

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών



UNIVERSITY of WEST ATTICA
FACULTY OF ENGINEERING
Department of Electrical & Electronics Engineering

Πρόγραμμα Μεταπτυχιακών Σπουδών
Διαδικτυωμένα Ηλεκτρονικά Συστήματα

Master of Science in
Internetworked Electronic Systems

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ΠΕΙΡΑΜΑΤΙΚΗΣ ΔΙΑΤΑΞΗΣ ΕΞΥΠΝΗΣ
ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΗΣ ΒΑΛΒΙΔΑΣ ΜΕ ΤΕΧΝΟΛΟΓΙΕΣ
ΠΛΑΤΦΟΡΜΑΣ ΜΙΚΡΟΕΛΕΓΚΤΗ ΑΝΟΙΚΤΟΥ ΚΩΔΙΚΑ,
ΑΣΥΡΜΑΤΗΣ ΔΙΚΤΥΩΣΗΣ ΚΑΙ CLOUD COMPUTING

Μεταπτυχιακός Φοιτητής : ΑΝΑΣΤΑΣΙΟΣ ΑΝΑΣΤΑΣΙΑΔΗΣ, Α.Μ. IES-0026

Επιβλέπων : ΠΑΝΑΓΙΩΤΗΣ ΠΑΠΑΓΕΩΡΓΑΣ, Καθηγητής

ΑΙΓΑΛΕΩ, ΙΟΥΝΙΟΣ 2020

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών



UNIVERSITY of WEST ATTICA
FACULTY OF ENGINEERING
Department of Electrical & Electronics Engineering

Πρόγραμμα Μεταπτυχιακών Σπουδών
Διαδικτυωμένα Ηλεκτρονικά Συστήματα

Master of Science in
Internetworked Electronic Systems

MSc THESIS

DEVELOPMENT OF A CLOUD INTERCONNECTED ELECTROMAGNETIC SWITCH BASED ON AN OPEN SOURCE MICROCONTROLLER PLATFORM

Student: ANASTASIADIS ANASTASIOS, Reg. Nr. IES-0026
Project Supervisor: Prof. PAPAGEORGAS PANAGIOTIS

ATHENS-EGALEO, JUNE 2020

Πίνακας περιεχομένων

ΕΙΣΑΓΩΓΗ: Αντικείμενο, ερευνητικά ερωτήματα και διάρθρωση της εργασίας	1
ΚΕΦΑΛΑΙΟ 1 : Θεωρητικό πλαίσιο του θέματος	4
1.1 : Μικροελεγκτές	4
1.1.1: Γενικές πληροφορίες	4
1.1.2: Χρήση των μικροελεγκτών	5
1.1.3: Λειτουργικές βαθμίδες ενός μικροελεγκτή	8
1.1.4: Αρχιτεκτονική μικροελεγκτών.....	9
1.2 : Η γλώσσα προγραμματισμού C	12
1.2.1 : Γενικές πληροφορίες της γλώσσας C	12
1.2.2 : Open-source υλικό (open-source hardware).....	14
1.2.3 : Πλατφόρμες μικροελεγκτή open source.....	15
1.3 : Δικτύωση υπολογιστών	21
1.3.1 : Εισαγωγή στα δίκτυα υπολογιστών	21
1.3.2 : Υλικό δικτύων	22
1.3.3 : Ταξινόμηση δικτύων ανάλογα την έκταση και τη τοπολογία τους	22
1.3.4 : Το μοντέλο OSI του ISO	26
1.3.5 : Βασικές έννοιες Διαδικτύου (Internet)	28
1.3.6 : Ασύρματη δικτύωση.....	32
1.3.6.α : Ραδιοκύματα.....	32
1.3.6.β : Ασύρματα δίκτυα δεδομένων	33
1.3.6.γ : Ασύρματες υπηρεσίες δεδομένων	34
1.3.7 : Λειτουργία τεχνολογίας Wi-Fi.....	37
1.4 : Cloud computing.....	38
1.4.1 : Ορισμός	38
1.4.2 : Μοντέλα Cloud computing.....	39
1.4.3 : Τύποι Cloud	41
1.5 : Internet of Things (IoT)	42
1.5.1 : Ορισμός	42
1.5.2 : Ασύρματες τεχνολογίες στο IoT.....	43

1.5.3 : IoT εφαρμογές.....	45
1.5.4 : IoT πλατφόρμες (IoT platforms).....	48
1.5.4.α : Γενικές πληροφορίες IoT πλατφόρμων.....	48
1.5.4.β: Κατηγορίες IoT πλατφόρμων.....	50
1.5.5 : IoT πρωτόκολλα επικοινωνίας	51
1.5.6 : IoT και Cloud, μια τέλεια ένωση	53
ΚΕΦΑΛΑΙΟ 2 : Πειραματικό πλαίσιο του θέματος	55
1 : Πλατφόρμα Arduino.....	55
1.1: Αναπτυξιακό Arduino board NodeMCUV3 με ESP8266.....	56
2 : Module (8 Channel 5V Optical Isolated Relay Module)	60
3 : Arduino IDE.....	74
4 : Προγράμματα - εφαρμογές του NodeMCU με το 8-channel 5V.....	80
5 : Απομακρυσμένος έλεγχος μέσω Cloud.....	98
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	117
ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ.....	118

Αντικείμενο της εργασίας αυτής είναι ο **απομακρυσμένος έλεγχος ενός ηλεκτρομαγνητικού διακόπτη (ρελέ) ή βαλβίδας με χρήση ανοικτού κώδικα πλατφόρμας μικροελεγκτή, cloud computing και ασύρματης δικτύωσης**. Εξαιτίας της πολύ μεγάλης χρήσης του Διαδικτύου (Internet) και μάλιστα σε επίπεδο συσκευών όπως ασύρματα δίκτυα αισθητήρων (**Wireless Sensor Networks**), δημιουργήθηκε η ανάγκη ελέγχου διακοπών (συσκευών και συστημάτων) μέσω του Internet (**Internet of Things, IoT**) διότι το Internet μάς βοηθά να ελέγξουμε με μεγάλη ακρίβεια διακόπτες μέσω ασύρματης δικτύωσης ώστε να μην χρειάζεται η παρουσία του χρήστη. Από αυτό συνεπάγεται ότι ο χρήστης ενώ βρίσκεται μακριά από την συσκευή – σύστημα, να μπορεί εύκολα από κινητό – smartphone, tablet ή υπολογιστή να ελέγχει τους διακόπτες της συσκευής αυτής. Η βασική προϋπόθεση είναι η συσκευή να είναι συνδεδεμένη στο Internet καθώς επίσης και το κινητό ή pc που έχουμε.

Καθημερινά όλο και περισσότερα IoT συστήματα αναπτύσσονται με σκοπό να γίνεται πιο εύκολη η ζωή μας σε συνδυασμό με τις υψηλές ταχύτητες Internet. Αυτό οφείλεται στο γεγονός ότι οι ανθρώπινες ανάγκες συνεχώς αλλάζουν και πρέπει να προσαρμόζονται στις εκάστοτε συνθήκες.

Ορισμένα ερωτήματα που δημιουργούνται σχετικά με την εργασία αυτή, είναι το πώς μπορεί ο μικροελεγκτής να συντονίζει τα δεδομένα μέσω του Internet ώστε να αλλάζει κατάσταση ο διακόπτης απομακρυσμένα, ποια η καλύτερη cloud computing εφαρμογή γι' αυτό και που μπορεί να εφαρμοστεί το όλο αυτό σύστημα. Παρακάτω, υπάρχει μια φωτογραφία που απεικονίζει το σύστημα μας ώστε να είναι σαφής η λειτουργία του.



Εικόνα 1. Παράδειγμα του project

Όπως παρατηρούμε στη παραπάνω εικόνα, ο χρήστης έχει εγκατεστημένη στο κινητό του μια εφαρμογή για cloud computing και μέσω αυτής και του Διαδικτύου, ελέγχει απομακρυσμένα ένα διακόπτη. Ο διακόπτης αυτός είναι συνδεδεμένος με το Διαδίκτυο ασύρματα με τη βοήθεια ενός ασύρματου router και έτσι ανταλλάσσονται δεδομένα και πληροφορίες μεταξύ του χρήστη και του διακόπτη. Έτσι και στη δική μας περίπτωση, μέσω του κινητού μας τηλεφώνου(ή υπολογιστή), θα ελέγχουμε τη κατάσταση του διακόπτη μέσω μιας cloud εφαρμογής. Ο διακόπτης μας θα ελέγχεται από μια πλατφόρμα μικροελεγκτή με ενσωματωμένο wi-fi προσαρμογέα για να μπορεί να συνδέεται με οποιοδήποτε router.

Παρόλο που ο απομακρυσμένος έλεγχος διακόπτη μέσω cloud είναι αρκετά χρήσιμος για τον έλεγχο συστημάτων, υπάρχουν και ορισμένα μειονεκτήματα του cloud computing. Αυτά είναι :

1) **Πιθανή αργία (downtime)** : Αν ο πάροχος Internet που έχουμε, παρέχει χαμηλές ταχύτητες και αποσυνδέσεις, τότε δεν θα μπορούμε εύκολα να ελέγξουμε το διακόπτη.

2) **Ασφάλεια** : Τα δεδομένα στο cloud μπορεί να είναι προσβάσιμα από το καθένα και χρειάζονται ισχυροί κώδικες ασφαλείας.

3) **Κόστος** : Στην προσπάθεια να προσαρμόσουμε το λογισμικό στο cloud , το κόστος μεγαλώνει ανάλογα την εφαρμογή που κάνουμε.

4) **Μη ευελιξία** : Κάποιες εφαρμογές cloud είναι ασύμβατες στο να μετατρέπουν δεδομένα σε άλλα συστήματα.

Πριν ξεκινήσουμε τη διάρθρωση της εργασίας σε κεφάλαια, στο πρώτο κεφάλαιο θα περιγράψουμε και θα αναλύσουμε έννοιες όπως **μικροελεγκτές, γλώσσα προγραμματισμού C, Cloud Computing, open source πλατφόρμες μικροελεγκτών, ασύρματη δικτύωση και Internet of Things.**

ΚΕΦΑΛΑΙΟ 1:

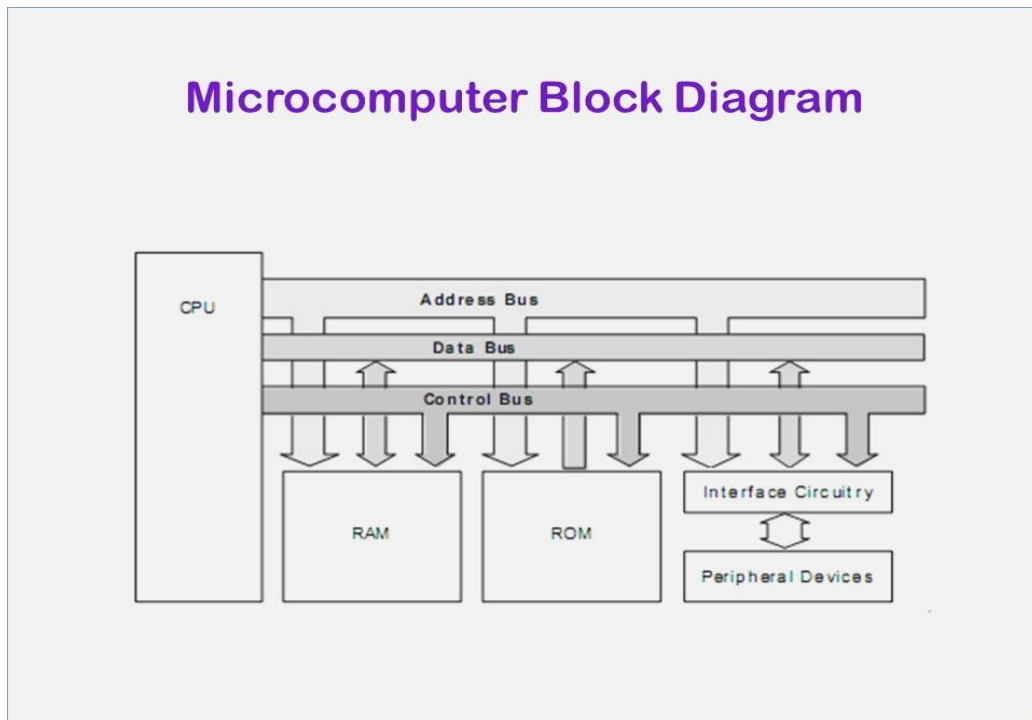
Θεωρητικό πλαίσιο του θέματος

1.1. ΜΙΚΡΟΕΛΕΓΚΤΕΣ

1.1.1. Γενικές πληροφορίες

Για να δώσουμε πλήρως τον ορισμό του μικροελεγκτή, πρέπει πρώτα να εξηγήσουμε τι είναι ο μικροϋπολογιστής. Κατ'αρχήν ξέρουμε ότι ο μικροεπεξεργαστής ή κεντρική μονάδα επεξεργασίας (C.P.U.) είναι ο πυρήνας κάθε μικροϋπολογιστικού συστήματος και είναι υλοποιημένος σε ολοκληρωμένο κύκλωμα. Αποτελείται από α) καταχωρητές γενικής και ειδικής χρήσης, β) από την αριθμητική και λογική μονάδα (A.L.U.) και γ) από την μονάδα ελέγχου. Οι καταχωρητές (registers) είναι ακολουθιακά κυκλώματα τα οποία χρησιμεύουν στην αποθήκευση και φόρτωση δεδομένων και ο αριθμός τους από C.P.U. σε C.P.U. διαφέρει. Η A.L.U. χρησιμεύει στην εκτέλεση αριθμητικών και λογικών πράξεων που ορίζονται απ'το πρόγραμμα και τέλος η μονάδα/δες ελέγχου συντονίζουν τη διαδικασία προσκόμισης των εντολών, τη διαδικασία αποκωδικοποίησης και τη παραγωγή σημάτων για την εκτέλεσή τους.

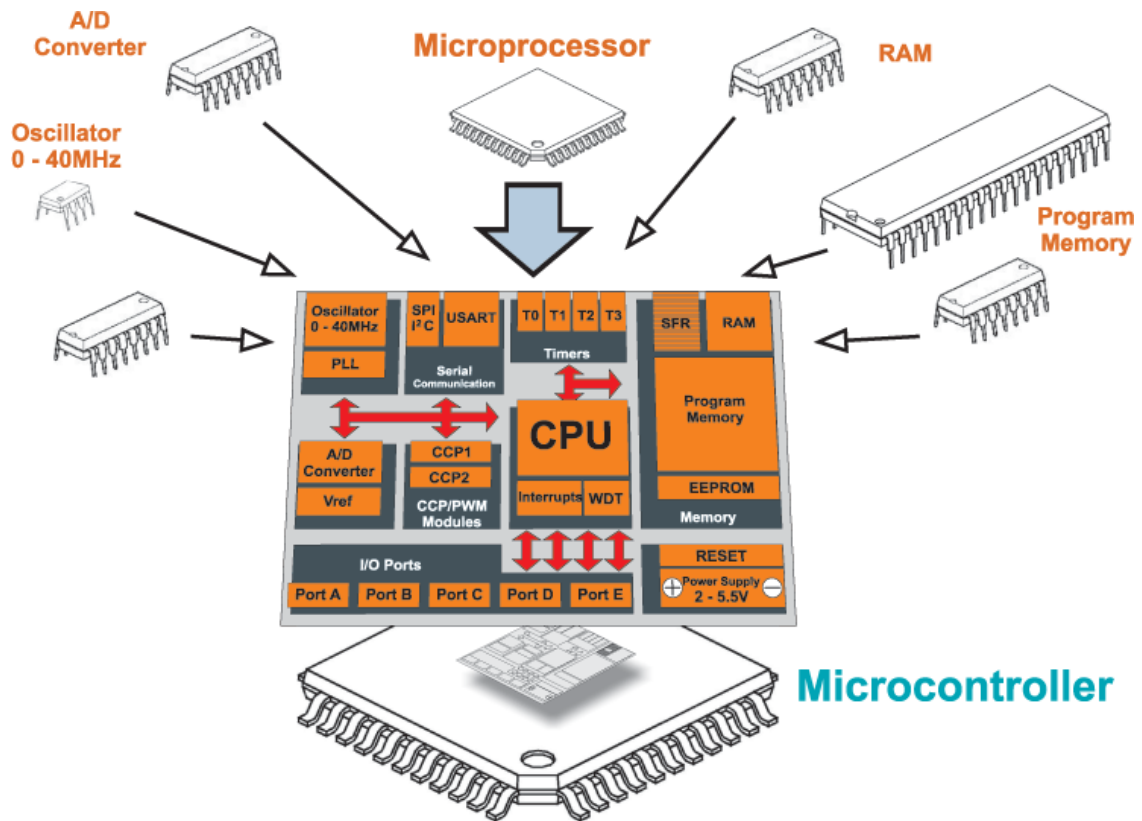
Εκτός απ'τη C.P.U. κάθε μικροϋπολογιστικό σύστημα περιλαμβάνει τη μνήμη που κρατεί δεδομένα και προγράμματα και τις βαθμίδες εισόδου-εξόδου(I/O) οι οποίες επιτρέπουν τη σύνδεση του μικροεπεξεργαστή με εξωτερικές διατάξεις, όπως πληκτρολόγια, εκτυπωτές, μονάδες δίσκων κ.ά. Όλα τα παραπάνω αποτελούν ένα οποιοδήποτε μικροϋπολογιστικό σύστημα (microcomputer).



Εικόνα 2. Σχηματικό διάγραμμα μικροϋπολογιστή

Ο **μικροελεγκτής** (microcontroller unit, MCU) λοιπόν είναι ένας μικροϋπολογιστής μέσα σ'ένα ολοκληρωμένο κύκλωμα. Δηλ. η C.P.U. , οι μονάδες μνήμης RAM και EEPROM, οι I/O βαθμίδες μαζί και μ'άλλες βαθμίδες όπως A/D μετατροπείς, timers/counters, κτλ βρίσκονται μέσα στο ίδιο ολοκληρωμένο κύκλωμα (chip).

Σήμερα πλέον έχουν εμφανιστεί πολλές και διαφορετικές οικογένειες μικροελεγκτών, όπως αυτοί της σειράς 8048 και 8051 της Intel, η σειρά 68HC11 της Motorola, η σειρά Z8 της Zilog, η σειρά PIC της Microchip, η σειρά MSP430 της Texas Instruments, οι μικροελεγκτές AVR της Atmel, κτλ.



Εικόνα 3. Σχηματικό διάγραμμα μικροελεγκτή



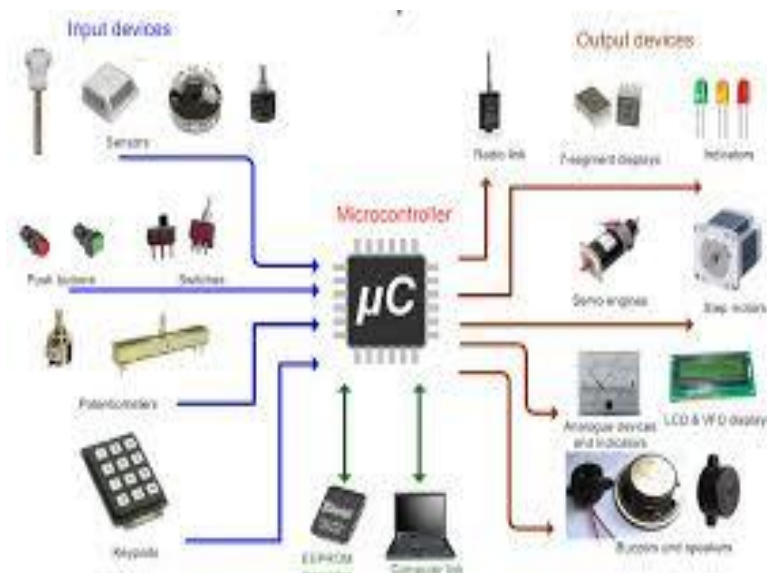
Εικόνα 4. Οικογένειες μικροελεγκτών

1.1.2 Χρήση των μικροελεγκτών

Κατά τη σχεδίαση ψηφιακών εφαρμογών με μικροελεγκτές, αυτά που πρέπει να λάβουμε υπόψη μας είναι ο τρόπος υλοποίησης, η απλότητα, η ελαχιστοποίηση του κόστους αλλά και η σχετικά χαμηλή ταχύτητα εφαρμογών. Πριν από τους μικροελεγκτές, χρησιμοποιούνταν διάφορες εναλλακτικές λύσεις για σχεδίαση ψηφιακών εφαρμογών οι οποίες κατηγοριοποιούνται παρακάτω ως εξής:

- 1) Χρήση ενός ψηφιακού κυκλώματος ειδικού τύπου. Δηλ. χρήση διακριτών ολοκληρωμένων κυκλωμάτων για λογικές πράξεις.
- 2) Χρήση ψηφιακών κυκλωμάτων που περιλαμβάνει ένα P.L.D.(προγραμματιζόμενη λογική διάταξη ή Programmable Logic Device). Ένα P.L.D. περιέχει ένα μεγάλο αριθμό λογικών κυκλωμάτων από τα οποία μόνο ένα τμήμα χρησιμοποιείται σε πολλές κοινές εφαρμογές.
- 3) Υλοποίηση βασισμένη σε Ολοκληρωμένα Κυκλώματα Εξειδικευμένων Εφαρμογών (Application Specific Integrated Circuits A.S.I.C.). Τα A.S.I.C. έχουν παρόμοια λειτουργία με τα P.L.D.'s αλλά είναι πιο βελτιωμένα.

Οι παραπάνω μέθοδοι εμφανίζουν σημαντικές ομοιότητες μεταξύ τους και ουσιαστικά διαφέρουν ως προς την υλοποίησή τους. Από σχεδιαστική άποψη η χρήση των μικροελεγκτών είναι εύκολη και προσιτή. Κάθε μικροελεγκτής μπορεί να συνδεθεί με διάφορα είδη κινητήρων, με προσωπικό υπολογιστή, να διαβάσει τιμές από εξωτερικούς αισθητήρες και να μπορεί επίσης να συνδεθεί με τοπικό δίκτυο μικροελεγκτών. Επιπρόσθετα, η χρήση τους είναι συχνή στα ενσωματωμένα συστήματα (embedded systems).



Εικόνα 5. Εφαρμογές μικροελεγκτών

1.1.3 Λειτουργικές βαθμίδες ενός μικροελεγκτή

Ας εξετάσουμε τις διάφορες βαθμίδες ενός μικροελεγκτή.

Κεντρική μονάδα επεξεργασίας: Η κεντρική μονάδα επεξεργασίας (C.P.U.) ή επεξεργαστής αποτελεί το «πυρήνα» ενός μικροελεγκτή και γενικά κάθε μικροϋπολογιστικού συστήματος. Κάνει ανάκληση δεδομένων (fetch) απ'τη μνήμη προγράμματος υπό μορφή εντολών, αποκωδικοποιεί τις εντολές και μετά τις εκτελεί. Η C.P.U. αποτελείται από καταχωρητές, την αριθμητική-λογική μονάδα, τον αποκωδικοποιητή εντολών και από διάφορα κυκλώματα ελέγχου.

Μνήμη προγράμματος: Στη μνήμη προγράμματος αποθηκεύονται οι εντολές που σχηματίζουν το κορμό του προγράμματος. Για τη χρήση μεγαλύτερων σε μήκος προγραμμάτων, το είδος της μνήμης αυτής μπορεί να χωριστεί σε εσωτερική και εξωτερική μνήμη προγράμματος. Η μνήμη αυτή είναι μη-πτητική δηλ. διατηρεί τα δεδομένα και χωρίς τροφοδοσία. Τέτοιες μνήμες είναι τύπου **EEPROM** (Electrically Erasable Read Only Memory).

Μνήμη RAM: Η μνήμη RAM (Random Access Memory) αποτελεί τη μνήμη δεδομένων του μικροελεγκτή. Η C.P.U. χρησιμοποιεί τη μνήμη αυτή για την αποθήκευση μεταβλητών καθώς επίσης και τη λεγόμενη Στοίβα (Stack). Η στοίβα χρησιμοποιείται απ'τη C.P.U. για τη προσωρινή αποθήκευση των διευθύνσεων επιστροφής από υπορουτίνα ή από σήμα διακοπής.

Ταλαντωτής χρονισμού: Ο μικροελεγκτής εκτελεί ένα πρόγραμμα με κάποιο ορισμένο ρυθμό. Ο ρυθμός αυτός καθορίζεται απ'τη συχνότητα λειτουργίας του ταλαντωτή χρονισμού. Ο ταλαντωτής χρονισμού (Clock Oscillator) μπορεί να'ναι ένας εσωτερικός ταλαντωτής τύπου RC ή ένας κρύσταλλος χαλαζία. Η λειτουργία του ξεκινά αμέσως μετά την εφαρμογή της τάσης τροφοδοσίας.

Σύστημα επανατοποθέτησης και κύκλωμα ανίχνευσης βυθίσεων τάσης: Το κύκλωμα αυτό λέγεται αλλιώς και Reset. Το Reset εξασφαλίζει το γεγονός ότι όλες οι εσωτερικές μονάδες και τα κυκλώματα ελέγχου του μικροελεγκτή θα ξεκινήσουν να λειτουργούν κατά την εφαρμογή της τροφοδοσίας, από κάποια προκαθορισμένη αρχική κατάσταση ενώ όλοι οι καταχωρητές βρίσκονται σε κατάλληλες αρχικές τιμές. Το κύκλωμα ανίχνευσης βυθίσεων τάσης είναι ένα κύκλωμα ελέγχου, το οποίο παρακολουθεί το επίπεδο τάσης τροφοδοσίας και αν ανιχνευτεί κάτι τέτοιο, θέτει το μικροελεγκτή σε κατάσταση επανατοποθέτησης.

Σειριακή θύρα επικοινωνίας: Η θύρα αυτή χρησιμοποιείται για την επικοινωνία του μικροελεγκτή με διάφορες εξωτερικές διατάξεις υπό τη μορφή σειριακής μετάδοσης δεδομένων. Η θύρα αυτή μπορεί να λειτουργήσει σε

οποιαδήποτε ταχύτητα μετάδοσης δεδομένων απαιτηθεί. Οι σειριακές θύρες χωρίζονται σε σύγχρονες και ασύγχρονες. Στις πρώτες απαιτείται πρόσθετο σήμα συγχρονισμού, ενώ στην ασύγχρονη δεν υφίσταται.

Ψηφιακή θύρα εισόδου-εξόδου: Ο μικροελεγκτής χρησιμοποιεί τις ψηφιακές θύρες εισόδου-εξόδου με σκοπό την ανταλλαγή δεδομένων από και προς το εξωτερικό περιβάλλον.

Αναλογική θύρα εισόδου: Στους μικροελεγκτές μπορούμε να'χουμε και αναλογικά σήματα εισόδου και αυτό χάρη στους μετατροπείς αναλογικού σήματος σε ψηφιακό (Analog to Digital Converters). Μπορούμε να'χουμε και αναλογικές εξόδους με τη χρήση DACs (Digital to Analog Converters).

Χρονοστής: Ένας χρονοστής (Timer) χρησιμοποιείται για το χρονισμό ή και τη σηματοδότηση διάφορων γεγονότων όπως π.χ. να θέλουμε να επαναλαμβάνεται μια διαδικασία κάθε δευτερόλεπτο. Ο χρονοστής παράγει αυτό το ζητούμενο ρυθμό. Επίσης ο χρονοστής χρησιμοποιείται και για τη καταμέτρηση γεγονότων. Γι'αυτό λέγεται και μετρητής (counter).

Χρονοστής επιτήρησης: Είναι ένας χρονοστής ειδικού σκοπού (Watch Dog Timer ή WDT) που υπάρχει σ'όλους τους σύγχρονους μικροελεγκτές. Ο χρονοστής αυτός χρησιμεύει στην αποφυγή πιθανής κατάρρευσης του προγράμματος. Όταν ενεργοποιηθεί, ένας εσωτερικός μετρητής ξεκινά να μετράει. Αν το πρόγραμμα δε μηδενίσει το μετρητή αυτόν, τότε κάποια στιγμή θα επέλθει η υπερχειλίση του και ο μικροελεγκτής θα επανατοποθετηθεί.

Ρολόι πραγματικού χρόνου: Πρόκειται για ένα χρονοστή ειδικού σκοπού, ο οποίος μετράει την ώρα και διατηρεί την ημερομηνία.

1.1.4 Αρχιτεκτονική μικροελεγκτών

Οι διάφορες αρχιτεκτονικές των μικροελεγκτών διακρίνονται με βάση τα ιδιαίτερα κάθε φορά χαρακτηριστικά και τις ιδιότητες που παρουσιάζει κάθε σύστημα. Ένα πολύ διαδεδομένο σχήμα είναι αυτό που λαμβάνει υπόψη τον συνολικό αριθμό εντολών μεταβλητού μήκους, μικρό αριθμό καταχωρητών και πολλαπλά είδη διευθυνσιοδότησης. Έτσι έχουμε την αρχιτεκτονική **C.I.S.C. (Complex Instruction Set Computer)**. Υπάρχει όμως και η αρχιτεκτονική **R.I.S.C. (Reduced Instruction Set Computer)** η οποία βασίζεται σ'ένα σχετικά μικρό αριθμό εντολών που χαρακτηρίζεται από μια απλή δομή σταθερού μήκους, σ'ένα είδος προσπέλασης μνήμης βασισμένης στο μοντέλο φόρτωσης αποθήκευσης και σ'ένα μεγάλο αριθμό καταχωρητών.

Ένα άλλο σχήμα με βάση το οποίο διακρίνουμε τις διατάξεις μικροελεγκτών είναι η πρόσβαση στη μνήμη προγράμματος και τη μνήμη δεδομένων. Ένα τέτοιο μοναδικό μοντέλο μνήμης, το οποίο είναι γνωστό με την ονομασία αρχιτεκτονική **Harvard** προβλέπει ξεχωριστούς διαύλους επικοινωνίας της μνήμης προγράμματος με τη μνήμη δεδομένων. Μ'αυτό το τρόπο υπάρχει ξεχωριστός χώρος μνήμης για αποθήκευση προγράμματος και δεδομένων. Αντίθετα με τη Harvard, η αρχιτεκτονική **Von Neumann** διαθέτει ενιαίο χώρο μνήμης για αποθήκευση προγράμματος και δεδομένων. Η αρχιτεκτονική Von Neumann χρησιμοποιούταν πιο παλιά για σχεδίαση μικροεπεξεργαστών και συγκριτικά με τη Harvard είναι πιο αργή διότι η C.P.U. ανακαλεί δεδομένα σε μεγαλύτερο χρόνο προσπέλασης απ'ό,τι με τη Harvard.

Επίσης οι μικροελεγκτές ταξινομούνται και με βάση τον τρόπο διαχείρισης και αποθήκευσης που υφίστανται τα δεδομένα εντός της C.P.U. Με κάθε πρόγραμμα επιτυγχάνεται η διαχείριση δεδομένων στους μικροελεγκτές. Ο τρόπος διαχείρισης και αποθήκευσης δεδομένων εντός της C.P.U. μαζί μ'άλλους τρόπους προσπέλασης, διαμορφώνουν μια βάση με την οποία ταξινομούνται οι διάφορες αρχιτεκτονικές των επεξεργαστών. Οπότε, σύμφωνα με τα παραπάνω, έχουμε τέσσερα πρότυπα : α) Το μοντέλο στοίβας, β) το μοντέλο συσσωρευτή (accumulator), γ) το μοντέλο καταχωρητή-μνήμης και δ) το μοντέλο πολλών καταχωρητών.

Στο πρώτο μοντέλο η αριθμητική-λογική μονάδα του συστήματος αποθηκεύει όλους τους τελεστές και τις μεταβλητές από και προς το σωρό. Η εντολή που το κάνει αυτό είναι η PUSH και τοποθετεί την πιο πρόσφατη μεταβλητή στην αρχή του σωρού (stack). Αν π.χ. θέλουμε να κάνουμε μια πρόσθεση δυο αριθμών, τότε η A.L.U. θα λάβει τις δυο αυτές τιμές απ'την υψηλότερη θέση του σωρού και μετά θα κάνει τη πράξη. Το αποτέλεσμα της πράξης αποθηκεύεται πάλι στο σωρό στην υψηλότερη θέση. Η εντολή POP «εξάγει» δεδομένα απ'το σωρό.


Στο δεύτερο μοντέλο, ένας απ'τους δυο τελεστές είναι ο συσσωρευτής, ο οποίος είναι ένας καταχωρητής στον οποίο αποθηκεύονται αποτελέσματα αριθμητικών και λογικών πράξεων. Η αποθήκευση τιμών γίνεται απ'το συσσωρευτή και οι εντολές «εμπεριέχουν» το συσσωρευτή χωρίς να τον γράφουμε εμείς.

Στο τρίτο μοντέλο, η πρόσβαση στη μνήμη γίνεται με τη βοήθεια εντολών φόρτωσης και αποθήκευσης. Συνεπώς οι καταχωρητές φορτώνονται πρώτα με τις τιμές των μεταβλητών, εκτελείται η πράξη και το αποτέλεσμα αποθηκεύεται σ'έναν απ'τους προηγούμενους καταχωρητές, το περιεχόμενο του οποίου με τη σειρά του, αποθηκεύεται στη μεταβλητή προορισμού. Τέλος, στο τελευταίο μοντέλο υπάρχουν πολλοί καταχωρητές τους οποίους μπορούμε να χρησιμοποιήσουμε για κάθε πράξη.

Στους παλαιότερους επεξεργαστές συναντούσαμε συχνά τις αρχιτεκτονικές στοίβας και συσσωρευτή. Στους πιο σύγχρονους, συναντάμε το τέταρτο μοντέλο, δηλ. την αρχιτεκτονική πολλών καταχωρητών. Οι λόγοι για τους

οποίους επικράτησε αυτή η αρχιτεκτονική είναι επειδή η διαδικασία πρόσβασης εσωτερικών καταχωρητών είναι ταχύτερη σε σχέση με τη πρόσβαση εξωτερικής μνήμης, γι'αυτό και υπάρχουν πολλοί καταχωρητές. Στο τομέα του μεταγλωττιστή, θεωρείται πιο εύκολη η πρόσβαση των καταχωρητών γενικής χρήσης κι όχι της στοίβας, παρά το γεγονός ότι η στοίβα είναι μονάδα της C.P.U.

RISC vs. CISC



RISC / CISC Architecture

<i>RISC</i>	<i>CISC</i>
<ul style="list-style-type: none">■ Single-clock■ Reduced instructions■ No microcode■ Data explicitly accessed■ Easier to validate■ Larger code sizes (~30%)■ Low cycles/second■ More transistors on memory registers■ Pipelining friendly■ Emphasis on software	<ul style="list-style-type: none">■ Multi-clock■ Complex instructions■ Complicated microcode■ Memory to memory operations■ Difficult to validate■ Smaller code sizes■ High cycles/second■ More transistors for complex instructions■ Compiler friendly■ Emphasis on hardware

BYU CS 224

ISA

9

Εικόνα 6. Συγκριτικός πίνακας RISC και CISC αρχιτεκτονικής επεξεργαστών

VAN-NEUMANN ARCHITECTURE	HARVARD ARCHITECTURE
Used in conventional processors found in PCs and Servers, and embedded systems with only control functions.	Used in DSPs and other processors found in latest embedded systems and Mobile communication systems, audio, speech, image processing systems
The data and program are stored in the same memory	The data and program memories are separate
The code is executed serially and takes more clock cycles	The code is executed in parallel
There is no exclusive Multiplier	It has MAC (Multiply Accumulate)
Absence of Barrel Shifter	Barrel Shifter help in shifting and rotating operations of the data
The programs can be optimized in lesser size	The program tend to grow big in size

Εικόνα 7. Συγκριτικός πίνακας αρχιτεκτονικής Von Neumann και Harvard

1.2. Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C

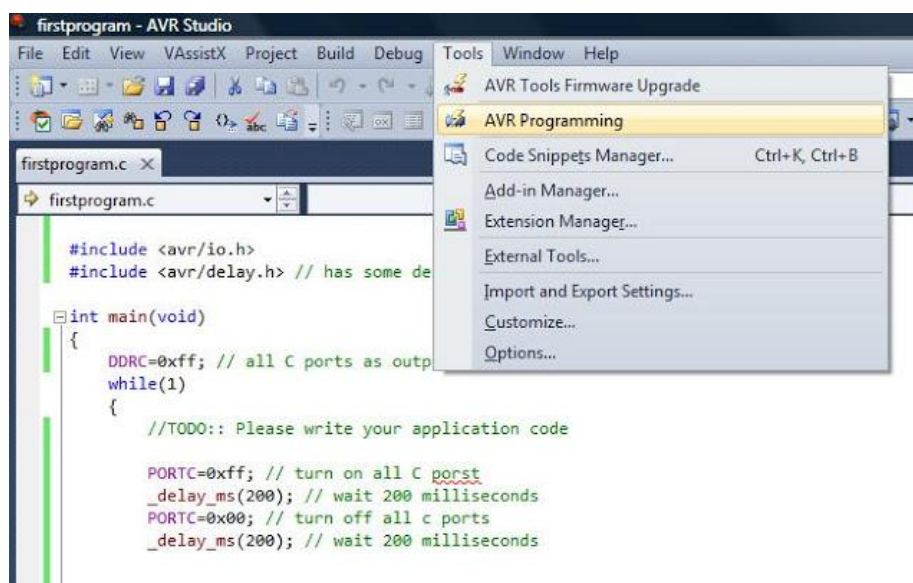
1.2.1. Γενικές πληροφορίες της γλώσσας C

Η γλώσσα προγραμματισμού C είναι από τις σημαντικότερες γλώσσες στο χώρο της Πληροφορικής και είναι θεμέλιο για γλώσσες «επαγγελματικού» επιπέδου όπως η C++, Java και C#. Η C εφευρέθηκε για πρώτη φορά το 1978 από τον Dennis Ritchie χρησιμοποιώντας το λειτουργικό σύστημα UNIX. Λόγω της μεγάλης δημοτικότητας που απέκτησε η C, οργανώθηκε μια επιτροπή το 1983 για την καθιέρωση ενός προτύπου για την C, η ANSI. Το πρότυπο της ANSI C

τελειοποιήθηκε το 1989 και τα πρώτα αντίγραφα της έγιναν διαθέσιμα στο κοινό το 1990.

Η γλώσσα C αναφέρεται συχνά σαν γλώσσα «μεσαίου επιπέδου». Πριν από την C υπήρχε ένας τύπος γλώσσας για το προγραμματισμό των υπολογιστών. Ο τύπος αυτός ήταν η **γλώσσα Assembly**, η οποία είναι και συμβολική αναπαράσταση των οδηγιών μηχανής που εκτελεί ο υπολογιστής. Η γλώσσα Assembly είναι γλώσσα χαμηλού επιπέδου, διότι ο προγραμματιστής δουλεύει σε συμβολική μορφή με τις πραγματικές οδηγίες που εκτελεί ο υπολογιστής. Επίσης, η γλώσσα Assembly μπορεί να χρησιμοποιείται για τη δημιουργία αποτελεσματικών προγραμμάτων, αλλά δεν παρέχει ενσωματωμένες δομές ελέγχου, εντολές εισόδου/εξόδου κτλ. Επειδή ο προγραμματισμός με την Assembly ήταν μια δύσκολη και κουραστική διαδικασία, υπήρχε πάντα η ανάγκη για μια γλώσσα που να εξισορροπούσε την ευχρηστία με την αποτελεσματικότητα.

Η γλώσσα C συνδυάζει επιτυχώς τη δομή μιας υψηλού επιπέδου γλώσσας με την ισχύ και αποτελεσματικότητα της Assembly και χρησιμεύει για τη δημιουργία λογισμικού προγράμματος. Λογισμικό προγράμματος λέμε τα προγράμματα που υποβοηθούν τη λειτουργία του υπολογιστή. Με την C προγραμματίζονται μικροεπεξεργαστές και μικροελεγκτές για εφαρμογές πχ ενσωματωμένων συστημάτων και γράφεται λογισμικό για εφαρμογές δικτύων υπολογιστών.



