



Σχολή Θετικών
Επιστημών
Τμήμα Πληροφορικής &
Τηλεπικοινωνιών

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**“ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΞΥΠΝΩΝ ΚΟΥΤΙΩΝ”**

ΚΥΡΙΑΚΟΣ ΜΠΑΤΖΑΚΗΣ
ΑΜ:2113126

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ
ΣΤΑΜΟΥΛΗΣ ΓΕΩΡΓΙΟΣ
ΣΥΝΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ
ΔΑΔΑΛΙΑΡΗΣ ΑΝΤΩΝΙΟΣ

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Περίληψη

Τα τελευταία χρόνια οι συσκευές που χρησιμοποιούνται στην καθημερινότητα των ανθρώπων συνεχώς και αυξάνονται. Για την καλύτερη αξιοποίηση του γεγονότος αυτού (προσφέροντας μας έλεγχο σε όλες αυτές τις συσκευές, με απλό και εύκολο τρόπο, οποιαδήποτε στιγμή, από οποιοδήποτε μέρος), έρχεται στην ζωή του ανθρώπου η έννοια του Internet of Things, με σκοπό να συνδέσει όλες αυτές τις συσκευές στο διαδίκτυο, κάνοντάς τις «έξυπνες». Αρχικά, η παρούσα εργασία αναφέρεται στις γενικές κατηγορίες του Internet of Things, όπως επίσης και σε βασικές πτυχές στις οποίες βρίσκει εφαρμογή, με στόχο την καλύτερη κατανόηση, της ταχύτατα εξελισσόμενης αυτής έννοιας (IoT). Ακόμα επεξηγείται η μεθοδολογία με την οποία φτιάχτηκε το Έξυπνο Κουτί, καθώς και αναλύονται όλα τα υλικά τα οποία χρησιμοποιήθηκαν. Στην συνέχεια, παρουσιάζονται τα αποτελέσματα, αναδεικνύοντας τον τρόπο λειτουργίας ελέγχου διαχείρισης του συστήματος, καθώς και της σωστής λειτουργικότητας του Έξυπνου Κουτιού. Τέλος, προτείνονται τρόποι και ιδέες, βάση υλοποίησης των οποίων, το προϊόν που κατασκευάστηκε, θα μπορούσε να αποτελεί κάλλιστα ένα βιομηχανικό παρασκεύασμα του Internet of Things.

Abstract

In recent years, devices that are used in people's daily routines are constantly growing. To make best use of this fact (by giving us control over all these devices, in a simple and easy way, at any time, from anywhere), the concept of Internet of Things comes to human life by connecting all these devices on the internet, which results in making them <smart>. Initially, this paper refers to the general categories of the Internet of Things, as well as to key aspects to which it applies, in order to better understand this rapidly evolving concept (IoT). It also presents the methodology based upon the Smart Box was constructed, as well as analyzing all the materials that were used. Next, the results are presented, highlighting how the systems's management control operates, as well as Smart Box's functionality. Finally, ideas and methods have been proposed, which if they were to be implemented, the Smart Box could might as well be an industrial Internet of Things production.

Ευχαριστίες

Η παρούσα πτυχιακή εργασία δεν θα μπορούσε να έχει ολοκληρωθεί χωρίς την υποστήριξη, παρότρυνση και ενθάρρυνση των παρακάτω ατόμων. Αρχικά, θα ήθελα να εκφράσω με ειλικρίνεια τη βαθύτατη ευγνωμοσύνη μου στον καθηγητή, συνεπιβλέποντα και επιστημονικό σύμβουλο της πτυχιακής μου κ Δαδαλιάρη Αντώνιο, που ήταν πάντα πρόθυμος να με συμβουλέψει και να με καθοδηγήσει. Οι ιδέες, η υπομονή και η εμπειρία του, έπαιξαν σημαντικό ρόλο στην ομαλή κατασκευή της εργασίας. Τον ευχαριστώ ακόμα και για την επιλογή του θέματος, καθώς οι γνώσεις που απέκτησα απ' αυτήν ήταν πολυάριθμες και πολύτιμες. Επιπλέον, θα ήθελα να εκφράσω τις ειλικρινείς ευχαριστίες μου στον επιβλέποντα της πτυχιακής μου εργασίας κ Σταμούλη Γεώργιο και καθηγητή του τμήματος Ηλεκτρολόγων Μηχανικών χάρη στον οποίο μπόρεσα να πραγματοποιήσω αυτήν την εργασία. Ιδιαίτερες ευχαριστίες στους γονείς μου , για την άνευ όρων αγάπη τους, καθώς και τη συνεχή υποστήριξή τους καθ' όλη τη διάρκεια των σπουδών μου με κάθε τρόπο.

Κυριάκος Μπατζάκης

Ιούλιος 2019

Περιεχόμενα

Περίληψη	2
Abstract	3
Ευχαριστίες	4
Περιεχόμενα	5
Κεφάλαια	5
Εικόνες	7
Πίνακες	9
1. Εισαγωγή.....	10
1.1 Υπόβαθρο.....	10
1.2 Στόχοι.....	10
1.3 Γενική Περιγραφή.....	11
1.4 Ανακεφαλαίωση.....	11
2. Internet of Things (IoT).....	12
2.1 Βασικά Χαρακτηριστικά.....	12
2.2 Πλεονεκτήματα.....	12
2.3 Μειονεκτήματα.....	12
2.4 5G.....	14
2.5 Επιχειρήσεις.....	15
2.6 Big Data.....	18
2.7 Εφαρμογές.....	19
2.7.1 Σπίτι.....	19
2.7.2 Υγεία και Ιατρική.....	20
2.7.3 Διαφημίσεις.....	20
2.7.4 Βιομηχανία.....	21
2.8 Νέφος (Cloud).....	22
2.8.1 Επικοινωνία Πύλης Δικτύου Με Νέφος.....	23
3. Υλοποίηση Έξυπνου Κουτιού (Smart Box).....	24
3.1 Υλικά που χρησιμοποιήθηκαν.....	24
3.1.1 Arduino Mega.....	24
3.1.2 Arduino Uno.....	26
3.1.3 Esp8266 Nodemcu.....	28
3.1.4 Bme280 sensor.....	29
3.1.5 Tsop1738 IR remote control receiver.....	30
3.1.6 2N2222A NPN Transistor.....	31
3.1.7 IR (Infrared) LED sensor.....	32
3.1.8 Breadboard.....	33
3.2 Μεθοδολογία.....	33

3.2.1 Λήψη Σήματος.....	33
3.2.2 Μετατροπή και Έλεγχος Σήματος.....	38
3.2.3 Ολοκλήρωση.....	43
4. Αποτελέσματα.....	52
5. Συμπεράσματα & Μελλοντικές Προεκτάσεις.....	70
Βιβλιογραφία	75
Παράρτημα Α: Πηγαίος Κώδικας.....	78
Παράρτημα Β: Ανάλυση Κόστους.....	96

Εικόνες

1) 5G & IoT.....	15
2) Χρήση του IoT σε επιχειρήσεις 2013-2018.....	16
3) Πλεονεκτήματα που παρατηρούν οι χρήστες του IoT στις επιχειρήσεις τους.....	17
4) Χρονολόγιο εξέλιξης των σύγχρονων υπολογιστών.....	18
5) Ροή Big Data στο IoT.....	19
6) Εφαρμογές IoT σε Έξυπνο Σπίτι.....	19
7) Εφαρμογές IoT στην Ιατρική.....	20
8) Πτυχές του IoT.....	21
9) Μερικές από τις καλύτερες πλατφόρμες νέφους IoT.....	22
10) Arduino Mega 2560 Αναλυτικά.....	25
11) Arduino Uno Αναλυτικά.....	26
12) ESP8266 Nodemcu Αναλυτικά.....	29
13) Bme280 sensor.....	30
14) Tsop1738.....	31
15) 2N2222A NPN TRANSISTOR όψη και προδιαγραφές.....	32
16) IR LED sensor.....	33
17) Συνδεσμολογία Arduino Mega 2560 - Tsop1738 για λήψη σήματος.....	35
18) Επιλογή σωστής πλακέτας και θύρας για Arduino Mega 2560.....	36
19) Serial Monitor του Long_IR_Receiver.....	37
20) Αποθήκευση των σημάτων σε text editor.....	37
21) Air Condition Controller.....	39
22) Συνδεσμολογία Uno/IR LED/Resistor/Transistor ελέγχου σημάτων.....	41
23) Επιλογή σωστής πλακέτας και θύρας για Arduino Uno.....	42
24) Σχηματική παράσταση τελικής συνδεσμολογίας.....	44
25) Συνδεσμολογία τελικής κατασκευής.....	45
26) Απώλεια εύρεσης πλακετών Esp8266.....	45
27) Βήμα 1.....	46
28) Βήμα 2.....	47
29) Βήμα 3.....	48
30) Βήμα 4.....	49
31) Ανέβασμα τελικού κώδικα στο Esp8266.....	50
32) Επισκόπηση συστήματος Έξυπνου Κουτιού.....	51
33) Serial Monitor έπειτα από ανέβασμα του κώδικα.....	52
34) Fullscreen της ιστοσελίδας στο Desktop.....	52
35) Serial Monitor έπειτα από την είσοδο στην ιστοσελίδα.....	53
36) Παράδειγμα λειτουργικότητας εικόνα 1.....	54
37) Παράδειγμα λειτουργικότητας εικόνα 1α.....	54
38) Παράδειγμα λειτουργικότητας εικόνα 2.....	55

39) Παράδειγμα λειτουργικότητας εικόνα 2α.....	55
40) Παράδειγμα λειτουργικότητας εικόνα 3.....	56
41) Παράδειγμα λειτουργικότητας εικόνα 3α.....	56
42) Παράδειγμα λειτουργικότητας εικόνα 4.....	57
43) Παράδειγμα λειτουργικότητας εικόνα 4α.....	57
44) Παράδειγμα λειτουργικότητας εικόνα 5.....	58
45) Παράδειγμα λειτουργικότητας εικόνα 6.....	58
46) Παράδειγμα λειτουργικότητας εικόνα 6α.....	59
47) Παράδειγμα λειτουργικότητας εικόνα 6β.....	60
48) Παράδειγμα λειτουργικότητας εικόνα 7.....	60
49) Παράδειγμα λειτουργικότητας εικόνα 7α.....	61
50) Παράδειγμα λειτουργικότητας εικόνα 8.....	61
51) Παράδειγμα λειτουργικότητας εικόνα 8α.....	62
52) Πρόσοψη Έξυπνου Κουτιού.....	62
53) Εσωτερικό Έξυπνου Κουτιού.....	63
54) Πλάγια όψη Έξυπνου Κουτιού.....	64
55) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 1.....	65
56) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 2.....	66
57) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 3.....	67
58) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 4.....	68
59) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 5.....	69
60) 128X64 OLED DISPLAY MODULE 0.96" I2C COMMUNICATION FOR ARDUINO.....	70
61) Σχεδιασμός πλακέτας τυπωμένου κυκλώματος (PCB) στο EasyDEA.....	71
62) Τελική μορφή πλακέτας τυπωμένου κυκλώματος.....	72
63) Ρυθμίσεις για Port Forwarding σε TP-LINK router.....	74

Πίνακες

1) Χαρακτηριστικά Πλακέτας Arduino Mega 2560.....	25
2) Χαρακτηριστικά Πλακέτας Arduino Uno.....	27
3) Ακρίβεια μετρήσεων του Bme280.....	29
4) Συνδεσμολογία για λήψη σήματος IR.....	34
5) Συνδεσμολογία για έλεγχο σημάτων.....	40
6) Συνδεσμολογία πλακετών για έλεγχο του αισθητήρα BME280.....	43
7) Συνολικό Κόστος Εξαρτημάτων.....	96

1. Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιαστούν το γενικό ερευνητικό πεδίο, οι στόχοι που θα επιτευχθούν στην συγκεκριμένη πτυχιακή εργασία, καθώς επίσης και μια σύνοψη των κεφαλαίων που θα ακολουθήσουν.

1.1 Υπόβαθρο

Το Internet of Things, ή αλλιώς εν συντομία IoT, είναι μια έννοια που στην ουσία αφορά την ολοκληρωτική δικτύωση των συσκευών . Τέτοιες μπορεί να είναι συσκευές που μπορούν να φορεθούν (γυαλιά/δαχτυλίδια/παπούτσια) αλλά και εργοστασιακές συσκευές. Οι συσκευές αυτές χρησιμοποιούν διάφορους αισθητήρες, με τους οποίους συγκεντρώνουν δεδομένα, τα οποία στην συνέχεια επεξεργάζονται ώστε να εκτελέσουν κάποια λειτουργία. Τρεις είναι οι λειτουργίες που μπορούν να εκτελέσουν: να συλλέγουν πληροφορίες, να στέλνουν πληροφορίες ή και τα δύο. Γενικά, το Internet Of Things προτρέπει στην συνδεσιμότητα των ανθρώπων αλλά και των επιχειρήσεων με τον κόσμο γύρω μας, δίνοντάς μας όχι μόνο την δυνατότητα ελέγχου των πραγμάτων, αλλά και ενημέρωσης της κατάστασης στην οποία βρίσκονται. Γίνεται επομένως εύκολα αντιληπτό, ότι αυτή η τεχνολογική εξέλιξη, έχει ως αποτέλεσμα τη βελτίωση του βιοτικού επιπέδου της ανθρωπότητας.

1.2 Στόχοι

Σκοπός της συγκεκριμένης εργασίας, είναι η υλοποίηση ενός Έξυπνου Κουτιού (Smart Box), το οποίο θα μπορούσε σύμφωνα με μερικές βελτιστοποιήσεις, οι οποίες θα αναφερθούν στο τελευταίο κεφάλαιο, να αποτελεί ένα βιομηχανικό προϊόν του Internet of Things . Το κουτί θα μπορεί να συλλέγει πληροφορίες από το περιβάλλον αλλά και να στέλνει πληροφορίες σε άλλες οικιακές συσκευές. Συγκεκριμένα θα μετράει την υγρασία και την θερμοκρασία του περιβάλλοντος στο οποίο βρίσκεται, με χρήση ανάλογου αισθητήρα, ο οποίος θα αναλυθεί περαιτέρω σε παρακάτω κεφάλαιο, στην κατάλληλη ενότητα. Ακόμα θα μπορεί να διαχειρίζεται το Air Condition, βάση επιλογών που θα κάνει ο χρήστης, μέσα από μια ιστοσελίδα στον υπολογιστή (desktop/laptop) του ή σαν εφαρμογή στο smartphone/tablet του.

1.3 Γενική Περιγραφή

Η πτυχιακή εργασία είναι δομημένη ως ακολούθως:

- Κεφάλαιο 2: Σκοπός αυτού του κεφαλαίου είναι η κατανόηση των γενικών και βασικών πληροφοριών, όσον αφορά το Internet of Things.
- Κεφάλαιο 3: Παρουσιάζεται η προτεινόμενη μεθοδολογία κατασκευής του τελικού μοντέλου, καθώς και των εξαρτημάτων από τα οποία αποτελείται.
- Κεφάλαιο 4: Παρουσιάζονται τα αποτελέσματα και αναλύεται η λειτουργικότητα, δηλαδή, ο τρόπος χρήσης και διαχείρισης του έξυπνου κουτιού.
- Κεφάλαιο 5: Αναφέρονται ιδέες και τρόποι ανάπτυξης, μελλοντικών προεκτάσεων, με σκοπό την βελτίωση του Έξυπνου Κουτιού (Smart Box).

1.4 Ανακεφαλαίωση

Το σχεδιασμένο και ανεπτυγμένο πρωτότυπο δίνει το προνόμιο της παρακολούθησης της κατάστασης των συσκευών καθώς και του ελέγχου, από οποιαδήποτε ηλεκτρονική συσκευή (desktop/laptop/notebook/tablet/smartphone), η οποία έχει πρόσβαση στο διαδίκτυο, με απλό και εύκολο τρόπο.

2. Internet Of Things (IoT)

2.1 Βασικά Χαρακτηριστικά

Τεχνητή Νοημοσύνη - Στην ουσία το Internet Of Things μπορεί να μετατρέψει οποιοδήποτε αντικείμενο με ηλεκτρονικό τρόπο, σε έξυπνο, εννοώντας δηλαδή ότι το βελτιώνει από κάθε άποψη, λόγω των δυνατοτήτων που προσφέρουν, η συλλογή δεδομένων, η τεχνητή νοημοσύνη και τα διαδίκτυα.

Συνδεδεσιμότητα – Λόγω της δικτύωσης του IoT, τα διαδίκτυα πλέον δεν χρειάζεται να είναι αποκλειστικά συνδεδεμένα με μεγάλες εταιρίες, αλλά μπορούν να υπάρχουν σε πολύ μικρότερη και φθηνότερη κλίμακα παραμένοντας πρακτικά. Το IoT δηλαδή δημιουργεί αυτά τα μικρά διαδίκτυα ανάμεσα στις συσκευές του συστήματός του.

Μικρές Συσκευές – Οι συσκευές με την πάροδο του χρόνου έχουν γίνει μικρότερες, φθηνότερες και πιο ισχυρές. Το IoT αξιοποιεί τις ειδικά κατασκευασμένες μικρές συσκευές του, ώστε να πετύχει ακρίβεια, δυνατότητα επέκτασης και πολυλειτουργικότητα.

Ενεργή Ενασχόληση – Στην σημερινή εποχή η αλληλεπίδραση με την συνδεδεμένη τεχνολογία γίνεται παθητικά, ωστόσο, το IoT προσφέρει νέους δρόμους για ενεργή επικοινωνία.

Αισθητήρες – Το IoT, χωρίς τους αισθητήρες, θα έχανε το χαρακτηριστικό του γνώρισμα. Είναι οι καθοριστικοί παράγοντες που μετατρέπουν το IoT από ένα απλό παθητικό διαδίκτυο σε ένα ενεργό σύστημα ικανό για ενοποίηση του πραγματικού κόσμου.

2.2 Πλεονεκτήματα

Δεδομένα – Το να έχεις περισσότερες πληροφορίες πάνω σε κάτι βοηθάει στο να κανεις καλύτερες επιλογές και να παίρνεις σωστές αποφάσεις. Ανεξάρτητα του αν είναι απλές αποφάσεις όπως το τι χρειάζεται να αγοράσεις από το παντοπωλείο ή εάν η εταιρία σου έχει αρκετές προμήθειες σε κάτι, το να γνωρίζεις τι χρειάζεται να αγοράσεις, χωρίς να χρειαστεί να ελέγξεις από μόνος σου, όχι μόνο εξοικονομεί χρόνο αλλά είναι και βολικό.

Εστιάζει στην αυτοματοποίηση και στον έλεγχο – Λόγω του ότι τα υλικά αντικείμενα συνδέονται και ελέγχονται ψηφιακά και κυρίως με ασύρματες υποδομές, υπάρχει τεράστια ποσότητα αυτοματισμού και ελέγχου στον τρόπο λειτουργίας.

Είναι αποδοτικό και εξοικονομεί χρόνο - Χωρίς την ανθρώπινη παρέμβαση οι μηχανές είναι ικανές να επικοινωνούν μεταξύ τους έχοντας ως αποτέλεσμα γρηγορότερη και έγκαιρη παραγωγή.

Προστασία σπιτιού – Το σενάριο του IoT χρησιμοποιείται για την προστασία του σπιτιού, με συσκευές που παρακολουθούνται και ελέγχονται είτε τοπικά είτε από απόσταση, με χρήση απλών εφαρμογών, οι οποίες θα είναι διαθέσιμες στις συσκευές των χρηστών. Συσκευές που χρησιμοποιούνται για προστασία σπιτιών μπορούν να είναι κάμερες, συναγερμοί, αισθητήρες, κλειδαριές και άλλα.

2.3 Μειονεκτήματα

Πολυπλοκότητα – Το Internet of Things αποτελείται από πολλαπλές τεχνολογίες των οποίων οι αρχιτεκτονικές είναι διαφορετικές μεταξύ τους. Αυτό κάνει το IoT διαδίκτυο ένα ιδιαίτερα πολύπλοκο σύστημα. Λόγω του γεγονότος αυτού, αποτυχίες στο διαδίκτυο ή σφάλματα στο λογισμικό (software) ή υλικό (hardware) θα έχουν σοβαρές επιπτώσεις. Για τον λόγο αυτό, θα απαιτούνται επιδέξιοι εργαζόμενοι που θα έχουν γνώση πολλαπλών τεχνολογιών. Συνεπώς, οι εργοδότες θα πρέπει να πληρώνουν σε αυτούς τους εργαζόμενους αρκετά υψηλούς μισθούς.

Ιδιωτικότητα – Η ιδιωτικότητα είναι μεγάλο πρόβλημα όσον αφορά το Internet of Things. Όλα τα δεδομένα θα πρέπει να είναι κρυπτογραφημένα, ώστε οι προσωπικές πληροφορίες του καθενός να μην είναι γνωστές από τους υπόλοιπους.

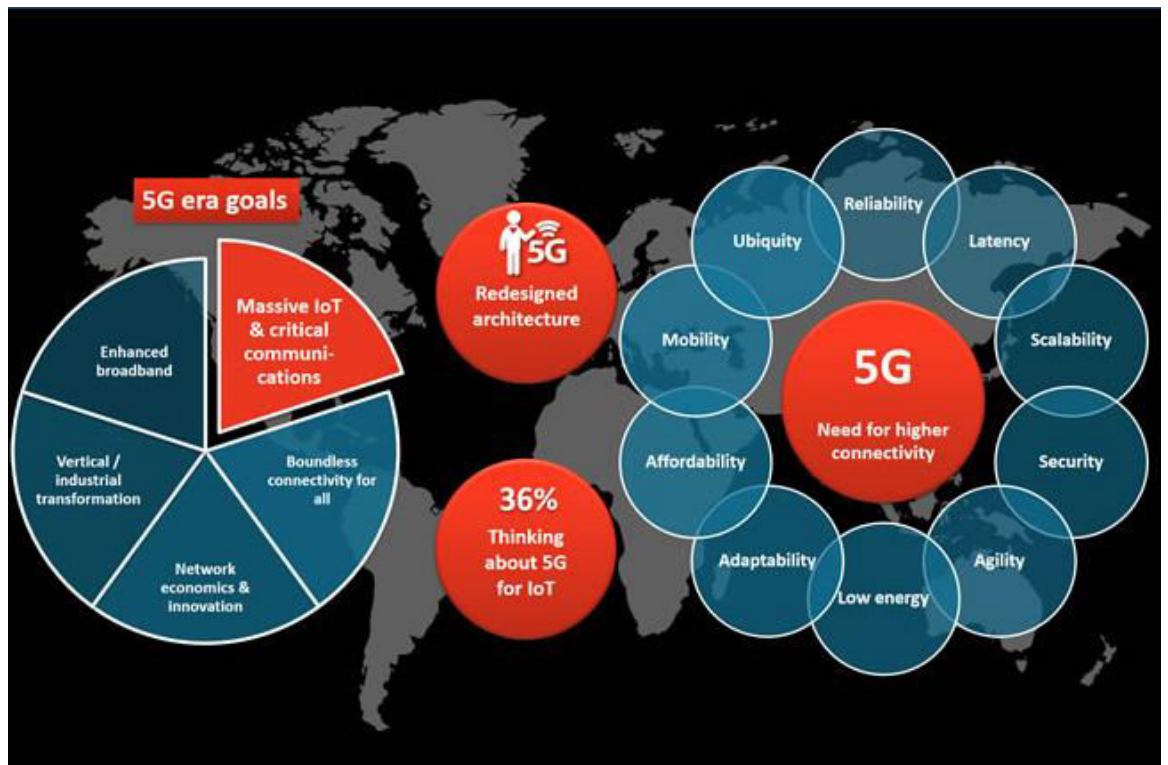
Απώλεια θέσεων εργασίας – Η εφαρμογή του IoT οδηγεί σε απώλεια θέσεων εργασίας για ανειδίκευτους εργαζόμενους, που δεν μπορούν να διαχειριστούν τις τελευταίες τάσεις απαιτήσεων σε υλικό (hardware) και λογισμικό (software), όπως άτομα που έχουν την ευθύνη της παρακολούθησης και συντήρησης των προμηθειών. Ωστόσο, κάτι τέτοιο αποτελεί πλεονέκτημα για τους εργοδότες, καθώς θα τους αποφέρει μεγάλα κέρδη από οικονομικής απόψεως.

Ασφάλεια – Υπάρχει δυνατότητα κάποιος να αποκτήσει παράνομη πρόσβαση στο λογισμικό κάποιου άλλου. Αυτό μπορεί να δημιουργήσει μεγάλα προβλήματα δεδομένου του μεγάλου εύρους πληροφοριών που θα αποκτήσει ο παραβάτης.

2.4 5G

«Το 5G είναι η πέμπτη γενιά των κινητών, κινητών τεχνολογιών, διαδικτύων και επιλύσεων. Η εξέλιξη των 5G διαδικτύων, γίνεται εύκολα ο κύριος οδηγός της ανάπτυξης των εφαρμογών του Internet of Things. Σύμφωνα με έρευνα του International Data corporation, οι παγκόσμιες 5G υπηρεσίες θα οδηγήσουν το 70% των επιχειρήσεων να ξοδέψουν 1,2 δισεκατομμύρια δολάρια σε λύσεις διαχείρισης συνδεσιμότητας (IDC, 1 Νοεμβρίου , 2017). Νέες εφαρμογές και επιχειρησιακά μοντέλα στο μέλλον του IoT απαιτούν νέα κριτήρια αποδοτικότητας όπως μαζική συνδεσιμότητα, ασφάλεια, εμπιστοσύνη, κάλυψη της ασύρματης επικοινωνίας, υπερβολικά χαμηλή αδράνεια, διακίνηση κ.α. για τεράστιο αριθμό IoT συσκευών. Για την ικανοποίηση αυτών των απαιτήσεων, οι εξελισσόμενες Long Term Evolution (LTE) και 5G τεχνολογίες αναμένονται να παρέχουν καινούριες διεπαφές συνδεσιμότητας για το μέλλον των IoT εφαρμογών.» [Shancang Li, Li Da Xu, Shanshan Zhao, 5G Internet of Things: A Survey, Journal of Industrial Information Integration (Volume 10, June 2018, Pages 1-9)]

«Το 5G μπορεί να συνεισφέρει σημαντικά στο μέλλον του Internet of Things συνδέοντας δισεκατομμύρια έξυπνες συσκευές ώστε να δημιουργηθεί ένα τεράστιο IoT, στο οποίο οι έξυπνες συσκευές θα μπορούν να αλληλεπιδρούν και να μοιράζονται δεδομένα χωρίς ανθρώπινη βοήθεια. Το IoT εξελίσσεται συνεχώς και με πολύ μεγάλη ταχύτητα, με νέα προτεινόμενη τεχνολογία, και με τις ήδη υπάρχουσες να μπαίνουν σε έναν νέο τομέα εφαρμογής. Πολλά σημερινά συστήματα του Internet of Things στοχεύουν στο να βελτιώσουν την ποιότητα της καθημερινής ζωής, το οποίο εμπλέκει την διασύνδεση ανάμεσα σε συσκευές έξυπνων σπιτιών και έξυπνων περιβάλλοντων, όπως έξυπνα σπίτια, έξυπνα κτίρια, ή ακόμα και έξυπνες πόλεις.» [Shancang Li, Li Da Xu, Shanshan Zhao, 5G Internet of Things: A Survey, Journal of Industrial Information Integration (Volume 10, June 2018, Pages 1-9)]



1) 5G & IoT [πηγή: <https://www.i-scoop.eu/internet-of-things-guide/5g-iot/>]

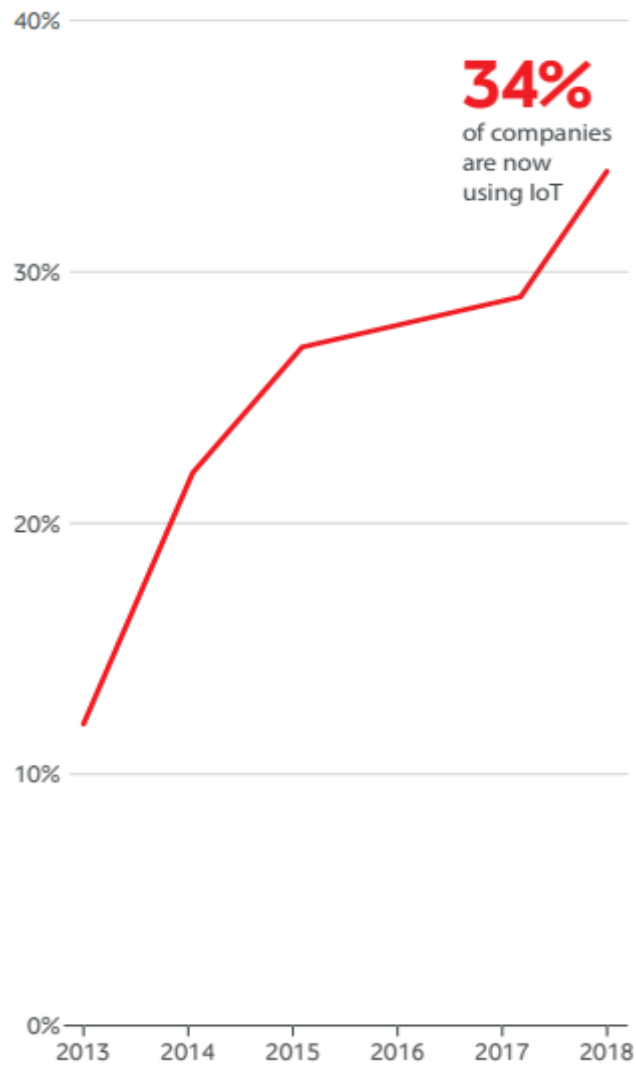
2.5 Επιχειρήσεις

Σύμφωνα με το Internet of Things Barometer της Vodafone του 2019, 34% των επιχειρήσεων χρησιμοποιούν IoT, εκ των οποίων το 76% υποστηρίζει ότι είναι υψίστης σημασίας. Το 95% αυτών που το χρησιμοποιούν έχουν ήδη δει μετρήσιμα οφέλη από την χρήση του Internet of Things, και πάνω από τους μισούς (52%) αναφέρουν ότι τα οφέλη είναι σημαντικά. Ακόμα και οργανισμοί που μόλις εφάρμοσαν τα πρώτα τους IoT έργα, αναφέρουν οφέλη.

Επίσης, το 58% χρησιμοποιούν πλατφόρμες ανάλυσης για να λαμβάνουν περισσότερα από τα IoT δεδομένα τους, ώστε να βελτιώσουν την επιχειρηματική λήψη αποφάσεων.

