



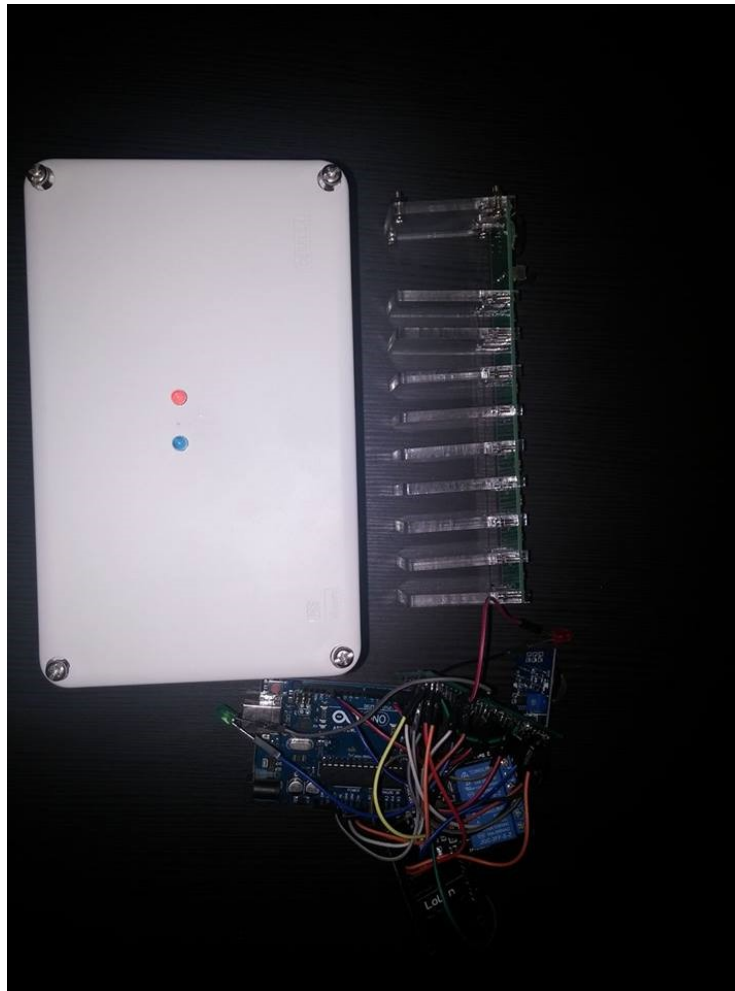
**Τμήμα Μηχανικών
Αυτοματισμού**



**Αλεξάνδρειο Τεχνολογικό
Εκπαιδευτικό Ίδρυμα
Θεσσαλονίκης**

**Απομακρυσμένος Έλεγχος φωτορυθμικού συστήματος και
εκπομπών αερίων χώρων διασκέδασης**

**Remote Control of Audio Spectrum Analyser with IoT and Android
Application**



Φοιτητής: Νικόλαος Ζαφειρέλλης

Εισηγητής: Σημίρα Παπαδοπούλου

Περιεχόμενα

.....	5
Πρόλογος:.....	5
ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ.....	6
1.1 Internet of things (IoT).....	6
1.2 Το ESP8266.....	8
1.3 Ο επεξεργαστής Arduino.....	9
1.4 Υλικός εξοπλισμός (Hardware).....	10
1.5 Λογισμικό (Software).....	12
1.6 Πλεονεκτήματα χρήσης Arduino.....	16
1.7 Δίοδος φωτοεκπομπής (LED).....	20
1.7.1 Βασική αρχή λειτουργίας διόδου φωτοεκπομπής.....	20
1.7.2 Διάρκεια ζωής των LED.....	21
1.8 Φωτορυθμικό.....	21
1.9 ΑΙΣΘΗΤΗΡΕΣ.....	22
1.9.1 Είδη και χαρακτηριστικά αισθητήρων.....	22
ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ.....	23
2.1 Ενισχυτής.....	23
2.2 Κατηγορίες Ενισχυτών.....	24
2.2.1 Ενισχυτική Τάξη "A".....	25
2.2.2 Ενισχυτική τάξη "B" και "AB".....	25
2.2.3 Ενισχυτική τάξη "C".....	26
2.3.1 Ανιχνευτής καπνού.....	26
2.3.2 Αυτόματο σύστημα συναγερμού.....	27
2.3.3 Αισθητήρια καπνού για Arduino.....	29
2.4 Σύνδεση με τον έξω κόσμο.....	30
2.5 Ο Server ThingSpeak.....	31
ΚΕΦΑΛΑΙΟ 3.....	35
3.1 Κύκλωμα προενίσχυσης.....	35

3.2 Προενίσχυση.....	36
3.3 Διάταξη των LED.....	38
3.4 Ανάλυση Κώδικα Συστήματος.....	39
3.5 NodeMcu.....	39
3.6 Arduino και ESP-01.....	40
3.7 Γενικότερη δομή του Κώδικα.....	40
3.8 Διαγράμματα Ροής.....	41
ΚΕΦΑΛΑΙΟ 4.....	45
4.1 Περιγραφή της Κατασκευής.....	45
4.1.1 Φωτορυθμικό.....	47
4.1.2 Χρήση του Arduino UnoR3.....	49
4.1.3 Αισθητήριο MQ-2.....	50
4.1.4 Ολοκληρωμένο ESP8266.....	50
4.1.5 Ολοκληρωμένο NodeMCU.....	51
4.1.6 Two Relay Module.....	52
4.2 Προγραμματισμός.....	53
4.2.1 Κώδικας Arduino Uno R3.....	53
4.2.2 Κωδικας NodeMcu.....	64
4.3 Κόστος Κατασκευής.....	69

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ κατ' αρχήν τους καθηγητές του τμήματος Μηχανικών Αυτοματισμού για την ορθή κατάρτιση που μου παρείχαν στις σπουδές μου. Επίσης θελω να ευχαριστήσω την κ. Σημίρα Παπαδοπούλου για την συνεχή υποστήριξη της καθώς και την δυνατότητα που μου έδωσε να παρουσιάσω αυτήν την εργασία.

Πρόλογος:

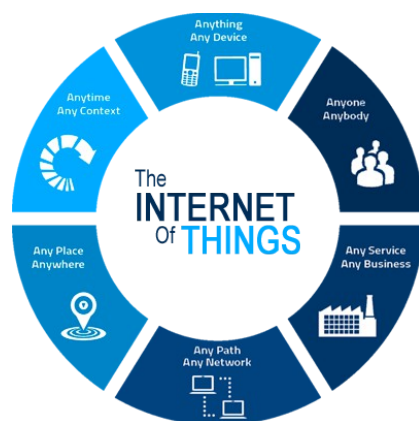
Τα τελευταία χρόνια αυξάνεται ραγδαία η χρήση διαφόρων μικροελεγκτών χαμηλού κόστους με μεγάλες όμως δυνατότητες. Ανάμεσα σ' αυτούς ξεχωρίζει ο μικροελεγκτής NODE MCU 8266, το κόστος του οποίου μειώνεται αντιστρόφως ανάλογα της χρήσης και αποδοχής που τυγχάνει από χιλιάδες χρήστες. Αν και οι εφαρμογές που ενσωματώνουν τον NODE MCU 8266 πληθαίνουν ραγδαία, οι περισσότερες απ' αυτές ανήκουν στον χώρο της Ρομποτικής και Αυτοματισμού, της Πληροφορικής και των Ηλεκτρονικών. Στο πρώτο κεφάλαιο της παρούσα εργασία θα περιγράψουμε αναλυτικά και την μητρική πλακέτα της κατασκευής που είναι ένας Arduino Uno: ιστορική αναδρομή, το ολοκληρωμένο ATmega328 και την μνήμη που χρησιμοποιεί, τις εισόδους/εξόδους του, την τροφοδοσία του, τα ενσωματωμένα πάνω στην πλακέτα κουμπιά και led, το ολοκληρωμένο περιβάλλον ανάπτυξης του (IDE) και την σύνδεση με τον υπολογιστή, την γλώσσα προγραμματισμού που χρησιμοποιεί προκειμένου να δημιουργήσουμε τα προγράμματά μας, διάφορα παραδείγματα και υλοποιήσεις προγραμμάτων για την κατανόηση της εύκολης και ευέλικτης χρήσης του Arduino.

Πιο συγκεκριμένα, η εργασία αυτή εστιάζει την προσοχή της στους αναλογικούς αισθητήρες, με τους οποίους αισθητοποιούνται δεδομένα του φυσικού περιβάλλοντος, στο σύστημα συλλογής και επεξεργασίας των δεδομένων με κατάλληλα ηλεκτρονικά κυκλώματα και τέλος στο τρόπο παρουσίασης των αποτελεσμάτων σε smartphones και Η/Υ μέσω IoT (σέρβερ). Στόχος της εργασίας είναι να αναδείξει τις δυνατότητες του απομακρυσμένου ελέγχου φωτορυθμικού και τα πλεονεκτήματα χρήσης των μικροελεγκτών λαμβάνοντας μετρήσεις από τα αισθητήρια και απεικονίζοντας τα σε smartphone συσκευή και IoT σέρβερ.

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

1.1 Internet of things (IoT)

Μια «έξυπνη συσκευή» είναι μια συσκευή, η οποία γενικά συνδέεται με άλλες συσκευές ή δίκτυα μέσω διαφόρων ασύρματων πρωτόκολλων όπως το bluetooth, wifi, 3G κλπ τα οποία λειτουργούν σε ένα βαθμό αυτόνομα. Μερικά γνωστά σε όλους παραδείγματα «έξυπνων» συσκευών, είναι τα smartphones τα tablet, οι «έξυπνες» τηλεοράσεις, τα smart watches (<http://www.mechanicalsolutions.gr>). Οι συσκευές αυτές μπορούν να λειτουργούν σε κάποιο βαθμό αυτόνομα, δηλαδή χωρίς ανθρώπινη παρέμβαση. Μπορούν να χειριστούν ένα πλήθος δυναμικών δράσεων και αλληλεπιδράσεων, ενώ διέπονται από έξυπνη λήψη αποφάσεων και οργανωτική αλληλεπίδραση. Αυτό μπορεί σε κάποιες περιπτώσεις να συνεπάγεται κάποια μορφή τεχνητής νοημοσύνης, προκειμένου να υπάρξει ομαλή σύνδεση και ένα είδος κατανομής εργασιών με τις άλλες συσκευές. Ωστόσο είναι δύσκολο να καθοριστεί ένα κλειστό σύνολο ιδιοτήτων που καθορίζει όλες τις υπολογιστικές συσκευές λόγω του μεγάλου εύρους και της ποικιλίας των ερευνών της εφαρμοσμένης πληροφορικής. Το internet of things (IoT) επιτρέπει σε αντικείμενα να αισθάνονται και να ελέγχονται εξ αποστάσεως σε όλη την υπάρχουσα υποδομή του δικτύου. Δημιουργεί προϋποθέσεις για πιο άμεση ενσωμάτωση του φυσικού κόσμου σε συστήματα που βασίζονται στον υπολογιστή. Αποτέλεσμα ,η βελτίωση της αποτελεσματικότητας, η ακρίβεια αλλά και το οικονομικό όφελος . Κάθε έξυπνη συσκευή συνεργάζεται και είναι μοναδικά αναγνωρίσιμη σε ένα ή πολλά συγκεκριμένα δίκτυα, στα οποία είναι ενσωματωμένη. Ο όρος Internet of things επινοήθηκε το 1999 από τον Kevin Ashton. Συνήθως το Internet of things, προσφέρει προηγμένη συνδεσιμότητα συσκευών των συστημάτων και των υπηρεσιών και προχωρά πέρα από το μηχάνημα, καθώς καλύπτει μία ποικιλία πολλών τομέων και εφαρμογών. Παράλληλα οι εφαρμογές παρακολούθησης περιβάλλοντος των Internet of things μπορούν να χρησιμοποιούν αισθητήρες για την καλύτερη αντίληψη του περιβάλλοντος. Η διασύνδεση αυτών των ενσωματωμένων συσκευών, συμπεριλαμβανομένων των έξυπνων αντικειμένων, αναμένεται να εισάγει την αυτοματοποίηση σε όλους σχεδόν τους τομείς. Με την χρήση αισθητήρων μπορούν να συνεισφέρουν στην προστασία του περιβάλλοντος. . Το «ιντερνέτ των πραγμάτων» μπορεί να βοηθήσει στην ενσωμάτωση των επικοινωνιών ,του ελέγχου και της επεξεργασίας πληροφοριών σε διάφορα συστήματα μεταφοράς. Άλλωστε ήδη εφαρμογές του IoT εκτείνονται σε όλες τις πτυχές των συστημάτων μεταφορών. Σαν συμπέρασμα λοιπόν μπορούμε να εξάγουμε ότι το IoT επρόκειτο στο άμεσο μέλλον να εισχωρήσει ολοκληρωτικά στην καθημερινότητα μας και στην κοινωνία, κάνοντας πιο εύκολη, ευχάριστη και ασφαλή τη ζωή μας αλλά και το περιβάλλον μας.



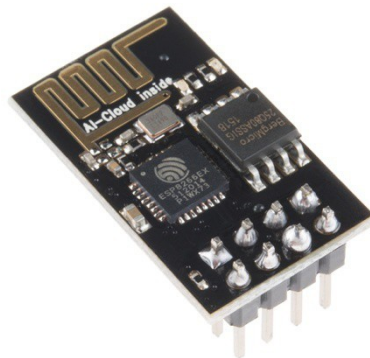
Εικόνα 1: Εφαρμογές του Internet Of Things ή «Διαδίκτυο των Πραγμάτων»(BetaNews)

Το Internet Of Things ή «Διαδίκτυο των Πραγμάτων» είναι η επερχόμενη εξέλιξη του Διαδικτύου των υπηρεσιών, που υπάρχει σήμερα. Πρόκειται για ένα δίκτυο όχι μόνο υπολογιστών αλλά και διασυνδεδεμένων αντικειμένων. Τα αντικείμενα αυτά θα περιέχουν ενσωματωμένα ηλεκτρονικά συστήματα και μπορούν να είναι διάφορες οικιακές συσκευές, μέσα μεταφοράς, μέσα τηλεπικοινωνίας, βιβλία, αυτοκίνητα, ακόμα και τρόφιμα. Πέρα από την εξασφάλιση της καλής λειτουργίας των διασυνδεδεμένων αυτών αντικειμένων, θα γίνει προσπάθεια να επιτευχθεί και συνεργασία μεταξύ των συστημάτων αυτών. Κάθε αντικείμενο θα χρησιμοποιεί συστήματα αναγνώρισης ραδιοσυχνοτήτων (τα γνωστά ως RFID), δηλαδή ένα είδος αισθητήρων. Απαραίτητη προϋπόθεση για την επιτυχία του καινούριου αυτού Διαδικτύου είναι να καταστεί το σημερινό Διαδίκτυο πιο ασφαλές. Το Διαδίκτυο των Πραγμάτων θα είναι η κορύφωση της προσπάθειας για την ολοκλήρωση και αυτοματοποίηση των υπηρεσιών που παρέχουν τα ενσωματωμένα συστήματα παντός είδους. Το διαδίκτυο θα γίνει διαδραστικό, ένα τεράστιο ιεραρχικά οργανωμένο «νευρικό σύστημα» που θα απολήγει σε συσκευές με αισθητήρες και ενεργοποιητές (actuators) που θα συνεργάζονται για έξυπνες υπηρεσίες για την υγεία, τις μεταφορές, τη διανομή και κατανάλωση ενέργειας κλπ. Στις μεταφορές σύντομα θα έχουμε συστήματα αυτόματης οδήγησης και οργάνωσης των μεταφορικών μέσων για περισσότερη ασφάλεια και οικονομία. Στον τομέα της υγείας προβλέπονται μία σειρά από καινοτομίες, από τη διαδραστική τηλεπαρακολούθηση των ασθενών, μέχρι την τηλεχειρουργική και τα έξυπνα φάρμακα. Η μεγάλη πρόκληση είναι η αυτοματοποίηση της διαχείρισης πόρων όπως για παράδειγμα σε αυτό που ονομάζεται smart grids, συνδυασμένη και αποτελεσματική χρήση εναλλακτικών μορφών ενέργειας. Όλες αυτές θα είναι μερικές εφαρμογές που θα αλλάξουν ριζικά το σημερινό τρόπο ζωής τις επρχόμενες δεκαετίες. Πρέπει να αναφερθεί ωστόσο ότι ακόμα η κατάσταση του διαδικτύου παραμένει ιδιαίτερα επισφαλής και χρειάζεται μεγάλη δράση και κινητοποίηση για να μπορέσουν όλα αυτά να λειτουργήσουν με ασφάλεια και αποτελεσματικότητα. Ο συνδυασμός του internet, των αντικειμένων και κινητών υπηρεσιών, ανοίγει το δρόμο σε αυτό που λέμε «διάχυτη νοημοσύνη». Πανταχού παρούσα και απρόσκοπτη πρόσβαση σε παντοειδείς υπηρεσίες, αποτελεσματικός έλεγχος πόρων, διαδραστικότητα και συνέργεια για την επίτευξη ολοκληρωμένων στόχων. Μεγάλες εταιρίες δεν αποσκοπούν πλέον στη μεμονωμένη πώληση λογισμικού ή υπολογιστών αλλά μελετούν ολοκληρωμένες λύσεις όπου τα 7 πληροφορικά συστήματα χρησιμοποιούνται για τη βέλτιστη

διαχείριση φυσικών πόρων και την ανάπτυξη έξυπνων υπηρεσιών για τη δημιουργία ενός «έξυπνου πλανήτη»

1.2 Το ESP8266

Το ESP8266 είναι ένα χαμηλού κόστους Wi-Fi chip με ενσωματωμένα πρωτοκολλά TCP/IP το οποίο μπορεί να δώσει πρόσβαση στο wifi σε οποιοδήποτε μικροελεγκτή. Το esp μπορεί να τρέξει το ίδιο μια εφαρμογή ή να φορτώσει μια, από άλλη συσκευή. Κατασκευάζεται στη Σαγκάη από την εταιρία Espressif. Πρόκειται για ένα αρκετά σύγχρονο τσιπ, το οποίο εμφανίστηκε στη δυτική αγορά τον Αύγουστο του 2014, με το μοντέλο ESP-01 από την Ai-Thinker. Είναι η πρώτη οικονομική μονάδα που επιτρέπει τη σύνδεση στο wifi αν και αρχικά δεν υποστήριζε την αγγλική γλώσσα. Η χαμηλή τιμή του και το μικρό μέγεθος του προσέλκυσε το ενδιαφέρον hacker , οι οποίοι τελικά το μετέφρασαν από την κινεζική στην αγγλική γλώσσα. Έκτοτε, ολοένα και περισσότεροι προγραμματιστές, το χρησιμοποιούν σε ποικίλες εφαρμογές. Η ικανότητα για γρήγορη επεξεργασία και ο μεγάλος αποθηκευτικός χώρος (1MB), επιτρέπουν την διασύνδεση με πολλούς αισθητήρες και άλλες συσκευές εφαρμογών (πχ arduino), μέσω των GPIOs, με ελάχιστο χρόνο φόρτωσης κατά τη διάρκεια λειτουργίας. Παράλληλα, το κύκλωμα του είναι σχεδιασμένο, ώστε να καταλαμβάνει τον ελάχιστο δυνατό χώρο πάνω σε μια πλακέτα PCB. (gmelectronic.com)



Εικόνα 2: Πλακέτα ESP8266(Wikipedia)

Υπάρχει πληθώρα πληροφοριών τόσο για τις δυνατότητες και τον τρόπο διασύνδεσης του με άλλες συσκευές και αισθητήρες, όσο και για τον προγραμματισμό του, οι οποίες παρέχονται κυρίως από την κοινότητα ανθρώπων που το χρησιμοποιούν. Στην κοινότητα μπορεί κανείς να διαβάσει πως μπορεί το esp να μετατραπεί σε μια IOT(Internet of Things) συσκευή.

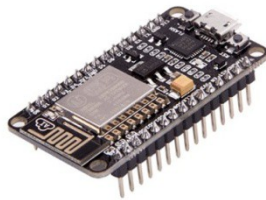
Τέλη του 2013, η ίδια εταιρία (Espressif) κατασκευάζει ένα software development kit (SDK), το οποίο επιτρέπει τον προγραμματισμό του esp χωρίς τη χρήση microcontroller. Από τότε, πολλές εταιρίες άρχισαν να διανέμουν τέτοια kit βασισμένα στο esp. Ένα από τα διασημότερα, οικονομικότερα και αποτελεσματικότερα SDK στην αγορά είναι το NodeMcu development board με firmware nodemcu. Το nodemcu development kit board based on esp8266, το οποίο θα χρησιμοποιήσουμε για την εργασία αυτή, διαθέτει θύρα micro usb για απευθείας σύνδεση με τον υπολογιστή, χωρίς τη χρήση μικροελεγκτή.

Κάποια πλεονεκτήματα που προσφέρει σε σχέση με άλλους μικροελεγκτές είναι τα παρακάτω:

-Κόστος: Οι πλακέτες του nodemcu είναι εξαιρετικά φθηνές σε σχέση με άλλες πλατφόρμες μικροελεγκτών. Το κόστος του μικροελεγκτή ανέρχεται στα 10 ευρώ.

- **Ευκολία στο προγραμματιστικό περιβάλλον:** Το περιβάλλον προγραμματισμού ενός nodemcu ενδείκνυται για αρχάριους, αλλά είναι ταυτόχρονα και ευέλικτο και για πιο προχωρημένους χρήστες.

- **Ανοιχτού λογισμικού και λογισμικού που επεκτείνεται και παραμετροποιείται:** Το software του nodemcu διανέμεται με την μορφή εργαλείων ανοιχτού λογισμικού και είναι διαθέσιμο προς επέκταση για έμπειρους προγραμματιστές. Η γλώσσα προγραμματισμού του μπορεί να επεκταθεί διαμέσου των βιβλιοθηκών την C++.



