



# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

## **ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

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

#### **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Συσκευή συλλογής δεδομένων και επικοινωνία με server με την  
χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη  
ενός smart home**

**Γιαννακέας Σταύρος-Κωνσταντίνος  
Α.Μ. 18390187**

**Εισηγητής: Χρήστος Τρούσσας, Επ. Καθηγητής**

**Αθήνα, Δεκέμβριος 2023**

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home

## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home**

**Γιαννακέας Σταύρος-Κωνσταντίνος  
Α.Μ. 18390187**

**Εισηγητής:**

**Χρήστος Τρούσσας, Επ. Καθηγητής**

**Εξεταστική Επιτροπή:**

**ΑΚΡΙΒΗ, ΚΡΟΥΣΚΑ, Μέλος ΕΔΙΠ  
ΠΑΝΑΓΙΩΤΑ, ΤΣΕΛΕΝΤΗ, Μέλος ΕΔΙΠ**

**Ημερομηνία εξέτασης 09/2024**

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home

## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Γιαννακέας Σταύρος - Κωνσταντίνος του Νικολάου, με αριθμό μητρώου 18390187 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά ειδικά για την συγκεκριμένη διπλωματική εργασία και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Βεβαιώνω ότι έχω αναφέρει ή παραπέμπει σε αυτή την Διπλωματική εργασία, αναφέρονται όλες τις πηγές από τις οποίες έκανα χρήση δεδομένων».

Ο Δηλών



Γιαννακέας Σταύρος - Κωνσταντίνος

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό των εφαρμογών στο Διαδίκτυο των αντικειμένων. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής Δρ. Χρήστος Τρούσσας, στον οποίο θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες, για την συνεχή καθοδήγησή του και προθυμία να βοηθήσει σε οποιοδήποτε απορία προέκυπτε. Ακόμα θα

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home

ήθελα να ευχαριστήσω την οικογένειά μου για τη συμπαράσταση κατά τη διάρκεια των σπουδών μου.

## ΠΕΡΙΛΗΨΗ

Η διπλωματική εργασία αυτή επικεντρώνεται στην ανάπτυξη μιας συσκευής που συλλέγει περιβαλλοντικά δεδομένα από αισθητήρες και τα μεταφέρει σε ένα διακομιστή για περαιτέρω ανάλυση. Σε έναν κόσμο όπου η τεχνολογία διαδραματίζει ολοένα και μεγαλύτερο ρόλο στην καθημερινότητά μας, η ψηφιοποίηση της αλληλεπίδρασης με το περιβάλλον γίνεται ολοένα πιο σημαντική.

Η εφαρμογή που περιγράφετε συνδυάζει την υλοποίηση του hardware μέρους με την ανάπτυξη του software που διαχειρίζεται και μεταφέρει τα συλλεγόμενα δεδομένα. Σε συνδυασμό με την ευρύτερη εξέλιξη προς την "έξυπνη" οικιακή τεχνολογία, η εφαρμογή αυτή αντιπροσωπεύει ένα βήμα προς την αναβάθμιση του περιβάλλοντος και την βελτίωση της αλληλεπίδρασης του ανθρώπου με το περιβάλλον του στο σπίτι.

Συνοψίζοντας, η εργασία επικεντρώνεται στην ανάπτυξη μιας ολοκληρωμένης λύσης που συνδυάζει την τεχνολογία με το περιβάλλον, βελτιώνοντας έτσι την καθημερινότητα των ανθρώπων.



ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Δικτύωση, IoT, Έξυπνο Σπίτι

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αυτοματοποίηση, Αισθητήρες, Επικοινωνία

## **ABSTRACT**

This thesis focuses on the development of a device that collects environmental data from sensors and transfers it to a server for further analysis. In a world where technology plays an increasingly important role in our daily lives, the digitization of interaction with the environment is becoming increasingly important.

The application you describe combines the implementation of the hardware part with the development of the software that manages and transfers the collected data. Combined with the wider evolution towards "smart" home technology, this application represents a step towards upgrading the environment and improving the interaction of man with his environment at home.

In summary, the work focuses on developing an integrated solution that combines technology with the environment, thus improving people's daily lives.

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home

**SCIENTIFIC AREA:** Networking, IoT, Smart Home

**KEYWORDS:** Automation, Sensors, Communication

## Περιεχόμενα

<b>ΠΕΡΙΛΗΨΗ .....</b>	<b>7</b>
<b>ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ .....</b>	<b>12</b>
<b>ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ.....</b>	<b>12</b>
<b>ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....</b>	<b>12</b>
<b>ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ .....</b>	<b>12</b>
<b>ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ .....</b>	<b>15</b>
<b>ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ.....</b>	<b>15</b>
<b>2.1 ΥΛΙΚΟ (HARDWARE) .....</b>	<b>16</b>
2.1.1 ARDUINO .....	16
2.1.2 NODEMCU .....	19
2.1.3 ΑΙΣΘΗΤΗΡΕΣ / ΕΝΕΡΓΟΠΟΙΗΤΕΣ.....	21
<b>2.2 ΔΙΑΔΙΚΤΥΟ .....</b>	<b>22</b>
2.2.1 ΠΡΩΤΟΚΟΛΛΟ ΔΙΑΔΙΚΤΥΟΥ / INTERNET PROTOCOL (IP) .....	23
2.2.2 CLIENT-SERVER .....	24
2.2.3 WEBSOCKETS.....	25
2.2.4 ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ .....	26
<b>ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΑ.....</b>	<b>30</b>
<b>3.1 ΣΤΟΧΟΣ ΚΑΙ ΤΑ ΥΠΟΠΡΟΒΛΗΜΑΤΑ .....</b>	<b>30</b>
<b>3.2 ΠΡΩΙΜΑ ΣΤΑΔΙΑ .....</b>	<b>30</b>
<b>3.3 ΕΜΠΟΔΙΑ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ .....</b>	<b>32</b>
<b>ΚΕΦΑΛΑΙΟ 4: ΤΟ ΣΥΣΤΗΜΑ ΜΟΥ .....</b>	<b>33</b>

<b>4.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΡΓΑΣΙΑΣ.....</b>	<b>34</b>
<b>4.2 ARDUINO .....</b>	<b>36</b>
4.2.1 ΒΙΒΛΙΟΘΗΚΗ Arduino.h.....	36
4.2.2 ΒΙΒΛΙΟΘΗΚΗ DHT_U.h και DHT.h .....	40
4.2.3 ΒΙΒΛΙΟΘΗΚΗ LiquidCrystal.h .....	43
4.2.4 ΒΙΒΛΙΟΘΗΚΗ Wire.h .....	48
4.2.5 ΒΙΒΛΙΟΘΗΚΗ SoftwareSerial.h .....	49
4.2.6 ΛΕΙΤΟΥΡΓΙΑ ARDUINO .....	51
<b>4.3 ESP.....</b>	<b>Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.</b>
<b>4.3.1 ΒΙΒΛΙΟΘΗΚΗ ESP8266WiFi.h .....</b>	<b>54</b>
4.3.2 ΒΙΒΛΙΟΘΗΚΗ WebSocketsServer.h .....	55
4.3.3 ΛΕΙΤΟΥΡΓΙΑ ESP .....	56
<b>4.4 SERVER.....</b>	<b>58</b>
4.4.1 ΒΙΒΛΙΟΘΗΚΗ threading .....	58
4.4.2 ΒΙΒΛΙΟΘΗΚΗ asyncio .....	59
4.4.3 ΒΙΒΛΙΟΘΗΚΗ websockets .....	60
4.4.4 ΒΙΒΛΙΟΘΗΚΗ thingspeak .....	61
4.4.5 ΛΕΙΤΟΥΡΓΙΑ SERVER.....	62
<b>ΚΕΦΑΛΑΙΟ 5: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ .....</b>	<b>65</b>
<b>5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ .....</b>	<b>65</b>
<b>5.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΞΕΛΙΞΕΙΣ .....</b>	<b>65</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>67</b>

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

<b>Εικόνα 2.1:</b> Ποικιλία πλακετών Arduino [1].....	18
<b>Εικόνα 2.2:</b> Πάνω όψη και τεχνικές λεπτομέρειες Arduino UNO [2].....	19
<b>Εικόνα 2.3:</b> Πάνω όψη και τεχνικές λεπτομέρειες NodeMCU [3].....	21
<b>Εικόνα 4.1:</b> Αρχιτεκτονική IoT εφαρμογής [58] .....	35
<b>Εικόνα 4.2:</b> Δίοδος εκπομπής φωτός και ο συμβολισμός της [6].....	39
<b>Εικόνα 4.3:</b> Φωτοδίοδος και ο συμβολισμός της [5] .....	39
<b>Εικόνα 4.4:</b> Συνδεσμολογία Arduino με απλά components [10] .....	41
<b>Εικόνα 4.5:</b> Τεχνικές λεπτομέρειες αισθητήρα DHT22 [4] .....	43
<b>Εικόνα 4.6:</b> Συνδεσμολογία Arduino με αισθητήρες [10].....	44
<b>Εικόνα 4.7:</b> LCD display και τεχνικές λεπτομέρειες [7].....	46
<b>Εικόνα 4.8:</b> Ποτενσιόμετρο και τεχνικές λεπτομέρειες [8] .....	47
<b>Εικόνα 4.9:</b> Συνδεσμολογία Arduino με LCD οθόνη [10] .....	48
<b>Εικόνα 4.10:</b> Τελική συνδεσμολογία Arduino [10] .....	51
<b>Εικόνα 4.11:</b> Ενδεικτικά διαγράμματα thingspeak με τιμές [11] .....	62

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

<b>Σχήμα 3.1:</b> Στάδια υλοποίησης εφαρμογής [9].....	32
<b>Σχήμα 3.2:</b> Σταδιακή υλοποίησης εφαρμογής [9].....	33
<b>Σχήμα 4.1:</b> Δομή και τοπολογία εφαρμογής [9].....	37
<b>Σχήμα 4.2:</b> Δομή μηνυμάτων Arduino [10].....	54

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

<b>Πίνακας 4.1:</b> Συνδεσμολογία φωτοδίοδου .....	41
<b>Πίνακας 4.2:</b> Συνδεσμολογία LED.....	41
<b>Πίνακας 4.3:</b> Συνδεσμολογία DHT22.....	44
<b>Πίνακας 4.4:</b> Συνδεσμολογία LCD οθόνης.....	49
<b>Πίνακας 4.5:</b> Συνδεσμολογία Ποτενσιόμετρου.....	49
<b>Πίνακας 4.6:</b> Συνδεσμολογία ESP.....	51

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

**IDE** Ενσωματωμένο Περιβάλλον Ανάπτυξης (Integrated Development Environment)

**TCP** Πρωτόκολλο Ελέγχου Μετάδοσης (Transmission Control Protocol)

**UI** Διεπαφή Χρήστη (User Interface)

**URL** Ενιαίος Εντοπιστής Πόρων (Uniform Resource Locator)

**HTML** Γλώσσα Σήμανσης Υπερκειμένου (HyperText Markup Language)

**CSS** Φύλλα Στυλ Συνοδευόμενης Μορφής (Cascading Style Sheets)

**LED** Δίοδος Εκπομπής Φωτός (Light Emitting Diode)

**LCD** Οθόνη Υγρών Κρυστάλλων (Liquid Crystal Display)

**DHT** Ψηφιακός Αισθητήρας Υγρασίας και Θερμοκρασίας (Digital Humidity and Temperature sensor)

**DIY** Κάν' το μόνος σου (Do It Yourself)

**MQTT** Πρωτόκολλο Μηνυμάτων Ανταλλαγής (Message Queuing Telemetry Transport)

**GPIO** Γενική Προγραμματιζόμενη Είσοδος/Εξόδος (General Purpose Input/Output)

**IP** Πρωτόκολλο Διαδικτύου (Internet Protocol)

**HTTP** Πρωτόκολλο Υπερκειμένου Μετάδοσης (Hypertext Transfer Protocol)

**FTP** Πρωτόκολλο Μεταφοράς Αρχείων (File Transfer Protocol)

**UDP** Πρωτόκολλο Διάσπασης Ενός Μηνύματος (User Datagram Protocol)

**URL** Ομοιόμορφος Πόρος Τοποθεσίας (Uniform Resource Locator)

**UUID** Μοναδικός Αναγνωριστικός Καθοριστής Παγκόσμιας Οντότητας (Universally Unique Identifier)

**IoT** Διαδίκτυο των Πραγμάτων (Internet of Things)

**UART** Διασύνδεση Ανάστροφης Αναμονής/Χρονοθυρίδας (Universal Asynchronous Receiver/Transmitter)

**I2C** Διασύνδεση Ενιαίας Μετάδοσης/Διαδικτύου (Inter-Integrated Circuit)

**SPI** Σειριακή Διεπαφή Προγραμματισμού (Serial Peripheral Interface)

**SSID** Ταυτότητα Περιοχής Εκπομπής (Service Set Identifier)

**WiFi** Ασύρματη Δικτύωση (Wireless Fidelity)

**API** Διεπαφή Προγραμματισμού Εφαρμογής (Application Programming Interface)

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home

## **ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ**

Η σύγχρονη εποχή διακατέχεται από την ταχεία εξέλιξη της τεχνολογίας και την ενσωμάτωσή της στην καθημερινότητά μας. Αυτή η προοδευτική διαδικασία έχει διαμορφώσει νέους τρόπους προσέγγισης σε πολλούς τομείς της ζωής, συμπεριλαμβανομένης και της οικιακής ζωής και της περιβαλλοντικής ευαισθητοποίησης. Σε αυτό το πλαίσιο, η παρούσα διπλωματική εργασία επικεντρώνεται στην ανάπτυξη μιας συσκευής για τη συλλογή, την ανάλυση και την αποστολή δεδομένων περιβαλλοντικών μετρήσεων από τον χώρο του σπιτιού, προσφέροντας ένα ολοκληρωμένο σύστημα για την παρακολούθηση και την ανάλυση του περιβάλλοντος.

Η ανάπτυξη της έξυπνης συσκευής βασίζεται σε μια συνέργεια πολλαπλών τεχνολογικών πτυχών. Τα δίκτυα αισθητήρων, η ανάλυση δεδομένων, η ασύρματη επικοινωνία και η προηγμένη προγραμματιζόμενη λογική συνδυάζονται για να δημιουργήσουν ένα ευφύες περιβάλλον. Οι τεχνολογικές εξελίξεις αυτές αποτελούν τη βάση για την υλοποίηση ενός πλήρους συστήματος παρακολούθησης που επιτρέπει στους χρήστες να παρακολουθούν και να διαχειρίζονται τις παραμέτρους περιβάλλοντος τους.

Οι κυριότερες αρχές που καθοδήγησαν την υλοποίηση αυτής της εφαρμογής βασίζονται στην επιθυμία για μια έξυπνη και ενεργειακά αποδοτική οικιακή περιβαλλοντική λύση. Η συλλογή περιβαλλοντικών δεδομένων γίνεται για να παρέχει αναλυτική ενημέρωση για τη θερμοκρασία, την υγρασία και την φωτεινότητα στο εσωτερικό του σπιτιού, ενώ παράλληλα προσφέρει τη δυνατότητα προσωπικής προσαρμογής του περιβάλλοντος σύμφωνα με τις προτιμήσεις του καθενός.

Μέσα από την εφαρμογή της τεχνολογίας σε πρακτικές εφαρμογές όπως αυτή, προσδίδεται νέα διάσταση στην έννοια του έξυπνου σπιτιού, επιτρέποντας τη συλλογή και την ανάλυση δεδομένων που έχουν άμεση σχέση με την καθημερινότητα και την ποιότητα ζωής. Η διασύνδεση των αισθητήρων με την ασύρματη μεταφορά δεδομένων παρέχει μια ολοκληρωμένη λύση που ανταποκρίνεται στις σύγχρονες ανάγκες ενός εξυπηρετικού, έξυπνου και φιλικού περιβάλλοντος κατοικίας.

## **ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ**



Σε αυτό το κεφάλαιο θα παρουσιαστούν τα θεωρητικά θεμέλια που υποστηρίζουν την ανάπτυξη του συστήματος έξυπνου σπιτιού με χρήση της πλατφόρμας Arduino. Θα εξεταστούν τα κύρια στοιχεία του υλικού που χρησιμοποιήθηκαν, καθώς και οι βασικές τεχνολογίες και πρωτόκολλα που αξιοποιήθηκαν για τη λειτουργία του συστήματος. Ειδικότερα, θα αναλύσουμε τη λειτουργία και τις δυνατότητες του Arduino, καθώς και άλλων σχετικών συσκευών και βιβλιοθηκών που συνέβαλαν στην υλοποίηση του έργου.

## **2.1 ΥΛΙΚΟ (HARDWARE)**

Το υλικό (hardware) αποτελεί τον πυρήνα κάθε συστήματος αυτοματισμού και ελέγχου. Στο πλαίσιο της ανάπτυξης ενός έξυπνου σπιτιού με χρήση Arduino, είναι κρίσιμο να κατανοήσουμε τις διάφορες συνιστώσες του υλικού και τον τρόπο με τον οποίο συνεργάζονται. Σε αυτή την ενότητα, θα παρουσιαστούν οι βασικές μονάδες και τα στοιχεία που απαρτίζουν το σύστημα, όπως οι πλακέτες Arduino, οι αισθητήρες, οι ενεργοποιητές και οι συσκευές επικοινωνίας. Επιπλέον, θα εξεταστούν τα χαρακτηριστικά και οι δυνατότητες του κάθε στοιχείου, προσφέροντας μια ολοκληρωμένη εικόνα για την επιλογή και χρήση του κατά την υλοποίηση του έργου. Η κατανόηση του υλικού είναι θεμελιώδης για τη σωστή σχεδίαση και βελτιστοποίηση του συστήματος, επιτρέποντας την επίτευξη των στόχων του έργου με αποτελεσματικό και αποδοτικό τρόπο.

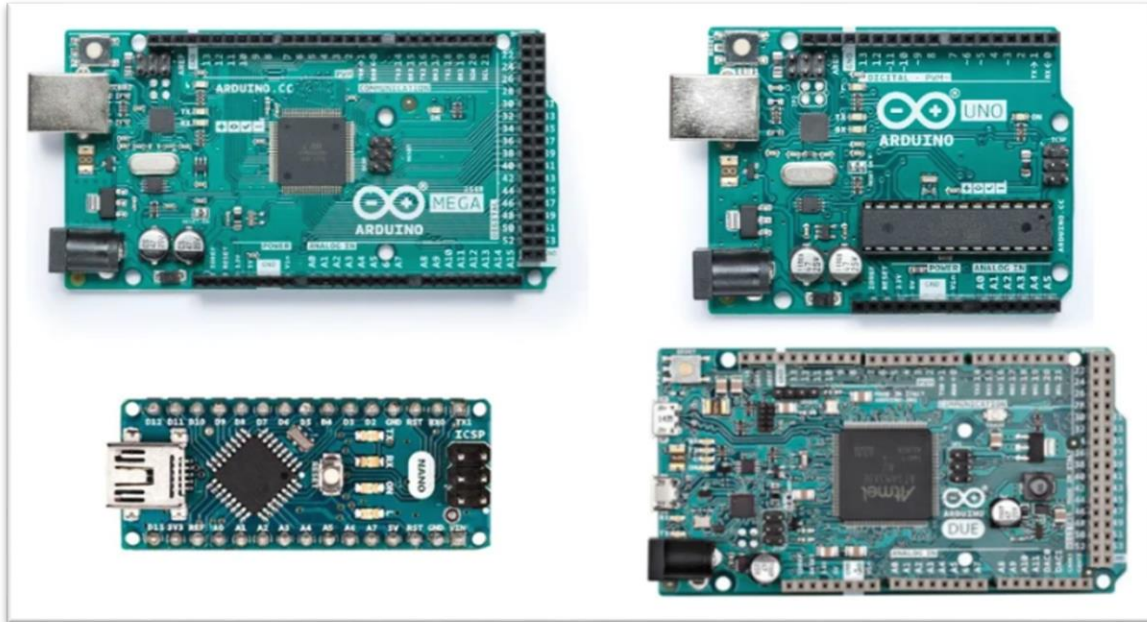
### **2.1.1 ARDUINO**

Το Arduino είναι μια ανοιχτού κώδικα πλατφόρμα ανάπτυξης υλικού και λογισμικού που ξεκίνησε το 2005 στο Interaction Design Institute Ivrea στην Ιταλία[12]. Αρχικά σχεδιάστηκε για να παρέχει στους φοιτητές έναν προσιτό και εύχρηστο τρόπο για να δημιουργούν συσκευές που αλληλεπιδρούν με το περιβάλλον τους χρησιμοποιώντας αισθητήρες και ενεργοποιητές. Με την πάροδο του χρόνου, οι πλακέτες Arduino εξελίχθηκαν και χρησιμοποιούνται σήμερα για μια πληθώρα εφαρμογών, όπως ρομπότ, θερμοστάτες, ανιχνευτές κίνησης, και πολλά άλλα[13]. Η ονομασία "Arduino" προέρχεται από ένα μπαρ στην Ivrea της Ιταλίας[14], όπου συνήθιζαν να συναντιούνται οι ιδρυτές του έργου.

Η πλατφόρμα Arduino βασίζεται σε έναν μικροελεγκτή, που είναι μια μικρή πλακέτα που περιλαμβάνει έναν επεξεργαστή, και παρέχει ένα περιβάλλον ανάπτυξης που είναι φιλικό προς τον χρήστη[15]. Η δημοτικότητα του Arduino στο κόσμο του DIY (Do It Yourself) και της πρωτοβουλίας maker οφείλεται στην ανοιχτού κώδικα φύση του και στη δυνατότητα που παρέχει στους χρήστες να δημιουργούν και να προσαρμόζουν τις δικές τους εφαρμογές. Χρησιμοποιώντας το Arduino IDE, ένα εύχρηστο περιβάλλον προγραμματισμού, οι χρήστες μπορούν να γράφουν κώδικα σε μια γλώσσα που

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home

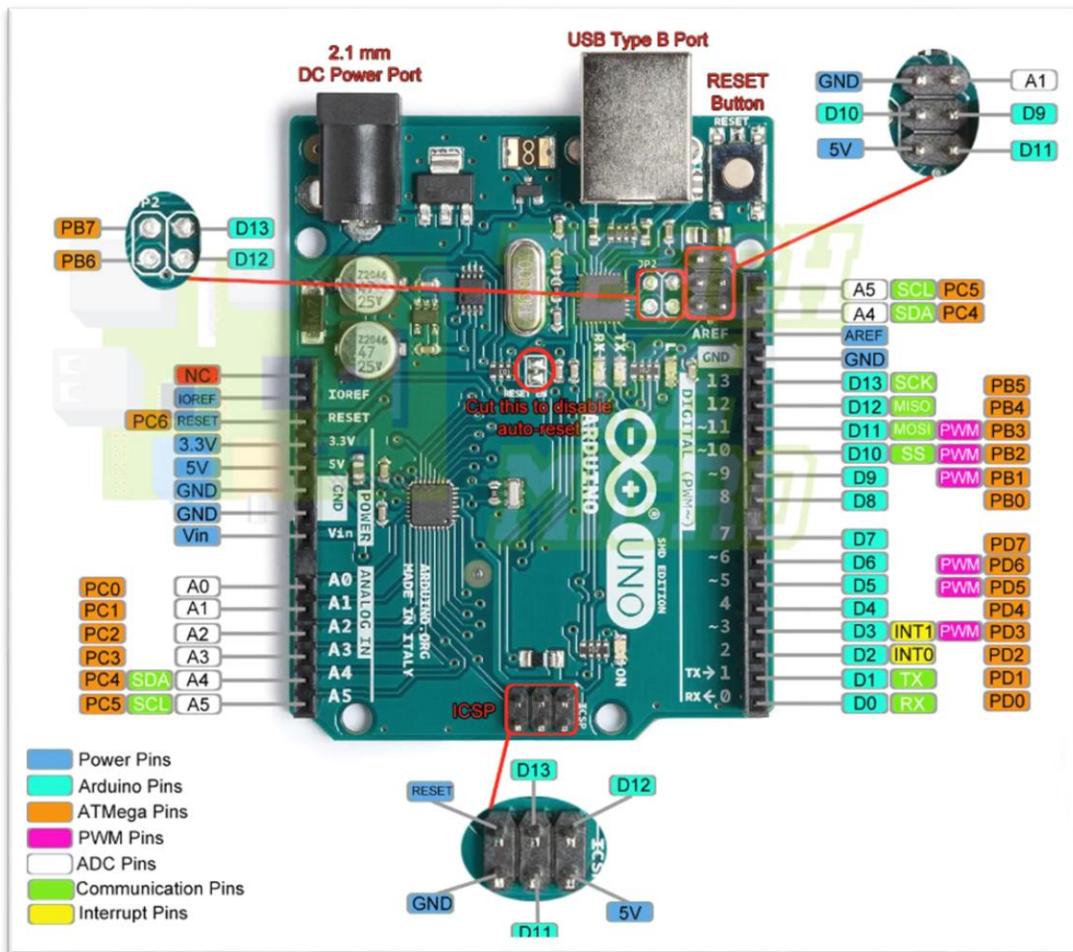
βασίζεται στη C/C++, να τον μεταγλωττίζουν και να τον μεταφορτώνουν στην πλακέτα Arduino για να εκτελέσει τις επιθυμητές λειτουργίες.



