



510  
ΑΥΤ

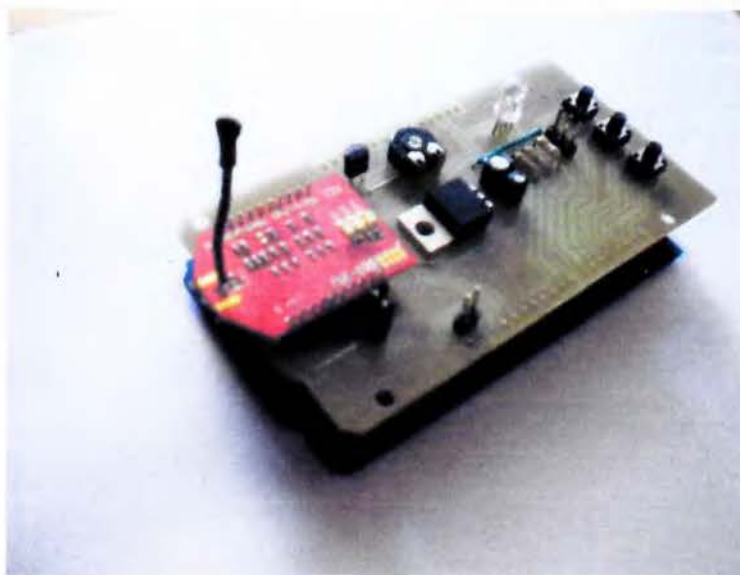
Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Αυτοματισμού

Πτυχιακή εργασία

Θέμα :

**"ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ WEB SERVER ΜΕ ARDUINO ΓΙΑ  
ΤΟΝ ΕΛΕΓΧΟ ΜΕΣΩ ΙΣΤΟΣΕΛΙΔΑΣ"**



**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΜΙΧΑΛΗΣ ΠΑΠΟΥΤΣΙΔΑΚΗΣ**

ΒΙΒΛΙΟΘΗΚΗ  
ΤΕΙ ΠΕΙΡΑΙΑ

**ΦΟΙΤΗΤΗΣ: ΦΟΥΝΤΟΥΚΙΔΗΣ ΠΑΝΑΓΙΩΤΗΣ Α.Μ: 35889**

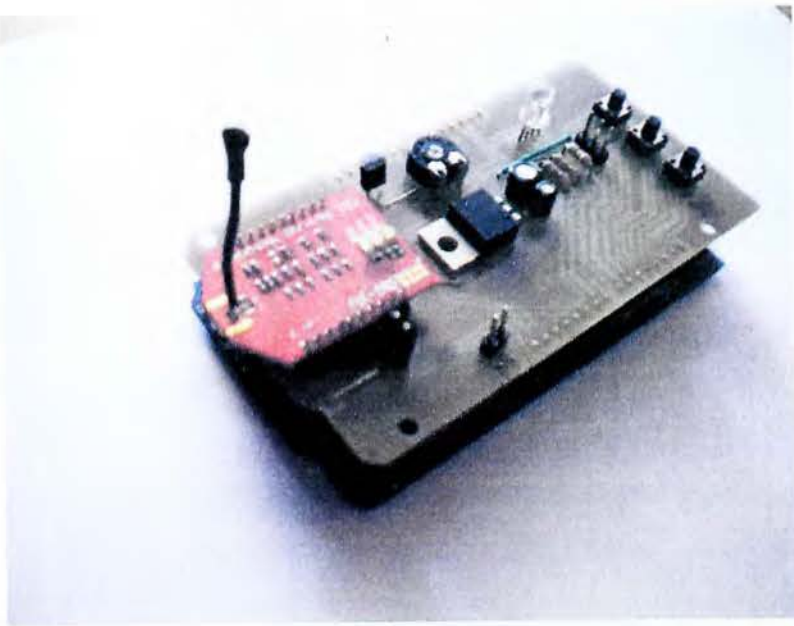


Faculty of Application Technology  
Department of Automation (Eng.)

**Final Year Project**

**Subject :**

**"CREATING A WEB SERVER WITH ARDUINO FOR THE  
CONTROL THROUGH A WEBSITE"**



***SUPERVISOR : MICHALIS PAPOUTSIDAKIS***

***STUDENT : FOYNTOUKIDIS PANAGIOTIS R.N. : 35889***

## ΕΥΧΑΡΙΣΤΙΕΣ

Οφείλω να εκφράσω τις θερμές μας ευχαριστίες στα άτομα που συνέβαλλαν στην αποπεράτωση της πτυχιακής μου εργασίας. Η έρευνά μου δεν θα είχε πραγματοποιηθεί χωρίς την πολύτιμη συμβολή:

του καθηγητή μου κ. Παπουτσιδάκη Μιχαήλ, τον οποίο ευχαριστώ για το ενδιαφέρον και τη συνεχή καθοδήγησή του.

του ξαδέλφου μου που δέχτηκε να συνεργαστεί μαζί μου και να με βοηθήσει στην κατασκευή και τη σχεδίαση της πλακέτας καθώς και του πατέρα μου για την πολύτιμη υποστήριξή του.

# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<b>ΠΕΡΙΕΧΟΜΕΝΑ</b> .....	3
<b>ΠΕΡΙΛΗΨΗ</b> .....	5
<b>ABSTRACT</b> .....	6
<b>1. ΕΙΣΑΓΩΓΗ</b> .....	7
<b>1.1 ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΤΟ ΔΙΑΔΙΚΤΥΟ</b> .....	7
<b>1.2 ΤΟ ΠΡΩΤΟΚΟΛΛΟ IP (Internet Protocol)</b> .....	12
1.2.1 ΔΡΟΜΟΛΟΓΗΣΗ ΠΑΚΕΤΩΝ.....	16
<b>1.3 ΤΟ ΠΡΩΤΟΚΟΛΛΟ TCP</b> .....	18
<b>1.4 ΤΟ ΠΡΟΤΥΠΟ OSI</b> .....	23
<b>1.5 ΤΟ ΠΡΩΤΟΚΟΛΛΟ HTTP</b> .....	27
1.5.1 ΜΟΝΙΜΕΣ ΚΑΙ ΜΗ ΜΟΝΙΜΕΣ ΣΥΝΔΕΣΕΙΣ .....	29
1.5.2 HTTP ΜΗΝΥΜΑΤΑ .....	31
<b>1.6 Η ΓΛΩΣΣΑ HTML</b> .....	33
<b>2. ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ ARDUINO</b> .....	35
<b>2.1 ARDUINO MEGA 2560</b> .....	38
2.1.1 ΕΙΔΙΚΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ARDUINO MEGA.....	39
2.1.2 ΤΡΟΦΟΔΟΤΗΣΗ .....	40

2.1.3 ΠΡΟΣΤΑΣΙΑ ΑΠΟ ΥΠΕΡΕΝΤΑΣΗ ΣΤΟ USB.....	41
2.1.4 ΕΙΣΟΔΟΙ ΚΑΙ ΕΞΟΔΟΙ.....	41
2.1.5 ΕΠΙΚΟΙΝΩΝΙΑ.....	42
2.1.6 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗ.....	43
2.1.7 ΑΥΤΟΜΑΤΟ RESET.....	43
<b>2.2 ΣΧΕΔΙΑΣΗ HARDWARE.....</b>	<b>44</b>
2.2.1 ΣΧΗΜΑΤΙΚΟ ΠΛΑΚΕΤΑΣ.....	46
2.2.2 ΠΑΡΟΥΣΙΑΣΗ ΠΛΑΚΕΤΑΣ.....	47
2.2.3 ΤΕΛΙΚΗ ΜΟΡΦΗ ΠΛΑΚΕΤΑΣ.....	48
<b>2.3 ΠΡΟΓΡΑΜΜΑ SOFTWARE.....</b>	<b>49</b>
2.3.1 ΚΩΔΙΚΑΣ HTML.....	49
2.3.2 ΚΩΔΙΚΑΣ ARDUINO.....	65
<b>3. ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΟΠΤΙΚΕΣ.....</b>	<b>76</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>78</b>

## ΠΕΡΙΛΗΨΗ

Σκοπός της πτυχιακής είναι ο έλεγχος και η διαχείριση δεδομένων μέσω μιας ιστοσελίδας με την οποία θα διευκολύνεται ο χρήστης χωρίς την ανάγκη να παρευρίσκεται ο ίδιος στον χώρο. Η ιστοσελίδα θα μας επιτρέπει τον απομακρυσμένο έλεγχο ορισμένων εισόδων και εξόδων του μικροελεγκτή. Χρησιμοποιούμε τον μικροελεγκτή arduino Mega 2560 που προγραμματίζεται σε γλώσσα C και θα τον συνδέσουμε σε Wi-fi συσκευή μέσω του οποίου συνδεόμαστε σε ένα τοπικό δίκτυο. Μπορούμε να συνδέσουμε επίσης στο δίκτυο και έναν δρομολογητή για να συνδεθούμε στο διαδίκτυο. Στη μνήμη του μικροελεγκτή δημιουργούμε μια ιστοσελίδα μέσω της οποίας επιτρέπεται ο έλεγχος και η προβολή των δεδομένων. Για την προβολή της ιστοσελίδας η Wi-fi συσκευή εφαρμόζει αυτόματα τα πρωτόκολλα TCP, IP και το HTTP. Ο χρήστης θα μπορεί να έχει πρόσβαση στην ιστοσελίδα από διάφορες συσκευές που διαθέτουν δυνατότητα σύνδεσης στο διαδίκτυο. Με αυτόν τον τρόπο θα εξοικονομεί χρόνο και φυσικούς πόρους έχοντας το ίδιο αποτέλεσμα.

## ABSTRACT

The aim of this project is the control and the data management through a website, which will facilitate the user without the need for the user to be preset. The website will allow us the remote control of certain inputs and outputs of the microcontroller. We use the arduino Mega2560 microcontroller which is programmed in C language we will connect it to a Wi-fi device through which the user can have access to a local network. We can also connect a router to the network for internet access. We create a website in the microcontroller memory through which we can control the data. To view the website the Wi-fi device implements the protocols TCP/IP, HTTP automatically. The user can access the site from various devices that have internet access. This will save time and physical resources having the same ultimate results.

# 1. ΕΙΣΑΓΩΓΗ

## 1.1 ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΤΟ ΔΙΑΔΙΚΤΥΟ

Με τον όρο διαδίκτυο (Internet) υποδηλώνεται το σύνολο των δικτύων της ανθρωπότητας τα οποία είναι διασυνδεδεμένα και προσπελάσιμα μέσω των προγραμμάτων πλοήγησης (explorers). Αποτελεί απώτερη εξέλιξη της διάδοσης των υπολογιστών και των μεταξύ τους ανταλλαγών πληροφοριών που οδήγησαν στην δημιουργία δικτύων δεδομένων. Στα τέλη της δεκαετίας του 60 άρχισε να γίνεται φανερό ότι η αξιοποίηση των υπολογιστικών αλλά και των ανθρώπινων πόρων (δηλ. αποδοτικότερη χρήση του πολύτιμου χρόνου των ερευνητών) των Πανεπιστημίων και των ερευνητικών κέντρων των ΗΠΑ δεν μπορούσε να βελτιστοποιηθεί αν έπρεπε να υποχρεώνει τους ερευνητές μα πηγαίνουν κοντά στους υπολογιστές απομακρυνόμενοι από τους εργαστηριακούς χώρους όπου τηρούσαν τα στοιχεία της εργασίας τους. Χρειαζόταν ένα πιο ανθρωποκεντρικό μοντέλο εργασίας όπου τα εργαλεία και η πληροφορία πάνε εκεί που βολεύει τον άνθρωπο και όχι ο άνθρωπος εκεί που βολεύει τα μηχανήματα.

Ξεκίνησε λοιπόν το 1968 από υπηρεσίες του Αμερικάνικου Πρακτορείου Προχωρημένων Ερευνών (ARPA) η προδιαγραφή ενός πειραματικού δικτύου με 4 κόμβους που θα συνέδεε αρχικά τα Πανεπιστήμια της Καλιφόρνια USLA και UCSB, το Πανεπιστήμιο της Utah και το Ίδρυμα Ερευνών Stanford. Την υλοποίηση ανέλαβε η εταιρία Bolt Beranek & Newman και το δίκτυο που ονομάστηκε ARPANET άρχισε να λειτουργεί το επόμενο έτος (1969). Το δίκτυο αυτό έμελλε να αποτελέσει το σπέρμα δημιουργίας ενός από τα πιο πολύπλοκα ανθρώπινα δημιουργήματα που εκτείνεται σε όλη την υδρόγειο και σε κάθε γωνιά όπου υπάρχει ανθρώπινη δραστηριότητα. Σε συνέργεια με τις εκρηκτικές εξελίξεις που ακολούθησαν στο τομέα των υπολογιστών και μικροϋπολογιστών, των οπτικών και ασύρματων επικοινωνιών αλλά και των πολυμέσων (ψηφιακή εικόνα και ήχος) που οδήγησαν σε ραγδαία επέκταση των δικτύων δεδομένων, γιγάντωσαν αυτό το δίκτυο ώστε να λάβει τη μορφή ενός ενιαίου παγκοσμιοποιημένου δικτύου υπό κατανεμημένη διαχείριση χιλιάδων οργανισμών και εταιριών. Αυτό το δίκτυο για το



οποίο γύρω στο 1980 θα αρχίσει να χρησιμοποιείται ο όρος internet, δεν αποτελεί απλά ένα μεγάλο τεχνολογικό βήμα αλλά και όχημα για την ραγδαία εξέλιξη και όλων των άλλων επιστημών, εμπορικών, κοινωνικών, πολιτιστικών αλλά και πολλών άλλων ανθρωπίνων δραστηριοτήτων επιφέροντας πρωτοφανείς βελτιώσεις στην οργάνωση, παραγωγή, διοίκηση και πολλούς άλλους τομείς της κοινωνικής ζωής.

Ο όρος Διαδίκτυο επελέγη για να τονίσει τη δυνατότητά του να επιτρέπει επικοινωνία δια μέσου ανομοιογενών φυσικών δικτύων ανατρέποντας τη μέχρι τότε κατάσταση, όπου, ενώ τα τερματικά επικοινωνούσαν απρόσκοπτα εντός κάθε τύπου δικτύου, δύο τερματικά σε δίκτυα διαφορετικής τεχνικής δεν είχαν συνήθως δυνατότητες επικοινωνίας. Την κατάσταση αυτή ανέτρεψε σταδιακά η εισαγωγή στα τερματικά λογισμικού που υλοποιούσε πρόδρομες μορφές των πρωτοκόλλων του μετέπειτα Διαδικτύου που έδιναν τη δυνατότητα ανταλλαγής δεδομενογραμμμάτων (datagrams). Οι υπολογιστές χρησιμοποιούσαν το πρωτόκολλο 1822 (από τον αριθμό του εγγράφου που το προδιέγραφε) για να επικοινωνούν με τους κόμβους (οι οποίοι τότε ονομάζονταν IMPs (Internet Message Processors)). Οι πρώτες εφαρμογές που ήταν remote login, ftp και αργότερα electronic mail, ήσαν ενσωματωμένες στο πρωτόκολλο Network Control Program (NCP) που μετεξελίχθηκε στο σημερινό TCP/IP (Internet Protocol δηλ. Πρωτόκολλο Διαδικτύου που δίνει το όνομα και στο όλο δίκτυο και στο TCP (Transmission Control Protocol). Το ζευγάρι αυτό των πρωτοκόλλων αποτελεί την καρδιά του Διαδικτύου και το κύριο αντικείμενο της μελέτης μας, αφού η κατανόησή τους ισοδυναμεί με γνώση της λειτουργίας του Διαδικτύου

Πρέπει να τονισθεί ότι τουλάχιστον στα αρχικά στάδια οι κόμβοι (και φυσικά τα τερματικά) του Διαδικτύου δεν βασιζόντουσαν σε ειδικά μηχανήματα αλλά κοινούς υπολογιστές με μόνο νέο στοιχείο το λογισμικό. Δεν υπήρχε ειδικό υλικό για το Διαδίκτυο. Αυτό που απαιτείτο ήταν η απλή προσθήκη του λογισμικού TCP/IP στα τερματικά καθώς και η τοποθέτηση ενός-δύο υπολογιστών σε θέση δρομολογητή ώστε να διασυνδέουν το συγκρότημα (Πανεπιστήμιο ή εταιρεία ή ερευνητικό οργανισμό) προς άλλα φυσικά δίκτυα μέσω μισθωμένων γραμμών που περαιτέρω διασυνδέονταν με άλλα δημιουργώντας ένα ενιαίο πια διαδίκτυο. Σημειωτέον ότι ο υπολογιστής που έμπαινε σε θέση δρομολογητή (π.χ. ένας mini computer PDP-11 της εταιρείας DEC) δεν έτρεχε κάποιο διαφορετικό λογισμικό αλλά το ίδιο πρόγραμμα το οποίο εμπεριείχε και την λειτουργία της δρομολόγησης η οποία ήταν ενσωματωμένη

στο σετ πρωτοκόλλων TCP/IP. Αλλά εκτός από την κάρτα του τοπικού δικτύου που είχαν αναγκαστικά όλοι οι υπολογιστές, ο δρομολογητής είχε και ένα modem ή κάρτα X.25 ή διεπαφή PCM που του επέτρεπε μέσω κάποιου τηλεφωνικού ή μισθωμένου κυκλώματος να συνδέεται προς την αντίστοιχη ομοειδή συσκευή κάποιου άλλου οργανισμού. Αυτός με τη σειρά του συνδεόταν και κάπου αλλού εις τρόπον ώστε να μπορεί ένα δεδομένογραμμα ξεκινώντας από οποιοδήποτε τέτοιο φυσικό υποδίκτυο να καταλήξει σε οποιοδήποτε άλλο. Έτσι το Διαδίκτυο συνέδεε υπάρχοντα φυσικά δίκτυα πάσης τεχνολογίας με την υλοποίηση των πρωτοκόλλων TCP/IP πάνω από ποικιλία υπολογιστών και φυσικών δικτύων. Τα δίκτυα αυτά, (π.χ. SNA, Ethernet, X.25, Token ring, κτλ), με την προσθήκη του λογισμικού που υλοποιούσε το TCP/IP αποκτούσαν κοινή γλώσσα. Η προσφυγή στην υποδομή μετάδοσης του τηλεφωνικού δίκτυο χρησιμοποιήθηκε πολλές φορές επικουρικά για διασύνδεση μέσω modem λόγω της μεγάλης διαδοσής του.

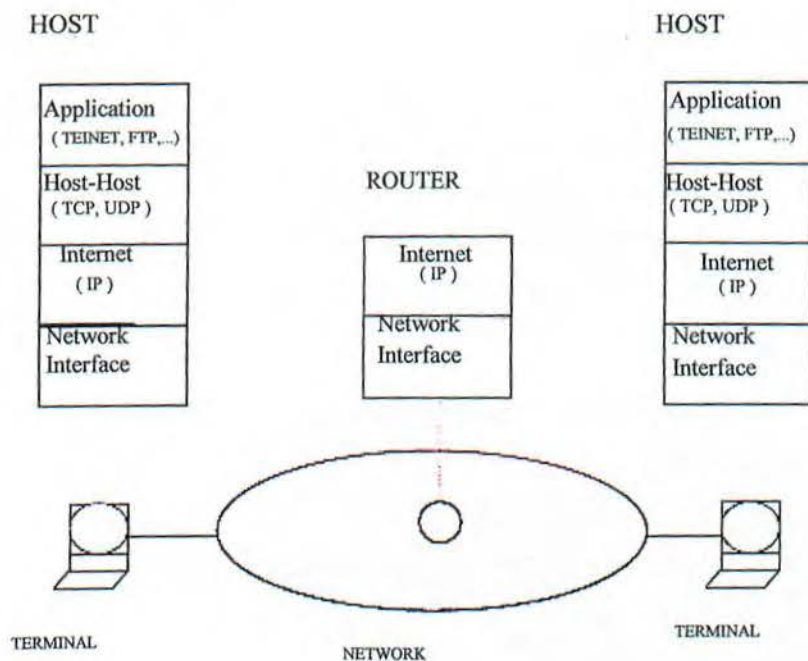
Η εκρηκτική εξέλιξη του Διαδικτύου οδήγησε σε πολλές μεταβολές και σήμερα η υλοποίηση του πρωτοκόλλου IP στους κόμβους (δρομολογητές) δεν βασίζεται πάντα σε λογισμικό. Παρότι μερικοί μικροί δρομολογητές σε μικρές εταιρείες εξακολουθούν να είναι κοινοί υπολογιστές με ειδικό λογισμικό, η διεκπεραίωση εκατομμυρίων δεδομένογραμμάτων το δευτερόλεπτο που απαιτούν οι σημερινές συνθήκες άλλαξε τη σχεδίαση των μεγάλων δρομολογητών οι οποίοι έγιναν ειδικά μηχανήματα με ειδικό υλικό και λογισμικό. Επίσης οι ζεύξεις ανάμεσα στους δρομολογητές βασίζονται σε ειδικές τεχνικές (IPoverSDH, IPoverWDM) και μόνο σε μικρές εταιρείες ή σπίτια βασίζονται σε modem ή PCM. Επίσης δημιουργήθηκαν εταιρείες παροχής υπηρεσιών Διαδικτύου (ISP = Internet Service Providers) που παρέχουν διασύνδεση σε χρήστες (σπίτια και εταιρείες) και άλλες που διασυνδέουν τις πρώτες και ονομάζονται NSP (Network Service Providers). Δηλαδή οι δεύτερες δεν διασυνδέουν τερματικά αλλά δρομολογητές των πρώτων δημιουργώντας ένα δεύτερο ιεραρχικό επίπεδο.