Εικόνα 3: NodeMcu Esp8266(Wikipedia)

1.3 Ο επεξεργαστής Arduino

Το 2005 τέθηκε σε λειτουργία ένα πλάνο προκειμένου να κατασκευαστεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από φοιτητές και μαθητές, το οποίο θα ήταν πιο οικονομικό από άλλα πρωτότυπα συστήματα, διαθέσιμα εκείνη την χρονική περίοδο. Το σχέδιο αυτό ονομάστηκε *Arduin* της *Ivrea* από τους δημιουργούς του Massimo Banzi και David Cueartielles. Έτσι, άρχισαν να κατασκευάζουν πλακέτες σε ένα μικρό εργοστάσιο στην βορειοδυτική, πιο συγκεκριμένα στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο στον ίδιο τόπο που στεγαζόταν και η επιχείρηση υπολογιστών Olivetti.

Όπως αναφέρουν οι ιδρυτές του, το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών, βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software. Απευθύνεται σε όλους όσους κατέχουν μια μικρή προγραμματιστική εμπειρία, βασικές γνώσεις ηλεκτρονικών και την θέληση να δημιουργήσουν διαδραστικά αντικείμενα ή περιβάλλοντα.

Ουσιαστικά, πρόκειται για ένα ηλεκτρονικό κύκλωμα που φέρει τον μικροελεγκτή ATmega της εταιρείας Atmel και του οποίου διακινούνται ελεύθερα και δωρεάν όλα τα σχέδια, καθώς και το software που απαιτείται για την λειτουργία του. Έτσι, μπορεί να κατασκευαστεί από οποιονδήποτε και αυτός είναι και ο λόγος για τον οποίο περιγράφετε με τον περίεργο – για hardware- χαρακτηρισμό: «ανοικτού κώδικα». Ο Arduino, είναι ικανός να λειτουργήσει σαν ένας

μικροσκοπικός υπολογιστής, επειδή ο χρήστης έχει την ικανότητα να συνδέσει πάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή ώστε να δέχεται πληροφορίες από τις μονάδες εισόδου, να τις επεξεργάζεται και να δίνει τις κατάλληλες εντολές στις μονάδες εξόδου. Θα μπορούσε μάλιστα κάποιος να ισχυριστεί – και θα ήταν μια παρά πολλοί πετυχημένη δήλωση – ότι λειτουργικά το Arduino έχει πολλές ομοιότητες με το NXT Brick των Lego Mindstorms NXT. Εξάλλου, μια από τις πολλές εφαρμογές στις οποίες το Arduino διαπρέπει είναι η ρομποτική. Το Arduino διαπιστωμένα, δεν είναι ο μοναδικός, άλλα ούτε και ο καλύτερος δυνατός τρόπος για την υλοποίηση μιας οποιασδήποτε διαδραστικής ηλεκτρονικής συσκευής. Το κυριότερο πλεονέκτημά του είναι η μεγάλη κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια αντίστοιχη σε μέγεθος online γνωσιακή βάση. Αυτός είναι τελικά και ο παράγοντας ο οποίος οδήγησε στην ευρεία χρήση του Arduino διότι, δεν χρειάζονται πολλές περισσότερες (ηλεκτρονικές) γνώσεις πέρα από τις λίγες που έχει μάθει ο καθένας μας στο σχολείο.

Το Arduino είναι μικροελεγκτής μονής πλακέτας. Αυτό σημαίνει ότι αποτελείται από μια απλή μητρική πλακέτα ανοικτού κώδικα και έχει έναν ενσωματωμένο μικροελεγκτή, καθώς και εισόδους/εξόδους. Η πλακέτα αυτή μπορεί να προγραμματιστεί με τη γλώσσα Wiring η οποία αποτελείται από την πολύ γνωστή γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες της. Έτσι μας δίνεται η δυνατότητα να αναπτύξουμε ανεξάρτητα διαδραστικά αντικείμενα, αλλά και να συνδέσουμε το Arduino σε υπολογιστή, με την βοήθεια ποικίλων λογισμικών, όπως Processing, Max/MSP, PureData, SuperCollider. Κυκλοφορούν προ-συναρμολογημένες εκδόσεις του Arduino (έτοιμα συστήματα), ενώ για τους περισσότερο τολμηρούς χρήστες είναι διαθέσιμα διαγράμματα και οδηγίες συναρμολόγησης. (grhotels.gr)

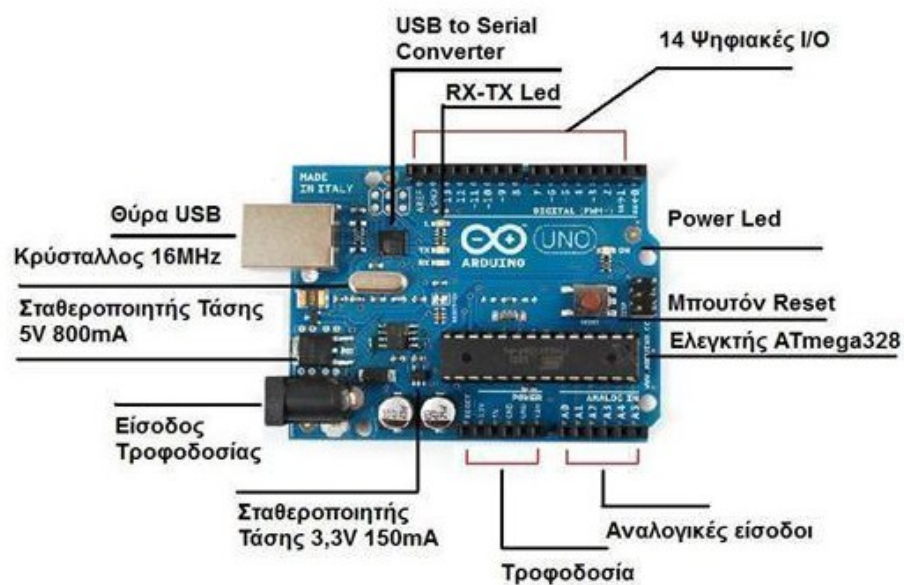
1.4 Υλικός εξοπλισμός (Hardware)

Όσο αφορά την υλική του δομή, το σύστημα Arduino περιλαμβάνει έναν γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz, ενώ κάποιες παραλλαγές έχουν κεραμικό αντηχητή. Για να μην απαιτείται εξωτερικός προγραμματιστής ο μικροελεγκτής φέρει ενσωματωμένο έναν φορτωτή εκκίνησης (bootloader).

Στην ακολουθία λογισμικού (software stack) που χρησιμοποιείται στις πλατφόρμες Arduino όλες οι πλάκες (boards) προγραμματίζονται με μία RS-232 σειριακή σύνδεση αλλά για κάθε διαφορετική εκδοχή υλικού εξοπλισμού η σύνδεση αυτή διαφέρει. Όσον αφορά τις σειριακές πλάκες, αυτές περιέχουν ένα απλό κύκλωμα μετατόπισης στάθμης (level shifter) που μετατρέπει το σήμα τάσης από RS-232 σε TTL, και το αντίστροφο. Πλέον τα Arduino προγραμματίζονται μέσω USB χρησιμοποιώντας εφαρμογές προσαρμογής chip USB-to-Serial, όπως το FTDI232. Όσον αφορά το

ανεπίσημο Boarduino ή το Arduino mini εφαρμόζονται άλλες μέθοδοι αλλά και ένα αφαιρούμενο USB-to-Serial καλώδιο ή πλάκα, ή ακόμα και σύνδεση με Bluetooth.

Σε σχέση με άλλα κυκλώματα, οι πίνακες Arduino περιέχουν περισσότερα microcontroller I/O pins, με διάφορες εφαρμογές plug-in ασπιδών να είναι διαθέσιμες. Τα Arduino Diecimila, Duemilanova και Uno έχουν 14 ψηφιακά I/O pins, έξι εκ των οποίων παράγουν pulse-width διαμορφωμένα σήματα, και έξι αναλογικά δεδομένα. Αυτά τα pins μπορούμε να τα βρούμε στην κορυφή του πίνακα με femaleheaders 0.1 ιντσών (2,2mm). Για την σύνδεση σε [Bread boards](#) τα Arduino nano, Arduino-CompatibleBareBonesBoard και Boarduino Board, παρέχουν κάποιες φορές maleheader pins.

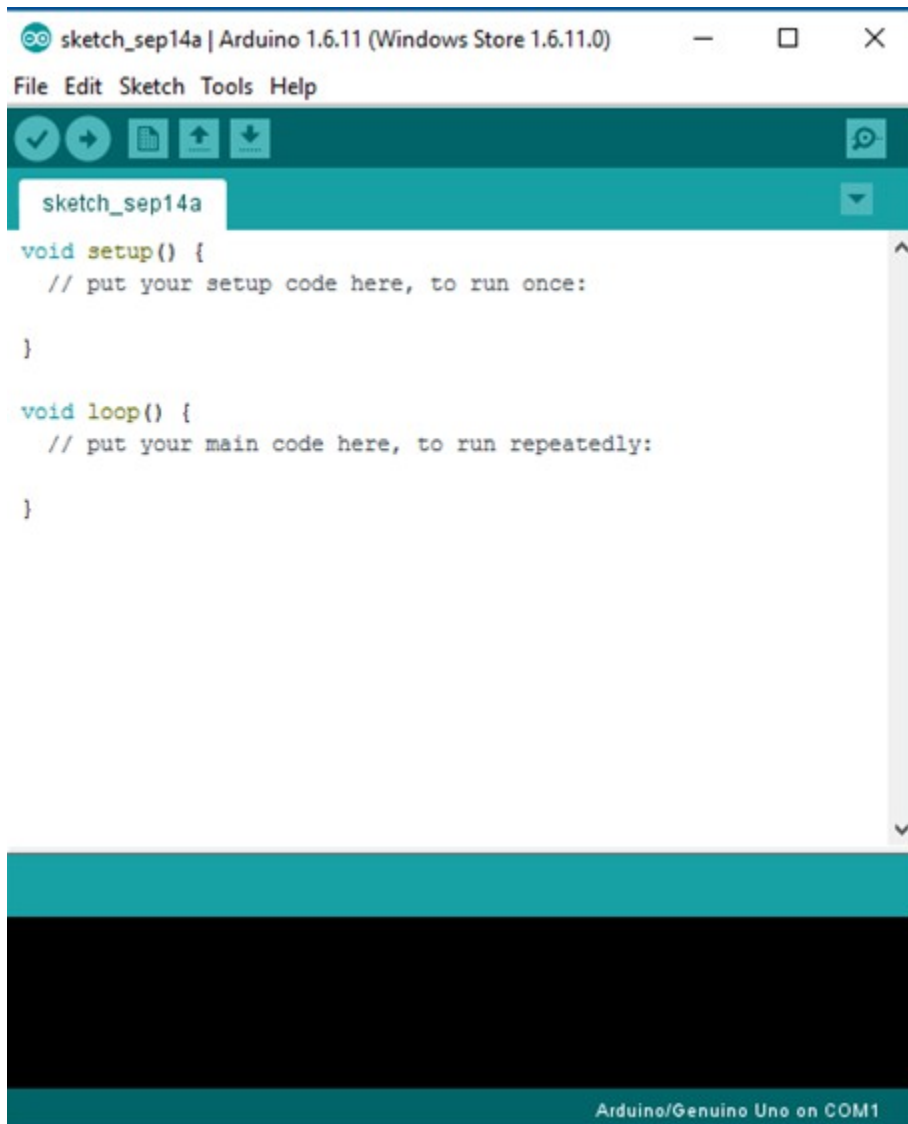


Εικόνα 4: Άποψη της πλακέτας Arduino.(Electronify)

Εναλλακτικά του Arduino, υπάρχουν πολλοί πίνακες (boards) που είναι είτε συμβατά είτε προερχόμενα από αυτό. Η συνηθέστερη μετατροπή των Arduino είναι η προσθήκη καινοτόμων οδηγών εξόδων (output drivers) στα πλαίσια χρήσης τους στην εκπαίδευση, ώστε να απλοποιηθούν και να γίνουν περισσότερο προσιτές στους μαθητές κατασκευές buggies ή μικρών ρομπότ. Κάποιες από αυτές τις μετατροπές είναι ισάξιες αλλάζοντας μόνο τον παράγοντα μορφής κι έτσι επιτρέποντας την συνεχόμενη χρήση ασπιδών (shields). Γενικά υπάρχουν πολλές παραλλαγές που χρησιμοποιούν πληθώρα επεξεργαστών και ποικίλα επίπεδα συμβατότητας.

1.5 Λογισμικό (Software)

Τα Arduino χρησιμοποιούν ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE) που έχει γραφτεί σε Java και είναι έτσι σχεδιασμένο ώστε να εισαγάγει στον προγραμματισμό άτομα που δεν είναι εξοικειωμένοι με το αντικείμενο όπως νέοι μαθητές ή καλλιτέχνες. Το IDE αυτό λειτουργεί σε πολλές πλατφόρμες και προέρχεται από τον προγραμματισμό με την γλώσσα Processing πάνω στο περιβάλλον Wiring. Αποτελείται από ένα τύπο επεξεργασίας κώδικα που τον χαρακτηρίζει η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με το πάτημα ενός κουμπιού. Τα προγράμματα ή ο ίδιος ο κώδικας που γράφτηκε για συστήματα Arduino ονομάζονται σκίτσο (sketch) ενώ στις περισσότερες περιπτώσεις δεν υπάρχει ανάγκη επεξεργασίας ή χρήσης περιβάλλοντος γραμμής εντολών.



Εικόνα 5: Ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Enviroment - IDE)

Το Arduino IDE τρέχει προγράμματα που είναι συνήθως γραμμένα σε C ή C++, αλλά με ειδικούς κανόνες δομής. Μαζί με το Arduino IDE έχουμε και μια βιβλιοθήκη λογισμικού που προέρχεται από το πρότζεκτ Wiring, για αυτό και έχουν πολλές κοινές διαδικασίες εισόδου/εξόδου (input/output).

Οι χρήστες πρέπει να ορίσουν μόνο τις εξής δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

-setup(): μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος και αρχικοποιεί τις ρυθμίσεις.

-loop(): μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί.

Υπάρχουν, κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που αποσκοπούν στην απλούστερη διαχείριση του ειδικού hardware του Arduino. Οι κυριότερες από αυτές αναλύονται στον πίνακα που ακολουθεί.

Πίνακας 1: Κυριότεροι παράμετροι και μεταβλητές του Arduino

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	–	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	int	–	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	int	–	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
OUTPUT	Σταθερά	int	–	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	–	(pin, mode)	Καθορίζει αν το συγκεκριμένο ψηφιακό pin θα είναι pin εισόδου ή pin εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο mode (INPUT ή OUTPUT αντίστοιχα). Θέτει την κατάσταση pinstatus (HIGH ή LOW) στο συγκεκριμένο ψηφιακό pin.
digitalWrite	Εντολή	–	(pin, pinstatus)	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού pin (0 για LOW και 1 για HIGH) εφόσον αυτό είναι pin εισόδου.
digitalRead	Συνάρτηση	int	(pin)	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο type για να καθορίσει την τάση αναφοράς (Vref) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το pinAREF αντίστοιχα) Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο pin αναλογικής εισόδου στην κλίμακα 0 ως Vref.
analogReference	Εντολή	–	(type)	Θέτει το συγκεκριμένο ψηφιακό pin σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος value καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με value 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
analogRead	Συνάρτηση	int	(pin)	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λάβετε υπόψη ότι λόγω του τύπου μεταβλητής (unsigned long δηλ. 32bit) θα γίνει overflow σε 2 ³² ms δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν.
analogWrite	Εντολή	–	(pin, value)	Σταματά προσωρινά την ροή του προγράμματος για χρόνο ms. Η παράμετρος time είναι unsigned long
millis	Συνάρτηση	unsigned long	()	
delay	Εντολή	–	(time)	

(από 0 ως 2^{32}). Σημειώστε ότι παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.

attachInterrupt	Εντολή	-	(interrupt,function,triggermode)	Θέτει σε λειτουργία το συγκεκριμένο interrupt, ώστε να ενεργοποιεί την συνάρτηση function, κάθε φορά που ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο triggermode: LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνεται LOW) RISING (όταν από LOW γίνεται HIGH) FALLING (όταν από HIGH γίνεται LOW) CHANGE (όταν αλλάξει κατάσταση γενικά)
detachInterrupt	Εντολή	-	(interrupt)	Απενεργοποιεί το συγκεκριμένο interrupt.
noInterrupts	Εντολή	-	()	Σταματά προσωρινά την λειτουργία όλων των interrupt
interrupts	Εντολή	-	()	Επαναφέρει την λειτουργία των interrupt που διακόπηκε προσωρινά από μια εντολή noInterrupts.
Serial.begin	Μέθοδος κλάσης	-	(datarate)	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
Serial.println	Μέθοδος κλάσης	-	(data)	Διοχετεύει τα δεδομένα data για αποστολή μέσω του σειριακού interface. Η παράμετρος data μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

Η βασική ρουτίνα setup() εκτελείται μια φορά μόνο κατά την εκκίνηση του προγράμματος ενώ η βασική ρουτίνα loop() περιέχει τον βασικό κορμό του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνέχεια σαν ένας βρόγχος while(true). (python.org.gr)

Ένα άλλο χαρακτηριστικό που κάνει τις πλακέτες Arduino ευρέως διαδεδομένες είναι ότι έχουν ένα LED και μία αντίσταση φορτίου (μεταξύ του pin 13 και της γείωσης). Στο περιβάλλον του Arduino, ο χρήστης μπορεί να γράψει ένα πρόγραμμα σαν αυτό της εικόνας 3: αυτό αποτελεί ένα τυπικό πρώτο πρόγραμμα για έναν μικροελεγκτή και κάνει ένα LED (Light-EmittingDiode) απλά να αναβοσβήνει επαναλαμβανόμενα.



Εικόνα 6: Παράδειγμα προγράμματος με Arduino IDE.(Engimusing)

Ο κανονικοί μεταγλωττιστές C++ δεν θα μπορούσαν να θεωρήσουν έγκυρο ένα πρόγραμμα περιβάλλοντος Arduino (όπως αυτό της εικόνας 3). Για να θεωρηθεί έγκυρο με το πάτημα του κουμπιού “Uploadtool/Onboard” το IDE δημιουργεί ένα προσωρινό αρχείο με ένα αντίγραφο του κώδικα περιλαμβάνοντας ένα «include» στην αρχή και μια απλή συνάρτηση “main()” στο τέλος.

Για την μεταγλώττιση των προγραμμάτων το Arduino χρησιμοποιεί το GNU toolchain και το AVR Libc, ενώ για να φορτώνει προγράμματα το εργαλείο γραμμής εντολών avrdude.

Το IDE του Arduino χρησιμοποιεί το GNU toolchain και το AVR Libc για να μεταγλωττίζει προγράμματα και το avrdude για να φορτώνει προγράμματα στην πλακέτα. Αφού χρησιμοποιεί τον μικροελεγκτή της εταιρείας Atmel-όπως αναφέρθηκε νωρίτερα, είναι και λογικό πως το περιβάλλον ανάπτυξης της εταιρίας, το AVR Studio αλλά και η νεότερη έκδοση του AtmelStudio χρησιμοποιούνται για την ανάπτυξη του λογισμικού σε συστήματα Arduino.

1.6 Πλεονεκτήματα χρήσης Arduino

Υπάρχουν πολλά συστήματα μικροελεγκτών και μικροεπεξεργαστών που προσφέρουν την δυνατότητα παρόμοιων λειτουργιών με το Arduino. Τέτοια είναι τα ParallelBasic, Stamp, Netmedia's BX-24, Phidgets και MIT's Handyboard. Οι ομοιότητα τους έγκειται στην απλοποίηση του χειρισμού των μικροελεγκτών μέσα από ένα χρηστικά εύκολο περιβάλλον λειτουργίας. Το

Arduino όμως υπερέχει καθώς είναι εύχρηστο για μια πληθώρα ομάδων (καθηγητές, μαθητές, ενδιαφερόμενοι ερασιτέχνες) λόγω των εξής στοιχείων του:

- **Χαμηλό κόστος.** Ένα πλήρες συναρμολογημένο Arduino κοστίζει λιγότερο από 15 ευρώ, μία τιμή που χαμηλώνει ακόμα περισσότερο αν συναρμολογηθεί από τον χρήστη (με το χέρι). Οι πλακέτες του είναι συγκριτικά με άλλες πλατφόρμες πιο φθηνές.

- **Γενική πλατφόρμα.** Ενώ οι περισσότεροι μικροελεγκτές χρειάζονται λειτουργικό σύστημα (OperatingSystem, OS) Windows για να λειτουργήσουν, το Arduino είναι γνωστό πως δουλεύει σε Windows, Macintosh και Linux.

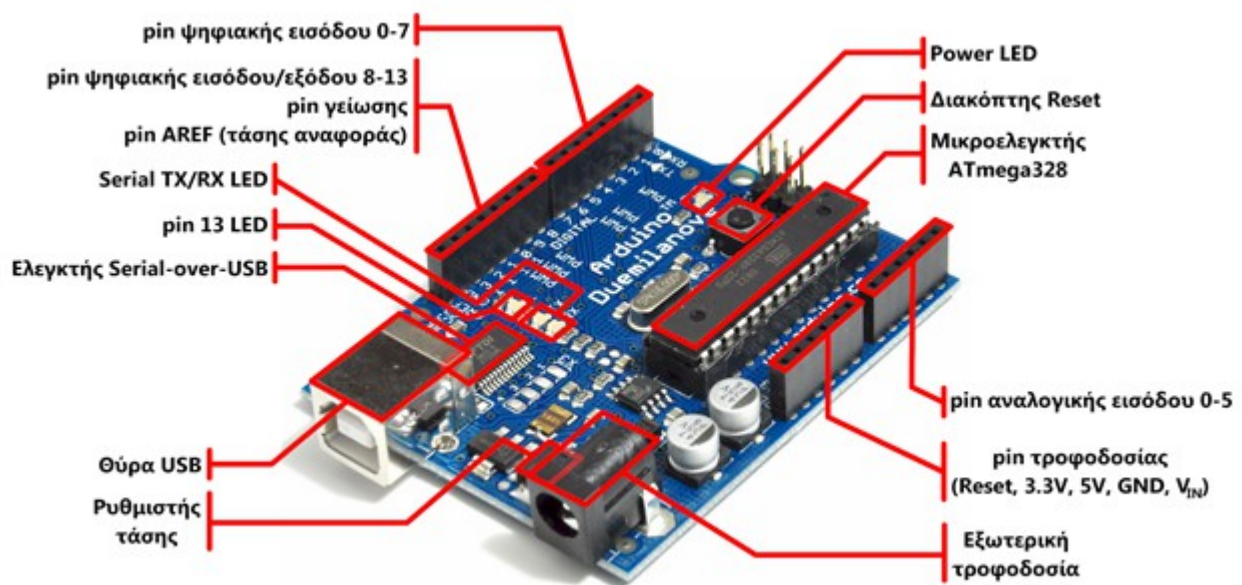
- **Απλό, κατανοητό προγραμματιστικό περιβάλλον.** Ενώ είναι εύκολο στη χρήση για τους αρχάριους, το διακρίνει ένας ευέλικτος χαρακτήρας απο τον οποίο μπορούν να επωφελούνται οι προχωρημένοι χρήστες. Η συγγένειά του με το περιβάλλον Processing το κάνει ιδανικό για καθηγητές και οικείο για τους μαθητές.

