

**ΑΛΕΞΑΝΔΡΕΙΑ ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ**

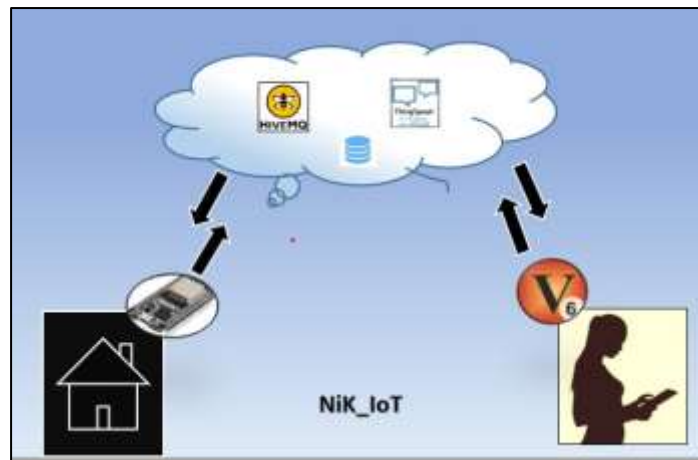
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ**

**ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ.**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

*Καθολικός Έλεγχος Ηλεκτρικού Φορτίου και Διαχείριση  
Ενέργειας μέσω Συστήματος IoT*



Του φοιτητή :

Νικολαΐδης Ν Δημήτριος  
Αρ.Μητρώου:2019/241

Επιβλέπων :

Δρ. Γιακουμής Αγγελος  
Βαθμίδα: Επίκουρος Καθηγητής

Τίτλος Δ.Ε. ‘Καθολικός Έλεγχος Ηλεκτρικού Φορτίου και Διαχείριση Ενέργειας μέσω Συστήματος IoT ’

Κωδικός Δ.Ε. 23206

Όνοματεπώνυμο φοιτητή:Νικολαϊδης Ν Δημητριος

Όνοματεπώνυμο εισηγητή :Dr Γιακουμής Άγγελος

Ημερομηνία ανάληψης :Δ.Ε 31-Μαρτιου -2023

Ημερομηνία περάτωσης : Δ.Ε 25-Μαιου 2023

Θεσσαλονίκη 2023

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

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

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

## Πίνακας Περιεχομένων

---

|   |    |
|---|----|
| Πίνακας Περιεχομένων .....  | 3  |
| Κατάλογος Διαγραμμάτων .....  | 6  |
| Πρόλογος.....   | 9  |
| Περίληψη .....  | 10 |
| Abstract .....  | 11 |
| 1 Εισαγωγή .....  | 12 |
| 1.1 Παρόμοιες μελέτες περίπτωσης. ....                                    | 13 |
| 1.2 Internet of Things (IoT).....   | 14 |
| 1.3 Το Έξυπνο σπίτι. ....   | 15 |
| 1.4 The Thing Speak Server. ....  | 16 |
| 1.5 The HiveMQ Cloud Server. ....   | 18 |
| 1.6 Wemos LoLIN 32 by Espressif.....                                      | 20 |
| 1.7 Η εφαρμογή Android Virtuino.....                                      | 22 |
| 1.7.1 Εισαγωγή στην εφαρμογή Virtuino.....                                | 22 |
| 1.7.2 Προγραμματισμός στο γραφικό περιβάλλον της εφαρμογής Virtuino. .... | 23 |
| 2 Το Σύστημα NiK_IoT και η αρχιτεκτονική του.....                         | 26 |
| 2.1 Εισαγωγή στο σύστημα NiK_IoT.....                                     | 26 |
| 2.2 The Data flow diagram of NiK_IoT.....                                 | 29 |
| 3 The NiK_IoT API .....   | 32 |
| 3.1 Εισαγωγή-Γενική περιγραφή της εφαρμογής API NiK_IoT. ....             | 32 |
| 3.2 Τα ‘panels’ των Servers.....  | 33 |
| 3.3 Η Εκκίνηση της Εφαρμογής.....   | 34 |
| 3.4 The Initial Panel Menu.....   | 34 |
| 3.5 The Control Panel .....   | 35 |
| 3.6 Load Current Consumption Panel .....                                  | 36 |
| 3.7 Indoor Temperature Panel .....  | 37 |
| 3.8 Wemos_Esp32Clients Connection Status Panel.....                       | 38 |
| 3.9 Alarms Panel Menu .....   | 40 |
| 3.10 Thermostat Panel .....   | 41 |
| 3.11 Energy Data Logger Panel.....  | 43 |
| 3.11.1 Επίδειξη μετρήσεων στο Energy Data logger Panel .....              | 45 |
| 3.12 Setting Hyperparameter Panel .....                                   | 48 |
| 3.13 Load status Panel.....   | 50 |
| 3.14 Programmable Timer Panel.....  | 52 |

|       |   |     |
|-------|---|-----|
| 3.15  | Outdoor Temperature Panel .....   | 52  |
| 3.16  | Visualization Data Panel .....  | 53  |
| 3.17  | Scale Analog to Current Value ( Hidden panel ) .....  | 54  |
| 4     | The Cal_En API.....   | 57  |
| 4.1   | Εισαγωγή στην εφαρμογή Cal_En.....  | 57  |
| 4.2   | Παρουσιαση της εφαρμογης Cal_En.....  | 58  |
| 4.3   | React App on ThingSpeak Server.....   | 60  |
| 4.4   | MATLAB Analysis App over ThingSpeak Server.....   | 61  |
| 4.5   | Ταυτοποίηση Ανάκλησης δεδομένων .....   | 62  |
|       | Στην ενότητα αυτή θα κάνουμε παρουσιάσουμε μια επίδειξη της εφαρμογης ώστε να πιστοποιήσουμε την ορθότητα της λειτουργίας της ..... | 62  |
| 5     | Κατασκευαστικό Μέρος .....  | 63  |
| 5.1   | Εισαγωγή στο Hardware .....   | 63  |
| 5.2   | InDoor Unit - Γενική περιγραφή .....  | 63  |
| 5.2.1 | Flow chart of Arduino code Wemos_1.....   | 67  |
| 5.2.2 | Flow chart of Arduino code Wemos_Command.....   | 69  |
| 5.3   | Outdoor Unit . .....  | 70  |
| 5.3.1 | Γενική περιγραφή της Εξωτερικής μονάδας.....  | 70  |
|       | Στο .....   | 71  |
| 5.3.2 | Flow chart of Arduino code Wemos_OutDoor.....   | 72  |
| 5.4   | Relay Unit . .....  | 73  |
| 5.4.1 | Γενική περιγραφή της Relay Unit . .....   | 73  |
| 5.4.2 | Block Diagram του κυκλώματος.....   | 74  |
| 5.4.3 | Μετρήσεις και σύγκριση τιμών . .....  | 75  |
| 6     | Αξιολόγηση.....   | 80  |
| 6.1   | Εισαγωγή.....   | 80  |
| 6.2   | Μετρήσεις -Ανάλυση Αποτελεσμάτων.....   | 80  |
| 6.3   | Τροποποιήσεις –Επεκτάσεις –Βελτιώσεις.....  | 81  |
| 6.4   | Συμπεράσματα .....  | 82  |
| 7     | Βιβλιογραφία – Αναφορές.....  | 84  |
| 8     | Παράρτημα 1-Κωδικας Arduino.....  | 88  |
| 8.1   | Arduino Code of Wemos_1_INDOOR_MQTT .....   | 88  |
| 8.2   | Arduino Code of WEMOS_Outdoor _MQTT .....   | 96  |
| 8.3   | Arduido Code of WEMOS_COMMAND .....   | 99  |
| 9     | Παράρτημα 2 Κώδικας Virtuino .....  | 101 |
| 10    | Παράρτημα 3 Κώδικας MATLAB.....   | 107 |

|      |  |     |
|------|--|-----|
| 10.1 | Παράρτημα 3 Κώδικας MATLAB “ tsimpao” .....  | 107 |
| 10.2 | Παράρτημα 3.2 Κώδικας MATLAB Visualitation A histogram of outdoor<br>tempeprature .....                        | 111 |
| 10.3 | Παράρτημα 3.3 Κώδικας MATLAB Visualitation Consumption of three days<br>Current - Last week - Last month ..... | 112 |
| 11   | Παράρτημα 4 Φωτογραφιες της κατασκευης τυ Νικ_IoT .....  | 113 |

## Κατάλογος Διαγραμμάτων

|  |    |
|--|----|
| Εικόνα 1: Εννοιολογικός χάρτης του IoT.....  | 14 |
| Εικόνα 2 : Τοπολογία και διασυνδεδεμένα συστήματα ενός Έξυπνου σπιτιού (Smart Home).....   | 15 |
| Εικόνα 3: ThingSpeak μια πλατφόρμα ανάλυσης του Internet of Things (IoT) .....             | 16 |
| Εικόνα 4: Η διακίνηση των μηνυμάτων στα τρία κανάλια ανά ημέρα .....                       | 17 |
| Εικόνα 5:Στοιχεία του λογαριασμού στο ThingSpeak.....                                      | 17 |
| Εικόνα 6: Η πλατφόρμα HiveMQ αξιόπιστη και κλιμακούμενη διαχείριση IoT στο cloud. ....     | 18 |
| Εικόνα 7: Στατιστικά των συνδέσεων στο Cluster του HIVEMQ Cloud Server .....               | 19 |
| Εικόνα 8:Συνδεση Web Client στο HiveMQ Cluster .....                                       | 19 |
| Εικόνα 9:Τα topics όπως εμφανίζονται στον Web client του HiveMQ Cluster.....               | 19 |
| Εικόνα 10: Wemos Lolin 32 – Μικροελεγκτής βασισμένος στο ESP32 chip της Espressif .....    | 20 |
| Εικόνα 11: Προγραμματισμός στο περιβάλλον της εφαρμογής Android Virtuino.....              | 23 |
| Εικόνα 12 :Advance value display Widget over Virtuino API.....                             | 23 |
| Εικόνα 13:Script widget της εφαρμογής Virtuino .....                                       | 24 |
| Εικόνα 14: Εναλλαγή λειτουργίας του Virtuino μεταξύ Run και Programming Mode .....         | 24 |
| Εικόνα 15: Servers που υποστηρίζουν την εφαρμογή NiK_IoT.....                              | 33 |
| Εικόνα 16: IoT Servers settings over NiK_IoT API.....                                      | 33 |
| Εικόνα 17 MQTT Server settings over NiK_IoT API. ....                                      | 33 |
| Εικόνα 18 Εκκίνηση και άνοιγμα της Εφαρμογής NiK_IoT API. ....                             | 34 |
| Εικόνα 19:Initial Menu Of NiK_IoT API.....   | 34 |
| Εικόνα 20 :Public view του Monitor Channel over ThingSpeak. ....                           | 35 |
| Εικόνα 21: Load Current Consumption Panel Of NiK_IoT API .....                             | 36 |
| Εικόνα 22: Indoor Temperature Panel Over NiK-IoT API.....                                  | 37 |
| Εικόνα 23: Wemos_Esp32 Clients Connection Status Panel Of NiK_IoT API .....                | 38 |
| Εικόνα 24: Alarms Panel Of NiK_IoT API.....  | 40 |
| Εικόνα 25 :SMS , Twitter and Phones Calls setting panel .....                              | 40 |
| Εικόνα 26: Thermostat Panel Of NiK_IoT API.....  | 41 |
| Εικόνα 28: Energy Data Logger Panel over NiK_IoT API.....                                  | 43 |
| Εικόνα 29 : Ηλεκτρική κατανάλωση κατά την διάρκεια της Επίδειξης.....                      | 45 |
| Εικόνα 30 : Παράμετροι της τιμολογιακής ενεργειακής κατανάλωσης .....                      | 45 |
| Εικόνα 31 :Αρχικοποίηση της μέτρησης .....   | 45 |
| Εικόνα 32:Επίδειξη λειτουργίας Πρώτη κλίμακα Χρέωσης .....                                 | 46 |
| Εικόνα 33 Δεύτερη κλίμακα Χρέωσης .....  | 46 |
| Εικόνα 34 : Επίδειξη λειτουργίας Νυκτερινό τιμολόγιο χρέωσης. ....                         | 47 |
| Εικόνα 35:Χρεώσεις στο Γ1/Γ1N προϊόν ηλεκτρικής ενέργειας [57] .....                       | 49 |
| Εικόνα 36: Load status Panel Of NiK_IoT API.....   | 50 |
| Εικόνα 37: Η τιμή της μεταβλητής M45 στον χρόνο .....                                      | 51 |
| Εικόνα 38: Searching ms timer a Hidden panel Of NiK_IoT API .....                          | 51 |
| Εικόνα 39: Programmable Timer Panel Of NiK_IoT API .....                                   | 52 |
| Εικόνα 40 : Outdoor Temperature Panel Of NiK_IoT API .....                                 | 52 |
| Εικόνα 41 Visualization Data Panel Of NiK_IoT API .....                                    | 53 |
| Εικόνα 42 Hidden panel Scale Analog Value to Current .....                                 | 54 |
| Εικόνα 43:Ευρεση Εξίσωση ευθείας που διέρχεται από 2 σημεία.....                           | 56 |
| Εικόνα 44:Βήματα 1 , 2 ,3 και 4 για την ανάκληση Δεδομένων μέσω της εφαρμογής Cal_En ..... | 58 |
| Εικόνα 45:Τα βήματα 1 και 2 για την ανάκληση Δεδομένων μέσω της εφαρμογής Cal_En .....     | 59 |
| Εικόνα 46:Εφαρμογή React του ThingSpeak για την υλοποίηση του Cal_En.....                  | 60 |
| Εικόνα 47 :Επίδειξη της Λειτουργίας ανάκλησης Δεδομένων μέσω της εφαρμογής Cal_En.....     | 62 |
| Εικόνα 48: Τοπολογία εξαρτημάτων και συνδέσεων στο InDoor Unit.....                        | 63 |
| Εικόνα 49:Ψηφιακός αισθητήρας θερμοκρασίας DS18820. ....                                   | 64 |
| Εικόνα 50 Successive-approximation ADC.....  | 66 |

|  |           |
|--|-----------|
| Εικόνα 51 Outdoor Unit.....  | 70        |
| Εικόνα 52: I- rex coaxial RF connector σε μεγέθυνση.....   | 70        |
| Εικόνα 53 Τοπολογία εξαρτημάτων και συνδέσεων στο Relay Unit.....  | 73        |
| Εικόνα 54 : Σύγκριση των αποτελεσμάτων μετρήσεων ρεύματος στην περιοχή 6 A.....                          | 76        |
| Εικόνα 55 :Σύγκριση των αποτελεσμάτων μετρήσεων στην περιοχή 14 A.....                                   | 76        |
| Εικόνα 56:Σύγκριση των αποτελεσμάτων μετρήσεων στην περιοχή των 20A.....                                 | 77        |
| Εικόνα 57:Εναλλακτικός τρόπος μέτρησης εναλλασσόμενου ρεύματος.....                                      | 77        |
| Εικόνα 58 Σύγκριση των αποτελεσμάτων μετρήσεων στην περιοχή κάτω των 4,5.....                            | 78        |
| Εικόνα 59:Γραφική παράσταση Χ.Υ των ενδείξεων του Nik_IoT vs Reference.....                              | 78        |
| <b>Εικόνα 60:Γραφική παράσταση Χ.Υ της τάσης εξόδου vs Reference.....</b>                                | <b>79</b> |
| Εικόνα 61 Σύνγκριση αποτελεσμάτων μέτρησης θερμοκρασίας.....   | 80        |
| <br>   |           |
| Block Διάγραμμα 1: Αρχιτεκτονική του συστήματος NiK_IoT.....   | 26        |
| Block Διάγραμμα 2 :Η ροή δεδομένων στο σύστημα NiK_IoT.....  | 29        |
| Block Διάγραμμα 3: NiK_IoT API Diagram.....  | 32        |
| Block Διάγραμμα 4 : Cal_En API DiagramAPI.....   | 57        |
| Block Διάγραμμα 5 :Διάγραμμα ροής της εφαρμογής React και της εφαρμογής Matlab Analysis 'TSIMPAO'.....   | 61        |
| Block Διάγραμμα 6: Διάγραμμα του Ηλεκτρονικού κυκλώματος της InDoor Unit.....                            | 66        |
| Block Διάγραμμα 7 : Flow chart of Arduino code Wemos_1.....  | 67        |
| Block Διάγραμμα 8 Wemos_ Command Arduino Code Flow Chart.....  | 69        |
| Block Διάγραμμα 9 : : Block Διάγραμμα του Ηλεκτρονικού κυκλώματος της OutDoor Unit.....                  | 71        |
| Block Διάγραμμα 10 Wemos_ Command Arduino Code Flow Chart.....   | 72        |
| Block Διάγραμμα 11 Block Διάγραμμα του Ηλεκτρονικού κυκλώματος της Relay Unit.....                       | 74        |
| <br>   |           |
| Πίνακας 1 :Τροποι λειτουργίας του μικροελεγκτή Esp32.....  | 20        |
| Πίνακας 2 Τα δεδομένα που αποστέλλονται από την μονάδα OutDoor Unit.....                                 | 30        |
| Πίνακας 3 Τα δεδομένα που αποστέλλονται Wemos_1.....   | 30        |
| Πίνακας 4:Τα δεδομένα που λαμβάνονται από την InDoor Unit Wemos_ Command.....                            | 30        |
| Πίνακας 5 Τα δεδομένα που αποστέλλονται από το Virtuino στην πλατφόρμα ανάλυσης ThingSpeak.....          | 30        |
| Πίνακας 6: Τα δεδομένα που αποστέλλονται από το Virtuino στον MQTT broker HiveMQ.....                    | 31        |
| Πίνακας 7:Τα δεδομένα που λαμβάνονται στο Virtuino από τον MQTT broker HiveMQ.....                       | 31        |
| Πίνακας 8: Συσκευατικά – συγκριτικά αποτελέσματα μετρήσεων ρεύματος.....                                 | 78        |
| <br>   |           |
| Figure 1: Setting Hyperparameter Panel over NiK-IoT API.....   | 48        |
| Figure 2:Προγραμματιστικό περιβάλλον δεξιά σε unlock mode.....   | 101       |
| Figure 3:Upload Widget για την δημιουργία Gateway.....   | 101       |
| Figure 4: Ενα μέρος από τον Κωδικά Widget στο Control Panel.....   | 102       |
| Figure 5: Gateway Operation in Outdoor temperature Field Using Upload Widget.....                        | 102       |
| Figure 6:Διαφορά panel και λειτουργίες της εφαρμογής Virtuino.....                                       | 103       |
| Figure 7:Κωδικας για την αλλαγή χρώματος στο frame του Outside Temperature στο panel του θερμοστάτη..... | 103       |
| Figure 8:Δείγμα κωδικα για την δημιουργία Λογικού κυκλώματος στο Control Panel;.....                     | 103       |
| Figure 9:Δήλωση μεταβλητών που τρέχουν στον emulator της εφαρμογής NiK_IoT.....                          | 104       |
| Figure 10: Δήλωση μεταβλητών που τρέχουν στον emulator της εφαρμογής NiK_IoT part 2.....                 | 104       |
| Figure 11: Δήλωση των topics στον HiveMQ Cloud Server.....   | 105       |
| Figure 12:Δήλωση μεταβλητών που τρέχουν στον emulator της εφαρμογής NiK_IoT part3.....                   | 105       |
| Figure 13:Ο Timer που τρεχει σε κάθε κλήση στην εφαρμογη Cal_En.....                                     | 106       |
| Figure 14: Στο εργαστήριο του τμήματος για έλεγχο και επίδειξη.....                                      | 113       |

|   |            |
|---|------------|
| <i>Figure 15:Video από τις δοκιμές φορτιού κάντε κλικ πάνω στην εικόνα .....</i>                      | <i>113</i> |
| <i>Figure 16 Scam στο Indoor Unit.....</i>  | <i>114</i> |
| <i>Figure 17:Κλεμοσειρες .....</i>  | <i>114</i> |
| <i>Figure 18: Εξοδος της Outdoor Unit με την ενδεικτικη λυχνια .....</i>                              | <i>115</i> |
| <i>Figure 19: Isolation Amplifier in Outdoor Unit.....</i>  | <i>115</i> |
| <i>Figure 20 :Η Outdoor Unit .....</i>  | <i>115</i> |
| <i>Figure 21:Κουμπωτή κλεμα για την σύνδεση του InDoor Unit με το OutDoor Unit Analog output.....</i> | <i>116</i> |
| <i>Figure 22:Η Εξωτερική κεραία της Outdoor Unit.....</i>   | <i>116</i> |
| <i>Figure 24 :Η Indoor με τους στηπιοθληπτες συνδεσης .....</i>                                       | <i>117</i> |
| <i>Figure 23:Εξοδος της Ν.Ο επαφής.....</i>   | <i>117</i> |

## Πρόλογος.

---

Ονομάζομαι Νικολαΐδης Δημήτριος και είμαι μόνιμος εκπαιδευτικός δευτεροβάθμιας εκπαίδευσης. Η πρώτη μου επαφή με τον κόσμο του IoT έγινε το 2018. Τη χρονιά εκείνη, στα πλαίσια της "Ερευνητικής Εργασίας στην Τεχνολογία" της πρώτης τάξης λυκείου στο 2ο ΕΠΑΛ Γιαννιτσών, κατασκευάσαμε έναν μετεωρολογικό σταθμό στην αυλή του σχολείου.

Όταν σχεδόν από καθαρή σύμπτωση βρέθηκα με την ιδιότητα του φοιτητή στο τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, ενώ δεν είχα ιδέα πότε θα περάτωνα τις σπουδές μου όντας γηραιός και εργαζόμενος ήξερα με σιγουριά το αντικείμενο της διπλωματικής μου. Η παρούσα εργασία έχει τίτλο "Καθολικός Έλεγχος Ηλεκτρικού Φορτίου και Διαχείριση Ενέργειας μέσω Συστήματος IoT" και είναι η συνέχεια του ταξιδιού στον μαγικό κόσμο του IoT.

Στόχος ήταν να δώσω μια πιο ολοκληρωμένη "κατασκευή" με επαγγελματικά χαρακτηριστικά και προοπτικές. Προσπάθησα να παρουσιάσω τις δυνατότητες του Virtuino, ένα πανίσχυρο εργαλείο API IoT Εφαρμογών, και να γράψω ένα πλήρες τεχνικό κείμενο χωρίς θεωρητικούς βερμπαλισμούς, ώστε ο αναγνώστης να το χρησιμοποιήσει ως βάση για άλλα Project.

Η εκπόνηση της εργασίας συνάντησε πολλές δυσκολίες και εμπόδια. Ήταν μια περίοδος δημιουργίας αλλά και απογοητεύσεων. Ασχέτως της επιτυχίας ή όχι του όλου εγχειρήματος, θεωρώ ότι με το πέρας αυτής, κατανόησα την ρήση «Ένας δρόμος, ένας μόνο οδηγεί στο Θεό, ο ανήφορος...» (Νίκος Καζαντζάκης - Αναφορά στον Γκρέκο).

## Περίληψη .

---

Αντικείμενο της παρούσης εργασίας είναι η διπλωματική εργασία που αναλύει την ανάπτυξη μιας εφαρμογής IoT καταγραφής ενέργειας, η οποία επιτρέπει τον έξυπνο έλεγχο της λειτουργίας ενός ηλεκτρικού φορτίου. Το NiK\_IoT επιτρέπει τον αξιόπιστο απομακρυσμένο έλεγχο με ταυτόχρονη καταγραφή λειτουργικών παραμέτρων για να βελτιστοποιηθεί η λειτουργία του φορτίου. Στόχος είναι ο έλεγχος της κατανάλωσης ενέργειας και του κόστους λειτουργίας. Η εφαρμογή περιλαμβάνει έλεγχο λειτουργίας, χρονοπρογραμματισμό, έλεγχο θερμοκρασίας με έξυπνο θερμοστάτη, καταγραφή και υπολογισμό κόστους ενέργειας σε πραγματικό χρόνο, πολλαπλούς χρήστες και μερική Οπτικοποίηση δεδομένων. Επίσης, παρέχει πλήρως προσαρμόσιμα alarms. Στην ενότητα αυτή παρουσιάζονται δύο εφαρμογές API οι οποίες στηρίζουν την εφαρμογή : το NiK\_IoT και το Cal\_En.

Κατά την τεχνική ανάλυση του συστήματος, δώσαμε έμφαση στη μέθοδο scaling και την μετατροπή τιμών και ενδείξεων. Παραθέτουμε με λεπτομέρεια διαγράμματα ροής για κάθε κώδικα που κατασκευάσαμε. Για την υλοποίηση του NiK\_IoT χρησιμοποιήθηκαν τρία είδη κώδικα: η "Arduino Code" ως εργαλείο για τον προγραμματισμό των μικροελεγκτών του Middleware, η "Matlab Analysis Code" για τον προγραμματισμό του Bank-End και το γραφικό περιβάλλον της εφαρμογής Android "Virtuino" για τον προγραμματισμό των API.

Η εφαρμογή Cal\_En ενσωματώνει και Front-End και Back-End προγραμματισμό και επιτρέπει στον χρήστη να καλέσει από τη βάση δεδομένων την ενέργεια που καταναλώθηκε σε ένα συγκεκριμένο χρονικό διάστημα. Η αρχιτεκτονική περιλαμβάνει τεχνολογία cloud computing και πιο συγκεκριμένα διάδραση με την πλατφόρμα ThingSpeak Server και HiveMQ. Το ThingSpeak είναι μια υπηρεσία IoT που παρέχει έναν απλό τρόπο για τη συλλογή, την ανάλυση και την απεικόνιση δεδομένων από συσκευές-πελάτες, ενώ από την άλλη η HiveMQ είναι μια πλατφόρμα μηνυμάτων MQTT (Message Queuing Telemetry Transport) για τον διαμοιρασμό δεδομένων στο Internet of Things (IoT). Το MQTT είναι ένα πρωτόκολλο ανοιχτού κώδικα που χρησιμοποιείται για τη μεταφορά δεδομένων σε συσκευές IoT. Το πάντρεμα των δύο αυτών διαφορετικών στη φιλοσοφία πλατφορμών ήταν μια πρόκληση. Ελπίζω να τα καταφέραμε..

Λέξεις Κλειδιά : *IoT, NiK\_IoT, Cal\_En, Virtuino, ThingSpeak, Matlab Analysis, Wemos LoLin32, Esp32, Arduino Code, MQTT, Remote device monitoring, Energy data logger,*

## Abstract.

---

The subject of this thesis is the development of an IoT energy monitoring application, which enables smart control of the operation of an electrical load. NiK\_IoT allows reliable remote control while simultaneously recording functional parameters to optimize the load operation. The goal is to control energy consumption and operating costs. The application includes operation control, scheduling, and temperature control with a smart thermostat, real-time energy cost recording and calculation, multiple users, and partial data visualization. Additionally, it provides fully customizable alarms. In this section, two API applications are presented that support the NiK\_IoT and Cal\_En application.

In the technical analysis of the system, we emphasized the scaling method and the conversion of values and indicators. We provide detailed flowcharts for each code we constructed. Three types of code were used for the implementation of NiK\_IoT: "Arduino Code" as a tool for programming the microcontrollers of the Middleware, "Matlab Analysis Code" for programming the Bank-End, and the graphical environment of the Android application "Virtuino" for programming the APIs.

The Cal\_En application integrates both Front-End and Back-End programming and allows the user to call the energy consumption action from the database in a specific time frame. The architecture includes cloud computing technology, specifically interacting with the ThingSpeak Server platform and HiveMQ. ThingSpeak is an IoT service that provides a simple way to collect, analyze, and visualize data from client devices, while HiveMQ is an MQTT (Message Queuing Telemetry Transport) messaging platform for sharing data on the Internet of Things (IoT). MQTT is an open-source protocol used for data transmission in IoT devices. The combination of these two different platforms was a challenge, but we hope we managed to accomplish it.

*Key Words :* IoT, NiK\_IoT, Cal\_En, Virtuino, ThingSpeak, Matlab Analysis, Wemos LoLin32, Esp32, Arduino Code, MQTT, Remote device monitoring, Energy data logger,

## 1 Εισαγωγή

Η τεχνολογία του Internet of Things (IoT) έχει επιφέρει τη δημιουργία νέων εφαρμογών και λύσεων στην καθημερινότητά μας. Μία από αυτές τις εφαρμογές είναι ο απομακρυσμένος έλεγχος λειτουργίας συστημάτων θέρμανσης, ο οποίος επιτρέπει στους χρήστες να ελέγχουν έξυπνα τη θερμοκρασία του σπιτιού τους από απόσταση. Το IoT αντιπροσωπεύει ένα παγκόσμιο δίκτυο μοναδικά διευθυνσιοδοτημένων συνδεδεμένων αντικειμένων. Το IoT είναι μια "συνδεδεμένη δικτύωση συσκευών ανίχνευσης και ενεργοποίησης που παρέχει τη δυνατότητα να μοιράζονται πληροφορίες σε διαδικτυακές πλατφόρμες μέσω ενός ενιαίου πλαισίου (Application).

Μια τέτοια εφαρμογή παρουσιάζεται στην παρούσα διπλωματική εργασία. Η διπλωματική εκπονήθηκε στα πλαίσια του μαθήματος 'Εφαρμογές Ενσωματωμένων Συστημάτων'. Κύρια θέματα και στόχοι που καλύπτονται στο μάθημα είναι

- Η σχεδίαση συστημάτων συναγερμού (Alarm systems)
- Η χρήση μικροελεγκτών και κατάλληλων Wi-Fi Module, για την διαχείριση δεδομένων μέσω του διαδικτύου (IOT)
- Η σχεδίαση συστημάτων μέτρησης και απεικόνισης διαφόρων μεγεθών (θερμοκρασία, πίεση)
- Η σχεδίαση συστήματα συλλογής και επεξεργασίας δεδομένων από τις ψηφιακές και αναλογικές εισόδους.
- Έλεγχος σειριακής θύρας και χρήση διαφόρων πρωτοκόλλων επικοινωνίας.
- Διαχείριση αναλογικών και ψηφιακών αισθητηρίων για την μέτρηση διαφόρων μεγεθών.

Όλο το σύστημα της διπλωματικής βαφτίστηκε «NiK\_IoT» και περιλαμβάνει όλα τα παραπάνω. Το NiK\_IoT είναι συνδυασμός από δομές Hardware και Software. Αρχικά, θα γίνει μια αναφορά στις βασικές έννοιες ώστε ο αναγνώστης να μπορέσει να κατανοήσει καλύτερα την τεχνική ανάλυση που θα ακολουθήσει. Θα αναφερθούμε στον όρο Smart Home (Έξυπνο σπίτι) και στο Internet of Things (Διαδίκτυο των Πραγμάτων). Στις ενότητες 1.4 και 1.5 θα παρουσιάσουμε τους Servers (εξυπηρετητές) HiveMQ και ThingSpeak που υποστηρίζουν την εφαρμογή. Στα κεφάλαια 3 και 4 παρουσιάζουμε στο ενιαίο πλαίσιο διάδρασης του συστήματος μέσω των εφαρμογών Android 'NiK\_IoT' και 'Cal\_End'. Στο κεφάλαιο 7 δίνεται το κατασκευαστικό μέρος και η συγγραφή τελειώνει με την αξιολόγηση και τα συμπεράσματα. Στο τέλος, στο παράρτημα, παραθέτουμε αυτούσιους τους κώδικες προγραμματισμού και μερικές φωτογραφίες από την όλο το έργο

Κατά τη συγγραφή έγινε προσπάθεια να αναδείξουμε τη διάδραση των Cloud Servers και του user interface με χρήση τόσο του Front-End programming όσο και του Back-End programming. Θα εξετάσουμε τα οφέλη και τους περιορισμούς αυτών των τεχνολογιών, καθώς και το δυναμικό τους για μελλοντικές εξελίξεις στον τομέα του IoT. Κύριος σκοπός της συγγραφής της διπλωματικής δεν είναι τόσο η παράθεση γνώσεων και κώδικα, όσο η σωστή τεχνικά παρουσίαση του 'NiK\_IoT', ώστε η εργασία να χρησιμοποιηθεί ως αναφορά για χρήση και επέκταση. Θα προσπαθήσουμε να αναδείξουμε τα χαρακτηριστικά και τις δυνατότητες που προκύπτουν από την ενσωμάτωση αυτής της συσκευής στο σπίτι μας. Τροποποιώντας την υπάρχουσα ηλεκτρική εγκατάσταση με σχετικά μικρό κόστος, μπορούμε να μετατρέψουμε την παλιά ηλεκτρική εγκατάσταση σε έξυπνη. Η τοποθέτηση μπορεί να γίνει είτε μέσα στον πίνακα διανομής είτε εξωτερικά. Έχοντας στη διάθεσή μας μια πλατφόρμα διαχείρισης της ενεργειακής κατανάλωσης, είναι εύκολο να εφαρμόσουμε μια πράσινη πολιτική στο σπίτι μας ή στην επιχείρησή μας. Ας ελπίσουμε ότι θα δώσουμε ένα έναυσμα στους νέους μηχανικούς για άμεσα οικονομικά έσοδα.

Δόθηκε μεγάλη βαρύτητα στα διαγράμματα, ένα πανίσχυρο εργαλείο στην επικοινωνία των πληροφοριών, με σκοπό την απλοποίηση ιδεών και την ανάδειξη των μοτίβων και των σχέσεων. Σε κάθε διάγραμμα δίνεται και υπερσύνδεσμος όπου μπορεί ο αναγνώστης να κατεβάσει το διάγραμμα σε μορφή pdf, όπου φαίνονται καλύτερα οι λεπτομέρειες και τα σχόλια. Σε πολλές περιπτώσεις κρίθηκε καλύτερο να γίνει αναφορά με τη χρήση

υπερσυνδέσμου στο κειμενογράφο Word Office που χρησιμοποιήσαμε αντί της πατροπαράδοτη μεθόδου παράθεσης αναφορών στο τέλος του κυρίως κειμένου.

Σε κάθε περίπτωση όπου κρίθηκε απαραίτητο, προσθέσαμε και το κόστος κάθε εξαρτήματος ή άδειας χρήσης. Το συνολικό κόστος του Project ήταν γύρω στα 500 ευρώ. Θεωρώ ότι είναι σημαντικό να συμπεριλάβετε το κόστος των υλικών εξαρτημάτων και των αδειών σε μια διπλωματική εργασία, καθώς αυτό αντιπροσωπεύει έναν σημαντικό παράγοντα στο συνολικό κόστος του έργου, πέραν του πνευματικού άχθους. Αυτό μπορεί να βοηθήσει στην αξιολόγηση της εφικτότητας του έργου και στην επιλογή της καλύτερης δυνατής λύσης για την υλοποίησή του. Εξάλλου, είναι κοινό μυστικό ότι πίσω από το engineering κρύβεται το κέρδος.

### 1.1 Παρόμοιες μελέτες περίπτωσης.

Στην ενότητα αυτή θα παρουσιάσουμε δυο συσκευές εμπορίου με παροιμίες λειτουργίες και δυνατότητες και θα επισημάνουμε τις διαφορές. Η πρώτη είναι η σειρά Smart temperature monitoring relays [1] της [ABB](#) και η δεύτερη είναι το [POWR3](#) [2]. Το Powr\_3 είναι ένας έξυπνος μετρητής ενέργειας της εταιρίας Son off

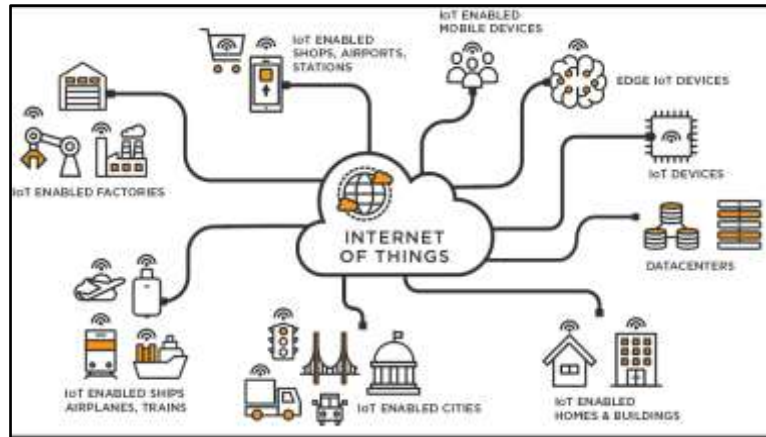
Τα Smart temperature monitoring relays της ABB παρέχουν αξιόπιστη προστασία στα συστήματα του χρήστη, αναλαμβάνοντας τον έλεγχο της θερμοκρασίας σε διάφορες εφαρμογές. Επιπλέον, διαθέτουν ενσωματωμένη οθόνη LCD που παρέχει πληροφορίες σχετικά με την κατάσταση του συστήματος και μια ακριβή μέθοδο μέτρησης της θερμοκρασίας. Η σειρά Smart temperature monitoring relays διαθέτει επίσης τη δυνατότητα One Touche - NFC parametrization μέσω της smartphone εφαρμογής ABB EPiC, η οποία επιτρέπει την εύκολη και γρήγορη επικοινωνία με το hardware είτε για ανάγνωση είτε για πέρασμα παραμέτρων. Το κόστος αγοράς ενός τέτοιου Smart temperature monitoring relay της ABB κυμαίνεται στα 300 ευρώ.

Η δεύτερη, το POWR3 είναι ένας έξυπνος μετρητής ενέργειας. Η συσκευή έχει μεγάλο εύρος λειτουργίας ως αναφορά την τάση δικτύου που επιβλέπει. Τοποθετείται στην αρχή των καταναλώσεων και το μέγιστο ρεύμα λειτουργίας είναι τα 25 A. Η συσκευή συνδέεται στο τοπικό δίκτυο WI-FI. Ο τρόπος σύνδεσης της συσκευής με το δίκτυο είναι εύκολος και γίνεται με την διαδικασία quick pairing. Το POWR3 συνεργάζεται με την εφαρμογή eWeLink η οποία είναι διαθέσιμη και σε συσκευές Android και σε IOS δωρεάν. Μπορούμε να βλέπουμε με λεπτομέρειες την κατανάλωση των συνδεδεμένων φορτίων. Ποιο συγκεκριμένα την ένταση του ρεύματος, την τάση λειτουργίας και την συνολική κατανάλωση. Η εφαρμογή κρατά τα δεδομένα κατανάλωσης για 100 μέρες. Συνεργάζεται με την εφαρμογή Alex Energy Dashboard με την οποία μπορούμε να έχουμε μια ποιο εποπτική εικόνα της ενεργειακής κατανάλωσης. Η εφαρμογή eWeLink δίνει την δυνατότητα προστασίας από υπερένταση το όριο το καθορίζει ο χρήστης, επιπρόσθετα δίνεται η δυνατότητα να μοιράσουμε την εφαρμογή και να δώσουμε τον έλεγχο του σπιτιού σε δεύτερο χρήστη. Το κόστος αγοράς του POWR3 κυμαίνεται στα 30 Ευρώ

Ας προσπαθήσουμε τώρα να κάνουμε μια σύγκριση των δυο αυτών αξιολόγων συστημάτων με το NiK\_IoT. Βασικό πλεονέκτημα των δυο παραπάνω είναι ο τρόπος σύνδεσης στο δίκτυο ο οποίος γίνεται μέσω αυτόματης μεθόδου. Στην περίπτωση του NiK\_IoT χρειάζεται παρέμβαση στον κώδικα. Στην πρώτη περίπτωση ο κωδικός ελέγχου θερμοκρασίας τρέχει μέσα στο Hardware και το εύρος θερμοκρασιών που μετράει το σύστημα είναι πολύ μεγάλο -200 °C +850 °C ενώ ο αισθητήρας Dallas που χρησιμοποιεί το NiK\_IoT έχει πολύ μικρότερο εύρος -55°C έως +125° [3]. Στον αντίποδα η καινοτομία που παρέχει το NiK\_IoT είναι η απεικόνιση on line τους κόστους κατανάλωσης της ενεργειακής κατανάλωσης μετά από την εισαγωγή των ιδιοπαραμετρών (βλέπε Εικόνα 29 :). Το POWR3 δεν κάνει έλεγχο θερμοκρασίας και το μέγιστο ρεύμα ελέγχου είναι 25 A. Αντίστοιχα το ρεύμα ελέγχου Smart temperature monitoring relays της ABB είναι μόλις 4 A. Στον αντίποδα το NiK\_IoT προσφέρει έλεγχο φορτίου έως 60 A και πραγματικό feed back μέσω και της τιμής του ρεύματος και της κατάστασης του ρελέ.