Φυσικά στα τερματικά η υλοποίηση με λογισμικό παρέμεινε καθώς και η συνήθεια να αποκαλούνται οι τερματικοί υπολογιστές που τρέχουν τα πρωτόκολλα του δικτύου (και κυρίως το TCP/IP) ως host computers δηλ. οικοδεσπότες, φιλοξενούντες το λογισμικό του δικτύου (σαν υπολογιστές γενικής χρήσεως που είναι τρέχουν και άλλο λογισμικό). , η καρδιά του Διαδικτύου είναι το Internet Protocol (IP) που καλύπτει το 3ο στρώμα (δικτύου), και το Transmission Control Protocol (TCP) που υλοποιεί το

4ο στρώμα (μεταφοράς). Τα δύο αυτά πρωτόκολλα πρωτοεμφανίσθηκαν το 1974 και έχουν έκτοτε υποστεί πολλές βελτιώσεις. Τα πακέτα (δεδομενογράμματα) TCP/IP πρέπει να ενθυλακωθούν μέσα στα πλαίσια του εκάστοτε φυσικού δικτύου (π.χ. LAN, X.25, SNA, PCM ή απλά RS-232 σε modem πάνω από μισθωμένες ή και επιλεγόμενες τηλεφωνικές γραμμές κτλ.) για να διαβιβαστούν στο άλλο άκρο του αφού αλλιώς το φυσικό δίκτυο δεν θα ήξερε πώς να τα χειριστεί. Το πακέτο εκθυλακώνεται προτού παραδοθεί στο στρώμα IP στον ξενιστή ή στον δρομολογητή.

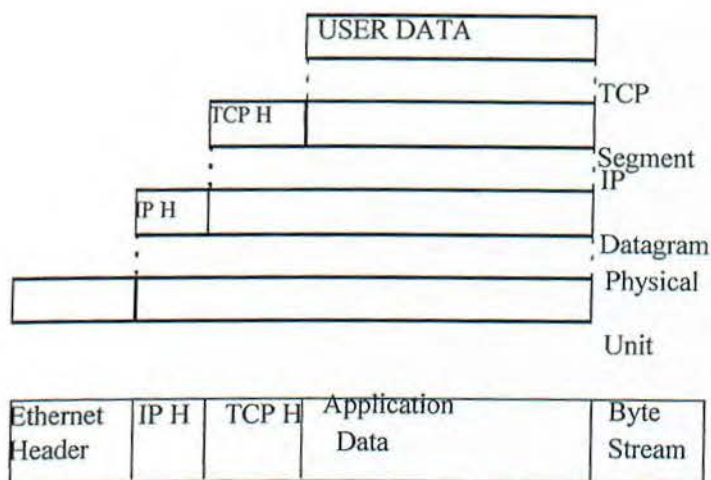
Το Διαδίκτυο δεν ακολουθούσε φυσικά την αρχιτεκτονική OSI αφού προηγήθηκε αυτής αλλά μια δική του που φαίνεται στο σχήμα. Στο ανώτερο επίπεδο βρίσκονται όπως πάντα οι εφαρμογές π.χ. FTP, Telnet, SMTP κτλ. (Αντιστοιχεί χονδρικά στα 3 ανώτερα στρώματα του OSI). Ακολουθεί το επίπεδο πρωτοκόλλων μεταξύ ξενιστών (host-to-host) δηλαδή απ' άκρου εις άκρο στο οποίο εδρεύουν οι λειτουργίες ελέγχου ροής και λαθών δηλαδή αυτό που αντιστοιχεί στις λειτουργίες του 4 στρώματος OSI. Εδώ χρησιμοποιείται το TCP όταν οι απαιτήσεις αξιοπιστίας είναι υψηλές και το UDP όταν ένα πιο ελαφρύ πρωτόκολλο χωρίς επιβεβαίωση είναι πιο κατάλληλο. Ακολουθεί το στρώμα δικτύου που εδώ είναι το στρώμα IP και εκτελεί τη παράδοση δεδομενογραμμάτων άνευ εγγυήσεων.

Στο κατώτερο επίπεδο (αντίστοιχο του φυσικού στρώματος και του στρώματος ζεύξης) βρίσκεται η πρόσβαση στο δίκτυο στην οποία ανήκουν η κάρτα δικτύου και το λογισμικό οδήγησης (driver) που εξαρτώνται από το εκάστοτε φυσικό δίκτυο που μπορεί να είναι όπως αναφέρθη μία απλή σειριακή πόρτα που οδηγεί σε ένα modem ή κάρτα τοπικού δικτύου ή X.25 κτλ. Το Διαδίκτυο δεν πρόσθεσε κάτι ιδιαίτερο σε αυτό το επίπεδο, απλά χρησιμοποίησε τα υπάρχοντα στοιχεία πατώντας σε αυτά. Η ικανότητα του Διαδικτύου να συνεργάζεται με κάθε φυσικό δίκτυο είναι και το μυστικό της τεράστιας επιτυχίας του. Είναι στην ουσία μία τεχνολογία διασύνδεσης δικτύων που δεν φιλοδοξεί να προδιαγράψει τα κατώτερα στρώματα και τα φυσικά μέσα αφήνοντάς τα να εξελίσσονται απρόσκοπτα. Η υλοποίηση μέσω λογισμικού παρέχει πλήρη ελευθερία στην επιλογή φόρμας πακέτων, διευθύνσεων, τεχνικών μετάδοσης κτλ. καθώς και τη δυνατότητα εύκολης τροποποίησης που οδηγεί σε νέες βερσιόν, δίνοντας και τη χαρακτηριστική ευελιξία που διακρίνει το Διαδίκτυο.



Σχήμα.1 Δομή διαδικτύου

Καθώς το Διαδίκτυο δεν μπορεί να νοηθεί χωρίς τα υποκείμενα φυσικά δίκτυα, είναι κεντρικής σημασίας ο μηχανισμός ενθυλάκωσης μέσα στις φόρμες των μονάδων των φυσικών δικτύων. Στο σχήμα φαίνονται αφ' ενός οι σχέσεις των μονάδων πρωτοκόλλων του Διαδικτύου μεταξύ τους και ο τρόπος ενθυλάκωσης εκάστης σε αυτήν του υποκείμενου στρώματος καθώς και την τελική ενθυλάκωση στο φυσικό δίκτυο.



Σχήμα 2. Τρόπος ενθυλάκωσης πρωτοκόλλων

## 1.2 ΤΟ ΠΡΩΤΟΚΟΛΛΟ IP (Internet Protocol)

Το πρωτόκολλο IP είναι πλέον το πιο διαδεδομένο πρωτόκολλο του στρώματος δικτύου και αποτελεί το κύριο πρωτόκολλο πάνω στο οποίο είναι βασισμένο το Διαδίκτυο. Αποτελεί το κύριο πρωτόκολλο επικοινωνίας για τη μετάδοση αυτοδύναμων πακέτων (datagrams), δηλ. πακέτων δεδομένων και είναι υπεύθυνο για τη δρομολόγηση των πακέτων δεδομένων ανάμεσα στα διάφορα δίκτυα.

Καθορίζει τη μορφή των πακέτων που στέλνονται μέσω ενός διαδικτύου, καθώς και τους μηχανισμούς που χρησιμοποιούνται για την προώθηση των πακέτων από έναν υπολογιστή προς έναν τελικό προορισμό μέσω ενός ή περισσότερων δρομολογητών.

Επειδή αυτό το πρωτόκολλο χαρακτηρίζεται από τον μη εγγυημένο τρόπο προώθησης πακέτων (γνωστός και ως «προώθηση βέλτιστης προσπάθειας») υιοθετήθηκε λοιπόν μία λύση άνευ συνδέσεων βασισμένη σε δεδομενογράμματα (datagrams) μεγέθους έως 64K οκτέτων τα οποία προωθούνται από δρομολογητή σε δρομολογητή χωρίς παρακολούθηση της πορείας τους αλλά ούτε και της τύχης τους μέσα στο δίκτυο και τα οποία στην πρώτη δυσκολία απορρίπτονται χωρίς πολλές διαδικασίες.

Η θεμελιώδης υπηρεσία του πρωτοκόλλου διαδικτύου IP είναι λοιπόν μία υπηρεσία μεταφοράς πακέτων που χαρακτηρίζεται σαν μη αξιόπιστη που στερείται των κατάλληλων μηχανισμών που θα εξασφάλιζαν εγγυημένη παράδοση των πακέτων. Αυτή η υπηρεσία είναι γνωστή και ως best-effort delivery ) γιατί το λογισμικό του Διαδικτύου καταβάλει προσπάθεια να παραδώσει τα πακέτα στον προορισμό τους χωρίς όμως και να έχει εξασφαλίσει τις προϋποθέσεις εκείνες που θα καθιστούσαν αυτή την προσπάθεια αρκετή για εγγυημένο αποτέλεσμα..Ο κύριος λόγος για την έλλειψη αξιοπιστίας είναι για να μειωθεί η πολυπλοκότητα των routers. Έτσι μπορούν να διαχειριστούν όπως θέλουν αυτοί τα πακέτα που τους έρχονται.

Το πρωτόκολλο IP χρησιμοποιεί την τεχνολογία μεταγωγής πακέτων για τη μεταφορά των δεδομένων από έναν υπολογιστή προς έναν τελικό προορισμό. Τα δεδομένα κόβονται σε κομμάτια που ονομάζονται πακέτα. Έτσι κάθε πακέτο IP, αποτελείται από μια κεφαλίδα και στη συνέχεια ακολουθούν τα δεδομένα. Στη κεφαλίδα αυτή εμπεριέχονται πληροφορίες: πρώτον, για τα δεδομένα που εμπεριέχονται στο πακέτο και δεύτερον, οι διευθύνσεις αφετηρίας και προορισμού. Η φόρμα ενός τέτοιου πακέτου ή αλλιώς δεδομενογράμματος είναι:

4	8	16	24	31
HLEN	TYPE OF SERVICE	TOTAL LENGTH		
IDENTIFICATION		D F	M F	FRAGMENT OFFSET
LIVE	PROTOCOL	HEADER CHECKSUM		
SOURCE IP ADDRESS				
DESTINATION IP ADDRESS				
IP OPTIONS (IF ANY)				PADDING
DATA				
...				

Σχήμα 3. Η δομή ενός πακέτου IP

## VERS

Το πεδίο αυτό τω 4 bits περιέχει την έκδοση του IP πρωτοκόλλου που χρησιμοποιείται για να δημιουργηθεί το δεδομένογραμμα. Συνηθίζεται ο αποστολέας, ο λήπτης και ο δρομολογητής να συμφωνούν μεταξύ τους στη μορφή των δεδομένογραμμάτων.

## HLEN

Αυτό το πεδίο αποτελείται από 4 bits και προσδιορίζει το μήκος της επικεφαλίδας του δεδομένογράφματος.

## TOTAL LENGTH

Περιέχει το μήκος του IP δεδομένογράφματος περιλαμβάνοντας την επικεφαλίδα και το πεδίο της πληροφορίας.

## FRAGMENT OFFSET

Με αυτό το πεδίο το IP περιγράφει τη διάσταση των τεμαχίων και το μέγεθος τεμαχισμού επιλέγεται να είναι το μεγαλύτερο πολλαπλάσιο του 8. Το fragment offset μαζί με τα πεδία FLAGS και IDENTIFICATION ελέγχουν την κατάτμηση και τη συναρμολόγηση των δεδομένογραμμάτων. Επίσης το πεδίο IDENTIFICATION περιέχει ένα μοναδικό αριθμό που αποτελεί την ταυτότητα του δεδομένογράφματος. Με αυτόν τον τρόπο όταν κόβεται ένα δεδομένογραμμα σε μικρότερα τεμάχια, σε κάθε τεμάχιο υπάρχει το ίδιο πεδίο identification για να μπορεί ο λήπτης να γνωρίζει σε ποια μηχανή συναρμολόγησης να οδηγήσει το εν λόγω κομμάτι, δεδομένου ότι τα τεμάχια δεν φθάνουν πάντα με σειρά.

## FLAGS

Το πεδίο FLAGS αποτελείται από 3 bits αλλά το πρώτο δεν χρησιμοποιείται. Τα 2 χαμηλότερης τάξης bits ελέγχουν την κατάτμηση. Το πρώτο bit χρησιμοποιείται για

να απαγορεύσει την κατάτμηση για λόγους ελέγχου και ονομάζεται DF δηλ. “do not fragment”. Όταν έχει τιμή 1 δηλώνει ότι το δεδομένογραμμα δεν πρέπει να κατατμηθεί.

Το τελευταίο bit MF του πεδίου FLAGS δηλώνει αν το τεμάχιο περιλαμβάνει δεδομένα από τη μέση ή από το τέλος του αρχικού δεδομένογραμματος. καλείται “more fragments” bit. Όταν ο προορισμός λάβει ένα τεμάχιο με το bit MF ίσο με μηδέν, γνωρίζει ότι αυτό το bit είναι το τελευταίο του αρχικού δεδομένογραμματος.

### **TIME TO LIVE**

Το πεδίο TIME TO LIVE (χρόνος ζωής) καθορίζει πόσο χρόνο σε δευτερόλεπτα το δεδομένογραμμα επιτρέπεται να παραμείνει στο Διαδίκτυο. Όταν ένας υπολογιστής ορίζει σε αυτό το πεδίο ένα μέγιστο χρόνο ζωής του δεδομένογραμματος, οι πύλες και ξενιστές που επεξεργάζονται το δεδομένογραμμα ελαττώνουν αυτό το χρόνο και εάν φθάσει η τιμή αυτού του πεδίου στο 0, το δεδομένογραμμα απορρίπτεται.

### **PROTOCOL**

Η τιμή του πεδίου αυτού καθορίζει ποιο υψηλού επιπέδου πρωτόκολλο χρησιμοποιήθηκε για να δημιουργήσει το μήνυμα που μεταφέρεται στο πεδίο δεδομένων του δεδομένογραμματος. Στην ουσία η τιμή του πεδίου PROTOCOL περιγράφει τη φόρμα των δεδομένων.

### **HEADER CHECKSUM**

Το πεδίο HEADER CHECKSUM διασφαλίζει την ακεραιότητα των τιμών της επικεφαλίδας. Το CHECKSUM του IP προκύπτει από την επεξεργασία της επικεφαλίδας σαν ακολουθία ακεραίων μήκους 16-bits οι οποίοι προστίθενται χρησιμοποιώντας αριθμητική συμπλήρωματος του ένα και παίρνοντας το συμπλήρωμα του ένα του προκύπτοντος αθροίσματος.



## SOURCE IP ADDRESS – DESTINATION IP ADDRESS

Περιέχουν τις IP διευθύνσεις των 32-bit του αποστολέα και του λήπτη των δεδομενογραμμάτων δηλαδή είναι τα κύρια πεδία βάσει των οποίων γίνεται η δρομολόγηση.

## PADDING

Απαρτίζεται από τόσα bits που περιέχουν 0 όσα χρειάζονται για να φέρουν το μήκος της επικεφαλίδας του δεδομενογράμματος σε ακριβές πολλαπλάσιο των 32 bits αφού το μήκος της επικεφαλίδας προδιαγράφεται σε μονάδες λέξεων μήκους 32 bits.

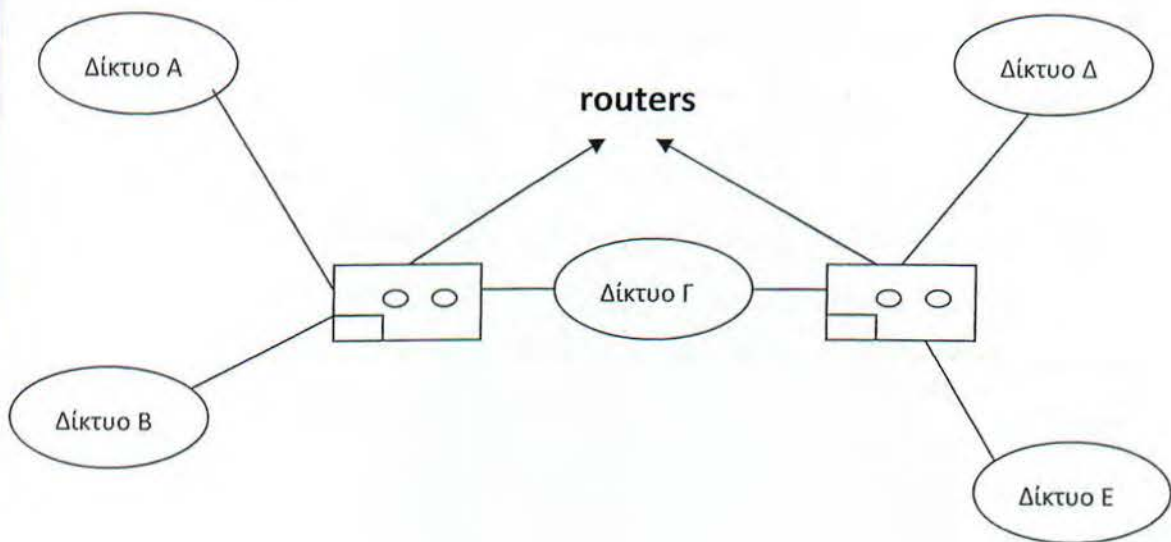
## DATA

Δείχνει την αρχή του πεδίου των δεδομένων του δεδομενογράμματος.

### 1.2.1 ΔΡΟΜΟΛΟΓΗΣΗ ΠΑΚΕΤΩΝ

Το πρωτόκολλο IP είναι υπεύθυνο για το πέρασμα ενός πακέτου δεδομένων από υπολογιστή σε υπολογιστή. Όλα τα δίκτυα που συνδέονται στο Internet “καταλαβαίνουν” τη γλώσσα IP κι έτσι μπορούν να συνεννοούνται και να ανταλλάσσουν δεδομένα με ομοιόμορφο τρόπο. Σημειώνουμε ότι σε κάθε υπολογιστή του Internet αντιστοιχίζεται μία διεύθυνση που ονομάζεται διεύθυνση IP.

Τα δίκτυα του Internet συνδέονται μεταξύ τους με ειδικούς υπολογιστές που ονομάζονται δρομολογητές (routers) ή πύλες (gateways). Ένας router είναι λοιπόν ένας υπολογιστής που συνδέει δύο ή περισσότερα δίκτυα (που μπορεί να είναι διαφορετικού τύπου) και έτσι ανήκει σε δύο ή περισσότερα δίκτυα ταυτόχρονα. Για τη δρομολόγηση πακέτων οι routers χρησιμοποιούν τις διευθύνσεις IP του υπολογιστή-αποστολέα και του υπολογιστή-παραλήπτη.



Σχήμα 4. Σύνδεση και λειτουργία του router

Η δουλειά των routers είναι να δρομολογούν τα πακέτα των δεδομένων μέσα από τα διάφορα δίκτυα που αποτελούν το Internet μέχρις ότου τα επιδώσουν στον προορισμό τους. Ας θεωρήσουμε ότι ένας υπολογιστής που βρίσκεται κάπου στο Internet θέλει να στείλει δεδομένα σε κάποιον άλλον υπολογιστή. Τα δεδομένα κόβονται σε πακέτα και το IP που εκτελείται στον υπολογιστή - αποστολέα ετοιμάζεται να στείλει το κάθε πακέτο. Εισάγει λοιπόν στην επικεφαλίδα του πακέτου τις IP διευθύνσεις του αποστολέα και του παραλήπτη και κατόπιν, βάσει των διευθύνσεων αυτών, ελέγχει αν ο παραλήπτης βρίσκεται στο ίδιο δίκτυο με τον αποστολέα. Εάν ναι, το πακέτο στέλνεται κατευθείαν στον παραλήπτη χωρίς να χρειαστεί να διαβεί τα όρια του δικτύου. Εάν όχι, προωθείται στον router που είναι συνδεδεμένος με το δίκτυο. Ο router με τη σειρά του ελέγχει αν ο παραλήπτης βρίσκεται σε κάποιο από τα υπόλοιπα δίκτυα με τα οποία είναι συνδεδεμένος. Εάν ναι, το πακέτο στέλνεται κατευθείαν στον παραλήπτη στο δίκτυο αυτό. Εάν όχι, το πακέτο προωθείται στον επόμενο router, κ.ο.κ. μέχρις ότου το πακέτο προωθηθεί τελικά στον router που είναι συνδεδεμένος στο ίδιο δίκτυο με τον παραλήπτη. Το πακέτο μπορεί έτσι να περάσει

από πολλούς routers μέχρις ότου φτάσει στον προορισμό του. Οι routers διατηρούν πίνακες που προσδιορίζουν την κατεύθυνση που πρέπει να πάρει ένα πακέτο προκειμένου να φτάσει στον προορισμό του. Βάσει αυτών των πινάκων αποφασίζουν ποιος θα είναι ο επόμενος router στον οποίο θα πρέπει να προωθήσουν το πακέτο. Κάθε φορά, το πακέτο μετακινείται όλο και πιο κοντά προς τον προορισμό του έως ότου τελικά τον φτάσει.