- **Εργαλείο Ανοιχτού κώδικα και επεκτάσιμου λογισμικού.** Οι βιβλιοθήκες της C++ είναι πολύ βοηθητικό εργαλείο για την επέκταση του λογισμικού Arduino, σε συνάρτηση με τον ανοιχτό κώδικα που είναι διαθέσιμος προς εξέλιξη στους πιο έμπειρους προγραμματιστές. Ακόμη, η στενή συγγένεια με τις βιβλιοθήκες της C++ προσφέρει εύκολη μετάβαση και κατανόηση και για τα άτομα που ενδιαφέρονται από Arduino να μεταβούν στην AVR C γλώσσα προγραμματισμού στην οποία είναι βασισμένο. Αυτός είναι και ο λόγος που μπορεί κάποιος να προσθέσει απευθείας AVR-C κώδικα σε Arduino χωρίς πρόβλημα.

- **Εργαλείο Ανοιχτού κώδικα και επεκτάσιμου υλικού.** Επιπλέον τα λειτουργικά σχέδια του λογισμικού Arduino είναι δημοσιευμένα υπό Δημιουργική Κοινή Άδεια (Creative Common License), Ταυτόχρονα είναι βασισμένο στους μικροελεγκτές ATMEGA8 και ATMEGA168 της Atmel, είναι εύκολο να τροποποιηθεί. Φυσικά μόνον οι έμπειροι σχεδιαστές και κατασκευαστές κυκλωμάτων μπορούν να δημιουργήσουν τροποποιημένες και βελτιωμένες εκδόσεις λειτουργικών τμημάτων του Arduino. (<http://brain.ee.auth.gr>)

Η συναρμολόγηση της μητρικής πλακέτας όμως, και η κατανόηση της βασικής δομής και λειτουργίας του, είναι πολύ εύκολη ενώ ταυτόχρονα μειώνει το κόστος του.

Όπως έχει αναφερθεί πως ο Arduino έχει σειριακή διεπιφάνεια (interface), αφού τέτοια είναι και η επικοινωνία την οποία υποστηρίζει ο μικροελεγκτής του ATmega. Υπάρχει όμως ένας ελεγκτής Serial-over-USB που βοηθά στην επικοινωνία με τον υπολογιστή μας μέσω μιας απλής θύρας USB. Η επικοινωνία αυτή είναι αμφίδρομη εξυπηρετώντας, αρχικά την μεταφορά των προγραμμάτων από τον υπολογιστή στο Arduino, αλλά και την επικοινωνίας με τον Arduino την στιγμή που τα εκτελεί βλέποντας τα αποτελέσματά του στην οθόνη ενός Η/Υ.



Εικόνα 7: Βασική δομή Arduino(Programino)

Τα 14 θηλυκά pin που βρίσκονται στην άνω πλευρά του (και φαίνονται χαρακτηρισμένα με τα νούμερα 0 ως 13) λειτουργούν στα 5V, με μέγιστο τα 40mA, και λειτουργούν ως είσοδοι του Arduino.Ενώ τα 8 ως 13 λειτουργούν ως είσοδοι αλλά και έξοδοι. Ο τρόπος που λειτουργούν τα pins ως ψηφιακή εξόδοι είναι απλά με διοχέτευση ή όχι ρεύματος. Από το πρόγραμμα λοιπόν μπορούμε να τα θέσουμε σε κατάσταση HIGH ή LOW κι έτσι το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα, ανάβοντας για παράδειγμα ή όχι ένα LED που συνδέουμε στο συγκεκριμένο pin. Στην περίπτωση που ένα από αυτά τα pin έχει ρυθμιστεί ως

ψηφιακή είσοδος από το πρόγραμμά μας, τα HIGH και LOW αντιστοιχούν σε καταστάσεις που αυτή η συσκευή διοχετεύει ή όχι ρεύμα σε αυτό το pin, δίνοντας μας μια πληροφορία (πχ την κατάσταση ενός διακόπτη).

Αναλυτικότερα και υπό περίπτωση:

- **Τα pin 0 και 1:** στις περιπτώσεις που το πρόγραμμά μας ενεργοποιεί την σειριακή θύρα αυτά λειτουργούν ως δέκτης (Receiver - RX) και ως πομπός (Transmitter- TX). Έτσι, όταν στο πρόγραμμά μας ενεργοποιούμε το σειριακό interface χάνουμε 2 ψηφιακές εισόδους/εξόδους, αφού αυτές λειτουργούν πλέον ως RX και TX. Για παράδειγμα, αν το πρόγραμμά μας μεταφέρει δεδομένα στην σειριακή αυτά μεταφέρονται και στην USB θύρα μέσω του ελεγκτή Serial-Over-USB, καθώς και στο pin 0 ώστε να διαβάσει μια άλλη πιθανή συσκευή που μπορεί να βρίσκεται ενδεχομένως στο pin1 (πχ ένα νέο σύστημα Arduino).

- **Τα pin 2 και 3:** χρησιμεύουν και ως εξωτερικοί διακόπτες (interrupt 0 και 1 αντίστοιχα). Έτσι μπορούμε να ρυθμίσουμε το πρόγραμμά μας ώστε τα συγκεκριμένα pin να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοί που μετά από συγκεκριμένες αλλαγές/συνθήκες προάγουν την άμεση στάση του τακτικού (regular) προγράμματος ώστε να εκτελεστεί μια νέα συνάρτηση. Αυτά τα pin είναι ιδιαίτερα σημαντικά και χρήσιμα σε καταστάσεις που απαιτούνται συγχρονισμοί μεγάλης ακρίβειας.

- **Τα pin 3, 5, 6, 9, 10 και 11:** χρησιμεύουν και ως ψευδοαναλογικές έξοδοι μέσω του PWM (Pulse Width Modulation) συστήματος, με το σύστημα δηλαδή που αξιοποιείται και στους ηλεκτρονικούς υπολογιστές (H/Y) από τις μητρικές για τον έλεγχο της ταχύτητας του ανεμιστήρα. Το PWM σύστημα δεν είναι πραγματικά αναλογικό, αλλά εναλλάσσει με μεγαλύτερη συχνότητα και για διαφορετικούς χρόνους τις δύο καταστάσεις HIGH και LOW. Ας πάρουμε για παράδειγμα την περίπτωση κατά την οποία συνδέουμε ένα LED σε κάποιο από αυτά τα pin. Τότε, μπορούμε να δούμε 256 καταστάσεις αυτού του LED (8bit), από σβηστό (τιμή 0) έως αναμμένο με τη μέγιστη δυνατή φωτεινότητά του (τιμή 255). Αν πχ θέσουμε στην έξοδο την τιμή 127 (σχεδόν η μέση του 256), η έξοδος θα δίνει και πάλι 5V (και όχι 2,5V) αλλά με τέτοιο παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Στην κάτω πλευρά του Arduino υπάρχουν 6 ακόμα pin, με νούμερα από 0 έως και 5, και την σήμανση ANALOGIN. Αυτά αποτελούν αναλογικές εισόδους εκμεταλλευόμενα τον Analog to Digital Converter (ADC) που διαθέτει πάνω του ο μικροελεγκτής. Έτσι, αν σε ένα από αυτά δώσουμε

μία κυμαινόμενη τάση από 0V έως την μέγιστη (προεπιλεγμένη Vref) 5V, τότε μπορούμε μέσα από το πρόγραμμά μας να δούμε την τιμή του ως ακέραιο αριθμό ανάλυσης (για τα 10bit) από 0 (που αντιστοιχεί στα 0V) έως 1023 (που αντιστοιχεί στα 5V). Η προεπιλεγμένη Vref τάση αναφοράς μπορεί βέβαια να αλλάξει και να ρυθμιστεί στο 1.1V, ή σε όποια άλλη επιθυμούμε (μεταξύ 2 και 5V), και να τροφοδοτήσουν εξωτερικά με την τάση αυτή τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν δώσουμε στο pin AREF 3.3V και στην συνέχεια αναγνώσουμε την τιμή κάποιου pin αναλογικής εισόδου στο οποίο εφαρμόζουμε τάση 1.65V, το Arduino μας δίνει την τιμή 512. Τέλος, αυτά τα 6 pin μπορούμε να τα τροποποιήσουμε με αντίστοιχη εντολή σε ψηφιακό pin εισόδου/εξόδου. Αυτό σημαίνει ότι μπορούν να γίνουν λειτουργικά ίδια με τα pin της απέναντι πλευράς (στα οποία έγινε αναφορά παραπάνω) και να ονομαστούν από 0-5 σε 14-19. (Reform of Social Welfare Programmes)

1.7 δίοδος φωτοεκπομπής (LED)

1.7.1 Βασική αρχή λειτουργίας δίοδου φωτοεκπομπής

Η δίοδος φωτοεκπομπής (LED, Light Emitting Diode) αποκαλείται ένας ημιαγωγός (επαφή P-N), ο οποίος όταν πολωθεί ορθά, εκπέμπει φωτεινή ακτινοβολία. Ακολουθούν η σχηματική αναπαράσταση καθώς και πραγματική μορφή διαφόρων τύπων LED.



Εικόνα 8: Δίοδος φωτοεκπομπής (LED, Light Emitting Diode).

Στην σημερινή εποχή, τα LED προσφέρονται σε διάφορα είδη αλλά και χρώματα. Το χρώμα του φωτός που εκπέμπεται, εξαρτάται από τη χημική σύσταση του ημιαγωγού που χρησιμοποιείται και

μπορεί να είναι φως ορατό, υπεριώδες ή υπέρυθρο. Επίσης ρόλο παίζει και το μήκος κύματος του φωτός που εκπέμπεται το οποίο εξαρτάται από τα υλικά από τα οποία αποτελείται ο ημιαγωγός.

1.7.2 Διάρκεια ζωής των LED

Τα LED είναι εκ κατασκευής τους ένα συμπαγές στοιχείο, που αντέχει αρκετά σε μηχανικές καταπονήσεις. Παρόλα αυτά, θα πρέπει να λειτουργεί σε συγκεκριμένες τιμές θερμοκρασίας, διότι η απόδοσή τους σχετίζεται με τη θερμοκρασία. Όσο χαμηλότερη η θερμοκρασία, τόσο το LED φτάνει τη μέγιστη φωτιστική του απόδοση, αλλά και καλύτερη απαγωγή της δημιουργούμενης θερμότητας από τη λειτουργία του γίνεται, εάν πρόκειται για LED υψηλής ισχύος. Η χρήση των LED σε περιβάλλον με υψηλή θερμοκρασία, σημαίνει μικρότερη φωτιστική ισχύ θα εκπέμπεται από το LED, αλλά μπορεί να οδηγήσει και στην καταστροφή του. Αντιθέτως, η χρήση του σε πολύ ψυχρά κλίματα, μπορεί να σημαίνει ότι θα έχουμε πολύ μεγάλη φωτιστική ισχύ, αλλά λόγω της πολύ μικρής θερμότητας που αναπτύσσεται στο LED, μπορεί να σχηματιστεί πάγος ή και χιόνι επάνω στο LED, εμποδίζοντας την εκπομπή του φωτός. Οι κατασκευαστές διόδων εκπομπής φωτός, δίνουν τιμές λειτουργίας και μετρήσεις για τα LED, στη θερμοκρασία των 25° C, και το προσδόκιμο ζωής των, μπορεί να κυμανθεί από 20.000 έως και 100.000 ώρες. Το πιο κοινό σύμπτωμα των LED είναι ότι με την πάροδο του χρόνου μειώνεται η φωτεινή τους ένταση, καθώς και η απόδοσή τους.

1.8 Φωτορυθμικό

Το χρώμα μας συγκινεί και μας εκφράζει. Αναμφίβολα, δεν μπορούμε να ζήσουμε χωρίς αυτό. Έχει αποδειχτεί ότι οι προτιμήσεις μας είναι έμφυτες. Μπορούμε να πούμε ότι το χρώμα είναι ταυτόχρονα μια θετική και μια αρνητική δύναμη. Το χρώμα είναι κραδασμός, κραδασμός είναι κίνηση, κίνηση είναι δραστηριότητα μιας θετικής ή αρνητικής δύναμης. Ο δημιουργικός κραδασμός και ο καταστρεπτικός κραδασμός προέρχονται από την ίδια πηγή. Η διαφορά έγκειται στον τρόπο με τον οποίο χρησιμοποιούμε αυτήν την ενέργεια.

Τα φωτορυθμικά συστήματα δίνουν μια διαφορετική διάσταση στον χώρο που τους περιβάλλει και προσθέτουν συναίσθημα στον χρήστη του χώρου. Το ρυθμιζόμενο φωτορυθμικό ανάλογα με την ένταση του περιβάλλοντος ήχου προσθέτει μια ξεχωριστή εμπειρία και ένα μοναδικό συναίσθημα στον χρήστη. Το φωτορυθμικό μπορεί να λειτουργήσει αυτόνομα χωρίς την ύπαρξη κονσόλας ελέγχου μιας και διαθέτει ενσωματωμένο μικρόφωνο που μετατρέπει τη μουσική σε φωτεινά εφε σύμφωνα με το ρυθμό.

Έχει παρέλθει προ πολλού η εποχή που ο τεχνητός φωτισμός χρησίμευε απλά για να μπορούμε να βλέπουμε μετά το σούρουπο. Στην αρχή του αιώνα που διανύουμε ο τεχνητός φωτισμός (και ο σχεδιασμός του) είναι ταυτόχρονα τέχνη και επιστήμη.

Οι οπτικές μας εντυπώσεις από ένα περιβάλλον παίζουν σημαντικό ρόλο. Ο φωτισμός είναι συνάρτηση πολλών παραμέτρων, δηλαδή των διαστάσεων (του μεγέθους) του χώρου, των χρωματισμών του δαπέδου (που απορροφούν και αντανακλούν το φως), της δραστηριότητας που ασκείται μέσα στον χώρο, του είδους του φωτισμού (γενικός-τοπικός-θεατρικός, κλπ) της κατεύθυνσης του φωτισμού (άμεσος-έμμεσος κλπ), της ομοιομορφίας, της χροιάς, της χρωματικής απόδοσης, της ύπαρξης φυσικού φωτισμού και τέλος των φωτιστικών και της διευθέτησής τους μέσα στον χώρο.



Εικόνα 9: Το φωτορυθμικό της κατασκευής.

1.9 ΑΙΣΘΗΤΗΡΕΣ

Αισθητήρας ονομάζεται μία συσκευή που ανιχνεύει ένα φυσικό μέγεθος και παράγει από αυτό μία μετρήσιμη έξοδο. Για παράδειγμα, το υδραργυρικό θερμομέτρο μετατρέπει τη μετρούμενη θερμοκρασία σε διαστολή, η οποία μπορεί να αναγνωσθεί από ένα βαθμονομημένο σωλήνα. Οι αισθητήρες χρησιμοποιούνται σε καθημερινά αντικείμενα, όπως κουμπιά ανελκυστήρων ευαίσθητα στην αφή και λάμπες φωτισμού που εκπέμπουν λαμπρότερα ή απαλότερα αγγίζοντας τη βάση τους. Υπάρχουν αναρίθμητες ακόμη χρήσεις που οι περισσότεροι άνθρωποι δεν αντιλαμβάνονται. Εφαρμογές τους συναντούμε στα αυτοκίνητα, σε μηχανές, στην αεροναυπηγική, την ιατρική, τη βιομηχανία και τη ρομποτική.

1.9.1 Είδη και χαρακτηριστικά αισθητήρων

Συχνά οι αισθητήρες δεν δίνουν στην έξοδο τους κατάλληλο ηλεκτρικό σήμα. τότε απαιτείται η χρήση ενός επιπρόσθετου ηλεκτρονικού κυκλώματος, το οποίο να λαμβάνει την έξοδο του αισθητήρα και να τη μετατρέπει σε κατάλληλο ηλεκτρικό σήμα, σύμφωνα με τις απαιτήσεις των επόμενων βαθμίδων. το κύκλωμα αυτό ονομάζεται κύκλωμα ρύθμισης σήματος (signal conditioning circuit), κύκλωμα ελέγχου (control circuit) η εξωτερική μονάδα (outer η external module). για παράδειγμα, υπάρχουν αισθητήρες στάθμης που μετρούν το χρόνο που απαιτείται για να ανακλαστεί ένα υπερηχητικό κύμα απο τη μετρούμενη επιφάνεια και να επιστρέψει στο σημείο από όπου εκ πέμφθηκε. σε αυτούς πρέπει να υπάρχει κατάλληλο κύκλωμα για τη μετατροπή των τιμών χρόνου σε ανάλογες τιμές τάσης.

Οι αισθητήρες που απαιτούν εξωτερική τροφοδοσία για να λειτουργήσουν ονομάζονται ενεργοί. για παράδειγμα, ένας αισθητήρας γραμμικής μετατόπισης 1ndt πρέπει να τροφοδοτείται από κατάλληλη εναλλασσόμενη τάση. Οι αισθητήρες που δημιουργούν μόνοι τους μια τάση και δε χρειάζονται εξωτερική τροφοδοσία ονομάζονται παθητικοί. Τέτοιοι είναι για παράδειγμα οι πιεζοηλεκτρικοί κρύσταλλοι, που όταν πιεστούν αναπτύσσουν στα άκρα τους ηλεκτρική τάση.

Τα χαρακτηριστικά των αισθητήρων συγκροτούν τις προδιαγραφές τους (specifications) και είναι πολλά. Παρότι οι ποικίλοι αισθητήρες που υπάρχουν σήμερα στηρίζονται σε διαφορετικές αρχές λειτουργίας, έχουν κοινά τα βασικά τους χαρακτηριστικά. Αυτά είναι τα ακόλουθα:

1: Γραμμικότητα

2: Ευαισθησία

3: Διακριτική ικανότητα

4: Ακρίβεια

5: Εύρος τιμών εισόδου

6: Εύρος τιμών εξόδου

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

2.1 Ενισχυτής

Όπου υπάρχει ήχος ο οποίος δεν προέρχεται από φυσική πηγή κάπου υπάρχει ένα μεγάφωνο και σίγουρα μια βαθμίδα ενίσχυσης ήχου πίσω από αυτό. Ο ολοκληρωμένος ενισχυτής είναι η βαθμίδα διαχείρισης και ενίσχυσης του σήματος που λαμβάνουμε από τις πηγές ήχου (CD, πικάπ, ραδιόφωνο κ.λ.π.). Το σήμα αυτό αφού διαμορφωθεί και ενισχυθεί κατάλληλα, οδηγεί τα ηχεία

μας.

Τους ενισχυτές γενικά τους χωρίζουμε σε στερεάς κατάστασης (με τρανζίστορ), σε ενισχυτές λυχνίας και σε υβριδικούς όταν κάποια κυκλώματα υλοποιούνται με τρανζίστορ ενώ άλλα με λυχνίες.

Ένας ενισχυτής ήχου αποτελείται από πολλά στάδια, εξού και ο χαρακτηρισμός του ως ολοκληρωμένος ενισχυτής. Έχουμε λοιπόν το στάδιο προενίσχυσης, τη βαθμίδα του τελικού ενισχυτή και τέλος το τροφοδοτικό αυτών. Το στάδιο προενίσχυσης επιτρέπει να επιλέξουμε τη προς ακρόαση πηγή, να διαμορφώσουμε ανάλογα το σήμα (μπάσα, πρίμα, μεσαία. loudness) και αφού αυτό ενισχυθεί με το μπουτόν της έντασης να οδηγήσει τη τελική βαθμίδα.



Εικόνα 10: Ενισχυτής Skytronic τάξης Delta (Skytronic)

Η τελική βαθμίδα ή αλλιώς βαθμίδα ισχύος, ενισχύει το σήμα όσον αφορά τη τάση αλλά κυρίως έχει τη δυνατότητα παροχής υψηλών ρευμάτων απαραίτητων για την οδήγηση των ηχείων.

Το τροφοδοτικό του ενισχυτή λαμβάνει εναλλασσόμενη τάση των 230 Volts του δικτύου και παράγει τις συνεχής τάσης για τη λειτουργία όλων των βαθμίδων.

Γενικά ισχύει ότι όσο περισσότερα και ανεξάρτητα για το κάθε στάδιο τροφοδοτικά υπάρχουν στο εσωτερικό ενός ενισχυτή τόσο καλύτερη είναι και η απόδοση του. Από εκεί προκύπτει και η ποιοτικότερη λύση των ανεξάρτητων κομματιών.