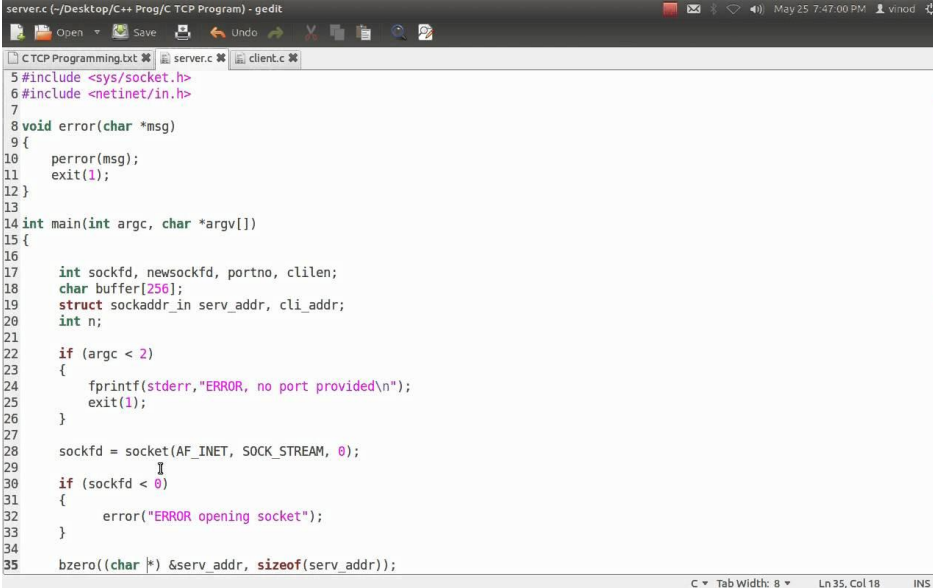
```
firstprogram - AVR Studio
File Edit View VAssistX Project Build Debug Tools Window Help
AVR Tools Firmware Upgrade
AVR Programming
Code Snippets Manager... Ctrl+K, Ctrl+B
Add-in Manager...
Extension Manager...
External Tools...
Import and Export Settings...
Customize...
Options...

firstprogram.c x
firstprogram.c
#include <avr/io.h>
#include <avr/delay.h> // has some de

int main(void)
{
    DDRC=0xff; // all C ports as outp
    while(1)
    {
        //TODO:: Please write your application code

        PORTC=0xff; // turn on all C port
        _delay_ms(200); // wait 200 milliseconds
        PORTC=0x00; // turn off all c ports
        _delay_ms(200); // wait 200 milliseconds
    }
}
```

Εικόνα 8. Παράδειγμα C προγράμματος για μικροελεγκτή AVR



```
server.c (-/Desktop/C++ Prog/C TCP Program) - gedit
5 #include <sys/socket.h>
6 #include <netinet/in.h>
7
8 void error(char *msg)
9 {
10     perror(msg);
11     exit(1);
12 }
13
14 int main(int argc, char *argv[])
15 {
16
17     int sockfd, newsockfd, portno, clien;
18     char buffer[256];
19     struct sockaddr_in serv_addr, cli_addr;
20     int n;
21
22     if (argc < 2)
23     {
24         fprintf(stderr, "ERROR, no port provided\n");
25         exit(1);
26     }
27
28     sockfd = socket(AF_INET, SOCK_STREAM, 0);
29
30     if (sockfd < 0)
31     {
32         error("ERROR opening socket");
33     }
34
35     bzero((char *) &serv_addr, sizeof(serv_addr));
```

Εικόνα 9. Παράδειγμα C προγράμματος για πακέτα socket για TCP/IP

1.2.2. Open-source υλικό (open-source hardware)

Το "ανοικτό υλικό" ή το "υλικό ανοικτού κώδικα" αναφέρεται στις προδιαγραφές σχεδιασμού ενός φυσικού αντικείμενου, οι οποίες διαθέτουν άδεια με τέτοιο τρόπο ώστε το εν λόγω αντικείμενο να μπορεί να μελετηθεί, να τροποποιηθεί, να δημιουργηθεί και να διανεμηθεί από τον οποιονδήποτε. Το ανοικτό υλικό είναι ένα σύνολο αρχών σχεδιασμού και νομικών πρακτικών, όχι ενός συγκεκριμένου τύπου αντικείμενου. Ως εκ τούτου, ο όρος μπορεί να αναφέρεται σε οποιοδήποτε αριθμό αντικείμενων όπως αυτοκίνητα, καρέκλες, υπολογιστές, ρομπότ ή ακόμα και σπίτια.

Σε αντίθεση με το λογισμικό ανοικτού κώδικα (open-source software) , το υλικό ανοικτού κώδικα ψάχνει τη συνεργασία όπου τα φυσικά αντικείμενα είναι το τελικό αποτέλεσμα. Αυτό απασχολεί ένα καταναμημένο μοντέλο ανάπτυξης υλικού με συνεισφέροντες από όλο τον κόσμο. Επίσης τα ανοικτού κώδικα αντικείμενα επιτρέπουν μια ξεχωριστή ελευθερία για πρόσβαση σε πηγαία αρχεία ενός σχεδίου, για βελτιώσεις σε μια μεγαλύτερη κοινότητα.

Η πλατφόρμα Arduino είναι, όπως αναφέρθηκε παραπάνω, μια ανοικτού κώδικα πλατφόρμα και συνδυάζει την ελευθερία στο σχέδιο, στην αρχιτεκτονική και στη συνεργασία. Το Arduino IDE είναι μια ανοικτού κώδικα εφαρμογή που εξηγείται παρακάτω.

1.2.3. ΠΛΑΤΦΟΡΜΕΣ ΜΙΚΡΟΕΛΕΓΚΤΗ OPEN SOURCE

Αφού περιγράψαμε τη γλώσσα προγραμματισμού C , θα αναφέρουμε ορισμένες - πιο γνωστές - πλατφόρμες μικροελεγκτή ανοικτού κώδικα (open source) οι οποίες χρησιμοποιούνται συχνά σε πολλές εφαρμογές.

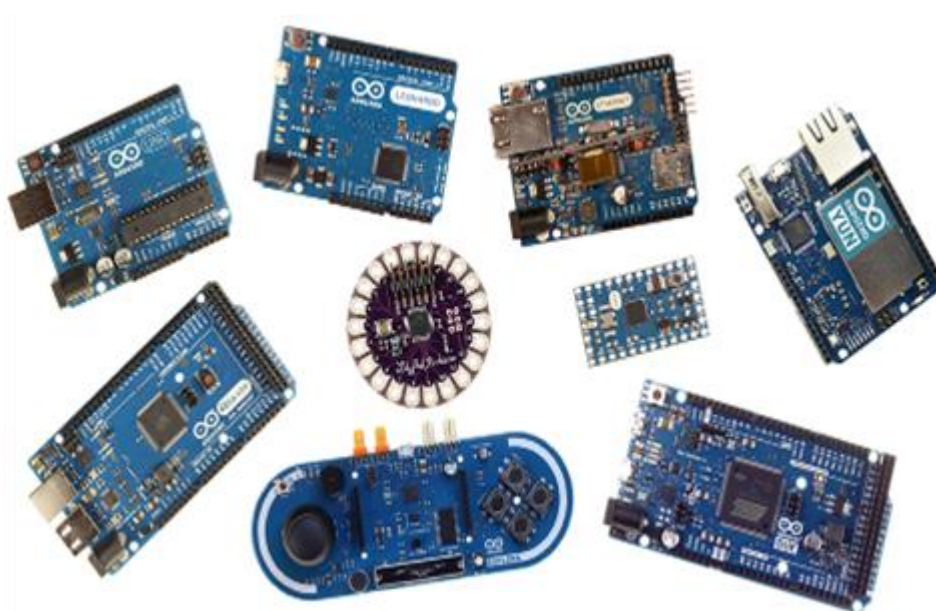
1) ARDUINO

To Arduino είναι πλατφόρμα ανάπτυξης έργων (projects) ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring C (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προσυναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους. Το πρόγραμμα Arduino έλαβε τιμητική μνεία στην κατηγορία Digital Communities στο Prix Ars Electronica το 2006.

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Σε εννοιολογικό επίπεδο, στην χρήση του Arduino software stack, όλα τα boards προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται αυτό διαφέρει σε κάθε hardware εκδοχή. Οι σειριακές πλάκες Arduino περιέχουν ένα απλό level shifter κύκλωμα για την μετατροπή του σήματος επιπέδου RS-232 σε TTL. Τα σημερινά Arduino προγραμματίζονται μέσω USB, το οποίο καθίσταται δυνατό μέσω της εφαρμογής προσαρμογών chip USB-to-Serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board, Bluetooth ή άλλες μεθόδους. (Όταν χρησιμοποιείται με παραδοσιακά εργαλεία microcontroller αντί για το Arduino IDE, χρησιμοποιείται

πρότυπος προγραμματισμός AVR ISP). Παρακάτω φαίνονται διάφορα μοντέλα Arduino (Arduino boards).



Εικόνα 10. Arduino boards

2) TI MSP430

Τα πακέτα ανοιχτού κώδικα MSP430™ και MSP432™ GCC είναι πλήρη εργαλεία μεταγλώττισης C/C++ για την υλοποίηση ενσωματωμένων εφαρμογών που βασίζονται σε μικροελεγκτές MSP430 και MSP432 της Texas Instruments. Αυτοί οι ελεύθεροι μεταγλωττιστές (compilers) GCC υποστηρίζουν όλες τις συσκευές MSP430 και MSP432 χωρίς περιορισμούς μεγέθους κώδικα. Επιπλέον, αυτοί οι μεταγλωττιστές μπορούν να χρησιμοποιηθούν αυτόνομα από τη γραμμή εντολών ή να επιλεγούν για χρήση στο Code Composer Studio v6.0 ή νεότερη έκδοση. Ξεκινήστε σήμερα αν χρησιμοποιείτε περιβάλλοντα Windows®, Linux® ή Mac OS X®.

Η οικογένεια MSP430 προσφέρει μια μεγάλη ποικιλία μικροελεγκτών RISC στα 16 bit με εξαιρετικά χαμηλή ισχύ, ενσωματωμένα αναλογικά και ψηφιακά περιφερειακά για εφαρμογές ανίχνευσης και μέτρησης. Επιπλέον, οι μικροελεγκτές MSP430 υποστηρίζονται από τα κιτ ανάπτυξης, τα σχέδια αναφοράς, το λογισμικό, την κατάρτιση, την τεκμηρίωση και την ηλεκτρονική υποστήριξη της Texas Instruments (www.ti.com).



Εικόνα 11. Κιτ ανάπτυξης MSP-EXP430FR2355 LaunchPad™

Το κιτ ανάπτυξης MSP-EXP430FR2355 LaunchPad™ είναι μια εύχρηστη μονάδα αξιολόγησης (EVM) που βασίζεται στον μικροελεγκτή MSP430FR2355 (MCU). Περιέχει όλα όσα χρειάζονται για να αρχίσουν να αναπτύσσονται στην πλατφόρμα MCU με εξαιρετικά χαμηλή ισχύ MSP430FR2x Value Line, συμπεριλαμβανομένου του ενσωματωμένου ανιχνευτή εντοπισμού σφαλμάτων για προγραμματισμό, εντοπισμό σφαλμάτων και μετρήσεις ενέργειας. Η κάρτα διαθέτει 2 κουμπιά και 2 LED για τη δημιουργία ενός απλού περιβάλλοντος εργασίας. Διαθέτει επίσης αισθητήρα περιβάλλοντος φωτός και υποδοχή για εξωτερικές αναλογικές πηγές.

Ο μικροελεγκτής MSP430FR2355 περιλαμβάνει έναν έξυπνο αναλογικό συνδυασμό διαμορφωμένων στοιχείων αλυσίδας σήματος, συμπεριλαμβανομένων των επιλογών για μετατροπείς ψηφιακού προς αναλογικό (DAC) και προγραμματιζόμενου ενισχυτή κέρδους (PGAs), καθώς και μετατροπέα αναλογικού προς ψηφιακό 12 bit (ADC)) και δύο βελτιωμένους συγκριτές (eCOMP). Ένα από τα demos από το κιβώτιο περιλαμβάνει ένα GUI για να παρουσιάσει μερικές από τις έξυπνες αναλογικές συνθέσεις combo.

Η 24MHz συχνότητας συσκευή MSP430FR2355 διαθέτει ενσωματωμένη μνήμη FRAM (Ferroelectric Random Access Memory) 32KB, μια μη πτητική μνήμη γνωστή για εξαιρετικά χαμηλή ισχύ, υψηλή αντοχή και πρόσβαση εγγραφής υψηλής ταχύτητας. Σε συνδυασμό με το 4KB του SRAM on-chip, οι χρήστες έχουν πρόσβαση σε 32KB μνήμης για να χωριστούν μεταξύ του προγράμματος και των δεδομένων, όπως απαιτείται. Για παράδειγμα, μια εφαρμογή καταγραφής δεδομένων μπορεί να απαιτεί μια μεγάλη μνήμη δεδομένων με σχετικά μικρή μνήμη προγράμματος, οπότε η μνήμη μπορεί να διανεμηθεί όπως απαιτείται μεταξύ μνήμης προγράμματος και μνήμης δεδομένων.



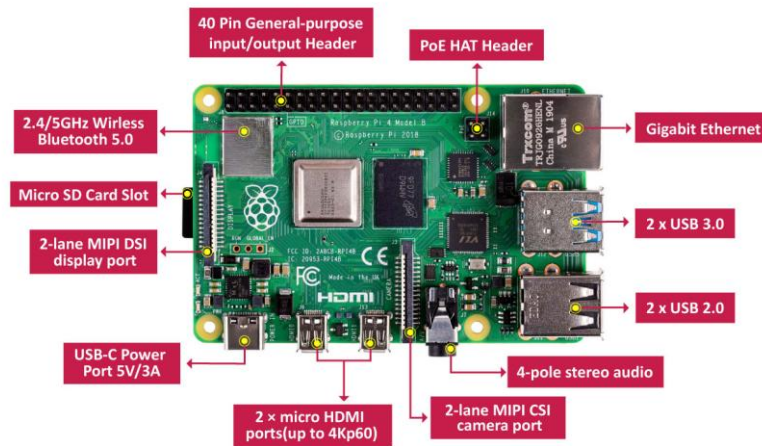
Εικόνα 12. Οικογένεια μικροελεγκτών MSP430G2xxx της Texas Instruments

3) RASPBERRY PI

Το Raspberry Pi είναι μια σειρά μικρών υπολογιστών με ένα μόνο σκέλος που αναπτύχθηκαν στο Ηνωμένο Βασίλειο από το Ίδρυμα Raspberry Pi για την προώθηση της διδασκαλίας της βασικής επιστήμης των υπολογιστών στα σχολεία και στις αναπτυσσόμενες χώρες. Το αρχικό μοντέλο έγινε πολύ πιο δημοφιλές από ό, τι αναμενόταν που πωλούσε εκτός της στοχευόμενης αγοράς για χρήσεις όπως η ρομποτική. Δεν περιλαμβάνει περιφερειακά (όπως πληκτρολόγια και ποντίκια) ή περιπτώσεις. Ωστόσο, ορισμένα εξαρτήματα έχουν συμπεριληφθεί σε διάφορες επίσημες και ανεπίσημες δέσμες. Πρόκειται στην ουσία για μητρικές πλακέτες υπολογιστή μικρότερων διαστάσεων.

Ορισμένα μοντέλα Raspberry φαίνονται παρακάτω

1) Raspberry Pi 4 Model B



Εικόνα 13. Raspberry Pi 4 Model B

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 1GB, 2GB or 4GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 x micro-HDMI ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.0 graphics
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)

- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient

2) Raspberry Pi 3 Model B



Technical Specification:

- Broadcom BCM2837 64bit ARMv7 Quad Core Processor powered Single Board Computer running at 1.2GHz
- 1GB RAM
- BCM43143 WiFi on board
- Bluetooth Low Energy (BLE) on board
- 40pin extended GPIO
- 4 x USB 2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source (now supports up to 2.4 Amps)
- Expected to have the same form factor has the Pi 2 Model B, however the LEDs will change position

Εικόνα 14. Raspberry Pi 3 Model B

1.3. ΔΙΚΤΥΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

1.3.1. Εισαγωγή στα δίκτυα υπολογιστών

Κατά τη διάρκεια του 20^{ου} αιώνα οι βασικές τεχνολογίες ήταν η συλλογή, η επεξεργασία και η διανομή πληροφοριών. Επίσης, μεταξύ των άλλων εξελίξεων είχαμε την εγκαθίδρυση των παγκόσμιων τηλεφωνικών δικτύων, την εφεύρεση του ραδιοφώνου και της τηλεόρασης, τη γένεση και ανάπτυξη της βιομηχανίας των υπολογιστών και τέλος την ανάπτυξη και εκτόξευση των τηλεπικοινωνιακών δορυφόρων με αρχή το δορυφόρο *Sputnik* τον Οκτώβριο του 1957 από την πρώτη Ε.Σ.Σ.Δ. Ως αποτέλεσμα της ραγδαίας τεχνολογικής προόδου, οι τεχνολογικοί αυτοί τομείς συγκλίνουν ταχύτατα και εξαφανίζονται οι διαφορές ανάμεσα στη συλλογή, στη μεταφορά, στην αποθήκευση και στην επεξεργασία των πληροφοριών.

Η συγχώνευση των υπολογιστών και των επικοινωνιών είχε βαθιά επίδραση στο τρόπο οργάνωσης των υπολογιστικών συστημάτων. Ο τρόπος με τον οποίο ένας υπολογιστής εξυπηρετούσε όλες τις υπολογιστικές ανάγκες ενός οργανισμού έχει πια ξεπεραστεί. Τώρα υπάρχει **πλήθος αυτόνομων αλλά διασυνδεδεμένων με κοινή τεχνολογία υπολογιστών, οι οποίοι μπορούν να ανταλλάσσουν μεταξύ τους δεδομένα**. Τέτοια συστήματα υπολογιστών λέγονται **δίκτυα υπολογιστών**.



Εικόνα 15. Παράδειγμα δικτύου υπολογιστών

1.3.2. Υλικό δικτύων

Υπάρχουν δυο ιδιότητες των δικτύων οι οποίες ξεχωρίζουν ως οι πλέον σημαντικές, δηλ. η τεχνολογία μετάδοσης και η κλίμακα. Με τη σειρά τους, υπάρχουν δυο είδη τεχνολογιών μετάδοσης που χρησιμοποιούνται ευρέως : 1) συνδέσεις εκπομπής και 2) συνδέσεις από σημείο σε σημείο.

Τα δίκτυα εκπομπής (broadcast networks) έχουν ένα μόνο κανάλι επικοινωνίας το οποίο είναι κοινόχρηστο από όλες τις μηχανές του δικτύου. Τα σύντομα μηνύματα που στέλνονται από κάθε μηχανήμα, τα οποία λέγονται πακέτα (packets), λαμβάνονται από όλες τις μηχανές (υπολογιστές, κινητά κτλ). Ένα πεδίο διεύθυνσης μέσα στο πακέτο προσδιορίζει τον επιθυμητό παραλήπτη και το κάθε μηχανήμα, μόλις λάβει το πακέτο, εξετάζει αυτό το πεδίο διεύθυνσης. Αν το πακέτο αυτό προορίζεται για το μηχανήμα που το έλαβε, αυτό προχωρά στην επεξεργασία του αλλιώς παραβλέπεται.

Τα δίκτυα από σημείο σε σημείο αποτελούνται από πολλές συνδέσεις ανάμεσα σε ζεύγη μηχανών. Για να φτάσει ένα πακέτο από τη προέλευση στον προορισμό του σε αυτόν τον τόπο δικτύου, μπορεί να χρειαστεί να επισκεφτεί πρώτα μια ή περισσότερες ενδιάμεσες μηχανές.

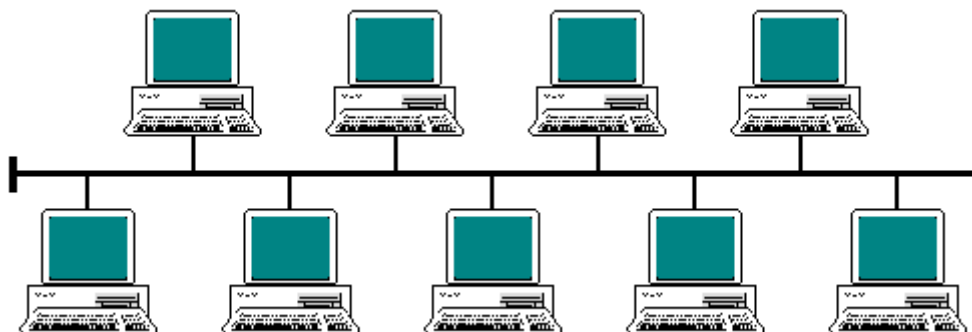
1.3.3. Ταξινόμηση δικτύων ανάλογα την έκταση και τη τοπολογία τους

Τα δίκτυα ταξινομούνται ανάλογα την έκτασή τους έτσι έχουμε τις παρακάτω κατηγορίες.

- 1) Τοπικά δίκτυα **LAN (Local Area Network)**
- 2) Μητροπολιτικά δίκτυα **MAN (Metropolitan Area Network)**
- 3) Δίκτυα ευρείας περιοχής **WAN (Wide Area Network)**

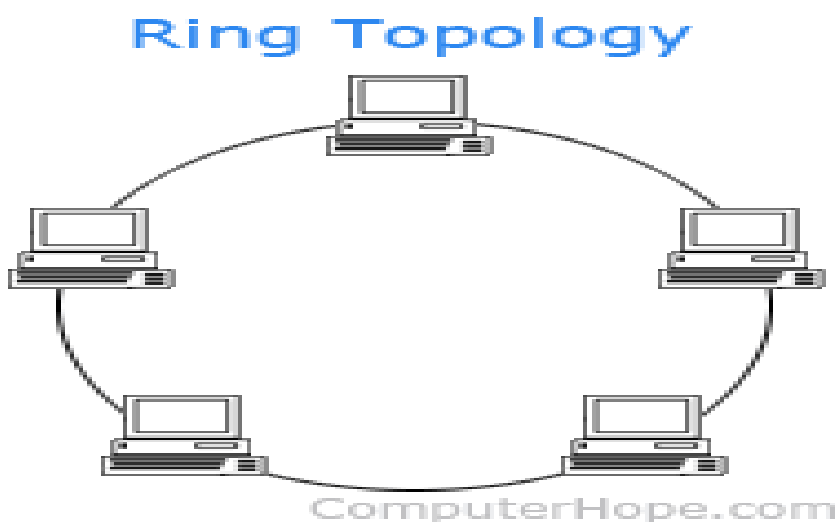
Τα **τοπικά δίκτυα** είναι ιδιωτικά δίκτυα τα οποία βρίσκονται μέσα σε ένα μόνο κτίριο ή κτιριακό συγκρότημα ή σε μια έκταση λίγων χιλιομέτρων και χρησιμοποιούνται για τη διασύνδεση προσωπικών υπολογιστών και σταθμών εργασίας σε γραφεία με στόχο την κοινοχρησία πόρων και ανταλλαγή πληροφοριών. Τα δίκτυα LAN μπορεί να χρησιμοποιούν μια τεχνολογία

μετάδοσης, η οποία συνίσταται σε ένα καλώδιο όπου είναι συνδεδεμένες όλες οι μηχανές. Ο ρυθμός μετάδοσης δεδομένων στα δίκτυα LAN είναι από 10 Mbps έως και 10 Gbps (τα πιο πρόσφατα). Πρότυπο LAN δικτύων είναι το γνωστό Ethernet (IEEE 802.3) και βασίζεται σε δίαυλο (bus) με αποκεντρωμένο έλεγχο και λειτουργεί με τις ταχύτητες αυτές και τα δεδομένα μεταδίδονται όποτε οι υπολογιστές θέλουν. Στη περίπτωση σύγκρουσης πακέτων, ο κάθε υπολογιστής περιμένει και δοκιμάζει αργότερα.



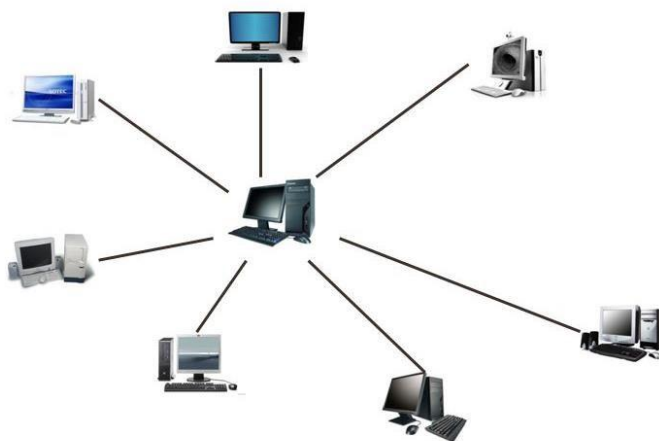
Εικόνα 16. Τοπολογία διαύλου LAN δικτύου

Επιπλέον, υπάρχει η τοπολογία δακτυλίου (ring) (IEEE 802.5) στην οποία τα κάθε bit δεδομένων μεταδίδεται μόνο του χωρίς να περιμένει για το υπόλοιπο πακέτο στο οποίο ανήκει. Συνήθως το κάθε bit μπορεί να καλύψει ολόκληρο το δακτύλιο στο διάστημα που απαιτείται για τη μετάδοση λίγων μόνο bit συχνά πριν καν μεταδοθεί ολόκληρο το πακέτο. Οι ταχύτητες εδώ είναι από 4 έως 16 Mbps.



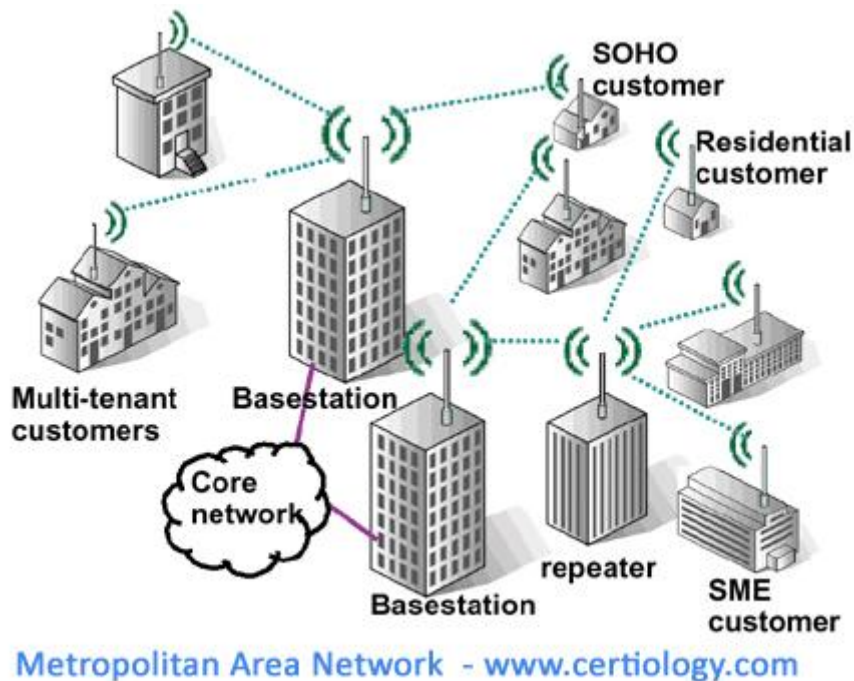
Εικόνα 17. Τοπολογία δακτυλίου LAN δικτύου

Τέλος, στην τοπολογία αστέρα, κάθε κόμβος (συσκευή) είναι συνδεδεμένος σε ένα "κεντρικό" κόμβο. Σε αυτή την τοπολογία ένα πακέτο δεδομένων που έχει αφετηρία έναν από τους περιφερειακούς κόμβους κατευθύνεται πάντα στον κεντρικό κόμβο ο οποίος το αναμεταδίδει σε όλους τους κόμβους. Οι περιφερειακοί κόμβοι επικοινωνούν μεταξύ τους με αποστολές και λήψεις στον κεντρικό κόμβο όπως φαίνεται παρακάτω.



Εικόνα 18. Τοπολογία αστέρα LAN δικτύου

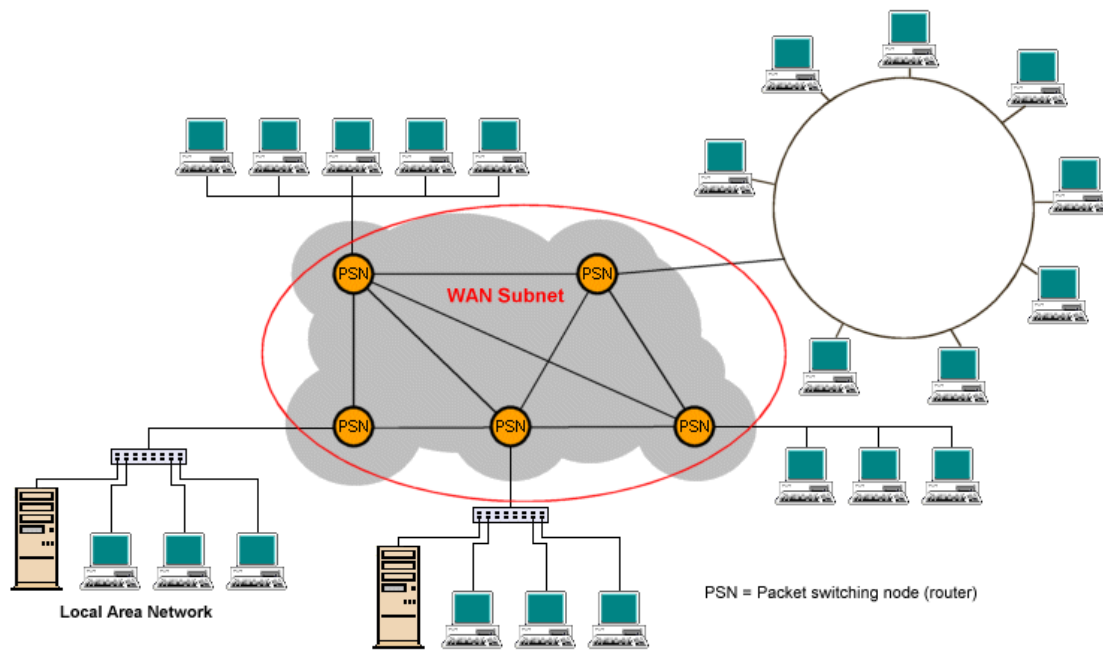
Τα **μητροπολιτικά δίκτυα** MAN καλύπτουν μια έκταση όσο μια πόλη. Ένα παράδειγμα τέτοιου δικτύου είναι το δίκτυο καλωδιακής τηλεόρασης. Σε αυτά τα πρώιμα συστήματα, μια μεγάλη κεραία ήταν τοποθετημένη στη κορυφή ενός λόφου και το σήμα στελνόταν στα σπίτια των συνδρομητών. Από τότε που το Internet άρχισε να προσελκύει το ενδιαφέρον του κόσμου, οι επιχειρήσεις καλωδιακής τηλεόρασης άρχισαν να αντιλαμβάνονται ότι με ορισμένες αλλαγές στο σύστημα θα μπορούσαν να παρέχουν αμφίδρομες υπηρεσίες Internet σε μη χρησιμοποιούμενα τμήματα του φάσματος.



Εικόνα 19. Μητροπολιτικό δίκτυο MAN

Το δίκτυο ευρείας περιοχής ή δίκτυο WAN εκτείνεται σε μια μεγάλη γεωγραφική περιοχή όπως μια χώρα ή μια ήπειρο και περιέχει ένα σύνολο μηχανών που προορίζονται για την εκτέλεση των προγραμμάτων των χρηστών. Τις μηχανές αυτές τις λέμε υπολογιστές υπηρεσίας και συνδέονται με ένα υποδίκτυο επικοινωνίας. Οι υπολογιστές υπηρεσίας δηλ, οι υπολογιστές των χρηστών ανήκουν στους πελάτες, ενώ το υποδίκτυο και η διαχείρισή του ανήκουν στην τηλεφωνική εταιρεία ή στο πάροχο υπηρεσιών Internet. Το υποδίκτυο μεταφέρει μηνύματα ανάμεσα στους υπολογιστές υπηρεσίας.

Στα περισσότερα δίκτυα WAN, το υποδίκτυο αποτελείται από δυο διακριτά συστατικά : τις γραμμές μετάδοσης και τα στοιχεία μεταγωγής. Οι γραμμές μετάδοσης μετακινούν bits ανάμεσα στους υπολογιστές του δικτύου και αυτό υαλοποιείται είτε με χάλκινα σύρματα, είτε με οπτικές ίνες, είτε ακόμα και ασύρματα. Τα στοιχεία μεταγωγής είναι εξειδικευμένοι υπολογιστές που συνδέουν τρεις ή περισσότερες γραμμές μετάδοσης. Όταν φτάνουν δεδομένα σε μια εισερχόμενη γραμμή, το στοιχείο μεταγωγής πρέπει να επιλέξει την εξερχόμενη γραμμή στην οποία θα τα προωθήσει. Ο υπολογιστής μεταγωγής λέγεται αλλιώς και δρομολογητής (router).



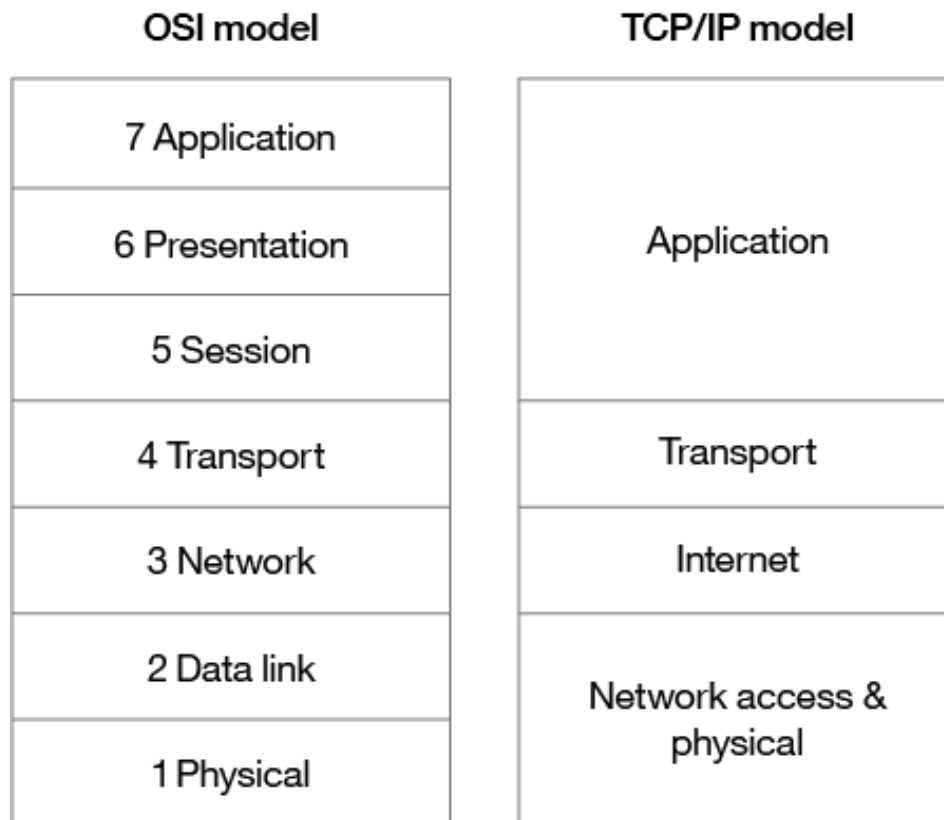
Εικόνα 20. WAN δίκτυο

1.3.4. Το μοντέλο OSI του ISO

Όταν οι πληροφορίες – οι οποίες παρίστανται σε δυαδικά δεδομένα-κινούνται σ'ένα δίκτυο, είναι σημαντικό να συμφωνήσουν όλα τα εμπλεκόμενα μέρη δηλ. ο δημιουργός του, ο τελικός παραλήπτης και όλοι οι ενδιάμεσοι, να χρησιμοποιούν τους ίδιους κανόνες, προδιαγραφές μορφοποίησης, χρονισμού και δρομολόγησης. Αυτοί οι κανόνες λέγονται πρωτόκολλα (protocols) και ορίζουν τις διαδρομές των δεδομένων στο δίκτυο καθώς επίσης και τη μορφή τους.

Καθώς η δικτυακή επικοινωνία έχει γίνει πιο πολύπλοκη, η κοινότητα των σχεδιαστών δικτύων έχει αποδεχθεί το μοντέλο Διασύνδεσης Ανοικτών Συστημάτων (Open Systems Interconnection, OSI) του Διεθνούς Οργανισμού Προτυποποίησης (International Organization for Standardization, ISO) για να καθορίζει τα ξεχωριστά στοιχεία μιας σύνδεσης δικτύου. Το μοντέλο OSI ισχύει για όλα τα είδη συστημάτων επικοινωνιών δεδομένων, επομένως και για τα ασύρματα δίκτυα. Επίσης επιτρέπει στο σχεδιαστή να αλλάζει ένα μόνο στοιχείο του δικτύου χωρίς να πρέπει να σχεδιάσει όλα τα υπόλοιπα εκ νέου.

Το μοντέλο OSI απεικονίζεται από μια στοίβα επτά επιπέδων όπως φαίνεται παρακάτω



Εικόνα 21. Τα επτά επίπεδα του μοντέλου OSI

1. Physical (Φυσικό επίπεδο) : Το φυσικό επίπεδο ορίζει το φυσικό μέσο ή υλικό που μεταφέρει τα σήματα μεταξύ των ακραίων σημείων μιας δικτυακής σύνδεσης. Το επίπεδο αυτό μπορεί να είναι ένα ομοαξονικό καλώδιο, ζεύγος τηλεφωνικών συρμάτων, ραδιοκύματα κτλ. Επίσης στο φυσικό επίπεδο συμπεριλαμβάνονται οι ραδιοσυχνότητες ακόμα και τα είδη των διαμορφώσεων σε πομπό και δέκτη.

2. Data link (Επίπεδο συνδέσμου μετάδοσης δεδομένων) : Το επίπεδο αυτό χειρίζεται τη μετάδοση των δεδομένων στο σύνδεσμο που ορίζει το φυσικό επίπεδο. Επίσης καθορίζει τη μορφή κάθε πακέτου δεδομένων που κινείται μέσα στο δίκτυο, το προορισμό κάθε πακέτου, τη φυσική δομή του δικτύου, την ακολουθία των πακέτων και το τύπο του ελέγχου ροής που διασφαλίζει ότι ο

πομπός δε θα στέλνει δεδομένα πιο γρήγορα από ό,τι μπορεί να χειριστεί ο παραλήπτης.

3. Network (Επίπεδο δικτύου) : Το επίπεδο αυτό καθορίζει το δρομολόγιο που χρησιμοποιεί κάποιο σήμα για να κινηθεί από τη πηγή στο προορισμό ανεξάρτητα από το φυσικό μέσο. Στο Internet η ανταλλαγή δεδομένων μεταξύ των LAN, WAN γίνεται στο επίπεδο δικτύου.

4. Transport (Επίπεδο μεταφοράς) : Με αρχή το επίπεδο μεταφοράς, το OSI ασχολείται με την επικοινωνία μεταξύ των προγραμμάτων σε δυο διαφορετικούς υπολογιστές. Η σύνδεση ανάμεσα στο φυλλομετρητή (browser) και το διακομιστή συμβαίνει στο επίπεδο μεταφοράς.

5. Session (Επίπεδο συνδιάλεξης) : Το επίπεδο αυτό ορίζει τη μορφή η οποία χρησιμοποιείται από τα προγράμματα που συνδέονται μέσω του επιπέδου μεταφοράς για την ανταλλαγή δεδομένων. Πχ. Αν έχουμε ζητήματα πιστοποίησης ταυτότητας, αυτή γίνεται στο επίπεδο συνδιάλεξης.

6. Presentation (Επίπεδο παρουσίασης) : Το επίπεδο αυτό ελέγχει τον τρόπο με τον οποίο κάθε υπολογιστής χειρίζεται το κείμενο, τον ήχο, το βίντεο και άλλες μορφές δεδομένων.

7. Application (Επίπεδο εφαρμογής) : Το επίπεδο αυτό χειρίζεται τις εντολές και τα δεδομένα που κινούνται στο δίκτυο. Πχ. Όταν στέλνουμε email, το περιεχόμενο του μηνύματος βρίσκεται στο επίπεδο εφαρμογής.