**Εικόνα 2.1 :** Ποικιλία πλακετών Arduino [1]

Οι πλακέτες Arduino διατίθενται σε διάφορα μοντέλα με διαφορετικά χαρακτηριστικά και μεγέθη. Η ποικιλία των διαθέσιμων πλακετών και η ευρεία υποστήριξη από την κοινότητα καθιστούν το Arduino ένα ισχυρό εργαλείο για κάθε είδους έργα. Οι χρήστες μπορούν να βρουν άφθονη τεκμηρίωση, παραδείγματα κώδικα, και tutorials, που τους βοηθούν να ξεκινήσουν και να προχωρήσουν στις δικές τους δημιουργίες. Οι πλακέτες αυτές είναι συνήθως εξοπλισμένες με ψηφιακές και αναλογικές εισόδους και εξόδους, οι οποίες επιτρέπουν τη σύνδεση με αισθητήρες, κινητήρες, οθόνες και άλλες συσκευές[16]. Για παράδειγμα, η Arduino Uno, ένα από τα πιο δημοφιλή μοντέλα, έχει τις ακόλουθες επιμέρους υλικές δυνατότητες όπως φαίνεται στο παρακάτω σχήμα:

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home



Εικόνα 2.2 : Πάνω όψη και τεχνικές λεπτομέρειες Arduino UNO [2]

### Επιμέρους στοιχεία υλικού

- **Ψηφιακές ακίδες (2-13):** Χρησιμοποιούνται με τις προγραμματιστικές λειτουργίες `digitalRead()`, `digitalWrite()` και `analogWrite()`. Η `analogWrite()` λειτουργεί μόνο στις ακίδες με το σύμβολο PWM (Pulse Width Modulation)(3, 5-6, 9-11).
- **LED (pin 13):** Το LED (light emitting diode) αυτό είναι ένας ενσωματωμένος ενεργοποιητής, χρήσιμος για τον εντοπισμό σφαλμάτων.
- **Ενδεικτική λυχνία λειτουργίας:** Υποδεικνύει ότι το Arduino ρευματοδοτείται
- **Μικροελεγκτής:** ATmega
- **Αναλογικές ακίδες (A0-A5):** Χρησιμοποιούνται με την προγραμματιστική λειτουργία `analogRead()`.
- **Ακίδες GND και 5V:** Μέσω αυτών παρέχεται ισχύς +5V και γείωση σε πρόσθετες κάρτες του Arduino.
- **Υποδοχή τροφοδοσίας:** Σύνδεση τροφοδοσίας ρεύματος του Arduino, όταν δεν τροφοδοτείται μέσω της θύρας USB. Δέχεται τάσεις μεταξύ 7-12V.

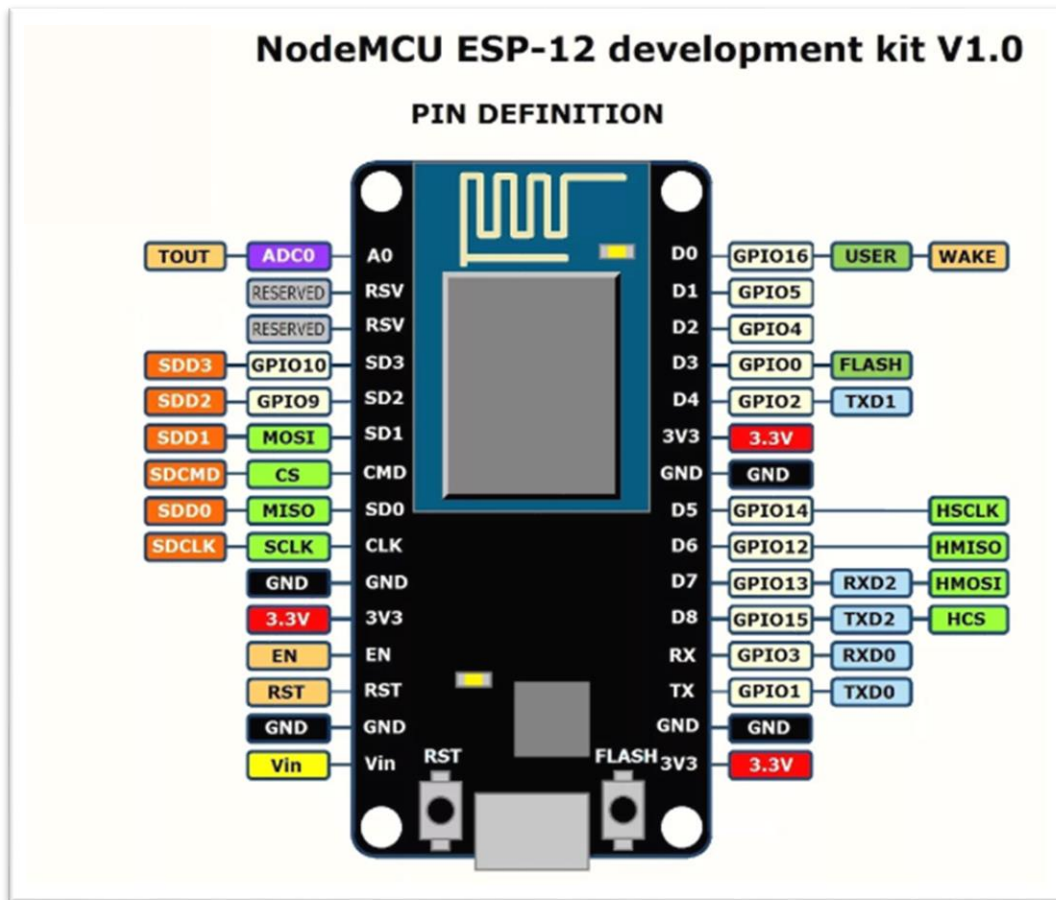
- **TX και RX LED:** Αυτές οι λυχνίες αποδεικνύουν την επικοινωνία μεταξύ του Arduino και του υπολογιστή. Αναμένεται να τρεμοπαίζουν γρήγορα κατά τη μεταφόρτωση κώδικα καθώς και κατά τη σειριακή επικοινωνία. Χρήσιμο για εντοπισμό σφαλμάτων.
- **Θύρα USB:** Χρησιμοποιείται για την τροφοδοσία του Arduino Uno, τη μεταφόρτωση κώδικα στο Arduino και για την επικοινωνία του Η/Υ με το Arduino (π.χ. με χρήση της προγραμματιστικής λειτουργίας `Serial.println()`).
- **Κουμπί επαναφοράς:** Επαναφέρει τον μικροελεγκτή ATmega[17].

Το Arduino συνεχίζει να εξελίσσεται, προσφέροντας νέα μοντέλα και λειτουργίες που καλύπτουν τις ανάγκες τόσο των αρχαρίων όσο και των έμπειρων χρηστών. Η πλατφόρμα ενσωματώνει συνεχώς νέες τεχνολογίες και τάσεις, καθιστώντας την ένα αναπόσπαστο κομμάτι της σύγχρονης ηλεκτρονικής και του προγραμματισμού.

### 2.1.2 NODEMCU

Το NodeMCU είναι μια από τις πιο διαδεδομένες και ευέλικτες πλατφόρμες για έργα Διαδικτύου των Πραγμάτων (IoT), βασισμένη αρχικά στο ESP8266 και αργότερα επεκταμένη στο ESP32. Το όνομα NodeMCU αναφέρεται τόσο στη firmware όσο και στο hardware της πλατφόρμας. Η firmware γράφεται κυρίως σε Lua[18], μια γλώσσα προγραμματισμού ελαφριά και εύκολη στη χρήση, ενώ το hardware είναι σχεδιασμένο να επιτρέπει την εύκολη ανάπτυξη πρωτοτύπων και την ενσωμάτωση σε διάφορα έργα.

Το hardware του NodeMCU είναι σχεδιασμένο για ευκολία χρήσης και ανάπτυξης. Συνήθως, αποτελείται από έναν πίνακα DIP με ελεγκτή USB, έναν μικροελεγκτή (MCU) και μια κεραία. Αυτή η σχεδίαση επιτρέπει τη χρήση του NodeMCU σε breadboards για γρήγορη ανάπτυξη πρωτοτύπων και δοκιμών. Η πρώτη έκδοση του NodeMCU ήταν η 0.9, ακολουθούμενη από τη βελτιωμένη έκδοση 1.0, η οποία περιλαμβάνει τη μονάδα ESP-12E[19]. Αυτή η έκδοση προσφέρει καλύτερη απόδοση και περισσότερες δυνατότητες συνδεσιμότητας, καθιστώντας την ακόμα πιο ιδανική για απαιτητικά έργα IoT.



Εικόνα 2.3 : Πάνω όψη και τεχνικές λεπτομέρειες NodeMCU [3]

Η ιστορία του NodeMCU ξεκινά το 2014, λίγο μετά την έναρξη της παραγωγής του ESP8266 από την Espressif Systems[20]. Το ESP8266 ήταν ένας επαναστατικός μικροελεγκτής, παρέχοντας ενσωματωμένες δυνατότητες Wi-Fi σε χαμηλή τιμή. Το NodeMCU ανέλαβε να εκμεταλλευτεί αυτές τις δυνατότητες, δημιουργώντας μια πλατφόρμα που θα επέτρεπε στους προγραμματιστές να δημιουργήσουν IoT εφαρμογές με ευκολία. Η πλατφόρμα εξελίχθηκε γρήγορα, ενσωματώνοντας ανοιχτό υλικό και υποστήριξη για διάφορα πρωτόκολλα, όπως το MQTT[21], το οποίο είναι ευρέως χρησιμοποιούμενο για την επικοινωνία μεταξύ συσκευών IoT.

Ένα από τα κύρια πλεονεκτήματα του NodeMCU είναι η συμβατότητά του με το περιβάλλον Arduino IDE μέσω του πυρήνα ESP8266 Arduino Core. Αυτή η συμβατότητα επιτρέπει στους προγραμματιστές να χρησιμοποιούν τη γνωστή και φιλική προς το χρήστη πλατφόρμα ανάπτυξης του Arduino για την ανάπτυξη εφαρμογών στο NodeMCU. Προσφέρει επίσης πρόσβαση σε GPIO (General Purpose Input/Output) και σε έναν πίνακα αντιστοίχισης που διευκολύνει την ανάπτυξη έργων. Αυτή η ευκολία χρήσης και η υποστήριξη πολλών εργαλείων και βιβλιοθηκών καθιστούν το NodeMCU μια κορυφαία επιλογή για έργα βασισμένα στο ESP8266 και το ESP32.

Συνολικά, το NodeMCU προσφέρει μια εξαιρετική πλατφόρμα για την ανάπτυξη έργων IoT, παρέχοντας τη δυνατότητα στους προγραμματιστές να δημιουργήσουν πρωτότυπα γρήγορα και αποτελεσματικά. Η ανοιχτή αρχιτεκτονική της πλατφόρμας και η υποστήριξη από μια ενεργή κοινότητα εξασφαλίζουν ότι το NodeMCU θα συνεχίσει να εξελίσσεται και να υποστηρίζει τις ανάγκες των προγραμματιστών σε όλο τον κόσμο.

### **2.1.3 ΑΙΣΘΗΤΗΡΕΣ / ΕΝΕΡΓΟΠΟΙΗΤΕΣ**

Καθώς το Arduino θεωρείται ο εγκέφαλος του συστήματος, οι αισθητήρες και οι ενεργοποιητές αποτελούν τα άκρα του. Οι αισθητήρες και οι ενεργοποιητές λειτουργούν ως είσοδοι και έξοδοι για τους μικροελεγκτές, επιτρέποντας την αλληλεπίδραση του συστήματος με το περιβάλλον.

Η χρήση των αισθητήρων και των ενεργοποιητών είναι κρίσιμη για τη λειτουργία των συστημάτων Arduino, καθώς επιτρέπουν την επικοινωνία του συστήματος με τον εξωτερικό κόσμο. Με τους αισθητήρες να συλλέγουν δεδομένα και τους ενεργοποιητές να εκτελούν ενέργειες, τα συστήματα Arduino μπορούν να υλοποιούν μια ποικιλία εφαρμογών που αλληλεπιδρούν δυναμικά με το περιβάλλον τους.

#### **2.1.3.1 ΑΙΣΘΗΤΗΡΕΣ**

Οι αισθητήρες είναι συσκευές που μετατρέπουν φυσικές ή χημικές μεταβολές στο περιβάλλον τους σε μια μορφή που μπορεί να μετρηθεί ή να ανιχνευθεί. Αυτή η μετατροπή γίνεται συνήθως σε ηλεκτρικό σήμα, το οποίο στη συνέχεια μπορεί να επεξεργαστεί ή να χρησιμοποιηθεί από άλλα συστήματα.

Οι αισθητήρες χρησιμοποιούνται σε πολλές εφαρμογές, όπως στην τεχνολογία των αυτοκινήτων (π.χ. αισθητήρες πίεσης ελαστικών), την ιατρική (π.χ. αισθητήρες καρδιακής σφαλμάτωσης), την βιομηχανία (π.χ. αισθητήρες θερμοκρασίας σε εργοστάσια), την περιβαλλοντική παρακολούθηση (π.χ. αισθητήρες ρύπανσης) και πολλές άλλες[22].

#### **2.1.3.2 ΕΝΕΡΓΟΠΟΙΗΤΕΣ**

Οι ενεργοποιητές είναι συσκευές ή μηχανισμοί που προκαλούν μια αντίδραση ή μεταβολή σε απάντηση σε ένα εισερχόμενο σήμα ή εντολή[27]. Αναλόγως της εφαρμογής, αυτή η αντίδραση

μπορεί να είναι φυσική (όπως μια κίνηση, η εκπομπή φωτός ή ήχου) ή μπορεί να προκαλέσει μια ηλεκτρονική αλλαγή (όπως μια αλλαγή σε ένα ηλεκτρικό σήμα).

Οι ενεργοποιητές χρησιμοποιούνται σε πολλές εφαρμογές. Για παράδειγμα, στον τομέα της αυτοματοποίησης και των ρομποτικών, ενεργοποιητές μπορεί να είναι κινητήρες που κινούν μηχανικά μέρη, όπως ένας βραχίονας ρομπότ, με βάση εντολές που λαμβάνουν από έναν ελεγκτή[28]. Σε έναν εξοπλισμό ή εγκατάσταση ασφαλείας, ενεργοποιητές μπορεί να είναι αισθητήρες που ανιχνεύουν κίνδυνο και ενεργοποιούν συναγερμό ή αντίδραση ασφαλείας.

## 2.2 ΔΙΑΔΙΚΤΥΟ

Το διαδίκτυο είναι ένα παγκόσμιο σύστημα διασύνδεσης δικτύων υπολογιστών που χρησιμοποιούν την οικογένεια πρωτοκόλλων TCP/IP (Transmission Control Protocol/Internet Protocol)[36] για να εξυπηρετήσουν δισεκατομμύρια χρήστες παγκοσμίως. Είναι ένα δίκτυο δικτύων που αποτελείται από εκατομμύρια ιδιωτικά, δημόσια, ακαδημαϊκά, επιχειρηματικά και κυβερνητικά δίκτυα, τα οποία συνδέονται μεταξύ τους μέσω καλωδίων οπτικών ινών, δορυφόρων, καλωδίων χαλκού, ασύρματων συνδέσεων και άλλων τεχνολογιών.

Το διαδίκτυο ξεκίνησε ως ένα ερευνητικό έργο του Υπουργείου Άμυνας των ΗΠΑ τη δεκαετία του 1960[35][38], γνωστό ως ARPANET (Advanced Research Projects Agency Network). Αρχικά σχεδιάστηκε για να συνδέσει πανεπιστήμια και ερευνητικά κέντρα, επιτρέποντας στους ερευνητές να μοιράζονται πόρους και πληροφορίες. Με την πάροδο του χρόνου, το ARPANET εξελίχθηκε και διαχωρίστηκε σε διάφορα δίκτυα που τελικά συνενώθηκαν για να σχηματίσουν το διαδίκτυο όπως το γνωρίζουμε σήμερα[37][39].

Το διαδίκτυο βασίζεται σε μια κατανεμημένη αρχιτεκτονική, χωρίς κεντρικό έλεγχο. Τα δίκτυα που το αποτελούν συνδέονται μεταξύ τους μέσω πυλών (gateways) και δρομολογητών (routers)[42], οι οποίοι διασφαλίζουν ότι τα δεδομένα φτάνουν στον προορισμό τους ακολουθώντας την πιο αποδοτική διαδρομή. Η βασική τεχνολογία πίσω από το διαδίκτυο είναι το πρωτόκολλο TCP/IP[36], το οποίο καθορίζει πώς τα δεδομένα χωρίζονται σε πακέτα, πώς δρομολογούνται μέσω του δικτύου και πώς επανασυναρμολογούνται στον προορισμό τους.

Το διαδίκτυο υποστηρίζει μια πληθώρα υπηρεσιών και εφαρμογών που έχουν αλλάξει τον τρόπο που επικοινωνούμε, εργαζόμαστε, μαθαίνουμε και ψυχαγωγούμαστε. Μερικές από τις πιο γνωστές υπηρεσίες περιλαμβάνουν:

- **World Wide Web (WWW):** Ένα σύστημα διασυνδεδεμένων εγγράφων και άλλων πόρων, προσβάσιμων μέσω του πρωτοκόλλου HTTP (Hypertext Transfer Protocol)[40].



- **Ηλεκτρονικό ταχυδρομείο (Email):** Μια από τις πρώτες και πιο διαδεδομένες υπηρεσίες του διαδικτύου, που επιτρέπει την αποστολή και λήψη μηνυμάτων μέσω δικτύων υπολογιστών[41].
- **Κοινωνικά δίκτυα:** Πλατφόρμες όπως το Facebook, Twitter, και Instagram που επιτρέπουν στους χρήστες να δημιουργούν και να μοιράζονται περιεχόμενο και να επικοινωνούν με άλλους χρήστες.
- **Ψηφιακή ψυχαγωγία:** Υπηρεσίες streaming, όπως το Netflix και το Spotify, που παρέχουν πρόσβαση σε βίντεο, μουσική και άλλα πολυμέσα.
- **Ηλεκτρονικό εμπόριο (e-commerce):** Πλατφόρμες όπως το Amazon και το eBay που επιτρέπουν την αγορά και πώληση προϊόντων και υπηρεσιών μέσω του διαδικτύου.

Η ασφάλεια είναι ένα από τα μεγαλύτερα ζητήματα που αντιμετωπίζει το διαδίκτυο. Οι απειλές περιλαμβάνουν την παραβίαση δεδομένων, τις επιθέσεις DDoS (Distributed Denial of Service)[43], το κακόβουλο λογισμικό και την ανεπιθύμητη αλληλογραφία (spam). Οι οργανισμοί και οι χρήστες πρέπει να εφαρμόζουν διάφορα μέτρα ασφαλείας, όπως κρυπτογράφηση, firewall, και λογισμικό προστασίας από ιούς για να προστατεύσουν τα δεδομένα τους και την ιδιωτικότητά τους.

Το διαδίκτυο έχει προκαλέσει σημαντικές αλλαγές σε πολλούς τομείς της κοινωνίας και της οικονομίας. Έχει διευκολύνει την παγκοσμιοποίηση, επιτρέποντας σε ανθρώπους και επιχειρήσεις από διαφορετικά μέρη του κόσμου να επικοινωνούν και να συνεργάζονται πιο εύκολα. Έχει επίσης αλλάξει τον τρόπο που οι άνθρωποι μαθαίνουν και αποκτούν πληροφορίες, με την πρόσβαση σε μια τεράστια ποικιλία πόρων και εκπαιδευτικών εργαλείων.

Το διαδίκτυο συνεχίζει να εξελίσσεται, με νέες τεχνολογίες και τάσεις να αναδύονται συνεχώς. Το Διαδίκτυο των Αντικειμένων (IoT) αναμένεται να συνδέσει δεκάτομμύρια συσκευές, από οικιακές συσκευές μέχρι βιομηχανικό εξοπλισμό, δημιουργώντας έξυπνα περιβάλλοντα που μπορούν να βελτιώσουν την αποδοτικότητα και την ποιότητα ζωής. Η τεχνολογία 5G υπόσχεται ταχύτερες και πιο αξιόπιστες συνδέσεις, επιτρέποντας νέες εφαρμογές και υπηρεσίες που απαιτούν υψηλές ταχύτητες δεδομένων και χαμηλή καθυστέρηση.

Συνολικά, το διαδίκτυο είναι ένας από τους πιο σημαντικούς και επιδραστικούς τεχνολογικούς καινοτομίες της σύγχρονης εποχής, συνεχίζοντας να επηρεάζει και να διαμορφώνει τον κόσμο με τρόπους που δεν μπορούσαν να προβλεφθούν πριν από λίγες δεκαετίες.

### 2.2.1 ΠΡΩΤΟΚΟΛΛΟ ΔΙΑΔΙΚΤΥΟΥ / INTERNET PROTOCOL (IP)

Το Πρωτόκολλο Διαδικτύου (IP) παίζει κρίσιμο ρόλο στην επικοινωνία μεταξύ δικτύων, διευκολύνοντας τη μεταφορά δεδομένων μεταξύ συσκευών. Είναι υπεύθυνο για τη διευθυνσιοδότηση αυτών των συσκευών και την ενσωμάτωση δεδομένων σε πακέτα για παράδοση[44]. Το IP λειτουργεί στο επίπεδο δικτύου, διευκολύνοντας τον διασυνδετισμό και δημιουργώντας τη βάση του Διαδικτύου.



Το IP δρομολογεί πακέτα από τη συσκευή πηγής στη συσκευή προορισμού, χρησιμοποιώντας τις διευθύνσεις IP που περιλαμβάνονται στις επικεφαλίδες των πακέτων. Κάθε πακέτο περιλαμβάνει μια επικεφαλίδα που περιέχει τις διευθύνσεις IP πηγής και προορισμού, μαζί με μεταδεδομένα για το δρομολόγηση, και ένα φορτίο που μεταφέρει τα πραγματικά δεδομένα.

Το IPv4, η κυρίαρχη έκδοση του IP, διαδέχτηκε το IPv6 λόγω της αυξανόμενης ζήτησης για διευθύνσεις IP. Το IPv4 χρησιμοποιεί διευθύνσεις 32 bit, ενώ το IPv6 χρησιμοποιεί διευθύνσεις 128 bit, προσφέροντας έναν αστρονομικά μεγαλύτερο χώρο διευθύνσεων[45][46].

Η σχεδίαση του IP ακολουθεί την αρχή της άκρο προς άκρο, θεωρώντας την υποδομή του δικτύου ως αναπόδραστα αναξιόπιστη σε οποιοδήποτε μοναδικό στοιχείο του δικτύου ή μέσου μετάδοσης. Ως αποτέλεσμα, το πρωτόκολλο IP παρέχει παράδοση με την καλύτερη δυνατή προσπάθεια και χαρακτηρίζεται ως μη αξιόπιστο. Υψηλότερα πρωτόκολλα εντός της ποικιλίας πρωτοκόλλων του Διαδικτύου, όπως το TCP και το UDP, αντιμετωπίζουν θέματα αξιοπιστίας και ανάλυσης σφαλμάτων[47].

Η ασφάλεια είναι μια σημαντική πτυχή στην εξέλιξη του IP καθώς το Διαδίκτυο έχει αυξηθεί. Αρχικά, τα πρωτόκολλα IP έλειπαν από αρκετά σημεία ασφαλείας, οδηγώντας σε ευπάθειες και επιθέσεις δικτύου. Έχουν γίνει σημαντικές προσπάθειες για την ενίσχυση της ασφαλείας, συμπεριλαμβανομένης της χρήσης τεχνολογιών όπως το IPsec για την κρυπτογράφηση και την προστασία των δεδομένων που μεταδίδονται μέσω των δικτύων IP.

### **2.2.2 CLIENT-SERVER**

Το μοντέλο client-server είναι μια δομή κατανεμημένων εφαρμογών που μοιράζει τις εργασίες μεταξύ των παρόχων πόρων ή υπηρεσιών, που ονομάζονται διακομιστές, και των αιτούντων υπηρεσίας, που ονομάζονται πελάτες[48]. Συνήθως, οι πελάτες και οι διακομιστές επικοινωνούν μέσω ενός υπολογιστικού δικτύου σε ξεχωριστό υλικό, αλλά και ο πελάτης και ο διακομιστής μπορεί να βρίσκονται στον ίδιο υπολογιστή. Ένας διακομιστής τρέχει ένα ή περισσότερα προγράμματα διακομιστή, τα οποία μοιράζονται τους πόρους τους με τους πελάτες. Ένας πελάτης συνήθως δε μοιράζεται κανέναν από τους πόρους του, αλλά ζητά περιεχόμενο ή υπηρεσία από έναν διακομιστή.

Η επικοινωνία πελάτη και διακομιστή συνήθως συμβαίνει με τη μορφή επικοινωνίας αιτήματος-απάντησης[49]. Ο πελάτης στέλνει ένα αίτημα και ο διακομιστής επιστρέφει μια απάντηση. Αυτή η ανταλλαγή μηνυμάτων είναι ένα παράδειγμα διαδικασίας επικοινωνίας μεταξύ διεργασιών.

Συνήθως, ο διακομιστής αναμένει εισερχόμενα αιτήματα από πολλούς διαφορετικούς πελάτες και πρέπει να χειρίζεται αυτά τα αιτήματα με κάποιον τρόπο για να τα εξυπηρετήσει αποτελεσματικά.

Ωστόσο, οι διάφοροι πελάτες μπορεί να επηρεάζουν τη δυναμική του διακομιστή. Για παράδειγμα, οι επιθέσεις αρνητικής υπηρεσίας είναι σχεδιασμένες για να εκμεταλλευτούν την υποχρέωση ενός διακομιστή να επεξεργάζεται αιτήματα φόρτου με υπερβολικές ταχύτητες[43].

Το παράδειγμα με την τράπεζα είναι ένα κλασικό παράδειγμα client-server. Όταν ένας πελάτης συνδέεται στις υπηρεσίες online τραπεζικών συναλλαγών με έναν web browser (τον πελάτη), αυτός ο πελάτης ξεκινά ένα αίτημα στον διακομιστή του τραπεζικού ιστότοπου. Οι κωδικοί πιστοποίησης σύνδεσης του πελάτη ενδέχεται να αποθηκεύονται σε μια βάση δεδομένων, και ο διακομιστής web προσπελαίνει το διακομιστή βάσης δεδομένων ως πελάτης. Ένας διακομιστής ερμηνεύει τα επιστρεφόμενα δεδομένα εφαρμόζοντας την επιχειρηματική λογική της τράπεζας και παρέχει το αποτέλεσμα στον διακομιστή web. Τέλος, ο διακομιστής web επιστρέφει τα αποτελέσματα στον πελάτη για εμφάνιση.