2.2 Κατηγορίες Ενισχυτών

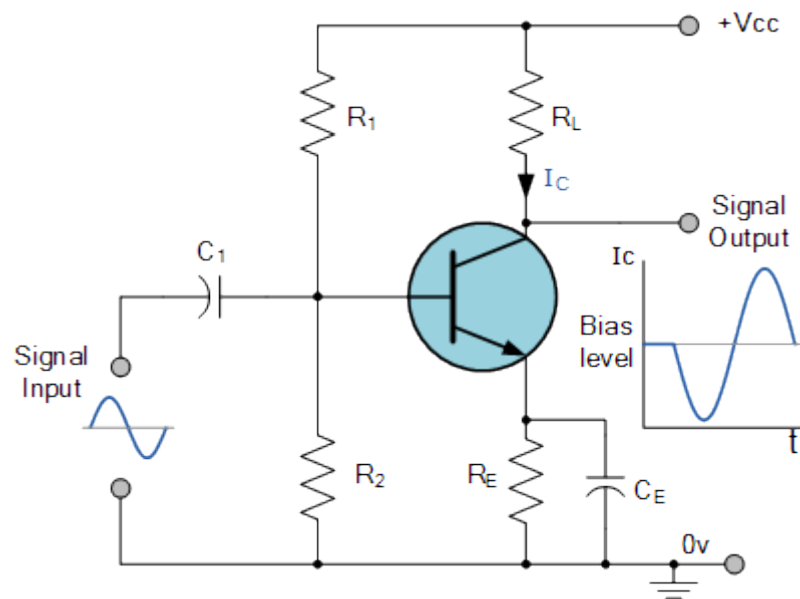
Οι βασικές και πιο συχνά κατασκευάσιμες κατηγορίες (κυκλωματικές διατάξεις) των ενισχυτών είναι τέσσερις, είναι η τάξη "A", η τάξη "B", η τάξη "AB", και η τάξη "D". Υπάρχουν και άλλες τάξεις ενισχυτών πιο σύνθετες αλλά χρησιμοποιούνται πολύ λιγότερο σε σχέση με τις προαναφέρουσες. Αναφορικά είναι η τάξη "C", η τάξη "E", τάξη "X".

Οι ενισχυτές με λειτουργία τάξεων "D", "E", "X", λειτουργούν κατά την παραγωγή του σήματος με "δια κοπτικό τρόπο" (switch mode) και γι' αυτό τον λόγο τους ονομάζουμε ψηφιακούς.

Οι υπόλοιποι ενισχυτές με τάξεις "A" , "B" , "AB" , "C" , λειτουργούν με "γραμμικό τρόπο" (line mode) και είναι οι γνωστοί σε όλους αναλογικοί. (repository.kallipos.gr/bitstream)

2.2.1 Ενισχυτική Τάξη "A"

Οι ενισχυτικές διατάξεις τάξης "A" λειτουργούν καθ' όλη την διάρκεια του κύκλου εισόδου έτσι ώστε το σήμα εισόδου να είναι ακριβές αντίγραφο του σήματος εξόδου αλλά ενισχυμένο και χωρίς παραμόρφωση. Οι ενισχυτές αυτοί είναι συνήθως μικρής ισχύος και με μικρό βαθμό απόδοσης (efficiency). Οι ενισχυτές αυτοί λόγω αυτής της διάταξης το στοιχείο ενίσχυσης τους άγει συνεχώς ακόμα και όταν δεν υπάρχει καθόλου εισόδος με αποτέλεσμα να καταναλώνει συνεχώς ηλεκτρική ισχύ. Αυτός είναι και ο βασικός λόγος χαμηλού βαθμού απόδοσης του ενισχυτή αφού για κάθε ένα βατ (W) που πηγαίνει στο μεγάφωνο μας ο ενισχυτής μας στην καλύτερη καταναλώνει άλλο ένα. Η κατανάλωση ισχύος του ενισχυτή σε κατάσταση ηρεμίας (χωρίς είσοδο) είναι σχεδόν ίδια με την κατανάλωση του σε πλήρη ένταση ήχου.



Εικόνα 11: Διάγραμμα ενισχυτή τάξης A (wikipedia)

2.2.2 Ενισχυτική τάξη "B" και "AB"

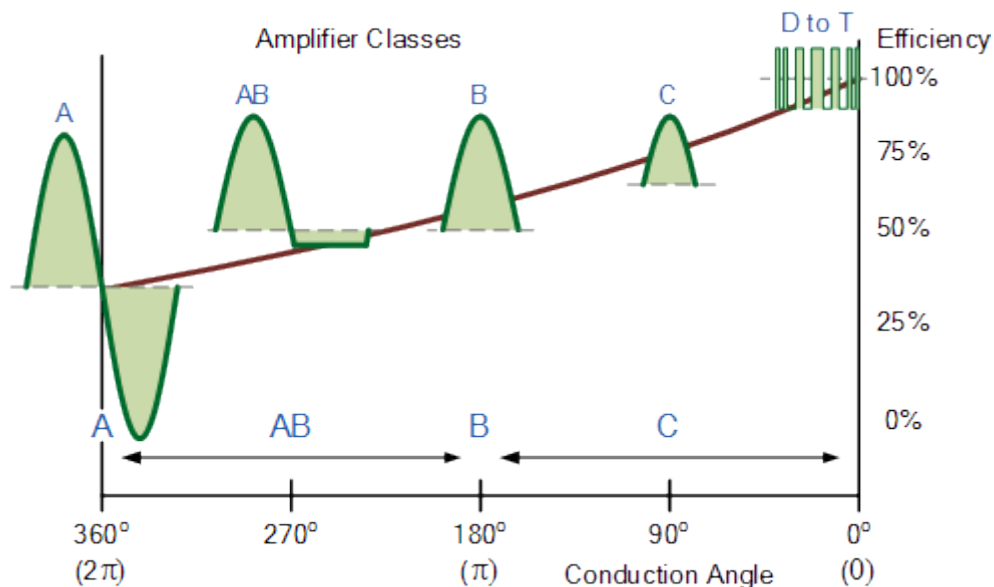
Οι ενισχυτικές διατάξεις τάξης "B" ενισχύουν μόνο τον μισό κύκλο του σήματος και αποκόπτουν το άλλο μισό με αποτέλεσμα να παράγουν υψηλή παραμόρφωση αλλά με πιο βελτιωμένο βαθμό απόδοσης σε σχέση με την τάξη "A". Ο βαθμός απόδοσης της τάξης "B" βαδίζει περίπου στο 75%. Για την παραγωγή ενός ολοκλήρου σήματος (κυματομορφή) χρειάζονται δυο στοιχεία όπου το ένα θα ενισχύει την θετική ημπερίοδο και το άλλο την αρνητική. Αυτή η διάταξη δίνει πολύ καλό

βαθμό απόδοσης (efficiency) αλλά πάσχει από ένα "σύνδρομο" που δεν ταυτίζει τα δυο μισά του σήματος. Αυτό ονομάζεται "παραμόρφωση διασταυρώσεως" (Crossover distortion).

Για την λύση αυτή του "συνδρόμου" έχει κατασκευαστεί ένα "φάρμακο" (μια άλλη κλάση) όπου πολώνει τα στοιχεία με τέτοιο τρόπο ώστε να μην "σβήνουν" όταν δεν χρησιμοποιούνται. Αυτό το "φάρμακο" λέγεται τάξη "AB". Σε αυτή την κλάση κάθε στοιχείο λειτουργεί με τον ίδιο τρόπο όπως στην κλάση "B" (μισή κυματομορφή), αλλά ταυτόχρονα ένα άλλο βρίσκεται σε αγωγή (άγει) με αποτέλεσμα να μειώνεται το νεκρό κομμάτι αναμεσα στις δυο μισές κυματομορφές. Αυτή η διάταξη λειτουργίας είναι η πιο πολυχρησιμοποιημένη για τον λόγο ότι έχει αρκετά καλό βαθμό απόδοσης 60-70% σε σχέση με την τάξη "A" αλλά με μικρότερη γραμμικότητα σε σχέση με την τελευταία. Είναι μια "συμβιβαστική" τάξη καθώς έχει αρκετά καλή, πιστότητα αναπαραγωγής σε χαμηλές εντάσεις αλλά και στις υψηλές η παραμόρφωση που παράγει δεν γίνεται εύκολα αντιληπτή γιατί η στάθμη της εξόδου του ήχου είναι κατά πολύ υψηλότερη.

2.2.3 Ενισχυτική τάξη "C"

Οι ενισχυτές ενισχυτική τάξη "C" άγουν λιγότερο από 50% κατά την λειτουργία τους. Η παραμόρφωση στην έξοδο είναι πολύ μεγάλη αλλά μπορούμε να πέτυχουμε απόδοση ισχύος μέχρι και 90%, απόδοση πολύ υψηλή αναλογιστούμε ότι πρόκειται για αναλογικό ενισχυτή. Οι ενισχυτές αυτοί δεν χρησιμοποιούνται για ενίσχυση ήχου λόγω τις πολύ υψηλής παραμόρφωσης τους καθώς το μόνο που βγάζουν στην έξοδο τους είναι ένα σήμα παλμού. Αυτή η τάξη χρησιμοποιείται σε ενισχυτές RF (radio frequency).



Εικόνα12: Διάγραμμα αποδοτικότητας διαφόρων κλάσεων των ενισχυτών (wikipedia)

2.3.1 Ανιχνευτής καπνού

Ο ανιχνευτής καπνού όπως λέει και το όνομά του είναι μια συσκευή που ανιχνεύει τον καπνό σαν ένδειξη έναρξης πυρκαγιάς. Σε χώρους συγκέντρωσης κοινού, σε βιομηχανίες, σε αποθήκες, κλπ είναι δυνατόν να είναι συνδεδεμένος με το σύστημα πυρόσβεσης, αντίθετα στα σπίτια είναι ένας τοπικός ανεξάρτητος ηχητικός ή και οπτικός συναγερμός ύπαρξης καπνού. Οι κοινοί ανιχνευτές

βρίσκονται μέσα σε ένα πλαστικό περίβλημα με περίμετρο γύρω στα 15 εκατοστά και πάχος γύρω 2,5 εκατοστά, με το υπόλοιπο σχήμα να διαφοροποιείται από κατασκευαστή σε κατασκευαστή.



Εικόνα 13: Ανιχνευτής Καπνού.

Οι περισσότεροι λειτουργούν με οπτική ανίχνευση (φωτοηλεκτρικά) ή με φυσική διεργασία (ιονισμός), άλλοι πάλι χρησιμοποιούν και τις δύο μεθόδους με σκεπτικό την αποτροπή του καπνίσματος σε χώρους όπου απαγορεύεται όπως τουαλέτες και σχολεία. Τροφοδοτούνται δε από μία μπαταρία όσον αφορά τους τοπικούς ανιχνευτές στα σπίτια ή είναι συνδεδεμένα με την ηλεκτρική εγκατάσταση του κτιρίου, με εφεδρική μπαταρία σε περιπτώσεις βιομηχανικών και γενικά μεγάλων εγκαταστάσεων.

2.3.2 Αυτόματο σύστημα συναγερμού

Το πρώτο αυτόματο ηλεκτρικό σύστημα συναγερμού πυρκαγιάς εφευρίσκει το 1890 από τον Francis Robbins Upton (ΗΠΑ δίπλωμα ευρεσιτεχνίας No 436961), συνεργάτη του Thomas Edison, χωρίς να υπάρχει κάποιο στοιχείο για την ανάμειξή του Edison. Το 1902, στο Birmingham της Αγγλίας ο George Andrew Darby κατοχυρώνει με δίπλωμα ευρεσιτεχνίας το πρώτο ηλεκτρικό ανιχνευτή θερμότητας και καπνού. Στα τέλη της δεκαετίας του 1930 ο Ελβετός φυσικός Walter Jaeger προσπαθεί να εφεύρει ένα αισθητήρα για τα δηλητηριώδη αέρια. Το σκεπτικό του είναι ότι το εισερχόμενο δηλητηριώδες αέριο θα δεσμεύεται από ιονισμένα μόρια αέρα και θα υπήρχε μια ηλεκτρική διαφοροποίηση σε ένα ηλεκτρικό κύκλωμα, αλλά οι μικρές συγκεντρώσεις αερίου δεν είχαν καμία επίδραση στο κύκλωμα ... πλήρης αποτυχία. Απογοητευμένος ανάβει ένα τσιγάρο και έκπληκτος διαπιστώνει ότι αυτό που δεν είχε κάνει το δηλητηριώδες αέριο το έκανε ο καπνός. Αυτό το πείραμα άνοιξε το δρόμο για το σύγχρονο ανιχνευτή καπνού. Χρειάστηκαν όμως 30 περίπου

χρόνια προόδου της Πυρηνικής Χημείας και των Ηλεκτρονικών εξαρτημάτων για να αποκτήσουμε ένα οικονομικά προσβάσιμο οικιακό ανιχνευτή. Τη δεκαετία του 1960 υπάρχουν μεν αλλά το κόστος αγοράς μπορούσαν να το σηκώσουν μόνο οι βιομηχανίες και οι μεγάλοι χώροι συνάθροισης κοινού. Η πρώτη πραγματικά προσιτή συσκευή ανίχνευσης καπνού για σπίτια, δημιουργήθηκε από τον Duane D. Pearsall το 1965, διαθέτει μπαταρία που μπορεί να εγκατασταθεί εύκολα και να αντικατασταθεί. Οι πρώτες μονάδες μαζικής παραγωγής κατασκευάζονται από την εταιρεία του Duane Pearsall, την Statitrol Corporation, στο Lakewood του Κολοράντο. Αυτές οι πρώτες μονάδες έγιναν από πυράντοχο ασάλι που διαμορφωνότανε σαν κυψέλη μέλισσας. Η μπαταρία ήταν επαναφορτιζόμενη και την κατασκεύαζε η Gates Energy. Σύντομα όμως απεδείχθη ότι η ανάγκη γρήγορης επαναλειτουργίας του ανιχνευτή μπορούσε να γίνει μόνο με αντικατάσταση αντί με επαναφόρτιση της μπαταρίας έτσι άρχισαν να χρησιμοποιούν ένα ζευγάρι μπαταρίες τύπου «AA» και τέλος ένα πλαστικό περίβλημα για να βελτιώσουν την εμφάνιση. Η Statitrol προωθούσε στην αγορά σε καθημερινή βάση γύρω στις 500 μονάδες μέχρι που το 1980 πουλά την πατέντα στην Emerson Electric η οποία προωθεί τους ανιχνευτές στη αγορά με το σύνθημα «τώρα χρειάζεται σε κάθε σπίτι». Από το 1969 που άρχισαν να κυκλοφορούν σε προσιτές τιμές, εκτιμάται ότι μέχρι σήμερα είναι εγκατεστημένοι ανιχνευτές στο 93% των Αμερικάνικων και στο 85% των Βρετανικών νοικοκυριών, αλλά πιθανολογείται ότι το 30% από αυτούς δεν λειτουργούν επειδή οι χρήστες ή έχουν αφαιρέσει τις μπαταρίες ή δεν τις έχουν αντικαταστήσει έγκαιρα.

Σύμφωνα με μελέτες στην Αυστραλία:

1. Τα άτομα άνω των 65 ετών αντιπροσωπεύουν περισσότερο από το 31% όλων των θανάτων από φωτιά.
2. Ο κίνδυνος θανάτου από φωτιά για άτομα άνω των 65 ετών είναι σχεδόν τριπλάσιος από ότι για τον υπόλοιπο πληθυσμό.

Έτσι δημιούργησαν το πρόγραμμα SABRE (Smoke Alarm Battery Replacement for the Elderly Αντικατάσταση Μπαταρίας Συναγερμού Καπνού για Ηλικιωμένους) το οποίο είναι σχεδιασμένο για την υποστήριξη των ηλικιωμένων, που διαμένουν στο σπίτι τους και που έχουν περιορισμένη οικιακή υποστήριξη, για να ζουν ανεξάρτητοι και να παραμένουν πολύτιμα κεφάλαια στην κοινωνία μας. Στα πλαίσια του προγράμματος επιτυγχάνεται συνεργασία των Πυροσβεστικών Υπηρεσιών με άλλους κοινοτικούς οργανισμούς και παρέχεται βοήθεια σε ηλικιωμένους πολίτες για την συντήρηση των συσκευών τους για πυρασφάλεια. Κυρίως για τις συσκευές συναγερμού καπνού που λειτουργούν με μπαταρία. Το πρόγραμμα SABRE δίνει την ευκαιρία στους πυροσβέστες της περιοχής όταν επισκέπτονται τους πολίτες για αλλαγή της μπαταρίας να κατανοήσουν καλύτερα τις ανάγκες τους και να τους δώσουν συμβουλές για την προστασία από φωτιές. (www.fire.gr)

2.3.3 Αισθητήρια καπνού για Arduino



Εικόνα 14: MQ-2 MQ2 Smoke Gas LPG Butane Hydrogen Gas αισθητήριο για Arduino.



Εικόνα 15: CO2 Carbon dioxide αισθητήριο για Arduino.

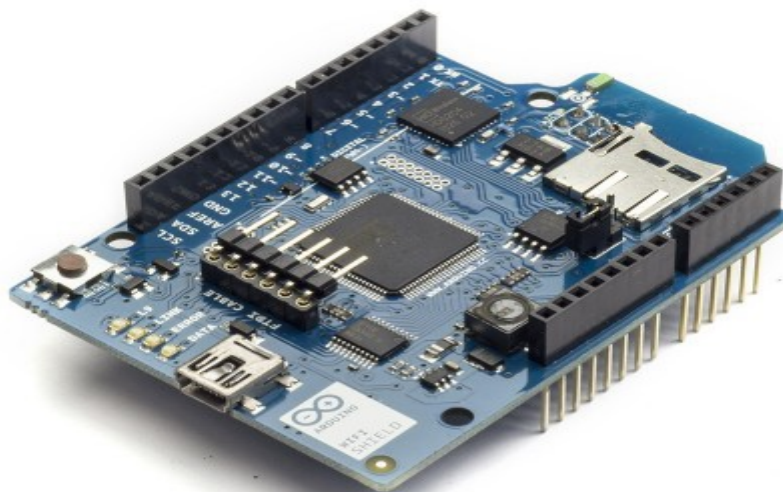
2.4 Σύνδεση με τον έξω κόσμο

Όπως περιεγράφηκε και στο κεφάλαιο (1.2), το σύστημα μας λειτουργεί με την υποστήριξη ενός ESP8266. Το ESP8266 αποτελεί ένα chip σύνδεσης με το διαδίκτυο το οποίο είναι συμβατό με την πλακέτα Arduino. Είναι ένα από τα σημαντικότερα κομμάτια του συστήματός μας. Με την βοήθεια του υπάρχει η δυνατότητα του απομακρυσμένου ελέγχου του συστήματός μας αλλά και η δημοσίευση των δεδομένων που ανακτούμε από αυτό στο διαδίκτυο. Με λίγα λόγια, μας επιτρέπει να πάμε την κατασκευή μας στο επόμενο επίπεδο.

Όμως, όπως αναφέρθηκε παραπάνω, δεν είναι μόνο το ESP8266 το οποίο μας προσφέρει την δυνατότητα σύνδεσης του Arduino με το διαδίκτυο. Μια από τις πιο δημοφιλείς επιλογές που επικρατούν είναι το WIFI Shield.

Το Arduino WIFI Shield αποτελεί ένα shield το οποίο είναι πλήρως συμβατό με την πλακέτα Arduino. Επιτρέπει την σύνδεση του Arduino με το διαδίκτυο ασύρματα. Αφού επιτευχθεί η σύνδεση του με ένα ασύρματο δίκτυο, προσφέρει σημαντικές δυνατότητες όπως είναι ο απομακρυσμένος έλεγχος του, η δημοσίευση δεδομένων που παράγει στο διαδίκτυο και πολλά άλλα. Για ένα σύστημα Arduino το οποίο με λίγη φαντασία και τα κατάλληλα εξαρτήματα μπορεί κυριολεκτικά να κάνει τα πάντα, η σύνδεση του με το διαδίκτυο απλά μας πάει ένα βήμα παραπάνω. Τα βασικά χαρακτηριστικά ενός WIFI Shield είναι τα παρακάτω:

- Λειτουργεί με τάση 5V (την οποία λαμβάνει από το Arduino Board)
- Είναι συμβατό με την πλακέτα Arduino Due
- Μέσο σύνδεσης: 802.11b/g networks
- Τύπος κρυπτογράφησης: WEP και WPA2 Personal
- Σύνδεση με τον Arduino στο SPI port
- Διαθέτει micro SD slot επάνω στην πλακέτα
- Διαθέτει ICSP headers
- Διαθέτει FTDI connection για συριακή αποσφαλμάτωση του WiFi shield
- Διαθέτει Mini-USB για να μπορούμε να αναβαθμίζουμε το WiFi shield firmware

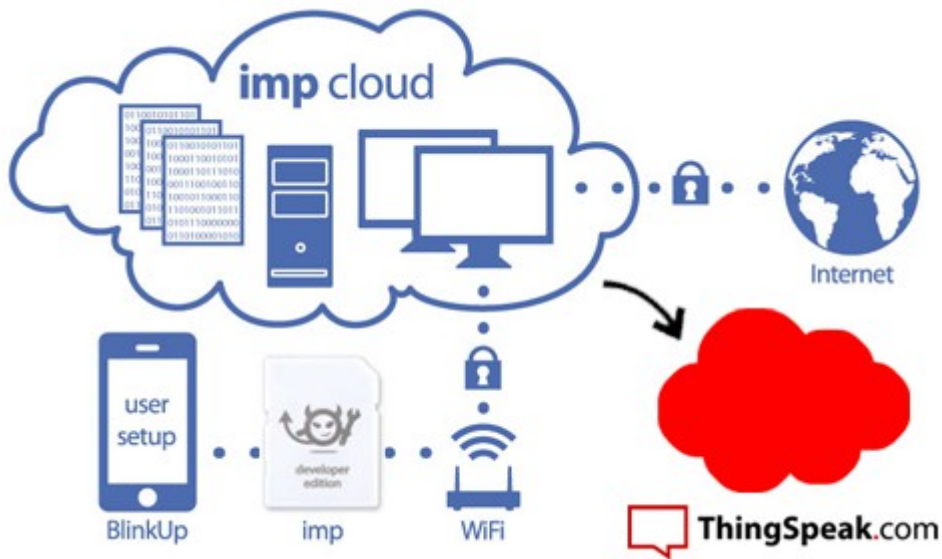


Εικόνα16: Arduino Wifi Shield (Wikipedia)