1.3.5. Βασικές έννοιες Διαδικτύου (Internet)

Internet : Το Διαδίκτυο είναι ένα παγκόσμιο δίκτυο ευρείας περιοχής που συνδέει συστήματα υπολογιστών σε όλο τον κόσμο. Περιλαμβάνει πολλές γραμμές δεδομένων υψηλού εύρους ζώνης που περιλαμβάνουν το "ραχοκοκαλιά" του Διαδικτύου. Αυτές οι γραμμές συνδέονται με μεγάλους κόμβους Διαδικτύου που διανέμουν δεδομένα σε άλλες τοποθεσίες, όπως διακομιστές ιστού και ISP.

Για να συνδεθείτε στο Διαδίκτυο, πρέπει να έχετε πρόσβαση σε έναν πάροχο υπηρεσιών Διαδικτύου (ISP), ο οποίος ενεργεί μεσάζων μεταξύ εσάς και του Διαδικτύου. Οι περισσότεροι πάροχοι υπηρεσιών Διαδικτύου προσφέρουν ευρυζωνική πρόσβαση στο Διαδίκτυο μέσω καλωδίου, DSL ή σύνδεσης ινών. Όταν συνδέεστε στο Διαδίκτυο χρησιμοποιώντας ένα δημόσιο σήμα Wi-Fi, ο δρομολογητής Wi-Fi εξακολουθεί να είναι συνδεδεμένος σε έναν ISP που παρέχει πρόσβαση στο Διαδίκτυο. Ακόμη και οι πύργοι δεδομένων κινητής τηλεφωνίας πρέπει να συνδεθούν σε έναν πάροχο υπηρεσιών Διαδικτύου για να παρέχουν σε συνδεδεμένες συσκευές πρόσβαση στο Διαδίκτυο.

Το Διαδίκτυο παρέχει διαφορετικές διαδικτυακές υπηρεσίες. Μερικά παραδείγματα περιλαμβάνουν:

Ιστός - μια συλλογή δισεκατομμυρίων ιστοσελίδων που μπορείτε να δείτε με ένα πρόγραμμα περιήγησης ιστού

Email - η πιο συνηθισμένη μέθοδος αποστολής και λήψης μηνυμάτων στο διαδίκτυο

Κοινωνικά μέσα - ιστότοποι και εφαρμογές που επιτρέπουν στους χρήστες να μοιράζονται σχόλια, φωτογραφίες και βίντεο

Διαδικτυακά τυχερά παιχνίδια - παιχνίδια που επιτρέπουν στους ανθρώπους να παίζουν μεταξύ τους μέσω του Διαδικτύου

Ενημερώσεις λογισμικού - οι ενημερώσεις λειτουργικού συστήματος και εφαρμογών μπορούν συνήθως να ληφθούν από το Διαδίκτυο

Στις πρώτες μέρες του Διαδικτύου, οι περισσότεροι άνθρωποι συνδέθηκαν στο Διαδίκτυο χρησιμοποιώντας έναν οικιακό υπολογιστή και ένα μόντεμ μέσω τηλεφώνου. Τα μόντεμ DSL και καλωδιακά έδωσαν τελικά στους χρήστες συνδέσεις "πάντα ενεργοποιημένες". Τώρα οι κινητές συσκευές, όπως τα tablet και τα smartphone, καθιστούν δυνατή τη σύνδεση των χρηστών στο Διαδίκτυο ανά πάσα στιγμή. Το Διαδίκτυο των πραγμάτων έχει μετατρέψει τις κοινές συσκευές και τα οικιακά συστήματα σε "έξυπνες" συσκευές που μπορούν να παρακολουθούνται και να ελέγχονται μέσω του Διαδικτύου. Καθώς το Διαδίκτυο συνεχίζει να αναπτύσσεται και να εξελίσσεται, μπορείτε να περιμένετε να γίνει ακόμη πιο αναπόσπαστο κομμάτι της καθημερινής ζωής.

TCP/IP : Σημαίνει "Πρωτόκολλο Ελέγχου Μετάδοσης / Πρωτόκολλο Διαδικτύου". Αυτά τα δύο πρωτόκολλα αναπτύχθηκαν στις πρώτες μέρες του Διαδικτύου από τον στρατό των ΗΠΑ. Ο σκοπός ήταν να επιτρέπεται στους υπολογιστές να επικοινωνούν μέσω δικτύων υπεραστικών. Το τμήμα TCP έχει να κάνει με την επαλήθευση της παράδοσης των πακέτων. Το τμήμα IP αναφέρεται στη μετακίνηση πακέτων δεδομένων μεταξύ κόμβων. Το TCP / IP έγινε έκτοτε το θεμέλιο του Διαδικτύου. Επομένως, το λογισμικό TCP / IP είναι ενσωματωμένο σε όλα τα μεγάλα λειτουργικά συστήματα, όπως τα Unix, Windows και το Mac OS.

IP διεύθυνση : Μια διεύθυνση IP ή απλά μια "IP" είναι μια μοναδική διεύθυνση που προσδιορίζει μια συσκευή στο Διαδίκτυο ή σε ένα τοπικό δίκτυο. Επιτρέπει σε ένα σύστημα να αναγνωρίζεται από άλλα συστήματα που συνδέονται μέσω του πρωτοκόλλου Διαδικτύου. Υπάρχουν δύο κύριοι τύποι μορφών διευθύνσεων IP που χρησιμοποιούνται σήμερα - IPv4 και IPv6.

Μια διεύθυνση IPv4 αποτελείται από τέσσερα σύνολα αριθμών από 0 έως 255, διαχωρισμένα με τρεις τελείες. Για παράδειγμα, η διεύθυνση IP του TechTerms.com είναι 67.43.14.98. Αυτός ο αριθμός χρησιμοποιείται για την αναγνώριση του ιστότοπου TechTerms στο Διαδίκτυο. Όταν επισκέπτεστε το <http://techterms.com> στο πρόγραμμα περιήγησης ιστού, το σύστημα DNS

μεταφράζει αυτόματα το όνομα τομέα "techterms.com" στη διεύθυνση IP "67.43.14.98."

Υπάρχουν τρεις κατηγορίες συνόλων διευθύνσεων IPv4 που μπορούν να καταχωρηθούν μέσω του InterNIC. Το μικρότερο είναι το Class C, το οποίο αποτελείται από 256 διευθύνσεις IP (π.χ. 123.123.123.xxx - όπου το xxx είναι 0 έως 255). Το επόμενο μεγαλύτερο είναι η κλάση B, η οποία περιέχει 65.536 διευθύνσεις IP (π.χ. 123.123.xxx.xxx). Το μεγαλύτερο μπλοκ είναι η κλάση A, η οποία περιέχει 16.777.216 διευθύνσεις IP (π.χ. 123.xxx.xxx.xxx).

Ο συνολικός αριθμός διευθύνσεων IPv4 κυμαίνεται από 000.000.000.000 έως 255.255.255.255. Επειδή $256 = 2^8$, υπάρχουν $2^8 \times 4$ ή 4.294.967.296 πιθανές διευθύνσεις IP. Παρόλο που αυτό φαίνεται να είναι μεγάλο αριθμό, δεν αρκεί πλέον να καλύπτει όλες τις συσκευές που είναι συνδεδεμένες στο Διαδίκτυο σε όλο τον κόσμο. Επομένως, πολλές συσκευές χρησιμοποιούν τώρα διευθύνσεις IPv6. Η μορφή διεύθυνσης IPv6 είναι πολύ διαφορετική από τη μορφή IPv4. Περιέχει οκτώ σύνολα τεσσάρων δεκαεξαδικών ψηφίων και χρησιμοποιεί άνω και κάτω τελεία για να διαχωρίσει κάθε μπλοκ. Ένα παράδειγμα διεύθυνσης IPv6 είναι: 2602:0445:0000:0000:a93e:5ca7:81e2:5f9d. Υπάρχουν $3,4 \times 10^{38}$ ή 340 undecillion) πιθανές διευθύνσεις IPv6, που σημαίνει ότι δεν θα πρέπει να εξαντλήσουμε τις διευθύνσεις IPv6 σύντομα.

DNS : Σημαίνει "Σύστημα Ονομάτων Τομέα". Τα ονόματα τομέα χρησιμεύουν ως απομνημόνευτα ονόματα για ιστότοπους και άλλες υπηρεσίες στο Διαδίκτυο. Ωστόσο, οι υπολογιστές έχουν πρόσβαση σε συσκευές Διαδικτύου από τις διευθύνσεις IP τους. Το DNS μεταφράζει τα ονόματα τομέα σε διευθύνσεις IP, επιτρέποντάς σας να αποκτήσετε πρόσβαση σε μια τοποθεσία στο Διαδίκτυο με το όνομα του τομέα.

Χάρη στο DNS, μπορείτε να επισκεφθείτε έναν ιστότοπο πληκτρολογώντας το όνομα τομέα και όχι τη διεύθυνση IP. Για παράδειγμα, για να επισκεφθείτε το Λεξικό Υπολογιστών Όρων Τεχνολογίας, μπορείτε απλά να πληκτρολογήσετε "techterms.com" στη γραμμή διευθύνσεων του προγράμματος περιήγησης ιστού αντί για τη διεύθυνση IP (67.43.14.98). Απλοποιεί επίσης τις διευθύνσεις email, αφού το DNS μεταφράζει το όνομα τομέα (ακολουθώντας το σύμβολο "@") στην κατάλληλη διεύθυνση IP.

Γλώσσα HTML : Σημαίνει "Γλώσσα Σήμανσης Υπερκειμένου". Η HTML είναι η γλώσσα που χρησιμοποιείται για τη δημιουργία ιστοσελίδων. Το "υπερκειμένου" αναφέρεται στους υπερσυνδέσμους που μπορεί να περιέχει μια σελίδα HTML. Η "γλώσσα σήμανσης" αναφέρεται στον τρόπο με τον οποίο χρησιμοποιούνται οι ετικέτες για τον καθορισμό της διάταξης και των στοιχείων της σελίδας.

Ακολουθεί ένα παράδειγμα HTML που χρησιμοποιείται για τον καθορισμό μιας βασικής ιστοσελίδας με τίτλο και μία μόνο παράγραφο κειμένου.

```
<!doctype html>  
<html>
```

```
<κεφάλι>  
<title> TechTerms.com </title>  
</head>  
<άτομο>  
<p> Αυτό είναι ένα παράδειγμα μιας παραγράφου σε HTML. </p>  
</body>  
</html>
```

Η πρώτη γραμμή καθορίζει τον τύπο περιεχομένου που περιέχει το έγγραφο. "<!doctype html>" σημαίνει ότι η σελίδα είναι γραμμένη σε HTML5. Οι σωστά μορφοποιημένες σελίδες HTML πρέπει να περιλαμβάνουν ετικέτες <html>, <head> και <body>, οι οποίες περιλαμβάνονται στο παραπάνω παράδειγμα. Ο τίτλος της σελίδας, τα μεταδεδομένα και οι σύνδεσμοι προς τα αρχεία αναφοράς τοποθετούνται μεταξύ των ετικετών <head>. Το πραγματικό περιεχόμενο της σελίδας βρίσκεται ανάμεσα στις ετικέτες <body>.

Ο ιστός έχει περάσει από πολλές αλλαγές τις τελευταίες δεκαετίες, αλλά η HTML ήταν πάντα η βασική γλώσσα που χρησιμοποιείται για την ανάπτυξη ιστοσελίδων. Είναι ενδιαφέρον, ενώ οι ιστότοποι έχουν γίνει πιο προηγμένοι και διαδραστικοί, το HTML έχει γίνει απλούστερο. Εάν συγκρίνετε την πηγή μιας σελίδας HTML5 με μια παρόμοια σελίδα γραμμένη σε HTML 4.01 ή XHTML 1.0, η σελίδα HTML5 θα περιέχει πιθανώς λιγότερο κώδικα. Αυτό συμβαίνει επειδή το σύγχρονο HTML βασίζεται σε επικαλυπτόμενα φύλλα στυλ ή JavaScript για τη μορφοποίηση σχεδόν όλων των στοιχείων σε μια σελίδα.

MAC διεύθυνση : Σημαίνει "Διεύθυνση Ελέγχου Πρόσβασης Πολυμέσων" και δεν σχετίζεται με υπολογιστές Apple Macintosh. Η διεύθυνση MAC είναι ένας αριθμός αναγνώρισης υλικού που προσδιορίζει μοναδικά κάθε συσκευή σε ένα δίκτυο. Η διεύθυνση MAC κατασκευάζεται σε κάθε κάρτα δικτύου, όπως κάρτα Ethernet ή κάρτα Wi-Fi, και ως εκ τούτου δεν μπορεί να αλλάξει.

Επειδή υπάρχουν εκατομμύρια συσκευές με δυνατότητα δικτύου και κάθε συσκευή πρέπει να έχει μια μοναδική διεύθυνση MAC, πρέπει να υπάρχει ένα πολύ ευρύ φάσμα πιθανών διευθύνσεων. Για το λόγο αυτό, οι διευθύνσεις MAC αποτελούνται από έξι διψήφιο δεκαεξαδικό αριθμό, χωρισμένο με άνω και κάτω τελεία. Για παράδειγμα, μια κάρτα Ethernet μπορεί να έχει διεύθυνση MAC 00:0d:83:b1:c0:8e. Ευτυχώς, δεν χρειάζεται να γνωρίζετε αυτήν τη διεύθυνση, καθώς αναγνωρίζεται αυτόματα από τα περισσότερα δίκτυα.

1.3.6. Ασύρματη δικτύωση

Η μεταφορά δεδομένων μέσω ασύρματου δικτύου περιλαμβάνει τρία ξεχωριστά στοιχεία : τα ραδιοκύματα, τη μορφή των δεδομένων και τη δομή του δικτύου. Στο μοντέλο OSI, το ραδιοκύμα (ή ραδιοσήμα) λειτουργεί στο φυσικό επίπεδο και η μορφή των δεδομένων ελέγχει πολλά από τα υψηλότερα επίπεδα. Η δομή του δικτύου περιλαμβάνει τους ασύρματους προσαρμογείς διασύνδεσης δικτύου και τους σταθμούς βάσης.

Στα ασύρματα δίκτυα (wireless networks), οι προσαρμογείς διασύνδεσης δικτύου μετατρέπουν τα ψηφιακά δεδομένα σε ραδιοσήματα, τα οποία και μεταδίδουν σε άλλες συσκευές του ίδιου δικτύου, λαμβάνουν και μετατρέπουν ξανά τα εισερχόμενα ραδιοσήματα από άλλα στοιχεία του δικτύου σε ψηφιακά δεδομένα.

1.3.6.α Ραδιοκύματα

Οι βασικοί νόμοι της Φυσικής που καθιστούν εφικτά τα ραδιοκύματα είναι γνωστοί ως εξισώσεις Maxwell. Αυτές οι εξισώσεις δείχνουν ότι ένα μεταβαλλόμενο μαγνητικό πεδίο έντασης H και μαγνητικής πυκνότητας ροής B παράγει ηλεκτρικό πεδίο έντασης E και ηλεκτρικής πυκνότητας ροής D . Συμβαίνει και το ανάποδο. Όταν ένας αγωγός διαρρέεται από AC ρεύμα, ένα μέρος της ενέργειας διαφεύγει στο περιβάλλον ως εναλλασσόμενο μαγνητικό πεδίο. Αυτό με τη σειρά του δημιουργεί εναλλασσόμενο ηλεκτρικό πεδίο στο χώρο και ούτω καθεξής.

Ηλεκτρομαγνητικά κύματα ή ραδιοκύματα είναι ,λοιπόν, διαταραχές που μεταφέρουν ενέργεια από ένα σημείο του χώρου σε άλλο με χωρική και χρονική μεταβολή της έντασης του ηλεκτρικού και του μαγνητικού πεδίου χωρίς να χρειάζεται να υπάρχει κάποιο μέσο.

$$\oint \mathbf{E} \cdot d\mathbf{A} = \frac{Q}{\epsilon_0}$$

$$\oint \mathbf{B} \cdot d\mathbf{A} = 0$$

$$\oint \mathbf{E} \cdot d\mathbf{s} = -\frac{d\Phi_m}{dt}$$

$$\oint \mathbf{B} \cdot d\mathbf{s} = \mu_0 I + \epsilon_0 \mu_0 \frac{d\Phi_e}{dt}$$

Εικόνα 22. Εξισώσεις Maxwell

Η συσκευή που παράγει ραδιοκύματα λέγεται πομπός ενώ η συσκευή που ανιχνεύει ραδιοκύματα στον αέρα και τα μετατρέπει σε κάποια άλλη μορφή ενέργειας λέγεται δέκτης. Στους πομπούς αλλά και στους δέκτες υπάρχουν διατάξεις με ειδικό σχήμα για να εστιάσουν το σήμα σε μια συγκεκριμένη κατεύθυνση. Αυτές οι διατάξεις λέγονται κεραιές. Οι ραδιοσυχνότητες (RF) εκφράζονται σε κύκλους ανά δευτερόλεπτο δηλ. σε Hertz, Hz (πολλαπλάσια KHz, MHz, GHz) και μας δείχνουν πόσο γρήγορα εκπέμπονται τα ραδιοκύματα από το πομπό. Ισχύει και η θεμελιώδης εξίσωση της Κυματικής $\lambda = c \cdot f$, όπου λ το μήκος κύματος της ακτινοβολίας, c η ταχύτητα φωτός στο κενό ($c = 3 \cdot 10^8 \text{ m/sec}$) και f η συχνότητα του κύματος.

1.3.2.β Ασύρματα δίκτυα δεδομένων

Κάθε τύπος ασύρματου δικτύου δεδομένων λειτουργεί σε συγκεκριμένο σύνολο ραδιοσυχνοτήτων. Πχ. Τα περισσότερα δίκτυα Wi-Fi λειτουργούν σε μια ειδική ζώνη γύρω στα 2,4 GHz, η οποία έχει δεσμευτεί στα περισσότερα μέρη του κόσμου για απαλλασσόμενες από άδεια χρήσης ραδιοπηρεσίες εξάπλωσης φάσματος από σημείο σε σημείο. Άλλα συστήματα Wi-Fi κάνουν χρήση διαφορετικής απαλλασσόμενης από άδεια χρήσης ζώνης γύρω στα 5 GHz.

Απαλλασσόμενες από άδεια χρήσης σημαίνει ότι όσοι χρησιμοποιούν εξοπλισμό που συμμορφώνεται με τις τεχνικές απαιτήσεις, μπορούν να στείλουν

και να λάβουν ραδιοσήματα σε αυτές τις συχνότητες χωρίς άδεια χρήσης ραδιοσταθμού. Μια υπηρεσία ραδιοκυμάτων από σημείο σε σημείο χειρίζεται ένα κανάλι επικοινωνίας που μεταφέρει πληροφορίες από ένα πομπό σε ένα μόνο δέκτη. Η **εξάπλωση φάσματος** αποτελεί ένα σύνολο μεθόδων για τη μετάδοση ενός ραδιοσήματος με τη χρήση ενός σχετικά μεγάλου εύρους τμήματος του ραδιοφάσματος. Τα ασύρματα δίκτυα Ethernet (Wireless LAN, WLAN) χρησιμοποιούν διαφορετικά συστήματα ραδιομετάδοσης εξάπλωσης φάσματος, τα οποία λέγονται εξάπλωση φάσματος με συνεχή αλλαγή συχνότητας (frequency-hopping spread spectrum, FHSS), εξάπλωση φάσματος με άμεση ακολουθία (direct-sequence spread spectrum, DSSS) και ορθογωνική πολύπλεξη με διαίρεση συχνότητας (orthogonal frequency division multiplexing, OFDM).

1.3.2.γ Ασύρματες υπηρεσίες δεδομένων

Επειδή τα ραδιοκύματα κινούνται στον αέρα, μπορούμε να διαμορφώσουμε μια σύνδεση δικτύου σε οποιοδήποτε σημείο στην εμβέλεια του πομπού. Με το που ανάβουμε τον ασύρματο προσαρμογέα του υπολογιστή μας, εκείνος εντοπίζει το σήμα του δικτύου. Παρόλο που στην ασύρματη δικτύωση δεν χρειάζονται σύρματα και καλώδια, η ασύρματη τεχνολογία δεν είναι τόσο ασφαλή όσο η ενσύρματη. Στην ασύρματη, λοιπόν, υπάρχουν πομποί, δέκτες, τα ραδιοκύματα και όλα μαζί λειτουργούν στις ίδιες συχνότητες με το ίδιο είδος διαμόρφωσης. Είναι διαθέσιμα πολλά διαφορετικά ασύρματα συστήματα και υπηρεσίες δεδομένων για τη σύνδεση υπολογιστών και άλλων συσκευών με τα τοπικά δίκτυα και το Internet όπως το Wi-Fi, WiMAX και τα δίκτυα κινητής τηλεφωνίας (2G, 3G, 4G, 4G+).

Wi-Fi : Για τα ασύρματα δίκτυα, το ίδρυμα IEEE (Institute of Electrical and Electronic Engineers) έχει δημιουργήσει ένα σύνολο προτύπων και προδιαγραφών, με τίτλο IEEE 802.11 και αυτά τα πρότυπα ορίζουν τις κορφές και τις δομές των σημάτων σχετικά μικρής εμβέλειας, τα οποία παρέχουν υπηρεσία Wi-Fi. Οι προδιαγραφές με τη πιο μεγάλη χρήση σήμερα είναι οι 802.11a, 802.11b, 802.11g και 802.11n. Χρησιμοποιούνται από όλα τα ασύρματα LAN και μάλιστα το τελευταίο πρότυπο, το 802.11n είναι το πιο ταχύ και πιο ασφαλές. Το Wi-Fi προοριζόταν αρχικά ως ασύρματη επέκταση ενός ενσύρματου LAN επομένως οι αποστάσεις μεταξύ των σταθμών βάσης Wi-Fi

και των υπολογιστών που επικοινωνούν μέσω αυτών περιορίζονται σε 35 μέτρα ή μέχρι 100 μέτρα σε εξωτερικούς χώρους χωρίς όμως φυσικά εμπόδια.

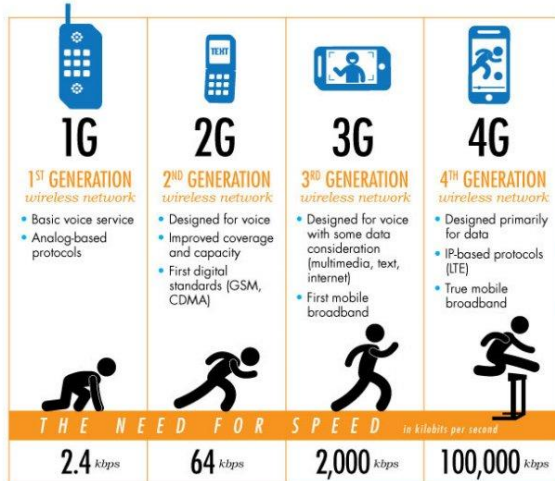
Μητροπολιτικές υπηρεσίες Wi-Fi : Σε κάποιες μητροπολιτικές περιοχές εγκαθίσταται μεγάλος αριθμός από διασυνδεδεμένους σταθμούς βάσης Wi-Fi, από δημόσιες υπηρεσίες ή από ιδιωτικές επιχειρήσεις για τη παροχή ασύρματης σύνδεσης στο Internet σε μια ολόκληρη περιοχή ή σε επιλεγμένες γειτονίες ως εναλλακτική λύση στις καλωδιακές και τις τηλεφωνικές (DSL) υπηρεσίες.

Ασύρματα δίκτυα κινητής τηλεφωνίας : Το μεγάλο πλεονέκτημα της κινητής ευρυζωνικής υπηρεσίας για εκείνους που χρησιμοποιούν τους υπολογιστές μακριά από το σπίτι είναι ότι καλύπτει πολύ μεγαλύτερη περιοχή από οποιονδήποτε σταθμό βάσης Wi-Fi. Βέβαια η τεχνολογία των υπολογιστών βελτιώνεται επίσης ταυτόχρονα , επομένως τα σημερινά κινητά τηλέφωνα (smartphones) 2.5G και 3G συχνά ενσωματώνουν αρκετή υπολογιστική ισχύ που τους επιτρέπει να χρησιμεύουν και ως τερματικά Διαδικτύου μεγέθους τσέπης.

Γενιές	Χωρητικότητα	Περιγραφή
1G	Χαμηλή	Αναλογικά κυψελωτά δίκτυα για υπηρεσίες φωνής
2G	10-14 kbps	Ψηφιακά ασύρματα δίκτυα, κυρίως για υπηρεσίες φωνής με περιορισμένη δυνατότητα μετάδοσης δεδομένων
2.5G	50-144 kbps	Ενδιάμεσο βήμα για το 3G στις ΗΠΑ
3G	144 kbps - 2+ Mbps	Υψηλή ταχύτητα, υποστήριξη video, email, περιήγηση στο διαδίκτυο

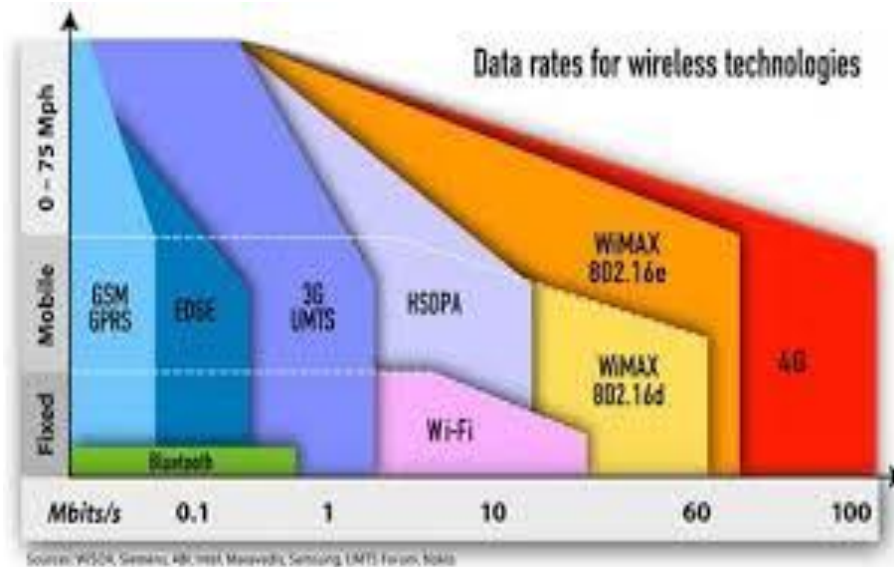
Εικόνα 23. Γενιές κυψελωτής κινητής τηλεφωνίας

EVOLUTION OF THE G



Εικόνα 24. Εξέλιξη γενεών κινητής τηλεφωνίας

WiMAX : Η Παγκόσμια Διαλειτουργικότητα για Πρόσβαση Μέσω Μικροκυμάτων (Worldwide Interoperability for Microwave Access-WiMAX) είναι μια ακόμη μέθοδος διανομής ασύρματων δεδομένων σε μεγάλες γεωγραφικές περιοχές. Αποτελεί υπηρεσία του MAN και το IEEE πρότυπό του είναι το 802.16. Όλοι οι πάροχοι υπηρεσιών του WiMAX χρησιμοποιούν μία ή περισσότερες συχνότητες λειτουργίας με άδεια χρήσης κάπου μεταξύ 2 και 11 GHz και οι ταχύτητες δεδομένων φτάνουν μέχρι τα 70 Mbps.



Εικόνα 25. Ρυθμοί δεδομένων ασύρματων τεχνολογιών

1.3.7. Λειτουργία τεχνολογίας Wi-Fi

Το ασύρματο Ethernet, γνωστό και ως Wi-Fi, χρησιμοποιεί ραδιοκύματα εξάπλωσης φάσματος για τη διανομή των δεδομένων σε τοπικά δίκτυα. Οι προδιαγραφές Wi-Fi ελέγχουν τον τρόπο που κινούνται τα δεδομένα μέσω του φυσικού επιπέδου και ορίζουν ένα επίπεδο πρόσβασης μέσων (media access control – MAC) που χειρίζεται τη διασύνδεση ανάμεσα στο φυσικό επίπεδο και την υπόλοιπη δομή του δικτύου.

Φυσικό επίπεδο : Σε ένα ασύρματο δίκτυο (802.11) ο πομπός προσθέτει ένα προοίμιο (δυναδική ακολουθία δεδομένων) των 144 bits σε κάθε πακέτο από τα οποία τα 128 bits χρησιμοποιούνται από το δέκτη για συγχρονισμό με το πομπό και ένα πεδίο αρχής πλαισίου των 16 bits. Το προοίμιο ακολουθείται από μια κεφαλίδα 48 bits που περιέχει πληροφορίες για τη ταχύτητα μεταφοράς των δεδομένων, το μήκος των δεδομένων του πακέτου και μια ακολουθία ελέγχου σφαλμάτων. Η κεφαλίδα αυτή λέγεται PHY επειδή ελέγχει το φυσικό επίπεδο του συνδέσμου επικοινωνίας. Το προοίμιο και η κεφαλίδα μεταδίδονται με ταχύτητα 1 Mbps και συνεπώς ακόμα και αν κάποια σύνδεση δικτύου λειτουργεί πχ στα 11 Mbps, η πραγματική ταχύτητα μεταφοράς δεδομένων είναι πιο μικρή. Το προοίμιο των 144 bits είναι από παλιότερα σήματα DSSS για να διασφαλιστεί ότι οι συσκευές 802.11 θα παραμείνουν συμβατές με τα παλαιότερα πρότυπα.

Επίπεδο MAC : Στο μοντέλο OSI, το επίπεδο MAC ελέγχει τη κυκλοφορία των δεδομένων μέσω του δικτύου ραδιοκυμάτων. Αποτρέπει τις συγκρούσεις και τις διενέξεις των δεδομένων με τη χρήση ενός συνόλου κανόνων που λέγονται πολλαπλή προσπέλαση με ανίχνευση φέροντος σήματος και αποφυγή σύγκρουσης (carrier sense multiple access/collision avoidance – CSMA/CA). Υποστηρίζει τις λειτουργίες ασφαλείας και όταν το δίκτυο περιλαμβάνει περισσότερα από ένα σημεία πρόσβασης, το επίπεδο MAC συσχετίζει κάθε πελάτη δικτύου με το σημείο πρόσβασης που παρέχει τη καλύτερη ποιότητα σήματος.

1.4. CLOUD COMPUTING

1.4.1. Ορισμός

Το **cloud computing** (Υπολογιστικό Σύννεφο) ονομάζεται η κατ' αίτηση διαδικτυακή κεντρική διάθεση υπολογιστικών πόρων (όπως δίκτυο, εξυπηρετητές, εφαρμογές και υπηρεσίες) με υψηλή ευελιξία, ελάχιστη προσπάθεια από τον χρήστη και υψηλή αυτοματοποίηση. Στο Υπολογιστικό Σύννεφο η αποθήκευση, η επεξεργασία και η χρήση δεδομένων, λογισμικού και υπηρεσιών γίνεται διαδικτυακά, μέσω απομακρυσμένων υπολογιστών σε κεντρικά datacenter. Υπηρεσίες όπως η κατ' αίτηση παροχή εικονικών μηχανών, το διαδικτυακό ηλεκτρονικό ταχυδρομείο ή τα κοινωνικά δίκτυα συχνά βασίζονται στην τεχνολογία του Υπολογιστικού Σύννεφου.

Οι χρήστες εξοικονομούν πόρους από την αγορά και συντήρηση λογισμικού, τη συντήρηση ακριβών εξυπηρετητών και εγκαταστάσεων αποθήκευσης δεδομένων.



Εικόνα 26. Cloud computing

1.4.2. Μοντέλα Cloud computing

Το cloud computing έχει τρία μοντέλα – υπηρεσίες, οι οποίες περιγράφονται παρακάτω.

IaaS (Infrastructure As A Service)

Το IaaS είναι το πρότυπο υπηρεσίας cloud με το χαμηλότερο επίπεδο και αναμφισβήτητα το πιο σημαντικό. Με το IaaS, οι προρυθμισμένοι πόροι υλικού παρέχονται στους χρήστες μέσω μιας εικονικής διεπαφής. Σε αντίθεση με τα προγράμματα PaaS και SaaS, το IaaS δεν περιλαμβάνει εφαρμογές ή ακόμα και λειτουργικό σύστημα (υλοποιώντας όλα αυτά που παραμένουν στον πελάτη), επιτρέπει απλώς την πρόσβαση στην υποδομή που απαιτείται για την τροφοδοσία ή την υποστήριξη αυτού του λογισμικού. Το IaaS μπορεί να παρέχει επιπλέον αποθηκευτικό χώρο για την υποστήριξη εταιρικών δεδομένων, το εύρος ζώνης δικτύου για έναν διακομιστή ιστότοπου της εταιρείας ή ακόμα και

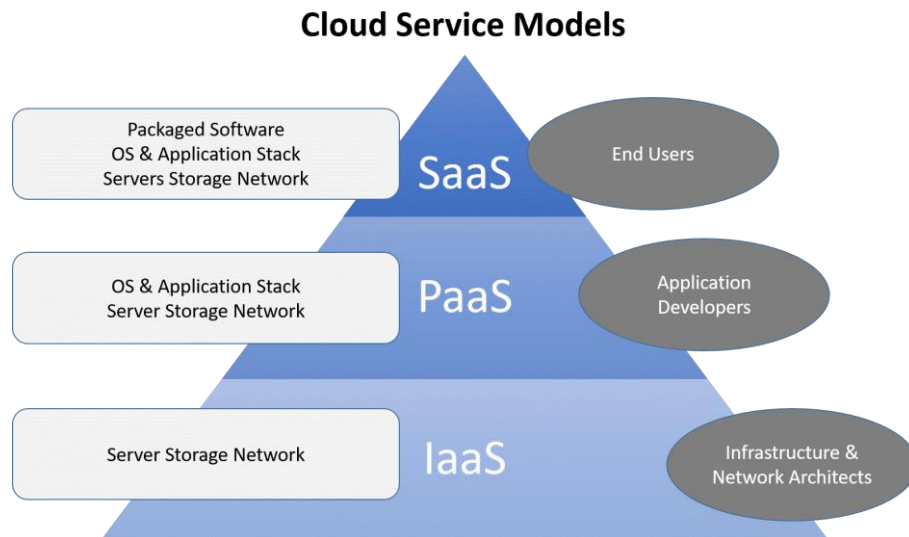
να επιτρέπει την πρόσβαση σε υπολογιστές υψηλής ισχύος, τα οποία προηγουμένως ήταν προσβάσιμα μόνο σε άτομα με υπερυπολογιστές. Οι δημοφιλείς προσφορές του IaaS, όπως το Amazon EC2, το IBM SoftLayer και το Compute Engine της Google (GCE), τροφοδοτούν σιωπηλά ένα τεράστιο τμήμα της ραχοκοκαλιάς του Διαδικτύου, είτε το συνειδητοποιούν είτε όχι.

PaaS (Platform As A Service)

Το PaaS είναι το μοντέλο υπηρεσίας σύννεφο όπου το cloud χρησιμοποιείται για να παρέχει μια πλατφόρμα στους χρήστες από τους οποίους μπορούν να αναπτύξουν, να αρχικοποιήσουν και να διαχειριστούν εφαρμογές. Οι προσφορές PaaS περιλαμβάνουν συνήθως ένα βασικό λειτουργικό σύστημα και μια σειρά εφαρμογών και εργαλείων ανάπτυξης. Το PaaS εξαλείφει την ανάγκη οι οργανισμοί να δημιουργήσουν και να διατηρήσουν την υποδομή που χρησιμοποιείται παραδοσιακά για την ανάπτυξη εφαρμογών. Το PaaS ονομάζεται μερικές φορές «μεσαία», αναφερόμενος στον τρόπο με τον οποίο τοποθετείται εννοιολογικά κάπου μεταξύ του SaaS και του IaaS. Οι πλατφόρμες όπως το App Engine της Google, το IBM BlueMix και το Stratos της Apache είναι δημοφιλή προϊόντα PaaS.

SaaS (Software As A Service)

Αν και μερικές φορές αναφέρεται ως "λογισμικό κατά παραγγελία", το SaaS είναι ένα μοντέλο χορήγησης άδειας και παράδοσης λογισμικού, όπου ένα πλήρως λειτουργικό και πλήρες προϊόν λογισμικού παραδίδεται στους χρήστες μέσω του διαδικτύου σε συνδρομή. Οι προσφορές του SaaS είναι συνήθως προσπελάσιμες από τους τελικούς χρήστες μέσω ενός προγράμματος περιήγησης ιστού (κάνοντας το λειτουργικό σύστημα του χρήστη σε μεγάλο βαθμό άσχετο) και μπορεί να χρεωθεί με βάση την κατανάλωση ή, απλά, με μια επίπεδη μηνιαία χρέωση. Οι προσφορές του SaaS είναι το πιο ευδιάκριτο από όλα τα μοντέλα υπηρεσιών υπολογιστικού νέφους. Στην πραγματικότητα, πολλοί χρήστες ενδέχεται να χρησιμοποιούν προϊόντα SaaS χωρίς να το συνειδητοποιούν. Δημοφιλή προϊόντα, όπως το Office365 και το Salesforce, προσφέρουν τις προσφορές SaaS στο προσκήνιο του χώρου εργασίας και χρησιμοποιούνται καθημερινά από χιλιάδες επιχειρήσεις.



Εικόνα 27. Υπηρεσίες cloud σε επίπεδα

1.4.3. Τύποι Cloud

Ανάλογα με τον τύπο δεδομένων με το οποίο δουλεύουμε υπάρχουν δημόσια, ιδιωτικά και υβριδικά cloud όσον αφορά τα διαφορετικά επίπεδα ασφάλειας και διαχείρισης που απαιτούνται.

Public Cloud : Ολόκληρη η υποδομή υπολογιστών βρίσκεται στις εγκαταστάσεις μιας εταιρείας cloud computing που προσφέρει την υπηρεσία cloud.

Private Cloud : Στη περίπτωση αυτή φιλοξενούμε μόνοι μας όλη την υποδομή υπολογιστών και δεν γίνεται κοινή χρήση. Το επίπεδο ασφάλειας και ελέγχου είναι το υψηλότερο κατά τη χρήση ενός ιδιωτικού δικτύου.

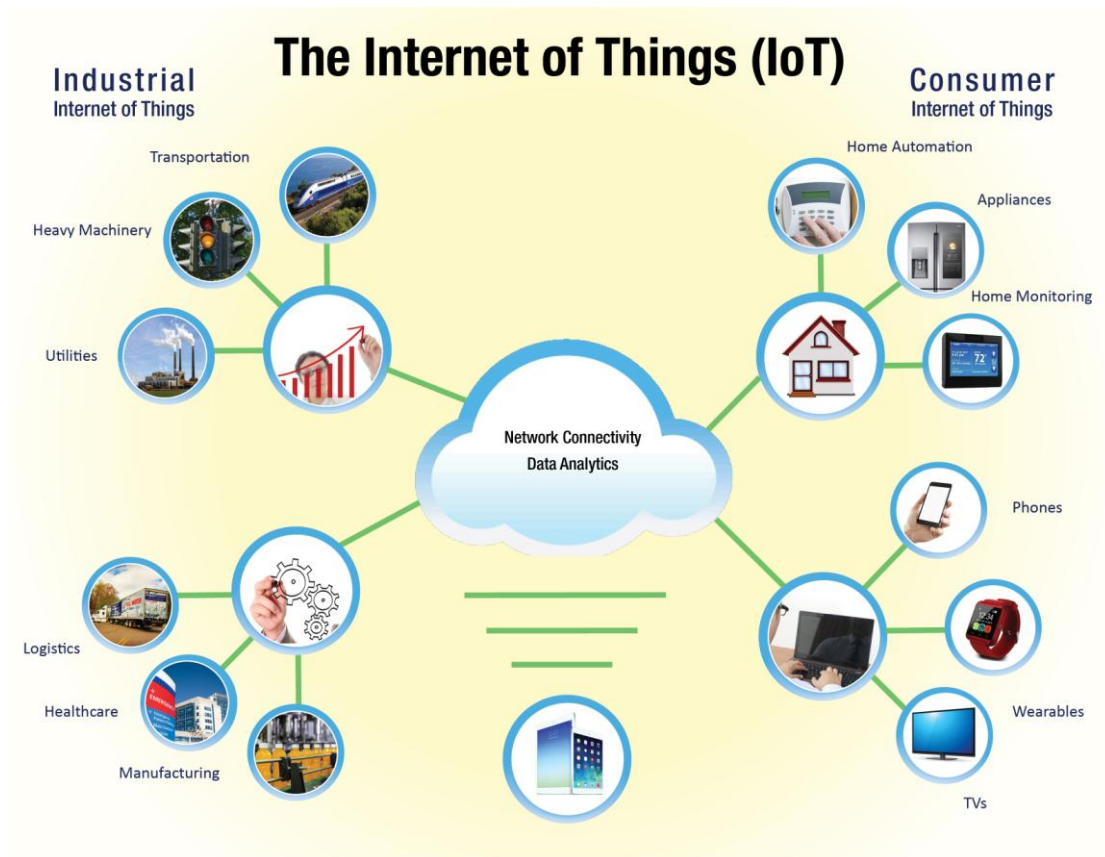
Hybrid Cloud : Χρησιμοποιώντας ιδιωτικά και δημόσια clouds, ανάλογα με τον σκοπό τους. Φιλοξενούμε τις πιο σημαντικές εφαρμογές μας στους δικούς μας διακομιστές για να τις κρατήσουμε πιο ασφαλείς και δευτερεύουσες εφαρμογές αλλού.

