε άλλη. Αντίθετα, η πολυπλοκότητα αλλά και οι αυξημένες απαιτήσεις που έχουν οι πελάτες, οδηγούν τους προγραμματιστές στο να χρησιμοποιήσουν και μερικές φορές να ενοποιήσουν διάφορες πλατφόρμες με σκοπό την ομαλή λειτουργία του project τους.

Στα πλαίσια της παρούσας διπλωματικής εργασίας, χρησιμοποιήθηκαν 4 διαφορετικές πλατφόρμες ανάπτυξης λογισμικού. Επιλέχθηκαν ανάλογα με τα χαρακτηριστικά τους, τις δυνατότητες που έχουν καθώς και το πεδίο εφαρμογής το οποίο απασχολούν.

* 1. 1. **Visual Studio Code**

Το Visual Studio Code είναι ένας επεξεργαστής πηγαίου κώδικα ο οποίος αναπτύχθηκε από την Microsoft και υποστηρίζεται σε λειτουργικά συστήματα όπως είναι τα Windows, τα Linux αλλά και τα macOS. Περιλαμβάνει υποστήριξη για debugging, ενσωματωμένο έλεγχο του Git, επισήμανση συντακτικού, έξυπνη αυτοσυμπλήρωση κώδικα και αποσπάσματα κώδικα. Η δημιουργία του Visual Studio Code έχει βασιστεί στην βιβλιοθήκη Electron που έχουμε αναφέρει παραπάνω και παρουσιάστηκε πρώτη φορά από την Microsoft τον Απρίλιο του 2015. Παρέχει υποστήριξη σε πάρα πολλές γλώσσες προγραμματισμού, οι πιο γνωστές εκ των οποίων είναι οι C και C++, C#, Go,HTML,CSS,JavaScript,TypeScript,PHP,Python,R και Swift [24]. Το Visual Studio Code, αποτέλεσε τον βασικό κινητήριο μοχλό της παρούσας διπλωματικής εργασίας καθώς υποστηρίζει όλες τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν και αρκετά κομμάτια από τον κώδικα αναπτύχθηκαν στον συγκεκριμένο επεξεργαστή κειμένου.

Βασικά πλεονεκτήματα που διαθέτει ο συγκεκριμένος επεξεργαστής κειμένου σε σχέση με άλλα λογισμικά είναι:

* **Διαθεσιμότητα σε γνωστά λειτουργικά**: Το Visual Studio Code υποστηρίζει λειτουργικά όπως macOS, Linux, και Windows. Καθίσταται λοιπόν μία σημαντική επιλογή για κάποιον ο οποίος θέλει να αναπτύξει εφαρμογές σε cross-platform επίπεδο.
* **Επεξεργασία, ανάπτυξη και αποσφαλμάτωση με ευκολία**: Με την ταχεία αλλά και την ογκώδη υποστήριξη πολλών γλωσσών προγραμματισμού, το Visual Studio Code δίνει την δυνατότητα στους χρήστες του να είναι πιο παραγωγικοί με τα χαρακτηριστικά που προσφέρει ανάλογα με την γλώσσα που επεξεργάζεται εκείνη την στιγμή. Διαθέτει αρκετές συντομεύσεις μέσα από τις οποίες μπορεί να αυξηθεί σε μεγάλο βαθμό η παραγωγικότητα και να μειωθεί ο χρόνος εκτέλεσης με αποτέλεσμα την μείωση τους κόστους.
* **Εύκολη παραμετροποίηση:** Κάθε χρήστης – προγραμματιστής ανάλογα με την εμπειρία αλλά και τις προτιμήσεις του, είναι πιθανό να χρειάζεται ένα διαφορετικό περιβάλλον εργασίας. Το Visual Studio Code, δίνει λύση σε αυτό το ζήτημα με τις απεριόριστες παραμετροποιήσεις που επιτρέπει στον χρήστη να κάνει καθώς και με τα διάφορα πρόσθετα που δημιουργούνται καθημερινά από την κοινότητα των προγραμματιστών.
* **Εστίαση στην ανάπτυξη περιεχομένου στο διαδίκτυο:** Επιλέγεται κυρίως ως πλατφόρμα ανάπτυξη περιεχομένου από web developers. Αυτό γίνεται για τον λόγο του ότι η Microsoft επενδύει καθημερινά στην out-of-the-box υποστήριξη του Node.js και των γλωσσών JavaScript και TypeScript που απασχολούν την πλειοψηφία των διαδικτυακών τόπων & εφαρμογών [25].

Σχήμα 2.17 Περιβάλλον Προγραμματισμού Visual Studio Code

* + 1. **Brackets**

****Το Brackets είναι ένα νέο, ανοικτού κώδικα επεξεργαστής κειμένου ο οποίος αντιλαμβάνεται τα ‘κενά’ στην ανάπτυξη εφαρμογών του διαδικτύου και προσφέρει λύσεις. Την παρούσα στιγμή κατά την οποία συγγράφεται η διπλωματική εργασία το Brackets βρίσκεται στην έκδοση 1.13 και οι δυνατότητες που προσφέρει το καθιστούν αξιόλογο για την ανάπτυξη του κώδικα μια διαδικτυακής εφαρμογής επάνω σε αυτό [26]. Δημιουργήθηκε από την Adobe Systems το 2014 και μέχρι σήμερα έχει γνωρίσει αρκετή ανάπτυξη. Πλέον το Brackets υποστηρίζει όλα τα γνωστά λειτουργικά συστήματα και ο πηγαίος κώδικας του συντηρείται σε ένα open source project στο GitHub.

Σχήμα 2.18 Περιβάλλον Προγραμματισμού Brackets

Βασικά πλεονεκτήματα που διαθέτει ο συγκεκριμένος επεξεργαστής κειμένου σε σχέση με άλλα λογισμικά είναι:

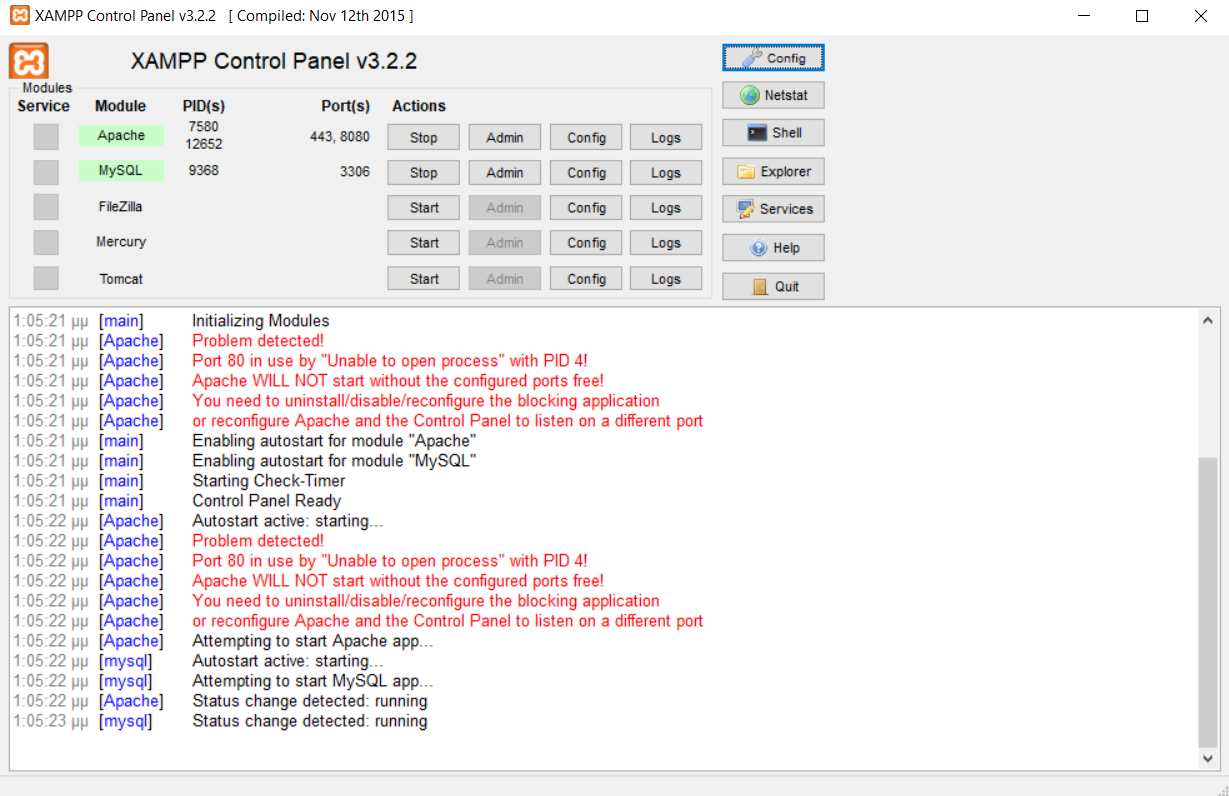
* **Εσωτερικοί επεξεργαστές:** Το Brackets δίνει την δυνατότητα να υπάρχει ένα παράθυρο ανοικτό μέσα σε άλλα παράθυρα για την καλύτερη επεξεργασία του κώδικα. Διαθέτει συγκεκριμένες συντομεύσεις μέσω των οποίων αυξάνεται η παραγωγικότητα του προγραμματιστή.
* **Ζωντανή προεπισκόπηση:** Με αυτό το χαρακτηριστικό μπορεί να υπάρξει ζωντανή σύνδεση με τον φυλλομετρητή (browser). Μπορούν να εκτελεστούν δυναμικές αλλαγές οι οποίες αυτόματα αναπαράγονται στην οθόνη χωρίς να χρειάζεται να κάνουμε χειροκίνητα ανανέωση στον φυλλομετρητή.
* **Υποστήριξη προεπεξεργαστών:** Το Brackets παρέχει υποστήριξη για προεπεξεργαστές κώδικα όπως είναι οι LESS και SCSS δίνοντας έτσι την δυνατότητα στους προγραμματιστές να έχουν ένα ενοποιημένο λογισμικό για την ανάπτυξη του κώδικα τους.
  + 1. **ΧAMPP**

To Xampp είναι ένα τοπικός διακομιστής που χρησιμοποιήθηκε για την ανάπτυξη του ιστόχωρου – διαδικτυακής εφαρμογής. Πρόκειται για ένα πακέτο προγραμμάτων ελεύθερου λογισμικού. Η ονομασία του οφείλεται στις γλώσσες προγραμματισμού και τον διακομιστή των βάσεων δεδομένων που χρησιμοποιούνται για την προσομοιωμένη φιλοξενία ενός ιστόχωρου σε web server (διακομιστής διαδικτύου) [27].

Η παρούσα διπλωματική βασίστηκε στο λειτουργικό Windows, και με την βοήθεια του XAMPP προσομοιώθηκε ο τοπικός web server. Λόγω περιορισμών, ήταν αναγκαία η αλλαγή της τοπικής θύρας για την λειτουργία της διαδικτυακής εφαρμογής από την θύρα :80 που είναι η προεπιλεγμένη στην θύρα :8080. Το πρωτόκολλο ασφαλείας SSL (Secure Sockets Layer) τοποθετήθηκε στην θύρα :443 που είναι και η προεπιλεγμένη και η βάση δεδομένων MySQL τοποθετήθηκε στην θύρα :3306. Δεν ήταν απαραίτητη η χρήση των πλήρη χαρακτηριστικών του XAMPP και έτσι δεν χρησιμοποιήθηκαν οι προσομοιώσεις του FileZilla, του Tomcat και του Mercury.

Το Xampp δημιουργείται από τα αρχικά:

* X: Cross platform- Σημαίνει ότι υποστηρίζει οποιοδήποτε λογισμικό από OS, W (Windows), M (Macintosh) και L (Linux).
* A: Apache server. To Xampp περιλαμβάνει Apache Server.
* M: Maria-DB. To Xampp υποστηρίζει ΜySQL.
* P: PHP. To Xampp υποστηρίζει PHP.
* P: Perl. To Xampp υποστηρίζει Perl.

****Σχήμα 2.19 Πίνακας Ελέγχου του XAMPP

* + 1. **Πλατφόρμα ανάπτυξης λογισμικού Arduino IDE**

Το Arduino IDE (Integrated Development Environment) είναι πολλαπλού λειτουργικού (multi-platform) και αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης, το οποίο σε αντίθεση με άλλα παρόμοια περιβάλλοντα ανάπτυξης όπως το Brackets και το Visual Studio είναι απλούστερο και πιο φιλικό προς το χρήστη.

Περιέχει έναν επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλμάτωση, συνδέτη, σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές. Sketch ονομάζεται το σχεδιάγραμμα που χρησιμοποιεί το Arduino για το εκάστοτε πρόγραμμα. Είναι η μονάδα κώδικα που φορτώνεται και εκτελείται στην πλακέτα Arduino.

Συνδέεται το hardware μέρος του Arduino για να φορτώσει-μεταφέρει προγράμματα και να επικοινωνεί μαζί τους. Η σύνδεση αυτή πραγματοποιείται με μια από τις θύρες USB του υπολογιστή και λόγω του ελεγκτή Serial-over-USB, θα πρέπει να αναγνωριστεί από το λειτουργικό σύστημα ως εικονική σειριακή θύρα.

****Σχήμα 2.20 Λογισμικό Arduino IDE

Οι συντομεύσεις αλλά και τα κουμπιά που περιέχει το περιβάλλον εργασίας του Arduino IDE είναι αρκετά εύχρηστα και προσφέρουν γρήγορες λύσεις στα προβλήματα που προκύπτουν κατά την διαδικασία του προγραμματισμού. Στον παρακάτω πίνακα επισημαίνονται οι επίσημες ορολογίες των κουμπιών αλλά και η περιγραφή της λειτουργίας τους.

***Πίνακας 2.6: Εργαλεία ανάπτυξης Arduino IDE***

|  |  |
| --- | --- |
| **Εργαλείο** | **Πειγραφή λειτουργίας** |
| Verify | Ελέγχει για συντακτικά λάθη στον κώδικα. |
| Upload | Μεταγλωττίζει τον κώδικα και τον φορτώνει στο Arduino. Αν δεν είναι συντακτικά σωστός δεν μπορεί να γίνει η φόρτωση. |
| New | Δημιουργεί ένα νέο sketch. |
| Open | Παραθέτει ένα μενού με όλα τα sketch. Ενεργοποιώντας ένα από αυτά, θα ανοίξει αυτόματα στο τρέχον παράθυρο. |
| Save | Αποθηκεύει ένα sketch. |
| Serial Monitor | Ανοίγει την σειριακή οθόνη και μέσω αυτής παρακολουθείται η ανταλλαγή δεδομένων που γίνεται στην σειριακή θύρα. |

Η γλώσσα συγγραφής κώδικα του Arduino, βασίζεται στη γλώσσα Wiring, μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C, καθώς και μερικά χαρακτηριστικά της C++. Ως compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Ακόμη υπάρχουν ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν τη διαχείριση του ειδικού hardware του Arduino. Στον παρακάτω πίνακα παρουσιάζονται οι βασικές δομές και εντολές του IDE.

