ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αυτόνομος ηλιακός εξυπηρετητής διαδικτύου με ασύρματη σύνδεση και αισθητήρες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νεκτάριος-Γεώργιος Σ. Τσούτσος

Επιβλέπων : Κιαμάλ Ζ. Πεκμεστζή
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2008

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αυτόνομος ηλιακός εξυπηρετητής διαδικτύου με ασύρματη σύνδεση και αισθητήρες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νεκτάριος-Γεώργιος Σ. Τσούτσος

Επιβλέπων : Κιαμάλ Ζ. Πεκμεστζή
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την .....η Σεπτεμβρίου 2008.

.....	.....	.....
Κ. Πεκμεστζή	Ν. Μήτρου	Γ. Οικονομάκος
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Λέκτορας Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2008

.....

Νεκτάριος-Γεώργιος Σ. Τσούτσος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Αυτή η σελίδα δεν εκτυπώνεται

Περίληψη

Στην εργασία αυτή περιγράφεται αναλυτικά ένα ενσωματωμένο σύστημα που βασίζεται στον μικροελεγκτή AVR mega168 της ATMEL. Το σύστημα υλοποιεί ένα εξυπηρετητή διαδικτύου, ο οποίος δέχεται αιτήματα μέσω ασύρματου τοπικού δικτύου. Το σύστημα διαθέτει επίσης ένα πλήθος αισθητήρων και έχει τη δυνατότητα να αποστέλλει τις μετρήσεις σε μορφή ιστοσελίδας. Η τροφοδοσία του συστήματος γίνεται με τη βοήθεια ηλιακού συλλέκτη και μπαταρίας. Το σύστημα συμπληρώνει ένας σταθμός βάσης που έχει τη δυνατότητα να αποθηκεύει τις μετρήσεις σε βάση δεδομένων.

Λέξεις κλειδιά

Ενσωματωμένο σύστημα, εξυπηρετητής διαδικτύου, megaAVR, Ethernet, μπαταρία οξέων μολύβδου, υποβιβασμός συνεχούς τάσης, ασύρματο δίκτυο, βάση δεδομένων, μετατροπέας αναλογικού σε ψηφιακό, στοίβα TCP/IP, μνήμη EEPROM, ηλιακός συλλέκτης, βιβλιοθήκη avr-libc, τριπλή χειραψία, γλώσσα XML.

Abstract

This diploma thesis describes in detail an embedded system based on ATMEL AVR mega168 microcontroller. The system includes a dedicated web server that accepts requests via a wireless network. The system incorporates many sensors and has the ability to create web pages with sensor measurements. The power comes from a solar panel and a battery. The project also includes a base station that stores every measurement in a database.

Keywords

Embedded system, web server, megaAVR, Ethernet, sealed lead acid battery, DC-to-DC converter, wireless network, data base, Analog-to-Digital Converter, TCP/IP stack, EEPROM memory, solar panel, solar charger, avr-libc, XML.

Ευχαριστίες

Από τη θέση αυτή θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον καθηγητή μου κ. Κιαμάλ Πεκμεστζή για τη θέρμη με την οποία δέχτηκε την ιδέα μου για τη διπλωματική αυτή, για την εύστοχη καθοδήγηση του και τη συμπαράστασή του κατά τη διάρκεια εκπόνησης της εργασίας. Επίσης θα ήθελα να ευχαριστήσω όσους συνέβαλαν στην υλοποίηση αυτού του συστήματος και ιδιαίτερα όλα τα παιδιά στο Microlab. Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για την στήριξη που μου προσέφεραν όλο αυτό το διάστημα.

Πίνακας Περιεχομένων

Ευρετήριο Εικόνων

Εισαγωγή

Στην εποχή μας η ανάγκη για μικρότερα υπολογιστικά συστήματα αυξάνεται διαρκώς. Οι λόγοι που προκαλούν αυτή την ανάγκη είναι κυρίως πρακτικοί: ένα μικρότερο υπολογιστικό σύστημα καταλαμβάνει λιγότερο χώρο, μεταφέρεται ευκολότερα και συνήθως απορροφά λιγότερη ενέργεια. Η σημερινή τεχνολογική πρόοδος μας επιτρέπει να αυξήσουμε δραματικά το επίπεδο ολοκλήρωσης των ηλεκτρονικών κυκλωμάτων, με αποτέλεσμα να κατασκευάζονται πολύ μικρά υπολογιστικά συστήματα γενικής χρήσης, όπως οι φορητοί υπολογιστές και οι υπολογιστές χειρός. Αν ζητάμε ένα υπολογιστικό σύστημα που εκτελεί μόνο ορισμένες λειτουργίες, τότε το μέγεθος μπορεί να μειωθεί ακόμα περισσότερο. Χαρακτηριστικό είναι το παράδειγμα των embedded συστημάτων, τα οποία συνήθως απαρτίζονται από κάποιο μικροϋπολογιστή που επικοινωνεί με περιφερειακές συσκευές [1].

Ο εξυπηρετητής διαδικτύου (web server) είναι ένα παράδειγμα υπολογιστικού συστήματος που εκτελεί μόνο μία λειτουργία. Η λειτουργία αυτή είναι να δέχεται αιτήσεις από άλλους υπολογιστές και να απαντά αποστέλλοντας μια ιστοσελίδα με κατάλληλο περιεχόμενο. Η ανάγκη για web servers γεννήθηκε το 1989 όταν ο Tim Berners-Lee πρότεινε ένα ευκολότερο σύστημα ανταλλαγής μηνυμάτων για το ερευνητικό κέντρο CERN ενώ πρώτος web server λειτούργησε σε έναν υπολογιστή NeXT [2]. Από τότε η τεράστια διάδοση του διαδικτύου οδήγησε σήμερα σε εκατομμύρια υπολογιστές γενικής χρήσης οι οποίοι λειτουργούν ως εξυπηρετητές διαδικτύου εκτελώντας κατάλληλα προγράμματα [3].

Περιγραφή του συστήματος

Παρόλο που ένας εξυπηρετητής διαδικτύου υλοποιείται συνήθως σε κάποιον υπολογιστή γενικής χρήσης, είναι πολύ ενδιαφέρουσα η περίπτωση στην οποία η υλοποίηση γίνεται σε ένα embedded σύστημα με τη βοήθεια μικροϋπολογιστή. Στην εργασία αυτή συνδυάζουμε την υλοποίηση ενός embedded web server με ένα σταθμό αισθητήρων, ο οποίος συλλέγει δεδομένα και τα αποστέλλει στο διαδίκτυο.

Πιο αναλυτικά το σύστημα που υλοποιήθηκε περιλαμβάνει:

- ένα web server που βασίζεται στον μικροελεγκτή megaAVR της ATMEL και τον ελεγκτή Ethernet της Microchip
- μια ομάδα από ειδικούς αισθητήρες
- μια οθόνη υγρών κρυστάλλων για παρουσίαση των δεδομένων

- μια μπαταρία οξέων μολύβδου (sealed lead acid) για αποθήκευση ενέργειας
- έναν ηλιακό συλλέκτη για ενεργειακή αυτονομία
- ένα φορτιστή μπαταρίας κατάλληλο για ηλιακούς συλλέκτες
- δύο μετατροπείς συνεχούς τάσης (DC-to-DC converters)
- μία κεραία WiFi για ασύρματη επικοινωνία
- ένα δευτερεύον σύστημα που αποθηκεύει πληροφορίες σε βάση δεδομένων

Όπως φαίνεται από τα προηγούμενα, το βασικό χαρακτηριστικό του συστήματος είναι η απουσία μόνιμης καλωδιακής σύνδεσης για την μεταφορά δεδομένων και την τροφοδοσία. Τα δεδομένα μεταδίδονται ασύρματα μέσω τοπικού δικτύου WiFi το οποίο έχει τυπική εμβέλεια τουλάχιστον 100 μέτρα σε εξωτερικούς χώρους [4] και δίνει τη δυνατότητα εγκατάστασης του συστήματος σε απομονωμένα σημεία όπως για παράδειγμα το ψηλότερο σημείο ενός κτηρίου. Ο ηλιακός συλλέκτης εξασφαλίζει την απαιτούμενη ηλεκτρική ενέργεια για καθημερινή λειτουργία ενώ στην περίπτωση ανεπάρκειας ηλιακής ακτινοβολίας γίνονται ειδικές περικοπές της κατανάλωσης ώστε το σύστημα να γίνει περισσότερο ευέλικτο ενεργειακά. Τα δεδομένα των αισθητήρων αποστέλλονται μέσω του εξυπηρετητή διαδικτύου σε ένα σταθμό βάσης, ο οποίος στη συνέχεια τα αποθηκεύει σε μια βάση δεδομένων MySQL. Παράλληλα το σύστημα δίνει τη δυνατότητα στους χρήστες του διαδικτύου να διαβάσουν τις τελευταίες μετρήσεις απευθείας από τον web server του συστήματος, ενώ τα δεδομένα της βάσης δεδομένων παρουσιάζονται και συγκεντρωτικά στο διαδίκτυο με τη μορφή γραφημάτων.

Γνώσεις που αποκτήθηκαν

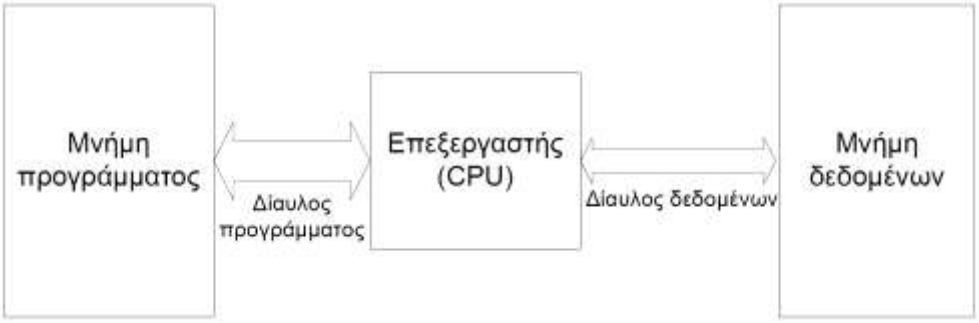
Στην παρούσα εργασία μελετήσαμε αναλυτικά τον προγραμματισμό μικροελεγκτών AVR στη γλώσσα C, με έμφαση στην υλοποίηση Analog to Digital Conversion, στην υλοποίηση διακοπών, στην υλοποίηση επικοινωνίας Serial Programming Interface, στον προγραμματισμό οθόνης LCD και στον προγραμματισμό μνήμης EEPROM. Επίσης μελετήθηκε διεξοδικά η στοίβα TCP/IP σε συνδυασμό με την κατασκευή πλαισίων Ethernet. Ασχοληθήκαμε ακόμα με μετατροπείς συνεχούς τάσης (buck converters) και με τη διαδικασία φόρτισης επαναφορτιζόμενων μπαταριών. Για την αποθήκευση δεδομένων στο επίκεντρο βρέθηκε η βάση δεδομένων MySQL σε συνδυασμό με τον προγραμματισμό σε περιβάλλον Visual Basic .NET και την χρήση της γλώσσας XML. Τέλος αυτή η εργασία ήταν η αφορμή για σχεδιασμό και κατασκευή τυπωμένων κυκλωμάτων με υπολογιστή, μια εμπειρία ιδιαίτερα χρήσιμη σε κάθε νέο Ηλεκτρολόγο Μηχανικό και Μηχανικό Υπολογιστών.

Μέρος 1ο Hardware

Κεφάλαιο 1
Server

1.1 Ο μικροελεγκτής AVR mega168

Οι μικροελεγκτές AVR ανήκουν στην οικογένεια επεξεργαστών RISC των 8-bit και κύριο χαρακτηριστικό τους είναι ότι ενσωματώνουν στο ίδιο chip ξεχωριστές μνήμες δεδομένων και εντολών. Αυτή η αρχιτεκτονική, γνωστή με το όνομα «Τροποποιημένη Αρχιτεκτονική Harvard», εξασφαλίζει ότι τα δεδομένα και οι εντολές βρίσκονται σε διαφορετικό χώρο διευθύνσεων και η επικοινωνία με την CPU γίνεται μέσω ανεξάρτητων διαύλων. Η ύπαρξη ανεξάρτητων διαύλων καθιστά την αρχιτεκτονική Harvard ιδιαίτερα αποδοτική, καθώς επιτρέπει παράλληλη προσπέλαση στις μνήμες. Για την μόνιμη αποθήκευση εντολών χρησιμοποιείται μνήμη Flash, ενώ για την προσωρινή αποθήκευση δεδομένων υπάρχει μνήμη SRAM. Στους περισσότερους μικροελεγκτές AVR υπάρχει ακόμα μνήμη EEPROM που προορίζεται για μόνιμη αποθήκευση δεδομένων.