## 1.2 Internet of Things (IoT).

Το Internet of Things (IoT) αναφέρεται στην σύνδεση συσκευών με το διαδίκτυο και την ανταλλαγή δεδομένων μεταξύ τους. Η ιδέα του IoT ξεκίνησε τη δεκαετία του 1980, όταν ο Kevin Ashton, Βρετανός επιχειρηματίας και επιστήμονας, πρότεινε τη χρήση της τεχνολογίας RFID (Radio Frequency Identification) για την αυτόματη αναγνώριση αντικειμένων.



**Εικόνα 1: Εννοιολογικός χάρτης του IoT**

Το IoT όμως έγινε πιο γνωστό στο ευρύ κοινό στη δεκαετία του 1990, όταν η General Electric χρησιμοποίησε αισθητήρες για τη συλλογή δεδομένων σε απομακρυσμένες τοποθεσίες και την αποστολή τους σε ένα κεντρικό σύστημα μέσω του διαδικτύου. Στη συνέχεια, η Cisco αναφέρει ότι το πρώτο συνδεδεμένο σκεύος IoT ήταν ένα αυτόματο αποθετήριο Coca-Cola το 1982, το οποίο είχε σχεδιαστεί για να ελέγχει τις πωλήσεις και τα αποθέματα.

Το Internet of Things (IoT) έχει πολλά πλεονεκτήματα και εφαρμόζεται σε διάφορους τομείς της ζωής. Μερικά από τα βασικά πλεονεκτήματα του IoT είναι τα εξής: **Βελτιωμένη παραγωγικότητα:** Οι συνδεδεμένες συσκευές IoT μπορούν να επιτρέψουν την αυτοματοποίηση διαφόρων διαδικασιών και την απομακρυσμένη παρακολούθηση της απόδοσης των συστημάτων, οδηγώντας έτσι σε αύξηση της παραγωγικότητας. **Βελτιωμένη ασφάλεια:** Οι συσκευές IoT μπορούν να επιτρέψουν την παρακολούθηση των συστημάτων ασφαλείας και την απομακρυσμένη ανίχνευση προβλημάτων, ενώ μπορούν επίσης να παρέχουν πληροφορίες για πιθανούς κινδύνους, βελτιώνοντας έτσι την ασφάλεια του συστήματος και των χρηστών. **Βελτιωμένη απόδοση των συστημάτων:** Οι συνδεδεμένες συσκευές IoT μπορούν να επιτρέψουν την αυτόματη ρύθμιση των συστημάτων, βελτιώνοντας έτσι την απόδοσή τους και μειώνοντας τη σπατάλη ενέργειας. [4]

Παρόλο που η τεχνολογία αυτή έχει αρκετά πλεονεκτήματα, όπως τη βελτιωμένη ευκολία και αποδοτικότητα στις εργασίες, υπάρχουν και ορισμένα μειονεκτήματα που πρέπει να ληφθούν υπόψη. [5]

Ένα από τα μειονεκτήματα του IoT είναι η ασφάλεια των συσκευών και των δεδομένων. Οι συσκευές IoT μπορεί να είναι ευάλωτες σε επιθέσεις και παραβίαση της ιδιωτικότητας. Επίσης, η υψηλή συνδεσιμότητα μπορεί να οδηγήσει σε αύξηση του αριθμού των σημείων εισόδου για επιθέσεις στο δίκτυο.

Τα smartphones αποτελούν σημαντικό κομμάτι του IoT καθώς είναι συνδεδεμένα με διάφορες συσκευές IoT και μπορούν να επικοινωνούν μεταξύ τους μέσω διαφόρων τεχνολογιών, όπως το Bluetooth, το NFC και το Wi-Fi. Τα smartphones μπορούν να λειτουργήσουν ως συσκευές ελέγχου για άλλες συνδεδεμένες συσκευές IoT, όπως έξυπνες λάμπες, θερμοστάτες, κλειδαριές και ηχεία, καθώς και να συλλέγουν δεδομένα από αισθητήρες στο περιβάλλον τους, όπως θερμοκρασίας, υγρασίας, κίνησης και φωτός. Επιπλέον, οι συσκευές IoT μπορούν να επιτρέψουν στους χρήστες να ελέγχουν και να διαχειρίζονται τις συνδεδεμένες συσκευές μέσω εφαρμογών που είναι διαθέσιμες στα smartphones τους. Αυτό σημαίνει ότι οι χρήστες μπορούν να ελέγχουν και να προγραμματίζουν τις συνδεδεμένες συσκευές τους από απόσταση, καθιστώντας τη ζωή τους πιο εύκολη και άνετη.



## 1.4 The Thing Speak Server.

Ο [Thing Speak](#) [7] είναι μια διαδικτυακή πλατφόρμα ανοικτού λογισμικού που παρέχει υπηρεσίες IoT. Στον διακομιστή στέλνονται τα δεδομένα από τον σταθμό (collect) όπου και αποθηκεύονται. Μπορούμε να έχουμε πρόσβαση στα δεδομένα τα οποία μπορούμε να τα δηλώσουμε είτε ιδιωτικά είτε δημόσια ανάλογα την χρήση τους. Παρέχει ενιαία μορφή διαχείρισης δεδομένων χρησιμοποιώντας γλώσσα προγραμματισμού MATLAB και την δυνατότητα να κάνεις on line οργάνωση δεδομένων και επεξεργασία (analyze) στο υπολογιστικό νέφος (μέθοδος Back-End programming) με στόχο την εξόρυξη πληροφορίας και την απόκριση (act).



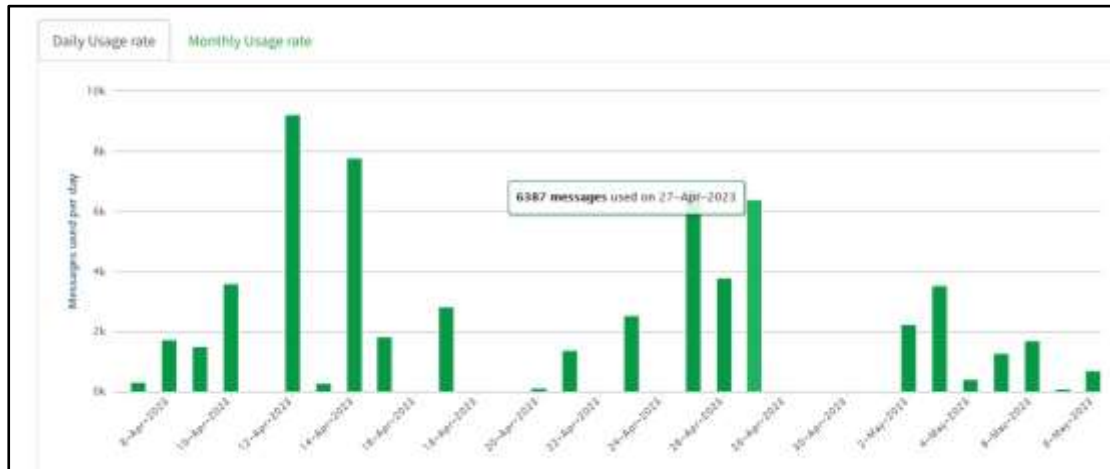
**Εικόνα 3: ThingSpeak μια πλατφόρμα ανάλυσης του Internet of Things (IoT)**

Συγκεκριμένα, κάνουμε εγγραφή και δημιουργούμε έναν λογαριασμό στην πλατφόρμα, έτσι δημιουργούμε το δικό μας κανάλι και ορίζουμε διαφορετικά πεδία (fields). Κάθε Field λειτουργεί σαν θέση μνήμης για τις μεταβλητές που θέλουμε να διαχειριστούμε. Κάθε κανάλι έχει ένα μοναδικό κλειδί εγγραφής (writing key) και ένα μοναδικό κλειδί ανάγνωσης (reading key). Χρησιμοποιούμε το κλειδί εγγραφής για να γράψουμε στο κανάλι και το κλειδί ανάγνωσης για να αναγνώσουμε τα πεδία εγγραφών. Οι τιμές των κλειδιών καταχωρούνται στον κώδικα που τρέχει στον Middleware για να γίνει η αυθεντικοποίηση. Οι τιμές αυτές των κλειδιών καταχωρούνται επίσης και στην εφαρμογή του κινητού τηλεφώνου για να γίνει και εκεί η απαιτούμενη αυθεντικοποίηση. Για τις ανάγκες της διπλωματικής Δημιουργήθηκε ένας λογαριασμός και αγοράστηκε μια άδεια χρήσης τύπου Student. Η άδεια χρήσης student δίνει την δυνατότητα να γράφεις δεδομένα στο κανάλι σου κάθε δευτερόλεπτο με κόστος της συνδρομής 55 ευρώ τον χρόνο. Στην ελεύθερη έκδοση το διάστημα εγγραφής (Message update interval limit) είναι 15 sec το οποίο δεν μας βοηθούσε καθόλου στις δοκιμές και στην ανάπτυξη ειδικά της εφαρμογής Cal\_En. Δημιουργήθηκαν τρία ξεχωριστά κανάλια για την διαχείριση των δεδομένων.

- Το πρώτο κανάλι βαφτίστηκε Request DBMS Channel και χρησιμοποιείται μόνο για την κλήση των ενεργειακών καταγράφων (βλέπε [Εισαγωγή στην εφαρμογή Cal\\_En](#)).
- Το δεύτερο κανάλι ονομάζεται Monitor Channel και χρησιμοποιείται αποκλειστικά για την καταγραφή των παραμέτρων λειτουργίας του συστήματος θερμοκρασίας και, καταναλισκόμενης ενέργειας. (βλέπε: [Load Current Consumption Panel](#)).
- Το τρίτο κανάλι ονομάζεται Command Channel και χρησιμοποιείται μόνο για την εντολή ενεργοποίησης ON-OFF. (βλέπε: [The](#)).

Κάθε κανάλι έχει μέγιστο αριθμό οκτώ πεδίων (fields) όπου μπορείς να γράφεις και να διαβάζεις τις τιμές εγγραφής. Στο Request DBMS Channel υπάρχουν τρία fields, στο Monitor Channel πέντε και στο Command Channel ένα. Το ThingSpeak μπορεί να συνδεθεί με το Matlab για την εκτέλεση προηγμένης ανάλυσης δεδομένων και οπτικοποίησης αυτών. Το Matlab παρέχει ισχυρά εργαλεία για τη δημιουργία προσαρμοσμένων αλγορίθμων. Η συνεργασία ThingSpeak και Matlab, γίνεται μέσω της εφαρμογής MATLAB Analysis [8]. Διαβάζουμε τα δεδομένα στα κανάλια μας, και στη συνέχεια χρησιμοποιώντας ενσωματωμένες

συναρτήσεις του Matlab επεξεργαζόμαστε τα δεδομένα και να τα οπτικοποιούμε κατά επιλογή. Υπάρχουν αρκετά παραδείγματα στην πλατφόρμα που βοηθούν τον αρχάριο προγραμματιστή για την ανάπτυξη custom εφαρμογών.



Εικόνα 4: Η διακίνηση των μηνυμάτων στα τρία κανάλια ανά ημέρα

### My Account

License Type: Student License: 41135634

Units: 1 Expiration Date: 30 Jan 2024

|          | Purchased | Remaining |
|----------|-----------|-----------|
| Messages | 35000000  | 32834865  |
| Channels | 10        | 6         |

\*Each unit can contain at most 33 million messages  
 \*\*Messages purchased may be granted [credit](#)

Recommended maximum daily usage based on your annual capacity: **121,611 messages**

For questions about exceeding suggested daily usage rate or for purchasing new units, see the [Licensing FAQ](#)

Εικόνα 5: Στοιχεία του λογαριασμού στο ThingSpeak

Στην Εικόνα 4 φαίνεται η διακίνηση των μηνυμάτων κατά τη διάρκεια των δοκιμών τον μήνα Απρίλιο. Στην Εικόνα 5 εμφανίζονται πόσα μηνύματα έχουμε αγοράσει και πόσα μας απομένουν μέχρι το τέλος της συνδρομής. Επιπροσθετα εμφανίζονται πόσα κανάλια μπορούμε ακόμα να δημιουργήσουμε και ποιος είναι ο βέλτιστος αριθμός μηνυμάτων ανά ημέρα, ώστε να μην έχουμε υπέρβαση των ορίων.

## 1.5 The HiveMQ Cloud Server.

Το HiveMQ Cloud [9], Εικόνα 6, είναι μια πλατφόρμα cloud του HiveMQ, που προσφέρει ένα διαχειρίσιμο και αξιόπιστο διαμεσολαβητή (broker MQTT). Το MQTT [10] είναι ένα πρωτόκολλο επικοινωνίας που χρησιμοποιείται στο Internet of Things (IoT) για τη μεταφορά μηνυμάτων μεταξύ συσκευών. Το HiveMQ Cloud προσφέρει σε εταιρείες και οργανισμούς τη δυνατότητα να χρησιμοποιούν έναν κεντρικό broker MQTT στο cloud για τη διαχείριση των συνδέσεων και των μηνυμάτων μεταξύ των συσκευών τους, εξοικονομώντας έτσι χρόνο και χρήματα από την ανάπτυξη και τη συντήρηση του δικού τους broker – Server. Αυτό είναι εξάλλου και ένα από τα πλεονεκτήματα του cloud computing.



Εικόνα 6: Η πλατφόρμα [HiveMQ](#) αξιόπιστη και κλιμακούμενη διαχείριση IoT στο cloud.

Το HiveMQ Cloud παρέχει επεκτασιμότητα, ασφάλεια και κλιμακωσιμότητα. Η ασφάλεια διασφαλίζεται μέσω των συνδέσεων TLS/SSL. Τα TLS (Transport Layer Security) και SSL (Secure Sockets Layer) είναι πρωτόκολλα ασφαλείας για την κρυπτογράφηση των επικοινωνιών στο Διαδίκτυο. Επιπροσθετα το HiveMQ Cloud παρέχει υποστήριξη πιστοποιητικών διαχείρισης συνδέσεων και δικαιωμάτων πρόσβασης.

Το MQTT (Message Queuing Telemetry Transport) είναι ένα πρωτόκολλο ανοιχτού κώδικα που χρησιμοποιείται για τη μεταφορά μηνυμάτων μεταξύ συσκευών μέσω του διαδικτύου. Το MQTT αναπτύχθηκε για να είναι αποδοτικό σε συνθήκες περιορισμένων πόρων δικτύου και ενέργειας, όπως είναι οι συσκευές IoT, ώστε να είναι αξιόπιστο σε ασύγχρονες συνδέσεις.

Το MQTT χρησιμοποιεί ένα μοντέλο εκδότη-συνδρομητή, όπου οι συσκευές εγγράφονται σε θέματα (topics) και λαμβάνουν μηνύματα μόνο για αυτά τα θέματα. Αυτό κάνει το MQTT πολύ αποδοτικό, καθώς οι συσκευές λαμβάνουν μόνο τα μηνύματα που τους αφορούν. Το MQTT χρησιμοποιεί ελάχιστη επικοινωνία με τον broker (μεσολαβητή), πράγμα που καθιστά τη μετάδοση μηνυμάτων γρήγορη και αξιόπιστη [11]. Επιπλέον, το MQTT παρέχει μηχανισμούς ασφαλείας, όπως κρυπτογράφηση και ταυτοποίηση, για την προστασία των μηνυμάτων από κακόβουλους χρήστες.

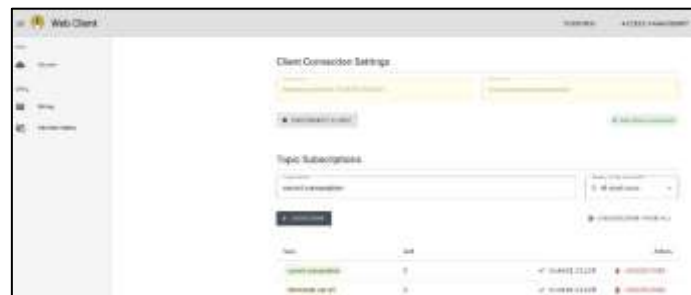
Λόγω των εκτεταμένων δοκιμών στα πλαίσια της διπλωματικής, υπερβήκαμε τα όρια της δωρεάν έκδοσης όσον αφορά τον αριθμό των συνεδριών και αναγκαστήκαμε να αγοράσουμε την έκδοση Pay-as-You-Go. Αυτή η έκδοση προσφέρει έως 1000 συνεδρίες και μέγιστο όγκο δεδομένων 100 GB. Κάθε συνεδρία κοστίζει 0,10 δολάρια. Για παράδειγμα, τον Απρίλιο πλήρωσα 7,1 δολάρια για τη χρήση του συστήματος ενώ τον Μάρτιο πλήρωσα 14,70 δολάρια.

Στην εικόνα 7 φαίνονται μέσω screenshot τα στατιστικά όλων των συνδέσεων που έχουν γίνει στο Server για τις ανάγκες της εργασίας. Στον HiveMQ μπορείς να συνδεθείς και μέσω browser με τη λειτουργία Web client (βλέπε εικόνα 8). Στην εικόνα 9 φαίνονται τα topics που

λαμβάνουν χώρα στην υλοποίηση της διπλωματικής. Για να γίνει η απεικόνιση τους στον Web client, θα πρέπει προηγουμένως να γίνει εγγραφή (subscribe).



Εικόνα 7: Στατιστικά των συνδέσεων στο Cluster του HIVEMQ Cloud Server



Εικόνα 8: Συνδεση Web Client στο HiveMQ Cluster

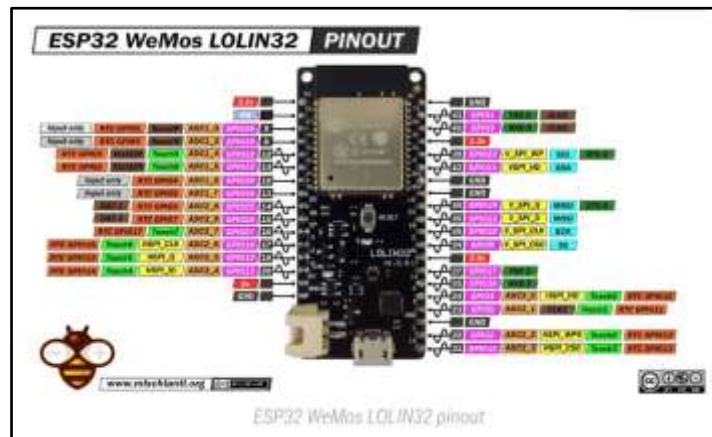
The screenshot shows the 'PUBLISH' page in the HiveMQ web interface, displaying a list of topics. The table has columns for 'Message', 'Topic', 'QoS', and 'Timestamp'. The topics listed include 'temperature\_outdoor', 'level\_back\_1', 'temperature', 'string\_value\_1', and 'status'.

| Message | Topic               | QoS | Timestamp     |
|---------|---------------------|-----|---------------|
| 18.00   | temperature_outdoor | 0   | 1682609648424 |
| 1       | level_back_1        | 0   | 1682609647702 |
| 22.25   | temperature         | 0   | 1682609647622 |
| 0.00    | string_value_1      | 0   | 1682609647485 |
| 16      | status              | 0   | 1682609647481 |
| 18.00   | temperature_outdoor | 0   | 1682609646423 |
| 1       | level_back_1        | 0   | 1682609644804 |
| 22.25   | temperature         | 0   | 1682609644803 |
| 0.00    | string_value_1      | 0   | 1682609644803 |
| 11      | status              | 0   | 1682609644801 |
| 17.94   | temperature_outdoor | 0   | 1682609644800 |

Εικόνα 9: Τα topics όπως εμφανίζονται στον Web client του HiveMQ Cluster

## 1.6 Wemos LoLIN 32 by Espressif.

Σε αυτό το επίπεδο έχουμε την σύνδεση μεταξύ του πραγματικού κόσμου ( Real world) με τις IoT υπηρεσίες ( cloud ). Το επίπεδο αυτό αναφέρεται στην βιβλιογραφία ως Middleware. Η πλακέτα Wemos Lolin 32 είναι μια αναπτυξιακή πλατφόρμα που βασίζεται στον



**Εικόνα 10: Wemos Lolin 32 – Μικροελεγκτής βασισμένος στο ESP32 chip της Espressif**

μικροεπεξεργαστή ESP32 της εταιρίας Espressif [12] και είναι ιδανική για εφαρμογές IoT λόγω της χαμηλής κατανάλωσης (220 mA) και του κόστους της .Η πλακέτα δίνεται στην Εικόνα 10 μαζί με τα περιφερειακά της Έχει μεγάλη υπολογιστική ισχύ , ενσωματωμένο Wi-Fi και Deep Sleep Operations. Προγραμματίζεται σε περιβάλλον Arduino IDE χρησιμοποιώντας την γλώσσα “Arduino” ( ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++) με υποστήριξη από ένα σύνολο βιβλιοθηκών. Επιπρόσθετα έχει ενσωματωμένη κεραία και Bluetooth module. Τα διαφορετικά είδη λειτουργίας και οι αντίστοιχες καταναλώσεις δίνονται στον πίνακα [13].

| Power Mode          | Description            | Power Consumption                              |
|---------------------|------------------------|--|
| Active (RF Working) | Wi-Fi and Bluetooth ON | 95-240 mA                                      |
| Modem Sleep         | CPU ON                 | Max Speed: 20mA<br>Normal: 5-10mA<br>Slow: 3mA |
| Light Sleep         |                        | 0.8 mA   |
| Deep Sleep          | ULP ON 150 uA          | ULP ON 150 uA                                  |
| Deep Sleep          | RTC Timer+ RTC Memory  | 10 uA  |
| Hibernation         | RTC timer Only         | 5 μA   |

**Πίνακας 1 :Τροποι λειτουργίας του μικροελεγκτη Esp32**

Οι βιβλιοθήκες μπορούν να χρησιμοποιηθούν για να απλουστεύσουν τη διαδικασία ανάπτυξης προγραμμάτων, να προσφέρουν λειτουργίες που δεν είναι διαθέσιμες στον πυρήνα της γλώσσας Arduino ή να επεκτείνουν τις λειτουργίες που ήδη υπάρχουν.

Τα Κύρια τεχνικά χαρακτηριστικά του Wemos Lolin32 είναι :

- Επεξεργαστής: ESP32-D0WDQ6-V3 με διπλούς πυρήνες 32-bit Xtensa LX6 στα 240MHz
- Μνήμη: 4MB Flash και 520KB SRAM (Static Random Access Memory).
- Συνδεσιμότητα: WiFi 802.11 b/g/n και Bluetooth 4.2 BR/EDR και BLE ( Bluetooth Low Energy)

Το WiFi 802.11 b/g/n είναι ένα ασύρματο πρότυπο δικτύωσης που χρησιμοποιείται για την ασύρματη επικοινωνία μεταξύ συσκευών σε ένα δίκτυο. Το πρότυπο αυτό υποστηρίζει ταχύτητες μεταφοράς δεδομένων μέχρι 600 Mbps στα 2.4 GHz, που είναι μια συχνότητα που μπορεί να διατρέξει μικρές αποστάσεις και να διαπεράσει τοίχους και άλλα εμπόδια. Το πρότυπο 802.11b υποστηρίζει ταχύτητες μεταφοράς δεδομένων έως και 11 Mbps στα 2.4 GHz, το 802.11g υποστηρίζει ταχύτητες μεταφοράς δεδομένων έως και 54 Mbps στα 2.4 GHz, ενώ το 802.11n υποστηρίζει ταχύτητες μεταφοράς δεδομένων έως και 600 Mbps στα 2.4 GHz και 5 GHz.

- Είσοδοι/Εξοδοι: 32 Inputs / Outputs
  - 26x Digital I / O (3.3V). All outputs can be PWM.
  - 18x Analog inputs.
  - 3x UART (Universal Asynchronous Receiver/Transmitter).
  - 3x SPI Serial Peripheral Interface Σύγχρονο πρωτόκολλο επικοινωνίας.
  - 2x I2S (Inter-IC Sound).
  - 2x DAC (Digital-to-Analog Converter).
  - 2x I2C (Inter-Integrated Circuit).
- Λειτουργικό σύστημα: Υποστηρίζει το πλαίσιο Arduino και το MicroPython.
- Τροφοδοσία: Λειτουργεί με 3.3V, ενώ η πλακέτα δέχεται τάσεις τροφοδοσίας από 5V έως 12V DC.
- Διαστάσεις: 25.5 x 14.2 x 2.5mm.
- Η τιμή του κυμαίνεται σε από 8 έως και 15 Ευρώ ( eBay).

(Το manual το μικροελεγκτή Esp32 είναι 730 σελίδες και δίνεται στο τέλος σε βιβλιογραφική αναφορά για όποιον θέλει να εντρυφήσει βαθύτερα [14] )

Να σημειώσουμε εδώ κάτι που αφορά τη χρήση γενικά των μικροελεγκτών (microcontroller) όπως ο ESP32, καθώς και τη διαφορά τους με τους μικροεπεξεργαστές (microprocessor) όπως π.χ. ο 8086. Ο μικροελεγκτής είναι ένα ελβετικός σουγιάς περιέχει σε ένα ολοκληρωμένο κύκλωμα - αξεσουάρ πέραν του πυρήνα του, επιπλέον κυκλώματα ώστε ο 'κάτοχος' του να εκτελεί εύκολα μια συγκεκριμένη διεργασία . Ενδεικτικά εργαλεία που εμπεριέχονται και μας κάνουν την ζωή εύκολη.

- I2C Controller (I2C).
- Remote Control Peripheral (RMT).
- Motor Control PWM (PWM).
- effuse Controller.
- Ethernet Media Access Controller (MAC).

Από την άλλη μεριά οι μικροεπεξεργαστές ( microprocessors) όπως ο Intel 8086 είναι ένας κεντρικός υπολογιστής σχεδιασμένος για να επεξεργάζεται δεδομένα και να εκτελεί λειτουργίες υπολογιστικών συστημάτων, έχει δικό του λειτουργικό σύστημα το οποίο δεν μπορείς να παρακάμψεις π.Χ. να του βάλεις να εκτελεί συνέχεια μια εργασία. Πέραν αυτού από μόνο του δεν κάνει τίποτα, θα πρέπει να του βάλεις επιπρόσθετα περιφερειακά (ότι δηλαδή έχει μέσα του ήδη ο microcontroller ) για να εκτελέσει μια εργασία π.χ. χρονισμός ανάφλεξης σε έναν κινητήρα θερμικής καύσης.

Οι microcontrollers βρίσκονται παντού. Οι μισές ηλεκτρικές συσκευές ενσωματώνουν έναν τέτοιο ολοκληρωμένο IC. Ενδεικτικά αναφέρω μερικές συσκευές όπως ποντίκια, πληκτρολόγια, τηλεχειριστήρια και αισθητήρες θερμοκρασίας και πυρανίχνευσης. Η χρήση τους θα γίνει ακόμα πιο ευρεία, καθώς πλέον κάθε συσκευή-αισθητήρας (End Points ) είναι μέρος ενός δικτύου και η επικοινωνία με αυτό γίνεται με τη βοήθεια ενός microcontroller. Μερικές φορές καταχρηστικά χρησιμοποιούμε τον όρο μικροεπεξεργαστής αντί για μικροελεγκτής.

## 1.7 Η εφαρμογή Android Virtuino

### 1.7.1 Εισαγωγή στην εφαρμογή Virtuino

Για την κατασκευή της εφαρμογής η οποία θα χειρίζεται όλα τα δεδομένα χρησιμοποιήσαμε την εφαρμογή Virtuino η οποία τρέχει μόνο σε λειτουργικό android. Το Virtuino είναι ένα λογισμικό που επιτρέπει στον άνθρωπο να παρακολουθεί την κατάσταση λειτουργίας ενός συστήματος αυτοματισμού καθώς και να το ελέγχει με εικονικά κουμπιά ή χειριστήρια από την οθόνη (Human Machine Interface). Θα μπορούσαμε να την περιγράψουμε και ως πρόγραμμα για δημιουργία SCADA (Supervisory Control and Data Acquisition) σε Smart phones. Παρουσιάζω μια λίστα με τις δυνατότητες της εφαρμογής που σημειωτέων είναι ελληνικής επινόησης και κατασκευής το συνάδερφο Πίος Lamprou.

- Εξειδικεύεται στο Διαδίκτυο των Πραγμάτων (IoT) δίνοντας στους χρήστες τη δυνατότητα να κάνουν Οπτικοποίηση και αλλαγές σε δεδομένα που αποθηκεύονται σε πλατφόρμες IoT. Υποστηρίζει τα πρωτόκολλα MQTT, και μέσω αυτών μπορεί να συνδεθεί με οποιαδήποτε πλατφόρμα IoT αλλά και Web Server που υποστηρίζει οποιοδήποτε Http API.
- Υποστηρίζει συνδέσεις με προγραμματιζόμενους λογικούς ελεγκτές (PLCs) όλων των εταιρειών, SCADA συστήματα, βιομηχανικές συσκευές και αισθητήρες μέσω του πρωτοκόλλου Modbus/TCP. Μπορεί να συνδεθεί ταυτόχρονα με PLCs διαφορετικών εταιρειών όπως Siemens, Schneider, ABB .
- Υποστηρίζει συνδέσεις με Arduino, ESP8266, ESP32, Raspberry Pi και γενικά οποιαδήποτε ηλεκτρονική συσκευή περιέχει μικροελεγκτή και μπορεί να προγραμματιστεί με κώδικα συμβατό με την εφαρμογή. Η επικοινωνία γίνεται μέσω WiFi ή Ethernet και χρησιμοποιούνται είτε το πρωτόκολλο MQTT είτε Web Sockets.
- Μπορεί να συγχρονιστεί με την IoT πλατφόρμα ThingSpeak και να κάνει ανταλλαγή δεδομένων και εμφάνιση γραφημάτων.
- Διαθέτει ένα πολύ εύχρηστο περιβάλλον οπτικοποίησης και ελέγχου με πλήθος από γραφικά στοιχεία με εκπληκτικές δυνατότητες παραμετροποίησης. Διαθέτει βιβλιοθήκες γραφικών στοιχείων στο cloud αλλά δίνει επίσης τη δυνατότητα στους δημιουργούς να εισάγουν δικά τους γραφικά και να δημιουργήσουν δικά τους γραφικά στοιχεία όπως μπουτόν, διακόπτες, ρυθμιστές, χρονόμετρα κλπ.
- Μπορεί να κάνει μια μικρή επεξεργασία τιμών που λαμβάνονται από συσκευές δεδομένων μέσω έτοιμων scripts όπως συγκρίσεις, μαθηματικές πράξεις, λογικές πράξεις κλπ.
- Μπορεί να λειτουργήσει τοπικά σαν πλατφόρμα IoT και να στέλνει ειδοποιήσεις, emails και SMS ενημερώνοντας του χρήστες για συμβάντα και συναγερμούς.
- Μπορεί να λειτουργήσει ως Gateway και να στέλνει αυτόματα δεδομένα από μια πλατφόρμα IoT ή συσκευή σε άλλη κάνοντας μετατροπή και στο πρωτόκολλο επικοινωνίας και στο format των δεδομένων.

Η Pro έκδοση μπορεί να δημιουργήσει απεριόριστα έργα, χωρίς περιορισμούς στον αριθμό των οθονών και τον αριθμό των γραφικών στοιχείων. Κάθε έργο μπορεί να εξαχθεί ως αρχείο με δυνατότητα κλειδώματος και να διανεμηθεί σε άλλους χρήστες ή πελάτες. Διατίθεται σε δύο εκδόσεις, τη Free έκδοση με περιορισμένο αριθμό εικονικών στοιχείων στη δημιουργία έργων και την Pro έκδοση χωρίς περιορισμούς. Η εφαρμογή είναι πλήρως προσαρμοσμένων χαρακτηριστικών (Full custom) και ο προγραμματισμός της γίνεται σε γραφικό περιβάλλον είναι διαθέσιμη από το Playstore καθώς και από τον ιστότοπο <https://virtuino.com/>. Η pro έκδοση με την οποία κατασκευάστηκαν οι εφαρμογές της διπλωματικής κοστίζει 10 ευρώ.

### 1.7.2 Προγραμματισμός στο γραφικό περιβάλλον της εφαρμογής Virtuino.

Η εφαρμογή Virtuino όπως προαναφέραμε διαθέτει ένα πολύ φιλικό περιβάλλον Διαθέτει βιβλιοθήκες γραφικών στοιχείων στο cloud και δίνει τη δυνατότητα στους δημιουργούς – προγραμματιστές να δημιουργήσουν δικά τους γραφικά στοιχεία.

Οι δυνατότητες της εφαρμογής είναι πολλές και δεν γίνεται να παρουσιαστούν όλες στα πλαίσια μιας διπλωματικής εργασίας. Η εφαρμογή δεν απευθύνεται σε κοινούς χρήστες. Από



**Εικόνα 11: Προγραμματισμός στο περιβάλλον της εφαρμογής Android Virtuino**

μόνη της δεν κάνει τίποτα. Με το άνοιγμα της εφαρμογής Virtuino εμφανίζεται ένα κενό Panel. Στο αρχικό αυτό Panel ο προγραμματιστής προσθέτει τα δικά του στοιχεία είτε απεικόνισης είτε ελέγχου. Στην ενότητα αυτή θα παρουσιάσουμε δύο παραδείγματα παραγωγής κώδικα .

Ας πάρουμε την απλή περίπτωση που ο προγραμματιστής- δημιουργός θέλει να απεικονίσει την τιμή ενός topics του HiveMQ στην οθόνη. Θα πρέπει αρχικά να προσθέσει τον αντίστοιχο Server (βλέπε [Εικόνα 17 MQTT Server settings over NiK\\_IoT API](#) ). Κατόπιν επιλέγεται από τη βιβλιοθήκη των widget π.χ. ένα value display ή ένα analog instrument. Στο widget αυτό επιλέγεται ο Server από τον οποίο θα παίρνει τιμές καθώς και η θέση μνήμης. Στην παρακάτω εικόνα 12 δίνεται ένα μέρος της παραμετροποίησης του display, όπου εμφανίζεται η τιμή της έντασης του ηλεκτρικού ρεύματος από το panel [Load Current Consumption](#). Το συγκεκριμένο widget έχει πολλές δυνατότητες όπως να δίνει εντολές ανάλογα με την τιμή της μεταβλητής που εμφανίζει.



**Εικόνα 12 :Advance value display Widget over Virtuino API**

Το δεύτερο παράδειγμα κώδικα αναφέρεται στο panel [Scale Analog to Current Value](#). (Hidden "Scale Analog to Current Value" και δίνει μια περίπτωση όπου το scaling γίνεται εύκολα στην πλευρά του SCADA, με δυνατότητα άμεσης παραμετροποίησης σε γλώσσα

υψηλού επιπέδου και όχι στη πλευρά του επεξεργαστή. Να σημειώσουμε εδώ ότι υπάρχει δυνατότητα να εισαγάγουμε password, ώστε ο χρήστης να μην μπορεί να πειράζει τέτοιες ευαίσθητες τιμές μεταβλητών. Το ζητούμενο εδώ είναι να μετατρέψουμε την ψηφιακή τιμή της αναλογικής τάσης εισόδου (εικόνα 14 αριστερά) σε τιμή ρεύματος (εικόνα 14 δεξιά). Θα χρησιμοποιήσουμε ένα script widget (Εικόνα 13). Αρχικά εισάγουμε την εξίσωση στο script :

Output memory = x \* y + z (συνήθως έχει την μορφή  $x = a * x + b$ )

- Input y: η ψηφιακή τιμή της αναλογικής τάσης εισόδου (συνήθως το βαφτίζουμε x).
- input x : η κλίση της ευθείας και (συνήθως το βαφτίζουμε a).
- input z : η μετατόπιση της ευθείας ως προς τον άξονα y και (συνήθως το ονομάζουμε b).
- output memory : η τιμή της έντασης του ρεύματος (συνήθως το ονομάζουμε y).



Εικόνα 13:Script widget της εφαρμογής Virtuino

Η εισαγωγή του script γίνεται με την εφαρμογή σε 'programming mode' επιλεγοντας το πρώτο κουμπι Unlock στην μεσαία εικόνα. Τα διαθέσιμα widget Panel εμφανίζονται



Εικόνα 14: Εναλλαγή λειτουργίας του Virtuino μεταξύ Run και Programming Mode

επιλεγοντας το + κάτω στην δεξιά εικόνα. Το μεγάλο πλεονέκτημα της μεθόδου scaling στην πλευρά της εφαρμογής φαίνεται στο κάτω μέρος της εικόνας όπου πατώντας στο display μπορούμε να εισάγουμε την επιθυμητή τιμή της μεταβλητής. Με αυτόν τον τρόπο είμαστε ευέλικτοι μπορούμε να πειραματιστούμε ευκολά και ασφαλή στην διάρκεια των δοκιμών και να αλλάξουμε την τιμή της μεταβλητής στον κώδικα μας εύκολα σε περίπτωση αλλαγής στο hardware μετέπειτα.