***Πίνακας 2.7: Βασικές δομές & εντολές - Arduino IDE***

|  |  |
| --- | --- |
| **Βασικές δομές /εντολές** | **Περιγραφή λειτουργίας** |
| if…else | Δομή επιλογής |
| switch case | Δομή επιλογής που εξαρτάται από την τιμή μιας μεταβλητής. Σε κάθε περίπτωση χρησιμοποιούμε την εντολή break για να τερματίσει η δομή. |
| for | Δομή επανάληψης με γνωστό αριθμό επαναλήψεων που ορίζεται στην αρχή του βρόγχου. |
| while | Δομή επανάληψης, ο βρόγχος εκτελείται συνέχεια έως ότου η συνθήκη που ακολουθεί τη while να γίνει ψευδής (false). |
| do…while | Δομή επανάληψης, ο βρόγχος εκτελείται τουλάχιστον μια φορά έως να γίνει έλεγχος της συνθήκης στο τέλος του βρόγχου. |
| break | Χρησιμοποιείται για την έξοδο από ένα βρόγχο (for, while,do...while) ή από μια switch...case παρακάμπτοντας την κανονική εκτέλεση των εντολών. |
| continue | Χρησιμοποιείται για την παράκαμψη μιας επανάληψης ενός βρόγχου και μεταπήδηση στην επόμενη επανάληψη. |
| return | Τερματίζει μια συνάρτηση ή επιστρέφει μια τιμή σε μια συνάρτηση που έχει κληθεί (τερματίζοντας την). |
| .goto | Μεταφέρει τη ροή του προγράμματος σε ένα άλλο σημείο χρησιμοποιώντας ετικέτες. |
| ; | Χρησιμοποιείται στο τέλος μιας εντολής. |
| {} | Περικλείουν ένα σύνολο εντολών. Ένα άγκιστρο ({) όταν ανοίγει θα πρέπει κάπου να κλείνει. |
| // | Εισάγεται από τον προγραμματιστή για σχολιασμούς και ευκολία ανάγνωσης του sketch. |
| #define | Χρησιμοποιείται για τη δήλωση μεταβλητών στην αρχή του sketch. Οι τιμές αυτών των μεταβλητών δεν θα αλλάξουν σε όλη τη διάρκεια της εκτέλεσης του sketch. |
| #include | Χρησιμοποιείται για τη δήλωση διάφορων βιβλιοθηκών στο sketch. |
| delay(x) | Σταματά προσωρινά την ροή του προγράμματος για time ms. Η παράμετρος time είναι unigned long (από 0 ως 232). Παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από Interrupt θα εκτελεστούν κανονικά κατά τη διάρκεια μιας delay. |
| millis | Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λόγω του τύπου μεταβλητής (unsigned long δηλ. 32bit) θα συμβεί υπερχείλιση σε 232 ms δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν. |

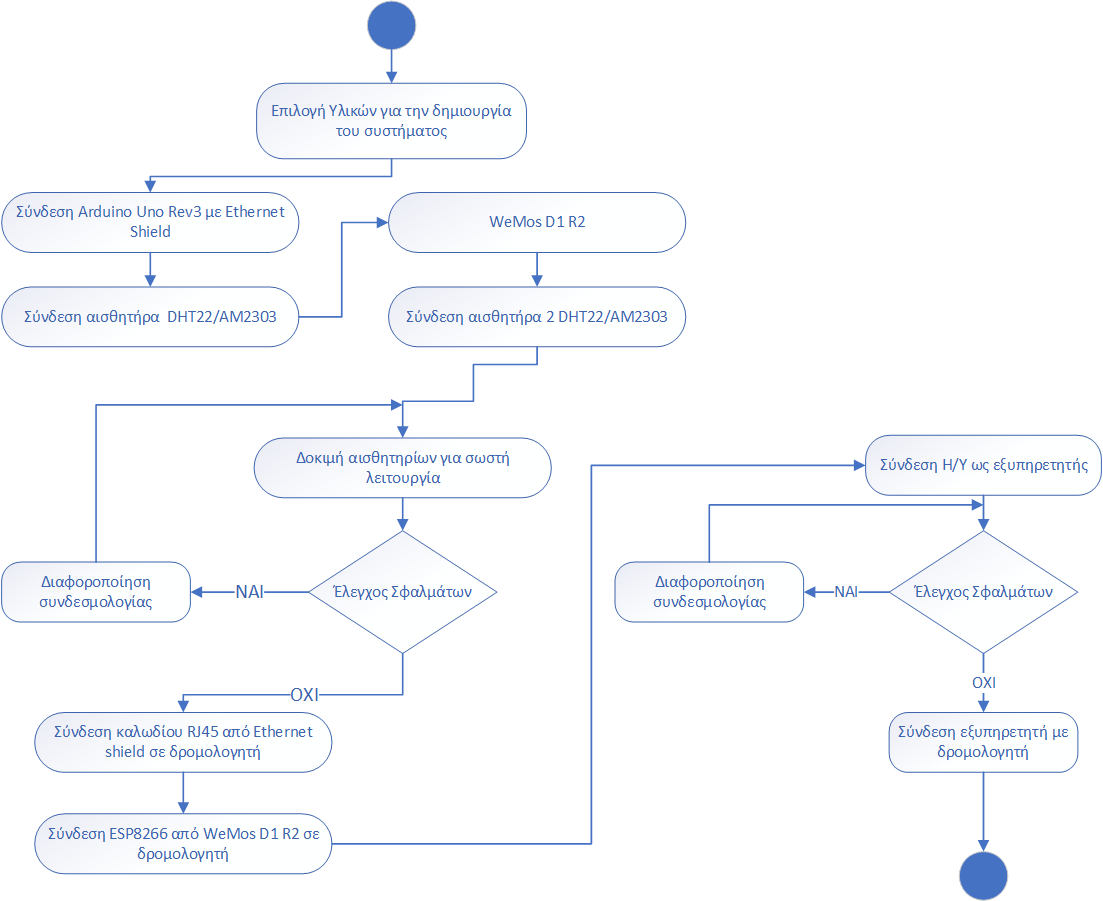
Κεφάλαιο 3

**Υλικό μέρος και συνδεσμολογία συστήματος**

Ένα από τα δύο σημαντικά μέρη της διπλωματικής εργασίας, όπως αναφέρθηκε παραπάνω είναι τα υλικά μέρη που χρησιμοποιήθηκαν. Απαραίτητη προϋπόθεση για την εγκυρότητα και αμεσότητα των δεδομένων προς εμφάνιση στην διαδικτυακή εφαρμογή έχει ο τρόπος με τον οποίο γίνεται η σύνδεση των υλικών καθώς και ο σωστός προγραμματισμός τους. Η συνδεσμολογία της διαδικτυακής εφαρμογής βασίστηκε στους μικροελεγκτές Arduino Uno Rev3 και στον WeMos D1 R2. Η αρχιτεκτονική και τα χαρακτηριστικά των οποίων αναλύθηκαν στο προηγούμενο κεφάλαιο. Έπειτα έγινε η απαραίτητη δρομολόγηση των στατικών IP και αργότερα έγινε η λήψη των δεδομένων από τους αισθητήρες. Εξαιτίας της αρχιτεκτονικής των μικροελεγκτών που χρησιμοποιήθηκαν η οποία είναι ανοικτού κώδικα, με τις κατάλληλες τροποποιήσεις στον προγραμματισμό είναι ικανή η σύνδεση με πληθώρα συσκευών όπως είναι διάφορες πλακέτες, αισθητήρες αλλά και κινητήρες. Έτσι, με την σωστή επιλογή των συσκευών αυτών μπορούν να επιτευχθούν διάφορα project.

Στην παρούσα διπλωματική εργασία, χρησιμοποιήθηκε ένα όχι και τόσο πολύπλοκο σύστημα το οποίο προσφέρει αρκετές επιλογές καθώς και επεκτασιμότητα εάν αυτή κριθεί αναγκαία. Οι αισθητήρες που χρησιμοποιήθηκαν μαζί με τους μικροελεγκτές, μας δίνουν την δυνατότητα παρακολούθησης των μετεωρολογικών φαινομένων ενός σπιτιού (στους εσωτερικούς αλλά και εξωτερικούς χώρους) και η δυνατότητα εναλλαγής των καταστάσεων στις έξυπνες συσκευές του σπιτιού (ενεργοποίηση/απενεργοποίηση). Σημαντικά χαρακτηριστικά της διαδικτυακής εφαρμογής είναι, η ζωντανή σε χρόνο αναμετάδοση δεδομένων από τους αισθητήρες προς τον χρήστη, η δημιουργία στατιστικών για την περαιτέρω λήψη αποφάσεων, η δυνατότητα εναλλαγής κατάστασης των συσκευών σε ζωντανό χρόνο καθώς και η ασφαλής λήψη των δεδομένων από τρίτους.

Στις ενότητες που ακολουθούν γίνεται μία εκτενής παρουσίαση όλων των εξαρτημάτων που συνετέλεσαν στην δημιουργία του συστήματος, την ανάλυση των τεχνικών χαρακτηριστικών και τον τρόπο λειτουργίας τους αλλά και στην ορθή σύνδεση αυτών. Παραθέτονται εικόνες και σχηματικές απεικονίσεις για τον τρόπο σύνδεσης των μικροελεγκτών και των αισθητηρίων οργάνων. Έπειτα αναλύεται και εξηγείται η λειτουργία του κώδικα βάσει του οποίου πραγματοποιείται η λήψη δεδομένων από την συσκευή. Τέλος, υπολογίζεται το κόστος της ολοκληρωμένης κατασκευής.

Το διάγραμμα ροής της διαδικασίας της συνδεσμολογίας που ακολουθεί στο σχήμα 3.1 περιγράφει συνοπτικά τα βήματα που εκτελέστηκαν για την υλοποίηση του συστήματος.

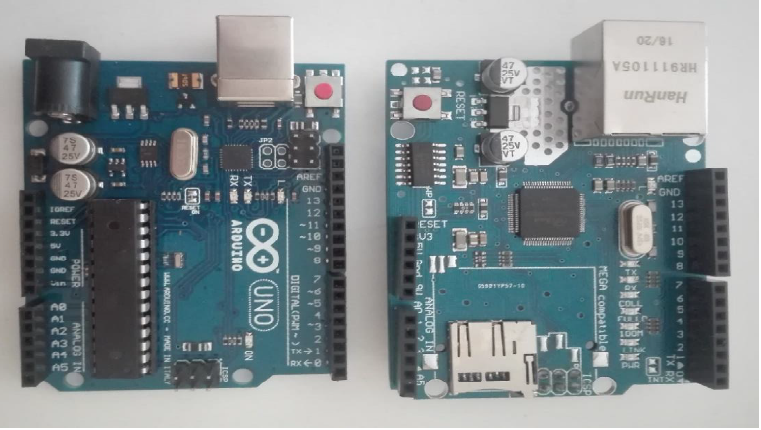
Σχήμα 3.1 Διάγραμμα ροής συνδεσμολογίας εξαρτημάτων

1. 1. **Hardware Υλοποίησης Συστήματος**

Η παρούσα διπλωματική εργασία έχει σκοπό, την δημιουργία ενός συστήματος μέσω του οποίου ο κάθε άνθρωπος θα μπορεί να έχει τον έλεγχο των έξυπνων συσκευών που έχει τοποθετήσει στο σπίτι του αλλά ταυτόχρονα με όσο το δυνατό μικρότερο κόστος. Έτσι λοιπόν, ο χρήστης θα πρέπει να προμηθευτεί τα βασικά εξαρτήματα που είναι τα εξής:

* Μικροελεγκτής Arduino Uno Rev3
* Ασπίδα Arduino Ethernet Shield
* Μικροελεγκτής WeMos D1 R2
* 2x Αισθητήρας θερμοκρασίας DHT22/AM2302
* 2x Πλακέτα δοκιμών (Breadboard 400)

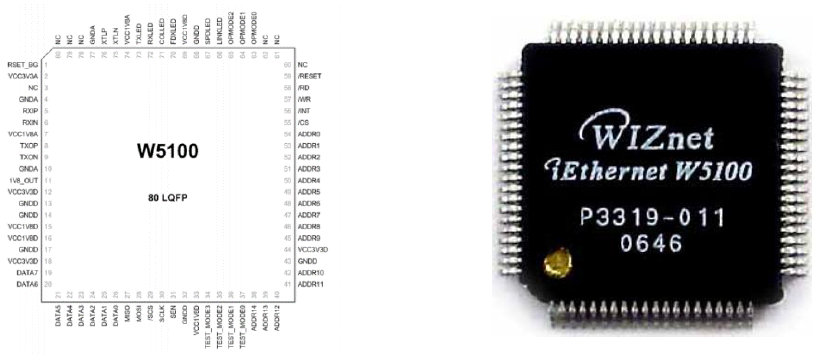
To σύνολο του συστήματος θα πρέπει να συμπληρωθεί από καλώδια σύνδεσης (jumper cables) καθώς και έναν δρομολογητή (router). Στις επόμενες ενότητες εξηγούνται πιο αναλυτικά το κάθε υλικό και ο τρόπος λειτουργίας αυτών.

1. 
   1. 1. **Arduino Uno Rev3 – Arduino Ethernet Shield**

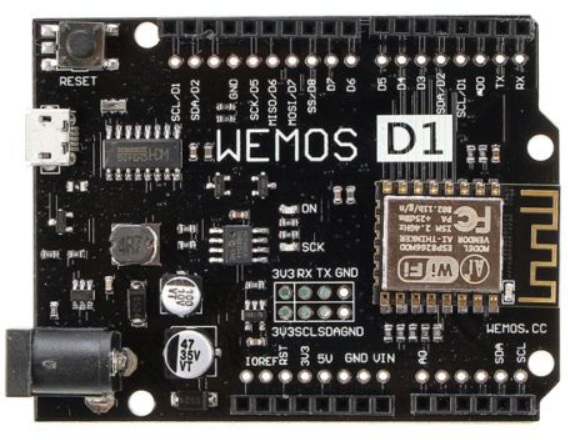
Σχήμα 3.2 Arduino Uno Rev3 και Ethernet Shield

Ξεκινώντας την συλλογή δεδομένων θερμοκρασίας και υγρασίας εντός του χώρου του σπιτιού κρίνεται απαραίτητη η δημιουργία ενός υποσυστήματος. Βασικά υλικά μέρη αυτού είναι ο μικροελεγκτής Arduino Uno Rev3 καθώς και το Ethernet Shield τα οποία φαίνονται στο σχήμα 3.2. Το υποσύστημα συμπληρώνουν ένας αισθήτηρας θερμοκρασίας/υγρασίας DHT22/AM2302 και μία πλακέτα δοκιμών Breadboard 400 οι οποίες θα περιγραφούν σε παρακάτω κεφάλαιο. Η χρήση αυτού του υποσυστήματος αποσκοπεί στην ενσύρματη μεταφορά των συλλεγόμενων δεδομένων στον τοπικό εξυπηρετητή και στη βάση δεδομένων μέσω καλωδίου Ethernet.

Τα τεχνικά χαρακτηριστικά των δύο αυτών πλακετών έχουν παρουσιαστεί αναλυτικά στο προηγούμενο κεφάλαιο. Στην ενότητα αυτή πραγματοποιείται η περιγραφή της λειτουργίας τους.

Η συλλογή των δεδομένων από τον μικροελεγκτή Arduino Uno Rev3 γίνεται μέσω της συνδεσμολογίας που έχει πραγματοποιηθεί μεταξύ αυτού και του αισθητήρα θερμοκρασίας/υγρασίας. Έπειτα ο μικροελεγκτής καθώς έχει συνδεθεί μέσω του δρομολογητή στον τοπικό εξυπηρετητή στέλνει τα δεδομένα που έχει συλλέξει μέσω του Ethernet Shield. Το τελευταίο, επιτελεί έναν πολύ σημαντικό ρόλο, καθώς μέσω αυτού γίνεται η ενσύρματη αποστολή δεδομένων στον τοπικό εξυπηρετητή μέσω του καλωδίου RJ45 Ethernet. Μπορεί να χρησιμοποιηθεί είτε σαν server είτε σαν client και υποστηρίζεται άμεσα από την επίσημη Ethernet βιβλιοθήκη. Χρησιμοποιώντας τη βιβλιοθήκη αυτή, είναι δυνατή η συγγραφή των sketches για την σύνδεση της κατασκευής στον τοπικό εξυπηρετητή. Όπως αναφέρθηκε και προηγουμένως βασίζεται στον ethernet ελεγκτή W5100 της Wiznet.

Σχήμα 3.3 WIZnet Ethernet W5100 Chipset & Schematic Diagram

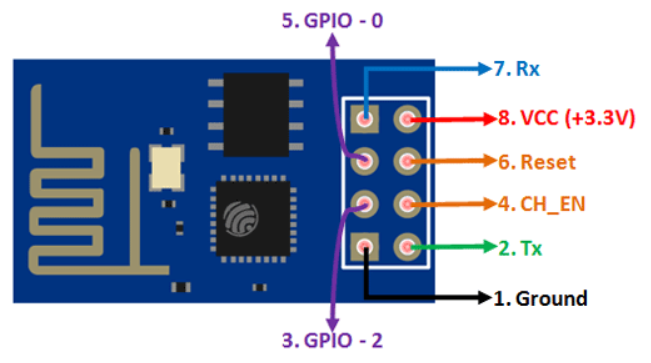
* + 1. **WeMos D****1 R2**

Σχήμα 3.4 WeMos D1 R2

Συνεχίζοντας την συλλογή δεδομένων θερμοκρασίας και υγρασίας εκτός του χώρου του σπιτιού κρίνεται απαραίτητη η δημιουργία ενός δεύτερου υποσυστήματος. Βασικά υλικά μέρη αυτού είναι ο μικροελεγκτής WeMos D1 R2 και το microchip ESP8266 τo οποίo είναι ενσωματωνμένο στην πλακέτα του μικροελεγκτή. Η πλακέτα φαίνεται στο σχήμα 3.3. Το υποσύστημα συμπληρώνουν ένας αισθήτηρας θερμοκρασίας/υγρασίας DHT22/AM2302 και μία πλακέτα δοκιμών Breadboard 400 οι οποίες προαναφέραμε πως θα περιγραφούν σε παρακάτω κεφάλαιο. Η χρήση αυτού του υποσυστήματος αποσκοπεί στην ασύρματη μεταφορά των συλλεγόμενων δεδομένων στον τοπικό εξυπηρετητή και στη βάση δεδομένων μέσω πρωτόκολλο WiFi.

Τα τεχνικά χαρακτηριστικά της πλακέτας έχουν παρουσιαστεί αναλυτικά στο προηγούμενο κεφάλαιο. Στην ενότητα αυτή πραγματοποιείται η περιγραφή της λειτουργίας τους.

