

**Ανάπτυξη διακομιστή διαδικτύου σε μικροελεγκτή
για ασύρματο σύστημα μετάδοσης ψηφιακής
πληροφορίας**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΑΡΛΑΝΤΖΑΣ ΑΘΑΝΑΣΙΟΣ
Ηλεκτρονικός Μηχανικός

Επίβλεψη: Αναπληρωτής Καθηγητής Ν. Μάνθος
Εργαστήριο Φυσικής Υψηλών Ενεργειών

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΙΣ ΣΥΓΧΡΟΝΕΣ
ΗΛΕΚΤΡΟΝΙΚΕΣ ΤΕΧΝΟΛΟΓΙΕΣ**

ΤΜΗΜΑ ΦΥΣΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΙΩΑΝΝΙΝΑ, ΟΚΤΩΒΡΙΟΣ 2009



ΒΙΒΛΙΟΘΗΚΗ
ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΙΩΑΝΝΙΝΩΝ



026000321430



Περίληψη

Στην παρούσα διπλωματική εργασία περιγράφεται μέρος ενός συστήματος μέτρησης διαφόρων παραμέτρων ποιότητας του νερού.

Το σύστημα ελέγχου της ποιότητας του νερού περιλαμβάνει το σύστημα που εκτελεί τις απαραίτητες λειτουργίες έτσι ώστε με τους κατάλληλους αισθητήρες να γίνουν οι μετρήσεις στο νερό και να μετατραπούν σε ψηφιακή μορφή, το ασύρματο σύστημα μετάδοσης WiFi που μεταφέρει μέσω αναμεταδοτών τις επεξεργασμένες πληροφορίες στον τελικό απομακρυσμένο ηλεκτρονικό υπολογιστή. Ενδιάμεσα από τα δύο συστήματα ένας διακομιστής διαδικτύου συλλέγει τα ψηφιακά αποτελέσματα των μετρήσεων, τα επεξεργάζεται, τα αποθηκεύει, τα στέλνει μέσω του ασύρματου δικτύου WiFi και τα αποθηκεύει σ' ένα απομακρυσμένο ηλεκτρονικό υπολογιστή.

Στην παρούσα διπλωματική περιγράφεται η ανάπτυξη του διακομιστή διαδικτύου.

Συγκεκριμένα χρησιμοποιήθηκε ένας διακομιστής διαδικτύου (web server) βασισμένος σε μικροελεγκτή. Το σύστημα του διακομιστή διαδικτύου έχει τη δυνατότητα να έχει αποθηκευμένη μια αρχική σελίδα καθώς και άλλες σελίδες HTML και να ελέγχεται μέσω του διαδικτύου από έναν απομακρυσμένο υπολογιστή.

Επίσης αναπτύχθηκε ένα σύστημα ρολογιού πραγματικού χρόνου βασισμένο σε μικροελεγκτή και ένα σύστημα που διασυνδέει λογικά το σύστημα που παίρνει τις μετρήσεις από τους αισθητήρες με το σύστημα που παρέχει τον πραγματικό χρόνο και με το διακομιστή διαδικτύου.



Abstract

In this McS thesis refers to a part of a system for measuring water quality parameters.

The water quality monitoring system is composed of a sensor read out system that executes all the necessary functions to read the water quality sensors and digitize the measurements and a WiFi system that transfers the processed information through WiFi repeaters to a remote computer. Between the two systems a web server collects the data, processes them, stores them, sends them through WiFi and finally store them again in a remote computer.

This document describes the last part of the water quality monitoring system.

In particular a pcb web server has been used based on a microcontroller. The web server system has the feature of storing a top and additional pages and can be controlled by a remote computer.

In addition a microcontroller based real time clock system has been developed together with a microcontroller based interface system between the webserver, the real time clock system and the sensor read out system.



Περιεχόμενα

Περίληψη.....	3
Abstract.....	3
Περιεχόμενα.....	5
Πρόλογος.....	11

1. Εισαγωγή

1.1 Συστήματα και πρωτόκολλα μεταφοράς δεδομένων.....	13
1.1.1 TCP/IP.....	15
1.1.2 Τοπολογίες δικτύων.....	18
1.1.3 Πρωτόκολλα Δικτύων.....	19
1.2 Περιγραφή του ασύρματου συστήματος μετάδοσης φυσικών πληροφοριών στο διαδίκτυο.....	24
1.3 Τοπολογία του συστήματος.....	25

2. Διακομιστής διαδικτύου - webserver

2.1 Γενικές πληροφορίες για τον ATMEGA128.....	29
2.2 Αριθμητική λογική μονάδα.....	33
2.3 Καταχωρητές.....	34

3. Το υλικό του συστήματος του διακομιστή διαδικτύου για την μετάδοση της πληροφορίας



3.1 Οι λειτουργίες της πλακέτας του διακομιστή διαδικτύου.....	37
3.2 Η πλακέτα χρονομέτρου πραγματικού χρόνου (real time clock) και διασύνδεσης με τον διακομιστή διαδικτύου και τους αισθητήρες.....	40
3.3 Διασύνδεση του συστήματος του διακομιστή διαδικτύου με τον αναγνώστη δεδομένων των αισθητήρων.....	48

4. Το λογισμικό του συστήματος του διακομιστή διαδικτύου

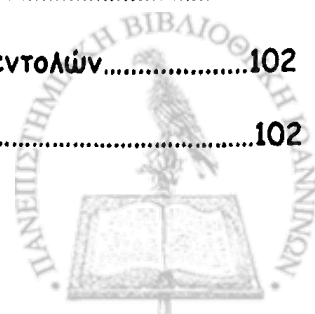
4.1 Λογισμικό για τον προγραμματισμό του διακομιστή διαδικτύου.....	51
4.2 Αρχική σελίδα του διακομιστή διαδικτύου σε γλώσσα html.....	66
4.2.1 Το πρόγραμμα της αρχικής σελίδας index.html.....	67
4.2.2 Η συνάρτηση CpuPortStatus του προγράμματος cgi.c.....	69
4.3 Το πρόγραμμα του χρονομέτρου πραγματικού χρόνου (real time clock).....	78
4.4 Πρόγραμμα της πλακέτας διασύνδεσης του διακομιστή διαδικτύου με την πλακέτα χρονομέτρου πραγματικού χρόνου - και την πλακέτα ανάγνωσης δεδομένων αισθητήρων....	86

5. Αποτίμηση του συστήματος του διακομιστή διαδικτύου

5.1 Γενικά.....	97
5.2 Αξιοπιστία του συστήματος.....	98
5.3 Αξιοπιστία του ασύρματου συστήματος μετάδοσης φυσικών πληροφοριών στο διαδίκτυο.....	99

6. Συμπεράσματα-Επεκτάσεις

6.1 Συμπεράσματα για το σύστημα του διακομιστή διαδικτύου.....	101
6.2 Σελίδα HTML μελλοντικής επέκτασης λειτουργιών και αποστολής εντολών.....	102
6.3 Επέκταση του διακομιστή διαδικτύου σε περισσότερες εφαρμογές.....	102



...από την ...

...από την ...

...από την ...

...από την ...

...από την ...

...από την ...



Πρόλογος

Το αρχικό ερέθισμα για τη παρούσα διπλωματική εργασία, ήταν η ευαισθησία τόσο των τριών μεταπτυχιακών φοιτητών που ανέλαβαν να την πραγματοποιήσουν όσο και του υπεύθυνου καθηγητή, για το περιβάλλον, την μόλυνση και γενικότερα την προφύλαξη του πλανήτη στον οποίο κατοικούμε έτσι ώστε να είμαστε αισιόδοξοι για ένα καλύτερο και «καθαρότερο» μέλλον.

Μέσα από το μεταπτυχιακό αυτό πρόγραμμα μας δόθηκε η ευκαιρία να αποδείξουμε την ευαισθησία μας αυτή, καθώς αναλάβαμε την ευθύνη να δημιουργήσουμε και να κατασκευάσουμε τόσο σε θεωρητικό επίπεδο όσο και σε εφαρμοσμένο, ένα αρκετά πολύπλοκο σύστημα, το οποίο να έχει πολλές δυνατότητες εφαρμογής σε μετρήσεις διαφόρων παραμέτρων του νερού αλλά και πολλές δυνατότητες επέκτασης και αναβάθμισης.

Εξαιτίας της πολυπλοκότητας, της δυσκολίας και το γεγονός ότι όλο το σύστημα θα έπρεπε να γίνει τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο, δηλαδή προϋποθέτει την εγκατάσταση κεραιών, αναμεταδοτών, αισθητήρων και άλλων αναγκαίων συστημάτων ώστε να λειτουργήσει, η διπλωματική αυτή χωρίστηκε σε τρία επιμέρους κομμάτια, το οποία ανατέθηκαν σε τρεις διαφορετικούς φοιτητές.

Φυσικά, για να δουλέψει ολοκληρωμένα το όλο σύστημα, απαιτήθηκε η άμεση συνεργασία και των τριων, έτσι ώστε να μπορέσουμε να βρούμε λύσεις σε διάφορα προβλήματα που προέκυπταν στη πορεία αλλά παράλληλα αυτή η λύση να είναι άμεσα αποδεκτή και από τους άλλους, καθώς η κάθε διπλωματική ήταν απλά ένα κομμάτι του συνόλου και θα έπρεπε να υπάρχει πλήρη συμβατότητα και συνεργασία έτσι ώστε να έχει ένα λογικό και ικανοποιητικό αποτέλεσμα.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τους ανθρώπους που συνέβαλλαν με διάφορους τρόπους στην πραγματοποίηση της παρούσας διπλωματικής εργασίας.

Πιο συγκεκριμένα τους δύο μεταπτυχιακούς φοιτητές, Πέτρογλου Θεόδωρο και Λούκα Νικήτα, η βοήθεια των οποίων ήταν αρκετά χρήσιμη στην πραγματοποίηση όλων των εξωτερικών εργασιών και όχι μόνο, τον επιβλέποντα της διπλωματικής εργασίας καθηγητή κ. Νικόλαο Μάνθο, μέλος του Εργαστηρίου Φυσικής Υψηλών



Ενεργειών, τον κ. Στάθη Μπλέτσα, μέλος ΕΤΕΠ του εργαστηρίου, την εξεταστική επιτροπή καθώς και όλο το υπόλοιπο προσωπικό που συνέβαλε με οποιοδήποτε τρόπο στην πραγματοποίησή της.



1. Εισαγωγή

1.1 Συστήματα και πρωτόκολλα μεταφοράς δεδομένων

Οι άνθρωποι έχουν την ανάγκη της επικοινωνίας. Ως επακόλουθο αυτής της ανάγκης έγιναν πολλές και ποικίλες εφευρέσεις που βοήθησαν στο πέρασμα του χρόνου έτσι ώστε οι επικοινωνίες να βελτιώνονται συνεχώς. Κάποιες από αυτές βοήθησαν στην άμεση επικοινωνία των ανθρώπων με ήχο, όπως τηλέφωνο, άλλες με εικόνα και ήχο, όπως κινητά τηλέφωνα και άλλες συνδιάζοντας πολλούς και διάφορους τρόπους όπως ήχο, εικόνα, μηνύματα και δεδομένα. Για να μπορέσει να αναπτυχθεί όμως ένας τρόπος που συνδιάζει όλα τα παραπάνω και οφείλεται κατά κύριο λόγο στους ηλεκτρονικούς υπολογιστές, έπρεπε να εφευρευθούν διάφορα συστήματα και πρωτόκολλα μεταφοράς δεδομένων. Στη συνέχεια αυτού του συγγράμματος αναφέρονται και αναλύονται κάποια από αυτά. Αρχικά αναφέρονται τα περισσότερα πρωτόκολλα περιληπτικά.

Το TCP/IP[1] αποτελεί μια συλλογή από διαφορετικά πρωτόκολλα επικοινωνίας, τα οποία βασίζονται στα δύο σημαντικότερα και αρχικά πρωτόκολλα, το TCP και το IP.

Το πρωτόκολλο TCP χρησιμοποιείται για τη μετάδοση των δεδομένων από μια εφαρμογή στο δίκτυο. Το TCP είναι υπεύθυνο για τη διάσπαση των δεδομένων σε μικρότερα πακέτα IP πριν αυτά αποσταλούν καθώς και για την αντίστοιχη συναρμολόγηση των πακέτων όταν αυτά φθάσουν στον προορισμό τους.

Το πρωτόκολλο IP φροντίζει για την επικοινωνία με τους άλλους υπολογιστές και είναι υπεύθυνο για την αποστολή και τη λήψη (δρομολόγηση) των πακέτων δεδομένων μέσω του διαδικτύου.

Το πρωτόκολλο HTTP[2] φροντίζει για την επικοινωνία ανάμεσα σ' έναν διακομιστή διαδικτύου (Web server) και έναν περιηγητή (Web browser). Χρησιμοποιείται για την αποστολή των αιτήσεων (requests) από έναν πελάτη διαδικτύου (Web client) (browser) σ' έναν διακομιστή διαδικτύου (Web server) καθώς και για την επιστροφή του περιεχομένου, δηλ. των ιστοσελίδων από τον server πίσω στον χρήστη.



Το πρωτόκολλο HTTP[2] φροντίζει για την ασφαλή επικοινωνία ανάμεσα σ' έναν διακομιστή διαδικτύου (Web server) και έναν περιηγητή διαδικτύου (Web browser). Χειρίζεται τυπικά τις συναλλαγές που γίνονται με πιστωτικές κάρτες καθώς και μ' άλλα ευαίσθητα δεδομένα.

Το πρωτόκολλο SSL[2] χρησιμοποιείται για την κωδικοποίηση (κρυπτογράφηση) των δεδομένων για να είναι έτσι ασφαλής η μεταφορά τους.

Το πρωτόκολλο SMTP[2] χρησιμοποιείται για την αποστολή των μηνυμάτων ηλεκτρονικού ταχυδρομείου από τον χρήστη σ' έναν e-mail server.

Το πρωτόκολλο IMAP[2] χρησιμοποιείται για την αποθήκευση και την ανάκτηση των μηνυμάτων ηλεκτρονικού ταχυδρομείου.

Το πρωτόκολλο POP[2] χρησιμοποιείται για το κατέβασμα των μηνυμάτων ηλεκτρονικού ταχυδρομείου από έναν διακομιστή ηλεκτρονικού ταχυδρομείου (e-mail server) στον προσωπικό υπολογιστή του χρήστη.

Το πρωτόκολλο FTP[2] φροντίζει για την αποστολή των αρχείων (ανέβασμα και κατέβασμα) ανάμεσα σε υπολογιστές και κυρίως ανάμεσα σ' έναν FTP server και έναν απλό προσωπικό υπολογιστή.

Το πρωτόκολλο NTP χρησιμοποιείται για τον συγχρονισμό του ρολογιού ανάμεσα στους υπολογιστές.

Το πρωτόκολλο DHCP χρησιμοποιείται για την κατανομή των δυναμικών IP διευθύνσεων στους υπολογιστές ενός δικτύου.

Το πρωτόκολλο SNMP χρησιμοποιείται για τη διαχείριση (administration) των δικτύων υπολογιστών.

Το πρωτόκολλο LDAP χρησιμοποιείται για τη συλλογή πληροφοριών σχετικά με χρήστες και διευθύνσεις e-mail από το διαδίκτυο.

Το πρωτόκολλο ICMP φροντίζει για την αντιμετώπιση των λαθών στο δίκτυο.

Το πρωτόκολλο ARP χρησιμοποιείται από το IP για να βρεθεί η διεύθυνση υλικού μιας κάρτας δικτύου υπολογιστή που βασίζεται στην διεύθυνση IP.



Το πρωτόκολλο RARP χρησιμοποιείται από το IP για να βρεθεί η διεύθυνση IP που βασίζεται στη διεύθυνση υλικού μιας κάρτας δικτύου υπολογιστή.

Το πρωτόκολλο BOOTP χρησιμοποιείται για το ξεκίνημα των υπολογιστών από το δίκτυο.

Το πρωτόκολλο RPTP χρησιμοποιείται για την εγκαθίδρυση μιας σύνδεσης ανάμεσα σε ιδιωτικά δίκτυα.

1.1.1 TCP/IP

Ένα πρωτόκολλο επικοινωνίας υπολογιστών είναι η περιγραφή των κανόνων που πρέπει να ακολουθούν οι υπολογιστές για να μπορούν να επικοινωνούν μεταξύ τους.

Το TCP/IP είναι το πρωτόκολλο επικοινωνίας για την επικοινωνία ανάμεσα σε υπολογιστές που είναι συνδεδεμένοι στο παγκόσμιο δίκτυο υπολογιστών (web). Τα αρχικά TCP/IP αναφέρονται σε δύο από τα σημαντικότερα πρωτόκολλα που χρησιμοποιούνται στο διαδίκτυο, δηλ. στο TCP και στο IP, και όχι σ' όλα τα πρωτόκολλα του διαδικτύου.

Το *Transmission Control Protocol / Internet Protocol* (TCP/IP) δηλ. *Πρωτόκολλο Ελέγχου Εκπομπής/Πρωτόκολλο του Διαδικτύου* ορίζει το πώς οι ηλεκτρονικές συσκευές, όπως είναι οι υπολογιστές, θα πρέπει να συνδέονται στο Διαδίκτυο και πώς πρέπει να μεταδίδονται τα δεδομένα μεταξύ τους.

Στο TCP/IP υπάρχουν πολλά πρωτόκολλα για τον χειρισμό της επικοινωνίας των δεδομένων:

- Το *TCP (Transmission Control Protocol)*, για επικοινωνία ανάμεσα σε εφαρμογές
- Το *UDP (User Datagram Protocol)*, για απλή επικοινωνία ανάμεσα σε εφαρμογές
- Το *IP (Διαδίκτυο Protocol)*, για επικοινωνία ανάμεσα σε υπολογιστές.
- Το *ICMP (Διαδίκτυο Control Message Protocol)*, για λάθη και στατιστικές.



- Το *DHCP* (*Dynamic Host Configuration Protocol*), για δυναμική διευθυνσιοδότηση (*dynamic addressing*).

Το πρωτόκολλο TCP προορίζεται για επικοινωνία ανάμεσα σε εφαρμογές. Όταν μια εφαρμογή πρέπει να επικοινωνήσει με μια άλλη εφαρμογή μέσω του TCP, στέλνει μια αίτηση επικοινωνίας. Αυτή η αίτηση πρέπει να σταλεί σε μια συγκεκριμένη διεύθυνση. Αφού επιβεβαιωθεί η επικοινωνία (*handshake*) ανάμεσα στις δύο εφαρμογές, το TCP καθιερώνει μια ταυτόχρονη αμφίπλευρη επικοινωνία ανάμεσα στις δύο εφαρμογές.

Η ταυτόχρονη αμφίπλευρη επικοινωνία καταλαμβάνει τη γραμμή επικοινωνίας ανάμεσα στους δύο υπολογιστές μέχρι αυτή να κλείσει από μια από τις δύο εφαρμογές. Το πρωτόκολλο UDP είναι παρόμοιο με το TCP αλλά είναι απλούστερο και λιγότερο αξιόπιστο.

Το πρωτόκολλο IP προορίζεται για επικοινωνία ανάμεσα σε υπολογιστές. Το IP είναι ένα χωρίς σύνδεση (*connection-less*) πρωτόκολλο επικοινωνίας. Δηλαδή δεν καταλαμβάνει τη γραμμή επικοινωνίας ανάμεσα σε δύο επικοινωνούντες υπολογιστές. Μ' αυτόν τον τρόπο το IP ελαττώνει την ανάγκη για γραμμές δικτύωσης. Έτσι, η κάθε γραμμή επικοινωνίας μπορεί να χρησιμοποιηθεί για επικοινωνία ανάμεσα σε πολλούς διαφορετικούς υπολογιστές την ίδια στιγμή.

Με το IP, τα μηνύματα ή και άλλα δεδομένα διασπώνται σε μικρά ανεξάρτητα *πακέτα* και στέλνονται ανάμεσα στους υπολογιστές μέσω του διαδικτύου. Το IP είναι υπεύθυνο για τη δρομολόγηση του κάθε πακέτου μέχρι να φθάσει στον τελικό του προορισμό. Η επικοινωνία μέσω του IP είναι σαν να στέλνουμε ένα μακρυσκελές γράμμα σαν ένα μεγάλο πλήθος από μικρές καρτ ποστάλ, όπου η καθεμία θα βρει την δική της, συχνά διαφορετική διαδρομή, μέχρι τον παραλήπτη.

Όπως ήδη αναφέρθηκε, το πρωτόκολλο TCP/IP αποτελείται από τα πρωτόκολλα TCP και το IP, τα οποία συνεργάζονται μεταξύ τους. Το TCP φροντίζει για την επικοινωνία ανάμεσα στο λογισμικό της εφαρμογής, δηλ. τον περιηγητή, και το λογισμικό του δικτύου.

Το IP φροντίζει για την επικοινωνία με τους άλλους υπολογιστές. Το TCP είναι υπεύθυνο για τη διάσπαση των δεδομένων σε πακέτα IP πριν αυτά αποσταλούν καθώς



και για την αντίστοιχη συναρμολόγησή τους όταν αυτά φθάσουν στον προορισμό τους. Το IP είναι υπεύθυνο για την αποστολή των πακέτων στον παραλήπτη. Το TCP/IP χρησιμοποιεί αριθμούς των 32 bits για να αποδώσει μια διεύθυνση σ' έναν υπολογιστή.

Ο κάθε υπολογιστής πρέπει να διαθέτει μια διεύθυνση IP πριν μπορέσει να συνδεθεί στο διαδίκτυο και το κάθε πακέτο IP πρέπει να διαθέτει μια διεύθυνση πριν μπορέσει να αποσταλεί σ' έναν άλλον υπολογιστή.

Για παράδειγμα, η παρακάτω διεύθυνση IP *194.63.237.7* ανήκει στο site *dide.flo.sch.gr*. Ένα άλλο παράδειγμα διεύθυνσης IP είναι το εξής : *83.235.160.51*. Το TCP/IP χρησιμοποιεί 4 ακέραιους αριθμούς για να αποδώσει μια διεύθυνση σ' έναν υπολογιστή. Έτσι, ο κάθε υπολογιστής θα πρέπει να διαθέτει μια μοναδική διεύθυνση με 4 ακέραιους αριθμούς. Οι αριθμοί αυτοί έχουν πάντα τιμές ανάμεσα στο 0 και το 255. Οι διευθύνσεις γράφονται με τους 4 αριθμούς χωρισμένους με τελείες, ως εξής : *193.106.1.51*.

Το TCP/IP χρησιμοποιεί διευθυνσιοδότηση (addressing) των 32 bits, δηλαδή 4 bytes και ένα byte μπορεί να περιέχει 256 διαφορετικές τιμές.

Επειδή είναι πολύ δύσκολο να θυμόμαστε τους αριθμούς και η χρήση των ονομάτων είναι σαφώς ευκολότερη, τα ονόματα που χρησιμοποιούνται και που αντιστοιχούν στις διευθύνσεις TCP/IP αποκαλούνται domain names (ονόματα χώρου ή ονόματα περιοχής). Για παράδειγμα, το *uoi.gr* είναι ένα domain name (όνομα χώρου). Όταν αποκτάμε πρόσβαση σ' έναν δικτυακό τόπο, όπως είναι το <http://www.uoi.gr>, για παράδειγμα, το όνομα αυτό μεταφράζεται σε μια σειρά από αριθμούς από μια διαδικασία που αποκαλείται DNS (Domain Name Server) ή σύστημα ονοματοδοσίας.

Σ' όλο τον κόσμο, υπάρχει ένας μεγάλος αριθμός από εξυπηρετητές ονομάτων τομέων (DNS servers) που είναι συνδεδεμένοι στο Διαδίκτυο. Οι DNS servers είναι υπεύθυνοι για τη μετάφραση των domain names σε αντίστοιχες διευθύνσεις του TCP/IP καθώς και για την αλληλοενημέρωσή τους με τις καινούργιες καταχωρήσεις ή και τις τροποποιήσεις και διαγραφές των domain names. Όταν καταχωρείται στο μητρώο ένα καινούργιο domain name μαζί με μια διεύθυνση TCP/IP, οι DNS servers σ' όλο τον κόσμο ενημερώνονται με τα καινούργια στοιχεία.



Όταν ένα πακέτο του IP στέλνεται από έναν υπολογιστή, αυτό φθάνει σ' έναν δρομολογητή IP. Ο δρομολογητής IP είναι υπεύθυνος για τη δρομολόγηση του πακέτου μέχρι τον τελικό του προορισμό, είτε απευθείας είτε μέσω ενός άλλου δρομολογητή.

Η διαδρομή που θα ακολουθήσει ένα πακέτο μπορεί να είναι διαφορετική από αυτήν που θα ακολουθήσουν άλλα πακέτα του ίδιου μηνύματος. Ο δρομολογητής είναι υπεύθυνος για τη σωστή διευθυνσιοδότηση των πακέτων ανάλογα και με τον όγκο της κυκλοφορίας, τα τυχόν λάθη που θα λάβουν χώρα στο δίκτυο ή και από άλλες παραμέτρους.

1.1.2 Τοπολογίες Δικτύων

Τοπολογία δικτύου είναι η γεωμετρική και φυσική διάταξη των καλωδίων και των συσκευών που συνδέονται με το δίκτυο. Η διάταξη μπορεί να είναι ένας δίαυλος, ένας δακτύλιος ή ένας αστέρας. Αν το δίκτυο είναι πολύ μεγάλο, ενδέχεται να υπάρχουν και τα τρία είδη σε διάφορες περιοχές του δικτύου.

Στην τοπολογία διαύλου (Bus)[3] ολόκληρο το δίκτυο στηρίζεται σε ένα καλώδιο, το οποίο συνήθως καλείται ραχοκοκαλιά (Backbone). Όλοι οι κόμβοι συνδέονται με το κεντρικό καλώδιο είτε απευθείας μέσω συνδέσμων T είτε μέσω άλλων καλωδίων. Τα καλώδια είναι συνήθως ομοαξονικά, ενώ η ραχοκοκαλιά είναι τερματισμένη στα δύο άκρα της, προκειμένου το σήμα να απορροφάται και να μην ανακλάται δημιουργώντας επιπλέον θόρυβο. Το δίκτυο της τοπολογίας διαύλου είναι εύκολο στην εγκατάσταση και σχετικά φθηνό. Αλλά, αν σε ένα σημείο του διαύλου διακοπεί η επικοινωνία, τότε καταρρέει όλο το δίκτυο, ενώ δύσκολα εντοπίζεται η βλάβη. Αυτή η τοπολογία συνδυάζεται συνήθως με τα πρωτόκολλα 10Base5 και 10Base2 που περιγράφονται σε επόμενη παράγραφο.

Στην τοπολογία δακτυλίου (Ring)[3] όλοι οι κόμβοι συνδέονται έτσι, ώστε να σχηματίζουν ένα κύκλο. Τα δεδομένα ακολουθούν πάντα την ίδια φορά και κάθε κόμβος αναπαράγει το σήμα. Αν λοιπόν κάποια συσκευή σταματήσει να λειτουργεί, τότε καταρρέει ολόκληρο το δίκτυο. Την τοπολογία αυτήν την συναντάμε σε πρωτόκολλα Token Ring και FDDI και χρησιμοποιείται σε δίκτυα με πολλούς κόμβους



όπου απαιτείται υψηλή ταχύτητα. Μειονέκτημά της είναι το υψηλό κόστος αφού για την δημιουργία ενός δακτυλίου με πρωτόκολλο FDDI χρειάζεται η αγορά οπτικών ινών και ανάλογες κάρτες δικτύου (NIC).

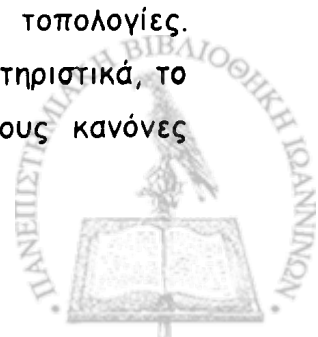
Στην τοπολογία αστέρα (star)[3], κάθε κόμβος έχει το καλώδιό του το οποίο συνδέεται με μία κεντρική συσκευή που καλείται Hub. Το Hub είναι συνήθως μία συσκευή με εισόδους RJ-45 όπου συνδέονται οι κόμβοι, ενώ συνήθως έχουν και λαμπάκια τα οποία δείχνουν την κατάσταση λειτουργίας τους. Το πλεονέκτημα της τοπολογίας είναι ότι, αν κάποιος κόμβος τεθεί εκτός λειτουργίας, δεν καταρρέει το δίκτυο. Εξίσου εύκολη είναι και η επέκταση του δικτύου με την προσθήκη παραπάνω κόμβων, αφού το δίκτυο δεν χρειάζεται να σταματήσει να λειτουργεί. Το μειονέκτημα εδώ είναι το αυξημένο κόστος λόγω των περισσότερων καλωδίων και των συσκευών Hub.

Τέλος υπάρχουν και κάθε συνδιασμός των παραπάνω μεθόδων που αναφέρθηκαν, υβριδική τοπολογία[3]. Ένα παράδειγμα είναι ο συνδυασμός αστέρα-διαύλου, όπου εκεί πολλοί μικροί αστέρες συνδέονται με ένα κεντρικό δίαυλο.

1.1.3 Πρωτόκολλα Δικτύων

Δίκτυο είναι ένα γκρουπ από δύο ή περισσότερους υπολογιστές που είναι συνδεδεμένοι μεταξύ τους με καλώδια. Στο δίκτυο μπορούν να συνδέονται και άλλες συσκευές, όπως εκτυπωτές και σαρωτές. Σκοπός του είναι η άνετη και η γρήγορη διακίνηση των πληροφοριών, όπως η ανταλλαγή εγγράφων και άλλων αρχείων και γενικότερα δεδομένων, π.χ. να βλέπουμε και να συνομιλούμε με το χρήστη του άλλου υπολογιστή χωρίς να υπάρχει φυσική επαφή.

Υπάρχουν δύο βασικά είδη δικτύων. Τα τοπικά (LAN Local Area Network) και τα ευρείας περιοχής (WAN Wide Area Network). Κάθε υπολογιστής ή συσκευή που είναι συνδεδεμένη με το δίκτυο καλείται κόμβος. Ο αριθμός των κόμβων μπορεί να αποτελείται από δύο και να φτάνει μέχρι δεκάδες ή και εκατοντάδες. Ο τρόπος που συνδέονται οι υπολογιστές και οι συσκευές μεταξύ τους ονομάζονται τοπολογίες. Εκτός από τις τοπολογίες, τα δίκτυα έχουν και άλλα δυο βασικά χαρακτηριστικά, το πρωτόκολλο και την αρχιτεκτονική. Το πρωτόκολλο αναφέρεται στους κανόνες



σύμφωνα με τους οποίους θα επικοινωνούν μεταξύ τους οι κόμβοι, ενώ η αρχιτεκτονική με το αν θα υπάρχει ένας κεντρικός υπολογιστής στο δίκτυο ή όχι.

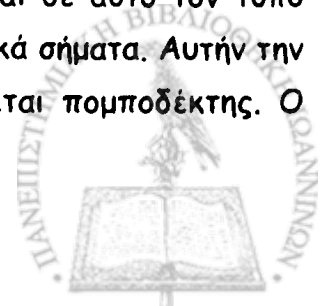
Τα δίκτυα χαρακτηρίζονται από την τοπολογία, το πρωτόκολλο και την αρχιτεκτονική. Το βασικότερο πρωτόκολλο που χρησιμοποιείται ευρέως για μικρά δίκτυα είναι το Ethernet[4] και αποτελεί την πλέον διαδεδομένη μέθοδο υλοποίησης τοπικών δικτύων (LAN).