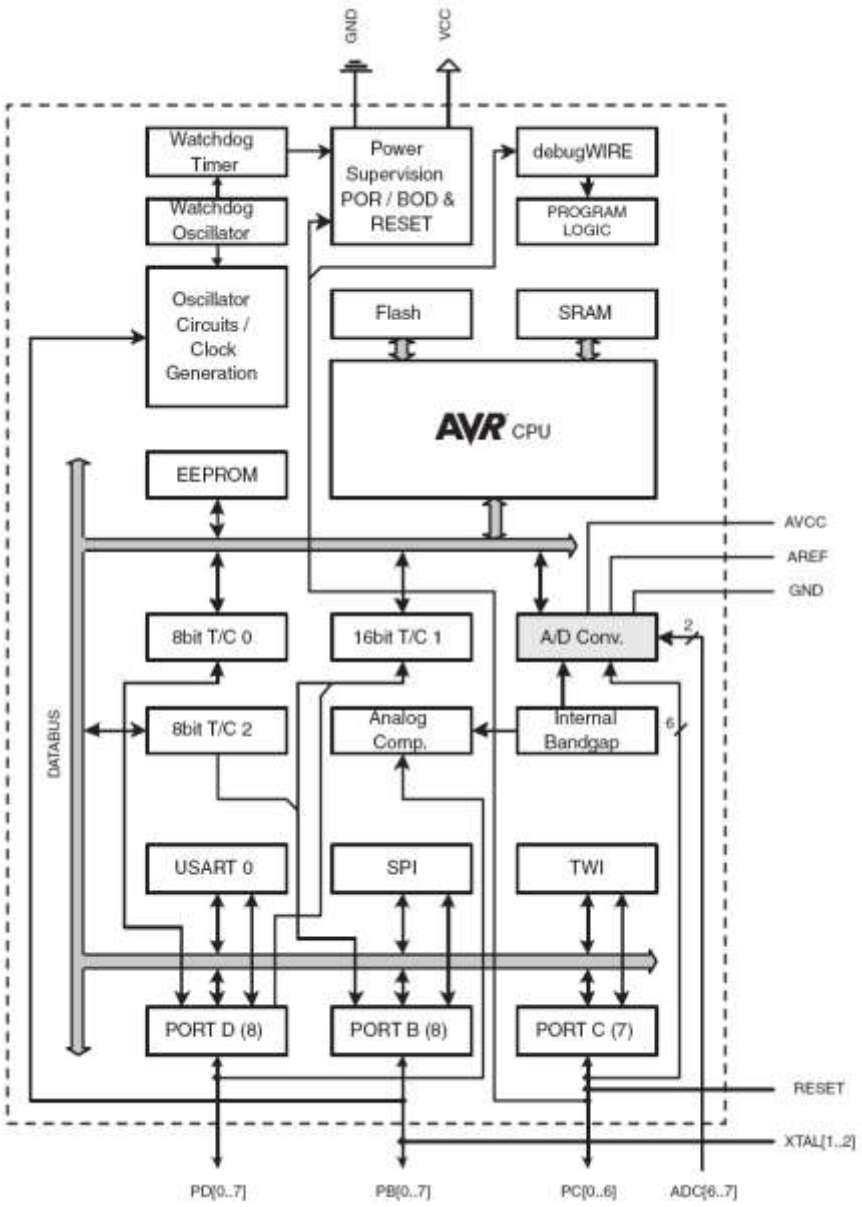
Εικόνα 1: Αρχιτεκτονική Harvard

1.1.1 Χαρακτηριστικά

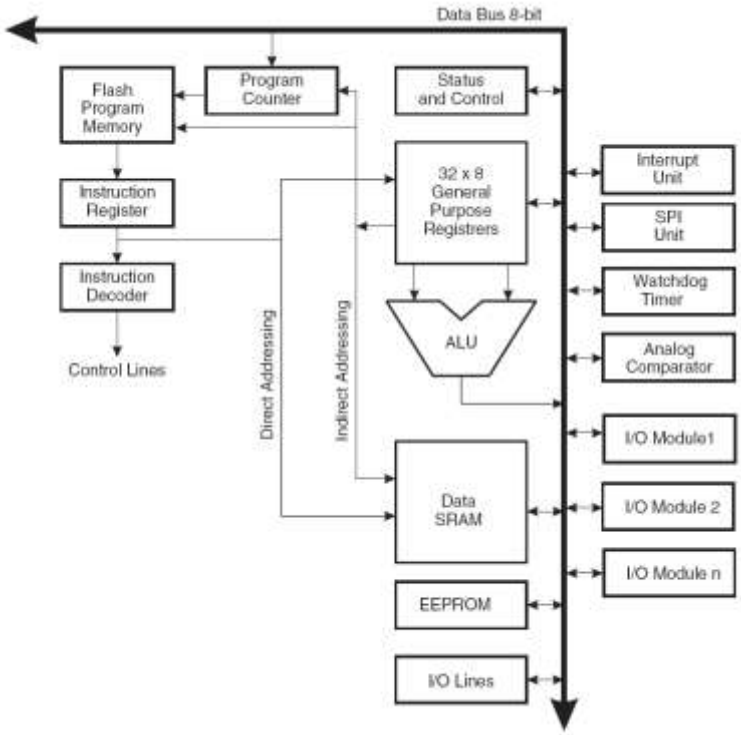
Η ATMEL πρόσφατα κυκλοφόρησε τον ATmega168 που αποτελεί βελτίωση του παλαιότερου ATmega16. Ο ATmega168 διαθέτει 16KB μνήμης Flash με δυνατότητα ανάγνωσης κατά τη διάρκεια της εγγραφής, 1KB μνήμης SRAM και 512 Bytes μνήμης EEPROM. Επίσης διαθέτει 32 καταχωρητές 8-bit γενικού σκοπού συνδεδεμένους απευθείας στην Αριθμητική και Λογική Μονάδα (ALU). Η απόδοση του επεξεργαστή φτάνει το 1MIPS ανά MHz, ενώ η μέγιστη συχνότητα λειτουργίας είναι τα 20MHz με χρήση εξωτερικού ταλαντωτή. Εναλλακτικά μπορεί να χρησιμοποιηθεί ο ενσωματωμένος ταλαντωτής ο οποίος εξασφαλίζει συχνότητες έως 8MHz. Η μνήμη

Flash έχει αντοχή 10^4 κύκλους εγγραφής ενώ η μνήμη EEPROM έχει αντοχή 10^5 κύκλους [5].

Πολύ σημαντικό περιφερειακό του μικροελεγκτή ATmega168 είναι ο ενσωματωμένος Analog to Digital Converter 6 καναλιών και εύρους 10 bit. Εξίσου σημαντική είναι η διεπαφή debugWire που προσφέρει δυνατότητα on-chip debugging με χρήση των σημάτων RESET, VCC και GND. Τα περιφερειακά τον μικροελεγκτή συμπληρώνουν ένας ελεγκτής USART, ένας χρονιστής επιφυλακής (Watchdog Timer), ένας ελεγκτής Serial Programming Interface και ένας ελεγκτής Two Wire Interface για σειριακή επικοινωνία με άλλες συσκευές [5]. Στα ακόλουθα σχήματα φαίνονται block διαγράμματα του μικροελεγκτή.



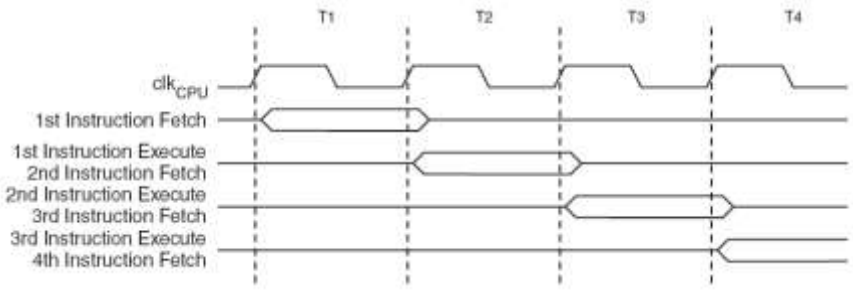
Εικόνα 2: Εσωτερική οργάνωση του μικροελεγκτή



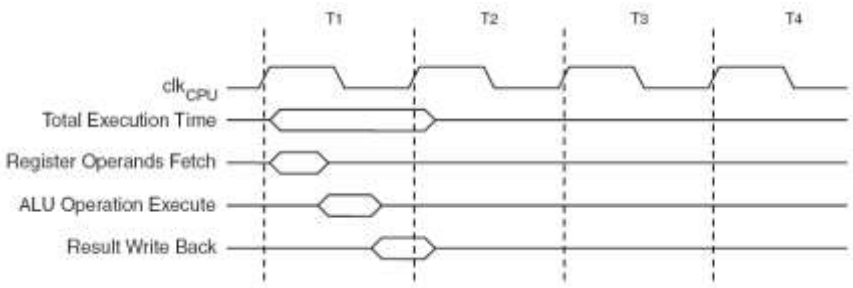
Εικόνα 3: Block διάγραμμα αρχιτεκτονικής AVR

1.1.2 Η Κεντρική Μονάδα Επεξεργασίας

Η CPU του ATmega168 για να εξασφαλίσει την ορθή εκτέλεση του προγράμματος, πρέπει να διαθέτει άμεση πρόσβαση στις μνήμες, να έχει τη δυνατότητα εκτέλεσης υπολογισμών, και να ελέγχει απευθείας τις περιφερειακές συσκευές. Οι εντολές του προγράμματος εκτελούνται σε pipeline ενός επιπέδου που εξασφαλίζει την έναρξη της εκτέλεσης μιας νέας εντολής προτού να ολοκληρωθεί η προηγούμενη.



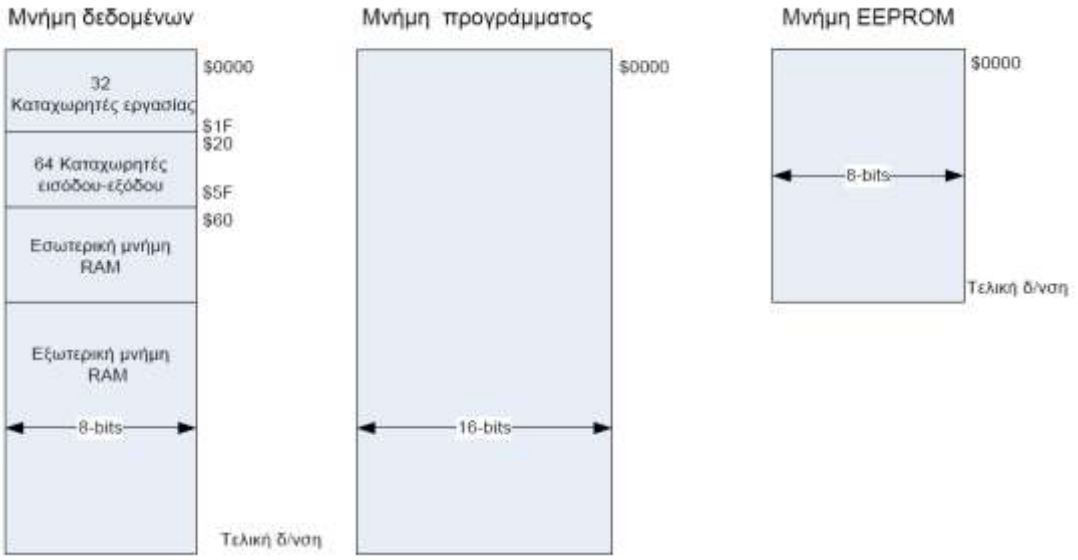
Εικόνα 4: Pipeline



Εικόνα 5: Ένας κύκλος ALU

1.1.3 Οι μνήμες

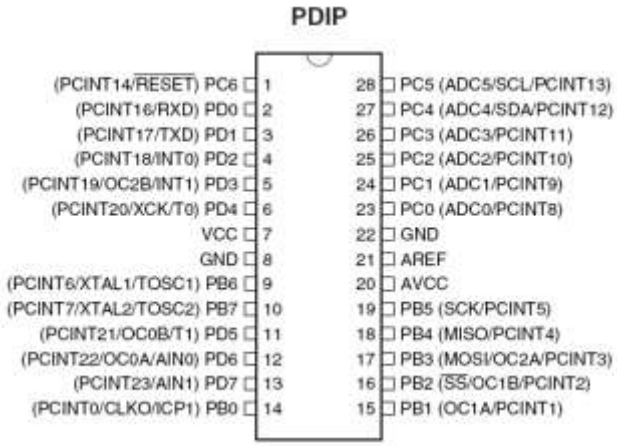
Η μνήμη Flash του μικροελεγκτή ATmega168 είναι η μνήμη που προορίζεται για την αποθήκευση των εντολών του προγράμματος. Ο προγραμματισμός της γίνεται από ειδικά αναπτυξιακά συστήματα όπως το STK500 και το AVRDragon. Αντιθέτως η μνήμη SRAM χρησιμοποιείται κυρίως για αποθήκευση προσωρινών δεδομένων κατά τη διάρκεια εκτέλεσης του προγράμματος, όπως για παράδειγμα τα δεδομένα της στοίβας. Η μνήμη SRAM είναι αρκετά μικρότερη από την Flash, και για το λόγο αυτό δίνεται η δυνατότητα αποθήκευσης δεδομένων στην μνήμη Flash από το ίδιο το πρόγραμμα κατά τον χρόνο εκτέλεσης (εντολή SPM). Η μνήμη EEPROM δίνει την δυνατότητα μόνιμης αποθήκευσης δεδομένων ανεξάρτητα από την μνήμη Flash. Η χρησιμότητα αυτής της μνήμης φαίνεται περισσότερο αν αναλογιστούμε ότι η EEPROM έχει δεκαπλάσια διάρκεια ζωής από την μνήμη Flash, και επομένως είναι ιδανική για επαναλαμβανόμενη αποθήκευση δεδομένων που δεν χάνονται αν σταματήσει η τροφοδοσία. Στο ακόλουθο σχήμα φαίνεται ο χάρτης μνήμης του μικροελεγκτή ATmega168.



Εικόνα 6: Οργάνωση μνήμης

1.1.4 Οι θύρες Εισόδου-Εξόδου

Ο ATmega168 έχει 28 ακροδέκτες εισόδου-εξόδου, από τους οποίους οι 23 είναι πλήρως προγραμματιζόμενοι. Οι ακροδέκτες αυτοί ομαδοποιούνται σε 3 θύρες B, C και D και μπορούν να χρησιμοποιηθούν είτε ως ψηφιακές θύρες γενικής χρήσης, είτε με βάση τις εναλλακτικές λειτουργίες τους. Κάθε ακροδέκτης μπορεί να χρησιμοποιηθεί ανεξάρτητα από τους υπόλοιπους είτε για είσοδο είτε για έξοδο δεδομένων. Οι εναλλακτικές λειτουργίες μια θύρας μπορεί να είναι: είσοδος του Analog to Digital Converter, είσοδος εξωτερικού ταλαντωτή, είσοδος-έξοδος σειριακής επικοινωνίας (USART, TWI, SPI) ή εξωτερική διακοπή.



Εικόνα 7: Διάταξη ακίδων της συσκευασίας PDIP

1.2 Προτίμηση του ATmega168

Η επιλογή μικροελεγκτή AVR έγινε κυρίως λόγω της κυρίαρχης θέσης που κατέχει η κατασκευάστρια εταιρία ATMEL τα τελευταία χρόνια, με αποτέλεσμα την ύπαρξη πληθώρας βοηθητικών εργαλείων υλοποίησης κυκλωμάτων, προσομοίωσης, ελέγχου και διόρθωσης προγραμμάτων. Σε σχέση με ανταγωνιστικά προϊόντα ο AVR παρέχει το πιο πλήρες πακέτο περιφερειακών συσκευών γενικής χρήσης με πολύ υψηλές συχνότητες ταλαντωτή και ιδιαίτερα χαμηλές τάσεις τροφοδοσίας.

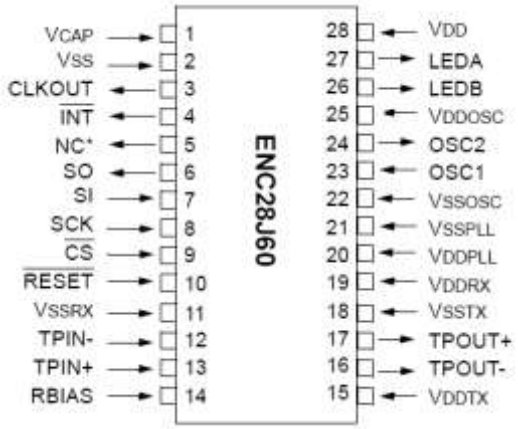
Η επιλογή συγκεκριμένα του μικροελεγκτή ATmega168 έγινε κυρίως επειδή συνδυάζει μεγάλη μνήμη Flash, με ικανοποιητική μνήμη SRAM και υψηλή συχνότητα ταλαντωτή (20 MHz) σε ένα ολοκληρωμένο 28 μόλις ακίδων. Πολύ σημαντική θεωρείται η ύπαρξη debugWire interface που εξασφαλίζει τον έλεγχο και τη διόρθωση του προγράμματος απευθείας στο τελικό κύκλωμα (on-chip debugging). Τα χαρακτηριστικά αυτά απουσίαζαν από τον παλαιότερο ATmega16. Με δεδομένες τις α-

παιτήσεις της εφαρμογής, η επιλογή του ATmega168 υπερκαλύπτει τις ανάγκες σε υπολογιστική ισχύ και αριθμό ακίδων.