Το 84% θεωρούν ότι η εμπιστοσύνη τους στο IoT έχει αυξηθεί τους τελευταίους 12 μήνες.

Το 55% υποστηρίζουν ότι το Internet of Things έχει επιφέρει μεγάλες και θετικές αλλαγές στην λειτουργία των βιομηχανιών τους. Το 74% θεωρούν ότι μέσα σε 5 χρόνια οι επιχειρήσεις που δεν έχουν ξεκινήσει να χρησιμοποιούν το IoT θα έχουν ως συνέπεια τη μείωση της ανταγωνιστικότητάς τους.



2) Χρήση του IoT σε επιχειρήσεις 2013-2018[πηγή: Vodafone Business – IoT Barometer 2019]

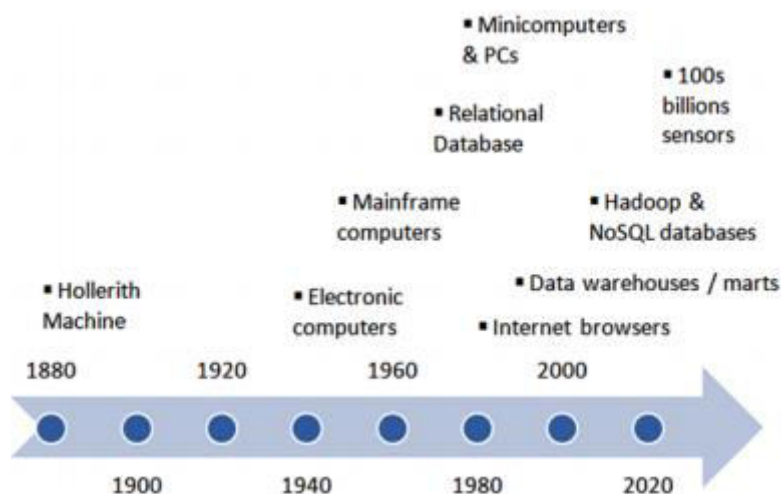
Ακόμα, το 45% που χρησιμοποιούν το Internet of Things στις επιχειρήσεις τους, ισχυρίζονται ότι τα IoT έργα τους διασχίζουν διεθνή σύνορα, με ένα επιπλέον 40% να υποστηρίζουν ότι θα τα διασχίσουν στο μέλλον. Επιπλέον, το 8% εξαρτώνται εξ' ολοκλήρου από το Internet of Things.



3) Πλεονεκτήματα που παρατηρούν οι χρήστες του IoT στις επιχειρήσεις τους [πηγή: Vodafone Business –IoT Barometer 2019]

2.6 Big Data

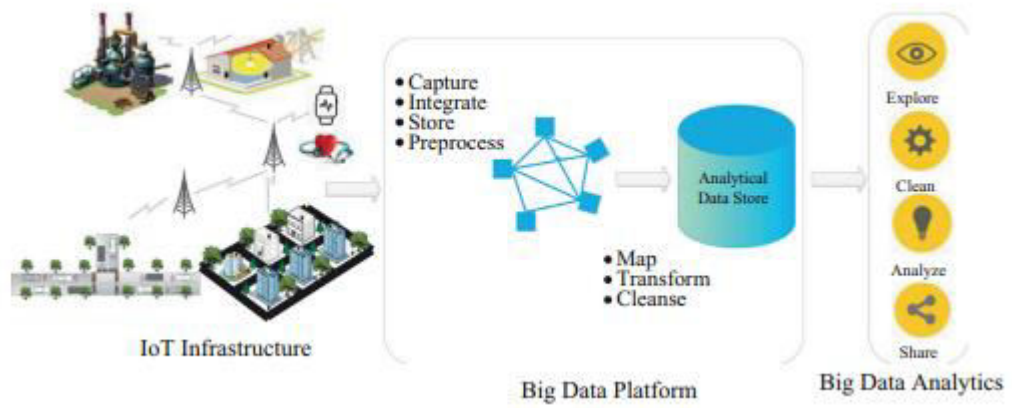
Ο όρος Big Data υπήρχε καιρό πριν έρθει το IoT ώστε να διεξάγει αναλύσεις. Όταν οι πληροφορίες καταδεικνύουν την ακρίβεια, την ταχύτητα, την ποικιλία και τον όγκο, τότε ερμηνεύονται ως Big Data.



4) Χρονολόγιο εξέλιξης των σύγχρονων υπολογιστών [πηγή: Big Data and The Internet of Things]

Το τεράστιο ποσό των εσόδων αλλά και των δεδομένων που θα δημιουργήσει το IoT, θα έχει επιπτώσεις σε όλο τον κόσμο του Big Data, αναγκάζοντας με αυτόν τον τρόπο τις επιχειρήσεις να αναβαθμίσουν τα εργαλεία και τις μεθόδους τους, καθώς και την τεχνολογία να εξελιχθεί.

«Το Internet of Things είναι μία απ' τις μεγαλύτερες πηγές Big Data, οι οποίες καθίστανται άχρηστες χωρίς την δύναμη της αναλυτικής. Το IoT αλληλεπιδρά με Big Data όταν ογκώδεις ποσότητες απαιτούνται για επεξεργασία, μετασχηματισμό και ανάλυση σε υψηλή συχνότητα. Χαρακτηριστικό γνώρισμα του Big Data Analytics είναι ότι μπορεί να διαχειρίζεται έναν μεγάλο όγκο δεδομένων που ρέουν συνεχόμενα από τις συσκευές IoT και αξιοποιούν το σύνολο δεδομένων ώστε να αυξήσουν την αποδοτικότητα των IoT εφαρμογών». [E. Ahmed et al., The role of big data analytics in Internet of Things, Computer Networks (Volume 129, Part 2, 24 December 2017, Pages 459-471)]



5) Ροή Big Data στο IoT [πηγή: E. Ahmed et al., The role of big data analytics in Internet of Things, Computer Networks (Volume 129, Part 2, 24 December 2017, Pages 459-471)]

2.7 Εφαρμογές

2.7.1 Σπίτι

Το Internet of Things, μας προσφέρει στο σπίτι μια εξατομικευμένη εμπειρία, η οποία αυξάνει την γενική μας ικανοποίηση, ενισχύει την παραγωγικότητα και βελτιώνει την υγεία και την ασφάλειά μας. Ο τρόπος που πετυχαίνει κάτι τέτοιο στο περιβάλλον του σπιτιού είναι μέσω οικιακών συσκευών, υπολογιστών, καμερών ασφάλειας κ.α.



6) Εφαρμογές IoT σε Έξυπνο Σπίτι [πηγή: <https://www.digitalvidya.com/blog/iot-and-its-impact-on-bigdata/>]

2.7.2 Υγεία και Ιατρική

Το IoT μπορεί να ενισχύσει δραματικά την ιατρική έρευνα, τις συσκευές, τη φροντίδα και την επείγουσα περίθαλψη. Η ενσωμάτωση όλων των στοιχείων παρέχει μεγαλύτερη ακρίβεια, περισσότερη προσοχή στη λεπτομέρεια, ταχύτερες αντιδράσεις στα γεγονότα και συνεχή βελτίωση με ταυτόχρονη μείωση των τυπικών εξόδων της ιατρικής έρευνας και των οργανισμών. Για παράδειγμα, συσκευές μπορούν να αισθανθούν την εμφάνιση κάποιας ασθένειας και να την αποτρέψουν.



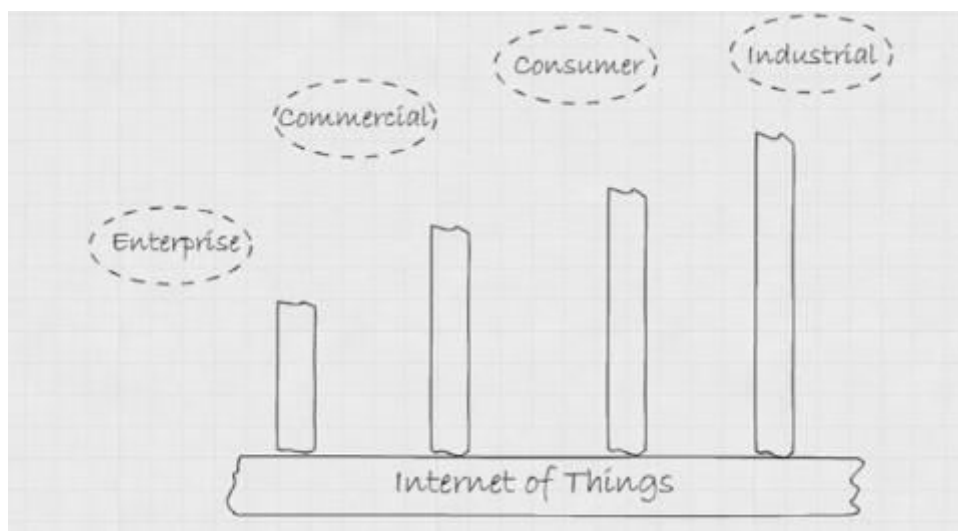
7) Εφαρμογές IoT στην Ιατρική [πηγή: <https://www.digitalvidya.com/blog/iot-and-its-impact-on-bigdata/>]

2.7.3 Διαφημίσεις

Οι σύγχρονες διαφημίσεις πάσχουν από υπερβολική και κακή στοχοποίηση. Ακόμα και με τις σημερινές αναλύσεις, η σύγχρονη διαφήμιση αποτυγχάνει. Το Internet of Things υπόσχεται διαφορετική και εξατομικευμένη διαφήμιση. Με αυτόν τον τρόπο μετατρέπει την διαφήμιση σε κάτι πρακτικό, επειδή οι καταναλωτές θα μπορούν να αλληλεπιδρούν μαζί της μέσω του IoT, αντί απλά να την λαμβάνουν. Αυτό κάνει την διαφήμιση πιο λειτουργική και χρήσιμη για άτομα που ψάχνουν στην αγορά για λύσεις ή αναρωτιούνται αν οι λύσεις αυτές υπάρχουν.

2.7.4 Βιομηχανία

Υπάρχουν διαφορετικές πτυχές όσον αφορά τις Internet of Things στρατηγικές. Τέτοιες είναι οι βιομηχανικές, οι εμπορικές, οι καταναλωτικές και οι ελεύθερες επιχειρήσεις, οι οποίες έχουν διαφορετικές τεχνικές απαιτήσεις, στρατηγικές, και στοχεύουν σε διαφορετικό κοινό. Η εμπορική αγορά, για παράδειγμα, έχει υψηλή εμπορευσιμότητα, καθώς διαθέτει υπηρεσίες που περιλαμβάνουν χρηματοπιστωτικά και επενδυτικά προϊόντα όπως τραπεζικές, ασφαλιστικές, χρηματοπιστωτικές υπηρεσίες και ηλεκτρονικό εμπόριο, οι οποίες επικεντρώνονται στο ιστορικό, την απόδοση και την αξία των καταναλωτών. Αντίστοιχα, η καταναλωτική αγορά, έχει την υψηλότερη ορατότητα, στην αγορά με την προσωπική συνδεσιμότητα έξυπνων κατοικιών μέσω συσκευών παρακολούθησης γυμναστικής, προσωπικών οθονών στο αυτοκίνητο, καθώς και ενσωματωμένων συσκευών ψυχαγωγίας. Οι ελεύθερες επιχειρήσεις IoT, από την άλλη πλευρά, είναι μια πτυχή που ενσαρκώνει μικρές, μεσαίες και μεγάλες επιχειρήσεις. Ωστόσο, η μεγαλύτερη πτυχή από αυτές είναι η βιομηχανική, η οποία περιλαμβάνει ένα τεράστιο αριθμό κλάδων όπως η παραγωγή ενέργειας, η υγειονομική περίθαλψη, η γεωργία, η μεταποίηση, το λιανικό εμπόριο, οι μεταφορές και πολλά άλλα. [Industry 4.0: The Industrial Internet of Things]



8) Πτυχές του IoT [πηγή: Industry 4.0: The Industrial Internet of Things]

2.8 Νέφος (Cloud)

Οι πλατφόρμες νέφους IoT συγκεντρώνουν τις δυνατότητες των IoT συσκευών και του υπολογιστικού νέφους (Cloud Computing), παρέχοντάς τες σαν υπηρεσία συνεχόμενης πλατφόρμας. Γενικά αναφέρονται και σαν πλατφόρμες υπηρεσίας νέφους IoT. Στην εποχή μας, που δισεκατομμύρια συσκευές είναι συνδεδεμένες στο διαδίκτυο, μπορούμε να δούμε αυξανόμενη δυνατότητα αξιοποίησης Big Data (Μεγάλων Δεδομένων) καθώς και αποτελεσματικής τους επεξεργασίας μέσω διαφόρων εφαρμογών. Οι IoT συσκευές είναι συνδεδεμένες με πολλούς αισθητήρες στο νέφος, συνήθως μέσω πύλης δικτύου (ή αλλιώς gateway, ονομάζεται το υλικό [hardware] ή το λογισμικό [software] που χρησιμοποιείται για τη σύνδεση ανάμεσα σε διαφορετικά δικτυακά περιβάλλοντα).

Σήμερα υπάρχουν πολλές πλατφόρμες νέφους IoT στην αγορά, που παρέχονται από διαφορετικούς παρόχους υπηρεσιών, μερικές από τις οποίες αναγράφονται στην εικόνα 9.



9) Μερικές από τις καλύτερες πλατφόρμες νέφους IoT [πηγή: <https://iot4beginners.com/best-internet-of-things-iot-cloud-platforms/>]

Γενικά το νέφος (Cloud) είναι όλα αυτά στα οποία μπορούμε να αποκτήσουμε απομακρυσμένη πρόσβαση μέσω του διαδικτύου. Αυτό σημαίνει πως οτιδήποτε βρίσκεται στο νέφος είναι αποθηκευμένο σε διακομιστές (servers) στο διαδίκτυο, αντί για τον υπολογιστή μας, το οποίο μας επιτρέπει να έχουμε πρόσβαση σε αυτά, από οποιονδήποτε υπολογιστή, απλά και εύκολα με σύνδεση στο διαδίκτυο.

2.8.1 Επικοινωνία Πύλης Δικτύου με Νέφος

Για την επικοινωνία της πύλης δικτύου (gateway) με το νέφος (cloud) χρησιμοποιήθηκε το Πρωτόκολλο Μεταφοράς Υπερκειμένου (Hypertext Transfer Protocol), το οποίο αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client), το οποίο πραγματοποιείται μέσω κάποιων λειτουργιών (αιτήσεων) που ονομάζονται μέθοδοι.

Οι πιο βασικές μέθοδοι είναι οι GET και POST αλλά υπάρχουν και άλλες, όπως οι HEAD / PUT / DELETE / TRACE / CONNECT / OPTIONS. Η μέθοδος η οποία χρησιμοποιήθηκε είναι η GET, η οποία ζητάει από τον διακομιστή να στείλει την σελίδα και η σελίδα κωδικοποιείται κατάλληλα σε μορφή MIME (Multipurpose Internet Mail Extensions).

Ακολουθεί ένα παράδειγμα από την λειτουργικότητα του κώδικα.

Έστω η ακόλουθη γραμμή:

GET /14/on HTTP/1.1

Η γραμμή αυτή λέγεται γραμμή αιτήσεως και περιέχει:

Το **GET** που είναι η μέθοδος που χρησιμοποιείται.

Το **/14/on** που είναι η διαδρομή του πόρου που πρέπει να προσκομιστεί.

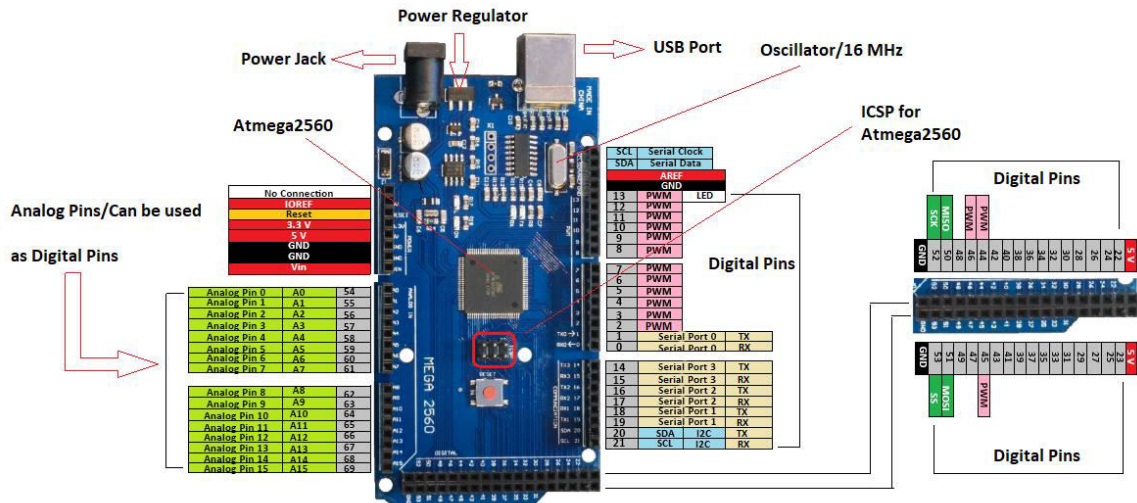
Και το **HTTP/1.1** που είναι η έκδοση του πρωτοκόλλου που χρησιμοποιείται.

3. Υλοποίηση Έξυπνου Κουτιού (Smart Box)

3.1 Υλικά που χρησιμοποιήθηκαν

3.1.1 Arduino Mega 2560

Υπάρχουν αρκετές πλακέτες βασισμένες σε μικροελεγκτές (microcontrollers), οι οποίες μπορούν να χρησιμοποιηθούν, σύμφωνα με το κατάλληλο υλικό (hardware) και τις σωστές μεταξύ τους διασυνδέσεις, όπως φυσικά και το σωστό λογισμικό (software), ώστε να δημιουργηθεί ένα Internet of Things project. Στο συγκεκριμένο project χρησιμοποιήθηκαν 3 τέτοιες πλακέτες, αν και οι δύο, όπως θα φανεί παρακάτω, δεν είναι απαραίτητες για την κατασκευή και ολοκλήρωση του Έξυπνου Κουτιού (Smart Box), παρ' ολ' αυτά χρησιμοποιήθηκαν για πρακτικούς λόγους. Η πρώτη πλακέτα που θα δούμε, είναι η Arduino Mega 2560, η οποία είναι βασισμένη στον μικροελεγκτή (microcontroller) ATmega2560 της Atmel. Το Arduino Mega 2560 περιλαμβάνει 54 ψηφιακές εισόδους/εξόδους από τις οποίες οι 15 μπορούν να χρησιμοποιηθούν σαν PWM (Pulse-Width Modulation, ρυθμιστές παλμικού πλάτους) έξοδοι, 16 αναλογικές εισόδους και 4 σειριακές θύρες. Ακόμα περιέχει μια θύρα για είσοδο USB τροφοδοσίας 5V DC, η οποία μπορεί να χρησιμοποιηθεί για προγραμματισμό (με χρήση γλώσσας C ή C++ στο Arduino IDE [Integrated Development Environment]), τροφοδοσία ή και debugging. Επίσης διαθέτει ακόμα μία είσοδο τροφοδοσίας 7V/12V DC. Γενικά το Arduino Mega 2560 είναι η πιο εξελιγμένη πλακέτα της τεχνολογίας Arduino και προτείνεται για περίπλοκες κατασκευές λόγω των πολλών εισόδων/εξόδων καθώς και της μεγάλης της μνήμης. Η τιμή της πλακέτας αυτής κυμαίνεται στο εύρος των 8-9 ευρώ (ebay) που κοστίζει ένα μοντέλο απομίμησης, μέχρι τα 35 ευρώ (arduino store) που κοστίζει το αυθεντικό μοντέλο.



10) Arduino Mega 2560 Αναλυτικά [πηγή:

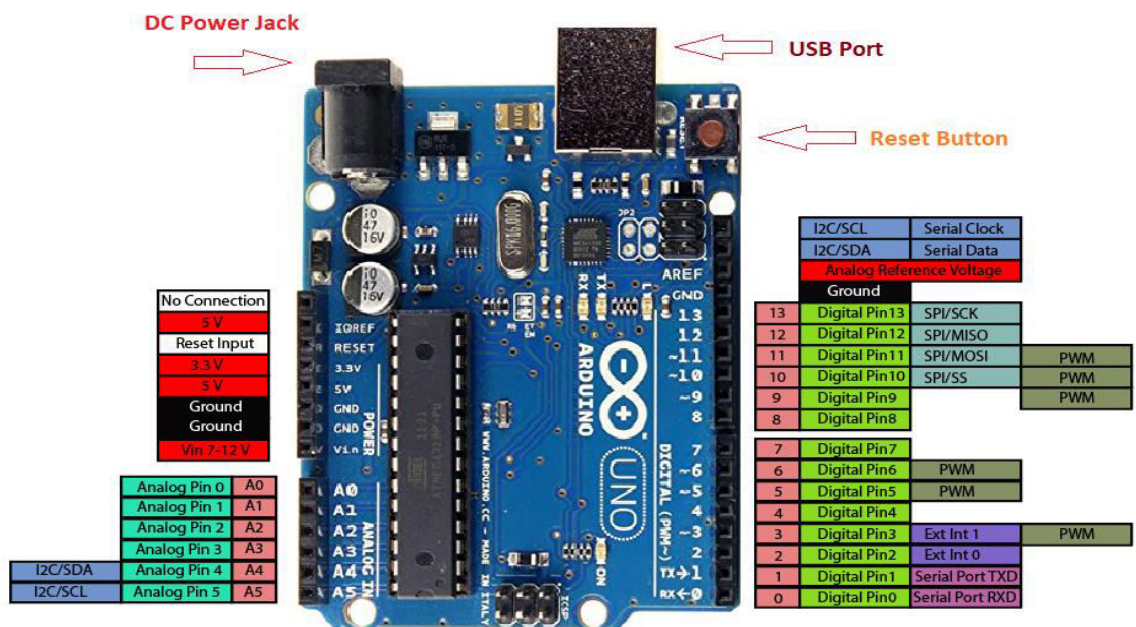
<https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-mega-2560.html>]

Μικροελεγκτής	ATmega2560
Τάση Λειτουργίας	5V
Τάση Εισόδου	7-12V
Τάση Εισόδου(min-max)	6-20V
Ψηφιακά I/O Pins	54
Αναλογικές εισοδοι	16
PWM εισοδοι	15
DC Ρεύμα ανά I/O Pin	40mA
DC Ρεύμα για 3.3V Pin	50mA
Μνήμη Flash	256 KB
Μνήμη SRAM	8 KB
Μνήμη EEPROM	4 KB
Ταχύτητα	16 MHz
Αρχιτεκτονική ελεγκτή	AVR
Διαστάσεις	101.5mm x 53.3mm
Βάρος	37g
Wi-Fi	Όχι
Συμβατότητα με Shield	Ναι

1) Χαρακτηριστικά Πλακέτας Arduino Mega 2560

3.1.2 Arduino Uno

Η δεύτερη πλακέτα που θα δούμε, είναι η Arduino Uno, που και αυτή ανήκει στην τεχνολογία Arduino, η οποία αποτελεί βασική πλακέτα της οικογένειας Arduino, γι' αυτό και πήρε το όνομά της UNO που στα Ιταλικά σημαίνει ENA, επειδή ήταν η πρώτη που είχε επικοινωνία απευθείας με τον υπολογιστή μέσω θύρας USB σε συνεργασία με το πρόγραμμα Arduino 1.0 (IDE 1.0). Η πλακέτα αυτή βασίζεται στον μικροελεγκτή (microcontroller) ATmega328 της Atmel. Το Arduino Uno δεν είναι τόσο εξελιγμένο όσο το Arduino Mega 2560, η τιμή της όμως είναι πιο προσιτή, αφού μπορεί να βρεθεί από το εύρος των 5-6 ευρώ (ebay) ενός μοντέλου απομίμησης, μέχρι και τα 20 ευρώ (arduino store) που κοστίζει το αυθεντικό μοντέλο. Το γεγονός αυτό την καθιστά μια καλή επιλογή για μικρότερα και πιο απλά project. Όμοια με την πλακέτα Arduino Mega 2560, το Arduino Uno περιέχει μια θύρα για είσοδο USB τροφοδοσίας 5V DC, η οποία μπορεί να χρησιμοποιηθεί για προγραμματισμό (με χρήση γλώσσας C ή C++ στο Arduino IDE [Integrated Development Environment]), τροφοδοσία ή και debugging. Επίσης διαθέτει ακόμα μία είσοδο τροφοδοσίας 7V/12V DC. Επίσης περιλαμβάνει 14 ψηφιακές εισόδους/εξόδους, από τις οποίες οι 6 μπορούν να χρησιμοποιηθούν σαν PWM έξοδοι και 6 αναλογικές εισόδους. Γενικά επειδή συχνά συγχέεται η έννοια μεταξύ των μικροελεγκτών (microcontrollers) και των Arduino, μπορεί να ειπωθεί με απλό τρόπο, ότι, όλα τα Arduino είναι μικροελεγκτές (microcontrollers), αλλά δεν είναι κάθε μικροελεγκτής, Arduino.



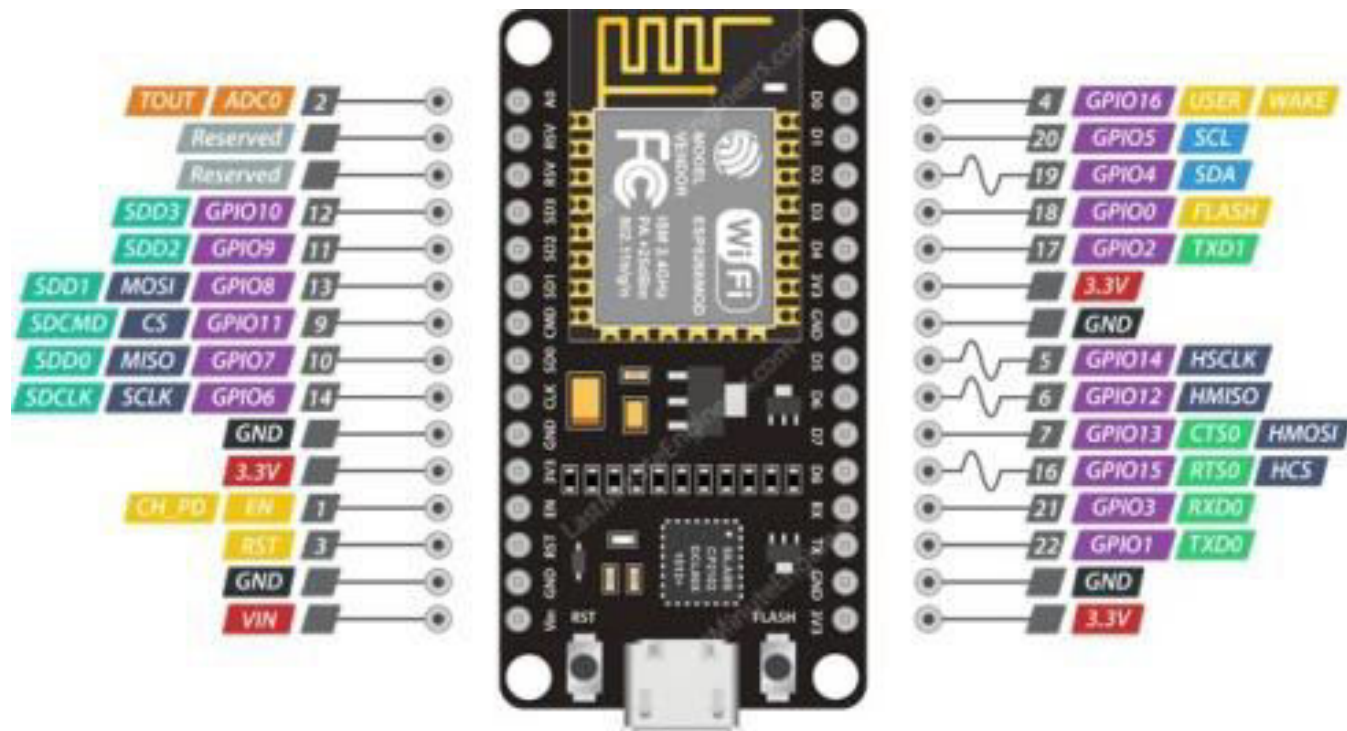
11) Arduino Uno Αναλυτικά [πηγή: <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html>]

Μικροελεγκτής	ATmega328
Τάση Λειτουργίας	5V
Τάση Εισόδου	7-12V
Τάση Εισόδου(min-max)	6-20V
Ψηφιακά I/O Pins	14
Αναλογικές εισοδοι	6
PWM εισοδοι	6
DC Ρεύμα ανά I/O Pin	20mA
DC Ρεύμα για 3.3V Pin	50mA
Μνήμη Flash	32 KB
Μνήμη SRAM	2 KB
Μνήμη EEPROM	1 KB
Ταχύτητα	16 MHz
Αρχιτεκτονική ελεγκτή	AVR
Διαστάσεις	68.6mm x 53.4mm
Βάρος	25g
Wi-Fi	Όχι
Συμβατότητα με Shield	Ναι

2) Χαρακτηριστικά πλακέτας Arduino Uno

3.1.3 ESP8266 Nodemcu

Όπως παρατηρήσαμε καμία απ' τις 2 προηγούμενες πλακέτες δεν έχει πρόσβαση στο Internet. Συνέπως από μόνες τους δεν αποτελούν προϊόν του Internet of Things. Αυτό όμως δεν αποτελεί κάποιου είδους πρόβλημα, ούτε ότι τα προϊόντα αυτά δεν μπορούν να χρησιμοποιηθούν για IoT projects. Υπάρχουν αρκετές επιλογές, για να συνδεθούν οι πλακέτες αυτές στο Internet. Ένας τρόπος είναι με τη χρήση σύνδεσης Arduino WiFi (ή Ethernet) Shield και ένας πιο απλός με την χρήση σύνδεσης κάποιου μικροελεγκτή που περιλαμβάνει WiFi, όπως το ESP-01 Serial WiFi ή το Wemos D1 mini, τα οποία κάποιος μπορεί να βρει με τιμές γύρω στα 5 ευρώ (skroutz). Φυσικά σε περίπτωση που κάποιος δεν θέλει να μπει στην διαδικασία αυτή, μπορεί να αγοράσει μικροελεγκτές που περιέχουν από μόνοι τους WiFi και δεν χρειάζονται επιπλέον σύνδεση. Τέτοιοι μικροελεγκτές, μπορεί να είναι Arduino, όπως το Arduino Uno Wifi REV2 με τιμή στα 38.90 ευρώ (arduino store), ή φθηνότεροι μικροελεγκτές που δεν ανήκουν στην οικογένεια Arduino, όπως το Wemos D1 R2 V2.1 με τιμή περίπου στα 12 ευρώ (skroutz) ή το ESP8266 Nodemcu με τιμή 8.99 ευρώ (cableworks), το οποίο και είναι ο μικροελεγκτής ο οποίος επιλέχθηκε για την δημιουργία του Έξυπνου Κουτιού (Smart Box). Η πλακέτα αυτή περιέχει το ESP-12E chip το οποίο έχει μικροεπεξεργαστή (microprocessor) TISILICA Xtensa® 32-bit LX106 RISC ο οποίος λειτουργεί με ρυθμιζόμενη ταχύτητα στα 80-160 MHz και υποστηρίζει RTOS (Real-Time Operating System). Έχει ακόμα μνήμη RAM 128 KB και μνήμη Flash 4 MB, παρατηρείται δηλαδή ότι έχει πολύ μεγαλύτερο χώρο για αποθήκευση προγραμμάτων και δεδομένων σε σχέση με τις προηγούμενες πιο ακριβές πλακέτες. Επιπλέον ενσωματώνει το 802.11b/g/n HT40 Wi-Fi transceiver, το οποίο όχι απλά το καθιστά δυνατό ώστε να συνδεθεί σε κάποιο WiFi δίκτυο και να αλληλεπιδρά με το Internet, αλλά επιπλέον μπορεί να δημιουργήσει ένα δίκτυο από μόνο του, επιτρέποντας άλλες συσκευές να συνδέονται απευθείας σε αυτό. Το γεγονός αυτό καθιστά το ESP8266 Nodemcu ακόμα πιο πολυχρηστικό. Σε αντίθεση με τις πλακέτες Arduino, το ESP8266, αντί για 2 θύρες τροφοδοσίας, έχει μόνο μία θύρα micro USB τροφοδοσίας 5V, η οποία χρησιμοποιείται για προγραμματισμό (με χρήση γλώσσας C ή C++ στο Arduino IDE), τροφοδοσία ή και debugging. Επίσης, καθώς η τάση λειτουργίας της πλακέτας κυμαίνεται στο εύρος των 3V-3.6V, διαθέτει έναν ρυθμιστή τάσης LDO ώστε να κρατάει σταθερή την τάση στα 3.3V. Ειδικότερα, το ESP8266 nodemcu διαθέτει 30 πινάκια, των οποίων οι λειτουργίες παρουσιάζονται αναλυτικότερα στην φωτογραφία παρακάτω (ESP8266 Nodemcu Αναλυτικά), από τις οποίες οι 17 χρησιμοποιούνται σαν είσοδοι/έξοδοι γενικού σκοπού, και από αυτές τις 17, οι 4 μπορούν να χρησιμοποιηθούν ως PWM έξοδοι.