Και οι δύο επιλογές έχουν τα πλεονεκτήματα αλλά και τα μειονεκτήματά τους. Ξεκινώντας από το WIFI Shield, καθώς αποτελεί ένα shield κατασκευής Arduino, είναι πλήρως συμβατό με αυτό. Με άλλα λόγια, εάν δεν εγκατασταθεί σωστά στην αρχή, το ESP8266 μπορεί να δημιουργήσει προβλήματα στην επικοινωνία του συστήματός μας με τον έξω κόσμο. Επιπλέον, ένα WIFI Shield προσφέρει πολλά επιπλέον χαρακτηριστικά, όπως μεγαλύτερο εύρος δικτύου από αυτό του ESP8266, SD slot και πολλά άλλα. Παρόλα αυτά, υπάρχουν μερικοί βασικοί λόγοι για τους οποίους η επιλογή ενός ESP8266 θεωρήθηκε καταλληλότερη από την επιλογή του WIFI Shield για το δικό μας σύστημα. Αρχικά, το ESP8266 χρειάζεται 3,3V για να λειτουργήσει, μικρότερη από τα 5V που απαιτεί το WIFI Shield. Με άλλα λόγια, με την χρήση του ESP8266 επιτυγχάνουμε την ίδια λειτουργικότητα στο σύστημα μας, με μικρότερο ενεργειακό κόστος. Επιπλέον, το WIFI Shield διαθέτει πολλά περισσότερα χαρακτηριστικά σε σχέση με το ESP8266, τα οποία στην δική μας περίπτωση δεν χρειάζονται. Το τελευταίο, σε συνδυασμό με τη αρκετά υψηλότερη τιμή του WIFI Shield καθιστούν το ESP8266 την καταλληλότερη λύση που προσφέρει αυτήν την στιγμή η αγορά για το σύστημα μας. (apothesis.lib.teicrete.gr/bitstream/)

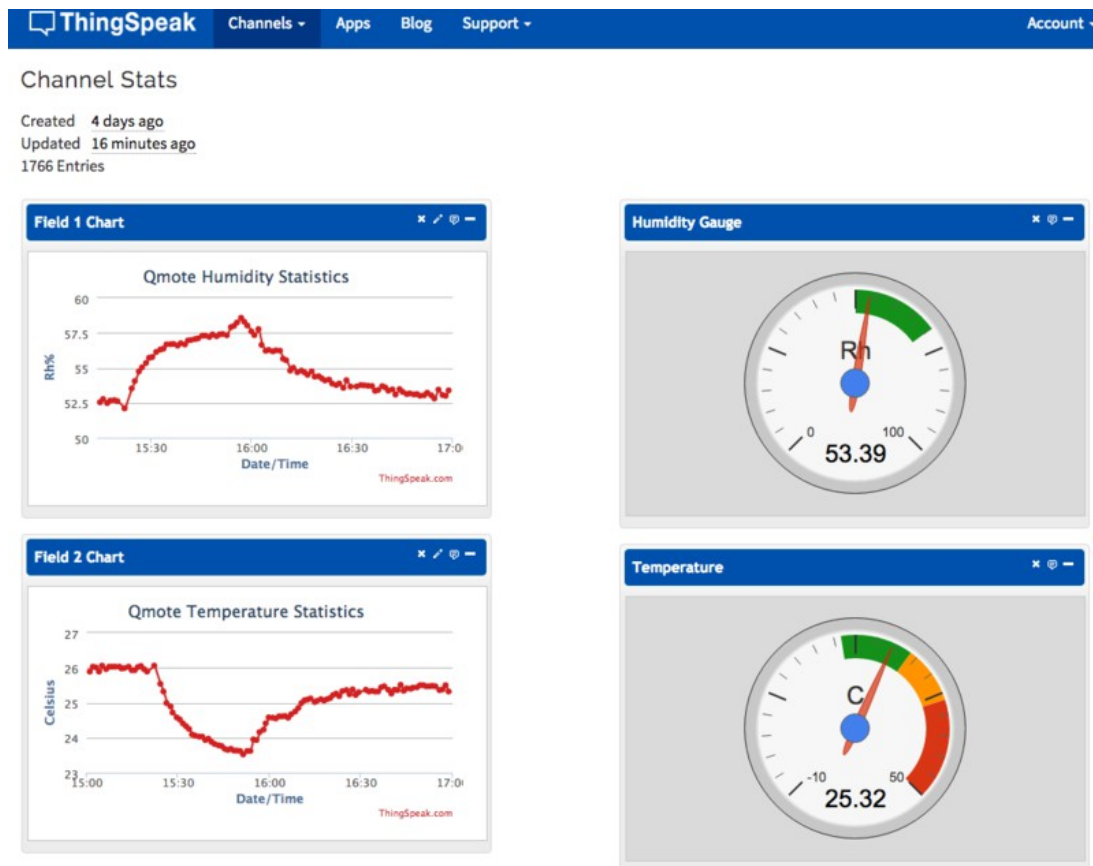
2.5 O Server ThingSpeak

Το Thingspeak είναι μια Internet of Things (IoT) πλατφόρμα που μας επιτρέπει να συλλέγουμε και από αποθηκεύουμε τα δεδομένα των αισθητηρίων μας σε ένα σύννεφο και να δημιουργήσουμε IoT εφαρμογές. Η ThingSpeak™ IoT πλατφόρμα μας παρέχει εφαρμογές που μας επιτρέπουν να αναλύσουμε και να οπτικοποιήσουμε τα δεδομένα μας στο MATLAB®, και στην συνέχεια να ενεργήσουμε κατά βούληση πάνω σε αυτά. Τα δεδομένα των αισθητηρίων μπορούν να σταλούν στο ThingSpeak από Arduino®, Raspberry Pi™, BeagleBone Black και άλλους εξοπλισμούς.



Εικόνα 17: Block διάγραμμα του ThingSpeak (ThingSpeak.com)

Με την χρήση της MATLAB Analysis εφαρμογής μπορούμε να αναλύσουμε δεδομένα όπως πχ να υπολογίζουμε την μέση υγρασία, το σημείο υγροποίησης ή να εξαλείψουμε ακραία δεδομένα από ένα ThingSpeak™ κανάλι, με την χρήση των MATLAB συναρτήσεων. Μετά την ανάλυση, μπορούμε να γράψουμε δεδομένα σε ένα κανάλι ή να δημιουργήσουμε μια οπτικοποίηση τους, όπως περιγράφετε στην MATLAB Visualizations εφαρμογή.



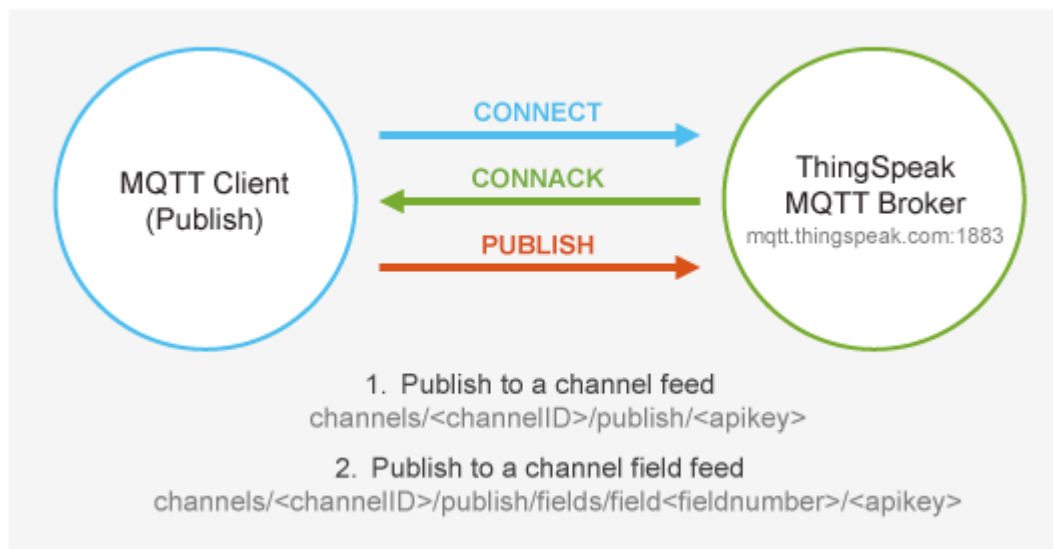
Εικόνα 18: Κανάλι απο το ThingSpeak (ThingSpeak.com)

Χρησιμοποιούμε την MATLAB Visualizations εφαρμογή για οπτικοποιήσουμε τα δεδομένα μας σε ένα ThingSpeak κανάλι. Μπορούμε να δούμε και να περιηγηθούμε στα δεδομένα μας με την χρήση διαδραστικών οπτικοποιήσεων, όπως γραφικές παραστάσεις, διαγράμματα ροής ή διαγράμματα διασποράς σε στατιστική οπτικοποίηση, με την χρήση άλλων Matlab διαγραμμάτων. Μπορούμε επίσης να θέσουμε τις οπτικοποιήσεις αυτές δημόσιες ούτως ώστε να χρησιμοποιήσουμε τον υπερσύνδεσμο τους για να τα προσθέσουμε στην ιστοσελίδα μας.

Μπορούμε επίσης να προσθέσουμε συναρτήσεις σε Scripts στο MATLAB Analysis και σε Visualization εφαρμογές ώστε να πετύχουμε τεχνικές τμηματικού προγραμματισμού.

Το ThingSpeak είναι μια IoT πλατφόρμα που χρησιμοποιεί κανάλια για την αποθήκευση δεδομένων που αποστέλλονται από εφαρμογές ή συσκευές. Με τις ρυθμίσεις που περιγράφονται στο Channel Configurations, δημιουργούμε ένα κανάλι και στην συνέχεια λαμβάνουμε ή στέλνουμε δεδομένα από και προς αυτό. Μπορούμε επίσης να θέσουμε το κανάλι μας δημόσιο για να δημοσιεύσουμε τα δεδομένα μας.

Με λίγα λόγια το κανάλι μας αποτελεί ένα REST Service. Χρησιμοποιώντας REST API calls όπως GET, POST, PUT και DELETE μπορούμε να δημιουργήσουμε ένα κανάλι και να πάρουμε το περιεχόμενο του, να ενημερώσουμε το περιεχόμενό του, να καθαρίσουμε το περιεχόμενό του ή να διαγράψουμε το κανάλι αυτό. Μπορούμε επίσης να χρησιμοποιήσουμε MQTT τεχνικές για να ενημερώσουμε το περιεχόμενο ενός καναλιού. Με μια μικρή έρευνα μπορούν να καταλήξουμε για το αν θέλουμε να χρησιμοποιήσουμε ένα REST API ή ένα MQTT API, με βάση το τι μας ταιριάζει καλύτερα.



Εικόνα 19: MQTT API.(sideshare)

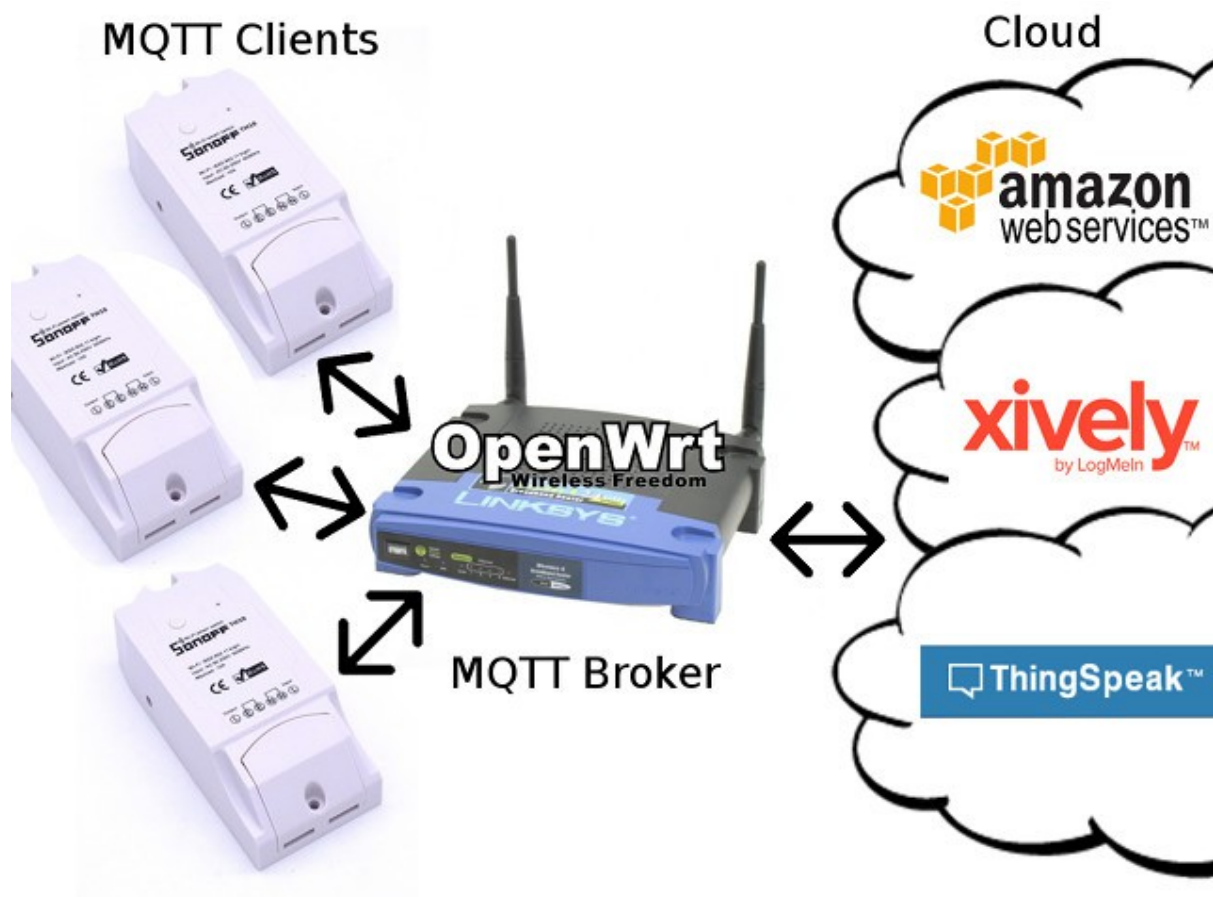
Το MATLAB analysis και οι Visualization εφαρμογές μας επιτρέπουν να περιηγηθούμε και να διαβάσουμε στα δεδομένα του καναλιού μας. Το ThingSpeak μας επιτρέπει να αλληλοεπιδράσουμε με social media, web services ή και άλλες εφαρμογές.

Το Representational state transfer (REST) είναι ένα αρχιτεκτονικό πρότυπο σχεδιασμένο ως ένα request/response μοντέλο επικοινωνίας πάνω σε HTTP πρωτόκολλο. Το ThingSpeak χρησιμοποιεί

REST API κλήσεις όπως GET, POST, PUT, και DELETE για να δημιουργήσει ένα κανάλι και να ενημερώσει το περιεχόμενό του, να ενημερώσει ένα υπάρχον κανάλι, να καθαρίσει το περιεχόμενο ενός καναλιού και να διαγράψει το κανάλι. Μέσω του φυλλομετρητή μας ή μέσω client αιτούμαστε από τον server χρησιμοποιώντας τα REST calls και αυτός ανταποκρίνεται με μια απάντηση. Οι φυλλομετρητές χρησιμοποιούν αυτόν τον τρόπο για να ανακτούν ιστοσελίδες ή να στείλουν δεδομένα σε απομακρυσμένους servers.

Μπορούμε να χρησιμοποιήσουμε τα REST calls αυτά μαζί με τα ThingSpeak apps τα οποία μας επιτρέπουν να αλληλοεπιδράσουμε με social media, web services ή και άλλες εφαρμογές.

Μπορούμε να χρησιμοποιήσουμε το MQTT API για να ενημερώσουμε τα ThingSpeak™ κανάλια μας. Η ThingSpeak IoT πλατφόρμα επιτρέπει στους clients να ενημερώνουν και να λαμβάνουν ενημερώσεις από τα δεδομένα των καναλιών μέσω ενός ThingSpeak MQTT broker. Το MQTT είναι ένα πρωτόκολλο επικοινωνίας βασισμένο στην publish/subscribe λογική το οποίο χρησιμοποιεί TCP/IP sockets ή WebSockets. Η χρήση του MQTT μέσω του WebSockets μπορεί να ασφαλιστεί με SSL.

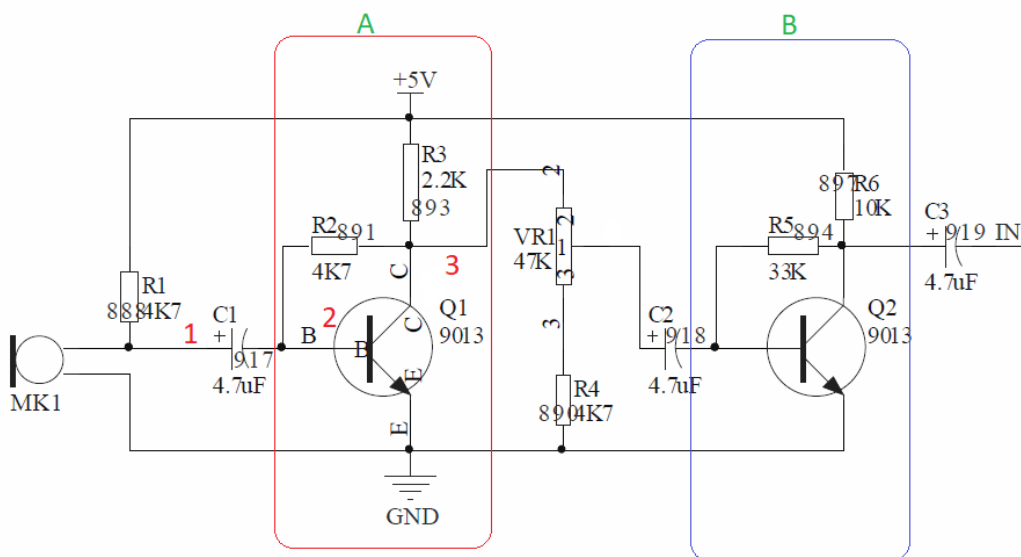


Εικόνα 20: Οπτική Αναπαράσταση λειτουργίας MQTT API(iot.do)

Μια client συσκευή συνδέεται με τον MQTT Broker και έχει την δυνατότητα να δημοσιεύσει (publish) σε ένα κανάλι ή να εγγραφεί σε αυτό (subscribe) με σκοπό να λαμβάνει αυτόματα ότι ενημερώσεις γίνονται σε αυτό.

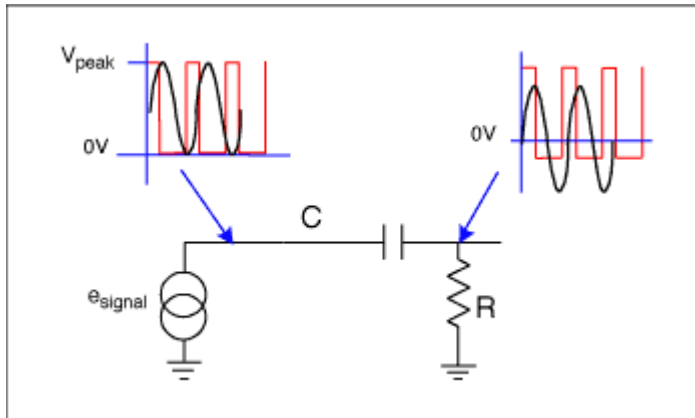
ΚΕΦΑΛΑΙΟ 3

3.1 Κύκλωμα προενίσχυσης



Εικόνα 21: Το πρώτο και το δεύτερο κύκλωμα προενίσχυσης της κατασκευής.

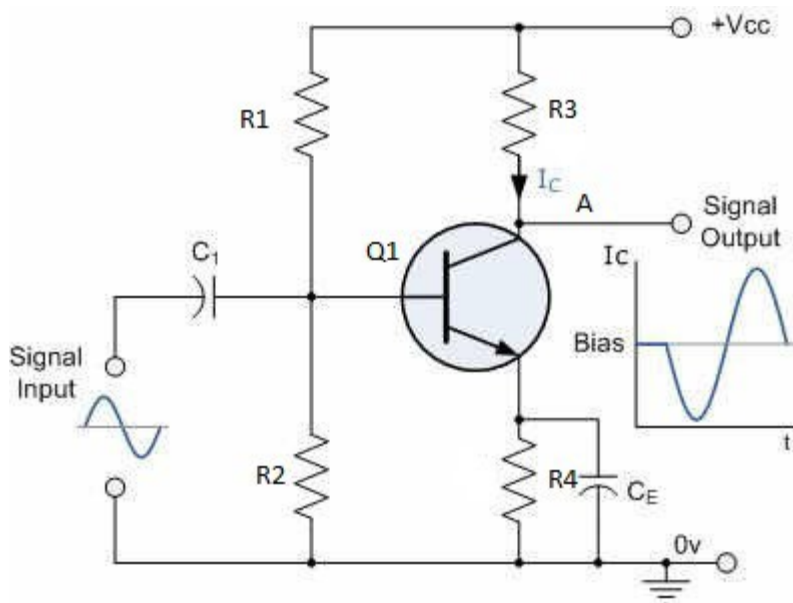
Στο παραπάνω σχήμα το (MK1) είναι ένα μικρόφωνο τύπου (electret) ο συγκεκριμένος τύπος μικροφώνου ονομάζεται έτσι λόγω της μεμβράνης που έχει εσωτερικά το μικρόφωνο και η οποία πρέπει να πολωθεί μέσω μιας αντίστασης της R1 στην τάση τροφοδοσίας +5V στην συνέχεια στο σημείο (1) υπάρχει το σήμα που έρχεται από το μικρόφωνο και στην συνέχεια υπάρχει ένας πυκνωτής C1 ο οποίος υποβιβάζει την κυματομορφή του μικροφώνου από τα 5V στα 0V.



Εικόνα 22:Υποβιβασμός κυματομορφής (DC Bias).

3.2 Προενίσχυση

Στην συνέχεια στο σημείο (2) έχουμε την πρώτη φάση προενίσχυσης όπου βασίζεται στην αρχή λειτουργίας ενός ενισχυτή κλάσης A.



Εικόνα 23:Διάγραμμα ενισχυτή κλάσης A.