Το Ethernet αναπτύχθηκε το 1960 από κοινού από τις εταιρίες Intel, Xerox και DEC και παρουσιάστηκε πρώτη φορά το 1973. Το πρωτόκολλο Ethernet περιλαμβάνει δύο βασικές υποκατηγορίες, οι οποίες ξεχωρίζουν κυρίως για το ρυθμό μεταφοράς δεδομένων. Η μία είναι η απλή Ethernet και χαρακτηρίζεται από την ταχύτητα των 10Mbps και η άλλη είναι η Fast Ethernet που έχει αντίστοιχη ταχύτητα τα 100Mbps. Υπάρχει και μία ακόμα υποκατηγορία η οποία υποστηρίζει ταχύτητες 1000Mbps(1Gbps) και ονομάζεται Gigabit Ethernet[20], αλλά δεν είναι τόσο διαδεδομένη ακόμα λόγω του υψηλού κόστους. Το Ethernet επιτρέπει τη μετάδοση πακέτων δεδομένων μεταβλητού μεγέθους, από 72 έως και 1518Byte. Κάθε πακέτο περιέχει μία επικεφαλίδα στην οποία περιλαμβάνονται πληροφορίες όπως η διεύθυνση του συστήματος-αποστολέα, καθώς και αυτή του παραλήπτη.

Η κατηγορία Απλή Ethernet χαρακτηρίζεται από την ταχύτητα των 10Mbps και αποτελείται από τρεις υποκατηγορίες. Τις 10Base5, 10Base2 και 10BaseT που έχουν κοινό χαρακτηριστικό το ρυθμό μεταφοράς δεδομένων.

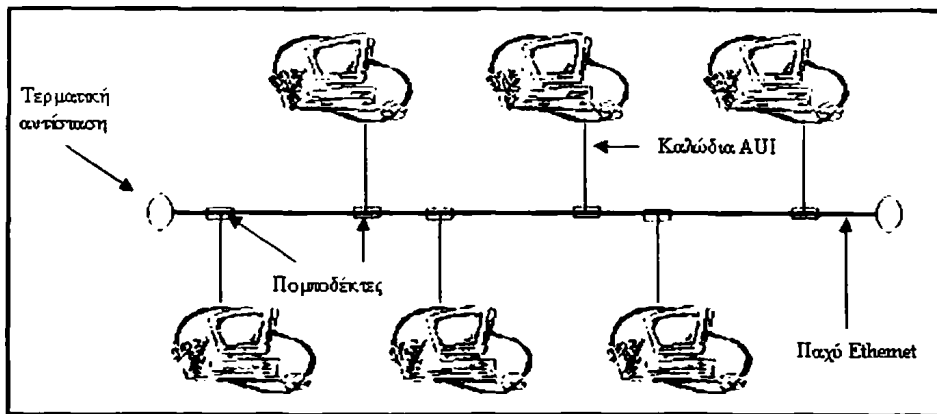
Η κατηγορία Fast Ethernet χαρακτηρίζεται από την ταχύτητα των 100Mbps και σε αυτή είναι η κατηγορία 100BaseT που χρησιμοποιείται στην τοπολογία τύπου αστέρα.

Ο τρόπος καλωδίωσης του αρχικού Ethernet λέγεται άτυπα Ethernet με παχύ καλώδιο επειδή το μέσο επικοινωνίας είναι ένα σχετικά παχύ ομοαξονικό καλώδιο. Επίσης αυτός ο τρόπος λέγεται 10Base5. Η κάρτα διασύνδεσης δικτύου (NIC- Network Interface Card) περιέχει κυκλώματα που χειρίζονται τις ψηφιακές πλευρές της επικοινωνίας, όπως είναι η ανίχνευση σφαλμάτων και η αναγνώριση διευθύνσεων. Η κάρτα διασύνδεσης δικτύου που χρησιμοποιείται σε αυτό τον τύπο δικτύου δεν περιέχει αναλογικό υλικό και δεν χειρίζεται αναλογικά σήματα. Αυτήν την εργασία την αναλαμβάνει μία εξωτερική συσκευή που ονομάζεται πομποδέκτης. Ο



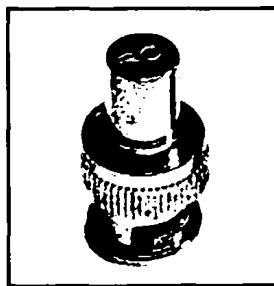
πομποδέκτης συνδέεται απευθείας με το καλώδιο του Ethernet και με ένα άλλο τύπο καλωδίου με την κάρτα διασύνδεσης δικτύου.

Το καλώδιο που συνδέει την κάρτα διασύνδεσης δικτύου με τον πομποδέκτη λέγεται καλώδιο AUI (Attachment Unit Interface). Η διασύνδεση μονάδας προσάρτησης, οι συνδέτες στην κάρτα διασύνδεσης δικτύου και ο πομποδέκτης λέγονται συνδέτες AUI. Το καλώδιο AUI περιέχει πολλά σύρματα, τα βασικά είναι μόνο δύο. Τα υπόλοιπα χρησιμοποιούνται για τον έλεγχο και την τροφοδοσία του πομποδέκτη.



Εικόνα 1.1 Έξι υπολογιστές συνδεδεμένοι σε παχύ Ethernet

Στο σχήμα 1.1 παρατηρούμε ότι εκτός από τις συσκευές που αναφέρθηκαν υπάρχει μία ακόμα συσκευή. Η συσκευή αυτή ονομάζεται τερματική αντίσταση. Η τερματική αντίσταση χρησιμοποιείται για τον τερματισμό του ομοαξονικού καλωδίου. Η τερματική αντίσταση είναι μία ηλεκτρική αντίσταση περίπου 50Ω που ενώνει το κεντρικό σύρμα του καλωδίου με την θωράκιση. Ουσιαστικά, όταν ένα ηλεκτρικό σήμα φτάνει στην αντίσταση τερματισμού, το σήμα αυτό αποβάλλεται.



Εικόνα 1.2 Τερματική αντίσταση

Ο πολυπλέκτης συνδέσεων είναι μία ηλεκτρονική συσκευή σχεδιασμένη να παρέχει τα ίδια ακριβώς σήματα όπως ένας συμβατικός πομποδέκτης. Για παράδειγμα, αν δύο υπολογιστές επιχειρήσουν να μεταδώσουν την ίδια στιγμή, ο πολυπλέκτης αναφέρει σύγκρουση, με τον ίδιο τρόπο που ένα πομποδέκτης θα αναφέρει μία σύγκρουση στο δίκτυο. Παρόμοια αν υπάρχει φέρον σήμα στο δίκτυο, ο πολυπλέκτης αναφέρει την παρουσία φέροντος σήματος σε όλους τους συνδεδεμένους σταθμούς.

Ένας διαφορετικός τρόπος καλωδίωσης είναι το Ethernet με λεπτό καλώδιο (Thin wire Ethernet ή Thinnet) επειδή το υλικό που χρησιμοποιεί είναι ένα πιο λεπτό και πιο εύκαμπτο ομοαξονικό καλώδιο από το αρχικό παχύ καλώδιο. Αυτός ο τρόπος καλωδίωσης επίσημα λέγεται 10Base2 και διαφέρει από το πρώτο παχύ Ethernet(10Base5, Thicknet) σε τρία σημεία.

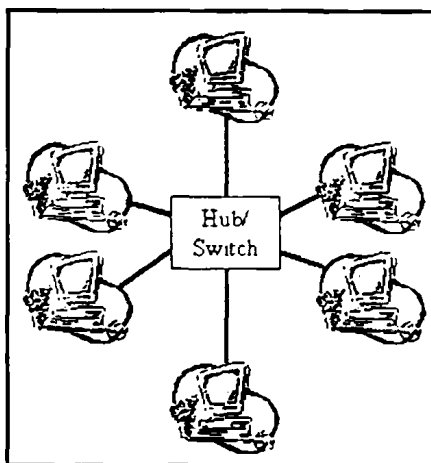
1. Η εγκατάσταση και η λειτουργία του Thinnet γενικά κοστίζει λιγότερο από το Thicknet.
2. Επειδή το υλικό που πραγματοποιεί τις λειτουργίες του πομποδέκτη είναι ενσωματωμένο στην κάρτα διασύνδεσης δικτύου (NIC), δεν χρειάζονται εξωτερικοί πομποδέκτες.
3. Το Thinnet δεν χρησιμοποιεί καλώδιο AUI για την σύνδεση της κάρτας διασύνδεσης δικτύου με το μέσο επικοινωνίας αλλά συνδέεται απευθείας στο πίσω μέρος του κάθε υπολογιστή με συνδέτη BNC.

Ο τρίτος τύπος καλωδίωσης διαφέρει ριζικά και από το παχύ και από το λεπτό Ethernet. Ο τρόπος αυτός λέγεται Ethernet συνεστραμένου ζεύγους καλωδίων ή TP Ethernet και επίσημα 10BaseT. Το 10BaseT το οποίο έχει γίνει ο καθιερωμένος τρόπος σύνδεσης για το Ethernet δεν χρησιμοποιεί καθόλου ομοαξονικό καλώδιο. Μάλιστα ένα δίκτυο Ethernet δεν έχει μεριζόμενο φυσικό μέσο επικοινωνίας όπως οι άλλοι τρόποι καλωδίωσης. Αντίθετα, το 10BaseT επεκτείνει την ιδέα που χρησιμοποιεί η πολύπλεξη συνδέσεων, δηλαδή μία κεντρική ηλεκτρονική συσκευή εξυπηρετεί το δίκτυο. Η κεντρική ηλεκτρονική συσκευή λέγεται Ethernet Hub.

Όπως και οι άλλοι τρόποι καλωδίωσης, το 10 BaseT απαιτεί κάθε υπολογιστής να έχει μία κάρτα διασύνδεσης δικτύου NIC και μία απευθείας σύνδεση από την κάρτα

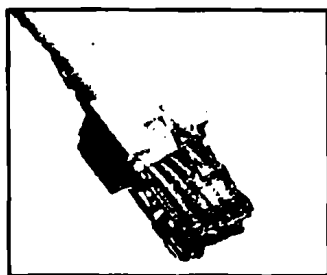


προς το δίκτυο. Η σύνδεση χρησιμοποιεί καλωδίωση συνεστραμένου ζεύγους καλωδίων με συνδέτες RJ-45, οι οποίοι είναι μεγαλύτερες παραλλαγές των συνδετών που χρησιμοποιούνται στα τηλέφωνα. Ο συνδέτης στο ένα άκρο του συνεστραμένου ζεύγους προσαρμόζεται στη διασύνδεση δικτύου ενός υπολογιστή, και ο συνδέτης στο άλλο άκρο προσαρμόζεται στο Hub. Έτσι ο κάθε υπολογιστής έχει την δική του αποκλειστική σύνδεση με την συσκευή του Hub.

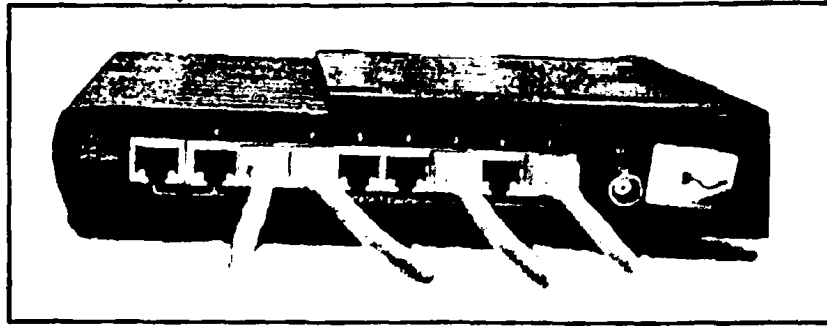


Εικόνα 1.3 Έξι υπολογιστές συνδεδεμένοι σε ένα Hub με καλωδίωση 10BaseT

Η τεχνολογία του Hub είναι μία επέκταση του πολυπλέκτη συνδέσεων. Ένας υπολογιστής συνδεδεμένος στο Hub πρέπει να έχει φυσική διεύθυνση Ethernet. Κάθε υπολογιστής πρέπει να χρησιμοποιεί το μηχανισμό CSMA/CD για να προσπελάσει το δίκτυο, και την καθιερωμένη μορφή πλαισίων του Ethernet. Μάλιστα, το λογισμικό δεν διακρίνει αν πρόκειται για παχύ Ethernet, Λεπτό Ethernet, ή 10BaseT. Η διασύνδεση δικτύου χειρίζεται τις λεπτομέρειες και κρύβει τις οποιεσδήποτε διαφορές.



Εικόνα 1.4 Συζευκτήρας RJ-45



Εικόνα 1.5 Hub με 8 θύρες UTP και μια BNC

Στην γρήγορη έκδοση του Ethernet, (Fast Ethernet 100BaseT) ο ρυθμός μεταφοράς δεδομένων φτάνει τα 100Mbps. Το κόστος είναι υψηλότερο, κυρίως λόγω των πιο γρήγορων και άρα πιο ακριβών καρτών δικτύου, όπως και των αντίστοιχων Hub. Ο υπόλοιπος εξοπλισμός είναι ίδιος με του πρωτοκόλλου 10BaseT.

Τέλος, στα ασύρματα δίκτυα WLAN (Wireless LAN)[5] τα δεδομένα μεταδίδονται ασύρματα. Οι κόμβοι του δικτύου επικοινωνούν ασύρματα με διάφορα σημεία πρόσβασης (access points), που είναι ειδικές συσκευές συνδεδεμένες πάνω σε hub ή σε διακομιστές. Μεταξύ ενός υπολογιστή και ενός σημείου πρόσβασης μπορεί να υπάρχουν και εμπόδια (π.χ., τοίχοι), αρκεί να μην είναι μεταλλικά. Ο οργανισμός IEEE έχει θεσπίσει τέσσερα πρότυπα για προϊόντα που συμμετέχουν σε WLAN. Το πρώτο από αυτά, το 802.11, παρέχει ταχύτητες μετάδοσης δεδομένων 1 ή 2Mbps στην περιοχή συχνοτήτων των 2,4GHz, το 802.11a ταχύτητες 24Mbps στην περιοχή συχνοτήτων των 5GHz, το 802.11b ταχύτητες μέχρι και 11Mbps στη συχνότητα των 2,4GHz ενώ το 802.11g μέχρι και 54Mbps.

1.2 Περιγραφή του ασύρματου συστήματος μετάδοσης φυσικών πληροφοριών στο διαδίκτυο

Το σύστημα το οποίο κατασκευάστηκε έχει την δυνατότητα μέτρησης διαφόρων παραμέτρων του νερού, ανάλογα με τους αισθητήρες που θα τοποθετηθούν.



Το συνολικό σύστημα μετρά διάφορες παραμέτρους ποιότητας του νερού. Οι παράμετροι που μετρώνται εξαρτώνται από τους αισθητήρες που έχουν τοποθετηθεί στο σύστημα. Στο συγκεκριμένο σύστημα έχουν τοποθετηθεί πέντε αισθητήρες. Αυτοί είναι: αισθητήρας θερμοκρασίας, pH, αγωγιμότητας, οξυγόνου, ORP/Redox. Δηλαδή ο μόνος ουσιαστικός περιορισμός είναι τι αισθητήρες παράγουν οι εταιρίες ή αντίστοιχα τι αισθητήρες θέλουμε και μπορούμε να κατασκευάσουμε εμείς.

Αρχικά, κατά την διαδικασία της μέτρησης, μια ποσότητα νερού μεταφέρεται σε ένα ανοξείδωτο δοχείο. Εκεί γίνονται οι μετρήσεις από τους αισθητήρες. Τα αποτελέσματα των μετρήσεων μεταφέρονται σε ένα μικροελεγκτή, επεξεργάζονται και αποθηκεύονται μαζί με την ημερομηνία και την ώρα της μέτρησης. Αναλόγως με την συχνότητα και τον αριθμό των αισθητήρων που έχουν τοποθετηθεί, το σύστημα μπορεί να αποθηκεύσει τις τιμές των αισθητήρων για τους τελευταίους δύο έως έξι μήνες καθώς και το έτος, τον μήνα, την ημερομηνία αλλά και την ώρα που έγιναν οι μετρήσεις. Οι ώρες της ημέρας που θέλουμε να παίρνουμε μετρήσεις ορίζονται αρχικά στον προγραμματισμό. Δηλαδή ο μόνος ουσιαστικός περιορισμός είναι ότι οι ώρες αυτές επαναλαμβάνονται καθημερινά και από το γεγονός ότι θα πρέπει να έχουν τελειώσει όλες οι προηγούμενες διαδικασίες μέτρησης, όπως άδειασμα, γέμισμα δοχείου, επεξεργασία προηγούμενων δεδομένων κ.ά. για να προγραμματιστεί η επόμενη.

Ο ηλεκτρονικός έλεγχος του συστήματος γίνεται από την συνεργασία τεσσάρων μικροεπεξεργαστών. Τα αποτελέσματα των αισθητήρων επεξεργάζονται από έναν Atmel ATMEGA128[6] και μεταφέρονται σε έναν άλλο ATMEGA128, ο οποίος σε συνεργασία με ένα ρολόι πραγματικού χρόνου από έναν ATMEGA16 τα μεταφέρει σ'έναν υπολογιστή, όταν ζητηθεί, μέσω ενός web server. Αυτός βασίζεται σ'ένα ATMEGA128.

Έπειτα τα αποτελέσματα μεταφέρονται με ασύρματο δίκτυο WiFi, και με δυνατότητα διαμεσολάβησης όσων αναμεταδοτών χρειάζονται, στο τμήμα Φυσικής του Πανεπιστημίου Ιωαννίνων.

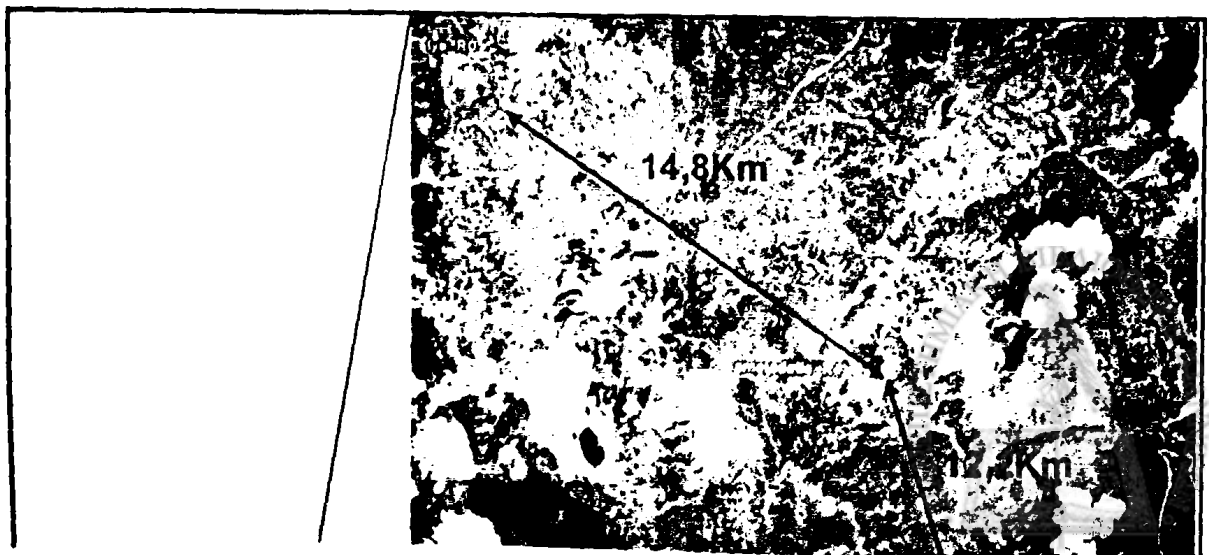
1.3 Τοπολογία του συστήματος



Το σύστημα εγκαταστάθηκε στον ποταμό Άραχθο, στην παλιά γέφυρα της Πλάκας, με συντεταγμένες $39^{\circ}27' 38.59''\text{N}$ [18], $21^{\circ}01' 48.95''\text{E}$. Από εκεί μεταφέρεται η απαραίτητη ποσότητα νερού με αντλία στο πιο κοντινό κτίριο που βρίσκεται δώδεκα μέτρα ψηλότερα από την επιφάνεια της στάθμης του νερού, στο παλιό τελωνείο. Εκεί αφού γίνουν όλες οι διαδικασίες μέτρησης και αποθήκευσης των αποτελεσμάτων, μεταφέρονται κατόπιν εντολής, με WiFi στην πιο κοντινή κορυφή με συντεταγμένες $39^{\circ}27' 28.78''\text{N}$, $21^{\circ}00' 36.30''\text{E}$. Εκεί εγκαταστάθηκε ο πρώτος αναμεταδότης. Σε εκείνο το σημείο επειδή δεν υπήρχε τροφοδοσία ηλεκτρικού ρεύματος, χρησιμοποιήθηκε φωτοβολταϊκό πάνελ σε συνδυασμό με μπαταρία ξηρής φόρτισης έτσι ώστε να τροφοδοτούνται με ρεύμα όλα τα συστήματα. Για την μέγιστη ενεργειακή απόδοση του παραπάνω συστήματος κατασκευάστηκε σύστημα ιχνηλάτισης του ήλιου (tracking system) και ενσωματώθηκε στο φωτοβολταϊκό πάνελ.

Στη συνέχεια τα δεδομένα στέλνονται δώδεκα χιλιόμετρα μακριά, στο σημείο με συντεταγμένες $39^{\circ}32' 51.42''\text{N}$, $20^{\circ}58' 23.24''\text{E}$ όπου εγκαταστάθηκε ο δεύτερος αναμεταδότης. Εκεί υπήρχε παροχή ρεύματος και έτσι δεν χρειάστηκε φωτοβολταϊκό πάνελ. Στο σημείο αυτό υπάρχει οπτική επαφή με την τaráτσα του τμήματος Φυσικής όπου τοποθετήθηκε ο κατάλληλος εξοπλισμός έτσι ώστε να είναι δυνατή η λήψη του σήματος στο πανεπιστήμιο με συντεταγμένες $39^{\circ}32' 51.36''\text{N}$ και $20^{\circ}58' 23.39''\text{E}$.

Μέσω ενός ηλεκτρονικού υπολογιστή που είναι συνδεδεμένος με το παραπάνω ασύρματο δίκτυο υπάρχει η δυνατότητα «διαβάσματος» και αποθήκευσης των τελευταίων μετρήσεων που υπάρχουν στη μνήμη του μικροελεγκτή μέσω ενός απλού προγράμματος περιήγησης στο διαδίκτυο όπως (internet explorer, firefox[19], κ.ά.).



Εικόνα 1.6 Δορυφορική φωτογραφία από Google Earth των σημείων εγκατάστασης του συνολικού συστήματος

Στην Εικόνα 1.6 φαίνεται η τοποθεσία του συστήματος πάνω στον χάρτη.

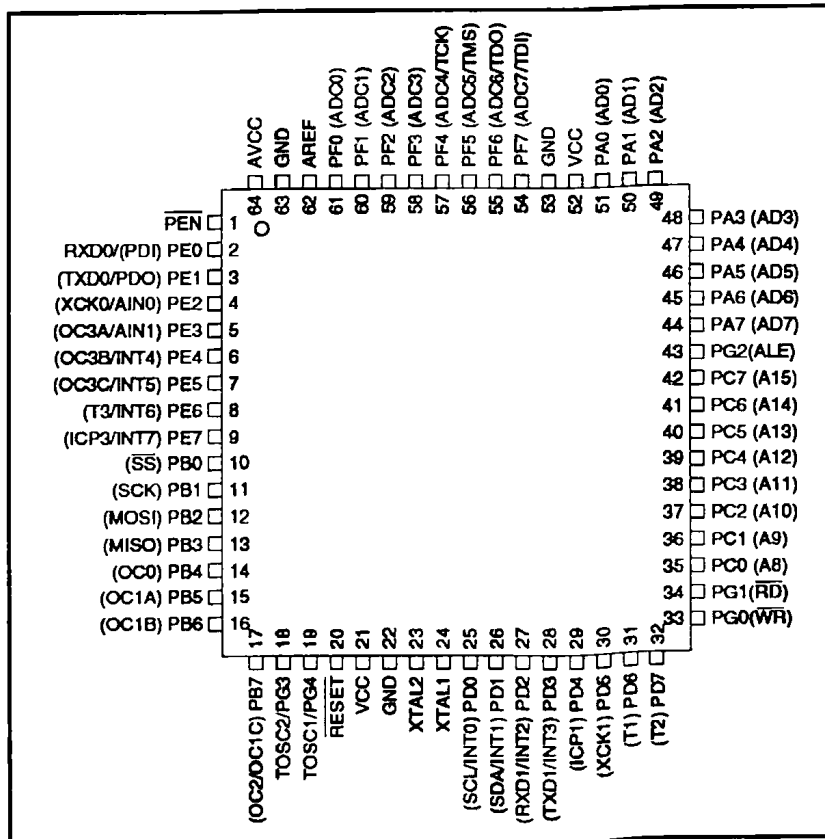


2. Διακομιστής διαδικτύου - webserver



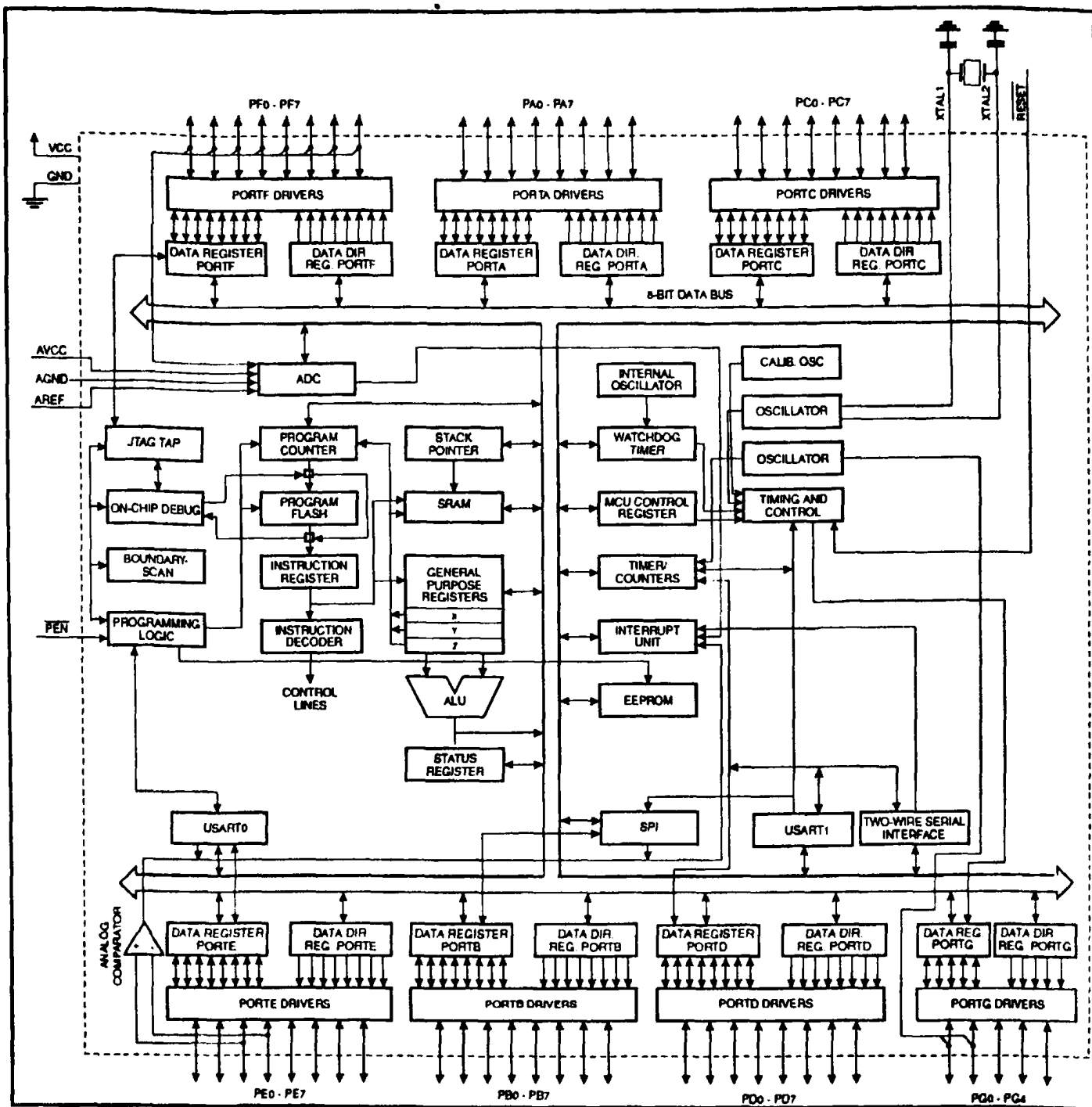
2.1 Γενικές πληροφορίες για τον ATMEGA128

Ο web server είναι βασισμένος σε μικροελεγκτή της Atmel, και συγκεκριμένα στο μοντέλο ATMEGA128[6]. Γι'αυτό θα πρέπει πρώτα να αναφερθεί πιο αναλυτικά ο συγκεκριμένος επεξεργαστής με τις δυνατότητές του και τις διάφορες λειτουργίες έτσι ώστε να γίνει πιο κατανοητή η λειτουργία του συστήματος που τον χρησιμοποιεί.



Εικόνα 2.1 Επεξεργαστής ATMEGA128 Της Atmel

Στην εικόνα 2.1 φαίνεται το σχεδιάγραμμα των pin του επεξεργαστή, με ποια σειρά είναι τα pin και τι λειτουργίες μπορεί να κάνει το καθένα.



Εικόνα 2.2 Γενικό διάγραμμα του εσωτερικού του επεξεργαστή ATMEGA128

Στην εικόνα 2.2 φαίνεται το γενικό διάγραμμα του επεξεργαστή με τους ακροδέκτες του. Έχει έξι πόρτες εισόδου-εξόδου με οχτώ bit η κάθε πόρτα και μία με πέντε bit. Έχει επίσης οχτώ μετατροπείς αναλογικού σήματος σε ψηφιακό (ADC) που συνδέονται με την πόρτα F.



Η ονοματολογία και οι λειτουργίες των ακροδεκτών είναι ως εξής:

VCC: Θετική τάση τροφοδοσίας

GND: Γείωση

Port A (PA7..PA0): Η πόρτα A είναι μια πόρτα με οχτώ bit, εισόδου-εξόδου με εσωτερικές αντιστάσεις pull up για κάθε bit. Όλοι οι ακροδέκτες έχουν αρκετά μεγάλη αντίσταση εισόδου έτσι ώστε αρκεί μια τάση με ελάχιστο ρεύμα για να ενεργοποιηθεί η κατάσταση high. Επίσης μπορούν να χρησιμοποιηθούν σαν δίαυλοι διευθύνσεων μνήμης για εξωτερική μνήμη.

Port B (PB7..PB0): Η πόρτα B έχει τις ίδιες ιδιότητες με την πόρτα A, αλλά κάθε pin έχει και δεύτερη λειτουργία:

PB7: Έξοδος σύγκρισης και διαμόρφωση πλάτους παλμού (Pulse Width Modulation PWM) για τον Timer/Counter2 ή έξοδος σύγκρισης C και PWM για Timer/Counter1.

PB6: Έξοδος σύγκρισης B και διαμόρφωση πλάτους παλμού για Timer/Counter1.

PB5: Έξοδος σύγκρισης A και διαμόρφωση πλάτους παλμού για Timer/Counter1.

PB4: Έξοδος σύγκρισης και διαμόρφωση πλάτους παλμού για Timer/Counter0

PB3: Master Input ή Slave Output για το SPI

PB2: Master Output ή Slave Input για το SPI

PB1: Serial Clock για το SPI

PB0: επιλογή SPI Slave

Port C (PC7..PC0): Παρόμοια με την πόρτα A έχει και η πόρτα C.