Αυτό το παράδειγμα δείχνει ένα πλάνο σχεδίασης που εφαρμόζεται στο μοντέλο client-server: απομόνωση των ανησυχιών. Από την αρχή του 1960 και μετά, οι όροι διακομιστής και πελάτης έχουν εξελιχθεί για να περιγράψουν τις σχέσεις μεταξύ διαφόρων υπολογιστικών συστημάτων[50]. Ο διακομιστής προσφέρει υπηρεσίες σε πολλούς πελάτες, ενώ ο πελάτης ζητά υπηρεσίες από τον διακομιστή. Αυτή η ανταλλαγή πληροφοριών είναι ένα βασικό στοιχείο της σύγχρονης υπολογιστικής αρχιτεκτονικής.

### 2.2.3 WEBSOCKETS

WebSocket είναι ένα πρωτόκολλο επικοινωνίας στον υπολογιστή που παρέχει ταυτόχρονα διπλής κατεύθυνσης κανάλια επικοινωνίας μέσω μίας μόνο σύνδεσης Transmission Control Protocol (TCP)[36]. Αυτό το πρωτόκολλο καθιερώθηκε από το IETF ως RFC 6455 το 2011 και επιτρέπει σε web εφαρμογές να χρησιμοποιούν αυτό το πρωτόκολλο για πραγματικού χρόνου μεταφορά δεδομένων από και προς τον διακομιστή[51].

Είναι διαφορετικό από το πρωτόκολλο HTTP που χρησιμοποιείται για την πλειοψηφία των ιστοσελίδων. Και τα δύο πρωτόκολλα βρίσκονται στο επίπεδο 7 στο μοντέλο OSI και εξαρτώνται από το TCP στο επίπεδο 4. Παρόλο που είναι διαφορετικά, το RFC 6455 αναφέρει ότι το WebSocket "σχεδιάστηκε για να λειτουργεί σε θύρες HTTP 443 και 80 καθώς και να υποστηρίζει διακομιστές και ενδιάμεσους HTTP", κάνοντάς το συμβατό με το HTTP. Η σύνδεση WebSocket χρησιμοποιεί την κεφαλίδα HTTP Upgrade για να μεταβεί από το πρωτόκολλο HTTP στο πρωτόκολλο WebSocket[52].

Το πρωτόκολλο WebSocket επιτρέπει την πλήρη αμφίδρομη αλληλεπίδραση μεταξύ ενός προγράμματος περιήγησης ιστού (ή άλλης εφαρμογής πελάτη) και ενός διακομιστή ιστού με μικρότερο υπερκεφάλαιο σε σύγκριση με εναλλακτικές λύσεις ημι-διπλής αμφίδρομης επικοινωνίας

όπως η ανάκτηση HTTP, διευκολύνοντας τη μεταφορά πραγματικού χρόνου δεδομένων από και προς τον διακομιστή.

Το πρωτόκολλο WebSocket έχει ενσωματωθεί στις περισσότερες περιηγητικές εφαρμογές, συμπεριλαμβανομένων του Google Chrome, του Firefox, του Microsoft Edge, του Internet Explorer, του Safari και του Opera.

Η προτύπωση του πρωτοκόλλου WebSocket καθορίζει το ws (WebSocket) και το wss (WebSocket Secure) ως δύο νέες ομοιόμορφες αναγνωριστικές ενότητες πόρων (URI) που χρησιμοποιούνται για μη κρυπτογραφημένες και κρυπτογραφημένες συνδέσεις αντίστοιχα.

Το πρωτόκολλο WebSocket ξεκινά με μια αίτηση/απάντηση HTTP, επιτρέποντας στους διακομιστές να διαχειριστούν συνδέσεις HTTP και συνδέσεις WebSocket στην ίδια θύρα. Αφού η σύνδεση ιδρυθεί, η επικοινωνία μεταβαίνει σε ένα αμφίδρομο δυαδικό πρωτόκολλο που δεν συμμορφώνεται με το πρωτόκολλο HTTP[53].

#### 2.2.4 ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

Το Διαδίκτυο των Αντικειμένων (IoT) αναφέρεται σε ένα εκτεταμένο δίκτυο φυσικών αντικειμένων που είναι εφοδιασμένα με αισθητήρες, λογισμικό και άλλες τεχνολογίες με στόχο τη σύνδεση και την ανταλλαγή δεδομένων με άλλες συσκευές και συστήματα μέσω του Διαδικτύου. Οι συσκευές αυτές, που περιλαμβάνουν από οικιακές συσκευές μέχρι βιομηχανικά εργαλεία, είναι σχεδιασμένες να συλλέγουν και να μοιράζονται δεδομένα, προσφέροντας νέες δυνατότητες αλληλεπίδρασης και αυτοματοποίησης. Η βασική ιδέα πίσω από το IoT είναι η δημιουργία ενός «έξυπνου» περιβάλλοντος όπου οι συσκευές μπορούν να επικοινωνούν και να συνεργάζονται αυτόνομα[54]. Αυτές οι συσκευές είναι ενσωματωμένες με τεχνολογίες που επιτρέπουν την παρακολούθηση, τον έλεγχο και την ανταπόκριση σε δεδομένα σε πραγματικό χρόνο. Οι εφαρμογές του IoT είναι απεριόριστες και καλύπτουν ένα ευρύ φάσμα τομέων, από την υγεία και την εκπαίδευση μέχρι τη γεωργία και τις μεταφορές.

Η λειτουργία του IoT βασίζεται στη σύνδεση των συσκευών με μια κεντρική πλατφόρμα ή κόμβο που διαχειρίζεται την επεξεργασία και την αποθήκευση των δεδομένων. Αυτή η πλατφόρμα επιτρέπει στις συσκευές να ανταλλάσσουν πληροφορίες και να ενεργούν με βάση προκαθορισμένους αλγόριθμους και κανόνες. Το IoT συχνά χρησιμοποιεί πρωτόκολλα επικοινωνίας όπως το MQTT, το CoAP και το HTTP για τη διασφάλιση της αποτελεσματικής μετάδοσης δεδομένων.

Ένα από τα κύρια χαρακτηριστικά του IoT είναι η αυτονομία και η αυτοματοποίηση. Οι συσκευές IoT μπορούν να λειτουργούν ανεξάρτητα, λαμβάνοντας αποφάσεις και εκτελώντας εργασίες χωρίς ανθρώπινη παρέμβαση. Για παράδειγμα, ένας έξυπνος θερμοστάτης μπορεί να

ρυθμίζει τη θερμοκρασία ενός δωματίου αυτόματα, με βάση τις προτιμήσεις του χρήστη και τα δεδομένα που συλλέγει από το περιβάλλον.

Παρά τα πολλά οφέλη, το IoT αντιμετωπίζει αρκετές προκλήσεις. Η ασφάλεια είναι μία από τις μεγαλύτερες ανησυχίες, καθώς οι συσκευές IoT συλλέγουν και μεταδίδουν ευαίσθητα δεδομένα που πρέπει να προστατεύονται από ανεπιθύμητη πρόσβαση. Η κρυπτογράφηση των δεδομένων και η ασφαλής διαχείριση των συσκευών είναι απαραίτητες για την αποτροπή επιθέσεων και την προστασία της ιδιωτικότητας[55].

Επιπλέον, η διαχείριση του τεράστιου όγκου δεδομένων που παράγονται από τις συσκευές IoT αποτελεί μια σημαντική πρόκληση. Οι πλατφόρμες IoT πρέπει να διαθέτουν ισχυρές υποδομές για την αποθήκευση, την επεξεργασία και την ανάλυση αυτών των δεδομένων[56]. Οι τεχνολογίες Big Data και η μηχανική μάθηση χρησιμοποιούνται συχνά για την εξαγωγή χρήσιμων πληροφοριών από τα δεδομένα του IoT.

Η χρησιμότητα του IoT είναι ευρεία και καλύπτει πολλές πτυχές της καθημερινής ζωής και των επιχειρηματικών διαδικασιών. Στην καθημερινή ζωή, το IoT μπορεί να περιλαμβάνει έξυπνες οικιακές συσκευές που ενημερώνουν για ελλείψεις τροφίμων ή παρακολουθούν την κατανάλωση ενέργειας. Στον τομέα της υγείας, οι φορητές συσκευές IoT μπορούν να παρακολουθούν ζωτικές ενδείξεις και να ενημερώνουν τους ιατρούς σε πραγματικό χρόνο. Στη γεωργία, τα συστήματα IoT μπορούν να παρακολουθούν τις συνθήκες του εδάφους και να βελτιστοποιούν την άρδευση[57]. Το IoT αναμένεται να επιφέρει σημαντική καινοτομία και να προωθήσει την ανάπτυξη νέων επιχειρηματικών μοντέλων και οικονομική ανάπτυξη. Οι επιχειρήσεις σχεδιάζουν να υιοθετήσουν συστήματα IoT για να βελτιώσουν τις λειτουργίες τους και να προσφέρουν νέες υπηρεσίες στους πελάτες τους. Παράλληλα, το IoT αναμένεται να συμβάλει στην αντιμετώπιση κοινωνικών και περιβαλλοντικών προκλήσεων.

## 2.3 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

Το άρθρο των Gubbi et al. (2013)[59], "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," αποτελεί μια από τις πιο επιδραστικές και αναλυτικές ανασκοπήσεις στον τομέα του IoT. Οι συγγραφείς προτείνουν ένα σαφές όραμα για το IoT, περιγράφοντας μια μελλοντική εποχή όπου οι καθημερινές συσκευές είναι διασυνδεδεμένες και αλληλεπιδρούν μέσω του Διαδικτύου. Οι Gubbi et al. αναλύουν τις βασικές αρχιτεκτονικές στοιχείων του IoT, περιλαμβάνοντας αισθητήρες, συνδεσιμότητα, υπηρεσίες και τις τεχνολογίες επεξεργασίας δεδομένων.

Σημαντική έμφαση δίνεται στις προκλήσεις της κλιμάκωσης και της διαλειτουργικότητας, καθώς το IoT περιλαμβάνει έναν τεράστιο αριθμό ετερογενών συσκευών που πρέπει να επικοινωνούν

απρόσκοπτα. Το άρθρο αναλύει επίσης τη σημασία της υποδομής υπολογιστικού νέφους για την αποθήκευση και ανάλυση των δεδομένων, επισημαίνοντας ότι η συλλογή και επεξεργασία μεγάλου όγκου δεδομένων σε πραγματικό χρόνο αποτελεί κεντρικό ζήτημα.

Τέλος, οι συγγραφείς εξετάζουν τις μελλοντικές κατευθύνσεις του IoT, περιλαμβάνοντας την ασφάλεια και την προστασία της ιδιωτικότητας, την ανάγκη για πρότυπα και την ανάπτυξη ευφών εφαρμογών που θα βελτιώσουν την καθημερινότητα των ανθρώπων. Το άρθρο κλείνει με την επισήμανση ότι η επιτυχής ανάπτυξη του IoT απαιτεί συνεργασία μεταξύ της ακαδημαϊκής κοινότητας, της βιομηχανίας και των κυβερνήσεων.

Το άρθρο των Al-Fuqaha et al. (2015)[60], "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," παρέχει μια εκτενή ανασκόπηση των βασικών τεχνολογιών που επιτρέπουν τη λειτουργία του IoT, συμπεριλαμβανομένων των πρωτοκόλλων επικοινωνίας και των δικτύων αισθητήρων. Οι συγγραφείς αναλύουν τις κύριες τεχνολογίες όπως τα ασύρματα δίκτυα αισθητήρων (WSN), τα πρωτόκολλα επικοινωνίας μικρής εμβέλειας όπως το Bluetooth και το Zigbee, καθώς και τα πρωτόκολλα μεγάλων αποστάσεων όπως το LTE και το Wi-Fi. Επιπλέον, γίνεται εκτενής αναφορά στις τεχνολογίες υπολογιστικού νέφους (cloud computing) και ανάλυσης μεγάλων δεδομένων (big data analytics), που είναι απαραίτητες για την αποθήκευση και ανάλυση των δεδομένων που συλλέγονται από συσκευές IoT.

Οι Al-Fuqaha et al. εστιάζουν επίσης στις εφαρμογές του IoT σε διάφορους τομείς, όπως η υγειονομική περίθαλψη, οι έξυπνες πόλεις, τα έξυπνα σπίτια, και οι έξυπνες μεταφορές. Τονίζεται ότι το IoT έχει τη δυνατότητα να επαναπροσδιορίσει τη λειτουργία αυτών των τομέων μέσω της βελτίωσης της αποτελεσματικότητας, της μείωσης του κόστους και της παροχής νέων υπηρεσιών. Παράλληλα, το άρθρο επισημαίνει τις προκλήσεις που παραμένουν, όπως η ασφάλεια των δεδομένων, η ιδιωτικότητα και η ανάγκη για διαλειτουργικότητα μεταξύ των συσκευών και των συστημάτων. Συμπερασματικά, οι συγγραφείς αναγνωρίζουν την ανάγκη για συνεχή έρευνα και ανάπτυξη προτύπων που θα υποστηρίξουν τη μαζική υιοθέτηση του IoT σε παγκόσμιο επίπεδο.

Το άρθρο των Bandyopadhyay και Sen (2011)[61], "Internet of Things: Applications and Challenges in Technology and Standardization," προσφέρει μια επισκόπηση των τεχνολογικών και τυποποιητικών προκλήσεων που αντιμετωπίζει το IoT. Οι συγγραφείς διερευνούν την αρχιτεκτονική του IoT, επισημαίνοντας την ανάγκη για ενσωμάτωση ετερογενών συστημάτων και πλατφορμών για την επίτευξη αποτελεσματικής διαχείρισης δεδομένων και διαλειτουργικότητας.

Επιπλέον, το άρθρο εστιάζει σε συγκεκριμένες εφαρμογές του IoT, όπως τα έξυπνα σπίτια, οι έξυπνες πόλεις, και τα συστήματα υγειονομικής περίθαλψης, αναδεικνύοντας τα οφέλη της βελτιστοποίησης πόρων και της βελτίωσης της ποιότητας ζωής. Ωστόσο, αναγνωρίζονται σημαντικές προκλήσεις, όπως η ασφάλεια, η ιδιωτικότητα και η ανάγκη για ανάπτυξη διεθνών προτύπων που θα διευκολύνουν τη διαλειτουργικότητα και την επικοινωνία μεταξύ των διαφόρων συσκευών. Οι συγγραφείς τονίζουν την ανάγκη για συνεχή έρευνα και συνεργασία μεταξύ των ακαδημαϊκών, των

βιομηχανιών και των ρυθμιστικών φορέων για την επίτευξη ενός πλήρως λειτουργικού και ασφαλούς περιβάλλοντος IoT.

Η διπλωματική εργασία με τίτλο «*Έξυπνες εφαρμογές γεωργίας ακριβείας στο πλαίσιο του Διαδικτύου των Αντικειμένων*» από τον Αναστάσιο Λαζάρου (2024)[62] επικεντρώνεται στην ενσωμάτωση των τεχνολογιών IoT στη γεωργία για τη βελτίωση των καλλιεργητικών μεθόδων και την αντιμετώπιση των προκλήσεων της κλιματικής αλλαγής. Το έργο περιγράφει τη δημιουργία ενός έξυπνου συστήματος που επιτρέπει την παρακολούθηση και την πρόβλεψη της κατάστασης των καλλιεργειών, παρέχοντας χρήσιμες πληροφορίες στους αγρότες μέσω αισθητήρων και αλγορίθμων μηχανικής μάθησης. Το σύστημα στοχεύει στη βελτιστοποίηση της χρήσης πόρων, όπως το νερό, και στη μείωση της σπατάλης.

Το σύστημα αυτό, το οποίο αποτελεί το ελάχιστο βιώσιμο προϊόν, ενσωματώνει τεχνολογίες όπως Arduino, Sensors, C, Python3, και πλατφόρμες όπως το ThingsBoard, για τη συλλογή, ανάλυση και παρουσίαση δεδομένων. Η έρευνα περιλαμβάνει επίσης την ανάλυση των υφιστάμενων γεωργικών μεθόδων και των επιπτώσεων της κλιματικής αλλαγής στη γεωργία, αναδεικνύοντας τη σημασία της τεχνολογίας στη βελτίωση της παραγωγικότητας και της βιωσιμότητας στη γεωργία.

## **ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΑ**

Στην παρούσα ενότητα, θα περιγραφεί η μεθοδολογία που χρησιμοποιήθηκε για τον σχεδιασμό και την υλοποίηση της εφαρμογής. Συγχρόνως, θα εξεταστούν οι στόχοι, τα προβλήματα και τα εμπόδια που προέκυψαν κατά τη διάρκεια της διαδικασίας ανάπτυξής της.

### **3.1 ΣΤΟΧΟΣ ΚΑΙ ΤΑ ΥΠΟΠΡΟΒΛΗΜΑΤΑ**

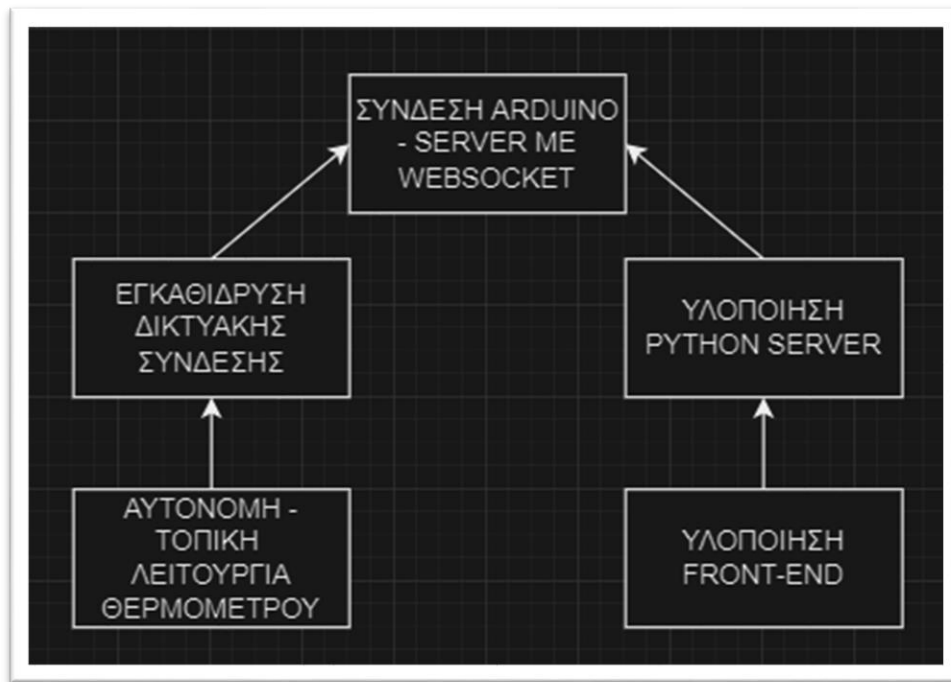
Βασικός στόχος της εργασίας είναι η δημιουργία ενός συστήματος ανίχνευσης θερμοκρασίας, υγρασίας και ισχύος φωτός. Οι πληροφορίες αυτές στέλνονται σε έναν διακομιστή όπου περνούν από μια λίστα αποφάσεων επιτυγχάνοντας ένα έξυπνο σπίτι. Το σύστημα συλλογής δεδομένων αποτελείται από Arduino σε συνδυασμό με έναν ESP και διάφορους αισθητήρες και ενεργοποιητές, καθώς και σύγχρονες τεχνικές για την επικοινωνία μεταξύ Arduino και server. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού C++.

Η ανάπτυξη του συστήματος περιλαμβάνει αρκετά στάδια και απαιτεί την επίλυση διαφόρων υποπροβλημάτων που εμφανίζονται κατά τη διαδικασία. Τα υποπροβλήματα που παρατηρήθηκαν και χρειάστηκε να αντιμετωπιστούν με ιδιαίτερη προσοχή είναι τα εξής:

1. Η επικοινωνία του Arduino με το διαδίκτυο
2. Η επικοινωνία του Arduino με το διακομιστή
3. Καλός συγχρονισμός στη μεταφορά μηνυμάτων/δεδομένων από και προς τον διακομιστή και το Arduino

### **3.2 ΠΡΩΙΜΑ ΣΤΑΔΙΑ**

Για την υλοποίηση μίας εφαρμογής μεγάλης κλίμακας όπως η δική μας, πρώτα πρέπει να διασπάσουμε το έργο σε υποπροβλήματα. Η υλοποίηση μίας εφαρμογής αποτελείται από διάφορα στάδια που απαιτούνται για την ολοκλήρωση του έργου.



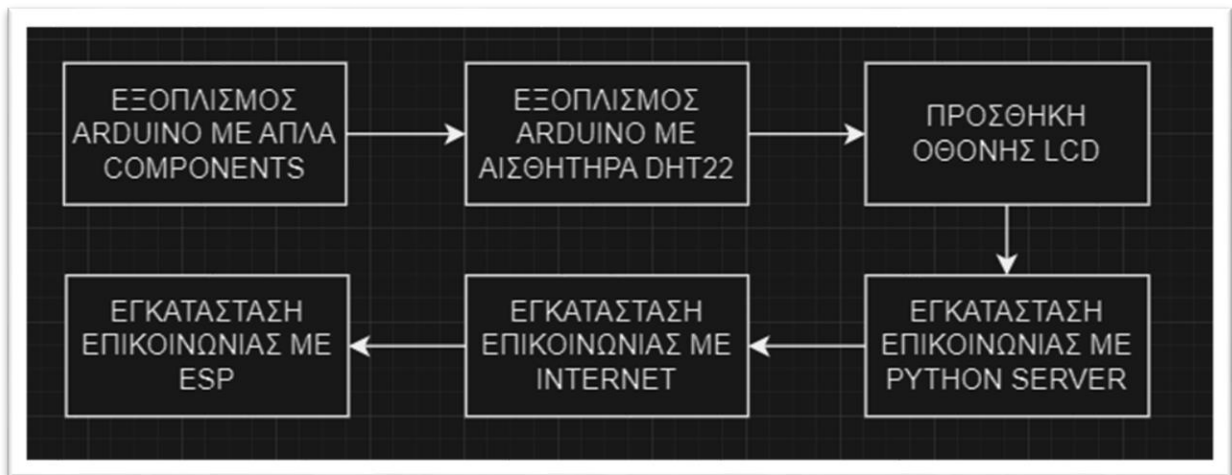
**Σχήμα 3.1 :** Στάδια υλοποίησης εφαρμογής [9]

Στο αρχικό στάδιο, δημιουργήθηκε ένα αυτόνομο θερμόμετρο που λειτουργεί τοπικά, χωρίς καμία δικτυακή σύνδεση. Αυτό το στάδιο περιλάμβανε τη χρήση του Arduino για τη λήψη και εμφάνιση των μετρήσεων θερμοκρασίας. Παράλληλα με την ανάπτυξη του αυτόνομου θερμόμετρου, πραγματοποιήθηκε η υλοποίηση του front-end, το οποίο θα επιτρέπει τη διαδραστική επικοινωνία των χρηστών με την εφαρμογή μέσω ενός φιλικού προς τον χρήστη γραφικού περιβάλλοντος.

Στη συνέχεια, υλοποιήθηκε η δικτυακή σύνδεση, επιτρέποντας στη συσκευή να συνδεθεί στο διαδίκτυο. Αυτό το στάδιο περιλάμβανε την ενσωμάτωση του ESP για τη δικτύωση της συσκευής. Παράλληλα με την εγκαθίδρυση της δικτυακής σύνδεσης, υλοποιήθηκε ο διακομιστής σε Python, ο οποίος θα λαμβάνει και θα επεξεργάζεται τα δεδομένα από το Arduino.

Το τελικό στάδιο περιλάμβανε τη σύνδεση του Arduino με τον διακομιστή μέσω WebSocket, εξασφαλίζοντας έτσι τη συνεχή και άμεση επικοινωνία μεταξύ της συσκευής και του server.





Σχήμα 3.2 : Σταδιακή υλοποίησης εφαρμογής [9]

Η ανάλυση της πρώτης στήλης περιγράφει λεπτομερώς τα στάδια εξοπλισμού και εγκατάστασης του Arduino και στην κορυφή είναι ο συνδετικός κρίκος που ενώνει τον ESP με τον διακομιστή. Το πρώτο βήμα ήταν να εξοπλιστεί το Arduino με τα απαραίτητα εξαρτήματα για τη βασική του λειτουργία. Αυτό περιλάμβανε την προσθήκη στοιχείων όπως αντιστάσεις, καλώδια και άλλους βασικούς αισθητήρες. Στη συνέχεια, προστέθηκε ο αισθητήρας θερμοκρασίας και υγρασίας DHT22, επιτρέποντας στο Arduino να λαμβάνει περιβαλλοντικές μετρήσεις.

Προστέθηκε μια οθόνη LCD για την άμεση εμφάνιση των μετρήσεων. Αυτό το βήμα ήταν κρίσιμο για τον εύκολο έλεγχο και τη διάγνωση των λειτουργιών της συσκευής. Ενσωματώθηκε το ESP, το οποίο επιτρέπει τη δικτύωση του Arduino και τη σύνδεσή του στο διαδίκτυο. Αυτό το βήμα ήταν καθοριστικό για τη δυνατότητα απομακρυσμένης παρακολούθησης και ελέγχου της συσκευής. Εξασφαλίστηκε η σύνδεση του Arduino στο διαδίκτυο, επιτρέποντας τη μεταφορά δεδομένων προς και από τον διακομιστή. Το τελικό στάδιο περιλάμβανε τη σύνδεση του Arduino με τον διακομιστή Python, διασφαλίζοντας τη συνεχή και αξιόπιστη επικοινωνία μεταξύ των δύο συστημάτων.

Μέσω αυτών των σταδίων, επιτεύχθηκε μια ολοκληρωμένη και λειτουργική εφαρμογή, ικανή να συλλέγει, να επεξεργάζεται και να παρουσιάζει δεδομένα σε πραγματικό χρόνο.