Community Cloud : Ένα cloud κοινότητας κοινοποιείται μεταξύ οργανισμών με κοινό στόχο ή που ταιριάζει σε μια συγκεκριμένη κοινότητα (επαγγελματική κοινότητα, γεωγραφική κοινότητα κ.λπ.).

1.5. Internet of Things (IoT)

1.5.1. Ορισμός

Με τον όρο IoT εννοούμε την διαδικτύωση φυσικών συσκευών, οχημάτων, κτιρίων και άλλων αντικειμένων με ενσωματωμένα ηλεκτρονικά συστήματα, λογισμικό, αισθητήρες, ενεργοποιητές (actuators) και σύνδεση Διαδικτύου που επιτρέπει σε όλα αυτά τα αντικείμενα να συλλέγουν και να ανταλλάσσουν δεδομένα μεταξύ τους. Επιπλέον, ο όρος IoT επινοήθηκε για να περιγράψει ένα πλήθος τεχνολογιών και ερευνητικών κλάδων που επιτρέπουν στο Διαδίκτυο να προσεγγίσει τα φυσικά αντικείμενα του πραγματικού κόσμου.



Εικόνα 28. Internet Of Things

1.5.2. Ασύρματες τεχνολογίες στο IoT

1. Bluetooth : Μια σημαντική τεχνολογία μικρής εμβέλειας επικοινωνιών είναι το Bluetooth. Είναι ο πιο γνωστός τρόπος σύνδεσης για φορητά προϊόντα που είναι ειδικά για σύνδεση με το IoT συνήθως με χρήση ενός smartphone ή tablet.

2. ZigBee: Το ZigBee διαθέτει μια μεγάλη εγκατεστημένη βάση λειτουργίας αν και παραδοσιακά χρησιμοποιούνταν περισσότερο για βιομηχανικές εφαρμογές. Βασίζεται στο πρωτόκολλο 802.15.4 του Ινστιτούτου Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών- Institute of Electrical and Electronics Engineers (IEEE) το οποίο είναι ένα βιομηχανικό πρότυπο ασύρματης δικτύωσης, που λειτουργεί στα 2.4 GHz και στοχεύει σε εφαρμογές που απαιτούν σχετικά σπάνιες ανταλλαγές δεδομένων με χαμηλές ροές σε μια περιορισμένη περιοχή και μέσα σε ένα εύρος 100 μέτρων, όπως π.χ. σε ένα σπίτι ή ένα κτίριο.

3. Z-Wave : Το Z-Wave είναι μια Radio Frequency (RF) τεχνολογία επικοινωνιών χαμηλής ισχύος που έχει σχεδιαστεί κυρίως για αυτοματοποίηση λειτουργιών σε οικιακό περιβάλλον. Είναι βελτιστοποιημένο για αξιόπιστη και χαμηλής καθυστέρησης επικοινωνία μικρών πακέτων δεδομένων με ταχύτητες μεταφοράς δεδομένων έως και τα 100 Kbit/s και λειτουργεί κάτω από την ζώνη του 1GHz που είναι αδιαπέραστη από άλλες ασύρματες τεχνολογίες που λειτουργούν στην ζώνη των 2.4 GHz.

4. 6LowPAN : Μια τεχνολογία βασισμένη στο IP πρωτόκολλο είναι το 6LowPan. Αντί να είναι ένα πρωτόκολλο εφαρμογής του IoT, όπως το Bluetooth και το ZigBee, το 6LowPAN είναι ένα πρωτόκολλο δικτύου που ορίζει μηχανισμούς ενθυσιασμού και συμπίεσης κεφαλίδας, έχει ελευθερία επιλογής ζώνης συχνοτήτων και φυσικού επιπέδου και μπορεί να χρησιμοποιηθεί σε πολλαπλές επικοινωνιακές πλατφόρμες, συμπεριλαμβανομένων των Ethernet, 802.11 και 802.15.4.

5. 802.11 : Η συνδεσιμότητα στο IEEE 802.11 είναι συχνά μια προφανής επιλογή για πολλούς προγραμματιστές, αν ειδικά ληφθεί υπόψη η διεισδυτικότητα του στο εσωτερικό περιβάλλον ενός σπιτιού μέσα από τοπικά δίκτυα. Είναι το πρότυπο για ασύρματα δίκτυα (WLAN) και προσφέρει γρήγορη μεταφορά δεδομένων και την ικανότητα διαχείρισης μεγάλης ποσότητας δεδομένων.

6. Cellular : Κάθε IoT εφαρμογή, που απαιτεί λειτουργία πάνω από μεγαλύτερες αποστάσεις μπορεί να εκμεταλλευτεί τις δυνατότητες των κυψελοειδών τηλεπικοινωνιών, όπως π.χ. των Global System for Mobile communications (GSM), της τρίτης-third Generation (3G), της τέταρτης-fourth Generation (4G) και πέμπτης-fifth Generation (5G) γενιάς δικτύων κινητής τηλεφωνίας.

7. NFC : Το Near Field Communication (NFC) είναι μια τεχνολογία που επιτρέπει εύκολη και ασφαλή αμφίδρομη αλληλεπίδραση μεταξύ ηλεκτρονικών συσκευών και ειδικά για smartphones, επιτρέποντας στους καταναλωτές να πραγματοποιούν ανέπαφες πληρωμές, να έχουν πρόσβαση σε ψηφιακό περιεχόμενο και να συνδέουν ηλεκτρονικές συσκευές μεταξύ τους.

8. Sigfox : Μια εναλλακτική τεχνολογία ευρείας κλίμακας είναι το Sigfox, το οποίο από άποψη εμβέλειας είναι μεταξύ του 802.11 και των κυψελοειδών επικοινωνιών. Χρησιμοποιεί τις Industrial, Scientific και Medical (ISM) ζώνες συχνοτήτων, οι οποίες είναι ελεύθερες για χρήση χωρίς την ανάγκη για απόκτηση αδειών.

9. Neul : Το Neul είναι παρόμοιας φιλοσοφίας με το Sigfox και λειτουργεί κάτω από την ζώνη συχνοτήτων του 1GHz. Το Neul αξιοποιεί πολύ μικρά κομμάτια του TeleVision (TV) φάσματος White Space για να παρέχει υψηλή κλιμάκωση, υψηλή κάλυψη, χαμηλή κατανάλωση και χαμηλού κόστους ασύρματα δίκτυα. Τα συστήματα βασίζονται στο Icen1 Chip, το οποίο

επικοινωνεί χρησιμοποιώντας το White Space Radio για να αποκτά πρόσβαση στο υψηλής ποιότητας Ultra-High Frequency (UHF) φάσμα, το οποίο είναι διαθέσιμο λόγω της μετάβασης από την αναλογική στην ψηφιακή τηλεόραση.

10. LoRaWAN : Το LoRaWAN στοχεύει σε WAN εφαρμογές και παρέχει χαμηλής κατανάλωσης WAN δίκτυα με χαρακτηριστικά ειδικά απαιτούμενα για την υποστήριξη ασφαλούς, αμφίδρομης και χαμηλού κόστους κινητή επικοινωνία σε IoT, Machine-to- Machine (M2M), βιομηχανικές και έξυπνων πόλεων εφαρμογές.

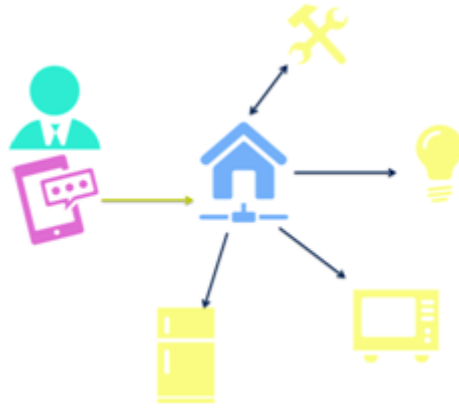
1.5.3. IoT εφαρμογές

Το IoT έχει τη δυνατότητα χρήσης σε όλες τις επικοινωνίες M2M (Machine 2 Machine). Κάποιο σημαντικό τμήμα μπορεί να ταξινομηθεί ως:

- 1) Οικιακός αυτοματισμός.
- 2) Φροντίδα υγείας.
- 3) Βιομηχανικός αυτοματισμός.

Οικιακός αυτοματισμός : Το IoT καθιστά δυνατή τη συνειδητοποίηση της δυνατότητας του Smart Home concept. Ο καταναλωτής μπορεί να παρακολουθεί το σπίτι τους 24 ώρες το 24ωρο από τα Smartphones τους ή μπορεί να δώσει εντολές στις ηλεκτρονικές συσκευές που είναι συνδεδεμένες στην πύλη για να ξυπνήσουν από την κατάσταση ύπνου. Μια ενιαία εφαρμογή μπορεί να χρησιμοποιηθεί για όλα τα συνδεδεμένα gadget που κάνουν τη ζωή του καταναλωτή άνετη. Τα ηλεκτρονικά gadgets μπορούν να είναι κλιματιστικά κλπ. Με την έλευση του BTLE, οι κόμβοι των αισθητήρων θα πρέπει να καταναλώνουν πολύ χαμηλή ισχύ, με αποτέλεσμα να αυξάνεται η διάρκεια ζωής της μπαταρίας.

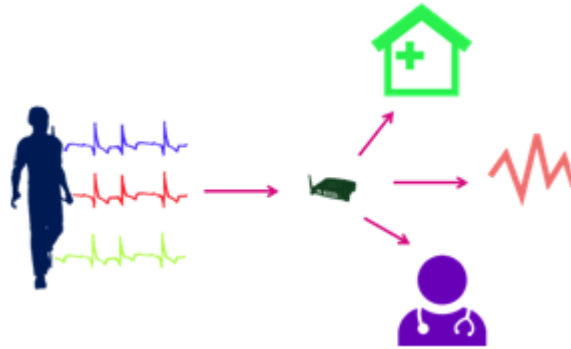
Η ασφάλεια μπορεί να είναι μία από τις μεγαλύτερες ανησυχίες, παρέχοντας ταυτόχρονα τα παραπάνω χαρακτηριστικά στον καταναλωτή. Το σύστημα θα πρέπει να εφαρμόζει χαρακτηριστικά ασφαλείας στο σύστημα, το οποίο να διασφαλίζει ότι κανένας εισβολέας δεν μπορεί να πάρει τον έλεγχο του συστήματος αυτοματισμού στο σπίτι.



Εικόνα 29. Αυτοματισμός με IoT

Φροντίδα υγείας : Πρόκειται για ένα επερχόμενο τμήμα όπου το Διαδίκτυο μπορεί να φέρει επαναστατική αλλαγή για τον τελικό χρήστη. Ο ασθενής μπορεί να συνδεθεί με το γιατρό, νοσοκομείο ή ασφαλιστικές εταιρείες. Πάρτε μια περίπτωση χρήσης όπου ο τελικός χρήστης μετρά τη θερμοκρασία και την αρτηριακή πίεση του σώματος και μεταδίδεται στο προφίλ χρήστη. Το μέσο υγειονομικής περίθαλψης μπορεί να έχει BTLE ή NFC που μεταδίδει δεδομένα στην πύλη και η πύλη μεταφορτώνει περαιτέρω δεδομένα σχετικά με το προσωπικό προφίλ των ασθενών τα οποία μπορούν να παρακολουθούνται από το νοσοκομείο ή το γιατρό. Η ανάγνωση μπορεί να αναλυθεί και να ενσωματωθεί σε περίπτωση έκτακτης ανάγκης, μπορεί να αναληφθεί άμεση δράση για την παροχή άμεσης φροντίδας στον ασθενή. Το σχήμα 3 μιλάει για τη χρήση του συστήματος απομακρυσμένης υγειονομικής περίθαλψης με βάση το IoT.

Το Διαδίκτυο μπορεί να μας βοηθήσει να μεταφέρουμε αποτελεσματικά τις δραστηριότητές μας στον τομέα της υγειονομικής περίθαλψης με κεντρικό σύστημα παρακολούθησης των ασθενών ακολουθώντας τους κανόνες συμμόρφωσης που ορίζει η κυβέρνηση. Αυτό θα βελτιώσει την ομαδική συνεργασία και θα μειώσει το φόρτο εργασίας των παρόχων φροντίδας. Οι προκλήσεις στην υγειονομική περίθαλψη θα μπορούσαν να είναι η ακεραιότητα των δεδομένων, η ανάγκη συμμόρφωσης με τους κανονισμούς της τοπικής κυβέρνησης και πρέπει να είμαστε ανταγωνιστικοί από πλευράς κόστους για να παρέχουμε μια λύση που μπορεί εύκολα να υιοθετηθεί από τον πελάτη μας.



Εικόνα 30. Υγεία με IoT

Βιομηχανικός αυτοματισμός : Οι εφαρμογές είναι πάντα πράσινες εφαρμογές που αναβαθμίζονται με την εμφάνιση της νέας τεχνολογίας. Το Διαδίκτυο μπορεί να διαδραματίσει σημαντικό ρόλο στην παρακολούθηση της βιομηχανικής εφαρμογής. Μπορεί να παρακολουθεί την περιοχή των βιομηχανικών γραφείων, εμπορική φθαρτή εφαρμογή. Υπάρχουν θέσεις στον βιομηχανικό τομέα όπου η ανθρώπινη παρέμβαση είναι σχεδόν αδύνατη και οι αισθητήρες με συνδεσιμότητα παίζουν σημαντικό ρόλο στη συντήρηση της εγκατάστασης.



Εικόνα 31. Αυτοματισμός μέσω IoT

1.5.4. IoT πλατφόρμες (IoT platforms)

1.5.4.α Γενικές πληροφορίες IoT πλατφόρμων

Μια πλατφόρμα IoT είναι μια τεχνολογία λογισμικού πολλαπλών επιπέδων που επιτρέπει την απλή παροχή, διαχείριση και αυτοματοποίηση των συνδεδεμένων συσκευών στο σύμπαν του Διαδικτύου των Πραγμάτων (Internet of Things). Μέσω μιας IoT πλατφόρμας, μπορεί ο οποιοσδήποτε να ελέγξει απομακρυσμένα και σε πραγματικό χρόνο τη κατάσταση των συσκευών, οι οποίες είναι συνδεδεμένες μεταξύ τους, ανταλλάσσουν δεδομένα και τα απεικονίζουν στο Internet. Όλες οι IoT πλατφόρμες έχουν GUI περιβάλλον και έτσι ο χειρισμός τους γίνεται πιο εύκολος.

Για παράδειγμα μια αναπτυξιακή πλακέτα μικροελεγκτή όπως το Raspberry είναι προγραμματισμένο να συλλέγει και να στέλνει δεδομένα σε ένα δίκτυο αισθητήρων και ο χρήστης να παρακολουθεί τη κατάσταση τους σε μια IoT πλατφόρμα. Φυσικά, οι αισθητήρες αλλά και η αναπτυξιακή πλακέτα είναι συνδεδεμένα μεταξύ τους με μια ασύρματη τεχνολογία π.χ. ZigBee, LoRa κτλ.

Σε υψηλό επίπεδο, μπορούμε να εντοπίσουμε λύσεις IoT που περιλαμβάνουν τέσσερα μεγάλα λειτουργικά τμήματα. Εάν κάποιο από αυτά τα μπλοκ λείπει, τότε δεν είναι συνετό να το ονομάσουμε λύση IoT.

Οι συσκευές ("πράγματα") είναι συνήθως φυσικοί αισθητήρες και ενεργοποιητές. Αυτοί να μετρούν διάφορες παραμέτρους και να τις μεταφράζουν σε ηλεκτρικά ή ψηφιακά δεδομένα. Αυτοί οι αισθητήρες είτε συνδέονται με τις κεντρικές συσκευές (τυπικές για αναβαθμίσεις κληρονομιάς) ή ενσωματωμένες στις συσκευές κεντρικού υπολογιστή (μοντέρνες). Αυτές οι συσκευές είναι κρίσιμοι κόμβοι μιας εφαρμογής IoT και απαιτούνται για να παρέχουν λειτουργικότητα πλήρους λύσης ενεργώντας ως είσοδοι, εξόδους ή και τα δύο. Τυπικά παραδείγματα τέτοιων συσκευών είναι οι θερμοστάτες, οι έξυπνες πτητικές συσκευές, τα συνδεδεμένα ψυγεία κ.ο.κ.

Οι πύλες είναι συσκευές ακμής που μπορούν να επικοινωνούν με την ανάντη συστήματος με έναν από τους δύο τρόπους: με ή χωρίς πύλη. Ορισμένες συσκευές έχουν τη δυνατότητα να επικοινωνούν απευθείας μέσω πρωτοκόλλου Internet (IP) χρησιμοποιώντας διάφορα πρωτόκολλα επικοινωνίας, όπως REST, MQTT, AMQP, CoAP, και ούτω καθεξής. Αυτές οι δυνατότητες είναι συνήθως αποτέλεσμα ολοκληρωμένων δομοστοιχείων επικοινωνίας, όπως τα τσιπ WiFi ή GSM, τα οποία επιτρέπουν τη χρήση ενός συσκευή για σύνδεση σε πύλες δικτύου, όπως δρομολογητές WiFi και κινητοί πύργοι και επικοινωνούν απευθείας με το στρώμα. Σε αυτές τις περιπτώσεις, οι δρομολογητές και οι κινητοί πύργοι εκτελούν τη δουλειά της πύλης.

Για οποιαδήποτε πλατφόρμα μεσαίου λογισμικού, η οποία αξίζει να είναι μέρος του IoT, είναι επιτακτική ανάγκη να έχει τις ακόλουθες λειτουργίες και δυνατότητες.

- **Επεκτασιμότητα.** Όπως κάθε νέα εφαρμογή ή προϊόν, τα πράγματα αρχίζουν μικρά και στη συνέχεια μεγαλώνουν αργότερα. Επομένως, εάν η πλατφόρμα middleware πρέπει να βρίσκεται στον πυρήνα του, η λύση πρέπει να είναι σε θέση να κλιμακωθεί στο ίδιο ποσοστό.
- **Αξιοπιστία.** Σε γενικές γραμμές, είναι μια προφανής προσδοκία ότι οτιδήποτε αποτελεί τον πυρήνα μιας λύσης ή προϊόντος πρέπει να είναι αξιόπιστη.
- **Προσαρμογή.** Δεδομένου ότι χτίζετε τη δική σας πλατφόρμα, μπορεί να προσαρμοστεί 100%. Αν δεν μπορείτε να προσαρμόσετε το middleware, τότε πρέπει να το τροποποιήσετε το προϊόν ή η υπηρεσία σας να ταιριάζει στην πλατφόρμα, η οποία ουσιαστικά εργάζεται στην αντίστροφη κατεύθυνση.
- **Υποστηριζόμενα πρωτόκολλα και διεπαφές.** Από θεμελιώδη ορισμό, μια πλατφόρμα IoT middleware βρίσκεται μεταξύ δύο ετερογενή συστημάτων: φυσικές συσκευές και σύννεφο λογισμικού (και υπάρχουν πολλοί αριθμοί συσκευών τύποι και λογισμικό). Η πλατφόρμα θα πρέπει να είναι σε θέση για συντονισμό με όλα αυτά, ενορχήστρωση πραγμάτων και ταυτόχρονα να μιλάει με όλες τις γλώσσες προγραμματισμού ή πρωτόκολλα.
- **Εξοπλισμός.** Το Διαδίκτυο των Πραγμάτων είναι ουσιαστικά μια ομάδα ετερογενών συνδεδεμένων πραγμάτων, υλικού συσκευές, συστήματα υπολογιστών και λογισμικό.
- **Cloud.** Υπάρχουν αρκετοί πάροχοι υπηρεσιών cloud - συμπεριλαμβανομένης της Google, Microsoft, και Amazon Web Services (AWS) - αλλά η πλατφόρμα δεν πρέπει να εξαρτάται από το cloud. Είτε πρόκειται για δική σας υπηρεσία είτε για σύννεφο τρίτων τρέχει πίσω από ένα NAS (δικτυακός χώρος αποθήκευσης), η πλατφόρμα θα πρέπει να μπορεί να λειτουργεί.
- **Αρχιτεκτονική και τεχνολογία στοίβας.** Μια καλά καθορισμένη αρχιτεκτονική και ο κατάλληλος συνδυασμός της τεχνολογίας stack είναι ένα βασικό πράγμα που διαφοροποιεί ένα καλό Πλατφόρμα IoT από τους άλλους.
- **Ασφάλεια.** Τα τελευταία χρόνια, το Διαδίκτυο των Πραγμάτων έχουν γίνει γέλιο, κυρίως λόγω ελάχιστα διαχειριζόμενων πτυχών ασφάλειας σε πάρα πολλές εφαρμογές και λύσεις IoT. Η ασφάλεια γίνεται ζωτικής σημασίας παράγοντας εάν επιλέξετε μια πλατφόρμα πολλαπλών παραγόντων. Η πολυπαραγοντική πτυχή

καθιστά το σύστημα πιο ευάλωτο, γιατί η δική σας εφαρμογή μπορεί να είναι μια χαρά αλλά μια άλλη εφαρμογή που χρησιμοποιεί την ίδια πλατφόρμα (συν-μισθωτής της εφαρμογής σας) μπορεί να δημιουργήσει στην ασφάλεια προβλήματα για κάθε άλλο μισθωτή. Ο κίνδυνος είναι πάντα παρών.

- **Κόστος.** Ο προϋπολογισμός που έχει οριστεί για μια πλατφόρμα IoT έχει σχετικά μεγαλύτερη επιρροή στους παράγοντες κόστους. Ωστόσο, συνολικά, εάν το κόστος της πλατφόρμας (είτε κατασκευάστηκε εσωτερικά είτε εξαγοράστηκε από το ράφι) δεν δικαιολογεί τη λειτουργικότητα και των χαρακτηριστικών της, τότε πρέπει να επανεξεταστεί. Εν ολίγοις, η πλατφόρμα θα πρέπει να προσθέσει αρκετή αξία για να δικαιολογήσει το κόστος της.
- **Υποστήριξη.** Όσο και η συνεχής υποστήριξη για πλατφόρμα η διαχείριση είναι απαραίτητη, απαιτείται επίσης υποστήριξη για σκοπούς ολοκλήρωσης λύσεων. Και ως υποχρεωτικό απαίτηση, η πλατφόρμα middleware θα πρέπει να έχει ισχυρή υποστήριξη στο σχεδιασμό, την ανάπτυξη, την ανάπτυξη και διαχείριση της λύσης σε συνεχή βάση.

1.5.4.b Κατηγορίες IoT πλατφόρμων

Σήμερα, υπάρχουν περισσότερες από 500 πλατφόρμες διαφόρων σχημάτων και μεγεθών, και αυτός ο αριθμός θα συνεχίσει να αυξάνεται. Είναι ενδιαφέρον να σημειωθεί ότι πολλές από τις πλατφόρμες χάνουν τη γοητεία τους, οπότε κλείνουν ή συγχωνεύονται με άλλους. Ταυτόχρονα, μερικές πλατφόρμες μεταμορφώνονται σε φουτουριστικές και ισχυρές. Εν ολίγοις, αλλαγές συμβαίνουν και στις δύο κατευθύνσεις. Μια επισκόπηση αυτών των πλατφορμών δείχνει ότι μπορούμε να κατηγοριοποιήσουμε όλες αυτές σε τρεις βασικούς τύπους.

- **Πλατφόρμες που υποστηρίζουν την εξυπηρέτηση δικτύου.** Αυτές οι πλατφόρμες υποστηρίζουν τμήματα συντήρησης δικτύου, όπως στρώμα MAC αποκωδικοποιητές και μετατροπείς επικοινωνίας. Αυτές οι πλατφόρμες ουσιαστικά ελέγχει και συντονίζει το μέρος των πραγμάτων των τηλεπικοινωνιών. Ένα καλό παράδειγμα είναι ένας διακομιστής δικτύου για επικοινωνίες LoRaWAN. Αυτά τα πλατφόρμες μετατρέπουν την επικοινωνία ραδιοεπίπεδου σε πληροφορίες πρώτων δεδομένων και να το μεταβιβάσετε σε ανάντη πλατφόρμες ή εφαρμογές για περαιτέρω επεξεργασία. Εκτός από τις πλατφόρμες υπηρεσιών

δικτύου, υπάρχουν μερικά άλλα μέρη, όπως υπηρεσίες ταυτότητας και διαχείρισης κλειδιών και συνδυασμοί αυτών.

1.5.5. IoT πρωτόκολλα επικοινωνίας

1) CoAP

Αυτό το πρωτόκολλο σχεδιάστηκε από την ομάδα εργασίας του Constrained RESTful Environments (CoRE) για χρήση σε περιορισμένες συσκευές με περιορισμένες δυνατότητες επεξεργασίας. Παρόμοια με το HTTP, ένα από τα πιο καθοριστικά χαρακτηριστικά του είναι η χρήση δοκιμασμένης και καλά αποδεκτής αρχιτεκτονικής REST.

Η αντιπροσωπευτική κρατική μεταφορά (REST) είναι ένα αρχιτεκτονικό στυλ λογισμικού που ορίζει ένα σύνολο περιορισμών που θα χρησιμοποιηθούν για τη δημιουργία υπηρεσιών Web. Οι υπηρεσίες Web που συμμορφώνονται με το αρχιτεκτονικό στυλ REST, που ονομάζεται RESTful Web Services, παρέχουν διαλειτουργικότητα μεταξύ συστημάτων υπολογιστών στο Internet. Οι υπηρεσίες RESTful Web επιτρέπουν στα αιτούμενα συστήματα να έχουν πρόσβαση και να χειρίζονται αντικειμενικές αναπαραστάσεις των πόρων του Διαδικτύου χρησιμοποιώντας ένα ομοιόμορφο και προκαθορισμένο σύνολο λειτουργιών ανιθαγένειας. Άλλα είδη υπηρεσιών Web, όπως οι υπηρεσίες Web SOAP, εκθέτουν τα δικά τους αυθαίρετα σύνολα λειτουργιών.

Με αυτό το χαρακτηριστικό το CoAP υποστηρίζει το παράδειγμα παραγγελίας / απάντησης ακριβώς όπως το REST HTTP και ειδικά αυτό για περιορισμένα περιβάλλοντα. Το CoAP θεωρείται ελαφρύ πρωτόκολλο, οπότε οι κεφαλίδες, οι μέθοδοι και οι κωδικοί κατάστασης είναι όλα δυαδικά κωδικοποιημένα, μειώνοντας έτσι το γενικό πρωτόκολλο σε σχέση με τα πολλά πρωτόκολλα. Εκτελείται επίσης μέσω του λιγότερο σύνθετου πρωτοκόλλου μεταφοράς UDP αντί του TCP, περαιτέρω μειώνοντας τα γενικά έξοδα.

Όταν ένας υπολογιστής-πελάτης CoAP αποστέλλει ένα ή περισσότερα αιτήματα του CoAP στο διακομιστή και λαμβάνει την απάντηση, η απάντηση αυτή δεν αποστέλλεται μέσω μιας προηγούμενως εγκατεστημένης σύνδεσης, αλλά ανταλλάσσεται ασύγχρονα στα μηνύματα του CoAP.

2) MQTT (Message Queue Telemetry Transport)

Το MQTT είναι ένα από τα ελαφριά πρωτόκολλα ανταλλαγής μηνυμάτων που ακολουθεί το παράδειγμα δημοσίευσης-εγγραφής, γεγονός που το καθιστά μάλλον κατάλληλο για συσκευές περιορισμένης πρόσβασης και για μη ιδανικές συνθήκες σύνδεσης στο δίκτυο, όπως με χαμηλό εύρος ζώνης και υψηλή λανθάνουσα κατάσταση. Το MQTT κυκλοφόρησε από την IBM, με τη τελευταία έκδοση MQTT v3.1 υιοθετήθηκε για το IoT από το OASIS. Λόγω της απλότητας της, και με πολύ μικρή κεφαλίδα μηνύματος σε σύγκριση με άλλα πρωτόκολλα ανταλλαγής μηνυμάτων, συνιστάται συχνά ως τη λύση επικοινωνίας της επιλογής στο διαδίκτυο. Το MQTT τρέχει πάνω από το πρωτόκολλο μεταφοράς TCP, που εξασφαλίζει την αξιοπιστία του.

Σε σύγκριση με άλλα αξιόπιστα πρωτόκολλα, όπως το HTTP, και ευχαριστίες στην ελαφρύτερη κεφαλίδα του, το MQTT έρχεται με πολύ χαμηλότερες απαιτήσεις ισχύος, καθιστώντας το μια από τις πιο σημαντικές λύσεις πρωτοκόλλου σε περιορισμένα περιβάλλοντα. Υπάρχουν δύο μέρη επικοινωνίας στην αρχιτεκτονική MQTT που συνήθως παίρνουν τους ρόλους του εκδότες και συνδρομητές, πελάτες και διακομιστές / μεσίτες. Οι πελάτες είναι οι συσκευές που μπορούν να δημοσιεύσουν μηνύματα, να εγγραφείτε για να λαμβάνετε μηνύματα ή και τα δύο.

3) AMQP (Advanced Message Queuing Protocol)

Το AMQP είναι ένα ανοιχτό πρότυπο πρωτόκολλο που ακολουθεί το παράδειγμα δημοσίευσης-εγγραφής όπως ορίζεται από το OASIS, που έχουν σχεδιαστεί για να επιτρέπουν τη διαλειτουργικότητα μεταξύ ενός ευρέος φάσματος διαφορετικών εφαρμογών και συστημάτων, ανεξάρτητα από τα εσωτερικά σχέδια τους. Αρχικά αναπτύχθηκε για την ανταλλαγή επιχειρηματικών μηνυμάτων με την ιδέα της προσφοράς μιας μη-αποκλειστικής λύσης που μπορεί να διαχειριστεί ένα μεγάλο αριθμό ανταλλαγών μηνυμάτων θα μπορούσε να συμβεί σε σύντομο χρονικό διάστημα σε ένα σύστημα. Αυτό το χαρακτηριστικό διαλειτουργικότητας AMQP είναι σημαντικό δεδομένου ότι επιτρέπει διαφορετικές πλατφόρμες, που εφαρμόζονται σε διάφορες γλώσσες, για την ανταλλαγή μηνυμάτων, τα οποία ίσως ιδιαίτερα χρήσιμη σε ετερογενή συστήματα.

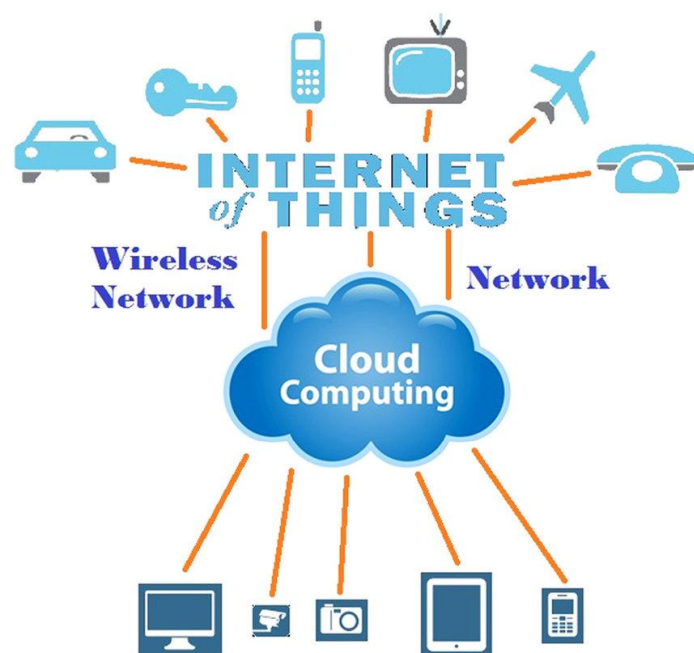
1.5.6. IoT και Cloud, μια τέλεια ένωση

Οι συσκευές IoT δημιουργούν μεγάλη ποσότητα δεδομένων, ασκώντας τεράστια επίδραση στην υποδομή του Διαδικτύου. Στο σημείο αυτό, έρχεται το Cloud Computing, το οποίο βοηθά στην αποθήκευση, επεξεργασία και μεταφορά δεδομένων στο σύννεφο (Cloud) αντί των συνδεδεμένων συσκευών.

Για παράδειγμα, πολλοί αισθητήρες εγκατεστημένοι σε διάφορες τοποθεσίες ενός εργοστασίου συλλέγουν δεδομένα συνεχώς από μηχανές και συσκευές, οι οποίες αργότερα αναλύονται σε πραγματικό χρόνο με τη βοήθεια εργαλείων ανάλυσης για την αναγνώριση βλαβών για την πρόληψη τυχόν μελλοντικών βλαβών. Το Cloud Computing βοηθά στην αποθήκευση όλων αυτών των δεδομένων από αυτές τις εκατοντάδες αισθητήρων και εφαρμόζει τους απαιτούμενους μηχανισμούς κανόνων και αλγορίθμους για την παροχή εκτιμώμενων αποτελεσμάτων αυτών των σημείων δεδομένων.

Όταν μια επιχείρηση χρησιμοποιεί χιλιάδες αισθητήρες για τη συλλογή δεδομένων, κάθε ένας από αυτούς τους αισθητήρες είναι φορτωμένος με μεγάλα ποσά υπολογιστικής ισχύος. Αυτό απαιτεί τεράστια ποσότητα ενέργειας και είναι δαπανηρή την ίδια στιγμή. Σε αυτήν την περίπτωση, τα δεδομένα μπορούν να μεταφερθούν στο νέφος από αυτούς τους αισθητήρες και να υποστούν επεξεργασία συνολικά.

Μπορούμε να πούμε ότι το σύννεφο είναι «ο εγκέφαλος» για μεγάλο μέρος του IoT, καθώς τα περισσότερα συλλεχθέντα δεδομένα τελικά επεξεργάζονται και αναλύονται στο σύννεφο.



Εικόνα 32. ΙοΤ με cloud computing

ΚΕΦΑΛΑΙΟ 2: Πειραματικό πλαίσιο του θέματος

Ενώ έχουμε αναλύσει στο θεωρητικό μέρος περί μικροελεγκτών, δικτύων, cloud computing και IoT μπορούμε τώρα αναπτύξουμε το πειραματικό κομμάτι της εργασίας αυτής.

Θα χρειαστούμε μια ανοικτού κώδικα πλατφόρμα μικροελεγκτή , βαλβίδες-διακόπτες καθώς επίσης και μια cloud εφαρμογή μέσω της οποίας θα μπορούμε να ελέγχουμε τη κατάσταση των διακοπών μέσω Internet.

Ας περιγράψουμε λίγο το hardware του project μας

1) ΠΛΑΤΦΟΡΜΑ ARDUINO



Εικόνα 33. Arduino boards

Γιατί Arduino; Η πλατφόρμα Arduino , πέρα του γεγονότος ότι είναι πολύ διαδεδομένη πλατφόρμα και χρησιμοποιείται ευρέως τόσο στην εκπαίδευση αλλά και σε επιχειρήσεις, έχει ορισμένα βασικά πλεονεκτήματα.

1) **Χαμηλό κόστος** : Μπορεί κάποιος να αγοράσει ένα Arduino πχ το Arduino Uno R3 μόνο με 7,60 € (μαζί τα μεταφορικά) από το e-bay.com. Συν τοις άλλοις, υπάρχουν starter kit τα οποία περιλαμβάνουν εκτός από τη πλακέτα αισθητήρες, καλώδια, LED, οθόνες LCD κτλ που κοστίζουν λιγότερο από 40 €.

2) **Συμβατότητα** : Το λογισμικό του Arduino εκτελείται εκτός από τα Windows και σε Linux και σε Macintosh.

3) **Απλότητα** : Το αναπτυξιακό περιβάλλον του Arduino μπορεί να χρησιμοποιηθεί και από αρχάριους χρήστες.

4) **Επεκτάσιμο λογισμικό ανοικτού κώδικα** : Ο καθένας μπορεί να βρει τμήματα πηγαίου κώδικα, να τα μελετήσει και να τα τροποποιήσει ανάλογα με τις ανάγκες του. Κάποιος έμπειρος χρήστης μπορεί να γράψει το δικό του κώδικα και να τον κάνει διαθέσιμο μέσω βιβλιοθηκών.

Η πλατφόρμα Arduino είναι, όπως αναφέρθηκε παραπάνω, μια ανοικτού κώδικα πλατφόρμα και συνδυάζει την ελευθερία στο σχέδιο, στην αρχιτεκτονική και στη συνεργασία. Το Arduino IDE είναι μια ανοικτού κώδικα εφαρμογή που εξηγείται παρακάτω.

1.1. Αναπτυξιακό Arduino board NodeMCU V3 με ESP8266

Το Arduino board που επιλέξαμε είναι το NodeMCU V3 , το οποίο θα αναλύσουμε πιο κάτω. Το NodeMCU V3 είναι ένα kit ανάπτυξης ανοικτού κώδικα ESP8266, εξοπλισμένο με το τσιπ Serial USB-TTL CH340G. Η σειρά τσιπ CH340 είναι γνωστή ως η προσιτή εναλλακτική λύση για το CP210x. Αν και φθηνότερο, το CH340 είναι εξαιρετικά αξιόπιστο ακόμη και σε βιομηχανικές εφαρμογές. Το CH340 είναι ένα τσιπ μετατροπής διαύλου USB και μπορεί να πραγματοποιήσει μετατροπή USB σε σειριακή διεπαφή, μετατροπή USB σε υπέρυθρες IrDA ή μετατροπή USB σε διεπαφή εκτυπωτή.

Σε λειτουργία σειριακής διασύνδεσης, το CH340 παρέχει κοινό σήμα σύνδεσης MODEM, το οποίο χρησιμοποιείται για τη μεγέθυνση της ασύγχρονης σειριακής διασύνδεσης του υπολογιστή ή την αναβάθμιση της κοινής σειριακής συσκευής σε USB bus απευθείας. Έχει δοκιμαστεί να είναι σταθερή και σε όλες τις υποστηριζόμενες πλατφόρμες.

Το ESP8266 NodeMCU είναι ένας δημοφιλής και ευρέως χρησιμοποιούμενος μικροελεγκτής που βασίζεται στο ESP-12E WiFi Module, συνδυάζει στοιχεία εύκολου προγραμματισμού με Arduino IDE (C \ C++) και

δυνατότητα WiFi. Μέσω του ενσωματωμένου προγραμματιστή και του CH340G USB-to-Serial chip, αναβοσβήνει το ESP8266 και η σειριακή έξοδος σε υπολογιστή, τα έργα ανάπτυξης και πρωτοτύπων γίνονται με ευκολία. Ακριβώς όπως οι πίνακες Arduino, το ESP8266 NodeMCU διαθέτει ακροδέκτες GPIO, ρυθμιστή τάσης, ADC, θύρα Micro-USB (για φλας και σειριακή έξοδο) - όλα σε μία πλακέτα. Επιπλέον, το ESP8266 NodeMCU διαθέτει ένα πλήρες WiFi που φροντίζει την επικοινωνία WiFi σε διακομιστή ή πελάτη.