12) ESP8266 Nodemcu Αναλυτικά[πηγή: <https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/>]

3.1.4 Bme280 Sensor

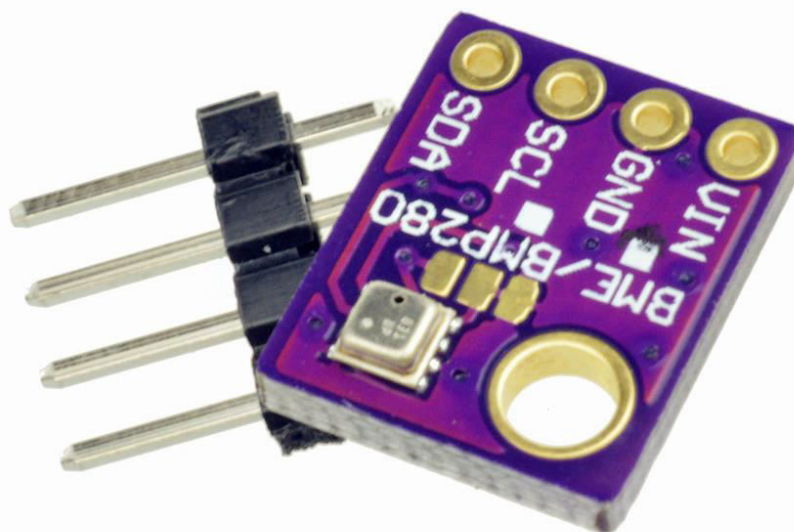
Ο αισθητήρας που χρησιμοποιήθηκε για την μέτρηση θερμοκρασίας και υγρασίας για την κατασκευή του Smart Box είναι ο Bme280, ο οποίος είναι ικανός για μετρήσεις θερμοκρασίας, υγρασίας, πίεσης και υψόμετρου. Είναι ένας αρκετά ακριβής αισθητήρας καθώς και πολύ απλός στην χρήση και σύνδεσή του. Αναλυτικά, έχει $\pm 1.0^{\circ}\text{C}$ ακρίβεια στην θερμοκρασία, $\pm 3\%$ στην υγρασία, $\pm 1\text{hPA}$ στην βαρομετρική πίεση και ± 1 μέτρο στο υψόμετρο, όπως φαίνεται και στον πίνακα.

Θερμοκρασία	-40°C με $+80^{\circ}\text{C}$ ($\pm 1.0^{\circ}\text{C}$ ακρίβεια)
Υγρασία	0 με 100% RH ($\pm 3\%$ ακρίβεια)
Βαρομετρική Πίεση	300hPA με 1100hPA ($\pm 1\text{hPA}$ ακρίβεια)
Υψόμετρο	0 με 9144μέτρα (± 1 μέτρο ακρίβεια)

3) Ακρίβεια μετρήσεων του Bme280

Ο αισθητήρας αυτός λειτουργεί στα 3.3V ή 5V, οπότε μπορεί να συνδεθεί είτε σε κάποια πλακέτα Arduino, είτε σε μια πλακέτα όπως το ESP8266 Nodemcu χωρίς κανένα πρόβλημα. Περιέχει μόνο 4 πινάκια, όπου αναλυτικά το ένα είναι το VIN το

οποίο είναι η τροφοδοσία ρεύματος της μονάδας, που όπως έχει ειπωθεί μπορεί να συνδεθεί σε 3.3V ή 5V, το GND το οποίο συνδέεται σε κάποια γείωση (GND) του Arduino (ή οποιουδήποτε μικροελεγκτή της επιλογής μας), και το SCL και SDA που χρησιμοποιούνται για I2C διασύνδεση και συνδέονται στις αντίστοιχες SCL και SDA θύρες του μικροελεγκτή της επιλογής μας. Η τιμή του συγκεκριμένου αισθητήρα κυμαίνεται στα περίπου 3 ευρώ (ebay) ή στα περίπου 2 ευρώ (aliexpress), αλλά έχει και φθηνότερες επιλογές όπως ο ίδιος αισθητήρας διαφορετικής κατασκευής ή ο dht11, που και οι 2 μπορούν να βρεθούν σε τιμή λίγο λιγότερο από 1 ευρώ (ebay).

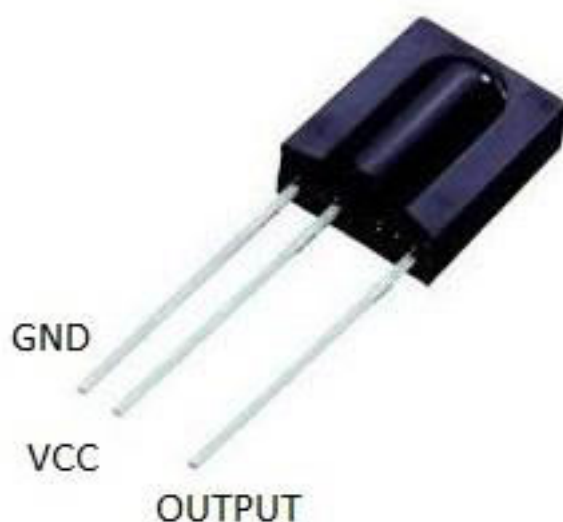


13) Bme280 sensor[πηγή: <https://www.aliexpress.com/item/32672210336.html>]

3.1.5 Tsop1738 IR Remote Control Receiver

Το Tsop1738 είναι ένας δέκτης υπέρυθρων ακτίνων τηλεχειριστηρίων, ο οποίος αποτελείται από μία δίοδο pin και έναν ενισχυτή. Η έξοδος του είναι ενεργά χαμηλή και δίνει τάση 5V σε κατάσταση εκτός λειτουργίας. Περιέχει ένα ενσωματωμένο κύκλωμα ελέγχου για την ενίσχυση των κωδικοποιημένων παλμών από τον πομπό IR (Infrared Radiation/Υπέρυθρες Ακτίνες ή αλλιώς Υπέρυθρη Ακτινοβολία]). Το σήμα δημιουργείται όταν η φωτοδίοδος pin λαμβάνει τα σήματα. Το σήμα αυτό λαμβάνεται από τον αυτόματο έλεγχο απολαβής (AGC/Automatic Gain Control). Μετά από κάποια συγκεκριμένη σειρά εισόδων, η έξοδος τροφοδοτείται πίσω στον αυτόματο έλεγχο απολαβής, ώστε να ρυθμιστεί η απολαβή αυτή σε κατάλληλο επίπεδο. Αφού ρυθμιστεί το σήμα, μεταφέρεται σε ζώνη διέλευσης φίλτρου για να φιλτράρει τις ανεπιθύμητες συχνότητες. Στη συνέχεια το σήμα πηγαίνει σε έναν αποδιαμορφωτή, όπου η έξοδος του οποίου πηγαίνει σε ένα NPN Transistor. Τέλος,

η έξοδος αυτή από το transistor, λαμβάνεται στο pin 3 του Tsop1738 (output). Υπάρχουν διαφορετικά μοντέλα της σειράς Tsop17xx και το κάθε ένα έχει διαφορετική κεντρική συχνότητα στην οποία είναι ευαίσθητο. Όπως για παράδειγμα το Tsop1738 είναι ευαίσθητο στην συχνότητα 38 kHz ενώ το Tsop1740 είναι ευαίσθητο στην συχνότητα 40 kHz.

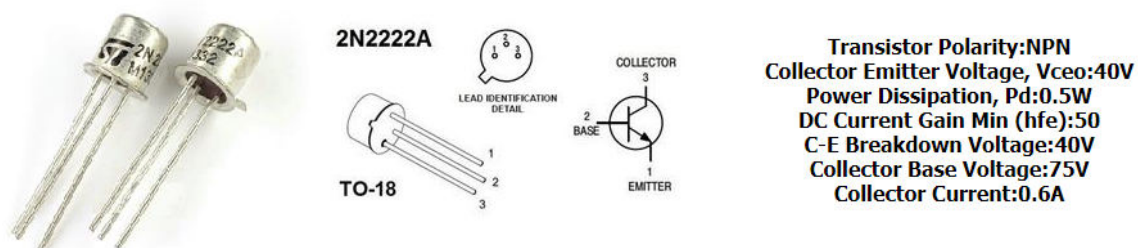


14) Tsop1738 [πηγή: <https://www.indiamart.com/proddetail/tsop-sensor-9432838012.html>]

3.1.6 2N2222A NPN Transistor

Το transistor (τρανζίστορ) , το οποίο στα ελληνικά λέγεται κρυσταλλοτρίοδος (και παλιότερα λεγόταν κρυσταλλολυχνία), είναι μία διάταξη ημιαγωγών στερεάς κατάστασης , η οποία βρίσκει διάφορες εφαρμογές στην ηλεκτρονική, μερικές από τις οποίες είναι η λειτουργία ως διακόπτης, η διαμόρφωση συχνότητας, η σταθεροποίηση τάσης, η λειτουργία ως μεταβλητή ωμική αντίσταση και η λειτουργία για την οποία και χρησιμοποιήθηκε στην συγκεκριμένη κατασκευή, η οποία δεν είναι άλλη από την ενίσχυση. Τα transistor κατασκευάζονται είτε ως τμήματα κάποιου ολοκληρωμένου κυκλώματος είτε και σαν ξεχωριστά ηλεκτρονικά εξαρτήματα. Γενικά, τα transistor μπορούν, ανάλογα με την τάση στην οποία πολώνονται, να ρυθμίζουν την ροή του ηλεκτρικού ρεύματος που απορροφούν από συνδεδεμένη πηγή τάσης. Το transistor που χρησιμοποιήθηκε στην κατασκευή του Smart Box ονομάζεται 2N2222A 2N2222 NPN Transistor 0.8A 40V TO-18 και έχει την ακόλουθη όψη και προδιαγραφές όπως βλέπουμε στην φωτογραφία που ακολουθεί. Το κόστος του βρίσκεται στα 2.46 ευρώ για την απόκτηση 10 τεμαχίων

(ebay) αλλά φυσικά μπορεί να βρεθεί κάποιο διαφορετικό transistor σε μικρότερη τιμή (για παράδειγμα κάτι λιγότερο από 1 ευρώ).

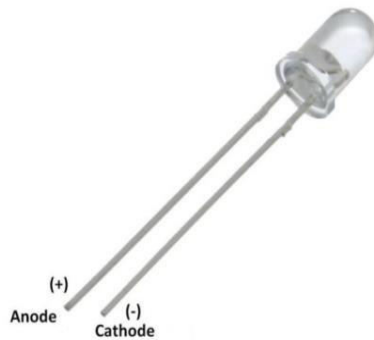


15) 2N2222A NPN TRANSISTOR όψη και προδιαγραφές

[πηγές: <https://www.ebay.com/itm/10PCS-NEW-Transistor-ST-MOTOROLA-TO-18-CAN-3-2N2222A-2N2222-AU/302185476317> + <https://www.ebay.com/itm/20-x-2N2222A-2N2222-NPN-Transistor-0-8A-40V-TO-18-/250867361510>]

3.1.7 IR (Infrared) LED sensor

Το IR (Infrared) LED (Light Emitting Diode) ή στα ελληνικά δίοδος εκπομπής φωτός υπέρυθρης ακτινοβολίας είναι ένα ειδικού σκοπού LED, το οποίο εκπέμπει υπέρυθρες ακτίνες που κυμαίνονται από 700nm έως 1mm μήκη κύματος. Όπως τα διαφορετικά LEDs παράγουν διαφορετικά είδη χρωμάτων, έτσι ακριβώς και τα διαφορετικά IR LEDs παράγουν υπέρυθρο φως με διαφορετικά μήκη κύματος. Το IR LED που χρησιμοποιήθηκε παράγει μήκος κύματος στα 940nm. Τα IR LED μαζί με τα IR Receivers (όπως το Tsop1738 που αναγράφεται παραπάνω), είναι απ' τους αισθητήρες που χρησιμοποιούνται αρκετά συχνά. Εξωτερικά το IR LED έχει την ίδια εμφάνιση με ένα απλό LED. Οι υπέρυθρες ακτίνες που παράγει ένα IR LED, παρόλο που δεν μπορούν να φανούν από ένα ανθρώπινο μάτι, με την χρήση μιας κάμερας ενός κινητού μπορούν να γίνουν ορατές, ώστε να μπορέσουμε να καταλάβουμε αν δουλεύει ή όχι. Συνήθως τα IR LED κατασκευάζονται από αρσενικό γάλλιο ή από αρσενικό γαλλίου αλουμινίου και η τιμή τους είναι αρκετά φθηνή καθώς μπορεί κάποιος να βρει τα 10, σε τιμή λιγότερο του ενός ευρώ (0.62 τα 10 – ebay). Σε ένα IR LED το θετικό πόδι (άνοδος) είναι πάντα το μεγάλο, και το αρνητικό πόδι (κάθοδος) είναι το μικρό – όπως φαίνεται και στην εικόνα 16 - συνεπώς είναι πολύ εύκολη η διάκριση τους. Για πρακτικούς λόγους, όπως για να έχει μεγαλύτερη διάρκεια ζωής το IR LED στην κατασκευή, προστέθηκε ένας αντιστάτης 120Ω. Η συνδεσμολογία τους θα αναλυθεί σε παρακάτω ενότητα.



16) IR LED sensor[πηγή: <https://electronicsforu.com/resources/learn-electronics/ir-led-infrared-sensor-basics>]

3.1.8 Breadboard

Το breadboard ή στα ελληνικά, πλακέτα γενικών συνδέσεων ή πλακέτα διασύνδεσης χωρίς κολλήσεις, είναι μία βάση πάνω στην οποία προσαρμόζονται αρκετά εύκολα ηλεκτρονικά εξαρτήματα με σκοπό να δημιουργηθεί ένα κύκλωμα για πειραματική και προσωρινή –συνήθως- χρήση. Όπως γίνεται αντιληπτό και από την μία ελληνική ονομασία, η τοποθέτηση των εξαρτημάτων δεν χρειάζεται συγκόλληση, και έτσι μπορούν να αφαιρούνται, να προσθέτονται και να αντικαθιστούνται εξαρτήματα και καλωδιώσεις εύκολα και γρήγορα. Για τον λόγο αυτό, χρησιμοποιείται αρκετά συχνά στην τεχνολογική εκπαίδευση και στην έρευνα. Επάνω σε ένα breadboard μπορούν να μοντελοποιηθούν πολλά και διαφορετικά ηλεκτρονικά κυκλώματα, είτε αναλογικά είτε ψηφιακά, μικρά ή και μεγάλα στην περίπτωση που συνδυαστούν πολλά breadboards. Για παράδειγμα, μπορεί ακόμα και να υλοποιηθεί πάνω σε ένα breadboard μέχρι και μία κανονική Κεντρική Μονάδα Επεξεργασίας (CPU), χωρίς όμως πρακτική αξία, αλλά καθαρά εκπαιδευτική.

3.2 Μεθοδολογία

3.2.1 Λήψη Σήματος

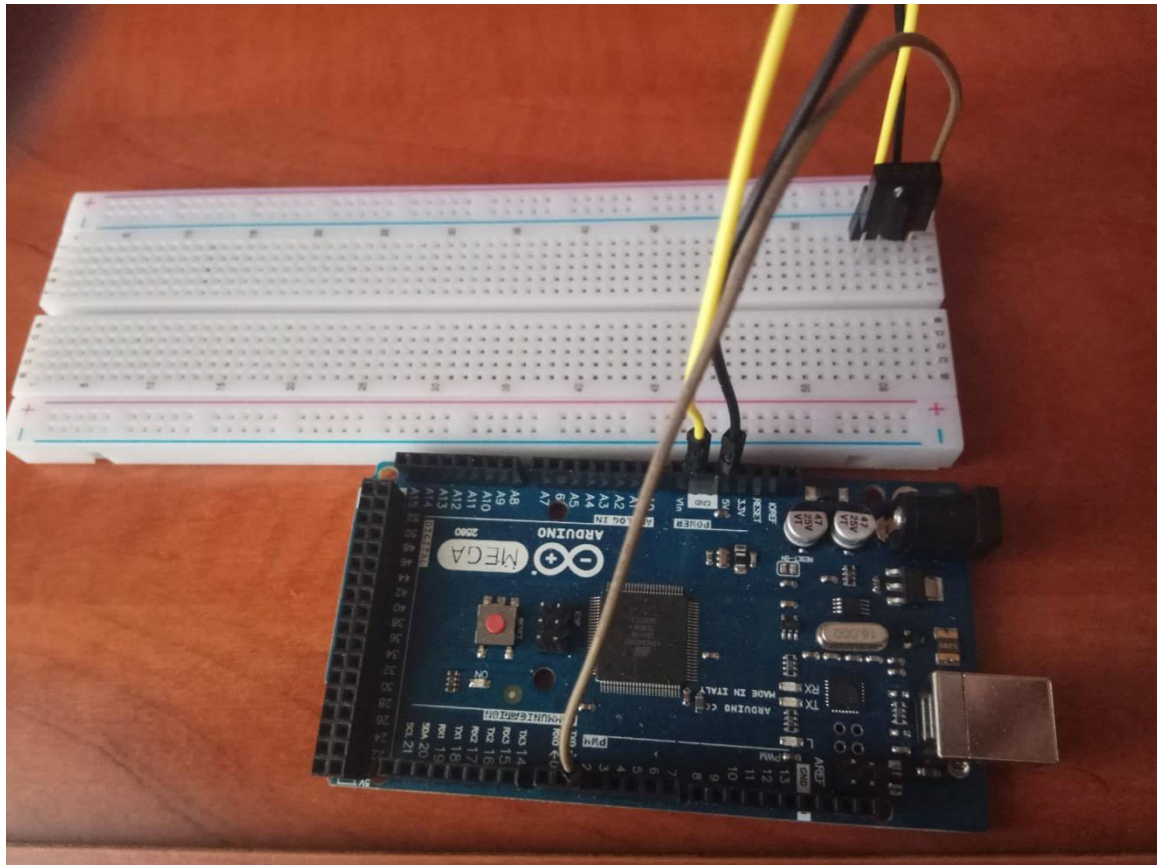
Το πρώτο πράγμα που πρέπει να ληφθεί υπόψη στην εργασία, αλλά και σε οποιαδήποτε παρόμοια κατασκευή, προκειμένου να υπάρχει έλεγχος κάποιας ηλεκτρονικής συσκευής, η οποία ελέγχεται από κάποιο τηλεχειριστήριο υπέρυθρων ακτίνων, είναι να ληφθούν από το τηλεχειριστήριο αυτό, οι εντολές βάσει των οποίων θα λειτουργεί η κατασκευή. Ωστόσο κάτι τέτοιο, μπορεί να μην είναι όσο απλό ακούγεται και να παρουσιάσει μερικά προβλήματα. Υπάρχουν αρκετές

βιβλιοθήκες για τα Arduino, οι οποίες περιέχουν κώδικες για την λήψη σήματος από τηλεχειριστήρια υπέρυθρων ακτίνων και οι καλύτερες από αυτές είναι οι βιβλιοθήκες IRLib και IRremote. Ενώ τα προγράμματα των βιβλιοθηκών αυτών φαίνεται να δουλεύουν χωρίς καμία ατέλεια για το μεγαλύτερο εύρος αυτών των συσκευών, όπως τηλεοράσεις και άλλες ηλεκτρονικές συσκευές πολυμέσων, το πρόβλημα εμφανίζεται όταν χρησιμοποιούνται για την λήψη σήματος συσκευών όπως τα Air Conditioners. Το πρόβλημα αυτό δημιουργείται λόγω του ότι, τα σήματα των Air Conditioners είναι αρκετά μεγαλύτερα, από τα σήματα που χρησιμοποιούν τα τηλεκοντρόλ των τηλεοράσεων και των υπόλοιπων συσκευών. Ένα ερώτημα που προκύπτει, είναι γιατί όμως να μην είναι κατασκευασμένες έτσι ώστε να μπορούν να λαμβάνουν και αυτά τα μεγαλύτερα σήματα. Ο λόγος είναι, ότι οι βιβλιοθήκες αυτές ορίζουν μια σταθερά, για να περιορίσουν τον μέγιστο αριθμό καταγεγραμμένων σημάτων (marks) και κενών (spaces) στα 100 (RAWBUF), ώστε να αφήσουν αρκετή SRAM διαθέσιμη για τον υπόλοιπο κώδικα του χρήστη. Φυσικά, μπορεί αυτό το όριο του RAWBUF να μετατραπεί από 100, στα 255 (περίπου στα 126 bits), που είναι το μέγιστο, μέσα από την βιβλιοθήκη. Παρ' ολ' αυτά, κάτι τέτοιο θα διορθώσει το πρόβλημα για το περίπου 50% των Air Conditioners, συνεπώς για τους υπόλοιπους χρήστες το πρόβλημα παραμένει. Την λύση σε αυτό το πρόβλημα, δίνει ο κώδικας **Long_IR_Reicever** (credit: AnalysIR), ο οποίος μπορεί να λάβει πολύ μεγαλύτερα σήματα, εύρους 800 marks & spaces (με δυνατότητα αλλαγής σε ακόμα μεγαλύτερο, αν το Arduino του χρήστη διαθέτει SRAM μεγαλύτερη από 2 KB).

Από στιγμή που έχει επιλεγεί ο κώδικας που πρόκειται να χρησιμοποιηθεί το επόμενο που πρέπει να γίνει, είναι η επιλογή των απαραίτητων υλικών που θα χρησιμοποιηθούν και η κατάλληλη σύνδεσή τους. Τα υλικά που επιλέχθηκαν για αυτήν την χρήση, είναι το Arduino Mega 2560 και το Tsop1738 IR Receiver, και η συνδεσμολογία τους είναι η εξής (η οποία αναγράφεται και για γενικότερη χρήση μέσα στον κώδικα):

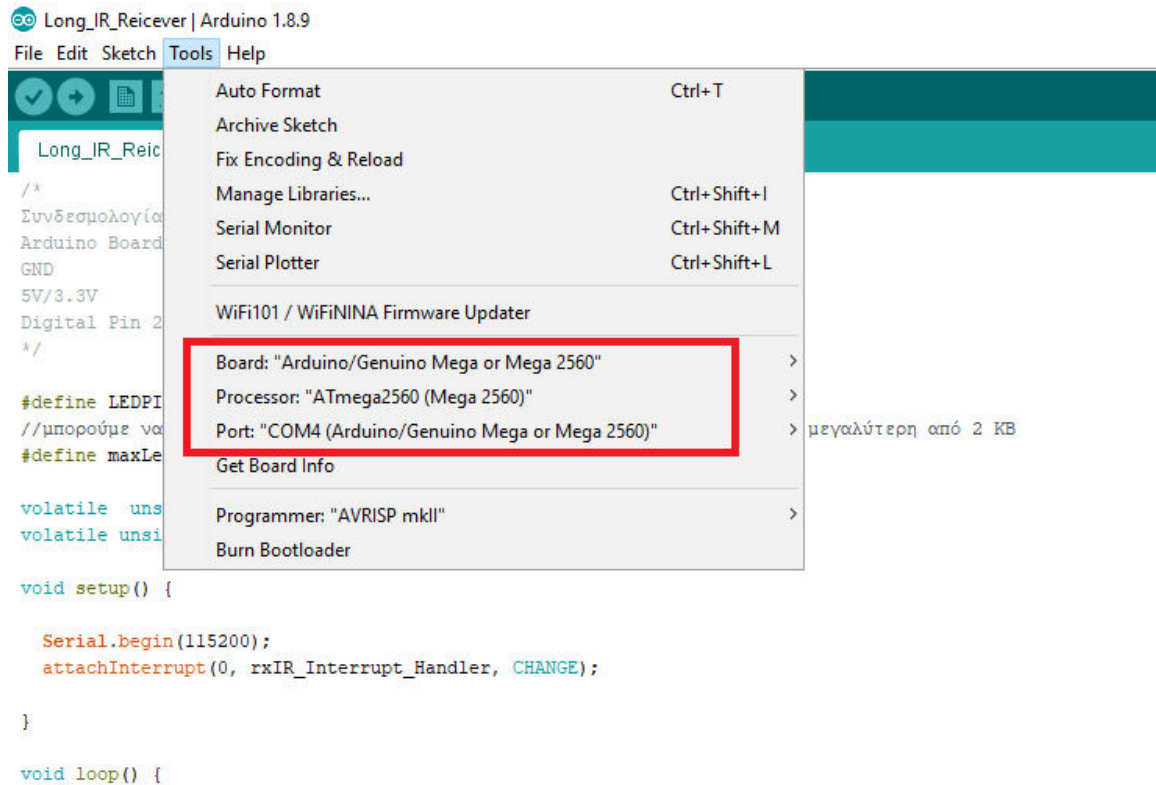
Arduino Mega 2560	Tsop1738
GND	GND
5V	VCC
Digital Pin 2	OUTPUT

4) Συνδεσμολογία για λήψη σήματος IR



17) Συνδεσμολογία Arduino Mega 2560 - Tso1738 για λήψη σήματος

Αφού έχουν επιτευχθεί λοιπόν οι κατάλληλες συνδέσεις των εξαρτημάτων, συνδέεται το Arduino στον υπολογιστή, ώστε να γίνει upload ο κώδικας στο Arduino. Πριν όμως γίνει το upload, πρέπει να πραγματοποιηθούν οι κατάλληλες επιλογές ρυθμίσεων.



18) Επιλογή σωστής πλακέτας και θύρας για Arduino Mega 2560

Στη συνέχεια, αφού γίνει verify και upload του κώδικα (sketch) και διαπιστωθεί ότι δεν υπάρχει κανένα πρόβλημα στον κώδικα και έχει γίνει όντως το upload, μπορεί να ανοιχθεί το Serial Monitor.

Ανοίγοντας το Serial Monitor, γίνεται αντιληπτό ότι το πρόγραμμα τρέχει κανονικά, και εμφανίζεται μήνυμα πατήματος κάποιου κουμπιού απ' το τηλεχειριστήριο. Προβάλλοντας το τηλεχειριστήριο προς το Tsop1738 (ή τον οποιονδήποτε IR Reicever) κρατώντας μια σχετικά μικρή απόσταση (μικρότερη των 2 μέτρων), δίνεται με το πάτημα του κουμπιού, η επιθυμητή εντολή.

```
COM4
10:37:46.769 ->
10:37:46.769 -> Press the button on the remote now - once only
10:37:51.765 -> Press the button on the remote now - once only
10:37:56.782 -> Press the button on the remote now - once only
10:38:01.764 -> Press the button on the remote now - once only
10:38:06.765 -> Press the button on the remote now - once only
10:38:11.782 -> Press the button on the remote now - once only
10:38:16.786 -> Press the button on the remote now - once only
10:38:21.760 -> Press the button on the remote now - once only
0:38:26.768 ->
0:38:26.768 -> Raw: (147) 8432, -4220, 580, -496, 560, -1544, 532, -524, 556, -1564, 560, -492, 564, -1540, 560, -1544,
0:38:26.838 ->
10:38:26.838 -> Press the button on the remote now - once only
10:38:31.846 -> Press the button on the remote now - once only
```

Autoscroll Show timestamp Newline 115200 baud Clear output

19) Serial Monitor του Long_IR_Receiver

Το σήμα αυτό εμφανίζεται στο Serial Monitor, και στην συνέχεια πρέπει να αποθηκευτεί σε κάποιον text editor, ώστε να κρατηθεί, με την ανάλογη ονομασία, για την καλύτερη απομνημόνευσή της.

```
signals temp - Notepad
File Edit Format View Help
ress the button on the remote now - once only
10:37:46.700 ->
18 - Raw: (147) 8456, -4220, 580, -496, 560, -1544, 560, -496, 556, -1564, 536, -520, 560, -1544,
10:37:46.769 ->

19
-> Press the button on the remote now - once only
10:38:26.768 ->
10:38:26.768 -> Raw: (147) 8432, -4220, 580, -496, 560, -1544, 532, -524, 556, -1564, 560, -492,
10:38:26.838 ->

21
Press the button on the remote now - once only
10:38:51.854 ->
10:38:51.854 -> Raw: (147) 8480, -4216, 556, -500, 560, -1540, 560, -496, 560, -1564, 508, -548,
10:38:51.923 ->

22
Press the button on the remote now - once only
10:39:06.904 ->
10:39:06.904 -> Raw: (147) 8480, -4196, 580, -496, 560, -1544, 556, -500, 556, -1564, 560, -496,
10:39:06.974 ->
```

20) Αποθήκευση των σημάτων σε text editor

3.2.2 Μετατροπή και Έλεγχος Σήματος

Ωστόσο παρ' ότι έχει αποθηκευθεί το επιθυμητό σήμα, είναι απαραίτητο να ακολουθηθούν μερικά βήματα προτού χρησιμοποιηθεί στον τελικό κώδικα. Παίρνοντας για παράδειγμα το σήμα θερμοκρασίας 19 που αναγράφεται στις εικόνες 18 και 19.