### 3.3 ΕΜΠΟΔΙΑ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ

Κατά τη διαδικασία υλοποίησης της εφαρμογής, προέκυψαν διάφορα εμπόδια που έπρεπε να αντιμετωπιστούν. Το μεγαλύτερο εμπόδιο ήταν η σύνδεση του Arduino με τον διακομιστή και η ανάγκη για τον σωστό συγχρονισμό τους. Η διασύνδεση αυτών των δύο στοιχείων απαιτούσε αξιόπιστη και συνεχή επικοινωνία, κάτι που αποδείχθηκε δύσκολο να επιτευχθεί.

Για την αντιμετώπιση αυτού του προβλήματος, επιλέχθηκε η χρήση του πρωτοκόλλου WebSocket. Σε αντίθεση με το παραδοσιακό πρωτόκολλο HTTP, το οποίο βασίζεται σε αιτήσεις και απαντήσεις, το WebSocket προσφέρει διμερή επικοινωνία σε πραγματικό χρόνο. Αυτό σημαίνει ότι τόσο ο διακομιστής όσο και το Arduino μπορούν να στέλνουν και να λαμβάνουν δεδομένα ταυτόχρονα, χωρίς την ανάγκη συνεχών αιτήσεων για ενημερώσεις.

Η επιλογή του WebSocket έγινε κυρίως λόγω της χαμηλής καθυστέρησης και της αποδοτικότητας του σε εφαρμογές πραγματικού χρόνου. Το WebSocket διατηρεί μια μόνιμη σύνδεση μεταξύ του πελάτη και του διακομιστή, επιτρέποντας την άμεση μεταφορά δεδομένων. Αυτό ήταν κρίσιμο για την εφαρμογή μας, καθώς οι μετρήσεις και οι εντολές έπρεπε να μεταφέρονται άμεσα και χωρίς καθυστερήσεις.

Η χρήση του WebSocket μας επέτρεψε να ξεπεράσουμε το πρόβλημα του συγχρονισμού, προσφέροντας μια σταθερή και αξιόπιστη σύνδεση μεταξύ του Arduino και του διακομιστή. Επιπλέον, η απλότητα της ενσωμάτωσης του WebSocket στο υπάρχον σύστημα, σε συνδυασμό με την αποδοτικότητα του, το κατέστησε την ιδανική επιλογή για την υλοποίηση της εφαρμογής μας. Παρόλα αυτά, η εγκατάσταση και η παραμετροποίηση της επικοινωνίας μέσω WebSocket απαιτούσε προσεκτικό σχεδιασμό και δοκιμές, για να εξασφαλιστεί η βέλτιστη λειτουργία του συστήματος.

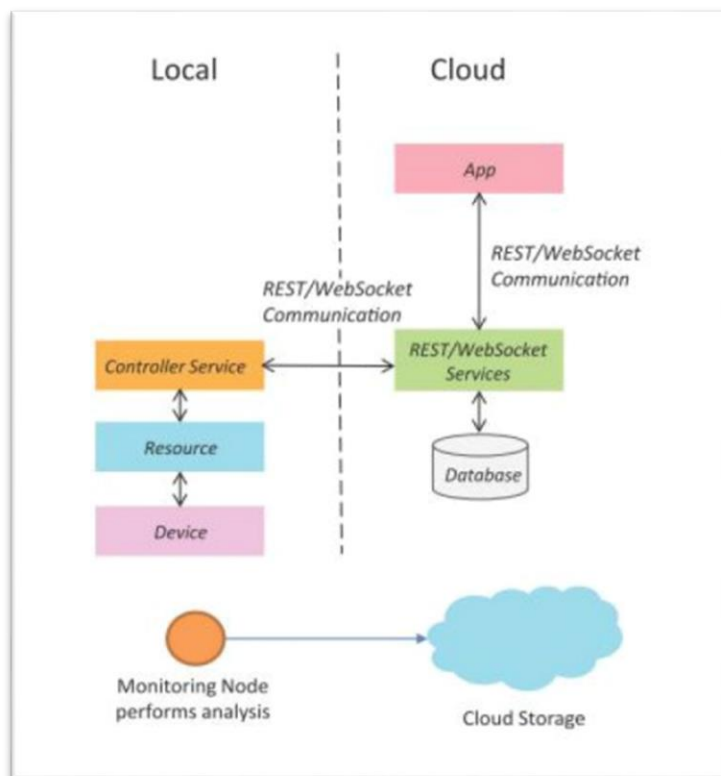
## **ΚΕΦΑΛΑΙΟ 4: ΤΟ ΣΥΣΤΗΜΑ**

Η ανάπτυξη ενός διαδικτυακού συστήματος ή εφαρμογής μπορεί να αποδειχθεί περίπλοκη και μακροσκελή διαδικασία καθώς αποτελείται από πολλές διαφορετικές διαδικασίες η κάθε μια από τις οποίες μπορεί να απαιτεί διαφορετικές τεχνολογίες για την επίτευξή της. Για την διευκόλυνση την

ανάπτυξης, συνηθίζεται η επιλογή τεχνολογιών και βιβλιοθηκών με ευρύ κοινό και υποστήριξη ώστε να υπάρχουν όσο το δυνατόν λιγότερα προβλήματα. Ακολουθούνται, επίσης, τεχνικές και μεθοδολογίες για την βελτίωση της κατανόησης και συγγραφής του κώδικα, όπως μια σωστή δομή των αρχείων της εφαρμογής.

#### 4.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΡΓΑΣΙΑΣ

Το αρχιτεκτονικό μοντέλο της εργασίας μας βασίζεται σε μια διακριτή διάρθρωση που αποτελείται από τοπικά και cloud στοιχεία. Η δομή αυτή διασφαλίζει την αποτελεσματική συλλογή, ανάλυση και απεικόνιση των δεδομένων, καθώς και τη διασύνδεση και επικοινωνία των διαφόρων συσκευών και υπηρεσιών. Αυτή η αρχιτεκτονική επιτρέπει την αποδοτική συλλογή, επεξεργασία και ανάλυση δεδομένων, προσφέροντας μια επεκτάσιμη λύση που συνδυάζει το τοπικό και το cloud περιβάλλον.



Εικόνα 4.1 : Αρχιτεκτονική IoT εφαρμογής [58]

Στο τοπικό επίπεδο, υπάρχουν διάφορα στοιχεία που συνεργάζονται για τη συλλογή και διαχείριση δεδομένων από μια συσκευή. Η συσκευή (Device) αποστέλλει δεδομένα σε πόρους (Resource), οι οποίοι στη συνέχεια επεξεργάζονται από την υπηρεσία ελέγχου (Controller Service). Η

υπηρεσία αυτή αναλαμβάνει να διαχειριστεί τα δεδομένα και να τα προωθήσει στο cloud μέσω επικοινωνίας REST ή WebSocket.

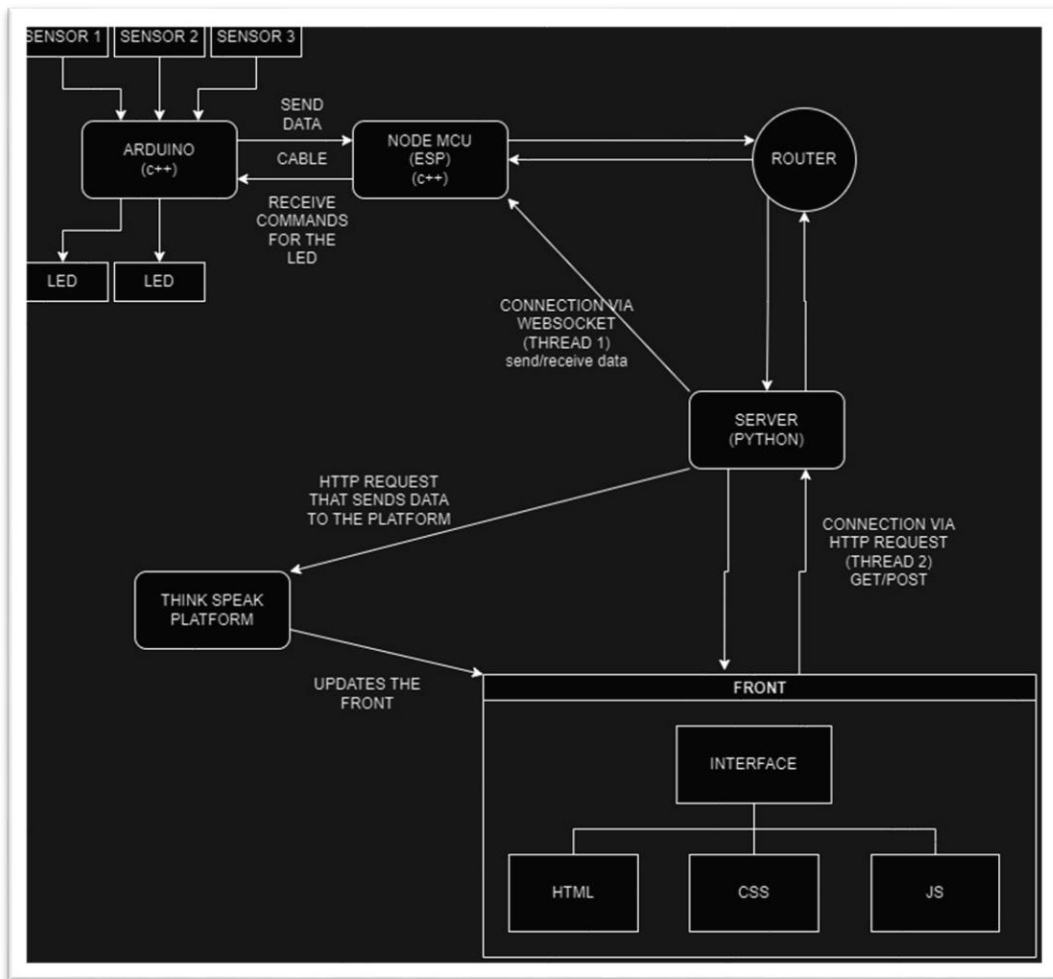
Στο cloud, τα δεδομένα λαμβάνονται και αποθηκεύονται σε μια βάση δεδομένων (Database) μέσω REST/WebSocket υπηρεσιών. Αυτές οι υπηρεσίες επιτρέπουν τη διασύνδεση με εφαρμογές (App) που χρησιμοποιούν τα δεδομένα για διάφορες λειτουργίες. Η επικοινωνία μεταξύ του τοπικού επιπέδου και του cloud πραγματοποιείται μέσω REST/WebSocket πρωτοκόλλων, εξασφαλίζοντας αξιόπιστη και άμεση μετάδοση δεδομένων.

Πιο συγκεκριμένα, η αρχιτεκτονική της εργασίας μας χαρακτηρίζεται από μια διακριτή διάρθρωση σε τρία βασικά μέρη: το Arduino και το ESP για το τοπικό επίπεδο και τον διακομιστή (server) για το cloud επίπεδο. Κάθε ένα από αυτά τα μέρη αναλαμβάνει έναν ξεχωριστό και συγκεκριμένο ρόλο στο έργο μας, εξασφαλίζοντας την ομαλή λειτουργία και την αλληλεπίδραση μεταξύ των διαφόρων στοιχείων του συστήματος.

Στο τοπικό επίπεδο, η κύρια συσκευή είναι το Arduino, το οποίο συνδέεται με διάφορους αισθητήρες όπως ο DHT22 για τη συλλογή δεδομένων θερμοκρασίας και υγρασίας. Τα δεδομένα αυτά στέλνονται μέσω ενός καλωδίου στο Node MCU (ESP), το οποίο λειτουργεί ως μεσολαβητής για την επικοινωνία με τον server. Το Node MCU χρησιμοποιεί WebSocket για τη σύνδεση με τον server, επιτρέποντας την αμφίδρομη επικοινωνία και τη μεταφορά δεδομένων σε πραγματικό χρόνο. Παράλληλα, το Arduino λαμβάνει εντολές από τον server για τον έλεγχο εξωτερικών συσκευών όπως LEDs.

Στο cloud, ο server είναι υλοποιημένος σε Python και είναι υπεύθυνος για την επεξεργασία και αποθήκευση των δεδομένων που λαμβάνει από το τοπικό επίπεδο. Ο server συνδέεται με το Node MCU μέσω WebSocket για την αποστολή και λήψη δεδομένων σε πραγματικό χρόνο. Επιπλέον, ο server χρησιμοποιεί HTTP αιτήσεις για την αποστολή δεδομένων σε πλατφόρμες όπως το ThinkSpeak για περαιτέρω ανάλυση και επεξεργασία.

Η front-end διεπαφή, υλοποιημένη με HTML, CSS και JavaScript, επικοινωνεί με τον server μέσω HTTP αιτήσεων για την προβολή των δεδομένων στον χρήστη. Η διεπαφή αυτή ανανεώνεται δυναμικά, παρέχοντας στους χρήστες άμεση πρόσβαση στις πληροφορίες που συλλέγονται και επεξεργάζονται από το σύστημα.



Σχήμα 4.1 : Δομή και τοπολογία εφαρμογής [9]

Μέσω αυτής της δομής, επιτυγχάνεται μια οργανωμένη και αποτελεσματική διαχείριση των δεδομένων και της επικοινωνίας. Το σύστημα είναι σε θέση να συλλέγει περιβαλλοντικά δεδομένα από το Arduino, να τα μεταφέρει μέσω του ESP στον διακομιστή, και στη συνέχεια να λαμβάνει αποφάσεις που επιστρέφονται πίσω στο Arduino για την ανάλογη δράση. Αυτή η διαδικασία διασφαλίζει ότι όλες οι συσκευές λειτουργούν αρμονικά και σε πραγματικό χρόνο, προσφέροντας έτσι αυξημένη λειτουργικότητα και αξιοπιστία στο σύστημα.

## 4.2 ARDUINO

### 4.2.1 ΒΙΒΛΙΟΘΗΚΗ Arduino.h

Η βιβλιοθήκη Arduino.h αποτελεί τον θεμέλιο λίθο για τον προγραμματισμό πλακετών Arduino, παρέχοντας τα απαραίτητα εργαλεία για την ανάπτυξη εφαρμογών σε αυτό το περιβάλλον.

Αυτή η βασική βιβλιοθήκη περιλαμβάνει δομές δεδομένων και συναρτήσεις που είναι κρίσιμες για τη λειτουργία και τον έλεγχο του Arduino. Μέσω της Arduino.h, οι προγραμματιστές αποκτούν πρόσβαση σε μια πληθώρα λειτουργιών του μικροελεγκτή, όπως η διαχείριση των ψηφιακών και αναλογικών εισόδων και εξόδων, ο χειρισμός των αισθητήρων και η επικοινωνία με άλλες συσκευές.

Οι ψηφιακές και αναλογικές εισοδοί και εξοδοί είναι ζωτικής σημασίας για την αλληλεπίδραση του Arduino με το φυσικό περιβάλλον. Η Arduino.h παρέχει εύχρηστες συναρτήσεις για την ανάγνωση τιμών από αισθητήρες (όπως θερμοκρασίας, υγρασίας, φωτός κ.ά.) και την αποστολή σημάτων σε ενεργοποιητές (όπως LED, μοτέρ, και άλλες ηλεκτρονικές συσκευές). Αυτές οι λειτουργίες επιτρέπουν στους χρήστες να δημιουργούν σύνθετες και διαδραστικές εφαρμογές με σχετική ευκολία.

Επιπλέον, η Arduino.h διευκολύνει την επικοινωνία του Arduino με άλλες συσκευές μέσω διαφόρων πρωτοκόλλων επικοινωνίας, όπως UART, I2C και SPI. Αυτό είναι ιδιαίτερα χρήσιμο όταν απαιτείται η σύνδεση του Arduino με εξωτερικές μονάδες, όπως οθόνες, πρόσθετους αισθητήρες ή άλλους μικροελεγκτές, καθιστώντας δυνατή τη δημιουργία πιο περίπλοκων και ολοκληρωμένων συστημάτων.

Η ενσωμάτωση της Arduino.h γίνεται αυτόματα όταν δημιουργείται ένα νέο πρόγραμμα στο Arduino IDE (Integrated Development Environment). Αυτή η ευκολία χρήσης σημαίνει ότι οι χρήστες δεν χρειάζεται να δηλώσουν χειροκίνητα τη βιβλιοθήκη στην αρχή του κώδικα τους. Η αυτόματη προσθήκη της Arduino.h διασφαλίζει ότι όλοι οι βασικοί πόροι και λειτουργίες είναι άμεσα διαθέσιμοι, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στην ανάπτυξη των εφαρμογών τους χωρίς να ανησυχούν για τις βασικές ρυθμίσεις και διαμορφώσεις.

## **ΥΛΙΚΗ ΥΛΟΠΟΙΗΣΗ**

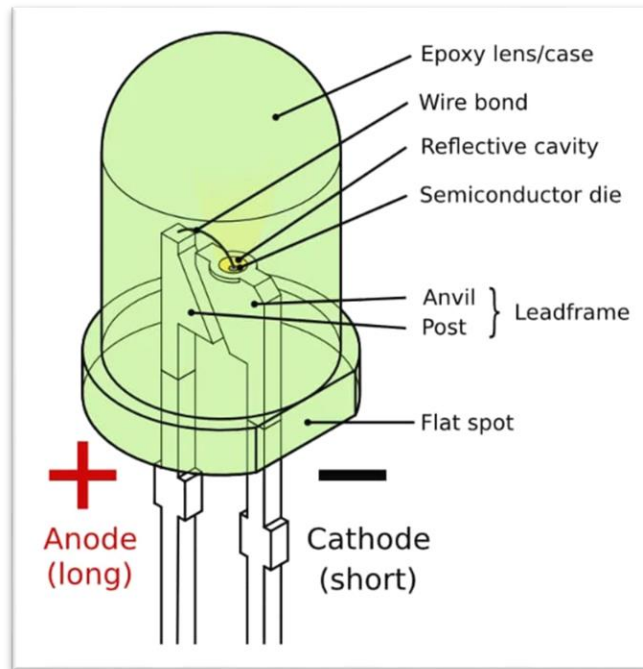
### **ΔΙΟΔΟΣ ΕΚΠΟΜΠΗΣ ΦΩΤΟΣ**

Το LED (Light-Emitting Diode) είναι μια ηλεκτρονική συσκευή η οποία εκπέμπει φως όταν διέρχεται ρεύμα μέσω αυτής[29]. Είναι ένας τύπος ημιαγωγικού φωτοδιόδου, ο οποίος λειτουργεί με την αρχή της εκπομπής φωτός μέσω της επανασύνδεσης των ηλεκτρονίων μέσα σε μια ημιαγωγική ουσία.

Οι LED έχουν πολλά πλεονεκτήματα σε σχέση με άλλες μορφές φωτισμού, όπως οι λάμπες φθορισμού και οι λαμπτήρες αλογόνου. Κάποια από αυτά περιλαμβάνουν την μεγαλύτερη διάρκεια ζωής, την χαμηλή κατανάλωση ενέργειας, το γρήγορο χρόνο ανταπόκρισης και την ποικιλία των χρωμάτων που μπορούν να παράγουν[30].

Λόγω των χαρακτηριστικών τους, τα LED χρησιμοποιούνται ευρέως σε διάφορες εφαρμογές, όπως ο φωτισμός σπιτιών και κτιρίων, η οθόνη φωτεινών ενδείξεων (LED displays), οι οπίσθιες

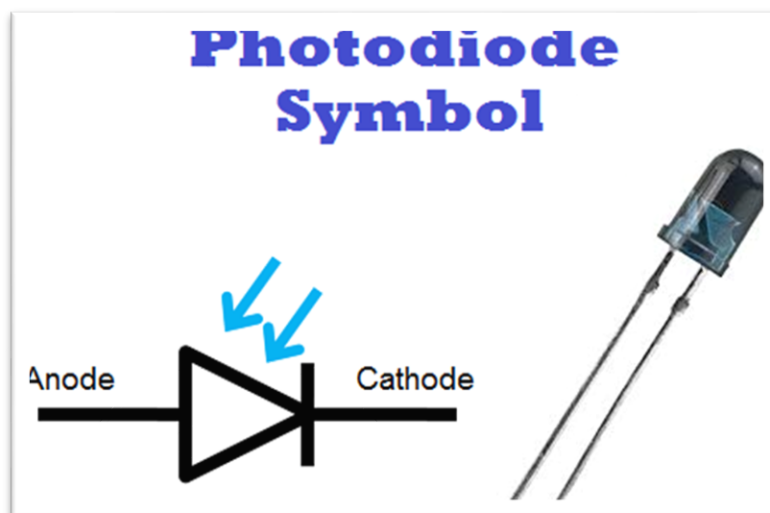
οθόνες τηλεόρασης, οι φωτεινοί φακοί και πολλές άλλες εφαρμογές στον τομέα της ηλεκτρονικής και της φωτιστικής.



Εικόνα 4.2 : Δίοδος εκπομπής φωτός και ο συμβολισμός της [6]

### ΦΩΤΟΔΙΟΔΟΣ

Η φωτοδίοδος είναι ένα ημιαγωγικό στοιχείο που μετατρέπει το φως σε ηλεκτρικό ρεύμα. Λειτουργεί παρόμοια με μια απλή δίοδο, αλλά έχει τη δυνατότητα να ανιχνεύει το φως. Η φωτοδίοδος είναι κατασκευασμένη από ημιαγωγικά υλικά όπως το πυρίτιο, και όταν το φως πέσει πάνω της, δημιουργεί ένα ηλεκτρικό ρεύμα μέσω του φωτοηλεκτρικού φαινομένου[24].



Εικόνα 4.3 : Φωτοδίοδος και ο συμβολισμός της [5]

Υπάρχουν διάφοροι τύποι φωτοδιόδων, όπως οι φωτοδιόδοι PIN και οι φωτοδιόδοι APD (Avalanche Photodiodes)[24]. Οι φωτοδιόδοι PIN είναι ιδιαίτερα ευαίσθητες στο φως και χρησιμοποιούνται σε εφαρμογές όπως οι αισθητήρες φωτός και οι φωτομετρικές συσκευές[25]. Οι φωτοδιόδοι APD λειτουργούν σε υψηλές τάσεις και παρέχουν μεγαλύτερη ευαισθησία, καθιστώντας τις κατάλληλες για τηλεπικοινωνίες και ανιχνευτές φωτός χαμηλής έντασης.

Οι **φωτοδιόδοι** έχουν ευρεία εφαρμογή σε διάφορους τομείς, όπως:

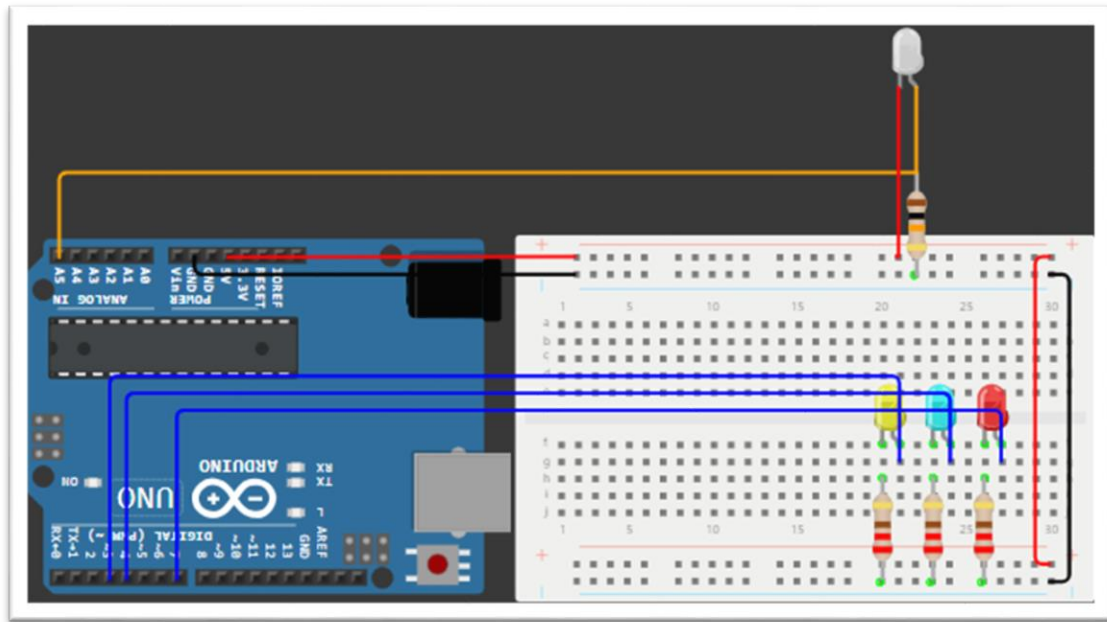
1. **Ανίχνευση φωτός:** Χρησιμοποιούνται σε συστήματα συναγερμών, τηλεχειριστήρια και αισθητήρες φωτός.
2. **Επικοινωνίες:** Χρησιμοποιούνται σε συστήματα οπτικών ινών για τη μετατροπή των οπτικών σημάτων σε ηλεκτρικά σήματα.
3. **Ιατρικές συσκευές:** Εφαρμόζονται σε οξύμετρα για τη μέτρηση της συγκέντρωσης του οξυγόνου στο αίμα.
4. **Βιομηχανία:** Χρησιμοποιούνται σε συστήματα αυτόματου ελέγχου και ρομποτικής για την ανίχνευση αντικειμένων και την πλοήγηση.

Η φωτοδιόδος είναι ένα κρίσιμο στοιχείο σε πολλές τεχνολογικές εφαρμογές λόγω της ικανότητάς της να μετατρέπει το φως σε ηλεκτρικό σήμα με υψηλή ακρίβεια και αξιοπιστία[26].

Στο συγκεκριμένο βήμα, προστέθηκε μια φωτοδιόδος στη θύρα A5 του Arduino και τρία LEDs στα pins 3, 4 και 7. Οι αναλογικές θύρες του Arduino μπορούν να διαβάζουν και να ερμηνεύουν μια συνεχή σειρά τιμών, επιτρέποντας την ακριβή μέτρηση παραμέτρων όπως η ένταση φωτός ή η θερμοκρασία. Αντίθετα, οι ψηφιακές θύρες λειτουργούν μόνο σε δύο καταστάσεις: υψηλή (HIGH) ή χαμηλή (LOW), δηλαδή αναγνωρίζουν μόνο αν υπάρχει τάση ή όχι, καθιστώντας τις ιδανικές για απλές λειτουργίες όπως η ενεργοποίηση ή απενεργοποίηση ενός LED. Αυτή η συνδεσμολογία επιτρέπει τη χρήση της βιβλιοθήκης Arduino.h για τον έλεγχο αυτών των στοιχείων.



Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home



Εικόνα 4.4 : Συνδεσμολογία Arduino με απλά components [10]

Συνδεσμολογία φωτοδιόδου	
<b>ANODE</b>	5V
<b>KATHODE</b>	GND(μέσα από αντίσταση 1k $\omega$ ) και A5 ARDUINO

Πίνακας 4.1 : Συνδεσμολογία φωτοδιόδου

Συνδεσμολογία LED	
<b>ANODE</b>	PIN 3,4,7 ARDUINO
<b>KATHODE</b>	GND(μέσα από αντίσταση 220 $\omega$ )

Πίνακας 4.2 : Συνδεσμολογία LED

#### 4.2.2 ΒΙΒΛΙΟΘΗΚΗ DHT\_U.h και DHT.h

Οι βιβλιοθήκες `DHT\_U.h` και `DHT.h` είναι κρίσιμες για την αλληλεπίδραση με τους αισθητήρες DHT (Digital Humidity and Temperature), οι οποίοι είναι ευρέως χρησιμοποιούμενοι για την ακριβή μέτρηση της θερμοκρασίας και της υγρασίας σε έργα Arduino. Αυτές οι βιβλιοθήκες παρέχουν τις απαραίτητες συναρτήσεις και εργαλεία για την ανάγνωση δεδομένων από τους αισθητήρες DHT11, DHT22 και AM2302.

Η `DHT.h` είναι η κύρια βιβλιοθήκη που χρησιμοποιείται για τη βασική λειτουργικότητα των αισθητήρων DHT. Περιλαμβάνει συναρτήσεις για την αρχικοποίηση του αισθητήρα, την ανάγνωση της θερμοκρασίας και της υγρασίας, καθώς και για την επεξεργασία των δεδομένων που

λαμβάνονται. Αυτή η βιβλιοθήκη είναι εύκολη στη χρήση και παρέχει έναν απλό και αποδοτικό τρόπο για τη συλλογή περιβαλλοντικών δεδομένων.

Η `DHT\_U.h` είναι μια επέκταση της βασικής βιβλιοθήκης `DHT.h` και προσφέρει επιπλέον λειτουργικότητες για τη διαχείριση των αισθητήρων DHT. Είναι μέρος του Adafruit Unified Sensor Library, το οποίο παρέχει ένα ενιαίο σύστημα διαχείρισης για διάφορους τύπους αισθητήρων. Η `DHT\_U.h` επιτρέπει την ενσωμάτωση των αισθητήρων DHT στο ευρύτερο πλαίσιο αυτής της ενοποιημένης βιβλιοθήκης, διευκολύνοντας τη χρήση πολλών διαφορετικών αισθητήρων σε ένα ενιαίο πρόγραμμα.

Η χρήση αυτών των βιβλιοθηκών είναι απλή και ευέλικτη. Οι προγραμματιστές πρέπει να συμπεριλάβουν τις γραμμές `#include <DHT.h>` και `#include <DHT\_U.h>` στην αρχή του προγράμματος τους. Ακολούθως, μπορούν να αρχικοποιήσουν τον αισθητήρα και να χρησιμοποιήσουν τις σχετικές συναρτήσεις για την ανάγνωση της θερμοκρασίας και της υγρασίας. Αυτές οι βιβλιοθήκες είναι ιδανικές για έργα που απαιτούν την παρακολούθηση των περιβαλλοντικών συνθηκών, όπως τα έξυπνα σπίτια, οι μετεωρολογικοί σταθμοί και τα συστήματα ελέγχου κλίματος.

Οι βιβλιοθήκες `DHT\_U.h` και `DHT.h` έχουν σχεδιαστεί για να είναι συμβατές με το Arduino IDE και προσφέρουν τεκμηρίωση για να διευκολύνουν τους χρήστες στην ανάπτυξη των εφαρμογών τους. Η ευρεία χρήση και υποστήριξή τους από την κοινότητα Arduino τις καθιστούν μια αξιόπιστη επιλογή για τη μέτρηση της θερμοκρασίας και της υγρασίας, παρέχοντας ακριβή και αξιόπιστα δεδομένα για διάφορες εφαρμογές.

## ΥΛΙΚΗ ΥΛΟΠΟΙΗΣΗ

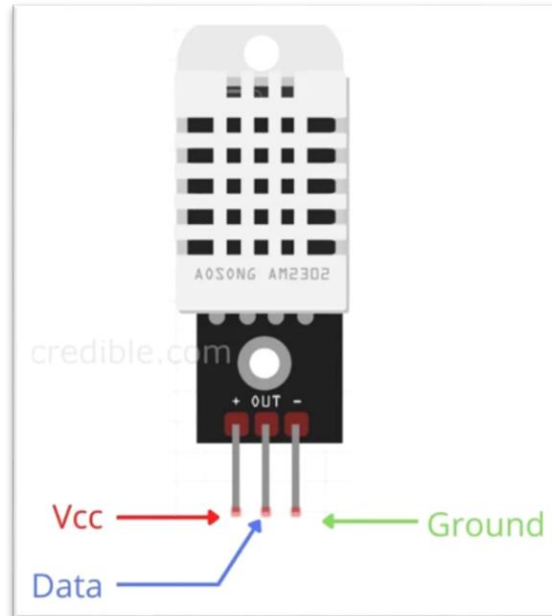
### DHT22

Το DHT22, γνωστό και ως AM2302[23], είναι ένας δημοφιλής αισθητήρας θερμοκρασίας και υγρασίας που χρησιμοποιείται ευρέως σε έργα Arduino και άλλες εφαρμογές μικροελεγκτών. Ο αισθητήρας είναι γνωστός για την ακρίβεια και την αξιοπιστία του, καθιστώντας τον ιδανικό για έργα που απαιτούν ακριβείς μετρήσεις περιβαλλοντικών συνθηκών.

#### Χαρακτηριστικά

1. **Ακρίβεια:** Ο DHT22 προσφέρει υψηλή ακρίβεια μετρήσεων. Η ακρίβεια θερμοκρασίας είναι  $\pm 0.5^{\circ}\text{C}$  και η ακρίβεια υγρασίας είναι  $\pm 2-5\%$ .
2. **Εύρος Μέτρησης:**
  - Θερμοκρασία:  $-40^{\circ}\text{C}$  έως  $80^{\circ}\text{C}$
  - Υγρασία: 0% έως 100% RH (Relative Humidity)

3. Συνδεσιμότητα: Ο αισθητήρας χρησιμοποιεί πρωτόκολλο επικοινωνίας 1-wire, καθιστώντας τον εύκολο στη σύνδεση με μικροελεγκτές όπως το Arduino.
4. Μέγεθος: Είναι μικρός και συμπαγής, διευκολύνοντας την ενσωμάτωσή σε διάφορα έργα.



Εικόνα 4.5 : Τεχνικές λεπτομέρειες αισθητήρα DHT22 [4]

#### Διάγραμμα Σύνδεσης

1. **VCC:** Παροχή ρεύματος (συνήθως 3.3V ή 5V).
2. **DATA:** Γραμμή δεδομένων για επικοινωνία με τον μικροελεγκτή.
3. **GND:** Γείωση.

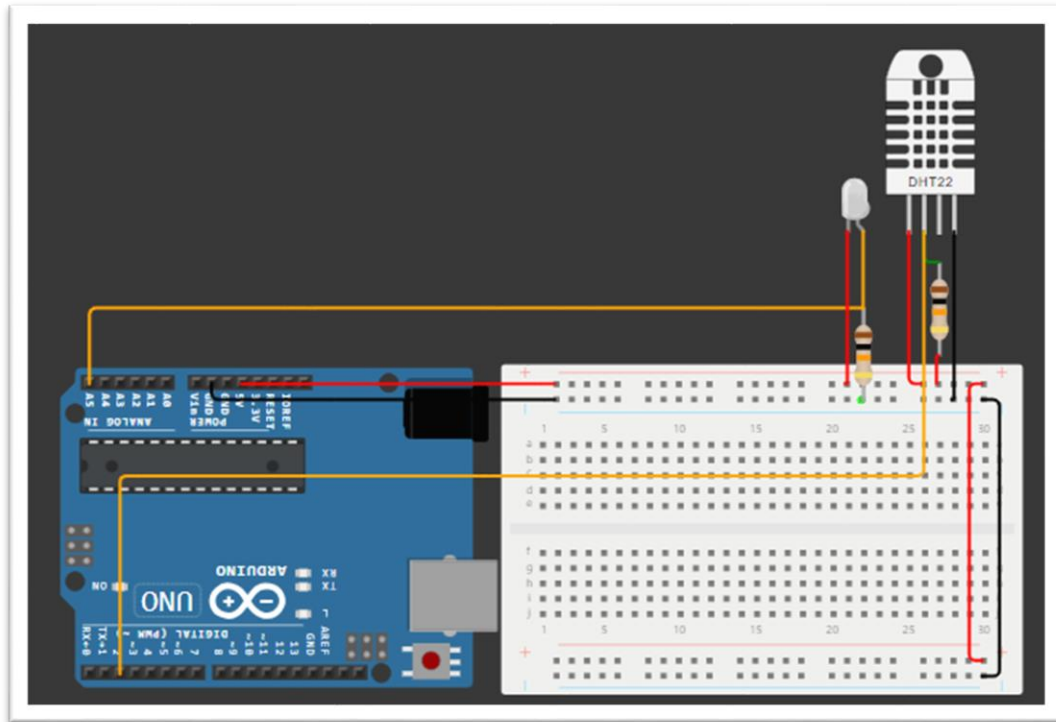
Οι εφαρμογές του DHT22 είναι ποικίλες και εκτείνονται σε διάφορους τομείς. Στα έξυπνα σπίτια, ο αισθητήρας χρησιμοποιείται για την παρακολούθηση και τη ρύθμιση της θερμοκρασίας και της υγρασίας σε διάφορα δωμάτια, βελτιώνοντας έτσι την άνεση και την ενεργειακή αποδοτικότητα. Στους μετεωρολογικούς σταθμούς, συμβάλλει στην καταγραφή περιβαλλοντικών δεδομένων που χρησιμοποιούνται για ανάλυση και πρόγνωση καιρού. Στη γεωργία, ο DHT22 παίζει σημαντικό ρόλο στον έλεγχο των συνθηκών θερμοκρασίας και υγρασίας μέσα στα θερμοκήπια, προάγοντας την υγιή ανάπτυξη των καλλιεργειών. Επιπλέον, στα συστήματα HVAC, ο αισθητήρας βοηθά στη βελτιστοποίηση της απόδοσης των συστημάτων θέρμανσης, αερισμού και κλιματισμού, εξασφαλίζοντας ένα άνετο και ασφαλές περιβάλλον διαβίωσης.

Ο DHT22 είναι ένας οικονομικός και αξιόπιστος αισθητήρας για τη μέτρηση της θερμοκρασίας και της υγρασίας, κατάλληλος για πολλά έργα DIY και επαγγελματικές εφαρμογές. Η

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home

εύκολη ενσωμάτωσή του με το Arduino και άλλα συστήματα μικροελεγκτών τον καθιστά δημοφιλή επιλογή για όσους επιθυμούν να αναπτύξουν έργα IoT και αυτοματισμών.

Στο συγκεκριμένο βήμα της υλοποίησης, προστέθηκε ο αισθητήρας DHT22 στο pin 2 του Arduino. Ο αισθητήρας συνδέθηκε σύμφωνα με το παρακάτω διάγραμμα σύνδεσης:



Εικόνα 4.6 : Συνδεσμολογία Arduino με αισθητήρες [10]

Συνδεσμολογία DHT22	
VCC	5V
DATA	5V (μεσα από αντισταση 1k $\omega$ ) και P2 ARDUINO
GND	GND

Πίνακας 4.3 : Συνδεσμολογία DHT22

#### 4.2.3 ΒΙΒΛΙΟΘΗΚΗ LiquidCrystal.h

Η βιβλιοθήκη 'LiquidCrystal.h' είναι μια βασική βιβλιοθήκη στο περιβάλλον ανάπτυξης Arduino, που χρησιμοποιείται για τον έλεγχο οθονών LCD (Liquid Crystal Display). Αυτή η βιβλιοθήκη επιτρέπει στους χρήστες να επικοινωνούν με οθόνες LCD που συνδέονται με το Arduino, επιτρέποντας την εμφάνιση κειμένου και συμβόλων. Οι οθόνες LCD είναι πολύ χρήσιμες για την προβολή πληροφοριών, μηνυμάτων και δεδομένων από διάφορους αισθητήρες σε πραγματικό χρόνο.

Με τη βοήθεια της `LiquidCrystal.h`, οι χρήστες μπορούν να διαχειριστούν τις οθόνες LCD εύκολα και αποτελεσματικά. Η βιβλιοθήκη περιλαμβάνει συναρτήσεις για τη ρύθμιση της οθόνης, την εκτύπωση κειμένου, την εκκαθάριση της οθόνης, τη μετακίνηση του κέρσορα και πολλές άλλες λειτουργίες. Οι οθόνες LCD μπορούν να έχουν διαφορετικές διαστάσεις, όπως 16x2 ή 20x4, και η βιβλιοθήκη υποστηρίζει πολλούς τύπους και μεγέθη.

Η χρήση της βιβλιοθήκης είναι πολύ απλή. Οι χρήστες πρέπει να προσθέσουν τη γραμμή `#include <LiquidCrystal.h>` στην αρχή του προγράμματος τους. Στη συνέχεια, μπορούν να δηλώσουν ένα αντικείμενο τύπου `LiquidCrystal` και να το αρχικοποιήσουν με τους κατάλληλους ακροδέκτες που είναι συνδεδεμένοι στην οθόνη LCD. Αφού γίνει η αρχικοποίηση, οι διάφορες συναρτήσεις της βιβλιοθήκης μπορούν να χρησιμοποιηθούν για την εμφάνιση δεδομένων στην οθόνη.

Η `LiquidCrystal.h` είναι ιδανική για έργα όπου η προβολή δεδομένων είναι κρίσιμη. Σε εφαρμογές όπως θερμοστάτες, μετεωρολογικοί σταθμοί, συστήματα παρακολούθησης και ελέγχου, οι οθόνες LCD παρέχουν μια εύκολη και γρήγορη μέθοδο για την οπτικοποίηση των πληροφοριών. Η βιβλιοθήκη αυτή επιτρέπει στους προγραμματιστές να δημιουργούν διαδραστικές εφαρμογές με ελάχιστο κώδικα.

## ΥΛΙΚΗ ΥΛΟΠΟΙΗΣΗ

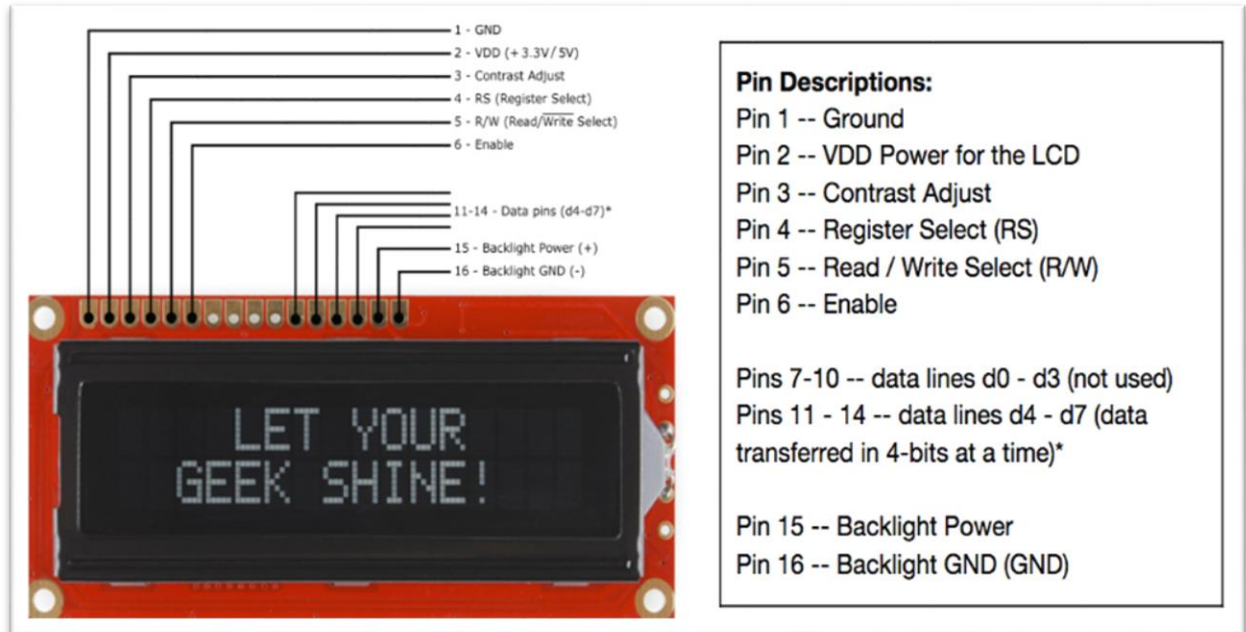
### ΟΘΟΝΗ ΥΓΡΩΝ ΚΡΥΣΤΑΛΛΩΝ

Το LCD (Liquid Crystal Display) είναι μια τεχνολογία οθόνης που χρησιμοποιεί υγρά κρυστάλλια για να παράγει εικόνες. Οι LCD οθόνες αποτελούνται από έναν αριθμό από μικροσκοπικά κρυσταλλικά κύτταρα τα οποία μπορούν να αλλάζουν την πολικότητα τους υπό την επίδραση της ηλεκτρικής τάσης[31].

#### Χαρακτηριστικά

1. **Μέγεθος Οθόνης:** Οι τυπικές LCD οθόνες που χρησιμοποιούνται με μικροελεγκτές έχουν διαστάσεις 16x2 ή 20x4, δηλαδή 16 ή 20 χαρακτήρες ανά γραμμή και 2 ή 4 γραμμές αντίστοιχα.
2. **Τύπος Ελέγχου:** Οι περισσότερες LCD οθόνες χρησιμοποιούν τον ελεγκτή HD44780 ή συμβατό, που επιτρέπει εύκολη σύνδεση με μικροελεγκτές όπως το Arduino.
3. **Τρόπος Εμφάνισης:** Μπορούν να εμφανίσουν χαρακτήρες ASCII καθώς και κάποιους ειδικούς χαρακτήρες ή εικονίδια.
4. **Οπίσθιος Φωτισμός:** Περιλαμβάνουν οπίσθιο φωτισμό που βελτιώνει την ορατότητα σε συνθήκες χαμηλού φωτισμού.
5. **Τροφοδοσία Ρεύματος:** Συνήθως λειτουργούν σε τάσεις 5V, αλλά υπάρχουν και εκδόσεις που λειτουργούν με 3.3V.

6. **Κατανάλωση Ρεύματος:** Η κατανάλωση ρεύματος εξαρτάται από τη χρήση του οπίσθιου φωτισμού, συνήθως είναι της τάξης των λίγων mA.
7. **Ευκολία Χρήσης:** Οι LCD οθόνες είναι εύκολες στη χρήση με μικροελεγκτές και υποστηρίζονται από πληθώρα βιβλιοθηκών, όπως η LiquidCrystal για το Arduino.



Εικόνα 4.7 : LCD display και τεχνικές λεπτομέρειες [7]

#### Διάγραμμα Σύνδεσης

1. **VSS:** Γείωση (Ground)
2. **VDD:** Παροχή ρεύματος (5V)
3. **V0:** Ρύθμιση αντίθεσης (συνήθως συνδέεται με έναν ποτενσιόμετρο)
4. **RS (Register Select):** Επιλογή καταχωρητή (συνδέεται σε ψηφιακή ακίδα του μικροελεγκτή)
5. **RW (Read/Write):** Ανάγνωση/Εγγραφή (συνήθως συνδέεται στη γείωση για εγγραφή)
6. **E (Enable):** Σήμα ενεργοποίησης (συνδέεται σε ψηφιακή ακίδα του μικροελεγκτή)
7. **D0-D7:** Δεδομένα (συνδέονται στις ψηφιακές ακίδες του μικροελεγκτή, συνήθως χρησιμοποιούνται μόνο D4-D7 για λειτουργία 4-bit)
8. **A (Anode):** Οπίσθιος φωτισμός θετικός (συνδέεται σε 5V μέσω αντιστάτη)
9. **K (Cathode):** Οπίσθιος φωτισμός αρνητικός (συνδέεται στη γείωση)

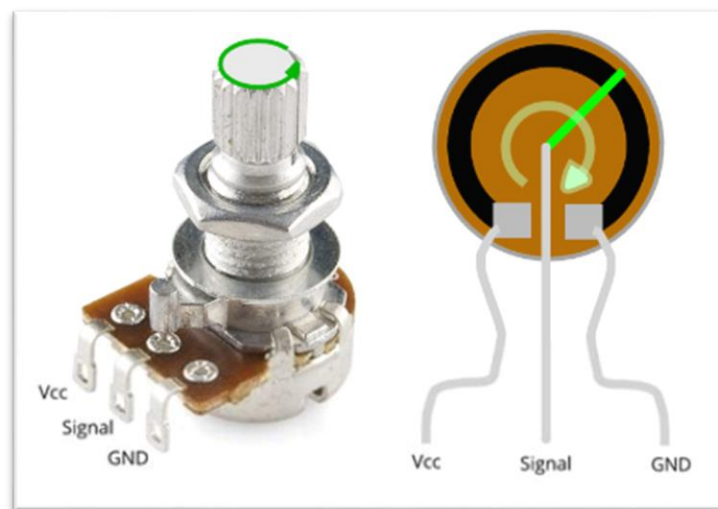
Η λειτουργία των LCD οθονών βασίζεται στην ιδιότητα των υγρών κρυστάλλων να αλλάζουν την φωτεινότητά τους ανάλογα με την εφαρμοζόμενη ηλεκτρική τάση. Κάθε κρυσταλλικό κύτταρο αντιπροσωπεύει ένα pixel της οθόνης και μπορεί να είναι είτε σε κατάσταση "on" (διαφανής), είτε σε κατάσταση "off" (αδιαφανής), ανάλογα με την εφαρμοζόμενη τάση[32].

Οι LCD οθόνες είναι δημοφιλείς λόγω της λεπτής τους κατασκευής, της χαμηλής κατανάλωσης ενέργειας και της ικανότητάς τους να παράγουν καθαρές εικόνες με υψηλή ανάλυση[33]. Χρησιμοποιούνται ευρέως σε φορητές συσκευές όπως τα κινητά τηλέφωνα, οι φορητοί υπολογιστές, τα ψηφιακά ρολόγια, οι τηλεοράσεις, τα ψηφιακά πλαίσια φωτογραφιών και πολλές άλλες εφαρμογές.

## ΠΟΤΕΝΣΙΟΜΕΤΡΟ

Το ποτενσιόμετρο είναι ένα ηλεκτρικό εξάρτημα που χρησιμοποιείται για τη μεταβολή της αντίστασής του, η οποία μπορεί να προσαρμοστεί ανάλογα με τη θέση ενός κινούμενου μέρους, όπως ένας περιστρεφόμενος άξονας ή ένας διακόπτης[34]. Αυτό το κινούμενο μέρος συνδέεται με έναν αγωγό αντίστασης, ο οποίος μπορεί να αλλάξει το μήκος του ή την επιφάνεια επαφής ανάλογα με την κίνηση που γίνεται.

Η χρήση ενός ποτενσιόμετρου επιτρέπει στον χρήστη να προσαρμόζει την αντίσταση του κυκλώματος, προκειμένου να επηρεάσει την τάση ή το ρεύμα που διαρρέει το κύκλωμα. Στην πράξη, αυτό σημαίνει ότι ο χρήστης μπορεί να αλλάξει την ένταση ή το επίπεδο της τάσης που φτάνει σε ένα συγκεκριμένο σημείο του κυκλώματος, προσαρμόζοντας τη θέση του κινούμενου μέρους του ποτενσιόμετρου.



Εικόνα 4.8 : Ποτενσιόμετρο και τεχνικές λεπτομέρειες [8]

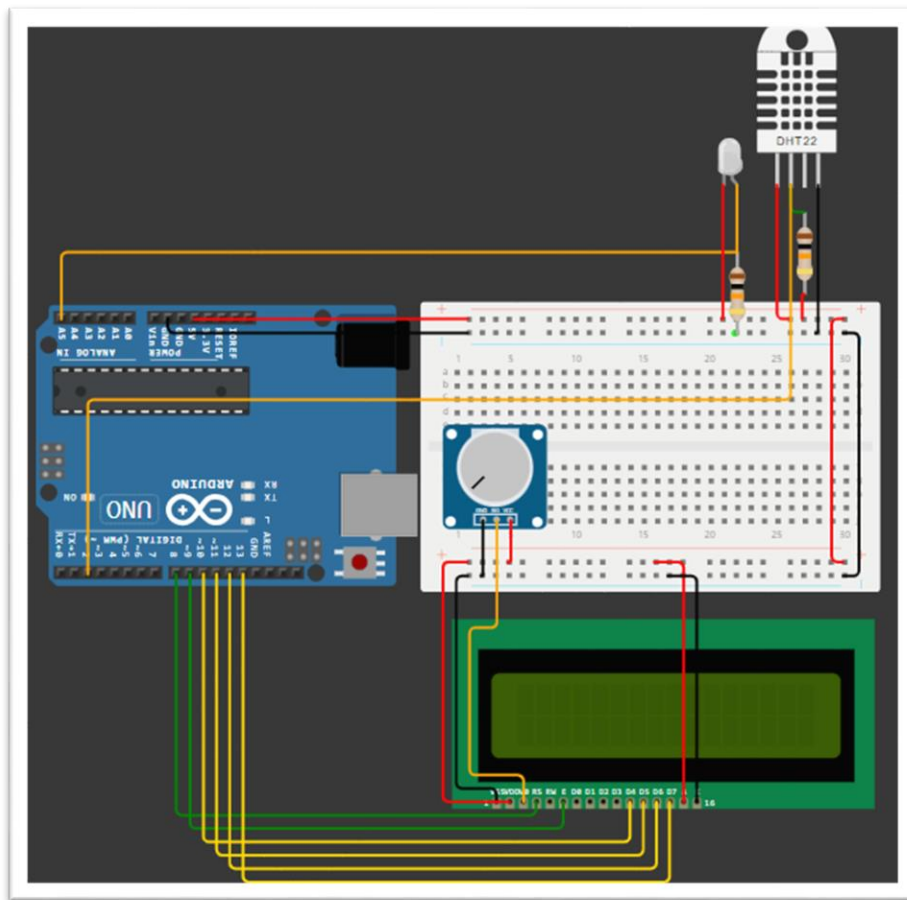
### Διάγραμμα Σύνδεσης

1. **Πόλος Καλωδίου:** Συνδέεται σε μια από τις αναλογικές εισόδους (A0-A5) του Arduino.
2. **Πόλος Καλωδίου (Κέντρο):** Συνδέεται στην τροφοδοσία (5V ή 3.3V) του Arduino.
3. **Πόλος Καλωδίου (Άκρο):** Συνδέεται στη γείωση (GND) του Arduino.