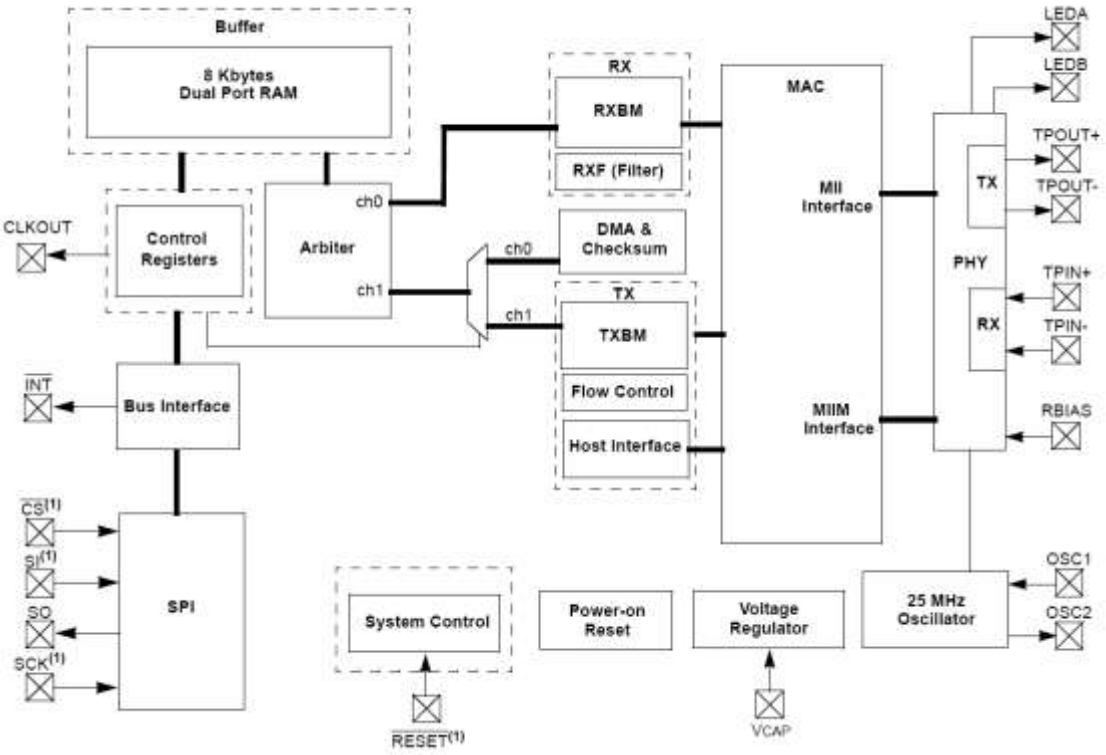
1.3 Ο Ethernet controller

Το Ethernet ήταν πάντοτε μια αρκετά πολύπλοκη διεπαφή και μέχρι σήμερα τα περισσότερα chip που την υλοποιούσαν είχαν 100 ή και περισσότερους ακροδέκτες. Το μειονέκτημα ήταν ότι χρειάζονταν μεγάλους μικροελεγκτές με άφθονη μνήμη για να λειτουργήσουν. Όλα όμως άλλαξαν όταν η εταιρία Microchip, γνωστή για την οικογένεια μικροελεγκτών PIC, κατασκεύασε τον ελεγκτή ENC28J60. Αυτός ο ελεγκτής Ethernet έχει 28 ακίδες και επικοινωνεί με τον μικροελεγκτή μέσω του ιδιαίτερα εύχρηστου Serial Programming Interface. Έτσι ανοίγει ο δρόμος για χιλιάδες εφαρμογές που βασίζονται στην μετάδοση δεδομένων μέσω Ethernet.

Ο ελεγκτής ENC28J60 είναι συμβατός με το πρότυπο IEEE 802.3 και συνδέεται με δίκτυα 10/100/1000Base-T. Στο εσωτερικό του υπάρχει υλοποιημένο το Physical Layer για δίκτυο 10Base-T σε συνδυασμό με το Medium Access Control Layer (MAC), ενώ υποστηρίζει αυτόματο εντοπισμό και διόρθωση της πολικότητας της θύρας Ethernet. Επίσης έχει δυνατότητα εντοπισμού συγκρούσεων και ελέγχου πλαισίων. Η ταχύτητα επικοινωνίας του διαύλου SPI μπορεί να φτάσει τα 20MHz και για την προσωρινή αποθήκευση των πλαισίων υπάρχει μνήμη 8KB. Το επίπεδο MAC υποστηρίζει μετάδοση Unicast, Multicast και Broadcast, ενώ υπάρχει και δυνατότητα Magic Packet για τη λειτουργία Wake-on-LAN [6]. Στο ακόλουθο σχήμα φαίνεται η διάταξη των ακίδων και το block διάγραμμα.

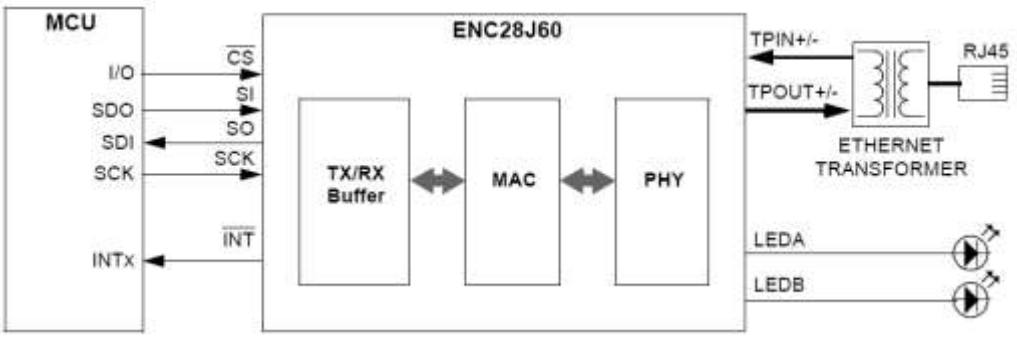


Εικόνα 8: Διάταξη ακίδων της συσκευασίας PDIP

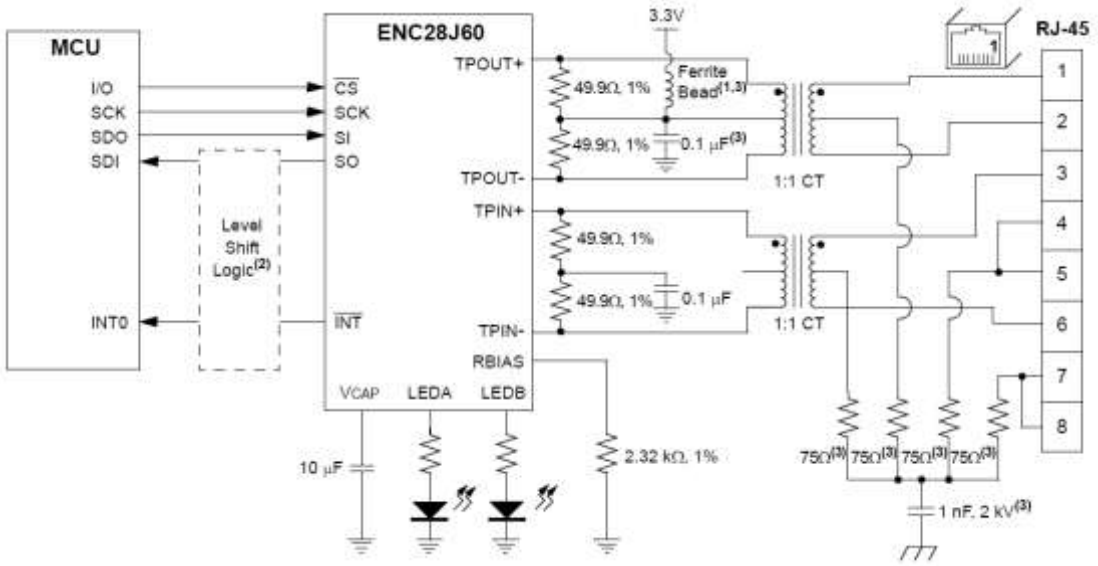


Εικόνα 9: Block διάγραμμα του ελεγκτή Ethernet

Για την ολοκλήρωση του ελεγκτή Ethernet απαιτούνται ακόμα ένας κρύσταλλος ταλάντωσης 25MHz και μία θύρα RJ45 με ενδεικτικά LED αποστολής-λήψης και ενσωματωμένο μετασχηματιστή σημάτων (Magnetics). Το καλώδιο που χρησιμοποιείται για την μετάδοση είναι ανεστραμμένου ζεύγους και το μήκος του μπορεί να φτάσει τα 100 μέτρα [7]. Στο ακόλουθο σχήμα παρουσιάζεται το ολοκληρωμένο σύστημα.



Εικόνα 10: Block διάγραμμα επικοινωνίας



Εικόνα 11: Κυκλωματικό διάγραμμα

1.4 Η σχεδίαση του κυκλώματος

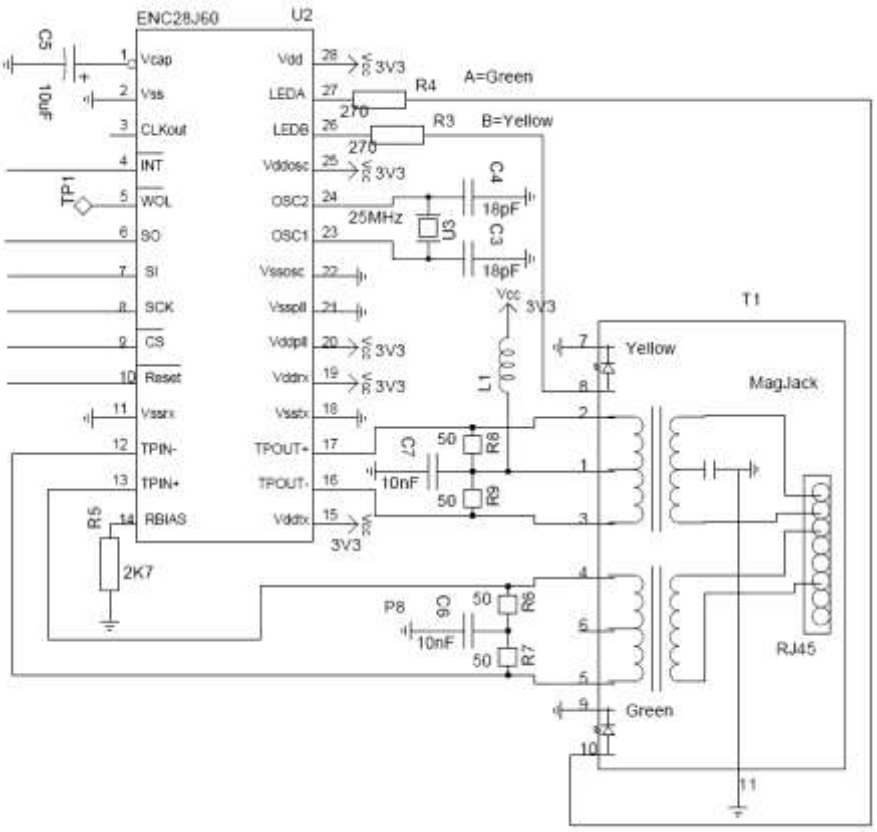
Το κύκλωμα που υλοποιήθηκε αποτελεί βελτίωση του κυκλώματος που προτείνει η εταιρία Microchip στο φύλλο προδιαγραφών του ελεγκτή ENC28J60. Ο μικροελεγκτής ATmega168 συνδέεται με τον Ethernet controller μέσω των γραμμών SCK, MISO, MOSI, INT, CS και RESET. Με βάση τις προδιαγραφές, στην γραμμή MISO απαιτείται αντίσταση 270Ω σε σειρά, ενώ στις γραμμές CS και RESET απαιτείται αντίσταση pull-up 10kΩ. Η αντίσταση pull-up βοηθά να διατηρείται ψηλά το επίπεδο της τάσης του ψηφιακού κυκλώματος όταν η αντίστοιχη ακίδα του μικροελεγκτή είναι απενεργοποιημένη. Η τάση λειτουργίας του κυκλώματος είναι τα 3.3Volt.