19 Raw: (147) 8432, -4220, 580, -496, 560, -1544, 532, -524, 556, -1564, 560, -492, 564, -1540, 560, -1544, 512, -564, 556, -1544, 560, -1540, 564, -1540, 560, -1564, 508, -548, 556, -496, 560, -1544, 560, -1560, 560, -496, 560, -496, 560, -496, 560, -1560, 532, -524, 560, -1540, 536, -524, 560, -1556, 540, -516, 564, -492, 564, -472, 580, -516, 560, -496, 560, -496, 536, -516, 564, -512, 560, -496, 560, -496, 560, -496, 560, -512, 560, -1544, 532, -524, 560, -492, 564, -512, 560, -496, 560, -496, 560, -496, 560, -512, 560, -496, 564, -492, 560, -496, 560, -512, 560, -496, 560, -496, 560, -512, 564, -492, 560, -496, 560, -496, 560, -516, 536, -516, 564, -476, 580, -492, 540, -536, 560, -1540, 560, -1548, 508, -1592, 560, -1560, 560, -1544, 536, -1564, 560, -496, 560, -496, 560

Το σήμα αυτό, προκειμένου να μπορέσει να χρησιμοποιηθεί, οφείλει να έρθει στην σωστή ακόλουθη μορφή:

{8432, 4220, 580, 496, 560, 1544, 532, 524, 556, 1564, 560, 492, 564, 1540, 560, 1544, 512, 564, 556, 1544, 560, 1540, 564, 1540, 560, 1564, 508, 548, 556, 496, 560, 1544, 560, 1560, 560, 496, 560, 496, 560, 496, 560, 1560, 532, 524, 560, 1540, 536, 524, 560, 1556, 540, 516, 564, 492, 564, 472, 580, 516, 560, 496, 560, 496, 536, 516, 564, 512, 560, 496, 560, 496, 560, 496, 560, 512, 560, 1544, 560, 496, 560, 496, 560, 512, 560, 1544, 532, 524, 560, 492, 564, 512, 560, 496, 560, 496, 560, 496, 560, 512, 560, 496, 564, 492, 560, 496, 560, 512, 560, 496, 560, 496, 560, 496, 560, 512, 564, 492, 560, 496, 560, 496, 560, 516, 536, 516, 564, 476, 580, 492, 540, 536, 560, 1540, 560, 1548, 508, 1592, 560, 1560, 560, 1544, 536, 1564, 560, 496, 560, 496, 560}

Έχοντας διαμορφώσει λοιπόν το σήμα στην σωστή μορφή, μένει ακόμα ένα βήμα προτού μπει στον τελικό κώδικα. Το βήμα αυτό είναι αρκετά σημαντικό και είναι ο έλεγχος του σήματος. Ο λόγος που χρειάζεται να μην παραλειφθεί αυτό το βήμα είναι λόγω του ότι μερικά τηλεχειριστήρια των Air Conditioners δεν στέλνουν για την ίδια εντολή το ίδιο σήμα συνέχεια, αλλά διαφορετικό. Για τον λόγο αυτό καλό είναι να καταγράφεται κάθε σήμα 2-3 φορές και μετά, αφού έχει έρθει στην κατάλληλη μορφή, να ελέγχεται.

Ο έλεγχος αυτός είναι αναγκαίος, όπως φαίνεται από τα παρακάτω παραδείγματα:



21) Air Condition Controller

Όπως φαίνεται στην εικόνα 21, το τηλεχειριστήριο, το οποίο χρησιμοποιήθηκε για την λήψη σημάτων, των εντολών που θα χρησιμοποιεί αργότερα το Smart Box, διαθέτει ένα κουμπί για την εντολή Power (ON/OFF) και από ένα κουμπί για το ανέβασμα/κατέβασμα της θερμοκρασίας αντίστοιχα. Βλέποντας αυτό, μπορεί να υπάρξει παρανόηση ότι πατώντας το κουμπί της εντολής Power, το Air Condition αν είναι ανοιχτό θα κλείσει και αν είναι κλειστό θα ανοίξει. Ωστόσο δεν είναι αυτή η λειτουργία του. Πατώντας το κουμπί Power, το τηλεχειριστήριο αυτό στέλνει σήμα είτε για να ανοίξει, είτε για να κλείσει. Συνεπώς πρέπει να βρεθεί, απ' το σήμα που λήφθηκε και μετατράπηκε, ποιο σήμα είναι το ON και ποιο σήμα είναι το OFF. Παρόμοιο πρόβλημα δημιουργείται και με τα κουμπιά temp up (temperature up/αύξηση θερμοκρασίας) και temp down (temperature down/μείωση θερμοκρασίας). Και εδώ μπορεί να υπάρξει παρανόηση όσον αφορά τον τρόπο λειτουργίας, θεωρώντας για παράδειγμα ότι το Air Condition Controller στέλνει ένα σήμα για το κουμπί temp up και ένα σήμα για το κουμπί temp down, αλλά η πραγματικότητα είναι ότι δεν συμβαίνει κάτι τέτοιο. Ένα παράδειγμα που αναδεικνύει τον σωστό τρόπο λειτουργίας είναι το εξής:

Έστω ότι η θερμοκρασία (όχι του δωματίου, όπως φαίνεται στην εικόνα 23) του Air Condition βρίσκεται στο 24. Μπορούν να υπάρξουν τα παρακάτω ενδεχόμενα:

-Πατώντας 1 φορά το temp up, η θερμοκρασία πηγαίνει στο 25, και η τελική θερμοκρασία (25) στέλνεται σαν σήμα.

-Πατώντας 5 φορές το temp up, η θερμοκρασία πηγαίνει στο 29, και η τελική θερμοκρασία (29) στέλνεται σαν σήμα.

-Πατώντας 6 φορές το temp up και 8 φορές το temp down, η θερμοκρασία πηγαίνει στο 22, και η τελική θερμοκρασία (22) στένεται σαν σήμα.

-Πατώντας 3 φορές το temp up και 3 φορές το temp down, η θερμοκρασία παραμένει στο 24, και η τελική θερμοκρασία (24) στέλνεται σαν σήμα.

Συνεπώς, διαπιστώνεται, ότι αν απλά πατηθεί μία φορά το temp up, ώστε να δοθεί σαν εντολή η 'αύξηση θερμοκρασίας' και μια φορά το temp down, ώστε να δοθεί σαν εντολή η 'μείωση θερμοκρασίας', το αποτέλεσμα θα είναι πολύ διαφορετικό, καθώς στην ουσία θα έχουν ληφθεί 2 σταθερές θερμοκρασίες (για παράδειγμα το 24 και το 23).

Για να υπάρχει λοιπόν η σιγουριά, ότι οι εντολές έχουν ληφθεί και μετατραπεί σε σωστό σήμα, κάνοντας αυτό ακριβώς που είναι επιθυμητό, προτού εισαχθούν στον κύριο κώδικα πρέπει να ελέγχονται.

Οι έλεγχοι αυτού πραγματοποιήθηκαν με τα ακόλουθα υλικά:

Arduino Uno

IR LED sensor

2N222A NPN Transistor

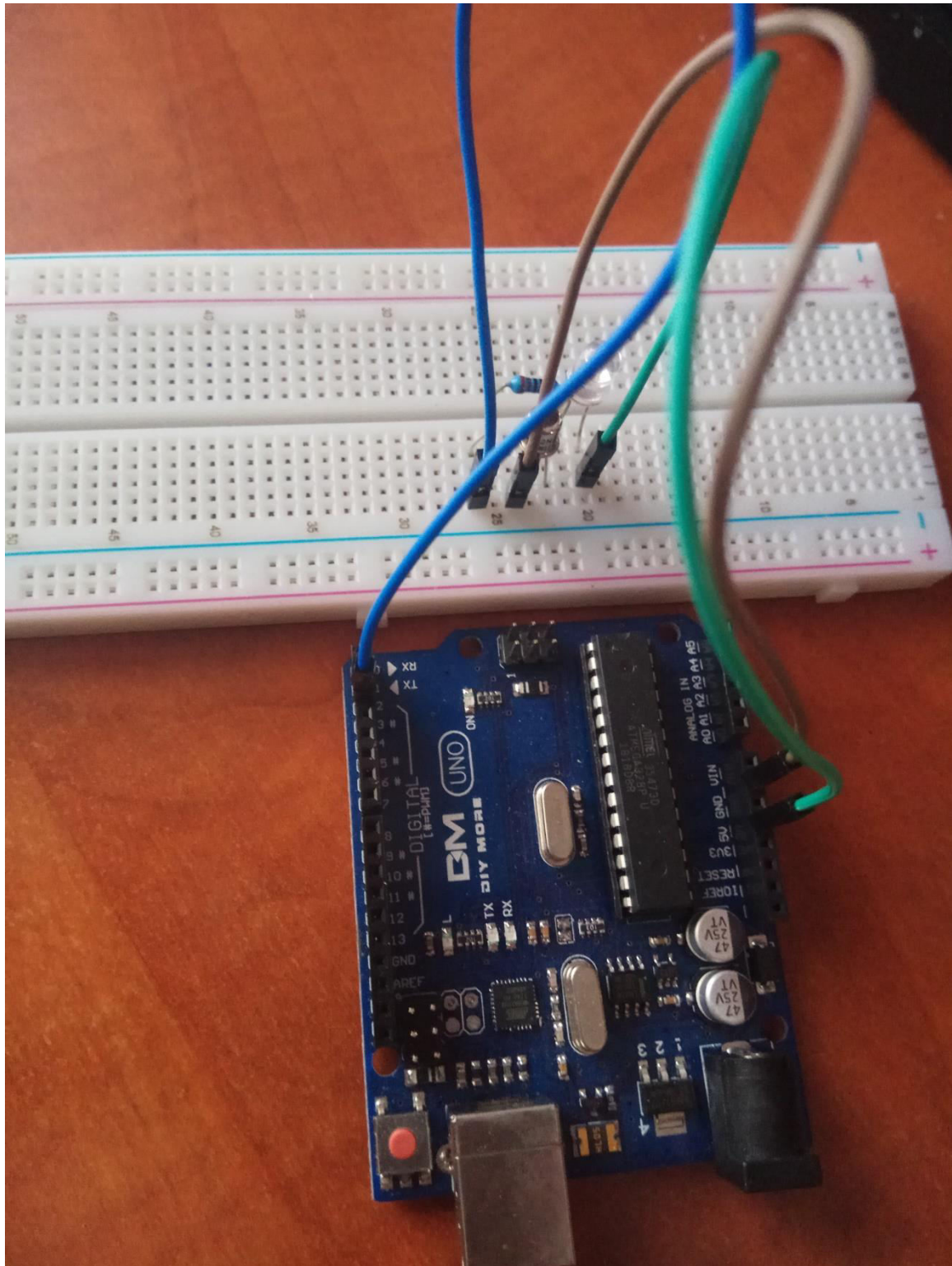
Resistor 120R

Breadboard

Σύμφωνα με τις ακόλουθες συνδέσεις(οι οποίες φαίνονται και στην εικόνα 22):

Θέσεις στο breadboard	Arduino Uno	IR LED sensor	2N222A NPN TRANSISTOR	RESISTOR 120R
20	5V	Anode (+)	-	-
21	-	-	-	-
22	-	Cathode (-)	Collector (3)	-
23	-	-	Base (2)	✓
24	GND	-	Emitter (1)	-
25	-	-	-	-
26	Digital Pin 2	-	-	✓

5) Συνδεσμολογία για έλεγχο σημάτων



22) Συνδεσμολογία Uno/IR LED/Resistor/Transistor ελέγχου σημάτων

Έχοντας γίνει λοιπόν οι κατάλληλες συνδέσεις και έχοντας επιλεγθεί το σωστό board και port στον κώδικα (όπως φαίνεται στην εικόνα 23) μπορεί να γίνει upload ο κώδικας **Test_IR_sender** στο Arduino Uno ώστε να επιτευχθούν οι κατάλληλοι

έλεγχου. Στον κώδικα Test_IR_sender ελέγχεται μόνο ένα σήμα κάθε φορά, αλλά μπορεί εύκολα να προστεθούν παραπάνω σήματα για έλεγχο.



23) Επιλογή σωστής πλακέτας και θύρας για Arduino Uno

3.2.3 Ολοκλήρωση

Αφού έχουν ολοκληρωθεί όλα τα προηγούμενα βήματα και τα σήματα είναι σωστά και εκτελούν την επιθυμητή λειτουργία, είναι εφικτό να συμπεριληφθούν στον τελικό κώδικα.

Πριν πραγματοποιηθεί η τελική συνδεσμολογία όλων των απαραίτητων υλικών της κατασκευής, μπορεί προαιρετικά να εκτελεστεί ένα σύντομο πρόγραμμα που έχει δημιουργηθεί για τον έλεγχο λειτουργίας του αισθητήρα BME280.

Το πρόγραμμα είναι το **simple_bme_test** και αυτό που κάνει είναι να τυπώνει την θερμοκρασία και την υγρασία στο Serial Monitor ανά 2 δευτερόλεπτα.

Ο συγκεκριμένος κώδικας μπορεί να γίνει upload και να τρέξει σε οποιαδήποτε από τις 3 πλακέτες της επιλογής μας, αρκεί να έχουν γίνει οι σωστές συνδέσεις (μεταξύ της πλακέτας και του αισθητήρα BME280).

Η συνδεσμολογία για την κάθε περίπτωση θα έχει ως εξής:

BME280	ESP8266 NODEMCU	ARDUINO MEGA 2560	ARDUINO UNO
VIN	3.3V	5V	5V
GND	GND	GND	GND
SCL	D1	21	A5
SDA	D2	20	A4

6) Συνδεσμολογία πλακετών για έλεγχο του αισθητήρα BME280

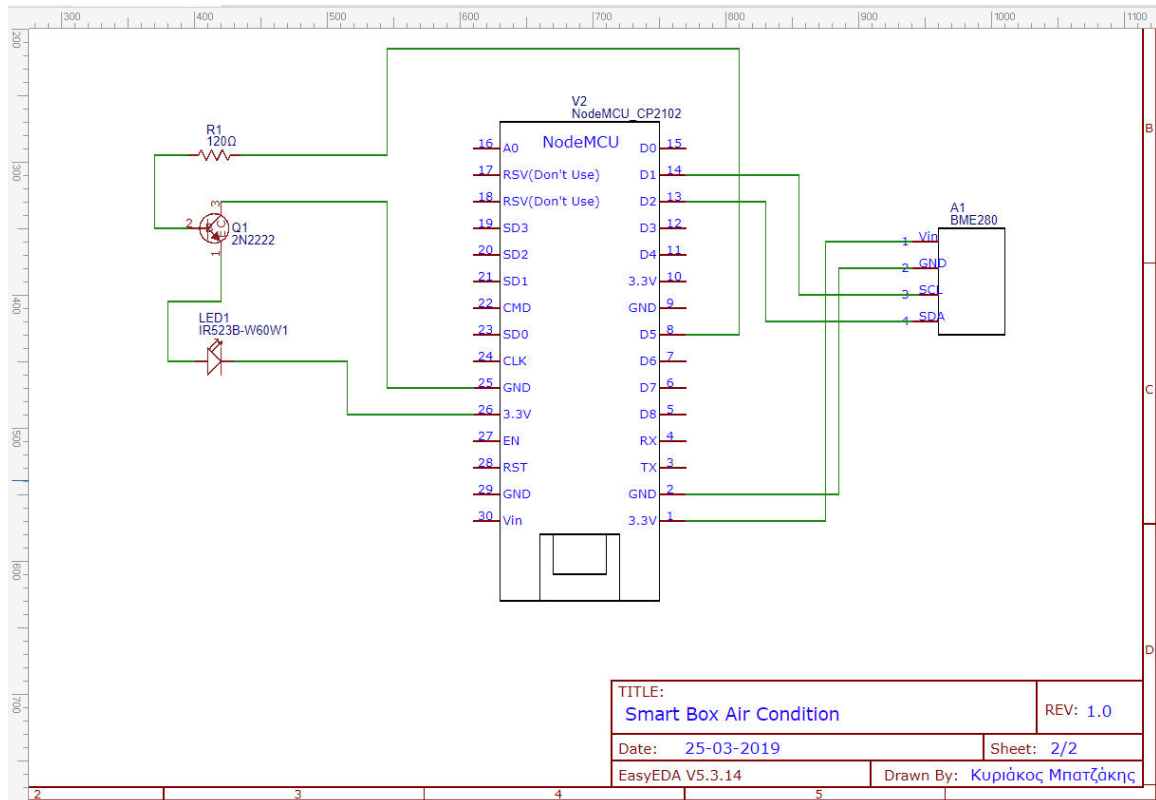
Όπως φαίνεται στις εικόνες 9, 10 και 11, τα pin D1 21 A5 και D2 20 A4 των Esp8266/Mega και Uno, είναι τα τα SCL και SDA τους αντίστοιχα. Παρατηρείται ωστόσο ότι το Esp8266 έχει μόνο έναν I2C δίαυλο σε αντίθεση με τα Arduino Mega και Uno που έχουν από 2. Φυσικά στις πλακέτες Mega και Uno, η σύνδεση μπορεί να γίνει και με εκείνον τον I2C δίαυλο, αντί για αυτά που αναδείχθηκαν στον πίνακα 6.

Ο I2C είναι ένας σειριακός δίαυλος που χρησιμοποιείται για την σύνδεση περιφερειακών μικρής ταχύτητας σε μητρικές πλακέτες (motherboards), ενσωματωμένων συστημάτων (embedded systems), και άλλων ηλεκτρονικών συσκευών. Εκτός από την επικοινωνία συσκευών που βρίσκονται πάνω σε ένα τυπωμένο κύκλωμα, χρησιμοποιείται και για την επικοινωνία συσκευών που συνδέονται με καλώδια. Ο δίαυλος αυτός χρησιμοποιεί 2 καλώδια για την μεταφορά δεδομένων, τα SCL και SDA. Η SCL είναι η γραμμή ρολογιού, ενώ η SDA είναι η γραμμή δεδομένων. Εκτός απ' αυτά τα 2 καλώδια απαιτείται και ένα τρίτο καλώδιο γείωσης (GND) και, συνήθως υπάρχει και ένα τέταρτο καλώδιο τροφοδοσίας (VIN/VCC/VDD), όπως έχει και ο αισθητήρας BM280 (αλλά και η οθόνη OLED που υπάρχει παρακάτω) με την οποία τροφοδοτούνται με ισχύ οι

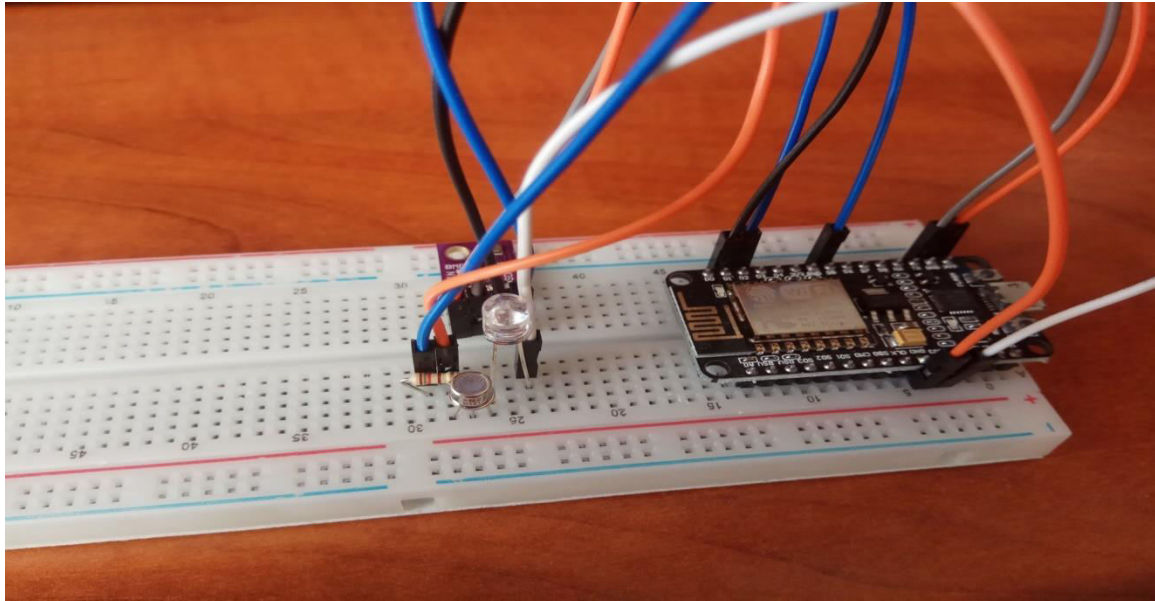
διάφορες συσκευές που συνδέονται στον δίαυλο. Συνήθως οι τάσεις που χρησιμοποιούνται είναι 3.3V και 5V, όπως βλέπουμε στον πίνακα 6.

Γνωρίζοντας πλέον ότι και ο αισθητήρας δουλεύει σωστά, μπορεί να γίνει η σύνδεση, ώστε να παραχθεί η τελική κατασκευή. Η συνδεσμολογία της τελικής κατασκευής θα αποτελείται από τα ακόλουθα υλικά:

Breadboard, Esp8266, Bme280, IR LED, 2N2222A και τον αντιστάτη 120Ω και θα έχει ως εξής:

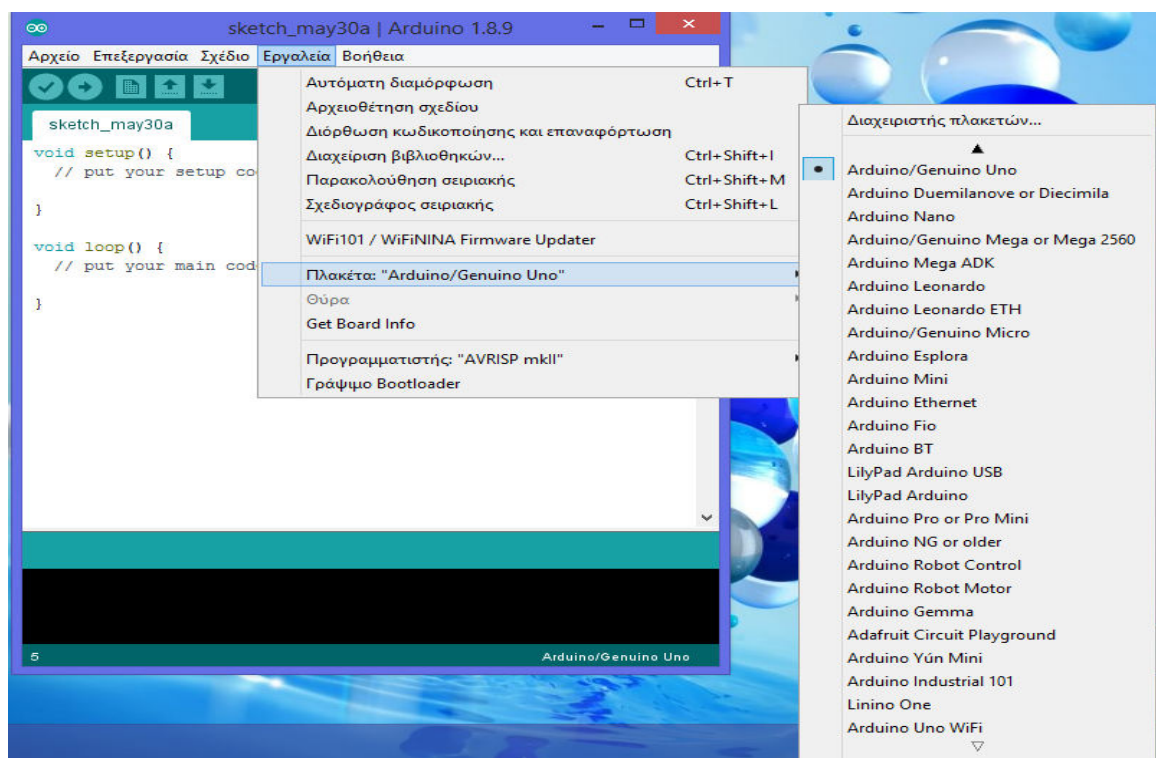


24) Σχηματική παράσταση τελικής συνδεσμολογίας



25) Συνδεσμολογία τελικής κατασκευής

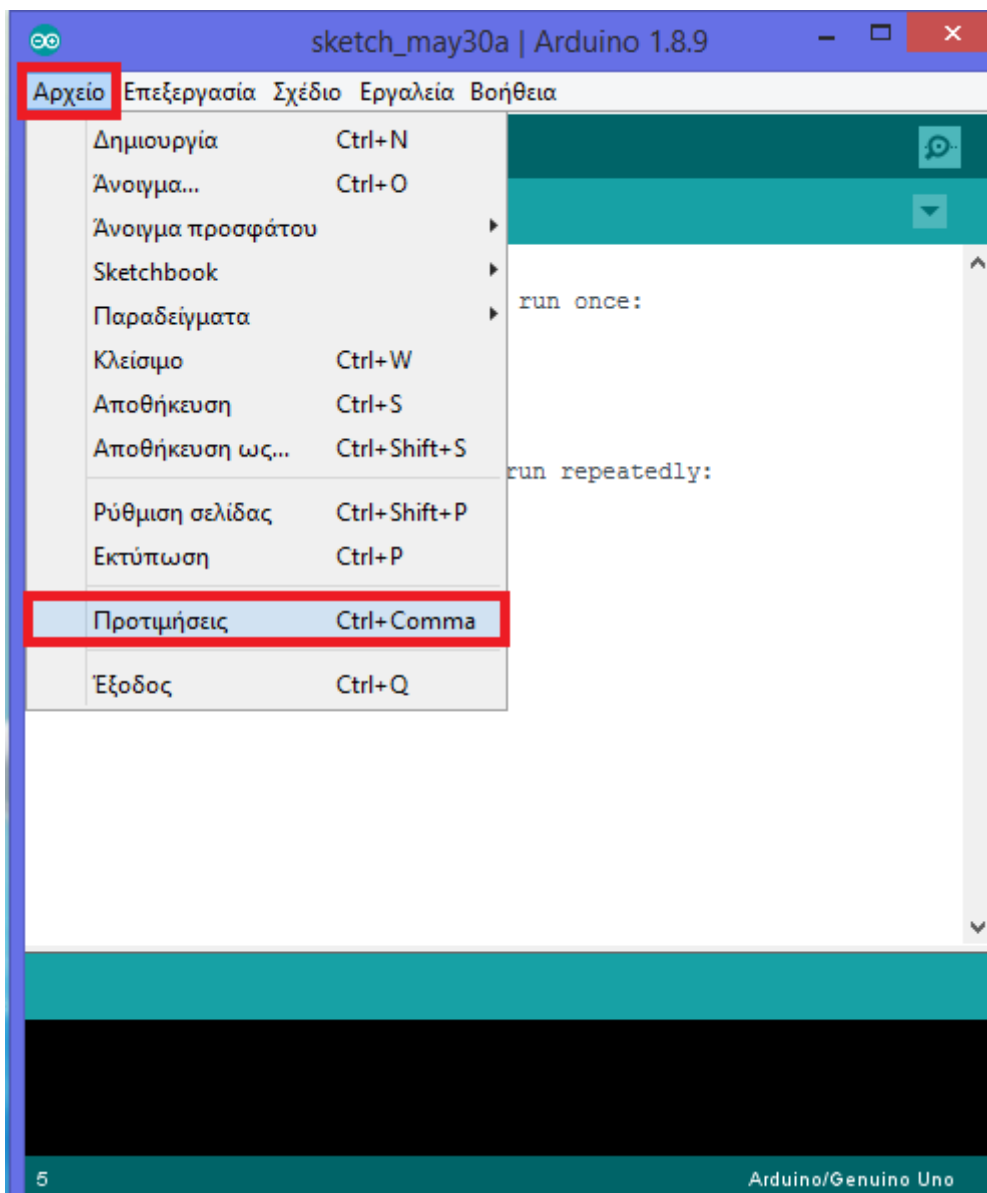
Στην φάση αυτή είναι εφικτό να συνδέσουμε το Esp8266 στον υπολογιστή για να γίνει upload ο κώδικας μέσω Arduino IDE. Όμως όταν αναζητηθεί στην κατηγορία Πλακέτα (Board) για να βρεθεί η σωστή πλακέτα, ώστε να επιλεγθεί και να γίνει upload ο κώδικας, γίνεται αντιληπτό, ότι είναι αδύνατο να βρεθεί.



26) Απόλεια εύρεσης πλακετών Esp8266

Αυτό συμβαίνει, διότι η πλακέτα Esp8266 Nodemcu δεν ανήκει στην οικογένεια Arduino. Για να το διορθωθεί αυτό πρέπει να ακολουθηθούν μερικά απλά βήματα.