Συσκευή συλλογής δεδομένων και επικοινωνία με server με την χρήση διαφόρων πρωτοκόλλων επικοινωνίας για την επίτευξη ενός smart home

Το ποτενσιόμετρο συνήθως χρησιμοποιείται για τον έλεγχο της τάσης ή του ρεύματος σε ένα κύκλωμα. Μεταβάλλοντας τη θέση του κινούμενου μέρους, ο χρήστης μπορεί να αυξήσει ή να μειώσει την αντίσταση του ποτενσιόμετρου, επηρεάζοντας έτσι την τάση ή το ρεύμα στο κύκλωμα.

Στο συγκεκριμένο βήμα, προστέθηκε μια LCD οθόνη στη θύρα 8-15 του Arduino. Αυτό επιτρέπει την εμφάνιση των δεδομένων από τους αισθητήρες σε πραγματικό χρόνο, διευκολύνοντας την παρακολούθηση και τον έλεγχο του συστήματος.



Εικόνα 4.9 : Συνδεσμολογία Arduino με LCD οθόνη [10]

Συνδεσμολογία LCD	
VSS	GND
VDD	5V
V0	SIGNAL PIN ΠΟΤΕΝΣΙΟΠΕΤΡΟΥ



<b>RS</b>	P8 ARDUINO
<b>RW</b>	
<b>E</b>	P9 ARDUINO
<b>DP4-DP7</b>	P10-P13 ARDUINO
<b>A</b>	5V
<b>K</b>	GND

**Πίνακας 4.4 :** Συνδεσμολογία LCD οθόνης

<b>Συνδεσμολογία Ποτενσιόμετρου</b>	
<b>VSS</b>	5V
<b>SIGNAL</b>	V0 LCD
<b>GRD</b>	GND

**Πίνακας 4.5 :** Συνδεσμολογία Ποτενσιόμετρου

#### 4.2.4 ΒΙΒΛΙΟΘΗΚΗ Wire.h

Η βιβλιοθήκη `Wire.h` είναι μια βασική βιβλιοθήκη στο περιβάλλον ανάπτυξης Arduino, που χρησιμοποιείται για την επικοινωνία μέσω του πρωτοκόλλου I2C (Inter-Integrated Circuit). Αυτή η βιβλιοθήκη επιτρέπει στο Arduino να επικοινωνεί με άλλα I2C συσκευές, όπως αισθητήρες, εκτελεστές και άλλους μικροελεγκτές, χρησιμοποιώντας μόνο δύο καλώδια (SDA για δεδομένα και SCL για χρονισμό).

Η `Wire.h` παρέχει λειτουργίες για την έναρξη της επικοινωνίας, την αποστολή και λήψη δεδομένων και τη διαχείριση των συσκευών I2C. Οι χρήστες πρέπει να προσθέσουν τη γραμμή `#include <Wire.h>` στην αρχή του προγράμματος τους και να καλέσουν τη συνάρτηση `Wire.begin()` στην ενότητα `setup()`. Στη συνέχεια, μπορούν να χρησιμοποιήσουν τις συναρτήσεις `Wire.requestFrom()`, `Wire.beginTransmission()`, `Wire.write()`, και `Wire.endTransmission()` για να επικοινωνήσουν με τις I2C συσκευές.

Η βιβλιοθήκη `Wire.h` είναι ιδιαίτερα χρήσιμη για έργα που απαιτούν τη σύνδεση πολλών συσκευών με το Arduino μέσω ενός κοινού διαύλου επικοινωνίας. Χρησιμοποιείται ευρέως σε εφαρμογές όπως μετρητές αισθητήρων, οθόνες, και συστήματα ελέγχου.

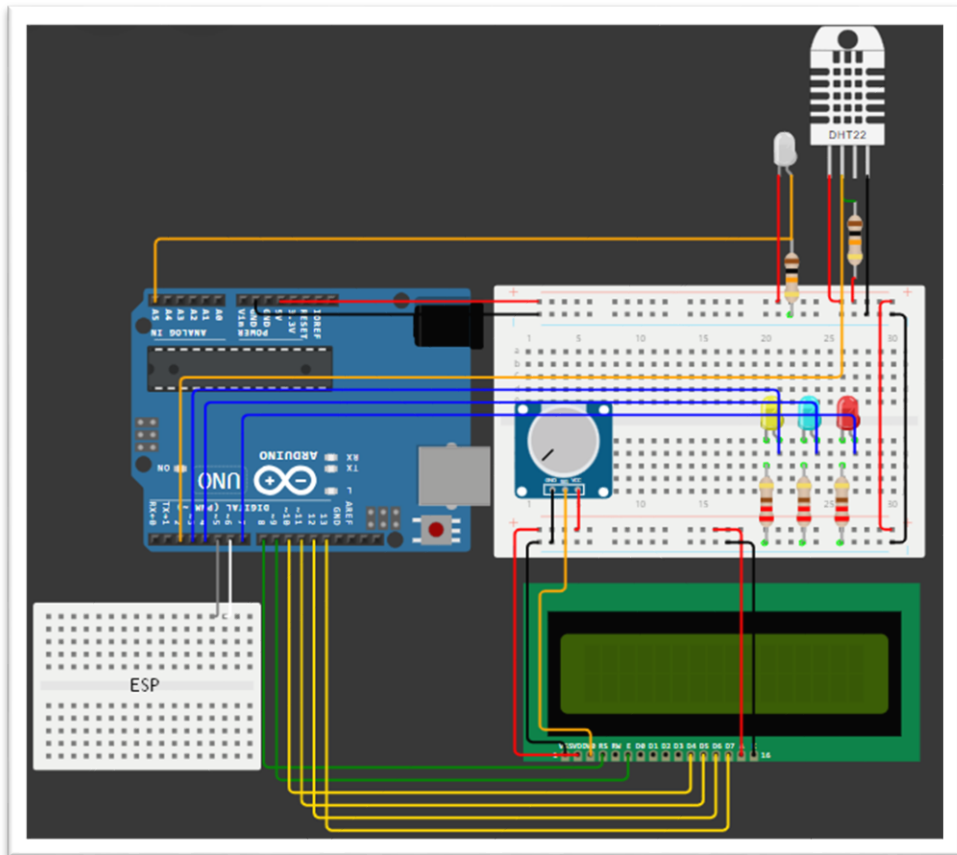
Η ευκολία χρήσης και η αποτελεσματικότητα της `Wire.h` την καθιστούν μια από τις πιο σημαντικές βιβλιοθήκες για την ανάπτυξη έργων με Arduino που περιλαμβάνουν I2C επικοινωνία. Η βιβλιοθήκη συνοδεύεται από παραδείγματα και τεκμηρίωση, διευκολύνοντας τους προγραμματιστές να ενσωματώσουν I2C συσκευές στα έργα τους.

#### **4.2.5 ΒΙΒΛΙΟΘΗΚΗ SoftwareSerial.h**

Η βιβλιοθήκη SoftwareSerial.h είναι ένα σημαντικό εργαλείο για τους χρήστες του Arduino που επιθυμούν να προσθέσουν πολλαπλές σειριακές θύρες στο πρόγραμμά τους. Το Arduino διαθέτει μία ενσωματωμένη σειριακή θύρα (Serial) που χρησιμοποιείται για επικοινωνία με τον υπολογιστή ή άλλες συσκευές μέσω του USB. Ωστόσο, πολλές φορές οι χρήστες χρειάζονται επιπλέον σειριακές θύρες για την επικοινωνία με άλλες συσκευές, όπως αισθητήρες, GPS μονάδες ή Bluetooth μονάδες. Η βιβλιοθήκη SoftwareSerial παρέχει τη δυνατότητα να δημιουργήσετε επιπλέον σειριακές θύρες χρησιμοποιώντας ψηφιακές ακίδες του Arduino.

Η βιβλιοθήκη SoftwareSerial επιτρέπει την επικοινωνία σειριακής διεπαφής σε οποιοδήποτε από τα ψηφιακά pins του Arduino, δημιουργώντας εικονικές σειριακές θύρες. Αυτό επιτυγχάνεται με τη χρήση της τεχνικής "bit-banging", η οποία επιτρέπει τη σειριακή επικοινωνία μέσω λογισμικού, χωρίς να απαιτείται ειδικό υλικό για επιπλέον σειριακές θύρες. Αυτό είναι ιδιαίτερα χρήσιμο για μοντέλα Arduino που διαθέτουν περιορισμένο αριθμό υλικών σειριακών θυρών, όπως τα Arduino Uno και Nano.

Στο συγκεκριμένο βήμα, προστέθηκε ο μικροελεγκτής NodeMCU και η σύνδεση μεταξύ του esp και του Arduino γίνεται στις θύρες 5 και 6 του esp και στις θύρες 5 και 6 του Arduino. Αυτό επιτρέπει την εμφάνιση των δεδομένων από τους αισθητήρες σε πραγματικό χρόνο, διευκολύνοντας την παρακολούθηση και τον έλεγχο του συστήματος.



Εικόνα 4.10 : Τελική συνδεσμολογία Arduino [10]

Συνδεσμολογία ESP	
D5	D5 ARDUINO
D6	D6 ARDUINO

Πίνακας 4.6 : Συνδεσμολογία ESP

Η βιβλιοθήκη SoftwareSerial χρησιμοποιείται για τη δημιουργία αντικειμένων τύπου SoftwareSerial. Κάθε αντικείμενο αντιπροσωπεύει μια εικονική σειριακή θύρα που μπορεί να διαμορφωθεί για να χρησιμοποιεί συγκεκριμένα pins για τη λήψη (RX) και αποστολή (TX) δεδομένων. Οι κύριες μέθοδοι της βιβλιοθήκης περιλαμβάνουν την `begin()`, που ρυθμίζει την ταχύτητα μετάδοσης (baud rate) της σειριακής επικοινωνίας και τις μεθόδους `read()`, `write()`, `available()`, και `listen()`.

- **begin(long speed):** Ρυθμίζει την ταχύτητα μετάδοσης της εικονικής σειριακής θύρας. Οι τυπικές ταχύτητες μετάδοσης είναι 9600, 14400, 19200, 38400, 57600 και 115200 baud.
- **read():** Διαβάζει τα δεδομένα που λαμβάνονται από την εικονική σειριακή θύρα.
- **write(byte val):** Στέλνει ένα byte δεδομένων μέσω της εικονικής σειριακής θύρας.

- **available():** Επιστρέφει τον αριθμό των bytes που είναι διαθέσιμα για ανάγνωση.
- **listen():** Ενεργοποιεί την ακρόαση της εικονικής σειριακής θύρας. Αυτό είναι απαραίτητο όταν υπάρχουν πολλαπλές εικονικές σειριακές θύρες, καθώς μόνο μία μπορεί να λαμβάνει δεδομένα κάθε φορά.

Ένα από τα κύρια πλεονεκτήματα της βιβλιοθήκης SoftwareSerial είναι η ευελιξία που παρέχει στους χρήστες να επεκτείνουν τις σειριακές δυνατότητες των Arduino τους χωρίς να απαιτείται επιπλέον υλικό. Είναι εξαιρετικά χρήσιμη σε έργα που απαιτούν την ταυτόχρονη επικοινωνία με πολλαπλές σειριακές συσκευές.

Ωστόσο, η βιβλιοθήκη SoftwareSerial έχει και κάποιους περιορισμούς. Η επικοινωνία μέσω λογισμικού είναι γενικά πιο αργή και λιγότερο αξιόπιστη από την επικοινωνία μέσω υλικού. Η ταχύτητα μετάδοσης περιορίζεται από την ταχύτητα του επεξεργαστή του Arduino και τις άλλες εργασίες που εκτελεί ο μικροελεγκτής. Επιπλέον, η χρήση της βιβλιοθήκης SoftwareSerial μπορεί να επηρεάσει την απόδοση άλλων χρονικά κρίσιμων λειτουργιών στο Arduino, καθώς η διαδικασία "bit-banging" καταλαμβάνει υπολογιστικούς πόρους.

Συνολικά, η βιβλιοθήκη SoftwareSerial.h είναι ένα πολύτιμο εργαλείο για τους χρήστες του Arduino που επιθυμούν να επεκτείνουν τις σειριακές δυνατότητες των έργων τους με ευελιξία και αποτελεσματικότητα. Παρότι έχει ορισμένους περιορισμούς, οι δυνατότητές της την καθιστούν αναντικατάστατη για πολλές εφαρμογές.

## 4.2.6 ΛΕΙΤΟΥΡΓΙΑ ARDUINO

### ΠΡΩΤΟ ΣΤΑΔΙΟ (SETUP)

Το πρώτο στάδιο της εκτέλεσης μιας εφαρμογής σε Arduino είναι το στάδιο της εκκίνησης. Σε αυτό το στάδιο, οι αρχικές ρυθμίσεις και αρχικοποιήσεις πραγματοποιούνται για να διασφαλιστεί η ομαλή λειτουργία του Arduino στο δεύτερο στάδιο.

Ο κώδικας ξεκινά με τον ορισμό διάφορων μεταβλητών και ακίδων του Arduino. Η μεταβλητή PHOTO\_pin δηλώνεται ως είσοδος για την ανάγνωση τιμών από έναν φωτοαντίσταση που είναι συνδεδεμένος στην αναλογική ακίδα A5. Οι μεταβλητές photo\_sensor\_val και photo\_val χρησιμοποιούνται για την αποθήκευση της αρχικής και της μετασχηματισμένης τιμής του αισθητήρα φωτοαντίστασης, αντίστοιχα. Η μεταβλητή photo\_led αντιστοιχεί στην ακίδα 3 του Arduino και χρησιμοποιείται για την έξοδο σε ένα LED που σχετίζεται με το φως.

Για τον αισθητήρα θερμοκρασίας και υγρασίας DHT22, χρησιμοποιείται η ακίδα 2 (DHT\_pin). Ο αισθητήρας αρχικοποιείται μέσω της κλάσης DHT\_Unified. Τα LEDs που σχετίζονται

με τη θερμοκρασία και την υγρασία αντιστοιχίζονται στις ακίδες 7 και 4, αντίστοιχα (tmp\_led και hmdt\_led).

Η οθόνη LCD αρχικοποιείται χρησιμοποιώντας την κλάση LiquidCrystal, με τις ακίδες ελέγχου 8, 9, 10, 11, 12 και 13 να συνδέονται στο Arduino. Δύο ειδικός χαρακτήρας, το σύμβολο του βαθμού (degree), ορίζεται ως πίνακας από bytes και δημιουργείται στη μνήμη της LCD με την εντολή lcd.createChar(1, degree);. Η οθόνη LCD χρησιμοποιείται για την εμφάνιση μηνυμάτων και δεδομένων από τους αισθητήρες.

Για την επικοινωνία με το ESP8266, χρησιμοποιείται η βιβλιοθήκη SoftwareSerial, η οποία δημιουργεί μια εικονική σειριακή σύνδεση στις ακίδες 5 και 6 (RX και TX αντίστοιχα). Η μεταβλητή receivedData χρησιμοποιείται για την αποθήκευση δεδομένων που λαμβάνονται από το ESP8266.

Η συνάρτηση setup εκτελείται μία φορά κατά την εκκίνηση του προγράμματος. Ξεκινά τη σειριακή επικοινωνία με το Arduino και το ESP8266 με ρυθμό baud 9600. Επιπλέον, αρχικοποιεί τον αισθητήρα DHT22 και την οθόνη LCD, εμφανίζοντας ένα μήνυμα "initialazing".

Οι ακίδες που χρησιμοποιούνται για τα LEDs ρυθμίζονται ως έξοδοι μέσω της εντολής pinMode. Αυτό επιτρέπει στο πρόγραμμα να ελέγχει την κατάσταση των LEDs, ανάλογα με τα δεδομένα που λαμβάνονται και τις ενέργειες που πρέπει να πραγματοποιηθούν.

Οι παραπάνω εντολές θα εκτελεστούν μόνο μια φορά κατά την ενεργοποίηση της συσκευής. Αυτές οι αρχικοποιήσεις επιτρέπουν στο Arduino να είναι έτοιμο για την λήψη και επεξεργασία δεδομένων από τους αισθητήρες και την ανταλλαγή πληροφοριών με άλλες συσκευές.

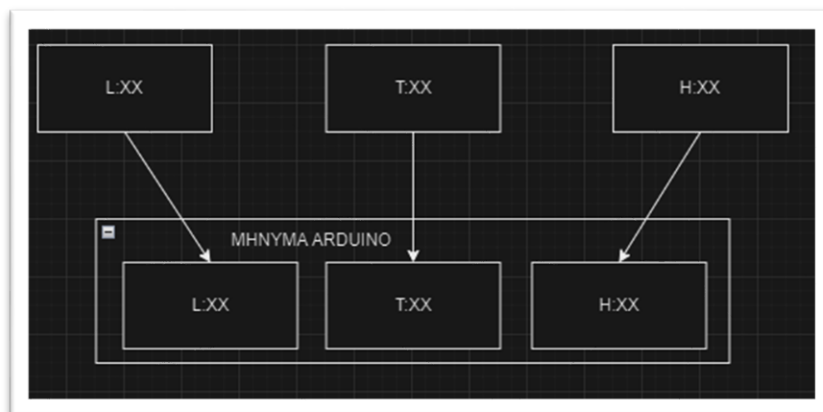
## **ΔΕΥΤΕΡΟ ΣΤΑΔΙΟ (LOOP)**

Στο δεύτερο κομμάτι της εφαρμογής, που λειτουργεί συνεχώς μετά την εκκίνηση και ο μικροελεγκτής εκτελεί επαναληπτικά ένα σύνολο εντολών, βρίσκεται η κύρια λειτουργία του Arduino. Εδώ, το Arduino αναλαμβάνει τη συλλογή περιβαλλοντικών μετρήσεων, συμπεριλαμβανομένης της θερμοκρασίας, της υγρασίας και της φωτεινότητας, μέσω αισθητήρων που συνδέονται με το Arduino.

Στην αρχή της συνάρτησης loop(), ο κώδικας διαβάζει την τιμή από έναν φωτοευαίσθητο αισθητήρα μέσω της αναλογικής εισόδου του Arduino. Αυτή η τιμή αποθηκεύεται στη μεταβλητή photo\_sensor\_val και εκτυπώνεται στη σειριακή οθόνη για επιθεώρηση. Στη συνέχεια, η τιμή αυτή μετατρέπεται σε μια κλίμακα 0-255 για χρήση σε LED, και αποθηκεύεται στη μεταβλητή photo\_val. Η LCD οθόνη καθαρίζεται και εμφανίζει τις μετρήσεις του φωτοευαίσθητου αισθητήρα.

Στη συνέχεια, ο κώδικας ασχολείται με τον αισθητήρα DHT22 που μετρά την θερμοκρασία και την υγρασία. Ο κώδικας ελέγχει εάν η ανάγνωση της θερμοκρασίας ή της υγρασίας είναι έγκυρη, και αν ναι, εκτυπώνει όλες τις τιμές στην LCD οθόνη. Διαφορετικά, εμφανίζει ένα μήνυμα σφάλματος στην οθόνη και στη σειριακή οθόνη.

Έπειτα από την συλλογή των δεδομένων γίνεται η κατασκευή του μηνύματος που θα σταλθεί στο esp. Αρχικά, ελέγχεται η σύνδεση με το esp. Στην περίπτωση που δεν είναι διαθέσιμο, για επικοινωνία τότε το Arduino εμφανίζει ανάλογο μήνυμα στην σειριακή θύρα. Διαφορετικά, κάθε μήνυμα που δημιουργείται είναι ένα string που περιλαμβάνει τις μετρήσεις του φωτοαισθητήρα, την μετρημένη θερμοκρασία και υγρασία. Με το να αποστέλλεται ένα μήνυμα έναντι των τριών (ένα μήνυμα για κάθε μέτρηση δηλαδή) επιτυγχάνεται ο καλός συγχρονισμός στην επικοινωνία των συσκευών και η αποσυμφόρηση των μηνυμάτων. Τα μηνύματα στέλνονται σε μορφή συμβολοσειράς μέσω σειριακής επικοινωνίας στο ESP8266.



Σχήμα 4.2 : Δομή μηνυμάτων Arduino [10]

Μετά την αποστολή του μηνύματος, ο κώδικας ελέγχει αν υπάρχουν διαθέσιμα δεδομένα από τον ESP8266. Εάν υπάρχουν, τα δεδομένα διαβάζονται και αποθηκεύονται σε μια μεταβλητή `receivedData`. Ο κώδικας αναλύει τα δεδομένα αυτά, τα οποία περιέχουν εντολές για έλεγχο LED και άλλες ενέργειες. Ανάλογα με την εντολή που λαμβάνεται, ενεργοποιούνται ή απενεργοποιούνται τα LED, ή ρυθμίζεται η φωτεινότητα ενός LED βάσει των τιμών του φωτοευαίσθητου αισθητήρα. Εάν δεν υπάρχουν διαθέσιμα δεδομένα από τον ESP8266, ο κώδικας εμφανίζει ένα μήνυμα σφάλματος στη σειριακή οθόνη. Ο κύκλος επαναλαμβάνεται κάθε 2 δευτερόλεπτα, παρέχοντας συνεχή παρακολούθηση και έλεγχο του συστήματος.

Με αυτόν τον τρόπο, το Arduino επιτρέπει την αναγνώριση και τη συλλογή των περιβαλλοντικών μετρήσεων, καθώς και τη διαμετάλλαξή τους με άλλες συσκευές μέσω του

διαδικτύου. Αυτό επιτυγχάνεται με απλά και αποτελεσματικά μηνύματα, επιτρέποντας την αξιοποίηση των δεδομένων για εφαρμογές ελέγχου και παρακολούθησης σε πραγματικό χρόνο.

### 4.3 ESP

#### 4.3.1 ΒΙΒΛΙΟΘΗΚΗ ESP8266WiFi.h

Η βιβλιοθήκη ESP8266WiFi.h είναι μια κρίσιμη συνιστώσα για την ανάπτυξη εφαρμογών που βασίζονται στο ESP8266, καθώς επιτρέπει στον μικροελεγκτή να συνδεθεί σε ασύρματα δίκτυα WiFi και να διαχειριστεί αυτή τη σύνδεση. Η βιβλιοθήκη παρέχει μια σειρά από συναρτήσεις και μεθόδους που διευκολύνουν την ενσωμάτωση της λειτουργικότητας WiFi στις εφαρμογές IoT (Internet of Things) και άλλες διασυνδεδεμένες συσκευές.

Η ESP8266WiFi.h περιλαμβάνει μεθόδους για τη διαχείριση της σύνδεσης WiFi, όπως η σύνδεση σε ένα δίκτυο με SSID (Service Set Identifier) και κωδικό πρόσβασης, η αποσύνδεση από το δίκτυο, και η ανανέωση της σύνδεσης. Η βιβλιοθήκη υποστηρίζει επίσης τη δυνατότητα σύνδεσης σε δίκτυα με κρυπτογράφηση WPA, WPA2, και WEP, καθώς και ανοικτά δίκτυα.

Μια από τις βασικές λειτουργίες της βιβλιοθήκης είναι η δυνατότητα διαχείρισης της διεύθυνσης IP του ESP8266. Η βιβλιοθήκη επιτρέπει τη χρήση δυναμικών διευθύνσεων IP μέσω DHCP (Dynamic Host Configuration Protocol), αλλά παρέχει επίσης την επιλογή ρύθμισης στατικής διεύθυνσης IP, καθορίζοντας συγκεκριμένες παραμέτρους δικτύου, όπως η διεύθυνση IP, η μάσκα υποδικτύου και η πύλη (gateway).

Η βιβλιοθήκη ESP8266WiFi.h προσφέρει επίσης λειτουργίες για την παρακολούθηση και τη διαχείριση της σύνδεσης WiFi. Παρέχει μεθόδους για την ανάκτηση της διεύθυνσης IP της συσκευής, τον έλεγχο της ισχύος του σήματος (RSSI - Received Signal Strength Indicator), και την ανανέωση της σύνδεσης σε περίπτωση απώλειας σύνδεσης. Επιπλέον, υποστηρίζει τη δυνατότητα αναζήτησης διαθέσιμων δικτύων WiFi, επιστρέφοντας μια λίστα με τα SSID των δικτύων που βρίσκονται στην εμβέλεια του ESP8266.

Εκτός από τις βασικές λειτουργίες σύνδεσης, η βιβλιοθήκη ESP8266WiFi.h επιτρέπει τη δημιουργία και διαχείριση διακομιστών (servers) και πελατών (clients) TCP/IP. Αυτό επιτρέπει τη δημιουργία web servers, μέσω των οποίων το ESP8266 μπορεί να εξυπηρετεί αιτήματα HTTP, καθώς και τη σύνδεση σε απομακρυσμένους servers για την αποστολή και λήψη δεδομένων. Η βιβλιοθήκη υποστηρίζει επίσης την επικοινωνία UDP (User Datagram Protocol), επιτρέποντας την αποστολή και λήψη μικρών πακέτων δεδομένων χωρίς την ανάγκη δημιουργίας μόνιμων συνδέσεων.