1.4.1 Περιφερειακά ENC28J60

Τον ελεγκτή Ethernet πλαισιώνουν τα απαραίτητα ηλεκτρονικά παθητικά στοιχεία (αντιστάσεις και πυκνωτές) όπως παρουσιάζονται στο φύλλο προδιαγραφών. Συγκεκριμένα χρειάζονται:

- Κρύσταλλος ταλάντωσης 25MHz
- Δύο πυκνωτές 18pF για τη σύνδεση του κρυστάλλου ταλάντωσης
- Θύρα RJ45 με ενσωματωμένο κύκλωμα Magnetics και 2 LED
- Δύο αντιστάσεις 270Ω για την οδήγηση των LED
- Δύο ζευγάρια αντιστάσεων 50Ω και δύο πυκνωτές 10nF για τη σύνδεση με το κύκλωμα Magnetics

- Ένα πηνίο 100μH ονομαστικού ρεύματος 80mA για μείωση των ηλεκτρομαγνητικών παρεμβολών
- Μία αντίσταση pull-down 2.7kΩ στη γραμμή RBIAS
- Ένας πυκνωτής 10μF decoupling στη γραμμή Vcap



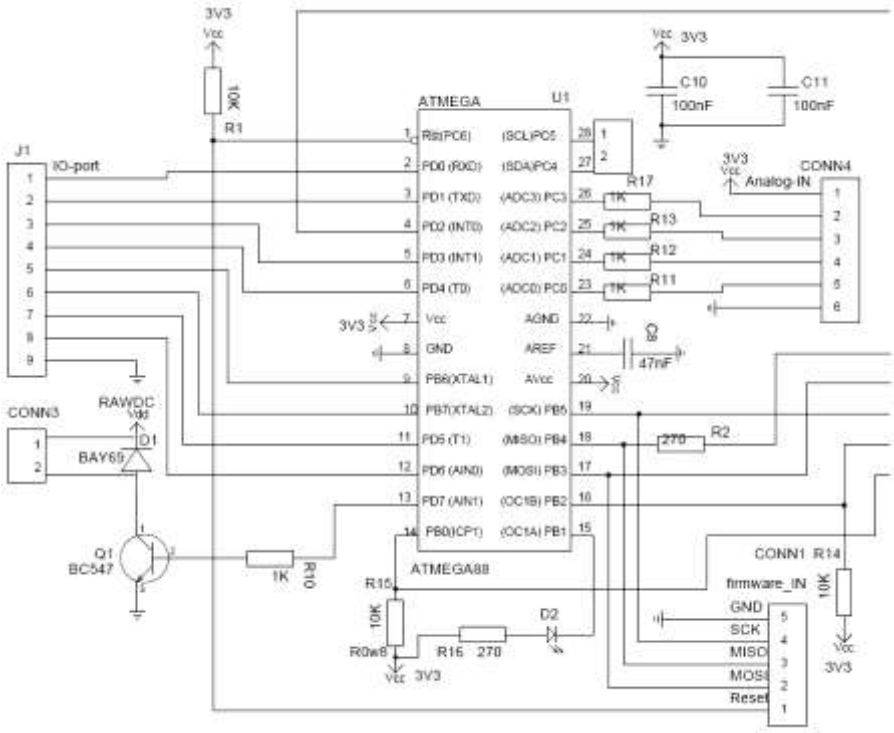
Εικόνα 12: Το κύκλωμα του ελεγκτή Ethernet

1.4.2 Περιφερειακά ATmega168

Τα στοιχεία που συνοδεύουν τον μικροελεγκτή είναι:

- Αντίσταση pull-up 10kΩ στη γραμμή PB0 που εξυπηρετεί και τη γραμμή RESET του ελεγκτή Ethernet
- Αντίσταση pull-up 10kΩ στη γραμμή PB2 που εξυπηρετεί και τη γραμμή CS του ελεγκτή Ethernet
- Πυκνωτής decoupling 47nF στη γραμμή AREF
- Θύρα επέκτασης γενικής χρήσης 9 επαφών
- Θύρα προγραμματισμού SPI και debugWire 6 επαφών
- Θύρα εισόδου αναλογικών σημάτων 6 επαφών (ADC)
- Τέσσερεις αντιστάσεις 1kΩ σε σειρά με τις γραμμές του ADC
- Θύρα επικοινωνίας TWI

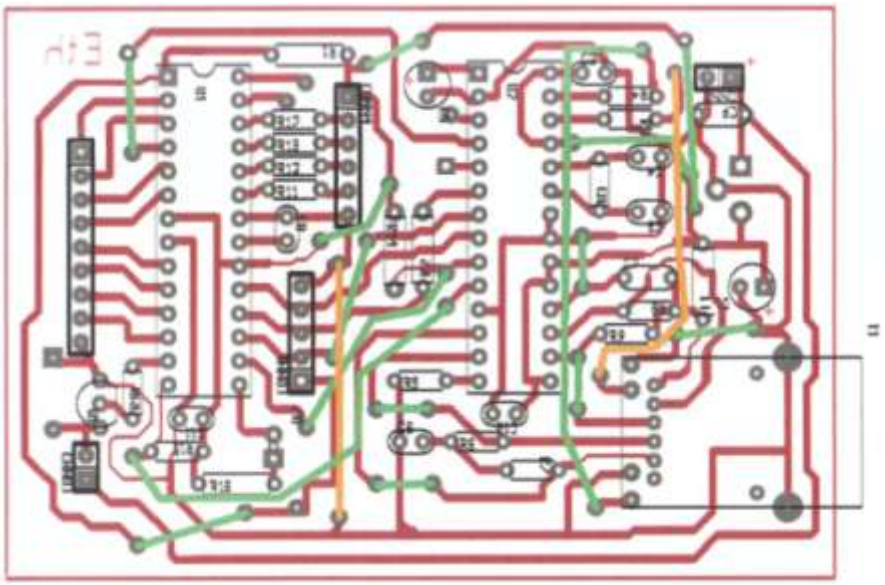
- Ενδεικτικό LED
- Αντίσταση 270Ω οδήγησης του LED
- Θύρα ελέγχου εξωτερικού φορτίου
- Διπολικό τρανζίστορ NPN οδήγησης εξωτερικού φορτίου
- Δίοδος προστασίας του τρανζίστορ από επαγωγικά φορτία
- Αντίσταση 1kΩ οδήγησης του τρανζίστορ
- Θύρα τροφοδοσίας
- Πυκνωτές 100nF και 10μF για τη μείωση του κυματισμού της τάσης τροφοδοσίας



Εικόνα 13: Το κύκλωμα του μικροελεγκτή

1.4.3 Εργαλεία σχεδίασης

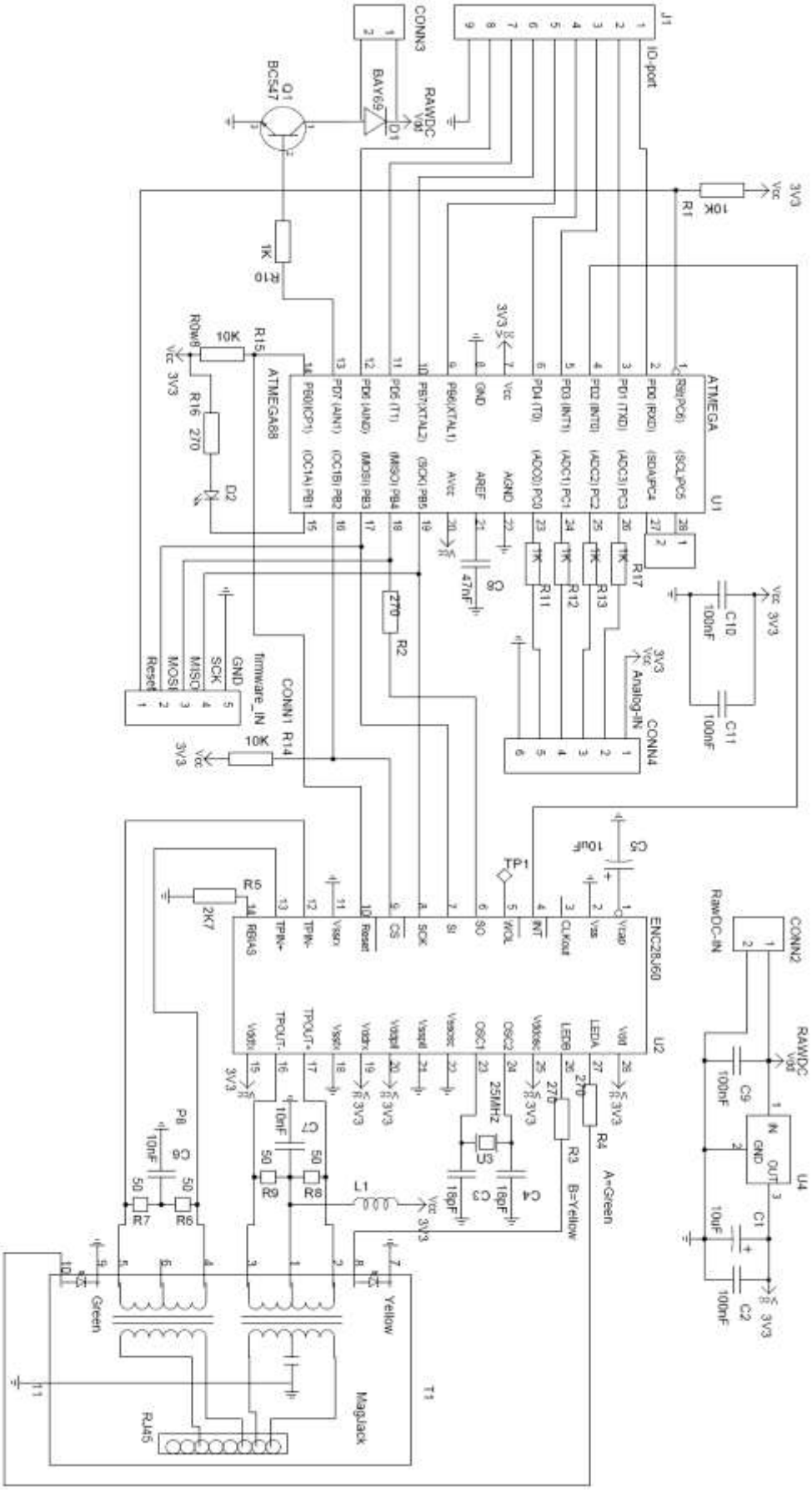
Το κύκλωμα σχεδιάστηκε με τη βοήθεια του σχεδιαστικού πακέτου gEDA που προσφέρει δυνατότητες σύλληψης σχεδίου (schematic capture), εξομοίωσης του κυκλώματος και σχεδιασμού της τελικής πλακέτας με δυνατότητα δρομολόγησης. Στο ακόλουθο σχήμα παρουσιάζεται η τυπωμένη πλακέτα και η θέση του κάθε εξαρτήματος.



Εικόνα 14: Η τυπωμένη πλακέτα του server

Στο παραπάνω σχήμα με κόκκινο (έντονο) χρώμα φαίνονται οι περιοχές του τυπωμένου κυκλώματος ενώ με πράσινο και πορτοκαλί (ασθενές) χρώμα φαίνονται ορισμένα απαραίτητα εξωτερικά καλώδια (jumpers).

Στο επόμενο σχήμα παρουσιάζεται ολοκληρωμένο το κύκλωμα του web server.



Εικόνα 15: Το συνολικό κύκλωμα του server

Κεφάλαιο 2
Power

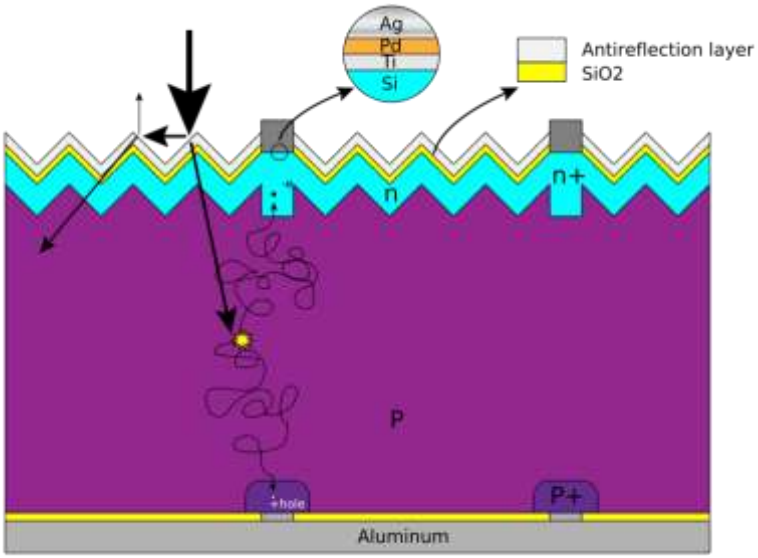
Η τροφοδοσία και η εξοικονόμηση ηλεκτρικής ενέργειας είναι η βασικότερη παράμετρος της παρούσας εργασίας. Η χρήση ηλιακού συλλέκτη (solar panel) σε συνδυασμό με την ασύρματη δικτύωση μας επιτρέπει να τοποθετήσουμε το υπολογιστικό μας σύστημα σε οποιοδήποτε σημείο δίχως την ανάγκη καλωδίων. Γίνεται λοιπόν προφανές ότι η τροφοδοσία του συστήματος παίζει καθοριστικό ρόλο και χρειάζεται ιδιαίτερη προσοχή.

2.1 Ο ηλιακός συλλέκτης

Ο ηλιακός συλλέκτης είναι μια συσκευή σχεδιασμένη να συλλέγει ηλιακή ακτινοβολία και να τη μετατρέπει σε ηλεκτρικό ρεύμα. Η λειτουργία του στηρίζεται στο φωτοβολταϊκό φαινόμενο που ανακαλύφθηκε το 1839 από τον Μπεκερέλ όταν παρατήρησε ότι ένα ηλεκτρολυτικό κελί που εκτίθεται στο ηλιακό φως παράγει ηλεκτρικό ρεύμα [8]. Το πρώτο πραγματικό φωτοβολταϊκό στοιχείο κατασκευάστηκε το 1883 από τον Κάρολο Φρίτς ενώ στις μέρες μας κατασκευάζονται ηλιακοί συλλέκτες με απόδοση έως και 28% [9].

2.1.1 Αρχή λειτουργίας

Ο ηλιακός συλλέκτης κατασκευάζεται από ημιαγώγιμα υλικά, όπως το πυρίτιο. Όταν εκτίθεται στην ηλιακή ακτινοβολία, τα φωτόνια απορροφώνται από τα ημιαγώγιμα υλικά με αποτέλεσμα να συγκρούονται με τα ηλεκτρόνια των ατόμων. Οι συγκρούσεις προκαλούν απελευθέρωση των ηλεκτρονίων, το οποία στη συνέχεια ρέουν μέσα στο υλικό δημιουργώντας ηλεκτρικό ρεύμα. Παράλληλα δημιουργούνται και οπές θετικά φορτισμένες οι οποίες ρέουν προς την αντίθετη κατεύθυνση. Το ηλεκτρικό ρεύμα που παράγεται είναι συνεχές και μπορεί να χρησιμοποιηθεί είτε απευθείας για τροφοδότηση ηλεκτρονικών κυκλωμάτων και φόρτιση μπαταριών, είτε μέσω αντιστροφέα και μετασχηματιστή να μετατραπεί σε εναλλασσόμενο για χρήση σε ηλεκτρικές συσκευές. Στο ακόλουθο σχήμα φαίνεται η τομή ενός ηλιακού συλλέκτη.



Εικόνα 16: Εσωτερική δομή του ηλιακού συλλέκτη

2.1.2 Επιλογή ηλιακού συλλέκτη

Για το σύστημά μας επιλέξαμε έναν ηλιακό συλλέκτη ονομαστικής ισχύος 12Watt από μονοκρυσταλλικό πυρίτιο. Η επιλογή αυτή έγινε με βάση τη θεωρητική κατανάλωση του συστήματος, που ορίστηκε στα 5Watt. Περισσότερες πληροφορίες για την κατανάλωση του συστήματος θα δούμε στο κεφάλαιο 9.