Ένα μεγάλο πλεονέκτημα αυτής της μεθόδου είναι ότι η διαδρομή που ακολουθεί ένα πακέτο δεν είναι προκαθορισμένη, αλλά επιλέγεται δυναμικά. Έτσι, οι routers μπορούν να επιλέγουν εναλλακτικούς δρόμους για ένα πακέτο σε περίπτωση που μια συγκεκριμένη σύνδεση του δικτύου παρουσιάζει πρόβλημα και βρίσκεται προσωρινά σε αχρηστία.

### 1.3 ΤΟ ΠΡΩΤΟΚΟΛΛΟ TCP

Όπως είδαμε, το πρωτόκολλο IP του 3ου στρώματος παρέχει υπηρεσίες αποστολής πακέτων χωρίς όμως να εξασφαλίζει την αξιοπιστία της ροής των δεδομένων.

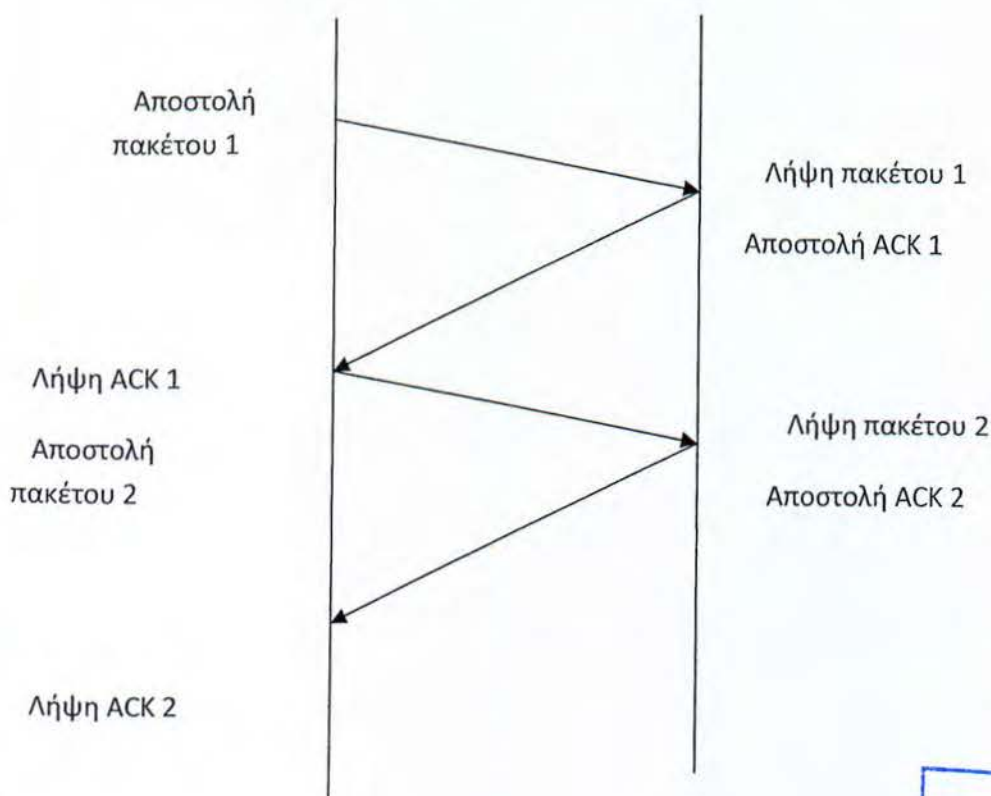
Τα πακέτα μπορούν να καταστραφούν από πολλές αιτίες π.χ. μπορεί να αλλοιωθούν αρκετά ψηφία στα δεδομένα από θορύβους, ή να χαθούν δεδομένα όταν το δίκτυο δεν μπορεί να αντεπεξέλθει στο φορτίο που υπάρχει τη δεδομένη στιγμή. Τα δίκτυα που δρομολογούν δυναμικά τα πακέτα, μπορεί να τα αποστείλουν στο δέκτη με διαφορετική σειρά, από αυτή που τα έστειλε ο πομπός, ή να τα στείλουν με κάποια καθυστέρηση, ή ακόμη να στείλουν μερικά πακέτα δυο φορές.

Είναι λοιπόν αναγκαίες κάποιες πρόσθετες λειτουργίες για την πρόσδοση αξιοπιστίας όπως π.χ. ανίχνευση και διόρθωση λαθών με επαναποστολή λανθασμένων και μη αφιχθέντων πακέτων, συναρμολόγηση κ.α. Τη λειτουργία αυτή αναλαμβάνει το ειδικό πρωτόκολλο μεταφοράς του δικτύου. Το πρωτόκολλο αυτό στη περίπτωση του Διαδικτύου είναι το Transmission Control Protocol (TCP).

Το TCP προσφέρει ένα αξιόπιστο πρωτόκολλο πάνω από το IP. Εγγυάται ότι τα πακέτα θα παραδοθούν στον προορισμό τους, ότι θα φτάσουν με τη σειρά με την οποία στάλθηκαν και ότι τα περιεχόμενα των πακέτων θα φτάσουν αναλλοίωτα (δηλ. όπως στάλθηκαν). Αυτή η διαδικασία επιτυγχάνεται ως εξής:

Όπως τα περισσότερα πρωτόκολλα έτσι και το πρωτόκολλο TCP χρησιμοποιεί μια απλή αλλά βασική ιδέα που είναι γνωστή σαν positive acknowledgement with retransmission (επιβεβαίωση λήψης με αναμετάδοση).

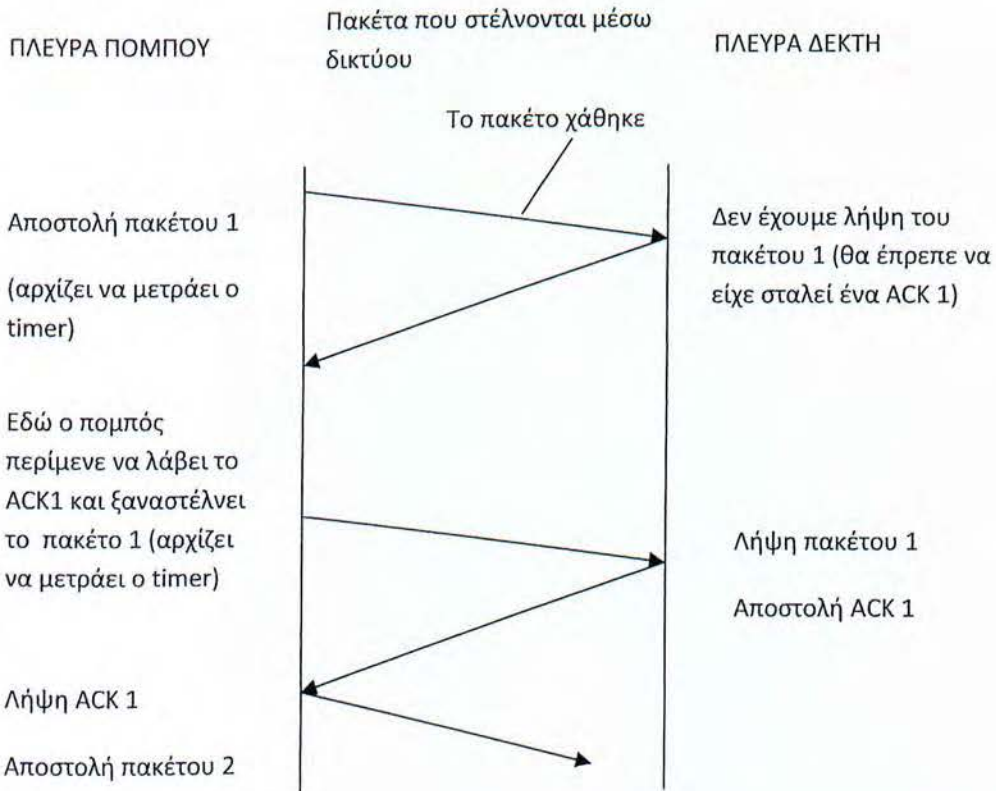
Η τεχνική αυτή απαιτεί να επικοινωνεί ο δέκτης με την πηγή και να στέλνει σ' αυτήν ένα μήνυμα επιβεβαίωσης (acknowledgement) καθώς λαμβάνει τα δεδομένα. Ο πομπός κρατάει ένα αντίγραφο του πακέτου που έστειλε και περιμένει το μήνυμα επιβεβαίωσης από το δέκτη για να στείλει το επόμενο. Επίσης μόλις στείλει ένα πακέτο θέτει σε λειτουργία έναν χρονιστή (timer) και εάν μέσα σε κάποιο προσδιορισμένο χρονικό διάστημα δεν πάρει μήνυμα επιβεβαίωσης, ξαναστέλνει το πακέτο.



ΒΙΒΛΙΟΘΗΚΗ  
ΤΕΙ ΠΕΙΡΑΙΑ

Σχήμα 5. Τεχνική πρωτοκόλλου TCP επιβεβαίωση λήψης και αναμετάδοσης

Στη περίπτωση που παρουσιαστεί κάποιο πρόβλημα στο δίκτυο με αποτέλεσμα να χαθεί κάποιο από τα πακέτα και ο πομπός δε λάβει μήνυμα επιβεβαίωσης μέσα στο χρονικό διάστημα τότε ο δέκτης ξαναζητάει την αποστολή του πακέτου και ο αποστολέας είναι υπεύθυνος για την αναμετάδοση του.



Σχήμα 6. Ανταλλαγή δεδομένων μεταξύ δέκτη και πομπού όταν χάνεται το πακέτο

Τα πακέτα του πρωτοκόλλου TCP καλούνται segments (τομείς). Ένα από τα κυριότερα μέρη ενός segment είναι η TCP επικεφαλίδα (TCP header), η οποία παρέχει συγκεκριμένες πληροφορίες για το πρωτόκολλο TCP. Η φόρμα του TCP φαίνεται παρακάτω

4	10	16	24	31
SOURCE PORT			DESTINATION PORT	
SEQUENCE NUMBER				
ACKNOWLEDGEMENT NUMBER				
LEN	RESERVED	CODE BITS	WINDOW ADVERTISEMENT	
CHECKSUM			URGENT POINTER	
OPTIONS (IF ANY)				PADDING
ΔΕΔΟΜΕΝΑ (DATA)				
...				

Σχήμα 7. Δομή ενός πακέτου TCP

Κάθε τμήμα χωρίζεται σε δύο μέρη, την επικεφαλίδα (header) και τα δεδομένα (data). Η επικεφαλίδα (TCP Header) μεταφέρει πληροφορίες ελέγχου (control) και αναγνώρισης (identification).

### Source Port – Destination Port

Αυτά τα δύο πεδία περιέχουν τους δύο αριθμούς TCP, του αποστολέα και του παραλήπτη αντίστοιχα και χρησιμοποιούνται στα σημεία τερματισμού για να προσδιορίσουν μια σύνδεση.

### **Sequence Number**

Προσδιορίζει τη θέση των δεδομένων του τμήματος, στην ακολουθία των bytes των δεδομένων του πομπού. Το TCP θεωρεί τα δεδομένα που ανταλλάσσουν δύο εφαρμογές σαν μια συνεχή ροή από οκτάδες.

### **Acknowledge Number**

Περιέχει τον αριθμό των byte που περιμένει να του στείλει η άλλη πλευρά. Το πεδίο αυτό αναφέρεται στα δεδομένα που στέλνει ο δέκτης σε αντίθεση με το Sequence Number που αναφέρεται στα δεδομένα που στέλνει ο πομπός.

### **HLEN**

Καθορίζει το μέγεθος της επικεφαλίδας (πολλαπλάσιο του 32) και επομένως δείχνει και την αρχή των δεδομένων.

### **Reserved**

Το πεδίο αυτό περιέχει 6 bits που έχουν δεσμευτεί για μελλοντική χρήση.

### **Checksum**

Το πεδίο checksum μεγέθους 16 bit χρησιμοποιείται για έλεγχο λαθών στην επικεφαλίδα και στα δεδομένα.

### **Code Bits**

Χρησιμοποιείται από το TCP για να αναγνωρίσει το είδος του τμήματος. Έχει 6 bits και η σημασία τους είναι η ακόλουθη.

### **Window**

Ο αριθμός από octets δεδομένων (bytes) που επιθυμεί να δεχτεί ο αποστολέας του πακέτου, αρχίζοντας από εκείνη που δείχνει το πεδίο επιβεβαίωσης (acknowledgment field).

### **Padding**

Περιέχει μηδενικά bits και μπαίνουν τόσα κάθε φορά όσα απαιτούνται για να είναι το συνολικό μήκος της επικεφαλίδας πολλαπλάσιο των 32 bits.

## Options

Σε αυτό το πεδίο περιγράφονται οι προδιαγραφές που έχουν συμφωνήσει τα πρωτόκολλα TCP των δύο άκρων για την σύνδεση. Για παράδειγμα μία πολύ σημαντική λειτουργία είναι ο καθορισμός του μέγιστου τμήματος (segment) που μπορεί να μεταφερθεί με αποτέλεσμα υπολογιστές που διαθέτουν περιορισμένο ελεύθερο χώρο για την προσωρινή αποθήκευση των bytes, να μπορούν να επεξεργάζονται τα segment που στέλνει ο πομπός.

## Urgent pointer

Εάν είναι ενεργοποιημένο το URG bit ελέγχου, τότε αυτό το πεδίο δείχνει τον αριθμό ακολουθίας (sequence number) της octet που βρίσκεται αμέσως μετά το τελευταίο byte από τα επείγοντα δεδομένα. Έτσι παρουσιάζει τη θέση του τελευταίου byte με επείγοντα δεδομένα.

## 1.4 ΤΟ ΠΡΟΤΥΠΟ OSI

Είναι γνωστό ότι για τη μελέτη και τη σχεδίαση δικτύων απαιτούνταν ένας μεγάλος αριθμός ειδικοτήτων (ηλεκτρονικών, προγραμματιστών, μελετητών κίνησης). Για την αντιμετώπιση λοιπόν αυτής της πολυπλοκότητας δημιουργήθηκε μια αρχιτεκτονική οργανωμένη σε μια στοίβα από στρώματα ή επίπεδα (layers ή levels), που το καθένα κτίζεται πάνω στο προηγούμενό του. Σκοπός κάθε επιπέδου σε όλα τα δίκτυα είναι να προσφέρει συγκεκριμένες υπηρεσίες στα υψηλότερα επίπεδα ακολουθώντας την αρχή απόκρυψης πληροφορίας δηλαδή αποκρύπτοντας από τα επίπεδα αυτά τις λεπτομέρειες σχετικά με το πώς υλοποιούνται οι παρεχόμενες υπηρεσίες.

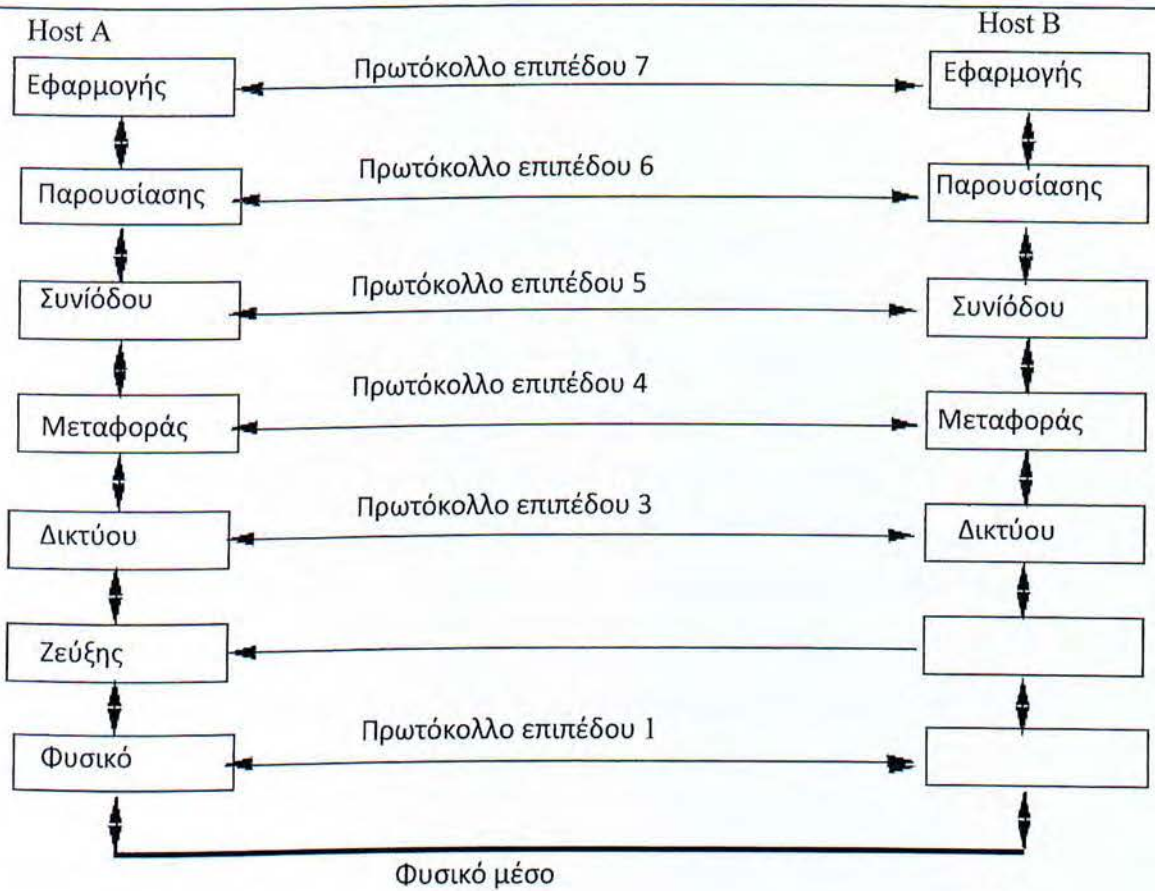
Η μεθοδολογία αυτή ονομάστηκε πρότυπο OSI και προτάθηκε από την ISO(International Standards Organization).

Το πρότυπο αυτό καθορίζει την αρχιτεκτονική στρωμάτωσης (layered architecture) με στόχο τη δημιουργία «ανοικτών συστημάτων» που επιτρέπουν την σύνδεση μέσω ενός επικοινωνιακού δικτύου υπολογιστών διαφορετικών κατασκευαστών και με διαφορετικά λειτουργικά συστήματα. Ο όρος ανοικτό σύστημα σημαίνει ανοικτό σε



κάθε κατασκευαστή που θα σχεδιάσει σύμφωνα με το πρότυπο OSI μέσω της τήρησης κοινών προδιαγραφών ανεξαρτήτως των διαφορών της εσωτερικής υλοποίησης. Ανάμεσα σε κάθε ζεύγος γειτονικών επιπέδων υπάρχει μια διεπαφή (interface) η οποία καθορίζει ποιες πρωτογενείς λειτουργίες και υπηρεσίες προσφέρει το ένα επίπεδο στο άλλο που βρίσκεται από πάνω του. Ένα από τα πιο σημαντικά θέματα είναι ο καθορισμός ξεκάθαρων διεπαφών ανάμεσα στα επίπεδα καθώς και ο καθορισμός ενός συγκεκριμένου συνόλου καλά κατανοητών λειτουργιών που απαιτείται να εκτελεί το κάθε επίπεδο. Επίσης στο κάθε επίπεδο αντιστοιχεί και ένα πρωτόκολλο. Το σύνολο των επιπέδων και πρωτοκόλλων ονομάζεται αρχιτεκτονική δικτύου (network architecture). Οι προδιαγραφές της αρχιτεκτονικής θα πρέπει να περιέχουν αρκετές πληροφορίες, ώστε να επιτρέπουν σ' έναν κατασκευαστή να γράψει το πρόγραμμα ή να κατασκευάσει το υλικό κάθε επιπέδου, έτσι ώστε αυτό να υπακούει σωστά στο κατάλληλο πρωτόκολλο.

Σχήμα 8. Τα επτά στρώματα σύμφωνα με το πρότυπο OSI



### Στρώμα εφαρμογής

Το επίπεδο εφαρμογής παρέχει στον χρήστη έναν τρόπο να προσπελάσει μέσω μιας εφαρμογής τις πληροφορίες ενός δικτύου. Αυτό το επίπεδο είναι η κύρια διασύνδεση του χρήστη με την εφαρμογή και συνεπώς, με το δίκτυο. Το στρώμα εφαρμογής παρέχει ένα πλαίσιο για την οργάνωση των εφαρμογών που ορίζουν το περιβάλλον του χρήστη. Το στρώμα εφαρμογής παρέχει υπηρεσίες κατευθείαν στον χρήστη και όχι σε άλλο στρώμα. Παραδείγματα πρωτοκόλλων επιπέδου εφαρμογών αποτελούν τα Telnet, FTP, SMTP και http.

### Στρώμα παρουσίασης

Το επίπεδο παρουσίασης μετασχηματίζει τα δεδομένα σε τυπική μορφή που την αναμένει το επίπεδο εφαρμογών. Στο επίπεδο αυτό τα δεδομένα υφίστανται κρυπτογράφηση, συμπίεση, κωδικοποίηση και όποια άλλη διαμόρφωση απαιτεί η μορφή δεδομένων ή ο σχεδιαστής του πρωτοκόλλου. Παραδείγματα αποτελούν η μετατροπή αρχείων από κώδικα EBCDIC σε κώδικα ASCII.