Η διαχείριση της ασφάλειας είναι ένα άλλο σημαντικό χαρακτηριστικό της ESP8266WiFi.h. Η βιβλιοθήκη περιλαμβάνει μεθόδους για την κρυπτογράφηση των συνδέσεων και την προστασία των δεδομένων που μεταδίδονται μέσω του δικτύου. Αυτό περιλαμβάνει τη χρήση πρωτοκόλλων ασφαλείας όπως το WPA/WPA2 για την εξασφάλιση της αυθεντικότητας και της εμπιστευτικότητας των συνδέσεων.

Συνολικά, η βιβλιοθήκη ESP8266WiFi.h είναι ένα ισχυρό εργαλείο που προσφέρει εκτεταμένη λειτουργικότητα για τη διαχείριση συνδέσεων WiFi και την επικοινωνία μέσω του διαδικτύου. Οι δυνατότητές της επιτρέπουν στους προγραμματιστές να δημιουργήσουν ευέλικτες και ασφαλείς εφαρμογές IoT, αξιοποιώντας τη συνδεσιμότητα του ESP8266 για την παρακολούθηση και τον έλεγχο απομακρυσμένων συσκευών.

#### **4.3.2 BIBΛΙΟΘΗΚΗ WebSocketsServer.h**

Η βιβλιοθήκη `WebSocketsServer.h` είναι ένα πολύτιμο εργαλείο για προγραμματιστές που επιθυμούν να υλοποιήσουν διακομιστές WebSocket χρησιμοποιώντας το ESP8266. Το πρωτόκολλο WebSocket προσφέρει ένα διάλογο επικοινωνίας σε πραγματικό χρόνο, επιτρέποντας την αμφίδρομη επικοινωνία μεταξύ πελάτη και διακομιστή σε μία συνεχή σύνδεση. Αυτό είναι ιδιαίτερα χρήσιμο για εφαρμογές που απαιτούν άμεση ανταλλαγή δεδομένων, όπως εφαρμογές IoT, όπου η άμεση ενημέρωση και ανταπόκριση είναι κρίσιμη.

Με τη βιβλιοθήκη `WebSocketsServer.h`, ο προγραμματιστής μπορεί να δημιουργήσει εύκολα έναν διακομιστή WebSocket στον ESP8266. Η βιβλιοθήκη παρέχει τις απαραίτητες λειτουργίες για να ξεκινήσει ο διακομιστής, να ακούει για εισερχόμενες συνδέσεις από πελάτες και να διαχειρίζεται τη διαδραστική επικοινωνία μεταξύ του διακομιστή και των πελατών. Αυτό σημαίνει ότι ο ESP8266 μπορεί να λάβει δεδομένα από συνδεδεμένους πελάτες, να επεξεργαστεί τα δεδομένα αυτά και να στείλει απαντήσεις σε πραγματικό χρόνο. Η συνεχή αυτή σύνδεση εξαλείφει την ανάγκη για επανειλημμένες αιτήσεις HTTP, κάνοντας την επικοινωνία πιο αποδοτική και γρήγορη.

Η βιβλιοθήκη `WebSocketsServer.h` απλοποιεί την ανάπτυξη εφαρμογών που απαιτούν WebSocket διακομιστή. Οι προγραμματιστές μπορούν να διαμορφώσουν και να χειριστούν τη συμπεριφορά του διακομιστή, όπως την αποδοχή συνδέσεων, την αποστολή και λήψη μηνυμάτων, και τη διαχείριση αποσυνδέσεων. Επιπλέον, η βιβλιοθήκη υποστηρίζει λειτουργίες όπως η μετάδοση μηνυμάτων σε όλους τους συνδεδεμένους πελάτες, προσφέροντας μεγάλη ευελιξία στην ανάπτυξη δικτυακών εφαρμογών.

Η χρήση της βιβλιοθήκης `WebSocketsServer.h` προσφέρει πολλά πλεονεκτήματα για την ανάπτυξη εφαρμογών IoT και άλλων εφαρμογών που απαιτούν δυναμική και διαδραστική επικοινωνία σε πραγματικό χρόνο. Ο διακομιστής WebSocket μπορεί να εξυπηρετήσει πολλαπλούς πελάτες ταυτόχρονα, διασφαλίζοντας ότι οι πληροφορίες μεταφέρονται αμέσως και αποτελεσματικά



μεταξύ των συνδεδεμένων συσκευών. Επιπλέον, η βιβλιοθήκη υποστηρίζει την κρυπτογράφηση της επικοινωνίας, προσφέροντας ένα επίπεδο ασφάλειας στις μεταδιδόμενες πληροφορίες.

Συνολικά, η βιβλιοθήκη `WebSocketsServer.h` είναι ένα αναπόσπαστο εργαλείο για τους προγραμματιστές που θέλουν να εκμεταλλευτούν την τεχνολογία WebSockets στο ESP8266, προσφέροντας απλοποιημένη ανάπτυξη και ισχυρές δυνατότητες για τη δημιουργία εφαρμογών επικοινωνίας σε πραγματικό χρόνο.

### 4.3.3 ΛΕΙΤΟΥΡΓΙΑ ESP

Η πλακέτα NodeMCU με μικροελεγκτή ESP8266 είναι βασισμένη στην αρχιτεκτονική του Arduino. Συνεπώς, η λειτουργία του ESP8266 χωρίζεται σε δύο βασικά μέρη: την αρχικοποίηση (setup) και τη βασική λειτουργία (loop).

#### ΠΡΩΤΟ ΣΤΑΔΙΟ (SETUP)

Το πρώτο στάδιο λειτουργίας του NodeMCU, βασισμένου στον μικροελεγκτή ESP8266, είναι η αρχικοποίηση του περιβάλλοντος λειτουργίας. Κατά αυτό το στάδιο, αρχικά, οι σειριακές επικοινωνίες ξεκινούν με την εντολή `Serial.begin(9600)` και `arduinoSerial.begin(9600)`, θέτοντας το baud rate στα 9600 bps για την επικοινωνία μεταξύ των συσκευών. Αυτό επιτρέπει την αποστολή και λήψη δεδομένων μεταξύ του Arduino και άλλων συσκευών, όπως ο υπολογιστής ή άλλες μονάδες.

Στη συνέχεια, ορίζονται οι παράμετροι σύνδεσης στο δίκτυο WiFi, όπως το όνομα δικτύου (SSID) και ο κωδικός πρόσβασης (password). Αυτή η διαδικασία είναι απαραίτητη για τη σύνδεση του ESP8266 στο ασύρματο δίκτυο. Η σύνδεση γίνεται με την εντολή `WiFi.begin(ssid, password)`, η οποία ξεκινά τη διαδικασία σύνδεσης στο καθορισμένο δίκτυο. Το ESP8266 προσπαθεί να συνδεθεί στο δίκτυο χρησιμοποιώντας τα παραπάνω στοιχεία. Η σύνδεση στο WiFi δίκτυο πραγματοποιείται μέσω της συνάρτησης `WiFi.begin(ssid, password)`. Εμφανίζονται μηνύματα στη σειριακή θύρα που ενημερώνουν το χρήστη για τη διαδικασία σύνδεσης. Η σύνδεση παρακολουθείται με τη βοήθεια του βρόχου `while (WiFi.status() != WL_CONNECTED)`, ο οποίος περιμένει έως ότου η συσκευή συνδεθεί επιτυχώς στο δίκτυο. Κατά τη διάρκεια αυτής της αναμονής, εκτυπώνεται το μήνυμα "Connecting to WiFi..." κάθε 1000 ms (1 δευτερόλεπτο), προκειμένου να παρέχει μια ένδειξη ότι η συσκευή προσπαθεί να συνδεθεί.

Όταν η συσκευή συνδεθεί επιτυχώς στο WiFi, εκτυπώνονται στην κονσόλα διάφορες πληροφορίες δικτύου όπως η διεύθυνση IP (`WiFi.localIP()`) και η μάσκα υποδικτύου (`WiFi.subnetMask()`). Αυτές οι πληροφορίες είναι χρήσιμες για την παρακολούθηση και τη

διαχείριση του δικτύου, καθώς και για τη διευκόλυνση της επικοινωνίας με άλλες συσκευές στο ίδιο δίκτυο.

Τέλος, η λειτουργικότητα του WebSocket ενεργοποιείται με την εντολή `websocket.begin()`, ενώ η συνάρτηση `websocket.onEvent(websocketEvent)` καθορίζει μια λειτουργία χειρισμού γεγονότων που θα εκτελείται όταν συμβαίνουν συμβάντα στο WebSocket, όπως η λήψη δεδομένων ή η διακοπή της σύνδεσης. Αυτή η διαδικασία επιτρέπει την ασύγχρονη επικοινωνία μεταξύ του ESP8266 και άλλων συσκευών ή διακομιστών μέσω του πρωτοκόλλου WebSocket, παρέχοντας τη δυνατότητα αποστολής και λήψης δεδομένων σε πραγματικό χρόνο.

### **ΔΕΥΤΕΡΟ ΣΤΑΔΙΟ (LOOP)**

Στο δεύτερο κομμάτι της εφαρμογής, δηλαδή στη συνάρτηση `loop()`, εκτελούνται επαναλαμβανόμενες λειτουργίες που αποτελούν τον κύριο πυρήνα της εφαρμογής. Αυτές περιλαμβάνουν την επικοινωνία μέσω WebSocket, τη λήψη δεδομένων από το Arduino, την αποστολή δεδομένων μέσω WebSocket, και την αποστολή δεδομένων πίσω στο Arduino.

Η συνάρτηση `websocketEvent` είναι υπεύθυνη για τη διαχείριση των γεγονότων που σχετίζονται με το WebSocket. Καθορίζει μια σειρά από ενέργειες που εκτελούνται όταν συμβαίνουν διαφορετικά γεγονότα, όπως αποσύνδεση (`WStype_DISCONNECTED`), σύνδεση (`WStype_CONNECTED`), και λήψη μηνύματος κειμένου (`WStype_TEXT`). Κατά την αποσύνδεση, το σύστημα καταγράφει το συμβάν και εμφανίζει ένα μήνυμα αποσύνδεσης. Κατά τη σύνδεση, καταγράφεται η διεύθυνση IP της συνδεδεμένης συσκευής και η διεύθυνση URL. Όταν λαμβάνεται ένα μήνυμα κειμένου, το μήνυμα αποθηκεύεται στη μεταβλητή `receivedString` και μπορεί να χρησιμοποιηθεί για περαιτέρω επεξεργασία.

Η βασική λειτουργία του προγράμματος βρίσκεται στη συνάρτηση `loop`, η οποία επαναλαμβάνεται συνεχώς. Αρχικά, καλείται η `websocket.loop()`, η οποία ελέγχει για νέα γεγονότα WebSocket και εκτελεί τις κατάλληλες ενέργειες που έχουν καθοριστεί στη συνάρτηση `websocketEvent`. Η εντολή `if (arduinoSerial.available() > 0)` ελέγχει αν υπάρχουν δεδομένα διαθέσιμα στη σειριακή θύρα που επικοινωνεί με το Arduino. Η εντολή `String dataFromArduino = arduinoSerial.readStringUntil('\n');` διαβάζει τη σειρά δεδομένων από το Arduino μέχρι να φτάσει σε χαρακτήρα νέας γραμμής (`\n`), αποθηκεύοντας την σε μια μεταβλητή τύπου `String`. Τα δεδομένα αυτά εκτυπώνονται στην κονσόλα για παρακολούθηση.

Αν τα δεδομένα είναι διαθέσιμα, στέλνονται σε όλους τους συνδεδεμένους WebSocket πελάτες χρησιμοποιώντας την εντολή `websocket.broadcastTXT(data.c_str())`. Αυτό επιτρέπει την αποστολή των δεδομένων από το Arduino σε άλλες συσκευές που είναι συνδεδεμένες μέσω

WebSocket. Η εντολή `if (arduinoSerial.available() > 0)` ελέγχει αν υπάρχουν δεδομένα διαθέσιμα στη σειριακή θύρα που επικοινωνεί με το Arduino. Η εντολή `String dataFromArduino = arduinoSerial.readStringUntil('\n');` διαβάζει τη σειρά δεδομένων από το Arduino μέχρι να φτάσει σε χαρακτήρα νέας γραμμής (`\n`), αποθηκεύοντας την σε μια μεταβλητή τύπου `String`. Επιπλέον, αν το μήνυμα που ελήφθη μέσω `WebSocket` είναι ένα κενό διάστημα (`spaceString`), εκτυπώνεται ένα μήνυμα "NO CHANGES MADE" τόσο στη σειριακή κονσόλα όσο και στο Arduino. Σε διαφορετική περίπτωση, το μήνυμα που ελήφθη από το `WebSocket` στέλνεται πίσω στο Arduino για περαιτέρω επεξεργασία.

Αν δεν υπάρχουν δεδομένα διαθέσιμα από το Arduino, η κονσόλα εκτυπώνει "No data available from Arduino.", ενημερώνοντας ότι δεν υπήρξε εισερχόμενη επικοινωνία. Ο βρόχος ολοκληρώνεται με μια καθυστέρηση 1000 ms (`delay(1000);`), παρέχοντας χρόνο στο σύστημα να αναμένει για νέα δεδομένα πριν επαναλάβει την επεξεργασία.

Αυτές οι διαδικασίες εξασφαλίζουν ότι το ESP8266 μπορεί να επικοινωνεί αποτελεσματικά τόσο με το Arduino όσο και με άλλες συσκευές μέσω του δικτύου, παρέχοντας μια ολοκληρωμένη λύση για την αυτοματοποίηση και τον έλεγχο διαφόρων συσκευών και αισθητήρων σε πραγματικό χρόνο.

## 4.4 SERVER

### 4.4.1 BIBΛΙΟΘΗΚΗ `threading`

Η βιβλιοθήκη `threading` στην Python χρησιμοποιείται για τη δημιουργία και τη διαχείριση νημάτων (`threads`), επιτρέποντας την εκτέλεση πολλαπλών εργασιών ταυτόχρονα. Αυτό μπορεί να βελτιώσει την απόδοση των εφαρμογών, ειδικά σε περιπτώσεις που περιλαμβάνουν αναμονή για εξωτερικούς πόρους, όπως εισόδους/εξόδους (I/O) από δίκτυα ή αρχεία.

Η βιβλιοθήκη `threading` παρέχει διάφορα εργαλεία και κλάσεις για τη δημιουργία και τον έλεγχο νημάτων. Η κλάση `Thread` είναι κεντρική στη βιβλιοθήκη και χρησιμοποιείται για τη δημιουργία νέων νημάτων. Οι χρήστες μπορούν να δημιουργήσουν ένα νέο νήμα υποκαθιστώντας τη μέθοδο `run` με μια συνάρτηση που επιθυμούν να εκτελεστεί στο νήμα.

Εκτός από τη δημιουργία νημάτων, η βιβλιοθήκη `threading` προσφέρει εργαλεία για τον συγχρονισμό των νημάτων, όπως κλειδαριές (`locks`), συμβάντα (`events`), και σημαφόρους (`semaphores`). Αυτά τα εργαλεία βοηθούν στη διαχείριση της πρόσβασης σε κοινόχρηστους πόρους, διασφαλίζοντας ότι οι πόροι χρησιμοποιούνται με ασφαλή και συντονισμένο τρόπο.

Η χρήση της βιβλιοθήκης `threading` είναι ιδιαίτερα επωφελής για εφαρμογές που είναι I/O-bound, δηλαδή περιλαμβάνουν πολλές λειτουργίες εισόδου/εξόδου, καθώς επιτρέπει την εκτέλεση άλλων εργασιών ενώ αναμένεται η ολοκλήρωση των λειτουργιών I/O. Για εφαρμογές που εκτελούν πολλές ανεξάρτητες εργασίες ταυτόχρονα, η πολυνηματική προσέγγιση μπορεί να οδηγήσει σε πιο αποδοτικό και ανταποκρινόμενο κώδικα.

Κατά τη χρήση της βιβλιοθήκης `threading`, οι προγραμματιστές μπορούν να επωφεληθούν από την ταυτόχρονη εκτέλεση πολλών διεργασιών, βελτιώνοντας την απόδοση και την αποδοτικότητα των εφαρμογών τους. Η βιβλιοθήκη παρέχει τα απαραίτητα εργαλεία για τη διαχείριση και τον συγχρονισμό των νημάτων, καθιστώντας την ανάπτυξη πολυνηματικών εφαρμογών στην Python μια απλή και αποτελεσματική διαδικασία.

#### 4.4.2 BIBΛΙΟΘΗΚΗ `asyncio`

Η βιβλιοθήκη `asyncio` στην Python χρησιμοποιείται για την υλοποίηση ταυτόχρονου προγραμματισμού μέσω της ασύγχρονης εκτέλεσης. Αυτή η βιβλιοθήκη επιτρέπει την εκτέλεση πολλαπλών λειτουργιών εισόδου/εξόδου (I/O) ταυτόχρονα χωρίς να μπλοκάρει η μία την άλλη, κάτι που είναι ιδιαίτερα χρήσιμο για εφαρμογές δικτύου, διακομιστές web, και άλλες εφαρμογές που βασίζονται σε εισόδους/εξόδους.

Η βιβλιοθήκη `asyncio` περιλαμβάνει έναν βρόχο εκδηλώσεων (event loop), ο οποίος διαχειρίζεται τη διαδοχική εκτέλεση των ασύγχρονων λειτουργιών. Οι ασύγχρονες λειτουργίες δηλώνονται με τη λέξη-κλειδί `async` και η εκτέλεσή τους προγραμματίζεται με τη χρήση της λέξης-κλειδί `await`. Αυτό επιτρέπει την αναμονή για την ολοκλήρωση μίας λειτουργίας χωρίς να σταματά η εκτέλεση του προγράμματος.

Η `asyncio` παρέχει επίσης εργαλεία για τη διαχείριση εργασιών (tasks), την επικοινωνία μέσω ουρών (queues), και τον συγχρονισμό μέσω κλειδαριών (locks) και άλλων συγχρονιστικών μηχανισμών. Αυτά τα εργαλεία επιτρέπουν την ανάπτυξη σύνθετων και αποδοτικών ασύγχρονων εφαρμογών με απλό και καθαρό κώδικα.

Η χρήση της βιβλιοθήκης `asyncio` είναι ιδιαίτερα επωφελής για εφαρμογές που περιλαμβάνουν πολλές λειτουργίες I/O, όπως δικτυακές εφαρμογές, όπου η αναμονή για δεδομένα από το δίκτυο μπορεί να εκμεταλλευτεί τον χρόνο για την εκτέλεση άλλων εργασιών. Αυτό καθιστά τις εφαρμογές πιο αποδοτικές και ανταποκρινόμενες, καθώς η αναμονή για πόρους δεν μπλοκάρει την εκτέλεση άλλων εργασιών.

Η βιβλιοθήκη `asyncio` παρέχει επίσης υποστήριξη για τη δημιουργία και τη διαχείριση πρωτοκόλλων δικτύου και μεταφορών (transports), επιτρέποντας την ανάπτυξη προσαρμοσμένων

λύσεων δικτύου. Αυτό προσφέρει μεγάλη ευελιξία και δυνατότητα παραμετροποίησης για την ανάπτυξη εφαρμογών που απαιτούν ασύγχρονη επικοινωνία.

Συνολικά, η βιβλιοθήκη `'asyncio'` στην Python επιτρέπει την αποδοτική διαχείριση ασύγχρονων λειτουργιών, βελτιώνοντας την απόδοση των εφαρμογών και καθιστώντας τον ταυτόχρονο προγραμματισμό πιο προσβάσιμο και ευέλικτο. Οι προγραμματιστές μπορούν να δημιουργούν εφαρμογές που εκμεταλλεύονται πλήρως τις δυνατότητες της ασύγχρονης εκτέλεσης, διατηρώντας τον κώδικα απλό και καθαρό.

#### 4.4.3 ΒΙΒΛΙΟΘΗΚΗ `websockets`

Η βιβλιοθήκη `'websockets'` στην Python χρησιμοποιείται για την υλοποίηση επικοινωνίας μέσω του πρωτοκόλλου WebSocket. Το πρωτόκολλο WebSocket επιτρέπει την αμφίδρομη επικοινωνία σε πραγματικό χρόνο μεταξύ ενός διακομιστή και ενός ή περισσότερων πελατών, καθιστώντας το ιδανικό για εφαρμογές όπως οι διαδικτυακές συνομιλίες, τα ζωντανά συστήματα ειδοποιήσεων, και τα διαδραστικά παιχνίδια.

Η βιβλιοθήκη `'websockets'` παρέχει έναν απλό και αποδοτικό τρόπο για τη δημιουργία διακομιστών και πελατών WebSocket. Χρησιμοποιεί τη λέξη-κλειδί `'async'` και `'await'` της Python για την υλοποίηση ασύγχρονων λειτουργιών, επιτρέποντας τη διαχείριση πολλαπλών συνδέσεων ταυτόχρονα χωρίς να μπλοκάρεται η εκτέλεση του προγράμματος.

Για να δημιουργηθεί ένας διακομιστής WebSocket, οι χρήστες μπορούν να χρησιμοποιήσουν τη συνάρτηση `'serve'`, η οποία δέχεται ως παραμέτρους τη συνάρτηση χειρισμού των μηνυμάτων και τη διεύθυνση IP και την πόρτα στην οποία θα ακούει ο διακομιστής. Αντίστοιχα, για τη δημιουργία ενός πελάτη WebSocket, χρησιμοποιείται η συνάρτηση `'connect'`, η οποία συνδέεται σε έναν διακομιστή WebSocket και επιτρέπει την αποστολή και λήψη μηνυμάτων.

Η βιβλιοθήκη `'websockets'` υποστηρίζει διάφορα χαρακτηριστικά, όπως την επικύρωση της σύνδεσης, την ασφαλή επικοινωνία μέσω WSS (WebSocket Secure), και την αυτόματη επανασύνδεση σε περίπτωση αποσύνδεσης. Παρέχει επίσης μηχανισμούς για την αντιμετώπιση σφαλμάτων και την διαχείριση του κύκλου ζωής των συνδέσεων.

Η χρήση της βιβλιοθήκης `'websockets'` είναι ιδιαίτερα επωφελής για εφαρμογές που απαιτούν χαμηλή καθυστέρηση και άμεση ανταπόκριση, καθώς το πρωτόκολλο WebSocket μειώνει την καθυστέρηση της επικοινωνίας σε σύγκριση με τα παραδοσιακά αιτήματα HTTP. Επιπλέον, η αμφίδρομη φύση του WebSocket επιτρέπει στον διακομιστή να στέλνει δεδομένα στους πελάτες αμέσως μόλις είναι διαθέσιμα, χωρίς να χρειάζεται οι πελάτες να κάνουν αιτήματα για νέα δεδομένα.

Συνολικά, η βιβλιοθήκη `websockets` στην Python διευκολύνει την ανάπτυξη εφαρμογών που απαιτούν πραγματικό χρόνο επικοινωνία, παρέχοντας ένα απλό και αποδοτικό API για τη δημιουργία διακομιστών και πελατών WebSocket. Οι προγραμματιστές μπορούν να επωφεληθούν από την ασύγχρονη φύση της βιβλιοθήκης για να διαχειρίζονται πολλαπλές συνδέσεις και να δημιουργούν αποδοτικές και ελέικτες εφαρμογές επικοινωνίας σε πραγματικό χρόνο.

#### 4.4.4 ΒΙΒΛΙΟΘΗΚΗ thingspeak

Η βιβλιοθήκη ThingSpeak στην Python χρησιμοποιείται για την αλληλεπίδραση με την πλατφόρμα IoT (Internet of Things) ThingSpeak. ThingSpeak είναι μια πλατφόρμα διασύνδεσης και ανάλυσης δεδομένων IoT που επιτρέπει στους χρήστες να στέλνουν, να αποθηκεύουν, να αναλύουν και να οπτικοποιούν δεδομένα από συσκευές IoT.

Η βιβλιοθήκη ThingSpeak για Python παρέχει μια εύχρηστη διεπαφή για την επικοινωνία με την πλατφόρμα ThingSpeak. Χρησιμοποιώντας αυτήν τη βιβλιοθήκη, δίνεται η δυνατότητα να στέλνονται δεδομένα από την εφαρμογή Python στην πλατφόρμα ThingSpeak για αποθήκευση, ανάλυση και οπτικοποίηση.



Εικόνα 4.11 : Ενδεικτικά διαγράμματα thingspeak με τιμές [11]

Ο τυπικός τρόπος χρήσης της βιβλιοθήκης ThingSpeak συνίσταται στη δημιουργία μιας σύνδεσης με την πλατφόρμα ThingSpeak χρησιμοποιώντας το ThingSpeak API και την αποστολή δεδομένων μέσω αιτημάτων HTTP ή MQTT.

Μερικά από τα βασικά χαρακτηριστικά της βιβλιοθήκης ThingSpeak στην Python περιλαμβάνουν:

1. Αποστολή δεδομένων προς αποθήκευση στην πλατφόρμα ThingSpeak.
2. Ανάγνωση δεδομένων από την πλατφόρμα ThingSpeak.
3. Δυνατότητα εκτέλεσης αναλύσεων και αποθήκευσης δεδομένων στο ThingSpeak.
4. Επικοινωνία με το ThingSpeak API για διαχείριση καναλιών και αιτημάτων.

Η χρήση της βιβλιοθήκης ThingSpeak είναι χρήσιμη για όσους θέλουν να δημιουργήσουν εφαρμογές IoT που να αποστέλλουν και να λαμβάνουν δεδομένα από την πλατφόρμα ThingSpeak. Επιπλέον, η δυνατότητα ανάλυσης και οπτικοποίησης των δεδομένων στο ThingSpeak κάνει τη βιβλιοθήκη αυτή αξιόλογο εργαλείο για την παρακολούθηση και την ανάλυση των δεδομένων αισθητήρων και άλλων συσκευών IoT.