Το ESP8266 είναι ένα σύστημα σε ένα τσιπ (SoC), που κατασκευάζεται από την κινεζική εταιρεία Espressif. Αποτελείται από μια μονάδα μικροελεγκτή Tensilica L106 32-bit (MCU) και έναν πομποδέκτη Wi-Fi. Διαθέτει 13 ακροδέκτες GPIO * (εισόδου / εξόδου γενικού σκοπού), καθώς και αναλογική είσοδο. Αυτό σημαίνει ότι μπορείτε να το προγραμματίσετε όπως οποιοδήποτε κανονικό Arduino ή άλλο μικροελεγκτή. Επιπλέον, λαμβάνετε επικοινωνία Wi-Fi, ώστε να μπορείτε να το χρησιμοποιήσετε για να συνδεθείτε στο δίκτυό σας Wi-Fi, να συνδεθείτε στο Διαδίκτυο, να φιλοξενήσετε έναν διακομιστή ιστού με πραγματικές ιστοσελίδες, να αφήσετε το smartphone σας να συνδεθεί σε αυτό, κ.λπ.... Οι πιθανότητες είναι ατελείωτες! Δεν είναι περίεργο που αυτό το τσιπ έχει γίνει η πιο δημοφιλής διαθέσιμη συσκευή IOT.

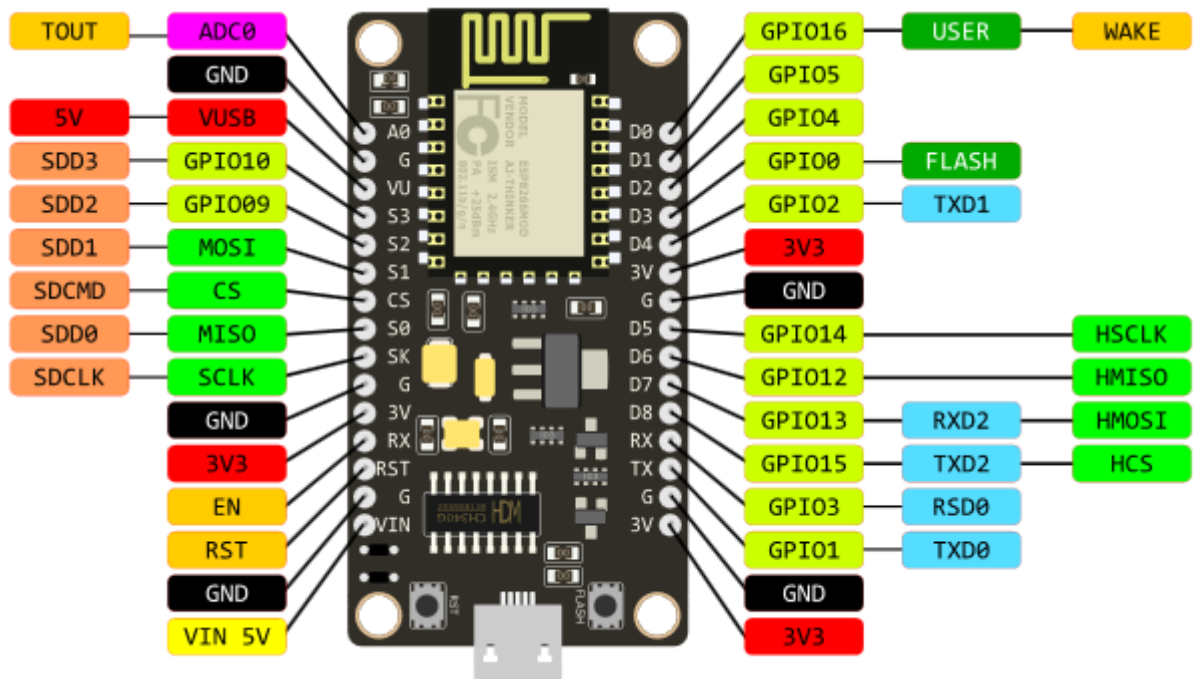
Περιέχει ενσωματωμένο CPU χαμηλής ισχύος 32-bit, ROM και RAM. Είναι μια ολοκληρωμένη και αυτόνομη λύση δικτύου Wi-Fi που μπορεί να μεταφέρει εφαρμογές λογισμικού ως αυτόνομη συσκευή ή συνδεδεμένη με μικροελεγκτή (MCU). Η μονάδα διαθέτει ενσωματωμένο υλικολογισμικό εντολών AT για χρήση με οποιοδήποτε MCU μέσω θύρας COM. Το ESP8266 μπορεί να αναβοσβήνει και να προγραμματιστεί χρησιμοποιώντας το Arduino IDE. Λόγω της μεγάλης κοινότητας προγραμματιστών ανοιχτού κώδικα, είναι διαθέσιμος μεγάλος αριθμός βιβλιοθηκών για αυτόν τον δημοφιλή μικροελεγκτή.

Το ESP8266EX ενσωματώνει έναν επεξεργαστή RISC 32-bit Tensilica L106, ο οποίος επιτυγχάνει εξαιρετικά χαμηλή κατανάλωση ισχύος και φτάνει σε μέγιστη ταχύτητα ρολογιού 160 MHz. Το λειτουργικό σύστημα σε πραγματικό χρόνο (RTOS) και η στοίβα Wi-Fi επιτρέπουν το 80% της ισχύος επεξεργασίας να είναι διαθέσιμο για προγραμματισμό και ανάπτυξη εφαρμογών χρήστη.

Με μια ενσωματωμένη διασύνδεση USB, μπορείτε εύκολα να τη χρησιμοποιήσετε με το Arduino IDE ή το NodeMCU Lua



Εικόνα 34. NodeMCU V3



Εικόνα 35. NodeMCU V3 pinout

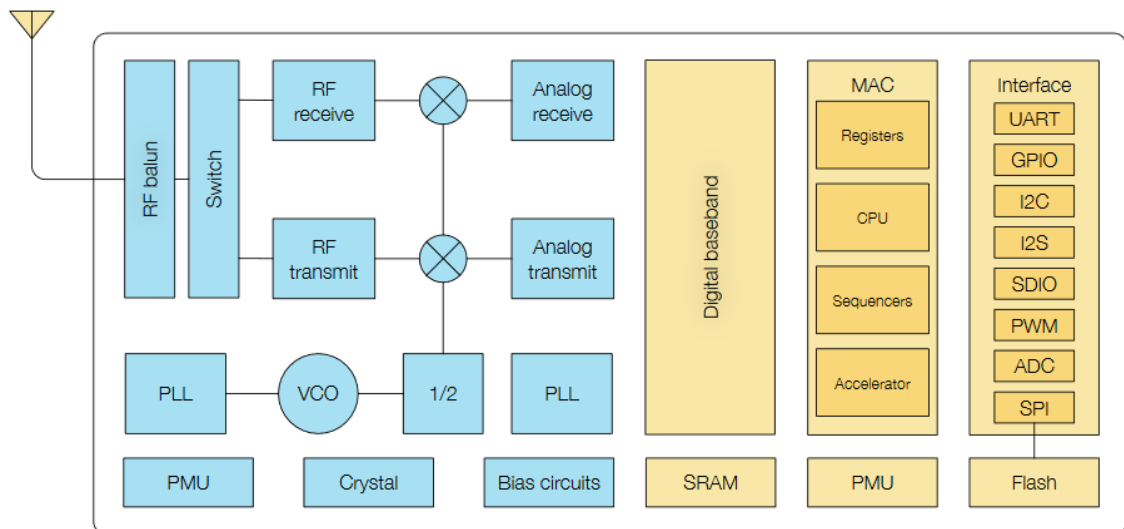
ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ NODEMCU V3

Όλες οι παραλλαγές ESP826 διαθέτουν επεξεργαστή πυρήνα ESP8266EX και μονάδα μικροελεγκτή Tensilica L106 32-bit. Πρόκειται για χαμηλό κόστος,

υψηλή απόδοση, χαμηλή κατανάλωση ενέργειας, εύκολο στον προγραμματισμό, ασύρματο SoC (System-On-Chip).

Παρέχει δυνατότητες για Wi-Fi 2,4 GHz (802,11 b / g / n, υποστήριξη WPA / WPA2), είσοδο / έξοδο γενικής χρήσης (16 GPIO), Inter-Integrated Circuit (I²C), μετατροπή αναλογικού σε ψηφιακό (10) -bit ADC), σειριακή περιφερειακή διασύνδεση (SPI), διασυνδέσεις I²S με DMA (καρφίτσες κοινής χρήσης με GPIO), UART (σε ειδικές ακίδες, καθώς και UART μόνο μετάδοσης μπορεί να ενεργοποιηθεί στο GPIO2) και διαμόρφωση πλάτους παλμού (PWM) .

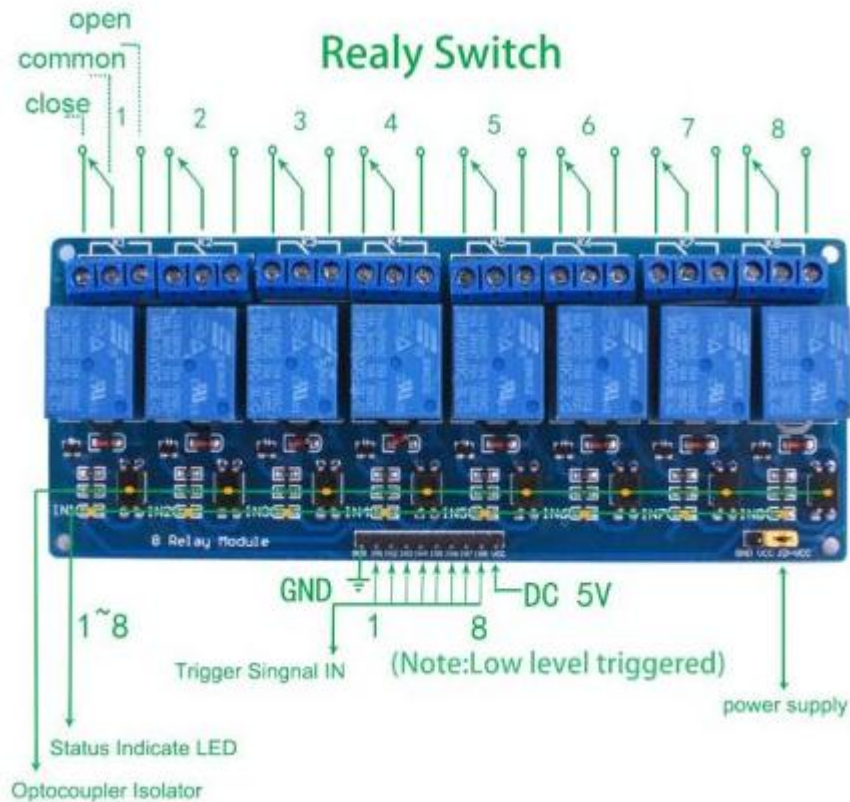
- Tensilica L106 32-bit μονάδα μικροεπεξεργαστή στα 80 MHz (ή overclocked στα 160 MHz).
- Μνήμη RAM 32 kB.
- RAM δεδομένων χρήστη 80 kB.
- RAM δεδομένων συστήματος 16 kB ETS.
- Μνήμη Flash 4Mb.
- USB - θύρα micro USB για τροφοδοσία, προγραμματισμό και εντοπισμό σφαλμάτων.
- 13 GPIO ακροδέκτες.
- WiFi 802.11 b / g / n συνδεσιμότητα, υποστηρίζοντας WPA / WPA2.
- Υποστήριξη λειτουργιών STA / AP.
- Στοίβα πρωτοκόλλου TCP / IP, μία υποδοχή.
- Διακομιστής και πελάτης TCP / UDP.
- Διαμόρφωση ρυθμού baud σειριακής θύρας: 1200/2400/4800/9600/19200/38400/57600/74800/115200 bps.
- Συμβατό με το Arduino UNO, Mega.
- Κουμπί KEY: διαμόρφωση τρόπων.
- Χρονόμετρο υλικού 32-bit.
- Ρεύμα λειτουργίας WiFi: λειτουργία συνεχούς μετάδοσης: ≈70mA (200mA MAX), λειτουργία βαθιάς ύπνου: <3mA.
- Ποσοστό σειριακής μετάδοσης WiFi: 110-460800bps.
- Θερμοκρασία: -40 °C ~ + 125 °C.
- Υγρασία: 10% -90% χωρίς συμπύκνωση.
- Βάρος: περίπου 20g (0,7 ουγκιές).
- Διαμόρφωση παλμού-πλάτους (PWM).
- Διακοπή δυνατότητας 3.3V τάση λειτουργίας, ο εσωτερικός ρυθμιστής τάσης επιτρέπει την είσοδο ισχύος 5V.
- Μέγιστο ρεύμα μέσω ακίδων GPIO: 12mA (πηγή), 20mA (αποστράγγιση)
- Διαθέσιμο υλικολογισμικό για το Arduino IDE.
- Διατίθενται βιβλιοθήκες Websocket.



Εικόνα 36. Λειτουργικό διάγραμμα ESP8266

2) MODULE 8 ΚΑΝΑΛΙΩΝ ΟΠΤΙΚΑ ΑΠΟΜΟΝΩΜΕΝΩΝ ΡΕΛΕ (8 Channel 5V Optical Isolated Relay Module)

Το module αυτό είναι το δεύτερο κομμάτι του project το οποίο θα συνδέσουμε στο NodeMCU V3 για να μπορούμε να ελέγχουμε τη κατάσταση των ρελέ προγραμματίζοντας το NodeMCU καταλλήλως. Πρόκειται για μια πλακέτα διασύνδεσης ρελέ 8 καναλιών LOW Level 5V και κάθε κανάλι χρειάζεται ρεύμα προγράμματος οδήγησης 15-20 mA. Μπορεί να χρησιμοποιηθεί για τον έλεγχο διαφόρων συσκευών και εξοπλισμού με μεγάλο ρεύμα. Είναι εξοπλισμένο με ρελέ υψηλής τάσης που λειτουργούν με AC 250V 10A ή DC 30V 10A και έχει μια τυπική διεπαφή που μπορεί να ελεγχθεί απευθείας από τον μικροελεγκτή. Αυτή η μονάδα είναι οπτικά απομονωμένη από την πλευρά υψηλής τάσης για απαίτηση ασφάλειας και επίσης αποτρέπει τον βρόχο γείωσης κατά τη διασύνδεση με τον μικροελεγκτή. Τα 8 αυτά ρελέ παίζουν το ρόλο των διακοπών (εναλλακτικά των βαλβίδων όπως είναι και ο αρχικός τίτλος της εργασίας).

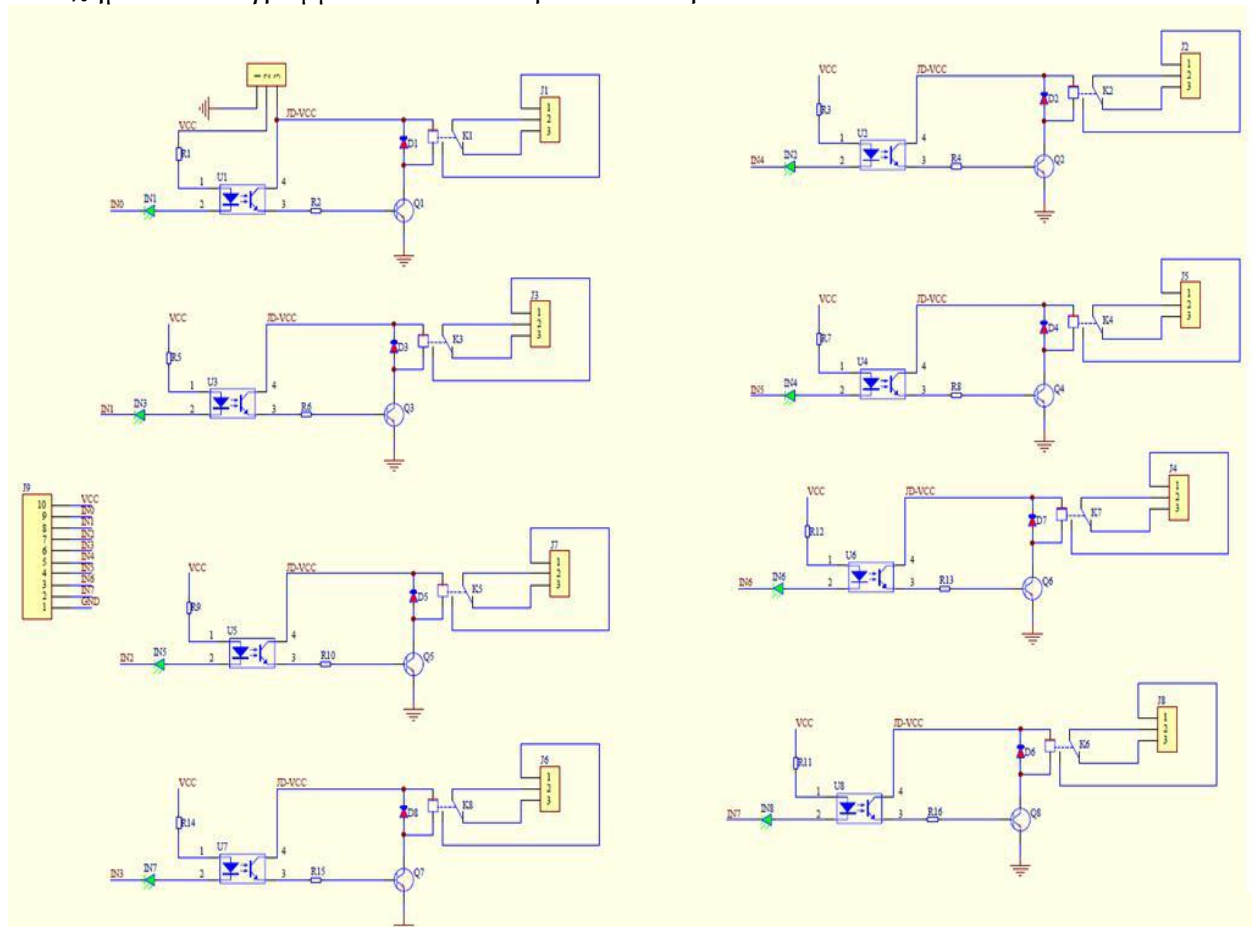


Εικόνα 37. 8-κάναλο κιτ οπτικά απομονωμένων ρελέ

ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

- Μέγιστη έξοδος ρελέ: DC 30V / 10A, AC 250V / 10A.
- Μονάδα ρελέ 8 καναλιών με οπτοζεύκτη. Πίνακας επέκτασης LOW Level Trigger, ο οποίος είναι συμβατός με τον πίνακα ελέγχου Arduino.
- Τυπική διεπαφή που μπορεί να ελεγχθεί απευθείας από μικροελεγκτή (8051, AVR, * PIC, DSP, ARM, ARM, MSP430, λογική TTL).
- Ρελέ υψηλής ποιότητας ρελέ χαμηλού θορύβου SPDT. Ένα κοινό τερματικό, ένα κανονικά ανοιχτό, ένα κανονικά κλειστό τερματικό.
- Απομόνωση Opto-Coupler, για ασφάλεια υψηλής τάσης και αποτροπή βρόχου γείωσης με μικροελεγκτή.
- Πλακέτα μονάδας: 138 x 56 mm

Το σχηματικό διάγραμμα του module φαίνεται παρακάτω



Εικόνα 38. Σχηματικό διάγραμμα του 8-channel 5V optical isolated relay

VCC: Συνδέεται με θετική τάση τροφοδοσίας (ισχύς τροφοδοσίας ανάλογα με την τάση ρελέ)

GND: Συνδέεται με γείωση.

IN1: Ακροδέκτης ενεργοποίησης σήματος 1 της μονάδας ρελέ

IN2: Ακροδέκτης ενεργοποίησης σήματος 2 της μονάδας ρελέ

IN3: Ακροδέκτης ενεργοποίησης σήματος 3 της μονάδας ρελέ

IN4: Ακροδέκτης ενεργοποίησης σήματος 4 της μονάδας ρελέ

IN5: Ακροδέκτης ενεργοποίησης σήματος 5 της μονάδας ρελέ

IN6: Ακροδέκτης ενεργοποίησης σήματος 6 της μονάδας ρελέ

IN7: Ακροδέκτης ενεργοποίησης σήματος 7 της μονάδας ρελέ

IN8: Ακροδέκτης ενεργοποίησης σήματος 8 της μονάδας ρελέ

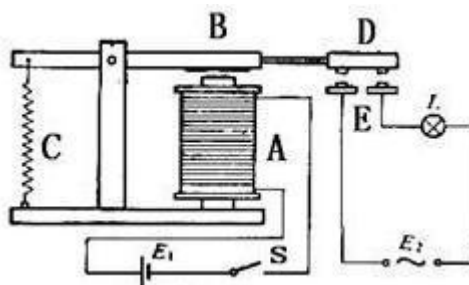
Όπως βλέπουμε έχουμε 8 ίδια blocks και το καθένα αποτελείται από

- ηρη τρανζίστορ MMBT5551 (G1)
- Μια αντίσταση 510 Ω
- Μια αντίσταση 1 ΚΩ
- Κόκκινο LED
- Οπτοcoupler DP817C909
- Ρελέ JQC-3FF S-Z
- Δίοδος D 1N4148
- 10 pin connector
- 3 pins connector

Τα VCC και RY-VCC είναι επίσης το τροφοδοτικό της μονάδας ρελέ. Όταν πρέπει να οδηγήσουμε ένα μεγάλο φορτίο ισχύος, μπορούμε να αφαιρέσουμε το κάλυμμα του βραχυκυκλωτήρα (jumper) και να συνδέσουμε μια επιπλέον ισχύ στο RY-VCC για να τροφοδοτήσουμε το ρελέ. Συνδέουμε το VCC στα 5V της πλακέτας MCU για να παρέχουμε σήματα εισόδου.

Εάν θέλουμε πλήρη οπτική απομόνωση, συνδέουμε το "Vcc" στο Arduino +5 volt αλλά δεν συνδέουμε το Arduino Ground. Αφαιρούμε τον βραχυκυκλωτήρα Vcc σε JD-Vcc. Συνδέουμε μια ξεχωριστή τροφοδοσία +5 στο "JD-Vcc" και το board στο Gnd. Αυτό θα παρέχει ισχύ στους οδηγούς τρανζίστορ και στα πηνία ρελέ. Εάν η απομόνωση ρελέ είναι αρκετή για την εφαρμογή μας, συνδέουμε το Arduino +5 και το Gnd και αφήνουμε το Vcc στο JD-Vcc jumper στη θέση του.

Το κάθε ρελέ παριστάνεται από το παρακάτω σχήμα



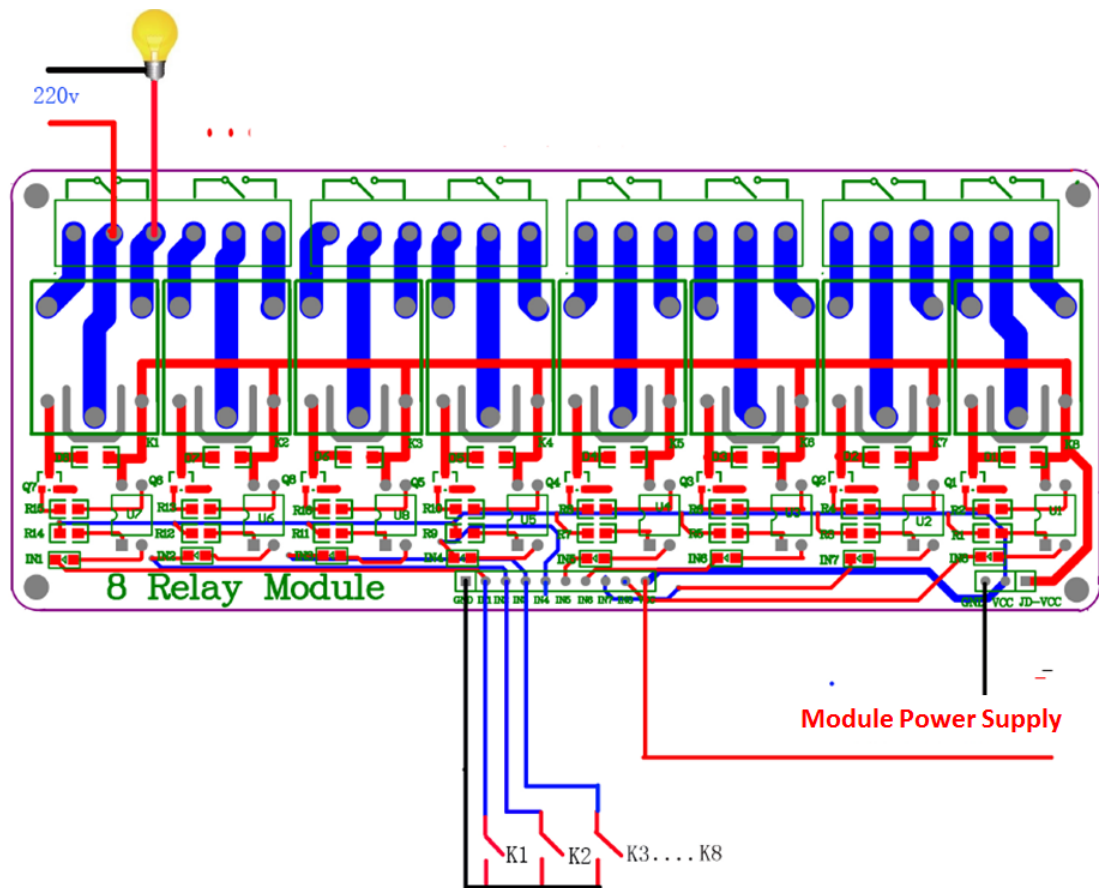
Εικόνα 39. Σχηματικό διάγραμμα του ρελέ του module

Η τάση τροφοδοσίας στο πηνίο και ορισμένα ρεύματα θα περάσουν μέσω του πηνίου δημιουργώντας έτσι το ηλεκτρομαγνητικό αποτέλεσμα. Έτσι, ο σπλισμός υπερνικά την τάση του ελατηρίου και προσελκύεται στον πυρήνα, κλείνοντας έτσι την κινούμενη επαφή του σπλισμού και την κανονικά ανοιχτή (NO) επαφή ή μπορείτε να πείτε ότι απελευθερώνετε την πρώτη και την κανονικά κλειστή (NC) επαφή.

Αφού το πηνίο απενεργοποιηθεί, η ηλεκτρομαγνητική δύναμη εξαφανίζεται και ο σπλισμός κινείται πίσω στην αρχική του θέση, απελευθερώνοντας την κινούμενη επαφή και την κανονικά κλειστή επαφή. Το κλείσιμο και η απελευθέρωση των επαφών έχει ως αποτέλεσμα την ενεργοποίηση και απενεργοποίηση του κυκλώματος.

Κάθε μονάδα του ρελέ έχει ένα NC (συνήθως κλειστό), ένα NO (κανονικά ανοιχτό) και ένα τερματικό COM (Common). Υπάρχουν συνολικά 8 NC, 8 NO και 8 COM του ρελέ καναλιών. Το NC σημαίνει την κανονική επαφή κλειστής θύρας και την κατάσταση χωρίς ρεύμα. NO σημαίνει την κανονική επαφή ανοιχτής θύρας και την κατάσταση με ισχύ. COM σημαίνει την κοινή θύρα. Μπορείτε να επιλέξετε θύρα NC ή θύρα OXI ανάλογα με την ισχύ ή όχι.

Όταν παρέχεται χαμηλή στάθμη στον ακροδέκτη σήματος του ρελέ 8 καναλιών, η λυχνία LED στον ακροδέκτη εξόδου θα ανάψει. Διαφορετικά, θα απενεργοποιηθεί. Εάν παρέχεται περιοδικό υψηλό και χαμηλό επίπεδο στον ακροδέκτη σήματος, μπορείτε να δείτε ότι η λυχνία LED θα περιστρέφεται μεταξύ ενεργοποίησης και απενεργοποίησης.



Εικόνα 40. Τυπωμένο κύκλωμα (layout) του 8-channel 5V optical isolated relay

Μερικές φορές είναι δυνατό να χρησιμοποιήσουμε αυτήν την πλακέτα ρελέ με σήματα 3.3V, εάν το JD-VCC (Ρελέ Ισχύς) παρέχεται από τροφοδοσία + 5V και ο βραχυκυκλωτήρας VCC σε JD-VCC αφαιρείται. Αυτή η παροχή ρελέ 5V θα μπορούσε να απομονωθεί εντελώς από τη συσκευή 3.3V ή να έχει κοινό έδαφος εάν δεν απαιτείται οπτική απομόνωση. Εάν χρησιμοποιούμε μεμονωμένα σήματα 3.3V, το VCC (Στην είσοδο του οπτο-απομονωτή, δίπλα στις ακίδες IN) πρέπει να συνδεθεί στην τροφοδοσία 3.3V της συσκευής 3.3V.

Ορισμένοι χρήστες του Raspberry-Pi διαπίστωσαν ότι ορισμένα ρελέ είναι αξιόπιστα και άλλοι δεν ενεργοποιούνται μερικές φορές. Ίσως χρειαστεί να αλλάξουμε την τιμή του R1 από 1000 Ω σε περίπου 220 Ω ή να τροφοδοτήσουμε + 5V στη σύνδεση VCC.

Οι ψηφιακές εισοδοί από το Arduino είναι Active LOW: Το ρελέ ενεργοποιείται και ανάβουν τα LED όταν το pin εισόδου είναι LOW και απενεργοποιείται σε HIGH.

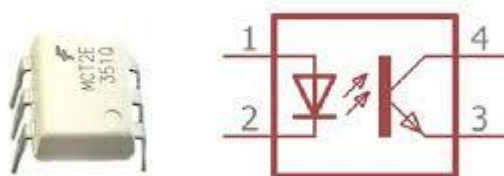
Στο module αυτό υπάρχουν και 8 οπτοζεύκτες (optocouplers). Πολλές φορές σε εφαρμογές που θέλουμε ηλεκτρική απομόνωση χρησιμοποιούμε

μετασχηματιστές, οι οποίοι απομονώνουν την πρωτογενή τάση εισόδου από τη δευτερεύουσα τάση εξόδου χρησιμοποιώντας ηλεκτρομαγνητική ζεύξη και αυτό επιτυγχάνεται με τη χρήση της μαγνητικής ροής που κυκλοφορεί εντός του στρωματοποιημένου πυρήνα σιδήρου.

Μπορούμε επίσης να παρέχουμε ηλεκτρική απομόνωση μεταξύ μιας πηγής εισόδου και ενός φορτίου εξόδου χρησιμοποιώντας μόνο φως χρησιμοποιώντας ένα πολύ κοινό και πολύτιμο ηλεκτρονικό στοιχείο που ονομάζεται Optocoupler.

Ο οπτικοαπομονωτής (ονομάζεται επίσης οπτικός συζευκτήρας, φωτοσυζεύκτης ή οπτικός απομονωτής) είναι ένα ηλεκτρονικό εξάρτημα που μεταφέρει ηλεκτρικά σήματα μεταξύ δύο απομονωμένων κυκλωμάτων χρησιμοποιώντας φως. Οι οπτικοαπομονωτές αποτρέπουν τις υψηλές τάσεις να επηρεάσουν το σύστημα που λαμβάνει το σήμα.

Ένας οπτοαπομονωτής περιέχει μια πηγή (εκπομπός) φωτός, σχεδόν πάντα μια σχεδόν υπέρυθη δίοδο εκπομπής φωτός (LED), η οποία μετατρέπει το σήμα ηλεκτρικής εισόδου σε φως, ένα κλειστό οπτικό κανάλι (ονομάζεται επίσης διηλεκτρικό κανάλι) και ένα φωτοαισθητήρας, ο οποίος ανιχνεύει το εισερχόμενο φως και είτε παράγει ηλεκτρική ενέργεια απευθείας, είτε ρυθμίζει το ηλεκτρικό ρεύμα που ρέει από μια εξωτερική τροφοδοσία. Ο αισθητήρας μπορεί να είναι φωτοαντίσταση, φωτοδίοδος, φωτοτρανζίστορ, ανορθωτής ελεγχόμενου πυριτίου (SCR) ή TRIAC. Δεδομένου ότι τα LED μπορούν να αισθάνονται φως εκτός από το που εκπέμπουν, είναι δυνατή η κατασκευή συμμετρικών, αμφίδρομων οπτικών απομονωτών.



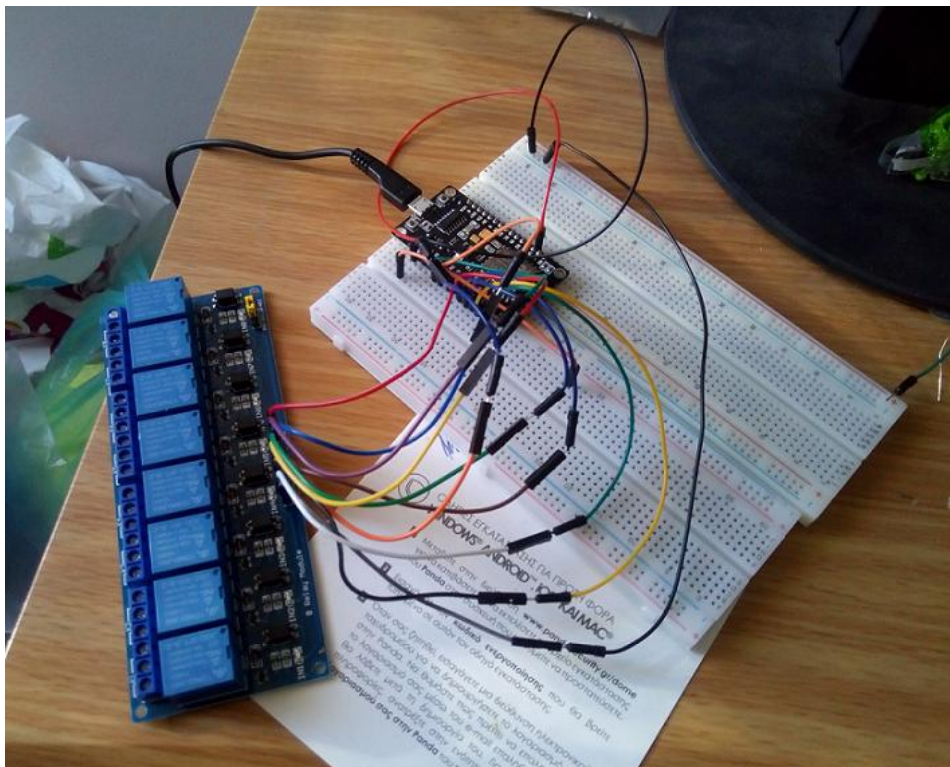
Εικόνα 41. Optocouler σε ic του module

Στο module που χρησιμοποιούμε, έχουμε το DP817 οπτοζεύκτη, ο οποίος είναι photo-transistor τύπου. Μια βασική παράμετρος λειτουργίας κάθε οπτοζεύκτη είναι ο λόγος ρεύματος μεταφοράς CTR (Current Transfer Ratio).

$$CTR = \frac{I_C}{I_F} \times 100\% \quad (1)$$

Όπου I_c το ρεύμα εκπομπού το φωτοτρανζίστορ του οπτοζεύκτη και I_f το ρεύμα του LED του οπτοζεύκτη. Με το CTR υπολογίζουμε τα ρεύματα αυτά βάσει των διαγραμμάτων που εκδίδουν οι διάφοροι κατασκευαστές ημιαγωγών.

Παρακάτω είναι το σχηματικό ενός από τα 8 (ίδια) blocks του 8-channel 5V optical isolated relay. Το ρελέ αποτελείται από το K1 (πηνίο ρελέ) μαζί με το S1.

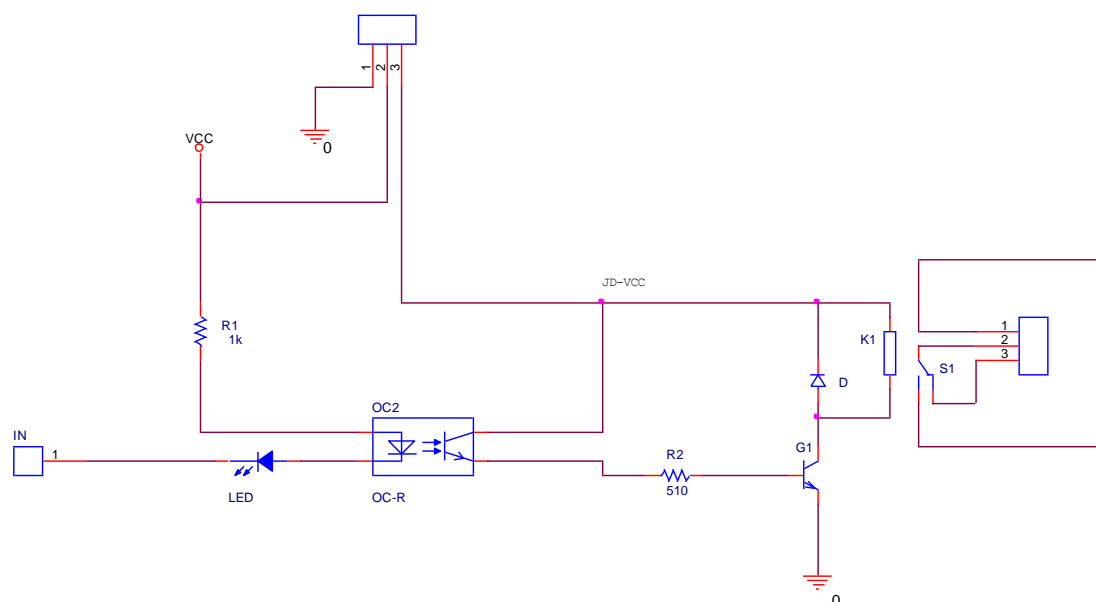


Εικόνα 42. Σύνδεση 8-channel module με το NodeMCU σε breadboard

Ανάλυση κυκλώματος ενός από τα 8 blocks του 8-channel 5V optical isolated relay

Έχοντας συνδέσει το 8-channel module με το NodeMCU (Εικόνα 43.) στο module παρέχεται τάση τροφοδοσίας $V_{cc} = +3,35$ Volts, διότι έχουμε βραχυκυκλωμένα με το jumper τα V_{cc} και $JD-V_{cc}$ στο module. Στο NodeMCU παρέχεται τάση +5 Volts από USB port.

Παρακάτω είναι το σχηματικό ενός από τα 8 (ίδια) blocks του 8-channel 5V optical isolated relay. Το ρελέ αποτελείται από το K1 (πηνίο ρελέ) μαζί με το S1.



Εικόνα 43. Σχηματικό διάγραμμα του ενός από τα 8 blocks του 8-channel 5V optical isolated relay

Δεδομένου ότι η είσοδος IN συνδέεται με την ψηφιακή έξοδο του NodeMCU, διακρίνουμε δυο περιπτώσεις για $IN = HIGH (+3,3 V)$ και $IN = LOW (0 V)$.

Για $I_N = +3,3$ Volts (HIGH)

Το LED είναι σε ανάστροφη πόλωση με αποτέλεσμα να μην ανάβει. Επομένως και ο οπτοζεύκτης DP817 και το NPN τρανζίστορ G1 είναι σε κατάσταση αποκοπής. Επομένως δε διαρρέει ρεύμα τις αντιστάσεις R_1 και R_2 και έχουμε μια τάση στα άκρα του φωτοτρανζίστορ του οπτοζεύκτη

Η τάση στο πηνίο K1 του ρελέ είναι και αυτή 0 Volts (άρα και στα άκρα της διόδου D), διότι στη περίπτωση αυτή ο διακόπτης του ρελέ κλείνει και βραχυκυκλώνει τη δίοδο.