### Στρώμα συνόδου

Το επίπεδο συνόδου ελέγχει τις συνόδους (δηλαδή τις ανταλλαγές δεδομένων) μεταξύ δύο υπολογιστών, του Α και του Β. Ξεκινά, διαχειρίζεται και τερματίζει τη σύνδεση μεταξύ μιας τοπικής και μιας απομακρυσμένης εφαρμογής. Αντιμετωπίζει λειτουργίες FDX (full duplex, οι Α και Β μιλούν ταυτόχρονα από δύο κανάλια) ή HDX (half-duplex, μιλάει ο Α και μετά απαντάει ο Β από το ένα διαθέσιμο κανάλι), ενώ υποστηρίζει διαδικασίες αποθήκευσης κατάστασης (checkpoint), αναβολής (adjournment), τερματισμού (termination) και επανεκκίνησης (restart). Αυτό το επίπεδο είναι επίσης υπεύθυνο για την ανάκαμψη από διακοπή της επικοινωνίας με επανάληψη της εκπομπής από το τελευταίο σημείο ελέγχου που είχε ολοκληρωμένη μετάδοση.

### Στρώμα μεταφοράς

Το στρώμα μεταφοράς έχει σαν ρόλο την παροχή αξιόπιστης ανταλλαγής των δεδομένων μεταξύ των τερματικών συσκευών (υπολογιστών). Εξασφαλίζει ότι τα

δέκτης το αντιλαμβάνεται σαν 1 και όταν στέλνει 0 και ο δέκτης το λαμβάνει σαν 0. Αυτό είναι πολύ πιο δύσκολο απ' ό τι φαίνεται εκ πρώτης όψεως όταν οι αποστάσεις είναι μεγάλες, οι ρυθμοί μετάδοσης υψηλοί, και οι θόρυβοι έντονοι. Πομπός και δέκτης πρέπει να έχουν συμφωνήσει στο σωστό πλάτος του σήματος, διάρκεια και μορφή του παλμού καθώς και πολλές άλλες λεπτομέρειες. Συσκευές φυσικού επιπέδου είναι οι διανεμητές, οι επαναλήπτες (repeaters), οι κάρτες δικτύου.

Τυπικό παράδειγμα προτύπου του φυσικού στρώματος είναι το RS-232 που χρησιμοποιείται για την σύνδεση περιφερειακών σε υπολογιστές.

## 1.5 ΤΟ ΠΡΩΤΟΚΟΛΛΟ HTTP

### Web page

Μία Ιστοσελίδα (Web page) αποτελείται από αντικείμενα. Με τον όρο αντικείμενο (object) εννοούμε ένα απλό αρχείο, όπως ένα αρχείο HTML, ένα αρχείο εικόνας ή ένα αρχείο βίντεο, το οποίο μπορεί να προσπελαστεί μέσω ενός URL (μια διεύθυνση). Οι περισσότερες Ιστοσελίδες αποτελούνται από ένα βασικό αρχείο HTML και διάφορα σχετικά αντικείμενα.

### Web browser

Ένας Web browser (φυλλομετρητής ιστοσελίδων, πλοηγός Web, πρόγραμμα περιήγησης Web ή περιηγητής Ιστού) είναι ένα λογισμικό που επιτρέπει στον χρήστη του να προβάλλει, και να αλληλεπιδρά με, κείμενα, εικόνες, βίντεο, μουσική, παιχνίδια και άλλες πληροφορίες συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Το κείμενο και οι εικόνες σε μια ιστοσελίδα μπορεί να περιέχουν υπερσυνδέσμους προς άλλες ιστοσελίδες του ίδιου ή διαφορετικού ιστότοπου. Ο Web browser επιτρέπει στον χρήστη την γρήγορη και εύκολη πρόσβαση σε πληροφορίες που βρίσκονται σε διάφορες ιστοσελίδες. Οι φυλλομετρητές χρησιμοποιούν τη γλώσσα μορφοποίησης HTML για την προβολή των ιστοσελίδων, για αυτό η εμφάνιση μιας ιστοσελίδας μπορεί να διαφέρει ανάλογα με τον browser. Επίσης, στους browser υλοποιείτε και η πλευρά του πελάτη του πρωτοκόλλου HTTP.

## Web server

Εξυπηρετητής ή διακομιστής (server) είναι υλικό ή λογισμικό που αναλαμβάνει την παροχή διάφορων υπηρεσιών, εξυπηρετώντας αιτήσεις από άλλους υπολογιστές γνωστούς ως πελάτες (clients). Παρομοίως, ως πελάτη μπορούμε να θεωρήσουμε είτε κάποιο λογισμικό που επικοινωνεί και υποβάλει αιτήματα στον εξυπηρετητή, είτε τον υπολογιστή στον οποίο εκτελείται το λογισμικό πελάτη.

Η επικοινωνία μεταξύ πελάτη και εξυπηρετητή γίνεται μέσω ενός τοπικού δικτύου, ή ακόμα και μέσω του Διαδικτύου. Ο εξυπηρετητής είναι συνήθως ένας υπολογιστής που διαφέρει ως προς τη σύνθεσή του από άλλους κοινούς υπολογιστές, μιας και οι δυνατότητες του είναι σαφώς αναβαθμισμένες. Κύρια χαρακτηριστικά ενός εξυπηρετητή είναι οι επεξεργαστές που υποστηρίζει και χρησιμοποιεί για την επεξεργασία των δεδομένων που δέχεται, οι γρήγοροι και μεγάλης χωρητικότητας σκληροί δίσκοι αλλά και οι ταχύτατες μνήμες που υποστηρίζει. Συνήθως συνοδεύεται από σύστημα διπλής τροφοδοσίας (dual power supply) και από συσκευή παροχής αδιάλειπτης ενέργειας (UPS), για μεγαλύτερη αξιοπιστία και σιγουριά στις παρεχόμενες υπηρεσίες του. Στους εξυπηρετητές Ιστού υλοποιείτε

και η πλευρά του εξυπηρετητή του πρωτοκόλλου HTTP.

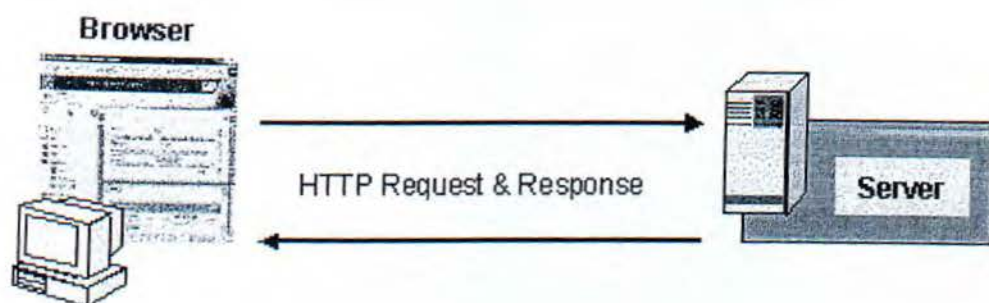
Το Πρωτόκολλο Μεταφοράς Υπερκειμένου HTTP (HyperText Transfer Protocol)

είναι η καρδιά του Ιστού. Το HTTP ανήκει στο στρώμα εφαρμογών του Διαδικτύου και υλοποιείτε ως δύο προγράμματα: ένα πρόγραμμα πελάτη (client program) και ένα πρόγραμμα εξυπηρετητή (server program). Τα δύο αυτά προγράμματα εκτελούνται σε διαφορετικά μηχανήματα επικοινωνώντας μεταξύ τους ανταλλάσσοντας HTTP μηνύματα. Συγκεκριμένα το HTTP ορίζει τη δομή των μηνυμάτων αυτών καθώς και τον τρόπο ανταλλαγής τους ανάμεσα στον πελάτη και στον εξυπηρετητή.

Επίσης ορίζει τον τρόπο με τον οποίο οι πελάτες του Ιστού (π.χ. οι browsers) ζητούν (request) Ιστοσελίδες από τους εξυπηρετητές του Ιστού (π.χ. τους Web servers) και πως οι εξυπηρετητές μεταφέρουν τις Ιστοσελίδες στους πελάτες.

Όταν ο χρήστης ζητά μία Ιστοσελίδα, ο browser στέλνει ένα μήνυμα HTTP αίτησης (HTTP request), για τα διάφορα αντικείμενα της σελίδας, στον εξυπηρετητή. Ο εξυπηρετητής όταν λάβει το μήνυμα αυτό ανταποκρίνεται με μηνύματα HTTP

απόκρισης (HTTP response) στα οποία περιέχονται τα αιτούμενα αντικείμενα.



Σχήμα 9. Ανταλλαγή δεδομένων μεταξύ server και browser

Οι browser και οι εξυπηρετητές υλοποιούν την έκδοση 1.0 του HTTP (HTTP/1.0) ή την έκδοση 1.1 (HTTP/1.1). Τόσο το HTTP/1.0 όσο και το HTTP/1.1 χρησιμοποιούν το TCP ως πρωτόκολλο μεταφοράς. Αφού ο πελάτης εγκαταστήσει μία σύνδεση TCP με τον εξυπηρετητή αρχίζει την αποστολή μηνυμάτων – αιτήσεων προς αυτόν και τη λήψη μηνυμάτων – απαντήσεων από αυτόν. Λόγω της χρήσης του TCP το HTTP δεν χρειάζεται να ασχοληθεί καθόλου με τη μεταφορά των δεδομένων. Το μόνο που πρέπει να κάνει είναι να στείλει τις αιτήσεις μέσω της TCP σύνδεσης και να περιμένει τις απαντήσεις. Το TCP εγγυάται την αξιόπιστη μεταφορά των δεδομένων καθώς και τον έλεγχο της συμφόρησης.

### 1.5.1 ΜΟΝΙΜΕΣ ΚΑΙ ΜΗ ΜΟΝΙΜΕΣ ΣΥΝΔΕΣΕΙΣ

Το HTTP μπορεί να χρησιμοποιήσει τόσο μόνιμες (persistent) όσο και μη μόνιμες (non-persistent) συνδέσεις. Η έκδοση HTTP/1.0 χρησιμοποιεί μη μόνιμες συνδέσεις ενώ η HTTP/1.1 χρησιμοποιεί κύρια μόνιμες συνδέσεις.

## Μη μόνιμες συνδέσεις

Για να εμφανίσουμε για παράδειγμα μια ιστοσελίδα στη οθόνη του υπολογιστή μας που αποτελείται από ένα βασικό HTML αρχείο και άλλα τέσσερα αρχεία εικόνων πρέπει να ακολουθηθούν κάποια βήματα για τη λήψη της ιστοσελίδας. Έστω ότι το URL του βασικού HTML αρχείου είναι το [www.ntua.gr/index..htm](http://www.ntua.gr/index..htm).

- Ο HTTP πελάτης εγκαθιστά μία σύνδεση TCP στην θύρα 80 του εξυπηρετητή HTTP. Η θύρα 80 είναι η προκαθορισμένη θύρα στην οποία ο εξυπηρετητής HTTP περιμένει για αφίξεις αιτήσεων HTTP από διάφορους πελάτες HTTP.
- Στη συνέχεια ο πελάτης HTTP στέλνει ένα μήνυμα HTTP αίτησης ζητώντας το αρχείο index.htm.
- Ο εξυπηρετητής στέλνει το αρχείο index.htm στον πελάτη μέσω ενός HTTP μηνύματος απόκρισης.
- Ο HTTP εξυπηρετητής κλείνει την TCP σύνδεση (στην πραγματικότητα η TCP σύνδεση κλείνει όταν επιβεβαιωθεί η λήψη του αρχείου από τον πελάτη).
- Ο browser λαμβάνει το αρχείο index.htm, κλείνει τη σύνδεση, και απεικονίζει τα περιεχόμενά του στην οθόνη του χρήστη. Επίσης ο browser αναγνωρίζει και τα URL των τεσσάρων αναφερόμενων εικόνων της Ιστοσελίδας.
- Τα τέσσερα πρώτα βήματα επαναλαμβάνονται για καθένα από τα αναφερόμενα αντικείμενα (στην περίπτωσή μας εικόνες).

Καταλαβαίνουμε για πιο λόγο αυτή η σύνδεση ονομάζεται μη μόνιμη αφού μετά την μεταφορά ενός αντικειμένου της ιστοσελίδας η TCP σύνδεση τερματίζεται και για κάθε άλλο αντικείμενο της ιστοσελίδας πρέπει να εγκατασταθεί μια καινούρια TCP σύνδεση.

## Μόνιμες συνδέσεις

Στις μόνιμες συνδέσεις ο εξυπηρετητής δεν τερματίζει τη TCP σύνδεση αμέσως μετά την αποστολή της απόκρισης. Αντίθετα, την τερματίζει μετά από κάποιο συγκεκριμένο χρονικό διάστημα κατά το οποίο η σύνδεση είναι ανενεργή (δηλαδή δεν γίνεται μεταφορά δεδομένων). Μέσα στο χρονικό διάστημα αυτό ο browser

μπορεί να ζητήσει την αποστολή και άλλων αντικειμένων από την ίδια Ιστοσελίδα αλλά και την αποστολή και άλλων Ιστοσελίδων από τον ίδιο εξυπηρετητή Ιστού.

Υπάρχουν δύο τύποι μόνιμων συνδέσεων: με pipelining και χωρίς pipelining. Στην πρώτη περίπτωση ο πελάτης στέλνει μία καινούργια αίτηση μόνο μετά τη λήψη της προηγούμενης απάντησης. Επίσης η χρήση pipelining καλύπτει και ένα άλλο μειονέκτημα που έχει η απουσία pipelining. Το μειονέκτημα αυτό είναι ο χρόνος αναμονής της λήψης απόκρισης από τους πελάτες και τη λήψη αιτήσεων από τους εξυπηρετητές, κατά τον οποίο γίνεται σπατάλη πόρων που θα μπορούσαν να χρησιμοποιηθούν σε άλλους σκοπούς.

## 1.5.2 HTTP ΜΗΝΥΜΑΤΑ

Το HTTP ορίζει μόνο δύο τύπους μηνυμάτων: τις HTTP αιτήσεις (requests) και τις HTTP αποκρίσεις (responses). Οι HTTP αιτήσεις γράφονται με χαρακτήρες ASCII και μπορούν να διαβαστούν από τους ανθρώπους και συνήθως αποτελούνται από 5 γραμμές κειμένου. Μία HTTP αίτηση έχει την ακόλουθη μορφή.

```
GET /lessons/index.htm HTTP/1.1
```

```
Connection: close
```

```
User-agent: Mozilla/4.0
```

```
Accept: text/html, image/gif, image/jpeg
```

```
Accept-language:gr
```

Η πρώτη γραμμή κειμένου ονομάζεται γραμμή αίτησης (request line), ενώ οι επόμενες γραμμές επικεφαλίδας (header lines).

- Η γραμμή αίτησης περιέχει τρία πεδία: το πεδίο μεθόδου, το πεδίο URL και το πεδίο HTTP έκδοσης. Το πεδίο μεθόδου μπορεί να έχει μία από τις ακόλουθες τιμές: GET, POST και HEAD. Η πιο συνηθισμένη μέθοδος στις HTTP αιτήσεις είναι η

GET, με την οποία ζητείται από τον εξυπηρετητή η αποστολή του αρχείου που εμφανίζεται στο πεδίο URL. Στο URL δεν είναι απαραίτητο να υπάρχει και το όνομα του host, αφού ήδη έχει εγκατασταθεί μία σύνδεση με τον host αυτό (δηλαδή τον HTTP εξυπηρετητή). Τέλος, στο πεδίο έκδοσης HTTP περιγράφεται η έκδοση του HTTP που χρησιμοποιεί ο αιτών host. Στο παραπάνω παράδειγμα χρησιμοποιείται το HTTP/1.1.

- Η γραμμή Connection:close λέει στον εξυπηρετητή να τερματιστεί η σύνδεση μετά την αποστολή του αρχείου, δηλαδή να μη γίνει χρήση μόνιμης σύνδεσης.
- Με το User-agent δηλώνει τον τύπο του browser του χρήστη. Στο παραπάνω παράδειγμα δηλώνεται ότι ο χρήστης χρησιμοποιεί τον browser Mozilla/4.0.
- Η Accept γραμμή δηλώνει τον τύπο των αντικειμένων που υποστηρίζει ο browser. Στο παραπάνω παράδειγμα δηλώνεται ότι ο browser υποστηρίζει αρχεία HTML και αρχεία εικόνων τύπου GIF και JPEG.
- Η Accept-language γραμμή δηλώνει την γλώσσα έκδοσης του αντικειμένου που επιθυμεί να λάβει ο browser.
- Στο παραπάνω παράδειγμα ο εξυπηρετητής στέλνει μία ελληνική έκδοση του αρχείου lessons/index.htm.

Μία HTTP απόκριση έχει την ακόλουθη μορφή:

HTTP/1.1 200 OK

Connection: close

Date: Thu, 06 Aug 1998 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT

Content-Length: 6821

Content-Type: text/html

data data data data data ...



Το HTTP μήνυμα απόκρισης αποτελείται από τρία μέρη: την γραμμή κατάστασης (status line), τις γραμμές επικεφαλίδας (header lines) και το τμήμα περιεχομένου (entity body). Το τμήμα περιεχομένου περιέχει τα δεδομένα του αιτούμενου αρχείου, τα οποία στο παραπάνω παράδειγμα φαίνονται ως data data data data data data

- Η γραμμή κατάστασης (status line) περιέχει τρία πεδία: το πεδίο HTTP έκδοσης, το πεδίο κωδικού κατάστασης και το πεδίο του αντίστοιχου μηνύματος κατάστασης. Στο παραπάνω παράδειγμα δηλώνεται ότι γίνεται χρήση του HTTP/1.1 και ότι δεν παρουσιάστηκε κάποιο σφάλμα (κωδικός κατάστασης 200 και αντίστοιχο μήνυμα κατάστασης OK).

Στο παράδειγμα περιλαμβάνονται έξι γραμμές επικεφαλίδας

- Η γραμμή Connection : close δηλώνει ότι μετά την αποστολή του μηνύματος αυτού η TCP σύνδεση θα τερματιστεί.
- Η γραμμή Date δηλώνει την ημερομηνία και την ώρα κατά την οποία στάλθηκε η απόκριση
- Η γραμμή Server δηλώνει τον τύπο του εξυπηρετητή.
- Η Last-Modified δηλώνει την ημερομηνία και την ώρα κατά την οποία το αντικείμενο δημιουργήθηκε ή τροποποιήθηκε για τελευταία φορά.
- Η γραμμή Content-Length δηλώνει το μήκος των δεδομένων που αποστέλλονται. Εδώ είναι 6821 bytes
- Η γραμμή Content-Type δηλώνει τον τύπο του αντικειμένου που περιέχεται στο τμήμα περιεχομένου του μηνύματος. Στο παράδειγμά μας θα αποσταλεί ένα αρχείο HTML.

## 1.6 Η ΓΛΩΣΣΑ HTML

Η γλώσσα HTML (HyperText Markup Language) είναι ο τρόπος με τον οποίο ο browser απεικονίζει τα έγγραφα. Με τον όρο έγγραφα εννοούμε κάθε είδους αρχείου, όπως αρχείο κειμένου, εικόνες, animation και βίντεο. Τα αρχεία της HTML είναι απλά αρχεία κειμένου στα οποία υπάρχουν ειδικοί κώδικες, οι οποίοι ονομάζονται

ετικέτες (tags), που δηλώνουν στον browser τον τρόπο με τον οποίο θα εμφανιστούν τα αρχεία αυτά.

Οι ετικέτες έχουν την παρακάτω γενική δομή:

```
<όνομα_ετικέτας>κείμενο</όνομα_ετικέτας>
```

Με το όνομα\_ετικέτας δηλώνεται στον browser η ενέργεια που θα εκτελεστεί στο κείμενο. Ένα αρχείο HTML είναι ένα αρχείο κειμένου (ASCII). Επομένως, για να δημιουργήσουμε ένα HTML αρχείο αρκεί να ανοίξουμε έναν επεξεργαστή κειμένου, να πληκτρολογήσουμε το κείμενό μας, να το αποθηκεύσουμε σε ένα αρχείο με κατάληξη .htm ή .html και στη συνέχεια να το εμφανίσουμε κάνοντας χρήση του browser.

Η HTML επίσης έχει τη δυνατότητα δημιουργίας δεσμών υπερκειμένου (hypertext links) προς άλλα έγγραφα (όπως ιστοσελίδες, γραφικά, βίντεο και animation). Ο χρήστης, με ένα απλό κλικ του ποντικιού, μπορεί να μεταβεί σε οποιαδήποτε τοποθεσία του Διαδικτύου.

## 2. ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ ARDUINO

Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη C++ με κάποιες μετατροπές). 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 προ-συναρμολογημένες. Το σχηματικό διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

### **Πλατφόρμα:**

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 ενσωμάτωση 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.

Γενικά όλες οι πλακέτες είναι προγραμματισμένες μέσω μιας σειριακής σύνδεσης RS-232, αλλά ο τρόπος με τον οποίο αυτό υλοποιείται ποικίλλει ανάλογα με την έκδοση. Οι σειριακές πλακέτες Arduino περιέχουν ένα απλό κύκλωμα αντιστροφής για την μετατροπή ανάμεσα στα σήματα των επιπέδων RS-232 και TTL.

Οι πλακέτες Arduino που κυκλοφορούν σήμερα στην αγορά, συμπεριλαμβανόμενης και της Diecimila, προγραμματίζονται μέσω USB, εφαρμόζοντας ένα τσίπ προσαρμογέα USB-to-serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το

Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν προσαρμογέα USB-to-serial σε μορφή πλακέτας ή καλωδίου.