Port D (PD7..PD0): Η πόρτα D έχει τις ίδιες ιδιότητες με την πόρτα A, αλλά κάθε pin έχει και δεύτερη λειτουργία:

PD7: είσοδος ρολογιού Timer/Counter2

PD6: είσοδος ρολογιού Timer/Counter1

PD5: εξωτερικό ρολόι είσοδος/έξοδος του USART1



PD4: είσοδος Timer/Counter1

PD3: Είσοδος εξωτερικού interrupt3 ή ακροδέκτης αποστολής του UART1

PD2: Είσοδος εξωτερικού interrupt2 ή ακροδέκτης λήψης του UART1

PD1: Είσοδος εξωτερικού interrupt1 ή σειριακά δεδομένα για το TWI

PD0: Είσοδος εξωτερικού interrupt0 ή σειριακό ρολόι για το TWI

Port E (PE7..PE0): Η πόρτα E έχει τις ίδιες ιδιότητες με την πόρτα A, αλλά κάθε pin έχει και δεύτερη λειτουργία:

PE7: Είσοδος εξωτερικού interrupt7 ή είσοδος timer/counter3

PE6: Είσοδος εξωτερικού interrupt6 ή είσοδος ρολογιού timer/counter3

PE5: Είσοδος εξωτερικού interrupt5 ή έξοδος σύγκρισης για την διαμόρφωση πλάτους παλμού C για timer/counter3

PE4: Είσοδος εξωτερικού interrupt4 ή έξοδος σύγκρισης για την διαμόρφωση πλάτους παλμού B για τον timer/counter3

PE3: Είσοδος αρνητικής τάσης αναλογικού συγκριτή ή έξοδος για την διαμόρφωση πλάτους παλμού για τον timer/counter3

PE2: Είσοδος θετικής αναλογικής τάσης ή εξωτερικό ρολόι για το UART0

PE1: Έξοδος προγραμματισμού ή ακροδέκτης αποστολής για το UART0

PE0: Είσοδος προγραμματισμού ή ακροδέκτης λήψης για το UART0

Port F (PF7..PF0) Η πόρτα F λειτουργεί σαν αναλογική είσοδος για τον μετατροπέα αναλογικού σήματος σε ψηφιακό (ADC). Επίσης είναι μια πόρτα με οχτώ bit, εισόδου-εξόδου με εσωτερικές pull up αντιστάσεις για κάθε bit. Όλα τα pin έχουν αρκετά μεγάλη αντίσταση εισόδου έτσι ώστε αρκεί μια τάση με ελάχιστο ρεύμα για να ενεργοποιηθεί η κατάσταση high. Επίσης λειτουργεί και σαν συνδέτης JTAG.

Port G (PG4..PG0): Παρόμοια με την πόρτα A έχει και η πόρτα G

RESET: Λογικό μηδέν για περισσότερο χρόνο από την διάρκεια ενός παλμού σε αυτόν τον ακροδέκτη, προκαλεί σήμα επανεκκίνησης του μικροελεγκτή ακόμα και αν το ρολόι δεν είναι ενεργοποιημένο. Εάν το λογικό μηδέν εφαρμοστεί για μικρότερο χρονικό



διάστημα δεν είναι σίγουρο τι ακριβώς θα γίνει, δηλαδή υπάρχει και πάλι η περίπτωση να γίνει επαννεκίνηση.

XTAL1: Ανεστραμμένη είσοδο ενισχυτή για τον κρύσταλλο και είσοδος για το εσωτερικό ρολόι

XTAL2: Ανεστραμμένη έξοδος ενισχυτή από τον κρύσταλλο.

AVCC: Είναι η τροφοδοσία για την πόρτα F και τον μετατροπέα αναλογικού σήματος σε ψηφιακό. Πρέπει να συνδεθεί εξωτερικά με το Vcc ακόμη και αν ο μετατροπέας από αναλογικό σε ψηφιακό δεν χρησιμοποιείται. Εάν ο μετατροπέας χρησιμοποιείται τότε πρέπει να συνδεθεί το Vcc μέσω ενός χαμηλοπερατού φίλτρου.

AREF: Είναι η αναλογική τάση αναφοράς για τον μετατροπέα αναλογικού σήματος σε ψηφιακό.

PEN: Είναι το pin ενεργοποίησης για τον σειριακό προγραμματισμό με SPI και είναι συνδεδεμένο εσωτερικά στη θέση high. Κρατώντας το στη θέση low κατά τη διάρκεια ενός σήματος επαννεκίνησης κατά την εφαρμογή τάσης στον μικροελεγκτή (power-on Reset), η συσκευή θα μπει σε λειτουργία προγραμματισμού με SPI. Δεν έχει καμία λειτουργία κατά τη διάρκεια κανονικής λειτουργίας.

Οι έλεγχοι εξομοίωσης αξιοπιστίας της εταιρίας Atmel έδειξαν ότι το σφάλμα στα δεδομένα είναι λιγότερο από 1 PPM για 20 χρόνια στους 85° C και για 100 χρόνια στους 25° C.

Με σκοπό την αύξηση της απόδοσης και τον παραλληλισμό, ο μικροελεγκτής χρησιμοποιεί μια αρχιτεκτονική με ξεχωριστές μνήμες και διαύλους για το πρόγραμμα και για τα δεδομένα. Οι εντολές στην μνήμη του προγράμματος εκτελούνται κυρίως σε ένα κύκλο μηχανής. Αυτό επιτυγχάνεται επειδή ενώ μια εντολή εκτελείται, η επόμενη προφορτώνεται. Η μνήμη προγράμματος είναι στην επαναπρογραμματιζόμενη flash μνήμη (System Reprogrammable Flash memory).

2.2 Αριθμητική λογική μονάδα



Ο μικροελεγκτής έχει τριανταδύο καταχωρητές γενικής χρήσης με οχτώ bit ο καθένας και επιτυγχάνεται η πρόσβαση σε αυτούς μόνο με ένα κύκλο μηχανής. Αυτό επιτρέπει την αριθμητική λογική μονάδα να λειτουργεί σ' ένα μόνο κύκλο μηχανής. Σε μια τυπική λειτουργία της αριθμητικής λογικής μονάδας, οι δύο συμβαλλόμενοι βγαίνουν από τους καταχωρητές, εκτελείται η πράξη και το αποτέλεσμα αποθηκεύεται πάλι στους καταχωρητές μόνο μέσα σ' ένα κύκλο μηχανής.

Η αριθμητική λογική μονάδα υποστηρίζει αριθμητικές και λογικές πράξεις μεταξύ καταχωρητών ή μεταξύ ενός σταθερού αριθμού και καταχωρητή. Μετά από μια πράξη, αριθμητική ή λογική, ο καταχωρητής status ενημερώνεται έτσι ώστε να αντικατοπτρίζει τις απαραίτητες πληροφορίες του αποτελέσματος της αριθμητικής λογικής μονάδας.

Η Program Flash μνήμη χωρίζεται σε δύο τομείς, τον boot program και application program. Και οι δύο τομείς έχουν κάποια bit για να κλειδώνουν την εγγραφή/διάβασμα.

Η υψηλής απόδοσης λογική μονάδα λειτουργεί με απευθείας σύνδεση με τους τριάντα δύο καταχωρητές γενικής χρήσεως. Μέσα σε ένα κύκλο μηχανής, αριθμητικές λειτουργίες μεταξύ καταχωρητών γενικής χρήσεως ή μεταξύ ενός καταχωρητή και ενός αριθμού μπορούν να εκτελεστούν. Οι λειτουργίες της λογικής μονάδας χωρίζονται σε τρεις κύριες κατηγορίες: τις αριθμητικές, τις λογικές και τις πράξεις μεταξύ bit.

2.3 Καταχωρητές

Οι καταχωρητές σε ένα μικροελεγκτή είναι θέσεις μνήμες στις οποίες κάθε δυαδική πληροφορία σημαίνει κάτι συγκεκριμένο (όπως ότι το αποτέλεσμα μιας αριθμητικής πράξης είναι 0). Πολλές από αυτές τις δυαδικές πληροφορίες αλλάζουν από το υλικό (hardware) ενώ υπάρχουν και κάποιες που αλλάζουν με λογισμικό. Πολλές από τις πληροφορίες των καταχωρητών είναι χρήσιμες και χρησιμοποιούνται για να γίνει ο προγραμματισμός των μικροεπεξεργαστών πιο κατανοητός και πιο εύκολος. Ο κυριότερος από αυτούς τους καταχωρητές είναι ο καταχωρητής κατάστασης (status).



Ο καταχωρητής κατάστασης περιέχει πληροφορίες σχετικές με το πιο πρόσφατο αποτέλεσμα της λογικής μονάδας. Αυτές οι πληροφορίες μπορεί να χρησιμοποιηθούν για να αλλάξει η ροή του προγράμματος με συνθήκες. Πρέπει να σημειωθεί ότι ο καταχωρητής κατάστασης ενημερώνεται κάθε φορά που η λογική μονάδα βγάζει αποτέλεσμα. Ο καταχωρητής κατάστασης δεν αποθηκεύεται αυτόματα όταν η ροή του προγράμματος μπαίνει σε υποπρόγραμμα διακοπής.

Τα δυαδικά ψηφία και οι λειτουργίες του κάθε δυαδικού ψηφίου του καταχωρητή φαίνονται στον Πίνακα 1.3.

Πίνακας 1.3 Καταχωρητής κατάστασης (status) του ATMEGA28

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	II	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 - I: Global Interrupt Enable: Δυαδικό ψηφίο ενεργοποίησης/απενεργοποίησης όλων των διακοπών (interrupt). Για να ενεργοποιηθούν οι διακοπές (interrupt) πρέπει να είναι στο λογικό ένα. Όταν είναι στο μηδέν, ανεξάρτητα από το τι είναι όλα τα τοπικά interrupt κανένα δεν είναι ενεργοποιημένο. Το I γίνεται μηδέν από το υλικό (hardware) μετά από μια διακοπή (interrupt) και γίνεται λογικό ένα από την εντολή επιστροφής από υποπρόγραμμα εξυπηρέτησης της διακοπής, RETI. Μπορεί επίσης να γίνει ένα ή μηδέν από το λογισμικό με τις εντολές SEI και CLI αντίστοιχα.

Bit 6 - T: Bit Copy Storage: Με την εντολή λογισμικού BST αντιγράφεται ένα bit από ένα καταχωρητή στο T, και με την εντολή BLD αντιγράφεται το bit T σ' ένα καταχωρητή.

Bit 5 - T: Half Carry Flag: Δυαδικό ψηφίο ειδικού κρατούμενου που χρησιμοποιείται κυρίως στον BCD κώδικα.



Bit 4 - S: Sign Bit, $S = N \oplus V$: Διαδικό ψηφίο αποκλειστικού ή (exclusive or) της αρνητικής σημαίας N και του δεύτερου συμπληρώματος της σημαίας υπερχύλησης V.

Bit 3 - V: Two's Complement OverFlow Flag: Υποστηρίζει πράξεις με δεύτερο συμπλήρωμα.

Bit 2 - N: Negative Flag: Υποδηλώνει αρνητικό αποτέλεσμα σε μια αριθμητική πράξη ή λογική λειτουργία.

Bit 1 - Z: Zero Flag: Υποδηλώνει μηδενικό αποτέλεσμα σε μια αριθμητική πράξη ή λογική λειτουργία.

Bit 0 - C: Carry Flag: Υποδηλώνει κρατούμενο σε μια αριθμητική πράξη ή λογική λειτουργία.



3. Το υλικό του συστήματος του διακομιστή διαδικτύου για την μετάδοση της πληροφορίας

Το υλικό του συστήματος του διακομιστή διαδικτύου για την μετάδοση της πληροφορίας, εκτός της πλακέτας του διακομιστή διαδικτύου περιλαμβάνει μια πλακέτα πάνω στην οποία υπάρχει το ρολόι πραγματικού χρόνου (real time clock) και η διασύνδεση μεταξύ του ρολογιού πραγματικού χρόνου, της πλακέτας του διακομιστή διαδικτύου και της πλακέτας η οποία στέλνει τα αποτελέσματα των μετρήσεων από τους αισθητήρες.

3.1 Οι λειτουργίες της πλακέτας του διακομιστή διαδικτύου

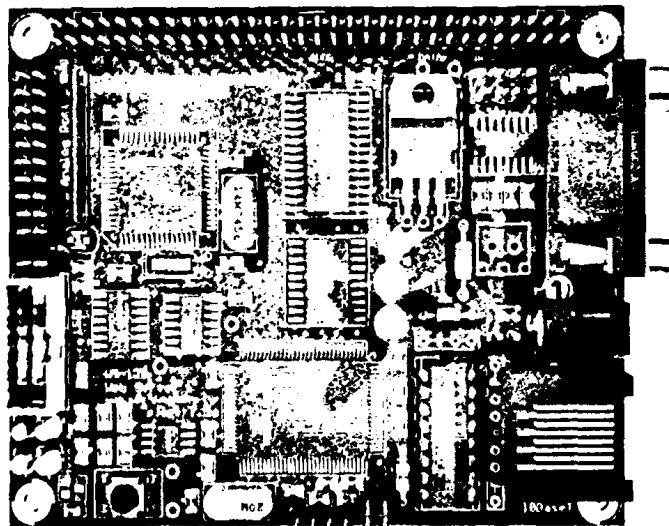
Η πλακέτα στην οποία λειτουργεί ο διακομιστής διαδικτύου (web server) στέλνει τα δεδομένα των μετρήσεων στο Πανεπιστήμιο Ιωαννίνων μέσω ασύρματου δικτύου WiFi. Η πλακέτα είναι η Ethernet Starter Kit Rev. 1.3H[10] και αγοράστηκε από την ελεύθερη αγορά.

Στην αρχική φάση της εκπόνησης της διπλωματικής εργασίας χρησιμοποιήθηκε η πλακέτα που φαίνεται στην εικόνα 3.1. Η πλακέτα βασίζεται σ'ένα 16 bit επεξεργαστή της Atmel, τον ATMEGA128. Η πλακέτα είναι η Atmel AVR ATmega128 14.756MHz processor της SOC MACHINES[7] και το μοντέλο της λέγεται Amber Web Server WS128 Plus PCB Rev 1.5a[8].

Κατά τον έλεγχο του διακομιστή διαδικτύου, πιθανόν από αδέξιο χειρισμό ή ελαττωματικό εξάρτημα, η πλακέτα καταστράφηκε. Έτσι αγοράστηκε η πλακέτα Ethernet Starter Kit Rev 1.3H της οποίας η όψη των ηλεκτρονικών στοιχείων φαίνεται στην εικόνα 3.3.



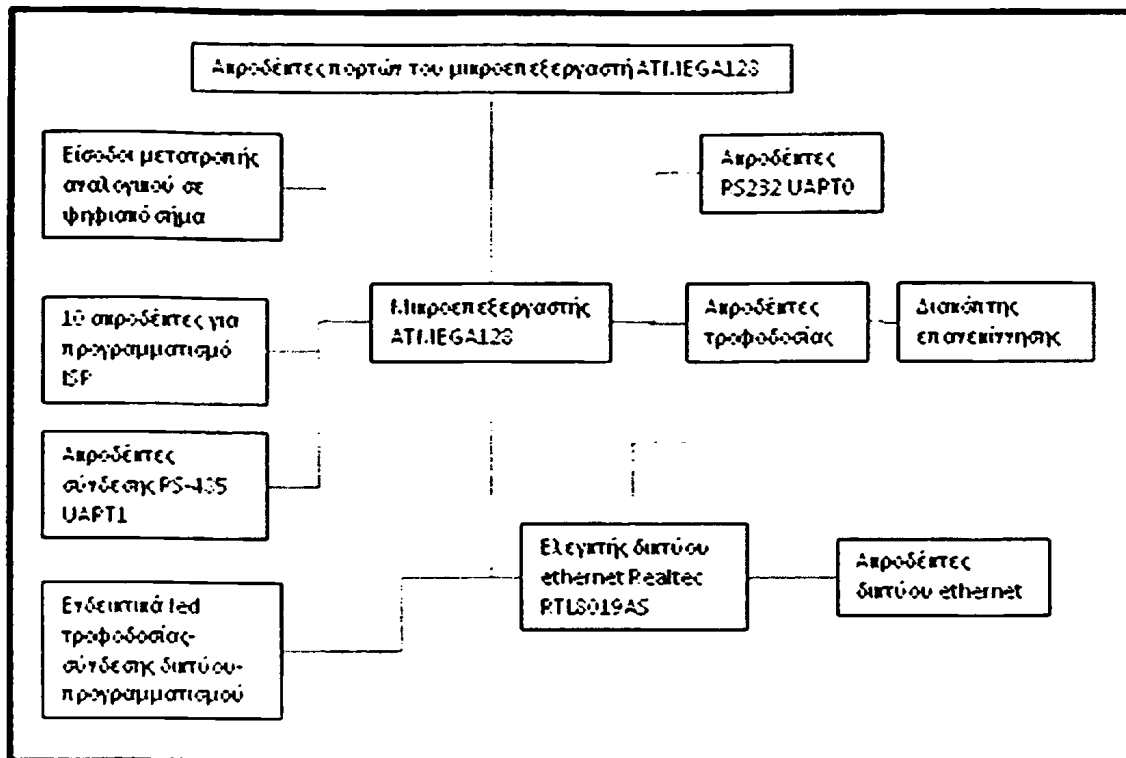
Η κύρια λειτουργία της συγκεκριμένης πλακέτας είναι η εμφάνιση ιστοσελίδων στο διαδίκτυο όταν χρησιμοποιηθεί η διεύθυνση IP της πλακέτας. Όταν κάποιος δώσει την συγκεκριμένη διεύθυνση IP της συσκευής, τότε φορτώνεται η αρχική σελίδα που είναι αποθηκευμένη στη πλακέτα. Η αρχική αυτή σελίδα μπορεί να τροποποιηθεί και να πάρει οποιαδήποτε μορφή, αρκεί το μέγεθός της να μπορεί να χωρέσει στην μνήμη ιστοσελίδων της πλακέτας. Από την αρχική σελίδα που έχει αναπτυχθεί για την συγκεκριμένη εφαρμογή μπορεί να ενεργοποιηθεί ο σύνδεσμος με τις τελευταίες μετρήσεις για να φορτωθούν τα δεδομένα των μετρήσεων. Με την ενεργοποίηση του συνδέσμου ανοίγει μια καινούρια ιστοσελίδα και ζητείται από την πλακέτα διασύνδεσης με τους αισθητήρες να στείλει όλες τις αποθηκευμένες μετρήσεις. Αν η πλακέτα διασύνδεσης δεν είναι απασχολημένη, δηλαδή δεν βρίσκεται στη διάρκεια μέτρησης, τότε στέλνει τις μετρήσεις και εμφανίζονται στη οθόνη, αν είναι απασχολημένη τότε ο διακομιστής διαδικτύου εμφανίζει το κατάλληλο μήνυμα στην οθόνη προτρέποντας τον χρήστη να δοκιμάσει αργότερα.



Εικόνα 3.1 Εικονική απεικόνιση της πλακέτας του διακομιστή διαδικτύου Amber Web Server WS128 Plus PCB Rev.15a

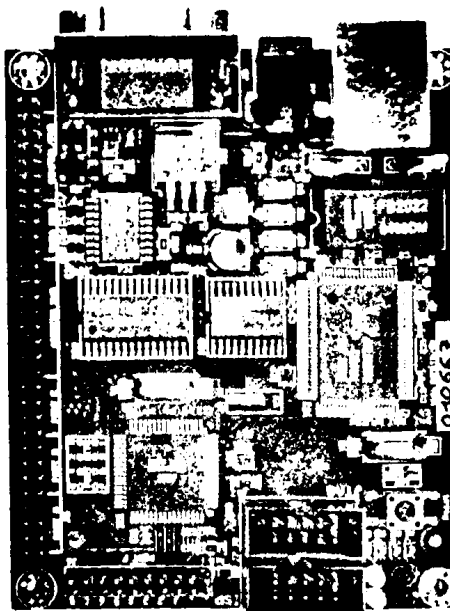
Όπως φαίνεται στην εικόνα 3.2 η πλακέτα του διακομιστή διαδικτύου έχει ένα σειριακό βύσμα σύνδεσης με ηλεκτρονικό υπολογιστή, μια είσοδο τροφοδοσίας τάσης της πλακέτας, το βύσμα σύνδεσης του καλωδίου Ethernet, το οποίο χρησιμοποιήθηκε για να αναπτυχθεί το δίκτυο Wifi, η είσοδος του καλωδίου ISP, το οποίο

χρησιμοποιείται για να προγραμματιστεί ο μικροελεγκτής μέσω ηλεκτρονικού υπολογιστή με το κατάλληλο λογισμικό, εφτά αναλογικές εισόδους και τους ακροδέκτες από τις πόρτες του μικροελεγκτή.



Εικόνα 3.2 Γενικό διάγραμμα της πλακέτας του διακομιστή διαδικτύου (webserver)

Η πλακέτα μπορεί να προγραμματιστεί με ένα καλώδιο προγραμματισμού ISP[21] το οποίο συνδέεται στον μετατροπέα των εννιά ακροδεκτών (9pin) και στην παράλληλη θύρα του ηλεκτρονικού υπολογιστή. Χρειάζεται επίσης και ένα πρόγραμμα το οποίο να έχει τη δυνατότητα να προγραμματίζει μικροεπεξεργαστές ATMEL της συγκεκριμένης σειράς, όπως το AVR Studio 4[22].

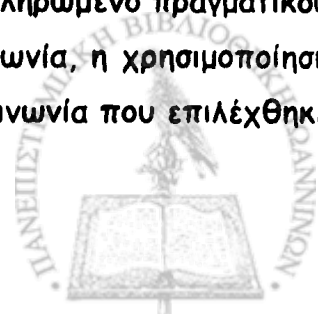


Εικόνα 3.3 Η πλακέτα του διακομιστή διαδικτύου Ethernet Starter Kit Rev. 1.3H

3.2 Η πλακέτα χρονομέτρου πραγματικού χρόνου (real time clock) και διασύνδεσης με τον διακομιστή διαδικτύου και τους αισθητήρες

Η πλακέτα χρονομέτρου πραγματικού χρόνου (real time clock) βασίζεται στον ίδιο επεξεργαστή με τον διακομιστή διαδικτύου, δηλαδή τον ATMEGA128[25] της Atmel. Η λειτουργία αυτής της πλακέτας είναι να δίνει την πραγματική ημερομηνία και ώρα, όταν της ζητηθεί, στην πλακέτα web server.

Ο λόγος που σχεδιάστηκε ξεχωριστή πλακέτα για την ημερομηνία και ώρα είναι η μη απασχόληση της πλακέτας του διακομιστή διαδικτύου (web server) με την παρακολούθηση της ώρας ώστε να μειώνεται η πιθανότητα ο διακομιστής διαδικτύου να κολλήσει γιατί πρέπει να συνεργάζονται τέλεια η πλακέτα του χρονομέτρου πραγματικού χρόνου (real time clock), οι εισοδοί από την πλακέτα μετρήσεων και φυσικά το δίκτυο TCP/IP το οποίο δεν είναι ιδιαίτερα αξιόπιστο σε τέτοιου είδους πλακέτες. Στις περισσότερες πλακέτες, το τελευταίο επιλύεται με μια απλή επανεκκίνηση (reset). Γεγονός όμως που δεν μπορεί να εφαρμοσθεί εύκολα σε ένα σύστημα, που βρίσκεται μακριά όπως στην περίπτωση του παρόντος διακομιστή διαδικτύου. Ο λόγος που δεν χρησιμοποιήθηκε ένα έτοιμο ολοκληρωμένο πραγματικού χρόνου[16] είναι επειδή αυτά χρησιμοποιούν σειριακή επικοινωνία, η χρησιμοποίηση της οποίας είναι λιγότερη αξιόπιστη από την παράλληλη επικοινωνία που επιλέχθηκε



στο συγκεκριμένο σύστημα μειώνοντας έτσι ακόμα περισσότερο την πιθανότητα κολλήματος.

Στην εικόνα 3.4 φαίνεται το σχηματικό διάγραμμα της εν λόγω πλακέτας.

Διακρίνονται οι δύο μικροελεγκτές ATMEGA128 της εταιρίας Atmel. Στον κύκλο 1 φαίνεται ο μικροελεγκτής ο οποίος χρησιμοποιείται για να παρέχει στο σύστημα την ημερομηνία και ώρα. Συνδέεται επίσης με φωτοδιόδους έτσι ώστε να μπορεί να γίνει η ρύθμιση της ημερομηνίας και ώρας. Στον κύκλο 2 φαίνεται ο δεύτερος ίδιος μικροελεγκτής ο οποίος χρησιμοποιείται για να παίρνει την ημερομηνία και ώρα από τον πρώτο, να δίνει εντολή να ξεκινήσει η διαδικασία των μετρήσεων των παραμέτρων του νερού, να αποθηκεύει τα δεδομένα των μετρήσεων και να τα στέλνει στον διακομιστή διαδικτύου. Και σ'αυτόν τον μικροελεγκτή είναι συνδεδεμένοι φωτοδιόδοι για να είναι πιο εύκολος ο έλεγχος της πλακέτας.

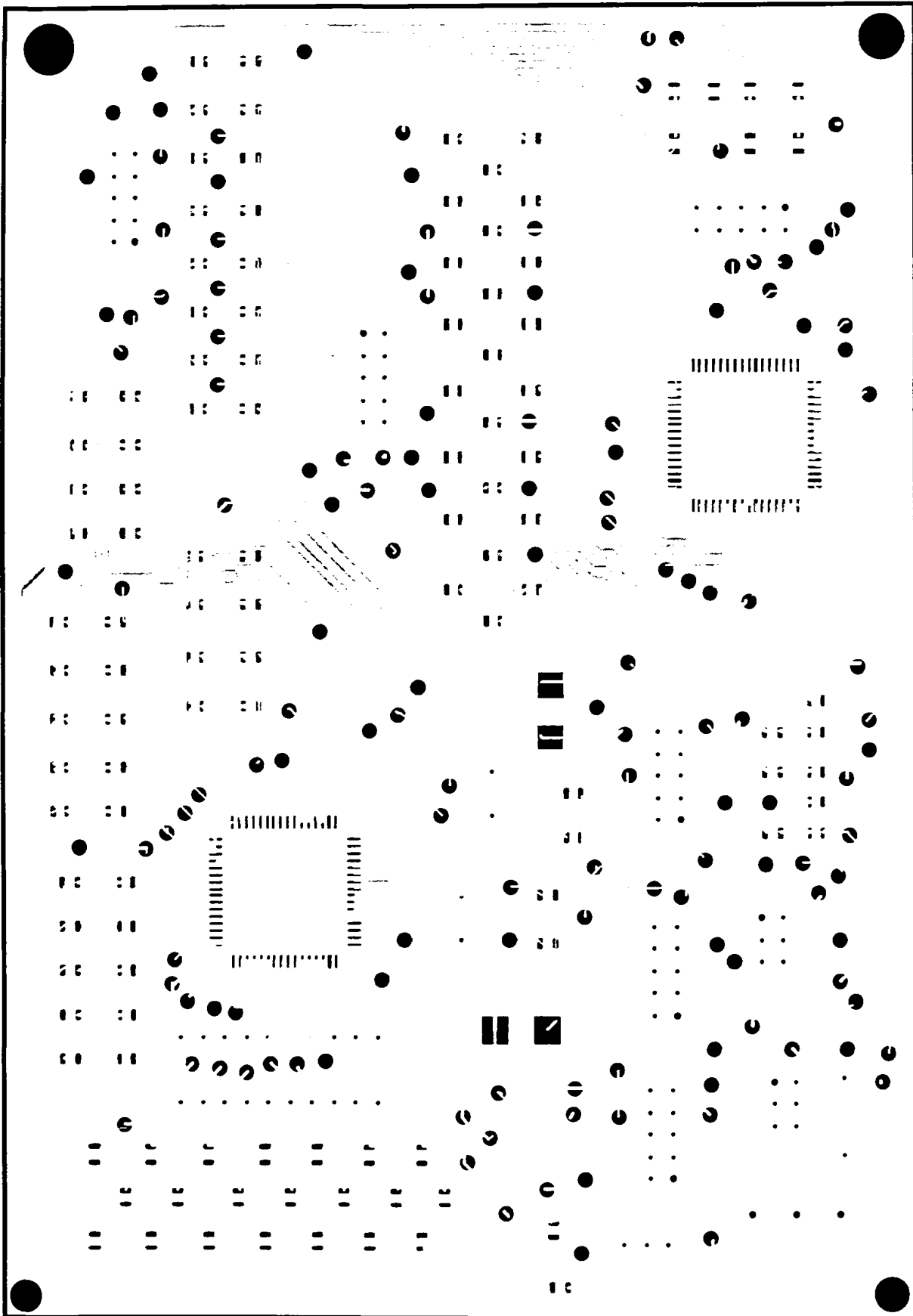
Στην εικόνα 3.5 φαίνεται η πάνω όψη της πλακέτας.

Στην εικόνα 3.6 φαίνεται η κάτω όψη της πλακέτας.

Στην εικόνα 3.7 φαίνεται η φωτορεαλιστική μορφή της πλακέτας της πάνω πλευράς.

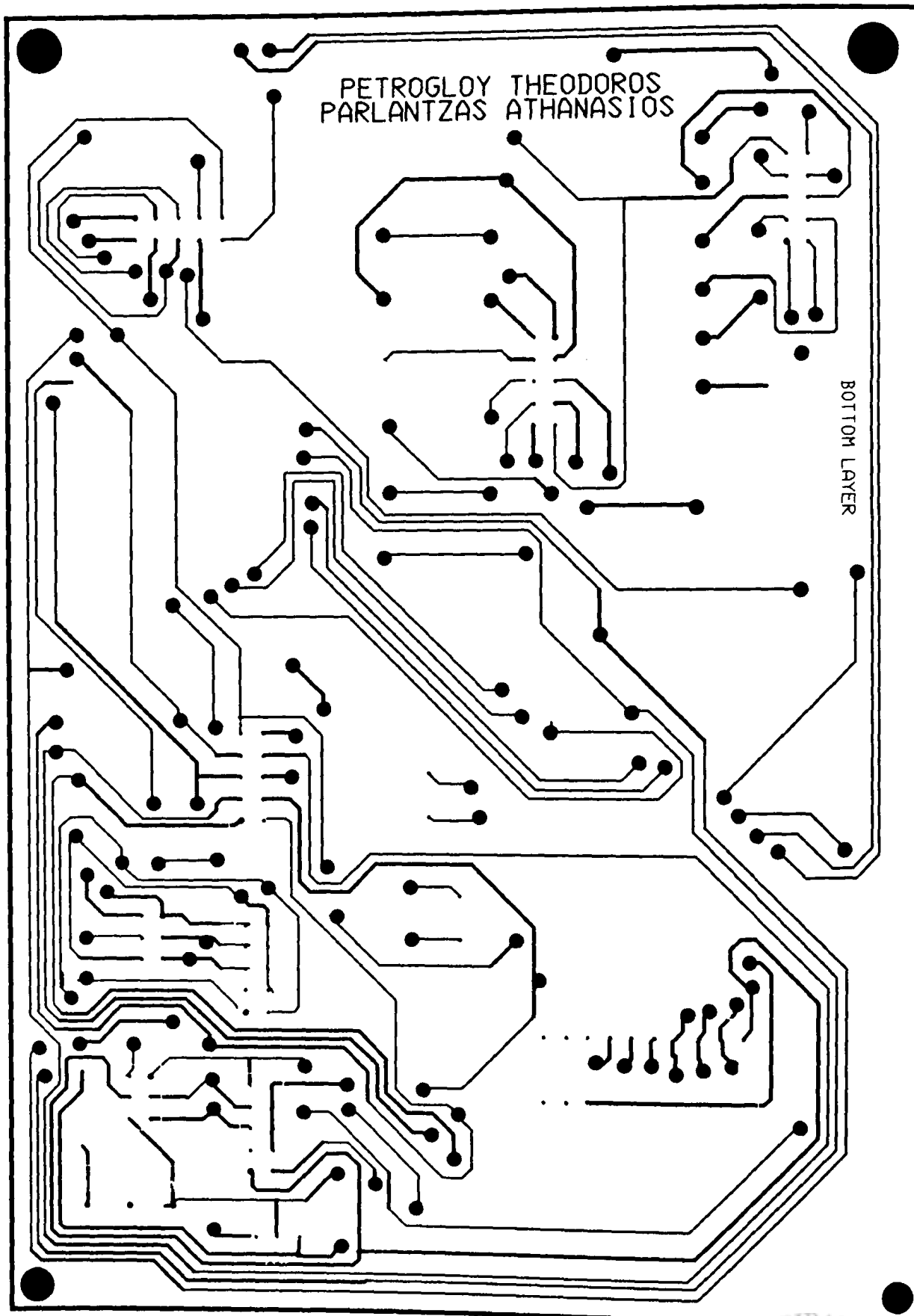
Στην εικόνα 3.8 φαίνεται η φωτορεαλιστική μορφή της πλακέτας της κάτω πλευράς.