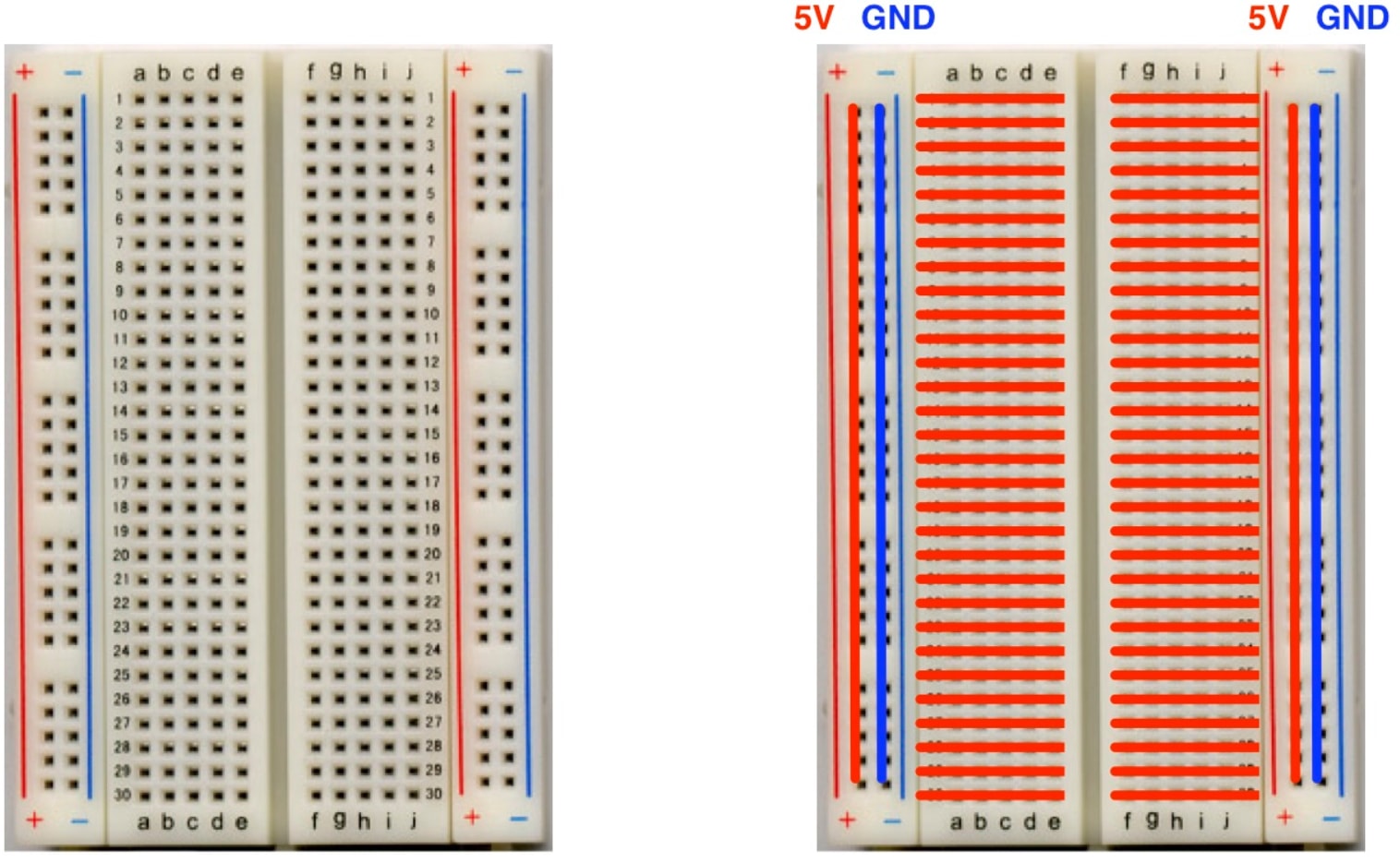
Η συλλογή των δεδομένων από τον μικροελεγκτή WeMos D1 R2 γίνεται μέσω της συνδεσμολογίας που έχει πραγματοποιηθεί μεταξύ αυτού και του αισθητήρα θερμοκρασίας/υγρασίας. Έπειτα ο μικροελεγκτής καθώς έχει συνδεθεί μέσω του δρομολογητή στον τοπικό εξυπηρετητή στέλνει τα δεδομένα που έχει συλλέξει μέσω του WiFi. Το microchip υποστηρίζεται από μεγάλη κοινότητα καθώς και από αρκετές βιβλιοθήκες οι οποίες έχουν συγγραφεί για τις διαφορετικές λειτουργίες του ESP8266. Χρησιμοποιώντας τις βιβλιοθήκες αυτές, είναι δυνατή η συγγραφή των sketches για την σύνδεση της κατασκευής στον τοπικό εξυπηρετητή.

Σχήμα 3.5 ESP8266 microchip Schematic

* + 1. **Πλακέτα δοκιμών (Breadboard 400)**

Η πλακέτα δοκιμών είναι ένα από τα βασικά κομμάτια και των δύο υποσυστημάτων που χρησιμοποιήθηκαν για την παρούσα διπλωματική εργασία καθώς πάνω σε αυτή βασίζεται η δημιουργία του κυκλώματος και η ταχεία και ασφαλή αποστολή των δεδομένων από τον αισθητήρα προς τον μικροελεγκτή. Τονίζουμε πως η πλακέτα δοκιμών που χρησιμοποιήσαμε είναι μια πλακέτα solderless breadboard η οποία διαθέτει 400 pins δίνοντας μας έτσι την δυνατότητα να μπορούν να γίνουν εύκολα τροποποιήσεις στην συνδεσμολογία καθώς δεν υπάρχουν κολλημένα μέρη.

Τα υλικά μέρη πέρα του μικροελεγκτή, τοποθετούνται επάνω στην πλακέτα δοκιμών, στην παρούσα περίπτωση ο αισθητήρας και τα καλώδια σύνδεσης, εισάγοντας τους ακροδέκτες τους στις ειδικές οπές που υπάρχουν στην επιφάνειά του. Οι οπές αυτές απέχουν η κάθε μία μεταξύ τους κατά 2,54mm. Κάτω από κάθε “γραμμή-σειρά” της πλαστικής επιφάνειας, στο εσωτερικό του breadboard, βρίσκονται μεταλλικά κλιπ που επιτρέπουν τη ροή ρεύματος και επιτυγχάνουν την αγωγιμότητά της. Κάθε ένα από αυτά τα κλιπ αντιστοιχεί σε πέντε οριζόντιες οπές που σημαίνει ότι σε κάθε ξεχωριστό τμήμα μπορούν να συνδεθούν μέχρι πέντε εξαρτήματα. Πέρα από τις οριζόντιες γραμμές, στις πλαϊνές πλευρές βρίσκονται και κάθετες γραμμές πανομοιότυπες με τις οριζόντιες με τη μόνη διαφορά, ότι αυτές συνδέονται σε όλο το μήκος και δίνουν ρεύμα οπουδήποτε χρειαστεί στο κύκλωμα. Οι ετικέτες “+” και “-” υποδεικνύουν τις θετικές και αρνητικές πλευρές.

Στο παρακάτω σχήμα εφμανίζεται η πλακέτα δοκιμών καθώς και η συνδέσεις αυτής.

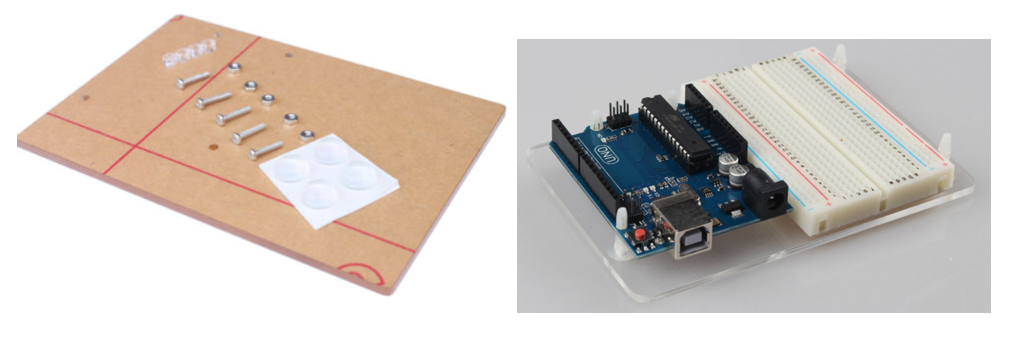
Σχήμα 3.6 Breadboard 400 Schematic

* + 1. **Ακρυλική βάση τοποθέτησης μικροελεγκτή – πλακέτας δοκιμών**

Για την ομαλότερη τοποθέτηση των υποσυστημάτων στους εξωτερικούς και εσωτερικούς χώρους του σπιτιού ήταν προαιρετική η ακρυλική βάση πάνω στην οποία τοποθετήθηκαν τα δύο υποσυστήματα. Η ακρυλική βάση είναι ουσιαστικά ένα κομμάτι από διαφανές πλαστικό πάνω στο οποίο έχουν δημιουργηθεί οπές σύμφωνα με τα πρότυπα του Arduino Uno Rev3 έτσι ώστε να υπάρχει η σωστή εφαρμογή του επάνω στην ακρυλική βάση. Υπάρχουν οι κατάλληλοι κοχλίες οι οποίοι βιδώνουν από το κάτω μέρος της ακρυλικής βάσης, διαπερνάνε τον μικροελεγκτή και ασφαλίζονται με παξιμάδια. Επίσης υπάρχει χώρος, ικανός να φιλοξενήσει επάνω της η βάση και μία πλακέτα δοκιμών breadboard 400.

Σκοπός της βάσης είναι ασφαλής και σταθερή τοποθέτηση του κυκλώματος. Λόγω της φύσης της βάσης (πλαστικό) προστίθεται και ένα επιπλέον μονωτικό επίπεδο έτσι ώστε ο μικροελεγκτής να μην εφάπτεται σε περιοχές που μπορεί να προκαλέσουν βραχυκύκλωμα.

Η ακρυλική βάση, οι κοχλίες και τα παξιμάδια καθώς και το ολοκληρωμένο οπτικό αποτέλεσμα παρουσιάζονται στο επόμενο σχήμα.

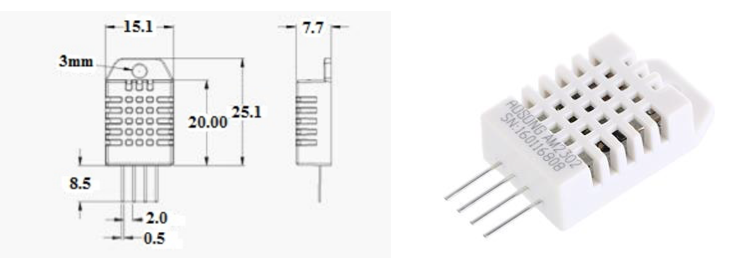


***Σχήμα 3.7 Ακρυλική Βάση***

* + 1. **Αισθητήριο θερμοκρασίας & υγρασίας DHT22/AM2302**

Για την συλλογή των δεδομένων στους χώρους που τοποθετήκαν τα υποσυστήματα χρησιμοποιήθηκε ο ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας DHT22/AM2302. Όσο αφορά την ποιότητα κατασκευής, ο DHT22/AM2302 είναι από πλαστικό σώμα, είναι διάτριτος σε αρκετά σημεία και στο κάτω μέρος του υπάρχουν 4 ακροδέκτες. Λόγω του μικρού μεγέθους του αλλά και του μικρής πλακέτας που φέρει, δικαιολογείται και η χαμηλή σε κόστος αγοράς, τιμή του.

Αποτελεί ένα σύνθετο αισθητήρα που περιέχει μια βαθμολογημένη παραγωγή ψηφιακών σημάτων. Έχει μικρή κατανάλωση ρεύματος, μπορεί να λαμβάνει τάση εισόδου την οποία μετατρέπει σε θερμοκρασία και την υγρασία του χώρου με ακρίβεια σε απόσταση 20 m και να μεταδίδει δεδομένα στην πλακέτα με την οποία έχει συνδεθεί κάθε 2 sec. Βάσει αυτών των ιδιαιτεροτήτων του χρησιμοποιείται σε πολλές εφαρμογές και υπάρχει πληθώρα βιβλιοθηκών στο Διαδίκτυο.



***Σχήμα 3.8 Σχηματική Απεικόνιση & Διαστάσεις του αισθητηρίου DHT22/AM2302***

Σημαντικό είναι να αναφέρουμε τα τεχνικά χαρακτηριστικά του εν λόγω αισθητήρα καθώς πληρεί τα τεχνικά χαρακτηριστικά που έχει θέσει ως πρότυπο ο παγκόσμιος μετεορολογικός οργανισμός (WMO) και έτσι ο DHT22/AM2302 καθίσταται ιδανικός αισθητήρας για τις μετρήσεις που χρειάζονται στην παρούσα διπλωματική εργασία. Τα πρότυπα του παγκόσμιου μετεορολογικού οργανισμού είναι:

* Εύρος τιμών περιβαλλοντικής καταγραφής θερμοκρασίας: -30°C έως +50°C
* Εύρος τιμών περιβαλλοντικής καταγραφής υγρασίας: 5% έως 100%

Τα αναλυτικά τεχνικά χαρακτηριστικά του αισθήτηρα αναλύονται στον παρακάτω πίνακα.

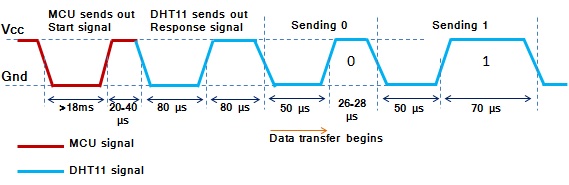
***Πίνακας 3.1: Τεχνικά Χαρακτηριστικά του DHT22/AM2302***

|  |  |  |
| --- | --- | --- |
| **Μοντέλο** | **DHT22** | |
| Τάση λειτουργίας | 3.3-6V DC | |
| Σήμα εξόδου | Ψηφιακό σήμα μέσω μονού-bus | |
| Στοιχείο αίσθησης | Πολυμερής Πυκνωτής | |
| Εύρος λειτουργίας | Υγρασία: 0-100% RH | Θερμοκρασία: -40 −80°C |
| Ακρίβεια | Υγρασία: ±2-5%RH | Θερμοκρασία: ±0,5°C |
| Ανάλυση/Ευαισθησία | Υγρασία: 0,1%RH | Θερμοκρασία: 0,1°C |
| Επαναληψιμότητα | Υγρασία: ±1%RH | Θερμοκρασία: ±0,2°C |
| Υστέρηση υγρασίας | ±0,3%RH | |
| Μακροσταθερότητα | +-0.5%RH/χρόνο | |
| Ανταλλαξιμότητα | Πλήρως έναλλάξιμος | |
| Δειγματοληψία | 2 δευτερόλεπτα | |

Η λειτουργία του αισθητήρα κατά την επικοινωνία του με το Arduino περιγράφεται με τον ακόλουθο τρόπο. Αρχικά, το Arduino μέσω της ψηφιακής του εξόδου στέλνει ένα σήμα έναρξης της τάξης των 500μs περίπου και αναμένει απάντηση από τον αισθητήρα. Στη συνέχεια, ο αισθητήρας μειώνει το σήμα στα 80μs και στέλνει εκ νέου ένα παρόμοιο σήμα. Ταυτόχρονα, στέλνει ένα ακόμα σήμα της τάξης των 50μs (διαδικασία χειραψίας). Τα επόμενα σήματα που ακολουθούν είναι ψηφιακά και κάθε 28μs για λογικό 0 και 70μs για λογικό 1. Κάθε δύο δευτερόλεπτα, ο αισθητήρας εκπέμπει 40 bit πληροφορίας τα οποία αντιστοιχούν σε:

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

H διαδικασία επικοινωνίας του αισθητήρα με τον μικροελεγκτή παρουσιάζεται στο παρακάτω σχήμα.

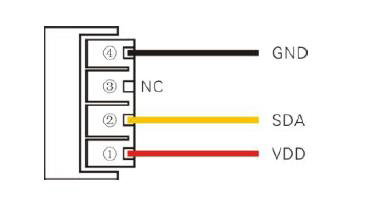


***Σχήμα 3.9 Διαδικασία επικοινωνίας DHT22/AM2302 με Arduinο***

Οι ακροδέκτες (pins) που διαθέτει είναι 4 και οι 3 από αυτούς χρησιμοποιούνται για την καταγραφή των απαραίτητων δεδομένων. Η αντιστοιχία τους καθώς και η λειτουργία καθενός περιγράφεται στον πίνακα και το σχήμα που ακολουθούν.

***Πίνακας 3.2: Περιγραφή λειτουργίας ακροδεκτών DHT22/AM2302***

|  |  |  |
| --- | --- | --- |
| **Ακροδέκτης (Pin)** | **Όνομα** | **Περιγραφή** |
| (1) | VDD | Τροφοδοσία (3.3-6V DC) |
| (2) | SDA | Θύρα σειριακών δεδομένων |
| (3) | NC | Κενός |
| (4) | GND | Γείωση |



***Σχήμα 3.10 Σχηματική Απεικόνιση ακροδεκτών DHT22/AM2302***

* + 1. **Καλώδια Jumper**

Τα καλώδια Jumper είναι κυλινδρικά ηλεκτρικά καλώδια με βύσματα (συνδέτες) ή αγωγούς στην κάθε άκρη τους. Χρησιμοποιούνται για τη σύνδεση εξαρτημάτων μεταξύ τους με ή χωρίς την βοήθεια κάποιου breadboard. Ανάλογα με τις άκρες τους τα καλώδια αυτά μπορεί να είναι αρσενικά (συνδέτες) ή θηλυκά (αγωγοί). Για να πραγματοποιηθούν οι συνδέσεις πάνω στο breadboard στη συγκεκριμένη περίπτωση χρησιμοποιήθηκαν αρσενικά καλώδια.



***Σχήμα 3.11 Καλώδια Jumper***

* + 1. **Καλώδια σύνδεσης πλακετών με Η/Υ**

Για την σωστή λειτουργία του συστήματος της παρούσας διπλωματικής εργασίας, είναι απαιραίτητη η χρήση ορισμένων καλωδίων για την σύνδεση των πλακετών και των ασπίδων με τον Η/Υ και τον τοπικό εξυπηρετητή. Η σύνδεση των μικροελεγκτών Arduino Uno Rev3 καθώς και του WeMos D1 R2 με τον Η/Υ γίνεται με τα καλώδια:

* **WeMos D1 R2 – Η/Υ**: micro USB σε USB
* **Arduino Uno Rev3 – Η/Υ:** USB Type-B σε USB

Τα καλώδια που αναφέρθηκαν παραθέτονται στα παρακάτω σχήματα.