Η πλακέτα του Arduino έχει εκτεθειμένες τις περισσότερες επαφές εισόδου/εξόδου για χρήση με άλλα κυκλώματα. Το Diecimila, για παράδειγμα, παρέχει 14 ψηφιακές επαφές εισόδου/εξόδου, από τις οποίες οι 6 μπορούν να παράξουν σήματα PWM, και 6 αναλογικές εισόδους. Αυτές οι επαφές είναι διαθέσιμες στην κορυφή της πλακέτας μέσω θηλυκών συνδέσεων μεγέθους 0,1 ιντσών.

Διάφορες plug-in πλακέτες εφαρμογών γνωστές σαν “shields” είναι, επίσης, διαθέσιμες στο εμπόριο.

### **Λογισμικό:**

Το IDE του Arduino είναι γραμμένο σε Java και μπορεί να τρέξει σε πολλαπλές πλατφόρμες. Περιλαμβάνει επεξεργαστή κώδικα (επεξεργαστή κειμένου με διάφορα εύχρηστα εργαλεία) και μεταγλωττιστή, και έχει την ικανότητα να φορτώνει εύκολα το πρόγραμμα μέσω σειριακής θύρας από τον υπολογιστή στην πλακέτα.

Το περιβάλλον ανάπτυξης είναι βασισμένο στην Processing, ένα περιβάλλον ανάπτυξης σχεδιασμένο να εισαγάγει στον προγραμματισμό νέους χρήστες μη εξοικειωμένους με την ανάπτυξη λογισμικού. Η συγκεκριμένη γλώσσα προγραμματισμού προέρχεται από την Wiring, μια γλώσσα που μοιάζει με την C η οποία παρέχει παρόμοια λειτουργικότητα για μια πιο περιορισμένη σχεδίασης πλακέτα, της οποίας το περιβάλλον ανάπτυξης βασίζεται επίσης στην Processing.

Είναι χρήσιμο εργαλείο για την πραγματοποίηση ιδεών προγραμματισμού επειδή χρησιμοποιείται σχετικά μικρός κώδικας επεξεργασίας για αρκετά μικρά πράγματα όπως η δημιουργία σύνδεσης δικτύων, η σύνδεση μιας εξωτερικής συσκευής μέσω μίας σειριακής θύρας ή ο έλεγχος ψηφιακής κάμερας μέσω Fire Wire. Επειδή βασίζεται στην προγραμματιστική γλώσσα Java, μπορεί να χρησιμοποιηθεί σε οποιοδήποτε λειτουργικό σύστημα όπως σε MAC OS X, Windows και Linux.

Σε περίπτωση που κάποιο κομμάτι της Processing δεν είναι επιθυμητό είναι σε θέση να χρησιμοποιήσει δείγματα του κώδικα και τα κατάλληλα σχόλια με τη χρήση ψευδογλώσσας για οποιοδήποτε περιβάλλον πολυμέσων είναι επιθυμητό. Η πλατφόρμα διαθέτει οκτώ κουμπιά επιλογών τα οποία είναι:

Επαλήθευση (verify)

Διακοπή (stop)

Νέο (new)

Άνοιγμα (open)

Αποθήκευση (save)

Μεταφόρτωση στην I/O πλατφόρμα (upload to I/O board)

Σειριακός έλεγχος (serial monitor)

Το κουμπί επιλογής Επαλήθευση ελέγχει αν ο κώδικας του χρήστη περιέχει λάθη και τα επισημαίνει, ενώ το κουμπί Διακοπή διακόπτει τη σειριακή επικοινωνία. Το κουμπί επιλογής Νέο δημιουργεί νέα καρτέλα για την καταγραφή κώδικα, το κουμπί επιλογής άνοιγμα παρουσιάζει όλους τους κώδικες που έχουν δημιουργηθεί και το κουμπί επιλογής μεταφόρτωση στην I/O πλατφόρμα μεταβιβάζει το πρόγραμμα στην I/O πλατφόρμα, αφού πρώτα έχει γίνει αποθήκευση και έλεγχος της σωστής σύνταξης του κώδικα. Το κουμπί επιλογής σειριακός έλεγχος εμφανίζει τα σειριακά δεδομένα στην οθόνη τα οποία αποστέλλονται από την πλατφόρμα.

Υπάρχει η δυνατότητα χρήσης υπάρχουσών βιβλιοθηκών σε οποιονδήποτε κώδικα με τη χρήση της επιλογής 'Import Library' ή χειρονακτικά προσθέτοντας στον κώδικα την εντολή #include(όνομα βιβλιοθήκης), Παρέχονται κάποιες βιβλιοθήκες με την εγκατάσταση του προγράμματος οδήγησης της πλατφόρμα και αυτές είναι:

**EEPROM-** ανάγνωση και εγγραφή σε μόνιμη αποθήκευση (εγγραφή στη μνήμη της πλατφόρμα)

**Ethernet-** Εφαρμόζεται για τη σύνδεση με το Διαδίκτυο χρησιμοποιώντας το Arduino Ethernet Shield

**Firmata-** Χρησιμοποιείται για την επικοινωνία με τις εφαρμογές του υπολογιστή που χρησιμοποιείται ένα τυποποιημένο τμηματικό πρωτόκολλο

**Liquid Crystal-** Εφαρμόζεται για τον έλεγχο των επιδείξεων υγρού κρυστάλλου (LCDs)

**Servo-** Εφαρμόζεται για τον έλεγχο των σερβομηχανών

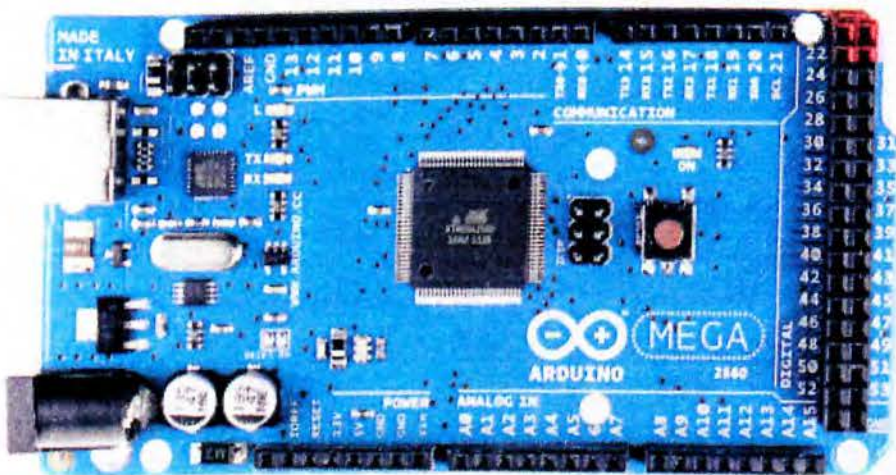
**Software Serial-** Εφαρμόζεται για την τμηματική ανακοίνωση σχετικά με οποιαδήποτε ψηφιακές εισόδους.

**Stepper-** Εφαρμόζεται για τον έλεγχο των stepper μηχανών

**Wire-** Η διεπαφή δύο καλωδίων (TWI/I2C) έχει τη δυνατότητα αποστολής και λήψης των δεδομένων εκτός των συσκευών και αισθητήρων

## 2.1 ARDUINO MEGA 2560

Το arduino mega 2560 είναι μια πλατφόρμα βασισμένη στον μικροεπεξεργαστή ATmega2560. Έχει 54 ψηφιακές εισόδους και εξόδους (εκ των οποίων οι 14 μπορούν να χρησιμοποιηθούν ως αποτέλεσμα PWM), 16 αναλογικές εισόδους, έναν ταλαντωτή κρυστάλλου 16MHZ, μια σύνδεση USB, μια είσοδο παροχής ενέργειας, έναν διασυνδετικό αγωγό ICSP και ένα κουμπί αναστοιχειοθέτησης (reset button). Περιέχει όλα όσα απαιτούνται για να υποστηρίξουν τη λειτουργία του μικροεπεξεργαστή. Μπορεί να συνδεθεί με έναν υπολογιστή διαμέσου ενός καλωδίου USB ή να τροφοδοτηθεί με έναν AC-to-DC προσαρμογέα ή με μια μπαταρία.



Σχήμα 10. Arduino Mega 2560

### 2.1.1 ΕΙΔΙΚΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ARDUINO MEGA

Μικροεπεξεργαστής	ATmega2560
Λειτουργούσα τάση	5V
Τάση εισαγωγής (που συνιστάται)	7-12V
Τάση εισαγωγής (όρια)	6-20V
Ψηφιακές I/O είσοδοι	54(εκ των οποίων οι 15 παρέχουν τη δυνατότητα παραγωγής PWM)
Αναλογικές είσοδοι	16
Συνεχές ρεύμα ανά I/O είσοδο	40 mA
Συνεχές ρεύμα για 3.3V είσοδο	50 mA
Μνήμη	256 KB (εκ των οποίων τα 8 KB χρησιμοποιούνται από τον bootloader)
SRAM	8 KB
EEPROM	4 KB
Ταχύτητα ρολογιών	16 MHz

## 2.1.2 ΤΡΟΦΟΔΟΤΗΣΗ

Το Arduino υπο μπορεί να τροφοδοτηθεί μέσω της σύνδεσης USB ή με μια εξωτερική παροχή ηλεκτρικού ρεύματος. Η πηγή ενέργειας επιλέγεται αυτόματα.

Η εξωτερική παροχή ηλεκτρικού ρεύματος ( εκτός της σύνδεσης USB) μπορεί να προέλθει είτε από έναν AC-to-DC προσαρμογέα είτε από μπαταρία.

Ο προσαρμογέα μπορεί να συνδεθεί στην είσοδο που βρίσκεται πάνω στη πλατφόρμα με διαστάσεις 2.1 mm. Κάποια καλώδια από μια μπαταρία μπορούν να συνδεθούν στις Gnd και Vin εισόδους που βρίσκονται στην τροφοδότηση μπαταρίας.

Η πλατφόρμα μπορεί να λειτουργήσει σε μια εξωτερική τροφοδότηση από 6 έως 20 volt. Εάν παρέχεται λιγότερο από 7 volt, η είσοδος τω 5 volt μπορεί να παρέχει λιγότερο από 5 volt και η πλατφόρμα μπορεί να είναι ασταθής. Εάν παρέχονται περισσότερα από 12 volt, ο ρυθμιστής τάσης μπορεί να υπερθερμανθεί και να βλάψει τη πλατφόρμα. Το προτεινόμενο φάσμα τροφοδότησης είναι 7-12 V.

### Είσοδοι τροφοδότησης

Οι είσοδοι τροφοδότησης είναι οι ακόλουθες

**VIN.** Η τάση εισαγωγής στον πίνακα Arduino όταν χρησιμοποιεί μια εξωτερική πηγή ενέργειας (σε αντιδιαστολή με 5 volt από τη σύνδεση USB ή άλλη ρυθμισμένη πηγή ενέργειας. Είναι δυνατόν να παρέχεται η τάση μέσω αυτής της εισόδου ή εάν παρέχοντας την τάση μέσω του προσαρμογέα εισόδου της μπαταρίας, εξασφαλίζεται πρόσβαση σε αυτήν την είσοδο.

**5V.** Η ρυθμισμένη παροχή ηλεκτρικού ρεύματος χρησιμοποιείται για να τροφοδοτήσει τον μικροεπεξεργαστή και άλλα συστατικά στον πίνακα. Αυτό μπορεί να προέλθει από την είσοδο VIN διαμέσου ενός ρυθμιστή της πλατφόρμας είτε να παρασχεθεί από USB είτε από έναν άλλο ρυθμισμένο 5 V ανεφοδιασμό.

**3V3.** Ένας ανεφοδιασμός 3V3 παράγεται από τον ρυθμιστή που βρίσκεται στην πλατφόρμα. Το μέγιστο ρεύμα που μπορεί να μεταδώσει είναι 50 mA .



### 2.1.3 ΠΡΟΣΤΑΣΙΑ ΑΠΟ ΥΠΕΡΕΝΤΑΣΗ ΣΤΟ USB

Το Arduino μπορεί να επαναριθμήσει τις ασφάλειες πολυμερούς, ώστε να προστατέψει τις θύρες USB από την χαμηλή ή την υψηλή τάση. Αν και οι περισσότεροι υπολογιστές παρέχουν τη δική τους εσωτερική προστασία, η ασφάλεια παρέχει ένα επιπλέον επίπεδο προστασίας. Εάν περισσότερα από 500 mA εφαρμοστούν στη θύρα USB, η ασφάλεια διακόπτει αυτόματα τη σύνδεση έως ότου διορθωθεί το πρόβλημα λόγω υψηλής ή χαμηλής τάσης.

### 2.1.4 ΕΙΣΟΔΟΙ ΚΑΙ ΕΞΟΔΟΙ

Κάθε μία από τις 54 ψηφιακές εισόδους στο Arduino Mega μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος χρησιμοποιώντας τις `pinMode()`, `digitalWrite()`, `digitalRead()` λειτουργίες. Λειτουργούν σε 5 volt. Κάθε είσοδος μπορεί να παρέχει ή να λάβει ένα μέγιστο 40 mA και έχει έναν εσωτερικό pull-up αντιστάτη (αποσυνδεδεμένο εξ ορισμού) των 20-50 kOhms. Επιπλέον μερικές εισοδοί έχουν ειδικευμένες λειτουργίες.

**Σειριακές : 0 (RX) and 1 (TX); . Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX)** Χρησιμοποιούνται για την παραλαβή (RX) και τη διαβίβαση (TX) των σειριακών στοιχείων TTL. Αυτές οι εισοδοί συνδέονται με τις αντίστοιχες εισόδους του ATmega16U2 USB-to-TTL Serial chip.

**Εξωτερικοί διακόπτες: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** Αυτές οι εισοδοί μπορούν να διαμορφωθούν για να προκαλέσουν διακοπή σε μια χαμηλή αξία, μια αύξηση ή μείωση άκρης ή μια αλλαγή στην αξία.

**PWM: 2 to 13 and 44 to 46** . Παρέχουν οκτάμπιτη παραγωγή PWM μαζί με την analogWrite ()λειτουργία

**SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)**. Αυτές οι εισοδοι υποστηρίζουν την επικοινωνία SPI, η οποία αν κα παρέχεται από το υλικό, δεν συμπεριλαμβάνεται στη γλώσσα Arduino.

**LED: 13**. Υπάρχει ενσωματωμένο led που συνδέεται με την ψηφιακή είσοδο 13. Όταν η τιμή της εισόδου είναι HIGH το led είναι αναμμένο, ενώ όταν η τιμή της εισόδου είναι LOW το led δεν είναι αναμμένο.

**TWI 20 (SDA) and 21 (SCL)**. Το Mega2560 έχει 16 αναλογικές εισόδους κάθε μία από τις οποίες παρέχει 10 bit της ανάλυσης (δηλ.1024 διαφορετικές τιμές). Μετρούν από τη γείωση έως τα 5 volt ενώ είναι πιθανό να αλλαχτεί το ανώτερο όριο της εμβέλειάς τους χρησιμοποιώντας την AREF είσοδο και την analogReference() λειτουργία. Επιπλέον μερικές εισοδοι έχουν ειδικές λειτουργίες.

**AREF**. Αναφορά τάσης για τις αναλογικές εισόδους χρησιμοποιώντας την εντολή analogReference().

**Reset**. Η επαναφορά της γραμμής αυτής σε LOW τιμή επαναριθμίζει τον μικροεπεξεργαστή. Συνήθως χρησιμοποιείται για την πρόσθεση ενός κουμπιού reset για να προστατεύει αυτά που μπλοκάρουν στην πλατφόρμα.

## 2.1.5 ΕΠΙΚΟΙΝΩΝΙΑ

Το Arduino Mega2560 έχει κάποιες ευκολίες για την επικοινωνία του με έναν υπολογιστή, με ένα άλλο Arduino ή με άλλους μικροεπεξεργαστές. Ο ATmega2560 παρέχει 4 hardware UARTs για TTL (5V) σειριακής επικοινωνία . Ένα ATmega16U2 διοχετεύει αυτή την σειριακή επικοινωνία διαμέσου USB και εμφανίζεται ως μια εικονική πόρτα com στο λογισμικό του υπολογιστή. Το ATmega16U2 υλικό λογισμικού χρησιμοποιεί τους στάνταρ USB COM οδηγούς και έτσι δεν χρειάζονται εξωτερικοί οδηγοί. Όμως στα Windows ένα .inf file απαιτείται. Το λογισμικό Arduino περιλαμβάνει ένα σειριακό όργανο ελέγχου που επιτρέπει στα απλά δεδομένα κειμένου να αποσταλούν προς αλλά και από την πλατφόρμα Arduino. Τα

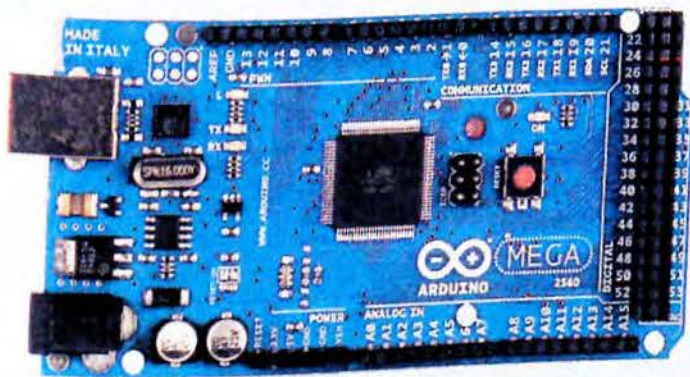
φορά που γίνεται μια σύνδεση από το λογισμικό (μέσω USB). Για το επόμενο μισό δευτερόλεπτο περίπου το bootloader τρέχει στο Uno. Ενώ προγραμματίζεται για να αγνοήσει τα δύσμορφα στοιχεία (δηλ. οτιδήποτε εκτός μια αποστολή νέου κώδικα), θα παρεμποδίσει τα πρώτα bytes που στέλνονται στη πλατφόρμα μετά την πραγματοποίηση της σύνδεσης. Εάν ένας τρέχον κώδικας στη πλατφόρμα λαμβάνει μια επιβεβαίωση ή άλλα δεδομένα όταν ξεκινά για πρώτη φορά, πρέπει να είναι σίγουρο ότι το λογισμικό με το οποίο επικοινωνεί περιμένει ένα δευτερόλεπτο μετά το άνοιγμα της σύνδεσης και πριν την αποστολή των δεδομένων.

Μπορούμε επίσης να ακυρώσουμε την επιλογή αυτόματου reset συνδέοντας μία αντίσταση 110 ohm από 5volt για να επαναφέρουμε τη λειτουργία.