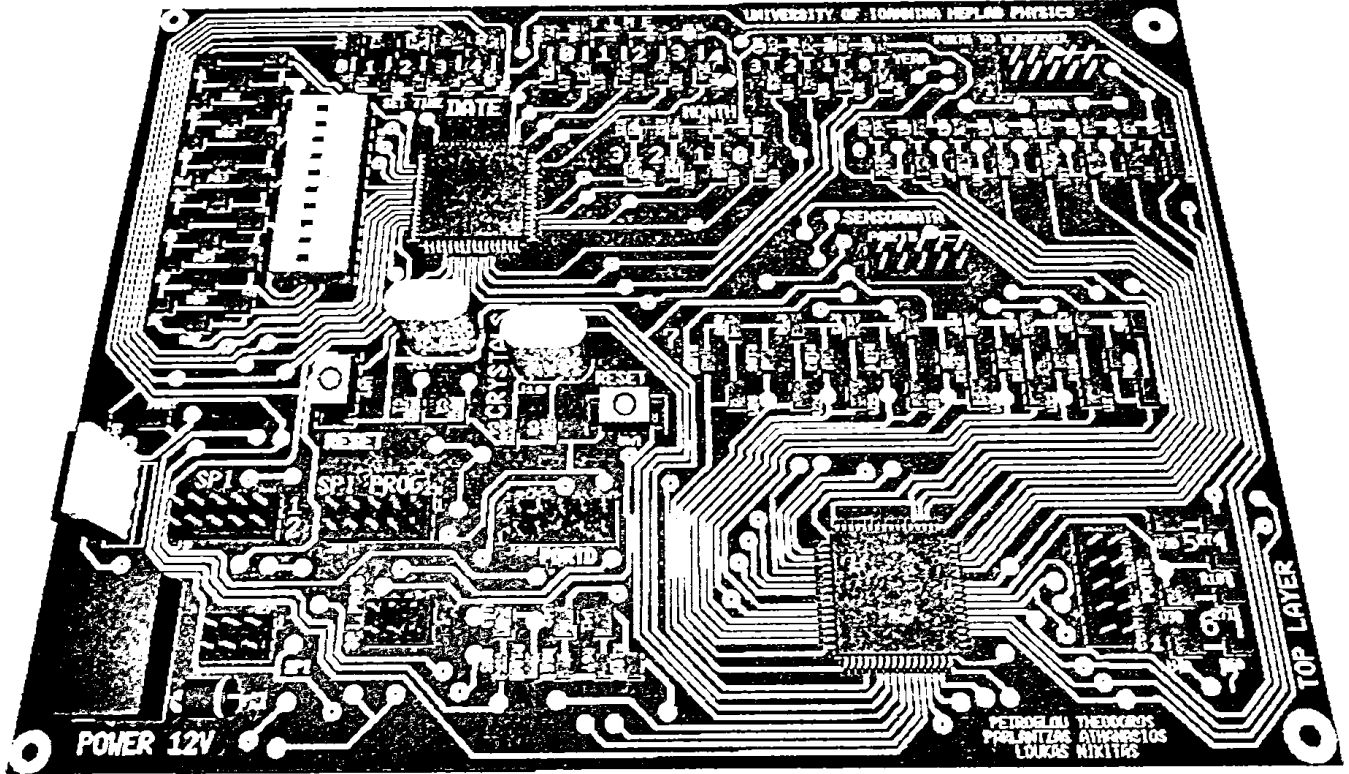
Εικόνα 3.5 Η πάνω όψη της πλακέτας πραγματικού χρόνου - διασύνδεσης του διακομιστή διαδικτύου με το σύστημα ανάγνωσης των αισθητήρων



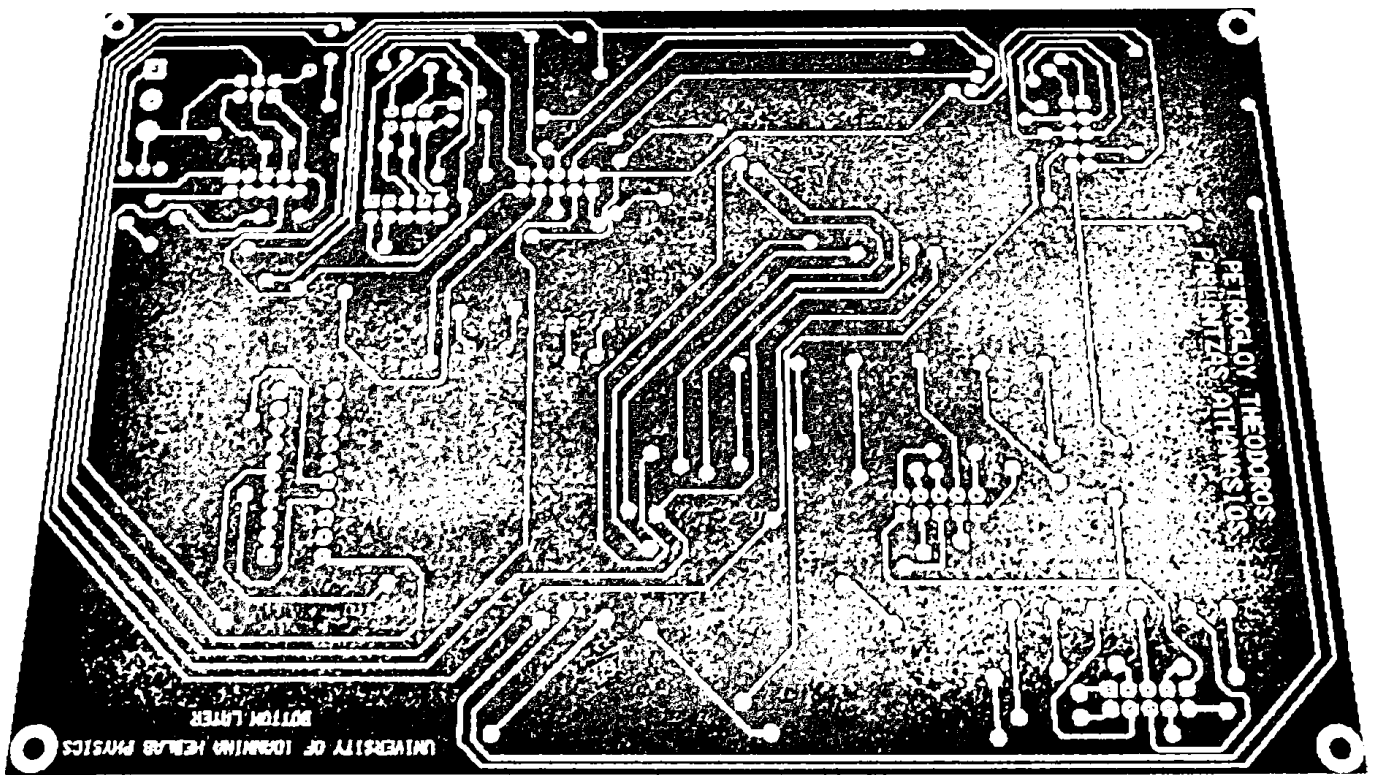


Εικόνα 3.6 Η κάτω όψη της πλακέτας πραγματικού χρόνου - διασυνδεσης του διακομιστή διαδικτύου με το σύστημα ανάγνωσης των αισθητήρων





Εικόνα3.7 Φωτορεαλιστική μορφή της πλακέτας της πάνω πλευράς

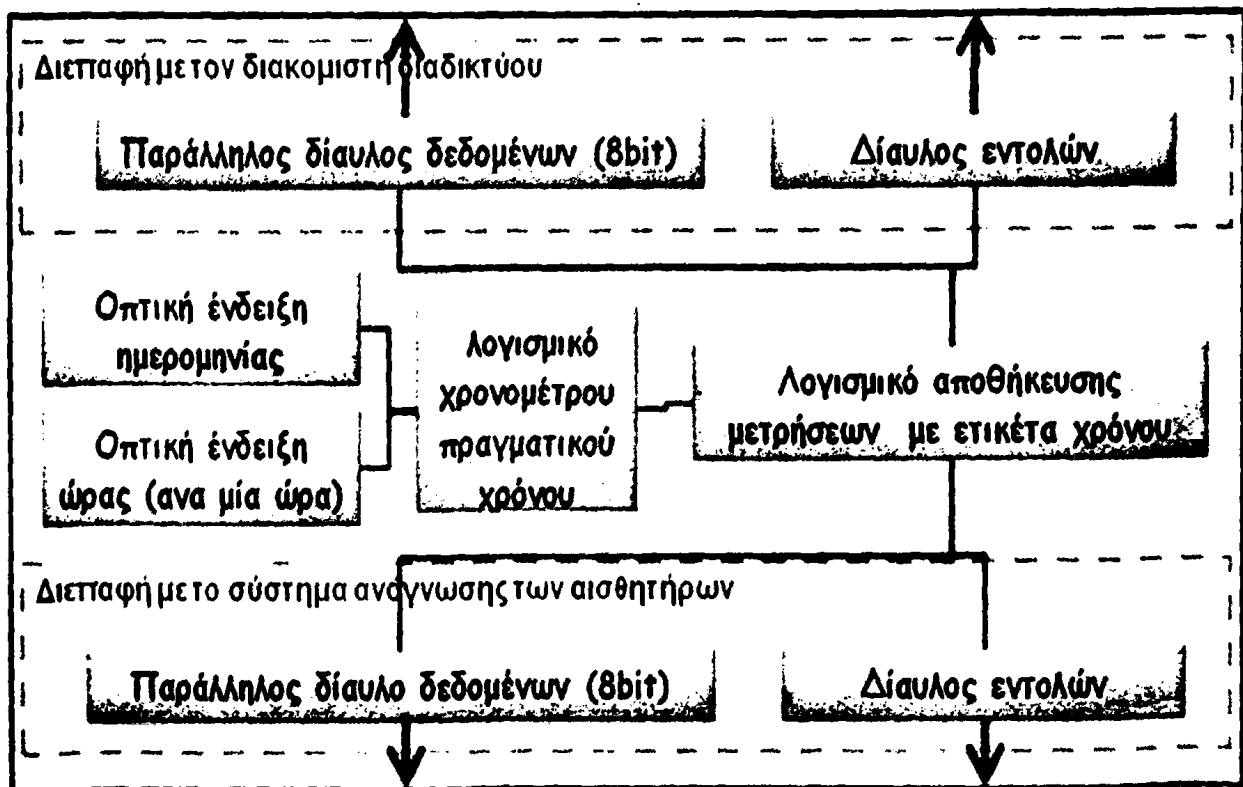


Εικόνα3.8 Φωτορεαλιστική μορφή της πλακέτας της κάτω πλευράς



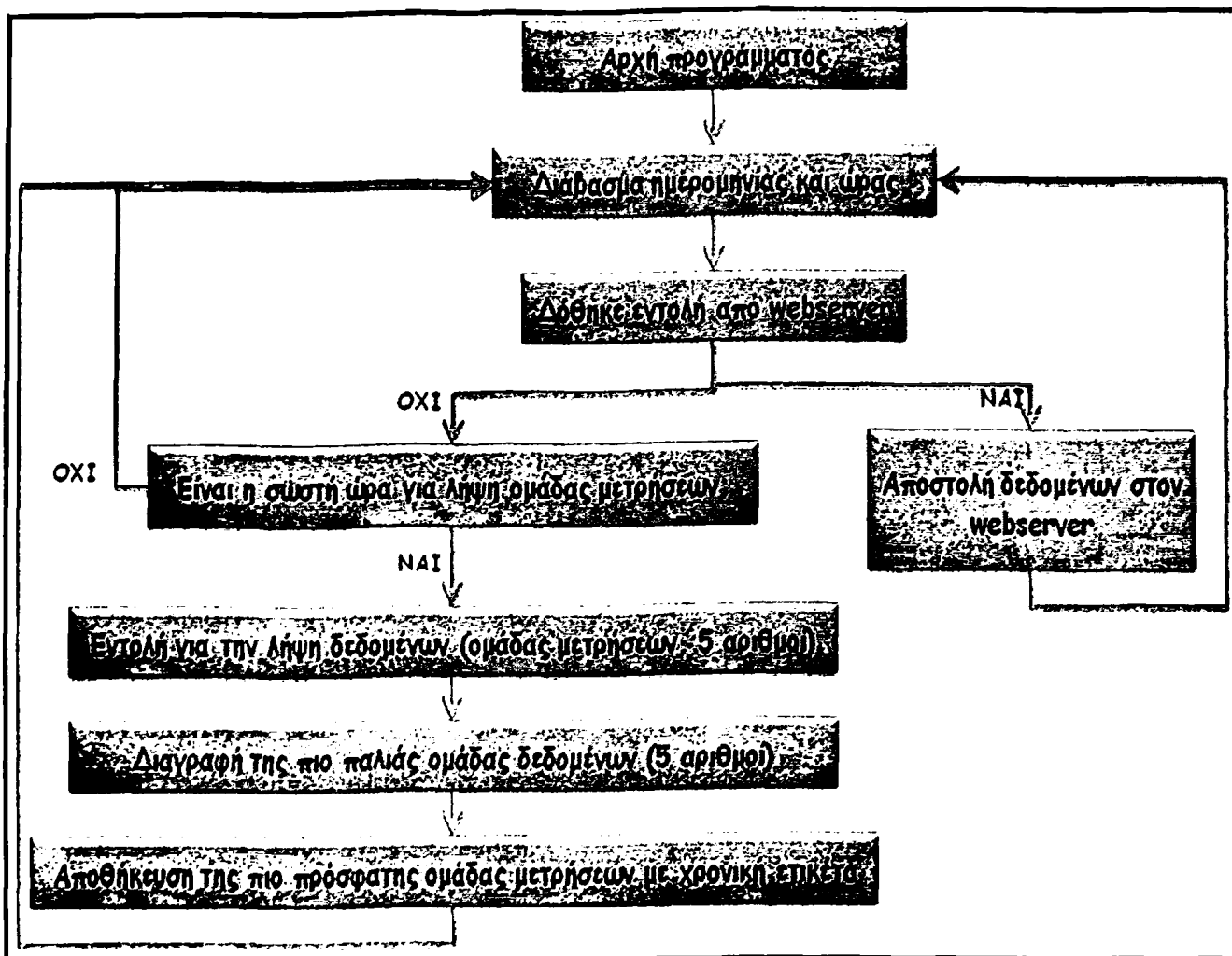
Αρχικά ρυθμίζεται η αρχική ώρα και ημερομηνία. Έπειτα εκτελείται καθυστέρηση μίας ώρας. Κατά συνέπεια, το συντομότερο δυνατόν που μπορούμε να ρυθμίσουμε το σύστημα να παίρνει μετρήσεις, είναι μία ώρα.

Αφού τελειώσει η καθυστέρηση της μιας ώρας, αυξάνει η ώρα κατά ένα ή γίνεται ένα αν η τελευταία ώρα ήταν εικοσιτέσσερα. Έπειτα ελέγχει τον ημερομηνία και ανάλογα αν η προηγούμενη ώρα ήταν εικοσιτρία τότε αυξάνει την ημερομηνία κατά ένα ή την κάνει ένα αν ήταν η τελευταία μέρα του μήνα. Αν το τελευταίο ισχύει, τότε αυξάνει τον μήνα κατά ένα. Εκτελείται μία ώρα καθυστέρηση και το πρόγραμμα επαναλαμβάνεται.



Εικόνα 3.5 Γενικό διάγραμμα της πλακέτας «χρονομέτρου πραγματικού χρόνου, διασύνδεσης με τον διακομιστή διαδικτύου και το σύστημα αποστολής δεδομένων από τους αισθητήρες»





Εικόνα 3.6 Διάγραμμα της διαδικασίας λειτουργίας της πλακέτας «χρονόμετρου πραγματικού χρόνου, διασύνδεσης με τον διακομιστή διαδικτύου και αποστολής δεδομένων από τους αισθητήρες»

Το συγκεκριμένο μέρος της πλακέτας δε χρειάζεται να έχει καμία είσοδο καθώς ακόμη και ο κρύσταλλος είναι εσωτερικός. Αντίθετα έχει πολλές εξόδους. Έχει πέντε bit που απεικονίζουν την συγκεκριμένη ώρα που βρίσκεται το ρολόι, πέντε bit που απεικονίζουν την ημέρα του μήνα και άλλα τέσσερα bit που απεικονίζουν σε ποιο μήνα βρίσκεται το ρολόι. Όλα είναι σε δυαδική μορφή.

Αυτό το μέρος της πλακέτας αποτελείται από έναν μικροελεγκτή ATMEΛ ATMEGA128, αντιστάσεις και led σε σειρά συνδεδεμένα σε κάθε έξοδο του μικροελεγκτή έτσι ώστε να μπορεί να γίνεται οπτικός έλεγχος για την ημερομηνία που βρίσκεται το πρόγραμμα, δηλαδή μπορεί να ελεγχθεί αν απεικονίζεται η σωστή ημερομηνία και ώρα και κατά συνέπεια αν το χρονόμετρο πραγματικού χρόνου



λειτουργεί σωστά. Οι αντιστάσεις που είναι συνδεδεμένες με τα led δεν χρειάζεται να έχουν κάποια συγκεκριμένη τιμή αλλά εξαρτώνται από την επιθυμητή φωτεινότητα που επιθυμεί ο χρήστης. Έτσι όσο πιο μικρή είναι η αντίσταση των led, τόσο πιο μεγάλη η φωτεινότητα. Συνήθως οι αντιστάσεις αυτές είναι της τάξης των μερικών εκατοντάδων Ωμ.

3.3 Διασύνδεση του συστήματος του διακομιστή διαδικτύου με τον αναγνώστη δεδομένων των αισθητήρων.

Ο διακομιστής διαδικτύου συνδέεται με την πλακέτα real time clock - διασύνδεσης του διακομιστή διαδικτύου με το σύστημα ανάγνωσης αισθητήρων. Δηλαδή η πλακέτα αυτή χρησιμοποιώντας την ημερομηνία και ώρα δίνει εντολή στο σύστημα ανάγνωσης αισθητήρων να πάρει μετρήσεις, έπειτα τις αποθηκεύει και όταν ο χρήστης του απομακρυσμένου υπολογιστή με τον οποίο είναι συνδεδεμένος ο διακομιστής διαδικτύου, χρησιμοποιήσει έναν περιηγητή για να δει τις τελευταίες αποθηκευμένες μετρήσεις, τότε ο διακομιστής διαδικτύου θα δώσει εντολή στην πλακέτα να του στείλει τις μετρήσεις και με τη σειρά του ο διακομιστής διαδικτύου θα τις στείλει στον απομακρυσμένο υπολογιστή. Αν τη συγκεκριμένη χρονική στιγμή που ζητήσει ο χρήστης να δει τις μετρήσεις, το σύστημα βρίσκεται σε διαδικασία μέτρησης, τότε εμφανίζεται στην οθόνη ένα μήνυμα που προτρέπει τον χρήστη να δοκιμάσει αργότερα. Από 'κει μπορεί να γίνει οποιαδήποτε επεξεργασία ή αποθήκευση των δεδομένων.

Η διασύνδεση του διακομιστή διαδικτύου με την πλακέτα real time clock και αποστολής δεδομένων από τους αισθητήρες, γίνεται με έντεκα ακροδέκτες. Οι οχτώ χρησιμοποιούνται για την λήψη των δεδομένων από τον αναγνώστη δεδομένων και άλλοι τρεις χρησιμοποιούνται για να επιτευχθεί η σωστή επικοινωνία μεταξύ τους χωρίς σφάλματα. Σε κάθε μέτρηση η συσκευή που πήρε την μέτρηση επιβεβαιώνει την λήψη από την άλλη συσκευή και τέλος επιβεβαιώνουν και οι δύο ότι είναι έτοιμες να συνεχίσουν την επικοινωνία.

Το μέρος της πλακέτας που συνδέεται με τον διακομιστή διαδικτύου, αποτελείται από έναν επεξεργαστή ATMEGA128, από led σε σειρά με αντιστάσεις στις εξόδους



του επεξεργαστή έτσι ώστε να γίνεται εύκολα έλεγχος της κατάστασης εξόδου, από led σε σειρά με αντιστάσεις στις εισόδους του επεξεργαστή έτσι ώστε να ελέγχεται η κατάσταση των εισόδων.



4. Το λογισμικό του συστήματος του διακομιστή διαδικτύου

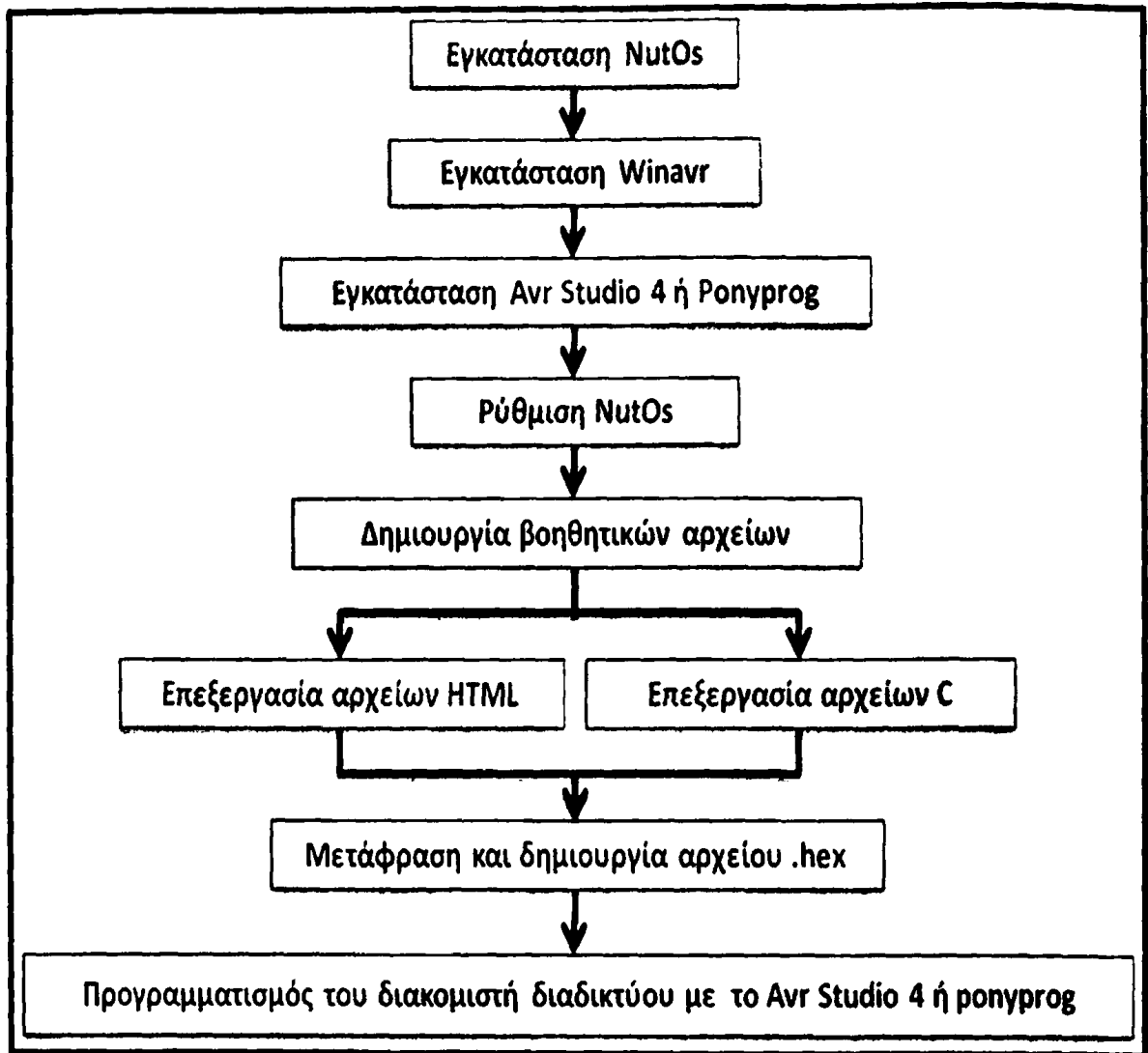
4.1 Λογισμικό για τον προγραμματισμό του διακομιστή διαδικτύου

Η πλακέτα του διακομιστή διαδικτύου είναι της εταιρίας Ethernut και είναι το μοντέλο Ethernut Starter Kit Rev 1.3H. Αγοράσθηκε από την ελεύθερη αγορά και προσφέρει ένα σύνολο από πακέτα λογισμικού με τα οποία μπορεί να γίνει εγγραφή ενός προγράμματος, διόρθωση και μετατροπή (compilation) και στη συνέχεια κατέβασμα του τελικού προγράμματος στον μικροελεγκτή ATMEGA128 της πλακέτας του διακομιστή διαδικτύου.

Η εγγραφή ενός καινούριου προγράμματος που έχει την δυνατότητα να συνεργάζεται με όλα τα περιφερειακά της πλακέτας, όπως το ολοκληρωμένο της εταιρίας Realtek[9] για την σύνδεση σε δίκτυο TCP/IP, ή οι αισθητήρες θερμοτήτας, (προσφέρονται προαιρετικά με την αγορά της πλακέτας) είναι δύσκολο να επιτευχθεί. Γι' αυτό το λόγο μαζί με το λογισμικό υπάρχουν και κάποια βασικά κομμάτια του προγράμματος τα οποία είναι έτοιμα και μπορούν να χρησιμοποιηθούν συμπληρωματικά με το πρόγραμμα της εφαρμογής έτσι ώστε να υπάρχει συνεργασία και με τα περιφερειακά της πλακέτας. Στην εικόνα 4.1 φαίνεται το διάγραμμα ροής για την δημιουργία του τελικού αρχείου το οποίο φορτώνεται (downloading) στον μικροελεγκτή.

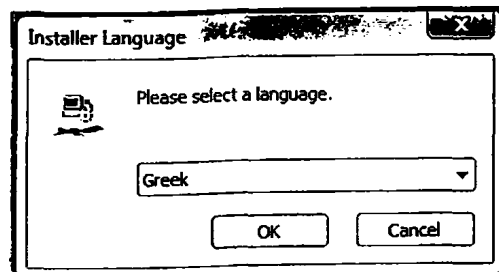
Το λογισμικό που προσφέρεται με την αγορά της πλακέτας του διακομιστή διαδικτύου είναι το NutOs[10], ένας GNU C compiler, δηλαδή το WinAVR[24], και δύο προγράμματα κατεβάσματος (downloading) του προγράμματος στον μικροελεγκτή, το Ronprog[26] και το Avr Studio 4[22], το οποίο και χρησιμοποιήθηκε.





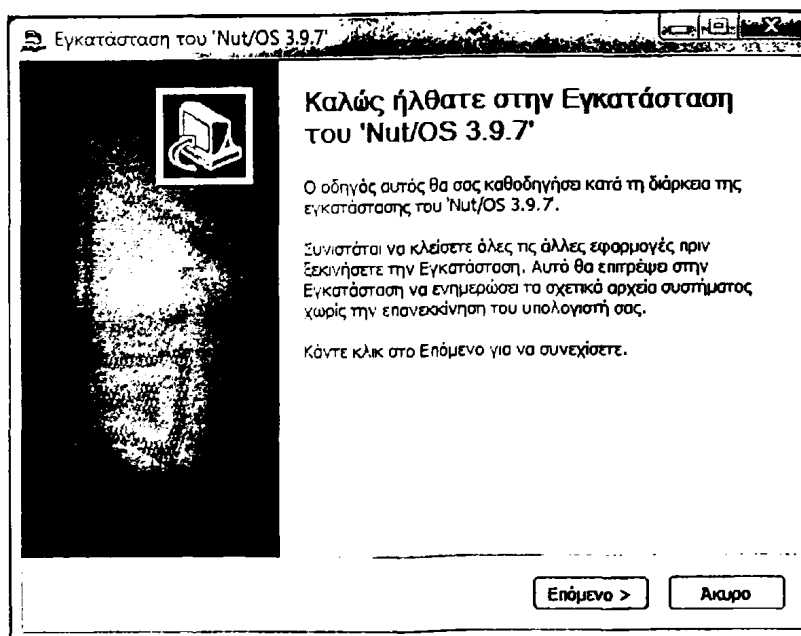
Εικόνα 4.1 Διαδικασία προγραμματισμού του διακομιστή διαδικτύου (web server)

Αρχικά πρέπει να γίνει η εγκατάσταση το NutOs, ή η τελευταία έκδοση του NutOs που διατίθεται ελεύθερα στο διαδίκτυο[10]. Επιλέγεται το αρχείο ethernut-3.9.7 (ethernut-x.x.x) όπου x.x.x. η έκδοση του NutOs. Μόλις υπάρχει συμφωνία με το μήνυμα ασφαλείας εμφανίζεται το παράθυρο που φαίνεται στην εικόνα 4.2 επιλέγουμε την ελληνική γλώσσα.

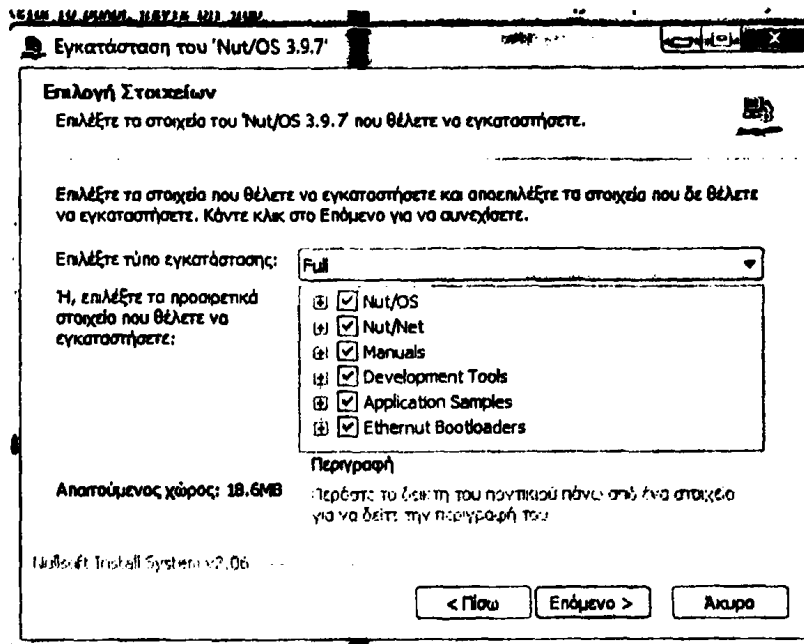


Εικόνα 4.2 Παράθυρο επιλογής γλώσσας για την εγκατάσταση του NutOs

Έπειτα εμφανίζεται το παράθυρο της εικόνας 4.3 και επιλέγεται επόμενο. Στη συνέχεια εμφανίζεται το παράθυρο της εικόνας 4.4. Αφού γίνει ο έλεγχος ότι όλα τα στοιχεία είναι τσεκαρισμένα και ο τύπος της εγκατάστασης είναι full, επιλέγεται επόμενο.