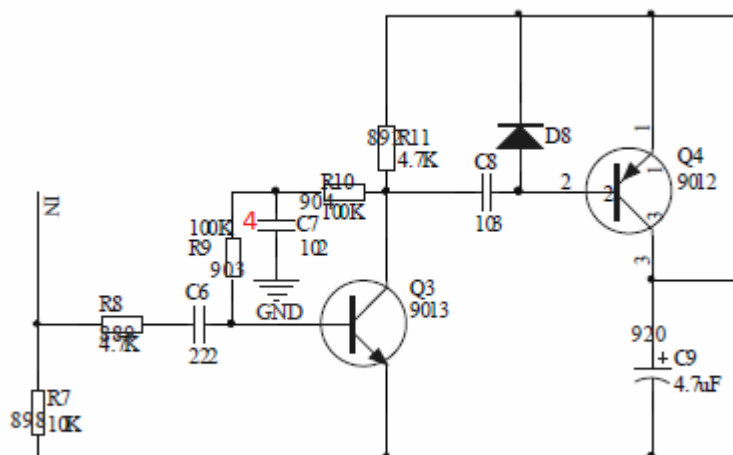
Από τον πυκνωτή (C1) έχουμε την είσοδο όπου γίνεται το (AC coupling) ώστε η κυματομορφή να κεντραριστεί στο μηδέν και στη συνέχεια έχουμε έναν διαιρέτη τάσης που σχηματίζουν οι αντιστάσεις

(R1,R2) που συνδέονται στην τροφοδοσία οπότε έχουμε μια πόλωση της βάσης του (Q1) και στο σημείο (A) παίρνουμε την έξοδο του σήματος όπου είναι πιστή αντιγραφή της εισόδου ίδια συχνότητα άλλα σήμα μεγαλύτερου πλάτους αφού προηγήθηκε ενίσχυση του σήματος. Συνηθίζεται παράλληλα στην R4 να συνδέεται και ένας πυκνωτής ο (CE) στο παραπάνω σχήμα ο οποίος αυξάνει το κέρδος.

Σε αντίθεση με το παραπάνω διάγραμμα ενός ενισχυτή κλάσης A στο κύκλωμα μας έχουμε 2 αντιστάσεις την (R2) και (R3) που είναι ίδιας φιλοσοφίας με τον κλάσης A ενισχυτή αλλά δεν κάνουν τόσο πιστή αντιγραφή του αρχικού σήματος όμως ο λόγος που δεν μας επηρεάζει την κατασκευή είναι γιατί έχουμε να ελέγξουμε ένα απλό φωτορυθμικό οπότε δεν μας επηρεάζει και τόσο η πίστη αντιγραφή και θα ήταν επιπλέον κόστος η προσθήκη εξαρτημάτων χωρίς λόγο.

Από την έξοδο της πρώτης βαθμίδας ενίσχυσης το σήμα πηγαίνει στο ποτενσιόμετρο όπου ρυθμίζουμε την απολαβή που θα δεχτεί ως είσοδο η δεύτερη βαθμίδα ενίσχυσης η οποία είναι ακριβώς ίδια με την πρώτη, έχουμε πάλι έναν πυκνωτή (C2) που κάνει το (AC coupling) και τέλος έχουμε πάλι έναν πυκνωτή τον (C3) που κάνει πάλι (AC coupling) για την επόμενη βαθμίδα ενίσχυσης.

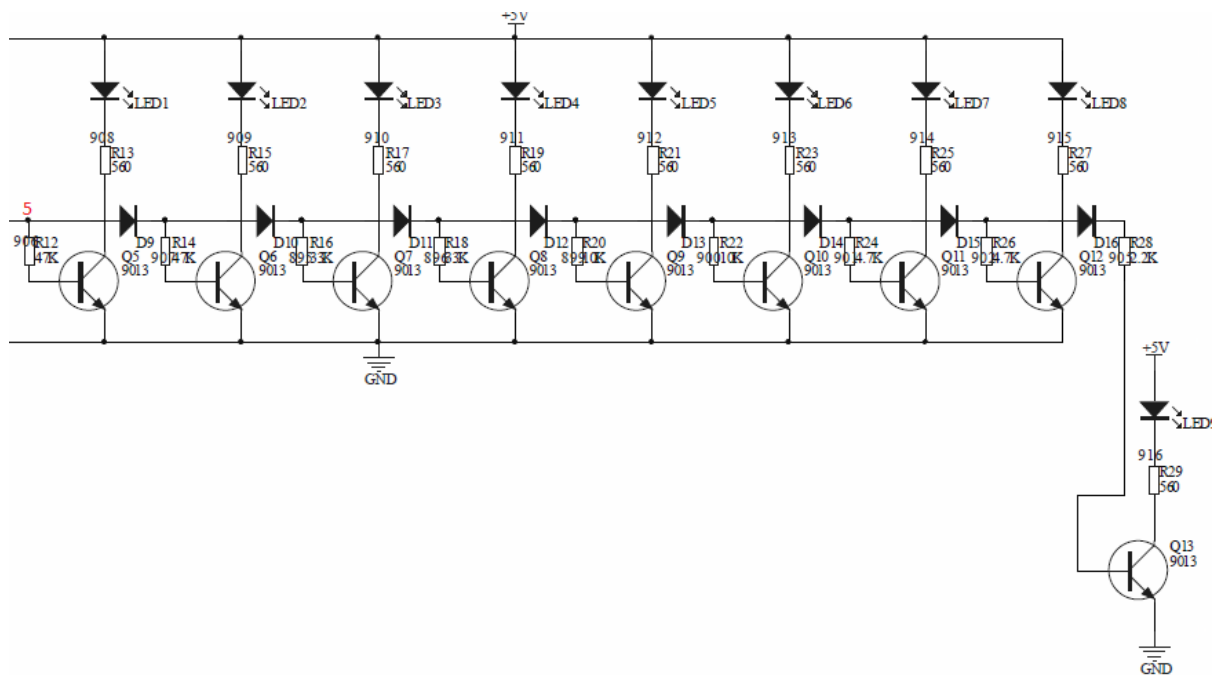
Εδώ πάλι βρίσκουμε την ίδια διάταξη ενίσχυσης με την μόνη διαφορά ότι στο σημείο 4 έχουμε τοποθετήσει έναν πυκνωτή τον C7 όπου συνδέεται με την γείωση, Ο λόγος που τον τοποθετήσαμε είναι για να προσθέσουμε μια μικρή αδράνεια στο σύστημα ώστε να υπάρχει καθυστέρηση στην είσοδο και την έξοδο.



Εικόνα 24:Διάταξη ενίσχυσης με αδράνεια.

Στην συνέχεια έχουμε τον πυκνωτή (C8) για το (AC coupling) και την είσοδο στην τελευταία βαθμίδα ενίσχυσης όπου εδώ έχουμε προσθέσει μια δίοδο την (D8) ο λόγος που υπάρχει είναι για να προστατέψει το τρανζίστορ (Q4) καθώς εάν αυξηθεί η τάση πάνω από 5.6 V στην γραμμή 2 τότε θα αρχίσει να άγει η (D8). Τέλος παρατηρούμε ότι στον συλλέκτη του (Q4) αντί για αντίσταση έχουμε τοποθετήσει τον πυκνωτή (C9) ο πυκνωτής στο AC λειτουργεί σαν αντίσταση όταν σταματήσει ο ήχος. Το (C9) θα έχει κρατήσει φορτίο οπότε θα αποφορτίζεται σταδιακά και έτσι θα έχουμε ένα πιο ομαλό οπτικό εφέ από το να σβήσει αμέσως η μπάρα. Ο λόγος που έχουμε το μοναδικό (pnp) τρανζίστορ (Q4) στο κύκλωμα ενώ όλα τα άλλα είναι (npn) οπότε όπως βλέπουμε συνδέεται και ανάποδα δηλαδή η έξοδος κάτω και το φορτίο από επάνω το κάνουμε γιατί όταν είναι απενεργοποιημένο το τρανζίστορ η τάση στο σημείο 3 εφόσον είναι αποφορτισμένος και ο πυκνωτής (C9) θα μηδενίσει οπότε δεν θα ανάβει καμία μπάρα. Αν είχαμε τοποθετήσει (npn) τρανζίστορ θα είχαμε τα αντίστροφα αποτελέσματα δηλαδή όταν είχαμε ησυχία που ουσιαστικά μεταφράζεται σε μηδενική είσοδο θα άναβαν όλες οι μπάρες και όσο ανέβαινε η ένταση του ήχου θα έσβησαν.

3.3 Διάταξη των LED



Εικόνα 25: Διάγραμμα της διάταξης των Led.

Τέλος έχουμε τα led που ανάβουν τις μπάρες του Πλέξιγκλας στην συνέχεια έχουμε μια σειρά τρανζίστορ και διόδους.

Έστω ότι έχουμε στο σημείο της (5) μια τάση 2V το κάθε τρανζίστορ χρειάζεται 0.5~0.6 V για να αρχίσει να άγει. Τα 2V είναι αρκετά πάνω από αυτήν την τιμή οπότε το πρώτο τρανζίστορ ενεργοποιείται και το πρώτο led ανάβει. Στην επόμενη βαθμίδα έχουμε ένα την δίοδο (D9) η οποία έχει μια πτώση τάσης 0.6V οπότε στο επόμενο σημείο (6) που είναι η βάση του δεύτερου led θα έχουμε τάση 1,4V οπότε θα ενεργοποιηθεί και αυτό το led αυτό θα συνεχίσει να γίνεται έως ότου η τάση γίνει 0,2V στο σημείο (D11) οπότε τα επόμενα led δεν θα ανάψουν. Το συμπέρασμα του παραπάνω παραδείγματος είναι πως όσο πιο μεγάλης έντασης ήχο λαμβάνει το μικρόφωνο τόσο μεγαλύτερη τάση θα έχουμε στην είσοδο των led και άρα τόσο περισσότερα led θα ανάψουν.

3.4 Ανάλυση Κώδικα Συστήματος

Όπως αναφέρθηκε παραπάνω, το σύστημά μας αποτελείται από ένα Arduino board, ένα ESP8266 NodeMcu και ένα ESP-01 ESP8266 serial. Τόσο το Arduino board όσο και το NodeMcu λειτουργούν ανεξάρτητα και τελούν το καθένα μια ξεχωριστή λειτουργία.

Σε γενικές γραμμές, σκοπός του NodeMcu είναι να επικοινωνεί με την android εφαρμογή μας με σκοπό να μας επιτρέπει να χειριζόμαστε απομακρυσμένα το φωτορυθμικό μας, ενώ επίσης στέλνει τις τιμές που παίρνει από τον χρήστη έτσι ώστε να ενημερώνεται άμεσα ο χρήστης για την κατάσταση του φωτορυθμικού. Το ESP-01 από την άλλη, είναι υπεύθυνο για την επικοινωνία του συστήματός μας με τον έξω κόσμο, ή αλλιώς με την πλατφόρμα ThingSpeak, όπου γίνεται η καταγραφή των εξόδων που δίνει το σύστημά μας.

Σε αυτό το σημείο θα γίνει η ανάλυση του κάθε κώδικα ξεχωριστά για να γίνει κατανοητό το πως δουλεύει το σύστημα αυτό.

3.5 NodeMcu

Αρχικά το module μας τροφοδοτείται με ενέργεια από τον Arduino. Με αυτόν τον τρόπο έχουμε μια πυγή ενέργειας, την συνδέουμε στον Arduino και έτσι τροφοδοτείτε όλο μας το σύστημα.

Ο κώδικας με τον οποίο λειτουργεί το NodeMcu είναι σχετικά απλός. Αρχικά, θέλουμε το module μας να συνδεθεί στο τοπικό μας δίκτυο. Του παρέχουμε ένα ssid και έναν κωδικό για τον σκοπό αυτό. Στην συνέχεια, για να συνδέσουμε το σύστημα μας με την εφαρμογή virtuino παρέχουμε επιπλέον μια πόρτα σύνδεσης και έναν κωδικό ασφαλείας. Αφού εκτελεστεί το πρόγραμμα, το NodeMcu θα λάβει ένα μοναδικό δικό του IP. Έτσι, για να συνδεθούμε στο σύστημά μας από το κινητό μας θα πρέπει να παρέχουμε το σωστό IP, την σωστή πόρτα και τον σωστό κωδικό ασφαλείας.

Τέλος, στο NodeMcu έχουμε συνδέσει την έξοδο του φωτορυθμικού και ένα LED με σκοπό να μας ενημερώνει για τυχών υπερβολική αύξηση της έντασης του ήχου.

Για να καταλάβουμε πόσο απλή είναι η χρήση του virtuino αρκεί να δούμε την loop() συνάρτηση του προγράμματος μας. Μέσα σε μια γραμμή (99) λέμε στο σύστημα μας να κρατήσει ανοικτή την επικοινωνία με το virtuino. Έτσι μπορούμε να ελέγχουμε το φωτορυθμικό μας, να παίρνουμε δεδομένα από το σύστημα και πολλά άλλα.

3.6 Arduino και ESP-01

Μιας και το σύστημα Arduino + ESP-01 είναι σαφώς πιο σύνθετο από ένα απλό NodeMcu, και καθώς ο κώδικας τον οποίο τρέχει ο Arduino είναι εκτενέστερος, θα ήταν καλό να αναλύσουμε μερικούς όρους πρώτου αναφερθούμε στην λειτουργικότητά του.

Αρχικά, το ESP-01 επικοινωνεί με το ThingSpeak μέσω http πρωτοκόλλου. Για να ενημερώσουμε το κανάλι μας με δεδομένα, χρησιμοποιούμε ένα POST request. Το request αυτό αποτελείται από headers και body, εμείς θα επικεντρωθούμε στο body. Το body του request μας είναι ένα data entity το οποίο περιέχει τα δεδομένα τα οποία θέλουμε να στείλουμε στο ThingSpeak. Η μορφή η οποία πρέπει να έχει το data entity είναι γνωστή ως Key-Value, δηλαδή θέτουμε ένα όνομα για την τιμή που θέλουμε να στείλουμε και διπλά βάζουμε την τιμή της.

Εμείς στο Request αυτό φορτώνουμε το μοναδικό κλειδί του καναλιού μας και τις τιμές των διαγραμμάτων που θα ενημερώσουμε. Το ThingSpeak στην συνέχεια θα συλλέξει τα δεδομένα που θα του στείλουμε και θα ενημερώσει με την σειρά του τα διαγράμματα μας.

Έτσι λοιπόν, αφού συνδέσουμε το ESP με το δίκτυο μας, ανοίγουμε μια TCP σύνδεση με το API του ThingSpeak. Όποτε θελήσουμε να στείλουμε δεδομένα, θα πραγματοποιήσουμε την διαδικασία που περιγράψαμε παραπάνω. Τα δεδομένα τα οποία αναλύουμε είναι τα αποτελέσματα τα οποία παίρνουμε από τον αισθητήρα καπνού τον οποίο έχουμε βάλει στο σύστημα μας. Σε περίπτωση που ο αισθητήρας μας αναγνωρίσει τιμές οι οποίες είναι μη φυσιολογικές τότε θέτουμε σε λειτουργία ένα LED λαμπάκι.

3.7 Γενικότερη δομή του Κώδικα

Καθώς η φιλοσοφία του Arduino και ο κώδικάς του είναι βασισμένα γλώσσα προγραμματισμού C, δεν πρέπει παρά να εκμεταλλευτούμε το γεγονός αυτό και να δομήσουμε τον κώδικά μας αναλόγως. Πρώτων και κυριότερων, σημαντική είναι η χρήση συναρτήσεων. Εκτός του ότι διατηρούν τον κώδικά μας ευανάγνωστο, δομημένο και εύκολο στην συντήρησή του, θα μας βοηθήσει αρκετά στην εξέλιξη του συστήματός μας.

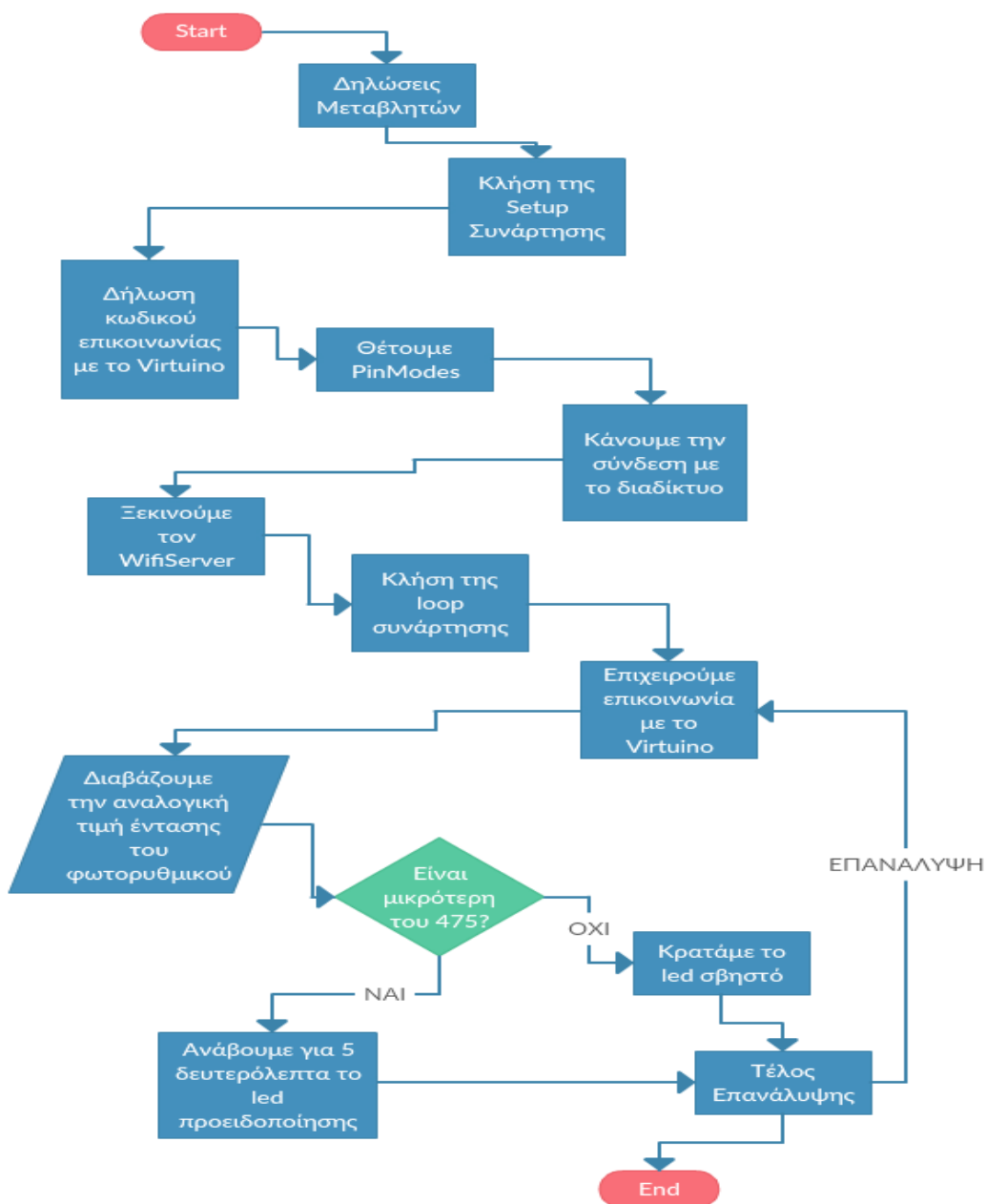
Γενικά, ένας κώδικας Arduino χωρίζεται σε 2 μέρη, την εκκίνηση και την επανάληψη. Επιπλέον στον κώδικά μας υπάρχουν πάνω πάνω οι δηλώσεις βιβλιοθηκών και μεταβλητών και οι επιμέρους συναρτήσεις τις οποίες έχουμε για να οργανώσουμε καλύτερα το πρόγραμμά μας. Με σωστή σχεδίαση πριν την συγγραφή του προγράμματος, η κατανόηση μέχρι και ενός πολύ σύνθετου προγράμματος γίνεται πολύ απλή.

3.8 Διαγράμματα Ροής

Η λειτουργία του συστήματος μας θα μπορούσε να περιγραφεί σε δύο διαγράμματα ροής, ένα για το NodeMcu και ένα για το Arduino.

Αρχικά στο NodeMcu δηλώνουμε τις μεταβλητές τις οποίες χρειαζόμαστε για το πρόγραμμα μας. Μέσα στην συνάρτηση Setup γίνεται η σύνδεση με το Wi-Fi, δηλώνουμε τα pinModes και ετοιμάζουμε την σύνδεση με το Virtuino. Στην loop συνάρτηση γίνεται η επικοινωνία του NodeMcu με το Virtuino και μετράμε την ένδειξη του αισθητήρα ήχου τον οποίο έχουμε.

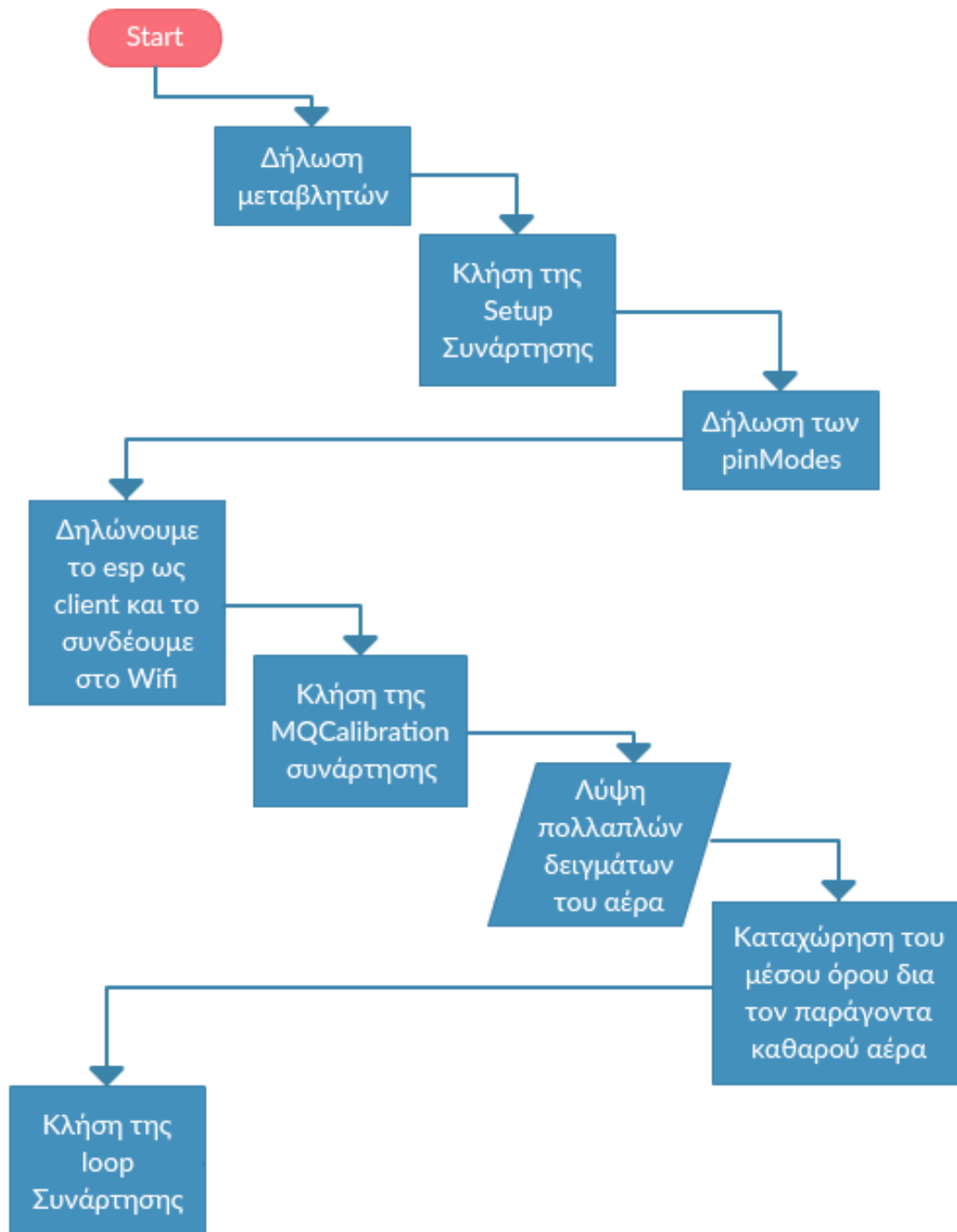
Σχηματικά, το διάγραμμα ροής είναι το παρακάτω.