Μετα την ‘συγγραφή’ του κώδικα στο Virtuino και την αποθήκευση του project δημιουργείται ένα αρχείο με κατάληξη \*.vrt. Το αρχείο αυτό μπορεί να μετατραπεί σε εκτελέσιμο αρχείο APK από τον δημιουργό της εφαρμογής Πίπας Λαμπρού κατόπιν επικοινωνίας. Το αρχείο APK είναι το πακέτο εγκατάστασης των εφαρμογών στο λειτουργικό σύστημα Android. Δυστυχώς υπάρχει μια εν γένη αδυναμία να παρουσιαστεί ολόκληρος ο δομημένος κώδικας της εφαρμογής και το αντίστοιχο flow chart. Έγινε προσπάθεια και γράφτηκαν σχόλια δίπλα σε κάθε script και λεπτομερής περιγραφή της μεταβλητής τόσο στα servers panels όσο και στα ίδια τα panel. Ένα μέρος του ‘κώδικα’ δίνεται στο [Παράρτημα Κώδικας Virtuino](#)

**Το αρχείο \*.vrt της εφαρμογής NiK\_IoT είναι διαθέσιμο από το παρακάτω Link:**

<https://drive.google.com/file/d/1DGLRRVS52gsfHaDR6gQxLowbgBHggAwb/view?usp=sharing>

**Ανοίγοντας το αρχείο μέσω της εφαρμογής Virtuino μπορείτε να δείτε όλο τον ‘κώδικα’ επιλεγοντας Unlock panel**

**Το αρχείο .vrt της εφαρμογής Cal\_En είναι διαθέσιμο από το παρακάτω Link:**

<https://drive.google.com/file/d/1PJFNwx4fpvHWu8dWnjCmb0guQWKpRmmu/view?usp=sharing>

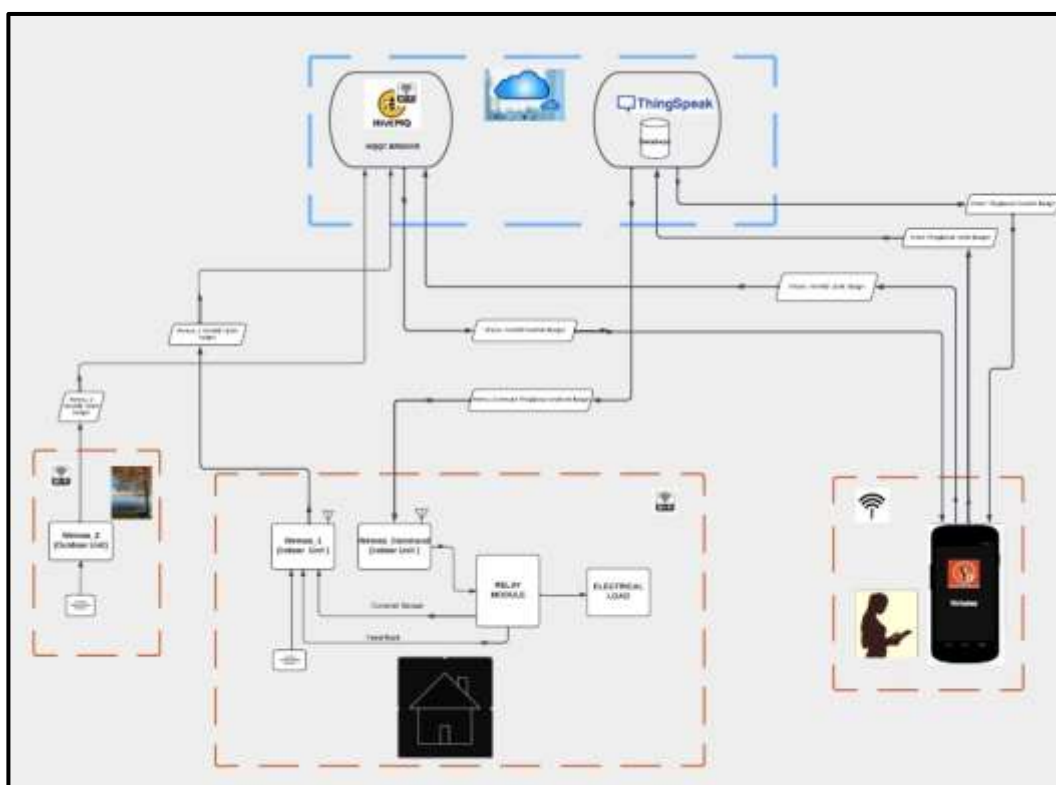
**Ανοίγοντας το αρχείο μέσω της εφαρμογής Virtuino μπορείτε να δείτε όλο τον ‘κώδικα’ επιλεγοντας Unlock panel**

## 2 Το Σύστημα NiK\_IoT και η αρχιτεκτονική του.

### 2.1 Εισαγωγή στο σύστημα NiK\_IoT.

Στο κεφάλαιο αυτό θα παρουσιάσουμε το σύστημα NiK\_IoT συνολικά. Αρχικά θα δώσουμε τον εννοιολογικό χάρτη του σε Block Diagram και κατόπιν θα περάσουμε στις λεπτομέρειες του. Το hardware του συστήματος θα παρουσιαστεί στο κεφάλαιο 5. Δόθηκε μεγάλη προσοχή στην κατασκευή των διαγραμμάτων έτσι ώστε ο αναγνώστης να καταλάβει την διακίνηση των δεδομένων καθώς και την συσχέτιση αυτών. Δυστυχώς οι εικόνες είναι μεγάλες και δεν χωρούν στην σελίδα A4. Θεωρήθηκε σκόπιμο να μην σχεδιαστούν αποσπασματικά αλλά καθ' ολοκληρία ώστε να φαίνεται όλες οι συσχετίσεις. Ο γενικός εννοιολογικός χάρτης δίνεται στο Block Διάγραμμα 1: Αρχιτεκτονική του συστήματος NiK\_IoT. Στην ενότητα 2.2 δίνεται Block Διάγραμμα 2: Η ροή δεδομένων στο σύστημα NiK\_IoT όπου φαίνονται η πορεία και η θέση κάθε μεταβλητής στον χωρόχρονο σε αντιστοιχία με τον γενικό εννοιολογικό χάρτη.

Για την δημιουργία των σχεδιαγραμμάτων χρησιμοποιήθηκε το πρόγραμμα <https://www.lucidchart.com>. Το Lucidchart είναι μια διαδικτυακή εφαρμογή δημιουργίας διαγραμμάτων και διανυσμάτων. Χρησιμοποιείται ευρέως από επιχειρήσεις και εκπαιδευτικά ιδρύματα για τη δημιουργία εικόνων, όπως διαγράμματα ροής, οργανογράμματα. Για τις ανάγκες της Διπλωματικής αποκτήθηκε η πληρωμένη άδεια χρήσης για ένα μήνα με κόστος 16 ευρώ.



**Block Διάγραμμα 1: Αρχιτεκτονική του συστήματος NiK\_IoT.**

Δίνεται παρακάτω Link με το παραπάνω Block Διάγραμμα 1 σε μορφή pdf :

<https://drive.google.com/file/d/1evUv3NqJDD2PHq9iZ1olZucctY7lUTwV/view?usp=sharing>

Ο σκοπός της εφαρμογής είναι ο έλεγχος της λειτουργίας ενός ηλεκτρικού φορτίου, όπως π.Χ. ένα ηλεκτρικό καλοριφέρ ή ένας ηλεκτρικός καυστήρας. Στόχος είναι ο αξιόπιστος απομακρυσμένος έλεγχος με ταυτόχρονη καταγραφή λειτουργικών παραμέτρων μες στόχο την βέλτιστη λειτουργία του φορτίου, την μείωση της κατανάλωσης ενέργειας και του κόστους λειτουργίας.

#### **Βασικά χαρακτηριστικά του NiK\_IoT είναι :**

1. Απομακρυσμένος έλεγχος λειτουργίας συστήματος θέρμανσης με πραγματικό Feed back .
2. Χρονοπρογραμματισμός.
3. Έλεγχος της θερμοκρασίας του σπιτιού μέσω έξυπνου θερμοστάτη με μεταβλητή υστέρηση.
4. Καταγραφή και Απεικόνιση των παραμέτρων λειτουργίας ( π.χ. Εσωτερική και Εξωτερική θερμοκρασία , Duty cycle Operation ).
5. Άμεση πρόσβαση και ανάκληση των τιμών της καταλισκωμένης ενέργειας για δεδομένο χρονικό διάστημα.
6. Απεικόνιση του τρέχοντος κόστους της καταναλισκόμενης ενέργειας σε πραγματικό χρόνο με δυο παραμετροποιημένες κλίμακες χρέωσης και νυκτερινή χρέωση. Δυνατότητα μηδενισμού από τον χρήστη . π.Χ. κάθε τετράμηνο όταν μηδενίζεται η χρέωσή.
7. Πολλαπλοί χρήστες , Διαχωρισμός Admin –User .
8. Μέτρηση κατανάλωσης ρεύματος από 4 A μέχρι και 60 A ( Μονοφασικό φορτίο )
9. Μερική Οπτικοποίηση δεδομένων.
10. Πλήρως προσαρμοσμένα Alarm καθώς και δυνατότητα ειδοποίησης του χρήστη μέσω SMS ή Email ή tweet .

Ας κάνουμε τώρα μια εμβάθυνση. Η Σουμελα μόνιμος κάτοικος Βέροιας θέλει να επισκεφτεί για την περίοδο του Πάσχα την εξοχική της κατοικία στην Χαλκιδική και θέλει το σπίτι να είναι ζεστό όταν πάει. Η Σουμελα ανοίγει την θέρμανση του σπιτιού πριν την έλευση της μέσω εφαρμογής στο κινητό. Το σπίτι έχει μόνιμη σύνδεση με το διαδίκτυο το ίδιο και το smart phone της .

Η διεύθυνση δικτύου του σπιτιού είναι δυναμική το ίδιο και η διεύθυνση του κινητού τηλεφώνου. Αυτό σημαίνει ότι δεν είναι δυνατή η απευθείας επικοινωνία μεταξύ των δυο αφού οι διευθύνσεις και των δυο αλλάζουν κατά διαστήματα. Η επικοινωνία γίνεται μέσω τρίτου ο οποίος βρίσκεται μόνιμα σε μια σταθερή διεύθυνση. Αυτός ο τρίτος είναι ο server. Όσοι είναι γεννημένοι πριν το 1980 ας θυμηθούν πως γινόταν η επικοινωνία όταν στο χωριό υπήρχε μόνο ένα τηλέφωνο και το “καφενείο του Γιωρκα” έπαιζε το ρολό του server. Κατά αντιστοιχία λοιπόν Η Σουμελα στέλνει (κάνει publish) το μήνυμα στο cloud και κατόπιν το ‘σπιτι’ διαβάζει (κάνει subscribed ) από το cloud .

Η επικοινωνία αυτή είναι διπλής κατεύθυνσης και αμφίδρομη, δηλαδή οι χρήστες μπορούν και να στείλουν και να λάμβανουν μηνύματα από το cloud. Στο σπίτι, οι συσκευές που επικοινωνούν με το cloud Server είναι δύο: η InDoor Unit και η OutDoor Unit. Η επικοινωνία "σπίτι-σύννεφο" γίνεται μέσω του [Wemos LoLIN 32 by Espressif](#). Οι Servers που επιλέχθηκαν για την επικοινωνία με το cloud είναι ο HiveMQ και ο ThingSpeak τους οποίους αναφεραμε στην ενότητα 1.5 και ενότητα 1.4 αντίστοιχα.

Για την καλύτερη κατανόηση των λεπτομερειών αλλά και της γενικής φιλοσοφίας του NiK\_IoT δημιουργήσαμε τρία σενάρια:

#### Σενάριο πρώτο η Σουμελα δίνει εντολή από το Android να ανάψει η θέρμανση

- Το Force ON bit μέσω της εφαρμογής γίνεται 1 και στέλνεται (Uplink) στο cluster στο HiveMQ οπου και γράφεται ( publish ) αποθηκεύεται ( μέχρι να αλλαχτεί από έναν άλλον client )

- Το κινητό ξαναδιαβάζει την εντολή Force on (Downlink) και ανάλογα την τρέχουσα κατάσταση γράφει (uplink) στο Field. 1 του command channel την τιμή 1 ή 0.
- Το Wemos\_3 Διαβάζει το παραπάνω field και αντίστοιχα ανοίγει η κλείνει το φορτίο.

Σενάριο δεύτερο η Σουμέλα επιλέγει μέσω του κινητού επιθυμητή εσωτερική θερμοκρασία έστω 22 βαθμούς

- Η τιμή set point 22 μέσω του Virtuino γράφεται (UpLiNK) στο HiveMQ το ίδιο και η τρέχουσα θερμοκρασία του σπιτιού (Real value) έστω 12.
- Το Virtuino κατεβάζει(DownLink) τις τιμές από το HiveMQ και κάνει την αφαίρεση (  $22 - 12 = 10$  άρα θετικό άρα γράφει γράφει στο Field. 1 του command Channel την τιμή 1.
- Η διαδικασία επαναλαμβάνεται.

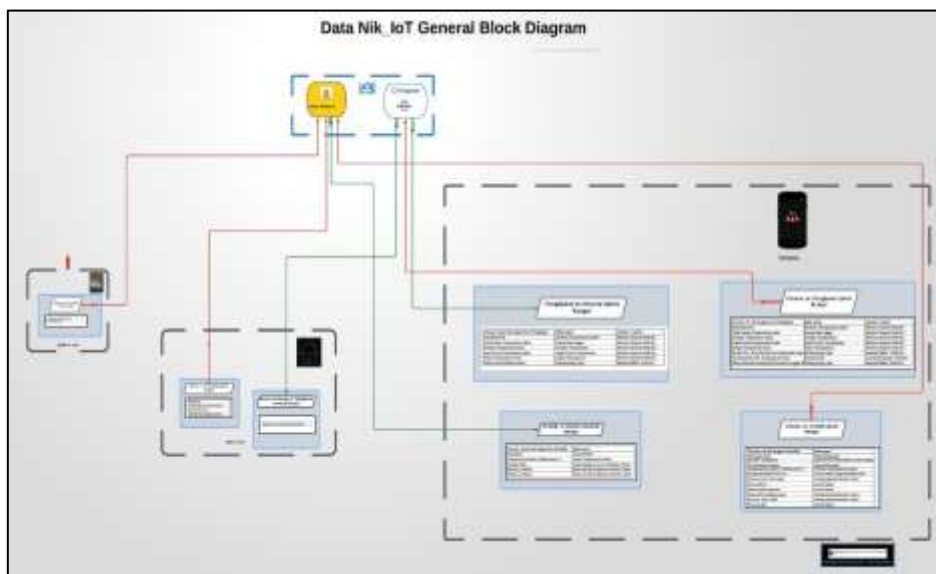
Όπως διαπιστώνεται αναφερόμαστε σε ένα σύστημα αυτομάτου ελέγχου με κλειστό βρόγχο ανάδρασης .

Η επεξεργασία για την ενεργοποίηση του θερμοστάτη γίνεται τοπικά στο Virtuino. Αρά για να λειτουργήσει ο βρόγχος ελέγχου θα πρέπει να είναι συνέχεια το Virtuino ανοιχτό για να κάνει την αφαίρεση , διαφορετικά η διαδικασία ελέγχου θα σταματήσει. Επομένως, θα πρέπει πάντα να υπάρχει συνέχεια ένα ‘Virtuino-server’ OnLine το οποίο θα λειτουργεί ως gateway. Για να ικανοποιηθεί αυτή η απαίτηση, χρησιμοποιήσαμε ένα Android BOX [15], το οποίο είναι σχετικά φθηνό και κοστίζει περίπου 23 ευρώ. Το Android BOX είναι μια συσκευή που τρέχει το λειτουργικό σύστημα Android και συνήθως συνδέεται με τηλεόραση όταν η τηλεόραση δεν είναι smart και προσφέρει δυνατότητες εκτέλεσης πολυμέσων και πλοήγησης στο διαδίκτυο. Στην παρούσα εργασία, προτείνουμε μια εναλλακτική χρήση η οποία σημειωτέων δοκιμάστηκε με απολυτή επιτυχία.

Σενάριο τρίτο: Ο χρήστης θέλει να δει κατά τη διάρκεια της πρώτης εβδομάδας του χειμώνα τον κύκλο λειτουργίας του συστήματος θέρμανσης, δηλαδή ποια χρονικά διαστήματα ήταν ενεργό και ποια όχι.

- Στο cluster μας στο HiveMQ στο topic Feedback\_1 γράφεται (uplink) από το Wemos\_1 η κατάσταση από το feedback input, Να υπενθυμίσουμε εδώ ότι ο HiveMQ broker δεν κρατάει ιστορικό, δρα σαν μεσολαβητής ( broker ) παίρνει και δίνει μέχρι εκεί.
- Εδώ έρχεται η λειτουργία του Virtuino ως gateway, το οποίο παίρνει την τιμή από το HiveMQ και την στέλνει στο ThingSpeak το οποίο σε αντίθεση με τον HiveMQ αποθηκεύει τις τιμές και κρατά ιστορικό καταγραφής με βάση τον χρόνο.
- Ο χρήστης κατεβάζει (DownLink) τις τιμές για την συγκεκριμένη χρονική περίοδο έχει πρόσβαση σε αυτές τόσο σε μορφή αρχείου xl Excel Office όσο και σε γράφημα (βλέπε [Εικόνα 21](#): Load Current Consumption Panel Of NiK\_IoT API ).

## 2.2 The Data flow diagram of NiK\_IoT system.



**Block Διάγραμμα 2 :Η ροή δεδομένων στο σύστημα NiK\_IoT**

**Δίνεται παρακάτω Link με το παραπάνω Block Διάγραμμα σε μορφή pdf :**

<https://drive.google.com/file/d/1mL6JEPruIThV2eIWxqDcLKzZwJN5PTIQ/view?usp=sharing>

Στο Block Διάγραμμα 2 δίνονται τα διαγράμματα ροής των δεδομένων. Τα διαγράμματα ροής δεδομένων (Data flow diagrams) είναι οπτικές αναπαραστάσεις του τρόπου με τον οποίο τα δεδομένα ρέουν μέσα από ένα σύστημα. Τα πλεονεκτήματα της χρήσης των διαγραμμάτων ροής είναι η πληρέστερη κατανόηση του συστήματος, η επικοινωνία ο εύκολος εντοπισμός προβλημάτων.

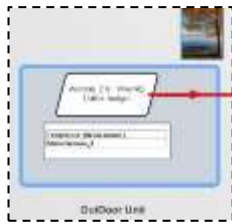
Τα δεδομένα που ανεβοκατεβαίνουν μεταξύ 'Γης και Ουρανού' μπορούν να κατηγοριοποιηθούν ανάλογα με το περιεχόμενό τους σε:

- Εντολές ελέγχου : Δεδομένα που αποστέλλονται από τον χρήστη προς το σύστημα, με σκοπό την ενεργοποίηση του φορτίου.
- Τιμές μετρήσεων από τα αισθητήρια.
- Δεδομένα που χρησιμοποιώντας για την ανίχνευση της κατάστασης σύνδεσης του συστήματος με το δίκτυο.
- Παράμετροι λειτουργίας του συστήματος π.χ. τιμή επιθυμητής θερμοκρασίας, τάση λειτουργίας φορτίου, τιμή Kwh, κλίμακες κόστους.

Μπορούν να κατηγοριοποιηθούν ανάλογα με το σε ποιο server αποστέλλονται στον ThinkSpeak ή στον HiveMQ.

Τέλος, διακρίνονται σε αυτά που ανεβαίνουν και σε αυτά που κατεβαίνουν. Αυτά που στέλνονται από το γη στο cloud αναφέρονται ως UpLiNK αντί για τον όρο publish που συνήθως χρησιμοποιείται. Αυτά που λαμβάνονται από το cloud αναφέρονται ως DownLink αντί για subscribed. Οι όροι UpLiNK και DownLink χρησιμοποιούνται περισσότερο στις ασύρματες τηλεπικοινωνίες, ένα γενικό πλαίσιο μέσα στο οποίο ανήκει το IoT, προτιμήθηκαν επειδή είναι ευρέως γνωστοί και δόκιμοι στο engineering. Εναλλακτικά Θα μπορούσαμε να τα βαφτίζαμε transmit και receive αντίστοιχα.

Στους παρακάτω πίνακες παραθετονται όλες οι μεταβλητές που λαμβάνουν χώρα στην υλοποίηση του συστήματος καθώς και λεπτομέρειες τους .Οι πίνακες είναι μεγέθυνση από το Block Διάγραμμα 2.



|                               |
|-------------------------------|
| <b>Wemos_2 Up link budget</b> |
| Temperature (Dallas sensor )  |
| Status Wemos_2                |

**Πίνακας 2 Τα δεδομένα που αποστέλλονται από την μονάδα OutDoor Unit .**

//-----



|                               |
|-------------------------------|
| <b>Wemos_1 Up link budget</b> |
| Feed Back                     |
| Analog Value (Current Value ) |
| Status Wemos_1                |
| Tempeprature (Dallas sensor ) |

**Πίνακας 3 Τα δεδομένα που αποστέλλονται Wemos\_1.**

//-----



**Πίνακας 4:Τα δεδομένα που λαμβάνονται από την InDoor Unit Wemos\_Command**

//-----



| <b>Virtuino Up link budget to HiveMQ ( Topics )</b> | <b>Refer Panel</b>  |
|---|---|
| Set point value                                     | Thermostat Panel  |
| Current Consumption                                 | Load Current Consumption ,Scale analog to Current Panel (Hidden ) |
| Final Hysterisis value                              | Thermostat Panel  |
| Temperature Outdoor (Dallas sensor )                | OutDoor Temperature panel ,Differential (Hidden )                 |
| Programmable Timer bit                              | Control Panel ,Programmable timer                                 |
| Overcurrent Limit value                             | Setting Hyperparameter Panel                                      |
| Force off bit                                       | Control Panel   |
| Thermostat status bit                               | Control Panel   |
| Operation Voltage value                             | Setting Hyperparameter Panel                                      |
| Current limit value                                 | Setting Hyperparameter Panel                                      |
| Force on bit  | Control Panel   |

**Πίνακας 5 Τα δεδομένα που αποστέλλονται από το Virtuino στην πλατφόρμα ανάλυσης ThingSpeak**

//-----



| Virtuino UP link budget to ThingSpeak                       | Refer Panel               | Channel - Field Id          |
|---|---------------------------|-----------------------------|
| Load Status bit   | Outdoor Temperature Panel | Monitor Channel-Field Id 1  |
| Total Energy Consumption value                              | Energy Data Logger        | Monitor Channel-Field Id 2  |
| Outdoor Temperature value                                   | Outdoor Temperature       | Monitor Channel-Field Id 3  |
| Load Current Consumption value                              | Load Current Consumption  | Monitor Channel-Field Id 4  |
| Indoor Temperature value                                    | Indoor Temperature        | Monitor Channel-Field Id 5  |
| M_80 Time form Pop Up Time -Date picker (number )           | Scaling Energy App        | Request DBMS - Field Id 6   |
| Command ON_OFF sending by Virtuino                          | Control Panel             | Common Channel - Field Id 1 |
| React Character sending from Virtuino to trigger Matlab app | Calling Energy App        | Request DBMS - Field Id 8   |

**Πίνακας 6: Τα δεδομένα που αποστέλλονται από το Virtuino στον MQTT broker HiveMQ**

//-----



| Virtuino Down link budget From HiveMQ | Refer Panel   |
|---------------------------------------|---|
| Feed Back                             | Control Panel   |
| Temperature Indoor (Dallas sensor )   | Indoor Temperature Panel  |
| Analog Value                          | Scale analog to current (Hidden Panel)  |
| Wemos_1 Status                        | Status of clients Wemos LoLIN 32 Panel ,Status Algorithm declaration (hidden) |
| Wemos_2 Status                        | Status of clients Wemos LoLIN 32 Panel  |

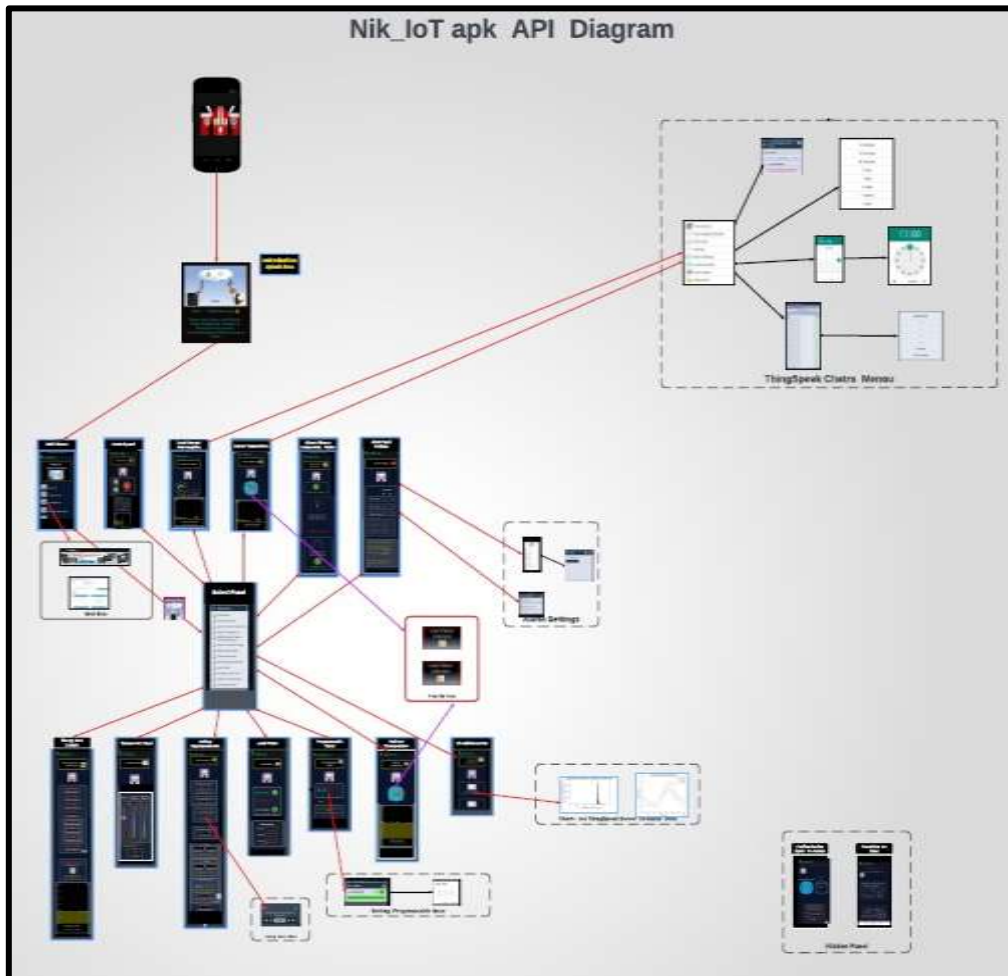
**Πίνακας 7: Τα δεδομένα που λαμβάνονται στο Virtuino από τον MQTT broker HiveMQ**

//-----

### 3 The NiK\_IoT API.

#### 3.1 Εισαγωγή-Γενική περιγραφή της εφαρμογής API NiK\_IoT.

Στο κεφάλαιο αυτό θα παρουσιάσουμε την εφαρμογή Application Programming Interface (API) η οποία υποστηρίζει όλο το σύστημα και ονομάζεται και αυτή NiK\_IoT. Πολλές φορές με τον ορο NiK\_IoT αναφερόμαστε στο όλο σύστημα διαχείρισης ενώ άλλες φορές μόνο στην εφαρμογή API. Η εφαρμογή αποτελείται από 13 λειτουργικά Panel και 2 κρυφά (hidden). Στο παρακάτω διάγραμμα εικόνα φαίνεται η συσχέτιση των πάνελ και πως πλοηγουμαστε.



**Block Διάγραμμα 3: NiK\_IoT API Diagram**

Το Apk της εφαρμογής NiK\_IoT είναι διαθέσιμο από το παρακάτω link  
<https://drive.google.com/file/d/1EOAvCr4pOTJrpdRvKGXd2t0LjOouuezZ/view?usp=sharing>

Στο παρακάτω link δίνεται το παραπάνω Block Διάγραμμα σε μορφή pdf :  
<https://drive.google.com/file/d/1wN-NL4Q2c85fiSCQada57JA56e-SsnfX/view?usp=sharing>

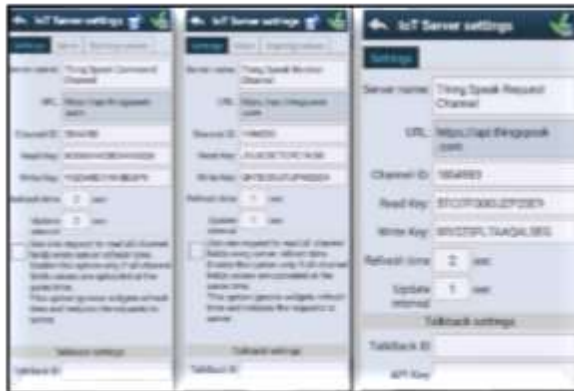
### 3.2 Τα 'panels' των Servers.

Όπως προαναφέρθηκε, η ανταλλαγή μηνυμάτων γίνεται μέσω Servers. Πάνω αριστερά στην οθόνη [Εικόνα 15](#) υπάρχει ένδειξη καταστάσεων των Servers που υποστηρίζουν την εφαρμογή. Από αριστερά προς τα δεξιά διακρίνουμε τους εξής Servers:

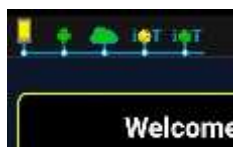
1. Τον emulator Server, ο οποίος είναι ουσιαστικά εκεί που γίνονται όλοι οι τοπικοί υπολογισμοί και αντιστοιχεί στις εσωτερικές διεργασίες και στην μνήμη της εφαρμογής.
2. Τον Server MQTT broker, ο οποίος αντιστοιχεί στον HiveMQ Server.
3. Τον Server που αντιστοιχεί στο κανάλι Monitor Channel του ThingSpeak.
4. Τον Server που αντιστοιχεί στο κανάλι Command Channel του ThingSpeak.

Στην [Εικόνα 16](#): IoT Servers settings over NiK\_IoT API φαίνονται τα Setting των ThingSpeak Servers οι ενδείξεις αυτές παίρνονται μόνο σε Unlock Panel, δηλαδή σε programming mode.

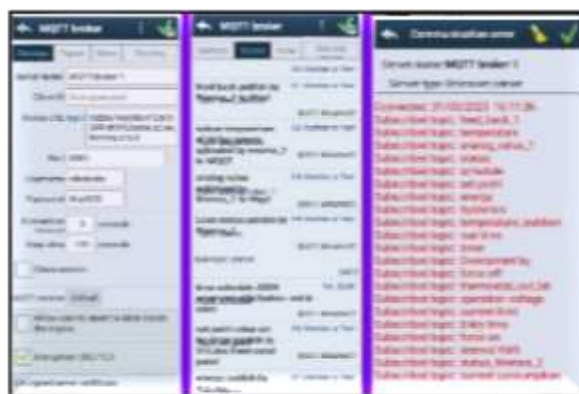
Στην [Εικόνα 17](#) αντίστοιχα φαίνονται από αριστερά προς τα δεξιά τα Setting του MQTT broker, τα topics τα οποία δηλώνει η εφαρμογή (τα οποία θα πρέπει να είναι ακριβώς ίδια με αυτά που στέλνει π.χ. το Wemos\_1 για να γίνει η επικοινωνία). Η εφαρμογή δίνει επίσης μηνύματα για τη λειτουργία της σύνδεσης.



**Εικόνα 16: IoT Servers settings over NiK\_IoT API.**



**Εικόνα 15: Servers που υποστηρίζουν την εφαρμογή NiK\_IoT.**



**Εικόνα 17 MQTT Server settings over NiK\_IoT API.**

### 3.3 Η Εκκίνηση της Εφαρμογής.

Η εκκίνηση της εφαρμογής γίνεται επιλέγοντας είτε κατευθείαν το εικονίδιο της εφαρμογής Nik\_IoT είτε επιλέγοντας την εφαρμογή Virtuino και κατόπιν φορτώνοντας το sketch NiK\_IoT. Επιλέγοντας τον δεύτερο τρόπο, μπορούμε να έχουμε πρόσβαση σε όλο το sketch ακόμα και στα κρυμμένα panels. Μπορούμε επίσης να επιλέξουμε το Unlock panel και να μπούμε σε programming mode και να κάνουμε αλλαγές στον κώδικα. Με την εφαρμογή Virtuino Viewer μπορούμε να τρέξουμε κανονικά το πρόγραμμα χωρίς όμως να μπούμε σε programming mode. Ένας κοινός χρήστης, ο οποίος δεν επεμβαίνει στον κώδικα, είναι αυτό που χρειάζεται. Το αρχείο Apk Nik\_IoT δημιουργήθηκε από τον προγραμματιστή του Virtuino, Ηλία Λάμπρου, και τον ευχαριστώ θερμά για τη συμβολή του. Μέσω της εφαρμογής



**Εικόνα 18 Εκκίνηση και άνοιγμα της Εφαρμογής NiK\_IoT API.**

Virtuino μπορεί να κατασκευαστεί πλήρως μια custom εφαρμογή. Στην εικόνα 13 βλέπουμε το splash screen image της δικής μας εφαρμογής καθώς και το welcome message. *’Τίποτα στην Τροία –« Ένα Είδωλο , μιας Πεταλούδας τίναγμα , το πούπουλο ενός κυκνου. Για ένα πουκάμισο αδειανό για μιαν Ελένη »* [16]

### 3.4 The Initial Panel Menu.

Η εφαρμογή ξεκινά με το Initial Menu, το οποίο περιλαμβάνει εννέα κουμπιά επιλογής που μπορούν να σε κατευθύνουν σε διάφορους ιστότοπους που σχετίζονται με την εφαρμογή. Αυτά τα hyperlink περιλαμβάνονται στο APK δίκην βιβλιογραφίας για όσους χρήστες επιθυμούν περισσότερες πληροφορίες αλλά δεν έχουν πρόσβαση στο γραπτό μέρος της διπλωματικής. Η Εικόνα 19 παρουσιάζει αυτές τις επιλογές στο μενού.



**Εικόνα 19:Initial Menu Of NiK\_IoT API.**

1. "Select Panel": Παρουσιάζει τα 13 λειτουργικά πάνελ της εφαρμογής. Μικρή αλλά σημαντική λεπτομέρεια για τους προγραμματιστές Το κουμπι αυτό είναι custom, δημιουργήθηκε μέσω του Virtuino.
2. "Connect to Servers": Επανεκκινεί τους Servers της εφαρμογής.
3. "HiveMQ Web page": Σας μεταφέρει στον ιστότοπο της HiveMQ.
4. "ThingSpeak Request Channel": Σας μεταφέρει στο public view του καναλιού Request Channel που υποστηρίζει η εφαρμογή .<https://thingspeak.com/channels/1854983>
5. "ThingSpeak Monitor Channel": Σας μεταφέρει στο public view του καναλιού Monitor Channel που υποστηρίζει η εφαρμογή.( Εικόνα 20 ) <https://thingspeak.com/channels/1994330>
6. "ThingSpeak Command Channel": Σας μεταφέρει στο public view του καναλιού Command Channel που υποστηρίζει η εφαρμογή .<https://thingspeak.com/channels/2064780>
7. "Virtuino Web page": Σας μεταφέρει στην ιστοσελίδα του Virtuino όπου μπορείτε να βρείτε περισσότερες πληροφορίες.
8. "My Personal Web page": Σας μεταφέρει στην προσωπική μου ιστοσελίδα. Ας ευλογήσουμε και λίγο τα γένια μας.  
<https://users.iewe.ihu.gr/~iewe2019241/site/index1.html?fbclid=IwAR2OozP0cpCA88dbm1xr3Lkr3Qx6gy1RavUdvYu0nT9-k30hnp2ej3CjRJM>
9. "Quit": Έξοδος την εφαρμογή.



**Εικόνα 20 :Public view του Monitor Channel over ThingSpeak.**

### 3.5 The Control Panel.

Στο Control Panel ( Εικόνα 16 ) το κεντρικό χειριστήριο για τον έλεγχο του φορτίου και την απεικόνιση των καταστάσεων του. Για να βελτιώσουμε την κατανόηση και την διάδραση του ελέγχου υπάρχει το automation circuit ( πάνω δεξιά Εικόνα 16 ).Το φορτίο ενεργοποιείται μέσω ενός λογικού κυκλώματος ( Automation Circuit ) από πέντε εισόδους ( input )

- Force ON Button : Εντολή από τον χρήστη
- Wi-Fi Thermostat :Εντολή από τον θερμοστάτη ( [Thermostat Panel](#) )
- Programmable Timer: Εντολή από τον χρονοδιακόπτη( [Programmable Timer Panel](#) )
- Force OFF Button: Εντολή από τον χρήστη

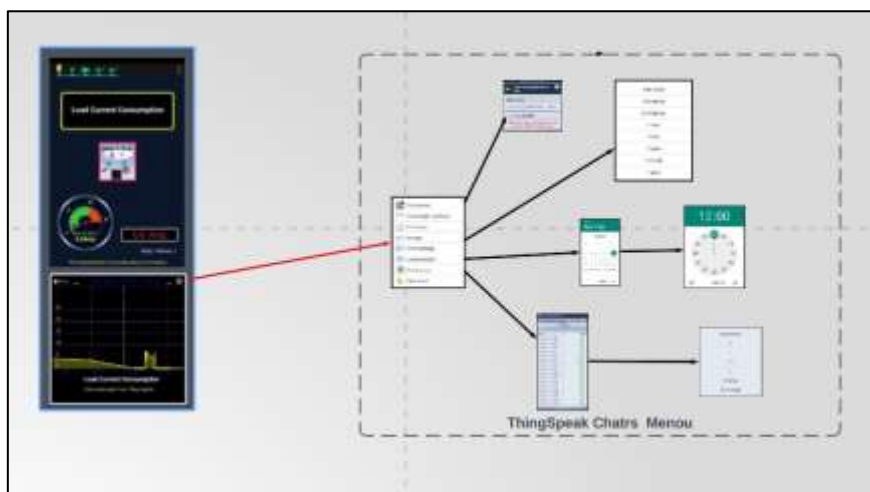
- Overcurrent Detect: Ανίχνευση υπερέντασης.



Εικόνα 16: Control Panel Of NiK\_IoT API

Οι τρεις πρώτες είναι σε συνδεσμολογία (Gate OR) και ενεργοποιούν το φορτίο. Ενώ οι τελευταίες δυο το απενεργοποιούν (Gate AND). Η εντολή Force OFF έχει προτεραιότητα και μπορεί να απενεργοποιήσει το φορτίο ανεξαρτήτως της κατάστασης της εισόδου. Στο κάτω μέρος του panel υπάρχει ένα γράφημα των καταστάσεων της εξόδου του φορτίου. Με αυτόν τον τρόπο, μπορούμε να έχουμε πρόσβαση σε ολόκληρο το ιστορικό της λειτουργίας του φορτίου και να αξιολογήσουμε την απόδοσή του.

### 3.6 Load Current Consumption Panel



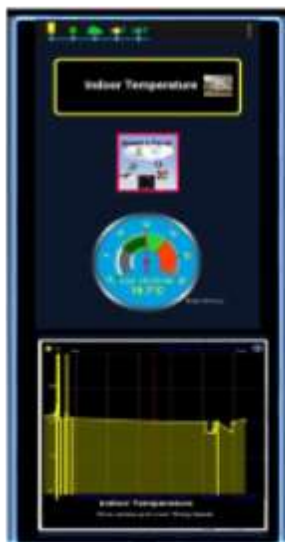
Εικόνα 21: Load Current Consumption Panel Of NiK\_IoT API

Στο panel αυτό έχουμε την ένδειξη από την τιμή της έντασης του ρεύματος του φορτίου. Η ένδειξη δίνεται και σε αναλογική και σε ψηφιακή μορφή. Στο Panel υπάρχει η ένδειξη του οργάνου (analog instrument), οι μονάδες μετρήσεις καθώς και η ημερομηνία και η ώρα που έγινε η τελευταία μέτρηση. Στο κάτω μέρος της οθόνης δίνεται το times series graf μέσω του ThingSpeak Server. Με την επιλογή chart (βλέπε Εικόνα 21) δεξιά πλαίσιο μπορούμε να επιλέξουμε χειροκίνητα (manual) από ποια χρονική στιγμή και μετά θέλουμε να διαβάσουμε τις τιμές π.χ. μπορούμε να επιλέξουμε της τελευταίας ώρας, τελευταίας μέρας, εβδομάδας ή του τελευταίου έτους. Επίσης, δίνεται η δυνατότητα στον χρήστη να κατεβάζει μέσες (average) τιμές ή όχι, δηλαδή ο χρήστης να επιλέγει π.χ. τον μέσο όρο των μετρήσεων ανά 15 λεπτά ή ανά 1 ώρα. Με αυτόν τον τρόπο ελαχιστοποιούμε τα σφάλματα που δημιουργούνται από αστοχία των αισθητήρων και κάνουμε εξοικονόμηση χώρου μνήμης στο κινητό και έχουμε λιγότερο όγκο δεδομένων για επεξεργασία. Μπορούμε επιπρόσθετα να πάρουμε τις τιμές αυτές σε πίνακα σε αντιστοιχία με τον χρόνο (time stamp). Υπάρχει φίλτρο τιμών έτσι ώστε να τις

επεξεργαστούμε πάνω στην εφαρμογή ή να τον εξάγουμε σε υπολογιστικό φύλλο OfficeXL για λεπτομερέστερη ανάλυση off line . Η εφαρμογή δίνει την δυνατότητα για απλή επεξεργασία δεδομένων από τον απλό χρήστη.

### 3.7 Indoor Temperature Panel.

Στο panel έχουμε αναλογική ένδειξη από την εσωτερική θερμοκρασία της οικίας, καθώς και γράφημα της (time series graph). Η εξωτερική θερμοκρασία διαβάζεται μέσω της InDoor Unit και του αισθητήρα DS18B20. Ας κάνουμε εδώ μια παρατήρηση σε σχέση με τα προλεγόμενα. Είχαμε πει στην



Εικόνα 22: Indoor Temperature Panel Over NiK-IoT API

ενότητα 1.7.1 ότι το Virtuino είναι μια ενιαία πλατφόρμα στην οποία ενσωματώνονται δεδομένα από διαφορετικούς Servers με διαφορετικά πρωτόκολλα επικοινωνίας και διαφορετικές δομές δεδομένων. Στο InDoor Temperature panel της η αναλογική ένδειξη βλέπει τον HiveMQ Server, ενώ το γράφημα βλέπει τον ThingSpeak. Αν για κάποιο λόγο η σύνδεση του Wemos\_1 χαθεί, το Virtuino σταματά να επαναεκπέμπει τα δεδομένα στον ThingSpeak και πετάγεται Banner που ενημερώνει τον ανυποψίαστο χρήστη.