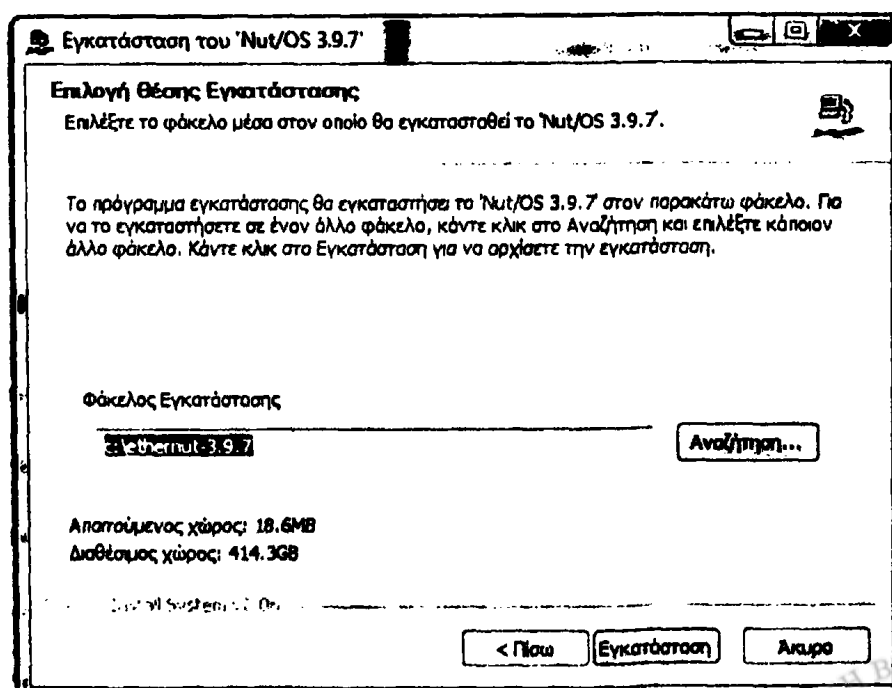


Εικόνα 4.3 Παράθυρο κατά την εγκατάσταση του NutOs



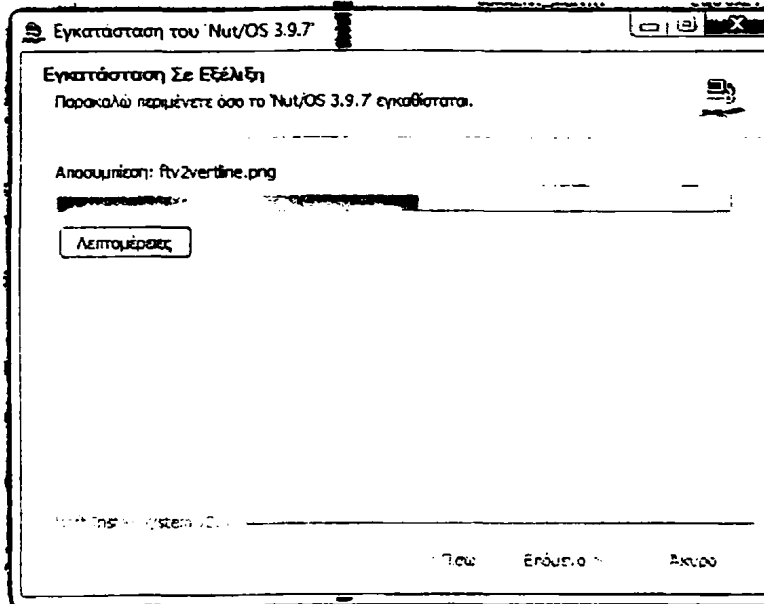
Εικόνα 4.4 Παράθυρο επιλογής στοιχείων κατά την εγκατάσταση του NutOs

Συνεχίζοντας εμφανίζεται το παράθυρο που ενημερώνει τον χρήστη σε πια διαδρομή του δίσκου θα γίνει η εγκατάσταση και φαίνεται στην εικόνα 4.5. Παρόλο που υπάρχει η δυνατότητα να αλλαχθεί η διαδρομή που θα γίνει εγκατάσταση, προτείνεται να διατηρηθεί η διαδρομή που υπάρχει ώστε να γίνει όσο το δυνατόν πιο εύκολη η εγκατάσταση. Επιλέγεται εγκατάσταση.

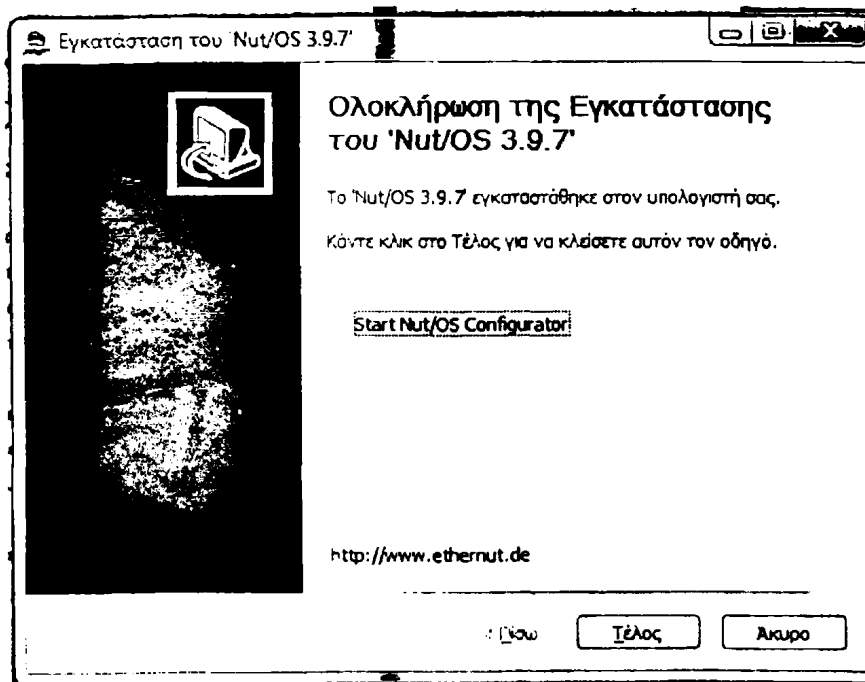


Εικόνα 4.5 Παράθυρο επιλογής διαδρομής δίσκου κατά την εγκατάσταση του NutOs

Αφού επιλεχθεί εγκατάσταση, εμφανίζεται για λίγο το παράθυρο της εικόνας 4.6 που μας ενημερώνει ότι η εγκατάσταση του NutOs πραγματοποιείται εμφανίζεται η πρόοδό της.

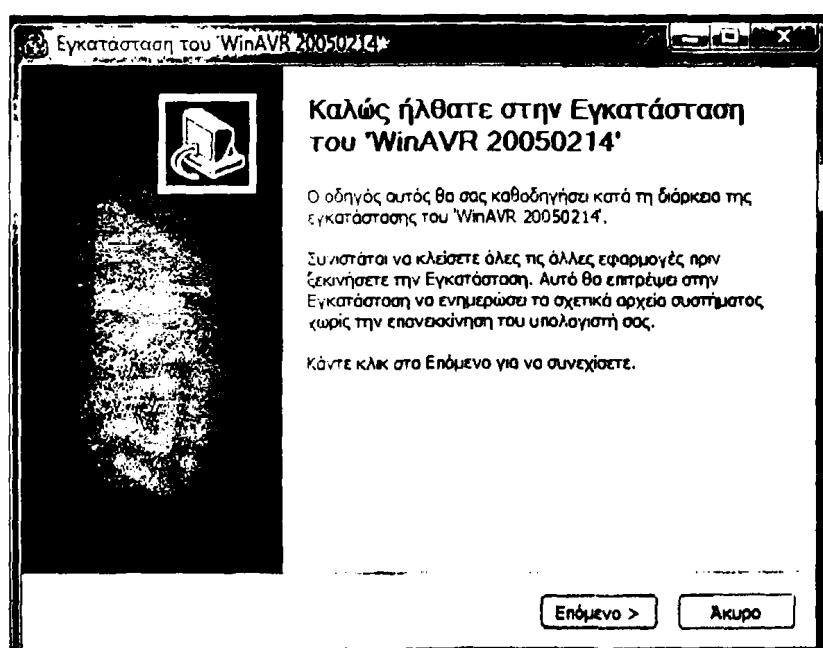


Εικόνα 4.6 Παράθυρο ενημέρωσης της πρόοδου εγκατάστασης του NutOs



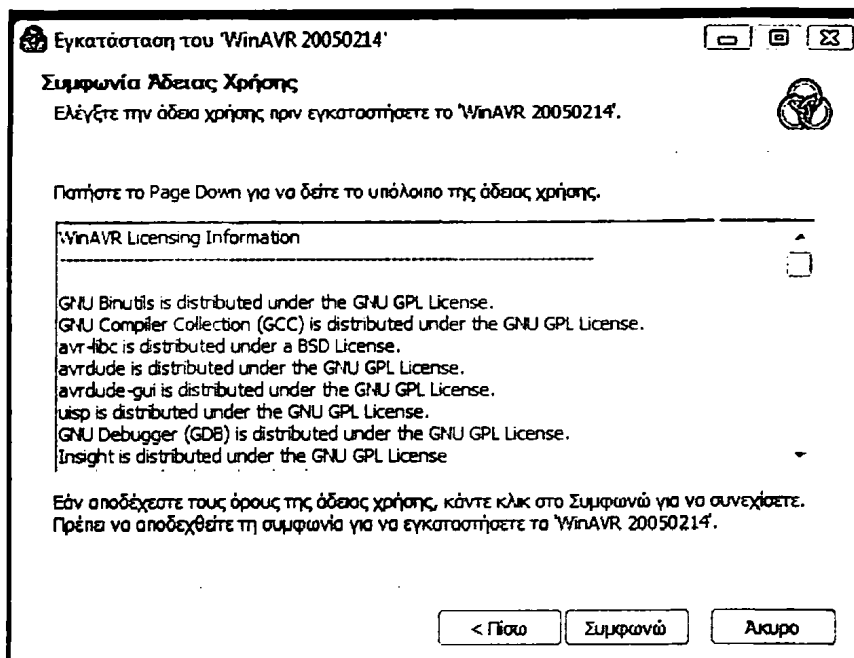
Εικόνα 4.7 Παράθυρο ολοκλήρωσης της εγκατάστασης του NutOs

Μόλις ολοκληρωθεί η εγκατάσταση του NutOs ακολουθεί η εγκατάσταση του AVR GNU C compiler, και συγκεκριμένα με τον compiler WinAVR που περιέχεται στο αρχείο WinAVR-20050214-install.exe ή οποιαδήποτε μεταγενέστερη έκδοση που διατίθεται ελεύθερα στο διαδίκτυο[11]. Με την ενεργοποίηση του συγκεκριμένου αρχείου εμφανίζεται το παράθυρο που ενημερώνει για την γλώσσα εγκατάστασης του WinAVR. Επιλέγεται η ελληνική (greek) και στη συνέχεια εμφανίζεται το παράθυρο της εικόνας 4.8.



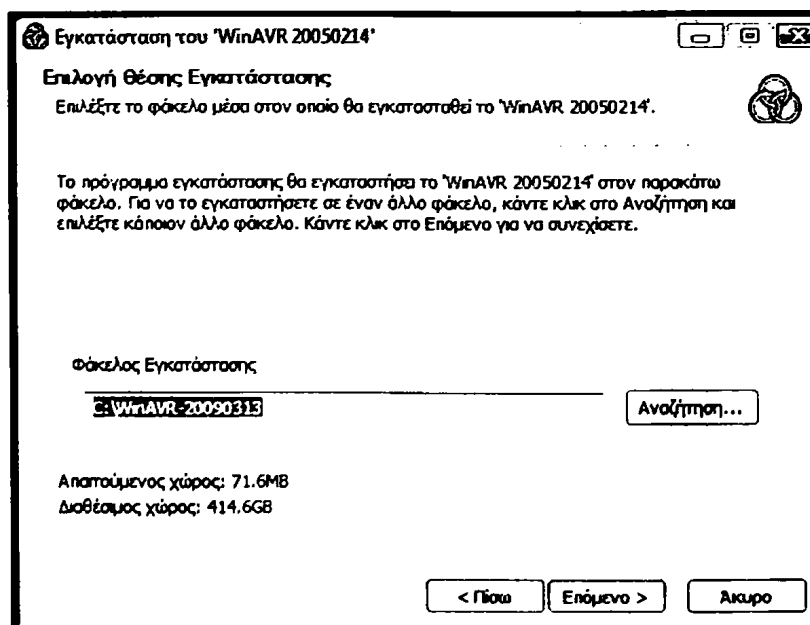
Εικόνα 4.8 Παράθυρο κατά την εγκατάσταση του WinAVR

Συνεχίζοντας εμφανίζεται το παράθυρο με την άδεια χρήσης του προγράμματος. Για να συνεχίσει η εγκατάσταση πρέπει να συμφωνήσει ο χρήστης στο παράθυρο που φαίνεται στην εικόνα 4.9.



Εικόνα 4.9 Παράθυρο άδειας χρήσης κατά την εγκατάσταση του WinAVR

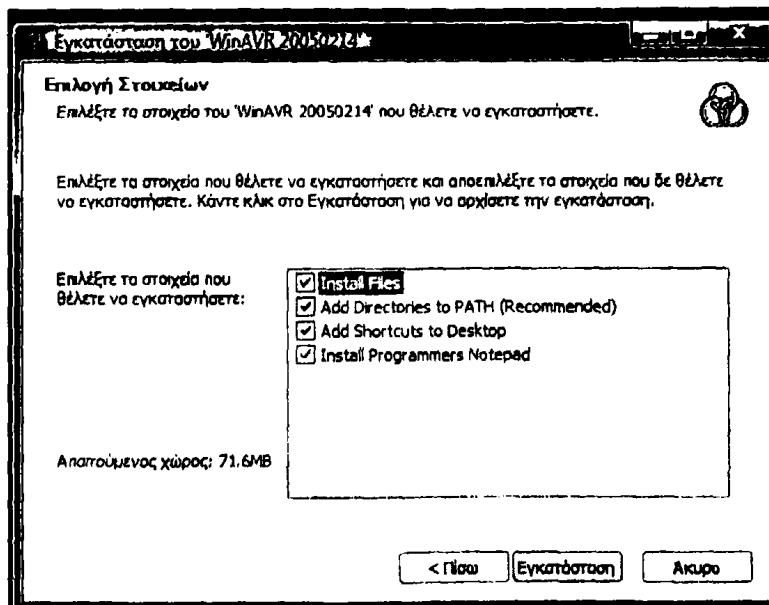
Μετά από το παράθυρο της άδειας χρήσης εμφανίζεται το παράθυρο της επιλογής θέσης εγκατάστασης του WinAVR, εικόνα 4.10. Όπως και με την εγκατάσταση του NutOs, υπάρχει η δυνατότητα να αλλάξει η διαδρομή του δίσκου που γίνεται η εγκατάσταση αλλά προτείνεται να μείνει η ίδια για τους λόγους που έχουν προαναφερθεί. Επιλέγεται επόμενο.



Εικόνα 4.10 Παράθυρο επιλογής θέσης εγκατάστασης κατά την εγκατάσταση του WinAVR

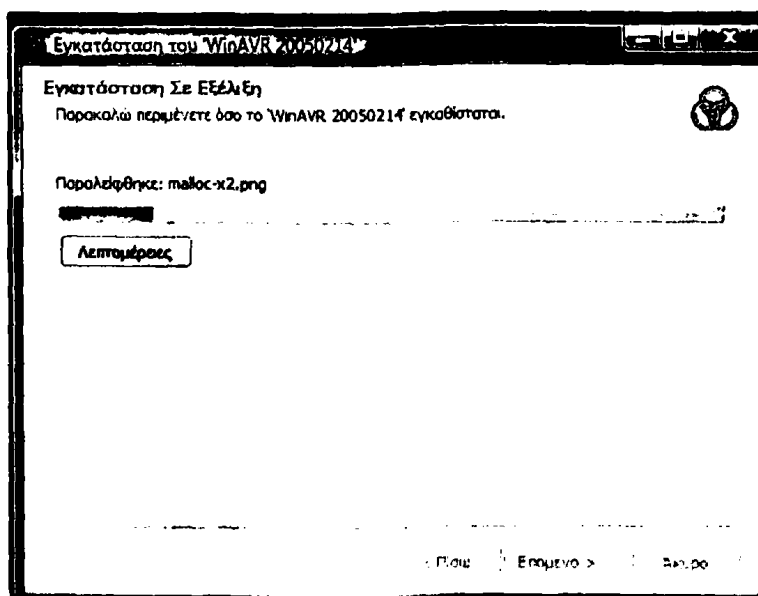


Στη συνέχεια εμφανίζεται το παράθυρο επιλογής στοιχείων προς εγκατάσταση, εικόνα 4.11. Αφού γίνει ο έλεγχος ότι όλα τα στοιχεία είναι τσεκαρισμένα, επιλέγεται εγκατάσταση.



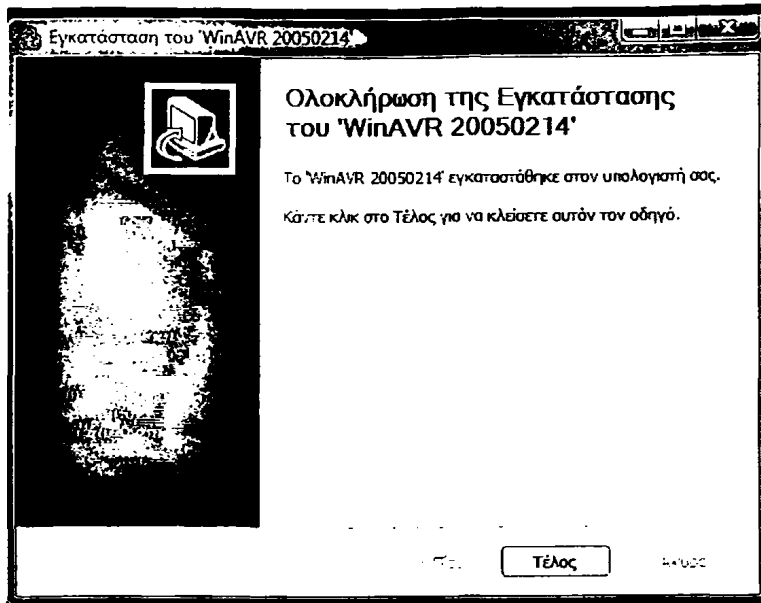
Εικόνα 4.11 Παράθυρο επιλογής στοιχείων προς εγκατάσταση, κατά την εγκατάσταση του WinAVR.

Μετά από αυτό, εμφανίζεται το παράθυρο που ενημερώνει για την εξέλιξη της εγκατάστασης, εικόνα 4.12 και ολοκληρώνεται η εγκατάσταση με το παράθυρο της εικόνας 4.13



Εικόνα 4.12 Παράθυρο εξέλιξης της εγκατάστασης του WinAVR

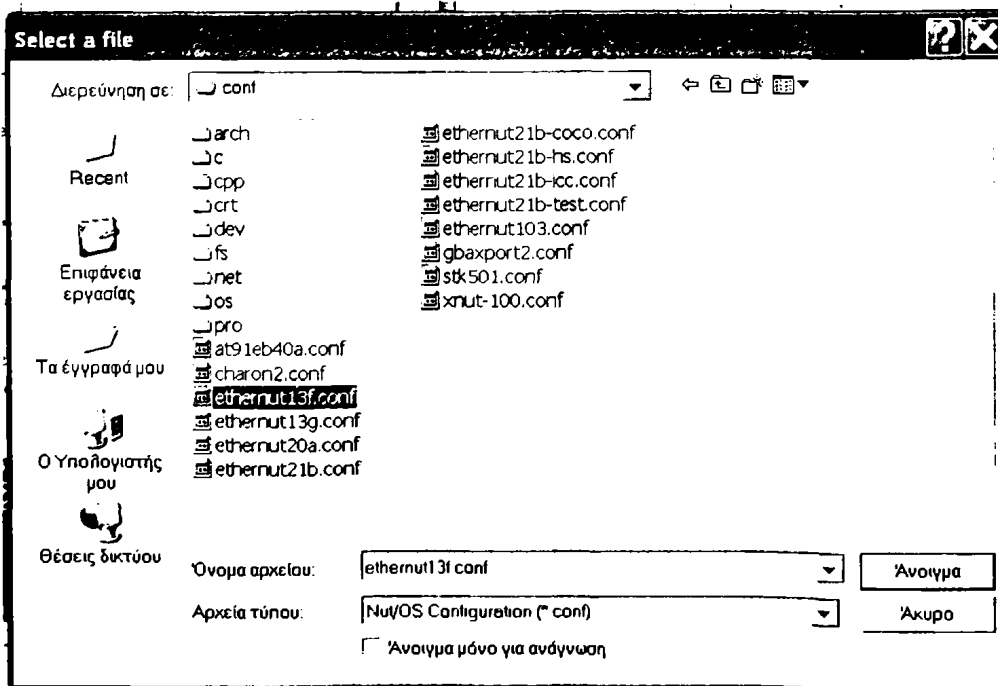




Εικόνα 4.13 Παράθυρο ολοκλήρωσης της εγκατάστασης του WinAVR

Αφού ολοκληρωθεί και η εγκατάσταση του WinAVR μπορεί στη συνέχεια να εγκατασταθεί και το Avr Studio 4 που είναι κοινό για όλους τους μικροεπεξεργαστές της εταιρίας Atmel. Έπειτα πρέπει να ρυθμιστεί σωστά το NutOs Configurator.

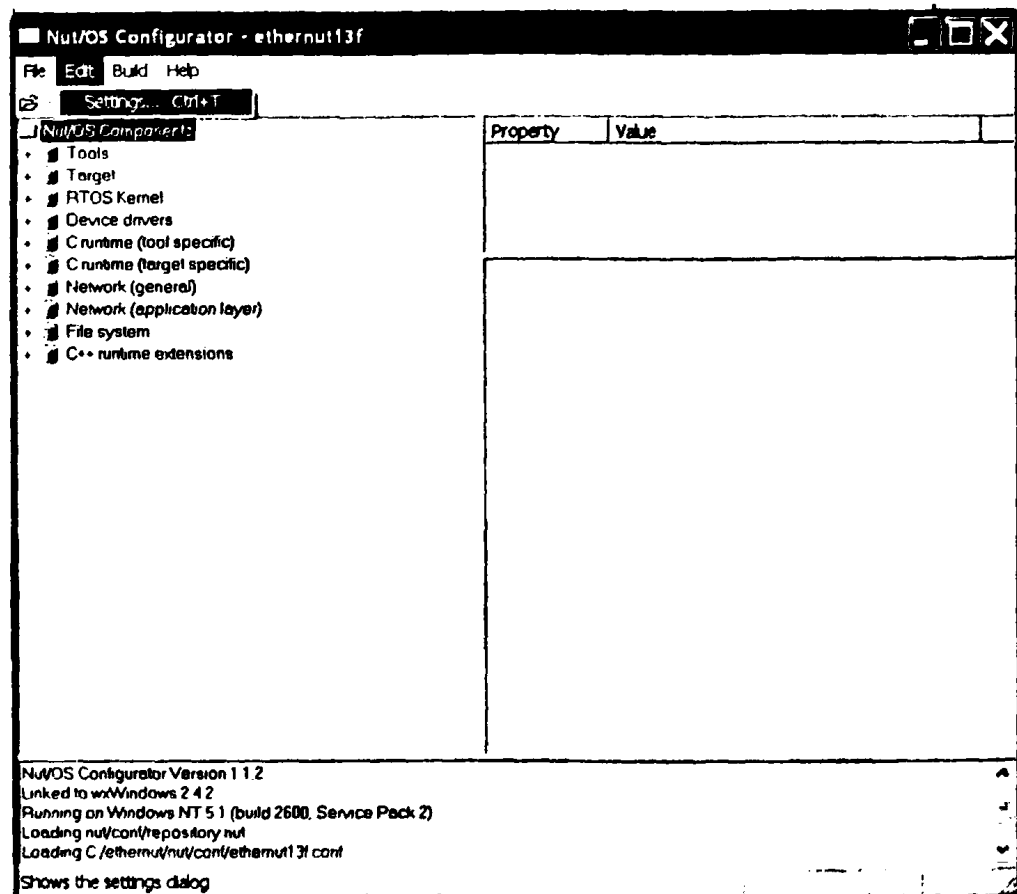
Ανοίγοντας για πρώτη φορά το NutOs Configurator από το μενού των windows, θα εμφανιστεί το παράθυρο της εικόνας 4.14.



Εικόνα 4.14 Παράθυρο επιλογής αρχείου ρυθμίσεων κατά το πρώτο άνοιγμα του NutOs configurator

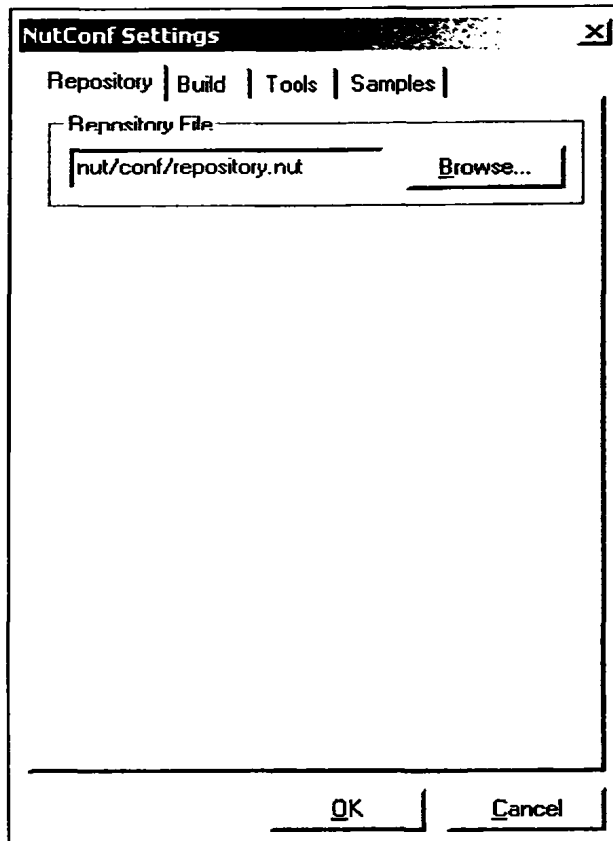


Επιλέγουμε την διαδρομή που έχει γίνει η εγκατάσταση, αν δεν έχουν αλλαχθεί οι προτεινόμενες διαδρομές κατά την εγκατάσταση, τότε επιλέγεται το αρχείο `c:\ethernut\nut\conf\ethernut13f.conf` και επιλέγεται άνοιγμα. Το αρχείο με τις ρυθμίσεις του υλικού (hardware) της συγκεκριμένης πλακέτας φορτώνεται και ανοίγει το NutOs configurator όπως φαίνεται στην εικόνα 4.15.



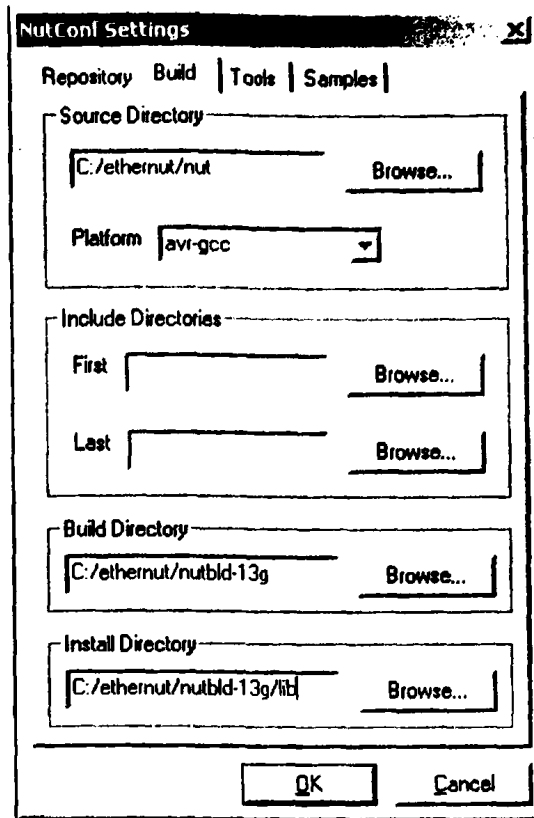
Εικόνα 4.15 Το παράθυρο του NutOs Configurator

Από το μενού edit επιλέγεται settings και εμφανίζεται το παράθυρο της εικόνας 4.16. Στην πρώτη καρτέλα Repository, στο πεδίο Repository File γράφει `nut/conf/repository.nut` το οποίο δεν πρέπει να αλλάξει.



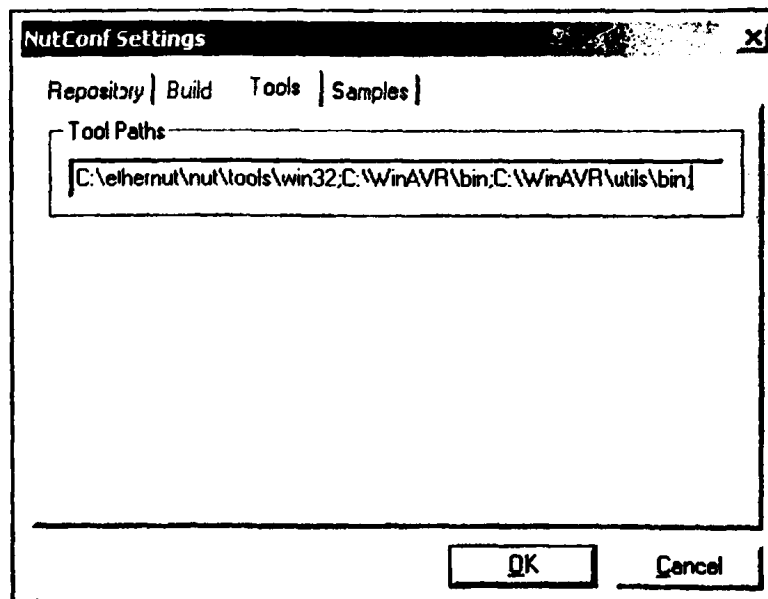
Εικόνα 4.16 Η καρτέλα Repository του παραθύρου NutConf Settings του NutOs Configurator

Συνεχίζοντας στην καρτέλα Build εμφανίζεται το παράθυρο της εικόνας 4.17. Στο πεδίο Source Directory πρέπει να γραφεί `c:/ethernut/nut`. Στο πεδίο Platform πρέπει να επιλεχθεί το `avr-gcc`. Το Include Directories μένει κενό. Στο πεδίο Build Directory πρέπει να γραφεί `c:/ethernut/nutbld-13g` και στο πεδίο Install Directory `c:/ethernut/nutbld-13g/lib`.