***Σχήμα 3.12 Καλώδιo micro USB σε USB Σχήμα 3.13 Καλώδιo USB Type-B σε USB***

Ακόμα, ένα καλώδιο Ethernet RJ45 για τη σύνδεση της πλακέτας Ethernet shield με το router του εκάστοτε χρήστη, έτσι ώστε να είναι εφικτή η διαδικτυακή παρακολούθηση των δεδομένων που συλλέγονται.

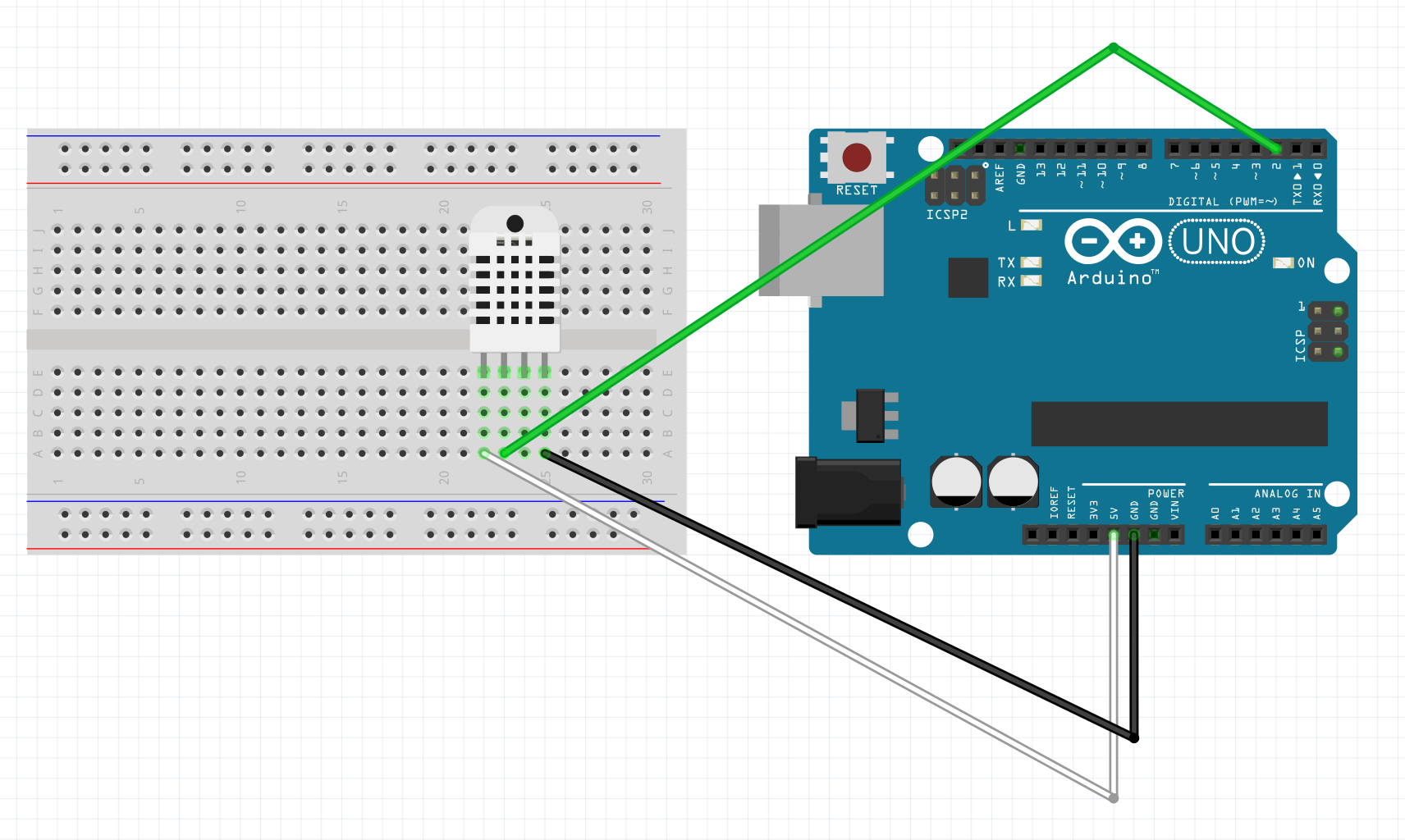


***Σχήμα 3.14 Καλώδιo Ethernet RJ45***

* 1. **Συνδεσμολογία Συστήματος (Hardware Wiring)**

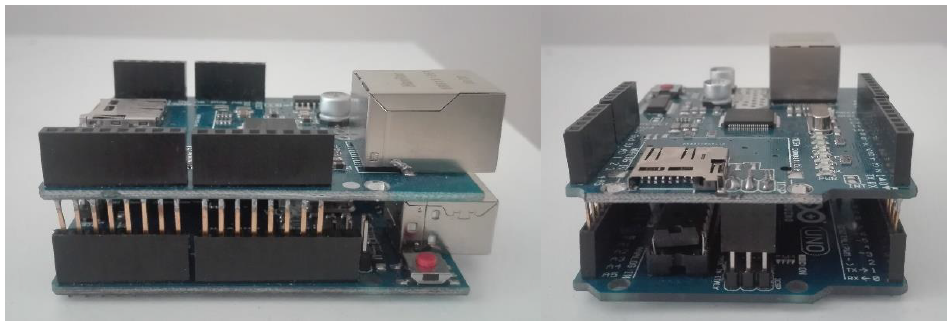
Για την λήψη των δεδομένων, όπως προαναφέρθηκε δημιουργήθηκαν δύο υποσυστήματα, στα οποία υπήρχε από ένας μικροελεγκτής και ένα αισθητήρας θερμοκρασίας/υγρασίας. Στην ενότητα αυτή παρουσιάζεται η βήμα προς βήμα διαδικασία που απαιτείται για τη σύνδεση όλων των επιμέρους υλικών που προαναφέρθηκαν.

Συνδεσμολογία Υποσυστήματος με Arduino Uno Rev3



***Σχήμα 3.15 Σχηματική απεικόνιση σύνδεσης Arduino Uno Rev3 και DHT22***

Αρχικά, στον μικροελεγκτή συνδέεται η πλακέτα Ethernet Shield. Οι ακροδέκτες των δύο συνεργαζόμενων προαναφερθέντων συσκευών έχουν ίδια τοποθέτηση στις πλακέτες, με αποτέλεσμα αυτοί του Ethernet Shield να εφάπτονται με απόλυτη ακρίβεια στις αντίστοιχες υποδοχές του Arduino Uno Rev3 όπως φαίνεται και στο σχήμα 3.15. Για την λειτουργία της πλακέτας Ethernet Shield απαιτείται η σύνδεση των ακροδεκτών 11,12 και 13 του Arduino με τους αντίστοιχους τις πλακέτας. Είναι οι ακροδέκτες που παρέχουν τη σύγχρονη σειριακή επικοινωνία δεομένων (SPI) μεταξύ των μικροελεγκτών που βρίσκονται στην κάθε μια πλακέτα αντίστοιχα. Αξίζει να σημειωθεί ότι, οι ακροδέκτες αυτοί δεν μπορούν να χρησιμοποιηθούν ως είσοδοι ή έξοδοι. Το Ethernet Shield δεν απαιτεί επιπλέον τροφοδοσία, τροφοδοτείται μέσω της πλακέτας στην οποία έχει ενσωματωθεί. Στην παρούσα κατασκευή από 5V dc/ 0,7mA που της παρέχει το Arduino Uno Rev3.

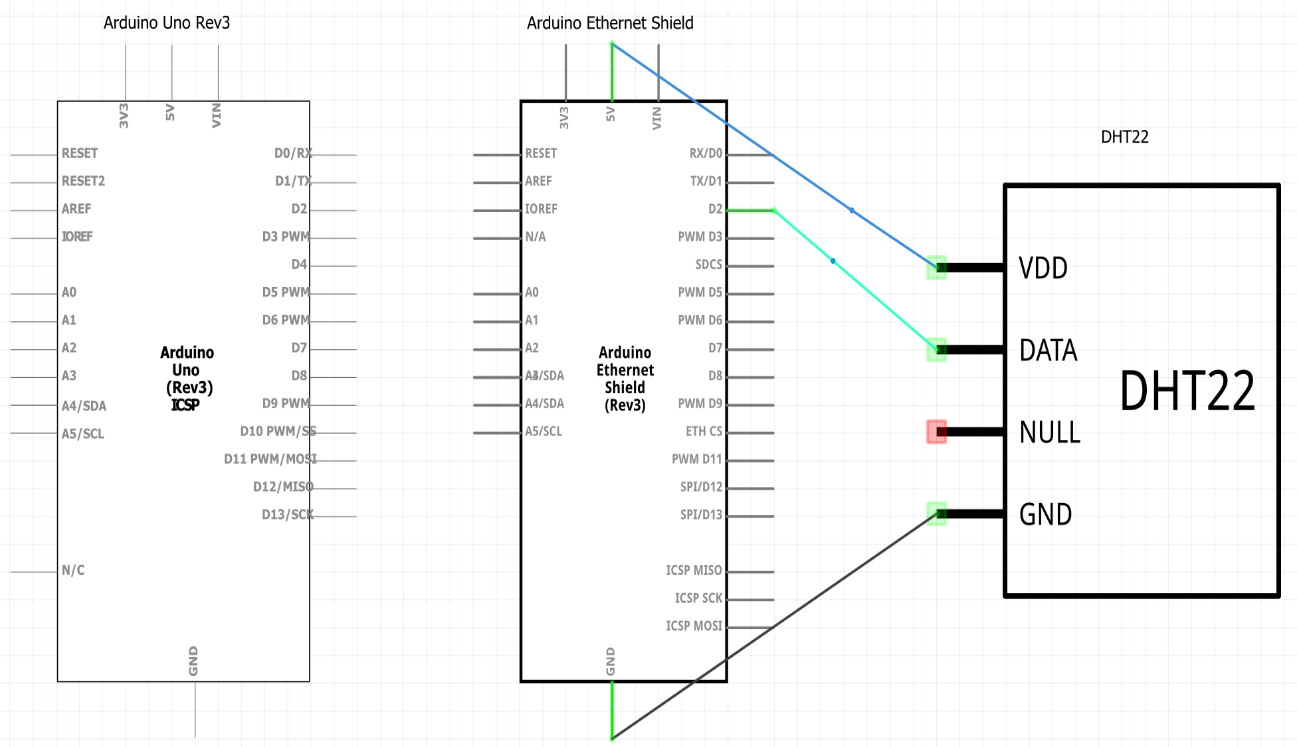


***Σχήμα 3.16 Σύνδεση Arduino Uno Rev3 και Arduino Ethernet Shield***

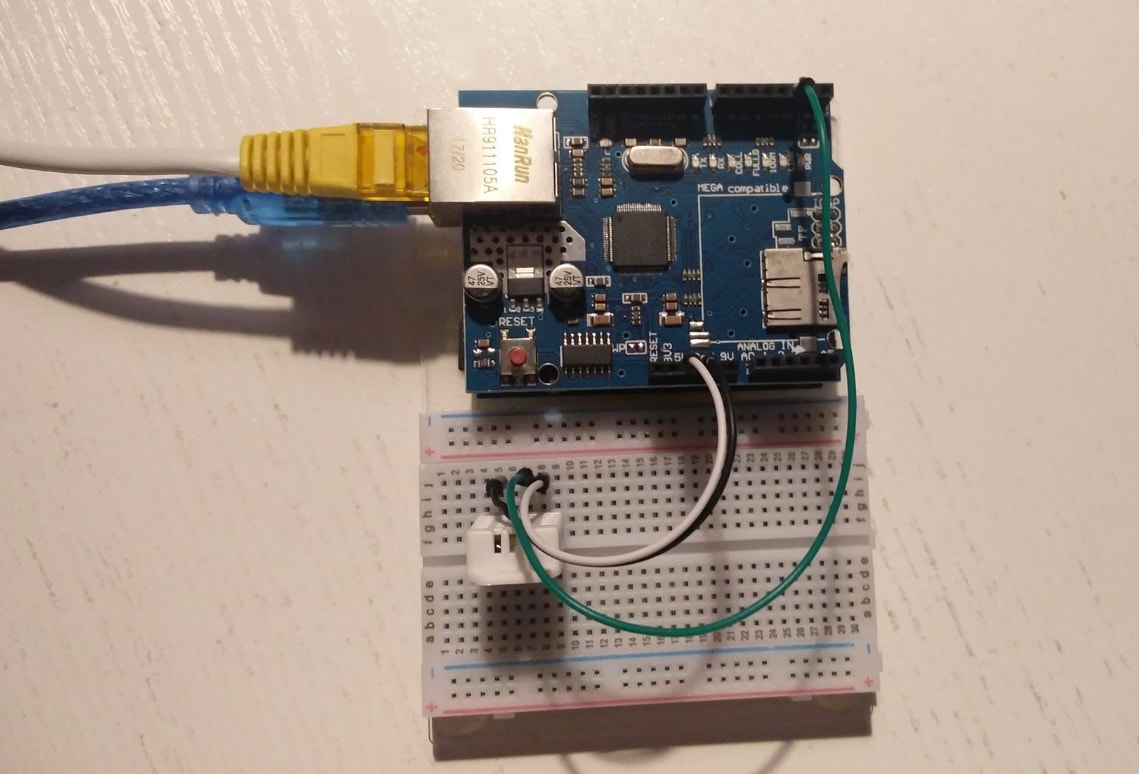
Σειρά έχει η τοποθέτηση του αισθητήρα DTH22/AM2302 πάνω στην πλακέτα δοκιμών καθώς και η καλωδίωση του μέσω των jumper cales και η σύνδεση αυτού με το ethernet shield έτσι ώστε να πραγματοποιηθεί η δημιουργία του πρώτου υποσυστήματος. Όπως τονίσαμε παραπάνω, τρεις από τους τέσσερεις ψηφιακούς ακροδέκτες του αντιστοιχούν σε υποδοχές, καθώς ο ένας είναι κενός. Η σύνδεση του με τον μικροελεγκτή Arduino γίνεται μέσω των υποδοχών της πλακέτας Ethernet Shield χωρίς να υπάρχει καμία διαφορά με την απευθείας σύνδεση πάνω στον πρώτο.

Είναι απαραίτητο να κατανοήσουμε την σωστή συνδεσμολογία που απαιτείται στον αισθητήρα DHT22/AM2302 έτσι ώστε να πάρουμε τα επιθυμητά αποτελέσματα. Πρέπει λοιπόν, να συνδέσουμε σωστά τον κάθε ακροδέκτη του αισθητήρα με κάποιο υποδοχή του μικροελεγκτή. Ξεκινώντας την αρίθμηση των ακροδεκτών από αριστερά προς τα δεξία όπως φαίνεται στο σχήμα 3.9, ο πρώτος ακροδέκτης χρησιμοποιείται για την τροφοδοσία των 5V και συνδέεται, με το άσπρο καλώδιο, στην υποδοχή με την αντίστοιχη υπόδειξη. Ο δεύτερος ακροδέκτης είναι υπεύθυνος για τη μεταφορά των συλλεγόμενων δεδομένων. Αποτελεί ουσιαστικά την ταυτότητα του αισθητήρα στην πλακέτα, στην παρούσα κατασκευή συνδέεται, με το πράσινο καλώδιο, στη ψηφιακή είσοδο της πλακέτας η οποία είναι αριθμημένη ως 2. Τέλος, η χρησιμότητα του τέταρτου ακροδέκτη αφορά την γείωση και συνδέεται, με το μαύρο καλώδιο, σε μια από τις αναγραφόμενες GND υποδοχές. Αξίζει να σημειωθεί σε αυτό το σημείο πώς το χρώμα των καλωδίων που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία δεν παίζουν κάποιο συγκεκριμένο ρόλο και πως ανά πάσα στιγμή μπορεί να γίνει η αλλαγή των καλωδίων με άλλα καλώδια διαφορετικού χρώματος.