### 3.8 Wemos\_Esp32Clients Connection Status Panel.

Το panel "Wemos\_Esp32Clients Connection Status", παρέχει πληροφορίες σχετικά με τη σύνδεση Android ή του σπιτιού με το cloud. Σύμφωνα με το [Block Διάγραμμα 1](#), για να λειτουργήσει το σύστημα θα πρέπει να υπάρχει επικοινωνία του χρήστη με το cloud και του σπιτιού με το cloud. Ενώ σχετικά εύκολο ο χρήστης 'Σουμελα' να πιστοποιήσει εάν το κινητό της είναι συνδεδεμένο στο Internet και εάν έχει επικοινωνία με τους Servers, (βλέπε Εικόνα 15) δεν μπορεί να γνωρίζει τι συμβαίνει στο 'Σπίτι στη Χαλκιδική'. Υπάρχει η πιθανότητα το "Wemos\_1" να είναι εκτός σύνδεσης και να ματαιοπονεί αδίκως



Εικόνα 23: Wemos\_Esp32 Clients Connection Status Panel Of NiK\_IoT API

- Είτε λόγω αστοχίας σύνδεσης του Wemos (θα αναφερθούμε εκτενέστερα σε αυτό το θέμα στο τελευταίο κεφάλαιο).
- Είτε λόγω έλλειψης ηλεκτρικής παροχής.
- Είτε λόγω προβλημάτων αποσυνδέσεων του τοπικού πάροχου ISP.

Βέβαια δεν μπορούμε ξέρουμε για ποιον λόγο δεν έχουμε σύνδεση. Αλλά τουλάχιστον αυτή η πληροφορία μας αποτρέπει να έχουμε προσδοκίες από ένα σύστημα που δεν δουλεύει. Το τέχνασμα για τον έλεγχο της σύνδεσης Wemos\_1 – internet είναι το παρακάτω Όπως φαίνεται στο Πίνακα 3 Τα δεδομένα που αποστέλλονται Wemos\_1 καθώς και στον κωδικα (βλεπε Παράρτημα 3 Arduino Code of [Wemos\\_1\\_INDOOR\\_MQTT](#)) το Esp32 στέλνει κάθε 1 δευτερόλεπτο ένα τυχαίο νούμερο από το 1 έως το 100 στο broker. Αυτό αντιστοιχεί με την καρδιά του Wemos\_1 εάν σταματήσει το Virtuino να βλέπει αυτόν κτύπο σημαίνει ότι το Wemos\_1 is dead και άρα το 'Σπίτι στη Χαλκιδική' είναι off line.

Στην Εικόνα 23 αριστερά διακρίνονται μέσω της εφαρμογής Android *Fing - Network Scanner* τα τρία Wemos του NiK\_IoT συνδεδεμένα στο τοπικό δίκτυο Wi-Fi. Στο panel **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** εμφανίζεται σε αναλογική ένδειξη ο « κτύπος » αυτός (βλέπε **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** δεξιά). Καθώς και ψηφιακή ένδειξη για την παρουσία της σύνδεσης, λογικό κύκλωμα δείχνει τη συνολική κατάσταση σύνδεσης υπολογίζοντας και την κατάσταση σύνδεσης μεταξύ του Virtuino και του MQTT broker. Το ίδιο πραγματοποιήθηκε για το Wemos\_2. Για το Wemos\_Command δεν έγινε αυτή πρόβλεψη. Στο κεφάλαιο τροποποιήσεις και επεκτάσεις, θα προτείνουμε λύσεις στο πρόβλημα των αποσυνδέσεων όταν αυτά οφείλονται στην δυσλειτουργία του Wemos. Στην Εικόνα 39 βλέπουμε το μήνυμα (banner) που εμφανίζεται και ενημερώνει τον χρήστη ακόμα και όταν βρίσκεται σε άλλα πάνελ ότι η σύνδεση μεταξύ του υπολογιστή Wemos\_1 και του δικτύου έχει χαθεί.



### 3.9 Alarms Panel Menu.

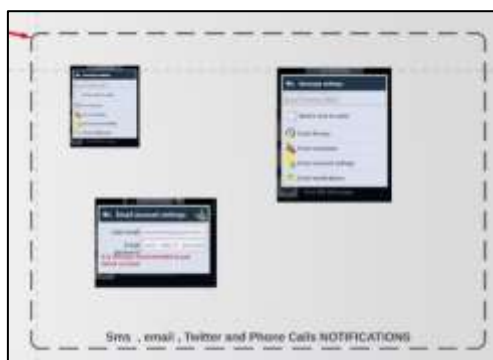
Η εφαρμογή VIRTUINO δίνει την επιλογή να ορίσουμε alarms, όταν μια τιμή είναι πάνω ή κάτω από ένα όριο. Ο χρήστης μπορεί να ορίσει τα μέγιστα και ελάχιστα κατώφλια ( max ή min) καθώς και να γράψει το δικό του προσωπικό μήνυμα όταν εμφανίζεται κάθε ειδοποίηση.



**Εικόνα 24: Alarms Panel Of NiK\_IoT API**

Υπάρχει η δυνατότητα να τα απενεργοποιούμε, εάν θέλουμε. Στη εικόνα Εικόνα 24 το πάνελ της εφαρμογής ενσωματώνει αυτήν λειτουργία καθώς και αντίστοιχα panel την ρύθμιση των κατωφλίων. Να υπενθυμίσουμε ότι οι ειδοποιήσεις θα στέλνονται μόνο όταν είναι ανοικτή η εφαρμογή που συνήθως δεν είναι .Ο κοινός χρήστης συνήθως ανοίγει την εφαρμογή και μετά την κλείνει .οπότε αυτό το panel δεν έχει μεγάλη λειτουργικότητα στα χέρια του client.

Εναλλακτικά οι ειδοποιήσεις στον χρήστη θα έρχονται μέσω του Virtuino-server' .στο οποίο μπορούμε να το ρυθμίσουμε ώστε να στέλνει ειδοποιήσεις μέσω SMS , τηλεφωνικών κλήσεων ή tweets. Η δυνατότητα αυτή είναι διαθέσιμη μόνο στην έκδοση του Virtuino που διατίθεται από την πλατφόρμα της εφαρμογής και όχι στην έκδοση από το Playstore. Στην Εικόνα 25 βλέπουμε τα πάνελ όπου εισαγάγουμε το email account από το οποίο θα στείλουμε τα ειδοποιητήρια email s, τα email των παραληπτών. Αντίστοιχες ρυθμίσεις γίνονται όταν επιλέγουμε ειδοποιήσεις μέσω τηλεφωνικών κλήσεων. Να σημειώσουμε ότι παρόμοια δυνατότητα αποστολής ειδοποιήσεων έχει και η πλατφόρμα ThingSpeak



**Εικόνα 25 :SMS , Twitter and Phones Calls setting panel**

### 3.10 Thermostat Panel.

Στο πάνελ του ηλεκτρονικού θερμοστάτη Εικόνα 26 ο χρήστης μπορεί να εισάγει την επιθυμητή θερμοκρασία μέσω του slider αριστερά η οποία αναγράφεται ως set point. Στη μέση του πάνελ, εμφανίζεται η τρέχουσα εσωτερική θερμοκρασία του σπιτιού (Real Value), ενώ στα αριστερά εμφανίζεται η τρέχουσα εξωτερική θερμοκρασία (Outside temperature) . Επίσης, υπάρχει ένδειξη για την κατάσταση του θερμοστάτη, εάν είναι ενεργοποιημένος ( ON ) ή όχι (OFF) .Η λειτουργία του θερμοστάτη είναι με μεταβλητή υστέρηση Ο θερμοστάτης με μεταβλητή υστέρηση είναι μια εξέλιξη του κλασικού θερμοστάτη που χρησιμοποιείται στα συστήματα κλιματισμού και θέρμανσης. Η βασική λειτουργία είναι να ελέγχει τη θερμοκρασία του χώρου και να επιτυγχάνει τη θερμοκρασία που έχει οριστεί από τον χρήστη.

Η μεταβλητή υστέρηση είναι μια παράμετρος που προστίθεται στη λειτουργία του



Εικόνα 26: Thermostat Panel Of NiK\_IoT API

θερμοστάτη και επιτρέπει την πιο αποτελεσματική διαχείριση της θερμοκρασίας στον χώρο. Όταν ο θερμοστάτης ανιχνεύει ότι η θερμοκρασία στον χώρο έχει φτάσει στην επιθυμητή τιμή, δεν σταματά αμέσως να λειτουργεί, αλλά συνεχίζει και να κάνει μικρές παρεκκλίσεις από την επιθυμητή τιμή. Η λογική της σχεδίασης είναι η εξής: εάν η θερμοκρασία έξω πέφτει, τότε κράτα λίγο ακόμα «αναμμένος». Εάν ο θερμοστάτης δεν είναι ενεργοποιημένος, η παράμετρος της μεταβλητής υστέρηση δεν εισέρχεται στον αλγόριθμο ελέγχου. Πιο αναλυτικά, η υστέρηση (Final Hysteresis) αποτελείται από δύο όρους (factors):

- Έναν σταθερό (Constant Hysteresis) όρο, τον οποίο εισάγει ο χρήστης από το [Setting Hyperparameter Panel](#) και
- Έναν δυναμικό –μεταβλητό όρο που ισούται με την διαφορά της τρέχουσας εξωτερικής θερμοκρασίας με αυτήν την εξωτερική θερμοκρασία πριν από ένα λεπτό.

Final Hysteresis = Constant Hysteresis + Last-minute difference of Outside Temperature

Error signal = Set point (επιθυμητή θερμοκρασία αριστερό slider) - [Real value (Indoor temperature value) + Final Hysteresis]

Εάν το Error signal είναι θετικό, τότε ο θερμοστάτης τίθεται ON, διαφορετικά είναι OFF. Ας υποθέσουμε ότι το σύστημα NiK\_IoT έχει προτοποθετημένο 0,5 ° C βαθμούς Constant Hysteresis και τα τελευταία είκοσι λεπτά η εξωτερική θερμοκρασία έχει πέσει τρεις βαθμούς. Η εσωτερική θερμοκρασία είναι 15 βαθμούς Celsius και ο χρήστης επιλέγει επιθυμητή θερμοκρασία 20 βαθμούς .Ο θερμοπομπός ανάβει και η εσωτερική θερμοκρασία αρχίζει να ανεβαίνει .Ποτέ θα κλείσει ;. Θα κλείσει σύμφωνα με την παρακάτω εξίσωση όταν το error signal γίνει αρνητικό

$$\text{Error signal} = 20 + [\text{Real value (Indoor temperature Value)} + 0,5 + 3]$$

Αρα σύμφωνα με την παραπάνω εξίσωση ο θερμοστάτης θα απενεργοποίηση το φορτίο ( βλέπε [Load Current Consumption Panel](#)) όταν η θερμοκρασία ξεπεράσει τους  $20+0,5+3=23,5$  βαθμούς. Ένας απλός θα απενεργοποιούσε το σύστημα όταν η θερμοκρασία δωματίου ξεπερνούσε τους 20 °C

Με τον παραπάνω τρόπο δίνουμε στο σύστημα θέρμανσης μια δυνατότητα αντιστάθμισης των μεταβολών της εξωτερικής θερμοκρασίας αντιστάθμιση μπορεί να βοηθήσει στη διατήρηση μιας σταθερής θερμοκρασίας στο σπίτι, ακόμα και όταν οι καιρικές συνθήκες αλλάζουν. Αυτό μπορεί να βελτιώσει την συνολική άνεση και να μειώσει την ανάγκη για χειροκίνητες ρυθμίσεις του θερμοστάτη. Στην παρούσα κατάσταση έχει ρυθμιστεί ώστε η μεταβλητή υστέρηση να λαμβάνει χώρα μόνο όταν ο θερμοστάτης είναι ενεργοποιημένος. Αυτό βεβαίως μπορεί να αλλάξει μέσω επαναπρογραμματισμού. Υπάρχουν πολλές δυνατότητες για παραλλαγές στη σχεδίαση του αυτοματισμού θέρμανσης και η υστέρηση μπορεί να λαμβάνει χώρα ακόμα και όταν η έξοδος του θερμοστάτη είναι OFF. Σε αυτήν την περίπτωση θα πρέπει να τοποθετηθεί ένας επιπλέον διακόπτης software λειτουργίας του θερμοστάτη, έτσι ώστε να μην ενεργοποιείται το σύστημα θέρμανσης όταν δεν το επιθυμεί ο χρήστης, αλλά η εξωτερική θερμοκρασία μειωθεί. Ο χρόνος 1 λεπτό, όπου θα λαμβάνεται η μέτρηση της διαφορά της εξωτερικής θερμοκρασίας, επιλέχθηκε κυρίως για πρακτικούς σκοπούς κατά τη διάρκεια των δοκιμών και κατά την επίδειξη . Στην πράξη, επειδή οι μεταβολές στην εξωτερική θερμοκρασία δεν είναι τόσο απότομες (διάρκειας ενός λεπτού), ίσως χρονικό διάστημα 15 λεπτών να είναι καταλληλότερη επιλογή που θα οδηγούσε σε αποδοτικότερο σύστημα. Αυτό βέβαια έχει να κάνει και με τις ιδιαίτερες τοπικές καιρικές συνθήκες, καθώς και τη θερμομόνωση του σπιτιού το κάθε πότε θα θέλουμε να μετράμε την εξωτερική θερμοκρασία θα το ξαναεξετάσουμε στην ενότητα [Outdoor](#) .

### 3.11 Energy Data Logger Panel

Στο πάνελ αυτό έχουμε τη λειτουργία του μετρητή Εικόνα 27. Η κύρια λειτουργία του πάνελ είναι η μέτρηση και η καταγραφή της ενέργειας. Πατήστε το κουμπί στο κάτω μέρος και η εφαρμογή θα αρχίσει να καταγράφει την ενέργεια που καταναλώνεται από τη στιγμή που πατήθηκε.



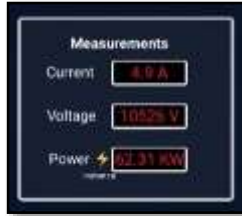
Εικόνα 27: Energy Data Logger Panel over NiK\_IoT API

1. Στο πρώτο πεδίο "Total Energy Consumption" φαίνεται η συνολική κατανάλωση. Για να μηδενιστεί αυτό το πεδίο, πρέπει να μηδενιστούν όλα τα δεδομένα στο κανάλι "Channel Monitor" στο ThinkSpeak. Δεν μπορεί να γίνει αυτή η ενέργεια από την εφαρμογή.
2. Στο πεδίο "Last Min Energy Consumption" φαίνεται η στιγμιαία κατανάλωση ενέργειας. Θεωρείται ότι η κατανάλωση είναι σταθερή κατά τη διάρκεια ενός λεπτού. Αυτή η κατανάλωση είναι η βάση μέτρησης όλων των αλγορίθμων της ενέργειας.

3. Στο πεδίο "Time-Space Energy Cost" φαίνεται το συνολικό κόστος κατανάλωσης σε ευρώ από την στιγμή που πατήθηκε το κουμπι . Η τιμή αυτή υπολογίζεται ως το άθροισμα τριών παρακάτω πεδίων:
  - 'Cost of Νυκτερινό Τιμολόγιο' Το Κόστος της ενέργειας που χρεώθηκε σύμφωνα με το Νυκτερινό τιμολόγιο σε Ευρώ
  - 'Κόστος Τιμολογίου Κλίμακας 1': Το κόστος της ενέργειας που χρεώθηκε σύμφωνα με την πρώτη κλίμακα, σε ευρώ.
  - 'Κόστος Τιμολογίου Κλίμακας 2': Το κόστος της ενέργειας που χρεώθηκε σύμφωνα με τη δεύτερη κλίμακα, σε ευρώ.
  
4. Στο πεδίο 'Τρέχον Κόστος ανά Κιλοβατόρα': Φαίνεται ο τρέχων συντελεστής κόστους ανά Κιλοβατόρα, δηλαδή με πόσα ευρώ θα χρεωθεί η τρέχουσα Κιλοβατόρα.

### 3.11.1 Επίδειξη μετρήσεων στο Energy Data logger Panel

Στην ενότητα αυτή θα προσπαθήσουμε να σας παρουσιάσουμε το panel σε λειτουργία. Τοποθετήσαμε ένα ηλεκτρικό φορτίο σε λειτουργία ποιο συγκεκριμένα ένα ηλεκτρικό καλοριφέρ το οποίο τραβούσε 5 Amere. Για τις ανάγκες τις επίδειξης αλλάξαμε την τάση λειτουργίας έτσι ώστε η 'ισχύς' να είναι 62 KW βλέπε Εικόνα 28. Έτσι η ανά λεπτό η



**Εικόνα 28 :** Ηλεκτρική κατανάλωση κατά την διάρκεια της Επίδειξης

ενεργειακή κατανάλωση θα είναι 1 KWh. Καταλληλά προσαρμόστηκαν οι τιμές το κατόπλι της πρώτης κλίμακας τέθηκε 1 KWh το κόστος της κιλοβατώρας είναι 1 ευρώ για την πρώτη κλίμακα 2 ευρώ για την και 0,5 για το νυκτερινό . βλέπε Εικόνα 29



**Εικόνα 29 :** Παράμετροι της τιμολογιακής ενεργειακής κατανάλωσης

- Ωρα 11.24 ( βλέπε την ένδειξη πάνω αριστερά Εικόνα 30 ). Το panel αρχίζει τη μέτρηση



**Εικόνα 30 :** Αρχικοποίηση της μέτρησης

- Ωρα 11:25 μετα από παρέλευση 1 λεπτού Πρώτη κλίμακα χρέωσης βλέπε Εικόνα 31 τον συντελεστή χρέωσής



Εικόνα 31:Επιδειξη λειτουργίας Πρώτη κλίμακα Χρέωσης

- Ωρα 11:26 έχουμε ξεπεράσει την πρώτη κλίμακα και χρεωνόμαστε με την δεύτερη 2 ευρώ /KWh Εικόνα 32



Εικόνα 32 Δεύτερη κλίμακα Χρέωσης

- Ωρα 11:27 και η βρόχα έπεφτε... right through Εικόνα 33 Καταγραφή κόστους μετα την ενεργοποίηση του νυκτερινού τιμολόγιου Σε ένα λεπτό έχω καταναλώσει και πάλι 1 KWh η χρέωση είναι 0,5 Ευρω /KWh \* 1KWh=0,5 Ευρω

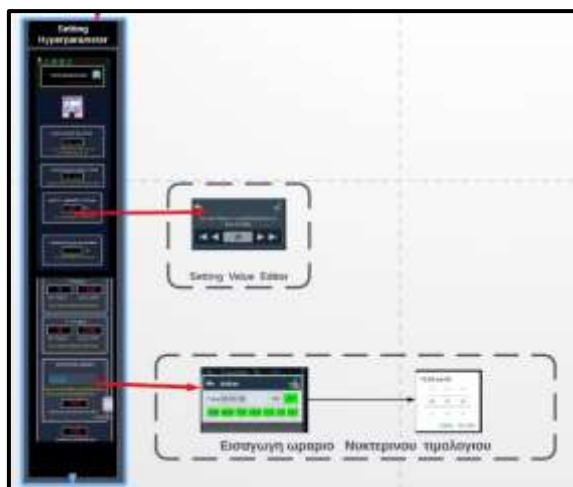


Εικόνα 33 : Επίδειξη λειτουργίας Νυκτερινό τιμολόγιο χρέωσης.

### 3.12 Setting Hyperparameter Panel.

Στο panel αυτό Figure 1: Setting Hyperparameter ο χρήστης εισάγει τιμές παραμέτρων στο σύστημα. Υπάρχει η δυνατότητα εισαγωγής κωδικού για την αλλαγή των τιμών των Value Display Οι παράμετροι αυτοί είναι:

1. Load Current up Limit Consumption είναι το ανώτατο όριο της έντασης που διαρρέει το φορτίο το εισάγει ο χρήστης ή καλύτερα ο administrator στο 'Virtuino server' ανάλογα τα χαρακτηριστικά του φορτίου που θα τοποθετήσει. Όταν η ένταση του ρεύματος ξεπεράσει αυτήν την τιμή το σύστημα απενεργοποιεί το ρελέ.
2. Load Current Down limit είναι το κάτω όριο της έντασης που διαρρέει το ηλεκτρικό φορτίο κάτω από αυτήν την τιμή το φορτίο θεωρείται ότι δεν διαρρέεται από ρεύμα βλέπε [Load status Panel](#).



**Figure 1: Setting Hyperparameter Panel over NiK-IoT API.**

Οι επόμενες 4 παράμετροι αφορούν τη χρέωση του ηλεκτρικού ρεύματος. Ένα κλιμακούμενο τιμολόγιο ηλεκτρικού ρεύματος είναι ένα σύστημα χρέωσης που βασίζεται στην ποσότητα της καταναλωθέντος ηλεκτρικής ενέργειας Υπάρχει διαφορετική τιμή για κάθε επίπεδο κατανάλωσης, με την χαμηλότερη τιμή για την πρώτη κλίμακα και υψηλότερη τιμή για την δεύτερη κλίμακα. Το σύστημα που υποστηρίζει το NiK\_IoT έχει 2 κλίμακες Η εφαρμογή ενός κλιμακούμενου τιμολογίου ηλεκτρικής ενέργειας μπορεί να ενθαρρύνει τους καταναλωτές να μειώσουν την κατανάλωσή τους και να αναζητήσουν εναλλακτικούς τρόπους ενέργειας, όπως τη χρήση φωτοβολταϊκών πάνελ ,τη βελτίωση της μόνωσης του σπιτιού, χοντρά μπουφάν .Κάθε εταιρία παροχής ηλεκτρικής ενέργειας έχει την δική της ενεργειακή πολιτική τιμολόγησης η διπλωματική Βασίστηκε στο κυμαινόμενο προϊόν Γ1/Γ1N της ΔΕΗ (Δημοσιά !! Επιχείρηση Ηλεκτρισμού) βλέπε Εικόνα 34. Εφαρμόζεται ενιαία τιμή χρέωσης για την κατανάλωση ημέρας, το ύψος της οποίας διαφοροποιείται ανάλογα με το συνολικό ύψος της 4μηνιαίας κατανάλωσης. Τις ώρες μειωμένης χρέωσης (νυκτερινό ωράριο) η χρέωση ενέργειας πραγματοποιείται με μειωμένη τιμή για όσους διαθέτουν νυκτερινό μετρητή.

3. Kwh threshold: Το όριο της πρώτης κλίμακας.
4. Cost Euro/KWh ( 1 Κλίμακα ): Κοςτος κιλοβατώρας της πρώτης κλίμακας.
5. Cost Euro/KWh ( 2 Κλίμακα ): Κοςτος κιλοβατώρας της δεύτερης κλίμακας.

6. Cost Euro/KWh ( Νυκτερινή ) Κοςτος κλοβατωρας για το νυκτερινό τιμολόγιο.
7. Load Operation Voltage :Ταση λειτουργίας π.χ. για χρήστες στην Ευρώπη 230 Vac.
8. Hysterisis Parameter (Constant Hysteresis βλεπε 3.10).
9. Οι παράμετροι τιμολόγησης (pricing parameters) αφορούν την ώρα έναρξης και λήξης του νυχτερινού τιμολογίου και εισάγονται στο πάνελ ρυθμίσεων μέσω του προγραμματιζόμενου χρονοδιακόπτη (Programmable Timer) .Οταν η έξοδος του είναι ενεργοποιημένη εφαρμόζεται νυκτερινή χρέωση. Ο προγραμματισμός είναι εβδομαδιαίος και γίνεται εύκολα μέσω του αντίστοιχου widget. Να υπενθυμίσουμε ότι οι ώρες χρήσης του νυκτερινού ρεύματος αλλάζουν ανά εποχή και τοποθεσία [17]

| Χρεώσεις Προμήθειες |             |  |               |               |  |                            |               |   |               |  |
|---------------------|-------------|--|---------------|---------------|--|----------------------------|---------------|---|---------------|--|
| Πρόσδιο             | Ποσοστό (€) | Μηνιαίο Ενέργειας Πληρωτέα (χωρίς προσαυκτικό 100% για όλους τους χρήστες) |               |               | Ετήσιο Ενέργειας Πληρωτέα (χωρίς προσαυκτικό 100%) | Αποδοτικότητα Πληρωτέα Πάκ |               | Ετήσιο Κόστος Ενέργειας Πληρωτέα (χωρίς προσαυκτικό 100%) |               | Ετήσιο Κόστος Φόρτισης Πληρωτέα (χωρίς προσαυκτικό 100%) |
|                     |             | 0-1000 kWh   | 1000-2000 kWh | 2000-4000 kWh |  | 0-1000 kWh                 | 1000-2000 kWh | 0-1000 kWh  | 1000-2000 kWh |  |
| Μηνιαίο 300€        | 0,0         | 0,10000  | 0,20000       | 0,30000       | Προμήθεια 1000 kWh (χωρίς προσαυκτικό 100%)        | 0,00000                    | 0,00000       | 0,10000   | 0,20000       |  |
|                     |             |  |               |               | Προμήθεια 2000 kWh (χωρίς προσαυκτικό 100%)        | 0,00000                    | 0,00000       | 0,10000   | 0,20000       |  |
|                     |             |  |               |               | Προμήθεια 4000 kWh (χωρίς προσαυκτικό 100%)        | 0,00000                    | 0,00000       | 0,10000   | 0,20000       |  |
|                     |             |  |               |               | Προμήθεια 1000 kWh (χωρίς προσαυκτικό 100%)        | 0,00000                    | 0,00000       | 0,10000   | 0,20000       |  |
|                     |             |  |               |               | Προμήθεια 2000 kWh (χωρίς προσαυκτικό 100%)        | 0,00000                    | 0,00000       | 0,10000   | 0,20000       |  |
|                     |             |  |               |               | Προμήθεια 4000 kWh (χωρίς προσαυκτικό 100%)        | 0,00000                    | 0,00000       | 0,10000   | 0,20000       |  |
|                     |             |  |               |               | Προμήθεια 1000 kWh (χωρίς προσαυκτικό 100%)        | 0,00000                    | 0,00000       | 0,10000   | 0,20000       |  |
|                     |             |  |               |               | Προμήθεια 2000 kWh (χωρίς προσαυκτικό 100%)        | 0,00000                    | 0,00000       | 0,10000   | 0,20000       |  |
|                     |             |  |               |               | Προμήθεια 4000 kWh (χωρίς προσαυκτικό 100%)        | 0,00000                    | 0,00000       | 0,10000   | 0,20000       |  |

Εικόνα 34:Χρεώσεις στο Γ1/Γ1N προϊόν ηλεκτρικής ενέργειας [57]

### 3.13 Load status Panel.

Σε αυτό το πάνελ έχουμε μια εικόνα για την κατάσταση λειτουργίας του ηλεκτρικού φορτίου. Υπάρχουν δυο πλαίσια στο πρώτο πλαίσιο που υποδεικνύουν την κατάσταση ενεργοποίησης.

Στο πρώτο πλαίσιο έχουμε δυο ενδείξεις:

1. Εάν ο ηλεκτρονόμος είναι οπλισμένος. (Relay is armed) ή όχι. Αυτή η ένδειξη παίρνεται μέσω πραγματικής ανατροφοδότησης από το relay (ψηφιακή είσοδος) μέσω της δεύτερης κύριας επαφής του intermediate relay. Είναι κατά την γνώμη μου μια σημαντική διαφορά από τις άλλα παρόμοια συστήματα απομακρυσμένου ελέγχου.

2. Εάν το φορτίο είναι αποσυνδεδεμένο (Load disconnect). Αυτή η ένδειξη εμφανίζεται όταν το relay είναι οπλισμένο αλλά δεν υπάρχει κατανάλωση ρεύματος. Το όριο της έντασης του ρεύματος κάτω από το οποίο θα θεωρώ ότι το φορτίο είναι αποσυνδεδεμένο καθορίζεται από το [Setting Hyperparameter Panel](#)



**Εικόνα 35: Load status Panel Of NiK\_IoT API.**

Στο δεύτερό κάτω πλαίσιο έχουμε 3 value display με ηλεκτρικά μεγέθη:

1. Η τιμή της έντασης του ηλεκτρικού ρεύματος στο φορτίο η τιμή αυτή παίρνεται μέσω του [Scale Analog to Current Value \(Hidden\)](#).
2. Η τιμή της ηλεκτρικής τάσης (θεωρείται σταθερή κατά συμβαση και είναι ίση με την τιμή που της δόθηκε στο Setting Hyperparameter).
3. Η ηλεκτρική ισχύ που καταναλώνεται ανά λεπτό (instance) σε KW. Η ισχύ που καταναλώνεται θεωρείτε de facto σταθερή για διάστημα ενός λεπτού (υπολογίζεται

με βάση την εξίσωση της ισχύος  $P=V*I$ ). Η τιμή αυτή είναι η βάση όλων των υπολογισμών της ενεργείας στο Energy Data Logger Panel.  
 Η συνολική ενέργεια δίνεται από παρακάτω αναδρομικό αλγόριθμο:

$$M31 = M34 * M45 + M31 \quad \text{σπού}$$

- M31 : Συνολική Ενέργεια (KWh)
- M34 : Η ενέργεια που έχει καταναλωθεί το τελευταίο min (θεωρείτε σταθερή ).
- M45 : ‘Μαγικός τελεστής’ είναι η έξοδος ενός timer ο οποίος βρίσκεται στο κρυφό πάνελ ‘Searching ms timer (ID=28) βλ. Το M45 γίνεται ανά ένα λεπτό για 1 ms 1 ( βλέπε Εικόνα 36: Η τιμή της μεταβλητής M45 στον Figure 2: Η έξοδος του χρονιστή M45)

Κατά την διάρκεια του ενός ms ο αναδρομικός τύπος τρέχει-εκτελείται μόνο μια φορά

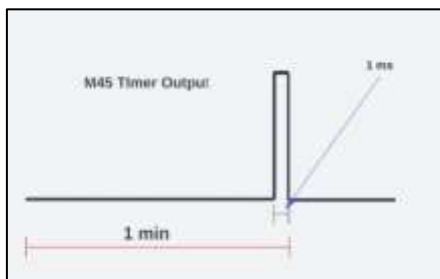


Figure 2: Η έξοδος του χρονιστή M45

Στην εικόνα 38 δίνεται η εικόνα του κρυφού πάνελ με όνομα Searching ms timer και ο timer ο οποίος χρονίζει την μεταβλητή M45



Εικόνα 37: Searching ms timer a Hidden panel Of NiK\_IoT API

### 3.14 Programmable Timer Panel.

Στο πάνελ αυτό Εικόνα 39 μπορούμε να προγραμματίσουμε μέσω του widget Programmable timer χρονικά διαστήματα κατά τα οποία θα είναι ανοικτό το φορτίο μας. Ο προγραμματισμός είναι εβδομαδιαίως και υλοποιείται εύκολα μέσω. Ο χρονοδιακόπτης παρέχει την ευελιξία στον χρήστη να ρυθμίζει τη θερμοκρασία στον χώρο του σπιτιού του τον επιτρέπει στον χρήστη να προσαρμόσει τη θέρμανση στις ανάγκες του και να βελτιώσει την άνεσή του. Με τη χρήση ενός χρονοδιακόπτη

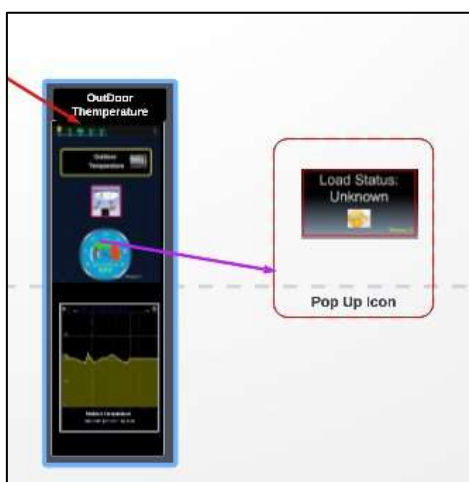


**Εικόνα 38: Programmable Timer Panel Of NiK\_IoT API.**

μπορούμε να επιτύχουμε άνεση, π.χ. μπορούμε να προγραμματίσουμε το σύστημα θέρμανσης να ενεργοποιηθεί μισή ώρα πριν το πρωινό γευτήριο ώστε το σπίτι να είναι ζεστό καθώς ετοιμαζόμαστε για το μεροκάματο, το ίδιο και κατά την επιστροφή μας. Το NiK\_IoT δίνει τη δυνατότητα ακύρωσης του προκαθορισμένου χρονοπρογραμματισμού μέσω του κομβίου Force OFF Button στο [The Control Panel](#). Να υπενθυμίσουμε και πάλι ότι το panel αυτό δεν έχει λειτουργικότητα να βρίσκεται στην εφαρμογή που έχει ο κοινός χρήστης στο κινητό του. Θα πρέπει να βρίσκεται μόνο στον ['Virtuino server'](#) του NiK\_IoT που είναι μόνιμα ανοικτός.

### 3.15 Outdoor Temperature Panel.

Στο panel αυτό Εικόνα 40 έχουμε αναλογική ένδειξη από την εξωτερική θερμοκρασία της οικίας, καθώς και αντίστοιχο γράφημα (time series graph). Το panel και οι δυνατότητές του είναι παρόμοιες με αυτές του Indoor panel. Η εξωτερική θερμοκρασία διαβάζεται μέσω της



**Εικόνα 39 : Outdoor Temperature Panel Of NiK\_IoT API.**

OutDoor Unit και του αισθητήρα DS18B20. Η λειτουργία του panel είναι παρόμοια με το

panel Indoor Temperature για κάποιο λόγο η σύνδεση του Wemos\_2 με το cloud χαθεί, το Virtuino σταματά να επαναεκπέμπει τα δεδομένα στον ThingSpeak. και πετάγεται Banner που ενημερώνει τον ανυποψίαστο χρήστη.

### 3.16 Visualization Data Panel.

Πριν προχωρήσουμε στην παρουσίαση του Visualization Data Panel Εικόνα 41 θα κάνουμε μια σύντομη αναφορά στην απεικόνιση δεδομένων. Η απεικόνιση δεδομένων (Visualization Data) επιτρέπει στον απλό χρήστη και όχι μόνο να διακρίνει πρότυπα, τάσεις και σχέσεις στα δεδομένα που μπορεί να μην είναι αμέσως εμφανή όταν κοιτάμε ένα μάζο αριθμούς (Big Data). Μέσω της απεικόνισης καθιστούμε τα δεδομένα πιο αντιληπτά και κατανοητά, επιτρέποντας μας να αποκτήσουμε μια βαθύτερη κατανόηση σε περίπλοκες πληροφορίες και συστήματα .

Ποιο συγκεκριμένα στην παρούσα Διπλωματική εκμεταλλευόμαστε τις δυνατότητες που μας προσφέρει η πλατφόρμα ThingSpeak η οποία συνεργάζεται με το Matlab. Το MATLAB είναι ένα λογισμικό αριθμητικών υπολογισμών και προγραμματισμού που αναπτύχθηκε από την MathWorks. Χρησιμοποιείται ευρέως στη βιομηχανία, την ακαδημαϊκή έρευνα για την ανάλυση δεδομένων, την επεξεργασία σήματος την ανάπτυξη αλγορίθμων Το MATLAB προσφέρει μια εύχρηστη γλώσσα προγραμματισμού και πλούσιες βιβλιοθήκες. Επιπλέον, περιλαμβάνει μια γραφική διεπαφή χρήστη που διευκολύνει την αλληλεπίδραση με το λογισμικό και την Οπτικοποίηση των αποτελεσμάτων.



**Εικόνα 40 Visualization Data Panel Of NiK\_IoT API**

Όπως διακρίνεται και στην παραπάνω εικόνα στο panel Visualization Data εμφανίζονται δυο διαγράμματα charts

Το πρώτο chart [https://thingspeak.com/apps/matlab\\_visualizations/499698](https://thingspeak.com/apps/matlab_visualizations/499698) εμφανίζει το ιστόγραμμα την εξωτερική θερμοκρασία των τελευταίων 7 ημερών, δηλαδή 168 ώρες από το κανάλι Monitor Channel field 3. Το ιστόγραμμα είναι ένα γραφικό που χρησιμοποιείται για την αναπαράσταση της συχνότητας εμφάνισης των διαφορετικών τιμών σε ένα σύνολο δεδομένων. Το ιστόγραμμα είναι ένα χρήσιμο εργαλείο για την εξέταση της κατανομής των δεδομένων και την αναγνώριση των πιο συχνών τιμών. Στο ιστόγραμμα, η οριζόντια άξονας αντιπροσωπεύει τις διαφορετικές τιμές που μπορεί να λάβει μια μεταβλητή, ενώ η κατακόρυφη άξονας αντιπροσωπεύει τον αριθμό των φορών που εμφανίζεται κάθε τιμή στο δείγμα ή σύνολο δεδομένων.

Το δεύτερο διάγραμμα (chart) στο [https://thingspeak.com/apps/matlab\\_visualizations/499666](https://thingspeak.com/apps/matlab_visualizations/499666) εμφανίζει τη γραφική παράσταση της κατανάλωσης ρεύματος τριών διαφορετικών ημερών:

1. Της τρέχουσας ημέρας με μπλε γραμμή.
2. Της ημέρας μία εβδομάδα πριν με κόκκινη γραμμή.
3. Της ημέρας ένα μήνα πριν με κίτρινη γραμμή.

Με αυτήν τη σύγκριση, μπορούμε εύκολα να κατανοήσουμε τα μοτίβα κατανάλωσης ενέργειας. Αυτό μπορεί να βοηθήσει στον εντοπισμό των τάσεων και στη λήψη πιο ενημερωμένων αποφάσεων. Η κατανόηση των προτύπων κατανάλωσης ενέργειας μπορεί να βοηθήσει στην βελτιστοποίηση της χρήση της ενέργειας προσαρμόζοντας τα συστήματα θέρμανσης. Επιπλέον, η σύγκριση δεδομένων κατανάλωσης ενέργειας από διαφορετικές ημέρες μπορεί να βοηθήσει στην ανίχνευση ανωμαλιών ή αλλαγών στην κατανάλωση ενέργειας, που μπορεί να υποδεικνύουν βλάβες στον εξοπλισμό ή ακόμα και ρευματοκλοπή

Στο παράρτημα 8.3.2 και στο παράρτημα 8.3.3 δίνονται ο κωδικας της διεπαφή Matlab Visualization Consumption of three days Current –Last week - Last month και A histogram of outdoor temperature αντίστοιχα

### 3.17 Scale Analog to Current Value (Hidden panel).

Το panel "Scale Analog Value to Current" Εικόνα 42 είναι ένα κρυφό panel της εφαρμογής. Αυτό σημαίνει ότι εμφανίζεται μόνο αν ανοίξετε το sketch της εφαρμογής μέσω της εφαρμογής Virtuino και όχι από το Ark. Η λειτουργία του panel είναι πολύ σημαντική, καθώς μετατρέπει την τιμή της αναλογικής τάσης



**Εικόνα 41 Hidden panel Scale Analog Value to Current**

σε τιμή ρεύματος. Συγκεκριμένα, το Wemos LOLIN32 Board διαθέτει έναν ενσωματωμένο μετατροπέα αναλογικού σήματος σε ψηφιακό (Analog to Digital Converter), που του επιτρέπει να διαβάζει αναλογικά σήματα και να τα μετατρέπει σε ψηφιακές τιμές. Το ADC στο LOLIN32 έχει ανάλυση 12 bit, που σημαίνει ότι μπορεί να παράγει ψηφιακές τιμές που κυμαίνονται από 0 έως 4095 για αναλογικές εισόδους από 0 έως 3,3 βολτ. Στο panel αυτό, μέσω με την χρήση μια πρωτοβάθμιας εξίσωσης, μετατρέπουμε τις τιμές από 0 έως 4095 σε τιμές έντασης ηλεκτρικού ρεύματος 0 A έως 60 Ampere. Η μέτρηση του ηλεκτρικού ρεύματος γίνεται μέσω μιας διάταξης (βλέπε λεπτομέρειες στην ενότητα [Relay Unit](#)).