Εικόνα 4.17 Η καρτέλα Build του παραθύρου NutConf Settings του NutOs Configurator

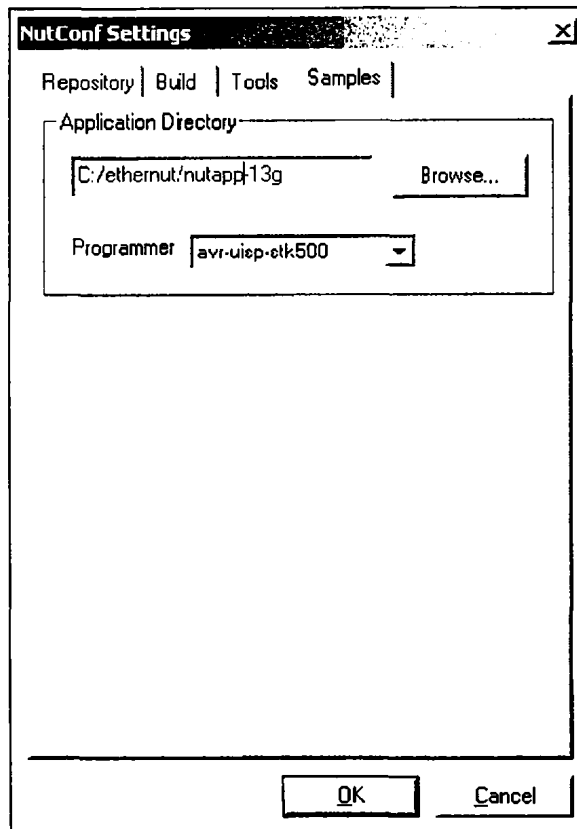
Επιλέγοντας την καρτέλα Tools εμφανίζεται το παράθυρο της εικόνας 4.18. Στο πεδίο Tool Paths πρέπει να γραφεί
`c:\ethernut\nut\tools\win32;c:\WinAVR\bin;c:\WinAVR\utils\bin;`



Εικόνα 4.18 Καρτέλα Tools του παραθύρου NutConf Settings του NutOs Configurator

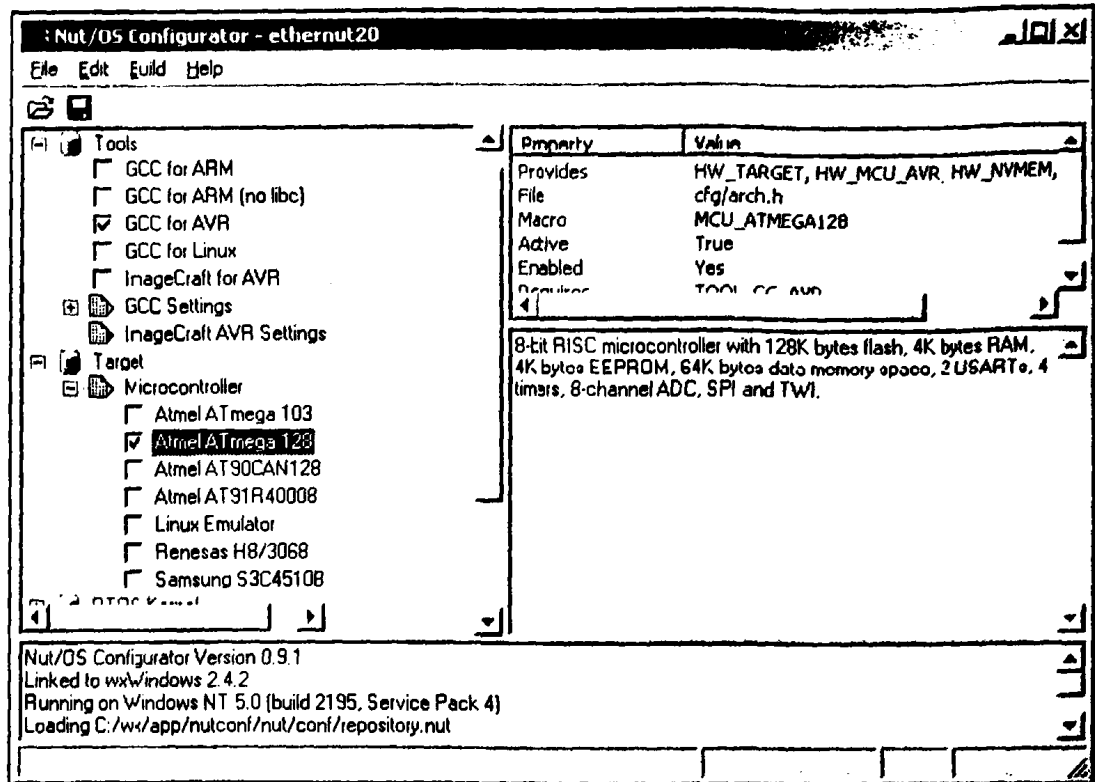


Επιλέγοντας την καρτέλα *Samples* εμφανίζεται το παράθυρο της εικόνας 4.19. Στο πεδίο *Application Directory* πρέπει να γραφεί `c:/ethernut/nutapp-13g` και στο πεδίο *Programmer* πρέπει να επιλεγθεί `avr-uisp-stk500` και επιλέγουμε **OK** τελειώνοντας τις ρυθμίσεις του παραθύρου *NutConf Settings*.



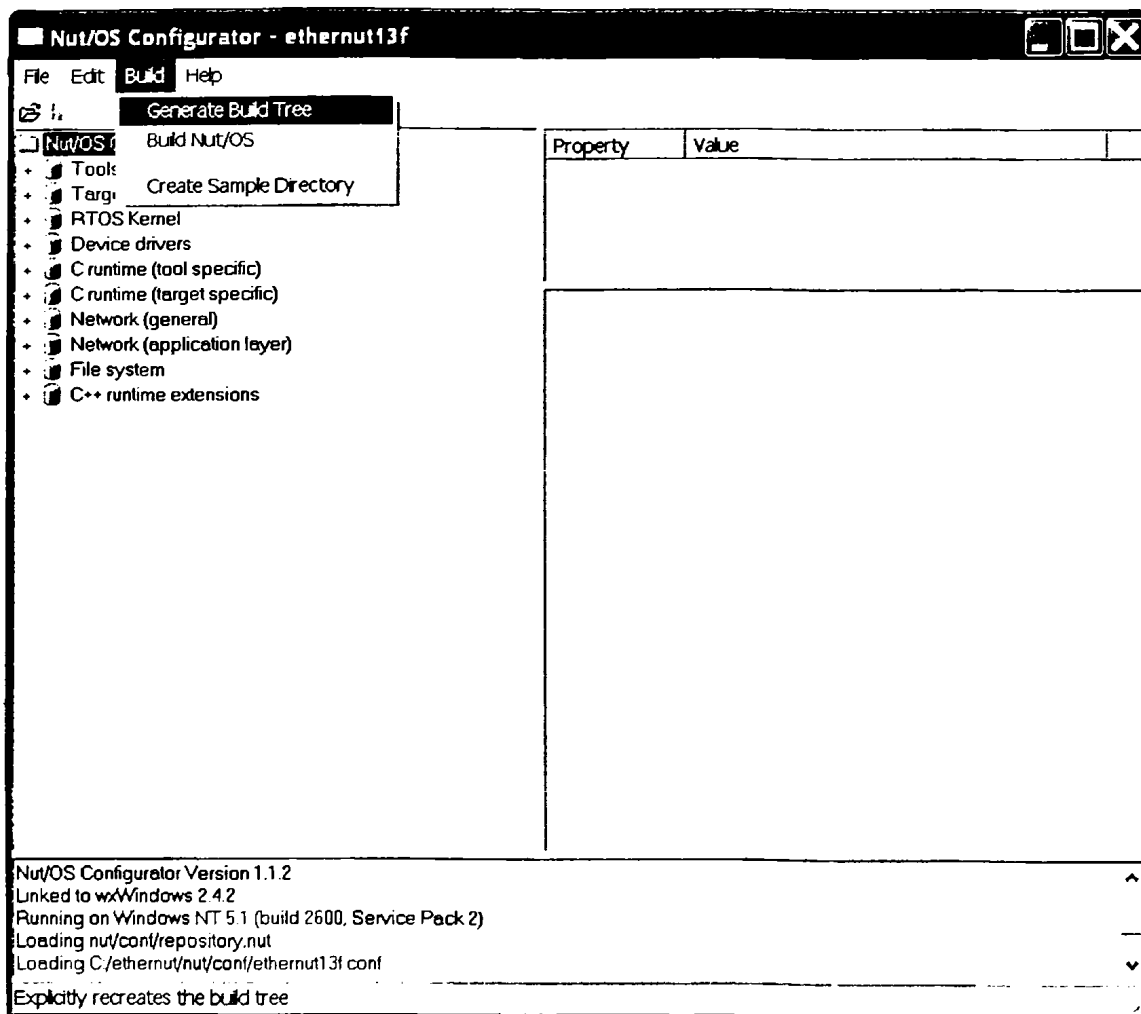
Εικόνα 4.19 Καρτέλα *Samples* του παραθύρου *NutConf Settings* του *NutOs Configurator*

Αφού πατηθεί το **OK** θα εμφανιστεί το κυρίως πρόγραμμα *NutOs Configurator*, εικόνα 4.20. Από το μενού *tools* επιλέγουμε το *GCC for AVR* και από το μενού *Target->Microcontroller* επιλέγουμε τον μικροελεγκτή *Atmel Atmega 128*.



Εικόνα 4.20 Το πρόγραμμα NutOs Configurator

Στη συνέχεια από το μενού Build επιλέγουμε *Generate Build Tree*, εικόνα 4.21. Με την εντολή αυτή δημιουργούνται μια ομάδα από αρχεία συμπεριλαμβανομένων και μερικές αρχικές παραμετροποιήσεις αρχείων *c*. Αυτά τα αρχεία χρησιμοποιούνται για να μπορεί το NutOs να προγραμματίσει την συγκεκριμένη πλακέτα με το συγκεκριμένο υλικό (*hardware*).



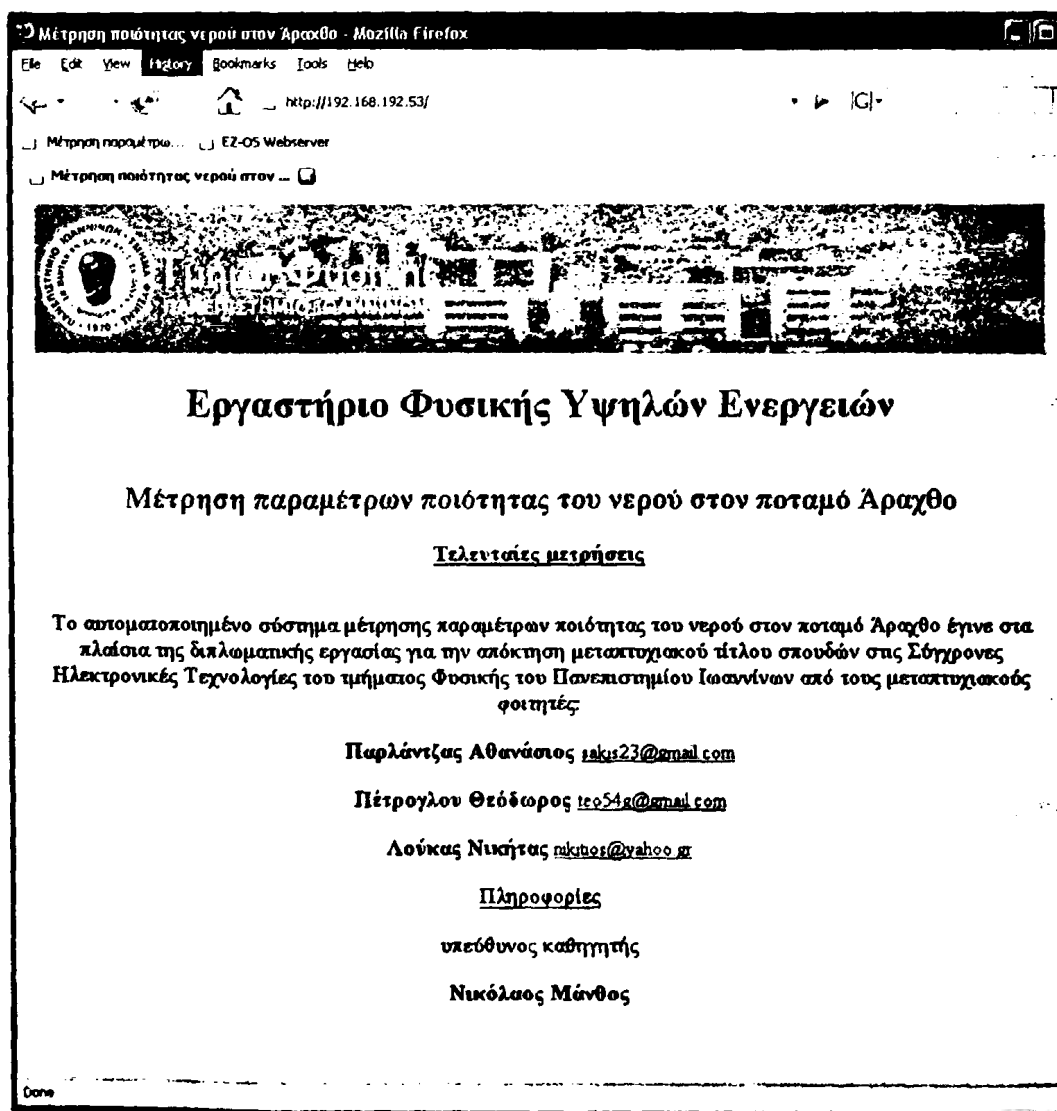
Εικόνα 4.21 Το μενού Generate Build Tree που χρησιμοποιείται για την δημιουργία των βοηθητικών αρχείων για τον προγραμματισμό του διακομιστή διαδικτύου

Στη συνέχεια επιλέγεται την εντολή Build Nut/OS και μετά Create Sample Directory. Μόλις ολοκληρωθεί και η τελευταία εντολή χωρίς σφάλματα σημαίνει ότι τα αρχεία που χρειάζονται έχουν δημιουργηθεί σωστά και έχουν αποθηκευθεί στην διαδρομή c:\ethernut\netbld-13g. Στον φάκελο html υπάρχει η αρχική σελίδα σε γλώσσα html, η οποία φορτώνεται όταν ο διακομιστής διαδικτύου συνδεθεί σε ένα δίκτυο TCP/IP και σε ένα περιηγητή διαδικτύου γραφεί η διεύθυνση IP του διακομιστή διαδικτύου. Το όνομα του αρχείου της αρχικής σελίδας είναι index.html. Επίσης δημιουργείται το αρχείο cgi.c, η χρησιμότητα του οποίου αναφέρεται εκτενέστερα σε επόμενο υποκεφάλαιο.

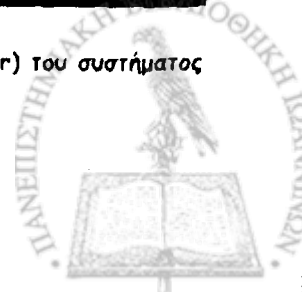


4.2 Αρχική σελίδα του διακομιστή διαδικτύου σε γλώσσα html

Όταν ο διακομιστής διαδικτύου είναι συνδεδεμένος σε δίκτυο TCP/IP και πληκτρολογηθεί η διεύθυνση IP του, εμφανίζεται η αρχική σελίδα html. Το αρχείο δηλαδή που εκτελείται είναι το index.html το οποίο ήδη έχει αναφερθεί πως δημιουργήθηκε και σε ποια θέση του δίσκου βρίσκεται. Επειδή είναι γραμμένο σε γλώσσα html, μπορεί να προσπελαστεί και να αλλάξει με οποιοδήποτε πρόγραμμα δημιουργίας ιστοσελίδων, όπως το Dreamweaver CS4[12] της εταιρίας Adobe το οποίο και χρησιμοποιήθηκε. Η αρχική σελίδα που δημιουργήθηκε για το συγκεκριμένο σύστημα φαίνεται στην εικόνα 4.22.



Εικόνα 4.22 Αρχική σελίδα HTML του διακομιστή διαδικτύου (web server) του συστήματος



Εκτός από το πρόγραμμα `index.html`, με την διαδικασία της μετάφρασης που έχει ήδη αναφερθεί, δημιουργούνται και τα αρχεία `webport.c` και `Db_shift.c`. Στο αρχείο `webport.c` τέθηκε η διεύθυνση IP του διακομιστή διαδικτύου στη μεταβλητή `MYIP`, δηλαδή `192.168.192.53`. Τέλος ένα σημαντικό αρχείο της εφαρμογής είναι το πρόγραμμα `cgi.c` στο οποίο γίνονται όλες οι διεργασίες ανάγνωσης και μεταφοράς των παραμέτρων ποιότητας του νερού.

4.2.1 Το πρόγραμμα της αρχικής σελίδας `index.html`

Όπως ήδη αναφέρθηκε για να μπορεί να γίνει σωστά η μετάφραση των αρχείων ώστε να προγραμματιστεί ο μικροελεγκτής και να υπάρχει μια σελίδα που φαίνεται από έναν περιηγητή, χρειάζεται και ένα αρχείο, σε γλώσσα προγραμματισμού `html`. Το συγκεκριμένο αρχείο ονομάζεται `index.html`. Αυτό το αρχείο αποτελεί την αρχική σελίδα του διακομιστή διαδικτύου. Επειδή δεν περιέχει καμία επιπλέον λειτουργία που να αφορά τον μικροελεγκτή, δηλαδή διάβασμα κάποιας εισόδου, αλλαγή κάποιας εξόδου, δεν χρειάζεται να περιέχει κώδικα σε γλώσσα `C`.

Στην αρχή του προγράμματος ορίζεται ο τίτλος ως «Μέτρηση παραμέτρων νερού στον Άραχθο». Στη συνέχεια γίνεται ο ορισμός διάφορων `στυλ` που θα χρησιμοποιηθούν στη σελίδα. Μετά ορίζεται το χρώμα του φόντου της σελίδας. Ορίζεται το μέγεθος, η θέση και το αρχείο της φωτογραφίας. Συνεχίζοντας ορίζεται η θέση και το `στυλ` των τίτλων: «Εργαστήριο Φυσικής Υψηλών Ενεργειών», «Μέτρηση παραμέτρων για την μόλυνση του νερού στον ποταμό Άραχθο». Έπειτα ορίζεται ο υπερσύνδεσμος που οδηγεί στη σελίδα των τελευταίων μετρήσεων και λέγεται «Τελευταίες Μετρήσεις». Ενεργοποιώντας τη συγκεκριμένη σελίδα δίνονται οι κατάλληλες εντολές στην πλακέτα του χρονομέτρου πραγματικού χρόνου και διασύνδεσης με τους αισθητήρες έτσι ώστε να εμφανιστούν στην ιστοσελίδα οι τελευταίες μετρήσεις. Ο τρόπος που δίνονται οι συγκεκριμένες εντολές αναλύονται στην παράγραφο που αναφέρεται στο αρχείο `cgi.c`. Συνεχίζοντας ορίζεται το `στυλ` και



η θέση για τους υπερσυνδέσμους των ηλεκτρονικών ταχυδρομείων. Ορίζεται η θέση και το στυλ για τον τίτλο «υπεύθυνος καθηγητής Νικόλαος Μάνθος» και τέλος ορίζεται κάποιες υπερσυνδέσεις που δεν χρειάζονται στο συγκεκριμένο σύστημα, αλλά μπορούν να χρησιμοποιηθούν σε μελλοντική χρήση.

Στη συνέχεια παρατίθεται το πρόγραμμα HTML index.html

```

<html>
<head>
<title>Μέτρηση παραμέτρων νερού στον Άραχθο</title> // ορισμός τίτλου της σελίδας

<meta http-equiv="Content-Type" content="text/html; charset=windows-1253">
<style type="text/css">
<!--
.style2 {font-size: 18px} // ορισμός του μεγέθους της γραμματοσειράς για το style2
.style4 {font-size: 1px} // ορισμός του μεγέθους της γραμματοσειράς για το style4
body {
}
.style5 {font-size: 18px; font-weight: bold; } // ορισμός του μεγέθους και του πάχους της
// γραμματοσειράς για το style5
.style7 {color: #0033FF} // ορισμός του χρώματος για το style7
.style8 {font-size: 16px} // ορισμός του μεγέθους της γραμματοσειράς για το style8
-->
</style>
</head>
<body bgcolor="#C7D0D9"> // ορισμός του χρώματος για το body

<h1 align="center"><span class="style4">κ</span></h1> // ορισμός του μεγέθους της
// φωτογραφίας, ορισμός του
// στυλ (style 4)

<h1 align="center"><span class="style7">Ε</span>ργαστήριο <span // ορισμός της θέσης και του στυλ
class="style7">Φ</span>υσικής <span class="style7" // για τον τίτλο Εργαστήριο Φυσικής
>Υ</span>ψηλών <span class="style7">Ε</span>νεργειών</h1> // Υψηλών Ενεργειών

<h2 align="center" class="style4">&nbsp;</h2> // ορισμός του στυλ (style4) για τον τίτλο
<h2 align="center">Μέτρηση παραμέτρων για την // Μέτρηση παραμέτρων για την μόλυνση

```




```

μόλυνση του νερού στον ποταμό Άραχθο </h2> // του νερού στον ποταμό Άραχθο

<p align="center"><span class="style2"> // ορισμός της υπερσύνδεσης
<a href="cgi-bin/csports.cgi" class="style5">Τελευταίες μετρήσεις </a></span></p> // Τελευταίες μετρήσεις,
<h2 align="center" class="style4">&nbsp;</h2> // το στυλ της και η στοίχιση

<p align="center" class="style2">Το αυτοματοποιημένο σύστημα μέτρησης παραμέτρων για την μόλυνση του
νερού στον ποταμό Άραχθο έγινε στα πλαίσια της διπλωματικής εργασίας για την απόκτηση μεταπτυχιακού
τίτλου σπουδών στις Σύγχρονες Ηλεκτρονικές Τεχνολογίες του τμήματος Φύσικης του Πανεπιστημίου
Ιωαννίνων από τους μεταπτυχιακούς φοιτητές: </p>

<p align="center" class="style2"><strong>Παρλάντζας Αθανάσιος </strong> // ορισμός στοίχισης και στυλ
// για τον τίτλο Παρλάντζας
// Αθανάσιος

<a href="mailto:sakis23@gmail.com" class="style8">sakis23@gmail.com</a></p> // ορισμός υπερσύνδεσης
// για το email

<p align="center" class="style2"><strong>Πέτρογλου Θεόδωρος</strong> // ορισμός στοίχισης και στυλ για
// τον τίτλο Πέτρογλου Θεόδωρος

<a href="mailto:teo54g@gmail.com" class="style8">teo54g@gmail.com</a></p> // ορισμός υπερσύνδεσης
// για το email

<p align="center" class="style2"><strong>Λούκας Νικήτας</strong> // ορισμός στοίχισης και στυλ
// για τον τίτλο Παρλάντζας
// Αθανάσιος

<a href="mailto:nikitios@yahoo.gr" class="style8">nikitio@yahoo.gr</a> </p> // ορισμός υπερσύνδεσης
// για το email

<p align="center" class="style2"> // ορισμός της υπερσύνδεσης
<a href="cgi-bin/ccports.cgi" class="style2">Πληροφορίες</a></p> // πληροφορίες

<p align="center" class="style2">υπεύθυνος καθηγητής</p> // ορισμός στοίχισης και στυλ
<p align="center" class="style5">Νικόλαος Μάνθος </p> // για τον τίτλο
<p align="center" class="style4">&nbsp;</p> // υπεύθυνος καθηγητής
<p align="center" class="style2">&nbsp;</p> // Νικόλαος Μάνθος
<p>&nbsp;</p>

</body>
</html>

```

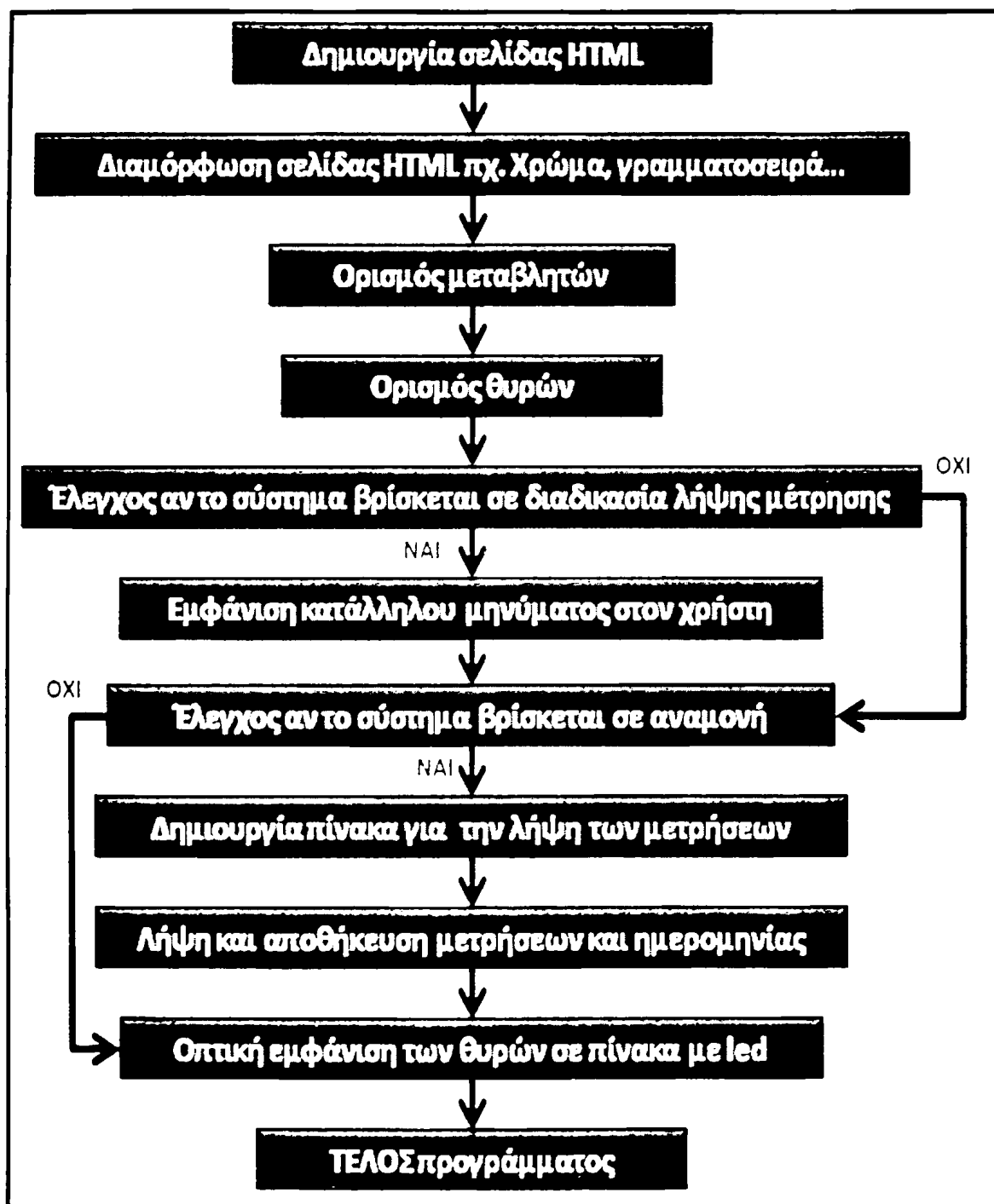
4.2.2 Η συνάρτηση CpuPortStatus του προγράμματος cgi.c



Στο αρχείο `cgi.c` που είναι γραμμένο σε γλώσσα προγραμματισμού C, έγιναν οι απαραίτητες αλλαγές έτσι ώστε να λειτουργεί η συγκεκριμένη εφαρμογή. Συγκεκριμένα έγιναν αλλαγές στο κομμάτι του προγράμματος που έχει σχέση με την ιστοσελίδα «Τελευταίες Μετρήσεις» η οποία δημιουργείται στην συνάρτηση `CrpPortStatus`. Δίπλα από τις εντολές της συνάρτησης σε κάθε κομμάτι ξεχωριστά, υπάρχουν διευκρινήσεις σχετικά με την λειτουργία της συνάρτησης.

Αρχικά ορίζονται το χρώμα, το στυλ και το μέγεθος της γραμματοσειράς. Έπειτα ορίζεται ο τίτλος της ιστοσελίδας και το χρώμα του φόντου. Στη συνέχεια ορίζεται ο πίνακας στον οποίο θα απεικονίζονται οι τελευταίες αποθηκευμένες μετρήσεις. Μετά ορίζονται οι ακέραιες μεταβλητές και οι μεταβλητές τύπου `float`. Ορίζονται οι πόρτες ως είσοδοι και έξοδοι και δίνονται οι αρχικές τιμές. Έπειτα ελέγχει αν το σύστημα βρίσκεται σε διαδικασία μέτρησης. Αν ναι, εμφανίζει στην οθόνη το κατάλληλο μήνυμα το οποίο προτρέπει τον χρήστη να δοκιμάσει αργότερα.