1) Αρχικά επιλέγεται -> **Αρχείο** -> **Προτιμήσεις**

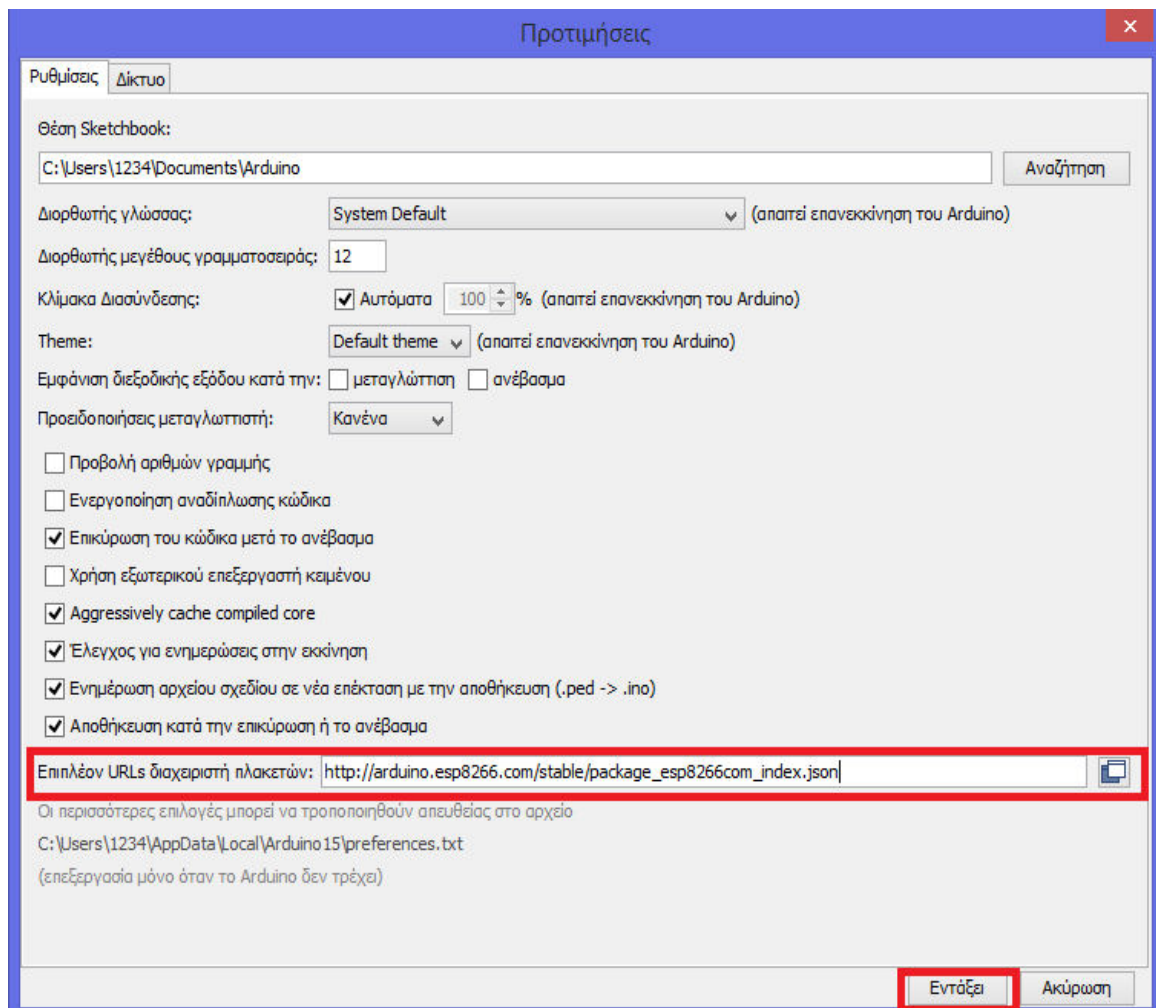


27) Βήμα 1

2) Εισάγεται το ακόλουθο λινκ

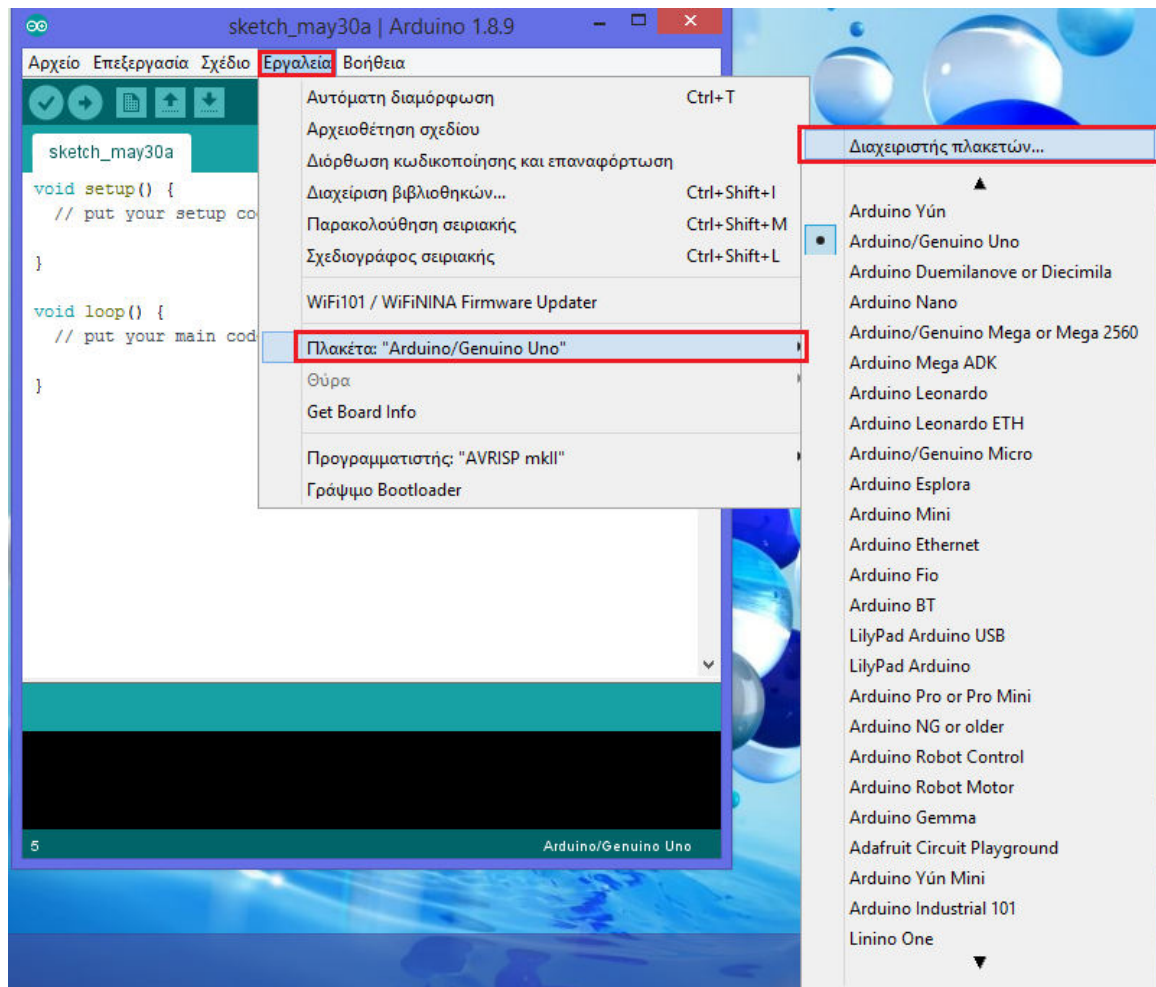
http://arduino.esp8266.com/stable/package_esp8266com_index.json στο πεδίο

‘Επιπλέον URLs διαχειριστή πλακετών’ και επιλέγεται **Εντάξει**.



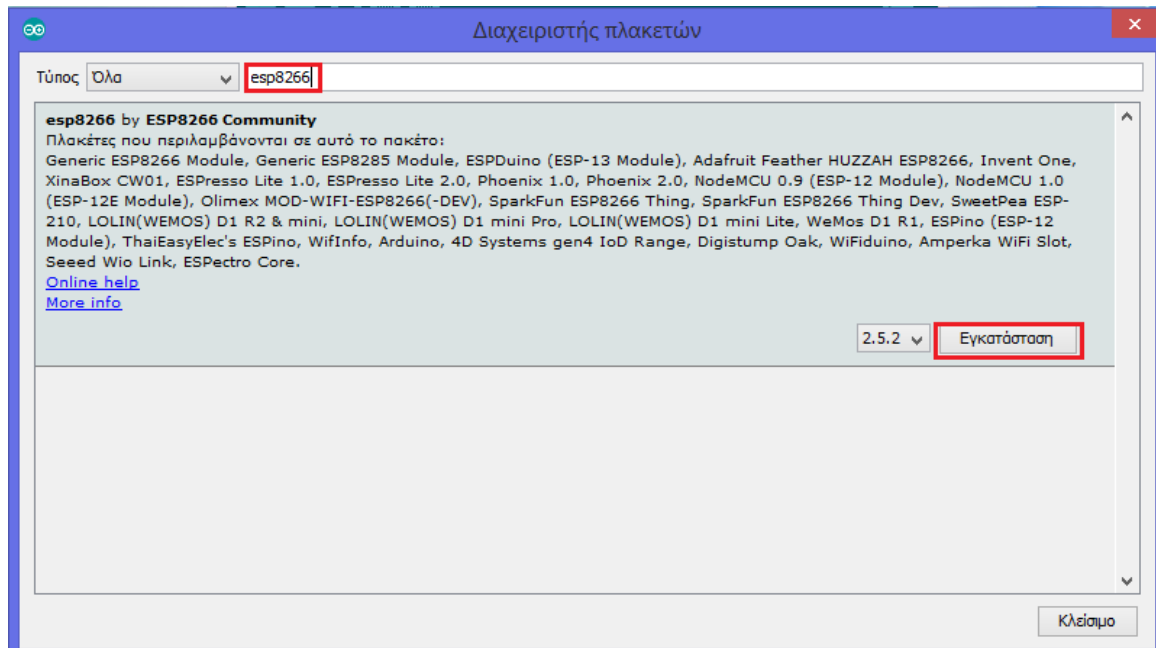
28) Βήμα 2

3) Στη συνέχεια επιλέγεται -> **Εργαλεία** -> **Πλακέτα** -> **Διαχειριστής πλακετών...**



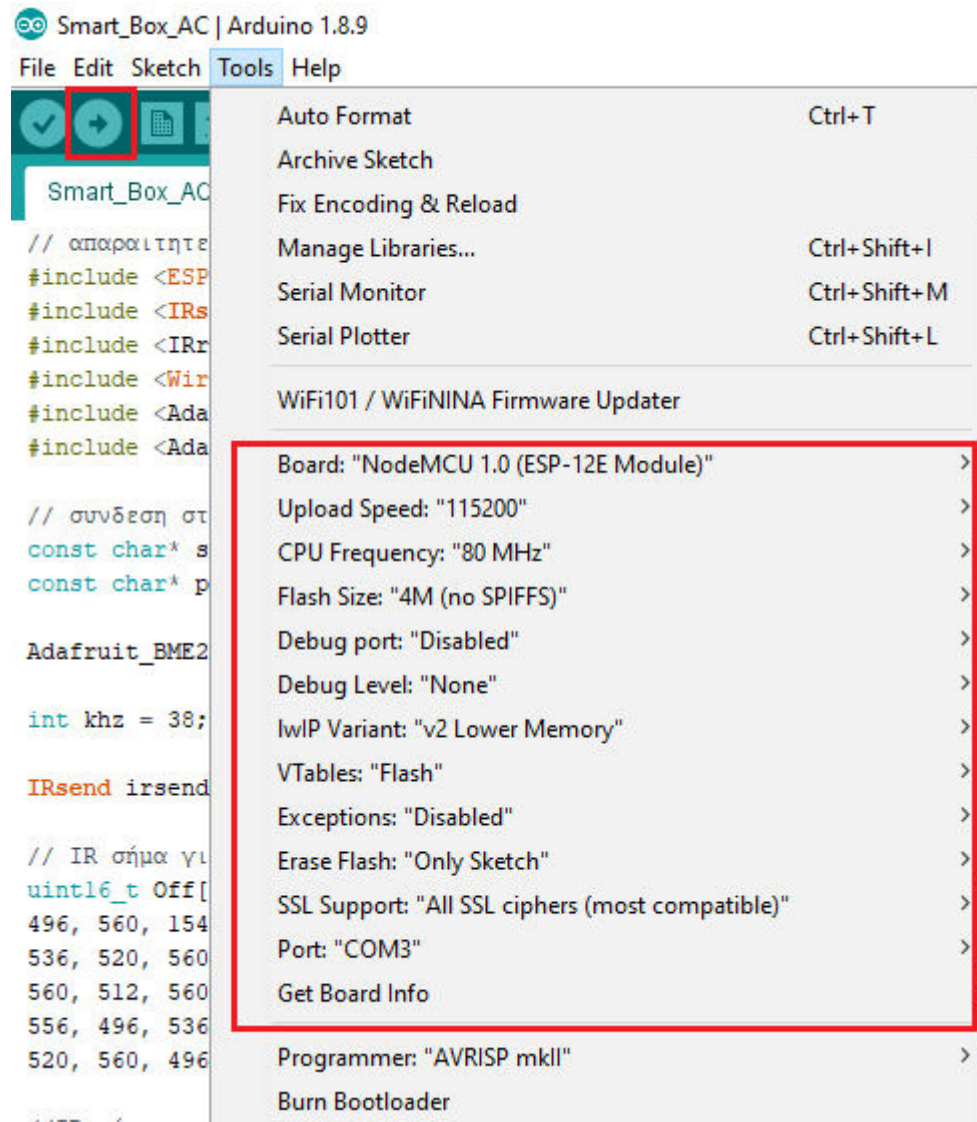
29) Βήμα 3

4) Στον διαχειριστή πλακετών, αναζητείται το **esp8266** από το **ESP8266 Community** και γίνεται **εγκατάσταση**.



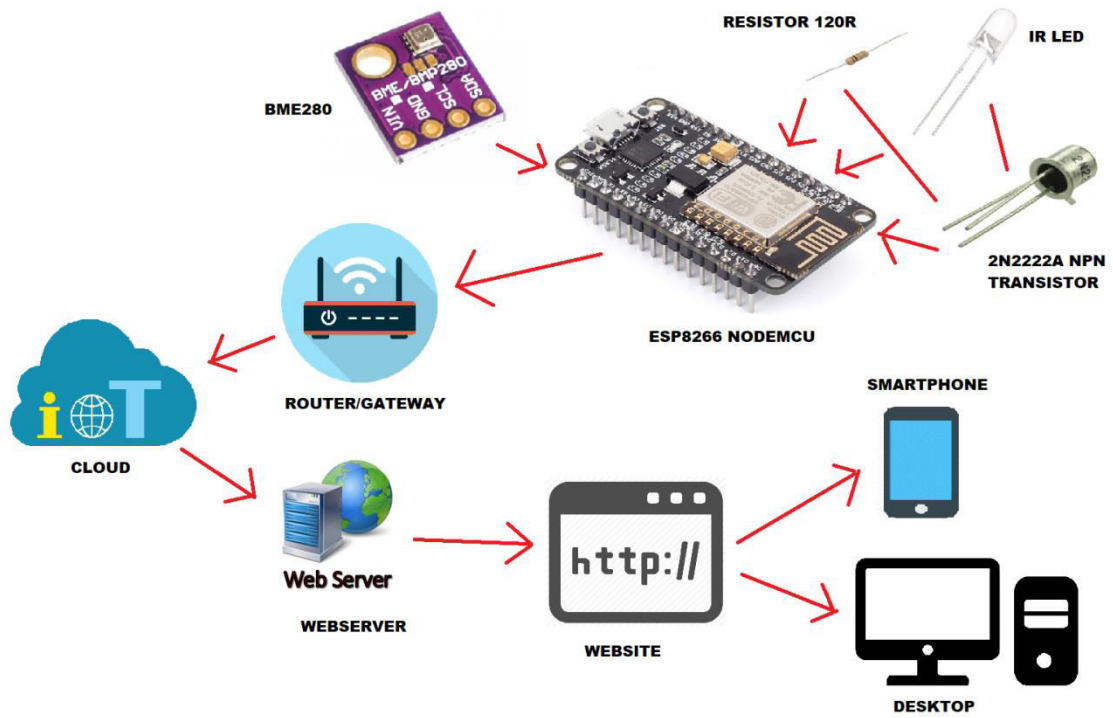
30) Βήμα 4

Είναι εφικτή λοιπόν η εύρεση των πλακετών Esp8266. Οπότε σε αυτό το σημείο, αφού επιλεγθεί η σωστή πλακέτα και θύρα, είναι δυνατό να ανεβεί ο κύριος κώδικας **Smart_Box_AC** στο Esp8266.



31) Ανέβασμα τελικού κώδικα στο Esp8266

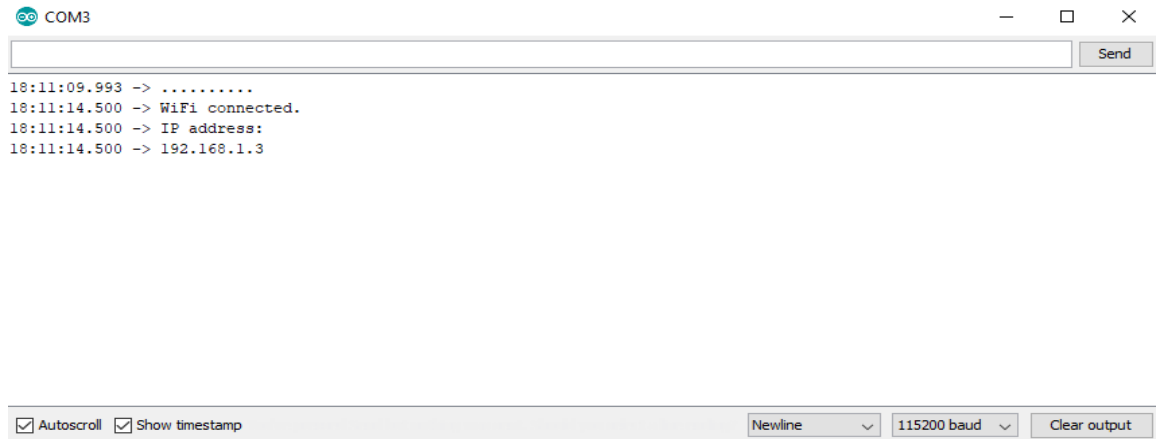
Πριν το επόμενο κεφάλαιο των αποτελεσμάτων, γίνεται μια γρήγορη επισκόπηση, της λειτουργίας του συστήματος του Έξυπνου Κουτιού.



32) Επισκόπηση συστήματος Έξυπνου Κουτιού

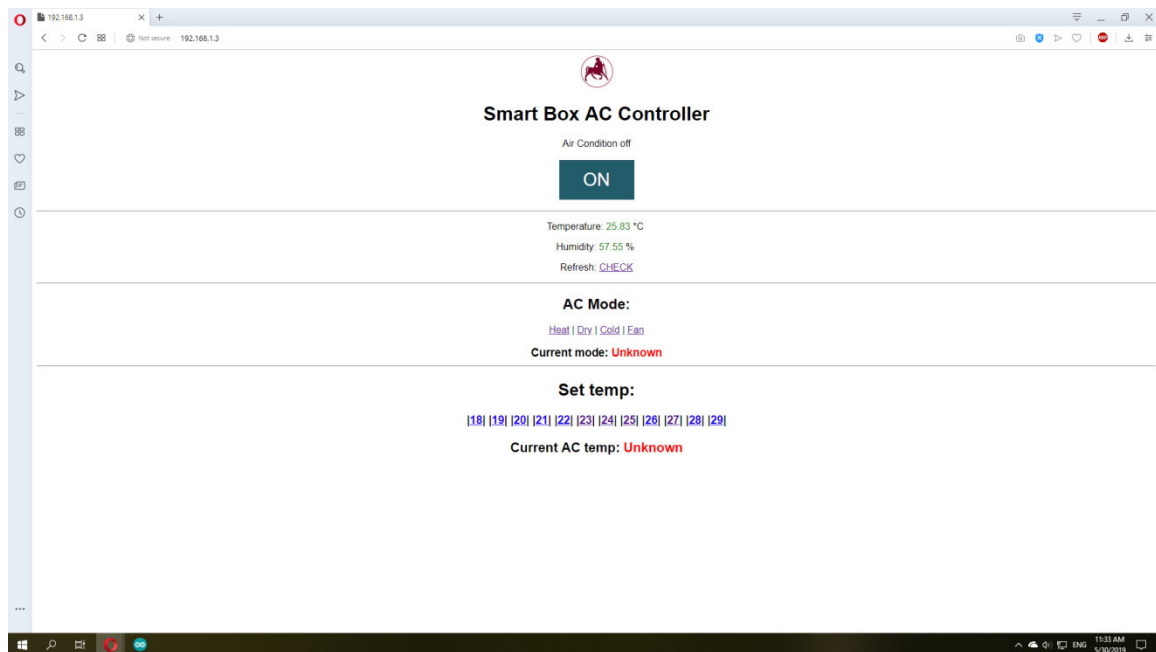
4. Αποτελέσματα

Αφού έχει ολοκληρωθεί το ανέβασμα του κώδικα στην πλακέτα μπορεί να ανοιχθεί το Serial Monitor.



33) Serial Monitor έπειτα από ανέβασμα του κώδικα

Επιλέγεται η IP address που εμφανίστηκε στην οθόνη και χρησιμοποιείται στην καρτέλα του browser μας.



34) Fullscreen της ιστοσελίδας στο Desktop

The screenshot shows a Serial Monitor window titled 'COM3'. The main area displays the following log entries:

```
11:33:43.753 -> New Client.
11:33:43.753 -> GET / HTTP/1.1
11:33:43.753 -> Host: 192.168.1.3
11:33:43.753 -> Connection: keep-alive
11:33:43.787 -> Upgrade-Insecure-Requests: 1
11:33:43.787 -> User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0
11:33:43.787 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
11:33:43.787 -> Accept-Encoding: gzip, deflate
11:33:43.787 -> Accept-Language: en-US,en;q=0.9
11:33:43.787 ->
11:33:43.925 -> Client disconnected.
11:33:43.925 ->
11:33:43.925 -> New Client.
```

At the bottom of the window, there are control elements: a 'Send' button, a scroll bar, and a status bar containing 'Autoscroll' (checked), 'Show timestamp' (checked), a 'Newline' dropdown menu, '115200 baud', and a 'Clear output' button.

35) Serial Monitor έπειτα από την είσοδο στην ιστοσελίδα

Στο σημείο γίνεται αντιληπτό ότι έχει πραγματοποιηθεί η σύνδεση.

Ακολουθεί ένα σύντομο παράδειγμα όσον αφορά την λειτουργικότητα.

Έστω ότι αρχικά επιλέγεται το κουμπί 'ON'.



Smart Box AC Controller

Air Condition on

OFF

Temperature: 26.03 *C

Humidity: 59.78 %

Refresh: [CHECK](#)

AC Mode:

[Heat](#) | [Dry](#) | [Cold](#) | [Fan](#)

Current mode: **Unknown**

Set temp:

[18](#) | [19](#) | [20](#) | [21](#) | [22](#) | [23](#) | [24](#) | [25](#) | [26](#) | [27](#) | [28](#) | [29](#)

Current AC temp: **Unknown**

36) Παράδειγμα λειτουργικότητας εικόνα 1

```
11:34:05.668 ->
11:34:05.668 -> New Client.
11:34:05.668 -> GET /14/on HTTP/1.1
11:34:05.668 -> Host: 192.168.1.3
11:34:05.668 -> Connection: keep-alive
11:34:05.668 -> Upgrade-Insecure-Requests: 1
11:34:05.668 -> User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 OPR/60.0.3255.109
11:34:05.702 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
11:34:05.702 -> Referer: http://192.168.1.3/
11:34:05.702 -> Accept-Encoding: gzip, deflate
11:34:05.702 -> Accept-Language: en-US,en;q=0.9
11:34:05.702 ->
11:34:05.805 -> IR power On signal sent
11:34:05.907 -> Client disconnected.
11:34:05.907 ->
```

37) Παράδειγμα λειτουργικότητας εικόνα 1α

Μετά η επιλογή 'HEAT'.



Smart Box AC Controller

Air Condition on

OFF

Temperature: 26.04 *C

Humidity: 59.94 %

Refresh: [CHECK](#)

AC Mode:

[Heat](#) | [Dry](#) | [Cold](#) | [Fan](#)

Current mode: **HEAT**

Set temp:

[|18|](#) [|19|](#) [|20|](#) [|21|](#) [|22|](#) [|23|](#) [|24|](#) [|25|](#) [|26|](#) [|27|](#) [|28|](#) [|29|](#)

Current AC temp: **Unknown**

38) Παράδειγμα λειτουργικότητας εικόνα 2

```
11:34:05.907 -> New Client.  
11:34:12.131 -> GET /14/heaton HTTP/1.1  
11:34:12.131 -> Host: 192.168.1.3  
11:34:12.131 -> Connection: keep-alive  
11:34:12.131 -> Upgrade-Insecure-Requests: 1  
11:34:12.131 -> User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 OPR/60.0.3255.109  
11:34:12.131 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8  
11:34:12.131 -> Referer: http://192.168.1.3/14/on  
11:34:12.131 -> Accept-Encoding: gzip, deflate  
11:34:12.165 -> Accept-Language: en-US,en;q=0.9  
11:34:12.165 ->  
11:34:12.266 -> IR heat mode signal sent  
11:34:12.369 -> Client disconnected.
```

39) Παράδειγμα λειτουργικότητας εικόνα 2α

Έπειτα επιλέγεται η θερμοκρασία των 26 βαθμών.



Smart Box AC Controller

Air Condition on

OFF

Temperature: 26.06 *C

Humidity: 58.75 %

Refresh: [CHECK](#)

AC Mode:

[Heat](#) | [Dry](#) | [Cold](#) | [Fan](#)

Current mode: **HEAT**

Set temp:

[|18|](#) [|19|](#) [|20|](#) [|21|](#) [|22|](#) [|23|](#) [|24|](#) [|25|](#) [|26|](#) [|27|](#) [|28|](#) [|29|](#)

Current AC temp: **26**

40) Παράδειγμα λειτουργικότητας εικόνα 3

```
11:34:19.694 -> New Client.  
11:34:23.471 -> GET /14/temp26 HTTP/1.1  
11:34:23.471 -> Host: 192.168.1.3  
11:34:23.471 -> Connection: Keep-alive  
11:34:23.505 -> Upgrade-Insecure-Requests: 1  
11:34:23.505 -> User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 OPR/60.0.3255.109  
11:34:23.505 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8  
11:34:23.505 -> Referer: http://192.168.1.3/14/temp26  
11:34:23.505 -> Accept-Encoding: gzip, deflate  
11:34:23.505 -> Accept-Language: en-US,en;q=0.9  
11:34:23.505 ->  
11:34:23.608 -> IR set temp to 26 signal sent  
11:34:23.713 -> Client disconnected.
```

41) Παράδειγμα λειτουργικότητας εικόνα 3α

Και στην συνέχεια η επιλογή 'COLD'.



Smart Box AC Controller

Air Condition on

OFF

Temperature: 26.12 *C

Humidity: 60.35 %

Refresh: [CHECK](#)

AC Mode:

[Heat](#) | [Dry](#) | [Cold](#) | [Fan](#)

Current mode: **COLD**

Set temp:

[|18|](#) [|19|](#) [|20|](#) [|21|](#) [|22|](#) [|23|](#) [|24|](#) [|25|](#) [|26|](#) [|27|](#) [|28|](#) [|29|](#)

Current AC temp: **26**

42) Παράδειγμα λειτουργικότητας εικόνα 4

```
11:34:23.713 -> New Client.  
11:34:27.386 -> GET /14/coldon HTTP/1.1  
11:34:27.386 -> Host: 192.168.1.3  
11:34:27.386 -> Connection: keep-alive  
11:34:27.386 -> Upgrade-Insecure-Requests: 1  
11:34:27.386 -> User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 OPR/60.0.3255.109  
11:34:27.386 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8  
11:34:27.386 -> Referer: http://192.168.1.3/14/temp26  
11:34:27.421 -> Accept-Encoding: gzip, deflate  
11:34:27.421 -> Accept-Language: en-US,en;q=0.9  
11:34:27.421 ->  
11:34:27.525 -> IR cold mode signal sent  
11:34:27.594 -> Client disconnected.
```

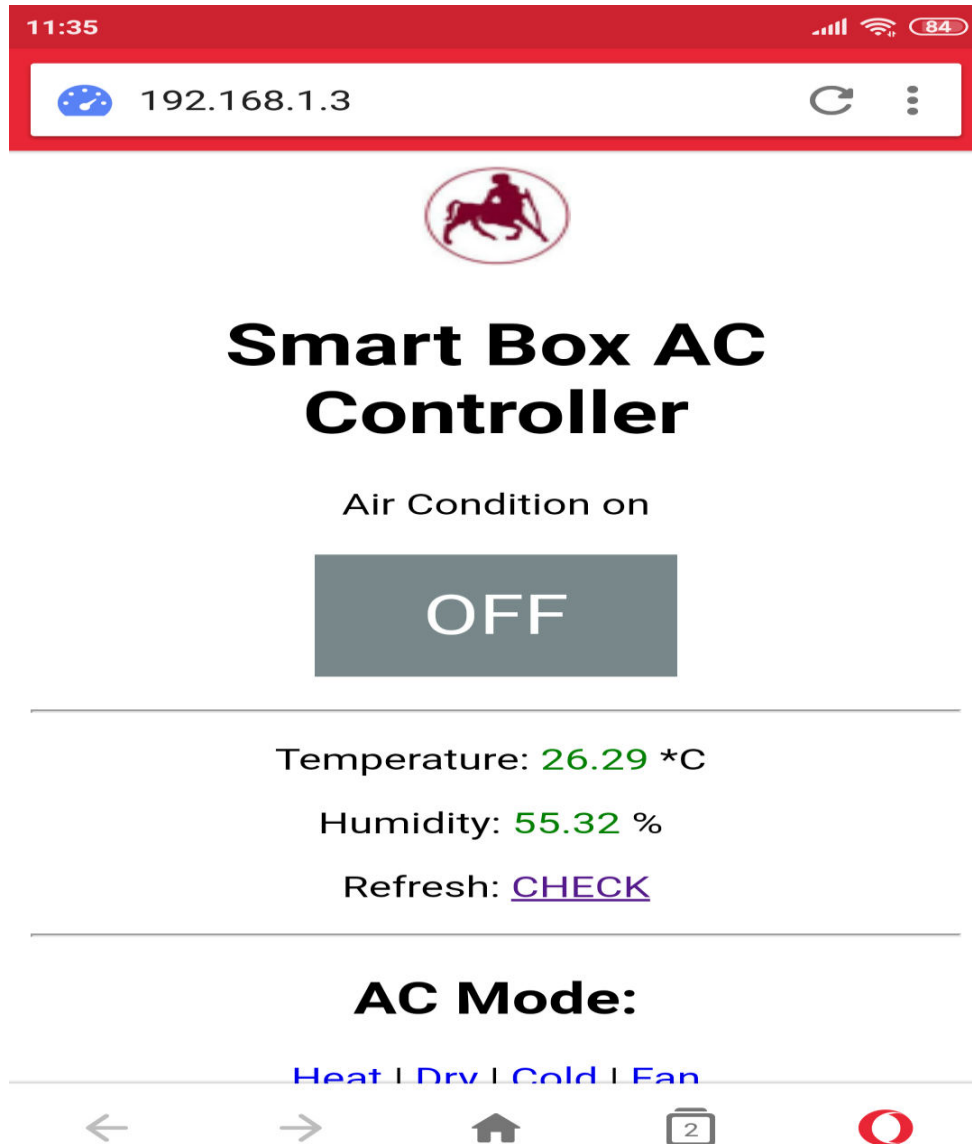
43) Παράδειγμα λειτουργικότητας εικόνα 4α

Έστω ότι ο χρήστης επιλέγει να κλείσει την καρτέλα του browser από το desktop γιατί θέλει να το διαχειριστεί μέσω κινητού.

```
11:34:27.594 -> New Client.  
11:34:44.795 -> Client disconnected.
```

44) Παράδειγμα λειτουργικότητας εικόνα 5

Και ανοίγει την εφαρμογή απ' το κινητό.