Γενική περιγραφή ). Η ηλεκτρονική αυτή διάταξη αυτή παράγει μια αναλογική τάση ανάλογη με την ένταση του ηλεκτρικού ρεύματος .Α υποθέσουμε ότι έχω ρεύμα φορτίου 6,28 A η διάταξη δίνει αναλογική τάση 0,250 Volt η οποία μεσω του A/D του Wemos μετατρέπεται σε ψηφιακή σύμφωνα με την εξίσωση

- Ψηφιακή τιμή = (Αναλογική τάση / 3.3) \* 4095.

Άρα, για μια αναλογική τάση 0,250 V, η αντίστοιχη ψηφιακή τιμή είναι:

$$\text{Ψηφιακή τιμή} = (0.250 / 3.3) * 4095 = 310.$$

Στο εν λόγω panel μετατρέπουμε (scaling) την ψηφιακή τιμή ,ένδειξη 310 σε ένδειξη 6,28A.

Για την μετατροπή χρησιμοποιήσαμε τα μαθηματικά εργαλεία της γραμμικής αναλογίας από στην ιστοσελίδα <https://planetcalc.com/8110/>.Κατ ουσία το scaling ανάγεται στο πρόβλημα της Εύρεσης της εξίσωσης ευθείας από 2 δεδομένα σημεία A και B του καρτεσιανού επιπέδου. Τα δυο σημεία είναι το A με συντεταγμένες (Xa,Ya) και το B με συντεταγμένες (Xb,Yb) όπου

- X είναι οι τιμές της ψηφιακής τάσης
- Y είναι η τιμές του ρεύματος που θέλουμε να αντιστοιχίσουμε .

Η διαδικασία επιλογής των σημείων ήταν χρονοβόρα και απαιτητική, βάζαμε ένα φορτίο και μετρούσαμε το ρεύμα που τραβούσε το φορτίο με το πολυμετρο. Τιμή Y ήταν η ψηφιακή ένδειξη του πολυμετρου , και κατόπιν διαβάζαμε την ψηφιακή ένδειξη τιμής X στο Virtuino. Τα σημεία έπρεπε να είναι όσο το δυνατόν πιο μακρινά .

Τα δυο σημεία που επιλέχθηκαν ήταν τα εξής :

- A (Xa, Ya) A (161 , 5,7 A )
- B (Xb,Yb) B (1250 , 21 A ).

Με την βοήθεια της ιστοσελίδας που προαναφέραμε η εξίσωση της ευθείας βρέθηκε ότι είναι:

$$y = 0,014655172413 * x + 3,405172413793 \quad (\text{Εικονα 43})$$

Το σημείο A επιλέχθηκε πάνω από 4, 5 A γιατί κάτω από τα 4, 5 A η διάταξη δεν μετράει σωστά και το B στα 25 A διότι παραπάνω δεν άντεχαν τα καλώδια της οικιακής παροχής. Ακόμα και στην έκδοση Ark ο χρήστης αν και οι μεταβλητές συντελεστές α και β αναφέρονται σε hidden panel μπορεί να ορίσει εκ νέου τις τιμές τους σε περίπτωση που γίνει αλλαγή στο hardware του κυκλώματος. Μια επιπλέον ανάλυση της παραπάνω μεθόδου από την πλευρά του προγραμματισμού δόθηκε επίσης στην ενότητα [Προγραμματισμός στο γραφικό περιβάλλον της εφαρμογής Virtuino](#)

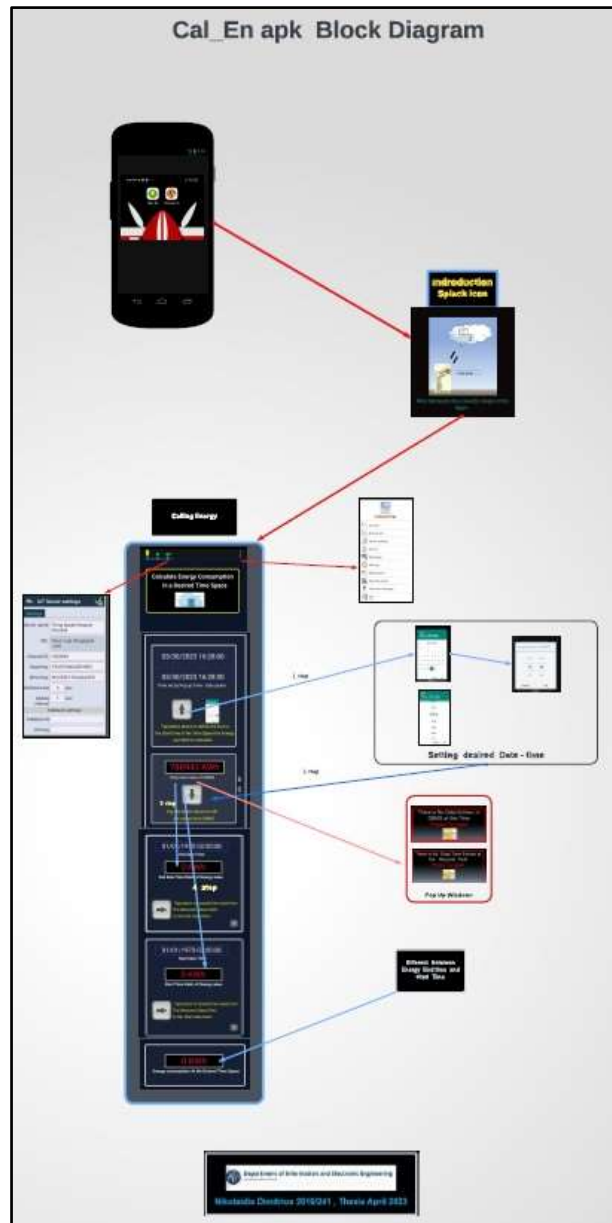


**Εικόνα 42:**Ευρεση Εξίσωση ευθείας που διέρχεται από 2 σημεία.

## 4 The Cal\_En API.

### 4.1 Εισαγωγή στην εφαρμογή Cal\_En.

Στο κεφάλαιο αυτό θα σας παρουσιάσουμε ίσως το πιο απαιτητικό μέρος της διπλωματικής από πλευράς σχεδιασμού και αρχιτεκτονικής. Θα προσπαθήσουμε να αναδείξουμε λεπτομέρειες στον τρόπο λειτουργίας της και την φιλοσοφία της. Σκοπός του όλου εγχειρήματος είναι η εφαρμογή να απαντάει στην υπαρξιακή ερώτηση του χρήστη: Πόσες κιλοβατώρες έκαψα την περίοδο του Πάσχα που ήμουν στο χωριό από τις 10 Απριλίου και ώρα 10.32 μέχρι τις 23 Απριλίου και ώρα 12.00



Block Διάγραμμα 4: Cal\_En API DiagramAPI.

Στο παρακάτω link δίνεται το παραπάνω Block Διάγραμμα σε μορφή pdf:

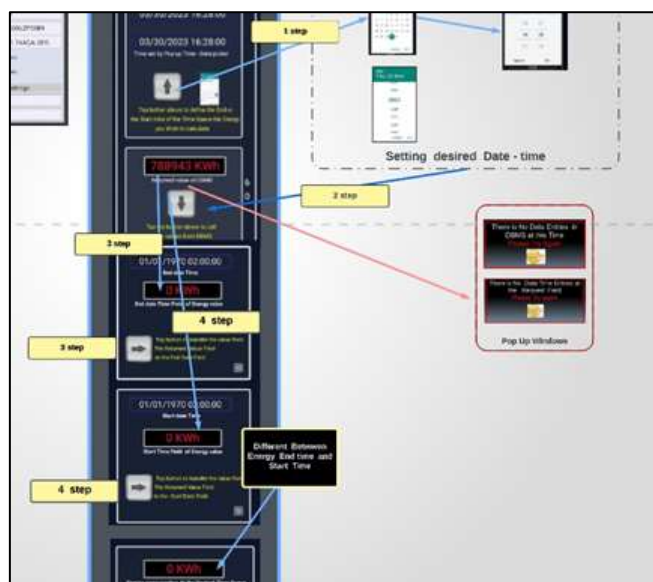
<https://drive.google.com/file/d/1rnZx3gQN1cZxEvta75ZmwBqRw-Wk0rCh/view?usp=sharing>

Για την υλοποίηση της εφαρμογής χρησιμοποιήσαμε το API "Matlab Analysis" που παρέχει η πλατφόρμα Thingspeak.com. Το API αυτό παρέχει μια σειρά από εφαρμογές βασισμένες σε MATLAB για ανάλυση δεδομένων. Αυτές οι εφαρμογές σχεδιάστηκαν για να λειτουργούν με δεδομένα που συλλέγονται και αποθηκεύονται από την πλατφόρμα και περιλαμβάνουν ανάλυση σειρών χρόνου. Το "Matlab Analysis" είναι μια εφαρμογή προγραμματισμού στο Back-End. Το Back-End programming αναφέρεται στον κώδικα που εκτελείται στην πλευρά του διακομιστή (Server). Ο κώδικας που γράφεται από τους Back-End developers είναι υπεύθυνος για τη διαχείριση της βάσης δεδομένων και την επεξεργασία των δεδομένων που λαμβάνονται

Στην καρδιά του ThingSpeak βρίσκεται μια βάση δεδομένων χρονοσειρών. Οι βάσεις χρονοσειρών σχεδιάζονται για τη διαχείριση μεγάλου όγκου δεδομένων και εκτελούν γρήγορες και αποτελεσματικές ερωτήσεις και αναλύσεις αυτών των δεδομένων. Ωστόσο, πρέπει να τονίσουμε ότι τα κανάλια του ThingSpeak Server δεν είναι σχεδιασμένα για ανάκληση δεδομένων όπως αυτή που γίνεται από το API Cal\_En. Παρ' όλα αυτά νομίζω ότι έχουμε καταφέρει να δώσουμε μια εναλλακτική και καινοτόμα χρήση του ThingSpeak Server σε συνδυασμό με το Virtuino.

## 4.2 Παρουσιαση της εφαρμογης Cal\_En

Η εφαρμογη αποτελείτε από ένα και μονο panel. Ολόκληρο το διάγραμμα του API και της λειτουργίας τους δίνεται στο Block Διάγραμμα 4: Cal\_En API Diagram. Αρχικά λόγω της δυσκολίας του το panel δουλεύτηκε μονο του . Δεν ενσωματώθηκε με την υπόλοιπη εφαρμογη καθώς θεωρήθηκε περιττό να φορτώσουμε την ήδη βαριά εφαρμογη με ακόμα ένα panel. Έστω λοιπόν ότι η Σουμελα θέλει να μάθει πόσες κιλοβατώρες έκαψε από τις 10 Απριλίου και ώρα 10 .32 (Start Data Time ) μέχρι τις 23 Απριλίου και ώρα 14.15 ( End Data Time) .Να υπενθυμίσουμε κάθε 5 sec αποθηκεύεται στο Field 2 του Monitor Channel η τιμή της συνολικής ενεργείας που εχει καταναλωθεί.

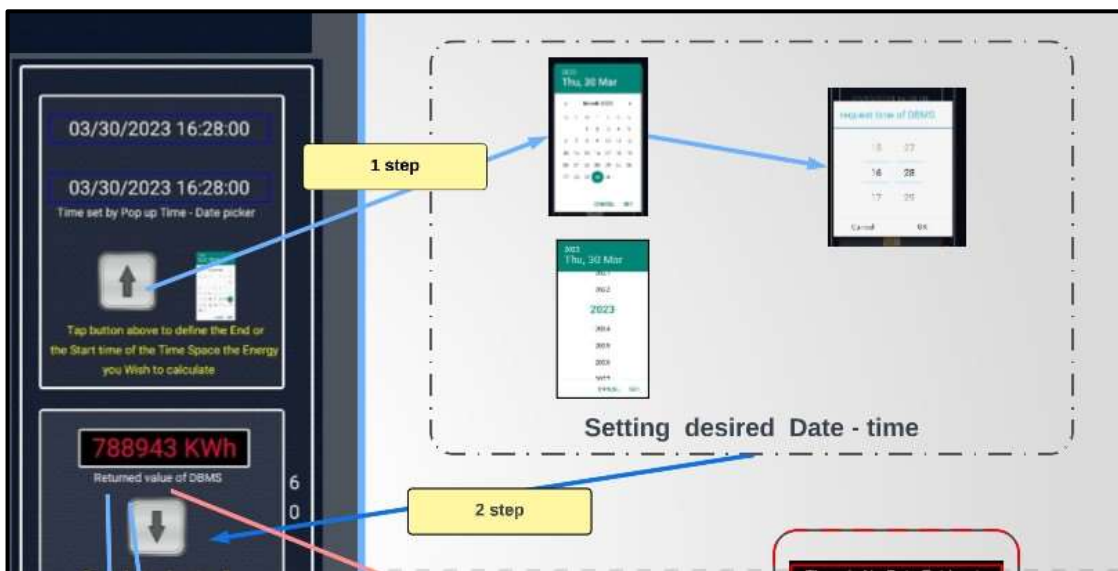


Εικόνα 43:Βήματα 1 , 2,3 και 4 για την ανάκληση Δεδομένων μέσω της εφαρμογής Cal\_En.

- **Step 1** :Επιλεγοντας το πρωτο κουμπι εισάγουμε την επιθυμητή ημερομηνία στο pop – Up Date time editor Αρχικά επιλέγουμε την End time date (βλέπε Εικόνα 4445
- **Step 2** :Επιλεγοντας το κουμπι στο από κάτω πλαίσιο επιστρέφει η τιμή της ενέργειας την «δεδομένη στιγμή» που εισήγαμε προηγουμένως στο value display του πλαισίου.
- **Step 3** :Επιλεγοντας το αντίστοιχο κουμπι βλέπε Εικόνα 43 μεταφέρεται η ανακλημένη τιμή στο πεδίο End time Field of Energy value.

Επαναλαμβάνουμε την διαδικασία ξεκινώντας και πάλι όπως πριν:

- **Step 1** εισάγουμε την επιθυμητή ημερομηνία στο pop – Up date time editor επιλεγοντας τώρα την Start Time date
- **Step 2** :Επιλεγοντας το κουμπι στο από κάτω πλαίσιο επιστρέφει η τιμή της ενέργειας την «δεδομένη στιγμή» που εισήγαμε προηγουμένως στο value display του πλαισίου
- **Step 4** :Επιλεγοντας το αντίστοιχο κουμπι βλέπε Εικόνα 4344 μεταφέρεται η τρέχουσα ανακλημένη τιμή στο πεδίο Start time Field of Energy value
- Στο πεδίο «Energy Consumption at the desired space time» εμφανίζεται στο value display η ενέργεια που καταναλώθηκε στο επιθυμητό χρονικό διάστημα



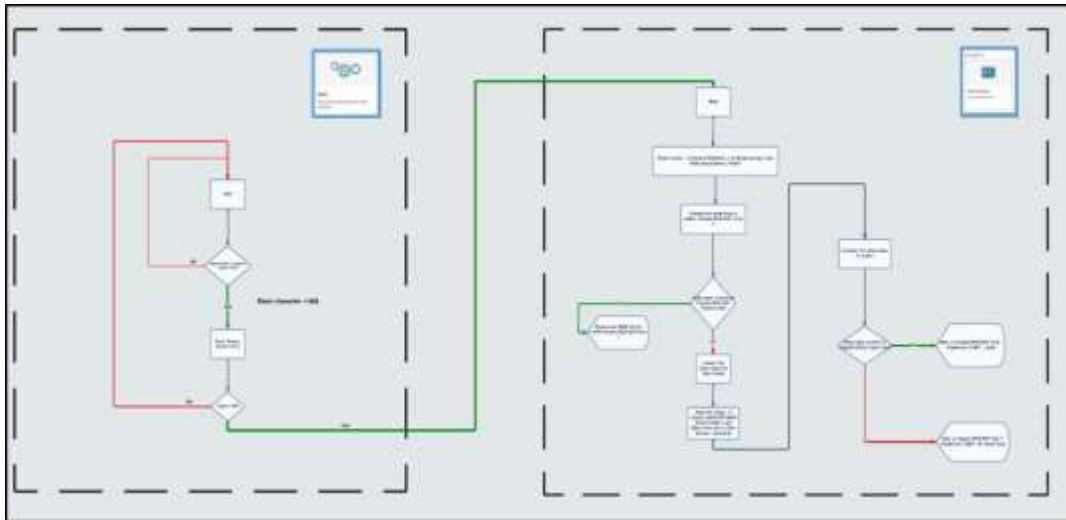
Εικόνα 44: Τα βήματα 1 και 2 για την ανάκληση Δεδομένων μέσω της εφαρμογής Cal\_En



#### 4.4 MATLAB Analysis App over ThingSpeak Server

Σε αυτήν την ενότητα θα προσπαθήσουμε να περιγράψουμε ολόκληρη τη φιλοσοφία του κώδικα και την αρχιτεκτονική του, καθώς και τις δυσκολίες του. Διότι Όπως αναφέραμε και προηγουμένως, τα κανάλια του ThingSpeak Server δεν είναι σχεδιασμένα για άμεση ανάκληση δεδομένων.

Αρχικά θα παρουσιάσουμε τον κώδικα React και κατόπιν τον κώδικα "TSIMPAO".



**Block Διάγραμμα 5 :Διάγραμμά ροής της εφαρμογής React και της εφαρμογής Matlab Analysis 'TSIMPAO'**

Στο παρακάτω link δίνεται το παραπάνω Block Διάγραμμα σε μορφή pdf

[:https://drive.google.com/file/d/1OKESIW0V3MzA7tOOe9xX\\_BbEHNcaX3pp/view?usp=sharing](https://drive.google.com/file/d/1OKESIW0V3MzA7tOOe9xX_BbEHNcaX3pp/view?usp=sharing)

Στο Block Διάγραμμα 5 φαίνεται αριστερά σε διάγραμμα ροής η εφαρμογή React και δεξιά η εφαρμογή MATLAB Analysis "TSIMPAO". Στο παράρτημα 3 παραθέτουμε τον κώδικα με εκτενή σχόλια. Επειδή στο παραπάνω Block Διάγραμμα δεν είναι ευκρινή τα γράμματα δίνεται στο πλαίσιο από κάτω, σε link το διάγραμμα σε μορφή PDF όπου φαίνονται όλα ξεκάθαρα. Όλη η διαδικασία τρέχει αποκλειστικά σε ένα μονο κανάλι του ThingSpeak το Request DBMS Channel

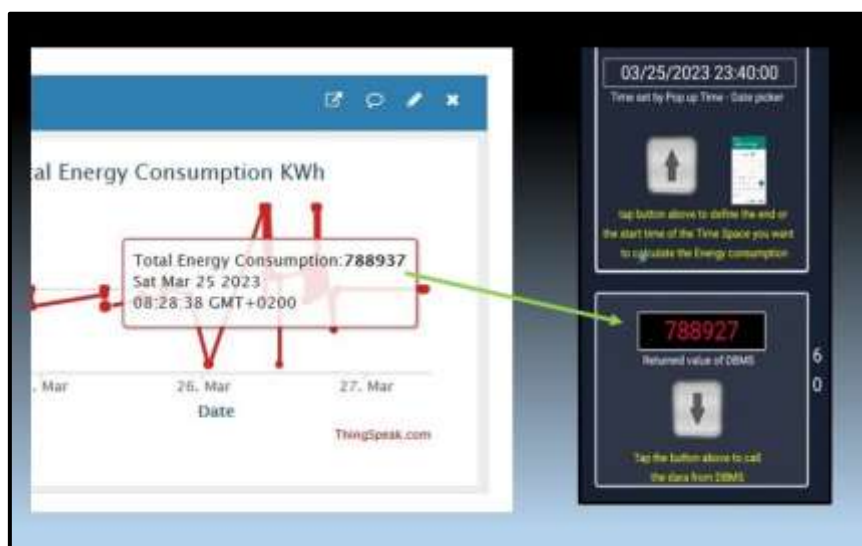
Έστω ότι κώδικας "TSIMPAO" είναι λειτουργεί τέλεια και επιστρέφει τις αναμενόμενες τιμές η ερώτηση είναι κάθε πότε θα τρέχει ;. Όταν ζητάω την τιμή της ενέργειας μιας δεδομένης στιγμής, το Virtuino στέλνει αρχικά ένα καθαρό νούμερο που αντιστοιχεί στο χρόνο στο Field 6 με όνομα M\_80 upload value Time-Date pop-Up picker και κατόπιν την τιμή 666 στο με όνομα React character Η εγγραφή στο Field 8 με τιμή 666 'τριγκαρει' τον κώδικα "TSIMPAO" ( βλεπε Block Διάγραμμα 5)

Ο κώδικας "TSIMPAO" διαβάζει τις πρόσφατες εγγραφές που έγιναν στο Field 6 Υπάρχει η πιθανότητα αστοχίας διότι λόγω αργοπορίας και συγχρονισμού (βήμα 2), όταν πάει να διαβάσει στο field 8 να διαβάσει NaN και ο κωδικας να κολλήσει .Τα 5 ανεβάσματα από το Virtuino γίνονται με τη βοήθεια Timer. Κατόπιν διαβάζει τις εγγραφές που έχουν γίνει ένα λεπτό πριν και ένα λεπτό μετά από τον χρόνο που επιλέχθηκε από τον χρήστη Αυτό γίνεται διότι μπορεί εκείνη τη χρονική στιγμή που έχει επιλέξει ο χρήστης να μην υπάρχουν εγγραφές.

Το πιθανότερο όμως είναι ότι υπάρχουν σίγουρα εντός αυτού του διαστήματος δύο λεπτών ( αφού οι εγγραφές γίνονται ανά 5 sec ) . Αφαιρεί τις εγγραφές NaN, βγάζει το μέσο όρο και γράφει την τιμή της ανακλημένης ενεργείας στο field 7 "Return from DMBS".

#### 4.5 Ταυτοποίηση Ανάκλησης δεδομένων

Στην ενότητα αυτή θα κάνουμε παρουσιάσουμε μια επίδειξη της εφαρμογής ώστε να πιστοποιήσουμε την ορθότητα της λειτουργίας της



**Εικόνα 46 :Επίδειξη της Λειτουργίας ανάκλησης Δεδομένων μέσω της εφαρμογής Cal\_En**

- Καλώ μέσω του API Cal\_En ( δεξιά στην εικόνα ) την τιμή της ενέργειας στις 25 /3/2023 και ώρα 23:40:00 9 (βλεπε πάνω δεξιά ) η τιμή που επιστρέφει η εφαρμογή φαίνεται στο κάτω πλαίσιο στο value Display Returned value of DBMS το οποίο έχει ένδειξη 788927
- Η τιμή συμπίπτει με αυτήν που μας υποδεικνύει ο Server αριστερά. Ο Κύβος εριφθι !!!

## 5 Κατασκευαστικό Μέρος.

### 5.1 Εισαγωγή στο Hardware.

Στο κεφάλαιο αυτό, θα αναφερθούμε στον τρόπο καθώς και στη διαδικασία υλοποίησης και κατασκευής. Το κεφάλαιο αυτό έχει τρεις ενότητες. Στην πρώτη ενότητα θα αναφερθούμε στην εσωτερική μονάδα "Indoor Unit". Στην δεύτερη ενότητα στην εξωτερική μονάδα "Outdoor Unit" και τέλος στο πιο δύσκολο κομμάτι του hardware, τη μονάδα που περιλαμβάνει το ρελέ ισχύος καθώς και τη διάταξη μέτρησης ρεύματος "Relay Unit". Η παρουσίαση των επιμέρους μονάδων θα γίνεται σε αυτόνομες ενότητες. Αρχικά θα δοθεί μια περιγραφή του hardware, έπειτα ένα Block diagram του ηλεκτρονικού κυκλώματος με έμφαση στα υλικά που χρησιμοποιήθηκαν, καθώς παίζουν σημαντικό ρόλο στην αξιοπιστία και στο συνολικό κόστος του όλου εκ πονήματος

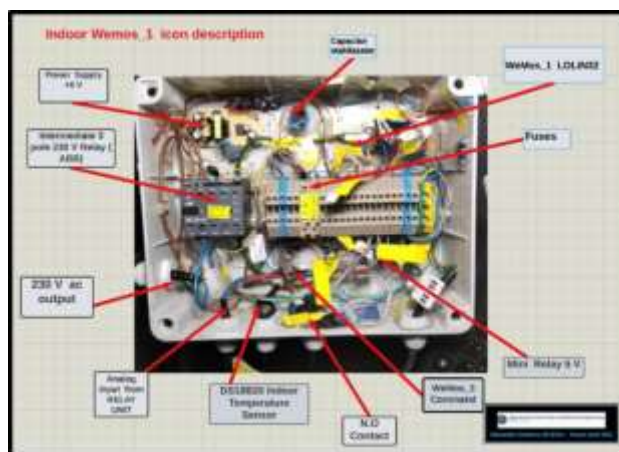
Στο τέλος κάθε ενότητα θα αναλύεται ο κωδικας του προγράμματος το οποίο τρέχει σε κάθε μικροελεγκτή. Σε κάθε περίπτωση θα δοθεί το flow chart του κώδικα και στο παράρτημα 8 ολόκληρος ο κώδικας με εκτενή σχόλια σε μορφή text. Ο κωδικας δίνεται επίσης και σε μορφή \*.ino μέσω υπερσύνδεσης. Σε κάθε εικόνα του Block diagram δίνεται το αντίστοιχο link του διαγράμματος σε μορφή pdf, έτσι ώστε ο μελετητής της διπλωματικής να μπορεί να δει μεγαλύτερη σαφήνεια στο διάγραμμα. Στο τέλος στο Παράρτημα [Φωτογραφίες](#) της κατασκευής του NiK\_IoT δίνονται λεπτομέρειες της κατασκευής

### 5.2 InDoor Unit - Γενική περιγραφή.

Στην ενότητα αυτή θα παρουσιάσουμε την εσωτερική μονάδα Indoor Unit. Στο Block Διάγραμμα 1: Αρχιτεκτονική του συστήματος NiK\_IoT 1: φαίνεται η λειτουργία της μονάδας και η συσχέτιση της με τις υπόλοιπες

Κυρία, η αποστολή του Indoor Unit είναι:

- Να συνδέεται στο τοπικό δίκτυο του σπιτιού.
- Να συνδέεται στους Servers στο cloud.
- Να μετρά την εσωτερική θερμοκρασία του σπιτιού και να την στέλνει στο cloud.
- Να μετρά το ρεύμα φορτίου και να το στέλνει στο cloud.
- Να ανοίγει και να κλείνει το ρελέ ανάλογα με την τιμή στο πεδίο που θα διαβάζει από το cloud.



Εικόνα 47: Τοπολογία εξαρτημάτων και συνδέσεων στο InDoor Unit

Για την κατασκευή χρησιμοποιήθηκε ένα κουτί ηλεκτρολογικό [DeLock Ηλεκτρολογικό Κουτί Εξωτερικής Τοποθέτησης Στεγανό IP65](#) κόστος 40 Ευρω και σπηλιοθλιπτες διαμέτρου: 8.5mm και βαθμού προστασίας: IP55. Οι εξωτερικές συνδέσεις γίναν με καλώδιο εξωτερικού

**Στο παρακάτω link δίνεται η παραπάνω εικόνα 48 με μεγαλύτερη ευκρίνεια σε μορφή pdf:**

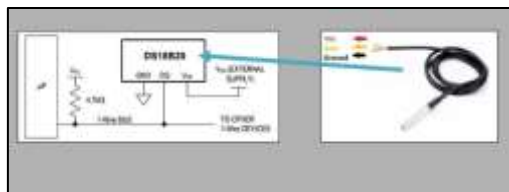
[https://drive.google.com/file/d/1j8\\_aRi8GkDLPTFZ1GURDIFTDMhFoKjVh/view?usp=sharing](https://drive.google.com/file/d/1j8_aRi8GkDLPTFZ1GURDIFTDMhFoKjVh/view?usp=sharing)

χώρου 3 X 2,5 mm και 2 X 0,7.mm. Στην εικόνα 44 δίνεται η φωτογραφία της μονάδας και η τοπολογία των υλικών και εξαρτημάτων.

Θα αναφερθουμε στα κυριότερα εξαρτήματα που απαρτίζουν την μονάδα

- Ψηφιακός Αισθητήρας θερμοκρασίας DS18B20 [18]

Το DS18B20 είναι ένας ψηφιακός αισθητήρας θερμοκρασίας της εταιρείας Dallas Semiconductor, που λειτουργεί με τη μέθοδο 1-Wire. Αυτό σημαίνει ότι όλες οι απαραίτητες συνδέσεις επικοινωνίας (τροφοδοσία, δεδομένα και γείωση) πραγματοποιούνται μέσω ενός μόνο καλωδίου. Ο DS18B20 μπορεί να μετρήσει θερμοκρασίες από -55°C έως +125°C με ακρίβεια  $\pm 0.5^\circ\text{C}$ . Η επικοινωνία με τον αισθητήρα γίνεται με τη χρήση σήματος σειριακής επικοινωνίας, και ο αισθητήρας είναι σε θέση να παράγει ακριβείς μετρήσεις θερμοκρασίας χωρίς την ανάγκη εξωτερικής βαθμονόμησης ή επιπλέον κυκλωμάτων. Στην εικόνα 49 δίνεται το pinout του αισθητήρα, καθώς και ο τρόπος σύνδεσής του.



**Εικόνα 48:Ψηφιακός αισθητήρας θερμοκρασίας DS18B20.**

- Το τροφοδοτικό της μονάδας [19].

Πάνω αριστερά διακρίνεται το τροφοδοτικό της μονάδας. Το τροφοδοτικό είναι αγορασμένο από Ebay και τα ηλεκτρικά του χαρακτηριστικά είναι τα εξής σύμφωνα με τον πωλητή.

- Input: AC100-240V 50/60Hz.
- Output: 6V 1000mA.
- Output voltage regulation:  $\pm 5.0\%$ .
- Output ripple and noise:  $\leq 200\text{mVp-p}$ .
- Κόστος 5 Euro.

Η μονάδα περιέχει 2 αναπτυξιακές πλακέτες στις οποίες δώσαμε τα εξής ονόματα:

- Wemos\_1 είναι το WEMOS LOLIN32 V1.0.0 -τα χαρακτηριστικά αυτής της πλακέτας τα παρουσιάσαμε στην ενότητα 1.6.
- Wemos Command η πλακέτα αυτή είναι ένα ESP32-CAM-Η ονομασία Wemos\_Command δόθηκε καταχρηστικός αλλά κάτ. ουσία δεν αλλάζει τίποτα βασικά χαρακτηριστικά του ESP32-CAM είναι :
  - χαμηλής ισχύος διπλού πυρήνα 32-bit CPU.
  - Η κύρια συχνότητα είναι μέχρι 240MHz και η υπολογιστική ισχύς φτάνει τα 600 DMIPS.

- Ενσωματωμένη μνήμη SRAM 520 KB, εξωτερική PSRAM 8MB.
- Ενσωματωμένα Lwip και FreeRTOS.  
Το Lwip (Lightweight IP) είναι μια υλοποίηση των πρωτοκόλλων TCP/IP για μικροελεγκτές με περιορισμένους πόρους, όπως οι επεξεργαστές με χαμηλό κόστος και μικρή μνήμη.  
Το FreeRTOS είναι ένα λειτουργικό σύστημα σε πραγματικό χρόνο, που παρέχει έναν πυρήνα προγραμματισμού που χρησιμοποιείται για την ανάπτυξη εφαρμογών σε ενσωματωμένα συστήματα.
- Ενσωματωμένη κεραία
- Υποστηρίζει λειτουργίες εργασίας STA/AP/STA+AP.  
Οι συντομογραφίες STA, AP και STA+AP αναφέρονται στις διαθέσιμες λειτουργίες των σημείων πρόσβασης (Access Point - As) και των συσκευών που συνδέονται σε αυτά.  
STA (Station) αναφέρεται σε μια συσκευή που συνδέεται σε ένα σημείο πρόσβασης (AP) για να έχει πρόσβαση στο δίκτυο. Δηλαδή, η συσκευή λειτουργεί ως πελάτης στο δίκτυο.  
AP (Access Point) αναφέρεται σε μια συσκευή που λειτουργεί ως σημείο πρόσβασης για τις συνδεδεμένες συσκευές. Δηλαδή, η συσκευή λειτουργεί ως δρομολογητής για το δίκτυο.  
STA+AP (Station + Access Point) αναφέρεται σε μια συσκευή που μπορεί να λειτουργεί ταυτόχρονα ως σταθμός και ως σημείο πρόσβασης. Δηλαδή, η συσκευή μπορεί να συνδέεται σε ένα δίκτυο ως πελάτης (STA) και ταυτόχρονα να παρέχει πρόσβαση σε άλλες συσκευές ως σημείο πρόσβασης (AP).
- Ποιο φτηνό από το Wemos Lol in 32 η τιμή του κυμαίνεται στα 5 ευρώ. Ένα πλεονέκτημα αυτής της αναπτυξιακής πλακέτας είναι ότι έχει ενσωματωμένη υποδοχή για σύνδεση με εξωτερική κεραία καθώς προορίζεται για χρήση με κάμερα.

- Mini relay [20]

- Τάση πηνίου 5 V.
- Max current 10 A.
- Κόστος 3 Ευρώ.

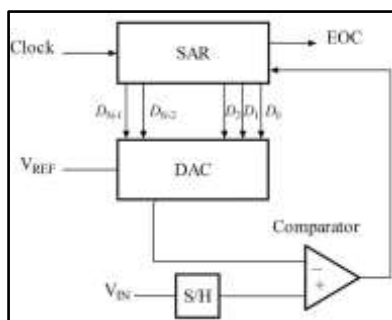
- Intermediate relay [21] 3 pole ABB B6-30-10-80 Mini Contactor με τα κάτωθι χαρακτηριστικά:

- 3 Pole.
- 1 Normally Open Auxiliary Contact.
- Max current 12 Amps ανά φάση.
- 220-240 Vac Coil.
- Κόστος 20 ευρώ .

Τοποθετήθηκε πυκνωτής 1000  $\mu\text{F}$  στην τροφοδοσία των 3.3 V για σταθεροποίηση. Να υπενθυμίσουμε εδώ ότι η μονάδα Wemos\_1 τροφοδοτείται με 6 Volt μέσω υποδοχής micro USB από το τροφοδοτικό. Η τάση αυτή υποβιβάζεται στα 3.3 V εσωτερικά στην πλακέτα. Αυτά τα 3.3 V κατόπιν τροφοδοτούν και τους 2 επεξεργαστές ESP32. Η σταθερότητα αυτής της τάσης είναι πάρα πολύ κρίσιμη. Πρώτον διότι είναι η τάση αναφοράς  $V_{\text{ref}}$  για το κύκλωμα μέτρησης αναλογικού σήματος σε ψηφιακό, Η μέθοδος που χρησιμοποιείται το εσωτερικά κύκλωμα A/D του Esp32 είναι η successive approximation. Ένα block διάγραμμα της μεθόδου δίνεται στην Εικόνα 49 Successive-approximation ADC Όπως παρατηρούμε εάν η τάση  $V_{\text{ref}}$  δεν είναι σταθερή, τότε και η έξοδος του μετατροπέα "παίζει", δημιουργώντας σωρεία προβλημάτων. Κάθε τάση που μετατρέπεται σε ψηφιακή τιμή συγκρίνεται με αυτήν την τάση. Ένα άλλο πρόβλημα που δημιουργείται από το ripple των 3.3 V είναι ότι το ESP32 κάνει restart με αποτέλεσμα να αποσυνδέεται από το τοπικό ασύρματο δίκτυο.

Όπως διακρίνεται και στο Block Διάγραμμα 6: Διάγραμμα του Ηλεκτρονικού κυκλώματος της InDoor Unit, έχει τοποθετηθεί ένας πυκνωτής μεταξύ αναλογικής εισόδου pin 36 και γειώσης.

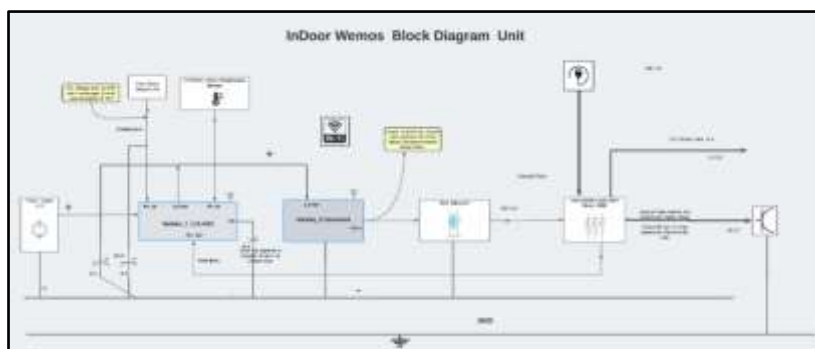
Αυτό βελτίωσε την 'κυματώση' στην ψηφιακή τιμή. Επιπροσθετα ένας άλλος πυκνωτής μεταξύ των pins En( Enable) και GND έλυσε το πρόβλημα



**Εικόνα 49 Successive-approximation ADC**

της φόρτωσης των προγραμμάτων στην πλακέτα. Η φόρτωση δεν μπορούσε να γίνει και το μήνυμα σφάλματος ήταν "Compilation error: exit status 1". Το πρόβλημα αυτό δεν παρουσιάζεται σε όλες τις πλακέτες του Wemos, αλλά είναι συχνό πρόβλημα σε ορισμένα firmware . Για τη διασύνδεση των εξαρτημάτων χρησιμοποιήθηκαν κλέμμες ράγας [22], οι οποίες είναι ογκώδεις αλλά είναι βιομηχανικού τύπου με σταθερότητα και αξιοπιστία. Στα καλώδια τοποθετήθηκαν ακροχιτωνία ανάλογης διατομής.

Εκτός από την εξοδο 230 V που παίρνουμε από την εξοδο του intermediate relay βγάλαμε και μια εξοδο επαφή N.O με την οποία μπορούμε να ελέγξουμε φορτία με διαφορετική τάση λειτουργίας ή μεσω επαφής ελέγχου π.Χ. ενεργοποίηση καυστήρα πετρελαίου με παρέμβαση στο ηλεκτρικό κύκλωμα του θερμοστάτη .