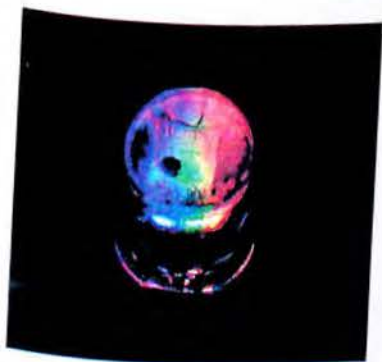
## 2.2 ΣΧΕΔΙΑΣΗ HARDWARE

Τα υλικά που χρειάστηκαν για την κατασκευή της πλακέτα είναι τα εξής:

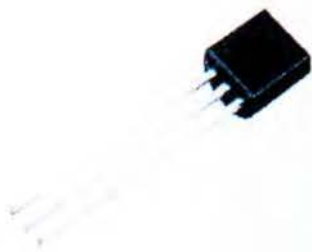
- Arduino Mega 2560



- Ένα led τύπου RGB



- Τρεις αντιστάσεις περιορισμού ρεύματος για το led RGB
- Ένα αισθητήριο θερμοκρασίας

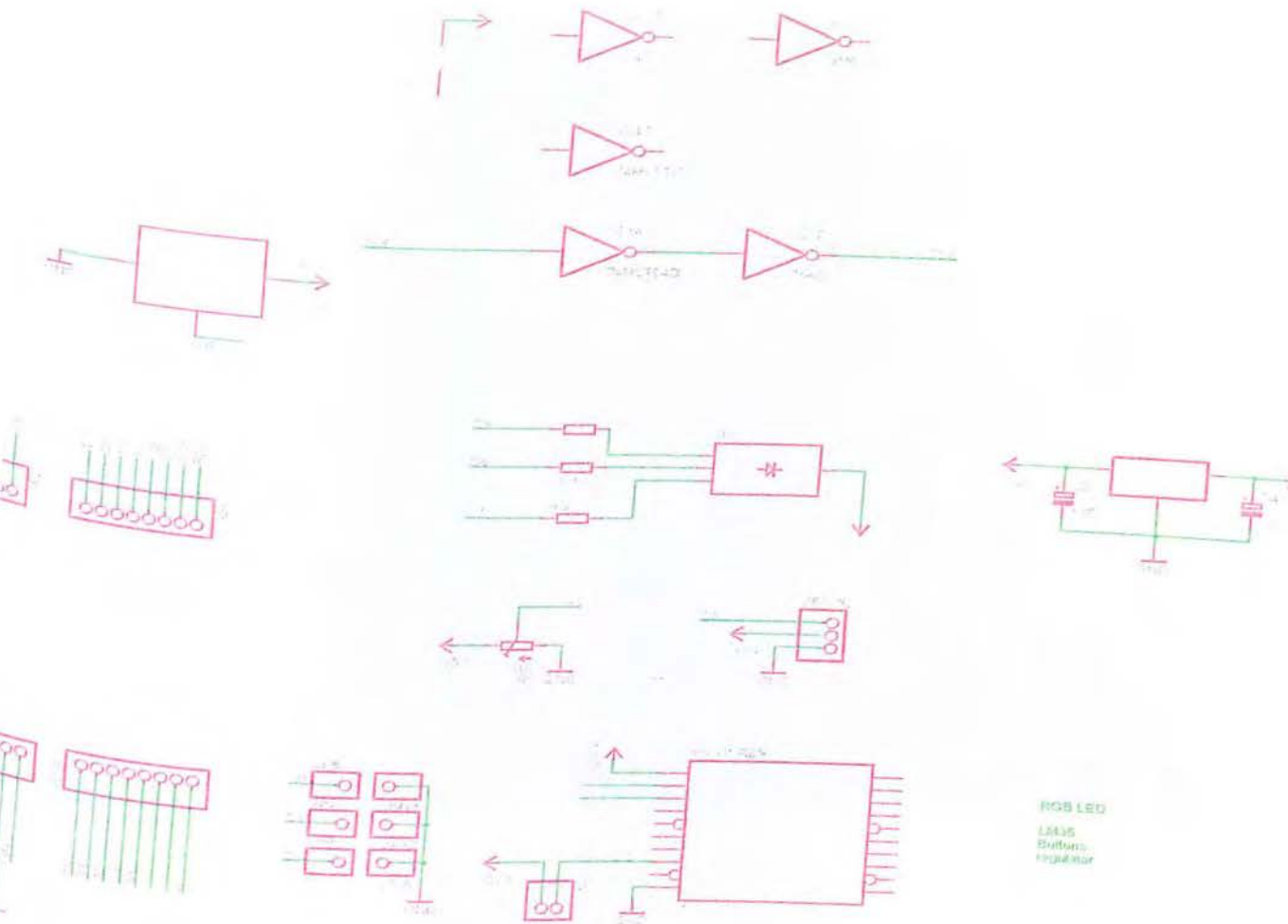


- Ένα ποτενσιόμετρο
- Τρεις διακόπτες
- Ένας σερβοκινητήρας



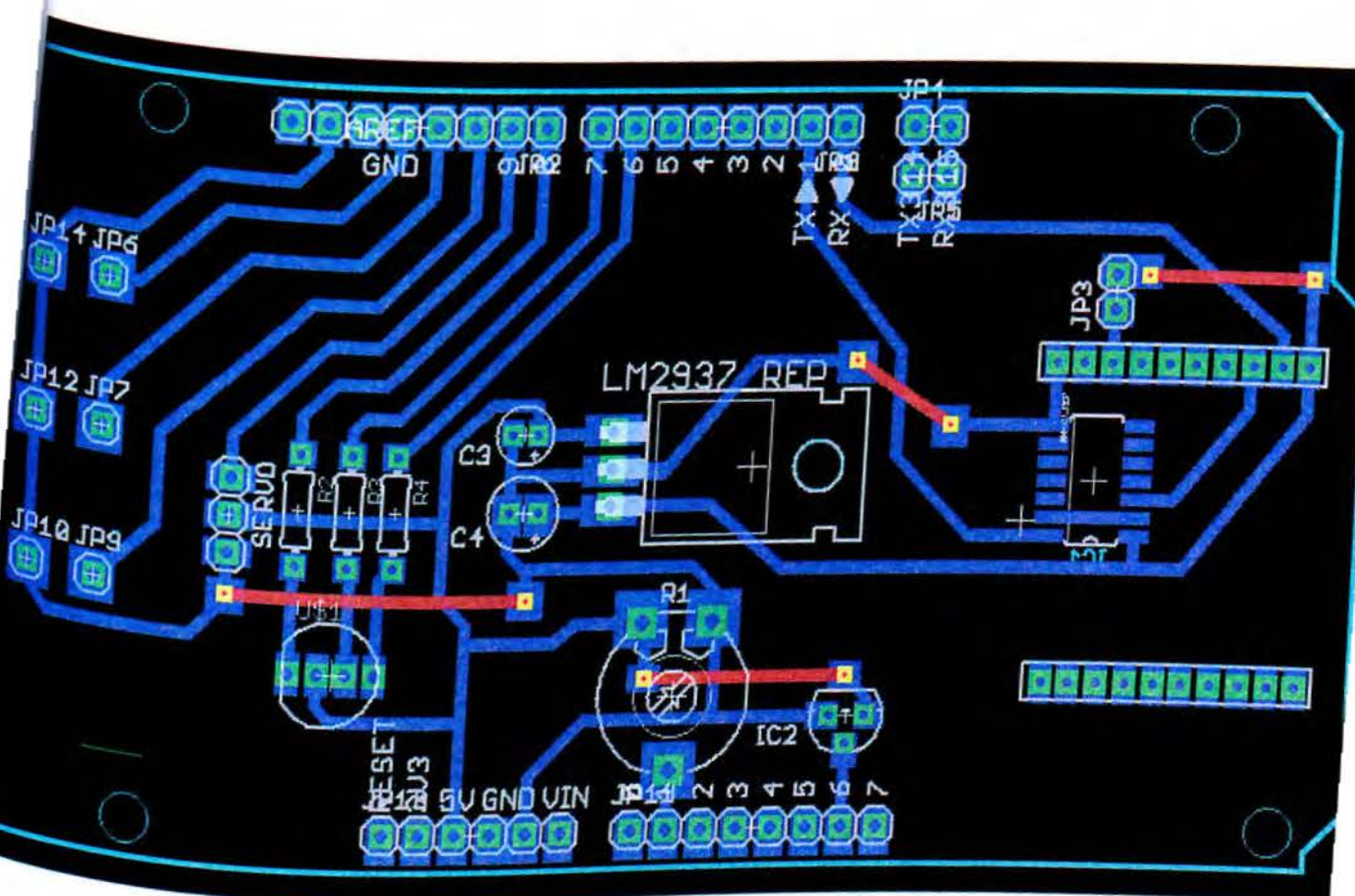
- Ένας μετατροπέας τάσης από 5V σε 3.5V με δύο πυκνωτές για την ασύρματη συσκευή Wi-fi. Ο arduino δίνει 5V τάση και το ασύρματο Wi-fi δουλεύει στα 3,5V.
- Μία ασύρματη συσκευή Wi-fi RN-XU Roving Networks.
- Ad-hoc για τη σύνδεση του arduino με τον υπολογιστή
- Καλώδιο usb για τη σύνδεση του arduino με το laptop

## 2.2.1 ΣΧΗΜΑΤΙΚΟ ΠΛΑΚΕΤΑΣ



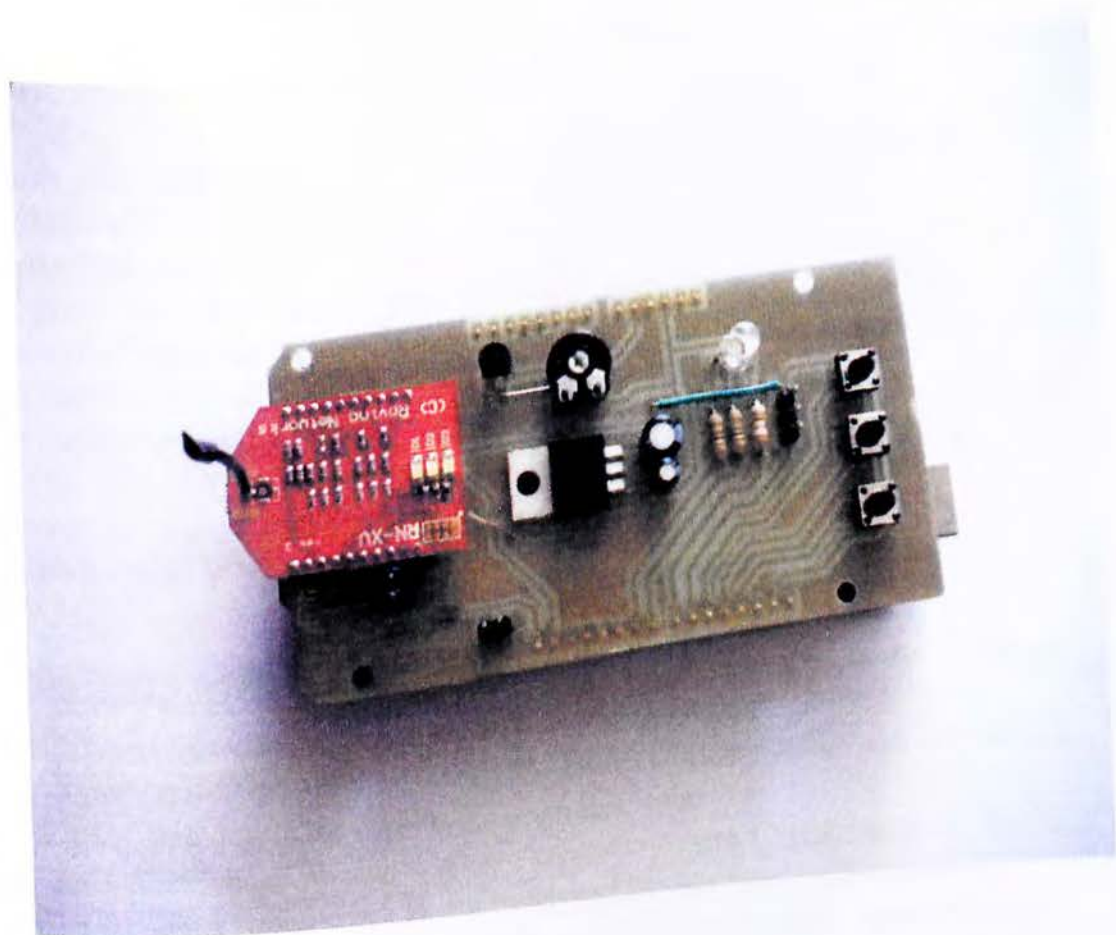
Σχήμα 11. Ηλεκτρονικό κύκλωμα πλακέτας

## 2.2.2 ΠΑΡΟΥΣΙΑΣΗ ΠΛΑΚΕΤΑΣ



Σχήμα 12.

### 2.2.3 ΤΕΛΙΚΗ ΜΟΡΦΗ ΠΛΑΚΕΤΑΣ



Σχήμα 13. Τελική μορφή πλακέτας





## "Δημιουργία ενός web server με arduino για τον έλεγχο μέσω μιας ιστοσελίδας"

Όνομα Σπουδαστή : Φουντουκίδης Παναγιώτης

ΑΜ: 35889

Επιβλέπων καθηγητής: Παπουτσιδάκης Μιχαήλ

[Αρχική Σελίδα](#)

### • Κώδικας index\_page σελίδας

```
PROGMEM prog_char index_page[]="<html>\n"  
"<head>\n"  
"<title>Αρχική</title>\n"  
"</head>\n"  
"<body bgcolor=\"#33CCFF\">\n"  
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"  
"\n"  
"<table border=\"0\" cellpadding=\"1\" cellspacing=\"1\" style=\"width: 100%; \">\n"  
"<tbody>\n"  
"<tr>\n"  
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a  
href=\"http://169.254.1.1/index\" style=\"color:#FF0000\">Αρχική</span></strong></  
a></td>\n"  
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a  
href=\"http://169.254.1.1/leds\" style=\"color:#FF0000\">Led</span></strong></a></t  
d>\n"
```

```

"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/sensor\" style=\"color:#FF0000\">Potentiometer</span></str
ong></a></td>\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/switches\" style=\"color:#FF0000\">Switches</span></stron
g></a></td>\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/servo\" style=\"color:#FF0000\">Servo</span></strong></a
></td>\n"
"</tr>\n"
"</tbody>\n"
"</table>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\"><strong><span style=\"font-size:
18pt\">Παρακαλώ επιλέξτε την σελίδα στην οποία επιθυμείτε να μεταφερθείτε από
την κορυφή της παρούσας.</span></strong></p>\n"
"\n"
"</html>\n";

```

Με αυτόν τον κώδικα html δημιουργούμε τη σελίδα με το όνομα index\_page. Σε αυτή τη σελίδα περιέχονται όλοι οι σύνδεσμοι των σελίδων και ο χρήστης επιλέγοντας τον σύνδεσμο που επιθυμεί θα μεταφέρεται αυτόματα στην σελίδα που επέλεξε. Αυτό έγινε με την εισαγωγή ενός πίνακα όπου εισάγαμε στα στοιχεία του τους αντίστοιχους συνδέσμους της κάθε σελίδας. Έτσι η σελίδα που δημιουργήσαμε είναι η εξής.

Παρακαλώ επιλέξτε την σελίδα στην οποία επιθυμείτε να μεταφερθείτε από την κορυφή της παρούσας.

#### • Κώδικας leds\_page σελίδας

```
PROGMEM prog_char leds_1_page[]="
"
"<html>\n"
"<head>\n"
"<title>Led</title>\n"
"</head>\n"
"<body bgcolor=\"#33CCFF\">\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"\n"
"<table border=\"0\" cellpadding=\"1\" cellspacing=\"1\" style=\"width: 100%; \">
"
"<tbody>\n"
"<tr>\n"
"<td style=\"text-align: center; \">
"
"<strong><span style=\"font-size: 16pt\"><a
"
"href=\"http://169.254.1.1/index\" style=\"color:#FF0000\">Αρχική</span></strong></
"
"a></td>\n"
"<td style=\"text-align: center; \">
"
"<strong><span style=\"font-size: 16pt\"><a
"
"href=\"http://169.254.1.1/leds\" style=\"color:#FF0000\">Led</span></strong></a></t
"
"d>\n"
```

```

"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/sensor\" style=\"color:#FF0000\">Potentiometer</span></str
ong></a></td>\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/switches\" style=\"color:#FF0000\">Switches</span></stron
g></a></td>\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/servo\" style=\"color:#FF0000\">Servo</span></strong></a
></td>\n"
"</tr>\n"
"</tbody>\n"
"</table>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"\n"
"\n"
"<p <span style=\"font-size: 16pt\"> Στην παρούσα σελίδα, ο χρήστης μπορεί να
επιλέξει το χρώμα που επιθυμεί, να ανάψει το led"
"τύπου RGB, το οποίο είναι συνδεδεμένο στον arduino. Δίνεται η δυνατότητα να
επιλέξει ανάμεσα από κόκκινο, πράσινο και μπλε."
"Επιπλέον ο χρήστης μπορεί να επιλέξει διαφορετικά χρώματα, δίνοντας μια τιμή από
0 έως 255 στο παρακάτω πεδίο.</span></p>\n"
"\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"\n"
"\n"
"<form      METHOD=get      action=\"http://169.254.1.1/leds\"><strong><span
style=\"font-size: 14pt\">\n"
"\n"
"<table style=\"width: 85%;\">\n"
"</tr>\n"
"<tr>\n"

```

```

" <td colspan="2" style="text-align:center">\n"
" <strong><span style="font-size: 14pt">Red:</span></strong><br />\n"
" </td>\n"
" </tr>\n"
" <tr>\n"
" <td colspan="2" style="text-align:center">\n"
" <input type="text" name="r" value="0" id="r"
style="width:160px;height:30px;background-color:#D0F18F"/>\n"
" </td>\n"
" </tr>\n"
"\n"
"\n"
" <tr>\n"
" <td colspan="2" style="text-align:center">\n"
" <strong><span style="font-size: 14pt">Green:</span></strong><br />\n"
" </td>\n"
" </tr>\n"
" <tr>\n"
" <td colspan="2" style="text-align:center">\n"
" <input type="text" name="g" value="0" id="g"
style="width:160px;height:30px;background-color:#D0F18F"/>\n"
" </td>\n"
" </tr>\n"
"\n"
"\n"
" <tr>\n"
" <td colspan="2" style="text-align:center">\n"
" <strong><span style="font-size: 14pt">Blue:</span></strong><br />\n"
" </td>\n"
" </tr>\n"
" <tr>\n"
" <td colspan="2" style="text-align:center">\n"
" <input type="text" name="b" value="0" id="b"
style="width:160px;height:30px;background-color:#D0F18F"/>\n"

```

```

" </td>\n"
"</tr>\n"
"\n"
"<tr>\n"
" <td colspan=\"2\" style=\"text-align:center\">\n"
" <input type=\"submit\" value=\"Submit\" /> \n"
" </td>\n"
"</tr>\n"
"\n"
"</table>\n"
"\n"
"\n"
"</form>\n"
"\n"
"\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"\n"
"<table align=\"center\" border=\"0\" cellpadding=\"1\" cellspacing=\"1\"
style=\"width: 50%; \">\n"
"<tbody>\n"
"<tr>\n"
"<td>\n"
" <form METHOD=get action=\"http://169.254.1.1/leds\">\n";
PROGMEM prog_char leds_red_on[]="\n"
" <input type=hidden name=cmd value=1>\n"
" <input type=submit value=\"Red ON\" style=\"background-
color:#CC0000;color:#FFFFFF;\" >\n"
"</form>\n"
"</td>\n"
"\n";
PROGMEM prog_char leds_red_off[]="\n"
' <input type=hidden name=cmd value=1>\n"

```

```

"      <input type=submit value=\"Red OFF\" style=\"background-
color:#CC0000;color:#FFFFFF;\" >\n"
" </form>\n"
"</td>\n"
"\n";
PROGMEM prog_char leds_green_on[]="\n"
"<td>\n"
" <form METHOD=get action=\"http://169.254.1.1/leds\">\n"
" <input type=hidden name=cmd value=2>\n"
"   <input type=submit value=\"Green ON \" style=\"background-
color:#00FF00;color:#000000;\">\n"
" </form>\n"
"</td>\n"
"\n";
PROGMEM prog_char leds_green_off[]="\n"
"<td>\n"
" <form METHOD=get action=\"http://169.254.1.1/leds\">\n"
" <input type=hidden name=cmd value=2>\n"
"   <input type=submit value=\"Green OFF \" style=\"background-
color:#00FF00;color:#000000;\">\n"
" </form>\n"
"</td>\n"
"\n";
PROGMEM prog_char leds_blue_off[]="\n"
"<td>\n"
" <form METHOD=get action=\"http://169.254.1.1/leds\">\n"
" <input type=hidden name=cmd value=3>\n"
" <input type=submit value=\"Blue OFF\" style=\"background-
color:#0000FF;color:#FFFFFF;\">\n"
" </form>\n"
"</td>\n"
"\n";
PROGMEM prog_char leds_blue_on[]="\n"
"<td>\n"

```

```

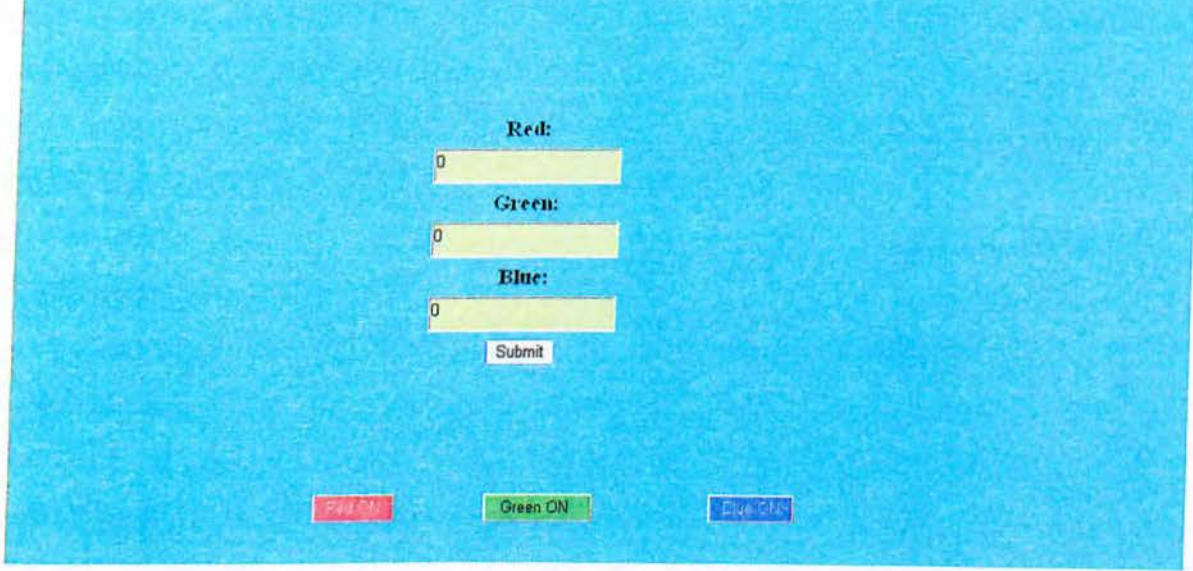
"<form METHOD=get action=\"http://169.254.1.1/leds\">\n"
"<input type=hidden name=cmd value=3>\n"
"<input type=submit value=\"Blue ON\" style=\"background-
color:#0000FF;color:#FFFFFF;\">\n"
"</form>\n"
"</td>\n"
"\n";
PROGMEM prog_char leds_2_page[]="\n"
"<td>\n"
"</tr>\n"
"</tbody>\n"
"</table>\n"
"\n"
"\n"
"</html>\n"
"";

```

Με αυτόν τον κώδικα δημιουργούμε τη σελίδα με το όνομα leds. Σε αυτή τη σελίδα θα μεταβαίνει ο χρήστης εάν θέλει να επεξεργαστεί το led RGB που υπάρχει πάνω στη πλακέτα. Κρατήσαμε σταθερό τον πίνακα με τους συνδέσμους των σελίδων στην κορυφή της σελίδας ώστε ο χρήστης ανά πάσα στιγμή να μπορεί να μεταφερθεί σε όποια σελίδα επιθυμεί με την επιλογή του κατάλληλου συνδέσμου. Δημιουργούμε τρία πλαίσια τύπου text μέσα στα οποία θα μπορεί ο χρήστης να πληκτρολογεί μία τιμή από 0-255 και θα μπορεί με αυτόν τον τρόπο να ελέγξει τη φωτεινότητα του κάθε led. Στη συνέχεια δημιουργήσαμε και τρία buttons που το καθένα αντιστοιχεί στο κάθε χρώμα του led ώστε ο χρήστης να μπορεί να ανάψει και να σβήσει το led με την επιλογή του αντίστοιχου button. Τέλος δημιουργήσαμε κάποιες επιπλέον σελίδες για το κάθε button για να πετύχουμε την αλλαγή της κατάστασης του button από ON σε OFF. Έτσι η τελική μορφή της σελίδας που θα πάρουμε θα είναι η εξής.