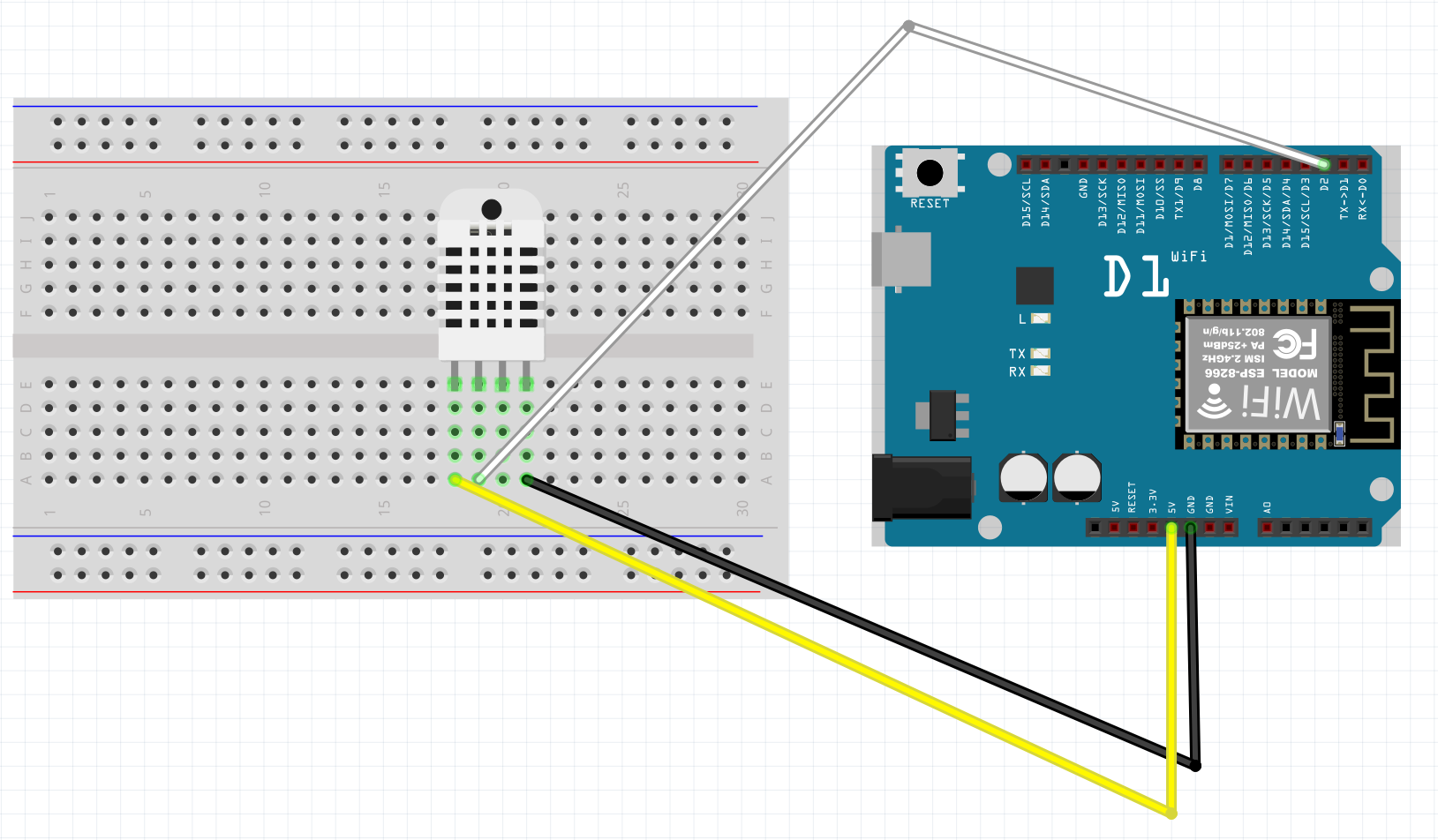
Στο παρακάτω σχήμα φαίνεται το ηλεκτρολογικό σχέδιο σύνδεσης του υποσυστήματος που αναλύσαμε παραπάνω. (Λόγω τεχνικών προβλημάτων στο πρόγραμμα που σχεδιάστηκαν οι απεικονίσεις των συστημάτων, έχει γίνει η αλλαγή του άσπρου καλωδίου με μπλε καλώδιο)



***Σχήμα 3.17 Ηλεκτρολογική σχηματική απεικόνιση συνδεσμολογίας***

Τέλος, είναι απραίτητη η σύνδεση του μικροελεγκτή Arduino Uno Rev3 με τον Η/Υ, μέσω ενός USB καλωδίου, τόσο για τροφοδοσία όσο και για την μεταφορά και συλλογή των δεδομένων, και το ανέβασμα του κώδικα στην πλακέτα. Επίσης πραγματοποιείται η σύνδεση του Ethernet Shield με το router αποτελούν τα τελικά αλλά αρκετά σημαντικά βήματα για την ολοκλήρωση της κατασκευής του πρώτου υποσυστήματος. Στο σχήμα 3.17, παρουσιάζεται η τελική μορφή της συνδεσμολογίας που περιγράφηκε.

***Σχήμα 3.18 Tελική μορφή υποσυστήματος***

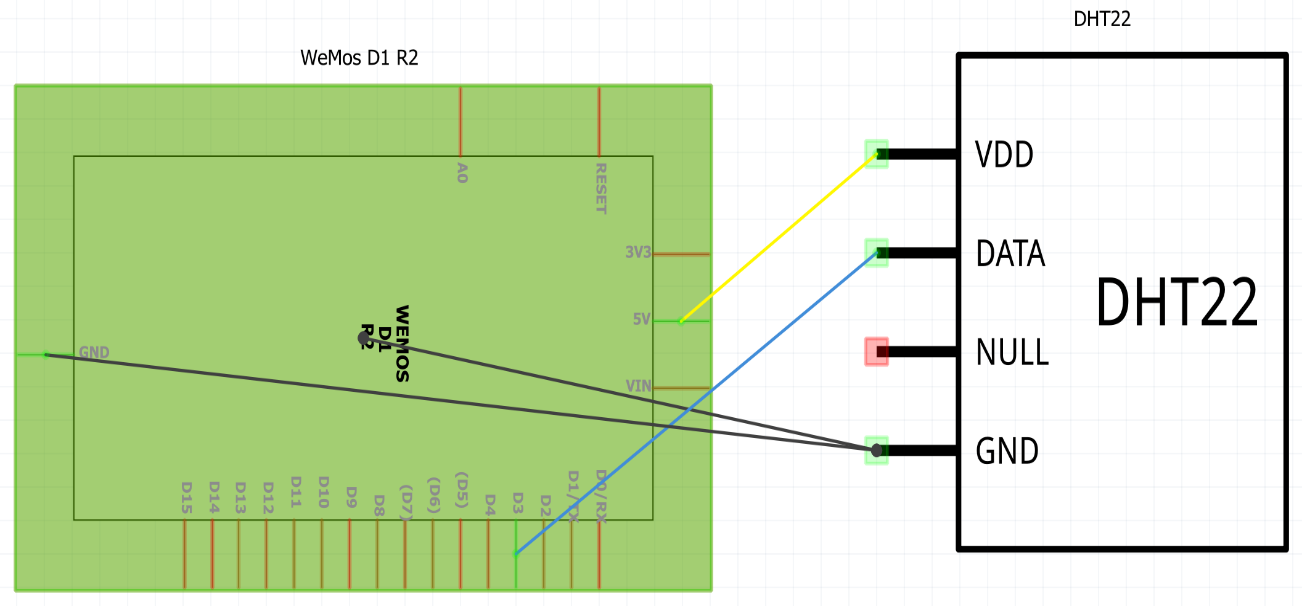
Συνδεσμολογία Υποσυστήματος με WeMos D1 R2

***Σχήμα 3.19 Σχηματική απεικόνιση σύνδεσης WeMos D1 R2 και DHT22***

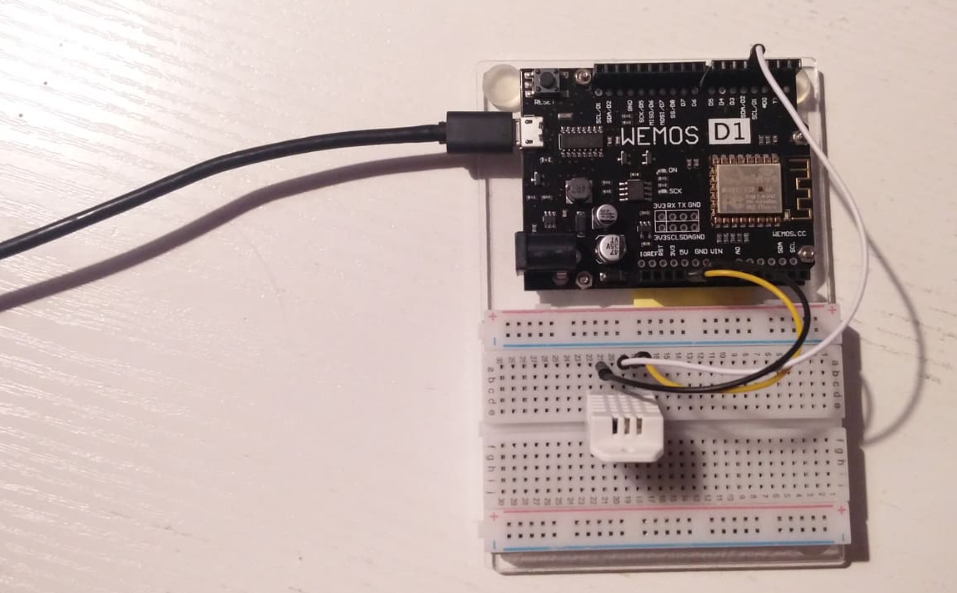
H συνδεσμολογία του δεύτερου υποσυστήματος που έχει ως μικροελεγκτή το WeMos D1 R2 διαφέρει σε σχέση με το προηγούμενο υποσύστημα. Σε αυτό το υποσύστημα δεν υπάρχει κάποιο shield με αποτέλεσμα μην υπάρχουν τέσσερα υλικά μέρη αλλά τρία.

Όσο αφορά τον αισθητήρα θερμοκρασίας & υγρασίας DHT22/AM2302, δεν υπάρχουν τροποποιήσεις σε ότι έχει να κάνει με την συνδεσμολογία ή τον κώδικα καθώς οι βιβλιοθήκες που χρησιμοποιούνται είναι κοινές για τους δύο μικροελεγκτές.

Στο παρακάτω σχήμα φαίνεται το ηλεκτρολογικό σχέδιο σύνδεσης του δεύτερου υποσυστήματος. (Λόγω τεχνικών προβλημάτων στο πρόγραμμα που σχεδιάστηκαν οι απεικονίσεις των συστημάτων, έχει γίνει η αλλαγή του άσπρου καλωδίου με μπλε καλώδιο.)



***Σχήμα 3.20 Ηλεκτρολογική σχηματική απεικόνιση συνδεσμολογίας δεύτερου υποσυστήματος***

Τέλος, είναι απραίτητη η σύνδεση του μικροελεγκτή WeMos D1 R2 με τον Η/Υ, μέσω ενός micro USB καλωδίου, τόσο για τροφοδοσία όσο και για την μεταφορά και συλλογή των δεδομένων, και το ανέβασμα του κώδικα στην πλακέτα. Στο σχήμα 3.20, παρουσιάζεται η τελική μορφή της συνδεσμολογίας που περιγράφηκε.

***Σχήμα 3.21 Tελική μορφή δεύτερου υποσυστήματος***

* 1. **Σύνδεση υλικού με τοπικό εξυπηρετητή**

Καθώς έχουν ολοκληρωθεί οι συνδεσμολογία στα δύο υποσυστήματα για την συλλογή και αποστολή των δεδομένων, σειρά έχει η σύνδεση του υλικού με τον τοπικό εξυπηρετητή καθώς και ο προγραμματισμός των υποσυστημάτων για την εύρυθμη αποστολή των δεδομένων που συλλέγονται. Στην παρούσα διπλωματική εργασία, αξίζει να σημειωθεί πώς οι μικροελεγκτές καθώς και ο τοπικός εξυπηρετητής έχουν το δικό του αναγνωριστικό συσκευής MAC Address, κάτι το οποίο έπαιξε σημαντικό ρόλο στον διαχωρισμό του κώδικα, αλλά και στον τρόπο με τον οποίο μεταφέρονται τα δεδομένα στην διαδικτυακή εφαρμογή. Όπως αναφέραμε παραπάνω, το λογισμικό που χρησιμοποιήθηκε για την συγγραφή του κώδικα που ‘ανέβηκε’ στους μικροελεγκτές είναι το Arduino IDE και το λειτουργικό σύστημα στο οποίο βασίστηκε η συγγραφή της παρούσας διπλωματικής εργασίας είναι τα Windows. Παρακάτω γίνεται μία ανάλυση σε ότι αφορά τον προγραμματισμό των μικροελεγκτών ώστε να γίνει η σωστή αποστολή των δεδομένων στις βάσεις δεδομένων καθώς και η σωστή - ασφαλή δρομολόγηση των δεδομένων μέσω του router στον τελικό τους προορισμό που είναι ο τοπικός εξυπηρετητής (Windows Server).

3. 3. 1. **Προγραμματισμός αισθητηρίων για λήψη δεδομένων**

Ξεκινώντας με τον προγραμματισμό των αισθητηρίων, πρέπει να αναφέρουμε πώς υπάρχουν διαφορές στον προγραμματισμό τους, καθώς αναφερόμαστε σε διαφορετικό μικροελεγκτή σε κάθε υποσύστημα. Οι βασικές βιβλιοθήκες παραμένουν ίδιες, αλλάζει όμως μερικώς, ο τρόπος γραφής ορισμένων μεταβλητών όπως είναι τα pins στα οποία είναι τοποθετημένοι οι ακροδέκτες για τα δεδομένα από τους αισθητήρες. Επίσης οι βιβλιοθήκες για τον τρόπο αποστολής των δεδομένων διαφέρουν, καθώς στο υποσύστημα με το μικροελεγκτή Arduino Uno Rev3 η μεταφορά δεδομένων γίνεται ενσύρματα μέσω καλωδίου ethernet και στο υποσύστημα με το μικροελεγκτή WeMos D1 R2 η μεταφορά δεδομένων γίνεται ασύρματα μέσω πρωτόκολλου WiFi. Το αρχεία που περιέχουν τον κώδικα για κάθε μικροελεγκτή είναι διαφορετικά και έχουν κατάληξη *.ino* έτσι ώστε να μπορούν να ανοίξουν με το λογισμικό Arduino IDE. Παρακάτω γίνεται μία πιο αναλυτική παρουσίαση των δύο αυτών αρχείων καθώς και των διαφορών που έχουν στην σύνταξη του πηγαίου κώδικα.

Υποσύστημα με μικροελεγκτή Arduino Uno Rev3

Για να παραμετροποιήσουμε το περιβάλλον εργασίας ώστε να γράψουμε τον πηγαίο κώδικα για το μικροελεγκτή Arduino Uno Rev3 δημιουργούμε ένα νέο sketch στο Arduino IDE και έπειτα το αποθηκεύουμε ως indoor\_sensor\_data.ino*.* Επίσης είναι απαραίτητο να επιλεγεί η πλακέτα Arduino και η θύρα σύνδεσής της με το περιβάλλον προγραμματισμού. Τα βήματα που ακολουθούνται για την επιλογή πλακέτας Arduino είναι: Tools →Board → και επιλογή της πλακέτας Arduino Uno Rev3 που χρησιμοποιείται. Τα βήματα που ακολουθούνται για την επιλογή θύρας είναι: Tools → Serial Port →και επιλογή της σειριακής θύρας (συνήθως COM#) ή θύρας USB που είναι συνδεδεμένο το Arduino.

Ξεκινώντας τον κώδικα, στην αρχή του προγράμματος, γίνονται οι δηλώσεις βιβλιοθηκών, μεταβλητών και έπειτα αυτό χωρίζεται σε δύο απλές συναρτήσεις. Η void setup() συνάρτηση χρησιμοποιείται για την αρχικοποίηση της Ethernet σύνδεσης. Τέλος, η void loop(), η οποία πραγματοποιεί συνεχείς επαναλήψεις στον κώδικα που έχει αυτή, δίνοντας τη δυνατότητα στο πρόγραμμα να εκτελείται συνεχώς και να στέλνει ή να λαμβάνει δεδομένα. Εκτελείται συνεχώς και καλείται αφού έχει πρώτα κληθεί και ολοκληρωθεί η void setup().

Για τη συγγραφή του αρχείου indoor\_sensor\_data.ino είναι αναγκαία η χρήση των βιβλιοθηκών <SPI.h> για την επικοινωνία Arduino-Shield , <Ethernet.h> για τη χρήση της πλακέτας Ethernet Shield και <DHT.h> για τη χρήση αισθητήρα.

* #include <SPI.h>
* #include <Ethernet.h>
* #include <dht.h>

Με τη λέξη *#include* εισάγεται κάθε βιβλιοθήκη, τιθέμενη σε εισαγωγικά ή αποσιωπητικά όταν αυτή βρίσκεται σε αρχείο. Οι βιβλιοθήκες είτε παρέχονται από το λογισμικό για τις πιο γνωστές λειτουργίες, είτε υπάρχουν στο Διαδίκτυο όπως η *<dht.h>* η λήψη της οποίας έγινε δωρεάν.

Η διασύνδεση του αισθητήρα DHT22/AM2302 γίνεται μέσω της βιβλιοθήκης “DHT.h” Στην ψηφιακή είσοδο (pin 2) εισέρχεται τάση η οποία έχει σταθμιστεί ανάλογα με τις τιμές θερμοκρασίας/υγρασίας του περιβάλλοντος και μετατρέπεται σε τιμή η οποία αντιστοιχεί στη θερμοκρασία/υγρασία. Η μετατροπή αυτή επιτυγχάνεται με τη χρήση της βιβλιοθήκης [28] .

Με βάση τo datasheet της εταιρίας κατασκευής ο αισθητήρας αναγνωρίζει τρεις συναρτήσεις:

1. readData()
2. readHumidity()
3. readTemperature()

H *readData(),* όπουεπιστρέφει την κατάσταση του αισθητήρα. Κατά την επιστροφή, έχουν οριστεί οι παρακάτω τιμές από την βιβλιοθήκη *<dht.h>:*

* **DHT\_ERROR\_NONE**: Εάν δεν υπάρχουν σφάλματα
* **DHT\_ERROR\_CHECKSUM**: Σφάλμα κατά τον έλεγχο με το Bit ισοτιμίας
* **DHT\_BUS\_HUNG**: Σφάλμα
* **DHT\_ERROR\_NOT\_PRESENT**: Δεν εντοπίστηκε ο αισθητήρας
* **DHT\_ERROR\_ACK\_TIMEOUT**: Το σήμα ACK έληξε
* **DHT\_ERROR\_SYNC\_TIMEOUT**: Σφάλμα κατά το συγχρονισμό
* **DHT\_ERROR\_DATA\_TIMEOUT**: Σφάλμα δεδομένων
* **DHT\_ERROR\_TOOQUICK**: Ζητήθηκε γρηγορότερα από τον αναμενόμενο χρόνο (< 2 seconds) αίτημα για νέα δεδομένα

Η *readHumidity(),* όπου επιστρέφει την τρέχουσα τιμή του αισθητήρα για την υγρασία σε ποσοστό %

Η *readTemperature(),* όπουεπιστρέφει την τρέχουσα τιμή της θερμοκρασίας σε βαθμούς Κελσίου

Στο τμήμα του κώδικα χρησιμοποιούνται οι *readHumidity()* και *readTemperature()* για τη λήψη δεδομένων ως εξής:

Void loop(){

int readData = DHT.read22(dataPin); // Επιστρέφεται η κατάσταση του αισθητήρα

float t = DHT.temperature; // Λαμβάνουμε την τιμή της θερμοκρασίας

float h = DHT.humidity; // Λαμβάνουμε την τιμή της υγρασίας

Μετά τη δήλωση των βιβλιοθηκών ακολουθεί η δήλωση και αρχικοποίηση των σταθερών μεταβλητών, των οποίων η τιμή δεν μπορεί να αλλάξει μέσα στο τμήμα κώδικα που ακολουθεί.