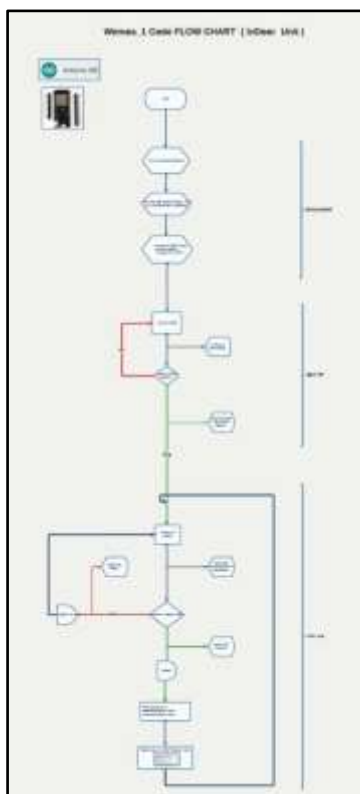
**Block Διάγραμμα 6: Διάγραμμα του Ηλεκτρονικού κυκλώματος της InDoor Unit**

Στο παρακάτω Link δίνεται το παραπάνω Block Διάγραμμα σε μορφή pdf :

<https://drive.google.com/file/d/1J2OSrtJafgeMT53jAC2IHZVwijaTIEeT/view?usp=sharing>

### 5.2.1 Flow chart of Arduino code Wemos\_1.

Ο κωδικας που ενσωματώνεται στο Wemos\_1 είναι γραμμένος σε γλώσσα προγραμματισμού 'Arduino' η οποία είναι βασισμένη στην γλώσσα C++. Πριν προχωρήσουμε στην παρουσίαση των προγραμμάτων θα αναφερθούμε εν συντομία για το προγραμματιστικό περιβάλλον Arduino IDE



**Block Διάγραμμα 7 : Flow chart of Arduino code Wemos\_1**

Δίνεται το Link του παραπάνω Block Διαγράμματος 7 σε μορφή pdf :

<https://drive.google.com/file/d/1HIuvs70ldpH8fpEODziKUea3AReadTf/view?usp=sharing>

Αυτούσιος ο παραπάνω κωδικας σε γλώσσα προγραμματισμού « Arduino » με εκτενή σχόλιά σε μορφή \*.ino

<https://drive.google.com/file/d/104937mAk1dPKdlAuRJV9YKRWoE-EfkBH/view?usp=sharing>

Το Arduino IDE είναι ένα λογισμικό ανοιχτού κώδικα που χρησιμοποιείται για τον προγραμματισμό των μικροελεγκτών Arduino. Παρέχει ένα περιβάλλον ανάπτυξης που επιτρέπει στους χρήστες να γράψουν, να μεταγλωττίσουν και να μεταφορτώσουν προγράμματα στις αναπτυξιακές πλατφόρμες τύπου Arduino υπάρχουν δεκάδες. Το Arduino IDE είναι διαθέσιμο για διάφορα λειτουργικά συστήματα, όπως Windows, Mac OS X και Linux, και είναι εύκολο στη χρήση ακόμα και από αρχάριους χρήστες. Η διαδικασία ανάπτυξης λογισμικού στο Arduino IDE περιλαμβάνει τη σύνταξη του κώδικα, τη μεταγλώττιση και τη μεταφόρτωση του προγράμματος στο μικροελεγκτή μέσω της σειριακής θύρας USB.

Η γλώσσα 'Arduino' έχει απλή σύνταξη, η οποία καθιστά εύκολη τη δημιουργία προγραμμάτων ακόμα και για αρχάριους προγραμματιστές. Η γλώσσα παρέχει βασικές δομές ελέγχου ροής όπως if, else και while. Η γλώσσα Arduino χρησιμοποιεί βιβλιοθήκες (libraries) που περιλαμβάνουν προγραμματιστικό κώδικα για τον έλεγχο διαφόρων ενεργοποιητών και αισθητήρων π.χ. στην δική μας περίπτωση των αισθητήρων θερμοκρασίας Dallas Αυτές οι βιβλιοθήκες καθιστούν εύκολο τον έλεγχο των συσκευών, αφαιρώντας την ανάγκη για την εγγραφή περίπλοκου κώδικα για τη λειτουργία τους.

Το Serial Monitor είναι μια ενσωματωμένη εφαρμογή στο Arduino IDE που χρησιμοποιείται για να επικοινωνεί το Arduino με τον υπολογιστή σας μέσω της σειριακής θύρας USB. Με το Serial Monitor μπορείτε να παρακολουθείτε τα δεδομένα που στέλνει ή λαμβάνει το Arduino, καθώς και να αλληλεπιδράτε με το Arduino στην περίπτωση που υπάρχει κώδικας που τον καθοδηγεί να περιμένει εισόδους από το Serial Monitor

Ας προσπαθήσουμε τώρα να παρουσιάσουμε τον κώδικα, επειδή το Block διάγραμμα 7, όπως και τα υπόλοιπα που παραθετονται στη διπλωματική, δεν είναι τόσο ευκρινή κάτω από κάθε flow chart δίνονται δύο πλαίσια: το ένα περιέχει link από όπου ο αναγνώστης μπορεί να κατεβάσει το flow chart σε μορφή PDF και το δεύτερο πλαίσιο περιέχει τον κώδικα σε μορφή \*.ino. Τέλος, υπάρχει και παραπομπή στο παράρτημα με τον αυθεντικό κώδικα σε text μορφή, με εκτενή σχόλια.

Ο κωδικας αποτελείται από τρία μέρη :

- Initialing
  - Εισάγουμε τις απαραίτητες βιβλιοθήκες για τους αισθητήρες και την ασύρματη σύνδεση
  - Εισάγουμε τις ρυθμίσεις του δικτύου Wi-Fi και τα διαπιστευτήρια του HIVEMQ broker
  - Δηλώνουμε τα MQTT topics
  - Δηλώνουμε το είδος του μεταβλητών.
  - Δηλώνουμε τα Pin του GPIO (General Purpose Input/Output)
- Set Up Ο κωδικας στο Initialing και στο set up τρεχει μονο μια φορά.
  - Σύνδεση στο τοπικο δίκτυο Wi-Fi
  - Έλεγχος συνδεσιμότητας.
- Void Loop όπως φαίνεται και από το μπλοκ διάγραμμα Το κωδικας στο Void Loop εκτελείται συνέχεια μέσα από ένα βρόγχο επανάληψης
  - Ελέγχει την Σύνδεση με το MQTT Cluster εάν είναι εντάξει συνεχίζει .Εάν δεν έχει γίνει σύνδεση, επιστρέφει και προσπαθεί να ξανασυνδεθεί μετά από παρέλευση κάποιου χρονικού διαστήματος.. Σε κάθε περίπτωση, έχει το ανάλογο μήνυμα στο Serial Monitor.
  - Διαβάζει την Analog input για την μετατροπή από αναλογική σε ψηφιακή μορφή
  - Διαβάζει την θερμοκρασία καλώντας τις κατάλληλες συναρτήσεις του αισθητήρα
  - Παράγει ένα τυχαίο νούμερο από το 1 έως το 100
  - Δημοσιεύει «Publish» τα topics στο MQTT cluster

Τα δεδομένα τα οποία διακινούνται από αυτόν τον μικροεπεξεργαστή δίνονται στον [Πίνακας 3](#)

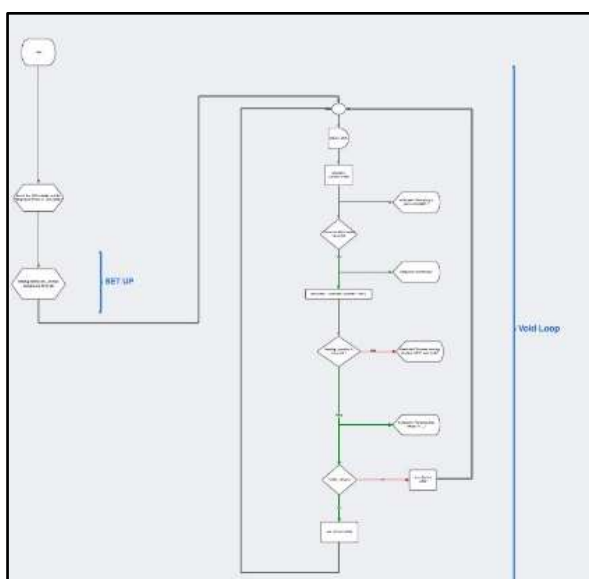
Το Serial Monitor είναι ιδιαίτερα χρήσιμο για την αποσφαλμάτωση του κώδικα του Arduino. Για παράδειγμα, στον παραπάνω κώδικα, εάν ο κώδικας δεν δουλεύει, μπορούμε να ελέγξουμε από το Serial Monitor αν υπάρχει πρόβλημα με τη σύνδεση στο Wi-Fi ή στο MQTT, ή αν δεν διαβάζει απλά τους αισθητήρες. Διαφορετικά και δεν θα μπορούσαμε να κάνουμε Debugging είτε λόγω λογικών σφαλμάτων είτε εξαιτίας λάθος συνδέσεων

### 5.2.2 Flow chart of Arduino code Wemos\_Command.

Κατά τα αντιστοιχία με την προηγούμενη ενότητα σε αυτήν την ενότητα θα παρουσιάσουμε σε flow chart στο Block Διάγραμμα 8 τον κώδικα που ενσωματώνεται στο Wemos\_Command .Το πρόγραμμα αποτελείται και πάλι από

- Την ρουτίνα αρχικοποίησης
- Την ρουτίνα Set-up και
- Την ρουτίνα Void loop. Στη γλώσσα προγραμματισμού C, ο όρος "void" χρησιμοποιείται για να δηλώσει μια συνάρτηση που δεν επιστρέφει κάποια τιμή.

Μια βασική διαφορά από τον κώδικα της προηγούμενης υποενοτητας είναι ότι μέσα στην ρουτίνα Void loop γίνεται έλεγχος σύνδεσης του Wemos\_Command με το Wi-Fi σε αντίθεση με τον προηγούμενο κώδικα που ο έλεγχος γινόταν μονο αρχικά στο set up .Με αυτόν τον τρόπο σε περίπτωση που το Wemos 'ξεκουμπώσει' από το Wi-Fi γίνεται προσπάθεια επανασύνδεσης Τα δεδομένα τα οποία διακινούνται δίνονται στον Πίνακας



4: Τα δεδομένα που λαμβάνονται από την InDoor Unit Wemos\_Command

### Block Διάγραμμα 8 Wemos\_Command Arduino Code Flow Chart

Link του παραπάνω Block Διαγράμματος 8 Wemos\_3 Arduino Code Flow Chart σε μορφή pdf:

[https://drive.google.com/file/d/1Ram\\_dqkKkAdC6U2-COf9TKJxISJkP4Zl/view?usp=sharing](https://drive.google.com/file/d/1Ram_dqkKkAdC6U2-COf9TKJxISJkP4Zl/view?usp=sharing)

Αυτούσιος ο κωδικας σε γλώσσα προγραμματισμού « Arduino » \*.ino με εκτενή σχόλια:

[Arduido Code of WEMOS\\_COMMAND](#)

## 5.3 Outdoor Unit .

### 5.3.1 Γενική περιγραφή της Εξωτερικής μονάδας.

Στην ενότητα αυτή θα παρουσιάσουμε την μονάδα OutDoor Unit. Με τη βοήθεια αυτής της "συσκευής" μετράμε την εξωτερική θερμοκρασία και την γράφουμε στον HiveMQ Broker. Για τη μέτρηση χρησιμοποιείται και πάλι ο αισθητήρας DS18B20. Επιλέξαμε την πλακέτα ESP32-DevKitC Core Board [23] διότι έχει ενσωματωμένη υποδοχή I- rex RF connector για τη σύνδεση με την εξωτερική κεραία. Στην πλακέτα δόθηκε το όνομα OutDoor\_Unit κατά σύμβαση .Στην εικόνα Εικόνα 51: I- rex coaxial RF connector σε φαίνονται οι κονεκτορες σύνδεσης. Η τοποθέτηση της κεραίας κρίθηκε απαραίτητη ώστε η μονάδα να συνδέεται στο δίκτυο ακόμα και όταν τοποθετείτε μακριά από την κεραία του access point του "router". Στη Εικόνα 50 Outdoor Unitφαίνεται η εξωτερική μονάδα OutDoor Unit



**Εικόνα 50 Outdoor Unit**

Η κεραία που τοποθέτησέ αγοράστηκε από την πλατφόρμα ηλεκτρονικών πωλήσεων eBay [24] και έχει τα παρακάτω χαρακτηριστικά

- Μήκος :30CM
- Connector : RP-SMA
- Impedance :50 Oh
- Gain:10dBi
- Frequency2400-2500 MHz
- Κόστος 6 Ευρώ



**Εικόνα 51: I- rex coaxial RF connector σε μεγέθυνση.**

Ο μικροελεγκτής τροφοδοτείται μέσω συστοιχίας τεσσάρων μπαταριών τύπου AA \* 1,5 V εν σειρά. Είναι τοποθετημένος μέσα σε ηλεκτρολογικό Κουτί επίτοιχο διαστάσεων 151 x 117 mm IP66 Ορθογώνιο με κόστος 10 Ευρω .Σε συνεχή λειτουργία οι μπαταρίες έχουν διάρκεια ζωής μόνο 6 ωρών. Οι μπαταρίες που τοποθετήθηκαν δεν είχαν αναγραφόμενη τη χωρητικότητα αλλά υπολογίζονται στα 2000 mAh. Ο τρόπος λειτουργίας και κατασκευής δεν είναι και αποδεκτός. Παρακάτω θα παραθέσουμε ορισμένες λύσεις ώστε η Outdoor Unit να λειτουργεί σωστά.

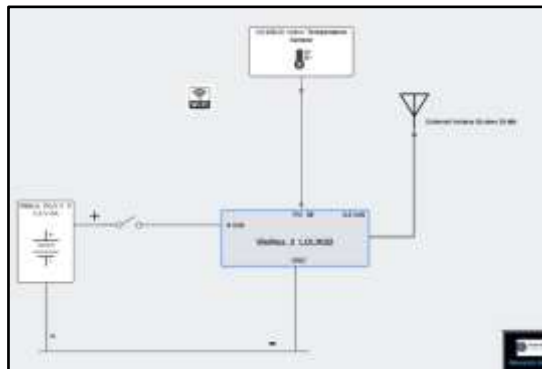
Η μία λύση είναι φυσικά να τη συνδέσουμε με εξωτερική τροφοδοσία όπως κάναμε και με τις άλλες μονάδες ,αυτό βέβαια δεν είναι και τόσο δόκιμο για εξωτερική μονάδα μακριά από ηλεκτρική παροχή .

Η άλλη λύση είναι να αλλάξουμε τη λειτουργία του επεξεργαστή και να τον θέσουμε σε λειτουργία Sleep mode Ο επεξεργαστής θα πέφτει σε ύπνωση για κάποια λεπτά της ώρας έστω 20 και κατόπιν θα ξυπνήσει μέσω εσωτερικού interrupt. θα ξανασυνδεθεί στο δίκτυο και θα ξαναμετρήσει την θερμοκρασία , θα στείλει τις τιμές στοcloud και θα ξανακοιμηθεί. Η τεχνική αυτή χρησιμοποιείται συχνά σε εξωτερικές μονάδες IoT για περιορισμό της ενεργειακής κατανάλωσης. Απαιτείται ειδικού τύπου κώδικας με κενή void loop Στην ενότητα 1.6 έχουμε δώσει κάποιες βασικές πληροφορίες για τις καταναλώσεις στα διάφορα είδη λειτουργίας Αν θέσουμε το Esp32 σε hibernation Sleep mode με κατανάλωση μονο 5  $\mu$ A ( ας το θεωρήσουμε μηδενικό) και έστω ότι τον duty cycle του επεξεργαστή είναι : κάθε 20 min SLEEP OFF και 0,07 min (4 sec ) ACTIVE πρακτικά σημαίνει ότι αυξάνω την διάρκεια των μπαταριών κατά 285 (= 20/0,07 ) φορές ήτοι 70 ημέρες

Η άλλη εναλλακτική λύση Που προτείνουμε είναι ένα ηλιακό πάνελ σε συνδυασμό με επαναφορτιζόμενες μπαταρίες power bank .Οι απαιτήσεις σε ισχύ είναι μικρές σχεδόν 0.1 watts οπότε ένα power bank σε συνδυασμό με solar panel κατάλληλης χωρητικότητας θα ήταν εφικτό να δώσουν λύση στην μικρή διάρκεια λειτουργίας .Υπάρχουν στο διαδίκτυο αρκετές επιλογές εμείς προτείνουμε την κάτωθι [25] με χωρητικότητα 10000 mAh και κόστος 16 ευρώ

Βέβαια, μπορεί να επιλεγεί συνδυασμός των δυο τεχνικών για μεγαλύτερη απόδοση . Η τεχνική της ύπνωσης δεν προτιμήθηκε λόγω έλλειψης χρόνου και η δεύτερη λόγω χρημάτων. Να σημειώσουμε εδώ έναν περιορισμό στο χρόνο ύπνωσης .Υπάρχει η απαίτηση να μετράμε την θερμοκρασία τουλάχιστον ανά 20 min ώστε να ‘βλέπουμε’ τις μεταβολές της και να κάνουμε αντιστάθμιση που περιγράφουμε ( βλέπε: Thermostat Panel ).

Στο Block Διάγραμμα 9: : Block Διάγραμμα του Ηλεκτρονικού κυκλώματος της OutDoor UnitBlock Διάγραμμα 9 δίνεται σε διάγραμμα το ηλεκτρονικό κύκλωμα της μονάδας



**Block Διάγραμμα 9: : Block Διάγραμμα του Ηλεκτρονικού κυκλώματος της OutDoor Unit**

**Link του παραπάνω Block Διαγράμματος 9 με περισσότερη λεπτομέρεια σε μορφή pdf:**

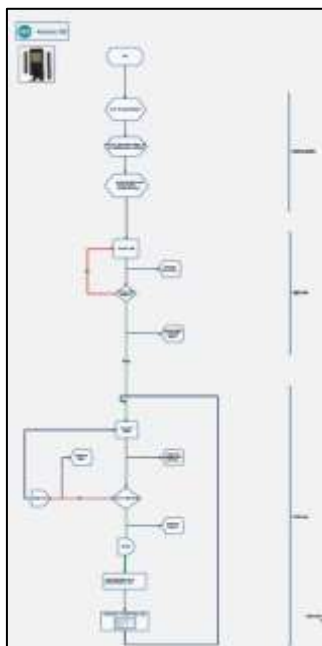
[https://drive.google.com/file/d/1miWwe2zoPUj2xTx3jqJSLT6G\\_WdVvRs6/view?usp=sharing](https://drive.google.com/file/d/1miWwe2zoPUj2xTx3jqJSLT6G_WdVvRs6/view?usp=sharing)

Στις αρχικές δοκιμές κάνοντας χρήση της ενσωματωμένης κεραίας η μονάδα έχανε το σήμα μόλις βγαίναμε έξω από το σπίτι ( 2 τοίχοι ενδιάμεσα ) , ετσι αναγκαστήκαμε να αγοράσουμε καινούργια πλακέτα με ενσωματωμένη υποδοχή για την εξωτερική κεραία .Αν και υπάρχουν ιστοτοποι που υποδεικνύουν την τοποθέτηση εξωτερικής κεραίας με

παρέμβαση στο pcb του Wemos κρίθηκε αναγκαίο να αγοραστεί πλακέτα με ενσωματωμένη υποδοχή ώστε να έχουμε σωστή προσαρμογή της αντίστασης εισόδου της κεραίας και άρα αποδοτικότερη λήψη

### 5.3.2 Flow chart of Arduino code Wemos\_OutDoor

Στο παρακάτω Block διάγραμμα δίνεται το διάγραμμα ροής του κώδικα που τρέχει στον μικροελεγκτή που ενσωματώνεται στην μονάδα OutDoor Unit .Ο κωδικας είναι παρόμοιος με αυτόν της μονάδας 5.2.1 για αυτό δεν γίνεται περαιτέρω αναφορά Τα δεδομένα που διακινούνται από την μονάδα στο cloud παραθετονται στον Πίνακας 2.



**Block Διάγραμμα 10 Wemos\_Command Arduino Code Flow Chart**

Αυτούσιος ο παραπάνω κωδικας σε γλώσσα προγραμματισμού « Arduino » με εκτενή σχόλια:

[Arduino Code of WEMOS Outdoor MQTT](#)

Link του παραπάνω Block Διαγράμματος με περισσότερη λεπτομέρεια σε μορφή pdf:

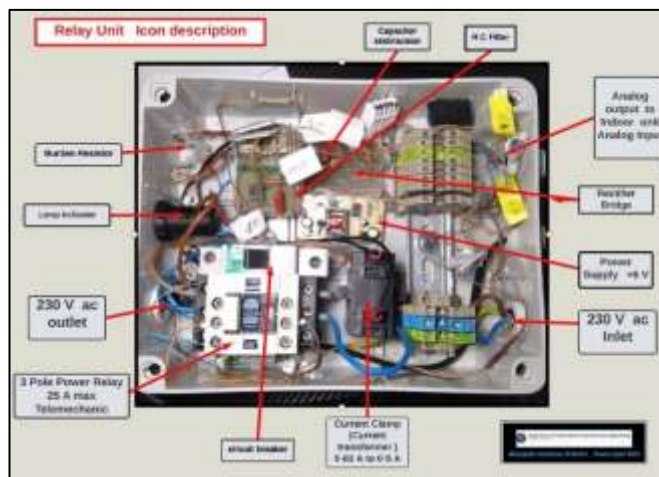
<https://drive.google.com/file/d/10tWdwGXc1oT5BLwyHq5vO6v7QO9MDUsw/view?usp=sharing>

## 5.4 Relay Unit .

### 5.4.1 Γενική περιγραφή της Relay Unit .

Στην ενότητα αυτή θα παρουσιάσουμε την μονάδα Relay module .Φωτογραφία της μονάδας δίνεται στην Εικόνα 53 καθώς η τοπολογία των εξαρτημάτων που την αποτελούν .Η μονάδα εκτελεί δυο λειτουργίες:

- Την οδήγηση του φορτίου μέσω του Relay.
- Την μέτρηση της έντασης του ρεύματος που καταναλώνεται από το φορτίο.



Εικόνα 52 Τοπολογία εξαρτημάτων και συνδέσεων στο Relay Unit

Στο παρακάτω link δίνεται η παραπάνω εικόνα σε μεγαλύτερη ευκρίνεια σε μορφή pdf:

[https://drive.google.com/file/d/1j8\\_aR18GkDLPTFZ1GURDIFTDMhFoKjVh/view?usp=sharing](https://drive.google.com/file/d/1j8_aR18GkDLPTFZ1GURDIFTDMhFoKjVh/view?usp=sharing)

Ποιο συγκεκριμένα η μονάδα αποτελείται από :

- Ένα Relay ισχύος LC 1 D09 [26] της Telemecanique με κύρια χαρακτηριστικά
  - Τρεις κύριες επαφές με μέγιστο ρεύμα λειτουργίας 25 A
  - Δυο βοηθητικές επαφές N.O and N.C με μέγιστο ρεύμα 10 A
  - Τάση λειτουργίας πηνίου 230 V ac 50 /60 Hz
  - Κόστος 40 Ευρώ
- Ένα μετασχηματιστή ρεύματος ( Current Clamp ) AC DX-30 60/5 VA [27] με χαρακτηριστικά:
  - (0-60 A) to (0-5 A)
  - Κόστος 8 Ευρώ
- Αντίσταση Burden: Η αντίσταση αυτή "μετατρέπει" το ρεύμα που "φτύνει" ο μετασχηματιστής σε τάση.. Αν και υπάρχει βιβλιογραφία [28] για τον υπολογισμό της τιμής της, η τιμή της βρέθηκε πειραματικά καθώς δεν υπήρχαν τεχνικά χαρακτηριστικά του μετασχηματιστή ρεύματος και πιο συγκεκριμένα οι σπείρες στο δευτερεύον .Επιλέχθηκε ένα δικτύωμα με συνολική αντίσταση 6.5 Ωhm με τη μέθοδο trial and

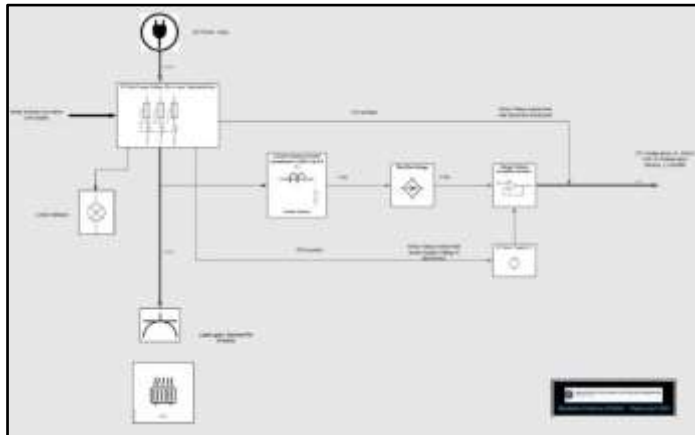
error.Σαν γενικό κανόνα η αντίσταση Burden δεν πρέπει να είναι πολύ μεγάλη γιατί θα γονατίσει την έξοδο του μετασχηματιστή ρεύματος και ούτε πάρα πολύ μικρή διότι η παραγομένη τάση θα είναι αντίστοιχα πολύ μικρή με συνέπεια να μην μπορούμε να την μετρήσουμε και να την επεξεργαστούμε.

- Κύκλωμα διπλής ανόρθωσης με πυκνωτή εξομάλυνσης ( Rectifier Bridge )
- Ενισχυτή Απομόνωσης Voltage Follower (Amplifier Isolation ) [29] .Για την υλοποίηση του χρησιμοποιήθηκε το ολοκληρωμένο κύκλωμα TLC 271 σε συνδεσμολογία rail to rail, high-bias mode In the high-bias mode, the TLC271 series features low offset voltage drift, high input impedance, and low noise.  
Το TLC271 είναι ένας τελεστικός ενισχυτής τεχνολογίας CMOS. Η επιλογή του TLC271 έγινε με βάση τα χαρακτηριστικά του :
  - Υψηλή αντίσταση εισόδου που καθιστά τον ενισχυτή κατάλληλο για εφαρμογές με μικρά σήματα εισόδου
  - Υψηλή αντίσταση εξόδου.
  - Χκάμηλό θόρυβο, που τον καθιστά κατάλληλο για εφαρμογές με μικρά σήματα
  - Ευρεία περιοχή τάσης τροφοδοσίας: Ο ενισχυτής λειτουργεί σε ευρεία περιοχή τάσης τροφοδοσίας, από 4 V έως 16 V
- Τροφοδοτικό 230 ac το 5 Volt 50 Hz

Όλη η κατασκευή είναι τοποθετημένη μέσα σε ηλεκτρολογικό κυτία ίδιων διαστάσεων με αυτού της InDoor Unit

#### 5.4.2 Block Diagram του κυκλώματος.

Στην παρακάτω εικόνα δίνεται το Block διάγραμμα του ηλεκτρονικού κυκλώματος της Relay Unit. Η κατασκευή της είχε αρκετές δυσκολίες. Για να γίνει λειτουργική αναγκαστήκαμε να κάνουμε κάποια τεχνάσματα. Το κύκλωμα έχει αρκετές αδυναμίες, με μεγαλύτερη αυτή της μη γραμμικότητας. Να υπενθυμίσουμε ότι σκοπός της μονάδας είναι να βγάξει στην έξοδο analog output μια αναλογική τάση ανάλογη της έντασης του ρεύματος που διαρρέει το φορτίο. Με το δεδομένο κύκλωμα αυτό δεν είναι εφικτό, δεν μπορούμε να μετρήσουμε ρεύματα κάτω από 4,5 A δηλαδή περιοχή από 0 A έως και 4,5 A βλεπε [Εικόνα 57](#) Σύγκριση των αποτελεσμάτων μετρήσεων στην περιοχή κάτω των 4,5. Οι λόγοι είναι οι εξής :



**Block Διάγραμμα 11 Block Διάγραμμα του Ηλεκτρονικού κυκλώματος της Relay Unit**

**Link του παραπάνω Block Διαγράμματος 11 Ηλεκτρονικού κυκλώματος της Relay Unit σε μορφή pdf:**

<https://drive.google.com/file/d/1C5x4bj1GgVIGbPC3YF9ciMNISXeY2bkX/view?usp=sharing>

- Έχω πτώση τάσης πάνω στην γέφυρα διπλής ανόρθωσης. Η τάση ορθής πόλωσης των διόδων είναι 0,6 V αυτό σημαίνει ότι θα εμφανίζεται τάση στην έξοδο πάνω από 1,2 V DC άρα το κύκλωμα μέτρησης είναι τυφλό για μικρά ρεύματα και άρα δεν μπορεί να τα ανιχνεύσει.
- Μη γραμμικότητα του Current Clamp (CT). Η γραμμικότητα ενός μετασχηματιστή ρεύματος (Current Transformer ) αναφέρεται στο βαθμό που η έξοδος ρεύματος του CT είναι ανάλογη με το εισερχόμενο ρεύμα. Ιδανικά, σε ένα CT το ρεύμα στην έξοδο έχει γραμμική σχέση με το ρεύμα πρωτεύοντος. Η πραγματικότητα όμως, ως συνήθως, είναι διαφορετική ο CT εμφανίζει μη γραμμικότητα στην αρχή της κλίμακας λόγω παραγόντων όπως το ρεύμα μαγνητισμού του πυρήνα και τις σιδηρομαγνητικές του ιδιότητες. Ο συγκεκριμένος μετασχηματιστής τοποθετείτε σε βιομηχανικούς πίνακες σε συνδυασμό με αναλογικό αμπερόμετρο, τα αμπερόμετρα αυτά δεν έχουν γραμμική ένδειξη
- Η τάση στην έξοδο του ενισχυτή απομόνωσης δεν είναι 0 volt όταν η είσοδος είναι 0. Αυτό είναι ένα εγγενή χαρακτηριστικό της συνδεσμολογίας rail to rail θα μπορούσε να λυθεί με την χρήση συμμετρικής τάσης [30]
- Στην πράξη και λόγω και της επιπρόσθετης μη γραμμικότητας του εσωτερικού μετατροπέα analog to digital του esp32 [11] η μέτρηση ρεύματος είναι αξιόπιστη πάνω από 4,5 Ampere αυτό βρέθηκε πειραματικά

Στη διάταξη τοποθετήσαμε zener 5 V στην είσοδο του ενισχυτή και στην αναλογική είσοδο του Esp32 διόδους για προστασία

Θα μπορούσαμε να κάνουμε ένα τέχνασμα ώστε το να υποδιπλασιάσαμε το όριο 4,5 A κάτω από το οποίο δεν μετρά η διάταξη. Αυτό επιτυγχάνεται εάν κάνουμε μια στροφή το καλώδιο γύρω από τον μετασχηματιστή (Νόμος του Αμπέρ). Βέβαια αυτό θα έριχνε στο μισό το εύρος μέτρησης από 0- 60 σε 0 – 30 αλλά έτσι συμβαίνει σηνυθως πάντα κάπου χάνεις κάπου κερδίζεις. Με αυτό το τέχνασμα θα μπορούσαμε να ελέγξουμε επίσης την ακρίβεια μέτρησης της διάταξης όταν «μετράει» 60 Amere 10 στροφές.

### 5.4.3 Μετρήσεις και σύγκριση τιμών .

Στην ενότητα αυτή θα παρουσιάσουμε με λεπτομέρεια τις μετρήσεις ρεύματος που έγιναν με την χρήση του NiK\_IoT σε αντιπαραβολή με όργανο μέτρησης εναλλασσόμενου ρεύματος Ταυτόχρονα θα δώσουμε και τις αντίστοιχές τιμές συνεχούς τάσης τις οποίες παράγει η διάταξη και μετατρέπονται κατόπιν σε ένδειξη ρεύματος. Ως όργανο μέτρησης εναλλασσόμενου ρεύματος χρησιμοποιήθηκε το πολυμετρο με αμπεροτσιμπίδα KONSTAR KSR 266D. Ως όργανο μέτρησης συνεχούς τάσης χρησιμοποιήθηκε το Digital Multimeter Protek 505

Παρακάτω παραθετονται 3 μετρήσεις

- Σημείο A στην αρχή της κλίμακας (Βλεπε Εικονα 54 )



**Εικόνα 53 : Σύγκριση των αποτελεσμάτων μετρήσεων ρεύματος στην περιοχή 6 A**

Αριστερα διακρίνεται το όργανο μέτρησης συνεχούς τάσης το οποίο μετρά την αναλογική τάση Στην μέση το όργανο αναφοράς και δεξιά σε κάθε εικόνα δίνεται σε μεγέθυνση η ένδειξη του NiK\_IoT

- Σημείο Β στην μέση της κλίμακας ( Βλεπε Εικόνα 55 )



**Εικόνα 54 :Σύγκριση των αποτελεσμάτων μετρήσεων στην περιοχή 14 A**

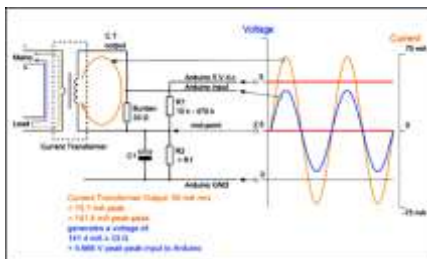
- Σημείο Γ το ορίσαμε ως ‘τέλος’ της κλίμακας γιατί δεν ήταν δυνατόν να γίνει έλεγχος με 60 Ampere φορτίο ( βλεπε Εικόνα 56 )



**Εικόνα 55:** Σύγκριση των αποτελεσμάτων μετρήσεων στην περιοχή των 20A

Ας σημειώσουμε εδώ ότι στα 60 Ampere η τάση εξόδου θα είναι 3V ,τάση η οποία δεν ξεπερνά τα 3,3 Volt, το εύρος μέτρησης του A to D του Esp32.

Η αρχιτεκτονικη ειδικά για την ακριβή μέτρηση μικρής τάξεως εναλλασσόμενου ρεύματος για είναι διαφορετική από αυτή που χρησιμοποιήσαμε [31] και δίνεται στην εικόνα 57 με την μέθοδο αυτή παρακάμπτουμε τις διόδους σηκώνουμε την εναλλασσόμενη τάση την κάνουμε μετατρέπουμε σε συνεχή. Μετα από συχνότητα δειγματοληψία μεγαλύτερη



**Εικόνα 56:**Εναλλακτικός τρόπος μέτρησης εναλλασσόμενου ρεύματος

από 50 Hz παίρνουμε την τιμή κατόπιν μαθηματικής επεξεργασίας των τιμών που προκύπτουν .Για καλύτερα αποτελέσματα μπορούμε επίσης να χρησιμοποιήσουμε το ολοκληρωμένο ADS1115 [32] αντί το εσωτερικό A/D του Esp32 το οποίο είναι ένας τετρακαναλος μετατροπέας. Κάνοντας χρήση δυο καναλιών αυτού του ολοκληρωμένου σε συνδεσμολογία διαφορικού ενισχυτή θα είχαμε μεγαλύτερη ακρίβεια , γραμμικότητα επιπρόσθετα θα μειώναμε το προβλήματα του floating ground -

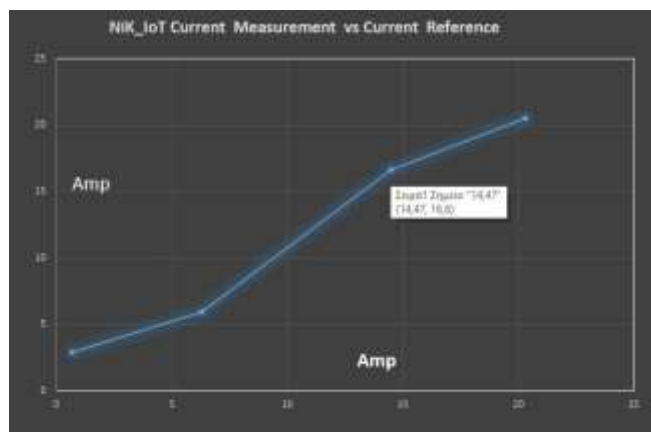


Εικόνα 57 Σύγκριση των αποτελεσμάτων μετρήσεων στην περιοχή κάτω των 4,5

Συνοπτικά δίνουμε τον παρακάτω πίνακα με τα αποτελέσματα στα οποία προαναφερθήκαμε

|          | Αναλογική τάση V | Όργανο αναφοράς ( A ) | NiK_IoT ( A ) |
|----------|------------------|-----------------------|---------------|
| < 4,5 A  | 0,089            | 0,65                  | 2,9           |
| Σημείο A | 0,25             | 6,28                  | 5,9           |
| Σημείο B | 0,766            | 14,47                 | 16,6          |
| Σημείο Γ | 1.062            | 20,3                  | 20,5          |

Πίνακας 8:“Συγκεντρωτικά – συγκριτικά αποτελέσματα μετρήσεων ρεύματος

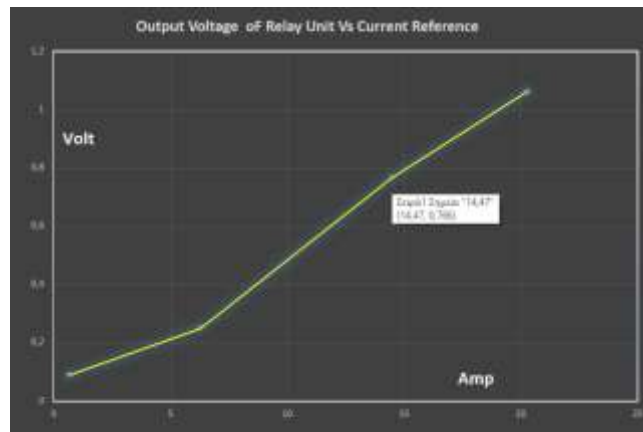


Εικόνα 58:Γραφική παράσταση X.Y των ενδείξεων του NiK\_IoT vs Reference

Με βάση τα αποτελέσματα του Πίνακα 8, κατασκευάσαμε δύο γραφικές παραστάσεις στο Excel, όπου φαίνεται η γραμμική συμπεριφορά της διάταξης. Η Εικόνα 59 δείχνει τη γραφική παράσταση της ένδειξης του NiK\_IoT ( άξονας Y) σε σχέση με το όργανο αναφοράς ( άξονας

X). Ενώ η Εικόνα 60 δείχνει το αντίστοιχο διάγραμμα X,Y της τάσης εξόδου της μονάδας Realy Unit (άξονας Y) σε σχέση με το όργανο αναφοράς ( άξονας X).

Θα ήταν πιο ακριβές και επιστημονικά ορθό να χρησιμοποιήσουμε το τέχνασμα με τις σπείρες ώστε να απεικονίσουμε τη συμπεριφορά της μετρητικής διάταξης σε όλο το εύρος τιμών της κλίμακας.



**Εικόνα 59:Γραφικ παρασταση X.Y της τάσης εξόδου vs Reference**

## 6 Αξιολόγηση.

### 6.1 Εισαγωγή.

Στην αρχική ενότητα της αξιολόγησης θα παρουσιάσουμε τις μετρήσεις που πραγματοποιήθηκαν, εξηγώντας τυχόν σφάλματα μέτρησης, αν υπάρχουν. Τα αποτελέσματα των μετρήσεων θα πιστοποιούν την ορθότητα της αντιμετώπισης του θέματος και θα καταδεικνύουν την χρησιμότητά του. Στην ενότητα των τροποποιήσεων θα παρουσιαστούν οι τρόποι προσαρμογής του συστήματος διπλωματικής σε άλλα συστήματα θέρμανσης και θα αναφερθούν οι δυσκολίες και οι παρούσες αστοχίες του συστήματος, καθώς και οι τρόποι αντιμετώπισής τους. Στην ενότητα των συμπερασμάτων θα παρουσιαστούν τα γενικά συμπεράσματα που προέκυψαν από την εκπόνηση της έρευνας και οι προτάσεις για περαιτέρω μελέτη και βελτίωση του συστήματος.

### 6.2 Μετρήσεις, Ανάλυση Αποτελεσμάτων.

Στην ενότητα 5.4.3 αναλύσαμε τα αποτελέσματα που προέκυψαν από τη μονάδα Relay Unit. Εξετάσαμε τις αποκλίσεις τιμών που παρατηρήθηκαν στις μετρήσεις και προτείνουμε λύσεις για την αποσφαλμάτωση αυτών.

Συγκρίναμε την ακρίβεια του ψηφιακού αισθητήρα DS18 με το αξιόπιστο βιομηχανικό



Εικόνα 60“Συγκριση αποτελεσμάτων μέτρησης θερμοκρασίας .

όργανο Stator-4849 της Chauvin Ardnoux. Στην εικόνα 61 δίνεται φωτογραφία από τις δοκιμές μετρήσεων. Μέσα στο δοχείο διακρίνονται τα δύο αισθητήρια, το Dallas από την Indoor Unit (μαύρο χρώμα) και το θερμόζευγος από το όργανο αναφοράς (άσπρο).