45) Παράδειγμα λειτουργικότητας εικόνα 6

Air Condition On

OFF

Temperature: 26.29 *C

Humidity: 55.32 %

Refresh: [CHECK](#)

AC Mode:

[Heat](#) | [Dry](#) | [Cold](#) | [Fan](#)Current mode: **COLD**

Set temp:

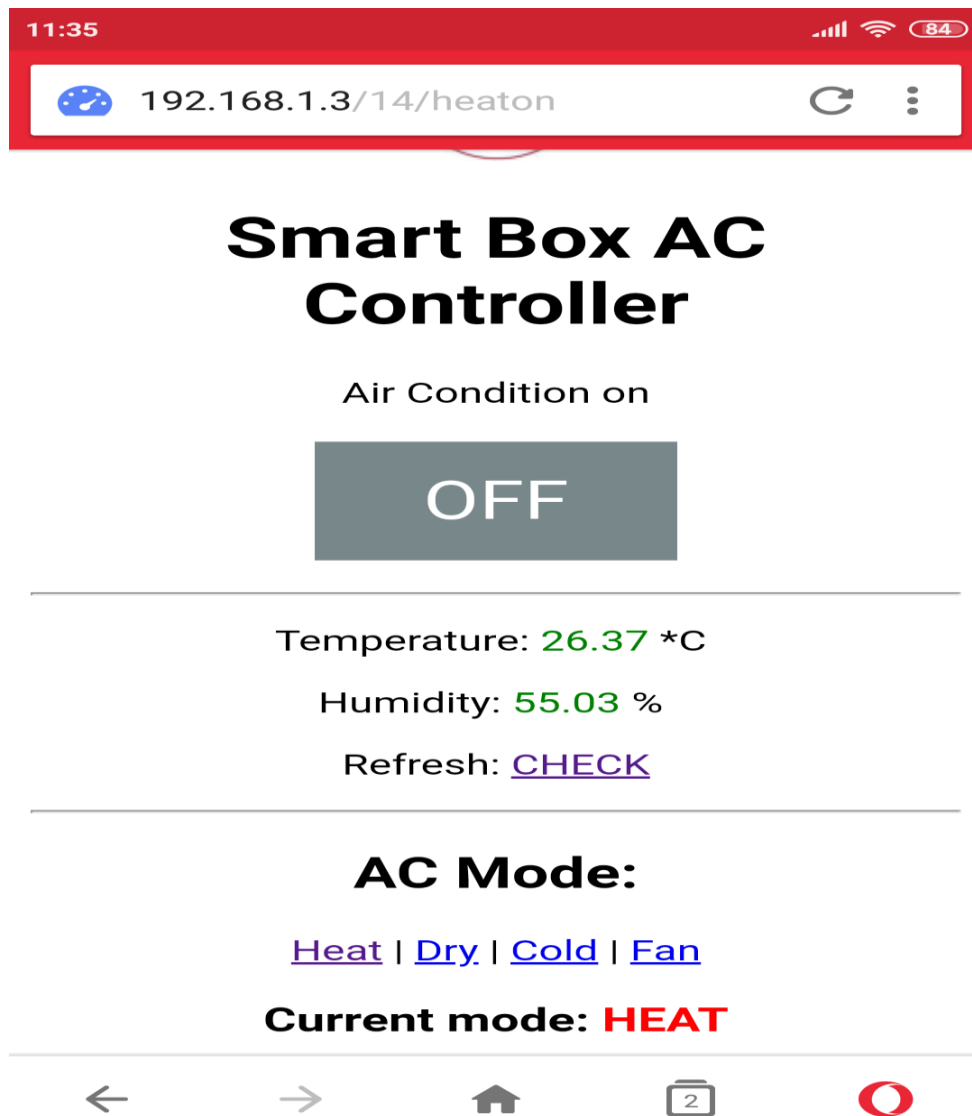
[18](#) | [19](#) | [20](#) | [21](#) | [22](#) | [23](#) | [24](#) | [25](#) | [26](#)
[27](#) | [28](#) | [29](#)Current AC temp: **26**

```
11:35:34.103 -> New Client.
11:35:34.103 -> GET / HTTP/1.1
11:35:34.103 -> Host: 192.168.1.3
11:35:34.103 -> Connection: keep-alive
11:35:34.103 -> Upgrade-Insecure-Requests: 1
11:35:34.103 -> User-Agent: Mozilla/5.0 (Linux; U; Android 6.0; Redmi Note 4 Build/MRA58K; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/74.0.3729.157 Mobile Safari/537.36 OPR/42.0.2254.139276
11:35:34.137 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
11:35:34.137 -> Accept-Encoding: gzip, deflate
11:35:34.137 -> Accept-Language: en-US,en;q=0.9
11:35:34.137 -> X-Requested-With: com.opera.mini.native
11:35:34.170 ->
11:35:34.373 -> Client disconnected.
11:35:34.373 ->
```

47) Παράδειγμα λειτουργικότητας εικόνα 6β

Παρατηρείται, ότι παρόλο που έκλεισε το interface επιλογών του desktop, οι τελευταίες επιλογές που πραγματοποιήθηκαν είναι αποθηκευμένες, στο interface του κινητού που μόλις ανοίχθηκε.

Τώρα έστω ότι επιλέγεται μέσω κινητού η επιλογή 'HEAT'.



48) Παράδειγμα λειτουργικότητας εικόνα 7


```
11:35:52.502 -> New Client.
11:35:52.502 -> GET /14/beacon HTTP/1.1
11:35:52.502 -> Host: 192.168.1.3
11:35:52.502 -> Connection: keep-alive
11:35:52.502 -> Upgrade-Insecure-Requests: 1
11:35:52.502 -> User-Agent: Mozilla/5.0 (Linux; U; Android 6.0; Redmi Note 4 Build/NRA58K; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/74.0.3729.157 Mobile Safari/537.36 OPR/42.0.2254.139276
11:35:52.536 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
11:35:52.536 -> Referer: http://192.168.1.3/
11:35:52.536 -> Accept-Encoding: gzip, deflate
11:35:52.536 -> Accept-Language: en-US,en;q=0.9
11:35:52.536 -> X-Requested-With: com.opera.mini.native
11:35:52.569 ->
11:35:52.638 -> IR heat mode signal sent
11:35:52.740 -> Client disconnected.
```

49) Παράδειγμα λειτουργικότητας εικόνα 7α

Και τώρα η επιλογή 'OFF'.

11:36

192.168.1.3/14/off



Smart Box AC Controller

Air Condition off

ON

Temperature: **26.40** *C

Humidity: **54.80** %

Refresh: [CHECK](#)

AC Mode:

Heat | Dry | Cold | Fan

← → 🏠 📅 🔴

50) Παράδειγμα λειτουργικότητας εικόνα 8

```
11:36:07.321 -> New Client.
11:36:07.321 -> GET /14/off HTTP/1.1
11:36:07.321 -> Host: 192.168.1.3
11:36:07.321 -> Connection: keep-alive
11:36:07.321 -> Upgrade-Insecure-Requests: 1
11:36:07.321 -> User-Agent: Mozilla/5.0 (Linux; U; Android 6.0; Redmi Note 4 Build/MRA58K; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/74.0.3729.157 Mobile Safari/537.36 OPR/42.0.2254.139276
11:36:07.356 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
11:36:07.356 -> Referer: http://192.168.1.3/14/heatmap
11:36:07.356 -> Accept-Encoding: gzip, deflate
11:36:07.356 -> Accept-Language: en-US,en;q=0.9
11:36:07.356 -> X-Requested-With: com.opera.mini.native
11:36:07.356 ->
11:36:07.457 -> IR power Off signal sent
11:36:07.525 -> Client disconnected.
11:36:07.525 ->
```

51) Παράδειγμα λειτουργικότητας εικόνα 8α

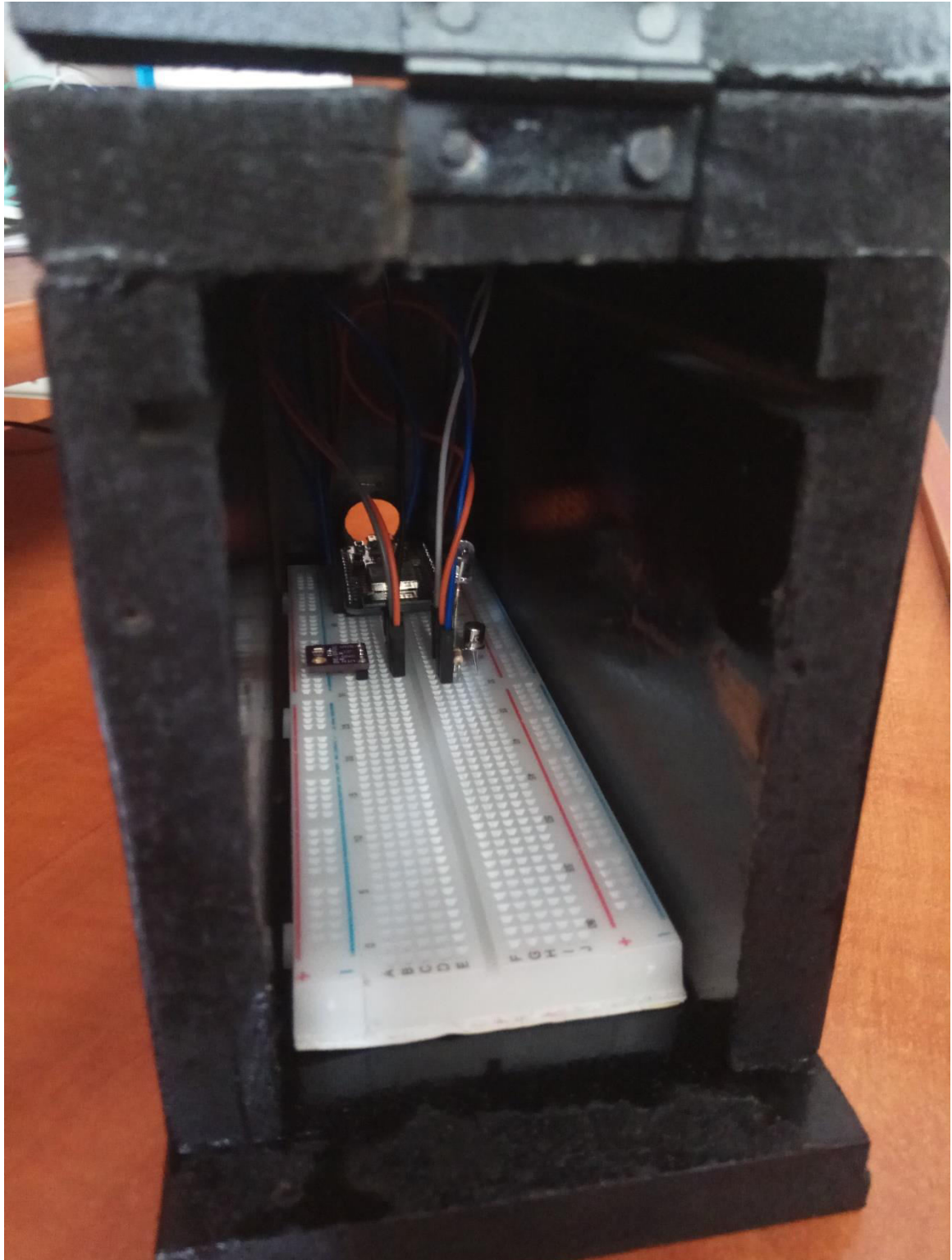
Στην συνέχεια παρουσιάζονται διαφορετικές όψεις του κουτιού.

Πρόσοψη:



52) Πρόσοψη Έξυπνου Κουτιού

Εσωτερικό:



53) Εσωτερικό Έξυπνου Κουτιού

Πλάγια:



54) Πλάγια όψη Έξυπνου Κουτιού

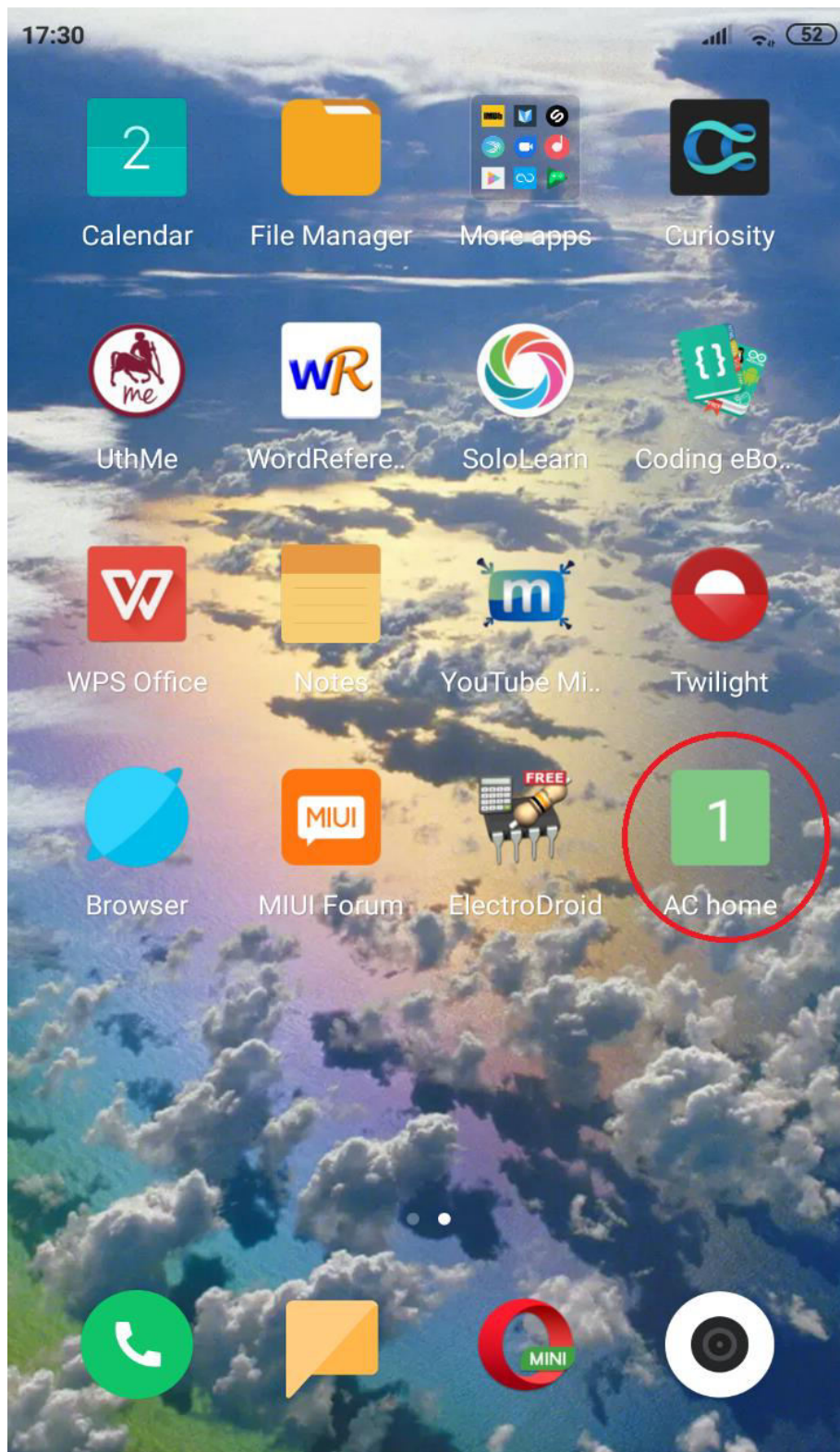
Παρατηρείται ότι το κουτί έχει 2 τρύπες. Η τρύπα στα πλάγια είναι για την είσοδο καλωδίου micro usb, και η τρύπα στο πάνω μέρος είναι για το σήμα που εκπέμπεται από το IR LED.

Στην συνέχεια ακολουθούν εικόνες που παρουσιάζουν την λειτουργία.



55) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 1

Ανοίγεται η εφαρμογή που έχει δημιουργηθεί στο το κινητό.




56) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 2

Επιλέγεται το κουμπί 'ON'.

17:30

192.168.1.3



Smart Box AC Controller

Air Condition off

ON

Temperature: 28.32 *C

Humidity: 48.58 %

Refresh: [CHECK](#)

AC Mode:


[Heat](#) | [Dry](#) | [Cold](#) | [Fan](#)

← → 🏠 📅 2 🔴

57) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 3

→

17:30 192.168.1.3/14/on



Smart Box AC Controller

Air Condition on

OFF

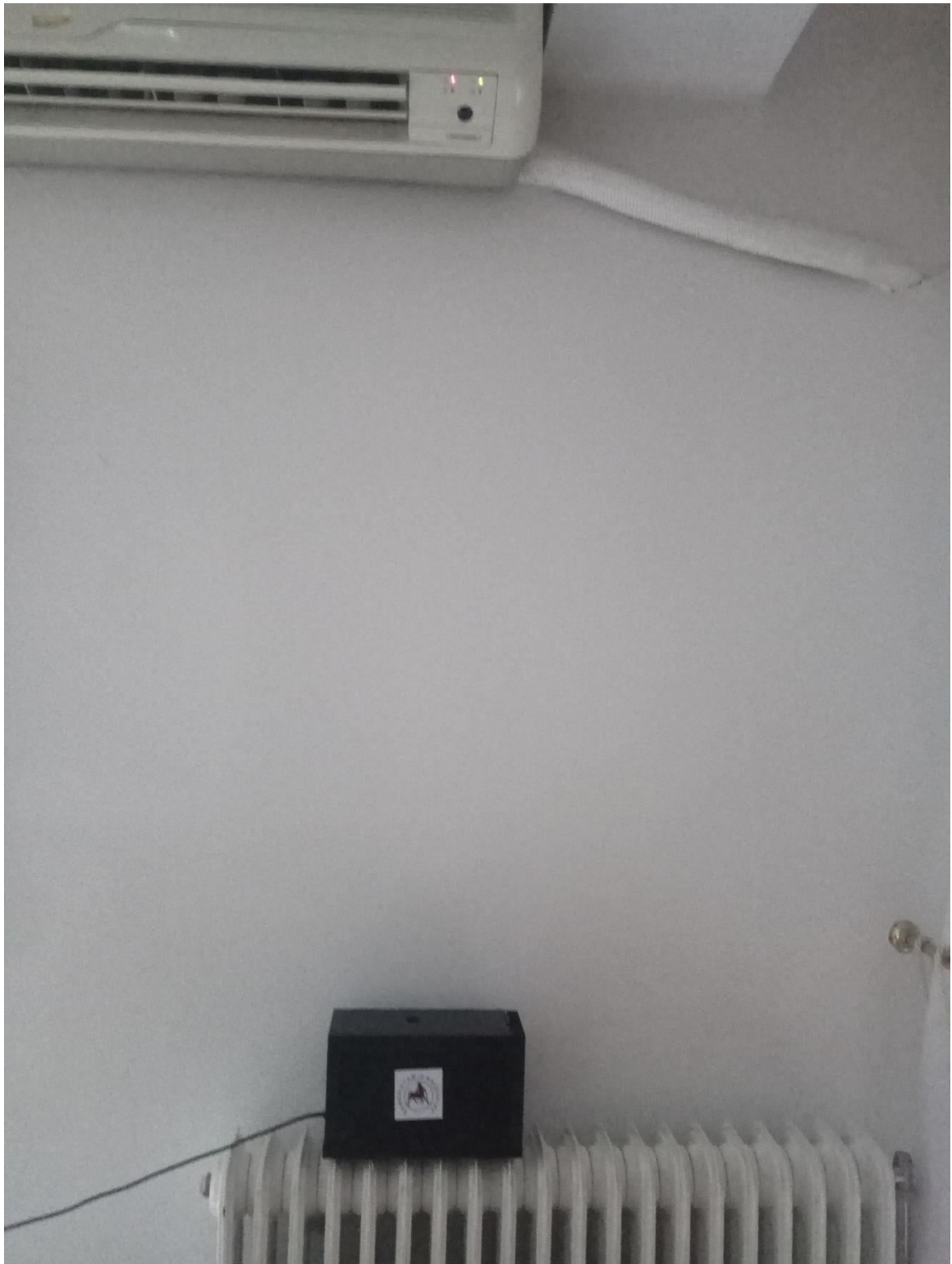
Temperature: 28.32 *C
Humidity: 48.41 %
Refresh: [CHECK](#)

AC Mode:
[Heat](#) | [Dry](#) | [Cold](#) | [Fan](#)

← → 🏠 📅 🌐

58) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 4

Παρατηρείται ότι το σήμα στάλθηκε και το Air Condition άνοιξε.



59) Παράδειγμα Έξυπνου Κουτιού σε λειτουργία εικόνα 5

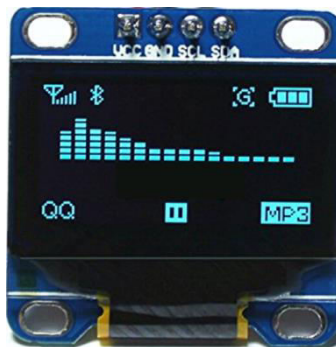
5. Συμπεράσματα & Μελλοντικές Προεκτάσεις

Α)Επέκταση σήματος και συσκευών ελέγχου

Αρχικά παρατηρείται ότι το Smart Box έχει τοποθετηθεί σε γωνία σχεδόν κάθετη από το Air Condition. Ο λόγος που γίνεται αυτό είναι, διότι το IR LED που έχει χρησιμοποιηθεί, λειτουργεί σε οπτική γωνία μέχρι 20 μοίρες (υπάρχουν και παρόμοια IR LED με οπτική γωνία 45 μοίρες). Συνεπώς για να μπορέσει το κουτί, να στείλει το σήμα θα πρέπει να βρίσκεται μέσα σε αυτό το πεδίο. Ένα ακόμα που πρέπει να προσεχθεί, είναι ότι η απόσταση την οποία έχει από το Air Condition, είναι σχεδόν η μέγιστη απόσταση, στην οποία μπορεί να μεταδώσει το σήμα. Αν είναι επιθυμητό να χρησιμοποιηθεί το Smart Box για τον έλεγχο μίας ηλεκτρικής συσκευής, η οπτική γωνία και η απόσταση λειτουργίας του IR LED δεν αποτελούν ιδιαίτερο πρόβλημα, καθώς είναι δυνατόν να προσαρμοστεί στην κατάλληλη απόσταση και γωνία με ποικίλους τρόπους. Έστω όμως, ότι είναι επιθυμητή η επέκταση της λειτουργικότητάς του και ο έλεγχος και άλλων υπέρυθρα τηλεχειριζόμενων ηλεκτρικών συσκευών μέσα στο σπίτι, όπως για παράδειγμα η τηλεόραση. Σε αυτήν την περίπτωση γίνεται κατανοητό ότι ο αισθητήρας αυτός δεν είναι πρακτικός. Μία απλή λύση για την επίλυση αυτού του προβλήματος είναι η αγορά ενός καλύτερου αισθητήρα IR, όπως ο Grove Infrared Emitter, ο οποίος μπορεί να στείλει σήμα σε απόσταση 10 μέτρων και κοστίζει γύρω στα 3-4 ευρώ (arduino store).

Β)Βελτιστοποίηση εξωτερικής κατασκευής (κουτιού)

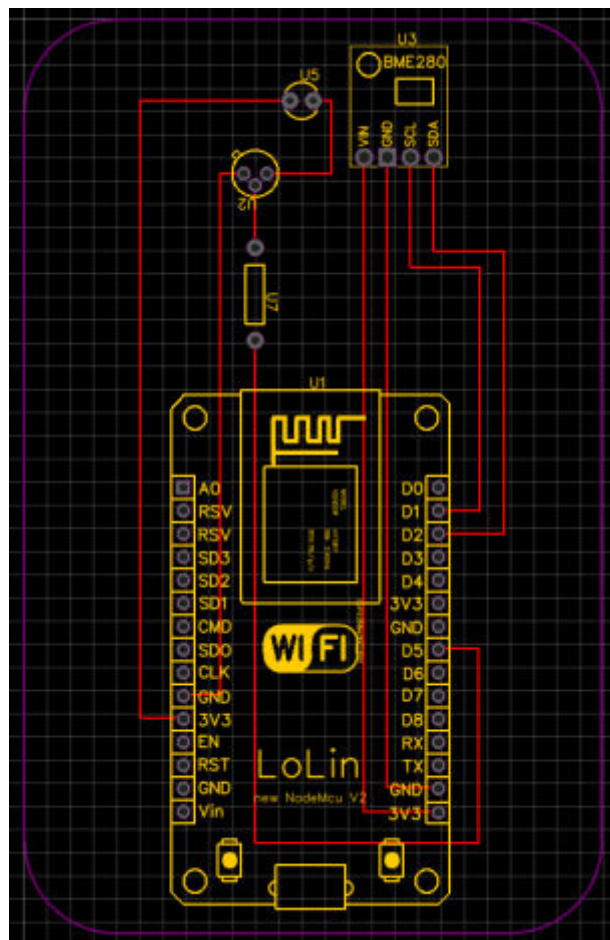
Μια εύκολη προσθήκη που μπορεί να κάνει το Smart Box πιο διαδραστικό με το περιβάλλον (ώστε οι μετρήσεις της θερμοκρασίας και της υγρασίας, εκτός από το website/app να εμφανίζονται και στο κουτί) είναι μια οθόνη LED, όπως η παρακάτω:



60) 128X64 OLED DISPLAY MODULE 0.96" I2C COMMUNICATION FOR ARDUINO

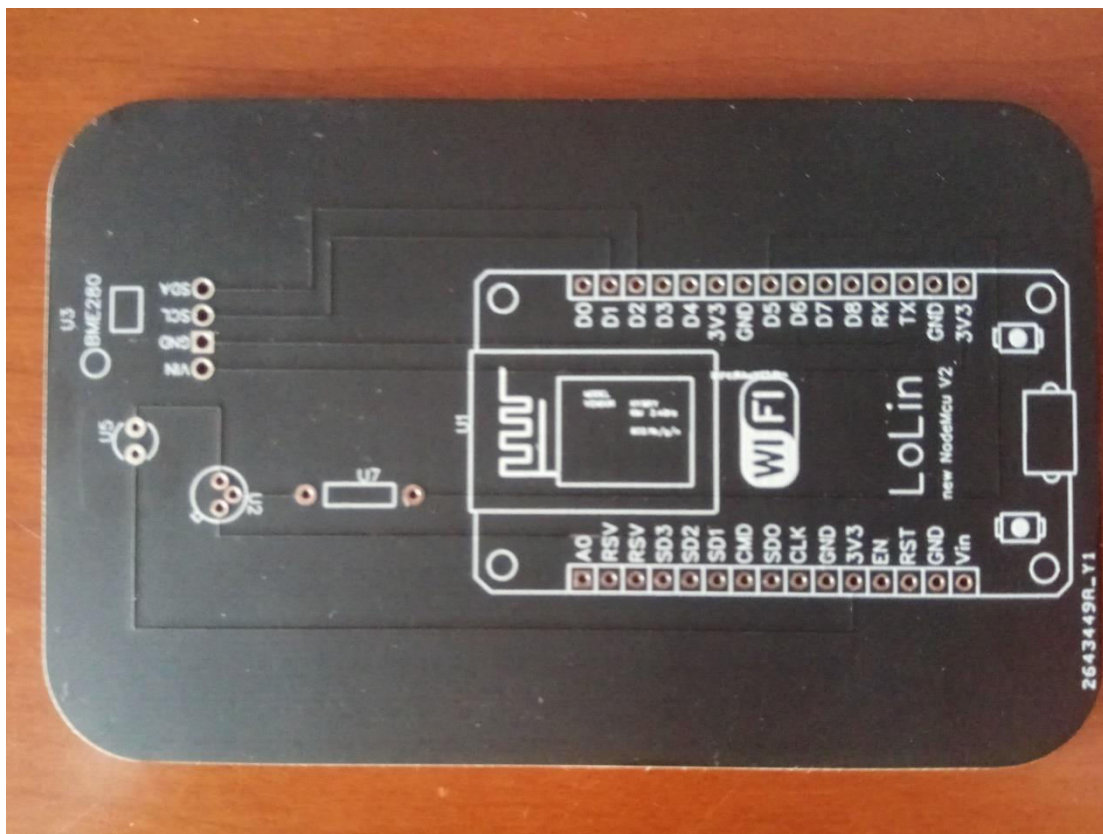
Ωστόσο, επειδή όπως αναφέρθηκε παραπάνω, το Esp8266 διαθέτει μόνο έναν I2C δίαυλο, για να συνδεθεί αυτή η οθόνη με το υπόλοιπο σύστημα, θα πρέπει να συνδεθεί το SCL της οθόνης με το SCL του αισθητήρα, και το SDA της οθόνης με το SDA του αισθητήρα. Σε διαφορετική περίπτωση δεν θα λειτουργήσει.

Όπως επισημάνθηκε στην επικεφαλίδα 3.1.8, τα breadboard, συνήθως χρησιμοποιούνται για προσωρινή χρήση. Για τον λόγο αυτό, προτείνεται η κατασκευή μιας PCB (printed circuit board/πλακέτα τυπωμένου κυκλώματος). Μια PCB μπορεί να σχεδιαστεί στο EasyDEA.com. Ένα τέτοιο σχέδιο μπορεί να είναι, σύμφωνα κατασκευή που σχεδιάστηκε το ακόλουθο:



61) Σχεδιασμός πλακέτας τυπωμένου κυκλώματος (PCB) στο EasyDEA

Έπειτα από τον επιθυμητό σχεδιασμό, μπορεί να γίνει η παραλαβή της πλακέτας από το JLCPCB.com όπου θα σχεδιαστεί και κατασκευαστεί, με κόστος 1.78 ευρώ. Η τελική πλακέτα όπως σχεδιάστηκε στην εικόνα 61, θα έχει για παράδειγμα την εξής μορφή:



62) Τελική μορφή πλακέτας τυπωμένου κυκλώματος

Φυσικά, για την κατάλληλη λειτουργία του συστήματος, θα χρειαστεί να συγκολληθούν όλα τα απαραίτητα υλικά, όπως για παράδειγμα ο αισθητήρας bme280, η οθόνη LED και στην συνέχεια όλα τα εξαρτήματα που είναι επιθυμητό να χρησιμοποιηθούν πάνω στην σχεδιασμένη πλακέτα τυπωμένου κυκλώματος.