#define datapin 2 // ακροδέκτης για την σύνδεση του αισθητήρα και μεταφορά των δεδομένων

#define DHTTYPE DHT22 // DHT 22 (AM2302)

DHT dht(DHTPIN, DHTTYPE); // Προετοιμασία αισθητήρα για 16mhz Arduino

Οι μεταβλητές t και h έχουν δηλωθεί στο αρχικό τμήμα κώδικα μαζί με τον domain της Διαδικτυακής εφαρμογής και τον κωδικό-όνομα της συσκευής στην οποία θα μεταφερθεί ο κώδικας.

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

byte ip[] = {192, 168, 1, 76 }; // Ανάθεση στατικής τοπικής IP

byte serv[] = {192, 168, 1, 100} ; // Πληροφορία για την στατική IP του τοπικού εξυπηρετητή

EthernetClient cliente; // Δημιουργία κλάσης client

Η μεταφορά των ληφθέντων τιμών γίνεται με τη χρήση της βιβλιοθήκης Ethernet.h. Αρχικά στο τμήμα του κώδικα με τη δήλωση των μεταβλητών, πραγματοποιείται η δημιουργία μιας κλάσης cliente. Μέσα στη συνάρτηση loop() μέσω της cliente εκτελούνται όλες οι συναρτήσεις και εντολές για τη δημιουργία διασύνδεσης της εκάστοτε συσκευής μετρήσεων με το Διαδίκτυο και κατά συνέπεια τη βάση και τη διαδικτυακή εφαρμογή. Η cliente.connect(server,8080)συνδέεται με τη διεύθυνση IP και τη θύρα που καθορίστηκε κατά τη δημιουργία. Έπειτα με την μέθοδο cliente.print() δημιουργείται ένα αίτημα GET από τον τοπικό εξυπηρετητή και έτσι οι τιμές θερμοκρασίας και υγρασίας περνάνε στη βάση δεδομένων μέσω του αρχείου data.php με ένα query που δημιουργείται και οι τιμές εισάγονται υπό μορφή JSON Format.

Στην void loop() εκτός από τη λήψη δεδομένων και τη διαχείρισή τους, ορίζεται ταυτόχρονα και ο ρυθμός επαναληψιμότητας της. Στη συγκεκριμένη εφαρμογή έχει επιλεγεί η συλλογή δεδομένων από τον αισθητήρα να γίνεται κάθε πέντε λεπτά.

Μόλις ολοκληρωθεί και ο έλεγχος για τυχόν σφάλματα και συντακτικά λάθη στον κώδικα, μέσω του εργαλείου “upload” γίνεται η μεταγλώττιση και φόρτωση του στην πλακέτα και η συσκευή άμεσα ξεκινά την καταγραφή των δεδομένων. Η τελική μορφή του αρχείου indoor\_sensor\_data.ino βρίσκεται στο Παράρτημα B.

Υποσύστημα με μικροελεγκτή WeMos D1 R2

Για να παραμετροποιήσουμε το περιβάλλον εργασίας ώστε να γράψουμε τον πηγαίο κώδικα για το μικροελεγκτή WeMos D1 R2 δημιουργούμε ένα νέο sketch στο Arduino IDE και έπειτα το αποθηκεύουμε ως outdoor\_sensor\_data.ino. Έπειτα η διαδικασία τροποιείται σε σχέση με το μικροελεγκτή Arduino Uno Rev3 καθώς πρέπει να εγκαταστήσουμε κάποια πακέτα ώστε να υπάρχει υποστήριξη από το Arduino IDE για το μικροελεγκτή WeMos D1 R2. Συγκεκριμένα, τα βήματα είναι: File → Preferences και έτσι μας ανοίγει ένα παράθυρο με τις προτιμήσεις. Στην καρτέλα Settings και στο πεδίο Additional Boards Manager URLs τοποθετούμε τον παρακάτω σύνδεσμο <http://arduino.esp8266.com/stable/package_esp8266com_index.json> και πατάμε OK. Μετά, πηγαίνουμε Tools → Board → Boards Manager και στο πεδίο αναζήτησης από το παράθυρο που έχει ανοίξει αναζητούμε την λέξη esp8266. Εγκαθιστούμε τα πακέτα που μας έχει προβάλει και πλέον είμαστε έτοιμοι να χρησιμοποιήσουμε το WeMos D1 R2. Τα βήματα που ακολουθούνται για την επιλογή πλακέτας Arduino είναι: Tools →Board → και επιλογή της πλακέτας WeMos D1 R2 & mini που χρησιμοποιείται. Τα βήματα που ακολουθούνται για την επιλογή θύρας είναι: Tools → Serial Port →και επιλογή της σειριακής θύρας (συνήθως COM#) ή θύρας USB που είναι συνδεδεμένο το WeMos D1 R2. Τελευταίο βήμα είναι να αλλάξουμε την ταχύτητα ανεβάσματος, έτσι πηγαίνουμε Tools → Upload Speed → και επιλέγουμε ταχύτητα 115200 διότι έτσι ορίζει ο κατασκευαστής της πλακέτας.

Για τη συγγραφή του αρχείου outdoor\_sensor\_data.ino είναι αναγκαία η χρήση των βιβλιοθηκών, <ESP8266WiFi.h> - <WiFiClient.h> - <ESP8266WebServer.h> - <ESP8266mDNS.h> για την αρχικοποίηση του ESP8266 module και για την χρήση του ως τοπικό υποσύστημα, αλλά και τις βιβλιοθήκες <DHT.h> - <Adafruit\_Sensor.h> για την χρήση του αισθητήρα.

* #include <ESP8266WiFi.h>
* #include <WiFiClient.h>
* #include <ESP8266WebServer.h>
* #include <ESP8266mDNS.h>
* #include <DHT.h>
* #include <Adafruit\_Sensor.h>

Μετά τη δήλωση των βιβλιοθηκών ακολουθεί η δήλωση και αρχικοποίηση των σταθερών μεταβλητών, των οποίων η τιμή δεν μπορεί να αλλάξει μέσα στο τμήμα κώδικα που ακολουθεί.

#define DHTPIN D2 // ακροδέκτης για την σύνδεση του αισθητήρα και μεταφορά των δεδομένων

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

DHT dht(DHTPIN, DHTTYPE); //Προετοιμασία αισθητήρα για WeMos D1 R2

const char\* ssid = "SmartHomie"; // Απόδοση ονόματος τοπικού δικτύου WiFi (WiFi SSID)

const char\* password = "!smarthomie10!"; // Απόδοση τοπικού password για το δίκτυο WiFi

Έπειτα ορίζονται οι στατικές IP για την σωστή δρομολόγηση των πακέτων.

IPAddress ip(192, 168, 1, 77); // Ανάθεση τοπικής στατικής IP

IPAddress gateway(192, 168, 1, 1); // Ανάθεση προεπιλεγμένης πύλης

IPAddress subnet(255, 255, 255, 0); // Ανάθεση τοπικής μάσκας δικτύου

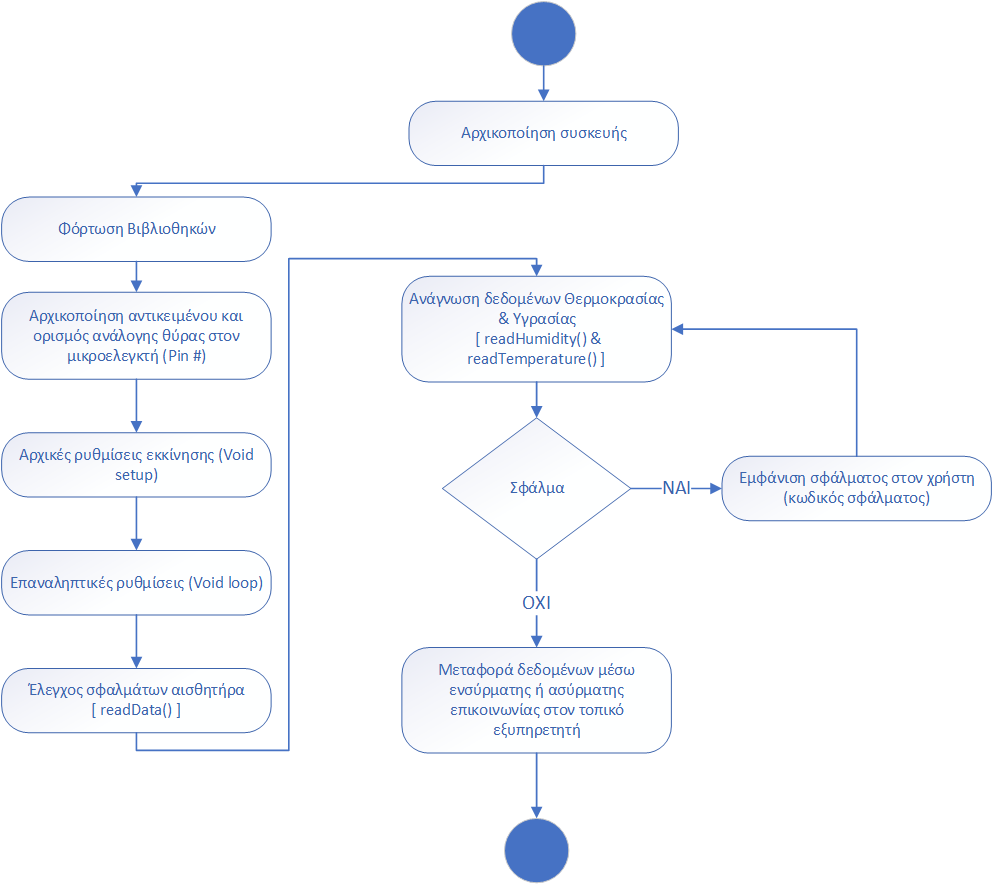
IPAddress dns1(192, 168, 1, 1); //DNS Server 1

IPAddress dns2(192, 168, 1, 2); //DNS Server 2

Όσο αφορά τις ρυθμίσεις του αισθητήρα DHT22/AM2302, δεν υπάρχουν τροποποιήσεις στην λήψη και αποστολή δεδομένων, και η διαδικασία με την μέθοδο GET και την μορφή JSON Format με οποία αποστέλλονται τα δεδομένα στις βάσεις δεδομένων παραμένει ίδια.

* + 1. **Διάγραμμα ροής λειτουργίας αισθητήρα DHT22/Am2302**

Η σωστή λειτουργία του αισθητήρα DHT22/Am2302 βασίζεται τόσο στη σωστή σύνδεσή του με τον μικροελεγκτή όσο και στον ορθό προγραμματισμό του. Η διαδικασία για τον τρόπο λειτουργίας του ως εξάρτημα της συσκευής και βασικό χαρακτηριστικό του κώδικα αναπαριστάται στο διάγραμμα ροής του σχήματος 3.21.

***Σχήμα 3.22 Διάγραμμα ροής λειτουργίας αισθητήρα DHT22/AM2302***

* 1. **Υπολογισμός κόστους εξαρτημάτων**

Σε αυτή την ενότητα, γίνεται καταγραφικών των υλικών μερών που χρησιμοποιήθηκαν για την κατασκευή των υποσυστημάτων καθώς και για το σύνολο τους συστήματος με το οποίο έγινε πραγματικότητα η διαδικτυακή εφαρμογή. Στον παρακάτω πίνακα γίνεται μία ανάλυση κόστους για κάθε τιμή μονάδας προϊόντος, αναλύονται οι ποσότητες και υπολογίζεται το συνολικό κόστος εγκατάστασης των συσκευών.

***Πίνακας 3.2: Υπολογισμός κόστους υλικών μερών***

|  |  |  |
| --- | --- | --- |
| **Υλικό** | **Ποσότητα** | **Τιμή μονάδας (€)** |
| Arduino Uno Rev3 + USB Καλώδιο | 1 | 20,00 |
| WeMos D1 R2 + microUSB καλώδιο | 1 | 12,00 |
| Ethernet Shield W5100 | 1 | 16,20 |
| Αισθητήρας DHT22/AM2302 | 2 | 7,00 |
| Mini Solderless Breadboard 400 | 2 | 3,00 |
| Jumper Wires (65 κομμάτια) | 1 | 3,50 |
| Ethernet RJ45 cable | 1 | 5,00 |
| Ακρυλική βάση | 2 | 2,80 |
| **Σύνολο: 79,50€** | | |

Όπως αναφέρθηκε και σε προηγούμενες ενότητες, καθίσταται δυνατό, με τις παραπάνω συσκσευές χαμηλού κόστους, να μπορεί κάθε χρήστης να καταγράφει και να παρακολουθεί σε ζωντανό χρόνο την κατάσταση θερμοκρασίας και υγρασίας καθώς και να εναλλάσει τις καταστάσεις στις έξυπνες συσκευές στο σπίτι του.

Κεφάλαιο 4

**Σχεδίαση και υλοποίηση διαδικτυακής εφαρμογής**

Έχοντας ολοκληρώσει την συνδεσμολογία των συστημάτων, το τελικό στάδιο είναι η σχεδίαση και δημιουργία της διαδικτυακής εφαρμογής και παράλληλα η σχεδίαση των βάσεων στις οποίες θα αποθηκεύονται όλα τα απαραίτητα δεδομένα και πληροφορίες οι οποίες εμφανίζονται στον χρήστη. Με την παρούσα διαδικτυακή εφαρμογή, ο χρήστης μπορεί να παρακολουθεί, να αναλύει, αλλά και να κάνει αλλαγές στην υπάρχουσα κατάσταση του χώρου τον οποίο έχει επιλέξει. H προβολή και παρακολούθηση αυτών των δεδομένων επιτυγχάνεται μέσω της διαδικτυακής εφαρμογής Smart Homie.

Στις ενότητες που ακολουθούν παρουσιάζονται όλες οι απαιτήσεις του συστήματος που κατασκευάστηκε. Περιγράφεται η διαδικασία σχεδίασης και δημιουργίας των βάσεων καθώς και οι λειτουργίες της διαδικτυακής εφαρμογής, που στηρίχτηκαν στις απαιτήσεις του συστήματος. Τέλος, εξηγείται η διαδικασία και οι πηγαίοι κώδικες που αναπτύχθηκαν για τη δημιουργία της διαδικτυακής εφαρμογής και της επικοινωνίας της με τη βάση.

2. 1. **Απαιτήσεις Συστήματος**

Βασικός στόχος της παρούσας διπλωματικής εργασίας είναι η υλοποίηση του συστήματος καταγραφής και παρουσίασης δεδομένων θερμοκρασίας και υγρασίας. Το σύστημα αυτό αποτελείται τόσο από τα υποσυστήματα για τον εσωτερικό και εξωτερικό χώρο, όσο και από τη διαδικτυακή εφαρμογή Smart Homie. Έτσι λοιπόν, κρίνεται αναγκαίο να διαμορφωθούν κάποιες απαιτήσεις από το σύστημα για την αποτελεσματική λειτουργία όλων των χαρακτηριστικών που έχουν αναπτυχθεί. Παρακάτω παρουσιάζεται μία λίστα με τις απαιτήσεις που υπάρχουν από την διαδυκτιακή εφαρμογή.