Πηραμε μετρήσεις σε τρεις περιοχές τιμών

| Stator -4849 ( reference ) °C | DS18B20 °C |
|-------------------------------|------------|
| 7                             | 8,8        |
| 20                            | 21,5       |
| 72                            | 74,8       |

Fun Fact: Η κούπα στην Εικόνα 61 είναι από την συνάντηση αποφοίτων του τμήματος Ηλεκτρονικής το 2015

Παρόλο που υπάρχουν αποκλίσεις, θεωρήθηκε ότι δεν είναι αναγκαία η βαθμονόμηση του αισθητήρα. Η διαδικασία βαθμονόμησης (calibration) είναι ίδια με το scaling που περιγράφουμε λεπτομερώς στη σελίδα 54.

Μετά από μεγάλο χρονικό διάστημα δοκιμών, παρατηρήσαμε ότι το σύστημα έχει μερικές αστοχίες. Η πλειονότητα αυτών οφείλεται στο γεγονός ότι τα Wemos αποσυνδέονται από το ασύρματο δίκτυο και αποτυγχάνουν να επανασυνδεθούν. Σε αυτήν την περίπτωση, πρέπει να

γίνει restart με το κουμπί που υπάρχει πάνω στην πλακέτα, ή καλύτερα αποσύνδεση από την τροφοδοσία και επανασύνδεση. Οι αποσυνδέσεις είναι ακόμα περισσότερες όταν βρισκόμαστε μακριά από το 'router' ή όταν το κανάλι WIFI είναι γεμάτο. Το πρόβλημα μειώνεται όταν μετακινούμαστε σε κανάλι με λιγότερη κίνηση και πιο κοντά στον κεραία του Access point.

Μια επιπλέον λύση για την αύξηση του σήματος WIFI θα ήταν να τοποθετήσουμε εξωτερικές κεραίες στα δύο Wemos, όπως κάναμε με την OutDoor Unit. Η πλακέτα Wemos\_Command τοποθετήθηκε κάθετα και όχι ξαπλωτά στο κουτί. Παρατηρήστε ότι τα οικιακά 'router' τα οποία έχουν ενσωματωμένη κεραία πάνω στο PCB τοποθετούνται κάθετα, με αυτόν τον προσανατολισμό έχουν καλύτερη λήψη, αφού η κεραία του access point εκπέμπει με οριζόντια πόλωση. Όπως αναφέρθηκε και στην ενότητα 5.4, το NiK\_IoT δεν μπορεί να μετρήσει ρεύμα μικρότερο από 4,5 A. Στο αρχικό στάδιο δοκιμών, το NiK\_IoT μετρούσε ρεύμα ακόμα και με μηδενικό φορτίο. Αυτό είχε σαν συνέπεια οι μετρήσεις ενέργειας να είναι λανθασμένες, ακόμα και όταν το relay είναι off, αφού το σύστημα διάβαζε ρεύμα της τάξης των 2,5 A. Αυτό λύθηκε εν μέρει με ένα τέχνασμα: όπου το relay είναι απενεργοποιημένο, βραχυκυκλώνουμε την έξοδο της αναλογικής έξοδου..

Ένα ακόμη μειονέκτημα της εφαρμογής είναι ο χρόνος εκκίνησής της. Η εφαρμογή NiK\_IoT είναι αρκετά μεγάλη σε μέγεθος με αποτέλεσμα να ανοίγει σε περίπου 11 δευτερόλεπτα στο κινητό μου. Το κινητό μου έχει τα παρακάτω τεχνικά χαρακτηριστικά: RAM 2.00 Gb, CPU octacore μέχρι 2.01 GHz. Είναι γνωστό από μελέτες σχετικά με την αλληλεπίδραση ανθρώπου-μηχανής ότι ο μέσος χρήστης δυσφορεί μετά από 3 δευτερόλεπτα από το πάτημα ενός κουμπιού αν δεν υπάρχει απόκριση. Στα κούφια αυτά δευτερόλεπτα το welcome message δίνει τροφή για σκέψη ή ματαιότητα των 11 δευτερολέπτων. Ο χρόνος εκκίνησης της εφαρμογής είναι ένας σημαντικός παράγοντας για την εμπειρία των χρηστών και μπορεί να τους αποτρέψει από τη χρήση της. Επομένως, είναι σημαντικό η μείωση του χρόνου απόκρισης ώστε να εξασφαλιστεί η βέλτιστη εμπειρία χρήστη. Στην παρακάτω ενότητα αναφέρουμε κάποιες προτάσεις για τη μείωση του χρόνου εκκίνησης της εφαρμογής.

### 6.3 Τροποποιήσεις –Επεκτάσεις –Βελτιώσεις.

Στην ενότητα αυτή θα προτείνουμε τροποποιήσεις ώστε να λυθούν τα προβλήματα που αναφέρθηκαν στις προηγούμενες ενότητες καθώς επίσης και επεκτάσεις του project .

Ας αναφερθούμε αρχικά στο πρόβλημα των αποσυνδέσεων. Η πρόταση - βελτίωσης του NiK\_IoT που προτείνουμε ώστε να εξαλειφθεί το πρόβλημα αυτό είναι η τοποθέτηση ενός ψηφιακού μαντρόσκυλου (Watchdog). Πιο συγκεκριμένα, όπως αναφερθήκαμε και στην εισαγωγή 2.1 προτείνουμε τη λειτουργία του 'Virtuino-server', η οποία θα τρέχει συνεχώς η εφαρμογή NiK\_IoT. Η εφαρμογή επιβλέπει την κατάσταση σύνδεσης των Wemos με το τοπικό δίκτυο. Έτσι θα μπορούσαμε να τοποθετήσουμε ένα [Bluetooth Relay Module](#), το οποίο θα παίρνει εντολή από το Virtuino μέσω Bluetooth, σε περίπτωση που έχουμε αποσύνδεση, θα επανεκκινεί τα Wemos με την διαδικασία hard reset, rst pin to gnd.

Μια προτεινόμενη τροποποίηση είναι εάν το σύστημα θέρμανσης βασίζεται στο πετρέλαιο ή το φυσικό αέριο. Σε αυτήν την περίπτωση, η ενεργειακή κατανάλωση μπορεί να μετρηθεί μέσω της ροής του καυσίμου. Ένας προτεινόμενος αισθητήρας είναι ο αισθητήρας ροής G1/2" Water Flow Hall Sensor Switch 1-25L/min Turbine [Flowmeter](#). Ο παραπάνω αισθητήρας θα αντικαταστήσει το current clamp. Ο έλεγχος του καυστήρα θα γίνεται χρησιμοποιώντας τη NO επαφή που παρέχεται από την Indoor Unit. Η μέτρηση της κατανάλωσης πετρελαίου και, κατά συνέπεια, το κόστος λειτουργίας μπορεί να γίνει με τροποποίηση του κώδικα του Virtuino στο Panel Energy Monitor και με την εισαγωγή της τιμής του πετρελαίου στο Hyperparameter Panel.

Μια άλλη βελτίωση που μπορεί να γίνει είναι η δημιουργία μιας εφαρμογής " NiK\_IoT client". Στην παρούσα φάση, η εφαρμογή NiK\_IoT κάνει όλους τους υπολογισμούς μέτρησης ενέργειας και τον αλγόριθμο θερμοστάτη. Αυτό δεν είναι απαραίτητο να γίνεται σε μια εφαρμογή που απευθύνεται στον χρήστη. Ο χρήστης (client) θα διαβάζει μόνο αυτά που στέλνει ο 'NiK\_IoT Server' στο σύννεφο και θα ελέγχει το φορτίο χωρίς όμως να παρεμβαίνει στις τιμές των παραμέτρων, ούτε θα κάνει υπολογισμούς. Δεδομένου ότι ο κοινός χρήστης θα ανοίγει την εφαρμογή και θα την κλείνει, δεν έχει ουσία η μέτρηση της ενέργειας από αυτόν,

βλέπε ενότητα 3.11 . Η εφαρμογή "NiK\_IoT client" θα είναι πιο ελαφριά, καθώς δεν θα ενσωματώνει κώδικες και scripts, και θα λύνει το πρόβλημα της αργής εκκίνησης. Μια άλλη λύση για την αργή εκκίνηση επιπρόσθετα θα ήταν η ομαδοποίηση των panel, δηλαδή η ολόκληρη εφαρμογή NiK\_IoT να χωριστεί σε δυο ή τρεις όπως έγινε π.χ. με το Cal\_En API. Ένας ενδεικτικός επιμερισμός θα δημιουργούσε δυο API. Ένα αποκλειστικά για τον έλεγχο του φορτίου και τη συνδεσιμότητα και ένα άλλο για την απεικόνιση των παραμέτρων λειτουργίας. Η ρύθμιση των alarm και η παραμετροποίηση βλέπε ενότητες 3.9 και 3.12 μπορεί να γίνεται αποκλειστικά στην έκδοση του " NiK\_IoT Server ".

## 6.4 Συμπεράσματα.

Η εκπόνηση της εργασίας συνάντησε πολλές δυσκολίες. Θα προσπαθήσουμε να τις συνοψίσουμε στην ενότητα αυτή και να δώσουμε κάποια συμπεράσματα.

Η εφαρμογή Virtuino είναι ένα πανίσχυρο εργαλείο για την ανάπτυξη διαδικτυακών εφαρμογών. Ωστόσο, το γραφικό περιβάλλον προγραμματισμού δεν προσφέρεται για τόσο μεγάλες εφαρμογές με πολλά panels. Ο λόγος είναι ότι δεν μπορείς να δεις καθολικά τις μεταβλητές και να ανατρέξεις στα σημεία που παίρνουν μέρος. Βέβαια, το πρόβλημα λύνεται εάν ομαδοποιήσεις τα panels ανά λειτουργία. Στην περίπτωση του NiK\_IoT δεν έγινε αρχικά σωστός σχεδιασμός και όλα τα panel τοποθετήθηκαν σε ένα API.

Ένα άλλο μειονέκτημα του Virtuino είναι ότι είναι πολύ δύσκολο να παρουσιάσεις τον κώδικα σε ένα τρίτο άτομο ή σε ένα τεχνικό κείμενο όπως αυτό εδώ , όπως γίνεται με άλλες γλώσσες χαμηλού επιπέδου όπως αυτή του Arduino IDE. Αυτό έχει σαν συνέπεια ο κώδικας να είναι δύσκολος στην επέκταση και στο debugging ειδικά από ένα δεύτερο άτομο .Επιπλέον δεν είναι δυνατόν να γίνει Copy paste. Βέβαια, από την άλλη πλευρά, οι πλούσιες βιβλιοθήκες του και η ευελιξία του το καθιστούν μοναδική επιλογή. Η νέα έκδοση του Virtuino IoT, η οποία λειτουργεί σε περιβάλλον Windows και επιτρέπει τη χρήση μεγάλης οθόνης, βελτιώνει την ευχρηστία του προγράμματος ακόμα και για τα μεγάλα έργα. Μεγαλύτερη οθόνη σημαίνει περισσότερο χώρο για την προβολή των δεδομένων και την εύκολη διαχείριση των στοιχείων του έργου.Ολη η εφαρμογη NiK\_IoT κατασκευάστηκε σε κινητό οθόνης 5,9 ιντσών.

Η διαδικασία αποσφαλμάτωσης στη διάταξη μέτρησης ρεύματος αποδείχθηκε πολύ χρονοβόρα και αναγκαστήκαμε να κατασκευάσουμε hardware που δεν είχε αρχικά προβλεφθεί στο σχεδιασμό. Αυτό είχε σαν αποτέλεσμα να μην υπάρξει αρκετός χρόνος για τη βελτίωση άλλων περιοχών της διπλωματικής εργασίας, όπως η αυτονομία της Outdoor Unit και ο χρόνος εκκίνησης της εφαρμογής.

Από την πλευρά του ηλεκτρονικού hardware, αντιμετωπίσαμε και εκεί δυσκολίες. Ζω σε μια επαρχιακή πόλη, όπου δεν υπάρχει εμπορικό κατάστημα που να προμηθεύει ηλεκτρονικό υλικό. Αυτό είχε ως αποτέλεσμα να καθυστερεί η προμήθεια των εξαρτημάτων που γίνονταν διαδικτυακά με αποτέλεσμα η ολοκλήρωση της εργασίας να γίνεται αποσπασματικά με διακοπές. Η ανάπτυξη ακόμα και ενός σχετικά απλού hardware όπως αυτό που υλοποιήσαμε απαιτεί τη χρήση ηλεκτρολογικών εργαλείων , βιολόγο, ποτηροτρύπανα, εργαλεία κοπής και διάνοιξης, για την κατασκευή καθώς και ηλεκτρονικών οργάνων μέτρησης. Το συνολικό πακέτο εργαλείων και εξαρτημάτων αυξάνει σημαντικά το κόστος της διπλωματικής. Είναι σημαντικό το πακέτο αυτό να είναι διαθέσιμο και προμηθευμένο εγκαίρα, προκειμένου να αποφεύγονται καθυστερήσεις. Οι αναπτυξιακές πλακέτες και οι αισθητήρες αγοράστηκαν από το eBay με χρόνο με παράδοσης δυο μήνες παραγγελία έγινε σε προκειμένο χρόνο με την πρόβλεψη και για εφεδρική πλακέτα σε περίπτωση ατυχήματος όπως και συνέβη .Γενικά τα θέματα των διπλωματικών που εμπεριέχουν hardware ένα ακριβό χόμπι χρειάζεται προσεκτική οικονομοτεχνική μελέτη ιδιαίτερα εάν το ταμείο είναι περιορισμένο και θα πρέπει να κριθούν αναλόγως.

Απαιτούνται τροποποιήσεις στην εξωτερική μονάδα (OutDoor Unit) προκειμένου να γίνει λειτουργική, οι μπαταρίες και ο τρόπος λειτουργίας παρέχουν μόνο έξι ώρες αυτονομίας. Στην ενότητα 5.4.2 αναφέρονται τροποποιήσεις τόσο στο hardware όσο και στο Software που μπορούν να καταστήσουν τη μονάδα αξιόπιστη με επαγγελματικό προφίλ.

Η υποστήριξη μέσω διαδικτύου για τον προγραμματισμό του Wemos είναι πλούσια και περιλαμβάνει πολλά παραδείγματα και βιβλιοθήκες που μπορούν να βοηθήσουν στον

προγραμματισμό. Ωστόσο, η αξιοπιστία του Wemos και των συναφών αναπτυξιακών πλακετών είναι υπό συζήτηση. Υπάρχουν προβλήματα σύνδεσης με το τοπικό δίκτυο και η δυσκολία επικοινωνίας μπορεί να καθιστά τη χρήση του δύσκολη. Για να διασφαλίσετε τη σταθερότητα του Wemos, θα πρέπει να χρησιμοποιούνται πιστοποιημένα αξεσουάρ και να ακολουθούνται η συστάσεις του κατασκευαστή. Επίσης, είναι σημαντικό να επιλέξετε μια αξιόπιστη πηγή εφοδιασμού για τις αναπτυξιακές πλακέτες και τους αισθητήρες σας, προκειμένου να αποφύγετε προβλήματα από αντιγραφή προϊόντων ή ανεπίσημους πωλητές.

Όπως φαίνεται και από το [NiK\\_IoT](#) όλο σύστημα είναι δαιδαλώδες με πολλά στοιχεία και συνδέσεις. Μια αστοχία στο κουμπί ελέγχου ON/OFF μπορεί να οφείλεται σε πολλούς παράγοντες, όπως την έκδοση του Virtuino, την παροδική απουσία σύνδεσης στο διαδίκτυο, την κακή σύνδεση των αγωγών, την αποτυχία επανασύνδεσης και ακόμα και σε τρίτους, όπως οι Servers.

Στην αρχική φάση των πειραματισμών, ο έλεγχος του ρελέ γινόταν μέσω topic στον HiveMQ. Κατόπιν διαπιστώθηκε ότι αυτή η λειτουργία είχε παροδικές αποτυχίες που είχαν να κάνουν με τον συγκεκριμένο Server. Για τον λόγο αυτό, αποφασίστηκε να αγοραστεί μια άδεια χρήσης και να τροποποιηθεί το hardware της συσκευής τοποθετώντας έναν δεύτερο επεξεργαστή, ο οποίος θα επικοινωνεί με τον ThingSpeak αυτή τη φορά και θα έχει αποκλειστική λειτουργία τον έλεγχο του ρελέ-φορτίου.

Ο ThingSpeak και η ανάλυση Matlab επιτρέπουν τη δημιουργία εξατομικευμένων εφαρμογών IoT που μπορούν να προσαρμοστούν στις ανάγκες του κάθε χρήστη. Η συνδυασμένη χρήση του ThingSpeak και του Matlab επιτρέπει στους χρήστες να συλλέγουν, να αναλύουν και να παρουσιάζουν δεδομένα από διάφορες συσκευές IoT με έναν πολύ αποτελεσματικό και ευέλικτο τρόπο. Παρέχουν εργαλεία στον προγραμματιστή για τη δημιουργία επαγγελματικών εφαρμογών με επιστημονική βάση, μέσω παραδειγμάτων.

Ένα μεγάλο λάθος μου στον αρχικό σχεδιασμό της διπλωματικής εργασίας είναι ότι δεν επέλεξα για τη δημιουργία της εφαρμογής NiK\_IoT τη νέα έκδοση του Virtuino, το "Virtuino IoT", διότι η τελευταία εκτελείται σε διάφορα γνωστά λειτουργικά συστήματα, όπως Android, iOS, MacOS, Windows, Raspberry Pi OS και όλες οι εκδόσεις LINUX. Είναι η μοναδική εφαρμογή του είδους της με δυνατότητα λειτουργίας σε πολλές πλατφόρμες. Εάν είχα επιλέξει τη νέα έκδοση, θα μπορούσα να δημιουργήσω την εφαρμογή στην οθόνη του υπολογιστή και όχι στο κινητό τηλέφωνο, επιπλέον η εφαρμογή NiK\_IoT θα δούλευε σε όλες τις παραπάνω πλατφόρμες και το πελατολόγιο θα ήταν ποιο ευρύ.

Είναι σημαντικό να συνειδητοποιήσουν οι αναγνώστες, και ιδιαίτερα οι νέοι μηχανικοί, την επεκτασιμότητα του έργου για εφαρμογές όπως το έξυπνο σπίτι, η έξυπνη γεωργία, η έξυπνη διαβίωση και οι νέοι επαγγελματικοί ορίζοντες που ανοίγονται με όριο τη φαντασία. Το Διαδίκτυο των Πραγμάτων (IoT) είναι ένας θαυμάσιος κόσμος και η επανάσταση που έχει συντελεστεί στις τεχνολογίες επικοινωνίας παρέχει τη δυνατότητα για πλήρη εποπτεία και έλεγχο των συστημάτων ηλεκτρικής ενέργειας, καθιστώντας τη λειτουργία τους αποδοτικότερη, οικονομικότερη και ευέλικτη. Το NiK\_IoT αποτελεί μια εφαρμογή διαχείρισης συστημάτων IoT που μπορεί να μετατρέψει ένα παλιομοδίτικο σπίτι σε ένα έξυπνο σπίτι και να παρέχει έλεγχο και διαχείριση συσκευών από οποιαδήποτε πλατφόρμα.

## 7 Βιβλιογραφία – Αναφορές.

---

- [1] «abb.com,» [Ηλεκτρονικό]. Available: <https://new.abb.com/low-voltage/products/electronicrelays/monitors/temperature-monitoring-relays>. [Πρόσβαση April 2023].
- [2] «sonoff,» [Ηλεκτρονικό]. Available: <https://sonoff.tech/product/diy-smart-switches/powr3/>. [Πρόσβαση April 2023 ].
- [3] «alldatasheet.com -Dallas temperature sensor,» [Ηλεκτρονικό]. Available: <https://www.alldatasheet.com/datasheet-pdf/pdf/58557/DALLAS/DS18B20.html>.
- [4] T. K. Risteska Stojkoska BL, «A review of Internet of Things for smart home: Challenges and solutions,» *Journal of Cleaner Production* , 2016.
- [5] S. G. H. Soumyalatha, «Study of IoT: Understanding IoT Architecture, Applications, Issues and Challenges,» *International Journal of Advanced Networking & Applications (IJANA)*, 2016.
- [6] M. B. I. R. a. M. A. M. M. R. Alam, «A Review of Smart Homes—Past, Present, and Future,» *in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190-1203, November 2012.
- [7] "Thing Speak," [Online]. Available: <https://thingspeak.com/>. [Accessed 2023].
- [8] «thingspeak,» April 2023. [Ηλεκτρονικό]. Available: [https://thingspeak.com/apps/matlab\\_analyses](https://thingspeak.com/apps/matlab_analyses).
- [9] «hivemq,» [Ηλεκτρονικό]. Available: <https://www.hivemq.com/>. [Πρόσβαση pril 2023].
- [10] «mqtt.org,» [Ηλεκτρονικό]. Available: <https://mqtt.org/>. [Πρόσβαση April 2023 ].
- [11] «ESP32 Analog To Digital Conversion Accuracy,» [Ηλεκτρονικό]. Available: <https://w4krl.com/esp32-analog-to-digital-conversion-accuracy/>. [Πρόσβαση April 2023].
- [12] «Esp32,» [Ηλεκτρονικό]. Available: <http://esp32.net/>. [Πρόσβαση 2023].
- [13] «lastminuteengineers.,» [Ηλεκτρονικό]. Available: <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>. [Πρόσβαση April 2023].
- [14] «esp32 refernece manual,» [Ηλεκτρονικό]. Available: [https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=0CAMQw7AJahcKEwjQqsNk18n-AhUAAAAAHQAAAAQAw&url=https%3A%2F%2Fwww.espressif.com%2Fsites%2Fdefault%2Ffiles%2Fdocumentation%2Fesp32\\_technical\\_reference\\_manual\\_en.pdf&psig=AOV](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=0CAMQw7AJahcKEwjQqsNk18n-AhUAAAAAHQAAAAQAw&url=https%3A%2F%2Fwww.espressif.com%2Fsites%2Fdefault%2Ffiles%2Fdocumentation%2Fesp32_technical_reference_manual_en.pdf&psig=AOV). [Πρόσβαση April 2023].
- [15] «aliexpress,» [Ηλεκτρονικό]. Available: [https://www.aliexpress.com/item/1005004200665806.html?spm=a2g0o.productlist.main.5.415b40c9VT02v&algo\\_pvid=3b17a60f-8a54-4300-bb25-d54a8f0141c7&algo\\_exp\\_id=3b17a60f-8a54-4300-bb25-d54a8f0141c7-](https://www.aliexpress.com/item/1005004200665806.html?spm=a2g0o.productlist.main.5.415b40c9VT02v&algo_pvid=3b17a60f-8a54-4300-bb25-d54a8f0141c7&algo_exp_id=3b17a60f-8a54-4300-bb25-d54a8f0141c7-)

2&pdp\_npi=3%40dis%21EUR%2151.08%2127.07%21%21%21%21%40211b.  
[Πρόσβαση April 2023 ].

- [16] Γ. Σεφερης, Ελενη, [http://ebooks.edu.gr/ebooks/v/html/8547/2702/Keimena-Neoellinikis-Logotechnias\\_B-Lykeiou\\_html-empl/index\\_c\\_02\\_02.html](http://ebooks.edu.gr/ebooks/v/html/8547/2702/Keimena-Neoellinikis-Logotechnias_B-Lykeiou_html-empl/index_c_02_02.html).
- [17] «ΔΕΔΔΗΕ,» [Ηλεκτρονικό]. Available: <https://deddie.gr/el/upiresies/nikterino-timologio/>. [Πρόσβαση April 2023].
- [18] «[https://www.alldatasheet.com/view.jsp?Searchword=Ds18b20%20datasheet&gclid=Cj0KCQjw3a2iBhCFARIsAD4jQB3421EW0rIKg2rrfZvAC5gQjVXi2t0APBYSHgtaE-YXoDsbGjLAnxAaAnaSEALw\\_wcB](https://www.alldatasheet.com/view.jsp?Searchword=Ds18b20%20datasheet&gclid=Cj0KCQjw3a2iBhCFARIsAD4jQB3421EW0rIKg2rrfZvAC5gQjVXi2t0APBYSHgtaE-YXoDsbGjLAnxAaAnaSEALw_wcB),» [Ηλεκτρονικό]. Available: [https://www.alldatasheet.com/view.jsp?Searchword=Ds18b20%20datasheet&gclid=Cj0KCQjw3a2iBhCFARIsAD4jQB3421EW0rIKg2rrfZvAC5gQjVXi2t0APBYSHgtaE-YXoDsbGjLAnxAaAnaSEALw\\_wcB](https://www.alldatasheet.com/view.jsp?Searchword=Ds18b20%20datasheet&gclid=Cj0KCQjw3a2iBhCFARIsAD4jQB3421EW0rIKg2rrfZvAC5gQjVXi2t0APBYSHgtaE-YXoDsbGjLAnxAaAnaSEALw_wcB). [Πρόσβαση April 2023].
- [19] «ebay.com,» [Ηλεκτρονικό]. Available: <https://www.ebay.com/itm/144884136048>.
- [20] «.ebay.com,» [Ηλεκτρονικό]. Available: <https://www.ebay.com/itm/400388077692>. [Πρόσβαση April 2023].
- [21] «horizonpfm.com,» [Ηλεκτρονικό]. Available: <https://horizonpfm.com/abb-b6-30-10-80-mini-contactor/b6-30-10-80/>. [Πρόσβαση April 2023].
- [22] «kaykas ηλεκτρολογικο υλικο,» [Ηλεκτρονικό]. Available: [https://www.kafkas.gr/viomichaniko-yliko/yliko-chamilis-tasis/viomich-revmatodotes-revmatoliptes/viomichanikoi-revmatodotes/abb-klemma-ragas-0-22-2-5mm-gkri\\_202336/?affid=4&skr\\_prm=WyJkZWE5MTkzNC05OGRILTQ4MGYtODRmOS0xZWVhMTVjYzc2N2YlDE2ODE4MDk4OTY1MTcsey](https://www.kafkas.gr/viomichaniko-yliko/yliko-chamilis-tasis/viomich-revmatodotes-revmatoliptes/viomichanikoi-revmatodotes/abb-klemma-ragas-0-22-2-5mm-gkri_202336/?affid=4&skr_prm=WyJkZWE5MTkzNC05OGRILTQ4MGYtODRmOS0xZWVhMTVjYzc2N2YlDE2ODE4MDk4OTY1MTcsey). [Πρόσβαση April 2023].
- [23] «Ebay.com,» [Ηλεκτρονικό]. Available: <https://www.ebay.com/itm/402598542550?var=672887347222>.
- [24] «.ebay.com,» April 2023. [Ηλεκτρονικό]. Available: <https://www.ebay.com/itm/394324490643>.
- [25] «ebay,» [Ηλεκτρονικό]. Available: [https://www.ebay.com/itm/374646203287?\\_trkparms=amclsrc%3DITM%26aid%3D1110013%26algo%3DHOMESPLICE.SIMRXI%26ao%3D1%26asc%3D245322%26meid%3Db8b1e7610f39457a8c114c29e9c19ccd%26pid%3D101196%26rk%3D3%26rkt%3D12%26sd%3D304833832965%26itm%3D374646203287%26rmt%3](https://www.ebay.com/itm/374646203287?_trkparms=amclsrc%3DITM%26aid%3D1110013%26algo%3DHOMESPLICE.SIMRXI%26ao%3D1%26asc%3D245322%26meid%3Db8b1e7610f39457a8c114c29e9c19ccd%26pid%3D101196%26rk%3D3%26rkt%3D12%26sd%3D304833832965%26itm%3D374646203287%26rmt%3). [Πρόσβαση April 2023].
- [26] «datasheet,» [Ηλεκτρονικό]. Available: <https://www.datasheet-pdf.info/entry/LC1-D09>. [Πρόσβαση April 2023].
- [27] «terabox.gr,» [Ηλεκτρονικό]. Available: <https://www.terabox.gr/product/metaschimatistis-entasis-ac-dx-30-60-5va/>.
- [28] «z3controls,» [Ηλεκτρονικό]. Available: <https://z3controls.com/ct-helper.php>. [Πρόσβαση April 2023].

- [29] «tutorialspoint.com,» [Ηλεκτρονικό]. Available: <https://www.tutorialspoint.com/voltage-follower-using-operational-amplifier-definition-circuit-diagram-working-and-applications>. [Πρόσβαση April 2023].
- [30] «fabmodules.com,» [Ηλεκτρονικό]. Available: <https://fabmodules.com/difference-between-single-supply-and-rail-to-rail-opamps/>. [Πρόσβαση April 2023].
- [31] I. E. Obayuwana Augustine, «researchgate.net,» [Ηλεκτρονικό]. Available: [https://www.researchgate.net/figure/Connecting-a-CT-Sensor-to-Arduino-learnopenenergymonitororg\\_fig2\\_347426251?fbclid=IwAR0IMFuLoKZaDnjVYHGXnwSfl44vbk3BQfLQm3ompPv0pw-emuJaSqzZ1Xw](https://www.researchgate.net/figure/Connecting-a-CT-Sensor-to-Arduino-learnopenenergymonitororg_fig2_347426251?fbclid=IwAR0IMFuLoKZaDnjVYHGXnwSfl44vbk3BQfLQm3ompPv0pw-emuJaSqzZ1Xw).
- [32] «adafruit.com,» [Ηλεκτρονικό]. Available: <https://www.adafruit.com/product/1085>. [Πρόσβαση April 2023].
- [33] K. V. T. Biljana L. Risteska Stojkoska, "A review of Internet of Things for smart home: Challenges and solutions," *Journal of Cleaner Production*, 1 October 2016.
- [34] «www.meidanis.gr,» [Ηλεκτρονικό]. Available: [https://www.meidanis.gr/el/shelly-em-smart-wifi-monofasikos-metritis-energeias-mini-metashimatistis-entaseos-50a-allterco-163-shellyem.html?gclid=CjwKCAjwrJ-hBhB7EiwAuyBVXS9tyaGTP4-Xs8Uq4l78mJE-LYvUMWJWRql\\_yS8ORj4KhA3f9XstxocgPwQAvD\\_BwE](https://www.meidanis.gr/el/shelly-em-smart-wifi-monofasikos-metritis-energeias-mini-metashimatistis-entaseos-50a-allterco-163-shellyem.html?gclid=CjwKCAjwrJ-hBhB7EiwAuyBVXS9tyaGTP4-Xs8Uq4l78mJE-LYvUMWJWRql_yS8ORj4KhA3f9XstxocgPwQAvD_BwE). [Πρόσβαση April 2023].
- [35] «Wemos\_esp32,» [Ηλεκτρονικό]. Available: <https://www.wemos.cc/en/latest/>. [Πρόσβαση 2023].
- [36] I. Lamprou, "Virtuino," Ilias Lamprou, [Online]. Available: <https://virtuino.com/>. [Accessed 2023].
- [37] S. t. m. rellay, *Smart temperature*.
- [38] «Public View Monitor channel,» [Ηλεκτρονικό]. Available: <https://thingspeak.com/channels/1994330>.
- [39] «MQTT: The Standard for IoT Messaging,» [Ηλεκτρονικό]. Available: <https://mqtt.org/>. [Πρόσβαση March 2023].
- [40] «Arduino IDE,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/software>. [Πρόσβαση March 2023].
- [41] «alldatasheet.com,» [Ηλεκτρονικό]. Available: <https://html.alldatasheet.com/html-pdf/1249939/TI/TLC271/92/3/TLC271.html>.
- [42] «lucid.app,» [Ηλεκτρονικό]. Available: <https://lucid.app/documents#/dashboard>. [Πρόσβαση April 2023].
- [43] «hivemq Broker,» [Ηλεκτρονικό]. Available: <https://www.hivemq.com/>. [Πρόσβαση 2023].
- [44] «grobotronics.com,» [Ηλεκτρονικό]. Available: <https://grobotronics.com/wemos-lolin32-v1.0.0-wifi-bluetooth-esp-32-based-board.html?sl=en>. [Πρόσβαση april 2023].
- [45] «lucid.app,» lucid, [Ηλεκτρονικό]. Available: <https://lucid.app/documents#/dashboard>. [Πρόσβαση April 2023].

- [46] "Ebay.com," [Online]. Available: <https://www.ebay.com/itm/254810284860>. [Accessed April 2023].
- [47] «planetcalc,» [Ηλεκτρονικό]. Available: <https://planetcalc.com/8110/>. [Πρόσβαση April 2023].
- [48] «/thingspeak,» [Ηλεκτρονικό]. Available: [https://thingspeak.com/apps/matlab\\_visualizations](https://thingspeak.com/apps/matlab_visualizations). [Πρόσβαση April 2023].
- [49] «.i-pex,» [Ηλεκτρονικό]. Available: <https://www.i-pex.com/products/rf>. [Πρόσβαση April 2023].
- [50] «ALLDATASHEET.COM,» [Ηλεκτρονικό]. Available: [https://www.alldatasheet.com/view.jsp?Searchword=Tlc271&gclid=CjwKCAjwov6hBhBsEiwAvrvN6Dc1d0ZKy\\_uk622cCKsMKzmqpuYGHUxtdv1oSl8Q9zd\\_7VW\\_fALYTRoCV8MQAvD\\_BwE](https://www.alldatasheet.com/view.jsp?Searchword=Tlc271&gclid=CjwKCAjwov6hBhBsEiwAvrvN6Dc1d0ZKy_uk622cCKsMKzmqpuYGHUxtdv1oSl8Q9zd_7VW_fALYTRoCV8MQAvD_BwE). [Πρόσβαση April 2023].
- [51] «researchgate.net,» [Ηλεκτρονικό]. Available: [https://www.researchgate.net/figure/Connecting-a-CT-Sensor-to-Arduino-learnopenenergymonitororg\\_fig2\\_347426251?fbclid=IwAR1ynFavQUj2YV0sk7dXR3GTfTFgP57YDil6uXD9Tw7AqS7O7\\_knRt9PzIM](https://www.researchgate.net/figure/Connecting-a-CT-Sensor-to-Arduino-learnopenenergymonitororg_fig2_347426251?fbclid=IwAR1ynFavQUj2YV0sk7dXR3GTfTFgP57YDil6uXD9Tw7AqS7O7_knRt9PzIM). [Πρόσβαση April 2023].
- [52] «ebay.com,» [Ηλεκτρονικό]. Available: <https://www.ebay.com/itm/154761641559?hash=item2408833e57:g:vc0AAOSwiYFhUI-0&amdata=enc%3AAQAIAAAA4B7N6H3NMlk8eQDZDEJXZitFveaY1pcqVVzPAwACTv%2Bkqi7GNt8LnOqr8E61OHj0wSoWCUYTN2c4hJkPLI9LETAfggTnafa2XsSZchKGqvaW6aaCygYiwxyznCmtalFdwnBQfCYIzLqmWYmmtK1N68VQDw>. [Πρόσβαση April 2023].
- [53] [Ηλεκτρονικό].
- [54] «Sonoff,» [Ηλεκτρονικό]. Available: <https://sonoff.tech/product/diy-smart-switches/powr3/>. [Πρόσβαση April 2023].
- [55] [Ηλεκτρονικό]. Available: <https://www.ebay.com/itm/374646204020?hash=item573aa76674:g:Rg4AAOSwfi5kQige&amdata=enc%3AAQAIAAAAwIs7Jd6BL7hEWCLV3CQIK2SRrhZR%2FXic35yqaZK8IAwZnV3KX2VynRL%2B6sbLiTEk8SQyLYTIRyfX84%2B6milCWEaxk9CMOXNODY1zmf5t%2BQYOFUAZCUafToh4hugiZ8NVkHwuDFPG0YIZbUM45v%2F>. [Πρόσβαση April 2023].
- [56] [Ηλεκτρονικό].
- [57] «ΔΕΗ,» [Ηλεκτρονικό]. Available: <https://www.dei.gr/el/gia-to-spiti/revma/g1-g1n/>. [Πρόσβαση April 2023].