Η συγκεκριμένη επιλογή του κουτιού έγινε για πρακτικούς και οικονομικούς λόγους, ωστόσο για μια πιο επαγγελματική κατασκευή, προτείνεται η δημιουργία κουτιού με 3D σχεδιασμό και εκτύπωση.

Γ) Ομορφότερος οπτικός σχεδιασμός του συστήματος διαχείρισης

Το σχεδιασμένο σύστημα διαχείρισης του Έξυπνου Κουτιού, δημιουργημένο με HTML, CSS και HTTP παρόλο που είναι απλό και εύκολο στην λειτουργία του, παρουσιάζει αδυναμίες όσον αφορά την περαιτέρω βελτίωση της δόμησής του. Για έναν πιο ιδανικό και ομορφότερο σχεδιασμό, θα μπορούσε να επιλεγεί η χρήση των HTML, CSS, HTTP, JavaScript, ArduinoJSON (JSON=JavaScript Object Notation), ArduinoOTA (OTA=Over The Air), mdns (multicast DNS[Domain Name System]) και MQTT (Message Queuing Telemetry Transport), όπως επίσης και η μετατροπή της κατασκευασμένης –μέσω αυτών- ιστοσελίδας (website), σε εφαρμογή κινητού /tablet (web app) με χρήση του manifest.json.

Δ)Έλεγχος Έξυπνου Κουτιού από παντού

Ο τρέχων έλεγχος του συστήματος του Έξυπνου Κουτιού μπορεί να γίνεται εφόσον ο χρήστης είναι συνδεδεμένος στο ίδιο διαδίκτυο με το οποίο είναι συνδεδεμένο και το Esp8266, το οποίο παρότι έχει μερικά πλεονεκτήματα έναντι του τηλεχειριστηρίου, υστερεί αρκετά μπροστά στο σύνολο δυνατοτήτων που μπορεί να προσφέρει η διαχείρισή του από οποιοδήποτε διαδίκτυο. Για παράδειγμα, ο χρήστης βρίσκεται στον δρόμο επιστροφής προς το σπίτι μετά από τις χειμερινές του διακοπές και δεν επιθυμεί να βρει το σπίτι του παγωμένο. Ανοίγει πολύ απλά (είτε με χρήση κάποιου Wi-Fi είτε με χρήση δεδομένων κινητού) την εφαρμογή διαχείρισης του Smart Box απ' το κινητό του και στην συνέχεια ανοίγει το Air Condition προσαρμόζοντας το στις κατάλληλες ρυθμίσεις που θέλει (λόγου χάρη: είδος λειτουργίας, ζέστη και θερμοκρασία 26). Ή για παράδειγμα, αν βρίσκεται στην δουλειά και έχει μικρά παιδιά σπίτι, ανοίγει την εφαρμογή στο κινητό ή την ιστοσελίδα απ' το desktop/laptop εργασίας, βλέπει την θερμοκρασία στην οποία βρίσκεται το σπίτι και προσαρμόζει τις ρυθμίσεις σύμφωνα με το αποτέλεσμα που επιδιώκει. Ένας τρόπος με τον οποίο μπορεί να επιτευχθεί κάτι τέτοιο είναι το Port Forwarding. Η διαδικασία αυτή διαφέρει από router σε router, αλλά η γενική ιδέα είναι η εξής:

Έστω ότι υπάρχουν οι παρακάτω διευθύνσεις:

Internal Ip:192.168.1.3 (private ip)

External Ip:11.22.33.444 (public ip)

Port:80

Αντί να πραγματοποιηθεί η είσοδος στην διεύθυνση **192.168.1.3** στην οποία θα γινόταν η σύνδεση από το ίδιο Wi-Fi (interface-esp8266), στην περίπτωση που είναι επιθυμητή η σύνδεση με διαφορετικό Wi-Fi απ' ότι είναι το Esp8266, θα πληκτρολογηθεί η διεύθυνση **11.22.33.444:80**.

Παρακάτω αναλύεται και ένας συγκεκριμένος τύπος router, όπως το TP-LINK:

The screenshot shows the TP-LINK web interface. On the left is a sidebar menu with the following items: Status, Quick Setup, QSS, Network, Wireless, DHCP, Forwarding (highlighted), Virtual Servers, Port Triggering, DMZ, UPnP, Security, Parental Control, Access Control, Advanced Routing, Bandwidth Control, IP & MAC Binding, Dynamic DNS, and System Tools. The main content area is titled 'Add or Modify a Virtual Server Entry'. It contains the following fields:

- Service Port: 80 (with a hint: 00-100 or 100)
- Internal Port: 80 (with a hint: 00, Only valid for single Service Port or leave a blank)
- IP Address: 192.168.0
- Protocol: ALL (dropdown menu)
- Status: Enabled (dropdown menu)
- Common Service Port: --Select One-- (dropdown menu)

At the bottom of the form are two buttons: 'Save' and 'Back'.

63) Ρυθμίσεις για Port Forwarding σε TP-LINK router

Στην συγκεκριμένη περίπτωση, στο Service Port, πρέπει να εισαχθεί το Port που έχει τεθεί στον κώδικα Arduino, και στην IP Address να εισαχθεί η External (public) IP.

Βιβλιογραφία

Big Data and The Internet of Things – Robert Stackowiak , Art Licht, Venu Mantha, and Louis Nagode

Journal Of Industrial Information Integration – Shancang Li , Li Da Xu , Shanshan Zhao

The role of big data analytics in Internet of Things - Ejaz Ahmeda , Ibrar Yaqooba , Ibrahim Abaker Targio Hashem , Imran Khan , Abdelmutilib Ibrahim Abdalla Ahmed , Muhammad Imran , Athanasios V. Vasilakos

Enabling technologies and user perception within integrated 5G-IoT ecosystem – Ing Jiri Hosek

Industry 4.0: The Industrial Internet of Things – Alasdair Gilchrist

<https://www.techopedia.com/definition/28247/internet-of-things-iot>

<https://www.newmoney.gr/palamos-oikonomias/tecnologia/312322-ti-einai-to-internet-of-things>

<https://www.iotforall.com/what-is-iot-simple-explanation/>

https://www.tutorialspoint.com/internet_of_things/

https://www.tutorialspoint.com/internet_of_things/internet_of_things_overview.htm

https://www.tutorialspoint.com/internet_of_things/internet_of_things_common_uses.htm

<https://www.quora.com/What-are-the-advantages-and-disadvantages-of-IOT>

<https://www.linkedin.com/pulse/advantages-disadvantages-internet-things-iot-tommy-quek>

<http://sticamsud.org/2018/10/09/the-advantages-of-internet-of-things-iot/>

<https://www.i-scoop.eu/internet-of-things-guide/5g-iot/>

<https://www.vodafone.com/business/news-and-insights/white-paper/vodafone-iot-barometer-2019>

<https://jaxenter.com/relationship-between-iot-big-data-138220.html>

<https://www.digitalvidya.com/blog/iot-and-its-impact-on-bigdata/>

<https://www.tutorialspoint.com/internet-of-things/internet-of-things-media-marketing-and-advertising.htm>

<https://el.wikipedia.org/wiki/I²C>

https://el.wikipedia.org/wiki/Πύλη_δικτύου

<https://www.slideshare.net/kokkinoulitsa/cloud-computing-71824176>

<https://devopedia.org/iot-cloud-platforms>

<https://iot4beginners.com/best-internet-of-things-iot-cloud-platforms/>

https://el.wikipedia.org/wiki/Πρωτόκολλο_Μεταφοράς_Υπερκειμένου

<https://grobotronics.com/arduino-mega-2560-rev3.html?sl=el>

<https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-mega-2560.html>

<https://store.arduino.cc>

<https://grobotronics.com/arduino-uno-rev3.html?sl=el>

<https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html>

<https://www.skroutz.gr/c/1607/Arduino.html?from=catspan&keyphrase=esp8266>

<https://www.cableworks.gr/ilektronika/arduino-and-microcontrollers/microcontrollers/esp8266/esp8266-nodemcu-with-cp2102-wifi-internet-development-board/>

<https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/>

<https://lastminuteengineers.com/bme280-arduino-tutorial/>

<https://www.aliexpress.com/item/32672210336.html>

<https://www.ebay.com/sch/i.html?from=R40&trksid=m570.l1313&nkw=bme280&sacat=0>

<https://www.engineersgarage.com/electronic-components/tsop1738-datasheet>

<https://www.indiamart.com/proddetail/tsop-sensor-9432838012.html>

<https://el.wikipedia.org/wiki/Τρανζίστορ>

<https://www.ebay.com/itm/10PCS-NEW-Transistor-ST-MOTOROLA-TO-18-CAN-3-2N2222A-2N2222-AU/302185476317>

<https://www.ebay.com/itm/20-x-2N2222A-2N2222-NPN-Transistor-0-8A-40V-TO-18-/250867361510>

<https://electronicsforu.com/resources/learn-electronics/ir-led-infrared-sensor-basics>

<https://www.ebay.com/itm/10PCS-5mm-940nm-IR-infrared-Launch-emission-tube-diode-LED-Lamp-Emitting/191736129142>

https://el.wikipedia.org/wiki/Πλακέτα_διασύνδεσης_χωρίς_κολλήσεις

<https://www.analysir.com/blog/2014/03/19/air-conditioners-problems-recording-long-infrared-remote-control-signals-arduino/>

Παράρτημα Α: Πηγαίος Κώδικας

Αρχείο: **Long_IR_Receiver.ino**

```
1 /*
2 Συνδεσμολογία:
3 Arduino Board - IR Reicever
4 GND - GND
5 5V/3.3V - VCC
6 Digital Pin 2 - OUTPUT
7 */
8
9 #define LEDPIN 13
10 //μπορούμε να αυξήσουμε κιάλλο αυτήν την τιμή σε Arduino που έχουν
SRAM μεγαλύτερη από 2 KB
11 #define maxLen 800
12
13 volatile unsigned int irBuffer[maxLen];
14 volatile unsigned int x = 0;
15
16 void setup() {
17
18   Serial.begin(115200);
19   attachInterrupt(0, rxIR_Interrupt_Handler, CHANGE);
20
21 }
22
23 void loop() {
24
25   Serial.println(F("Press the button on the remote now - once
only"));
26   delay(5000);
27   if (x) {
28     digitalWrite(LEDPIN, HIGH);
29     Serial.println();
30     Serial.print(F("Raw: ("));
31     Serial.print((x - 1));
32     Serial.print(F(") "));
33     detachInterrupt(0);
34     for (int i = 1; i < x; i++) {
35       if (!(i & 0x1)) Serial.print(F("-"));
36       Serial.print(irBuffer[i] - irBuffer[i - 1]);
37       Serial.print(F(", "));
38     }
39     x = 0;
40     Serial.println();
41     Serial.println();
42     digitalWrite(LEDPIN, LOW);
43     attachInterrupt(0, rxIR_Interrupt_Handler, CHANGE);
44   }
45 }
46 }
47
48 void rxIR_Interrupt_Handler() {
49
50   if (x > maxLen) return;
51   irBuffer[x++] = micros();
52 }
```

```
53 }
54
```

Αρχείο: Test_IR_sender.ino

```
1 #include <IRremote.h>
2
3 IRsend irsend;
4
5 void setup()
6 {
7 }
8
9 unsigned int sendbuf1[] = {
10 8476, 4196, 580, 496, 560, 1544, 560, 496, 536, 1588, 508, 544,
11 560, 1544, 556, 1544, 560, 512, 560, 1544, 560, 496, 560, 1548,
12 504, 568, 532, 1568, 532, 524, 532, 1568, 560, 1564, 508, 548,
13 556, 496, 560, 496, 560, 1564, 556, 496, 560, 1544, 536, 520,
14 560, 1564, 532, 524, 532, 520, 536, 520, 556, 516, 560, 496,
15 536, 520, 536, 520, 532, 540, 560, 496, 536, 520, 536, 520,
16 560, 512, 560, 1544, 560, 496, 536, 520, 560, 512, 560, 1544,
17 560, 496, 560, 496, 560, 512, 560, 496, 560, 496, 536, 1572,
18 528, 544, 556, 496, 536, 520, 560, 496, 556, 516, 560, 496,
19 560, 496, 536, 520, 536, 536, 560, 496, 560, 496, 560, 496,
20 560, 512, 560, 496, 536, 520, 560, 496, 560, 516, 556, 496,
21 560, 1544, 560, 496, 560, 1560, 560, 1548, 532, 1568, 532,
22 524, 532, 520, 560};
23
24 void loop() {
25
26   testRaw("RAW1", sendbuf1, sizeof(sendbuf1)/sizeof(int));
27   delay(3000);
28
29 }
30
31 void testRaw(char *label, unsigned int *rawbuf, int rawlen) {
32
33   irsend.sendRaw(rawbuf, rawlen, 38 /* kHz */);
34   delay(200);
35
36 }
37
```

Αρχείο: simple_bme_test.ino

```
1 #include <Wire.h>
2 #include <Adafruit_Sensor.h>
3 #include <Adafruit_BME280.h>
4
5
6 Adafruit_BME280 bme;
7
8 void setup() {
9
10  Serial.begin(115200);
11  delay(100);
12
13  bme.begin(0x76);
```



```

14
15 }
16
17 void loop() {
18
19   Serial.println("Temperature = ");
20   Serial.println(bme.readTemperature());
21   Serial.println("Humidity = ");
22   Serial.println(bme.readHumidity());
23
24   delay(2000);
25 }
26

```

Αρχείο: Smart_Box_AC.ino

```

1 // απαραίτητες βιβλιοθήκες
2 #include <ESP8266WiFi.h>
3 #include <IRsend.h>
4 #include <IRremoteESP8266.h>
5 #include <Wire.h>
6 #include <Adafruit_Sensor.h>
7 #include <Adafruit_BME280.h>
8
9 // συνδεση στο wifi network
10 const char* ssid = "Forthnet-7Eght";
11 const char* password = "zbtysjDj";
12
13 Adafruit_BME280 bme;
14
15 int khz = 38; // 38kHz συχνότητα για το σήμα που στέλνει
16
17 IRsend irsend(14); //το IR LED είναι συνδεδεμένο στο GPIO pin
14(D5)
18
19 // IR σήμα για κλείσιμο
20 uint16_t Off[] = {8476, 4196, 580, 496, 560, 1544, 560, 496, 536,
21 1588, 508, 544, 560, 1544, 556, 1544, 560, 512, 560, 1544, 560,
22 496, 560, 1548, 504, 568, 532, 1568, 532, 524, 532, 1568, 560,
23 1564, 508, 548, 556, 496, 560, 496, 560, 1564, 556, 496, 560,
24 1544, 536, 520, 560, 1564, 532, 524, 532, 520, 536, 520, 556,
25 516, 560, 496, 536, 520, 536, 520, 532, 540, 560, 496, 536, 520,
26 536, 520, 560, 512, 560, 1544, 560, 496, 536, 520, 560, 512, 560,
27 1544, 560, 496, 560, 496, 560, 512, 560, 496, 560, 496, 536, 1572,
28 528, 544, 556, 496, 536, 520, 560, 496, 556, 516, 560, 496, 560,
29 496, 536, 520, 536, 536, 560, 496, 560, 496, 560, 496, 560, 512,
30 560, 496, 536, 520, 560, 496, 560, 516, 556, 496, 560, 1544, 560,
31 496, 560, 1560, 560, 1548, 532, 1568, 532, 524, 532, 520, 560};
32
33 //IR σήμα για ανοίγμα
34 uint16_t On[] = {8476, 4192, 584, 496, 536, 1568, 532, 524, 556,
35 1564, 532, 524, 556, 1544, 536, 1568, 560, 512, 560, 1544, 556,
36 500, 556, 1548, 556, 516, 532, 1572, 532, 520, 536, 1572, 532,
37 1588, 556, 496, 560, 496, 560, 496, 536, 1588, 532, 520, 560,
38 1544, 532, 524, 556, 1564, 556, 500, 556, 500, 556, 496, 560,
39 516, 560, 496, 556, 500, 556, 500, 556, 516, 536, 520, 560, 496,
40 560, 496, 536, 536, 560, 1552, 528, 520, 556, 500, 532, 540,
41 536, 1568, 532, 524, 556, 500, 532, 540, 556, 500, 556, 500,
42 532, 524, 532, 1588, 532, 524, 560, 496, 556, 500, 556, 516,

```

```

43 560, 496, 560, 496, 556, 500, 532, 540, 560, 496, 560, 496, 560,
44 496, 556, 516, 560, 496, 560, 496, 536, 520, 536, 544, 552, 496,
45 560, 1544, 560, 1544, 532, 1588, 560, 1540, 560, 1544, 536, 520,
46 536, 520, 560};
47
48 // IR σήμα για θέση θερμοκρασίας στο 18
49 uint16_t temp18[] = {8456, 4220, 580, 496, 560, 1544, 560, 496,
50 556, 1564, 536, 520, 560, 1544, 508, 1592, 560, 512, 564, 492,
51 560, 1544, 560, 1524, 556, 1584, 512, 544, 560, 496, 560, 1540,
52 560, 1560, 536, 520, 560, 496, 560, 496, 560, 1560, 560, 496,
53 560, 1544, 560, 496, 560, 1560, 560, 496, 560, 496, 560, 496,
54 560, 512, 560, 496, 560, 496, 536, 520, 560, 512, 564, 492,
55 560, 496, 560, 496, 560, 516, 556, 1544, 560, 496, 560, 496,
56 560, 512, 560, 1548, 508, 544, 560, 496, 560, 512, 560, 496,
57 560, 496, 560, 496, 560, 512, 564, 492, 564, 492, 560, 496, 560,
58 516, 560, 496, 560, 492, 564, 492, 564, 512, 560, 496, 560,
59 496, 560, 496, 560, 512, 560, 496, 560, 496, 560, 496, 560, 512,
60 560, 496, 560, 1544, 512, 1592, 560, 1560, 560, 1540, 560, 1544,
61 560, 496, 560, 496, 560};
62
63 // IR σήμα για θέση θερμοκρασίας στο 19
64 uint16_t temp19[] = {8432, 4220, 580, 496, 560, 1544, 532, 524,
65 556, 1564, 560, 492, 564, 1540, 560, 1544, 512, 564, 556, 1544,
66 560, 1540, 564, 1540, 560, 1564, 508, 548, 556, 496, 560, 1544,
67 560, 1560, 560, 496, 560, 496, 560, 496, 560, 1560, 532, 524,
68 560, 1540, 536, 524, 560, 1556, 540, 516, 564, 492, 564, 472,
69 580, 516, 560, 496, 560, 496, 536, 516, 564, 512, 560, 496,
70 560, 496, 560, 496, 560, 512, 560, 1544, 560, 496, 560, 496,
71 560, 512, 560, 1544, 532, 524, 560, 492, 564, 512, 560, 496,
72 560, 496, 560, 496, 560, 512, 560, 496, 564, 492, 560, 496,
73 560, 512, 560, 496, 560, 496, 560, 496, 560, 512, 564, 492,
74 560, 496, 560, 496, 560, 516, 536, 516, 564, 476, 580, 492,
75 540, 536, 560, 1540, 560, 1548, 508, 1592, 560, 1560, 560,
76 1544, 536, 1564, 560, 496, 560, 496, 560};
77
78
79 // IR σήμα για θέση θερμοκρασίας στο 20
80 uint16_t temp20[] = {8480, 4196, 580, 496, 560, 1540, 560, 496,
81 560, 1564, 532, 524, 556, 1544, 556, 1548, 532, 540, 560, 496,
82 560, 496, 560, 496, 560, 512, 560, 1544, 560, 496, 556, 1544,
83 560, 1564, 532, 524, 556, 496, 560, 496, 560, 1564, 532, 520,
84 560, 1544, 560, 496, 560, 1560, 508, 548, 560, 496, 556, 500,
85 560, 512, 560, 496, 560, 496, 560, 496, 560, 512, 560, 496,
86 536, 520, 560, 496, 560, 512, 560, 1544, 560, 496, 560, 496,
87 556, 516, 560, 1544, 560, 496, 556, 500, 556, 516, 536, 520,
88 560, 496, 560, 496, 560, 512, 560, 496, 560, 496, 560, 496,
89 560, 512, 560, 496, 560, 496, 560, 496, 536, 536, 536, 520,
90 560, 496, 560, 496, 560, 516, 556, 500, 560, 496, 556, 496,
91 536, 540, 560, 1540, 560, 496, 560, 496, 560, 516, 532, 1568,
92 560, 1548, 504, 548, 556, 500, 560};
93
94 // IR σήμα για θέση θερμοκρασίας στο 21
95 uint16_t temp21[] = {8480, 4216, 556, 500, 560, 1540, 560, 496,
96 560, 1564, 508, 548, 556, 1544, 560, 1544, 556, 516, 560, 1544,
97 532, 524, 560, 496, 556, 516, 560, 1544, 560, 496, 556, 1544,
98 560, 1564, 532, 524, 556, 496, 560, 496, 560, 1560, 536, 520,
99 560, 1544, 560, 496, 536, 1584, 532, 524, 560, 496, 560, 496,
100 532, 540, 560, 496, 560, 496, 560, 496, 560, 512, 560, 496,
101 560, 496, 560, 496, 560, 512, 560, 1544, 560, 496, 560, 496,
102 560, 512, 560, 1524, 580, 496, 536, 520, 560, 512, 560, 496,
103 560, 496, 560, 496, 536, 536, 536, 520, 560, 496, 560, 496,

```

```

104 560, 516, 556, 500, 556, 496, 560, 496, 560, 516, 556, 500,
105 560, 496, 560, 492, 560, 516, 560, 496, 560, 496, 560, 496,
106 560, 512, 560, 496, 560, 1544, 560, 496, 560, 512, 560, 1544,
107 560, 1544, 532, 524, 532, 520, 560};
108
109 // IR σήμα για θέση θερμοκρασίας στο 22
110 uint16_t temp22[] = {8480, 4196, 580, 496, 560, 1544, 556, 500,
111 556, 1564, 560, 496, 536, 1568, 532, 1568, 560, 512, 560, 496,
112 560, 1544, 560, 496, 560, 512, 560, 1548, 532, 520, 560, 1544,
113 560, 1560, 560, 496, 532, 524, 560, 496, 556, 1564, 508, 548,
114 556, 1544, 560, 496, 536, 1584, 560, 496, 560, 496, 560, 496,
115 560, 512, 560, 500, 556, 496, 536, 520, 536, 540, 560, 496,
116 532, 520, 560, 496, 560, 516, 560, 1540, 560, 496, 560, 496,
117 560, 516, 560, 1544, 508, 548, 556, 496, 560, 516, 556, 500,
118 556, 496, 536, 520, 560, 516, 560, 496, 560, 496, 560, 496,
119 556, 520, 556, 496, 560, 496, 536, 520, 560, 512, 560, 496,
120 560, 496, 560, 496, 560, 512, 560, 496, 536, 520, 560, 496,
121 560, 516, 556, 1548, 508, 1592, 512, 544, 560, 512, 560, 1544,
122 536, 1564, 560, 496, 560, 496, 560};
123
124 // IR σήμα για θέση θερμοκρασίας στο 23
125 uint16_t temp23[] = {8452, 4196, 556, 520, 560, 1544, 536, 520,
126 556, 1564, 560, 496, 556, 1544, 560, 1548, 532, 540, 556, 1544,
127 560, 1544, 532, 524, 556, 516, 536, 1568, 532, 524, 556, 1544,
128 560, 1560, 560, 496, 560, 496, 560, 496, 560, 1564, 532, 524,
129 556, 1544, 560, 496, 556, 1564, 556, 500, 560, 496, 556, 500,
130 556, 516, 560, 496, 560, 496, 536, 520, 560, 512, 560, 496,
131 560, 496, 560, 496, 560, 512, 536, 1568, 536, 520, 536, 520,
132 560, 512, 536, 1568, 560, 496, 560, 496, 532, 540, 560, 496,
133 536, 520, 560, 496, 560, 512, 560, 500, 556, 496, 560, 496,
134 560, 516, 532, 520, 536, 520, 560, 496, 560, 516, 556, 500,
135 556, 500, 532, 520, 560, 516, 560, 496, 556, 500, 560, 496,
136 556, 516, 556, 500, 560, 496, 560, 1544, 532, 540, 560, 1544,
137 532, 1568, 560, 496, 560, 496, 560};
138
139 // IR σήμα για θέση θερμοκρασίας στο 24
140 uint16_t temp24[] = {8452, 4216, 584, 496, 560, 1544, 560, 496,
141 560, 1560, 560, 496, 560, 1544, 508, 1596, 556, 520, 556, 492,
142 560, 496, 560, 1548, 532, 540, 560, 1540, 560, 496, 560, 1544,
143 560, 1560, 560, 496, 560, 496, 556, 500, 560, 1560, 556, 500,
144 560, 1540, 560, 496, 560, 1564, 532, 524, 556, 496, 560, 496,
145 560, 512, 536, 520, 536, 520, 560, 500, 556, 516, 560, 496,
146 556, 500, 556, 500, 556, 516, 560, 1540, 560, 496, 560, 496,
147 560, 516, 560, 1544, 532, 524, 556, 496, 560, 516, 556, 500,
148 556, 500, 556, 496, 564, 512, 560, 496, 560, 496, 556, 500,
149 556, 516, 560, 496, 560, 496, 560, 496, 560, 492, 580, 496,
150 560, 496, 560, 496, 560, 512, 560, 496, 560, 496, 560, 496,
151 560, 512, 560, 1548, 532, 520, 560, 1544, 560, 512, 560, 1544,
152 560, 1544, 508, 548, 556, 480, 552};
153
154 // IR σήμα για θέση θερμοκρασίας στο 25
155 uint16_t temp25[] = {8480, 4192, 580, 500, 532, 1568, 560, 496,
156 560, 1564, 532, 524, 556, 1544, 560, 1544, 556, 516, 560, 1544,
157 556, 500, 556, 1548, 532, 540, 532, 1568, 560, 496, 536, 1568,
158 536, 1588, 532, 524, 556, 496, 556, 500, 532, 1588, 560, 496,
159 556, 1548, 532, 520, 536, 1588, 532, 524, 556, 500, 556, 496,
160 536, 540, 532, 524, 556, 496, 536, 520, 560, 516, 560, 496,
161 556, 500, 556, 500, 532, 540, 560, 1544, 532, 524, 532, 524,
162 556, 516, 536, 1568, 556, 500, 556, 496, 536, 540, 560, 496,
163 556, 500, 532, 524, 556, 516, 560, 496, 560, 496, 560, 496,
164 532, 540, 560, 496, 560, 496, 536, 520, 560, 512, 560, 496,

```

```

165 560, 496, 536, 520, 536, 536, 560, 496, 560, 496, 560, 496,
166 560, 516, 556, 500, 532, 1568, 536, 1568, 532, 540, 536, 1572,
167 532, 1568, 532, 524, 556, 500, 556};
168
169 // IR σήμα για ανέβασμα θερμοκρασίας στο 26
170 uint16_t temp26[] = {8476, 4216, 560, 496, 560, 1544, 532, 524,
171 556, 1564, 556, 500, 556, 1544, 536, 1568, 536, 536, 536, 520,
172 560, 1544, 560, 1544, 532, 540, 560, 1544, 556, 500, 556, 1548,
173 532, 1588, 556, 496, 560, 496, 560, 496, 560, 1564, 532, 520,
174 560, 1544, 556, 500, 556, 1564, 532, 524, 556, 500, 556, 500,
175 532, 540, 560, 496, 532, 524, 560, 496, 556, 516, 560, 496,
176 560, 496, 536, 520, 560, 512, 560, 1544, 560, 496, 536, 520,
177 560, 512, 560, 1544, 556, 500, 556, 500, 532, 540, 536, 520,
178 532, 524, 560, 496, 556, 516, 560, 496, 560, 496, 560, 496,
179 556, 516, 536, 520, 560, 496, 560, 496, 560, 512, 560, 496,
180 536, 520, 560, 496, 560, 516, 532, 520, 560, 496, 560, 496,
181 536, 540, 556, 1544, 560, 1544, 532, 1572, 532, 540, 532,
182 1572, 556, 1544, 536, 520, 560, 496, 536};
183
184 // IR σήμα για θέση θερμοκρασίας στο 27
185 uint16_t temp27[] = {8480, 4196, 576, 496, 560, 1548, 532, 524,
186 556, 1556, 540, 516, 540, 1564, 540, 1572, 528, 544, 532, 1564,
187 560, 1544, 560, 1540, 564, 508, 564, 1548, 532, 520, 536, 1564,
188 560, 1560, 560, 496, 536, 520, 536, 520, 536, 1588, 508, 548,
189 556, 1544, 560, 492, 564, 1560, 536, 516, 564, 492, 564, 492,
190 564, 508, 540, 516, 564, 496, 560, 492, 564, 512, 560, 492,
191 540, 516, 540, 516, 540, 536, 564, 1540, 560, 492, 564, 492,
192 564, 512, 560, 1544, 560, 492, 540, 516, 564, 508, 564, 496,
193 560, 492, 540, 516, 540, 536, 560, 496, 536, 520, 536, 520,
194 560, 512, 560, 496, 564, 492, 560, 496, 536, 536, 540, 516,
195 564, 492, 560, 496, 560, 512, 564, 492, 564, 492, 564, 492,
196 564, 508, 564, 492, 564, 492, 564, 492, 540, 1588, 532, 1568,
197 560, 1540, 560, 496, 536, 516, 540};
198
199 // IR σήμα για θέση θερμοκρασίας στο 28
200 uint16_t temp28[] = {8456, 4196, 580, 492, 564, 1540, 560, 496,
201 560, 1560, 560, 496, 560, 1544, 512, 1592, 556, 516, 560, 496,
202 560, 496, 560, 496, 560, 1560, 512, 1592, 508, 548, 556, 1544,
203 560, 1560, 560, 496, 560, 496, 560, 496, 560, 1560, 512, 544,
204 560, 1544, 560, 492, 564, 1560, 560, 496, 560, 492, 564, 492,
205 564, 512, 560, 496, 560, 496, 560, 496, 560, 512, 560, 496,
206 536, 520, 536, 520, 560, 512, 560, 1544, 560, 496, 560, 496,
207 560, 512, 560, 1544, 532, 524, 560, 492, 564, 512, 560, 496,
208 560, 496, 560, 496, 560, 512, 560, 496, 560, 496, 560, 496,
209 560, 512, 560, 496, 560, 496, 560, 496, 560, 512, 560, 496,
210 560, 496, 560, 496, 560, 512, 560, 496, 560, 496, 560, 496,
211 560, 516, 560, 1544, 508, 548, 556, 496, 560, 1560, 564, 1540,
212 560, 1544, 560, 496, 536, 516, 560};
213
214 // IR σήμα για θέση θερμοκρασίας στο 29
215 uint16_t temp29[] = {8452, 4220, 580, 496, 536, 1568, 556, 496,
216 560, 1564, 556, 500, 556, 1548, 532, 1568, 536, 536, 536, 1568,
217 560, 496, 560, 496, 560, 1564, 556, 1544, 556, 496, 560, 1544,
218 560, 1560, 560, 496, 536, 520, 560, 496, 560, 1560, 536, 520,
219 560, 1544, 532, 524, 556, 1564, 532, 524, 556, 500, 556, 500,
220 556, 516, 536, 520, 560, 496, 560, 496, 556, 516, 560, 496,
221 536, 520, 560, 496, 536, 536, 564, 1540, 536, 520, 560, 496,
222 556, 516, 536, 1568, 532, 524, 556, 500, 556, 516, 536, 520,
223 560, 496, 560, 496, 556, 516, 536, 520, 560, 496, 560, 496,
224 560, 512, 536, 524, 556, 496, 560, 496, 536, 540, 532, 524,
225 556, 496, 560, 496, 560, 516, 556, 500, 556, 500, 556, 500,