1. Για την σωστή καταγραφή της δραστηριότητας αλλά και για την ασφάλεια της διαδυκτιακής εφαρμογής είναι απαραίτητη η εγγραφή κάποιου χρήστη κατά την εκκίνηση της διαδυκτιακής εφαρμογής για πρώτη φορά. Όταν τοποθετηθεί το σύστημα για πρώτη φορά σε κάποιον οικιακό χώρο, ο πρώτος χρήστης παίρνει δικαιώματα διαχειριστή και έπειτα μπορεί να προσθέσει νέους διαχειριστές/χρήστες, ανάλογα με τα δικαιώματα που θέλει να δώσει.
2. Κάθε υποσύστημα καταγραφής, απαιτεί την ύπαρξη του κατάλληλου πηγαίου κώδικα με σκοπό την επικοινωνία του με τον τοπικό εξυπηρετητή. Με αυτόν τον τρόπο, καθορίζεται και η τοποθεσία στην οποία θα εγκατασταθεί, είτε σε εξωτερικό είτε σε εσωτερικό χώρου του σπιτιού. Ακόμη, πρέπει να τοποθετηθεί το ανάλογο αρχείο .ino, ανάλογα με τον τρόπο αποστολής των δεδομένων (ασύρματα ή ενσύρματα).
3. Το σύστημα πρέπει να προσφέρει εξασφάλιση γρήγορης και ακίνδυνης σύνδεσης στην εφαρμογή από τον χρήστη. Κάθε χρήστης κατά την εγγραφή του πληκτρολογεί έναν προσωπικό κωδικό πρόσβασης. Ο κρυπτογραφημένος αυτός κωδικός αποθηκεύεται στη βάση με τη χρήση της εντολής του κώδικα.
4. Το περιβάλλον την εφαρμογής πρέπει να είναι απλό, ξεκάθαρο και φιλικό προς τον χρήστη. Η διαδικτυακή εφαρμογή ενδέχεται να χρησιμοποιηθεί από άτομα με έλλειψη τεχνολογικής εκπαίδευσης. Ο σχεδιασμός της πρέπει να είναι συμβατός με την εύκολη και ομαλή περιήγηση του κάθε χρήστη μέσα σε αυτή.
5. Οι χρήστες να μπορούν να επεξεργάζονται τα στοιχεία των προφίλ τους. Κάθε χρήστης θα πρέπει να έχει τη δυνατότητα να τροποποιήσει τα προσωπικά του στοιχεία οποιαδήποτε στιγμή αυτός επιθυμεί (π.χ. αλλαγή κωδικού πρόσβασης για καλύτερη ασφάλεια,αλλαγή του email τους σε περίπτωση που έχουν αλλάξει).
6. Οι χρήστες να μπορούν να ενημερώνονται για την κατάσταση των συσκευών σε πραγματικό χρόνο. Πέρα από την καταγραφή των δεδομένων που συλλέγουν οι συσκευές ο κάθε χρήστης θα πρέπει να μπορεί να παρακολουθεί τις τρέχουσες τιμές υγρασίας και θερμοκρασίας του χώρου που έχει επιλέξει όποια στιγμή επισκεφτεί τη διαδικτυακή εφαρμογή.
7. Οι χρήστες να μπορούν να αποθηκεύουν αρχεία από την εφαρμογή. Εκτός του αρχείου καταγραφής δεδομένων στην εφαρμογή ο χρήστης θα μπορεί να αποθηκεύει και να εκτυπώνει τα διαγράμματα που παρακολουθεί στην εφαρμογή.
   1. **Διαδικτυακή ασφάλεια & Προστασία προσωπικών δεδομένων**

Στις μέρες μας, μεγάλο μέρος των προσωπικών δεδομένων των ανθρώπων βρίσκεται στο διαδίκτυο. Άλλοτε γίνεται με την θέληση τους και άλλοτε εν αγνοία τους. Προσωπικά δεδομένα είναι κάθε πληροφορία που αναφέρεται στο πρόσωπο του κάθε ατόμου, όπως: το όνομα και το επάγγελμά του, η οικογενειακή του κατάσταση, η ηλικία του, ο τόπος κατοικίας, η φυλετική του προέλευση, τα πολιτικά του φρονήματα, η θρησκεία που πιστεύει, οι φιλοσοφικές του απόψεις, η συνδικαλιστική του δράση, η υγεία του, η ερωτική του ζωή και οι τυχόν ποινικές του διώξεις και καταδίκες. Δεν θεωρούνται προσωπικά δεδομένα πληροφορίες από τις οποίες δεν δυνάται να ταυτοποιηθεί ένα συγκεκριμένο άτομο [29].

Σημαντικό κομμάτι της παρούσας διπλωματικής εργασίας αποτέλεσε ο τρόπος με τον οποίο συλλέγονται τα δεδομένα από τους αισθητήρες, ο τρόπος αποθήκευσης τους στην βάση δεδομένων καθώς και η εμφάνιση τους στον τελικό χρήστη. Ακολουθήθηκαν κάποια βήματα για την ασφαλέστερη λειτουργία της διαδικτυακής εφαρμογής και τα δεδομένα του χρήστη τα οποία είναι:

* **Η διαδικτυακή εφαρμογή δουλεύει σε τοπικό δίκτυο:** Πρακτικά, αυτό σημαίνει πώς ο χρήστης δεν μπορεί κατά την διάρκεια της αρχικής εγκατάστασης να εισέλθει στην εφαρμογή από το διαδίκτυο. Αυτό μειώνει τον κίνδυνο απώλειας δεδομένων κατά ένα μεγάλο βαθμό, καθώς είναι πρακτικά αδύνατο κάποιος ο οποίος δεν βρίσκεται στο τοπικό δίκτυο της εφαρμογής να υποκλέψει τα δεδομένα.
* **Η χρήση της διαδικτυακής εφαρμογής μπορεί να γίνει μέσω δικτύου:** Με την άνωθεν λειτουργία της εφαρμογής ο χρήστης θα πρέπει να βρίσκεται σε μία απόσταση, ώστε να μπορεί να συνδεθεί στο τοπικό δίκτυο με WiFi είτε με καλώδιο Ethernet. Έτσι, η λειτουργικότητα της εφαρμογής δυσκολεύει σε ορισμένες περιπτώσεις τον χρήστη. Λύση σε αυτό το πρόβλημα, έρχεται να δώσει η δυνατότητα του δρομολογητή, να ανοίξει μία διαδικτυακή πόρτα στην οποία θα γίνει αντιστοίχιση του τοπικού εξυπηρετητή και έτσι εφόσον ο χρήστης έχει ορίσει το όνομα τομέα (domain name) που αυτός θέλει καθώς και την θύρα στον δρομολογητή, θα μπορεί να χρησιμοποιήσει την εφαρμογή από οπουδήποτε στον κόσμο μέσω διαδικτύου. Για τις ανάγκες της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε ένα τοπικό όνομα τομέα <https://smarthomie.local/> καθώς και η θύρα 61457 στον δρομολογητή.
* **Κατά την εγγραφή του χρήστη, ο κωδικός αποστέλλεται κρυπτογραφημένος:** Η φόρμα εγγραφής αποστέλλει τα στοιχεία του χρήστη στη βάση δεδομένων. Πριν την αποστολή ο κωδικός που έχει εισάγει ο χρήστης περνάει από μία συνάρτηση της PHP την password\_hash()με την βοήθεια της οποίας δημιουργείται ένα αλγόριθμος κρυπτογράφησης και έτσι ο κωδικός αποστέλλεται κρυπτογραφημένος[30].
* **Στον τοπικό εξυπηρετητή έχει ενεργοποιηθεί το πρωτόκολλο SSL:** Το πρωτόκολλο SSL (Secure Sockets Layer) αναπτύχθηκε από την εταιρεία Netscape και σχεδιάστηκε για να παρέχει ασφάλεια κατά την μετάδοση ευαίσθητων δεδομένων στο διαδίκτυο. Το SSL χρησιμοποιεί μεθόδους κρυπτογράφησης των δεδομένων που ανταλλάσσονται μεταξύ δύο συσκευών (συνηθέστερα Ηλεκτρονικών Υπολογιστών) εγκαθιδρύοντας μία ασφαλή σύνδεση μεταξύ τους μέσω του διαδικτύου. Το πρωτόκολλο αυτό χρησιμοποιεί το TCP/IP για τη μεταφορά των δεδομένων και είναι ανεξάρτητο από την εφαρμογή που χρησιμοποιεί ο τελικός χρήστης. Για τον λόγο αυτό μπορεί να παρέχει υπηρεσίες ασφαλούς μετάδοσης πληροφοριών σε πρωτόκολλα ανώτερου επιπέδου όπως για παράδειγμα το HTTP [31]. Έτσι για την ασφαλέστερη μεταφορά των δεδομένων προς τον χρήστη, στον πίνακα ελέγχου του XAMPP έχει ενεργοποιηθεί το πρωτόκολλο SSL και έχει ανοιχθεί η απαραίτητη 443 θύρα στον δρομολογητή.
  1. **Δημιουργία και περιγραφή των βάσεων δεδομένων**

Η βάση δεδομένων που δημιουργήθηκε για τη διαδικτυακή εφαρμογή Smart Homie ονομάστηκε smart\_homie και αποτελείται από πέντε πίνακες οι οποίοι δημιουργήθηκαν για να υποστηρίζουν τις απαιτήσεις του συστήματος. Ο σχεδιασμός της έγινε έτσι ώστε να μπορούν να εξυπηρετούνται πολλοί χρήστες με μία ή και περισσότερες συσκευές στην κατοχή τους ταυτόχρονα. Αφού γίνει περιγραφή των πινάκων και των γνωρισμάτων τους πρέπει να καθοριστούν και οι συσχετίσεις μεταξύ αυτών. Οι πίνακες παρατίθενται στο τέλος της συγκεκριμένης ενότητας.

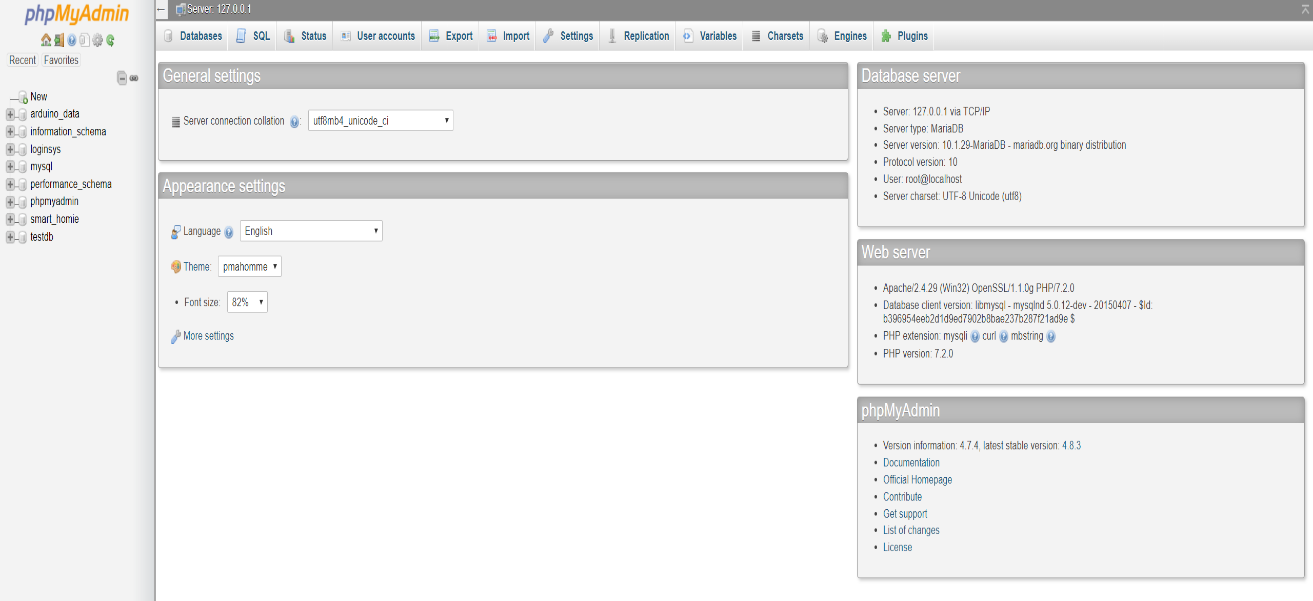
1. 3. 1. **Εργαλείο δημιουργίας βάσεων δεδομένων phpMyAdmin**

Για την επεξεργασία των βάσεων δεδομένων είναι απαραίτητη η χρήση ενός εργαλείου διαχείρισης βάσεων δεδομένων. Στην παρούσα διπλωματική εργασία επιλέχθηκε το phpMyAdmin, (μια εφαρμογή) που δημιουργήθηκε για τη διαχείριση εξυπηρετητών MySQL. To εργαλείο phpMyAdmin έρχεται προεγκατεστήμενο στον πίνακα ελέγχου του XAMPP και έτσι καθίσταται αρκετά εύκολη η πρόσβαση του. Τo εργαλείο είναι ικανό να είναι διαχειρίσιμο από απλές βάσεις δεδομένων μέχρι έναν ολόκληρο mysql server. Είναι εύκολο και κατανοητό στην χρήση και παράλληλα πάροχος πολλών δυνατοτήτων για κάθε χρήστη.

Οι δυνατότητες που προσφέρει το phpMyAdmin, το καθιστούν ένα δυνατό εργαλείο έναντι σε άλλα ανταγωνιστικά. Παρακάτω αναλύονται μερικές από τις δυνατότητες που έχει αυτό το εργαλείο.

* Δημιουργεί και διαγράφει βάσεις δεδομένων
* Δημιουργεί, τροποποιεί, διαγράφει, αντιγράφει και μετονομάζει πίνακες
* Κάνει συντήρηση της βάσης
* Προσθέτει, διαγράφει και τροποποιεί πεδία πινάκων
* Εκτελεί ερωτήματα SQL ακόμα και ομαδικά (batch
* Διαχειρίζεται κλειδιά σε πεδία
* Φορτώνει αρχεία κειμένου σε πίνακες
* Δημιουργεί και διαβάζει πίνακες (που προέρχονται από dump βάσεις)
* Εξάγει δεδομένα σε μορφή CSV, XML και LATEX
* Εισάγει δεδομένα και δομές MySQL από CSV, XML και SQL αρχεία
* Διαχειρίζεται πολλούς διακομιστές
* Προσθέτει, επεξεργάζεται και μετακινεί τους χρήστες MySQL και τα δικαιώματά τους
* Ελέγχει την αναφορική δραστηριότητα των δεδομένων των MyISAM πινάκων
* Δημιουργεί αυτόματα περίπλοκα ερωτήματα για τη σύνδεση των πινάκων χρησιμοποιώντας QBE(Query-by-example)
* Δημιουργεί PDF γραφικών του layout της βάσης δεδομένων
* Εκτελεί αναζητήσεις σε όλη τη βάση δεδομένων ή σε μέρος αυτής
* Εντοπίζει και επισημάνει τυχόν αλλαγές στις βάσεις και τους πίνακες
* Υποστηρίζει πίνακες InnoDB και ξένα κλειδιά
* Υποστηρίζει MySQLi, μια βελτιωμένη επέκταση του MySQL PhpMyAdmin
* Δημιουργεί, επεξεργάζεται, καλεί, εξάγει αποθηκευμένες διαδικασίες και συναρτήσεις
* Επικοινωνεί σε 80 διαφορετικές γλώσσες

Η διαδικασία την οποία ακολουθούμε για να δούμε το γραφικό περιβάλλον αυτού του εργαλείου είναι απλή. Χρειάζεται απλά να γράψουμε το τοπικό όνομα τομέα <http://localhost:8080/phpMyAdmin> το οποίο έχει παραχθεί από τις ρυθμίσεις που χρησιμοποιούμε στο XAMPP και το αποτέλεσμα φαίνεται στο σχήμα 4.1.



***Σχήμα 4.1 Γραφικό περιβάλλον phpMyAdmin***

* + 1. **Αρχιτεκτονική βάσης δεδομένων διαδικτυακής εφαρμογής**

Όλες οι βάσεις δεδομένων αποτελούνται από πίνακες όπου εκεί αποθηκεύονται τα δεδομένα. Με αυτό τον τρόπο επιτυγχάνεται ένας διαχωρισμός δεύτερου επιπέδου, καθώς πολλά δεδομένα μπορεί να αναφέρονται στην ίδια εφαρμογή αλλά να χρησιμοποιούνται για διαφορετικό σκοπό. Έτσι για την καλύτερη επεξεργασία των δεδομένων που λαμβάνει η διαδικτυακή εφαρμογή έχουν δημιουργηθεί πέντε πίνακες μέσα στη βάση δεδομένων. Αφού γίνει περιγραφή των πινάκων και των γνωρισμάτων τους πρέπει να καθοριστούν και οι συσχετίσεις μεταξύ αυτών. Οι πίνακες παρατίθενται στο τέλος της συγκεκριμένης ενότητας.

1. Πίνακας Users

Στον Πίνακα 4.1: Πίνακας users, αποθηκεύονται όλες οι πληροφορίες και τα στοιχεία κάθε χρήστη που έχει εγγραφεί στην εφαρμογή και έχει δυνατότητα πρόσβασης σε αυτήν.

Ανάλυση των στοιχείων του πίνακα:

* **id:** Ο αναγνωριστικός αύξων αριθμός του χρήστη στον πίνακα, είναι μοναδικός για κάθε εγγραφή και αποτελεί το πρωτεύον κλειδί του. O τύπος του int(11) επιλέχθηκε γιατί η τιμή των αναγνωριστικών αριθμών είναι πάντα ακέραια. Με την ιδιότητα UNSIGNED εξασφαλίζονται 2,147,483,647 διαφορετικοί χρήστες για το σύστημα. Η πρόσθετη επιλογή AUTO\_INCREMENT δείχνει ότι ο αριθμός αυτός αυξάνεται αυτόματα.
* **username:** Το όνομα χρήστη με το οποίο, ο χρήστης θα εισέρχεται στην εφαρμογή. Ο τύπος varchar(50) χρησιμοποιείται για να καλυφθεί το μήκος των χαρακτήρων που συμφωνούν με τους κανόνες ασφαλείας.
* **email:** Η διεύθυνση ηλεκτρονικού ταχυδρομείου με την οποία ο χρήστης θα κάνει εγγραφή στην ιστοσελίδα. Ο τύπος varchar(255) χρησιμοποιείται για να καλυφθεί το μέγιστο δυνατό όριο χαρακτήρων που μπορεί να έχει μια ηλεκτρονική διεύθυνση.
* **adminuser:** Το αναγνωριστικό με το οποίο κάποιος χρήστης διαχωρίζεται σε διαχεχιριστή και απλό χρήστη. Τα δικαιώματα που έχει εξαρτώνται από την τιμή του πεδίου adminuser. Ο τύπος είναι TINYINT(1), η τιμή 1 περιγράφει τον χρήστη ως διαχειριστή και η τιμή 0 ως απλό χρήστη.
* **password:** Ο κωδικός με τον οποίο ενεργοποιείται η εγγραφή και πραγματοποιείται η είσοδος του χρήστη στο σύστημα. Για την ασφάλεια του χρήστη επίσης δόθηκε ο τύπος varchar(255) έτσι ώστε να καλυφθεί και ο πιο σύνθετος κωδικός του χρήστη. Αξίζει να σημειωθεί ότι ο κωδικός κάθε χρήστη στη βάση αποθηκεύεται σε κρυπτογραφημένη μορφή.
* **created\_at:** Η ημερομηνία δημιουργίας του χρήστη. Έχει ως προεπιλογή την αυτόματη λήψη ημερομηνίας με το πέρας της διαδικασίας εγγραφής.