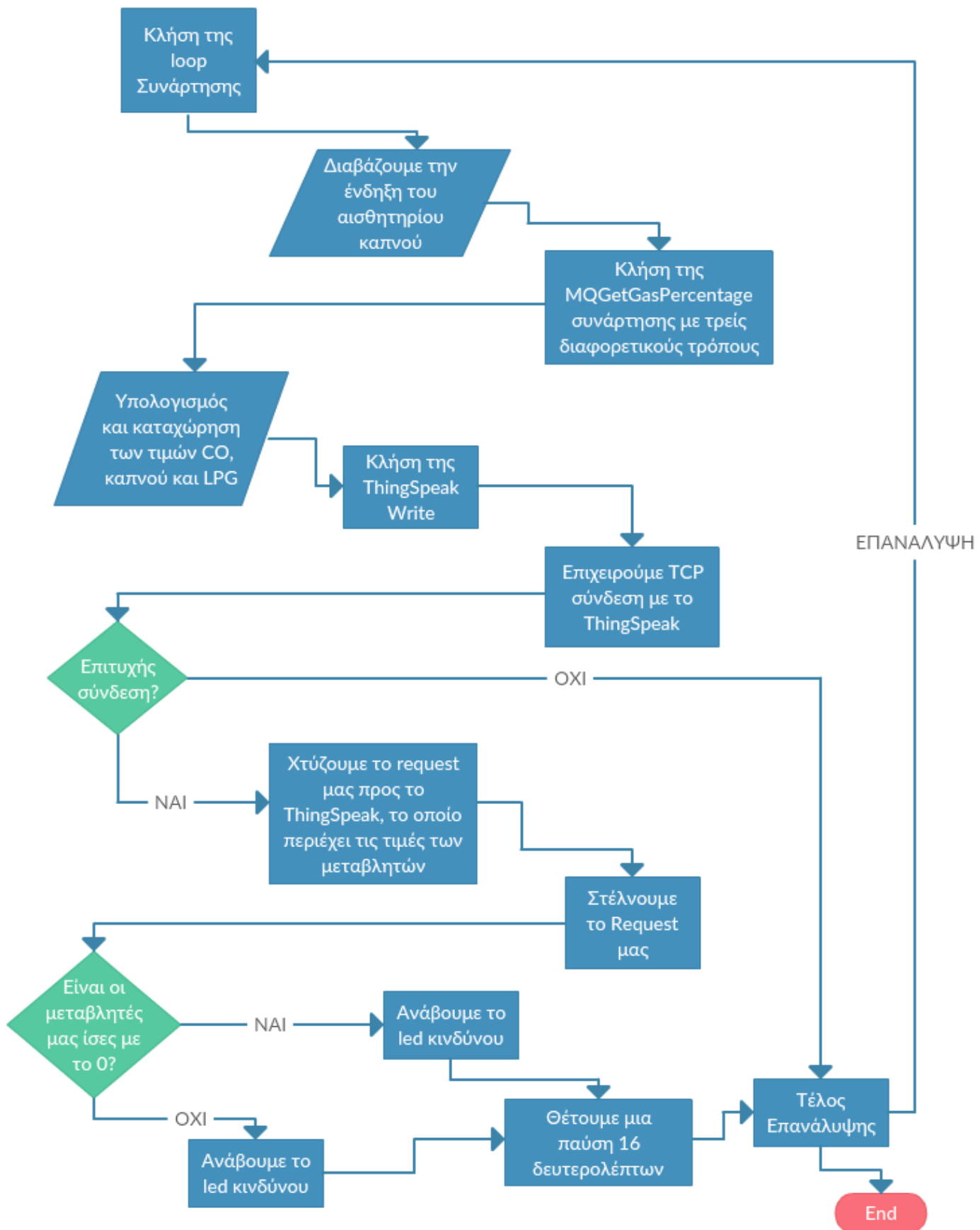
Διάγραμμα ροής NodeMcu.

Στην περίπτωση του Arduino, αρχίζουμε όπως και στο NodeMcu με την δήλωση των μεταβλητών που χρησιμοποιεί το πρόγραμμά μας. Στην συνέχεια καλούμε την συνάρτηση Setup. Εκεί δηλώνουμε τα pinModes, συνδέουμε το esp-01 με το Wi-Fi και κάνουμε το calibration του αισθητήρα καπνού. Στην συνέχεια προχωράμε στην loop συνάρτηση. Εκεί λαμβάνουμε την τιμή του αισθητήριου καπνού και με βάση αυτήν μπορούμε να υπολογίσουμε τα επίπεδα CO, καπνού και LPG στον αέρα. Στέλνουμε τις τιμές αυτές στο ThingSpeak και στην συνέχεια, ανάλογα με τις τιμές που λάβαμε, ανάβουμε και το κατάλληλο led ενδείξεως κατάστασης.

Μια αποικόνιση του διαγράμματος ροής για τον Arduino είναι η παρακάτω.



Διάγραμμα ροής Arduino Α μέρος.



Διάγραμμα ροής Arduino B μέρος.

ΚΕΦΑΛΑΙΟ 4 (ΚΑΤΑΣΚΕΥΗ)

4.1 Περιγραφή της Κατασκευής

Η Κατασκευή αποτελείται από τα παρακάτω 6 βασικά κομμάτια που παρουσιάζονται στην εικόνα (29) στα οποία γίνεται αναλυτική περιγραφή των χαρακτηριστικών τους στην συνέχεια του κεφαλαίου.



1. Φωτορυθμικό



2. Arduino Uno R3



3. MQ-2 Gas Sensor



4. WiFi Module - ESP8266

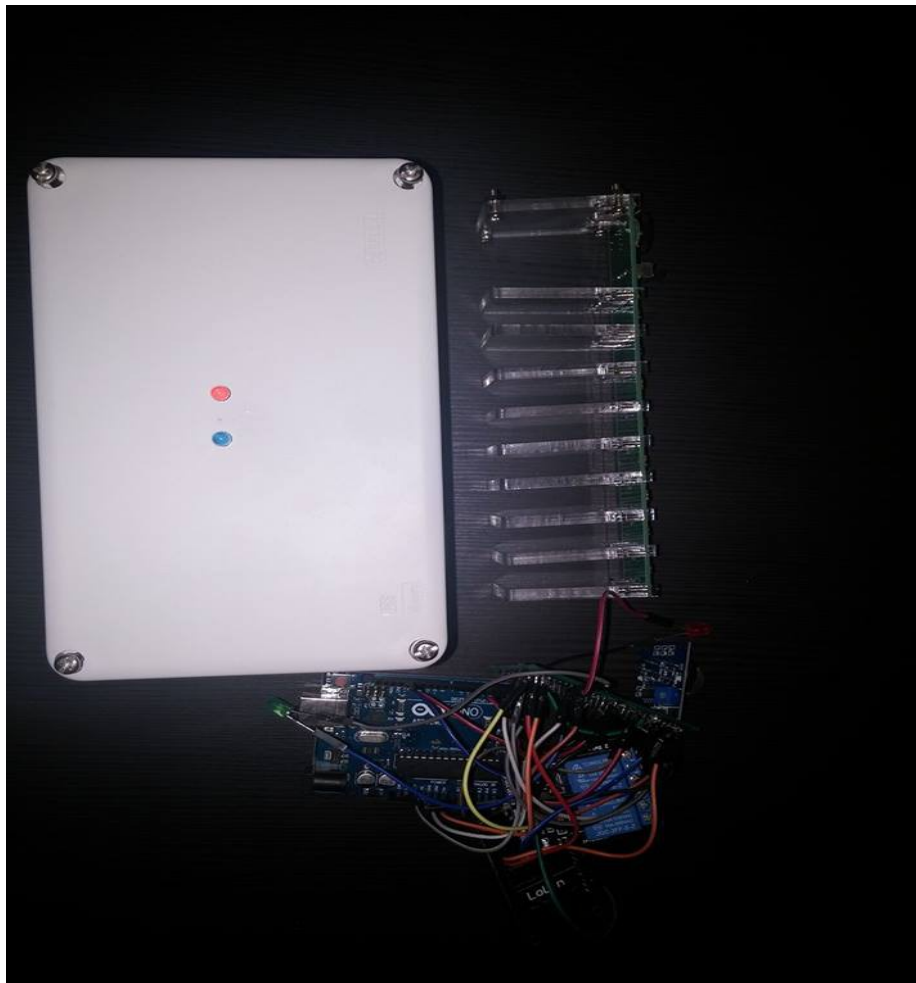


5. NodeMCU-based
ESP8266 development kit



6. Two Relay Module
5V, 10A

Εικόνα 26 Τα βασικά τμήματα της κατασκευής.



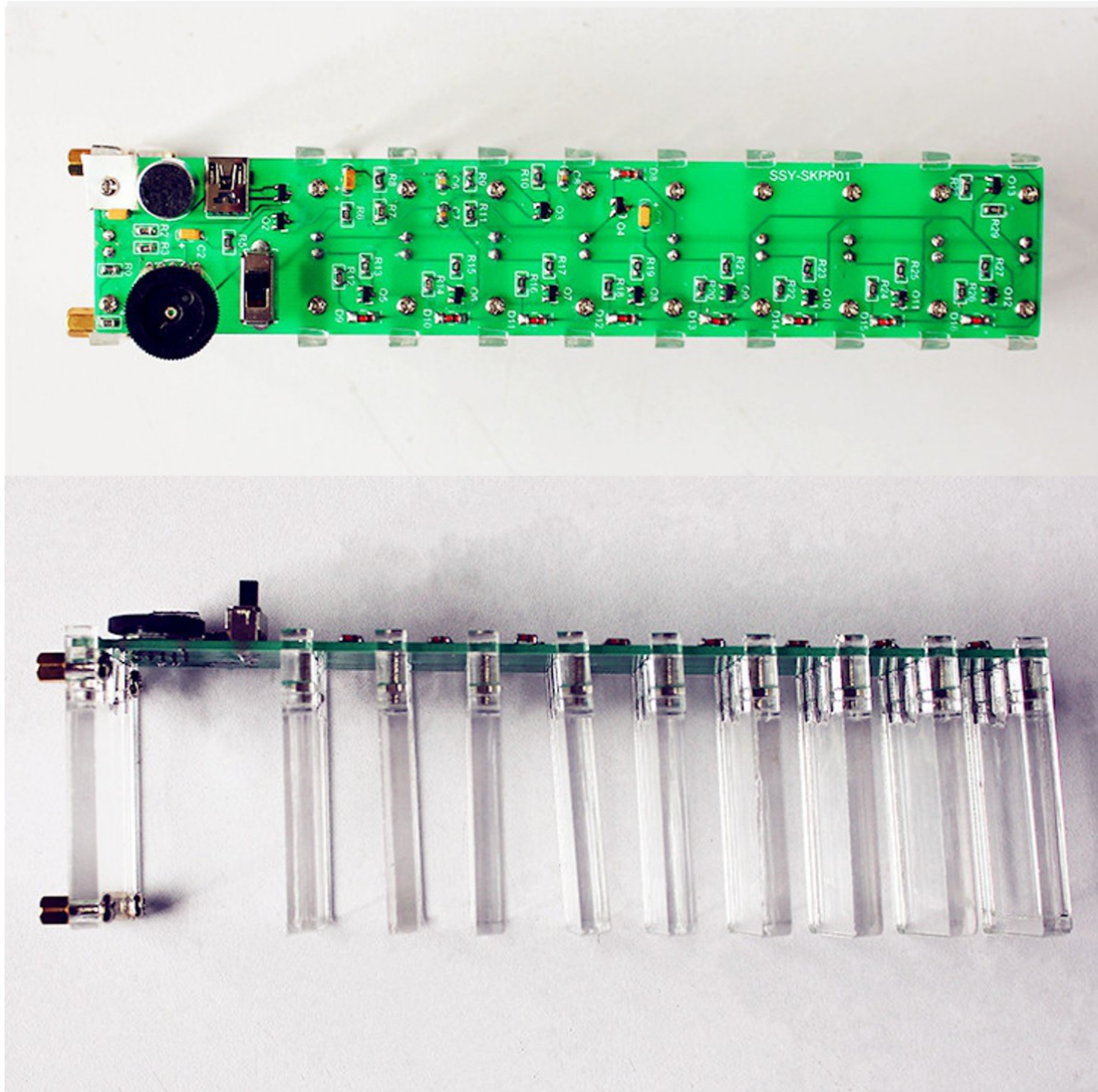
Εικόνα 27 Η κατασκευή.

Συμπεράσματα: Η κατασκευή αποτελεί μια φτηνή και ικανοποιητική λύση καθώς με φτηνά και εύκολα στην εύρεση υλικά επιτύχαμε το επιθυμητό αποτέλεσμα για τις απαιτήσεις μιας πτυχιακής εργασίας. Έχουμε μια άμεση απόκριση του φωτορυθμικού σε εξωτερικά ερεθίσματα. Τα δεδομένα από το περιβάλλον που λαμβάνουν τα αισθητήρια αποστέλλονται με επιτυχία μέσω δικτύου Wifi και στην εφαρμογή android αλλά και στον IoT server (thingspeak.com).

Αυτό που παρατηρήσαμε μετά από αρκετά πειράματα είναι ότι η κατασκευή έχει ορισμένες αποκλίσεις στις μετρήσεις των διαφόρων αέριων και αυτό οφείλεται στο φτηνό αισθητήριο (MQ2) που επιλέξαμε για να κάνουμε τις μετρήσεις μας. Σε επίπεδο software ορισμένες φορές έχουμε αδυναμία στο connectivity και αυτό οφείλεται κυρίως στην πλατφόρμα του arduino IDE και το μικρό σε κόστος hardware (esp8266 & arduino uno). Το arduino IDE είναι μια open source πλατφόρμα που βελτιώνεται από τους χρηστές και δεν παρέχεται από πουθενά εγγύηση καλής λειτουργίας καθώς δεν κοστίζει. Οπότε δεν έχουμε πάντα το επιθυμητό αποτέλεσμα λόγω δημιουργίας διαφόρων bugs και loops που μπορεί να κολλήσει ο κώδικας.

Σε γενικές γραμμές πρόκειται για μια αξιόπιστη κατασκευή που για το μικρό της κόστος τα αποτελέσματα είναι αρκετά ικανοποιητικά.

4.1.1 Φωτορυθμικό



Εικόνα 28 Το φωτορυθμικό της κατασκευής.

Το πρώτο και βασικό κομμάτι της κατασκευής είναι το φωτορυθμικό του οποίου η λειτουργία έχει περιγραφή αναλυτικά στο κεφάλαιο 3.

Τα υλικά που χρησιμοποιήθηκαν για την κατασκευή του φωτοριθμικού αναλύονται στον παρακάτω πίνακα:

ΕΞΑΡΤΗΜΑ	ΤΙΜΗ	ΣΥΜΒΟΛΟ	ΠΟΣΟΤΗΤΑ
ΑΝΤΙΣΤΑΣΗ	4.7K	R1/R2/R4/R8/R11/R24/R26	7
ΑΝΤΙΣΤΑΣΗ	2.2K	R3/R28	2
ΑΝΤΙΣΤΑΣΗ	33K	R5/R16/R18	3
ΑΝΤΙΣΤΑΣΗ	10K	R6/R7/R20/R22	4
ΑΝΤΙΣΤΑΣΗ	100K	R9/R10	2
ΑΝΤΙΣΤΑΣΗ	47K	R12/R14	2
ΑΝΤΙΣΤΑΣΗ	560	R0/R13/R15/R17/R19/R21 /R23/R25/R27/R29	10
ΠΟΤΕΝΣΙΟΜΕΤΡΟ	100K	VR1	1
ΠΥΚΝΩΤΗΣ ΤΑΝΤΑΛΙΟΥ	4.7μF	C1/C2/C3/C9	4
ΠΥΚΝΩΤΗΣ	2.2nF	C6	1
ΠΥΚΝΩΤΗΣ	1nF	C7	1
ΠΥΚΝΩΤΗΣ	10nF	C8	1
ΜΙΚΡΟΦΩΝΟ	-	MK1	1
ΤΡΑΝΖΙΣΤΟΡ	NPN	Q1/Q2/Q3/Q5/Q6/Q7/Q8 / Q9/Q10/Q11/Q12/Q13	14
ΤΡΑΝΖΙΣΤΟΡ		Q4	1

	PNP		
LED	ΜΠΛΕ	LE0/LED1/LED2/LED3/ LED8/LED9	6
LED	ΚΟΚΚΙΝΟ	LED4/LED5/LED6/LED7	4
ΔΙΑΚΟΠΤΗΣ ΔΙΟΔΟΣ	-	-	1
	-	D8/D9/D10/D11/D12/D13/D14 / D15/D16	9
USB PORT	-	P1	1

4.1.2 Χρήση του Arduino UnoR3

Το δεύτερο κομμάτι της κατασκευής το οποίο αποτελεί και την καρδιά του συστήματος καθώς επεξεργάζεται όλα τα σήματα εισόδου και δημιουργεί κατάλληλα σήματα εξόδου είναι ο Arduino Uno R3. Λαμβάνει όλα τα σήματα από τα αισθητήρια όπως το MQ-2 και ενεργοποιεί το relay module όταν αυτό απαιτείται.

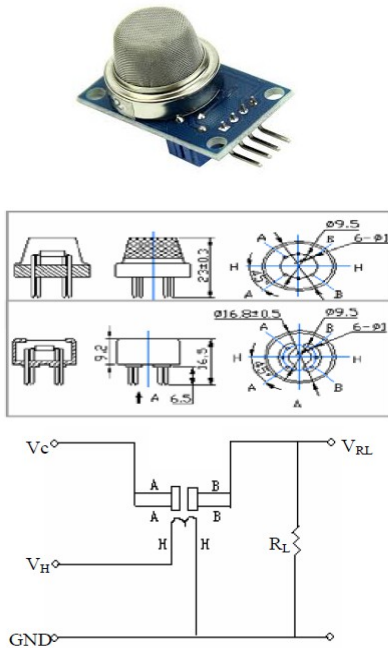
Το arduino είναι υπεύθυνο και για την επικοινωνία του συστήματος με το internet και την εφαρμογή android καθώς στέλνει τις απαραίτητες πληροφορίες στο Esp8266 και στο NodeMCU.



Εικόνα 29 Arduino Uno R3.

4.1.3 Αισθητήριο MQ-2

Το τρίτο κομμάτι της κατασκευής είναι το αισθητήριο MQ-2 Gas Sensor το οποίο αφού γίνει calibration μέσω του προγράμματος που έχουμε δημιουργήσει στέλνει τις μετρήσεις που λαμβάνει από το εξωτερικό περιβάλλον από υγραέριο, προπάνιο και υδρογόνο στον arduino.

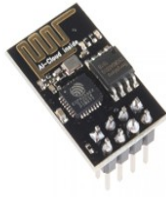


Model No.		MQ-2	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite (Black Bakelite)	
Detection Gas		Combustible gas and smoke	
Concentration		300-10000ppm (Combustible gas)	
Circuit	Loop Voltage	V_c	$\leq 24V$ DC
	Heater Voltage	V_H	$5.0V \pm 0.2V$ AC or DC
	Load Resistance	R_L	Adjustable
Character	Heater Resistance	R_H	$31\Omega \pm 3\Omega$ (Room Tem.)
	Heater consumption	P_H	$\leq 900mW$
	Sensing Resistance	R_s	$2K\Omega - 20K\Omega$ (in 2000ppm C_2H_6)
	Sensitivity	S	$R_s(\text{in air})/R_s(1000ppm \text{ isobutane}) \geq 5$
	Slope	α	$\leq 0.6 (R_{s000ppm}/R_{s000ppm} CH_4)$
Condition	Tem. Humidity	$20^\circ C \pm 2^\circ C$; $65\% \pm 5\% RH$	
	Standard test circuit	$V_c: 5.0V \pm 0.1V$; $V_H: 5.0V \pm 0.1V$	
	Preheat time	Over 48 hours	

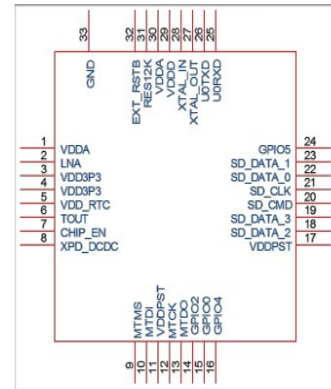
Εικόνα 30 Χαρακτηριστικά του αισθητηρίου MQ-2 Ga Sensor.