Άρα έχουμε

Τάση στην αντίσταση R_1 , $V_{R1} = 0$ Volts

Τάση στην αντίσταση R_2 , $V_{R2} = 0$ Volts

Τάση στη δίοδο D, $V_D = 0$ Volts

Τάση συλλέκτη-εκπομπού για το G1, $V_{CE} = 3,35$ Volts

Τάση βάσης-εκπομπού για το G1, $V_{BE} = 0$ Volts

Τάση συλλέκτη - εκπομπού για το φωτοτρανζίστορ στο optocoupler, $V_{CEF} = 3,35$ Volts

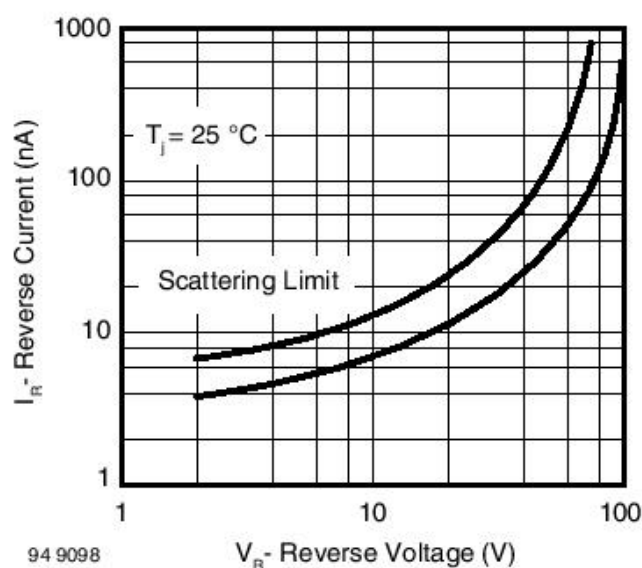


Fig. 3 - Reverse Current vs. Reverse Voltage

Εικόνα 44. Διάγραμμα διόδου 1N4148 για ανάστροφη πόλωση

Για $I_N = 0$ Volts (LOW)

Το LED ανάβει αφού βρίσκεται σε ορθή πόλωση (Εικόνα 43) και είναι σε σειρά με την αντίσταση R_1 και με το LED μέσα στον οπτοζεύκτη. Αυτό σημαίνει ότι περνά ρεύμα από το LED του οπτοζεύκτη, το οποίο είναι το I_F και θα υπολογιστεί με βάση τα διαγράμματα του οπτοζεύκτη. Στα διπολικά τρανζίστορ ισχύει ότι $I_E = I_B + I_C$, δηλ. το ρεύμα στον εκπομπό ισούται με το άθροισμα των ρευμάτων βάσης και συλλέκτη. Όμως για τα φωτοτρανζίστορ, ισχύει $I_E = I_C$, διότι δεν έχουμε ακροδέκτη - βάση.

Στην εικόνα 43, βλέπουμε ότι το LED, η φωτοδίοδος του οπτοζεύκτη και η αντίσταση $1\text{ K}\Omega$ είναι σε σειρά και τα $3,35$ Volts, V_{CC} , εφαρμόζονται πάνω τους. Δεδομένου ότι $I_N = 0$ Volts, έχουμε τα εξής

Τάση στο LED $V_{LED} = 1,82$ Volts (από ψηφιακό πολύμετρο)

Τάση στην αντίσταση R_1 , $V_{R1} = 0,45$ Volts

Τάση στην αντίσταση R_2 , $V_{R2} = 0,878$ Volts

Τάση στη δίοδο D, $V_D = 3,05$ Volts (ανάστροφη)

Τάση συλλέκτη-εκπομπού για το G1, $V_{CE} = 0,2$ Volts (από ψηφιακό πολύμετρο)

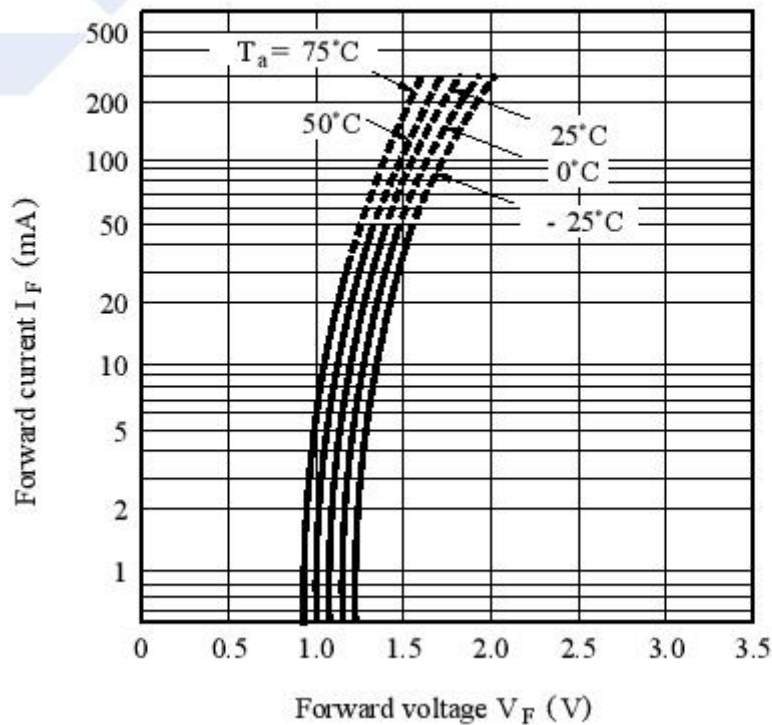
Τάση βάσης-εκπομπού για το G1, $V_{BE} = 0,775$ Volts (από ψηφιακό πολύμετρο)

Τάση συλλέκτη - εκπομπού για το φωτοτρανζίστορ στο optocoupler, $V_{CEF} = 1,69$ Volts

Τάση στον οπτοζεύκτη $V_F = 1,08$ Volts

Αφού λοιπόν $R_1 = 1\text{ K}\Omega$ (Εικόνα 43) και $V_{R1} = 0,45$ Volts, έχουμε $I_F = I_{R1} = 450\ \mu\text{A}$.

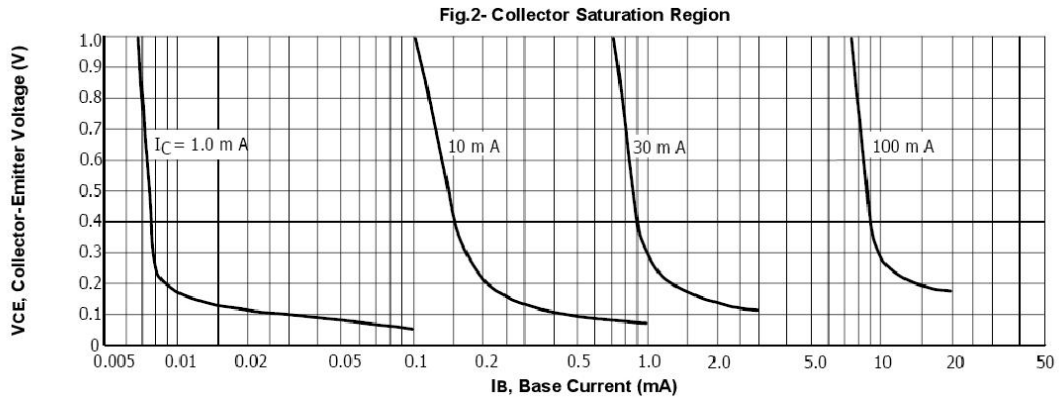
Fig. 5 Forward Current vs. Forward Voltage



Εικόνα 45. Διάγραμμα οπτοζεύκτη DP817

Η τάση στη φωτοδίοδο του οπτοζεύκτη $V_F = 1,08$ Volts από το παραπάνω διάγραμμα για $I_F = 0,45$ mA. Δεδομένου ότι η τάση στην αντίσταση R_2 ($R_2 = 510 \Omega$ από την Εικόνα 43) είναι στο $0,878$ Volt , το ρεύμα της είναι $1,72$ mA, το οποίο είναι και το ρεύμα βάσης για το τρανζίστορ G1 αλλά ισούται και με το ρεύμα I_C του φωτοτρανζίστορ του οπτοζεύκτη. Επομένως, $CTR = I_C / I_F = 1,72 / 0,45 = 3,822$ ή $382,2 \%$.

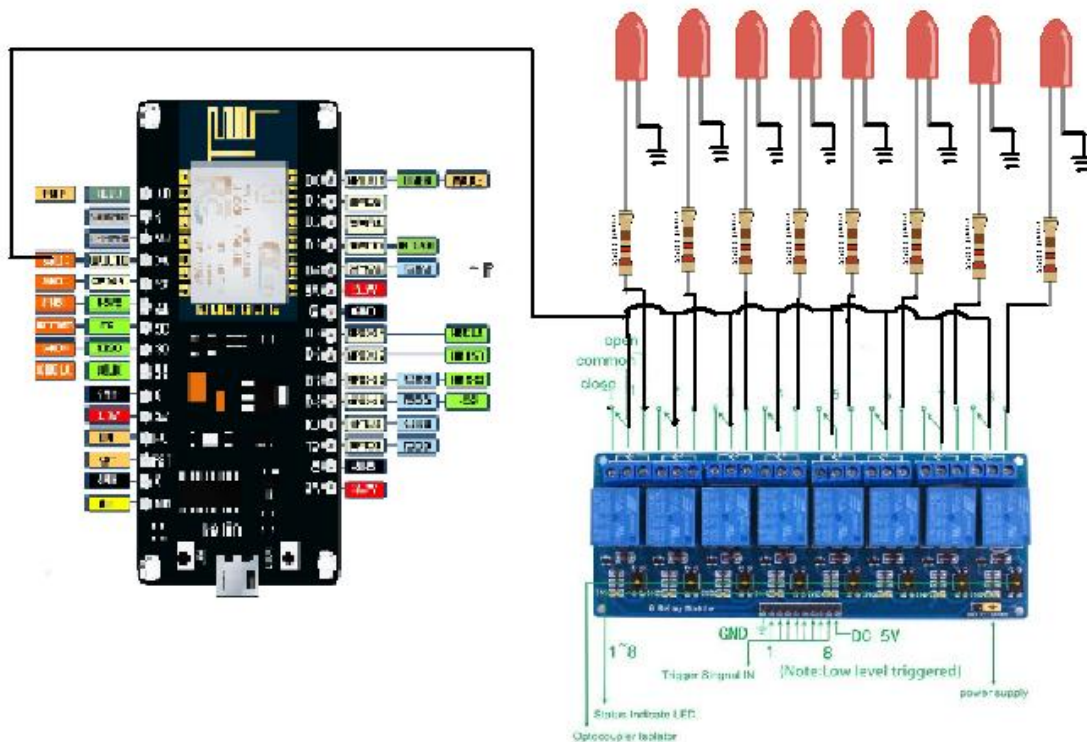
Για το G1 τρανζίστορ οι επαφές βάση-εκπομπός και βάση-συλλέκτης είναι ορθά πολωμένες, όντως $V_{BE} = 0,775$ Volts και $V_{BC} = 0,575$ Volts. Αυτό σημαίνει ότι είναι στο κορεσμό το G1 και ισχύει το παρακάτω διάγραμμα



Εικόνα 46. Διάγραμμα κορεσμού για το τρανζίστορ MMBT5550 (G1) ($V_{CE} - I_B$)

Από το παραπάνω διάγραμμα για $I_B = 1,72 \text{ mA}$ όπως έχουμε και στη περίπτωση μας και $V_{CE} = 0,2 \text{ Volts}$, έχουμε $I_C \approx 30 \text{ mA}$. Επομένως η ενίσχυση h_{FE} του τρανζίστορ G1 βγαίνει $I_C / I_B = 17,44$. Άρα το ρεύμα εκπομπού του G1 I_E που προκύπτει από το άθροισμα $I_B + I_C$ βγαίνει $31,72 \text{ mA}$.

Παρακάτω απεικονίζεται το σχήμα του 8-channel module με το NodeMCU και με 8 κόκκινα LED συνδεδεμένα στα ρελε του module. Όπως φαίνεται και στην εικόνα, στον ακροδέκτη COMMON του κάθε ρελέ βάζουμε τη τάση τροφοδοσίας $V_{CC} = +3,35 \text{ Volts}$ και στον ακροδέκτη OPEN συνδέουμε το κάθε LED αντίστοιχα σε σειρά με μια 220Ω αντίσταση.

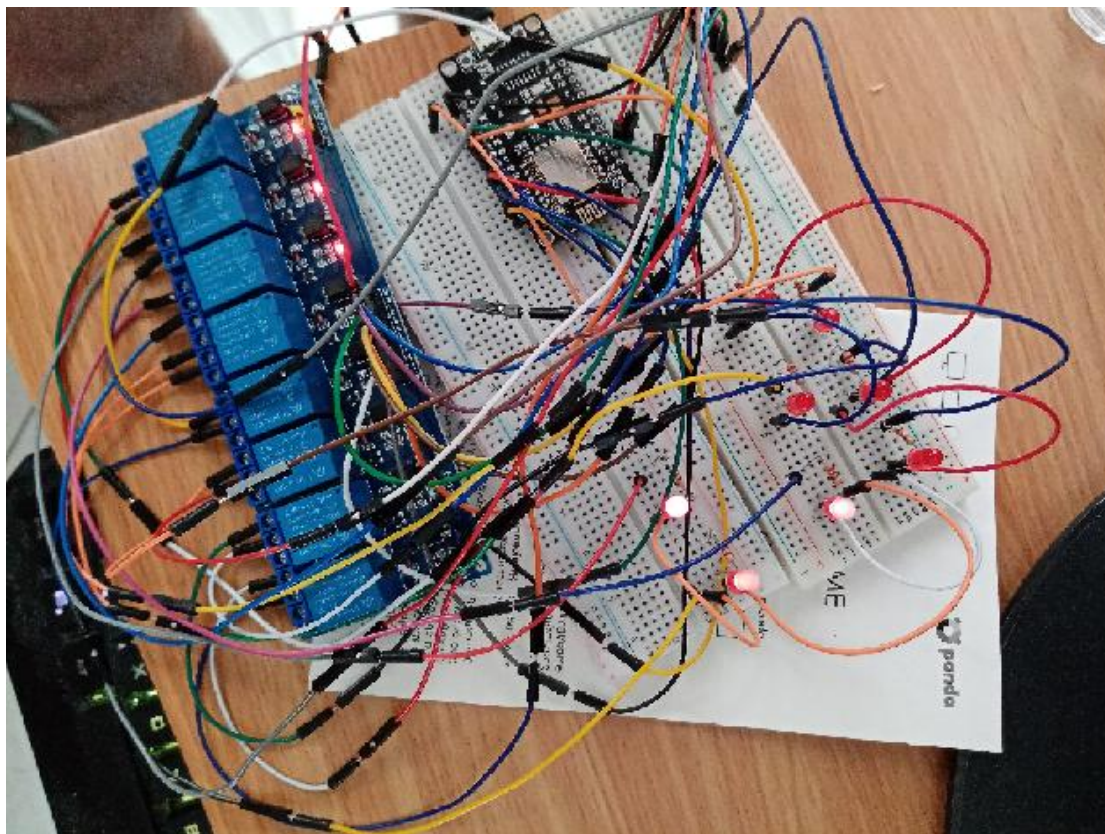


Εικόνα 47. NodeMCU, 8-channel module με 8 LED

Όταν η έξοδος από τα pin του NodeMCU είναι 0 (IN = LOW) , το ρελέ κλείνει και η τάση Vcc εφαρμόζεται στην αντίσταση και στο LED κάνοντας το LED να ανάβει.

Τάση στο LED, $V_{LED} = 2,05$ Volts

Τάση στην αντίσταση 220Ω , $V_{220} = 1,3$ Volts, άρα το ρεύμα που ρέει στην αντίσταση και το LED είναι 5,9 mA.

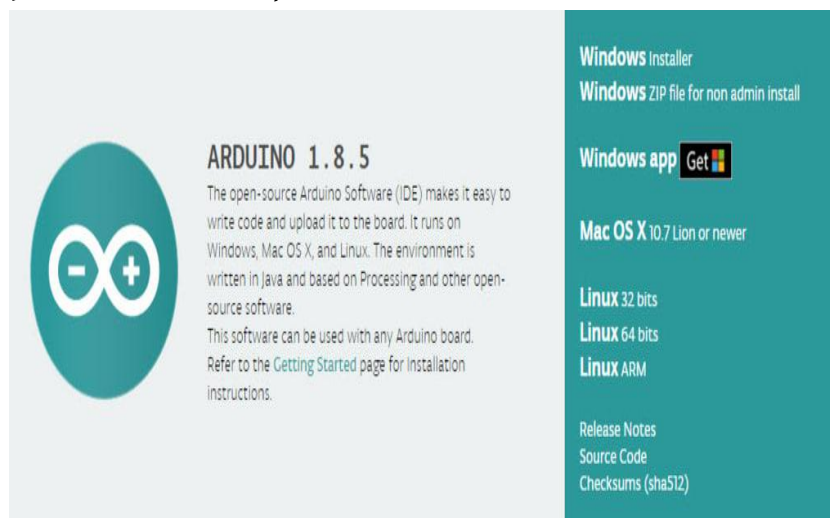


Εικόνα 48. Το σύστημα με ενεργά τα τρία τελευταία ρελέ με αναμμένα τα τρία τελευταία LED

3) ARDUINO IDE

Ας δούμε πώς προγραμματίζεται ένα Arduino board γενικά. Για τον προγραμματισμό των Arduino έχει αναπτυχθεί το ολοκληρωμένο περιβάλλον **Arduino IDE** (Integrated Development Environment, <https://www.arduino.cc/en/Main/Software>), δηλ. μια πλατφόρμα βασισμένη στο περιβάλλον ανάπτυξης processing development environment (<http://processing.org>) με το οποίο γράφουμε, επεξεργαζόμαστε, αποσφαλματώνουμε και τέλος φορτώνουμε τον arduino κώδικα της εφαρμογής μας στο board που έχουμε επιλέξει. Η ομάδα προγραμματιστών του Arduino έχει

κάνει τη παραδοχή ότι οι χρήστες δεν ενδιαφέρονται για τις χιλιάδες ειδικότητες που έχουν σχέση με την αρχιτεκτονική μικροελεγκτών. Γι' αυτόν ακριβώς το λόγο, προσάρμοσαν το ρεπερτόριο εντολών του μικροελεγκτή AVR σε μια απλή βιβλιοθήκη εντολών Arduino για κάθε board.



Εικόνα 49. Το περιβάλλον προγραμματισμού Arduino IDE

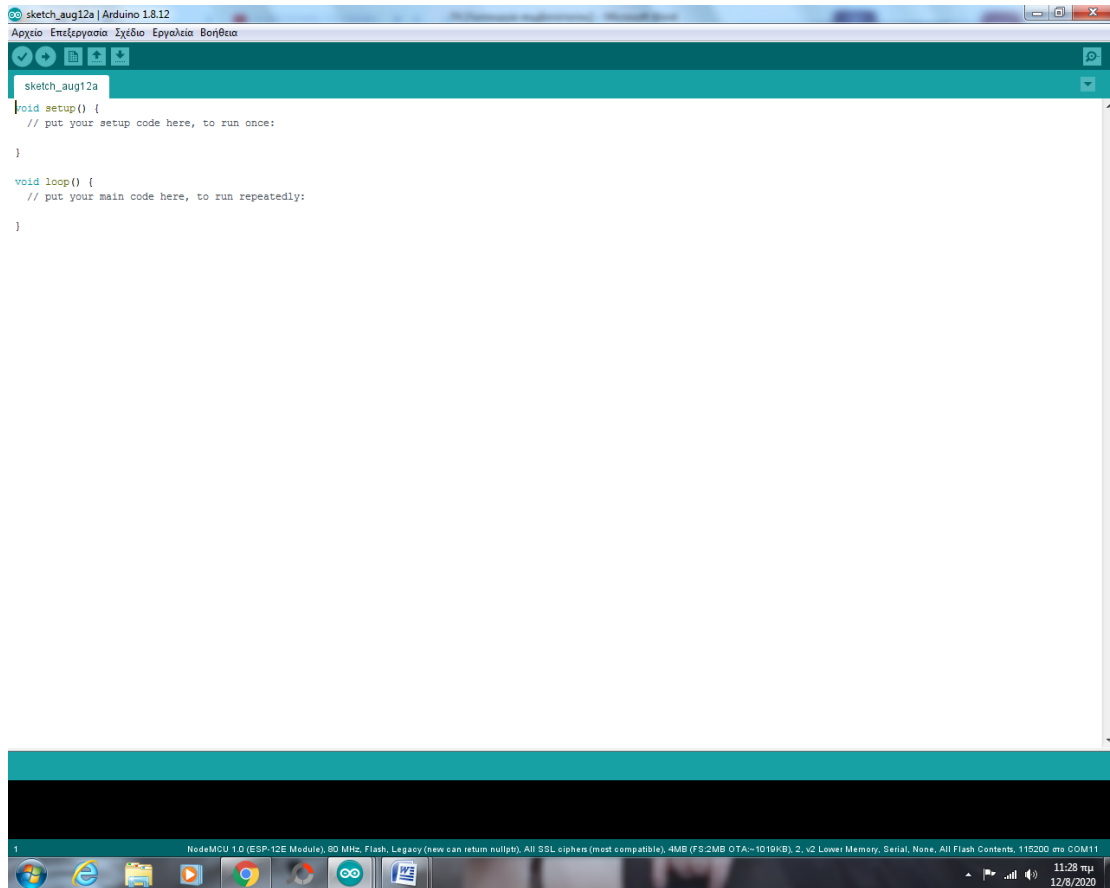
Το Arduino IDE είναι γραμμένο σε γλώσσα Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch).

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring", από το πρωτότυπο σχέδιο Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Κάθε εφαρμογή απαιτεί κύκλωμα και πρόγραμμα (κώδικα). Ο κώδικας εκτελείται από τον μικροελεγκτή και το αποτέλεσμα εμφανίζεται στο κύκλωμα. Όπως αναφέραμε παραπάνω, ο κώδικας για κάθε Arduino εφαρμογή είναι γραμμένος σε γλώσσα Wiring C, η οποία βασίζεται στους κλασικούς κανόνες δομημένου προγραμματισμού (structural programming). Οι χρήστες Arduino πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

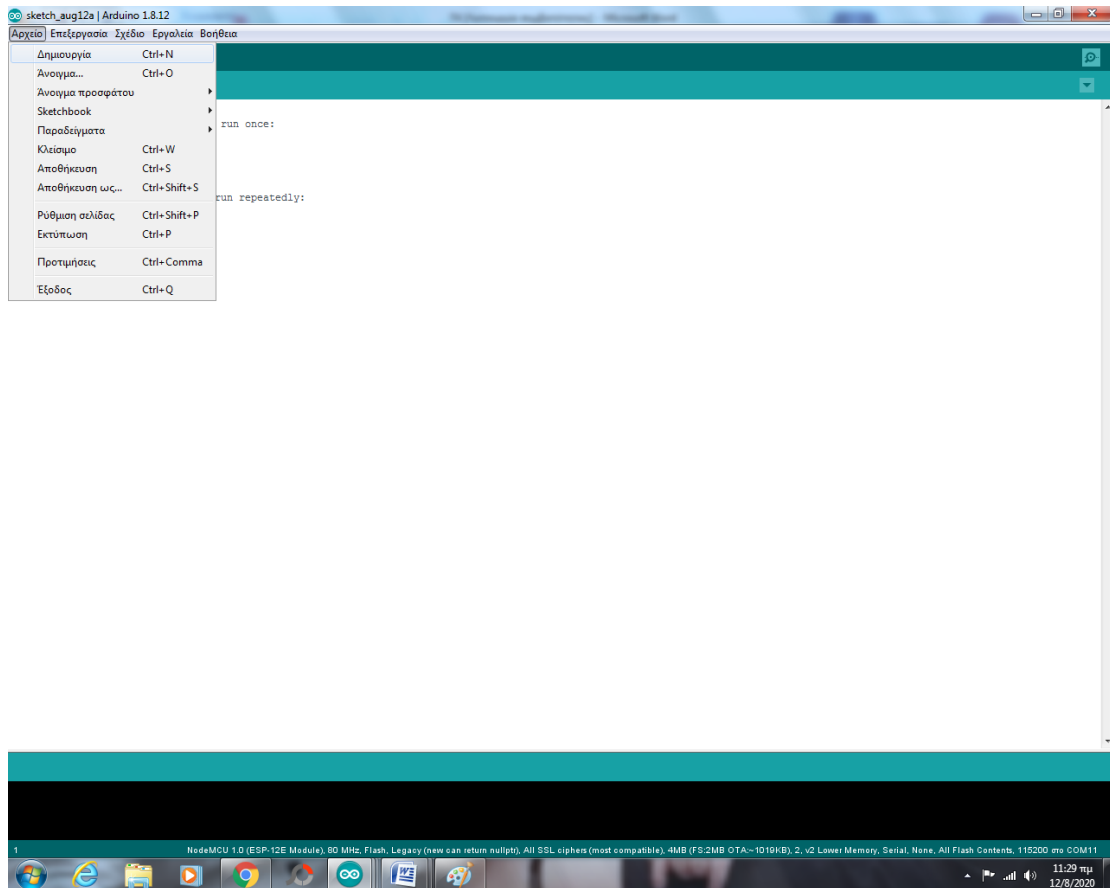
-setup() : μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις

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

Η αρχική σελίδα του φαίνεται πιο κάτω

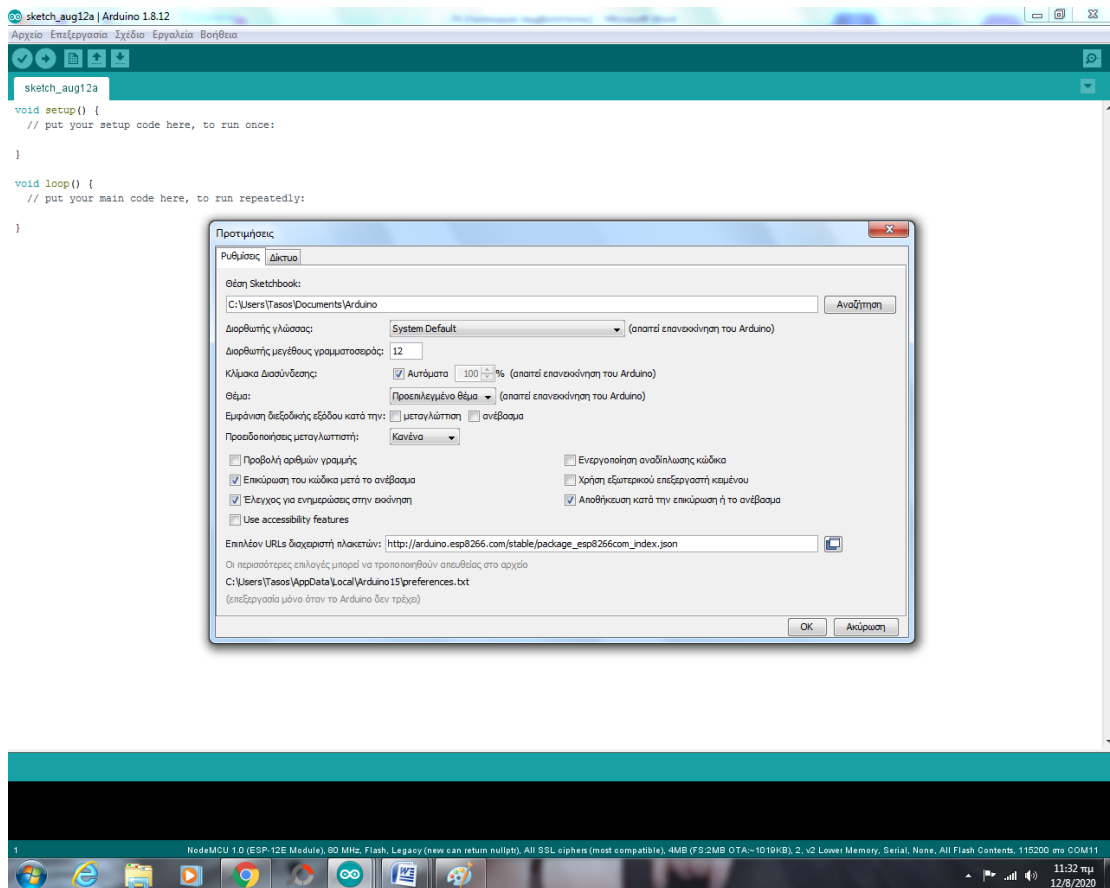


Εικόνα 50. Αρχικό παράθυρο του Arduino IDE



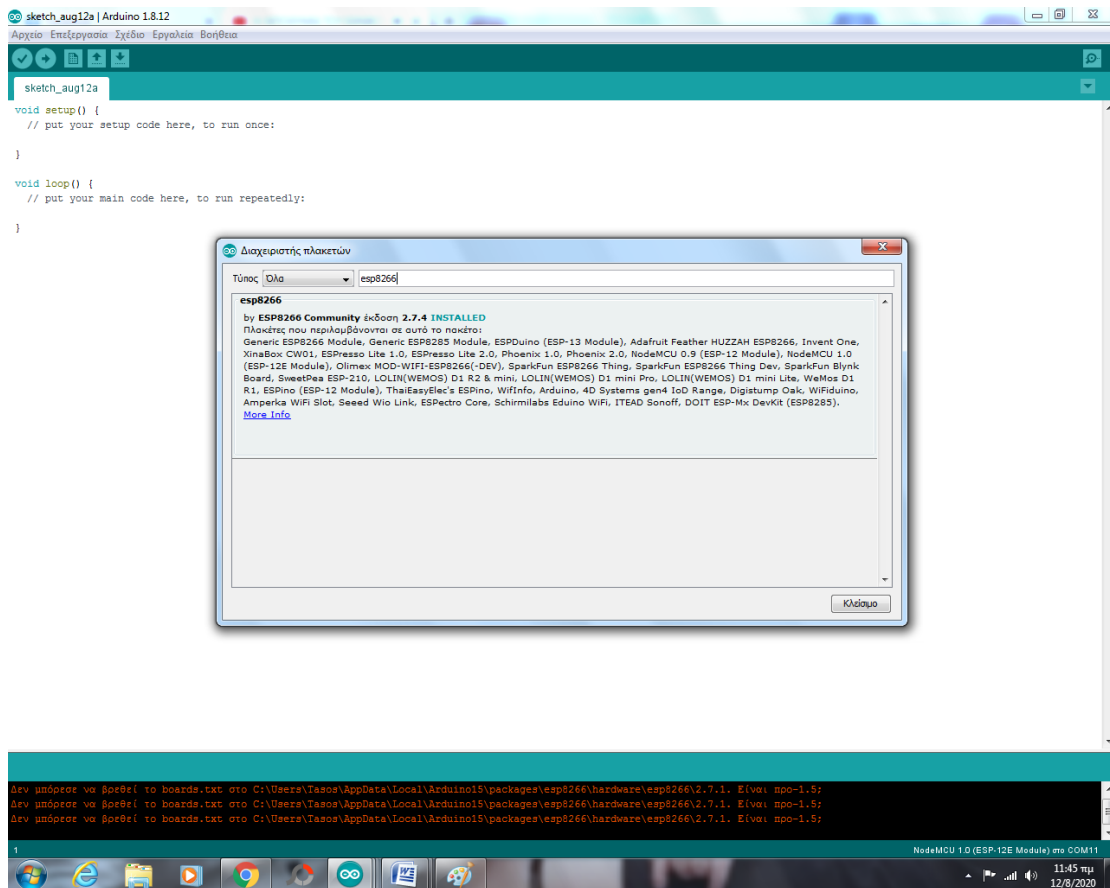
Εικόνα 51. Αρχικό παράθυρο του Arduino IDE

Πατώντας Αρχείο-->Δημιουργία, ανοίγει το παράθυρο και εκεί αρχίζουμε και συντάσσουμε το εκάστοτε πρόγραμμα της εφαρμογής μας. Επειδή στο board μας υπάρχει το ESP8266 , χρειάζεται να κάνουμε μια ρύθμιση στο Arduino IDE, (Αρχείο-->Προτιμήσεις) όπως φαίνεται παρακάτω



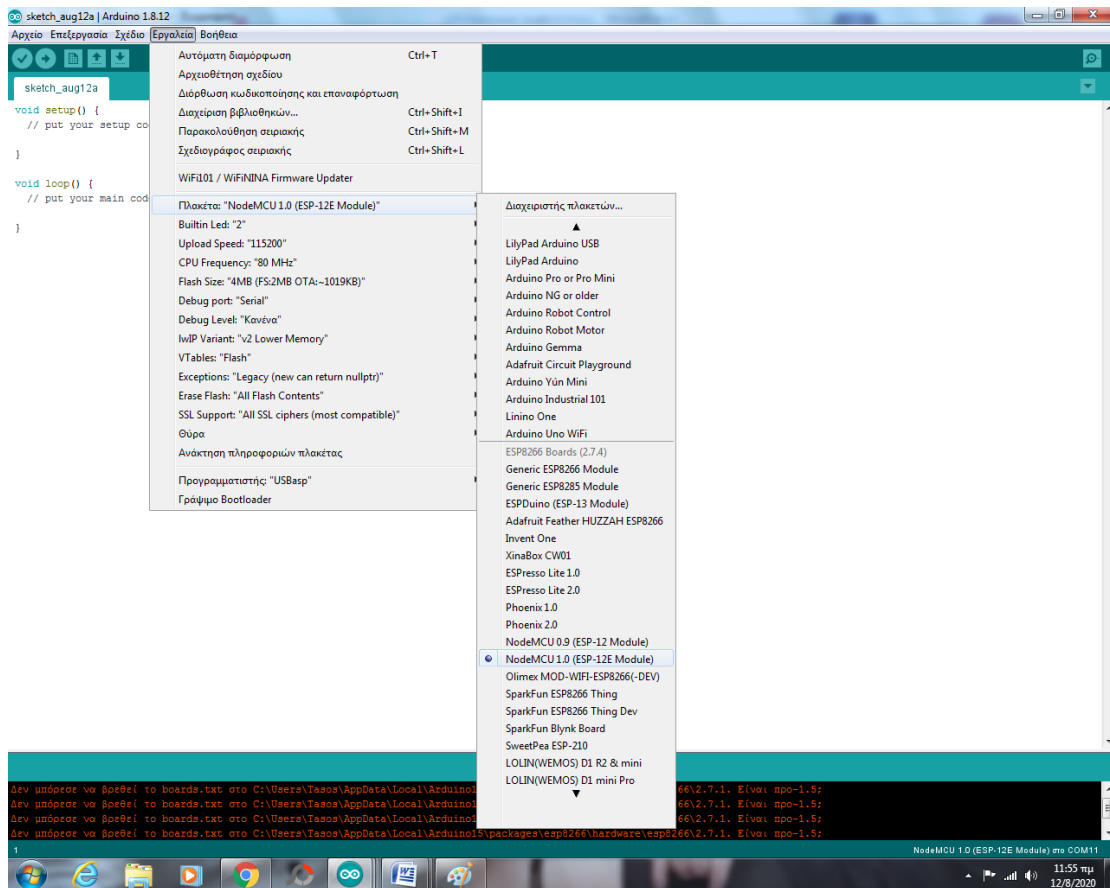
Εικόνα 52. Arduino IDE Προτιμήσεις

Στο μενού "Επιπλέον URLs διαχειριστή πλακετών" στο πεδίο δεξιά βάζουμε http://arduino.esp8266.com/stable/package_esp8266com_index.json. Επίσης κλικάρουμε Εργαλεία--> Διαχείριση πλακετών και στο πεδίο βάζουμε esp8266 όπως φαίνεται πιο κάτω



Εικόνα 53. Arduino IDE Διαχειριστής πλακετών

Τέλος, δε ξεχνάμε να κλικάρουμε Εργαλεία--> Πλακέτα και στο μενού δίπλα επιλέγουμε το board που έχουμε επιλέξει για το project μας.



Εικόνα 54. Arduino IDE Διαχειριστής πλακετών επιλογή board

4) ΠΡΟΓΡΑΜΜΑΤΑ - ΕΦΑΡΜΟΓΕΣ ΤΟΥ NODEMCU ME TO 8-CHANNEL 5V MODULE

Αρχικά, θα δημιουργήσουμε ένα πρόγραμμα όπου θα αναβοσβήνουν διαδοχικά 8 LEDs ένα-ένα τα οποία είναι συνδεδεμένα σε κάθε GPIO του Node MCU μας (από D0 έως D7, Εικόνα 35) με διάρκεια 1 δευτερόλεπτο (δηλ. για 1 δευτερόλεπτο κάθε LED θα είναι ON και για 1 δευτερόλεπτο θα είναι OFF) και αυτό φαίνεται παρακάτω

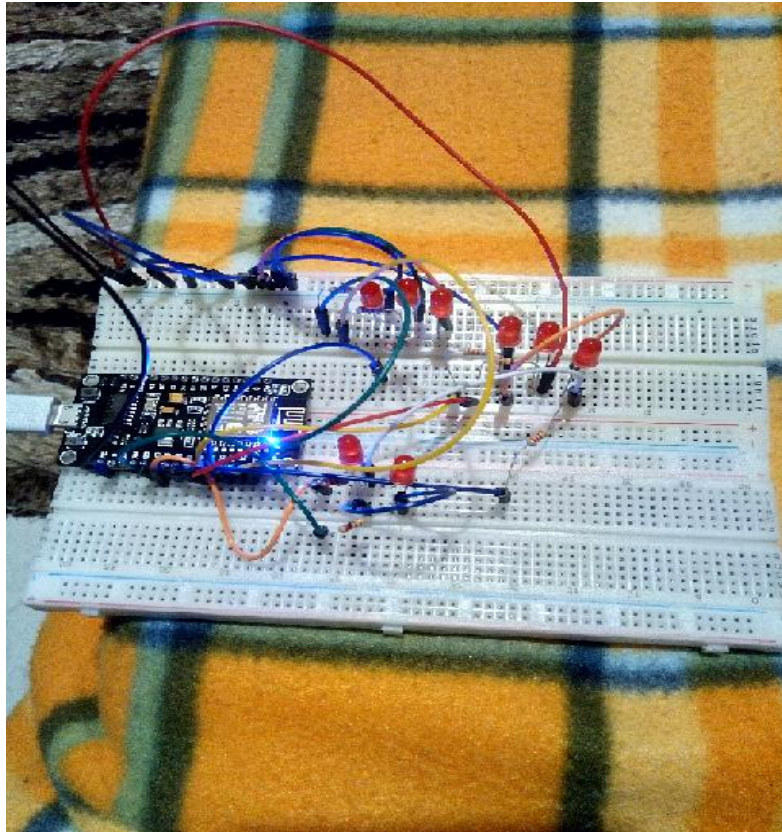
```
#define SIZE 8
```

```
int exodoi[SIZE]={D0,D1,D2,D3,D4,D5,D6,D7}; //ορισμός πίνακα 8 pins του board
```

```
void setup()
{
  for(int i=0; i<SIZE; i++) //να γίνουν έξοδοι και τα 8 pins D0-D7
  {
    pinMode(exodoi[i],OUTPUT);
  }
}

void loop()
{

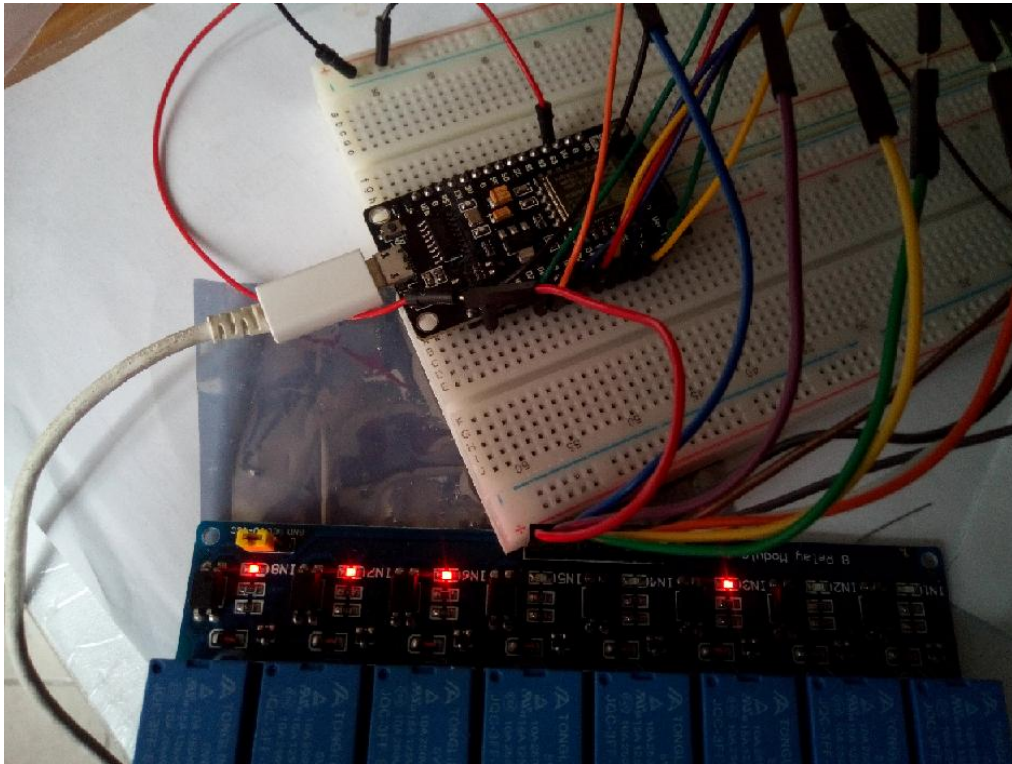
  for(int i=0; i<SIZE; i++) //να αναβοσβήνουν τα LEDES διαδοχικά κάθε 1 sec
  {
    digitalWrite(exodoi[i],LOW);
    delay(1000);
    digitalWrite(exodoi[i],HIGH);
    delay(1000);
  }
}
```



Εικόνα 55. NodeMCU με 8 LEDs συνδεδεμένα στα GPIOs

Το επόμενο πρόγραμμα είναι για να ελέγξουμε μέσω Internet (χρησιμοποιώντας WiFi) τα ρελέ του module αφού όμως έχουμε συνδέσει τα GPIOs (D0-D7) του NodeMCU με τα IN1-IN8 του module αντίστοιχα και έχουμε επίσης συνδέσει το NodeMCU με το WiFi μας . Στο ίδιο WiFi (SSID και password) συντονίζουμε το PC ή smartphone ή tablet.