Στην παρούσα σελίδα, ο χρήστης μπορεί να επιλέξει το χρώμα που επιθυμεί να αναφαι το led του RGB, το οποίο είναι συνδεδεμένο στον arduino. Δίνεται η δυνατότητα να επιλέξει ανάμεσα από κόκκινο, πράσινο και μπλε. Επιπλέον ο χρήστης μπορεί να επιλέξει διαφορετικά χρώματα, δίνοντας μία τιμή από 0 έως 255 στο παρακάτω πεδίο.



#### ▪ Κώδικας sensor\_page σελίδας

```
PROGMEM prog_char sensor_page_1[]="n"
```

```
"<html>\n"
```

```
"<head>\n"
```

```
"<title>Potensiometer</title>\n"
```

```
"</head>\n"
```

```
"<body bgcolor=\"#33CCFF\">\n"
```

```
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
```

```
"\n"
```

```
"<table border=\"0\" cellpadding=\"1\" cellspacing=\"1\" style=\"width: 100%; \">\n"
```

```
"<tbody>\n"
```

```
"<tr>\n"
```

```
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
```

```
href=\"http://169.254.1.1/index\" style=\"color:#FF0000\">Αρχική</span></strong></
```

```
a></td>\n"
```

```
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
```

```
href=\"http://169.254.1.1/leds\" style=\"color:#FF0000\">Led</span></strong></a></t
```

```
d>\n"
```

```
"<td style="text-align: center; \"><strong><span style="font-size: 16pt"><a href="http://169.254.1.1/sensor" style="color:#FF0000">Potentiometer</span></strong></a></td>\n"
```

```
"<td style="text-align: center; \"><strong><span style="font-size: 16pt"><a href="http://169.254.1.1/switches" style="color:#FF0000">Switches</span></strong></a></td>\n"
```

```
"<td style="text-align: center; \"><strong><span style="font-size: 16pt"><a href="http://169.254.1.1/servo" style="color:#FF0000">Servo</span></strong></a></td>\n"
```

```
"</tr>\n"
```

```
"</tbody>\n"
```

```
"</table>\n"
```

```
"<p style="text-align: center" align="center">&nbsp;</p>\n"
```

```
"<p style="text-align: center" align="center">&nbsp;</p>\n"
```

```
"<p style="text-align: center" align="center">&nbsp;</p>\n"
```

```
"\n"
```

```
"<p <span style="font-size: 16pt">Στην παρούσα σελίδα, ο χρήστης μπορεί να ελέγξει την τιμή της θερμοκρασίας. Η τιμή αυτή, λαμβάνεται\
```

```
"από το αισθητήριο θερμοκρασίας LM35.</span></p>\n"
```

```
"\n"
```

```
"<p style="text-align: center" align="center">&nbsp;</p>\n"
```

```
"<p style="text-align: center" align="center">&nbsp;</p>\n"
```

```
"\n"
```

```
"<table border="1" cellspacing="0" cellpadding="2" width="400" align="center"><tbody>\n"
```

```
"<tr><td align=center><strong><span style="font-size: 16pt">Θερμοκρασία</span></strong></td>\n"
```

```
"<td align=center><strong><span style="font-size: 20pt">">
```

```
PROGMEM prog_char sensor_page_2[]=" °C</span></strong></td></tr>\n"
```

```
"</table>\n"
```

```
"\n"
```

```
"<p style="text-align: center" align="center">&nbsp;</p>\n"
```

```
"<p style="text-align: center" align="center">&nbsp;</p>\n"
```

```
"<p style="text-align: center" align="center">&nbsp;</p>\n"
```

```

"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"\n"
"<p <span style=\"font-size: 16pt\">Επίσης, ο χρήστης μπορεί να δει την τιμή που
μεταβάλλεται από το ποτενσιόμετρο.</span></p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"\n"
"<table border=\"2\" cellspacing=\"0\" cellpadding=\"2\" width=\"400\"
align=\"center\"><tbody>\n"
"<tr><td align=center><strong><span style=\"font-size:
16pt\">Τάση</span></strong></td>\n"
"<td align=center><strong><span style=\"font-size: 20pt\">";
PROGMEM prog_char sensor_page_3[]=" V</span></strong></td></tr>\n"
"</table>\n"
"\n"
"</html>";

```

στη σελίδα αυτή ο χρήστης θα μπορεί να ελέγξει τη θερμοκρασία του περιβάλλοντος με τη βοήθεια του αισθητηρίου θεοκρασίας που υπάρχει πάνω στη πλακέτα. Έτσι έχουμε φτιάξει ένα πίνακα με δύο στοιχεία. Το πρώτο στοιχείο θα περιέχει την τιμή της θερμοκρασίας που παίρνουμε από το αισθητήριο κα στο δεύτερο θα υπάρχει το σύμβολο 0C. Επίσης ο χρήστης θα μπορεί να βλέπει την τάση που παίρνουμε από ένα pin του arduino μεταβάλλοντας ένα ποτενσιόμετρο που είναι συνδεδεμένο στη πλακέτα. Αντίστοιχα φτιάχνουμε ένα ακόμη πίνακα στον οποίο το πρώτο στοιχείο θα είναι η τιμή της τάσης και το δεύτερο θα είναι το σύμβολο V (VOLT). Η τελική κόντα που θα πάρουμε θα είναι η εξής.

Στην παρούσα σελίδα, ο χρήστης μπορεί να ελέγξει την τιμή της θερμοκρασίας. Η τιμή αυτή λαμβάνεται από το αισθητήριο θερμοκρασίας LM35.

Θερμοκρασία

31.74 °C

Επίσης, ο χρήστης μπορεί να δει την τιμή που μεταβάλλεται από το ποτενσιόμετρο

Τάση

1.85 V

#### ▪ Κώδικας servo\_page σελίδας

```
PROGMEM prog_char servo_page[]="
"<html>\n"
"<head>\n"
"<title>Servo</title>\n"
"</head>\n"
"<body bgcolor=\"#33CCFF\">\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"\n"
"<table border=\"0\" cellpadding=\"1\" cellspacing=\"1\" style=\"width: 100%; \">\n"
"<tbody>\n"
"<tr>\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/index\" style=\"color:
#FF0000\">Αρχική</span></strong></a></td>\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/leds\" style=\"color:
#FF0000\">Led</span></strong></a></td>\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/sensor\" style=\"color:
#FF0000\">Potentiometer</span></strong></a></td>\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/switches\" style=\"color:
#FF0000\">Switches</span></strong></a></td>\n"
```

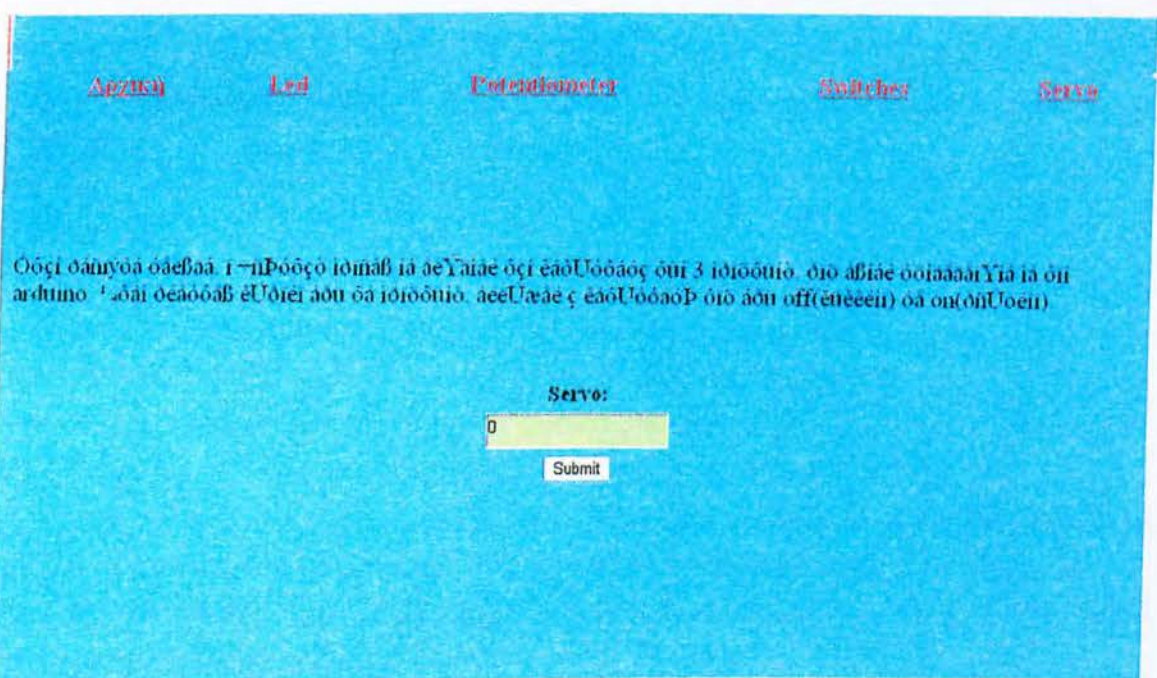
```

"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a
href=\"http://169.254.1.1/servo\"
style=\"color:
#FF0000\">Servo</span></strong></a></td>\n"
"</tr>\n"
"</tbody>\n"
"</table>\n"
"\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"\n"
"<p <span style=\"font-size: 16pt\"> Óôçí ðáñĩýóá óãëßää, ĩ ÷ñßóôçò ïðĩñâß íá
ãëÝãñáé ôçí éáoÛóôáóç ôúí 3 ïðĩðôüíð, ðĩð åßíáé óðíääääĩÝíá ïå ôĩĩ arduino.\n"
"¼ôáí ðéåóôåß êÛðĩéĩ áðü ôá ïðĩðôüíð, áéëÛæåé ç éáoÛóôáóç ð ðĩð áðü off(ëüêêéí) óå
on(ðñÛóéí).</span></p>\n"
"\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\n"
"\n"
"\n"
"<form METHOD=get action=\"http://169.254.1.1/servo\"><strong><span
style=\"font-size: 14pt\">\n"
"\n"
"<table style=\"width: 100%;\">\n"
"</tr>\n"
"<tr>\n"
" <td colspan=\"2\" style=\"text-align:center\">\n"
" <strong><span style=\"font-size: 14pt\">Servo:</span></strong><br />\n"
" </td>\n"
" </tr>\n"
"<tr>\n"
" <td colspan=\"2\" style=\"text-align:center\">\n"
" <input type=\"text\" name=\"v\" value=\"0\" id=\"v\"
style=\"width:160px;height:30px;background-color:#D0F18F\";/>\n"
" </td>\n"
" </tr>\n"
"\n"
"\n"
"<tr>\n"
" <td colspan=\"2\" style=\"text-align:center\">\n"
" <input type=\"submit\" value=\"Submit\" /> \n"
" </td>\n"
" </tr>\n"
"\n"
"</table>\n"

```

```
"\n"
"\n"
"</form>\n"
"\n"
"</body>\n"
"</html>";
```

Σε αυτή τη σελίδα εμφανίζεται το αρχικό μενού με το οποίο μπορούμε να μεταβούμε στις υπόλοιπες σελίδες που επιθυμούμε. Ακόμα μας εμφανίζεται μήνυμα το οποίο μας λέει να πληκτρολογήσουμε έναν αριθμό και πατώντας του κουμπι submit το δημιουργήθηκε με την εντολή Form method και την input type=submit value=submit μετακινεί τον σερβοκινητήρα στην επιθυμητή γωνία. Η εικόνα που θα πάρουμε είναι η εξής.



**Κώδικας sw\_page σελίδας**

```
PROGMEM prog_char sw_1_page[]="\n"
"<html>\n"
"<head>\n"
"<title>Switches</title>\n"
"</head>\n"
"<body bgcolor=\"#33CCFF\">\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;&nbsp;&nbsp;</p>\n"
```

```

"\n"
"<table border=\"0\" cellpadding=\"1\" cellspacing=\"1\" style=\"width: 100%; \">\\n"
"<tbody>\\n"
"<tr>\\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a href=\"http://169.254.1.1/index\" style=\"color:#FF0000\">Αρχική</span></strong></a></td>\\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a href=\"http://169.254.1.1/leds\" style=\"color:#FF0000\">Led</span></strong></a></td>\\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a href=\"http://169.254.1.1/sensor\" style=\"color:#FF0000\">Potentiometer</span></strong></a></td>\\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a href=\"http://169.254.1.1/switches\" style=\"color:#FF0000\">Switches</span></strong></a></td>\\n"
"<td style=\"text-align: center; \"><strong><span style=\"font-size: 16pt\"><a href=\"http://169.254.1.1/servo\" style=\"color:#FF0000\">Servo</span></strong></a></td>\\n"
"</tr>\\n"
"</tbody>\\n"
"</table>\\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\\n"
"\n"
"<p <span style=\"font-size: 16pt\"> Στην παρούσα σελίδα, ο χρήστης μπορεί να ελέγξει την κατάσταση των 3 μπουτόνς, που είναι συνδεδεμένα με τον arduino.\\n"
"Όταν πιεστεί κάποιο από τα μπουτόνς, αλλάζει η κατάστασή του από off(κόκκινο) σε on(πράσινο).</span></p>\\n"
"\n"
"<p style=\"text-align: center\" align=\"center\">&nbsp;</p>\\n"
"<table border=\"1\" cellspacing=\"0\" cellpadding=\"2\" width=\"400\" align=\"center\"><tbody>\\n";
PROGMEM prog_char sw_1_high[]="<tr><td align=center>Button 1</td><td bgcolor=\"#00FF00\" align=center>ON</td></tr>\\n";
PROGMEM prog_char sw_1_low[]="<tr><td align=center>Button 1</td><td bgcolor=\"#FF0000\" align=center>OFF</td></tr>\\n";
PROGMEM prog_char sw_2_high[]="<tr><td align=center>Button 2</td><td bgcolor=\"#00FF00\" align=center>ON</td></tr>\\n";
PROGMEM prog_char sw_2_low[]="<tr><td align=center>Button 2</td><td bgcolor=\"#FF0000\" align=center>OFF</td></tr>\\n";
PROGMEM prog_char sw_3_high[]="<tr><td align=center>Button 3</td><td bgcolor=\"#00FF00\" align=center>ON</td></tr>\\n";

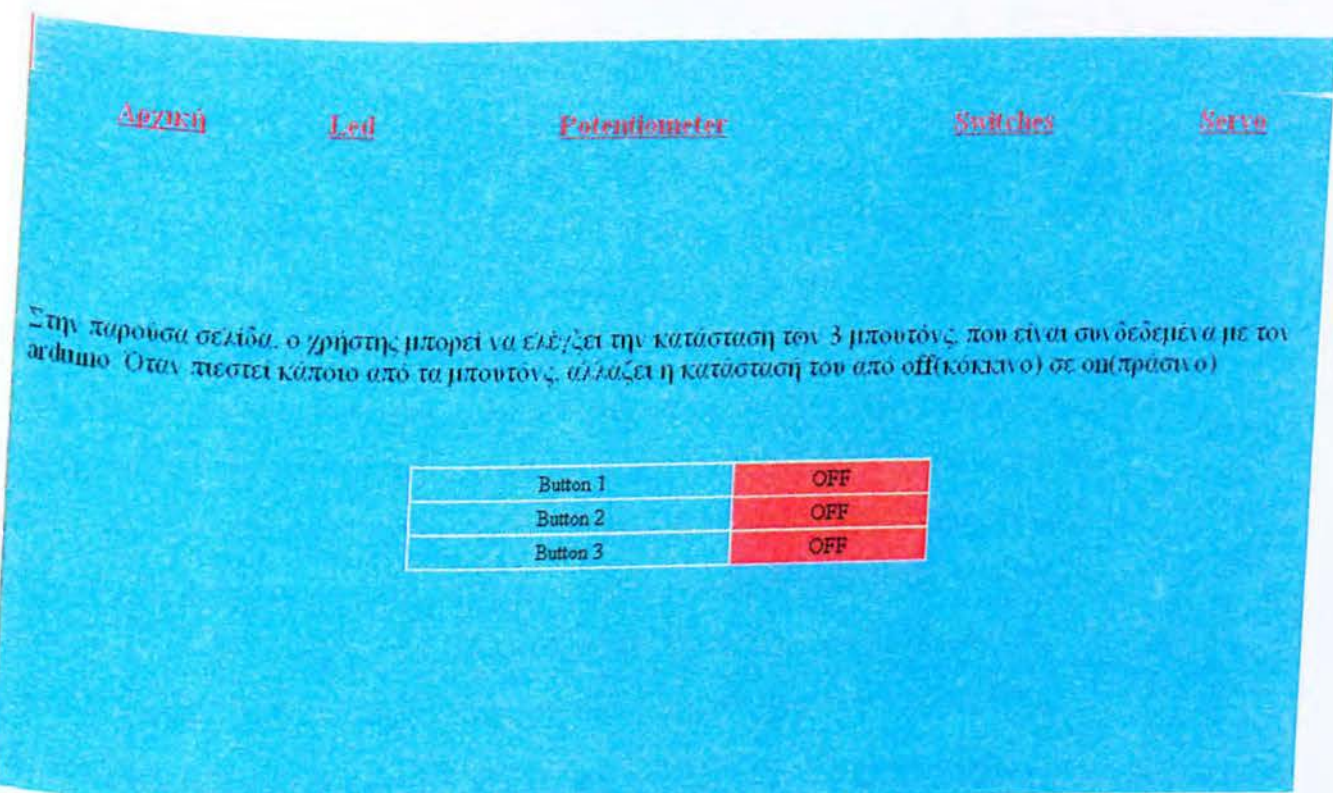
```

```

PROGMEM prog_char sw_3_low[]="<tr><td align=center>Button 3</td><td
bgcolor="#FF0000" align=center>OFF</td></tr>\n";
PROGMEM prog_char sw_2_page[]="</table>";

```

Σε αυτή τη σελίδα ο χρήστης θα μπορεί να ελέγξει την κατάσταση τριών buttons. Έτσι πατώντας ένα button από τη πλακέτα που είναι συνδεδεμένη με τον arduino θα μπορεί να βλέπει την αλλαγή της κατάστασης του αντίστοιχου button στην οθόνη του υπολογιστή του. Έχουμε φτιάξει έναν πίνακα όπου στη μία στήλη έχουμε γράψει το όνομα του κάθε button(button1,button 2,button 3) και στη δεύτερη στήλη αν το button είναι σε κατάσταση ON ή OFF.Η τελική μορφή της σελίδας που θα πάρουμε είναι η εξής



### 3.2 ΚΩΔΙΚΑΣ ARDUINO

Πρώτα επιλέγουμε κάποιες βιβλιοθήκες που χρειαζόμαστε. Συγκεκριμένα τη βιβλιοθήκη servo.h που τη χρειαζόμαστε για τη λειτουργία του σερβοκινητήρα και τη βιβλιοθήκη pgmspace.h. Ορίζουμε μία μεταβλητή με το όνομα string\_byffer\_size που μας βοηθάει γιατί χωρίζει την κάθε σελίδα ανά 300 bytes μέσα στη flash memory του arduino και στη συνέχεια στέλνει ανά 300bytes τη σελίδα στη Ram και στη συνέχεια στο ασύρματο wifi και τέλος στην οθόνη του υπολογιστή. Αυτό γίνεται



γιατί τα στοιχεία δεν χωράνε απευθείας στη μνήμη RAM. Στη συνέχεια ορίζουμε τα ονόματα των μεταβλητών που θα χρησιμοποιήσουμε για το led, για τους διακόπτες κλπ. και ορίζουμε τον αριθμό του pin που αντιστοιχεί σε κάθε μεταβλητή.

```
#include <Servo.h>
#include <avr/pgmspace.h>
#define STRING_BUFFER_SIZE 300
#define POT A1
#define LM A6
#define SW1 12
#define SW2 11
#define SW3 10
#define LED_RED 8
#define LED_GREEN 7
#define LED_BLUE 6
```