4.1.4 Ολοκληρωμένο ESP8266

Το τέταρτο κομμάτι της κατασκευής είναι το ολοκληρωμένο ESP8266 το οποίο είναι υπεύθυνο να στέλνει τις πληροφορίες από τις μετρήσεις που έχουν γίνει στον IOT server.



Categories	Items	Values
WiFi Parameters	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocles	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
802.11 n: -72 dbm (MCS7)		
Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip	
Hardware Paramaters	Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	5x5mm
	External Interface	N/A

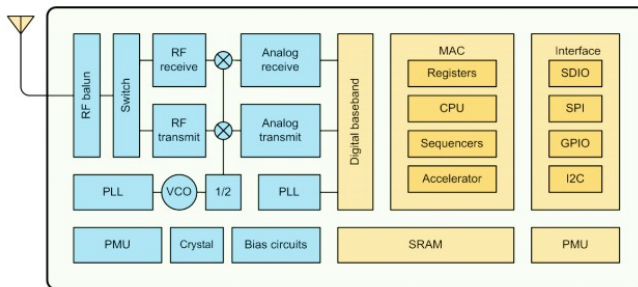


Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0 ~3.6V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50Ω No matching required but we recommend that the n-type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0~3.6V
4	VDD3P3	P	Amplifier Power 3.0~3.6V
5	VDD_RTC	P	NC (1.1V)

Εικόνα 31 Χαρακτηριστικά του ολοκληρωμένο ESP8266.

4.1.5 Ολοκληρωμένο NodeMCU

Το πέμπτο κομμάτι της κατασκευής είναι το ολοκληρωμένο NodeMCU το οποίο δέχεται εντολή μέσω της εφαρμογής android και ενεργοποιεί το relay module.



Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0 ~3.6V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50Ω No matching required but we recommend that the n-type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0~3.6V
4	VDD3P3	P	Amplifier Power 3.0~3.6V
5	VDD_RTC	P	NC (1.1V)

Parameters	Conditions	Min	Typical	Max	Unit
Storage Temperature Range		-40	Normal	125	°C
Maximum Soldering Temperature	IPC/JEDEC J-STD-020			260	°C
Working Voltage Value		3.0	3.3	3.6	V
I/O	V_{IL}/V_{IH}	-0.3/0.75V _{IO}		0.25V _{IO} /3.6	V
	V_{OL}/V_{OH}	N/0.8V _{IO}		0.1V _{IO} /N	V
	I _{MAX}			12	mA
Electrostatic Discharge (HBM)	TAMB=25°C			2	KV
Electrostatic Discharge (CDM)	TAMB=25°C			0.5	KV

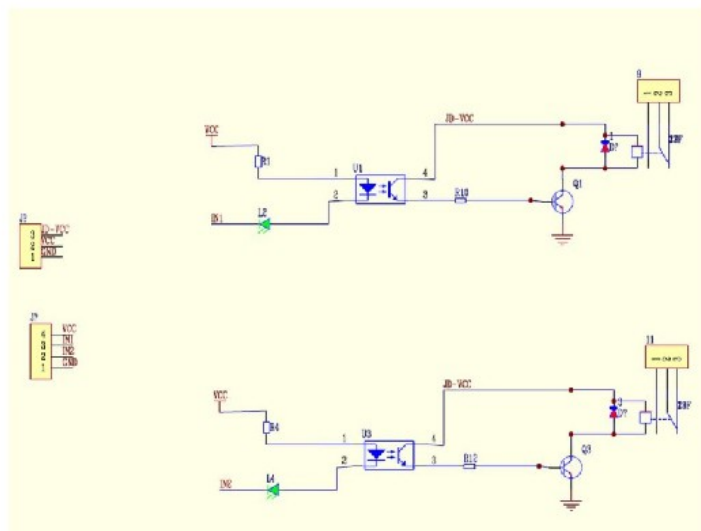
Εικόνα 32 Χαρακτηριστικά του ολοκληρωμένο NodeMCU.

4.1.6 Two Relay Module

Το έκτο κομμάτι της κατασκευής είναι το Two Relay το οποίο χρειάζεται μόνιμη τροφοδοσία 5V που την παίρνει από τον arduino και ένα σήμα εισόδου που το δέχεται από το NodeMCU ώστε να ενεργοποιηθούν οι επαφές του και να πάρει τροφοδοσία το φωτορυθμικό.



1. VCC: 5V DC
2. COM: 5V DC
3. IN1: high/low output
4. IN2: high/low output
5. GND: ground



Εικόνα 33 Χαρακτηριστικά του Relay Module.

4.2 Προγραμματισμός

4.2.1 Κώδικας Arduino Uno R3

Αρχίζουμε με την δήλωση των μεταβλητών που χρησιμοποιεί το πρόγραμμά μας. Στην συνέχεια καλούμε την συνάρτηση Setup. Εκεί δηλώνουμε τα pinModes, συνδέουμε το esp-01 με το Wi-Fi και κάνουμε το calibration του αισθητήρα καπνού. Στην συνέχεια προχωράμε στην loop συνάρτηση. Εκεί λαμβάνουμε την τιμή του αισθητηρίου καπνού και με βάση αυτήν μπορούμε να υπολογίσουμε τα επίπεδα CO, καπνού και LPG στον αέρα. Στέλνουμε τις τιμές αυτές στο ThingSpeak και στην συνέχεια, ανάλογα με τις τιμές που λάβαμε, ανάβουμε και το κατάλληλο led ενδείξεως κατάστασης.

Ακολουθεί ο κώδικας του Arduino:

```
/* Arduino Code By zafirelis
```

```
*Arduino+esp8266 thingSpeak example
```

```
* Download latest Virtuino android app from the link:https://play.google.com/store/apps/details?id=com.virtuino\_automations.virtuino&hl
```

```
*/
```

```
/******Demo for MQ-2 Gas Sensor Module V1.0*****
```

```
Support: Tiequan Shao: support\[at\]sandboxelectronics.com
```

```
Lisence: Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)
```

Note: This piece of source code is supposed to be used as a demonstration ONLY. More sophisticated calibration is required for industrial field application.

```
Sandbox Electronics 2011-04-25
```

```
*****
```

```
**/
```

```
//SMOKE SENSOR
```

```

#define    MQ_PIN            (0) //define which analog input channel you are going to use

#define    RL_VALUE          (5) //define the load resistance on the board, in kilo ohms

#define    RO_CLEAN_AIR_FACTOR    (9.83) //RO_CLEAN_AIR_FACTOR=(Sensor resistance in
clean air)/RO,

                                //which is derived from the chart in datasheet

/*****Software Related Macros*****/

#define    CALIBRATION_SAMPLE_TIMES    (50) //define how many samples you are going to
take in the calibration phase

#define    CALIBRATION_SAMPLE_INTERVAL    (500) //define the time interval(in milisecond)
between each samples in the

                                //calibration phase

#define    READ_SAMPLE_INTERVAL    (50) //define how many samples you are going to take
in normal operation

#define    READ_SAMPLE_TIMES    (5) //define the time interval(in milisecond) between each
samples in

                                //normal operation

/*****Application Related Macros*****/

#define    GAS_LPG            (0)

#define    GAS_CO              (1)

#define    GAS_SMOKE          (2)

/
*****Globals*****/
/

float    LPGCurve[3] = {2.3,0.21,-0.47}; //two points are taken from the curve.

                                //with these two points, a line is formed which is "approximately
equivalent"

                                //to the original curve.

```

```

//data format:{ x, y, slope}; point1: (lg200, 0.21), point2: (lg10000,
-0.59)

float    COCurve[3] = {2.3,0.72,-0.34}; //two points are taken from the curve.

//with these two points, a line is formed which is "approximately
equivalent"

//to the original curve.

//data format:{ x, y, slope}; point1: (lg200, 0.72), point2: (lg10000,
0.15)

float    SmokeCurve[3] =(2.3,0.53,-0.44); //two points are taken from the curve.

//with these two points, a line is formed which is "approximately
equivalent"

//to the original curve.

//data format:{ x, y, slope}; point1: (lg200, 0.53), point2: (lg10000,
-0.22)

float    Ro    = 10;    //Ro is initialized to 10 kilo ohms

// Code to use SoftwareSerial

#include <SoftwareSerial.h>

SoftwareSerial espSerial = SoftwareSerial(2,3); // arduino RX pin=2 arduino TX pin=3 connect
the arduino RX pin to esp8266 module TX pin - connect the arduino TX pin to esp8266 module RX
pin

String apiKey = "BBKFOG0IGMIO62M3"; // replace with your channel's thingspeak WRITE API key

String ssid="zafirelis wifi"; // Wifi network SSID

String password ="zafirelispassword"; // Wifi network password

boolean DEBUG=true;

```

```
//=====
showResponse

void showResponse(int waitTime){

    long t=millis();

    char c;

    while (t+waitTime>millis()){

        if (espSerial.available()){

            c=espSerial.read();

            if (DEBUG) Serial.print(c);

        }

    }

}

```

```
//=====

boolean thingSpeakWrite(float value1, float value2, float value3){

    String cmd = "AT+CIPSTART=\"TCP\", \"";          // TCP connection
    cmd += "184.106.153.149";                       // api.thingspeak.com
    cmd += "\",80";

    espSerial.println(cmd);

    if (DEBUG) Serial.println(cmd);

    if(espSerial.find("Error")){

        if (DEBUG) Serial.println("AT+CIPSTART error");

        return false;

    }

}

```



```

String getStr = "GET /update?api_key="; // prepare GET string
getStr += apiKey;

getStr += "&field1=";
getStr += String(value1);
getStr += "&field2=";
getStr += String(value2);
getStr += "&field3=";
getStr += String(value3);
getStr += "\r\n\r\n";

// send data length
cmd = "AT+CIPSEND=";
cmd += String(getStr.length());
espSerial.println(cmd);
if (DEBUG) Serial.println(cmd);

delay(100);
espSerial.print(getStr);
if (DEBUG) Serial.print(getStr);
return true;
}

//===== SMOKE

/***** MQResistanceCalculation *****/
Input: raw_adc - raw value read from adc, which represents the voltage

```

Output: the calculated sensor resistance

Remarks: The sensor and the load resistor forms a voltage divider. Given the voltage across the load resistor and its resistance, the resistance of the sensor could be derived.

```
*****  
**/  
  
float MQResistanceCalculation(int raw_adc)  
{  
    return ( ((float)RL_VALUE*(1023-raw_adc)/raw_adc));  
}
```

```
/****** MQCalibration  
*****
```

Input: mq_pin - analog channel

Output: Ro of the sensor

Remarks: This function assumes that the sensor is in clean air. It use

MQResistanceCalculation to calculates the sensor resistance in clean air and then divides it with RO_CLEAN_AIR_FACTOR. RO_CLEAN_AIR_FACTOR is about 10, which differs slightly between different sensors.

```
*****  
**/  
  
float MQCalibration(int mq_pin)  
{  
    int i;  
    float val=0;  
  
    for (i=0;i<CALIBARAION_SAMPLE_TIMES;i++) { //take multiple samples  
        val += MQResistanceCalculation(analogRead(mq_pin));  
        delay(CALIBRATION_SAMPLE_INTERVAL);  
    }  
}
```

```

}

val = val/CALIBARAION_SAMPLE_TIMES;           //calculate the average value

val = val/RO_CLEAN_AIR_FACTOR;               //divided by RO_CLEAN_AIR_FACTOR yields the Ro
                                              //according to the chart in the datasheet

return val;
}

```

```

/***** MQRead
*****

```

Input: mq_pin - analog channel

Output: Rs of the sensor

Remarks: This function use MQResistanceCalculation to caculate the sensor resistenc (Rs).

The Rs changes as the sensor is in the different consentration of the target gas. The sample times and the time interval between samples could be configured by changing the definition of the macros.

```

*****/

```

```

float MQRead(int mq_pin)
{
  int i;
  float rs=0;

  for (i=0;i<READ_SAMPLE_TIMES;i++) {
    rs += MQResistanceCalculation(analogRead(mq_pin));
    delay(READ_SAMPLE_INTERVAL);
  }
}

```

```
rs = rs/READ_SAMPLE_TIMES;
```

```
return rs;
```

```
}
```

```
/****** MQGetGasPercentage  
*****
```

Input: rs_ro_ratio - Rs divided by Ro

gas_id - target gas type

Output: ppm of the target gas

Remarks: This function passes different curves to the MQGetPercentage function which
calculates the ppm (parts per million) of the target gas.

```
*****  
**/
```

```
int MQGetGasPercentage(float rs_ro_ratio, int gas_id)
```

```
{
```

```
if ( gas_id == GAS_LPG ) {
```

```
return MQGetPercentage(rs_ro_ratio,LPGCurve);
```

```
} else if ( gas_id == GAS_CO ) {
```

```
return MQGetPercentage(rs_ro_ratio,COCurve);
```

```
} else if ( gas_id == GAS_SMOKE ) {
```

```
return MQGetPercentage(rs_ro_ratio,SmokeCurve);
```

```
}
```

```
return 0;
```

```
}
```

```
/****** MQGetPercentage *****
```

Input: rs_ro_ratio - Rs divided by Ro

pcurve - pointer to the curve of the target gas

Output: ppm of the target gas

Remarks: By using the slope and a point of the line. The x(logarithmic value of ppm)

of the line could be derived if y(rs_ro_ratio) is provided. As it is a

logarithmic coordinate, power of 10 is used to convert the result to non-logarithmic

value.

```
*****
```

```
**/
```

```
int MQGetPercentage(float rs_ro_ratio, float *pcurve)
```

```
{
```

```
return (pow(10,((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) + pcurve[0]));
```

```
}
```

```
//=====
```

```
setup
```

```
void setup() {
```

```
    //LEDS
```

```
    pinMode(12, OUTPUT);
```

```
    pinMode(13, OUTPUT);
```

```
    DEBUG=true;    // enable debug serial
```

```
    Serial.begin(9600);
```

```
    espSerial.begin(115200); // enable software serial
```

```
        // Your esp8266 module's speed is probably at 115200.
```

```
        // For this reason the first time set the speed to 115200 or to your esp8266 configured speed
```

// and upload. Then change to 9600 and upload again

```
//espSerial.println("AT+RST"); // Enable this line to reset the module;
```

```
//showResponse(1000);
```

```
//espSerial.println("AT+UART_CUR=9600,8,1,0,0"); // Enable this line to set esp8266 serial speed  
to 9600 bps
```

```
//showResponse(1000);
```

```
espSerial.println("AT+CWMODE=1"); // set esp8266 as client
```

```
showResponse(1000);
```

```
espSerial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\""); // set your home router SSID and  
password
```

```
showResponse(5000);
```

```
if (DEBUG) Serial.println("Setup completed");
```

```
Serial.print("Calibrating...\n");
```

```
Ro = MQCalibration(MQ_PIN); //Calibrating the sensor. Please make sure the sensor is  
in clean air
```

```
//when you perform the calibration
```

```
Serial.print("Calibration is done...\n");
```

```
Serial.print("Ro=");
```

```
Serial.print(Ro);
```

```
Serial.print("kohm");
```

```
Serial.print("\n");
```

```

}

// ===== loop
void loop() {

  int var1 = MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_LPG);
  int var2 = MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_CO);
  int var3 = MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_SMOKE);

  thingSpeakWrite(var1, var2, var3);

  if(var1!=0 || var2 !=0 || var3!=0){
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
    Serial.print("TRUE");
    Serial.print("\n");
  }
  else{
    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);
    Serial.print("FALSE");
    Serial.print("\n");
  }

  // thingspeak needs 15 sec delay between updates,
  delay(16000);
}

```

```
}
```

4.2.2 Κωδικας NodeMcu

Αρχικά στο NodeMcu δηλώνουμε τις μεταβλητές τις οποίες χρειαζόμαστε για το πρόγραμμα μας. Μέσα στην συνάρτηση Setup γίνεται η σύνδεση με το Wi-Fi, δηλώνουμε τα pinModes και ετοιμάζουμε την σύνδεση με το Virtuino. Στην loop συνάρτηση γίνεται η επικοινωνία του NodeMcu με το Virtuino και μετράμε την ένδειξη του αισθητήρα ήχου τον οποίο έχουμε.

Ακολουθεί ο κώδικας του NodeMCU:

```
/* Virtuino NodeMCU ESP8266 web server and WiFi access point example No1

* Created by nikos zafirelis

* Before running this code config the settings below as the instructions on the right

*

*

* Download latest Virtuino android app from the link: https://play.google.com/store/apps/details?id=com.virtuino\_automations.virtuino&hl=el

*/

/*===== Virtuino General methods

*

* void vDigitalMemoryWrite(int digitalMemoryIndex, int value)    write a value to a Virtuino digital
memory (digitalMemoryIndex=0..31, value range = 0 or 1)

* int vDigitalMemoryRead(int digitalMemoryIndex)                read the value of a Virtuino digital
memory (digitalMemoryIndex=0..31, returned value range = 0 or 1)

* void vMemoryWrite(int memoryIndex, float value);              write a value to Virtuino memory
(memoryIndex=0..31, value range as float value)

* float vMemoryRead(int memoryIndex);                            read a value of Virtuino memory
(memoryIndex=0..31, returned a float value)

* run();                                                         necessary command to communicate with Virtuino android
app (on start of void loop)

* int getPinValue(int pin);                                       read the value of a Pin. Usefull to read the value of
a PWM pin
```



```
* long lastCommunicationTime;           Stores the last communication with Virtuino
time
*/
```

```
// 1. On Arduino IDE software select the board NodeMCU. If the board isn't in boards list, open the
boards manager, on the search line enter esp8266 and download all ESP8266 boards
```

```
// 2. Open the Virtuino library file: VirtuinoESP.h, enable the line #define NODEMCU and disable
the other boards
```

```
// 3. Connect the NODEMCU ESP8266 module to usb and upload this code
```

```
#include <ESP8266WiFi.h>
```

```
#include "Virtuino_ESP_WifiServer.h"
```

```
int MicPin = A0;
```

```
int MicValue1 = 0;
```

```
int ledPin = D1;
```

```
const char* ssid = "G3_2109";
```

```
const char* password = "1234567890";
```

```
WiFiServer server(8000);           // Server port
```

```
Virtuino_ESP_WifiServer virtuino(&server);
```

```
//NodeMCU ESP8266 pins= D0,D1,D2,D3,D4,D5,D6,D7,D8,D9,D10
```

```
void setup() {
```

```
    //----- Virtuino settings
```

```

virtuino.DEBUG=true;           // set this value TRUE to enable the serial monitor status

virtuino.password="1234";      // Set a password to your web server for more protection

                                // avoid special characters like ! $ = @ # % & * on your password. Use only
                                // numbers or text characters

Serial.begin(9600);           // Enable this line only if DEBUG=true

delay(10);

pinMode(MicPin, INPUT);

pinMode(ledPin, OUTPUT);

//----- NodeMCU module settings

//----- prepare GPIO2

pinMode(2, OUTPUT);

digitalWrite(2, 0);

//---- 1. Settings as Station - Connect to a WiFi network

Serial.println("Connecting to "+String(ssid));

// If you don't want to config IP manually disable the next four lines

// IPAddress ip(192, 168, 2, 150);    // where 150 is the desired IP Address

// IPAddress gateway(192, 168, 2, 1); // set gateway to match your network

// IPAddress subnet(255, 255, 255, 0); // set subnet mask to match your network

// WiFi.config(ip, gateway, subnet); // If you don't want to config IP manually disable this line

WiFi.mode(WIFI_STA);          // Config module as station only.

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

```

```

    delay(500);

    Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println(WiFi.localIP());

//---- 2. Settings as Access point - Create a private Wifi Network. Enable the next five lines to use
module as Acces point

// Serial.print("Setting soft-AP ... ");          // Default IP: 192.168.4.1

// WiFi.mode(WIFI_AP);                          // Config module as Acces point only. Set
WiFi.mode(WIFI_AP_STA); to config module as Acces point and station

// boolean result = WiFi.softAP("NodeMCU", "12345678"); // SSID: NodeMCU
Password:12345678

// if(result == true) Serial.println("Server Ready");

// else Serial.println("Failed!");

// ---- Start the server

server.begin();

Serial.println("Server started");

//---- Enter your setup code below

pinMode(D3,OUTPUT); // connect a relay or a led to this pin. On Virtuino panel add a switch
to pin D3

}

void loop() {

```

```
virtuino.run();

MicValue1 = analogRead(MicPin);

delay(100);
Serial.println(MicValue1);

if(MicValue1<=475){
  digitalWrite(ledPin, HIGH);
  delay(5000);
}
else{
  digitalWrite(ledPin, LOW);
}

}
```

4.3 Κόστος Κατασκευής

ΕΞΑΡΤΗΜΑΤΑ	ΚΟΣΤΟΣ
Arduino Uno R3	15€
NodeMCU	15€
ESP8266 WIFI	10€
MQ-2 Gas Sensor	10€
2 Module Relay	15€
Φωτορυθμικό	35€
Μπουατ, διάτρητη πλαकेτα και διάφορα	30€
Συνολικό κόστος	130€

Βιβλιογραφία

Arampatzis, Th.& Lygeros, J. (2005). A Survey of Applications of Wireless Sensors and Wireless Sensor Networks.

Albert Malvino David J. Bates ΗΛΕΚΤΡΟΝΙΚΗ (7η Έκδοση)

Leach & Malvino Ψηφιακά Ηλεκτρονικά (5η Έκδοση)

ΤΗΛΕΜΑΤΙΚΗ ΚΑΙ ΝΕΕΣ ΥΠΗΡΕΣΙΕΣ - ΦΩΤΙΟΣ ΣΟΛΔΑΤΟΣ

BUILDING AND PROGRAMMING A HOUSEHOLD

APPLIANCE CONTROL CIRCUIT - Χατζηλούκας Δημήτριος

Σύστημα Βασισμένο σε Arduino για την παρατήρηση ροής υγρού - Βλαγοϊδης Νικόλαος

WEBSITES

<http://www.hlektronika.gr/> (Κοινότητα ηλεκτρονικών)

https://www.sas.com/el_gr/insights/big-data/internet-of-things.html

https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF_%CF%84%CF%89%CE%BD_%CF%80%CF%81%CE%B1%CE%B3%CE%BC%CE%AC%CF%84%CF%89%CE%BD

<https://www.arduino.cc/>

<https://www.youtube.com/channel/UCEKKd7DkhJ6fnQNX9mKvGw>

<https://grobotronics.com/esp8266-wifi-module.html?sl=el>