Εικόνα 4.23 Διάγραμμα ροής της συνάρτησης CruPortStatus του αρχείου cgi.c

Αν όχι, δημιουργεί την πρώτη σειρά του πίνακα, δηλαδή τις στήλες με το έ μήνα, ημέρα, ώρα και αισθητήρε, παίρνει από την πλακέτα χρονομέτρου πραγματ χρόνου και διασύνδεσης με τους αισθητήρες, με τον τρόπο που αναλύεται εξηγήσεις του προγράμματος, αρχικά την ημερομηνία, μετά την ώρα και μετά τις τ από τους πέντε αισθητήρες. Επαναλαμβάνεται η διαδικασία αυτή μέχρι

εμφανιστούν στην οθόνη όλες οι αποθηκευμένες μετρήσεις. Τέλος δημιουργεί και την τελευταία σειρά του πίνακα, όμοια με την πρώτη. Στην Εικόνα 4.23 φαίνεται το διάγραμμα ροής της CpuPortStatus. Στη συνέχεια παρατίθεται ο κώδικας της συνάρτησης CpuPortStatus.

```
int CpuPortStatus(FILE *stream, REQUEST *req)
{
    static prog_char head[] = "<html>"
        "<head>"

    "<style type=\"text/css\">"           //
    "<!--"                               // ορίζονται τα στυλ
    ".style2 {color: #CCCCCC;}"         // το χρώμα
    "font-size: 10px;}"                // και το μέγεθος της
    "-->"                               // γραμματοσειράς
    "</style>"                             //

    "<meta http-equiv=\"refresh\" content=\"200; URL= \" PORT_STATUS_CGI \">"
    "<title>Μέτρηση παραμέτρων νερού στον ποταμό Άραχθο</title>" // ορίζεται ο τίτλος της
                                                                    // ιστοσελίδας
    "</head>"

    "<body bgcolor=\"#C7D0D9\"><a href= \" / \">" // ορίζεται το χρώμα του φόντου

    "</a><div align=\"center\">"           // ορίζεται η στοίχιση

    "<table border=\"1\" cellspacing=\"0\">\r\n" // του πίνακα, το κενό

    "<thead><tr><th rowspan=\"2\"> PORT </th>" // ανάμεσα στα κελιά

    "<th colspan=\"8\">Bit</th></tr>"      //
    "<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th>" //
    "<th>2</th><th>1</th><th>0</th></tr>"    //

```



```

"</thead><tfoot>\r\n";           //

NutHttpSendHeaderTop(stream, req, 200, "Ok");
NutHttpSendHeaderBot(stream, "text/html", -1);

int BBB,EEE,DDD,i,k,TEMP1,TEMP2,TEMP3;    // ορίζεται οι
double PH,TEMP,COND,DIS,ORP,A[5],B[5];    // μεταβλητές

char VARS[5][10];

A[0] = 0.13;           // ορίζονται οι μεταβλητές του
B[0] = -76.8;         // 1ο αισθητήρα
A[1] = 0.018;         // ορίζονται οι μεταβλητές του
B[1] = -3.7;          // 2ο αισθητήρα
A[2] = 12.752;        // ορίζονται οι μεταβλητές του
B[2] = -2614.3;       // 3ο αισθητήρα
A[3] = 0.128;         // ορίζονται οι μεταβλητές του
B[3] = -26.2;         // 4ο αισθητήρα
A[4] = 1.301;         // ορίζονται οι μεταβλητές του
B[4] = -765.8;        // 5ο αισθητήρα

DDRA=0;               // ορίζεται η PORTA ως είσοδος
DDRB=0;               // ορίζεται η PORTB ως είσοδος
DDRC=0;               // ορίζεται η PORTC ως είσοδος
DDRD=255;             // ορίζεται η PORTD ως έξοδος
DDRE=0;               // ορίζεται η PORTE ως είσοδος
BBB=PINB;              // ορίζεται η PORTB ως BBB
EEE=PINE;              // ορίζεται η τιμή 0 στην μεταβλητή DDD
DDD=0;                 // ορίζεται η PORTE ως EEE
PORTD=DDD;             // παίρνει η πόρτα D την τιμή της μεταβλητής DDD

if (BBB==85) fputs("<body>Το σύστημα βρίσκεται σε διαδικασία μέτρησης. Αργότερα μπορεί
να έχετε περισσότερη τύχη...</body>", stream); // εάν η τιμή στην πόρτα B (που

```



```

// συνδέεται με την πλακέτα
// χρονομέτρου πραγματικού χρόνου
// και διασύνδεσης με τους
// αισθητήρες) πάρει την τιμή 85,
// δηλαδή 0101 0101 σε δυαδικό
// τότε εμφανίζεται στην οθόνη το
// μήνυμα που προτρέπει τον
// χρήστη να δοκιμάσει αργότερα,
// επειδή το σύστημα βρίσκεται σε
// διαδικασία μέτρησης

if (BBB==0) { // εάν η τιμή της πόρτας B είναι 0, τότε το σύστημα βρίσκεται
// σε αναμονή, άρα μπορεί να συνεχιστεί η διαδικασία της
// λήψης των τελευταίων αποθηκευμένων μετρήσεων

```

```

if (EEE<128) {
DDD=1; // δίνεται η τιμή 0000 0001 στην πόρτα D δηλαδή δίνεται
PORTD=DDD; // εντολή στην πλακέτα χρονομέτρου πραγματικού χρόνου
// και διασύνδεσης με τους αισθητήρες να στείλει τις
// τελευταίες αποθηκευμένες μετρήσεις

```

```

fputs("<p align=\"center\" class=\"style2\">Πίνακας των πιο πρόσφατων
αποθηκευμένων μετρήσεων ποιότητας νερού στον ποταμό Άραχθο </p>", stream);

```

```

fputs("<table width=\"950\" border=\"1\">", stream); //
fputs("<tr>", stream); //
fputs("<td>", stream); //
fprintf(stream, "ΕΤΟΣ"); //
fputs("</td>", stream); //
fputs("<td>", stream); //
fprintf(stream, "ΜΗΝΑΣ"); //

```



```

fputs("</td>", stream); //
fputs("<td>", stream); //
fprintf(stream, "ΗΜΕΡΑ"); // δημιουργείται η πρώτη
fputs("</td>", stream); // σειρά του πίνακα στον
fputs("<td>", stream); // οποίο θα εμφανιστούν
fprintf(stream, "ΩΡΑ"); // οι μετρήσεις των
fputs("</td>", stream); // αισθητήρων

fputs("<td>", stream); //
fprintf(stream, "ΘΕΡΜΟΚΡΑΣΙΑ Cel"); //
fputs("</td>", stream); //
fputs("<td>", stream); //
fprintf(stream, "PH"); //
fputs("</td>", stream); //
fputs("<td>", stream); //
fprintf(stream, "ΑΓΩΓΙΜΟΤΗΤΑ"); //
fputs("</td>", stream); //
fputs("<td>", stream); //
fprintf(stream, "ΟΞΥΓΟΝΟ"); //
fputs("</td>", stream); //
fputs("<td>", stream); //
fprintf(stream, "ORP/Redox"); //
fputs("</td>", stream); //
fputs("</tr>", stream); //

for (i=0;i<200;i++){
    fputs("<tr>", stream);

    for (k=0;k<4;k++){
        do {
            EEE=PINE; // περιμένει μέχρι να επικοινωνήσει
            }while (EEE<128); // με την πλακέτα του χρονομέτρου
            EEE=PINE; // πραγματικού χρόνου και
            fputs("<td>", stream); // διασύνδεσης με τους αισθητήρες,
            fprintf(stream, "%d",PINB); // δηλαδή περιμένει μέχρι να γίνει ο
            fputs("</td>", stream); // ακροδέκτης 7της πόρτας D

```



```

DDD=2; // λογικό μηδέν, δηλαδή μέχρι να
PORTD=DDD; // έρθει στην πόρτα D ο δυαδικός
do { // αριθμός 1XXX XXXX. Έπειτα
    EEE=PINE; // παίρνει την τιμή της πόρτας E
    }while (EEE>127); // και κάνει την πόρτα D 0
DDD=0; // αυτό επαναλαμβάνεται για όλα
PORTD=DDD; // τα στοιχεία της ημερομηνίας του πίνακα
} // των μετρήσεων

```

```

for (k=0;k<5;k++){
    do {
        EEE=PINE; // περιμένει μέχρι να επικοινωνήσει
    }while (EEE<128); // με την πλακέτα του χρονομέτρου
    EEE=PINE; // πραγματικού χρόνου και
    TEMP = A[k]*PINB*4 + B[k]; // διασύνδεσης με τους αισθητήρες,
    TEMP1 = TEMP; // δηλαδή περιμένει μέχρι να γίνει ο
    TEMP2 = TEMP*10.0; // ακροδέκτης 7της πόρτας D
    TEMP3 = TEMP2 - 10*TEMP1; // λογικό μηδέν, δηλαδή μέχρι να
    if (TEMP1<0){ // έρθει στην πόρτα D ο δυαδικός
        TEMP1 = 0; // αριθμός 1XXX XXXX. Έπειτα
        TEMP3 = 0; // παίρνει την τιμή της πόρτας E
    } // (μήνας) και κάνει την πόρτα D 0.
    if (TEMP3<0) // αυτό επαναλαμβάνεται για όλα
        TEMP3 = 0; // τα στοιχεία του πίνακα
    fputs("<td>", stream); // των μετρήσεων
    fprintf(stream, "%d",TEMP1);//
    fprintf(stream, "."); //
    fprintf(stream, "%d",TEMP3);//
    DDD=2; //
    PORTD=DDD; //
    do { //
        EEE=PINE; //
    }while (EEE>127); //
    DDD=0; //
    PORTD=DDD; //
}

```




```

    }

    fputs("</tr>", stream);           //

fputs("<tr>", stream);

    fputs("<td>", stream);           //
    fprintf(stream, "ΕΤΟΣ");         //
    fputs("</td>", stream);         //
    fputs("<td>", stream);           //
    fprintf(stream, "ΜΗΝΑΣ");        //
    fputs("</td>", stream);         //
    fputs("<td>", stream);           //
    fprintf(stream, "ΗΜΕΡΑ");        //
    fputs("</td>", stream);         //
    fputs("<td>", stream);           // δημιουργείται η τελευταία
    fprintf(stream, "ΩΡΑ");           // σειρά του πίνακα στον
    fputs("</td>", stream);         // οποίο εμφανίζονται
    fputs("<td>", stream);           // οι μετρήσεις των
    fprintf(stream, "ΘΕΡΜΟΚΡΑΣΙΑ Cel"); // αισθητήρων
    fputs("</td>", stream);         //
    fputs("<td>", stream);           //
    fprintf(stream, "PH");            //
    fputs("</td>", stream);         //
    fputs("<td>", stream);           //
    fprintf(stream, "ΑΓΩΓΙΜΟΤΗΤΑ");   //
    fputs("</td>", stream);         //
    fputs("<td>", stream);           //
    fprintf(stream, "ΟΞΥΓΟΝΟ");       //
    fputs("</td>", stream);         //
    fputs("<td>", stream);           //
    fprintf(stream, "ORP/Redox");     //
    fputs("</td>", stream);         //
    fputs("</tr>", stream);         //

```



```

        fputs("</table>", stream);           //
    }                                         //
}

fputs_P(head, stream);                      //
                                           //
fputs("<tr><th>B</th>", stream);           //
HtmlLedRow(stream, 8, 1, inp(PINB), inp(PORTB), inp(DDRB)); //
fputs("</tr>\r\n", stream);               //
                                           //
fputs("<tr><th>D</th>", stream);           //
HtmlLedRow(stream, 8, 1, inp(PIND), inp(PORTD), inp(DDRD)); //
fputs("</tr>\r\n", stream);               // δημιουργείται ένας
                                           // πίνακας με led
fputs("<tr><th>E</th>", stream);           //
HtmlLedRow(stream, 8, 1, inp(PINE), inp(PORTE), inp(DDRE)); //
fputs("</tr>\r\n", stream);               //
                                           //
fputs("<tr><th>F</th>", stream);           //
HtmlLedRow(stream, 8, 1, inp(PINF), inp(PINF), 0);           //
fputs("</tr>\r\n", stream);               //
                                           //
fputs("</tfoot></table><br></div></body>\r\n</html>", stream); //
fflush(stream);                             //

return 0;
}

```

4.3 Το πρόγραμμα του χρονομέτρου πραγματικού χρόνου (real time clock)



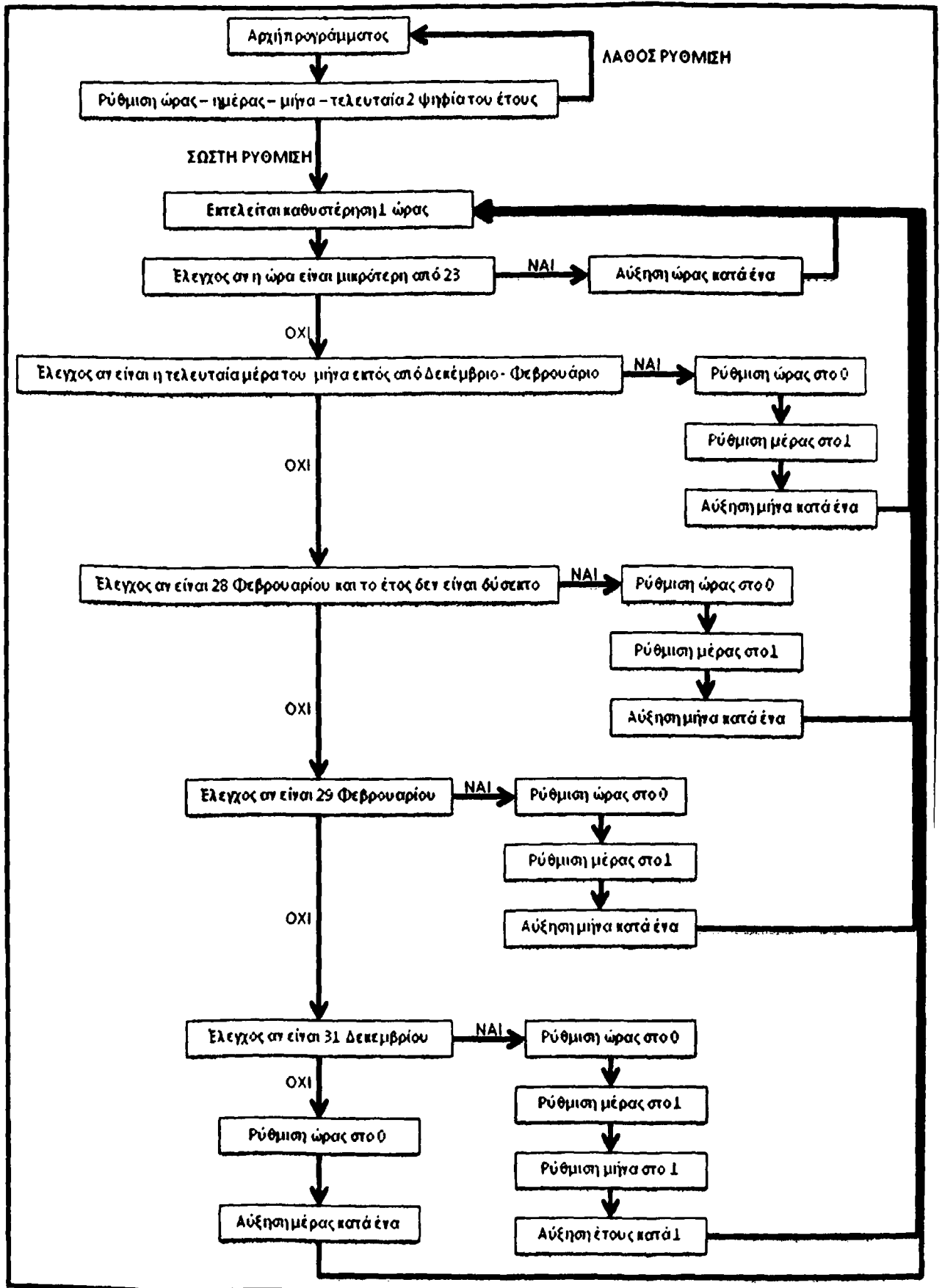
Το πρόγραμμα αυτό είναι φορτωμένο στον μικροελεγκτή ATMEGA128 της εταιρίας Atmel της πλακέτας χρονομέτρου πραγματικού χρόνου - διασύνδεσης με τους αισθητήρες. Χρησιμοποιείται για να υπολογίζει και να ενημερώνει το υπόλοιπο σύστημα σχετικά με την ημερομηνία και ώρα. Είναι γραμμένο σε γλώσσα ANSI C[17], και έχει μεταφραστεί (compile) και φορτωθεί στον μικροελεγκτή με το AVR Studio 4, αφού προηγουμένως έχουν γίνει οι εγκαταστάσεις που έχουν αναφερθεί. Το διάγραμμα ροής του προγράμματος φαίνεται στην εικόνα 4.24.

Αρχικά ορίζονται ποιες βιβλιοθήκες συναρτήσεων θα χρησιμοποιηθούν. Εκτός από τις βασικές βιβλιοθήκες, χρειάζεται και η βιβλιοθήκη των συναρτήσεων της χρονοκαθυστέρησης για να μπορεί να δημιουργηθεί η καθυστέρηση μίας ώρας.

Ορίζονται οι είσοδοι και οι έξοδοι. Ορίζονται οι εντολές setbit και clrbit έτσι ώστε να γίνονται λογικό ένα και λογικό μηδέν οι ακροδέκτες των πορτών. Δηλώνονται οι μεταβλητές τύπου ακεραίου που θα χρειαστούν. Ορίζεται η πραγματική ώρα, η μέρα του μήνα, ο μήνας και τα τελευταία δύο ψηφία του έτους.

Στη συνέχεια ορίζεται το κομμάτι του προγράμματος που θα επαναλαμβάνεται με τις κατάλληλες αυξήσεις της ώρας, της μέρας, μήνα και χρόνου.





Εικόνα 4.24 Διάγραμμα ροής του προγράμματος χρονομέτρου πραγματικού χρόνου σε γλώσσα προγραμματισμού C



Παρακάτω φαίνεται το πρόγραμμα σε γλώσσα C.

```
#include <avr/io.h>           // περιλαμβάνει την βασική
                               // βιβλιοθήκη io.h
#include <util/delay.h>       // περιλαμβάνει την βιβλιοθήκη
                               // χρονοκαθυστέρησης delay.h
#include <stdio.h>            // περιλαμβάνει την βασική
                               // βιβλιοθήκη stdio.h

int main(void){               // απαραίτητη αρχική εντολή

-                               // αρχικοποίηση των πορτών
DDRA = 0x1F;                  // I/O: 0 0 0 1 1 1 1 1 (έξοδος, ώρα ημέρας)

DDRB = 0x1F;                  // I/O: 0 0 0 1 1 1 1 1 (έξοδος, ημέρα του μήνα)

DDRC = 0x0F;                  // I/O: 0 0 0 0 1 1 1 1 (έξοδος, μήνας)

DDRD = 0x1F;                  // I/O: 0 0 0 1 1 1 1 1 (έξοδος, τα
                               // τελευταία δύο ψηφία του έτους)

DDRE = 0x00;                  // I/O: 0 0 0 0 0 0 0 0 (είσοδος, γίνεται η αρχική
                               // ρύθμιση της ημερομηνία και ώρας

PORTA = 0x00;                 //
PORTB = 0x00;                 // μηδενίζονται οι έξοδοι
PORTC = 0x00;                 //
PORTD = 0x00;                 //
```



```

#define SETBIT(ADDRESS,BIT) (ADDRESS |= (1<<BIT)); // παραμετροποίηση
#define CLRBIT(ADDRESS,BIT) (ADDRESS &= ~(1<<BIT)); // για setting και
#define TOGBIT(ADDRESS,BIT) (ADDRESS ^= (1<<BIT)); // clearing bits

```

```

int i, time, day, month, year; // δήλωση μεταβλητών

```

```

time = 0; //
day = 0; // αρχικοποίηση μεταβλητών
month = 0; //
year = 0; //

```

```

arxika:

```

```

if (PINE >= 32){ // εάν έστω ένα από τα bit 5,6,7 είναι 1, τότε
    PORTA = 0xFF; // γίνονται όλοι οι έξοδοι λογικό 1
    PORTB = 0xFF; // έτσι ώστε να καταλάβει ο χρήστης
    PORTC = 0xFF; // ότι η ώρα και η ημερομηνία δεν
    PORTD = 0xFF; // έχουν ρυθμιστεί σωστά
    goto arxika;
}

```

```

PORTA = 0x00;
PORTB = 0x00;
PORTC = 0x00;
PORTD = 0x00;

```

```

if (PINE < 32) { //
    while ((PINE & (1<<PINE5)) == 0); // ορίζεται από τον χρήστη
    while ((PINE & (1<<PINE6)) == 0); // η ώρα
    time = (PINE & 31); //
    PORTA = time; //
}

```



```

while (PINE >= 32); //

while ((PINE & (1<<PINE5)) == 0); //
while ((PINE & (1<<PINE6)) == 0); // ορίζεται από τον
day = (PINE & 31); // χρήση η μέρα του
PORTB = day; // μήνα
while (PINE >= 32); //

while ((PINE & (1<<PINE5)) == 0); //
while ((PINE & (1<<PINE6)) == 0); // ορίζεται από τον
month = (PINE & 15); // χρήση ο μήνας
PORTC = month; //
- while (PINE >= 32); //

while ((PINE & (1<<PINE5)) == 0); //
while ((PINE & (1<<PINE6)) == 0); // ορίζεται από τον
year = (PINE & 31); // χρήση τα τελευταία
PORTD = year; // 2 ψηφία του έτους
while (PINE >= 32); //
}

repeat:
for (i=0;i<3604;i++) { // ορίζεται καθυστέρηση
_delay_ms(961); // μίας ώρας λαμβάνοντας υπόψιν και την
// επιπλέον καθυστέρηση που χρειάζεται για
// να εκτελεστούν οι εντολές σε συνδιασμό
// με την εντολή delay
}

if (time < 23){ // εάν η ώρα είναι μικρότερη από 23
// τότε η ώρα αυξάνεται κατά 1
time = time + 1; // και δεν επηρεάζονται μέρα,
PORTA = time; // μήνας και έτος. Μετά πηγαίνει να

```



```

// εκτελέσει καθυστέρηση μίας ώρας
//
goto repeat;
}
if (time == 23){ //ελέγχει αν η ώρα είναι 23

    if (((day == 31) && (month<12)) || // ελέγχει αν είναι η
        ((day == 30) && ((month == 4) || // τελευταία μέρα ενός από
            (month == 6) || (month == 9) || // τους μήνες που έχουν 31
            (month == 11)))){ // μέρες ή 30, εκτός του
        time=0; // Δεκεμβρίου. Αν ισχύει τότε
        PORTA=time; // η ώρα γίνεται 0 και
        day=1; // ανεβαίνει ο μήνας κατά ένα.
        PORTB=day; // Μετά πηγαίνει να εκτελέσει
        month=month + 1; // καθυστέρηση μίας ώρας.
        PORTC=month; //
        goto repeat; //
    }

    if ((day == 28) && (month == 2) && // ελέγχει αν δεν είναι
        (year != 12) && (year != 16) && (year != 20) // δύσεκτο έτος και είναι
        && (year != 24) && (year != 28)){ // η τελευταία μέρα του
        time=0; // Φεβρουαρίου. Αν
        PORTA=time; // ισχύει τότε η ώρα
        day=1; // γίνεται μηδέν και ο
        PORTB=day; // μήνας ανεβαίνει κατά
        month=month + 1; // ένα και πηγαίνει να
        PORTC=month; // εκτελέσει καθυστέρηση
        goto repeat; // μίας ώρας
    }

    if ((day == 29) && (month == 2)){ // ελέγχει αν είναι 29
        time=0; // Φεβρουαρίου.
        PORTA=time; // Αν ισχύει η ώρα γίνεται

```




```

day=1; // μηδέν και ο μήνας ανεβαίνει
PORTB=day; // κατά ένα. Πηγαίνει να
month=month + 1; // εκτελέσει καθυστέρηση
PORTC=month; // μίας ώρας
goto repeat; //
}

if ((day == 31) && (month == 12)){ // ελέγχει αν είναι 31
time=0; // Δεκεμβρίου. Αν
PORTA=time; // ισχύει τότε η ώρα
day=1; // γίνεται 0, η μέρα
PORTB=day; // γίνεται 1, ο μήνας
month=1; // γίνεται 1, το έτος
PORTC=month; // ανεβαίνει κατά ένα
year=year + 1; // και πηγαίνει να
PORTD=year; // εκτελέσει καθυστέρηση
goto repeat; // μίας ώρας
} //

time=0; // αν δεν ισχύει τίποτα από τα
PORTA=time; // παραπάνω, η ώρα γίνεται 0,
day=day + 1; // η μέρα ανεβαίνει κατά 1 και
PORTB=day; // πηγαίνει να εκτελέσει
goto repeat; // καθυστέρηση μίας ώρας
}

return (0); // τελευταία εντολή προγράμματος
}

```



4.4 Το πρόγραμμα ` της πλακέτας διασύνδεσης του διακομιστή διαδικτύου με την πλακέτα χρονομέτρου πραγματικού χρόνου και την πλακέτα ανάγνωσης αισθητήρων

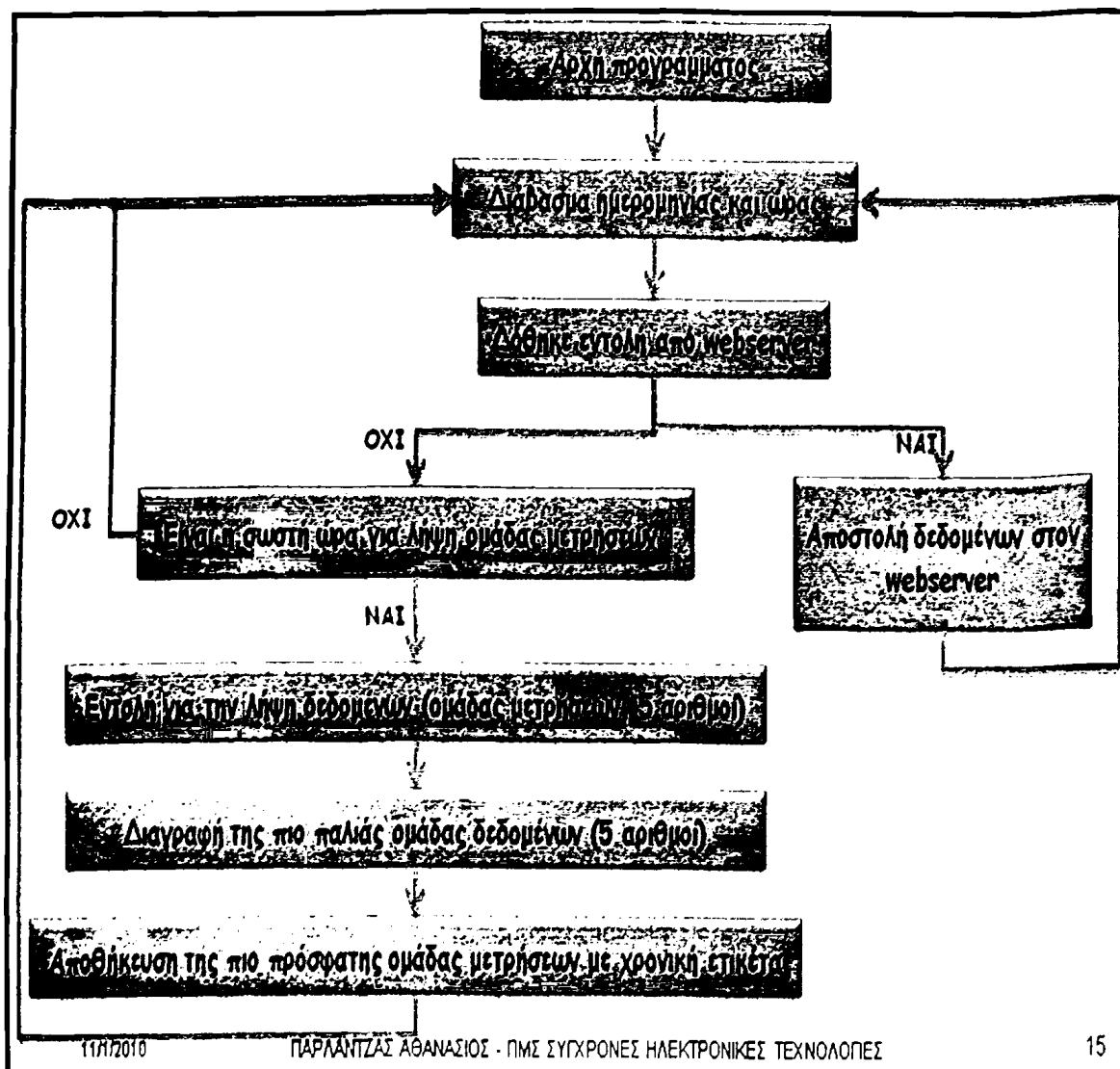
Το πρόγραμμα αυτό είναι φορτωμένο στο δεύτερο μικροελεγκτή ATMEGA128 της εταιρίας Atmel της πλακέτας πραγματικού χρόνου - διασύνδεσης του διακομιστή διαδικτύου με το σύστημα ανάγνωσης αισθητήρων. Είναι γραμμένο σε γλώσσα C, και έχει μεταφραστεί (compile) και φορτωθεί στον μικροελεγκτή από το AVR Studio 4 όπως και το πρόγραμμα που έχει αναφερθεί σε προηγούμενη παράγραφο.

Χρησιμοποιείται για να αποφασίζει αν πρέπει να ξεκινήσει την διαδικασία της μέτρησης, για να αποθηκεύει την τελευταία σειρά μετρήσεων από τους αισθητήρες και να στέλνει στο διακομιστή διαδικτύου και κατ' επέκταση στον απομακρυσμένο υπολογιστή, τις τελευταίες σειρές αποθηκευμένων μετρήσεων. Στην εικόνα 4.25 φαίνεται το διάγραμμα ροής του προγράμματος.

Αρχικά ορίζονται οι εισοδοί και οι έξοδοι που χρειάζονται. Δίνονται οι αρχικές τιμές στις εξόδους και δηλώνονται οι μεταβλητές που θα χρησιμοποιηθούν. Οι μεταβλητές i , k , m είναι βοηθητικές. Οι μεταβλητές $time1$ έως $time6$ αντιστοιχούν στις ώρες που θα πρέπει να γίνουν οι μετρήσεις από τους αισθητήρες. Η μεταβλητή $real_time$ αντιστοιχεί στην πραγματική ώρα που διαβάζει από την πλακέτα χρονομέτρου πραγματικού χρόνου και διασύνδεσης με τους αισθητήρες. Αντίστοιχα $date$ είναι ημέρα του μήνα, $month$ ο μήνας και $year$ τα τελευταία δύο ψηφία του έτους. Στον πίνακα δύο διαστάσεων mem αποθηκεύονται οι μετρήσεις από τους αισθητήρες και η ημερομηνία και ώρα που μετρήθηκαν.

Στη συνέχεια ορίζονται ποιες ώρες κατά τη διάρκεια εικοσιτετραώρου παίρνονται μετρήσεις από τους αισθητήρες. Στο συγκεκριμένο σύστημα απαιτούνται τέσσερις φορές την ημέρα, γι' αυτό στις μεταβλητές που αντιστοιχούν στην πέμπτη και έκτη φορά δίνεται τιμή μεγαλύτερη από εικοσιτέσσερα. Έπειτα ακολουθεί το κυρίως πρόγραμμα το οποίο είναι μέσα σε βρόγχο επανάληψης.





Εικόνα 4.25 Διάγραμμα ροής του προγράμματος διασύνδεσης με τους αισθητήρες

Αποθηκεύονται η ημερομηνία και ώρα στους καταχωρητές που έχει αναφερθεί. Έπειτα ελέγχει αν δόθηκε εντολή από τον διακομιστή διαδικτύου να στείλει τις τελευταίες αποθηκευμένες μετρήσεις. Αν ναι, τις στέλνει. Στη συνέχεια ελέγχει αν πρέπει να δοθεί εντολή στην πλακέτα που συνδέεται με τους αισθητήρες, να πάρει μετρήσεις από τους αισθητήρες. Αν έχει δοθεί, βγάζει στην έξοδο που συνδέεται με την πλακέτα του διακομιστή διαδικτύου την δεκαδική τιμή 85 (δυαδική μορφή 0101 0101) δηλώνοντας με αυτό τον τρόπο στο διακομιστή διαδικτύου ότι βρίσκεται σε διαδικασία μέτρησης. Αλλάζει θέσεις στις αποθηκευμένες τελευταίες μετρήσεις έτσι ώστε να χαθεί η παλαιότερη χρονολογικά και να ελευθερωθεί μία θέση μνήμης. Αποθηκεύεται η ημερομηνία και η ώρα και δίνεται εντολή στη πλακέτα που συνδέεται με τους αισθητήρες να ξεκινήσει η διαδικασία της μέτρησης από τους αισθητήρες.

Περιμένει να γίνει επιβεβαίωση ότι τελείωσαν οι μετρήσεις και μετά αποθηκεύει τις μετρήσεις από τους αισθητήρες στην πιο πρόσφατη θέση μνήμης, στην οποία έχουν ήδη αποθηκευθεί η ημερομηνία και ώρα. Έπειτα ενημερώνει την πλακέτα του διακομιστή διαδικτύου ότι τελείωσε τις μετρήσεις και βρίσκεται σε διαδικασία αναμονής.