```

```

226 556, 516, 536, 520, 560, 1544, 556, 496, 560, 1564, 560, 1540,
227 560, 1548, 532, 520, 536, 520, 536};
228
229 //IR σήμα για να μπει σε λειτουργία ανεμηστήρα(φυσικός αέρας)
230 uint16_t fan[] = {8456, 4216, 584, 496, 560, 1540, 564, 492, 564,
231 1560, 508, 548, 560, 1540, 560, 1544, 560, 512, 536, 520, 536,
232 520, 560, 1544, 512, 560, 560, 1540, 564, 492, 564, 1540, 560,
233 1564, 508, 548, 556, 496, 560, 496, 560, 1560, 560, 496, 560,
234 1544, 560, 496, 560, 1560, 512, 544, 560, 496, 560, 492, 540,
235 536, 560, 496, 560, 496, 560, 496, 560, 512, 560, 496, 560, 496,
236 560, 496, 560, 512, 536, 520, 560, 496, 560, 496, 560, 512, 560,
237 1544, 560, 496, 560, 496, 536, 536, 560, 496, 560, 1544, 512,
238 544, 560, 512, 560, 496, 560, 496, 560, 496, 536, 536, 560, 496,
239 560, 496, 560, 496, 560, 512, 560, 496, 560, 496, 560, 496, 560,
240 512, 564, 492, 564, 492, 560, 496, 560, 512, 536, 524, 560,
241 1540, 560, 1544, 560, 512, 560, 1544, 512, 1592, 556, 496, 560,
242 496, 560};
243
244 //IR σήμα για να μπει σε λειτουργία υγρασίας
245 uint16_t dry[] = {8476, 4200, 576, 496, 536, 1568, 560, 496, 560,
246 1560, 560, 496, 560, 1544, 556, 1544, 560, 516, 556, 500, 556,
247 496, 536, 1568, 536, 540, 556, 1544, 560, 496, 560, 1544, 532,
248 1588, 508, 548, 560, 496, 560, 496, 556, 1564, 556, 500, 556,
249 1548, 532, 524, 532, 1584, 560, 496, 536, 520, 536, 520, 560,
250 512, 560, 496, 560, 500, 532, 520, 560, 516, 532, 524, 556, 500,
251 556, 496, 536, 540, 560, 1544, 532, 1572, 556, 496, 536, 540,
252 556, 1544, 560, 496, 536, 520, 560, 512, 560, 496, 560, 496,
253 560, 496, 536, 536, 560, 496, 536, 520, 560, 496, 560, 516, 556,
254 500, 556, 496, 560, 496, 536, 540, 556, 500, 560, 496, 532, 524,
255 556, 516, 556, 500, 560, 496, 556, 500, 560, 512, 560, 1544,
256 556, 1544, 560, 1548, 532, 540, 532, 1568, 560, 1544, 532, 524,
257 560, 496, 556};
258
259 //IR σήμα για να μπει σε λειτουργία κρύου
260 uint16_t cold[] = {8476, 4196, 580, 496, 560, 1544, 536, 520,
560,
261 1560, 560, 496, 536, 1568, 532, 1572, 556, 516, 556, 500, 556,
262 500, 556, 1544, 560, 516, 556, 1544, 560, 496, 560, 1544, 532,
263 1588, 536, 520, 560, 496, 560, 496, 556, 1564, 556, 500, 560,
264 1540, 536, 520, 536, 1588, 532, 524, 532, 520, 560, 496, 560,
516,
265 556, 500, 532, 520, 560, 500, 532, 540, 536, 520, 556, 500, 556,
266 500, 532, 540, 556, 500, 560, 1540, 536, 520, 560, 516, 556,
1548,
267 532, 524, 556, 496, 560, 516, 560, 496, 532, 524, 532, 520, 560,
268 516, 560, 496, 560, 496, 556, 500, 556, 516, 536, 520, 532, 524,
269 536, 520, 556, 516, 560, 496, 560, 496, 560, 496, 532, 540, 536,
270 520, 536, 520, 536, 520, 560, 516, 556, 496, 536, 1572, 528,
1572,
271 532, 540, 560, 1544, 532, 1568, 560, 496, 560, 496, 560};
272
273 //IR σήμα για να μπει σε λειτουργία ζέστης
274 uint16_t heat[] = {8456, 4220, 580, 496, 560, 1540, 560, 496,
560,
275 1564, 532, 520, 536, 1568, 560, 1544, 532, 540, 560, 496, 560,
276 496, 560, 1544, 560, 512, 560, 1544, 508, 548, 560, 1540, 536,
277 1584, 560, 496, 560, 496, 560, 496, 560, 1564, 508, 548, 556,
278 1544, 560, 496, 560, 1560, 560, 496, 556, 500, 560, 496, 560,
279 512, 560, 496, 560, 496, 536, 520, 556, 516, 536, 520, 536, 520,
280 560, 496, 560, 512, 560, 1544, 560, 496, 560, 496, 560, 512, 560,
281 1548, 508, 544, 560, 496, 560, 512, 560, 496, 560, 496, 560, 496,

```

```

282 560, 512, 560, 496, 560, 496, 560, 496, 560, 512, 560, 496, 560,
283 496, 560, 496, 560, 512, 560, 496, 560, 496, 536, 520, 560, 516,
284 560, 496, 556, 500, 560, 492, 560, 516, 560, 1544, 560, 492, 560,
285 1548, 532, 540, 556, 1544, 560, 1544, 560, 496, 560, 496, 560};
286
287 // Θέσε το port του web server στο 80
288 WiFiServer server(80);
289
290 // Μεταβλητή στην οποία αποθηκεύουμε το αίτημα HTTP
291 String header;
292
293 String AC_status = "off";
294
295 boolean heaton = false;
296 boolean coldon = false;
297 boolean fanon = false;
298 boolean dryon = false;
299
300 boolean actemp18 = false;
301 boolean actemp19 = false;
302 boolean actemp20 = false;
303 boolean actemp21 = false;
304 boolean actemp22 = false;
305 boolean actemp23 = false;
306 boolean actemp24 = false;
307 boolean actemp25 = false;
308 boolean actemp26 = false;
309 boolean actemp27 = false;
310 boolean actemp28 = false;
311 boolean actemp29 = false;
312
313 void setup() {
314     irsend.begin();
315     Serial.begin(115200);
316     delay(200);
317
318     // εκκίνηση του bme
319     bme.begin(0x76);
320
321     // Σύνδεση στο Wi-Fi
322     Serial.print("Connecting to ");
323     Serial.println(ssid);
324     WiFi.begin(ssid, password);
325     while (WiFi.status() != WL_CONNECTED) {
326         delay(500);
327         Serial.print(".");
328     }
329     // Εμφάνιση της local IP address και εκκίνηση του web server
330     Serial.println("");
331     Serial.println("WiFi connected.");
332     Serial.println("IP address: ");
333     Serial.println(WiFi.localIP());
334     server.begin();
335 }
336
337 void loop() {
338
339     //θερμοκρασία
340     float temp = bme.readTemperature();
341     //υγρασία
342     float humi = bme.readHumidity();

```

```

343
344 // server διαθέσιμος για είσοδο νέων χρηστών
345 WiFiClient client = server.available();
346
347 // Αν εισέλθει νέος χρήστης
348 if (client) {
349 // Εμφάνισέ το στο Serial Monitor
350 Serial.println("New Client.");
351 // Φτιάξε ένα String για να κρατάει
352 //τα εισερχόμενα δεδομένα από τον χρήστη
353 String currentLine = "";
354 // Όσο ο χρήστης είναι συνδεδεμένος -> loop
355 while (client.connected()) {
356 // Αν υπάρχουν bytes για να διαβάσει από τον χρήστη
357 if (client.available()) {
358 // Διάβασέ το , και μετά
359 char c = client.read();
360 // Εμφάνισέ το στο Serial Monitor
361 Serial.write(c);
362 header += c;
363 // Αν το byte είναι χαρακτήρας νέας γραμμής
364 if (c == '\n') {
365 // Αν η τρέχον γραμμή είναι κενή
366 // τότε έχεις δύο νέες γραμμές στη σειρά
367 // Αυτό είναι το τέλος του HTTP αιτήματος του χρήστη
368 //συνεπώς στείλε απάντηση:
369 if (currentLine.length() == 0) {
370 // Οι HTTP επικεφαλίδες πάντα ξεκινάνε με κωδικούς κατάστασης
371 //(για παράδειγμα HTTP/1.1 200 OK)
372 // Και μετά content-type ώστε ο χρήστης να γνωρίζει τι ακολουθεί
373 // και μετά κενή γραμμή:
374 client.println("HTTP/1.1 200 OK");
375 client.println("Content-type:text/html");
376 client.println("Connection: close");
377 client.println();
378
379 // Άνοιγμα Air Condition
380 if (header.indexOf("GET /14/on") >= 0) {
381 // στείλε σήμα για να ανοίξει το Air Condition
382 irsend.sendRaw(On, sizeof(On) / sizeof(On[0]), khz);
383 // εμφάνισε στο Serial Mode ότι το σήμα για το άνοιγμα
384 // του AC στάλθηκε
385 Serial.println("IR power On signal sent");
386 // κάνε την μεταβλητή AC_status 'on'
387 AC_status = "on";
388 }
389 // Κλείσιμο Air Condition
390 else if (header.indexOf("GET /14/off") >= 0) {
391 // στείλε σήμα για να κλείσει το Air Condition
392 irsend.sendRaw(Off, sizeof(Off) / sizeof(Off[0]), khz);
393 // εμφάνισε στο Serial Mode ότι το σήμα για το κλείσιμο
394 // του AC στάλθηκε
395 Serial.println("IR power Off signal sent");
396 // κάνε την μεταβλητή AC_status 'off'
397 AC_status = "off";
398 }
399 // Θερμοκρασία 18
400 else if (header.indexOf("GET /14/temp18") >= 0) {
401 στείλε σήμα για θερμοκρασία 18
402 irsend.sendRaw(temp18, sizeof(temp18) / sizeof(temp18[0]), khz);
403 //εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 18

```

```

404 //στάλθηκε
405 Serial.println("IR set temp to 18 signal sent");
406 // κάνε την μεταβλητή actemp18 αληθή και όλες τις
407 // υπόλοιπες ψευδείς
408 actemp18 = true;
409 actemp19 = false;
410 actemp20 = false;
411 actemp21 = false;
412 actemp22 = false;
413 actemp23 = false;
414 actemp24 = false;
415 actemp25 = false;
416 actemp26 = false;
417 actemp27 = false;
418 actemp28 = false;
419 actemp29 = false;
420 }
421 // Θερμοκρασία 19
422 else if (header.indexOf("GET /14/temp19") >= 0) {
423 // στείλε σήμα για θερμοκρασία 19
424 irsend.sendRaw(temp19, sizeof(temp19) / sizeof(temp19[0]), khz);
425 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 19
426 // στάλθηκε
427 Serial.println("IR set temp to 19 signal sent");
428 // κάνε την μεταβλητή actemp19 αληθή και όλες τις
429 // υπόλοιπες ψευδείς
430 actemp18 = false;
431 actemp19 = true;
432 actemp20 = false;
433 actemp21 = false;
434 actemp22 = false;
435 actemp23 = false;
436 actemp24 = false;
437 actemp25 = false;
438 actemp26 = false;
439 actemp27 = false;
440 actemp28 = false;
441 actemp29 = false;
442 }
443 // Θερμοκρασία 20
444 else if (header.indexOf("GET /14/temp20") >= 0) {
445 // στείλε σήμα για θερμοκρασία 20
446 irsend.sendRaw(temp20, sizeof(temp20) / sizeof(temp20[0]), khz);
447 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 20
448 // στάλθηκε
449 Serial.println("IR set temp to 20 signal sent");
450 // κάνε την μεταβλητή actemp20 αληθή και όλες τις
451 // υπόλοιπες ψευδείς
452 actemp18 = false;
453 actemp19 = false;
454 actemp20 = true;
455 actemp21 = false;
456 actemp22 = false;
457 actemp23 = false;
458 actemp24 = false;
459 actemp25 = false;
460 actemp26 = false;
461 actemp27 = false;
462 actemp28 = false;
463 actemp29 = false;
464 }

```



```

465 // Θερμοκρασία 21
466 else if (header.indexOf("GET /14/temp21") >= 0) {
467 // στείλε σήμα για θερμοκρασία 21
468 irsend.sendRaw(temp21, sizeof(temp21) / sizeof(temp21[0]), khz);
469 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 21
470 // στάλθηκε
471 Serial.println("IR set temp to 21 signal sent");
472 // κάνε την μεταβλητή actemp21 αληθή και όλες τις
473 // υπόλοιπες ψευδείς
474 actemp18 = false;
475 actemp19 = false;
476 actemp20 = false;
477 actemp21 = true;
478 actemp22 = false;
479 actemp23 = false;
480 actemp24 = false;
481 actemp25 = false;
482 actemp26 = false;
483 actemp27 = false;
484 actemp28 = false;
485 actemp29 = false;
486 }
487 // Θερμοκρασία 22
488 else if (header.indexOf("GET /14/temp22") >= 0) {
489 // στείλε σήμα για θερμοκρασία 22
490 irsend.sendRaw(temp22, sizeof(temp22) / sizeof(temp22[0]), khz);
491 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 22
492 // στάλθηκε
493 Serial.println("IR set temp to 22 signal sent");
494 // κάνε την μεταβλητή actemp22 αληθή και όλες τις
495 // υπόλοιπες ψευδείς
496 actemp18 = false;
497 actemp19 = false;
498 actemp20 = false;
499 actemp21 = false;
500 actemp22 = true;
501 actemp23 = false;
502 actemp24 = false;
503 actemp25 = false;
504 actemp26 = false;
505 actemp27 = false;
506 actemp28 = false;
507 actemp29 = false;
508 }
509 // Θερμοκρασία 23
510 else if (header.indexOf("GET /14/temp23") >= 0) {
511 // στείλε σήμα για θερμοκρασία 23
512 irsend.sendRaw(temp23, sizeof(temp23) / sizeof(temp23[0]), khz);
513 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 23
514 // στάλθηκε
515 Serial.println("IR set temp to 23 signal sent");
516 // κάνε την μεταβλητή actemp23 αληθή και όλες τις
517 // υπόλοιπες ψευδείς
518 actemp18 = false;
519 actemp19 = false;
520 actemp20 = false;
521 actemp21 = false;
522 actemp22 = false;
523 actemp23 = true;
524 actemp24 = false;
525 actemp25 = false;

```

```

526 actemp26 = false;
527 actemp27 = false;
528 actemp28 = false;
529 actemp29 = false;
530 }
531 // Θερμοκρασία 24
532 else if (header.indexOf("GET /14/temp24") >= 0) {
533 // στείλε σήμα για θερμοκρασία 24
534 irsend.sendRaw(temp24, sizeof(temp24) / sizeof(temp24[0]), khz);
535 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 24
536 // στάλθηκε
537 Serial.println("IR set temp to 24 signal sent");
538 // κάνε την μεταβλητή actemp24 αληθή και όλες τις
539 // υπόλοιπες ψευδείς
540 actemp18 = false;
541 actemp19 = false;
542 actemp20 = false;
543 actemp21 = false;
544 actemp22 = false;
545 actemp23 = false;
546 actemp24 = true;
547 actemp25 = false;
548 actemp26 = false;
549 actemp27 = false;
550 actemp28 = false;
551 actemp29 = false;
552 }
553 // Θερμοκρασία 25
554 else if (header.indexOf("GET /14/temp25") >= 0) {
555 // στείλε σήμα για θερμοκρασία 25
556 irsend.sendRaw(temp25, sizeof(temp25) / sizeof(temp25[0]), khz);
557 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 25
558 // στάλθηκε
559 Serial.println("IR set temp to 25 signal sent");
560 // κάνε την μεταβλητή actemp25 αληθή και όλες τις
561 // υπόλοιπες ψευδείς
562 actemp18 = false;
563 actemp19 = false;
564 actemp20 = false;
565 actemp21 = false;
566 actemp22 = false;
567 actemp23 = false;
568 actemp24 = false;
569 actemp25 = true;
570 actemp26 = false;
571 actemp27 = false;
572 actemp28 = false;
573 actemp29 = false;
574 }
575 // Θερμοκρασία 26
576 else if (header.indexOf("GET /14/temp26") >= 0) {
577 // στείλε σήμα για θερμοκρασία 26
578 irsend.sendRaw(temp26, sizeof(temp26) / sizeof(temp26[0]), khz);
579 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 26
580 // στάλθηκε
581 Serial.println("IR set temp to 26 signal sent");
582 // κάνε την μεταβλητή actemp26 αληθή και όλες τις
583 // υπόλοιπες ψευδείς
584 actemp18 = false;
585 actemp19 = false;
586 actemp20 = false;

```

```

587 actemp21 = false;
588 actemp22 = false;
589 actemp23 = false;
590 actemp24 = false;
591 actemp25 = false;
592 actemp26 = true;
593 actemp27 = false;
594 actemp28 = false;
595 actemp29 = false;
596 }
597 // Θερμοκρασία 27
598 else if (header.indexOf("GET /14/temp27") >= 0) {
599 // στείλε σήμα για θερμοκρασία 27
600 irsend.sendRaw(temp27, sizeof(temp27) / sizeof(temp27[0]), khz);
601 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 27
602 // στάλθηκε
603 Serial.println("IR set temp to 27 signal sent");
604 // κάνε την μεταβλητή actemp27 αληθή και όλες τις
605 // υπόλοιπες ψευδείς
606 actemp18 = false;
607 actemp19 = false;
608 actemp20 = false;
609 actemp21 = false;
610 actemp22 = false;
611 actemp23 = false;
612 actemp24 = false;
613 actemp25 = false;
614 actemp26 = false;
615 actemp27 = true;
616 actemp28 = false;
617 actemp29 = false;
618 }
619 // Θερμοκρασία 28
620 else if (header.indexOf("GET /14/temp28") >= 0) {
621 // στείλε σήμα για θερμοκρασία 28
622 irsend.sendRaw(temp28, sizeof(temp28) / sizeof(temp28[0]), khz);
623 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 28
624 // στάλθηκε
625 Serial.println("IR set temp to 28 signal sent");
626 // κάνε την μεταβλητή actemp28 αληθή και όλες τις
627 // υπόλοιπες ψευδείς
628 actemp18 = false;
629 actemp19 = false;
630 actemp20 = false;
631 actemp21 = false;
632 actemp22 = false;
633 actemp23 = false;
634 actemp24 = false;
635 actemp25 = false;
636 actemp26 = false;
637 actemp27 = false;
638 actemp28 = true;
639 actemp29 = false;
640 }
641 // Θερμοκρασία 29
642 else if (header.indexOf("GET /14/temp29") >= 0) {
643 // στείλε σήμα για θερμοκρασία 29
644 irsend.sendRaw(temp29, sizeof(temp29) / sizeof(temp29[0]), khz);
645 // εμφάνισε στο Serial Mode ότι το σήμα για θερμοκρασία 29
646 // στάλθηκε
647 Serial.println("IR set temp to 29 signal sent");

```

```

648 // κάνε την μεταβλητή actemp29 αληθή και όλες τις
649 // υπόλοιπες ψευδείς
650 actemp18 = false;
651 actemp19 = false;
652 actemp20 = false;
653 actemp21 = false;
654 actemp22 = false;
655 actemp23 = false;
656 actemp24 = false;
657 actemp25 = false;
658 actemp26 = false;
659 actemp27 = false;
660 actemp28 = false;
661 actemp29 = true;
662 }
663 // Επιλογή ζέστης
664 else if (header.indexOf("GET /14/heaton") >= 0) {
665 // στείλε σήμα για να μπει το AC σε λειτουργία ζέστης
666 irsend.sendRaw(heat, sizeof(heat) / sizeof(heat[0]), khz);
667 // εμφάνισε στο Serial Mode ότι το σήμα λειτουργίας ζέστης
668 // στάλθηκε
669 Serial.println("IR heat mode signal sent");
670 // κάνε την μεταβλητή heaton αληθή και όλες τις
671 // υπόλοιπες ψευδείς
672 heaton = true;
673 dryon = false;
674 coldon = false;
675 fanon = false;
676 }
677 // Επιλογή υγρασίας
678 else if (header.indexOf("GET /14/dryon") >= 0) {
679 // στείλε σήμα για να μπει το AC σε λειτουργία υγρασίας
680 irsend.sendRaw(dry, sizeof(dry) / sizeof(dry[0]), khz);
681 // εμφάνισε στο Serial Mode ότι το σήμα λειτουργίας υγρασίας
682 // στάλθηκε
683 Serial.println("IR dry mode signal sent");
684 // κάνε την μεταβλητή dryon αληθή και όλες τις
685 // υπόλοιπες ψευδείς
686 heaton = false;
687 dryon = true;
688 coldon = false;
689 fanon = false;
690 }
691 // Επιλογή κρύου
692 else if (header.indexOf("GET /14/coldon") >= 0) {
693 // στείλε σήμα για να μπει το AC σε λειτουργία κρύου
694 irsend.sendRaw(cold, sizeof(cold) / sizeof(cold[0]), khz);
695 // εμφάνισε στο Serial Mode ότι το σήμα λειτουργίας κρύου
696 // στάλθηκε
697 Serial.println("IR cold mode signal sent");
698 // κάνε την μεταβλητή coldon αληθή και όλες τις
699 // υπόλοιπες ψευδείς
700 heaton = false;
701 dryon = false;
702 coldon = true;
703 fanon = false;
704 }
705 // Επιλογή ανεμιστήρα
706 else if (header.indexOf("GET /14/fanon") >= 0) {
707 // στείλε σήμα για να μπει το AC σε λειτουργία ανεμιστήρα
708 irsend.sendRaw(fan, sizeof(fan) / sizeof(fan[0]), khz);

```

```

709 // εμφάνισε στο Serial Mode ότι το σήμα λειτουργίας ανεμιστήρα
710 // στάλθηκε
711 Serial.println("IR fan mode signal sent");
712 // κάνει την μεταβλητή fanon αληθή και όλες τις
713 // υπόλοιπες ψευδείς
714 heaton = false;
715 dryon = false;
716 coldon = false;
717 fanon = true;
718 }
719
720 // HTML
721 client.println("<!DOCTYPE html><html>");
722 client.println("<head><meta name=\"viewport\"
content=\"width=device-width, initial-scale=1\">");
723 client.println("<link rel=\"icon\" href=\"data:,\>");
724 // CSS
725 client.println("<style>html { font-family: Helvetica; display:
inline-block; margin: 0px auto; text-align: center;}");
726 client.println(".button { background-color: #195B6A; border:
none; color: white; padding: 16px 40px;}");
727 client.println("text-decoration: none; font-size: 30px; margin:
2px; cursor: pointer;}");
728 client.println(".button2 {background-color:
#77878A;}</style></head>");
729
730
731 // Web Page Heading
732 client.println("<body>");
733 client.println("<center>");
734 client.println("εδώ περιέχεται το url του icon που έχουμε βάλει
αλλά είναι υπερβολικά μεγάλο για το word");
735 //εμπεριέχεται ωστόσο κανονικά στον κώδικα Smart_Box_AC
736 // και στο uni icon.txt
737 client.println("</center>");
738 client.println("<h1>Smart Box AC Controller</h1>");
739
740 // Δείξε το τρέχον στάτους του Air Condition
741 client.println("<p>Air Condition " + AC_status + "</p>");
742 // Αν το Ac_status είναι off , δείχνει το ON κουμπί
743 if (AC_status == "off") {
744 client.println("<p><a href=\"/14/on\"><button
class=\"button\">ON</button></a></p>");
745 }
746 // Αλλιώς δείχνει το κουμπί OFF
747 else {
748 client.println("<p><a href=\"/14/off\"><button class=\"button
button2\">OFF</button></a></p>");
749 }
750
751 client.println("<hr/>");
752
753 // Δείχνει την θερμοκρασία και την υγρασία
754 client.println("<p>Temperature: ");
755 client.println("<span style=\"color:green\";>" + String(temp) +
"</span>");
756 client.println(" *C <br/></p>");
757 client.println("<p>Humidity: ");
758 client.println("<span style=\"color:green\";>" + String(humi) +
"</span>");
759 client.println(" % <br/></p>");

```

```

760 // Επιλογή για να κάνει ο χρήστης ανανέωση τις μετρήσεις
761 // θερμοκρασίας και υγρασίας
762 client.println("<p>Refresh: <a href=''>CHECK</a></p>");
763
764 client.println("<hr/>");
765
766 //Διάλεξε είδος λειτουργίας
767 client.println("<h2>AC Mode: </h2>");
768 client.print("<a href='\"/14/heaton\">Heat</a> | <a
href='\"/14/dryon\">Dry</a> | <a href='\"/14/coldon\">Cold</a> | <a
href='\"/14/fanon\">Fan</a>");
769 //Το τρέχον είδος λειτουργίας είναι:
770 client.println("<h3>Current mode: ");
771 if (heaton) {
772 client.print("<span
style='\"color:red\"';><strong>HEAT</strong></span>");
773 }
774 else if (dryon) {
775 client.print("<span
style='\"color:red\"';><strong>DRY</strong></span>");
776 }
777 else if (coldon) {
778 client.print("<span
style='\"color:red\"';><strong>COLD</strong></span>");
779 }
780 else if (fanon) {
781 client.print("<span
style='\"color:red\"';><strong>FAN</strong></span>");
782 }
783 else {
784 client.print("<span
style='\"color:red\"';><strong>Unkown</strong></span>");
785 }
786
787 client.println("<hr/>");
788
789 //Διάλεξε θερμοκρασία
790 client.println("<h2>Set temp: </h2>");
791 client.print("<a href='\"/14/temp18\">18</a> | <a
href='\"/14/temp19\">19</a> | <a href='\"/14/temp20\">20</a> | ");
792 client.print("<a href='\"/14/temp21\">21</a> | <a
href='\"/14/temp22\">22</a> | <a href='\"/14/temp23\">23</a> | ");
793 client.print("<a href='\"/14/temp24\">24</a> | <a
href='\"/14/temp25\">25</a> | <a href='\"/14/temp26\">26</a> | ");
794 client.print("<a href='\"/14/temp27\">27</a> | <a
href='\"/14/temp28\">28</a> | <a href='\"/14/temp29\">29</a> |");
795 //Η τρέχουσα θερμοκρασία είναι:
796 client.println("<h3>Current AC temp: ");
797 if (actemp18) {
798 client.print("<span
style='\"color:red\"';><strong>18</strong></span>");
799 }
800 else if (actemp19) {
801 client.print("<span
style='\"color:red\"';><strong>19</strong></span>");
802 }
803 else if (actemp20) {
804 client.print("<span
style='\"color:red\"';><strong>20</strong></span>");
805 }
806 else if (actemp21) {

```

```

807 client.print("<span
style=\"color:red\";><strong>21<strong></span>");
808 }
809 else if (actemp22) {
810 client.print("<span
style=\"color:red\";><strong>22<strong></span>");
811 }
812 else if (actemp23) {
813 client.print("<span
style=\"color:red\";><strong>23<strong></span>");
814 }
815 else if (actemp24) {
816 client.print("<span
style=\"color:red\";><strong>24<strong></span>");
817 }
818 else if (actemp25) {
819 client.print("<span
style=\"color:red\";><strong>25<strong></span>");
820 }
821 else if (actemp26) {
822 client.print("<span
style=\"color:red\";><strong>26<strong></span>");
823 }
824 else if (actemp27) {
825 client.print("<span
style=\"color:red\";><strong>27<strong></span>");
826 }
827 else if (actemp28) {
828 client.print("<span
style=\"color:red\";><strong>28<strong></span>");
829 }
830 else if (actemp29) {
831 client.print("<span
style=\"color:red\";><strong>29<strong></span>");
832 }
833 else {
834 client.print("<span
style=\"color:red\";><strong>Unknown<strong></span>");
835 }
836
837 client.println("</body></html>");
838
839 // Η HTTP απόκριση τελειώνει με μια κενή γραμμή
840 client.println();
841 // Break από την while loop
842 break;
843 }
844 // αν υπάρχει καινούρια γραμμή , καθάρισε την currentLine
845 else {
846 currentLine = "";
847 }
848 }
849 // αν υπάρχει κάτι άλλο εκτός από χαρακτήρα επιστροφής μεταφοράς
850 else if (c != '\r') {
851 // πρόσθεσέ το στο τέλος του currentLine
852 currentLine += c;
853 }
854 }
855 }
856 // Καθάρισε την μεταβλητή header
857 header = "";

```

```
858 // Κλείσε την σύνδεση
859 client.stop();
860 Serial.println("Client disconnected.");
861 Serial.println("");
862 }
863
864 }
865
```


Παράρτημα Β: Ανάλυση Κόστους

Arduino Mega 2560 (ebay)	X1	8.83€
Arduino Uno (ebay)	X1	5.47€
Esp8266 Nodemcu (cableworks)	X1	8.99€
Bme280 (cableworks)	X1	8.49€
Breadboard (cableworks)	X1	4.50€
Tsop1738 (ebay)	X2	0.89€
IR LED (ebay)	X10	0.62€
Electric Soldering Iron Welding tool (ebay)	X1	15.17€
Wires (ebay)	X120	3.01€
2N2222A NPN TRANSISTOR (ebay)	X10	2.47€
Resistor (ebay)	X50	0.88€
Κουτί (custom)	X1	10€
Σύνολο		69.32€

7) Συνολικό Κόστος Εξαρτημάτων