Τα χαρακτηριστικά του ηλιακού συλλέκτη αναλυτικά είναι:

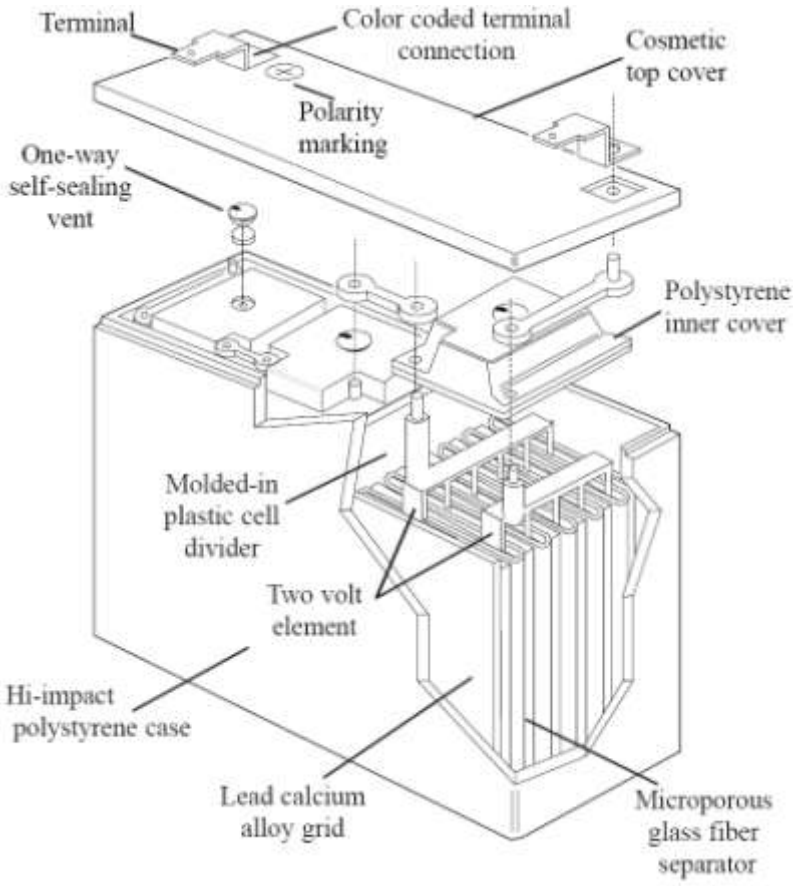
- Τάση ανοικτού κυκλώματος 21Volt
- Τάση λειτουργίας 16.8Volt
- Ρεύμα βραχυκυκλώματος 0.8Amp
- Ρεύμα λειτουργίας 0.7Amp
- Θερμοκρασία λειτουργίας από -40°C έως +80°C
- Θερμοκρασιακός συντελεστής ρεύματος λειτουργίας +0.1%/°C
- Θερμοκρασιακός συντελεστής τάσης λειτουργίας -0.38%/°C
- Αδιάβροχο πλαίσιο από αλουμίνιο και γυαλί υψηλής διαφάνειας

2.2 Η μπαταρία

Το σύστημά μας περιλαμβάνει μια επαναφορτιζόμενη μπαταρία CSB 12Volt 7.2Ah οξέων μολύβδου ξηρού τύπου (sealed lead acid battery). Η επιλογή αυτή έγινε καθώς οι μπαταρίες οξέων μολύβδου έχουν μεγάλη περιεκτικότητα ενέργειας σε ιδιαίτερα χαμηλό κόστος. Αναλυτικά τα χαρακτηριστικά της μπαταρίας είναι [10]:

- Ονομαστικό ενεργειακό περιεχόμενο 86.4Wh
- Αριθμός ηλεκτρολυτικών στοιχείων 6
- Ονομαστική τάση 12Volt
- Βάρος 2.4kg
- Ρεύμα βραχυκύκλωσης 100Amp
- Εσωτερική αντίσταση 23mΩ
- Θερμοκρασία λειτουργίας από -15°C έως +40°C
- Ρυθμός αποφόρτισης 3.4%/μήνα στους 25°C
- Διάρκεια ζωής τουλάχιστον 260 πλήρεις κύκλοι φόρτισης
- Αποδοτικότητα φόρτισης τουλάχιστον 96% στους πρώτους 100 πλήρεις κύκλους
- Τερματική τάση κάθε ηλεκτρολυτικού στοιχείου 1.75Volt
- Μέγιστη τάση κάθε ηλεκτρολυτικού στοιχείου 2.15Volt
- Μέγιστο ρεύμα φόρτισης 2.16Amp

Στο ακόλουθο σχήμα φαίνεται ενδεικτικά το εσωτερικό μας μπαταρίας οξέων μολύβδου ξηρού τύπου.



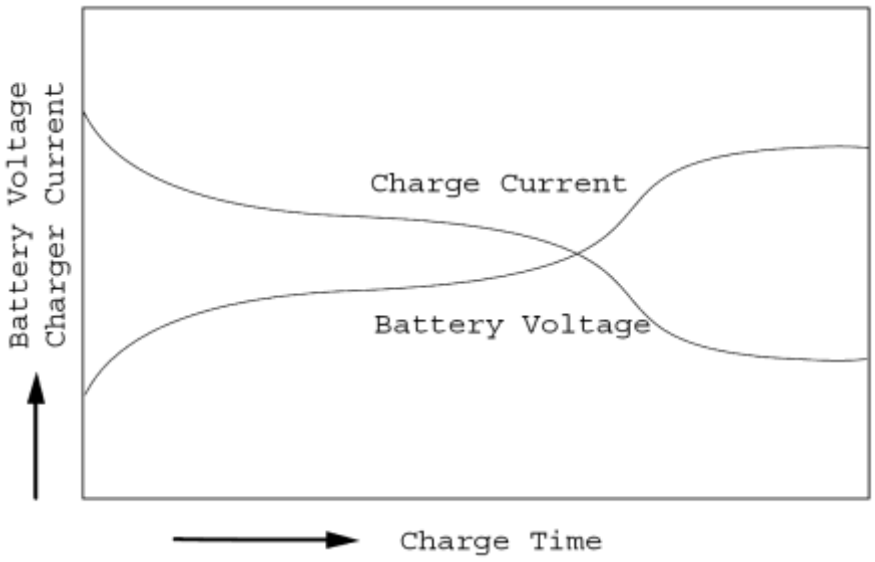
Εικόνα 17: Η μπαταρία τύπου SLA

2.3 Ο ηλιακός φορτιστής

2.3.1 Τύποι φορτιστών

2.3.1.1 Taper charger

Ο πιο απλός τύπος φορτιστή αποτελείται από μία πηγή συνεχούς τάσης, μία δίοδο και μία αντίσταση σε σειρά με την μπαταρία [11]. Η αντίσταση χρειάζεται για να ορισθετεί το ρεύμα φόρτισης κάτω από μία ορισμένη τιμή, ενώ η δίοδος αποτρέπει αντίθετη ροή ρεύματος από την προβλεπόμενη. Κατά τη διάρκεια της φόρτισης η τάση της μπαταρίας αυξάνεται με αποτέλεσμα το ρεύμα φόρτισης να μειώνεται συνεχώς. Όταν η τάση της μπαταρίας περάσει τη μέγιστη επιτρεπόμενη τιμή, η μπαταρία αρχίζει να υπερθερμαίνεται και η εσωτερική της αντίσταση μειώνεται σημαντικά [12] [13]. Όταν διαπιστωθεί υπερθέρμανση ο χρήστης πρέπει να αποσυνδέσει το φορτιστή, διαφορετικά υπάρχει κίνδυνος έκρηξης της μπαταρίας [14]. Ο πιο απλός τρόπος διακοπής της φόρτισης για αυτόν τον τύπο φορτιστή είναι μέσω χρονοδιακόπτη, αλλά είναι ευθύνη του χρήστη να υπολογίσει σωστά τον χρόνο της φόρτισης. Στο ακόλουθο σχήμα φαίνεται ενδεικτικά η καμπύλη ρεύματος φόρτισης και τάσης της μπαταρίας.

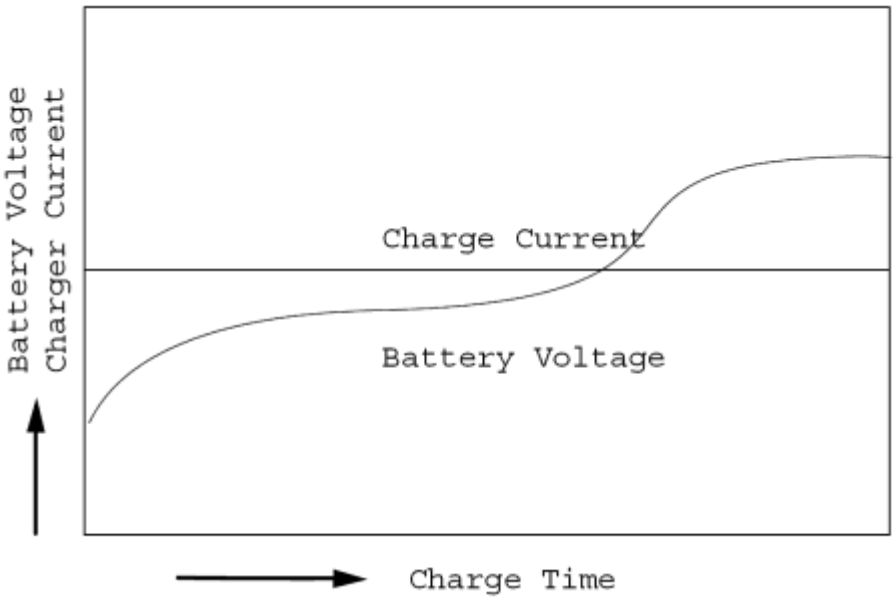


Εικόνα 18: Οι καμπύλες φόρτισης του taper charger

2.3.1.2 Constant current charger

Αυτός ο τύπος φορτιστή διατηρεί σταθερό το ρεύμα φόρτισης μέχρι η τάση της μπαταρίας να φτάσει τη μέγιστη επιτρεπόμενη τιμή, οπότε και διακόπτει τη φόρτιση. Το κύκλωμα ενός τέτοιου φορτιστή στηρίζεται στη χρήση τρανζίστορ και υλοποιεί ένα σύστημα αυτομάτου ελέγχου κλειστού βρόχου. Η χρήση αυτού του είδους

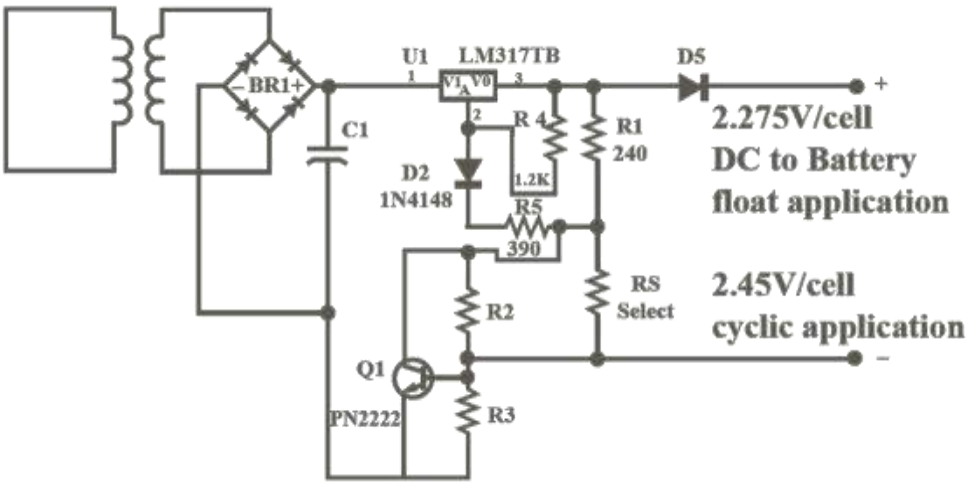
φορτιστή προτείνεται κυρίως όταν είναι γνωστό το ποσοστό αποφόρτισης κάθε κύκλου [11]. Στο ακόλουθο σχήμα φαίνεται ενδεικτικά η καμπύλη ρεύματος φόρτισης και τάσης της μπαταρίας.



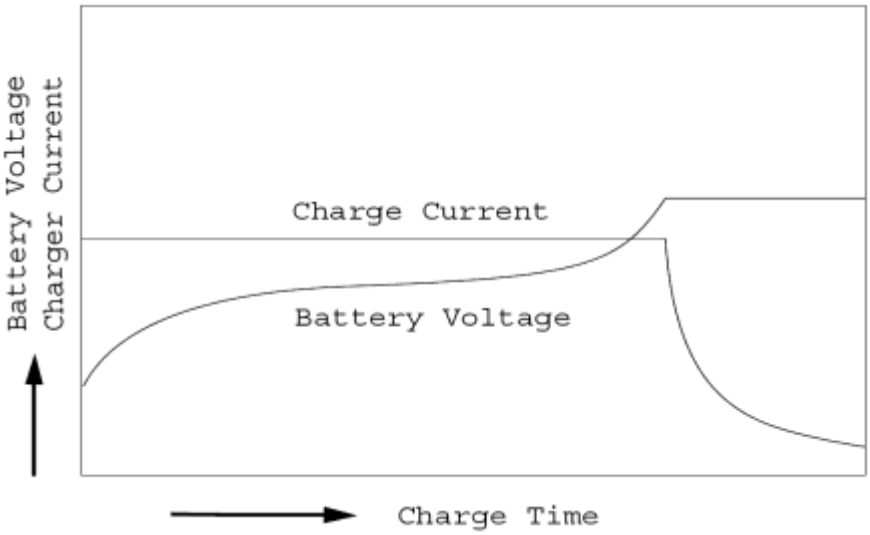
Εικόνα 19: Οι καμπύλες φόρτισης του constant current charger

2.3.1.3 Constant voltage charger

Αυτός ο τύπος φορτιστή διατηρεί σταθερή την τάση σε όλη τη διάρκεια της φόρτισης [11]. Όταν η τάση της μπαταρίας φτάσει μια επιθυμητή τιμή, το ρεύμα φόρτισης μειώνεται και η φόρτιση μεταβαίνει στην κατάσταση συντήρησης. Το κύκλωμα φορτιστή σταθερής τάσης περιλαμβάνει τρανζίστορ και voltage regulators. Στα ακόλουθα σχήματα φαίνεται ένα συνηθισμένο κύκλωμα φορτιστή συνεχούς τάσης, μαζί με τις αντίστοιχες καμπύλες ρεύματος φόρτισης και τάσης της μπαταρίας.