Παρακάτω απεικονίζεται φωτογραφία με το 8-channel 5V relay module σε σύνδεση με το NodeMCU καθώς επίσης και το πρόγραμμα



Εικόνα 56. NodeMCU συνδεδεμένο με το 8-channel module

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

const char* ssid = "WiFi SSID"; // Το όνομα του WiFi δικτύου μας
const char* password = "WiFi access password"; //Κωδικός WiFi

ESP8266WebServer server(80);

uint8_t LED0pin = D0;
bool rel0status = LOW;

uint8_t LED1pin = D1;
bool rel1status = LOW;

uint8_t LED2pin = D2;
```

```
bool rel2status = LOW;
```

```
uint8_t LED3pin = D3;  
bool rel3status = LOW;
```

```
uint8_t LED4pin = D4;  
bool rel4status = LOW;
```

```
uint8_t LED5pin = D5;  
bool rel5status = LOW;
```

```
uint8_t LED6pin = D6;  
bool rel6status = LOW;
```

```
uint8_t LED7pin = D7;  
bool rel7status = LOW;
```

```
void setup()
```

```
{
```

```
  Serial.begin(115200);
```

```
  delay(100);
```

```
  pinMode(LED0pin, OUTPUT);
```

```
  pinMode(LED1pin, OUTPUT);
```

```
  pinMode(LED2pin, OUTPUT);
```

```
  pinMode(LED3pin, OUTPUT);
```

```
  pinMode(LED4pin, OUTPUT);
```

```
  pinMode(LED5pin, OUTPUT);
```

```
  pinMode(LED6pin, OUTPUT);
```

```
  pinMode(LED7pin, OUTPUT);
```

```
  Serial.println("Connecting to ");
```



```

Serial.println(ssid);

//connect to your local wi-fi network
WiFi.begin(ssid, password);

//check wi-fi is connected to wi-fi network
while (WiFi.status() != WL_CONNECTED)
{
  delay(1000);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected..!");
Serial.print("Got IP: ");
Serial.println(WiFi.localIP());

server.on("/", handle_OnConnect);
server.on("/rel0on", handle_rel0on);
server.on("/rel0off", handle_rel0off);
server.on("/rel1on", handle_rel1on);
server.on("/rel1off", handle_rel1off);
server.on("/rel2on", handle_rel2on);
server.on("/rel2off", handle_rel2off);
server.on("/rel3on", handle_rel3on);
server.on("/rel3off", handle_rel3off);
server.on("/rel4on", handle_rel4on);
server.on("/rel4off", handle_rel4off);
server.on("/rel5on", handle_rel5on);
server.on("/rel5off", handle_rel5off);
server.on("/rel6on", handle_rel6on);
server.on("/rel6off", handle_rel6off);
server.on("/rel7on", handle_rel7on);
server.on("/rel7off", handle_rel7off);

server.onNotFound(handle_NotFound);

```

```
server.begin();
Serial.println("HTTP server started");
}
void loop()
{
server.handleClient();

if(rel0status)
{
digitalWrite(LED0pin, HIGH);
}
else
{
digitalWrite(LED0pin, LOW);
}

if(rel1status)
{
digitalWrite(LED1pin, HIGH);
}
else
{
digitalWrite(LED1pin, LOW);
}
if(rel2status)
{
digitalWrite(LED2pin, HIGH);
}
else
{
digitalWrite(LED2pin, LOW);
}
if(rel3status)
{
```

```
    digitalWrite(LED3pin, HIGH);
  }
else
{
  digitalWrite(LED3pin, LOW);
  }
  if(rel4status)
  {
    digitalWrite(LED4pin, HIGH);
  }
else
{
  digitalWrite(LED4pin, LOW);
  }
  if(rel5status)
  {
    digitalWrite(LED5pin, HIGH);
  }
else
{
  digitalWrite(LED5pin, LOW);
  }
  if(rel6status)
  {
    digitalWrite(LED6pin, HIGH);
  }
else
{
  digitalWrite(LED6pin, LOW);
  }
  if(rel7status)
  {
    digitalWrite(LED7pin, HIGH);
  }
else
```

```

    {
        digitalWrite(LED7pin, LOW);
    }

}

void handle_OnConnect()
{
    rel0status = LOW;
    rel1status = LOW;
    rel2status = LOW;
    rel3status = LOW;
    rel4status = LOW;
    rel5status = LOW;
    rel6status = LOW;
    rel7status = LOW;
    Serial.println("GPIO0 Status: OFF | GPIO1 Status: OFF | GPIO2 Status: OFF |
GPIO3 Status: OFF | GPIO4 Status: OFF | GPIO5 Status: OFF | GPIO6 Status: OFF
| GPIO7 Status: OFF" );
    server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,rel3status,rel4status,rel5status,rel6statu
s,rel7status));

}

void handle_rel0on()
{
    rel0status = HIGH;
    Serial.println("GPIO0 Status: ON");
    server.send(200, "text/html",
SendHTML(true,rel1status,rel2status,rel3status,rel4status,rel5status,rel6status,rel7s
tatus));
}

void handle_rel0off()
{

```

```

    rel0status = LOW;
    Serial.println("GPIO0 Status: OFF");
    server.send(200, "text/html",
SendHTML(false,rel1status,rel2status,rel3status,rel4status,rel5status,rel6status,rel7
status));
}
void handle_rel1on()
{
    rel1status = HIGH;
    Serial.println("GPIO1 Status: ON");
    server.send(200, "text/html",
SendHTML(rel0status,true,rel2status,rel3status,rel4status,rel5status,rel6status,rel7s
tatus));
}

void handle_rel1off()
{
    rel1status = LOW;
    Serial.println("GPIO1 Status: OFF");
    server.send(200, "text/html",
SendHTML(rel0status,false,rel2status,rel3status,rel4status,rel5status,rel6status,rel7
status));
}

void handle_rel2on()
{
    rel2status = HIGH;
    Serial.println("GPIO2 Status: ON");
    server.send(200, "text/html",
SendHTML(rel0status,rel1status,true,rel3status,rel4status,rel5status,rel6status,rel7s
tatus));
}

void handle_rel2off()
{
    rel2status = LOW;

```

```

Serial.println("GPIO2 Status: OFF");
server.send(200, "text/html",
SendHTML(rel0status,rel1status,false,rel3status,rel4status,rel5status,rel6status,rel7
status));
}

void handle_rel3on()
{
rel3status = HIGH;
Serial.println("GPIO3 Status: ON");
server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,true,rel4status,rel5status,rel6status,rel7s
tatus));
}

void handle_rel3off()
{
rel3status = LOW;
Serial.println("GPIO3 Status: OFF");
server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,false,rel4status,rel5status,rel6status,rel7
status));
}

void handle_rel4on()
{
rel4status = HIGH;
Serial.println("GPIO4 Status: ON");
server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,rel3status,true,rel5status,rel6status,rel7s
tatus));
}

void handle_rel4off()
{
rel4status = LOW;
Serial.println("GPIO4 Status: OFF");

```

```

    server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,rel3status,false,rel5status,rel6status,rel7
status));
}

void handle_rel5on()
{
    rel5status = HIGH;
    Serial.println("GPIO5 Status: ON");
    server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,rel3status,rel4status,true,rel6status,rel7s
tatus));
}

void handle_rel5off()
{
    rel5status = LOW;
    Serial.println("GPIO5 Status: OFF");
    server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,rel3status,rel4status,false,rel6status,rel7
status));
}

void handle_rel6on()
{
    rel6status = HIGH;
    Serial.println("GPIO6 Status: ON");
    server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,rel3status,rel4status,rel5status,true,rel7s
tatus));
}

void handle_rel6off()
{
    rel6status = LOW;
    Serial.println("GPIO6 Status: OFF");

```

```

    server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,rel3status,rel4status,rel5status,false,rel7
status));
}
void handle_rel7on()
{
    rel7status = HIGH;
    Serial.println("GPIO7 Status: ON");
    server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,rel3status,rel4status,rel5status,rel6statu
s,true));
}

void handle_rel7off()
{
    rel7status = LOW;
    Serial.println("GPIO7 Status: OFF");
    server.send(200, "text/html",
SendHTML(rel0status,rel1status,rel2status,rel3status,rel4status,rel5status,rel6statu
s,false));
}

void handle_NotFound()
{
    server.send(404, "text/plain", "Not found");
}
// ΣΥΝΑΡΤΗΣΗ SendHTML για τη δημιουργία ιστοσελίδας ελέγχου των
String SendHTML(uint8_t rel1stat,uint8_t rel2stat,uint8_t rel3stat,uint8_t
rel4stat,uint8_t rel5stat,uint8_t rel6stat,uint8_t rel7stat,uint8_t rel8stat)
{
    String ptr = "<!DOCTYPE html> <html>\n";

    ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1.0, user-scalable=no\">\n";
    ptr += "<title>LED Control</title>\n";
    ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px
auto; text-align: center;}\n";

```



```

ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;} h3
{color: #444444;margin-bottom: 50px;}\n";
ptr += ".button {display: block;width: 80px;background-color: #1abc9c;border:
none;color: white;padding: 13px 30px;text-decoration: none;font-size:
25px;margin: 0px auto 35px;cursor: pointer;border-radius: 4px;}\n";
ptr += ".button-on {background-color: #1abc9c;}\n";
ptr += ".button-on:active {background-color: #16a085;}\n";
ptr += ".button-off {background-color: #34495e;}\n";
ptr += ".button-off:active {background-color: #2c3e50;}\n";
ptr += "p {font-size: 14px;color: #888;margin-bottom: 10px;}\n";
ptr += "</style>\n";
ptr += "</head>\n";
ptr += "<body>\n";
ptr += "<h1>ESP8266 Web Server</h1>\n";
ptr += "<h3>Using Station(STA) Mode</h3>\n";

```

```

if(rel1stat)
{ptr += "<p>REL0 Status: OFF</p><a class=\"button button-off\"
href=\"/rel0off\">OFF</a>\n";}
else
{ptr += "<p>REL0 Status: ON</p><a class=\"button button-on\"
href=\"/rel0on\">ON</a>\n";}

```

```

if(rel2stat)
{ptr += "<p>REL1 Status: OFF</p><a class=\"button button-off\"
href=\"/rel1off\">OFF</a>\n";}
else
{ptr += "<p>REL1 Status: ON</p><a class=\"button button-on\"
href=\"/rel1on\">ON</a>\n";}

```

```

if(rel3stat)
{ptr += "<p>REL2 Status: OFF</p><a class=\"button button-off\"
href=\"/rel2off\">OFF</a>\n";}
else
{ptr += "<p>REL2 Status: ON</p><a class=\"button button-on\"
href=\"/rel2on\">ON</a>\n";}

```

```

if(rel4stat)
{ptr += "<p>REL3 Status: OFF</p><a class=\"button button-off\"
href=\"/rel3off\">OFF</a>\n";}
else
{ptr += "<p>REL3 Status: ON</p><a class=\"button button-on\"
href=\"/rel3on\">ON</a>\n";}

if(rel5stat)
{ptr += "<p>REL4 Status: OFF</p><a class=\"button button-off\"
href=\"/rel4off\">OFF</a>\n";}
else
{ptr += "<p>REL4 Status: ON</p><a class=\"button button-on\"
href=\"/rel4on\">ON</a>\n";}

if(rel6stat)
{ptr += "<p>REL5 Status: OFF</p><a class=\"button button-off\"
href=\"/rel5off\">OFF</a>\n";}
else
{ptr += "<p>REL5 Status: ON</p><a class=\"button button-on\"
href=\"/rel5on\">ON</a>\n";}

if(rel7stat)
{ptr += "<p>REL6 Status: OFF</p><a class=\"button button-off\"
href=\"/rel6off\">OFF</a>\n";}
else
{ptr += "<p>REL6 Status: ON</p><a class=\"button button-on\"
href=\"/rel6on\">ON</a>\n";}

if(rel8stat)
{ptr += "<p>REL7 Status: OFF</p><a class=\"button button-off\"
href=\"/rel7off\">OFF</a>\n";}
else
{ptr += "<p>REL7 Status: ON</p><a class=\"button button-on\"
href=\"/rel7on\">ON</a>\n";
}
}

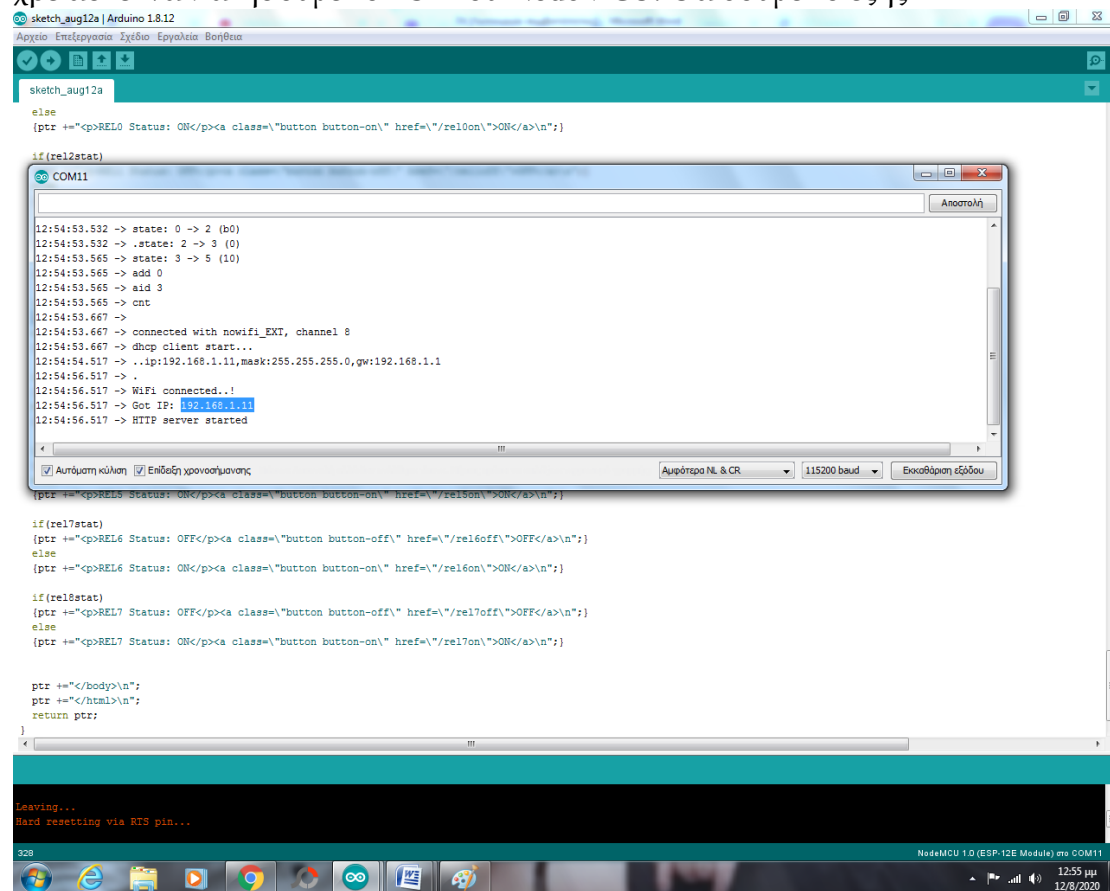
```

```

ptr += "</body>\n";
ptr += "</html>\n";
return ptr;
}

```

Έχοντας μεταγλωττίσει το παραπάνω πρόγραμμα στο Arduino IDE όπως έχουμε περιγράψει παραπάνω, πάμε Εργαλεία--> Παρακολούθηση σειριακής για να δούμε σε real time τη παραμετροποίηση του NodeMCU. Ίσως χρειαστεί να πατήσουμε το RST του NodeMCU. Θα δούμε το εξής

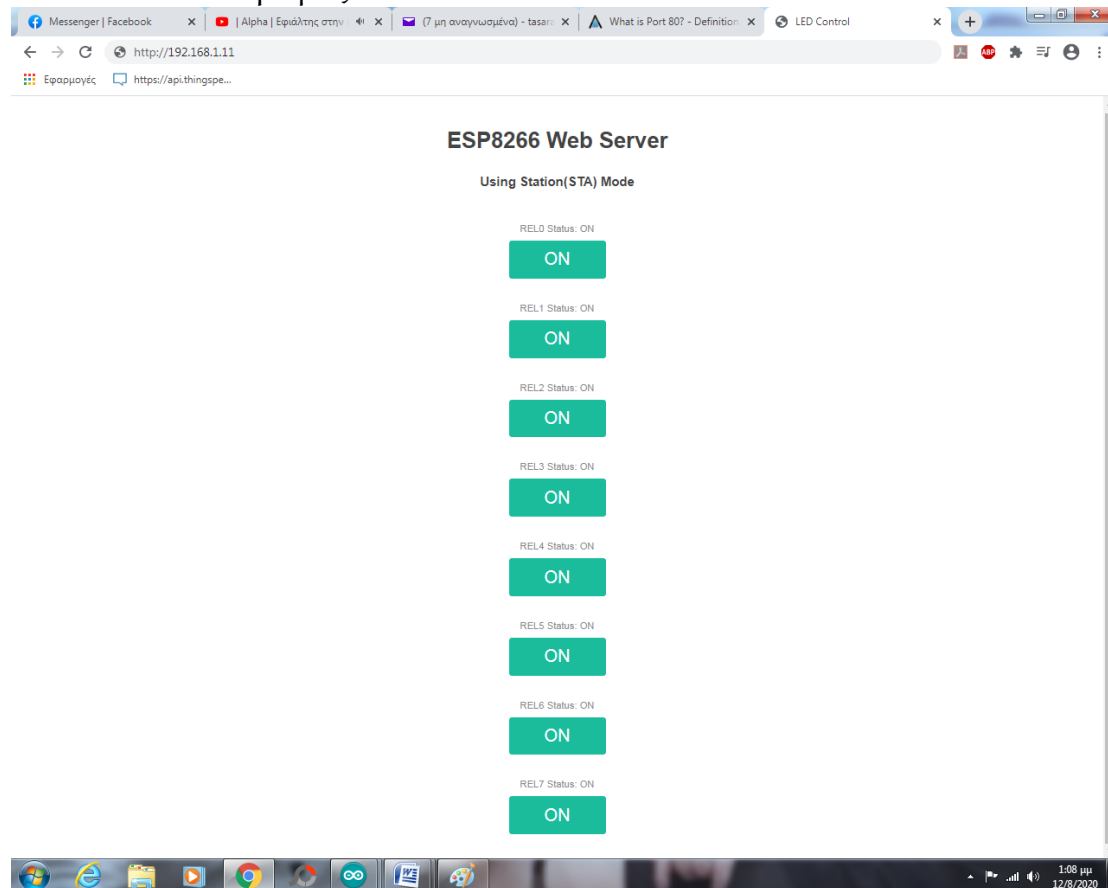


Εικόνα 57. Παρακολούθηση σειριακής

Θα δούμε μια διεύθυνση IP : 192.168.1.11 (μπορεί να βγει εναλλακτικά και η 192.168.1.8) και ανοίγοντας στο browser μας του PC ή κινητού ή tablet τα οποία έχουμε συνδέσει στο Internet στο ίδιο SSID με αυτό που έχουμε ορίσει στην αρχή του προγράμματος μας (const char* ssid = "WiFi SSID"; // Το όνομα του WiFi δικτύου μας const char* password = "WiFi access password"; //Κωδικός WiFi) τη διεύθυνση <http://192.168.1.11/> (ή όποια IP μας βγάλει) ανοίγει η

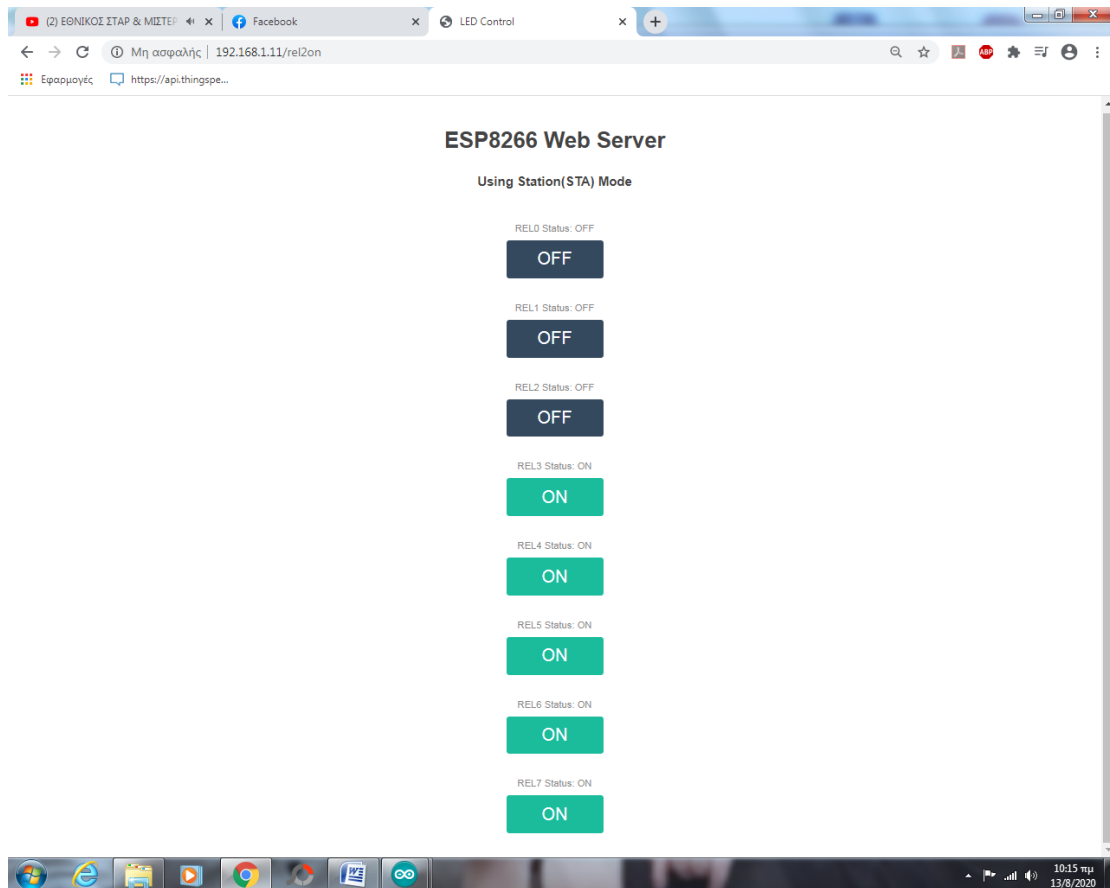
ιστοσελίδα ελέγχου. Όταν τα "κουτάκια" πράσινα σημαίνει ότι τα LED στη πλακέτα του module είναι αναμμένα και αυτό σημαίνει ότι και τα 8 ρελέ είναι ενεργά. Όταν ένα LED είναι αναμμένο, το αντίστοιχο ρελέ του είναι ενεργό.

Κλικάροντας τα "κουτάκια" αλλάζουμε τη κατάσταση των ρελέ από ON σε OFF και αντιστρόφως.

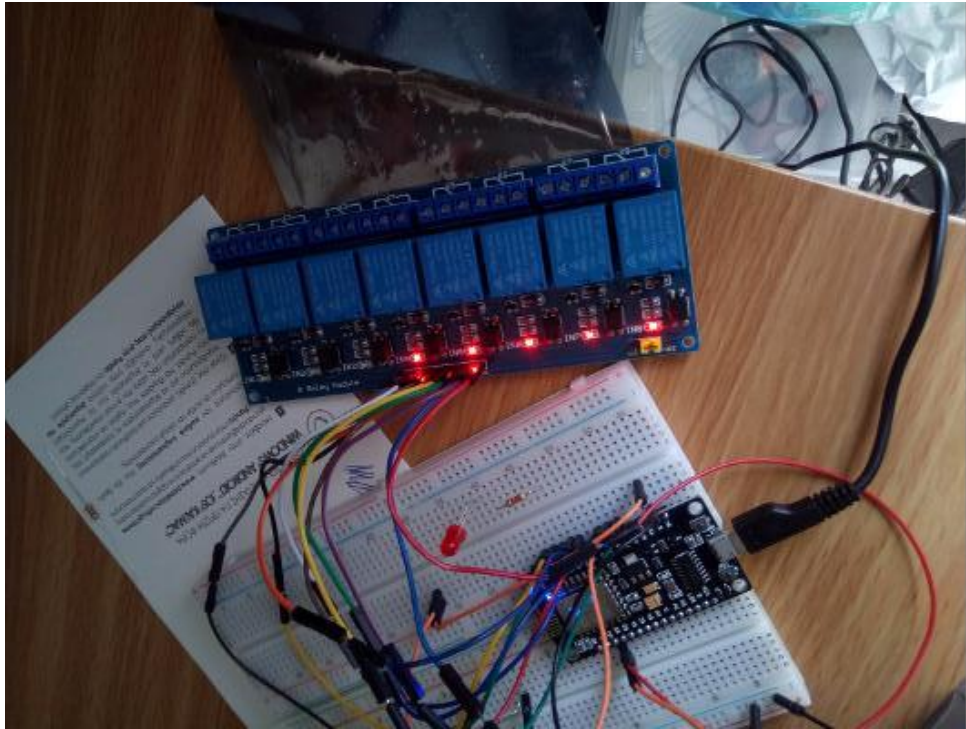


Εικόνα 58. Ιστοσελίδα ελέγχου

Παρακάτω απεικονίζονται φωτογραφίες όπου τα LED1 (REL0), LED2 (REL1), LED3 (REL2) είναι σβηστά και στη ιστοσελίδα ελέγχου τα αντίστοιχα "κουτάκια" είναι με γκρι χρώμα.



Εικόνα 59. Ιστοσελίδα ελέγχου με τα REL0, REL1, REL2 ανενεργά



Εικόνα 60. Φωτογραφία του 8-channel module με το NodeMCU με τα τρία πρώτα LED σβηστά

Το παραπάνω πρόγραμμα που δείξαμε παρουσιάζει ένα μειονέκτημα , το οποίο πρέπει να ξεπεράσουμε. Για να δουλέψει το όλο σύστημα πρέπει να είναι συνεχώς συνδεδεμένα στο ΙΔΙΟ WiFi δίκτυο το NodeMCU και το PC μας ή το smartphone ή το tablet μας. Αυτό βέβαια δεν μας δίνει τη δυνατότητα να είμαστε οπουδήποτε στο κόσμο και να ελέγχουμε τα ρελέ του module μας από πολύ μακριά παρά μόνο εντός της εμβέλειας του WiFi.

Ο σκοπός είναι να είμαστε όπου θέλουμε (αρκεί να έχουμε πρόσβαση στο Internet και το NodeMCU να είναι μόνιμα συνδεδεμένο σε WiFi) και να ελέγχουμε απομακρυσμένα τα ρελέ του module μας από PC, smartphone, tablet.

Για το λόγο αυτό, θα αναπτύξουμε ένα τρόπο να ελέγχουμε τα ρελέ του module χρησιμοποιώντας Cloud Computing ή απλά Cloud. Το Cloud είναι πια αναπόσπαστο κομμάτι του IoT.

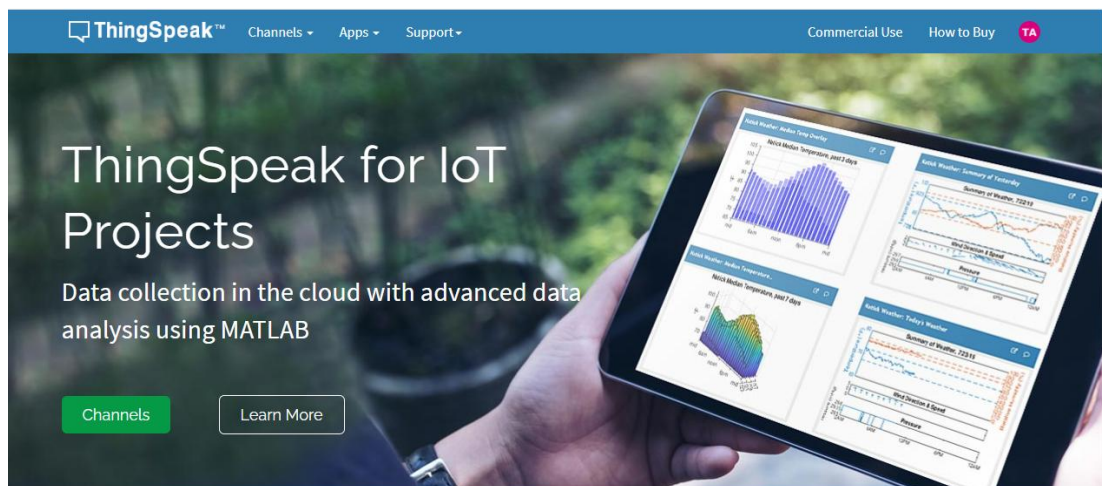
5) ΑΠΟΜΑΚΡΥΣΜΕΝΟΣ ΕΛΕΓΧΟΣ ΜΕΣΩ CLOUD

Cloud εφαρμογές - περιβάλλοντα υπάρχουν πολλά στο Διαδίκτυο, ενώ άλλα κατασκευάζονται από τους χρήστες. Υπάρχουν πολλά IoT cloud τα οποία μπορεί κάποιος να βρει και να τα προσαρμόσει στο ή στα project που ετοιμάζει. Κάποια από αυτά είναι

- Thingworx 8 IoT platform
- Microsoft Azure IoT suite
- IBM Watson IoT Platform
- ThingSpeak IoT

Για το project αυτό η πλατφόρμα που έχει χρησιμοποιηθεί είναι το ThingSpeak IoT. (www.thingSpeak.com), στο οποίο πρέπει να δημιουργήσουμε ένα λογαριασμό - account για να προχωρήσουμε.

Το ThingSpeak είναι μια πλατφόρμα ανοιχτού κώδικα που μας επιτρέπει να συλλέγουμε και να αποθηκεύουμε δεδομένα αισθητήρων στο cloud. Μας παρέχει την εφαρμογή για ανάλυση και οπτικοποίηση των δεδομένων μας στο Matlab. Μπορούμε να χρησιμοποιήσουμε το Arduino, το Raspberry Pi και το Beaglebone για να στείλουμε π.χ. δεδομένα αισθητήρα και να δημιουργήσουμε ένα ξεχωριστό κανάλι για την αποθήκευση δεδομένων



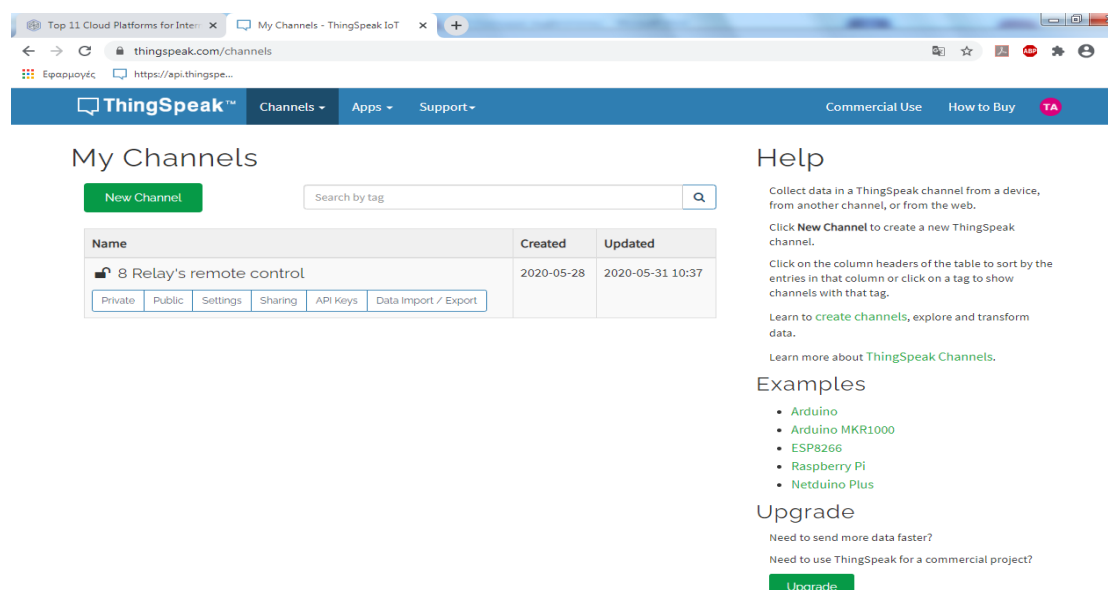
Εικόνα 61. ThingSpeak αρχική σελίδα

Πλεονεκτήματα

- Δωρεάν φιλοξενία για κανάλια
- Εύκολη απεικόνιση
- Παρέχει πρόσθετες δυνατότητες για Ruby, Node.js και Python

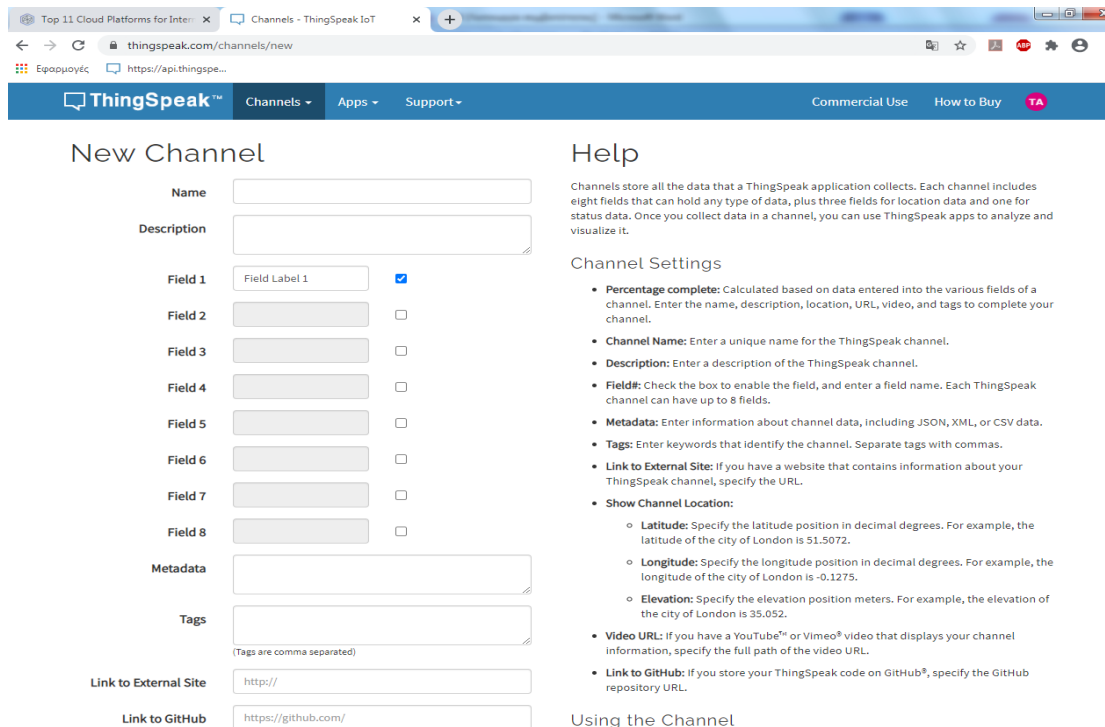
Μειονεκτήματα

- Περιορισμένη μεταφόρτωση δεδομένων για API
- Το ThingSpeak API μπορεί να αποτελέσει εμπόδιο για αρχάριους



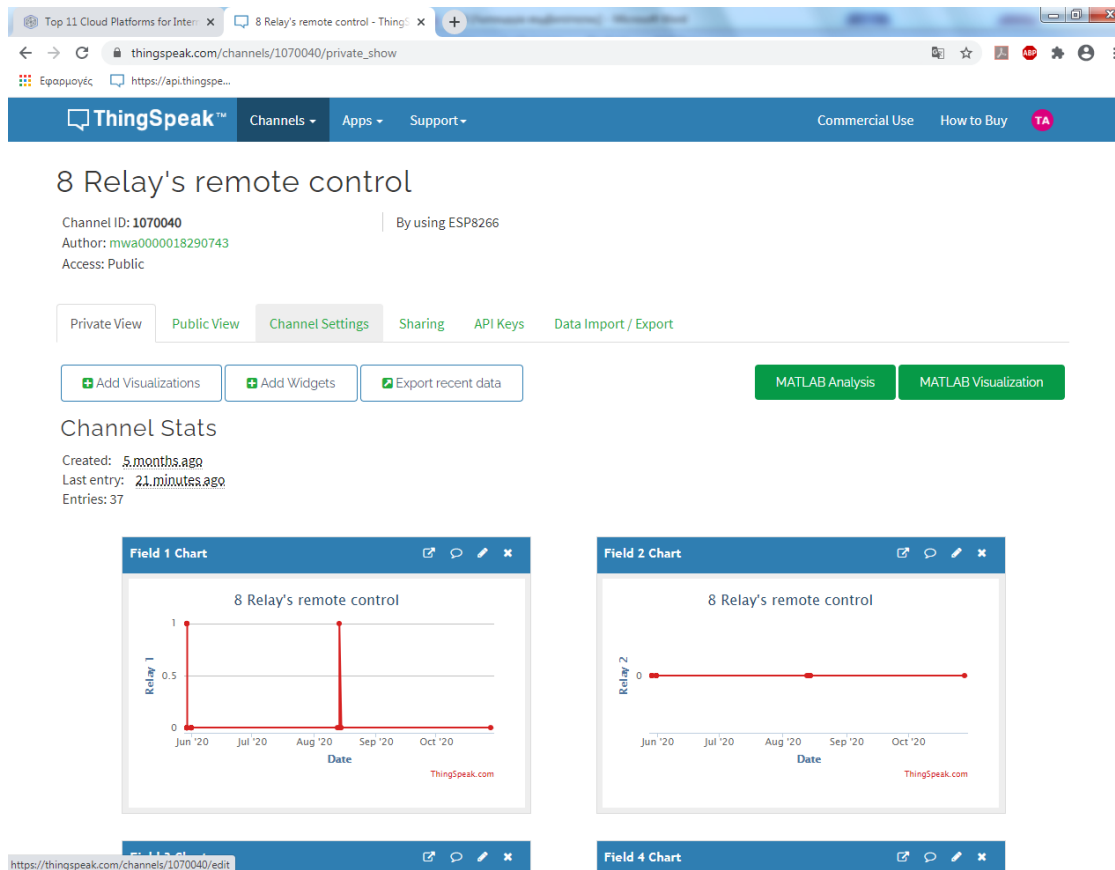
Εικόνα 62. ThingSpeak My Channels

Σύμφωνα με τη πάνω σελίδα, πρέπει να ορίσουμε τον αριθμό των καναλιών που χρειαζόμαστε για το project μας. Στο Name βάζουμε ένα δικό μας όνομα, στο Description μια περιγραφή και στα Fields βάζουμε όνομα (προαιρετικά). Δημιουργείται μετά ένα Channel ID όπως φαίνεται στην εικόνα 60.



Εικόνα 63. ThingSpeak My Channels ορισμός

Παρακάτω φαίνεται η σελίδα στην οποία ρυθμίζουμε τα κανάλια του project (στη δική μας περίπτωση είναι 8) από τη καρτέλα Channel Settings, υπάρχει και η καρτέλα API Keys (Application Programming Interface) με την οποία θα ελέγχουμε τη κατάσταση των ρελέ.



Εικόνα 64. Ρυθμίσεις καναλιών στο ThingSpeak

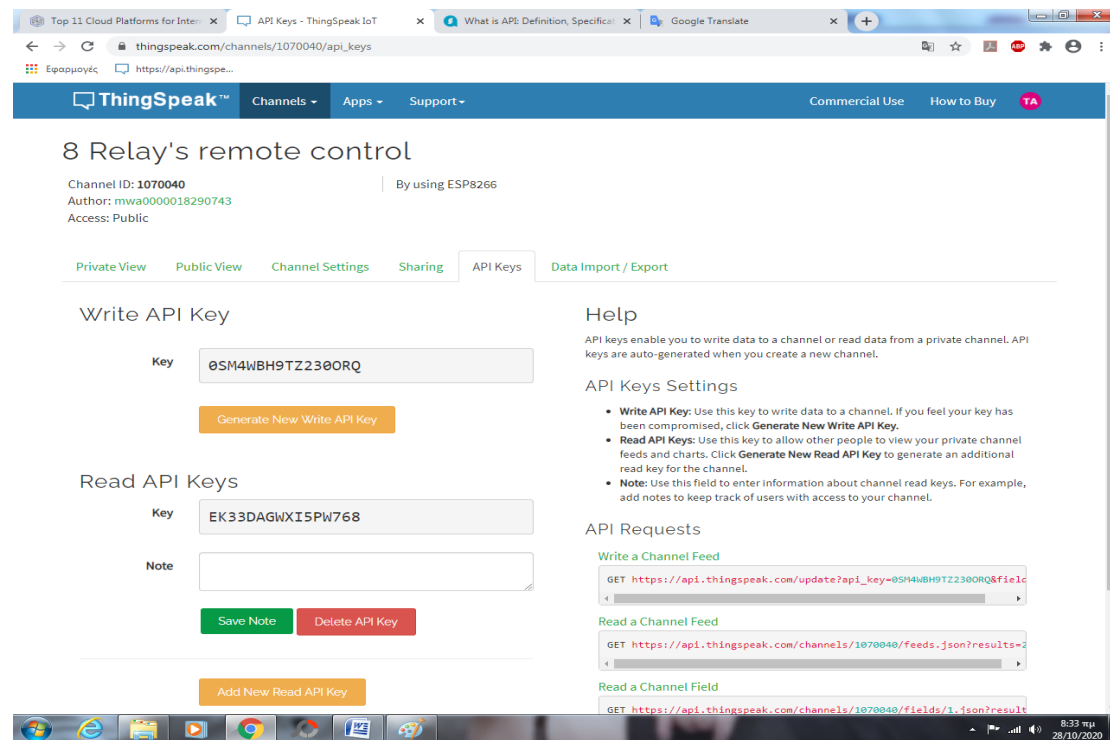
Το API είναι ένα σύνολο κώδικα προγραμματισμού που επιτρέπει τη μετάδοση δεδομένων μεταξύ ενός προϊόντος λογισμικού και ενός άλλου. Περιέχει επίσης τους όρους αυτής της ανταλλαγής δεδομένων.

Τα API's αποτελούνται από δύο στοιχεία:

- Τεχνική προδιαγραφή που περιγράφει τις επιλογές ανταλλαγής δεδομένων μεταξύ λύσεων με την προδιαγραφή που έχει τη μορφή αιτήματος για επεξεργασία και πρωτόκολλα παράδοσης δεδομένων
- Διεπαφή λογισμικού γραμμένη σύμφωνα με τις προδιαγραφές που την αντιπροσωπεύουν

Το λογισμικό που πρέπει να έχει πρόσβαση σε πληροφορίες (δηλαδή, τιμές δωματίων ξενοδοχείου X για συγκεκριμένες ημερομηνίες) ή λειτουργικότητα (δηλαδή, μια διαδρομή από το σημείο A έως το σημείο B σε έναν χάρτη με βάση την τοποθεσία ενός χρήστη) από άλλο λογισμικό, καλεί το API του ενώ καθορίζει το απαιτήσεις για τον τρόπο παροχής δεδομένων /

λειτουργικότητας. Το άλλο λογισμικό επιστρέφει δεδομένα / λειτουργίες που ζητήθηκαν από την προηγούμενη εφαρμογή.



Εικόνα 65. Εμφάνιση API's στο ThingSpeak

Χρειάζεται όμως απαραίτητα να φορτώσουμε στο NodeMCU το κατάλληλα γραμμένο πρόγραμμα στο Arduino IDE προκειμένου να λειτουργήσει το όλο σύστημα.

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