1. Πίνακας users\_log

Στον Πίνακα 4.2: Πίνακας users\_log, αποθηκεύεται η δραστηριότητα του χρήστη κατά την διάρκεια χρήσης της εφαρμογής.

Ανάλυση των στοιχείων του πίνακα:

* **id:** Είναι ο αναγνωριστικός αύξων αριθμός ο οποίος ξεχωρίζει τις δραστηριότητες μεταξύ τους. Ο τύπος int(11) επιλέχθηκε γιατί η τιμή των αναγνωριστικών αριθμών είναι πάντα ακέραια. Με την ιδιότητα UNSIGNED εξασφαλίζεται ένα μεγάλο πλήθος για καταχωρήσεις διαφορετικών συσκευών. Η πρόσθετη επιλογή AUTO\_INCREMENT δείχνει ότι ο αριθμός αυτός αυξάνεται αυτόματα.
* **user\_id:** Είναι ο αναγνωριστικός αριθμός που έχει τοποθετηθεί για τον συγκεκριμένο χρήστη στον πίνακα users. Έτσι, μπορεί να γίνει ο διαχωρισμός των δραστηριοτήτων ανά χρήστη. Ο τύπος varchar(20) επιλέχθηκε καθώς το αναγνωριστικό εδώ εξετάζεται ως χαρακτήρας και όχι ως κάποιος αριθμός.
* **date:** Η ημερομηνία στην οποία καταγράφηκε κάποια δραστηριότητα από τον χρήστη. Ο τύπος date επιλέχθηκε καθώς το πεδίο έχει μόνο ημερομηνίες με την μορφή YYYY-MM-DD.
* **time:** Η ώρα στην οποία καταγράφηκε κάποια δραστηριότητα από τον χρήστη. Ο τύπος time επιλέχθηκε καθώς το πεδίο έχει μόνο την ώρα με την μορφή HH-MM-SS**.**
* **log:** Το μήνυμα με το οποίο υποδεικνύει η διαδικτυακή εφαρμογή στον χρήστη, το είδος της δραστηριότητας που καταγράφηκε. Ο τύπος varchar(100) επιλέχθηκε καθώς το πεδίο θα έχει ως τιμές κάποιο μήνυμα το οποίο δεν πρέπει να ξεπερνάει τους 100 χαρακτήρες.

1. Πίνακας devices\_states

Στον Πίνακα 4.3: devices\_states καταχωρούνται όλες οι έξυπνες συσκευές του χώρου καθώς και η κατάσταση στην οποία βρίσκονται (ενεργή/ανενεργή κατάσταση)

Ανάλυση των στοιχείων του πίνακα:

* **id:** Είναι ο αναγνωριστικός αύξων αριθμός ο οποίος ξεχωρίζει τις συσκευές μεταξύ τους. Ο τύπος int(11) επιλέχθηκε γιατί η τιμή των αναγνωριστικών αριθμών είναι πάντα ακέραια. Με την ιδιότητα UNSIGNED εξασφαλίζεται ένα μεγάλο πλήθος για καταχωρήσεις διαφορετικών συσκευών. Η πρόσθετη επιλογή AUTO\_INCREMENT δείχνει ότι ο αριθμός αυτός αυξάνεται αυτόματα.
* **devicename:** Είναι η περιγραφή που δίνεται για την αναγνώριση της συσκευής και τον χώρο όπου είναι τοποθετημένη. Ο τύπος varchar(20) επιλέχθηκε καθώς η περιγραφή περιέχει χαρακτήρες και δεν πρέπει να ξεπερνάει τους 20 χαρακτήρες.
* **status:** Είναι η κατάσταση στην οποία βρίσκεται η συσκευή. Ο τύπος varchar(20) επιλέχθηκε καθώς η κατάσταση περιέχει χαρακτήρες και δεν πρέπει να ξεπερνάει τους 20 χαρακτήρες.

1. Πίνακας indoor\_data

Στον πίνακα 4.4: indoor\_data καταχωρούνται οι μετρήσεις θερμοκρασίας και υγρασίας που συλλέγονται από τους αισθητήρες στον εσωτερικό χώρο.

Ανάλυση των στοιχείων του πίνακα:

* **id:** Είναι ο αναγνωριστικός αύξων αριθμός ο οποίος ξεχωρίζει τις καταγραφές μεταξύ τους. Ο τύπος int(11) επιλέχθηκε γιατί η τιμή των αναγνωριστικών αριθμών είναι πάντα ακέραια. Με την ιδιότητα UNSIGNED εξασφαλίζεται ένα μεγάλο πλήθος για καταχωρήσεις διαφορετικών δεδομένων. Η πρόσθετη επιλογή AUTO\_INCREMENT δείχνει ότι ο αριθμός αυτός αυξάνεται αυτόματα.
* **date:** Η ημερομηνία στην οποία καταγράφηκαν κάποια δεδομένα από τον αισθητήρα. Ο τύπος date επιλέχθηκε καθώς το πεδίο έχει μόνο ημερομηνίες με την μορφή YYYY-MM-DD.
* **time:** Η ώρα στην οποία καταγράφηκαν κάποια δεδομένα από τον αισθητήρα. Ο τύπος time επιλέχθηκε καθώς το πεδίο έχει μόνο την ώρα με την μορφή HH-MM-SS.
* **temperature:** Η τιμή της θερμοκρασίας που καταγράφηκε από τον αισθητήρα. Ο τύπος varchar(10) επιλέχθηκε καθώς ο αριθμός περιέχει και χαρακτήρες όπως είναι το σύμβολο υποδιαστολής.
* **humidity:** Η τιμή της υγρασίας που καταγράφηκε από τον αισθητήρα. Ο τύπος varchar(10) επιλέχθηκε καθώς ο αριθμός περιέχει και χαρακτήρες όπως είναι το σύμβολο υποδιαστολής.

1. Πίνακας outdoor\_data

Στον πίνακα 4.45 outdoor\_data καταχωρούνται οι μετρήσεις θερμοκρασίας και υγρασίας που συλλέγονται από τους αισθητήρες στον εξωτερικό χώρο.

Ανάλυση των στοιχείων του πίνακα:

* **id:** Είναι ο αναγνωριστικός αύξων αριθμός ο οποίος ξεχωρίζει τις καταγραφές μεταξύ τους. Ο τύπος int(11) επιλέχθηκε γιατί η τιμή των αναγνωριστικών αριθμών είναι πάντα ακέραια. Με την ιδιότητα UNSIGNED εξασφαλίζεται ένα μεγάλο πλήθος για καταχωρήσεις διαφορετικών δεδομένων. Η πρόσθετη επιλογή AUTO\_INCREMENT δείχνει ότι ο αριθμός αυτός αυξάνεται αυτόματα.
* **date:** Η ημερομηνία στην οποία καταγράφηκαν κάποια δεδομένα από τον αισθητήρα. Ο τύπος date επιλέχθηκε καθώς το πεδίο έχει μόνο ημερομηνίες με την μορφή YYYY-MM-DD.
* **time:** Η ώρα στην οποία καταγράφηκαν κάποια δεδομένα από τον αισθητήρα. Ο τύπος time επιλέχθηκε καθώς το πεδίο έχει μόνο την ώρα με την μορφή HH-MM-SS.
* **temperature:** Η τιμή της θερμοκρασίας που καταγράφηκε από τον αισθητήρα. Ο τύπος varchar(10) επιλέχθηκε καθώς ο αριθμός περιέχει και χαρακτήρες όπως είναι το σύμβολο υποδιαστολής.
* **humidity:** Η τιμή της υγρασίας που καταγράφηκε από τον αισθητήρα. Ο τύπος varchar(10) επιλέχθηκε καθώς ο αριθμός περιέχει και χαρακτήρες όπως είναι το σύμβολο υποδιαστολής.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **ΟΝΟΜΑ** | **ΤΥΠΟΣ** | **ΣΥΝΘΕΣΗ** | **ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ** | **ΚΕΝΟ** | **ΠΡΟΕΠΙΛΟΓΗ** | **ΠΡΟΣΘΕΤΑ** |
| 1 | id | int(11) | - | - | Όχι | Καμία | AUTO\_INCREMENT |
| 2 | username | varchar(50) | utf8\_general\_ci | - | Όχι | Καμία | - |
| 3 | email | varchar(50) | utf8\_general\_ci | - | Όχι | Καμία | - |
| 4 | adminuser | TINYINT(1) | utf8\_general\_ci | - | Ναι | yes | - |
| 5 | password | varchar(50) | utf8\_general\_ci | - | Όχι | Καμία | - |
| 6 | created\_at | datetime | - | - | Ναι | CURRENT\_TIMESTAMP | - |

***Πίνακας 4.1: Πίνακας Users***

***Πίνακας 4.2: Πίνακας user\_log***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **ΟΝΟΜΑ** | **ΤΥΠΟΣ** | **ΣΥΝΘΕΣΗ** | **ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ** | **ΚΕΝΟ** | **ΠΡΟΕΠΙΛΟΓΗ** | **ΠΡΟΣΘΕΤΑ** |
| 1 | id | int(11) | - | - | Όχι | Καμία | AUTO\_INCREMENT |
| 2 | user\_id | varchar(20) | utf8\_general\_ci | - | Όχι | Καμία | - |
| 3 | date | date | - | - | Ναι | Καμία | - |
| 4 | time | time | - | - | Ναι | Καμία | - |
| 5 | log | varchar(100) | utf8\_general\_ci | - | Όχι | Καμία | - |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **ΟΝΟΜΑ** | **ΤΥΠΟΣ** | **ΣΥΝΘΕΣΗ** | **ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ** | **ΚΕΝΟ** | **ΠΡΟΕΠΙΛΟΓΗ** | **ΠΡΟΣΘΕΤΑ** |
| 1 | id | int(11) | - | - | Όχι | Καμία | AUTO\_INCREMENT |
| 2 | devicename | varchar(20) | utf8\_general\_ci | - | Όχι | Καμία | - |
| 3 | status | varchar(10) | utf8\_general\_ci | - | Όχι | Καμία | - |

***Πίνακας 4.3: Πίνακας devices\_states***

***Πίνακας 4.4: Πίνακας Indoor\_data***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **ΟΝΟΜΑ** | **ΤΥΠΟΣ** | **ΣΥΝΘΕΣΗ** | **ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ** | **ΚΕΝΟ** | **ΠΡΟΕΠΙΛΟΓΗ** | **ΠΡΟΣΘΕΤΑ** |
| 1 | id | int(11) | - | - | Όχι | Καμία | AUTO\_INCREMENT |
| 2 | date | date | - | - | Ναι | Καμία | - |
| 3 | time | time | - | - | Όχι | Καμία | - |
| 4 | temperature | varchar(10) | utf8\_general\_ci | - | Ναι | Καμία | - |
| 5 | humidity | varchar(10) | utf8\_general\_ci | - | Όχι | Καμία | - |

***Πίνακας 4.5: Πίνακας outdoor\_data***

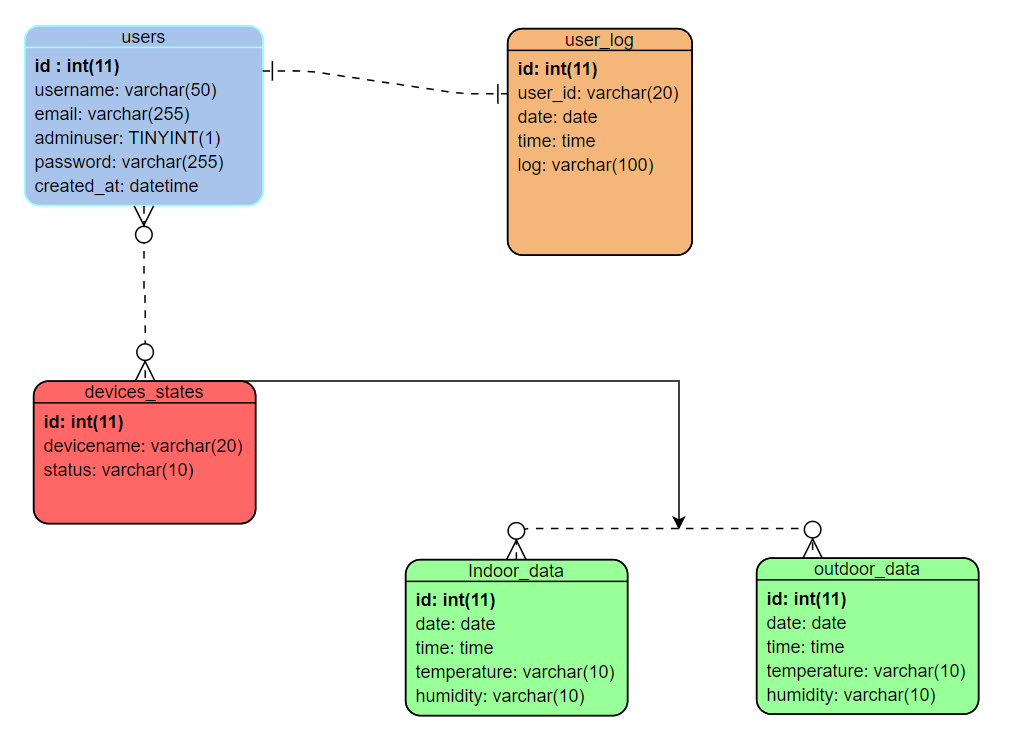
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **ΟΝΟΜΑ** | **ΤΥΠΟΣ** | **ΣΥΝΘΕΣΗ** | **ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ** | **ΚΕΝΟ** | **ΠΡΟΕΠΙΛΟΓΗ** | **ΠΡΟΣΘΕΤΑ** |
| 1 | id | int(11) | - | - | Όχι | Καμία | AUTO\_INCREMENT |
| 2 | date | date | - | - | Όχι | Καμία | - |
| 3 | time | time | - | - | Όχι | Καμία | - |
| 4 | temperature | varchar(10) | utf8\_general\_ci | - | Όχι | Καμία | - |
| 5 | humidity | varchar(10) | utf8\_general\_ci | - | Όχι | Καμία | - |

1. Συσχετισμός μεταξύ πινάκων της βάσης δεδομένων

Όπως παρατηρήθηκε και στην προηγούμενη παράγραφο, κάποια στοιχεία πινάκων βρίσκονται με το ίδιο ή διαφορετικό όνομα και σε άλλους πίνακες. Με τον τρόπο αυτό επιτυγχάνεται η συσχέτιση ενός με έναν ή περισσότερους πίνακες.

Ως αρχικός πίνακας, ο πίνακας users δεν περιέχει κανένα στοιχείο κάποιου άλλου. Παράλληλα όμως, ο αναγνωριστικός αριθμός αυτού (στοιχείο id) βρίσκεται στον πίνακα user\_log με όνομα user\_id, έτσι η κάθε καταγεγραμμένη δραστηριότητα αντιστοιχεί στον χρήστη που ανήκει.

Όσο αφορά τους πίνακες indoor\_data & outdoor\_data, περιέχουν ίδιου τύπου πληροφορίες και διαχειρίζονται με τον ίδιο τρόπο. Η διαφορά τους είναι πώς προέρχονται από διαφορετικούς χώρους του σπιτιού.

Ακόμη, ο πίνακας devices\_states έχει άμεση σχέση με τους δύο πίνακες indoor\_data & outdoor\_data, καθώς ανάλογα με τις πληροφορίες που έχει ο χρήστης από τους πίνακες θα λάβει την απόφαση να αλλάξει την κατάσταση στις έξυπνες συσκευές. Παράλληλα όμως, θα πρέπει να έχει και τα δικαιώματα τα οποία δίνονται από το πεδίο adminuser του πίνακα users.

***Σχήμα 4.2 Συσχετισμός πινάκων της βάσης δεδομένων.***

* 1. **Περιγραφή λειτουργίας διαδικτυακής εφαρμογής Smart Homie**

Η διαδικτυακή εφαρμογή Smart Homie έχει σκοπό στη παρακολούθηση των έξυπνων συσκευών ενός σπιτιού καθώς και η μέτρηση της θερμοκρασίας και υγρασίας στους εσωτερικούς & εξωτερικούς χώρους του σπιτιού. Οι λειτουργίες αυτής διατυπώθηκαν και υλοποιήθηκαν από τις απαιτήσεις των πιθανών χρηστών και κατ’ επέκταση του συστήματος. Η διαδικτυακή εφαρμογή χωρίζεται σε έξι βασικές καρτέλες, οι οποίες εμφανίζονται μετά την είσοδο του χρήστη στην εφαρμογή. Υπάρχουν επίσης δύο καρτέλες οι οποίες αφορούν στην εγγραφή και είσοδο του χρήστη. Αξίζει να σημειωθεί ότι παρά το γεγονός ότι, η γλώσσα της διαδικτυακής εφαρμογής είναι η αγγλική, οι ετικέτες οι τίτλοι, τα διαγράμματα καθώς και τα ενημερωτικά μηνύματα είναι διατυπωμένα και έχουν συνταχθεί με τον πιο δυνατό κατανοητό τρόπο ακόμα και για χρήστης με χαμηλή γνώση της αγγλικής γλώσσας.

* + 1. **Λειτουργία εγγραφής και λειτουργία σύνδεσης (Register & Log In)**