Εικόνα 20: Κυκλωματικό διάγραμμα του constant voltage charger



Εικόνα 21: Οι καμπύλες φόρτισης του constant voltage charger

2.3.2 Επιλογή φορτιστή

Η φόρτιση της μπαταρίας με ηλιακή ενέργεια χρειάζεται ιδιαίτερη προσοχή καθώς εμφανίζονται τα ακόλουθα προβλήματα:

- η παροχή ρεύματος από τον ηλιακό συλλέκτη γενικά δεν είναι σταθερή
- πρέπει να εμποδίζεται η αποφόρτιση της μπαταρίας μέσω του κυκλώματος του ηλιακού συλλέκτη
- η φόρτιση πρέπει να σταματά αυτόματα όταν η τάση της μπαταρίας υπερβεί μια ορισμένη τιμή
- η μπαταρία πρέπει να προστατεύεται από υπερβολική αποφόρτιση

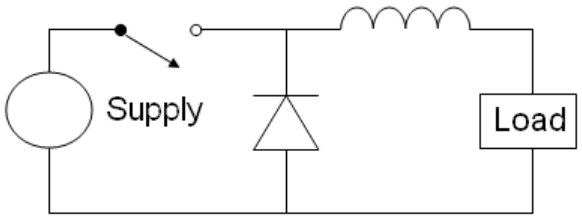
Με βάση τους παραπάνω περιορισμούς, ο ιδανικός τύπος φορτιστή για το σύστημά μας είναι ένας φορτιστής συνεχούς τάσης. Τα χαρακτηριστικά του φορτιστή που χρησιμοποιήσαμε είναι:

- Ονομαστική τάση 12Volt
- Μέγιστο ρεύμα 5Amp
- Ενδεικτικά LED λειτουργιών charging, over-charge και over-discharge
- Θύρα σύνδεσης ηλιακού συλλέκτη
- Θύρα σύνδεσης μπαταρίας
- Θύρα σύνδεσης φορτίου
- Αυτόματη αποσύνδεση φορτίου όταν η τάση της μπαταρίας υποχωρήσει κάτω από 11.0Volt

- Αυτόματη αποσύνδεση του ηλιακού συλλέκτη όταν η τάση της μπαταρίας υπερβεί τα 14.2Volt
- Εμποδίζει την αποφόρτιση της μπαταρίας μέσω του κυκλώματος του ηλιακού συλλέκτη
- Μέγιστη ισχύς ηλιακού συλλέκτη 60Watt

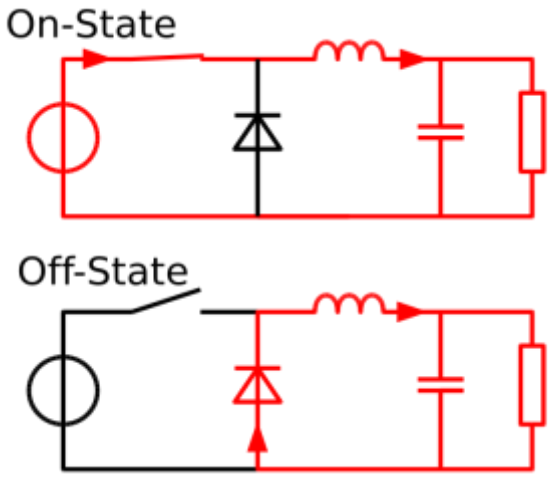
2.4 Step-down converter

Ο step-down converter είναι ένας μετατροπέας υποβιβασμού συνεχούς τάσης. Το βασικό κύκλωμα ενός step-down converter περιλαμβάνει ένα πηνίο, μία δίοδο και ένα διακόπτη [15].



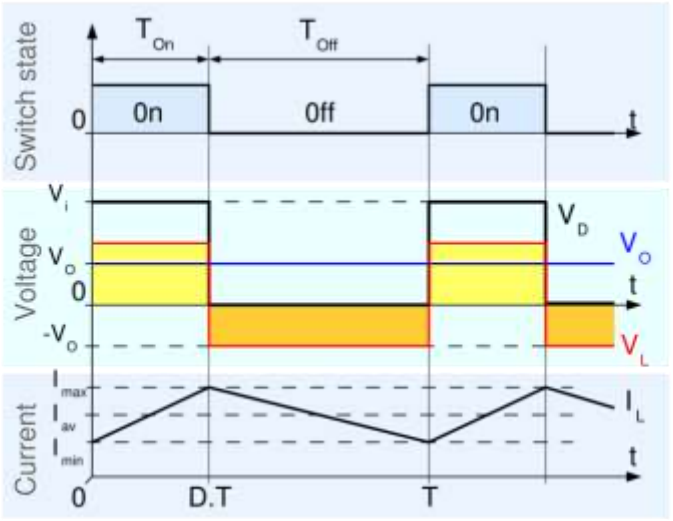
Εικόνα 22: Κυκλωματικό διάγραμμα του step-down converter

Όταν ο διακόπτης είναι κλειστός, η πηγή τροφοδοτεί απευθείας το φορτίο και το πηνίο αποθηκεύει ενέργεια μαγνητικού πεδίου. Όταν ο διακόπτης ανοίξει, η ενέργεια του μαγνητικού πεδίου μετατρέπεται σε ρεύμα επαγωγής, το οποίο τροφοδοτεί το φορτίο. Ανάλογα με τη συχνότητα που ανοίγει και κλείνει ο διακόπτης, μεταβάλλεται και το επίπεδο της τάσης τροφοδοσίας του φορτίου. Η τάση αυτή σταθεροποιείται με την προσθήκη ενός πυκνωτή παράλληλα με το φορτίο.



Εικόνα 23: Οι δύο καταστάσεις λειτουργίας

Στο ακόλουθο σχήμα φαίνεται το διάγραμμα της τάσης και του ρεύματος φορτίου κατά τη διάρκεια μίας περιόδου:



Εικόνα 24: Η εναλλαγή των καταστάσεων σε μία περίοδο

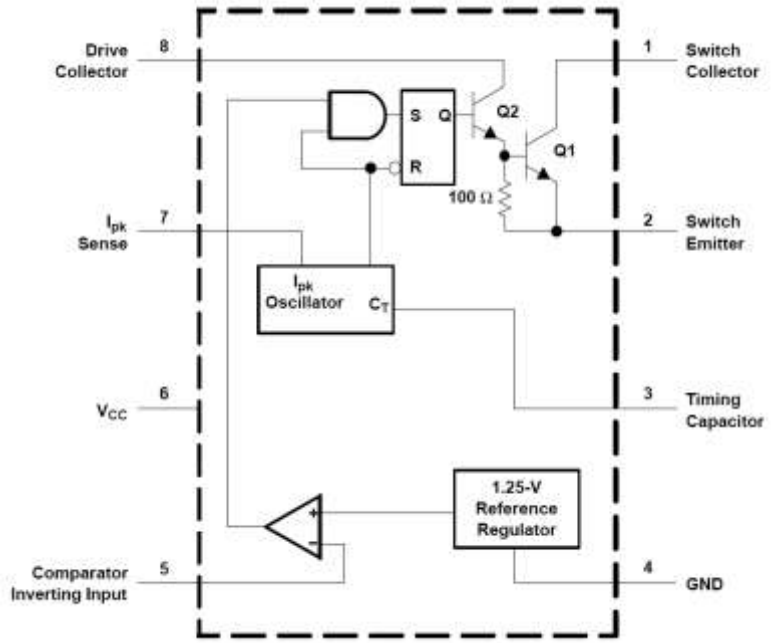
2.4.1 Το ολοκληρωμένο MC34063A

Η υλοποίηση του διακόπτη ενός step-down converter μπορεί να γίνει μέσω ολοκληρωμένου κυκλώματος. Το MC34063A της Texas Instruments είναι ένα τέτοιο ολοκληρωμένο που ελέγχει την κατάσταση ενός εσωτερικού διακόπτη ανάλογα με το επίπεδο μίας τάσης αναφοράς. Πιο συγκεκριμένα το ολοκληρωμένο περιλαμβάνει [16]:

- Τάση αναφοράς 1.25Volt ανεξάρτητη από τη θερμοκρασία
- Ένα τελεστικό ενισχυτή σε συνδεσμολογία συγκριτή
- Ένα ταλαντωτή συχνοτήτων τάξεως 10^4 Hz
- Ένα flip-flop που υλοποιεί ελεγκτή PWM
- Δύο τρανζίστορ σε συνδεσμολογία Darlington που υλοποιούν ένα διακόπτη ισχύος για ρεύμα έως 1.5Amp

Το ολοκληρωμένο MC34063A μπορεί να διατηρεί την τάση εξόδου σταθερή και ανεξάρτητη από το φορτίο ή την τάση εισόδου, αρκεί η αποδιδόμενη ισχύς εισόδου να είναι μεγαλύτερη από την ισχύ εξόδου. Στο εσωτερικό του ολοκληρωμένου υλοποιείται ένα σύστημα αυτομάτου ελέγχου κλειστού βρόχου, το οποίο μεταβάλλοντας κατάλληλα το duty cycle του ελεγκτή PWM εξασφαλίζει τη σταθερότητα της τάσης εξόδου.

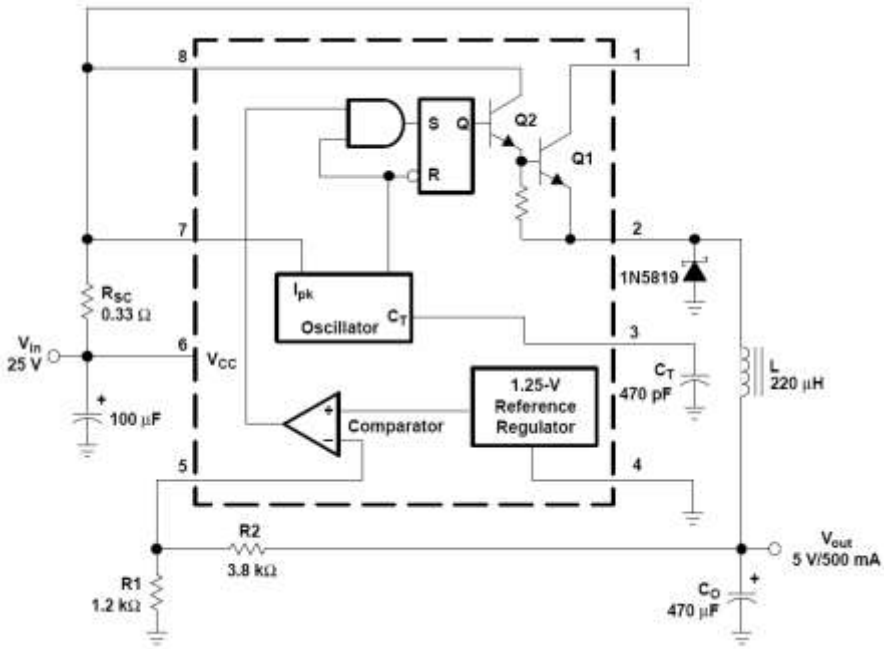
Στο ακόλουθο σχήμα φαίνεται το μπλοκ διάγραμμα του ολοκληρωμένου MC34063A.



Εικόνα 25: Η εσωτερική οργάνωση του ολοκληρωμένου MC34063A

2.4.2 Το κύκλωμα υποβιβασμού τάσης

Για την υλοποίηση του step-down converter θα χρησιμοποιήσουμε το ολοκληρωμένο MC34063A. Η συνδεσμολογία που προτείνεται στο φύλλο προδιαγραφών του ολοκληρωμένου, παρουσιάζεται στο ακόλουθο σχήμα [16]:



Εικόνα 26: Κυκλωματικό διάγραμμα υποβιβασμού τάσης

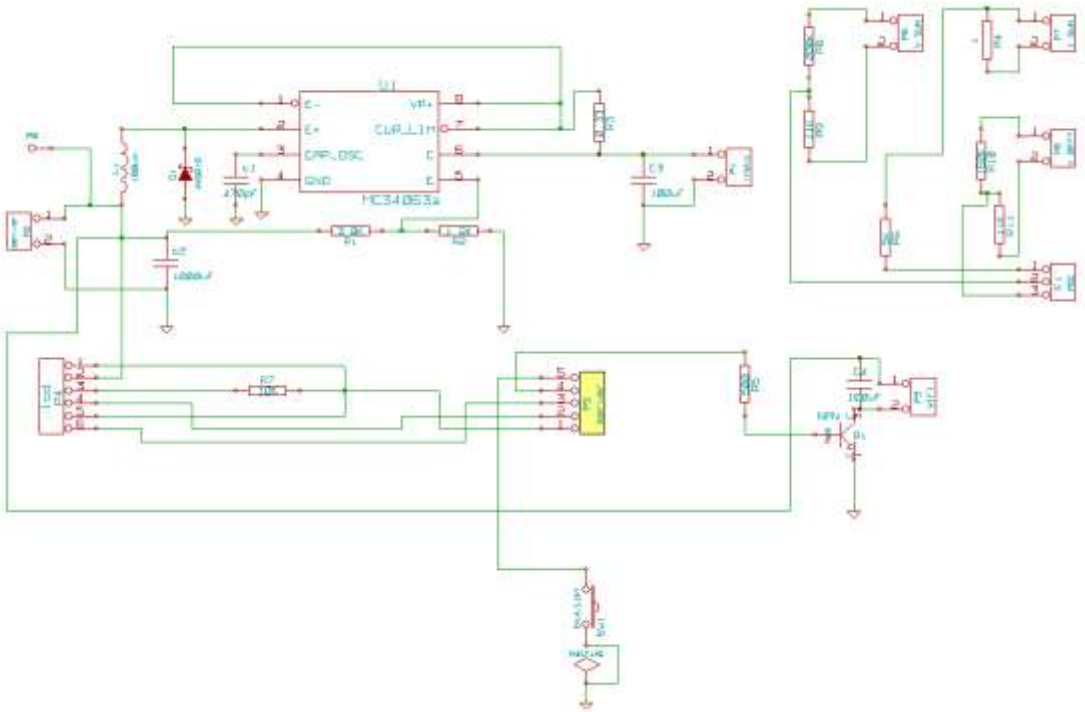
Η παραπάνω συνδεσμολογία φέρει τα ακόλουθα χαρακτηριστικά [16]:

- Για τάση εισόδου από 15Volt έως 25Volt και ρεύμα φορτίου 0.5Amp, η μεταβολή της τάσης εξόδου είναι 12mV
- Για τάση εισόδου 25Volt και ρεύμα φορτίου από 50mA έως 500mA, η μεταβολή της τάσης εξόδου είναι 3mV
- Για τάση εισόδου 25Volt και ρεύμα φορτίου 500mA, η κυμάτωση της τάσης εξόδου είναι 120mV peek-to-peek
- Το ρεύμα βραχυκύκλωσης της εξόδου είναι 1.1Amp
- Η απόδοση ενέργειας του μετατροπέα είναι 83.7%

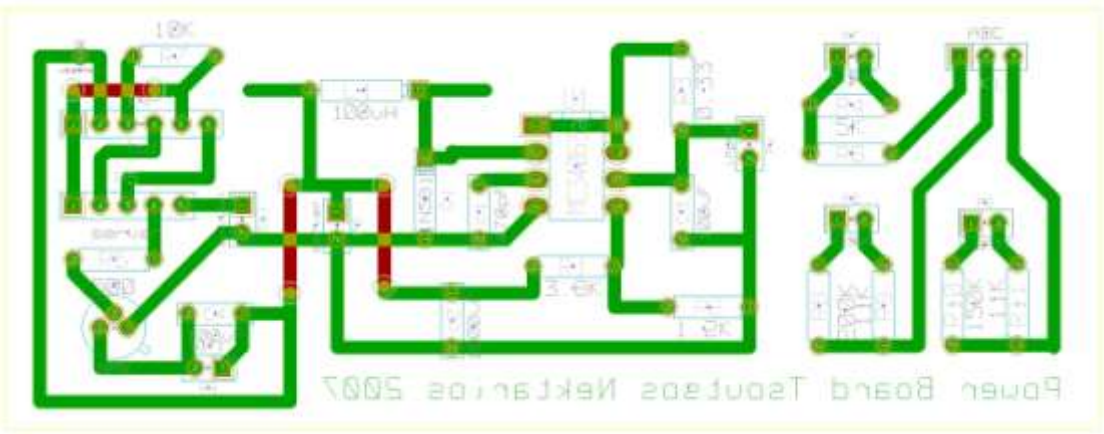
Στο σύστημα που υλοποιήσαμε η τάση της μπαταρίας υποβιβάζεται μέσω δύο DC-to-DC converters στα 5Volt και στα 3.3Volt αντίστοιχα. Η σχεδίαση των δύο αυτών κυκλωμάτων έγινε με το σχεδιαστικό πακέτο kicad που υποστηρίζει σύλληψη σχεδίου και δρομολόγηση της τελικής πλακέτας.