Έχουμε φτιάξει έναν πίνακα με το όνομα led[] τριών στοιχείων που κάθε στοιχείο αντιστοιχεί σε ένα χρώμα του led RGB. Μέσα στον πίνακα αποθηκεύουμε την κατάσταση του led δηλαδή αν είναι σβηστό ή ανοιχτό (LOW ή HIGH). Επίσης έχουμε χωρίσει τις σελίδες HTML sensor, switches και leds σε κομμάτια που μας ώστε να μπορούμε να επέμβουμε και να καλούμε κάθε φορά από τη μνήμη flash το κομμάτι που χρειαζόμαστε. Για παράδειγμα εάν θέλουμε να φανεί στην οθόνη μας ότι το button για το κόκκινο χρώμα του led άλλαξε κατάσταση και από red on έγινε red off, τότε θα ζητήσουμε να μας σταλεί το κομμάτι του κώδικα HTML που κάνει μόνο αυτή τη λειτουργία. Δηλώνουμε τις σελίδες HTML σε διαφορετικά αρχεία με την extern και με αυτόν τον τρόπο ο ελεγκτής θα ψάξει στα αρχεία για να τις βρει. Αποθηκεύουμε τις σελίδες HTML στη μνήμη του arduino σε μορφή string.

```
#define RED 0
#define GREEN 1
#define BLUE 2
#define SERVO 9
char buf[100];
unsigned char led[3] = { LOW, LOW, LOW };
char connected=0;
extern PROGMEM prog_char index_page[];
extern PROGMEM prog_char init_page[];
extern PROGMEM prog_char servo_page[];
extern PROGMEM prog_char sensor_page_1[];extern PROGMEM prog_char
sensor_page_2[];extern PROGMEM prog_char sensor_page_3[];
```

```

extern PROGMEM prog_char sw_1_page[];
extern PROGMEM prog_char sw_1_high[];
extern PROGMEM prog_char sw_1_low[];
extern PROGMEM prog_char sw_2_high[];
extern PROGMEM prog_char sw_2_low[];
extern PROGMEM prog_char sw_3_high[];
extern PROGMEM prog_char sw_3_low[];
extern PROGMEM prog_char sw_2_page[];
extern PROGMEM prog_char leds_1_page[];
extern PROGMEM prog_char leds_2_page[];
extern PROGMEM prog_char leds_red_on[];
extern PROGMEM prog_char leds_red_off[];
extern PROGMEM prog_char leds_green_on[];
extern PROGMEM prog_char leds_green_off[];
extern PROGMEM prog_char leds_blue_on[];
extern PROGMEM prog_char leds_blue_off[];

```

Με αυτή τη ρουτίνα ο arduino παίρνει την αποθηκευμένη σελίδα σε μορφή string που βρίσκετε μέσα στη μνήμη flash και τη στέλνει στο ασύρματο wifi για να την εμφανίσει στην οθόνη.

```

void contentPrinter(char *realword)
{
  int b;
  int total = 0;
  int start = 0;
  char buffer[STRING_BUFFER_SIZE];
  int realLen = strlen_P(realword);

  memset(buffer,0,STRING_BUFFER_SIZE);

  while (total <= realLen) {
    // print content
    strncpy_P(buffer, realword+start, STRING_BUFFER_SIZE-1);
    for(b=0; b<STRING_BUFFER_SIZE-1; b++) { Serial3.write(buffer[b]); }

    // more content to print?
    total = start + STRING_BUFFER_SIZE-1;
    start = start + STRING_BUFFER_SIZE-1;
  }
}

```

```
Servo myservo;
void setup()
{
  // start the Ethernet connection and the server:
```

```
  myservo.attach(SERVO);
  myservo.write(90);
```

Επιλέγουμε τα pin να είναι είσοδοι Η έξοδοι αντίστοιχα (INPUT,OUTPUT) και επιλέγουμε το led και οι διακόπτες να είναι σβηστοί δίνοντας HIGH(5 Volt) στο pin.Στη συνέχεια ενεργοποιούμε τις σειριακές θύρες για την επικοινωνία του arduino με τον υπολογιστή.

```
pinMode(LED_RED, OUTPUT);
pinMode(LED_GREEN, OUTPUT);
pinMode(LED_BLUE, OUTPUT);
```

```
digitalWrite(LED_RED, HIGH);
digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_BLUE, HIGH);
```

```
pinMode(SW1, INPUT);
pinMode(SW2, INPUT);
pinMode(SW3, INPUT);
```

```
digitalWrite(SW1, HIGH);
digitalWrite(SW2, HIGH);
digitalWrite(SW3, HIGH);
```

```
pinMode(LM, INPUT);
pinMode(POT, INPUT);
```

```
Serial3.begin(115200);
Serial.begin(115200);
}
```

Εδώ ξεκινάει το κυρίως πρόγραμμα. Αρχικά υπάρχει ένα loop που λέει ότι όσο υπάρχει διαθέσιμη σειριακή σύνδεση να διαβάζει, να αποθηκεύει τα δεδομένα και να τα εκτυπώνει στην οθόνη. Με το `if(c=='*') { i=0; Serial.print("NEW")` ελέγχει αν έχουμε νέο στοιχείο και αν έχουμε εκτυπώνει \*NEW και αποθηκεύει στον πίνακα `buf[]` τα δεδομένα της σειριακής. Στη συνέχεια συγκρίνει τα πέντε πρώτα στοιχεία του

πίνακα buf[] να δει αν είναι ίσα με \*OPEN και εάν είναι τότε εκτυπώνει CONNECTED.

```
void loop()
{
  static int i=0;
  // listen for incoming clients

  boolean currentLineIsBlank = true;

  while(Serial3.available()>0)
  {
    char c = Serial3.read();
    Serial.print(c);
    // if you've gotten to the end of the line (received a newline
    // character) and the line is blank, the http request has ended,
    // so you can send a reply
    if(c=='\n') { i=0; Serial.print("NEW");}
    buf[i]=c;
    i++;
    if(i>98) { i=98; }

    if(strncmp("OPEN",buf,5)==0) { connected=1; i=0; Serial.print("CONNECTED");
    char c = Serial3.read(); Serial.print(c);}
    while(connected==1)
    {
      if(Serial3.available()>0)
      {
        char c = Serial3.read();
        Serial.print(c);
        buf[i]=c;
        i++;
        if(i>98) { i=98; }

```

Εδώ ελέγχει εάν έχουμε δύο συνεχόμενα σύμβολα newline. Αν ισχύει αυτό τότε το browser request έχει τελειώσει και μπορεί ο arduino να επεξεργαστεί τα δεδομένα

```
if(c == '\n' && currentLineIsBlank)
{
  i=0;

```

Σε αυτό το σημείο συγκρίνει πάλι τα στοιχεία για να καταλαβεί ποια σελίδα έχουμε ζητήσει να μας στείλει ο server. Αρχικά ελέγχει να δει αν έχουμε ζητήσει τη σελίδα με

το όνομα index και μας τη στέλνει, αλλιώς συνεχίζει να συγκρίνει τα δεδομένα του πίνακα buf[] με τη σελίδα που του έχουμε ζητήσει για να μας στείλει την κατάλληλη σελίδα.

```
if(strncmp("GET /index",buf,10)==0) { contentPrinter((char*)index_page); }
```

Εάν έχουμε ζητήσει τη σελίδα sensor τότε μετατρέπει τη τιμή που έχει το pin του αισθητηρίου από 0v-5v σε 0-1023 ακέραιο αριθμό και με τον κατάλληλο τύπο τον μετατρέπει σε βαθμούς κελσίου. Με την ίδια λογική μας εμφανίζει και πόσα volt έχουμε στο pin με το ποτενσιόμετρο.

```
else if(strncmp("GET /sensor",buf,11)==0)
{
float temp;
contentPrinter((char*)sensor_page_1);
temp=(5.0 * analogRead(LM) * 100.0)/1024.0;
Serial3.print(temp);
contentPrinter((char*)sensor_page_2);
temp=analogRead(POT)*0.0048828125;
Serial3.print(temp);
contentPrinter((char*)sensor_page_3);
```

Εάν έχουμε ζητήσει τη σελίδα switches μας στέλνει την ανάλογη σελίδα και στη συνέχεια με τη συνθήκη να εμφανίζει τον διακόπτη σε κατάσταση ON όταν έχουμε στο pin SW1 0V(LOW) μας στέλνει και την αντίστοιχη σελίδα που να δείχνει την κατάσταση του διακόπτη σε ON και το αντίστροφο.

```
else if(strncmp("GET /switches",buf,13)==0)
{
contentPrinter((char*)sw_1_page);
if( digitalRead(SW1)==LOW) { contentPrinter((char*)sw_1_high); }
else { contentPrinter((char*)sw_1_low); }
if( digitalRead(SW2)==LOW) { contentPrinter((char*)sw_2_high); }
else { contentPrinter((char*)sw_2_low); }
if( digitalRead(SW3)==LOW) { contentPrinter((char*)sw_3_high); }
else { contentPrinter((char*)sw_3_low); }
contentPrinter((char*)sw_2_page);
}
```

Εάν έχουμε ζητήσει τη σελίδα servo μας τη στέλνει και ελέγχει εάν του έχουμε πληκτρολογήσει κάποια τιμή. Εάν ναι τότε μετατρέπει αυτή τη τιμή σε ακέραιο αριθμό και τη βάζει σε μια μεταβλητή. Στη συνέχεια ελέγχει εάν αυτή η τιμή είναι

μέσα στα επιτρεπτά όρια από 0-180 και αν ισχύει και αυτό τότε κινεί τον σερβοκινητήρα στις ανάλογες μοίρες που πληκτρολογήσαμε.

```
else if(strncmp("GET /servo",buf,10)==0)
{
contentPrinter((char*)servo_page);
if(strncmp("GET /servo?v=",buf,13)==0)
{
char con=1,b=13;
int ang=0;
char temp[10];
while(con==1 && b<20)
{
if(buf[b]==' ')
{
temp[b-13]=0;
ang=atoi(temp);
if(ang>0 && ang<181) { myservo.write(ang); }
con=0;
}
}
Else
{
temp[b-13]=buf[b];
b++;
}
}
}
```

Εάν έχουμε ζητήσει τη σελίδα leds τότε ελέγχει να δει αν έχουμε πατήσει κάποιο από τα buttons. Τότε βάζουμε συνθήκη ώστε το led να ανάβει όταν στο pin του led έχουμε 0volt(LOW) και να σβήνει όταν έχουμε 5volt(HIGH). Ταυτόχρονα με αυτό αλλάζει και την κατάσταση του led στον πίνακα led[] (από HIGH σε LOW και από LOW σε HIGH).

```
else if(strncmp("GET /leds",buf,9)==0)
{
if(strncmp("GET /leds?cmd=1",buf,15)==0)
{
if(led[RED]==HIGH) { digitalWrite(LED_RED,LOW); led[RED]=LOW; }
else { digitalWrite(LED_RED,HIGH); led[RED]=HIGH; }
}
else if(strncmp("GET /leds?cmd=2",buf,15)==0)
```

```

{
if(led[GREEN]==HIGH) { digitalWrite(LED_GREEN,LOW); led[GREEN]=LOW; }
else { digitalWrite(LED_GREEN,HIGH); led[GREEN]=HIGH; }
}
else if(strncmp("GET /leds?cmd=3",buf,15)==0)
{
if(led[BLUE]==HIGH) { digitalWrite(LED_BLUE,LOW); led[BLUE]=LOW; }
else { digitalWrite(LED_BLUE,HIGH); led[BLUE]=HIGH; }
}

```

Σε αυτό το σημείο ελέγχει αν έχουμε εισάγει κάποια τιμή από 0-255 στα πλαίσια τύπου text του led. Εάν έχουμε πληκτρολογήσει τότε αποθηκεύει την τιμή αυτή προσωρινά σε έναν πίνακα temp[] και τη μετατρέπει σε ακέραια μορφή και την αποθηκεύει σε μία μεταβλητή με το όνομα ang. Μετά ελέγχει αν αυτή η τιμή είναι μέσα στα επιτρεπτά όρια από 0-255 και τότε με την analogWRITE() βάζει την αναλογική τιμή στο pin του led. Ουσιαστικά με την αναλογική τιμή διαμορφώνουμε το εύρος του παλμού PWM στο pin του led και έτσι μπορούμε να ελέγξουμε τη φωτεινότητα του led. Για τιμή 255 το led θα είναι σβηστό γιατί η τιμή 255 αντιστοιχίζεται σε 5VOLT και η τιμή μηδέν αντιστοιχίζεται σε 0VOLT . Στη συνέχεια ο κώδικας με την ίδια λογική βρίσκει τις υπόλοιπες τιμές που έχουμε πληκτρολογήσει για το χρώμα πράσινο και μπλε.

```

else if(strncmp("GET /leds?r=",buf,12)==0)
{
char con=1,b=12,te=0;
int ang=0;
char temp[10];
te=0;
while(con==1 && b<60)
{
if(buf[b]=='&')
{
temp[te]=0;
ang=atoi(temp);
if(ang>=0 && ang<256) { analogWrite(LED_RED,ang); }
con=0;
b++;
}
}
Else
{
temp[te]=buf[b];
te++;
}

```

```

b++;
}
}
con=1;
while(con==1 && b<60)
{
if(buf[b]!='=') { con=0; b++; }
else { b++; }
}

```

```

con=1;
te=0;
while(con==1 && b<60)
{
if(buf[b]=='&')
{
temp[te]=0;
ang=atoi(temp);
if(ang>=0 && ang<256) { analogWrite(LED_GREEN,ang); }
con=0;
b++;
}
else
{
temp[te]=buf[b];
te++;
b++;
}
}

```

```

con=1;
te=0;
while(con==1 && b<60)
{
if(buf[b]!='=') { con=0; b++; }
else { b++; }
}

```

```

con=1;
while(con==1 && b<60)
{
if(buf[b]==' ')
{
temp[te]=0;

```



```

ang=atoi(temp);
if(ang>=0 && ang<256) { analogWrite(LED_BLUE,ang); }
con=0;
b++;
}
else
{
temp[te]=buf[b];
te++;
b++;
}
}
}

```

Σε αυτό το σημείο μας εμφανίζει την καταλληλη σελίδα για το button,αν είναι για παράδειγμα RED ON ή RED OFF.Ελέγχει τη κατάσταση του led στον πίνακα led[] και αν είναι led[HIGH] τότε σημαίνει ότι το led είναι σβηστό και μας στέλνει τη σελίδα leds\_red-on που εμφανίζει το button RED ON.

```

contentPrinter((char*)leds_1_page);
if(led[RED]==HIGH) { contentPrinter((char*)leds_red_on); }
else { contentPrinter((char*)leds_red_off); }

if(led[GREEN]==HIGH) { contentPrinter((char*)leds_green_on); }
else { contentPrinter((char*)leds_green_off); }

if(led[BLUE]==HIGH) { contentPrinter((char*)leds_blue_on); }
else { contentPrinter((char*)leds_blue_off); }
contentPrinter((char*)leds_2_page);
}

```

Εάν δεν του έχουμε ζητήσει καμία σελίδα από τις παραπάνω ή εάν του έχουμε ζητήσει να μας στείλει μια ανύπαρκτη σελίδα τότε μας στέλνει τη σελίδα init\_page.

```

else { contentPrinter((char*)init_page); }
Τερματίζεται η σύνδεση και εκτυπώνεται η λέξη CLOSE
connected=0;
Serial.print("CLOSE");
break;
}
if (c == '\n')
{
// you're starting a new line
currentLineIsBlank = true;

```

```
}  
else if (c != '\r')  
{  
// you've gotten a character on the current line  
currentLineIsBlank = false;  
}  
}  
}  
}  
// give the web browser time to receive the data  
delay(1);  
}
```

### 3. ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΟΠΤΙΚΕΣ

Στις μέρες μας, ο χρόνος έχει γίνει πολύτιμος λόγω της πληθώρας εργασιών και υποχρεώσεων που καλείται να εκπληρώσει ο κάθε σύγχρονος άνθρωπος κατά την διάρκεια του εικοσιτετραώρου. Κάθε υποχρέωση, σε όλους σχεδόν τους τομείς της ζωής μας, γίνεται ολοένα και πιο απαιτητική από άποψη χρόνου, ολοένα και πιο εύκολη από άποψη χρήσεως. Υπάρχει δηλαδή μια τάση προς την αυτοματοποίηση των λειτουργιών. Σε αυτό βέβαια, έχει βοηθήσει η ανάπτυξη της τεχνολογίας. Η παρούσα πτυχιακή εργασία, θέλει να αναδείξει την ευκολία, την απλότητα, την ευελιξία, αλλά και την χρησιμότητα που προσφέρει μία συσκευή για τον απομακρυσμένο έλεγχο.

Ο απομακρυσμένος έλεγχος που παρέχει μία τέτοια συσκευή καθώς και η επεξεργασία δεδομένων μέσω μιας ιστοσελίδας μπορεί να βρει εφαρμογές σε πολλούς τομείς.

Σε φωτοβολταϊκές εγκαταστάσεις η υπηρεσία απομακρυσμένου ελέγχου χρησιμοποιείται για να παρακολουθεί το φωτοβολταϊκό σύστημα από βλάβες ή απρόβλεπτες καταστάσεις ώστε να μπορεί να ανταποκριθεί έγκαιρα δίνοντας λύση σε οποιοδήποτε ζήτημα προκύψει. Με την υπηρεσία απομακρυσμένου ελέγχου θα επιτυγχάνεται η 24ωρη επίβλεψη και παρακολούθηση του φωτοβολταϊκού συστήματος όπου θα καταγράφεται η απόδοσή του. Η υπηρεσία αυτή επιτρέπει στον χρήστη να ανταποκριθεί έγκαιρα με στόχο την εξασφάλιση της βιωσιμότητας του συστήματος μακροπρόθεσμα.

Στη καθημερινότητά μας μία τέτοια υπηρεσία μπορεί να δώσει τη δυνατότητα στο χρήστη να μπορεί να οπλίσει και να αφοπλίσει το συναγερμό του σπιτιού του από μακριά. Η ιδιότητα αυτή είναι ιδιαίτερα χρήσιμη. Με τον τρόπο αυτό για παράδειγμα όταν ο χρήστης βρίσκεται στην εργασία του ή σε διακοπές μπορεί να επιτρέψει την είσοδο στο σπίτι του σε κάποιον συγγενή ή οικιακή βοηθό αποπλίζοντας το συναγερμό από το κινητό του ή από κάποια συσκευή που έχει πρόσβαση στο διαδίκτυο. Επίσης ο χρήστης μπορεί να ενεργοποιεί και να απενεργοποιεί το σύστημα θέρμανσης από μακριά ώστε το σπίτι να αρχίσει να θερμαίνεται μέχρι την ώρα που θα επιστρέψει σε αυτό.

Ακόμα μία τέτοια εφαρμογή μπορεί να βρει εφαρμογή και στο αυτοκίνητο. Ο χρήστης θα έχει τη δυνατότητα μέσω της συσκευής που θα συνδέεται στο διαδίκτυο να ελέγχει τη ποσότητα των καυσίμων και τη διάρκεια ζωής των λαδιών εύκολα και γρήγορα από απόσταση κερδίζοντας με αυτόν τον τρόπο χρόνο και αποφεύγοντας ανεπιθύμητα αποτελέσματα.

Στις βιομηχανίες ο απομακρυσμένος έλεγχος είναι πολύ σημαντικός. Για παράδειγμα ο υπεύθυνος μιας μονάδας παραγωγής θα μπορεί να ελέγχει από τον υπολογιστή του τη λειτουργία κάποιων μηχανών καθώς και θα μπορεί να επέμβει από απόσταση στην ενεργοποίηση ή στην απενεργοποίηση κάποιων από αυτών χωρίς να χρειάζεται να στείλει κάποιον από τους εργάτες διακινδυνεύοντας τη ζωή του. Επίσης θα μπορεί να ελέγχει αν σε κάποια μονάδα η θερμοκρασία είναι πολύ υψηλή που σημαίνει κίνδυνος πυρκαγιάς και έτσι να μπορεί να επέμβει έγκαιρα ενεργοποιώντας πάλι από απόσταση το σύστημα συναγερμού και το σύστημα πυρόσβεσης.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [www.arduinoohnev.blogspot.com](http://www.arduinoohnev.blogspot.com)
- Αγγελόπουλος Δ. Ι., (2006) *Σημειώσεις μαθήματος Δίκτυα Υπολογιστών*
- Αγγελόπουλος Δ. Ι., (2006) *Σημειώσεις μαθήματος Διαδικτυακά Πρωτόκολλα*
- [www.wiring.org](http://www.wiring.org)
- <http://el.wikipedia.org/wiki/Arduino>
- Banzi, M., (2009). *Getting Started with Arduino*, 1st edition
- Evans, B., (2007), *Arduino Programming Notebook*, 2st edition,
- <http://arduino.cc/it/Main/ArduinoBoardMega2560>
- <http://el.wikipedia.org/wiki/TCP/IP>
- Καλουκάση Ν.,(2002). *Εφαρμογές ψηφιακής συνδυαστικής λογικής*. β' έκδοση, Αθήνα, ISBN 960- 90500- 2- 6
- Κυτάγιας Δ., *Προγραμματισμός*, Σύγχρονη εκδοτική, Αθήνα, ISBN 960-7344- 77- 4
- Μπούσλης Π., Πολίτης Γ., (1995). *Ηλεκτροτεχνία Ι*, εκδόσεις Μ & Ρ, Αθήνα
- Βάκαλη Α., Γιαννόπουλος Η., Ιωαννίδης Ν., Κοιλιάς Χ., Μάλαμας Κ., Μανόπουλος Ι., Πολίτης Π., *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*, οργανισμός εκδόσεων διδακτικών βιβλίων, Αθήνα
- <http://arduino.cc/en/Reference/HomePage>
- <http://www.sparkfun.com/products/11047>