Στη συνέχεια παρατίθεται ο κώδικας του προγράμματος σε γλώσσα προγραμματισμού C.

```
#include <avr/io.h>           // περιλαμβάνει την βασική βιβλιοθήκη io.h

#include <util/delay.h>       // περιλαμβάνει την βιβλιοθήκη
                              // χρονοκαθυστέρησης delay.h

#include <stdio.h>           // περιλαμβάνει την βασική βιβλιοθήκη stdio.h

int main(void){              // απαραίτητη αρχική εντολή

                              // αρχικοποίηση των πορτών
DDRA = 0xFF;                // I/O: 1 1 1 1 1 1 1 1 (έξοδος προς τον
                              // διακομιστή διαδικτύου)

DDR B = 0x00;               // I/O: 0 0 0 0 0 0 0 0 (παίρνει είσοδο από
                              // την πλακέτα που συνδέεται με τους αισθητήρες

DDRC = 0xA0;                // I/O: 1 0 1 0 0 0 0 0 (pin 0-4: παίρνει
                              // είσοδο την ώρα από την πλακέτα χρονομέτρου
                              // πραγματικού χρόνου και διασύνδεσης με τους
                              // αισθητήρες
```



```

// pin 5: βγάζει έξοδο εντολής εκκίνησης στην
// πλακέτα που συνδέεται με τους αισθητήρες
// pin 6: παίρνει είσοδο από την πλακέτα που
// συνδέεται με τους αισθητήρες ότι είναι έτοιμη
// Pin 7: βγάζει έξοδο στην πλακέτα που συνδέεται
// με τους αισθητήρες ότι είναι έτοιμη

```

```

DDR0 = 0x40; // I/O: 0 1 0 0 0 0 0 0 (pin 0-4: παίρνει
// είσοδο την ημέρα του μήνα από την
// πλακέτα χρονομέτρου πραγματικού χρόνου - διασύνδεσης
// με τους αισθητήρες)
// pin 5: παίρνει εντολή από την πλακέτα
// του διακομιστή διαδικτύου για να
// δώσει τις αποθηκευμένες μετρήσεις
// pin 6: βγάζει έξοδο στην πλακέτα του
// διακομιστή διαδικτύου ότι είναι έτοιμη
// Pin 7: παίρνει είσοδο από την
// πλακέτα που συνδέεται με τους
// αισθητήρες ότι είναι έτοιμη

```

```

DDRE = 0x00; // I/O: 0 0 0 0 0 0 0 0 (pin 0-3: παίρνει είσοδο τον
// μήνα, από την πλακέτα χρονομέτρου πραγματικού
// χρόνου και διασύνδεσης με τους αισθητήρες
// pin 4-7: παίρνει είσοδο το έτος από την πλακέτα
// χρονομέτρου πραγματικού χρόνου και διασύνδεσης
// με τους αισθητήρες

```

```

#define SETBIT(ADDRESS,BIT) (ADDRESS |= (1<<BIT)); // παραμετροποίηση
#define CLRBIT(ADDRESS,BIT) (ADDRESS &= ~(1<<BIT)); // για setting και
#define TOGBIT(ADDRESS,BIT) (ADDRESS ^= (1<<BIT)); // clearing bits

```



```

int mem[199][9]; // ορισμός του πίνακα
int i,k, m, n, temp, time1, time2, time3, time4, time5, // για την αποθήκευση
time6, real_time, stored_time, day, month, year, // των μετρήσεων και
AA,BB,CC,DD,EE; // ορισμός των
// υπόλοιπων μεταβλητών

CLRBIT(PORTD,6); // γίνεται μηδενισμός του Bit 6 της πόρτας D
CLRBIT(PORTC,5); // γίνεται μηδενισμός του Bit 5 της πόρτας D
CLRBIT(PORTC,7); // γίνεται μηδενισμός του Bit 7 της πόρτας D

PORTA = 0; // μηδενίζεται η πόρτα A

time1=0; // ρυθμίζεται η ώρα της πρώτης φοράς
// που θα παίρνει μέτρηση το σύστημα
time2=6; // ρυθμίζεται η ώρα της δεύτερης φοράς
// που θα παίρνει μέτρηση το σύστημα
time3=12; // ρυθμίζεται η ώρα της τρίτης φοράς
// που θα παίρνει μέτρηση το σύστημα
time4=18; // ρυθμίζεται η ώρα της τέταρτης φοράς
// που θα παίρνει μέτρηση το σύστημα

time5=100; // ρυθμίζεται η ώρα της πέμπτης φοράς που θα παίρνει
// μέτρηση το σύστημα. Επειδή στο συγκεκριμένο
// σύστημα δεν χρειάζεται η πέμπτη φορά, δίνεται
// στη μεταβλητή μια τιμή πάνω από εικοσιτέσσερα

time6=100; // ρυθμίζεται η ώρα της έκτης φοράς που θα παίρνει
// μέτρηση το σύστημα. Επειδή στο συγκεκριμένο
// σύστημα δεν χρειάζεται η έκτη φορά, δίνεται
// στη μεταβλητή μια τιμή πάνω από εικοσιτέσσερα

stored_time = 200; // ορίζεται η αρχική αποθηκευμένη ώρα που
// χρησιμοποιείται για να μην πάρει 2 φορές

```



```

// μέτρηση την ίδια ώρα

for (i=0;i<200;i++) {
    k=0;
    for (k=0;k<9;k++){
        mem[i][k] = 0;
    }
}

repeat:
CC=PINC;
real_time = (CC & 31);
-
// αποθηκεύεται η ώρα στη θέση
// real_time απορρίπτοντας τα bit
// της πόρτας που δεν χρειάζονται

DD = PIND;
day = (DD & 31);
// αποθηκεύεται η ημέρα του μήνα στη θέση date
// απορρίπτοντας τα bit της πόρτας που δεν χρειάζονται

EE = PINE;
month = (EE & 15);
// αποθηκεύεται ο μήνας στη θέση
// month απορρίπτοντας τα bit της
// πόρτας που δεν χρειάζονται

EE = PINE;
year= (EE & 240);
year = year>>4;
// αποθηκεύει την χρονολογία
// στη θέση year απορρίπτοντας
// τα bit που δεν χρειάζονται

if ((PIND & (1<<PIND5)) != 0){
// ελέγχει αν δόθηκε εντολή από την πλακέτα
// του διακομιστή διαδικτύου να στείλει τις
// τελευταίες αποθηκευμένες μετρήσεις

i=0;
for (i=0;i<200;i++) {
    k=0;
    for (k=0;k<9;k++){
// βρόγχος αποστολής των
// τελευταίων αποθηκευμένων
// μετρήσεων στην πλακέτα
// του διακομιστή διαδικτύου

```



```

PORTA = mem[i][k];
SETBIT(PORTD,6);           // ενημερώνει το διακομιστή δια-
                           // δικτύου ότι έστειλε την τιμή

while ((PIND & (1<<PIND5)) != 0);
while ((PIND & (1<<PIND7)) == 0); // περιμένει επιβεβαίωση
                                   // από τον διακομιστή
                                   // διαδικτύου ότι έλαβε
                                   // την τιμή

CLRBIT(PORTD,6);           // μηδενίζει το bit αποστολής

while ((PIND & (1<<PIND7)) != 0); // περιμένει να
                                   // μηδενιστεί το bit λήψης
    }
}
}

```

```

if (((real_time == time1) || (real_time == time2) || // ελέγχει αν είναι η σωστή
    (real_time == time3) || (real_time == time4) || // ώρα για να δοθεί η εντολή
    (real_time == time5) || (real_time == time6)) && // για να ξεκινήσει η
    (stored_time != real_time)){ // διαδικασία της μέτρησης

```

```

PORTA = 170;           // ενημερώνει το διακομιστή
                       // διαδικτύου ότι το σύστημα βρίσκεται
                       // σε διαδικασία μέτρησης

```

```

i = 0;
m = 201;
n = 200;

```

```

for (i=0;i<200;i++){ //
    m = m-1;         //
    n = n-1;         // αλλαγή των θέσεων των

```




```

    for (k=0;k<9;k++){           // αποθηκευμένων μετρήσεων
    temp = mem[n][k];           // για να αποθηκευθεί η
    mem[m][k] = temp;           // τελευταία μέτρηση
    }                             //
}                                 //

mem[0][0] = year;               // αποθήκευση έτους
mem[0][1] = month;              // αποθήκευση μήνα
mem[0][2] = day;                // αποθήκευση ημέρας
mem[0][3]= real_time;           // αποθήκευση ώρας

SETBIT(PORTC,5);                // 0010 0000: δίνει εντολή εκίνησης
// μετρήσεων στην πλακέτα που
// συνδέεται με τους αισθητήρες

while ((PINC & (1<<PINC6)) == 0); // περιμένει να τελειώσει η
// πλακέτα που συνδέεται με
// τους αισθητήρες τη
// διαδικασία της μέτρησης

CLRBIT(PORTC,5);

mem[0][4] = PINB;               // αποθηκεύει την τιμή του 1ου αισθητήρα

SETBIT(PORTC,7);                // 1000 0000: επιβεβαιώνει στην πλακέτα
// που συνδέεται με τους αισθητήρες ότι
// διάβασε την τιμή του 1ου αισθητήρα

while ((PINC & (1<<PINC6)) != 0);
CLRBIT(PORTC,7);

while ((PINC & (1<<PINC6)) == 0); // περιμένει την πλακέτα που
// συνδέεται με τους αισθητήρες να

```



```

// στείλει τη τιμή του 2ου αισθητήρα

mem[0][5] = PINB; // αποθηκεύει τη τιμή του 2ου αισθητήρα

SETBIT(PORTC,7); // 1000 0000: επιβεβαιώνει στην πλακέτα
// που συνδέεται με τους αισθητήρες ότι
// διάβασε την τιμή του 2ου αισθητήρα

while ((PINC & (1<<PINC6)) != 0);
CLRBIT(PORTC,7);

while ((PINC & (1<<PINC6)) == 0); // περιμένει την πλακέτα που
// συνδέεται με τους αισθητήρες να
// στείλει τη τιμή του 3ου αισθητήρα

mem[0][6] = PINB; // αποθηκεύει τη τιμή του 3ου αισθητήρα

SETBIT(PORTC,7); // 1000 0000: επιβεβαιώνει στην πλακέτα
// που συνδέεται με τους αισθητήρες ότι
// διάβασε την τιμή του 3ου αισθητήρα

while ((PINC & (1<<PINC6)) != 0);
CLRBIT(PORTC,7);

while ((PINC & (1<<PINC6)) == 0); // περιμένει την πλακέτα που
// συνδέεται με τους αισθητήρες να
// στείλει τη τιμή του 4ου αισθητήρα

mem[0][7] = PINB; // αποθηκεύει τη τιμή του 4ου αισθητήρα

SETBIT(PORTC,7); // 1000 0000: επιβεβαιώνει στην πλακέτα
// που συνδέεται με τους αισθητήρες ότι
// διάβασε την τιμή του 4ου αισθητήρα

```



```

while ((PINC & (1<<PINC6)) != 0);
CLRBIT(PORTC,7);

while ((PINC & (1<<PINC6)) == 0); // περιμένει την πλακέτα που
// συνδέεται με τους αισθητήρες να
// στείλει τη τιμή του 5ου αισθητήρα

mem[0][8] = PINB; // αποθηκεύει τη τιμή του 5ου αισθητήρα

SETBIT(PORTC,7); // 1000 0000: επιβεβαιώνει στην πλακέτα
// που συνδέεται με τους αισθητήρες ότι
// διάβασε την τιμή του 5ου αισθητήρα

while ((PINC & (1<<PINC6)) != 0); // ενημερώνει την πλακέτα του
CLRBIT(PORTC,7); // διακομιστή διαδικτύου ότι το
// σύστημα έχει τελειώσει τη
// διαδικασία μέτρησης και
// βρίσκεται σε διαδικασία αναμονής
}
PORTA = 0; // μηδενίζεται η πόρτα A
stored_time = real_time; // αποθηκεύεται η ώρα που πάρθηκε η μέτρηση

goto repeat; // επαναλαμβάνει το πρόγραμμα

return (0); // απαραίτητη τελευταία εντολή προγράμματος
}

```



5. Αποτίμηση του συστήματος του διακομιστή διαδικτύου

5.1 Γενικά

Έπειτα από την ολοκλήρωση της κατασκευής ολόκληρου του συστήματος μετρήθηκε η ταχύτητα αποστολής δεδομένων σε ολόκληρο το σύστημα αλλά και στον διακομιστή διαδικτύου. Η ταχύτητα ασφαλούς μετάδοσης δεδομένων μετρήθηκε κοντά στα 11Mbps[13]. Αναλόγως με τις καιρικές συνθήκες την σταθερότητα των κεραιών που τοποθετήθηκαν, η ταχύτητα μπορούσε να φτάσει και τα 40Mbps. Σε αυτές τις περιπτώσεις όμως, μπορεί η ταχύτητα να ήταν μέγιστη άρα και η απόδοση του ασύρματου δικτύου, αλλά δεν υπήρχε σταθερότητα. Έτσι με οποιαδήποτε αλλαγή του καιρού, ακόμη και με χαμηλό αέρα, ολόκληρη η σύνδεση στην καλύτερη περίπτωση γινόταν υπερβολικά αργή, περίπου 0,5 Mbps ή και χανόταν. Στην δεύτερη περίπτωση έπρεπε είτε να περιμένουμε ώστε να επανέρθουν οι συνθήκες της μέγιστης δυνατής ταχύτητας είτε τα access point να ρυθμιστούν σε χαμηλότερη ταχύτητα.

Για να ξεπεραστεί αυτό το πρόβλημα θα μπορούσαν να τοποθετηθούν μεγαλύτερες και πιο αποδοτικές κεραιές ή να τοποθετηθούν ενισχυτές σήματος ή και τα δύο. Η ταχύτητα όμως των 11Mbps είναι υπεραρκετή στην παρούσα εφαρμογή καθώς τα δεδομένα της εφαρμογής είναι μηδαμινά. Έτσι επιλέχθηκε η σταθερή ταχύτητα των 11Mbps.

Ενώ το ασύρματο δίκτυο υποστηρίζει αυτή τη ταχύτητα, η μέγιστη ταχύτητα μεταφοράς δεδομένων περιορίζεται από τον διακομιστή διαδικτύου. Η θεωρητική μέγιστη ταχύτητα της συγκεκριμένης πλακέτας είναι 10Mbps. Η μέγιστη ταχύτητα που μετρήθηκε όμως σε πραγματικές συνθήκες ήταν περίπου 2Mbps. Παρατηρείται λοιπόν ότι στο υπάρχον ασύρματο δίκτυο WiFi χωρίς καμία επιπλέον αναβάθμιση



μπορούν να συνδεθούν αρκετοί διακομιστές διαδικτύου χωρίς να υπάρξει το παραμικρό πρόβλημα με την συνολική ταχύτητα του συστήματος.

Εδώ θα πρέπει να αναφερθούν και οι χρόνοι εκτέλεσης της εντολής ping που μετρήθηκαν από το σημείο στον ποταμό Άραχθο μέχρι το σημείο της ταράτσας του Τμήματος Φυσικής του Πανεπιστημίου Ιωαννίνων. Κατά τη διάρκεια των δοκιμών και της εξέλιξης του συστήματος πριν ολοκληρωθεί μετρήθηκαν χρόνοι αρκετά μεγαλύτεροι από αυτούς που αναφέρονται στη συνέχεια. Οι χρόνοι οι οποίοι μετρήθηκαν στην τελευταία δοκιμή κυμαίνονται από 6 έως 11ms ανά 32 bytes.

5.2 Αξιοπιστία του συστήματος

Κατά τη διάρκεια της ανάπτυξης του hardware και του λογισμικού υπήρχαν πάρα πολλές δυσλειτουργίες που οφείλονταν σε διάφορους λόγους. Ένας από τους βασικότερους λόγους, αλλά ευτυχώς όχι ο πιο συχνός, ήταν στην φύση του συστήματος, καθώς βασίζεται σ' ένα μικροελεγκτή, που δυσλειτουργεί σε μικρές μεταβολές των συνθηκών λειτουργίας. Έτσι έπρεπε να παρθούν όλα τα απαραίτητα μέτρα ώστε τα κολλήματα εξαιτίας αυτού το λόγου να ελαχιστοποιηθούν. Στην ανάπτυξη του συστήματος χρησιμοποιήθηκε αρκετά σταθερή τροφοδοσία, το λογισμικό απλοποιήθηκε όσο το δυνατόν περισσότερο και το φορτίο εργασίας κατανεμήθηκε σε περισσότερες πλακέτες και κατά συνέπεια μικροελεγκτές. Το φορτίο εργασίας του κάθε επεξεργαστή ελαττώθηκε αρκετά με παράλληλη αύξηση της αξιοπιστίας.

Ένας δεύτερος και κυριότερος λόγος που απλοποιήθηκε το λογισμικό του διακομιστή διαδικτύου, ήταν η χρήση του πρωτοκόλλου TCP/IP. Η χρήση του TCP/IP σε συνεργασία με μικροελεγκτή το καθιστά τον μικροελεγκτή ευάλωτο σε κολλήματα καθώς πρέπει και να συνεργάζονται άψογα και παράλληλα με όλα τα υπόλοιπα συστήματα όπως access point, router που το υποστηρίζουν και να είναι εξίσου αξιόπιστα και λειτουργικά. Έτσι αν κάποιο από αυτά δε λειτουργήσει με τον τρόπο που έπρεπε ολόκληρο το σύστημα καταρρέει. Φυσικά δε συμβαίνει το ίδιο τόσο συχνά σ' ένα δίκτυο που αποτελείται μόνο από υπολογιστές, αλλά είναι αδύνατον να αντικατασταθούν όλες οι δυνατότητες και οι λειτουργίες ενός ηλεκτρονικού υπολογιστή από έναν σχετικά απλό μικροελεγκτή. Επίσης έπρεπε να ληφθούν υπ' όψιν



και άλλοι παράγοντες και φυσικές παράμετροι όπως η θερμοκρασία η οποία και αυτή είναι αρκετά σημαντική.

Λαμβάνοντας υπ' όψιν τα παραπάνω και κάνοντας αρκετές δοκιμές διαπιστώθηκε ότι ο διακομιστής έγινε αρκετά αξιόπιστος σε σχέση με αυτό που είχε αρχικά σχεδιαστεί. Παράλληλα για να αυξηθεί η αξιοπιστία, τα αποτελέσματα των μετρήσεων εκτός από τον ηλεκτρονικό υπολογιστή που μπορούν να αποθηκευθούν όταν υπάρχει σε λειτουργία το ασύρματο δίκτυο, αποθηκεύονται και σε ένα άλλο μικροελεγκτή ATMEGA16.

Θα μπορούσε να χρησιμοποιηθεί εξωτερική μνήμη έτσι ώστε οι τελευταίες μετρήσεις να αποθηκεύονται εκεί. Ο λόγος που δεν προτιμήθηκε είναι γιατί μέσα στον μικροελεγκτή υπάρχουν λιγότερες πιθανότητες να υπάρξει σφάλμα στην αποθήκευση της μέτρησης καθώς υπάρχουν περισσότερες δυνατότητες επέκτασης και αναβάθμισης. Μπορούν δηλαδή μελλοντικά τα δεδομένα να στέλνονται με RS232 ή με usb κατευθείαν από τον μικροελεγκτή.

Αν για οποιοδήποτε λόγο το ασύρματο δίκτυο δεν λειτουργεί για αρκετές μέρες, π.χ. μετά από μια αρκετά παρατεταμένη χιονόπτωση αφού εξαντληθεί η ενέργεια της μπαταρίας στον αναμεταδότη, και ακόμα και στην περίπτωση που ο διακομιστής διαδικτύου δυσλειτουργεί, οι μετρήσεις θα συνεχίσουν να αποθηκεύονται στη συγκεκριμένη πλακέτα. Έτσι όταν το δίκτυο ανακτηθεί, στην πραγματικότητα δεν θα έχει χαθεί ούτε μια μέτρηση εκτός αν ξεπεραστεί κάποιο συγκεκριμένο όριο μετρήσεων και εξαντληθεί η μνήμη αποθήκευσης.

5.3 Αξιοπιστία του ασύρματου συστήματος μετάδοσης φυσικών πληροφοριών στο διαδίκτυο

Αρχικά έγινε προσπάθεια όλα τα μέρη του συστήματος του διακομιστή διαδικτύου να ενσωματωθούν σε μια και μόνο μια πλακέτα η οποία θα συνδέονταν με το ασύρματο δίκτυο και θα διαχειριζόταν όλες τις πληροφορίες που θα μπορούσαν να ζητηθούν από τον απομακρυσμένο ηλεκτρονικό υπολογιστή και ταυτόχρονα θα έκανε όλες τις απαραίτητες εργασίες τον κατάλληλο χρόνο.



Διαπιστώθηκε όμως ότι αυτό ήταν δύσκολο να πραγματοποιηθεί. Διαπιστώθηκε ότι μια «απλή» πλακέτα διακομιστή διαδικτύου σε σχετικά μικρό μέγεθος χρειαζόταν τουλάχιστον τέσσερα διαφορετικά επίπεδα (layers) το οποίο ήταν αρκετά δύσκολο και χρονοβόρο για να πραγματοποιηθεί.

Έτσι κατανεμήθηκε η εργασία με τέτοιο τρόπο ώστε ο τρόπος εργασίας να γίνει όσο το δυνατόν πιο αποδοτικός και αποτελεσματικός και τα επιμέρους συστήματα να συνεργάζονται αξιόπιστα.

Κατασκευάστηκε τελικά ένα αξιόπιστο σύστημα που λειτουργεί σε δυσμενείς καιρικές συνθήκες, καθώς η θερμοκρασία στους αναμεταδότες που βρίσκονται σε μεγάλο υψόμετρο, τους χειμερινούς μήνες μπορεί να φτάσει και τους -15°C χωρίς να υπάρχει δυσλειτουργία στο δίκτυο. Το ίδιο απροβλημάτιστα μπορεί να λειτουργεί και το σύστημα ιχνηλάτισης του ήλιου για το φωτοβολταϊκό, το οποίο τροφοδοτεί τον έναν από τους αναμεταδότες.

Σε περίπτωση παρατεταμένης χιονόπτωσης και αφού καταναλωθεί η αποθηκευμένη ενέργεια του συσσωρευτή που τροφοδοτεί τον αναμεταδότη, δηλαδή σε περίπτωση που το δίκτυο δε λειτουργεί, έχει προβλεφθεί να μη χάνεται καμία από τις μετρήσεις που παίρνονται στο ποταμό Άραχθο, αφού οι πιο πρόσφατες μετρήσεις είναι αποθηκευμένες σε ένα μικροελεγκτή, ο οποίος εκτός από την κανονική τροφοδοσία, τροφοδοτείται και από ένα συσσωρευτή. Δηλαδή, ακόμα και αν υπάρξει διακοπή ρεύματος στο παλιό τελωνείο στο ποταμό Άραχθο, δε θα χαθεί καμία αποθηκευμένη μέτρηση καθώς ο συσσωρευτής είναι ικανός να τροφοδοτεί τον μικροελεγκτή για τουλάχιστον είκοσι μέρες.



6. Συμπεράσματα-Επεκτάσεις

6.1 Συμπεράσματα για το σύστημα του διακομιστή διαδικτύου

Το σύστημα του διακομιστή διαδικτύου βασισμένο στον μικροελεγκτή ATMEGA128 της Atmel μπορεί να θεωρηθεί ως μια πολύ καλή και φθηνή λύση για τις δυνατότητες και τις λειτουργίες που προσφέρει. Αρκεί να σκεφτεί κανείς ότι ένας μικρός και φθηνός υπολογιστής που έχει περίπου την ίδια λειτουργία και περίπου τις ίδιες δυνατότητες έχει στην καλύτερη περίπτωση πενταπλάσιο κόστος από το κόστος της πλακέτας. Και πάλι δεν έχει όλες τις δυνατότητες του συστήματος αφού πρέπει να αγοραστεί και λογισμικό και απαιτείται η ανάπτυξη του λογισμικού της εφαρμογής. Άρα στην πραγματικότητα το κόστος είναι αρκετά μεγαλύτερο.

Το σύστημα του διακομιστή διαδικτύου έχει πολλές λειτουργίες καθώς μπορεί να φιλοξενεί μια ή και περισσότερες ιστοσελίδες αλλά το κυριότερο μπορεί να εκτελεί διάφορες αυτοματοποιημένες εργασίες, τα αποτελέσματα των οποίων φαίνονται σε μια ή σε περισσότερες ιστοσελίδες. Το μέγεθος των ιστοσελίδων όμως δεν μπορεί να είναι αρκετά μεγάλο καθώς περιορίζεται από την σχετικά μικρή μνήμη που διαθέτει σε σχέση μ'ένα ηλεκτρονικό υπολογιστή. Όμως η φύση της γλώσσας HTML και κατά συνέπεια των ιστοσελίδων είναι τέτοια που δεν χρειάζεται μεγάλη μνήμη για να πάρουμε τα επιθυμητά αποτελέσματα. Το σύστημα του διακομιστή διαδικτύου έχει επίσης την δυνατότητα να διαχειρίζεται απομακρυσμένα με τον κατάλληλο προγραμματισμό πολλές συσκευές. Έχει επίσης και πολλές εισόδους και έτσι είναι δυνατόν να παίρνει και ανάδραση ξέροντας κάθε στιγμή αν πραγματικά εκτελέστηκε η εντολή που δόθηκε απομακρυσμένα.



Έχοντας όλες τις παραπάνω δυνατότητες μπορεί κανείς εύκολα να συμπεράνει ότι η χρησιμότητα του είναι μεγάλη και μπορεί να χρησιμοποιηθεί σ' ένα μεγάλο εύρος εφαρμογών ακόμη και σε αρκετά αντίξοες περιβαλλοντικές συνθήκες καθώς το εύρος λειτουργίας αυτής της πλακέτας, π.χ. όσο αναφορά τη θερμοκρασία λειτουργίας είναι αρκετά μεγάλο.

6.2 Σελίδα HTML μελλοντικής επέκτασης λειτουργιών και αποστολής εντολών

Η δομή του προγράμματος μπορεί να υποστηρίξει περισσότερες από τρεις σελίδες που έχει γίνει στην παρούσα διπλωματική. Συγκεκριμένα, μπορεί να υποστηρίξει αρκετά μεγάλο αριθμό σελίδων αρκεί να μην υπερβεί σε όγκο την μνήμη που διαθέτει η συγκεκριμένη πλακέτα με τον επεξεργαστή. Επειδή η κατασκευή μιας ακόμα υποσελίδας δεν απαιτεί μεγάλη μνήμη, είναι δυνατόν στη συγκεκριμένη σελίδα να δίνονται απομακρυσμένα κάποιες εντολές, δηλαδή να ελέγχονται οι εισοδοί και έξοδοι του επεξεργαστή, (αυτοί που δεν χρησιμοποιούνται στη συγκεκριμένη διπλωματική) έτσι ώστε να ελέγχονται διάφορες συσκευές. Επειδή υπάρχει η δυνατότητα να ελέγχονται και οι εισοδοί, μπορεί να υπάρχει ανάδραση από τις συσκευές έτσι ώστε να επιβεβαιώνεται αν εκτελέστηκαν σωστά οι εντολές που δόθηκαν.

6.3 Επέκταση του διακομιστή διαδικτύου σε περισσότερες εφαρμογές

Το σύστημα του διακομιστή διαδικτύου, με μια περαιτέρω αναβάθμιση, μπορεί σχετικά εύκολα να χρησιμοποιηθεί σε περισσότερες εφαρμογές. Όταν δεν βρίσκεται σε διαδικασία μέτρησης, μπορεί να χρησιμοποιηθεί για να ελέγχει απομακρυσμένα κάποιες συσκευές, είτε αυτές έχουν σχέση με το συγκεκριμένο σύστημα, είτε είναι κάτι τελείως διαφορετικό. Μπορεί δηλαδή να χρησιμοποιηθεί για να θέσει σε λειτουργία την αντλία νερού και όλα τα άλλα απαραίτητα συστήματα, όπως ηλεκτροβαλβίδες και άλλα έτσι ώστε να παρθεί μια μέτρηση την συγκεκριμένη χρονική



στιγμή. Μπορεί επίσης να χρησιμοποιηθεί ώστε να ελέγχει άλλες εργασίες που χρειάζεται να γίνονται στο χώρο που έχει εγκατασταθεί. Μια πιθανή εφαρμογή είναι να ποτίζει όποτε χρειάζεται τα δέντρα και το πράσινο του χώρου, να ενεργοποιεί, απενεργοποιεί και να ελέγχει ένα συναγερμό, να ελέγχει αν κάποιες συσκευές λειτουργούν σωστά, όπως αν έχει την απαραίτητη ψύξη το ψυγείο. Γενικότερα θα μπορούσε κάποιος να πει ότι μπορεί να χρησιμοποιηθεί έτσι ώστε να επισκέπτεται κάποιος το μέρος μόνο όταν υπάρχει ανάγκη.

Είναι αρκετά πιθανό, ένα πιο εξελιγμένο μοντέλο από αυτό, με περισσότερες εισόδους εξόδους και πιθανόν παράλληλη επεξεργασία, να μπορεί να ελέγχει ταυτόχρονα όλες τις συσκευές ενός σπιτιού, ή στη συγκεκριμένη περίπτωση να ελέγχει όλες τις παραμέτρους έτσι ώστε σε περίπτωση κάποιου μικρού ή μεγάλου προβλήματος να ειδοποιεί αυτόματα, με τηλέφωνο, ηλεκτρονικό ταχυδρομείο, γραπτό μήνυμα σε κινητό την κατάσταση που βρίσκεται το σύστημα και αν γενικότερα υπάρχει κάποιο πρόβλημα ή αλλαγή κατάστασης. Δηλαδή σε περίπτωση που κάποια συγκεκριμένη παράμετρος ή και περισσότερες περάσουν ένα συγκεκριμένο όριο, τότε υπάρχει λόγος ανησυχίας και έρευνας, να μη χρειάζεται να παρακολουθούνται καθημερινά οι μετρήσεις αλλά με μια απλή ειδοποίηση να υπάρχει ενημέρωση ότι υπάρχει πρόβλημα.

Όλες οι παραπάνω λειτουργίες είναι αρκετά χρήσιμες και θα μπορούσαν να γίνουν περισσότερο χρήσιμες αν συνδυαστούν με τη χρήση μιας ή και περισσότερων καμερών IP. Έτσι θα μπορεί κάποιος ταυτόχρονα με τις παραπάνω μετρήσεις μέσα από μια IP κάμερα να βλέπει τον καιρό, αν υπάρχει κάποια διαρροή νερού από το δοχείο, αν κάποιος αισθητήρας έχει πρόβλημα και το δοχείο είτε γεμίζει ανεξέλεγκτα είτε δε γεμίζει στο απαιτούμενο ύψος, αν υπάρχει παραβίαση ασφαλείας να καταγράφεται στην κάμερα και να υπάρχει γενικότερα μια επίβλεψη του χώρου. Βέβαια για να επιτευχθεί αυτό, εκτός των άλλων, χρειάζεται απαραίτητα και αναβάθμιση του ασύρματου δικτύου έτσι ώστε να υποστηρίζει μεγαλύτερες ταχύτητες δεδομένων.



Αναφορές

- [1] John Ray, Using TCP/IP Special Edition, Pearson Education (US), 1999
- [2] McMahon Richard, Εισαγωγή στα δίκτυα υπολογιστών, Γκιούρδας Μ., 2004
- [3] Ciccarelli, Patrick, Faulkner, Christina, Δίκτυα υπολογιστών, Γκιούρδας Μ., 2005
- [4] Gilbert Held, Ethernet Networks, John Wiley and Sons Ltd, 1998
- [5] Raymond Smith, WiFi Home Networking, McGraw-Hill Education - Europe, 2003
- [6] Michael Predko, Programming and Customizing the 8051 Microcontroller, McGraw-Hill Education - Europe, 1999
- [7] www.soc-machines.com
- [8] www.ethernut.de/en/hardware/ethernuts.html
- [9] www.realtek.com
- [10] www.ethernut.de/en/download/index.html
- [11] winavr.sourceforge.net/download.html
- [12] www.adobe.com/products/dreamweaver/?promoid=BPDEC
- [13] , NICOPOLITIDIS, OBAIDAT, PAPADIMITRIOY Ασύρματα Δίκτυα, ΚΛΕΙΔΑΡΙΘΜΟΣ, 2006
- [14] Bruce Archambeault, PCB Design for Real-world EMI Control, Kluwer Academic Publishers Group, 2002
- [15] www.atmel.com



[16] www.datasheet.com

[17] David C. Black, Jack, System C, Donovan, Kluwer Academic Publishers Group, 2004

[18] el.wikipedia.org/wiki/Γεωγραφικές_συντεταγμένες

[19] www.mozilla.com/firefox

[20] en.wikipedia.org/wiki/Gigabit_Ethernet

[21] <http://www.ikalogic.com/isp.php>

[22] http://www.atmel.com/dyn/Products/tools_card.asp?tool_id=2725

[23] http://www.atmel.com/dyn/products/Product_card.asp?part_id=2010

[24] <http://winavr.sourceforge.net/>

[25] http://www.atmel.com/dyn/products/product_card.asp?part_id=2018

[26] <http://www.lancos.com/prog.html>