2.4.2.1 Κύκλωμα υποβιβασμού τάσης 5Volt

Για λόγους οικονομίας σχεδίασης η πλακέτα υποβιβασμού τάσης στα 5Volt περιλαμβάνει ένα κύκλωμα αισθητήρων και ένα βοηθητικό κύκλωμα της οθόνης LCD, τα οποία θα μελετήσουμε αναλυτικά στο 3ο κεφάλαιο. Στα σχήματα που ακολουθούν φαίνεται το σχέδιο του κυκλώματος και η τυπωμένη πλακέτα.



Εικόνα 27: Το κύκλωμα υποβιβασμού στα 5 Volt

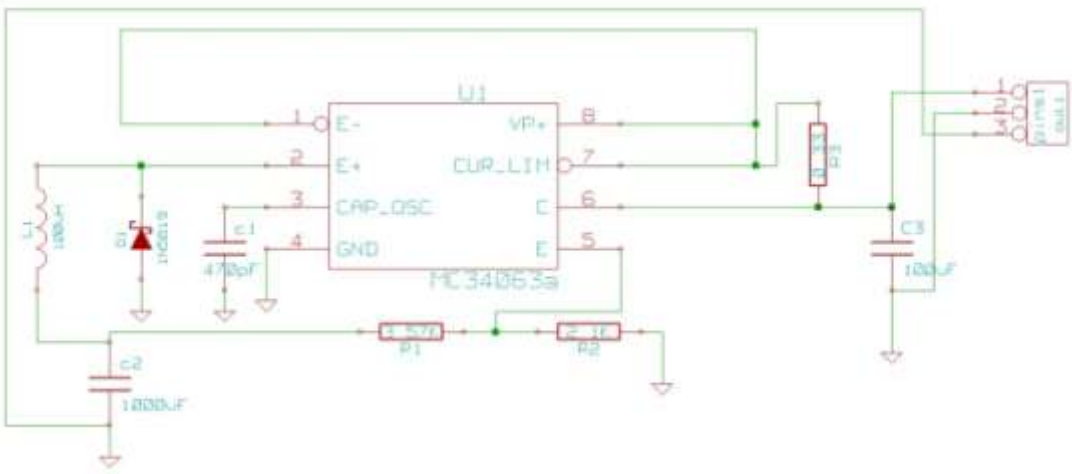


Εικόνα 28: Η τυπωμένη πλακέτα υποβιβασμού στα 5 Volt

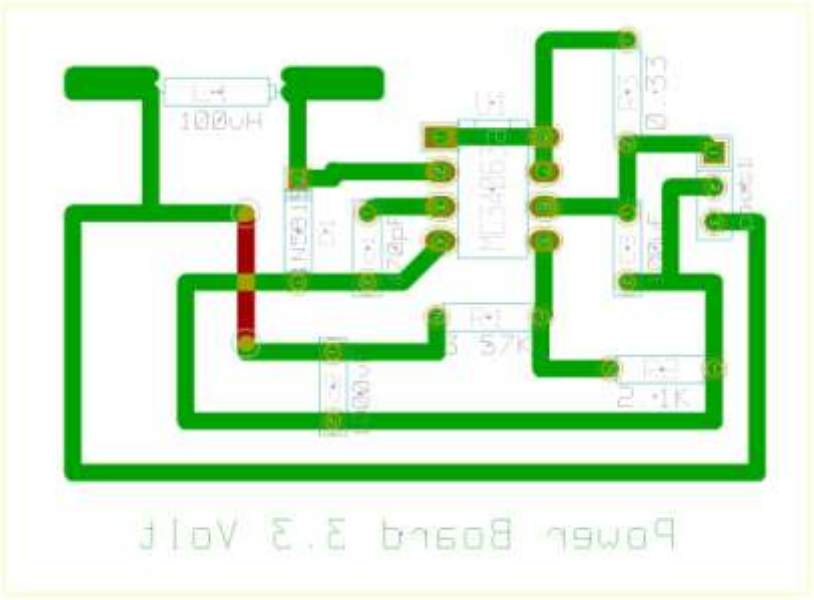
Στο παραπάνω σχήμα με πράσινο (ασθενές) χρώμα φαίνονται οι περιοχές του τυπωμένου κυκλώματος ενώ με κόκκινο (έντονο) χρώμα συμβολίζονται τα απαραίτητα εξωτερικά καλώδια (jumpers). Στο ίδιο σχήμα παρουσιάζεται και η θέση του κάθε ηλεκτρονικού εξαρτήματος. Ο υπολογισμός της τελικής τάσης εξόδου γίνεται με βάση τις αντιστάσεις R1 και R2 από τη σχέση V_{out} = 1.25·(1 + R1/R2) [16]. Έτσι επιλέγουμε αντιστάσεις R1 = 3.8kΩ και R2 = 1.2kΩ. Παρατηρούμε επίσης ότι η τάση εξόδου είναι ανεξάρτητη της τάσης εισόδου. Για τη μείωση της κυμάτωσης στην τάση εξόδου χρησιμοποιούμε πυκνωτή C2 = 1000μF και πηνίο L1 = 100μH ονομαστικού ρεύματος 1.8Amp.

2.4.2.2 Κύκλωμα υποβιβασμού τάσης 3.3Volt

Για να πετύχουμε υποβιβασμό της τάσης στα 3.3Volt ακολουθούμε τη σχέση V_{out} = 1.25·(1 + R1/R2) [16] και επιλέγουμε αντίσταση R1 = 3.57kΩ και R2 = 2.1kΩ. Το σχέδιο του κυκλώματος και η τυπωμένη πλακέτα παρουσιάζονται στα ακόλουθα σχήματα.



Εικόνα 29: Το κύκλωμα υποβιβασμού στα 3.3 Volt



Εικόνα 30: Η τυπωμένη πλακέτα υποβιβασμού στα 3.3 Volt

Στο παραπάνω σχήμα με πράσινο (ασθενές) χρώμα φαίνονται οι περιοχές του τυπωμένου κυκλώματος, με κόκκινο (έντονο) χρώμα συμβολίζεται ένα απαραίτητο εξωτερικό καλώδιο (jumper), ενώ παρουσιάζεται και η θέση του κάθε ηλεκτρονικού εξαρτήματος. Για τη μείωση της κυμάτωσης στην τάση εξόδου χρησιμοποιούμε πυκνωτή C2 = 1000μF και πηνίο L1 = 100μH ονομαστικού ρεύματος 1.8Amp.

Κεφάλαιο 3
Data

Η συλλογή δεδομένων αποτελεί ένα αρκετά σημαντικό κομμάτι της εφαρμογής μας και είναι άμεσα συνδεδεμένη με τη μετατροπή αναλογικών σημάτων σε ψηφιακά. Η μετατροπή αυτή γίνεται από ειδικό hardware που ονομάζεται Analog-to-Digital Converter (ADC) και μπορεί είτε να βρίσκεται ενσωματωμένο σε ένα μικροελεγκτή είτε να υπάρχει σαν ξεχωριστή συσκευή που επικοινωνεί μέσω μιας γραμμής δεδομένων. Τα δεδομένα λαμβάνονται από κατάλληλους αισθητήρες ανάλογα με την εφαρμογή, ενώ η παρουσίασή τους μπορεί να γίνει μέσω μιας συνηθισμένης οθόνης LCD.

3.1 Analog-to-Digital Converter

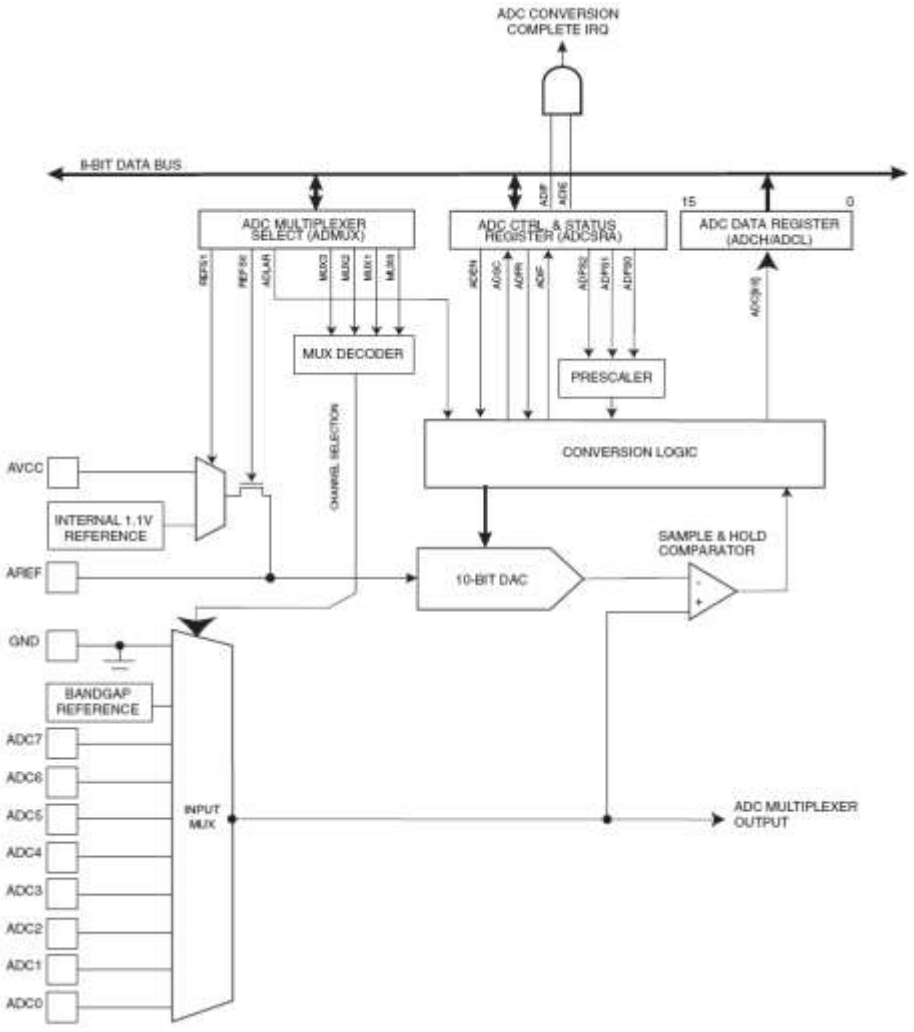
Ένα πολύ βασικό χαρακτηριστικό του ADC είναι η ανάλυση (resolution) που προσφέρει. Η ανάλυση εκφράζει το πλήθος των διακριτών τιμών που μπορούν να αντιστοιχιστούν σε ένα συνεχές εύρος τιμών. Οι τιμές αυτές συνήθως αποθηκεύονται σε δυαδική μορφή με αποτέλεσμα η ανάλυση να εκφράζεται σε αριθμό bits. Σε αυτή την περίπτωση το πλήθος των διαφορετικών τιμών ή επιπέδων (levels) του ADC είναι ίσο με μία δύναμη του δύο. Για παράδειγμα αν η ανάλυση του ADC είναι 8 bits, τότε τα επίπεδα είναι 256, ενώ ανάλογα με την εφαρμογή μπορούν να κυμαίνονται από 0 έως 255 ή από -128 έως 127.

Ανάλογα με την εφαρμογή, ένας ADC μπορεί να είναι γραμμικός ή μη γραμμικός. Στους γραμμικούς ADC η διαφορά μεταξύ δύο οποιωνδήποτε διαδοχικών επιπέδων, αντιστοιχεί πάντα σε σταθερή διαφορά στο πεδίο των αναλογικών τιμών. Έτσι αν η είσοδος ενός γραμμικού ADC είναι ηλεκτρική τάση, τότε η ανάλυση μπορεί να εκφραστεί σε Volt ανά επίπεδο από τη σχέση (Volts per level) = (Analog Range) / ($2^{\text{resolution bits}}$). Αντίθετα στους μη γραμμικούς ADC η κατανομή των επιπέδων δεν είναι συμμετρική σε όλο το εύρος αναλογικών τιμών, αλλά μεταβάλλεται ανάλογα με τα χαρακτηριστικά και τη σημασία κάθε περιοχής. Έτσι σε περιοχές ενδιαφέροντος, ένας μη γραμμικός ADC μπορεί να πετύχει την ίδια ανάλυση με ένα γραμμικό ADC περισσότερων bits.