```
/*Put your SSID & Password*/
const char* ssid = "nowifi_EXT"; // Enter SSID here
const char* password = "xxxxxxxxxxxx"; //Enter Password here
```

```
unsigned long channel =1070040;
```

```
unsigned int Relay1=1;  
unsigned int Relay2=2;  
unsigned int Relay3=3;  
unsigned int Relay4=4;  
unsigned int Relay5=5;  
unsigned int Relay6=6;  
unsigned int Relay7=7;  
unsigned int Relay8=8;
```

```
WiFiClient client;
```

```
void setup()  
{
```

```
    Serial.begin(115200);  
    delay(100);
```

```
    pinMode(D0,OUTPUT);  
    pinMode(D1,OUTPUT);  
    pinMode(D2,OUTPUT);  
    pinMode(D3,OUTPUT);  
    pinMode(D4,OUTPUT);  
    pinMode(D5,OUTPUT);  
    pinMode(D6,OUTPUT);  
    pinMode(D7,OUTPUT);
```

```
    digitalWrite(D0,HIGH);  
    digitalWrite(D1,HIGH);  
    digitalWrite(D2,HIGH);  
    digitalWrite(D3,HIGH);  
    digitalWrite(D4,HIGH);  
    digitalWrite(D5,HIGH);  
    digitalWrite(D6,HIGH);  
    digitalWrite(D7,HIGH);
```

```
    Serial.println();  
    Serial.println();  
    Serial.println("Connecting to ");  
    Serial.println(ssid);
```

```

//connect to your local wi-fi network
WiFi.begin(ssid, password);

//check wi-fi is connected to wi-fi network
while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected..!");
Serial.println("IP address :");
Serial.println(WiFi.localIP());
Serial.print("Netmask:");
Serial.println(WiFi.subnetMask());
Serial.print("Gateway:");
Serial.println(WiFi.gatewayIP());

ThingSpeak.begin(client);

}
void loop()
{

  int Relay_1=ThingSpeak.readFloatField(channel,Relay1);
  int Relay_2=ThingSpeak.readFloatField(channel,Relay2);
  int Relay_3=ThingSpeak.readFloatField(channel,Relay3);
  int Relay_4=ThingSpeak.readFloatField(channel,Relay4);
  int Relay_5=ThingSpeak.readFloatField(channel,Relay5);
  int Relay_6=ThingSpeak.readFloatField(channel,Relay6);
  int Relay_7=ThingSpeak.readFloatField(channel,Relay7);
  int Relay_8=ThingSpeak.readFloatField(channel,Relay8);

  if(Relay_1==1)
  {
    digitalWrite(D0,LOW);
    Serial.println("Relay 1 is ON");
  }
  else if(Relay_1==0)

```

```

{
  digitalWrite(D0,HIGH);
  Serial.println("Relay 1 is OFF");
}

if(Relay_2==1)
{
  digitalWrite(D1,LOW);
  Serial.println("Relay 2 is ON");
}
else if(Relay_2==0)
{
  digitalWrite(D1,HIGH);
  Serial.println("Relay 2 is OFF");
}

if(Relay_3==1)
{
  digitalWrite(D2,LOW);
  Serial.println("Relay 3 is ON");
}
else if(Relay_3==0)
{
  digitalWrite(D2,HIGH);
  Serial.println("Relay 3 is OFF");
}

if(Relay_4==1)
{
  digitalWrite(D3,LOW);
  Serial.println("Relay 4 is ON");
}
else if(Relay_4==0)
{
  digitalWrite(D3,HIGH);
  Serial.println("Relay 4 is OFF");
}

if(Relay_5==1)
{
  digitalWrite(D4,LOW);
  Serial.println("Relay 5 is ON");
}

```

```

}
else if(Relay_5==0)
{
    digitalWrite(D4,HIGH);
    Serial.println("Relay 5 is OFF");
}
if(Relay_6==1)
{
    digitalWrite(D5,LOW);
    Serial.println("Relay 6 is ON");
}
else if(Relay_6==0)
{
    digitalWrite(D5,HIGH);
    Serial.println("Relay 6 is OFF");
}
if(Relay_7==1)
{
    digitalWrite(D6,LOW);
    Serial.println("Relay 7 is ON");
}
else if(Relay_7==0)
{
    digitalWrite(D6,HIGH);
    Serial.println("Relay 7 is OFF");
}
if(Relay_8==1)
{
    digitalWrite(D7,LOW);
    Serial.println("Relay 8 is ON");
}
else if(Relay_8==0)
{
    digitalWrite(D7,HIGH);
    Serial.println("Relay 8 is OFF");
}
Serial.println(Relay_1);
Serial.println(Relay_2);
Serial.println(Relay_3);
Serial.println(Relay_4);
Serial.println(Relay_5);
Serial.println(Relay_6);

```

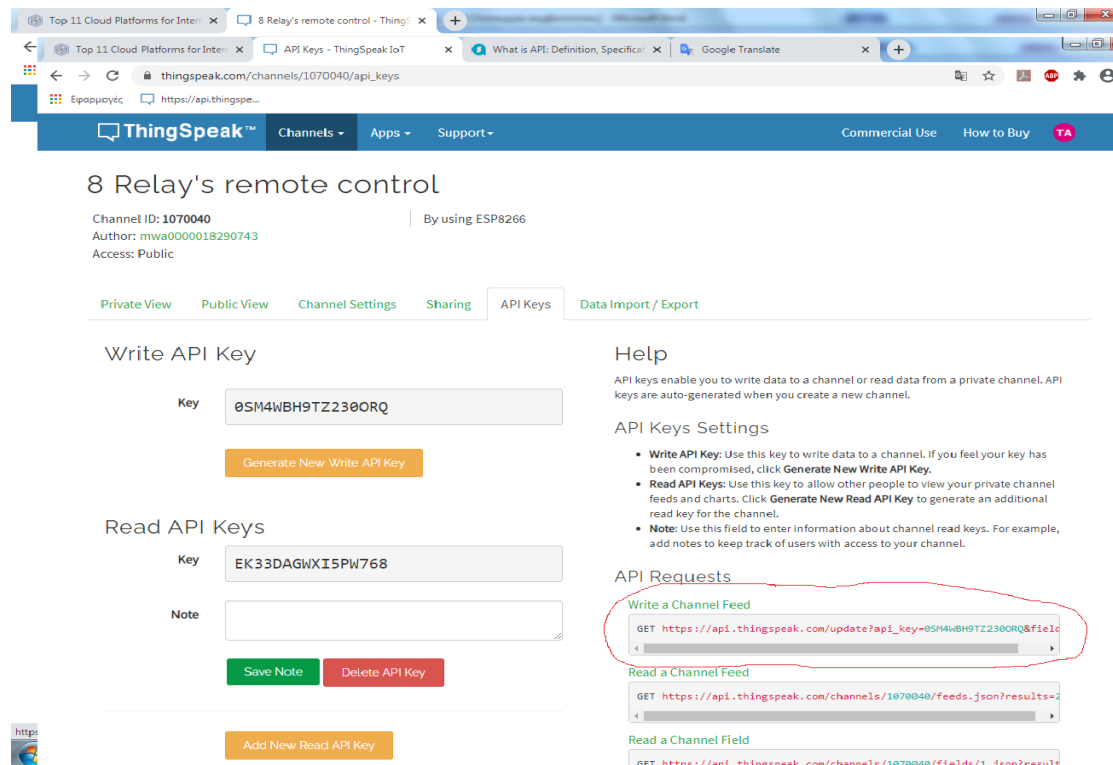
```

Serial.println(Relay_7);
Serial.println(Relay_8);

delay(2000);
}

```

Φορτώνοντας το παραπάνω πρόγραμμα μέσω του Arduino IDE στο NodeMCU με τη γνωστή διαδικασία, πηγαίνουμε στη σελίδα των API's του Thingspeak και κάνουμε αντιγραφή το link που φαίνεται κυκλωμένο με κόκκινο.



Εικόνα 66. API Thingspeak

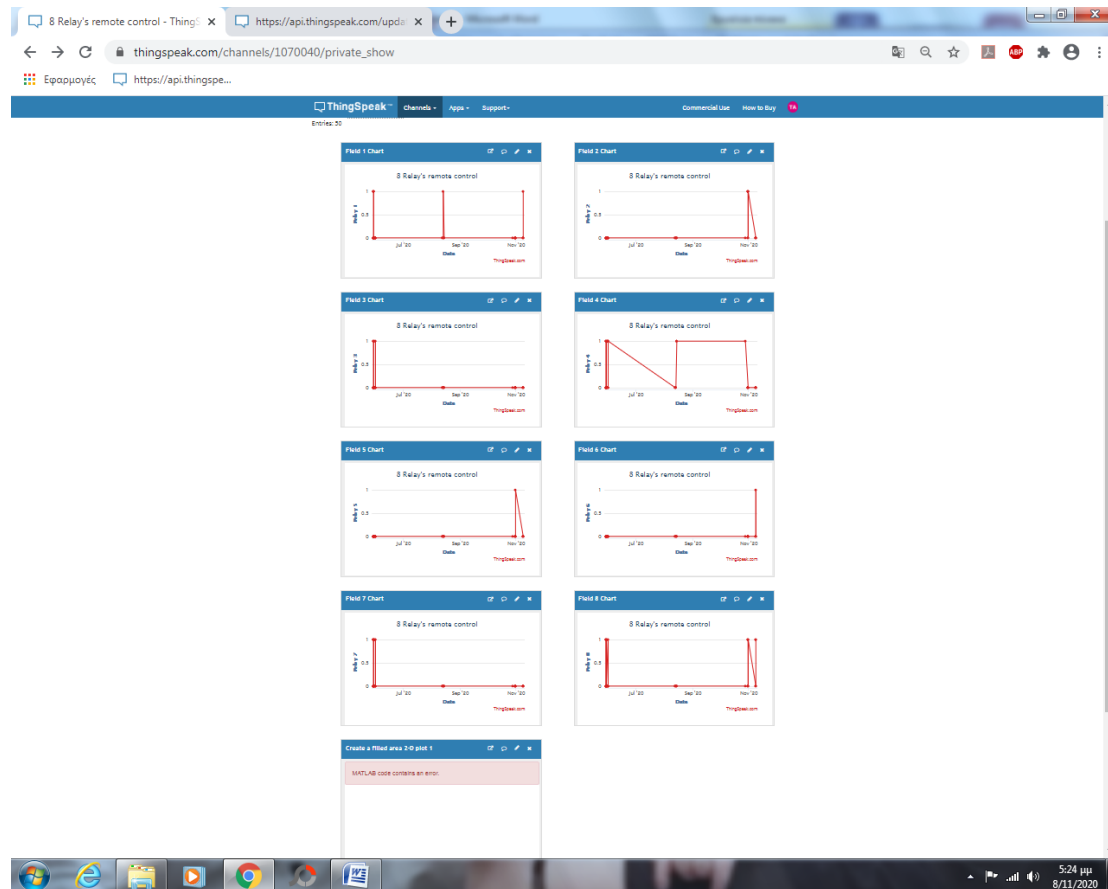
Προσθέτουμε και τα 8 fields στο URL αυτό και ανοίγοντας νέα καρτέλα στο browser μας (πχ. το Google Chrome)

https://api.thingspeak.com/update?api_key=0SM4WBH9TZ230ORQ&field1=0&field2=0&field3=0&field4=0&field5=0&field6=0&field7=0&field8=0 και μετά πατάμε Enter.

Αν εμείς θελήσουμε να ενεργοποιήσουμε το ρελέ 1 και ρελέ 8, πρέπει το URL να είναι

https://api.thingspeak.com/update?api_key=0SM4WBH9TZ230ORQ&field1=1&field2=0&field3=0&field4=0&field5=0&field6=0&field7=0&field8=1

ld2=0&field3=0&field4=0&field5=1&field6=0&field7=0&field8=1, μετά πατάμε Enter ανοίγουν τα ρελέ 1 και 8, ανάβοντας την ίδια στιγμή τα αντίστοιχα LED.



Εικόνα 67. Ημερολογιακή κατάσταση των 8 ρελέ

Επειδή η παραπάνω διαδικασία αφορά έλεγχο του συστήματός μας μέσω PC και επειδή είναι πολλές φορές άβολο να κουβαλάμε μαζί μας το laptop, πιο κάτω θα δείξουμε πως γίνεται ο απομακρυσμένος έλεγχος μέσω smartphone ή tablet.

Διαμορφώνοντας το κατάλληλα το πρόγραμμα ,το φορτώνουμε μέσω του Arduino IDE όπως έχουμε περιγράψει παραπάνω, μπορούμε μέσω της εφαρμογής ThingSpeak κατεβάζοντάς τη από το Google Play στο κινητό μας τηλέφωνο ,να ελέγχουμε πάλι απομακρυσμένα το σύστημά μας. Το πρόγραμμα είναι παρακάτω

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

```

#define RELAY1 D0
#define RELAY2 D1
#define RELAY3 D2
#define RELAY4 D3
#define RELAY5 D4
#define RELAY6 D5
#define RELAY7 D6
#define RELAY8 D7

/*Put your SSID & Password*/
const char *ssid = "xxxxxx"; // Enter SSID here
const char *password = "xxxxxxxxxxxx"; //Enter Password here

int status = WL_IDLE_STATUS;
WiFiClient client;
int value1,value2,value3,value4,value5,value6,value7,value8; // variable to save
channel field reading

unsigned long myChannelNumber = 1070040;
const char *myAPIread = "EK33DAGWXI5PW768";

void setup()
{

  Serial.begin(115200);
  delay(1000);

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

  //connect to your local wi-fi network
  WiFi.begin(ssid, password);

  //check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);

```

```

Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected..!");
Serial.println("IP address :");
Serial.println(WiFi.localIP());
Serial.print("Netmask:");
Serial.println(WiFi.subnetMask());
Serial.print("Gateway:");
Serial.println(WiFi.gatewayIP());

ThingSpeak.begin(client);

pinMode(RELAY1,OUTPUT);
pinMode(RELAY2,OUTPUT);
pinMode(RELAY3,OUTPUT);
pinMode(RELAY4,OUTPUT);
pinMode(RELAY5,OUTPUT);
pinMode(RELAY6,OUTPUT);
pinMode(RELAY7,OUTPUT);
pinMode(RELAY8,OUTPUT);

digitalWrite(RELAY1,HIGH);
digitalWrite(RELAY2,HIGH);
digitalWrite(RELAY3,HIGH);
digitalWrite(RELAY4,HIGH);
digitalWrite(RELAY5,HIGH);
digitalWrite(RELAY6,HIGH);
digitalWrite(RELAY7,HIGH);
digitalWrite(RELAY8,HIGH);

}
void loop()
{
  // Read the latest value from field 1 of your channel
  value1= ThingSpeak.readIntField(myChannelNumber, 1, myAPIread);
  Serial.print("value1 = ");
  Serial.println(value1);

  if (value1 == 1)
  {
    digitalWrite(RELAY1, LOW);

```

```

}
else
{
    digitalWrite(RELAY1, HIGH);
}
// Read the latest value from field 2 of your channel
value2= ThingSpeak.readIntField(myChannelNumber, 2, myAPIread);
Serial.print("value2 = ");
Serial.println(value2);

if (value2 == 1)
{
    digitalWrite(RELAY2, LOW);

}
else
{
    digitalWrite(RELAY2, HIGH);

}
// Read the latest value from field 3 of your channel
value3= ThingSpeak.readIntField(myChannelNumber, 3, myAPIread);
Serial.print("value3 = ");
Serial.println(value3);

if (value3 == 1)
{
    digitalWrite(RELAY3, LOW);

}
else
{
    digitalWrite(RELAY3, HIGH);

}
// Read the latest value from field 4 of your channel
value4= ThingSpeak.readIntField(myChannelNumber, 4, myAPIread);
Serial.print("value4 = ");
Serial.println(value4);

if (value4 == 1)

```

```

{
    digitalWrite(RELAY4, LOW);

}
else
{
    digitalWrite(RELAY4, HIGH);

}
// Read the latest value from field 5 of your channel
value5= ThingSpeak.readIntField(myChannelNumber, 5, myAPIread);
Serial.print("value5 = ");
Serial.println(value5);

if (value5 == 1)
{
    digitalWrite(RELAY5, LOW);

}
else
{
    digitalWrite(RELAY5, HIGH);

}
// Read the latest value from field 6 of your channel
value6= ThingSpeak.readIntField(myChannelNumber, 6, myAPIread);
Serial.print("value6 = ");
Serial.println(value6);

if (value6 == 1)
{
    digitalWrite(RELAY6, LOW);

}
else
{
    digitalWrite(RELAY6, HIGH);

}
// Read the latest value from field 7 of your channel
value7= ThingSpeak.readIntField(myChannelNumber, 7, myAPIread);
Serial.print("value7 = ");

```

```

Serial.println(value7); // debugging instrument

if (value7 == 1)
{
    digitalWrite(RELAY7, LOW);

}
else
{
    digitalWrite(RELAY7, HIGH);

}
// Read the latest value from field 8 of your channel
value8= ThingSpeak.readIntField(myChannelNumber, 8, myAPIread);
Serial.print("value8 = ");
Serial.println(value8);

if (value8 == 1)
{
    digitalWrite(RELAY8, LOW);

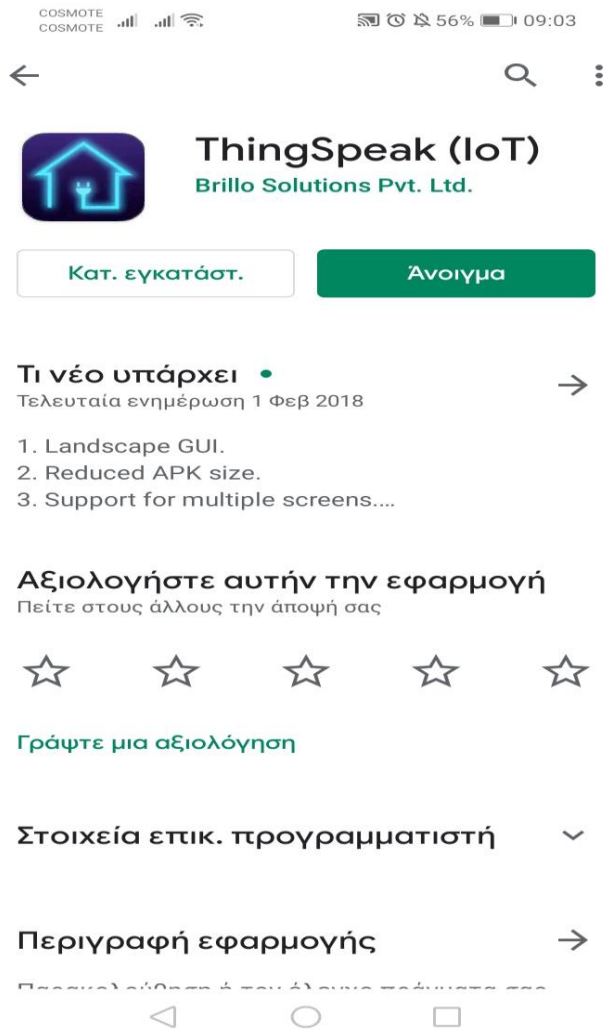
}
else
{
    digitalWrite(RELAY8, HIGH);

}

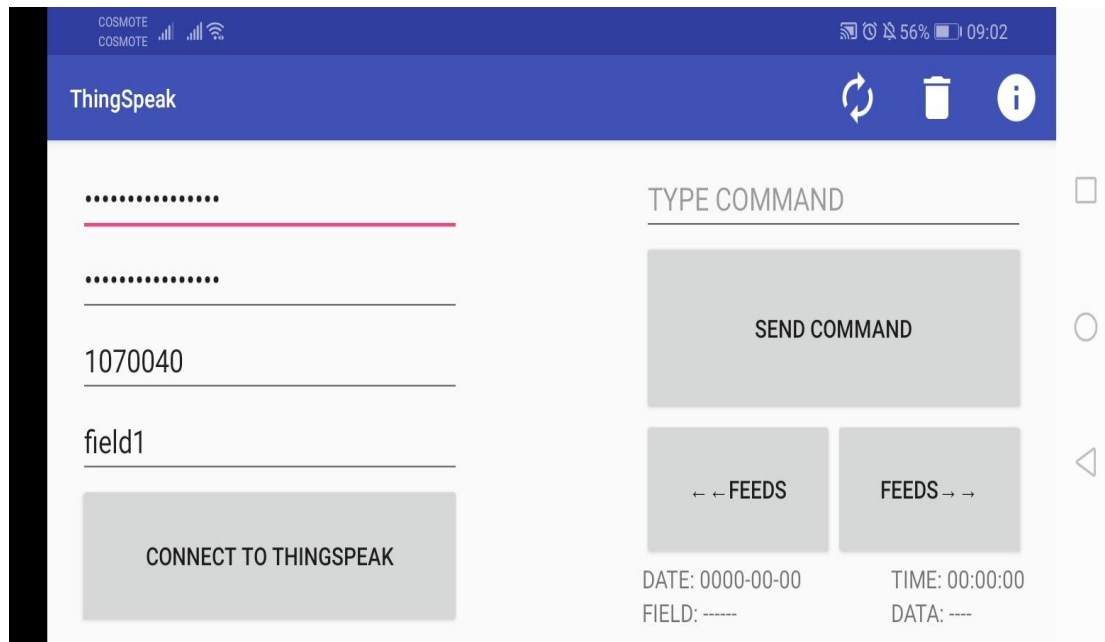
delay(2000);
}

```

Μόλις μπούμε στο Google Play από το android κινητό μας ,
 πληκτρολογούμε στην Αναζήτηση "thingspeak" και κατεβάζουμε την εφαρμογή
 ThingSpeak (IoT)



Εικόνα 68. Εφαρμογή ThingSpeak (IoT) στο Google Play



Εικόνα 69. Μενού της εφαρμογής

Αφού κατεβάσουμε την εφαρμογή αυτή, μπαίνουμε στο Internet από το κινητό είτε μέσω WiFi είτε μέσω του δικτύου κινητής τηλεφωνίας (3G, 4G, 4G+, 5G) .

Στην Εικόνα 65 είναι το μενού της εφαρμογής , στην οποία πρέπει να συμπληρώσουμε τα API keys, το κανάλι, τα fields . Μετά πατάμε CONNECT TO THINGSPEAK , ύστερα δίνουμε εντολές στα fields πχ. 1 ή 0 και μετά πατάμε SEND COMMAND. Αλλάζοντας από field1 σε field2, field3 κτλ , δίνουμε εντολές σε κάθε ένα field κάθε φορά.

Έτσι λοιπόν, έχοντας το σύστημά μας μόνιμα συνδεδεμένο σε κάποιο WiFi δίκτυο, μπορούμε να το ελέγχουμε μέσω Internet ενώ βρισκόμαστε οπουδήποτε στο κόσμο αρκεί να υπάρχει πρόσβαση στο Internet.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Το project αυτό μάς βοηθά να κατανοήσουμε τη λειτουργία αλλά και τα οφέλη του IoT. Μπορούμε να ελέγχουμε γενικότερα μέσω του Internet οποιοδήποτε σύστημα από οποιοδήποτε σημείο του πλανήτη και αν βρισκόμαστε αρκεί να έχουμε πρόσβαση στο Internet.

Με τη συνεχή εξέλιξη του Internet, των smartphones, των tablets αλλά και των υπολογιστών, το IoT γίνεται ολοένα πιο γρήγορο, πιο αξιόπιστο δίνοντας μας μεγάλο όγκο πληροφοριών και δεδομένων, σχετικά με τα συστήματα που αποτελούν την εκάστοτε εφαρμογή μας , σε πραγματικό χρόνο. Σε αυτό , μεγάλο ρόλο παίζουν και οι cloud IoT πλατφόρμες που συνεχώς αυξάνονται και γίνονται και αυτές πιο λειτουργικές παρέχοντας στο κάθε χρήστη ευκολία και αξιοπιστία. Με τον απομακρυσμένο έλεγχο μέσω του IoT μπορούμε π.χ. να παρακολουθούμε τα δεδομένα και τη κατάσταση ενός δικτύου αισθητήρων και να βλέπουμε τα αποτελέσματα (πχ. θερμοκρασία χώρου, υγρασία) σε μια cloud πλατφόρμα.

Το σύστημα που αναπτύξαμε μάς βοηθά επίσης να ελέγξουμε όχι μόνο LED και λάμπες φωτισμού, αλλά και DC/AC κινητήρες , θα μπορούσε να ήταν κομμάτι του όλου hardware π.χ. ενός φούρνου , ο οποίος να είχε WiFi συνδεσιμότητα και ο χρήστης να τον ενεργοποιήσει από το κινητό του τηλέφωνο, να ανοιγοκλείνει ηλεκτροβαλβίδες αερίου ή νερού ακόμα και κλιματιστικά με WiFi όπου κάποιος να μπορεί να το ανοίγει/κλείνει από το κινητό του τηλέφωνο.

Αυτό το project μάς βοηθά να προγραμματίζουμε τα Arduino boards ή όποιο άλλο board είναι συμβατό με Arduino (π.χ. NodeMCU) αναλόγως την εφαρμογή μας και γενικά τα Arduino είναι τόσο διαδεδομένα και πολύ εφαρμόσιμα με τα οποία μπορούμε ακόμα να υλοποιήσουμε και ρομπότ. Με τις cloud πλατφόρμες μάς δίνεται η δυνατότητα να ελέγχουμε τα συστήματά μας χωρίς να είναι απαραίτητο να είναι και ο χρήστης συνδεδεμένος στο ίδιο WiFi δίκτυο με το σύστημα. Επιλέγοντας το κατάλληλο board και IoT πλατφόρμα , μπορούμε να υλοποιήσουμε όποιο IoT σύστημα θέλουμε και να είναι όσο πιο εύρωστο γίνεται.

Η πολυπλοκότητα τόσο σε software όσο και σε hardware ενός IoT συστήματος πηγάζει πάντα από την ανάγκη και αφού υπάρχουν τα κατάλληλα εργαλεία και οι τεχνικές , το IoT θα συνεχίζει να μας εκπλήσσει συνεχώς βελτιώνοντάς μας και σαν μηχανικούς αλλά και σαν ανθρώπους.

ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

- 1) [http://linksprite.com/wiki/index.php5?title=LinkNode_R8: Arduino compatible WiFi relay controller](http://linksprite.com/wiki/index.php5?title=LinkNode_R8:_Arduino_compatible_WiFi_relay_controller)
- 2) <http://www.linksprite.io>
- 3) <http://www.hfremote.us/files/Arduino.pdf>
- 4) <https://dzone.com/articles/building-an-iot-system-using-arduino-and-iot-cloud>
- 5) [https://iot-playground.com/blog/2-uncategorised/77-esp8266-internet controlled-switch-easyiot-cloud-mqtt-api](https://iot-playground.com/blog/2-uncategorised/77-esp8266-internet-controlled-switch-easyiot-cloud-mqtt-api)
- 6) [http://www.astrj.com/EDUCATIONAL-USE-OF-CLOUD-COMPUTING-AND-AT-MEGA-MICROCONTROLLER-A-CASE-STUDY-OF-AN-ALARM -SYSTEM,62701,0,2.html](http://www.astrj.com/EDUCATIONAL-USE-OF-CLOUD-COMPUTING-AND-AT-MEGA-MICROCONTROLLER-A-CASE-STUDY-OF-AN-ALARM-SYSTEM,62701,0,2.html)
- 7) <http://freepdf-books.com/download/?file=2176>
- 8) [http://www.circuitbasics.com/wp-content/uploads/2015/11/SRD-05VDC-SL-C -Datasheet.pdf](http://www.circuitbasics.com/wp-content/uploads/2015/11/SRD-05VDC-SL-C-Datasheet.pdf)
- 9) <https://www.hellasdigital.gr/electronics/relays/songle-relay-5v-t73-srd-5vdc-sl-c-5pin-10a-for-arduino/>
- 10) <https://cloudacademy.com/blog/disadvantages-of-cloud-computing/>
- 11) <https://www.thebalancesmb.com/disadvantages-of-cloud-computing-4067218>
- 12) ΒΙΒΛΙΟ : Προγραμματίζοντας τον μικροελεγκτή AVR, Dhananjay V. Gadre, εκδ. ΤΖΙΟΛΑ
- 13) ΒΙΒΛΙΟ : Οδηγός της C, Herbert Schildt, εκδ. Μ. Γκιούρδας
- 14) <https://el.wikipedia.org/wiki/Arduino>
- 15) <https://opensource.com/resources/what-open-hardware>
- 16) ΒΙΒΛΙΟ : Beginning Arduino Programming : Writing Code For the Most Popular Microcontroller Board In The World
- 17) ΒΙΒΛΙΟ : Εισαγωγή στην ασύρματη δικτύωση 2η αμερικάνικη έκδοση, John Ross, Εκδ. Κλειδάριθμος
- 18) ΒΙΒΛΙΟ : Δίκτυα Υπολογιστών, 4η αμερικάνικη έκδοση, Andrew S. Tanenbaum, Εκδ. Κλειδάριθμος
- 19) http://teachers.teicm.gr/mademlis/networks_I_1.pdf

- 20) <http://www.epset.gr/el/content/ypologistiko-nefos-cloud-computing>
 - 21) <https://doublehorn.com/saas-paas-and-iaas-understanding/>
 - 22) Πτυχιακή εργασία του κου Φώτιου Σολδάτου "Internet Of Things", Πολυτεχνική Σχολή Πανεπιστημίου Πατρών, Τμημ. Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, 30/6/2017
 - 23) https://thingspeak.com/pages/how_to
 - 24) <https://maker.pro/arduino/tutorial/how-to-make-a-web-controlled-servo-with-arduino-and-esp8266>
 - 25) <https://www.electronicshub.org/web-controlled-servo-using-esp8266/>
 - 26) [https://electrosome.com/arduino-web-controlled-servo/#Arduino Uno Program](https://electrosome.com/arduino-web-controlled-servo/#Arduino_Uno_Program)
 - 27) <https://store.arduino.cc/arduino-uno-wifi>
 - 28) <https://bitnine.net/blog-useful-information/top-9-iot-open-source-platforms/>
 - 29) [https://www.w3.org/WoT/IG/wiki/images/9/95/The_essential_guide_to_Open Source in IoT - White Paper.pdf](https://www.w3.org/WoT/IG/wiki/images/9/95/The_essential_guide_to_Open_Source_in_IoT_-_White_Paper.pdf)
 - 30) <https://arxiv.org/pdf/1804.01747.pdf>
 - 31) <https://circuitdigest.com/microcontroller-projects/getting-started-with-esp8266-module>
 - 32) <http://qqtrading.com.my/nodemcu-v3-esp8266-wifi-ch340>
 - 33) <https://lastminuteengineers.com/creating-esp8266-web-server-arduino-ide/>
 - 34) <https://www.youtube.com/watch?v=X4r3QCaujOk>
 - 35) https://www.youtube.com/watch?v=h3_sOcrZwYw
 - 36) <https://www.youtube.com/watch?v=r9Wvw4-CGXQ>
 - 37) <http://www.handsontec.com/dataspecs/module/8Ch-relay.pdf>
 - 38) <https://techterms.com/definition>
 - 39) <https://www.youtube.com/watch?v=giACxpN0cGc>
 - 40) <https://www.vishay.com/docs/83172/tlmo1000.pdf>
 - 41) https://www.futurlec.com/LED/Red_0805_SMD_LED.shtml
 - 42) <https://dzone.com/articles/10-cloud-platforms-for-internet-of-things-iot>
-

ΑΝΑΦΟΡΕΣ ΕΙΚΟΝΩΝ ΕΡΓΑΣΙΑΣ

- 1) <https://images.app.goo.gl/BMTeVvUZNtVeAHWCA>
- 2) <https://images.app.goo.gl/n38eQLMtmg79Y4hHA>
- 3) <https://images.app.goo.gl/4CeHZvsLoZDz3LXr5>
- 4) <https://images.app.goo.gl/4uDyugR3DK1P2R8r5>
- 5) <https://images.app.goo.gl/sgy6gMnUx4o1ysrHA>
- 6) <https://images.app.goo.gl/55RknbVFZVLmQdYS6>
- 7) <https://images.app.goo.gl/rm6TDoVsuXKrSjYb8>
- 10) <https://images.app.goo.gl/yXWxZ4LCu6wbxoJy5>
- 11) <https://images.app.goo.gl/Vvq8CPZLUDHheuhi8>
- 12) <https://images.app.goo.gl/yqJvNwjLt6kCDTDX7>
- 13) <https://images.app.goo.gl/R7qXsgA13abRLLb97>
- 14) <https://images.app.goo.gl/EU48mPqYqiUiKSUx9>
- 15) <https://images.app.goo.gl/nX3EZJXdrGs5baxg8>
- 16) <https://images.app.goo.gl/nFAKAnefarwGSh5d7>
- 17) <https://images.app.goo.gl/ZdQH3x4oZzcGC9EC9>
- 18) <https://images.app.goo.gl/GYyRFtRjnHCPYyK7>
- 19) <https://images.app.goo.gl/vp8HPRhGhjVJMvxB9>
- 20) <https://images.app.goo.gl/89JvJLtpV2WubjQS9>
- 21) <https://images.app.goo.gl/oi5WaDubu4zsuWyu7>
- 22) <https://images.app.goo.gl/WVU2Q76rQi4oNXnB7>
- 23) <https://docplayer.gr/1099083-Oi-genies-ton-diktyon-kinitis-tilefonias-kai-paroysiasi-ton-systimaton-4is-genias-lte-kai-lte-advanced.html>
- 24) <https://images.app.goo.gl/513to9uLYrBv6foa9>
- 25) <https://images.app.goo.gl/PmuSsvXbFhz65fkTA>
- 26) <https://images.app.goo.gl/6gYEvngAD4wXxFeS6>
- 27) <https://images.app.goo.gl/A9QudUiHWaEtyFAEA>
- 28) <https://images.app.goo.gl/pvVXCDhQgLSqa4g88>
- 29) <https://images.app.goo.gl/oMeujKfiufe6TLgY9>
- 32) <https://images.app.goo.gl/caPrdAP9AQ4N3Xhw6>
- 33) <https://images.app.goo.gl/V4Tv57dQZ3QX7XY86>
- 34) <https://images.app.goo.gl/YdLwHHVamCZ4kmRb8>

- 35) <https://images.app.goo.gl/2tRpwNH2a9tTYC6N7>
- 36) <https://images.app.goo.gl/AYguXPTHhKRbKxaM8>
- 37) <https://images.app.goo.gl/vSC7YZwWKwtzCoV76>
- 38) <http://www.handsontec.com/dataspecs/module/8Ch-relay.pdf>
- 39) <http://www.handsontec.com/dataspecs/module/8Ch-relay.pdf>
- 40) <http://www.handsontec.com/dataspecs/module/8Ch-relay.pdf>
- 41) <https://images.app.goo.gl/kiEhCq7yCx5pEFxo7>
- 43) <http://www.handsontec.com/dataspecs/module/8Ch-relay.pdf>
- 44) <https://images.app.goo.gl/zPdehSNKg4sBL4bT8>
- 45) <https://www.everlight.com/file/ProductFile/EL817.pdf>
- 46) <https://www.everlight.com/file/ProductFile/EL817.pdf>
- 49) <https://www.arduino.cc/en/software>
- 61) <https://thingspeak.com/>