## 8 Παράρτημα.

### 8.1 Arduino Code

#### 8.1.1 Arduino Code of Wemos\_1\_INDOOR\_MQTT

Στο παρακάτω link είναι διαθέσιμος ο κωδικας σε μορφή \*.ino  
[:https://drive.google.com/file/d/1dha9VX4mY6LYdfeYpbWz\\_DqbkNvyfzGO/view?usp=sharing](https://drive.google.com/file/d/1dha9VX4mY6LYdfeYpbWz_DqbkNvyfzGO/view?usp=sharing)

```
//=====

// code for wemos_1 read and publish sensors value and status
value to MQTT BROKER

// TO Status value is a random number which I sent to broker in order
to insure that the WEMOS IS ALIVE
#ifdef ESP8266
#include <ESP8266WiFi.h> // Pins for board ESP8266 Wemos-NodeMCU
#else
#include <WiFi.h>
#endif
//-----
#include <PubSubClient.h>
#include <WiFiClientSecure.h>

//---- WiFi settings-----
-----
const char* ssid = "JM-0431A0";
const char* password = "22021018200619390967";

//---- MQTT Broker settings CREDENTIAL=====
const char* mqtt_server =
"0d26b766658b4726935891df9f52ada6.s2.eu.hivemq.cloud"; // replace with
your broker url
const char* mqtt_username = "nikolaidis";
const char* mqtt_password = "!Kia6953";
const int mqtt_port =8883;

WiFiClientSecure espClient; // for no secure connection use
WiFiClient instead of WiFiClientSecure -----
-----
//WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];
```

```

// Dallas DIGITAL SENSOR SETTING PLUS LIBRARY

//read temperature values from a Dallas temperature sensor connected
to pin 15,
//Defines a global variable used to indicate an error condition. It
also defines a boolean variable used for debugging purposes.

#define DS18B20_PIN 15
#include <OneWire.h>//The OneWire library provides a way to communicate
with one-wire devices,
#include <DallasTemperature.h>//DallasTemperature library, which
provides a way to read temperature values from Dallas temperature
sensors.
OneWire oneWire(DS18B20_PIN);
DallasTemperature sensors(&oneWire);
static float ERROR_TEMPERATURE=-1001;//The line static float
ERROR_TEMPERATURE=-1001; defines a variable named ERROR_TEMPERATURE and
sets its initial value to -1001.
boolean debug=true;

//-----4 METABLHTES initialing -----
-----
//analog_value_1: a float variable that will be used to store the
value read from an analog sensor connected to pin A0 on the device.
//temperature: a float variable that will be used to store the
temperature value read from a Dallas sensor connected to pin 15 on the
device.
//feed_back_1: an integer variable that will be used to store the
value read from a digital input pin A17 on the device.
//status: an integer variable that will be used to store a randomly
generated value between 0 and 100.
float analog_value_1 = 0;
float temperature = 0;
int feed_back_1=0;
int status=0;

//----- δηλωνω τα topic ΣΟΣ ΘΑ πρεπει ως "feed_back_1" να δηλωθει
και στο Public-topic στο virtuino -----
const char* sensor1_topic = "analog_value_1";
const char* sensor2_topic = "temperature";
const char* sensor3_topic = "feed_back_1";
const char* status_topic = "status";

static const char *root_ca PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIIFazCCA10gAwIBAgIRAIIQz7DSQONZRGpGu20CiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFN1Y3VyYXR5IFJlc2Vh

```

```

cmNoIEdyb3VwMRUwEwYDVQQDEwXJU1JHIFJvb3QgWDEwHhcNMTUwNjA0MTEwNDM4
WhcNMzUwNjA0MTEwNDM4WjBPMQswCQYDVQGEwJVUzEpMCCGA1UEChMgSw50ZXJv
ZXQGU2VjdXJpdHkgUmVzZWYyZggR3JvdXAxFtATBgNVBAMTDE1TUkcGUm9vdCBY
MTCCAIiWdQYJKoZIhvcNAQEBBQADggIPADCCAgocGgIBAK3oJHP0FDfzm54rVygC
h77ct984kIxuPOZxHj3dcKi/vVqbvYATyjb3miGbESTtrFj/RQSa78f0uoxmyF+
0TM8ukj13Xnfs7j/EvEhmkvBioZxaUpmZmyPffjxwv60pIgbz5MDmgK7iS4+3mX6U
A5/TR5d8mUgjU+g4rK8Kb4Mu0U1XjIB0ttov0DiNewNwIRt18jA8+o+u3dpjq+sW
T8KOEUt+zwvo/7V3LvSye0rgTBI1DHCNAymg4VMk7BPZ7hm/ELNKjD+Jo2FR3qyH
B5T0Y3HsLuJvW5iB4Y1cNHlsdu87kGJ55tukmi8mxdAQ4Q7e2RCOFvu396j3x+UC
B5iPNgiV5+I3lg02dZ77DnKxHZu8A/1JBdiB3QW0KtZB6awBdpUKD9jf1b0SHzUv
KBds0pjBqAlkd25HN7rOrFleaJ1/ctaJxQZBKT5ZPt0m9STJEadao0xAH0ahmbWn
0lFuhjuefXKNegV4We0+UXgVCwOPjdAvBbI+e0ocS3MFEVzG6uBQE3xDk3SzynTn
jh8BCNAw1FtxNrQHusEwMFxIt4I7mKZ9YIqioymCzLq9gwQbooMDQaHwBfEbwrbw
qHyG00aoSCqI3Haadr8faqU9GY/rOPnk3sgrDQoo//fb4hVC1CLQJ13hef4Y53CI
rU7m2Ys6xt0nUW7/vGT1M0NPAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBRR5tFnme7b15AFzgAiIyBpY9umbbjANBgkq
hkiG9w0BAQsFAAOCAgEAVR9YqbyyqFDQDLHYGmkGjYkIrGF1XIpu+ILlaS/V9lZL
ubhzEFnTIzd+50xx+7LSYK05qAvqFyFWhfFQDlnrzuBZ6brJFe+GnY+EgPbk6ZGQ
3BebYhtF8GaV0nxvwuo77x/Py9auJ/GpsMiu/X1+mvoiB0v/2X/qkSsisRc0j/KK
NFtY2PwByVS5uCbMiozziUwthDyC3+6WVwW6LLv3xLfHTjuCvjHIInNzktHCgKQ5
ORAZI4JMPJ+Gs1WYHb4phowim57iaztX0oJwTdwJx4nLCgdNbOhdjsnqvHu7Ur
TkXWStAmz0VyyghqPZxjFaH3p03JLF+l+/+sKAIuvtd7u+Nxe5AW0wdeR1N8NwdC
jNPElpzVmbUq4JUagEiuTDkHszxHpFKVK7q4+63SM1N95R1NbdWhscdCb+ZAJzVc
oyi3B43njTOQ5yOf+1CceWxG1bQVs5ZufpsMlj4Ui0/1lvh+wjChP4kqK0J2qxq
4RgqsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d11TPA
mRGunUHBcnWEvgJBQ19nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwxA57d
emyPxgcYxn/eR44/KJ4EBs+1VDR3veyJm+kXQ99b21/+jh5Xos1AnX5iItreGCc=
-----END CERTIFICATE-----
)EOF";

```

```

//=====
//--- Prints a message to the serial monitor indicating that the
device is attempting to connect to a Wi-Fi network.
//Sets the Wi-Fi mode to station mode using the WiFi.mode() function.
//Attempts to connect to the specified Wi-Fi network using the
WiFi.begin() function.
// Waits for the Wi-Fi connection to be established using a while
loop that continuously checks the Wi-Fi status with the WiFi.status()
function.
//This loop also prints a series of dots to the serial monitor to
indicate that the device is attempting to connect to the network.
// Seeds the random number generator with the current value of
microseconds using the randomSeed() function.
// Prints a message to the serial monitor indicating that the device
has successfully connected to the Wi-Fi network, along with the
device's IP address.
void setup_wifi()
{

```

```

delay(10);
Serial.print("\nConnecting to.... ");
Serial.println(ssid);
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}

randomSeed(micros());
Serial.println("\nWiFi connected\nIP address: ");
Serial.println(WiFi.localIP());
}

//=====
//=====
//=====
// Loops until the MQTT client is connected to the broker using a
while loop that checks the connection status with the
client.connected() function.
// Prints a message to the serial monitor indicating that the
function is attempting to connect to the MQTT broker.
// Generates a random client ID using the random() function and
appends it to a string to create a unique client ID for the MQTT
client.
// Attempts to connect to the MQTT broker using the client.connect()
function, passing the generated client ID, username, and password as
parameters.
// If the connection is successful, prints a message to the serial
monitor indicating that the client is connected to the broker.
// If the connection fails, prints a message to the serial monitor
indicating the reason for the failure, waits for 5 seconds using the
delay() function,
// and tries to connect again.

void reconnect()
{
    // Loop until we're reconnected
    while (!client.connected())
    {
        Serial.print("Attempting MQTT connection...");
        String clientId = "wemos_Nikolaidis_Client-"; // Create a random
client ID
        clientId += String(random(0xffff), HEX);
        // Attempt to connect

```

```

// If client ESP connect
if (client.connect(clientId.c_str(), mqtt_username, mqtt_password))
{
  Serial.println("connected");
}
else
{
  Serial.print("failed, rc=");
  Serial.print(client.state());
  Serial.println(" try again in 5 seconds"); // Wait 5 seconds
before retrying
  delay(5000);
}
}
}

//===== setup
=====
//---
// Sets the Wi-Fi mode to station mode using the WiFi.mode()
function.
// Attempts to connect to the specified Wi-Fi network using the
WiFi.begin() function.
// Waits for the Wi-Fi connection to be established using a while
loop that continuously checks the Wi-Fi status with the WiFi.status()
function.
// This loop also prints a series of dots to the serial monitor to
indicate that the device is attempting to connect to the network.
// Prints a message to the serial monitor indicating that the device
has successfully connected to the Wi-Fi network, along with the
device's IP address.
void setup()
{
  // initializes a serial connection at a baud rate of 9600.
  //This will allow the microcontroller to send and receive messages
over a serial connection, which can be useful for debugging or
monitoring the system.
  Serial.begin(9600);
  //waits until a serial connection is established before continuing.
  //This is important because if the serial connection is not
established,
  // the microcontroller may not be able to send or receive data
properly.
  while (!Serial) delay(1);
  setup_wifi();
  //ifdef ESP8266 and #else are preprocessor directives that allow
the code to be compiled differently depending on which microcontroller
board is being used. In this case,

```

```

    // if an ESP8266 is being used, the next line of code
    (espClient.setInsecure()); will be executed.
    // If an ESP32 is being used, the code in the #else block will be
    executed instead
    #ifndef ESP8266
        // enables certificate verification for the MQTT connection.
        //This is more secure, but it requires a valid root certificate to
        be installed on the microcontroller.
        espClient.setInsecure();
    #else // for the ESP32
        espClient.setCACert(root_ca); // enable this line and the the
        "certificate" code for secure connection
    #endif
    // sets the MQTT server and port that the microcontroller will
    connect to.
    //This line initializes the MQTT client object,
    //which is used to send and receive messages from the server.
    client.setServer(mqtt_server, mqtt_port);

    //----- SEETINGS
    =====
    //Analog Input analog input PIN A0/36/VP in wemos
    #define ANALOG_PIN_0 36
    // sets the pinMode of pin A17 to INPUT,
    // indicating that it will be used as a digital input.
    //This is used to read feedback from a relay that has been
    connected to the Arduino.
    pinMode(A17,INPUT);
}

//===== loop
=====

void loop()
{
    //Checks if the MQTT client is connected to the broker using the
    client.connected() function.
    // If not, calls the reconnect() function to attempt to establish
    a connection.
    if (!client.connected()) reconnect();
    client.loop();

    //===== example: how to publish sensor values every 3 sec=====

    //Periodically publishes sensor values to the MQTT broker using a
    timer.

```

```

// The timer is implemented using the millis() function, which
returns the number of milliseconds since the device was powered on.
//If the elapsed time since the last message was published is
greater than 3 seconds, reads the sensor values and publishes them to
the broker
unsigned long now = millis();
if (now - lastMsg > 3000) //delay
{
  lastMsg = now;

  //====READ - GENERATE =====
  status=random(100); //generates a random integer between 0 and
99 (inclusive) and assigns it to the variable status.
  analog_value_1 = analogRead(ANALOG_PIN_0); //PIN A0/36/VP STO
WEMOS // reads an analog voltage value from pin A0 (or pin 36 or VP
on a Wemos board) and assigns it to the variable analog_value_1.
  sensors.requestTemperatures(); // Send the command to get
temperatures from pin 15 wich dalla is connected
  portDISABLE_INTERRUPTS();//disables interrupts on the
microcontroller. This is done to prevent the code from being
interrupted while reading from the Dallas temperature sensor, which
could cause errors.
  float temperature = sensors.getTempCByIndex(0);// reads the
temperature in Celsius from the first Dallas temperature sensor
connected to the microcontroller and assigns it to the variable
temperature.
  portENABLE_INTERRUPTS();//enables interrupts on the microcontroller
again.
  feed_back_1= digitalRead(A17); // reads a digital input value from
pin A17 and assigns it to the variable feed_back_1.

  //=====publish to broker
=====
  // publishing a message to an MQTT server with a topic and
payload The message will be retained on the server.
  publishMessage(status_topic,String(status),true);
  publishMessage(sensor1_topic,String(analog_value_1),true);
  publishMessage(sensor2_topic,String(temperature),true);
  publishMessage(sensor3_topic,String(feed_back_1),true);
}
}

//=====publishMessage Fuction which is used above
===== publishing as string-----

//topic: a string that specifies the topic to publish the message
to.
//payload: a string that contains the message to be published.

```

```
//retained: a boolean value that indicates whether the message  
should be retained on the broker.
```

```
//Calls the client.publish() function to publish the message to the  
specified topic on the broker, passing the topic, payload, and retained  
parameters as arguments.
```

```
void publishMessage(const char* topic, String payload , boolean  
retained)  
{  
    if (client.publish(topic, payload.c_str(), true))  
        Serial.println("Message publised ["+String(topic)+"]:  
"+payload); //If the message is successfully published, prints a  
message to the serial monitor indicating that the message has been  
published, including the topic and payload.  
}
```

## 8.1.2 Arduino Code of WEMOS\_Outdoor\_MQTT

**Στο παρακάτω Link δίνεται ο κωδικας σε μορφή \*.ino**

[:https://drive.google.com/file/d/1xyRXqmR\\_758y0N9pem11LHlr0Pfu78Wa/view?usp=sharing](https://drive.google.com/file/d/1xyRXqmR_758y0N9pem11LHlr0Pfu78Wa/view?usp=sharing)

```

#ifdef ESP8266
#include <ESP8266WiFi.h> // Pins for board ESP8266 Wemos-NodeMCU
#else
#include <WiFi.h>
#endif

//-----
// PubSubClient.h library, which is used for connecting to an MQTT broker and publishing and subscribing to
topics.
// includes the WiFiClientSecure.h library, used for establishing a secure connection to the MQTT broker over TLS
(Transport Layer Security) protocol.
#include <PubSubClient.h>
#include <WiFiClientSecure.h>

//---- WiFi settings-----
const char* ssid = "JM-0431A0";
const char* password = "22021018200619390967";

//---- MQTT Broker settings-----
const char* mqtt_server = "0d26b766658b4726935891df9f52ada6.s2.eu.hivemq.cloud"; // replace with your
broker url
const char* mqtt_username = "nikolaidis";
const char* mqtt_password = "!Kia6953";
const int mqtt_port =8883;

WiFiClientSecure espClient; // for no secure connection use WiFiClient instead of WiFiClientSecure -----
//-----
//WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];

// Dallas DIGITAAL SENSOR declaration //- ΕΔΩ ΔΗΛΩΝΩ ΤΟ PIN
//The first line defines a constant called "DS18B20_PIN" and assigns it the value 15, which specifies the pin
number used to connect the temperature sensor to the Arduino.
//The code then includes the OneWire and DallasTemperature libraries functions for communicating with the
DS18B20 temperature sensor over the OneWire bus.
// instance of the OneWire class called "oneWire" using the DS18B20_PIN constant, which initializes the
communication with the temperature sensor.
// creates an instance of the DallasTemperature class called "sensors" using the previously created OneWire
instance "oneWire",
// which allows for reading the temperature from the DS18B20 sensor.

#define DS18B20_PIN 15
#include <OneWire.h>
#include <DallasTemperature.h>
OneWire oneWire(DS18B20_PIN);
DallasTemperature sensors(&oneWire);
static float ERROR_TEMPERATURE=-1001;
boolean debug=true;

//-----initializing ---variable -----

float temperature = 0;
int status2=0;

```

```
//----- Στο virtuiino θα πρέπει γίνει ατιστοιχη δηλωση στο πεδιο Pub -topic ετσι ωστε το
//-----esp32 και το virtuiino να βλεπουν την ιδια μεταβλητη
const char* sensor3_topic = "temperature_outdoor";
const char* status_topic2 = "status_Wemos_2";

//-----
static const char *root_ca PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIIFazCCA1OgAwIbAgIRAIQz7DSQONZRGPgu2OCiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlYXV0eS1FJlc2Vh
cmNoEEdyb3VwMRUwEwYDVQQDEwxFb3VwMRUwEwYDVQQDEwxFb3VwMRUwEwYDV
Wb3VwMRUwEwYDVQDEwxFb3VwMRUwEwYDVQDEwxFb3VwMRUwEwYDVQDEwxFb3
ZXQgU2VjdXJpdHkgUmVzZWZyY2ggR3JvdXAFTAgTBGQVNTBAMTDEITUkcqUm9vdCBY
MTCCAILwDQYJKoZIhvcNAQEBBQADggIPADCCAggIBAK3oJHP0FDfzm54rVvyc
h77ct984klxuPOZXoHj3dcKi/vVqbnYATyjb3miGbESTrFj/RQSa78f0uoxmyF+
0TM8ukj13Xnfs7j/EvEhmkvBioZxaUpmZmyPfxwv60plgbz5MDmgk7iS4+3mX6U
A5/TR5d8mUgUj+g4rk8Kb4Mu0UIXjIB0ttov0DiNewNwRt18jA8+o+u3djq+sW
T8KOEUt+zwvo/7V3LvSye0rgTBIIDHCNAymg4VMk7BPZ7hm/ELNKjD+Jo2FR3qyH
B5T0Y3HsLuJvW5iB4YlcNHlsdu87kGJ55tukmi8mxdAQ4Q7e2RCOFvu396j3x+UC
B5iPNgiV5+I3lg02dZ77DnKxHZu8A/IJBdiB3QW0KtZB6awBdpUKD9jf1b0SHzUv
KBds0pjBqAlkd25HN7rOrFleaJ1/ctaJxQZBKt5ZPt0m9STJEadao0xAH0ahmbWn
OIFuhjuefXKnEgV4We0+UXgVCwOPjdAvBbl+e0ocS3MFEvzG6uBQE3xDk3SzynTn
jh8BCNAw1FtxNrQHusEwMFxt47mKZ9YlqioymCzLq9gWQbooMDQaHwBfEbwrwb
qHyGO0aoSCq13Haadr8faqU9Gy/rOPNk3sgrDQoo//fb4hVC1CLQJ13hef4Y53Cl
rU7m2Ys6xt0nUW7/vGT1MONPAGMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV
HRMBAI8EBTADAQH/MB0GA1UdDgQWBBR5tFnme7b15AFzgaAilyBpY9umbbjANBgkq
hkiG9w0BAQsFAAOCAgEAAR9YqbyyqFDQDLHYGmkGjYkIrgF1Xlpu+ILlaS/V9ZL
ubhzEFnTIZd+50xx+7LSYK05qAvqFyFWhfQDlnrzuBZ6brJFe+GnY+EgPbk6ZGQ
3BebYhtF8GaU0nxvwuo77x/Py9auJ/GpsMiu/X1+mvoiBOv/2X/qkSsisRcOj/KK
NFY2PwByVS5uCbMiozgiUwthDyC3+6WVwW6LLv3xLFTjuCvjHlInNzktHCgKQ5
ORAZl4JMPJ+GslWYHb4phowim57iazTXXOoJwTdwJx4nLCgdNbOhdjsnvzqvHu7Ur
TkXWStAmzOVyyghqpxZjFaH3pO3JLF+!/+sKAluvtd7u+Nxe5AW0wdeRIN8NwdC
jNPElpzVmbUq4JUagEiuTDkHszxHpFKVK7q4+63SM1N95R1NbdWWhscdCb+ZAJzVc
oyi3B43njTOQ5yOf+1CceWxG1bQVs5ZufpsMlj4Ui0/1lvh+wjChP4kqKOJ2qxq
4RqgsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d11TPA
mRGunUHBcnWEvgjBQl9nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwxA57d
emyPxcYxn/eR44/KJ4EBs+IVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iltreGCC=
-----END CERTIFICATE-----
)EOF";

//=====wifi=====

//a function called setup_wifi() which is responsible for connecting the Arduino to a Wi-Fi network.
//the function sets the Wi-Fi module to station mode using the WiFi.mode() function, and attempts to
//connect to the Wi-Fi network using the WiFi.begin() function.
//The function then enters a loop that waits for the Wi-Fi module to connect to the network.
//It does this by checking the status of the Wi-Fi module using the WiFi.status() function.

void setup_wifi()
{
  delay(10);
  Serial.print("\nConnecting to.... ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("\nWiFi connected\nIP address: ");
  Serial.println(WiFi.localIP());
}

//=====
//a function called reconnect() that is responsible for reconnecting to an MQTT broker if the client becomes
disconnected.
//The client.connected() function is used to check if the client is currently connected to the broker.
//If the client is not connected, the loop continues.
//A random client ID is generated using the String class and the random() function.
//The HEX parameter is used to specify that the random number should be formatted as a hexadecimal string.
```

//The client.connect() function is then called with the generated client ID, MQTT username and password as arguments. If the client is able to connect to the broker,  
 //the function prints a message to the serial monitor indicating that it has successfully connected

```

void reconnect()
{
  // Loop until we're reconnected
  while (!client.connected())
  {
    Serial.print("Attempting MQTT connection...");
    String clientId = "wemos_Nikolaidis_Client_2"; // Create a random client ID
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    if (client.connect(clientId.c_str(), mqtt_username, mqtt_password))
    {
      Serial.println("connected");
    }
    else
    {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds"); // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

//===================================================== setup=====

void setup()
{
  Serial.begin(9600);
  while (!Serial) delay(1);
  setup_wifi();
  #ifdef ESP8266
  espClient.setInsecure();
  #else // for the ESP32
  espClient.setCACert(root_ca); // enable this line and the the "certificate" code for secure connection
  #endif
  client.setServer(mqtt_server, mqtt_port);
}

//===================================================== loop =====

void loop()
{
  if (!client.connected()) reconnect();
  client.loop();
  //--- example: how to publish sensor values every 3 sec
  //The code then checks if now - lastMsg > 3000, which means that 3 seconds have passed since the last time
the sensor values were published.
  // If 3 seconds have passed, the code proceeds to publish the sensor values to the MQTT broker.
  unsigned long now = millis();
  if (now - lastMsg > 3000)
  {
    //the lastMsg variable is updated with the current time to track when the next set of sensor values should be
published.
    lastMsg = now;
    status2=random(100);
    sensors.requestTemperatures(); // Send the command to get temperatures pin 15
    portDISABLE_INTERRUPTS();
    float temperature = sensors.getTempCByIndex(0);
    portENABLE_INTERRUPTS();
    //=====publish to broker στελνω την τιμή του dallas outdoor και του random status για
ελεγχο συνδεσιμότητας -----
    // The MQTT topic for the outdoor temperature is "temperature_outdoor",
    // while the topic for the status of the Wemos 2 board is "status_Wemos_2".
    publishMessage(sensor3_topic,String(temperature),true);
    publishMessage(status_topic2 ,String(status2),true);
  }
}

```

```

//===== publish message call when i have to publish to broker =====
publishing as string
//the publishMessage() function, which is used to publish messages to the MQTT broker.
void publishMessage(const char* topic, String payload , boolean retained)
{
  if (client.publish(topic, payload.c_str(), true))
    Serial.println("Message publised ["+String(topic)+"]: "+payload);
}

```

### 8.1.3 Arduido Code of WEMOS\_COMMAND

**Στο παρακάτω Link δίνεται ο κωδικας σε μορφή \*.ino**

[https://drive.google.com/file/d/1Y3cd4rest2wH5HPUETMg2QF94PgdUZm5Tf/view?usp=sha  
ring](https://drive.google.com/file/d/1Y3cd4rest2wH5HPUETMg2QF94PgdUZm5Tf/view?usp=sharing)

```

////data from ThingSpeak channels and controls a relay
// Control a pin 4 via virtuino app and via thing speak server command channel
// Στο αρχείο secrets.h δηλώνω τα credentials toy WIFI AND THINGSPEAK
// THIS file must in the same folder of this programam

#include <WiFi.h>
#include "secrets.h"
#include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros

char ssid[] = SECRET_SSID; // your network SSID (name)
char pass[] = SECRET_PASS; // your network password
int keyIndex = 0; // your network key Index number (needed only for WEP)
WiFiClient client;

// Setting channel CREDENTIALS on secrets.h file
unsigned long Command_Channel_Number = SECRET_CH_ID_COMMAND;
unsigned int Command_FieldNumber = 1;

void setup()
{
  Serial.begin(9600); //initializes the serial communication with the baud rate of 9600.
  while (!Serial)
  {
    ;
  }
  WiFi.mode(WIFI_STA); //sets the WiFi module to Station mode, which means it will connect to an existing WiFi
network as a client.
  ThingSpeak.begin(client); // ThingSpeak.begin() initializes the ThingSpeak library with the client object created
earlier.
  //-----Εδώ καθεται το relay
  pinMode(4,OUTPUT);//configures pin 4 of the Arduino board as an output pin for controlling a relay.
}

void loop()
{
  int statusCode = 0;
  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED)//checks if the WiFi module is connected to a WiFi network. If it's not
connected
  {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED)
    {
      WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected");
  }

  ///ThingSpeak.readFloatField() function reads the value of the command field (Field 1) of the COMMAND
channel and stores it in the command variable.

```

```

int command = ThingSpeak.readFloatField(Command_Channel_Number, Command_FieldNumber);

// Check the status of the read operation to see if it was successful
statusCode = ThingSpeak.getLastReadStatus();
//checks the status of the read operation using the ThingSpeak.getLastReadStatus() function.
//If the status code is 200, the read operation was successful,
// and the program prints the value of the command field to the serial monitor. If the status code is not 200,
// there was a problem with the read operation, and the program prints the HTTP error code to the serial monitor.
if(statusCode == 200)
{
  Serial.println("Command by virtuino is " + String(command) );
}
else
{
  Serial.println("Problem reading channel. HTTP error code " + String(statusCode));
}

//-----κωδικας να ανοιγει και να κλεινει το rellay
//check the value of the command variable and turn the relay on or off based on its value. If
command is 1, the relay is turned on, otherwise, it's turned off.
if (command==1 )
{
  digitalWrite(4, HIGH);
}
else
{
  digitalWrite(4, LOW);
}

  delay(1000);
}

```

## 8.2 Παράρτημα Κώδικας Virtuino



Figure 3: Προγραμματιστικό περιβάλλον δεξιά σε unlock mode

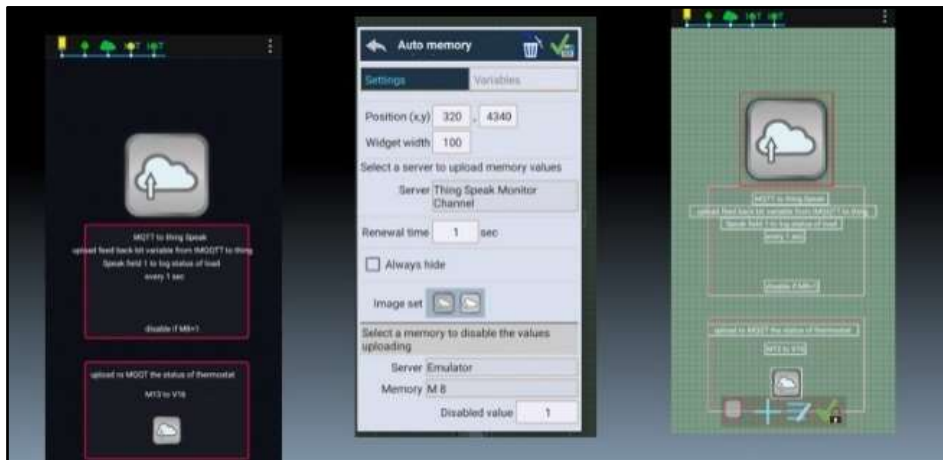


Figure 4: Upload Widget για την δημιουργία Gateway



Figure 5: Ένα μέρος από τον Κωδικά Widget στο Control Panel



Figure 6: Gateway Operation in Outdoor temperature Field Using Upload Widget



Figure 7: Διαφορα panel και λειτουργίες της εφαρμογής Virtuino



Figure 8: Κωδικας για την αλλαγή χρώματος στο frame του Outside Temperature στο panel του θερμοστάτη

Figure 9: Δειγμα κωδικα για την δημιουργία Λογικού κυκλώματος στο Control Panel;



Figure 10: Δηλώση μεταβλητών που τρέχουν στον emulator της εφαρμογής NiK\_IoT



Figure 11: Δήλωση μεταβλητών που τρέχουν στον emulator της εφαρμογής NiK\_IoT part 2



Figure 12: Δήλωση των topics στον HiveMQ Cloud Server



Figure 13: Δήλωση μεταβλητών που τρέχουν στον emulator της εφαρμογής NiK\_IoT part3



Figure 14:Ο Timer που τρέχει σε κάθε κλήση στην εφαρμογή Cal\_En

## 8.3 Παράρτημα Κώδικας MATLAB

### 8.3.1 Κώδικας MATLAB “tsimpao”

```

% (tsimpaei ) the energy value upon the request time ( M_80
Time point form Pop Up senting by virtuino )
% and put this value to Request DBMS Channel field 7

% GIAKOURIS MONITOR channel FIELD 2
% HERE i store the total energy values
% this is my DBMS

% Request DBMS Channel SETTIGS
readChannelID = 1994330;
EnergyFieldID = 2;
readAPIKey = '';
readChannelID;
writeAPIKey='WV2ZSFL7AAQALSEG';

% Ο ΠΑΡΩΝ κωδικας αρχικα
% ΔιαβαΖΕΙ τις εγγραφες ( Numspoints=1..12) στο field 6 Channel
1854983 (request channel)
%
% Στο πεδιο αυτο ανεβαζω ( απο το Virtuino ) την ημερομηνια της
οποιας της ενεργεια
% ΘΕΛΩ να καλεσω απο την DBMS ( Giakoumis channel monitor - field 2
)

% ΤΟ VIRTUINO κανει 5 εγγραφες με τον (timer στο Virtuino στο 6 )
οταν πατας
% το δευτερο απο πανω κατα σειρα κουμπι

% τα ([data]) που stelnei to virtuino apo to pop up window M80 einai
ενας καθαρος αριθμος msec
% ισον με ποσα ms απο την αρχη του χρονου του bing bang του unix
1970
[data , timestamps]= thingSpeakRead(1854983,Fields=[6], Numpoints=5)
display([data], 'field 6 request time ( uploading pop up virtuino
time) field 6 pure ms');

% η συναρτηση "omitnan" δεν λαμβανει υποψη του τις εγγραφες NaN
% Req_T einai η μεση τιμη των 5 (Numspoints ) διπλοτυπων εγγραφων
ανακατεμενες εγγραφες
% με NaN .

```

```

%Τα numspoint δεν πρπει να είναι πολλά γιατί μπορεί να πάρει και
εγγραφή απο
% προηγούμενο request

Req_T=mean(data, 'omitnan')

%-----
% δίνει το periergo special number character -6666 όταν
% η τιμή που επιστρεφέΤΕ απο το Field 6 is NaN
% όπως στην περίπτωση AN τρεχω το scriptaki TSIMPAO META απο πλθ
χρονο αφο
% στου έχω ζητήσει σηκώσει request πρώτο κουμπι στο panel virtuino
calling energy
TF=isnan(Req_T);
while TF
    thingSpeakWrite(1854983, 'fields', [7], 'Values', -
6666, 'WriteKey', writeAPIKey);
    display (-66666, 'there is no entrie of end or start time upon
which i am going to search for data entrie ')
return
end

%-----
time
%edw είναι η αρχή του χρόνου μέτρησης του virtuino
% για value = 0 pop up window σε βγαζει αυτην την ημερομηνια
Y=1970;% ετος
M=1;% μηνας
D=1;% ημερα
H=0;% ωρα
MI=0;% minute
S=[Req_T]/1000; % sec apo to virtuino request time .....

% μετατρεπει τα msec ( αριθμο ) που σελνει το virtuino σε date
Format
% (syndyasma me thn την αρχη χρόνου μέτρησης ) σε ημερομηνια
Y/M/D.....

dt2=datetime(Y,M,D,H,MI,S);
display(dt2, '----dt2----μετατροπη των ms του pop up virtuino το date
φορματ απο το matlab')
%-----

format longG ;
[data];

```

```

%-----
--
% epistrefei ton year /month /day/minute orisma as date format
% ορισμα tnh συναρτησεων ειναι το dt2
% h ημερομηνια
% Τα Y_st,M_st ..... τα χρησιμοποιω σαν ορισματα
Y_st=year(dt2);
M_st=month(dt2);
D_st=day(dt2);
H_st=hour(dt2);
MM_st=minute(dt2);
%-----

% Calculate the average energy value of last two minitew of my data
base ( ston elegxo χρησιμοποιω το
% το channel 12397 9 ( θα πρπει να το αλλαξω με το δικο μου )

% WRITE CHANNEL Giakoumis REquest Commant ( setting to καναλιου
εγγραφης )
writeChannelID = [1854983];
writeAPIKey = 'WV2ZSFL7AAQALSEG';

%-----για να παρω την τιμη της ενεργιας για μια στιγμη θα
πρπει να ζητησω για
% ενα range 2 λεπτων -1 +1 απο το date-time οπου σιγουρα θα εχω
entries και θα παρω την μεση τιμη αυτων

[data,timestamps,channelInfo]=thingSpeakRead(readChannelID,'Fields',Ene
rgyFieldID,DateRange=[datetime(Y_st,M_st,D_st,H_st,MM_st,00),datetime(Y
_st,M_st,D_st,H_st,MM_st + 1 ,00)])

%-----
% κωδικας εαν υπαρχουν εγγραφες στην data base στο ζητουμενο
χρονικο NaN να μην χτυπαει

%-----
% χρησιμοποιω και παλι την συναρτηση 'omitnan' γαιτι μπορει να μην εχω
εγγραφη
% π.χ σε περιπτωση που ζητησω μια ημερομηνια π.χ 1980
% ή απο σφαλμα μη εγγραφης
display (mean(data,'omitnan'),'Average value of given range malakas
will be writen to field 7')
%-----
TF2=isnan(mean(data,'omitnan'));

```

```

while TF2
    display ( -9999,'there is no entries on DBMS in this given
    Fucking time ');
    thingSpeakWrite(1854983,'fields',[7],'Values',-
9999,'WriteKey',writeAPIKey);
return
end

%-----
% γραφει την μεση τιμη των μετρησεων στο REQUEST channel στο field
7
    thingSpeakWrite(1854983,'fields',[7]
,'Values',mean(data,'omitnan'),'WriteKey',writeAPIKey);

//=====

//-----
Here's an outline of the steps involved in the code:

    Read the energy value from the request time sent by Virtuino.
    Store the total energy values in the Giakoumis Monitor channel Field 2.
    Read the records from Field 6 Channel 1854983 (request channel).
    Upload the date of the energy that needs to be called from the DBMS to Field 6 (from Virtuino).
    Read 5 data points using the timer in Virtuino.
    Calculate the mean of the 5 data points with nan.
    Check if the value returned from the Field 6 is NaN.
    If it is NaN, return a special number character (-6666).
    If not, convert the milliseconds (number) sent by Virtuino to date format, relative to the start time of measurement
on Virtuino.

```

### 8.3.2 Κώδικας MATLAB Visualitation A histogram of outdoor temperature

```
% This code in MATLAB uses ThingSpeak to read temperature data from
%a specified channel and then creates a histogram of the data.
%The data is collected once every second
% 5 contains temperature data.
% Channel ID to read data from
readChannelID =1994330 ;

% Outdoor Temperature Field ID
TemperatureFieldID = 3;

% Channel Read API Key
% If your channel is private, then enter the read API
% Key between the '' below:
readAPIKey = '';

% Get temperature data from field 3 for the last week ( 1 week = 168
hours )

tempF = thingSpeakRead(readChannelID,'Fields',TemperatureFieldID,...
'NumMinutes',168*60, 'ReadKey',readAPIKey);
% The last three lines add labels and a title to the histogram.
%The xlabel and ylabel functions add labels to the x-axis and y-axis,
respectively.
%The title function adds a title to the histogram.
histogram(tempF);
xlabel('Temperature (Celcius)');
ylabel('Number of Measurements\nnewline for Each Temperature');
title('Histogram of OutDoor Temperature ');
```

### 8.3.3 Κώδικας MATLAB Visualitation Consumption of three days Current - Last week - Last month

```

% Read current consumption from a ThingSpeak channel for three separate
days
% and visualize the data in a single plot using the PLOT function.
% Channel ID to read data from
% Giakoumis Monitor Channel
% Field 4 Current consumption Load
readChannelID = 1994330;
% Temperature Field ID
myFieldID = 4;
% One day date range
oneDay = [datetime('yesterday') datetime('today')];

% Channel Read API Key
% If your channel is private, then enter the read API key between the
'' below:
readAPIKey = '';

% Read Temperature Data. Learn more about the THINGSPEAKREAD function
by
% going to the Documentation tab on the right side pane of this page.
temperatureDay1 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
    'dateRange', oneDay,
    'ReadKey',readAPIKey);
temperatureDay2 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
    'dateRange',oneDay-
days(7),'ReadKey',readAPIKey);
temperatureDay3 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
    'dateRange', oneDay-
days(30),'ReadKey',readAPIKey);
y_min = 0;
y_max = 30;
% Create array of durations
myTimes1 = minutes(1:length(temperatureDay1));
myTimes2 = minutes(1:length(temperatureDay2));
myTimes3 = minutes(1:length(temperatureDay3));

% Visualize the data

plot(myTimes1,temperatureDay1, myTimes2,temperatureDay2, myTimes3,
temperatureDay3);
ylim([y_min, y_max]);
legend({'Current Day ', 'Day Before 1 week ', 'Day Before One month '});
xlabel('Minutes');
ylabel('Current Consumption');
title('3-Day Current Consumption Comparison ');

```

## 8.4 Παράρτημα Φωτογραφίες της κατασκευής του Νικ\_IoT



Figure 15: Στο εργαστήριο του τμήματος για έλεγχο και επίδειξη



Figure 16: Video από τις δοκιμές φορτιού κάντε κλικ πάνω στην εικόνα

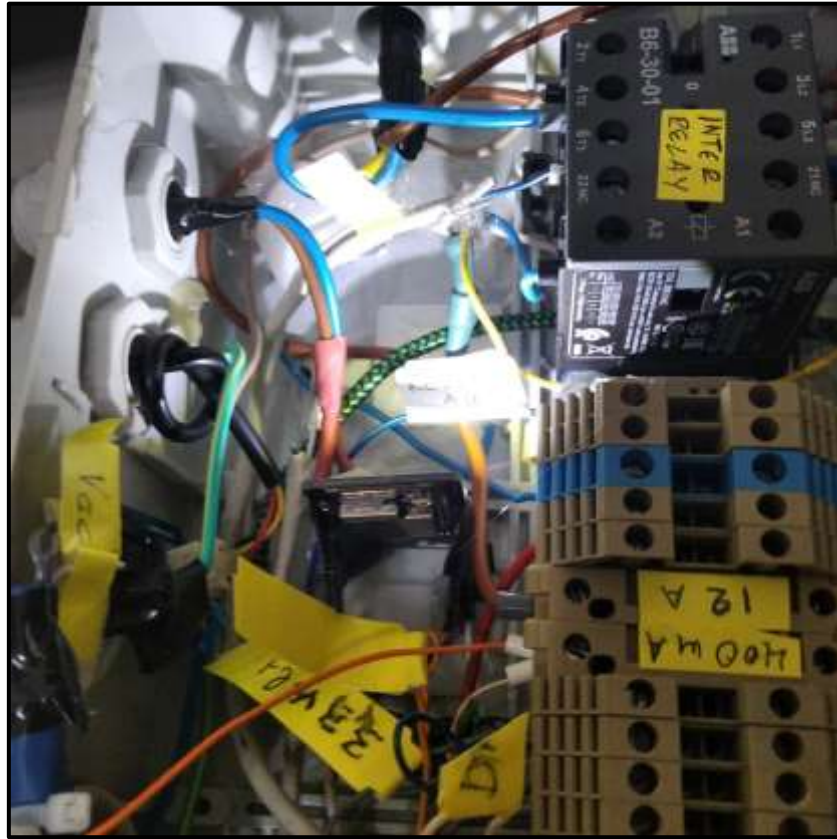


Figure 17 Scam στο Indoor Unit

:

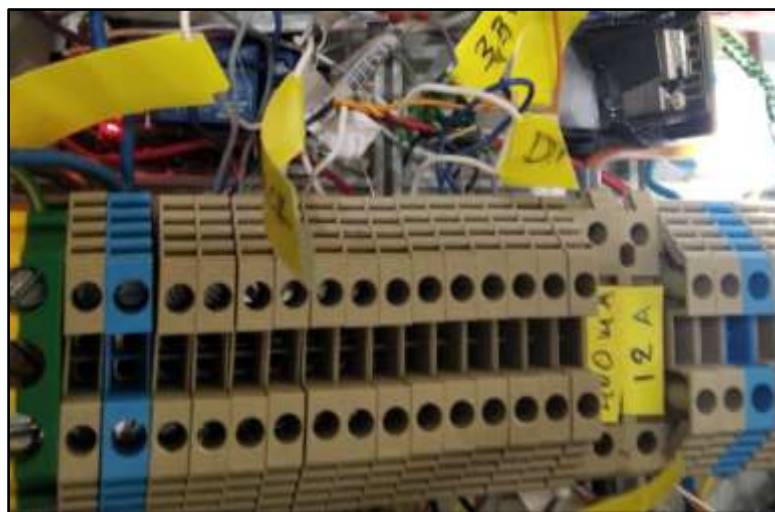


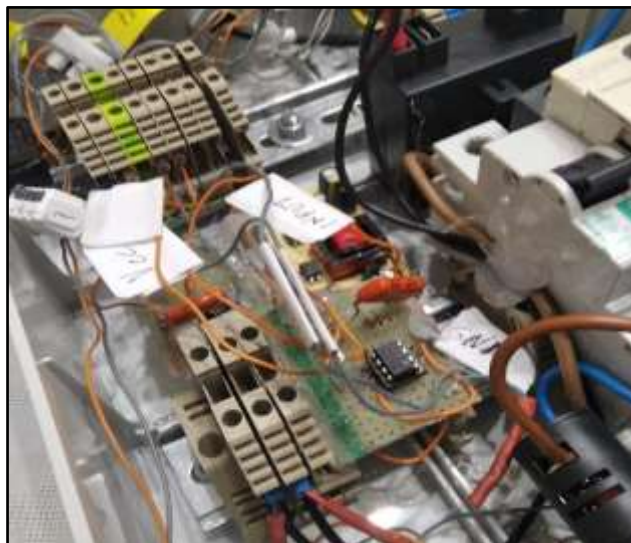
Figure 18:Κλεμμοσειρες



**Figure 19:** Εξοδος της Outdoor Unit με την ενδεικτική λυχνία



**Figure 21 :**H Outdoor Unit



**Figure 20:** Isolation Amplifier in Outdoor Unit



**Figure 23:** Η Εξωτερική κεραία της Outdoor Unit



**Figure 22:** Κουμπωτή κλεμα για την σύνδεση του InDoor Unit με το OutDoor Unit Analog output



**Figure 24 :Η Indoor με τους στηπιοθληπτες συνδεσης**



**Figure 25:Εξοδος της N.O επαφής**