## ΠΛΑΤΦΟΡΜΑ THINGSPEAK

Το ThingSpeak είναι μια πλατφόρμα ανάλυσης δεδομένων και IoT που επιτρέπει στους χρήστες να συλλέγουν, αποθηκεύουν, αναλύουν και οπτικοποιούν δεδομένα από αισθητήρες σε πραγματικό χρόνο. Δημιουργήθηκε από την MathWorks και χρησιμοποιείται ευρέως για έργα που απαιτούν την παρακολούθηση και ανάλυση δεδομένων από διάφορες συσκευές IoT.

Η πλατφόρμα υποστηρίζει τη σύνδεση και τη διαχείριση πολλαπλών συσκευών, επιτρέποντας την αποστολή δεδομένων μέσω πρωτοκόλλων όπως το HTTP και το MQTT. Κάθε χρήστης μπορεί να δημιουργήσει κανάλια στο ThingSpeak, όπου αποθηκεύονται τα δεδομένα από τις συσκευές τους. Τα δεδομένα μπορούν να οπτικοποιηθούν μέσω γραφημάτων και διαγραμμάτων που παρέχονται από την πλατφόρμα.

Το ThingSpeak περιλαμβάνει επίσης δυνατότητες επεξεργασίας δεδομένων, όπως μαθηματικές αναλύσεις, λειτουργίες φιλτραρίσματος και ενσωμάτωση με το MATLAB για προηγμένη ανάλυση και μοντελοποίηση. Οι χρήστες μπορούν να δημιουργήσουν triggers και να στέλνουν ειδοποιήσεις όταν τα δεδομένα υπερβαίνουν συγκεκριμένα όρια.

Η πλατφόρμα χρησιμοποιείται σε διάφορους τομείς, όπως η γεωργία, η περιβαλλοντική παρακολούθηση, η υγεία και οι έξυπνες πόλεις, για την ανάλυση και τη λήψη αποφάσεων με βάση τα δεδομένα που συλλέγονται από τους αισθητήρες.

#### 4.4.5 ΛΕΙΤΟΥΡΓΙΑ SERVER

Ο server σε αυτό το σύστημα έχει δύο κύριες λειτουργίες, οι οποίες διαχωρίζονται σε νήματα. Κάθε νήμα είναι υπεύθυνο για την επικοινωνία με το front end και το ESP8266, αντίστοιχα. Σε αυτή την εργασία, θα επικεντρωθούμε στη χρήση του νήματος που είναι υπεύθυνο για την επικοινωνία με το ESP8266.

Το νήμα που διαχειρίζεται τη WebSocket σύνδεση ξεκινά με την εκτέλεση της συνάρτησης `connect_to_websocket()`. Η συνάρτηση αυτή ορίζεται ως ασύγχρονη (`async`) και υλοποιεί έναν βρόχο που προσπαθεί συνεχώς να συνδεθεί στον WebSocket server. Ο WebSocket server υποτίθεται ότι τρέχει σε μια συγκεκριμένη διεύθυνση IP (στην προκειμένη περίπτωση, `ws://192.168.1.10:88`).

Μέσα στο βρόχο, η συνάρτηση προσπαθεί να συνδεθεί στον WebSocket server χρησιμοποιώντας τη συνάρτηση `websockets.connect(uri)`. Όταν η σύνδεση είναι επιτυχής, το πρόγραμμα τυπώνει μήνυμα "Connected to WebSocket Server" και αναθέτει το αντικείμενο του WebSocket στην παγκόσμια μεταβλητή `global_websocket`, η οποία χρησιμοποιείται για την αποστολή δεδομένων στον server.

Κατά τη διάρκεια της σύνδεσης, το νήμα ακούει συνεχώς για μηνύματα που αποστέλλονται από τον server χρησιμοποιώντας τη συνάρτηση `await websocket.recv()`. Όταν ληφθεί ένα μήνυμα, αυτό τυπώνεται και αναλύεται. Τα δεδομένα των αισθητήρων (όπως η θερμοκρασία, η υγρασία, και το φως) εξάγονται από το μήνυμα και ενημερώνουν τις αντίστοιχες μεταβλητές του συστήματος.

Μετά τον κατακερματισμό του μηνύματος του Arduino, τα περιβαλλοντικά δεδομένα στέλνονται στην πλατφόρμα ThingSpeak ως ένα HTTP update αίτημα. Η συνάρτηση `urllib.request.urlopen()` χρησιμοποιείται για να στείλει μια GET αίτηση με τα δεδομένα στους κατάλληλους καναλιούς του Thingspeak. Ειδικότερα, η γραμμή `urllib.request.urlopen('https://api.thingspeak.com/update?api_key=8NRKPGE1C2802J8O&field1='+ R_value + '&field2=' + H_value + '&field3=' + T_value)` στέλνει τα δεδομένα των αισθητήρων R (φωτεινότητα), H (υγρασία) και T (θερμοκρασία) στην πλατφόρμα. Ο κάθε αισθητήρας αντιστοιχεί σε ένα πεδίο (`field1`, `field2`, `field3`) του Thingspeak. Η αίτηση περιέχει το κλειδί API (`api_key`) που χρησιμοποιείται για να εξουσιοδοτήσει την αποστολή δεδομένων στο συγκεκριμένο κανάλι του Thingspeak. Στη συνέχεια, τα διαγράμματα αυτά εμφανίζονται στο front end με τη χρήση των ειδικών API της πλατφόρμας.

Με βάση τα δεδομένα που λήφθηκαν και τις τρέχουσες ρυθμίσεις του συστήματος (π.χ., αν είναι ενεργοποιημένο το air conditioner ή ο αφυγραντήρας), η συνάρτηση `process_sensor_data(sensor_type, sensor_value)` καλείται για να δημιουργήσει τις κατάλληλες εντολές για τις συσκευές. Οι εντολές αυτές στέλνονται πίσω στον server μέσω του WebSocket χρησιμοποιώντας τη συνάρτηση `await websocket.send(command)`.



Σε περίπτωση που η σύνδεση κλείσει απροσδόκητα (π.χ., λόγω προβλήματος δικτύου ή διακοπής του server), ο κώδικας πιάνει την εξαίρεση `websockets.exceptions.ConnectionClosedError`. Σε αυτή την περίπτωση, τυπώνεται μήνυμα που αναφέρει την αποσύνδεση και το νήμα επανεκκινεί τη διαδικασία σύνδεσης, προσπαθώντας να επανασυνδεθεί στον WebSocket server.

Η χρήση του `asyncio` για την εκτέλεση του νήματος WebSocket επιτρέπει στο πρόγραμμα να χειρίζεται ταυτόχρονα την επικοινωνία μέσω WebSocket και την επεξεργασία των δεδομένων, χωρίς να εμποδίζει άλλες λειτουργίες του συστήματος. Αυτό είναι σημαντικό για να εξασφαλιστεί η ομαλή λειτουργία του συστήματος, ακόμη και αν η επικοινωνία μέσω του WebSocket διαρκεί για μεγάλο χρονικό διάστημα ή αντιμετωπίζει καθυστερήσεις.

Συνολικά, το νήμα WebSocket είναι κρίσιμο για τη συνεχή ροή δεδομένων και την επικοινωνία μεταξύ του συστήματος και του εξωτερικού κόσμου, επιτρέποντας την ενημέρωση των τιμών των αισθητήρων και τον έλεγχο των συσκευών σε πραγματικό χρόνο.

## **ΚΕΦΑΛΑΙΟ 5: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ**

### **5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ**

Η παρούσα εργασία αναδεικνύει τις δυνατότητες και τις προοπτικές που προσφέρει η πλατφόρμα Arduino στην ανάπτυξη συστημάτων αυτοματισμού για έξυπνα σπίτια. Καταφέραμε να δημιουργήσουμε ένα αξιόπιστο και λειτουργικό σύστημα που επιτρέπει την παρακολούθηση και τον έλεγχο διαφόρων παραμέτρων του οικιακού περιβάλλοντος. Η χρήση αισθητήρων για τη μέτρηση περιβαλλοντικών δεδομένων, σε συνδυασμό με ενεργοποιητές για τον έλεγχο συσκευών, μας επέτρεψε να προσφέρουμε ένα ολοκληρωμένο σύστημα αυτοματισμού.

Ένα από τα κύρια πλεονεκτήματα του συστήματος είναι η φιλικότητα προς τον χρήστη, που επιτυγχάνεται μέσω της χρήσης μιας LCD οθόνης. Η οθόνη αυτή επιτρέπει την άμεση προβολή των δεδομένων και την εύκολη αλληλεπίδραση με το σύστημα. Επιπλέον, η ευελιξία του Arduino και η υποστήριξη πολλών βιβλιοθηκών και επεκτάσεων προσφέρει απεριόριστες δυνατότητες για μελλοντικές επεκτάσεις και βελτιώσεις.

Το σύστημα που αναπτύχθηκε είναι επεκτάσιμο και μπορεί να προσαρμοστεί στις ανάγκες του χρήστη, επιτρέποντας την ενσωμάτωση νέων αισθητήρων και ενεργοποιητών. Η πλατφόρμα Arduino, με την ανοιχτή αρχιτεκτονική της, προσφέρει ένα ιδανικό περιβάλλον για την ανάπτυξη καινοτόμων λύσεων στον τομέα του αυτοματισμού.

Συνολικά, η εφαρμογή μας αποτελεί ένα σημαντικό βήμα προς την κατεύθυνση της δημιουργίας έξυπνων σπιτιών που παρέχουν μεγαλύτερη άνεση, εξοικονόμηση ενέργειας και βελτίωση της ποιότητας ζωής. Με την υποστήριξη της κοινότητας και τη συνεχή ανάπτυξη νέων τεχνολογιών, οι δυνατότητες για περαιτέρω βελτιώσεις και εξελίξεις είναι απεριόριστες, κάνοντάς το σύστημα ακόμα πιο αποτελεσματικό και ευέλικτο.

### **5.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΞΕΛΙΞΕΙΣ**

Έως τώρα έχουμε αναπτύξει ένα αρκετά αξιόπιστο σύστημα. Ωστόσο, υπάρχουν κάποιες επαναλήψεις και ορισμένα σημεία που μειώνουν την ποιότητα της εφαρμογής.

Μία αρχική ιδέα για βελτίωση είναι η πλήρης αντικατάσταση του Arduino με το ESP. Με αυτόν τον τρόπο εξαλείφεται η ανάγκη για συγχρονισμό μεταξύ των μικροελεγκτών, μειώνοντας παράλληλα το κόστος υλοποίησης του έργου. Η προαναφερόμενη αλλαγή επιτυγχάνει επίσης τη μείωση της κατανάλωσης ηλεκτρικού ρεύματος, γεγονός που σημαίνει ότι εάν ο χρήστης επιλέξει να χρησιμοποιήσει τη συσκευή με μπαταρίες, η διάρκεια λειτουργίας της θα επεκταθεί.

Επιπλέον, με τη χρήση κάποιων επιπρόσθετων βιβλιοθηκών, θα μπορούσε ο ESP να λειτουργεί ως διακομιστής και να φιλοξενεί την ιστοσελίδα με τα δεδομένα, εξαλείφοντας έτσι την ανάγκη για χρήση ενός εξωτερικού διακομιστή και τις δυσκολίες στην επικοινωνία. Αυτό θα έχει ως αποτέλεσμα το σύστημα να είναι πιο φιλικό και εύχρηστο προς τον χρήστη, καθώς η εφαρμογή θα είναι πλέον πιο αυτόνομη και plug-and-play.

Βάσει των παραπάνω αλλαγών, ο χρήστης θα χρειάζεται μόνο να συνδέσει τη συσκευή σε κάποια πηγή ρεύματος και να την συνδέσει στο διαδίκτυο. Αυτό θα κάνει τη διαδικασία εγκατάστασης και χρήσης πιο απλή και αποτελεσματική, βελτιώνοντας συνολικά την εμπειρία χρήσης.

Τέλος, πέρα από τη συμπύκνωση και απλοποίηση του συστήματος, υπάρχει και η δυνατότητα επέκτασης. Με την προσθήκη νέων αισθητήρων και ενεργοποιητών, μπορούμε να αυξήσουμε τις δυνατότητες και να εξειδικεύσουμε περισσότερο το έργο. Για παράδειγμα, με την προσθήκη αισθητήρα ανίχνευσης καπνού και ενεργοποιητών νερού, η εφαρμογή μπορεί να μετατραπεί σε σύστημα πυρασφάλειας.

Μια άλλη ιδέα είναι η προσθήκη αισθητήρων κίνησης, καμερών και συναγερμών σε συνδυασμό με κάποιο μοντέλο τεχνητής νοημοσύνης για την επίτευξη ενός συστήματος ασφάλειας και παρακολούθησης. Αυτές οι επεκτάσεις όχι μόνο θα βελτιώσουν την απόδοση του συστήματος, αλλά θα επιτρέψουν επίσης τη δημιουργία εξατομικευμένων λύσεων για διαφορετικές ανάγκες, όπως η ασφάλεια του σπιτιού και η προστασία από κινδύνους.

Η ενσωμάτωση τεχνητής νοημοσύνης (AI) στο project προσφέρει σημαντικά πλεονεκτήματα. Η AI μπορεί να αναλύει δεδομένα από αισθητήρες σε πραγματικό χρόνο και να λαμβάνει έξυπνες αποφάσεις, βελτιώνοντας την απόδοση και την ακρίβεια του συστήματος. Για παράδειγμα, μπορεί να προβλέπει και να αποτρέπει κινδύνους, όπως πυρκαγιές ή εισβολές, αναγνωρίζοντας μοτίβα και ανωμαλίες στα δεδομένα. Επίσης, η AI μπορεί να αυτοματοποιεί διαδικασίες, μειώνοντας την ανάγκη για ανθρώπινη παρέμβαση, και να προσαρμόζεται στις συνθήκες του περιβάλλοντος, κάνοντας το σύστημα πιο αποδοτικό και αξιόπιστο.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://maker.pro/storage/I0PDRAK/I0PDRAKhQmwm6yr03C9UVNPb4i4hMPF0tbIZM6vX.jpg>
- [2] <https://i2.wp.com/www.teachmemicro.com/wp-content/uploads/2019/01/Arduino-UNO-pinout-1.jpg>
- [3] <https://avatars.mds.yandex.net/get-pdb/1394998/1be46f98-ab05-4184-b1dc-324a68fdde67/s1200?webp=false>
- [4] <https://electrocredible.com/wp-content/uploads/2022/06/DHT22-Pinout-1024x724.webp>
- [5] <https://projectiot123.com/wp-content/uploads/2019/10/What-is-Photodiode.png>
- [6] <https://soldered.com/productdata/2023/03/led-5mm-green-wikipedia.jpg>
- [7] [https://cdn.sparkfun.com/assets/learn\\_tutorials/9/0/5/Character\\_LCD\\_Pinout.png](https://cdn.sparkfun.com/assets/learn_tutorials/9/0/5/Character_LCD_Pinout.png)
- [8] [https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Fse4.mm.bing.net%2Fth%3Fid%3DOIP.2s\\_HdsFN\\_XxO7IvJb\\_ZYkgHaFE%26pid%3DApi&f=1&ipt=134a37cdbf2ee02f859f99cfdaf0dc3e8f241ed7f12678d2ec42a8aa327a5cb5&ipo=images](https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Fse4.mm.bing.net%2Fth%3Fid%3DOIP.2s_HdsFN_XxO7IvJb_ZYkgHaFE%26pid%3DApi&f=1&ipt=134a37cdbf2ee02f859f99cfdaf0dc3e8f241ed7f12678d2ec42a8aa327a5cb5&ipo=images)
- [9] <https://app.diagrams.net>
- [10] <https://wokwi.com>
- [11] <https://thingspeak.com>
- [12] Kushner, David (2011-10-26). "[The Making of Arduino](#)". *IEEE Spectrum*.
- [13] Barragán, Hernando (2016-01-01). "[The Untold History of Arduino](#)". *arduinohistory.github.io*. Retrieved 2016-03-06.
- [14] Lahart, Justin (27 November 2009). "[Taking an Open-Source Approach to Hardware](#)". *The Wall Street Journal*. Retrieved 2014-09-07.
- [15] "[Chip Hall of Fame: Atmel ATmega8](#)". *IEEE Spectrum: Technology, Engineering, and Science News*. Retrieved 2017-10-10.
- [16] "[Arduino - Products](#)". *www.arduino.cc*. Retrieved 20 February 2018.

- [17] ["Arduino - ArduinoBoardUnoSMD"](#). *www.arduino.cc*. Retrieved 20 February 2018.
- [18] Mpx. ["Lua cJSON is a fast JSON encoding/parsing module for Lua"](#). *Github*. Retrieved 2 April 2015.
- [19] ["NodeMCU - A Perfect Board for IoT"](#). *circuitio.io blog*. 2018-11-21. Retrieved 2021-05-27.
- [20] Hong. ["First commit of NodeMCU Firmware"](#). *Github*. Retrieved 2 April 2015.
- [21] Tuan PM. ["MQTT client library for ESP8266"](#). *Github*. Retrieved 2 April 2015.
- [22] Bennett, S. (1993). *A History of Control Engineering 1930–1955*. London: Peter Peregrinus Ltd. on behalf of the Institution of Electrical Engineers. [ISBN 978-0-86341-280-6](#)The source states "controls" rather than "sensors", so its applicability is assumed. Many units are derived from the basic measurements to which it refers, such as a liquid's level measured by a differential pressure sensor.
- [23] [https://www.waveshare.com/wiki/DHT22\\_Temperature-Humidity\\_Sensor](https://www.waveshare.com/wiki/DHT22_Temperature-Humidity_Sensor)
- [24] Tavernier, Filip and Steyaert, Michiel (2011) *High-Speed Optical Receivers with Integrated Photodiode in Nanoscale CMOS*. Springer. [ISBN 1-4419-9924-8](#). Chapter 3 *From Light to Electric Current – The Photodiode*
- [25] Fossum, Eric R.; Hondongwa, D. B. (2014). ["A Review of the Pinned Photodiode for CCD and CMOS Image Sensors"](#). *IEEE Journal of the Electron Devices Society*. **2** (3): 33–43. [doi:10.1109/JEDS.2014.2306412](#).
- [26] E. Aguilar Pelaez et al., "LED power reduction trade-offs for ambulatory pulse oximetry," 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon, 2007, pp. 2296–2299. doi: 10.1109/IEMBS.2007.4352784, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4352784&isnumber=4352185>
- [27] Escudier, Marcel; Atkins, Tony (2019). ["A Dictionary of Mechanical Engineering"](#). *Oxford Reference*. [doi:10.1093/acref/9780198832102.001.0001](#). [ISBN 978-0-19-883210-2](#).
- [28] Shabestari, N. P. (2019). "Fabrication of a simple and easy-to-make piezoelectric actuator and its use as phase shifter in digital speckle pattern interferometry". *Journal of Optics*. **48** (2): 272–282. [doi:10.1007/s12596-019-00522-4](#). [S2CID 155531221](#).
- [29] Edwards, Kimberly D. ["Light Emitting Diodes"](#) (PDF). [University of California, Irvine](#). p. 2. Archived from [the original](#) (PDF) on February 14, 2019. Retrieved January 12, 2019.
- [30] ["Solid-State Lighting: Comparing LEDs to Traditional Light Sources"](#). *eere.energy.gov*. Archived from [the original](#) on May 5, 2009.

- [31] Ulrich, Lawrence (2020). "[Bosch's smart visual visor tracks sun](#)". [Archived](#) March 1, 2021, at the [Wayback Machine IEEE Spectrum](#), January 29, 2020. Retrieved March 17, 2020.
- [32] "[Definition of LCD](#)". *Merriam-Webster.com*. [Archived](#) from the original on February 25, 2021. Retrieved February 15, 2015.
- [33] Tom's Hardware: Power Consumption Benchmark Results for CRT versus TFT LCD "[Benchmark Results: Different Brightness Testing](#)" [Archived](#) June 6, 2020, at the [Wayback Machine](#)
- [34] *The Authoritative Dictionary of IEEE Standards Terms (IEEE 100)* (seventh ed.). Piscataway, New Jersey: IEEE Press. 2000. [ISBN 0-7381-2601-2](#).
- [35] "[A Flaw in the Design](#)". *The Washington Post*. 30 May 2015. [Archived](#) from the original on 8 November 2020. Retrieved 20 February 2020. The Internet was born of a big idea: Messages could be chopped into chunks, sent through a network in a series of transmissions, then reassembled by destination computers quickly and efficiently. Historians credit seminal insights to Welsh scientist Donald W. Davies and American engineer Paul Baran. ... The most important institutional force ... was the Pentagon's Advanced Research Projects Agency (ARPA) ... as ARPA began work on a groundbreaking computer network, the agency recruited scientists affiliated with the nation's top universities.
- [36] Amogh Dhamdhere. "[Internet Traffic Characterization](#)". Retrieved 6 May 2022.
- [37] by Vinton Cerf, as told to Bernard Aboba (1993). "[How the Internet Came to Be](#)". Archived from [the original](#) on 26 September 2017. Retrieved 25 September 2017. We began doing concurrent implementations at Stanford, BBN, and University College London. So effort at developing the Internet protocols was international from the beginning.
- [38] Lee, J.A.N.; Rosin, Robert F (1992). "[Time-Sharing at MIT](#)". *IEEE Annals of the History of Computing*. **14** (1): 16. [doi:10.1109/85.145316](#). [S2CID 30976386](#). Retrieved 3 October 2022.
- [39] Gromov, Gregory (1995). "[Roads and Crossroads of Internet History](#)". Archived from [the original](#) on 27 January 2016.
- [40] "[The Difference Between the Internet and the World Wide Web](#)". *Webopedia*. QuinStreet Inc. 24 June 2010. [Archived](#) from the original on 2 May 2014. Retrieved 1 May 2014.
- [41] Brown, Ron (26 October 1972). "[Fax invades the mail market](#)". *New Scientist*. **56** (817): 218–221.

- [42] Fisher, Tim. ["How to Find Your Default Gateway IP Address"](#). *Lifewire*. [Archived](#) from the original on 25 February 2019. Retrieved 25 February 2019.
- [43] Andriole, Steve. ["Cyberwarfare Will Explode In 2020 \(Because It's Cheap, Easy And Effective\)"](#). *Forbes*. Retrieved 18 May 2021.
- [44] [DOD Standard Internet Protocol](#). DARPA, Information Sciences Institute. January 1980. [doi:10.17487/RFC0760](#). [RFC 760](#).
- [45] [The Economics of Transition to Internet Protocol version 6 \(IPv6\)](#) (Report). OECD Digital Economy Papers. OECD. 2014-11-06. [doi:10.1787/5jxt46d07bhc-en](#). [Archived](#) from the original on 2021-03-07. Retrieved 2020-12-04.
- [46] DeLong, Owen. ["Why does IP have versions? Why do I care?"](#) (PDF). *Scale15x*. Retrieved 24 January 2020.
- [47] [Braden, R.](#), ed. (October 1989). [Requirements for Internet Hosts – Communication Layers](#). [doi:10.17487/RFC1122](#). [RFC 1122](#).
- [48] Hohpe, Gregor. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. [ISBN 0-321-20068-3](#). p. 184
- [49] Hohpe, Gregor. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. [ISBN 0-321-20068-3](#). p. 184
- [50] Shapiro, Elmer B. (March 1969). [Network Timetable](#). IETF. [doi:10.17487/RFC0004](#). [RFC 4](#). Retrieved 30 November 2013.
- [51] ["WebSockets Standard"](#). *websockets.spec.whatwg.org*. [Archived](#) from the original on 2023-03-12. Retrieved 2022-05-16.
- [52] Ian Fette; Alexey Melnikov (December 2011). ["Relationship to TCP and HTTP"](#). [RFC 6455 The WebSocket Protocol](#). IETF. sec. 1.7. [doi:10.17487/RFC6455](#). [RFC 6455](#).
- [53] ["Adobe Flash Platform – Sockets"](#). *help.adobe.com*. [Archived](#) from the original on 2021-04-18. Retrieved 2021-07-28. TCP connections require a "client" and a "server". Flash Player can create client sockets.
- [54] Gillis, Alexander (2021). ["What is internet of things \(IoT\)?"](#). *IOT Agenda*. Retrieved 17 August 2021.
- [55] [Howard, Philip N.](#) (2015). *Pax Technica: How the internet of things May Set Us Free, Or Lock Us Up*. New Haven, CT: Yale University Press. [ISBN 978-0-30019-947-5](#).

- [56] McEwan, Adrian (2014). "[Designing the Internet of Things](#)" (PDF). Retrieved 1 June 2016.
- [57] Vongsingthong, S.; Smanchat, S. (2014). "[Internet of Things: A review of applications & technologies](#)" (PDF). *Suranaree Journal of Science and Technology*.
- [58][https://eclass.uniwa.gr/modules/document/file.php/ICE303/%CE%98%CE%B5%CF%89%CF%81%CE%AF%CE%B1/lec\\_01.pdf](https://eclass.uniwa.gr/modules/document/file.php/ICE303/%CE%98%CE%B5%CF%89%CF%81%CE%AF%CE%B1/lec_01.pdf)
- [59] Gubbi et al. (2013) <https://arxiv.org/pdf/1207.0203> Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions
- [60] Al-Fuqaha et al. (2015) <https://ieeexplore.ieee.org/ielx7/9739/7331734/07123563.pdf> Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications
- [61] Bandyopadhyay και Sen (2011) <https://arxiv.org/pdf/1105.1693> Internet of Things - Applications and Challenges in Technology and Standardization
- [62] Λαζάρου Αναστάσιος  
[https://polynoe.lib.uniwa.gr/xmlui/bitstream/handle/11400/6467/ΛΑΖΑΡΟΥ\\_ΑΝΑΣΤΑΣΙΟΣ-AM711-141290.pdf?sequence=1&isAllowed=y](https://polynoe.lib.uniwa.gr/xmlui/bitstream/handle/11400/6467/ΛΑΖΑΡΟΥ_ΑΝΑΣΤΑΣΙΟΣ-AM711-141290.pdf?sequence=1&isAllowed=y) Έξυπνες εφαρμογές γεωργίας ακριβείας στο πλαίσιο του Διαδικτύου των Αντικειμένων