3.2 Ο ADC του μικροελεγκτή ATmega168

Ο μικροελεγκτής ATmega168 περιλαμβάνει έναν Analog-to-Digital Converter ανάλυσης 10bit. Ο ADC συνδέεται με αναλογικό πολυπλέκτη 8 καναλιών και επιτρέπει τη μετατροπή μέχρι 8 αναλογικών τάσεων. Η είσοδος των τάσεων στη συσκευασία PDIP γίνεται από τις ακίδες 0 έως 5 της θύρας C, με αναφορά τα 0Volt. Αναλυτικά τα χαρακτηριστικά του ADC είναι [5]:

- Ανάλυση 10bit
- Μέγιστο απόλυτο σφάλμα 2 least significant bits
- Χρόνος μετατροπής από 13μsec έως 260μsec
- Μέγιστη χρονική ανάλυση 15000 δείγματα ανά δευτερόλεπτο
- Έως 8 κανάλια εισόδου ενός ακροδέκτη
- Περιοχή τιμών από 0Volt έως 1.1Volt
- Λειτουργία μείωσης θορύβου



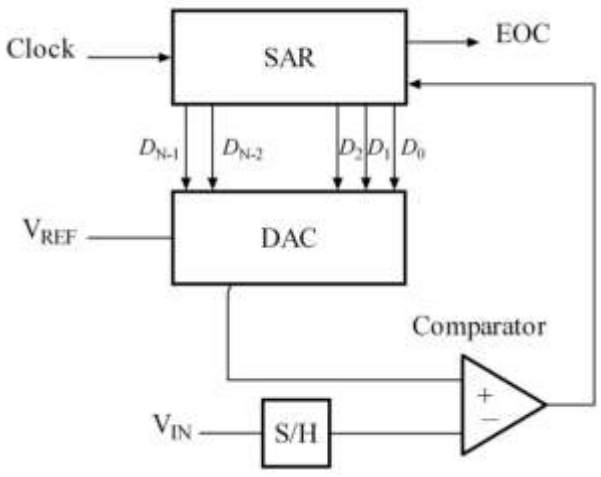
Εικόνα 31: Εσωτερική οργάνωση του ADC

3.2.1 Αρχή λειτουργίας

Η μετατροπή της τάσης εισόδου γίνεται με τη μέθοδο της διαδοχικής προσέγγισης (successive approximation). Η μέθοδος αυτή εντοπίζει την ψηφιακή αναπαράσταση μιας αναλογικής τάσης μέσω δυαδικής αναζήτησης (binary search) ανάμεσα σε όλες τις πιθανές αναπαραστάσεις [17] [18]. Η υλοποίηση γίνεται με τέσσερα βασικά κυκλώματα [5]:

- Ένα κύκλωμα “Sample and Hold” που εξασφαλίζει ότι η τάση εισόδου διατηρείται σταθερή κατά τη διάρκεια της μετατροπής.
- Ένα τελεστικό ενισχυτή σε συνδεσμολογία συγκριτή, ο οποίος συγκρίνει την τάση εισόδου (V_{in}) με την έξοδο ενός εσωτερικού Digital-to-Analog Converter (DAC) και στέλνει το αποτέλεσμα σε ένα ειδικό καταχωρητή διαδοχικής προσέγγισης (SAR)
- Ένα κύκλωμα που στέλνει στον DAC την νέα τιμή του καταχωρητή SAR
- Ένα κύκλωμα που τροφοδοτεί τον DAC με σταθερή τάση αναφοράς (V_{ref})

Όλα τα bit του καταχωρητή SAR αρχικοποιούνται στην τιμή 0 με εξαίρεση το most significant bit που αρχικοποιείται στην τιμή 1. Η τιμή του καταχωρητή SAR αποστέλλεται στον DAC, ο οποίος στέλνει την τάση V_{ref} / 2 στον τελεστικό ενισχυτή για να συγκριθεί με την τάση εισόδου V_{in}. Αν η τάση του DAC υπερβαίνει την τάση V_{in}, τότε το most significant bit του καταχωρητή SAR μηδενίζεται, διαφορετικά παραμένει στην τιμή 1. Η αναζήτηση προχωρά με τον ίδιο τρόπο σε κάθε επόμενο bit, το οποίο τίθεται αρχικά στην τιμή 1. Όταν γίνει η σάρωση ολόκληρου του καταχωρητή SAR, εξάγεται το αποτέλεσμα. Στο ακόλουθο σχήμα φαίνεται το μπλοκ διάγραμμα της μεθόδου μετατροπής.



Εικόνα 32: Block διάγραμμα της μετατροπής

3.3 Αισθητήρες

Το σύστημά μας περιλαμβάνει τους εξής αισθητήρες:

- Αναλογικός αισθητήρας ατμοσφαιρικής πίεσης
- Αναλογικός αισθητήρας ρεύματος ηλιακού συλλέκτη
- Αναλογικός αισθητήρας τάσης μπαταρίας
- Αναλογικός αισθητήρας τάσης ηλιακού συλλέκτη
- Ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας

Όλοι οι παραπάνω αναλογικοί αισθητήρες συνδέονται στον ADC του μικροελεγκτή ATmega168, ενώ ο ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας περιλαμβάνει το δικό του ADC και επικοινωνεί με τον μικροελεγκτή μέσω Two Wire Interface.

3.3.1 Αισθητήρας ατμοσφαιρικής πίεσης

Για τη μέτρηση της ατμοσφαιρικής πίεσης χρησιμοποιούμε τον αισθητήρα MPX4115A της Freescale. Ο αισθητήρας αυτός λειτουργεί με τροφοδοσία 5Volt και η έξοδος του κοντά στο επίπεδο της θάλασσας κυμαίνεται μεταξύ 3.6 και 4.15Volt [19]. Όμως ο ADC του μικροελεγκτή ATmega168 δέχεται είσοδο τάσης από 0 έως 1.1Volt, με αποτέλεσμα να χρειάζεται ο μετασχηματισμός της εξόδου του αισθητήρα σε αυτά τα επίπεδα. Για το σκοπό αυτό χρησιμοποιούμε ένα τελεστικό ενισχυτή σε συνδεσμολογία διαφορικού ενισχυτή με κέρδος 2. Η μία είσοδος του διαφορικού ενισχυτή είναι τάση αναφοράς 3.6Volt, ενώ στην άλλη είσοδο συνδέεται η έξοδος του αισθητήρα πίεσης. Στην έξοδο του διαφορικού ενισχυτή εμφανίζεται τάση από 0 έως 1.1Volt που μπορεί πλέον να χρησιμοποιηθεί από τον ADC του μικροελεγκτή.

3.3.2 Αισθητήρας ρεύματος ηλιακού συλλέκτη

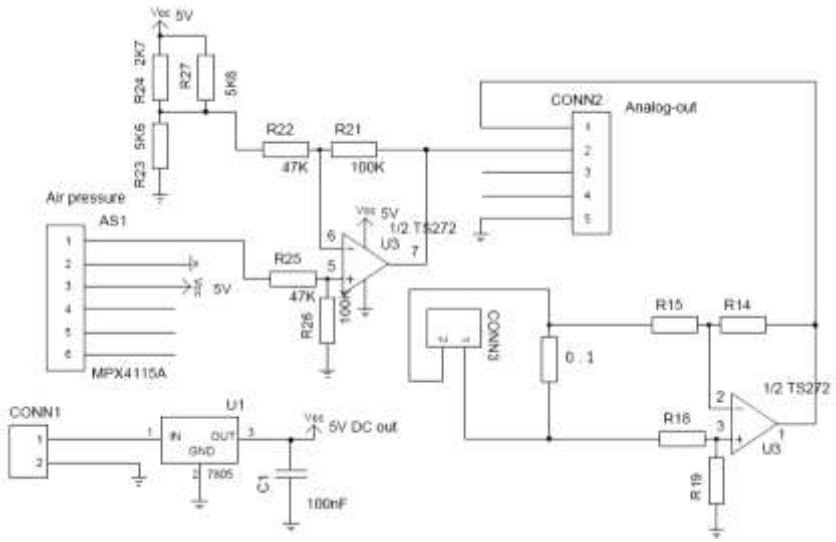
Για τη μέτρηση του ρεύματος του ηλιακού συλλέκτη υπολογίζουμε την πτώση τάσης πάνω σε μία αντίσταση 0.1Ω [20]. Η μέτρηση της πτώσης τάσης γίνεται με τη βοήθεια ενός τελεστικού ενισχυτή σε συνδεσμολογία διαφορικού ενισχυτή με κέρδος 10. Έτσι για μέγιστο ρεύμα 1Amp, η πτώση τάσης στην αντίσταση είναι 100mVolt και η έξοδος του τελεστικού ενισχυτή είναι 1Volt, που μπορεί να χρησιμοποιηθεί απευθείας από τον ADC του μικροελεγκτή.

3.3.3 Το κύκλωμα των αισθητήρων πίεσης και ρεύματος

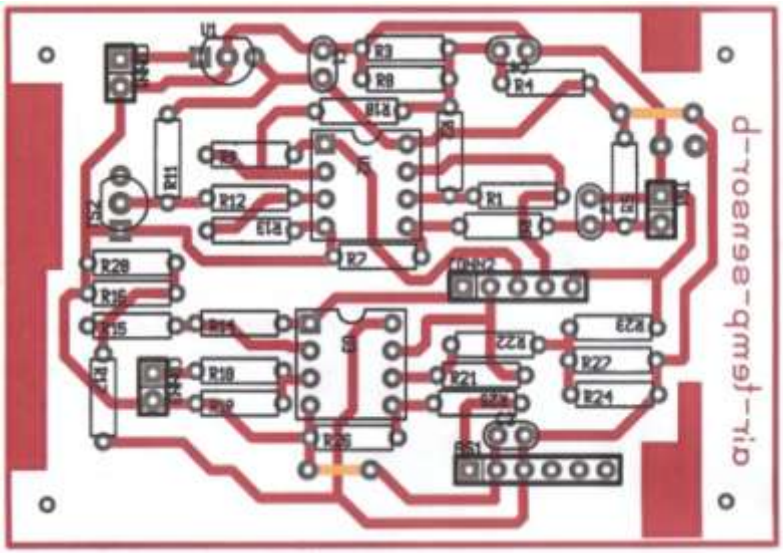
Στο κύκλωμα των αισθητήρων χρησιμοποιούμε το ολοκληρωμένο TS272 που περιλαμβάνει δύο τελεστικούς ενισχυτές. Για τη συνδεσμολογία του πρώτου διαφορικού ενισχυτή χρησιμοποιήσαμε ζεύγη αντιστάσεων 47kΩ και 100kΩ, ενώ η τάση

αναφοράς των 3.6Volt δίνεται από διαίρεση τάσης με τη βοήθεια δύο αντιστάσεων 5.6kΩ και μίας αντίστασης 2.7kΩ. Για τη συνδεσμολογία του δεύτερου διαφορικού ενισχυτή χρησιμοποιήσαμε ζεύγη αντιστάσεων 2.2kΩ και 21kΩ. Η επικοινωνία με την μικροελεγκτή γίνεται μέσω μίας θύρας 5 επαφών, ενώ η τροφοδοσία του κυκλώματος γίνεται μέσω σταθεροποιητή τάσης 7805 και πυκνωτή 100nF. Ιδιαίτερη προσοχή χρειάζεται στη γείωση ώστε να αποφευχθεί βρόχος γείωσης (ground loop) με το κύκλωμα του μικροελεγκτή [21] [22]. Βρόχος γείωσης μπορεί να προκύψει αν συνδέσουμε την επαφή 5 της θύρας CONN2 με την γείωση του μικροελεγκτή.

Η σχεδίαση του κυκλώματος και η δρομολόγηση της πλακέτας έγινε με το σχεδιαστικό πακέτο gEDA. Στα ακόλουθα σχήματα φαίνεται το σχέδιο του κυκλώματος και η τελική πλακέτα.



Εικόνα 33: Το κύκλωμα των αισθητήρων

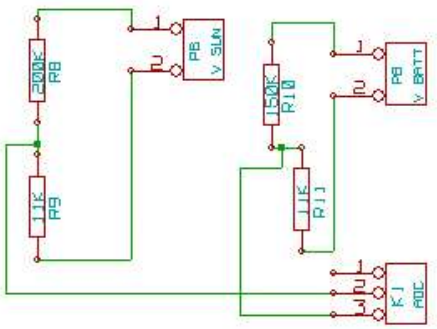


Εικόνα 34: Η τυπωμένη πλακέτα των αισθητήρων

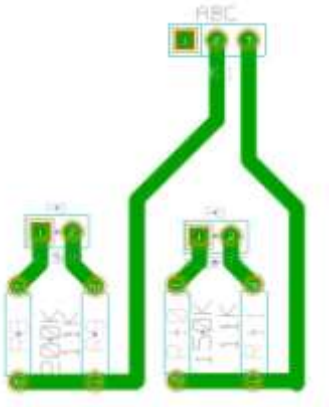
Στο παραπάνω σχήμα με κόκκινο (έντονο) χρώμα φαίνονται οι περιοχές του τυπωμένου κυκλώματος, με πορτοκαλί (ασθενές) χρώμα φαίνονται δύο απαραίτητα εξωτερικά καλώδια (jumpers), ενώ παρουσιάζεται και η θέση κάθε ηλεκτρονικού εξαρτήματος. Το τυπωμένο κύκλωμα περιλαμβάνει καλωδίωση για ένα ακόμα ζεύγος προαιρετικών αισθητήρων.

3.3.4 Το κύκλωμα των αισθητήρων τάσης

Η μέτρηση της τάσης του ηλιακού συλλέκτη και της μπαταρίας γίνεται με τη βοήθεια του ADC του μικροελεγκτή και ενός ειδικού κυκλώματος διαίρεσης τάσης. Η τάση του ηλιακού συλλέκτη έχει μέγιστη τιμή τα 20Volt, και επομένως χρειάζεται διαίρεση κατά 20 φορές πριν συνδεθεί στον ADC, ο οποίος δέχεται μέγιστη είσοδο 1.1Volt. Η διαίρεση γίνεται μέσω αντιστάσεων 220kΩ και 11kΩ. Αντίστοιχα η τάση της μπαταρίας που έχει μέγιστη τιμή τα 15Volt, διαιρείται 15 φορές με τη βοήθεια αντιστάσεων 150kΩ και 11kΩ, πριν συνδεθεί στον ADC. Το κύκλωμα διαίρεσης τάσης βρίσκεται στην ίδια πλακέτα με τον DC-to-DC converter των 5Volt όπως είδαμε στο προηγούμενο κεφάλαιο. Στα σχήματα που ακολουθούν παρουσιάζεται το σχέδιο του κυκλώματος και η αντίστοιχη περιοχή της τυπωμένης πλακέτας.



Εικόνα 35: Το κύκλωμα μέτρησης τάσης



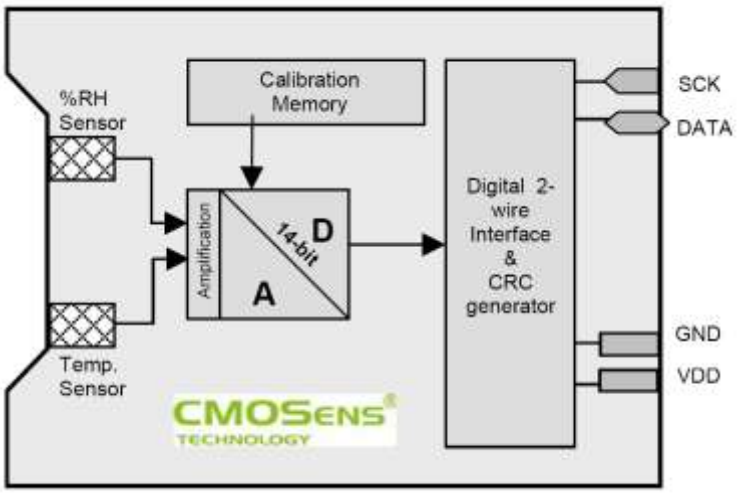
Εικόνα 36: Τμήμα της τυπωμένης πλακέτας

Η σχεδίαση του κυκλώματος και η δρομολόγηση της πλακέτας έγινε με το σχεδιαστικό πακέτο kicad.

3.3.5 Αισθητήρας θερμοκρασίας και υγρασίας

Στο σύστημά μας χρησιμοποιούμε στον ψηφιακό αισθητήρα SHT11 της Sensirion. Ο αισθητήρας αυτός περιλαμβάνει δικό του ADC και η επικοινωνία με τον μικροελεγκτή ATmega168 γίνεται ψηφιακά μέσω σύνδεσης Two Wire Interface από τις ακίδες 5 και 6 της θύρας C. Ο αισθητήρας είναι αδιάβροχος και μας δίνει μετρήσεις της θερμοκρασίας και του ποσοστού υγρασίας. Ο ADC που περιλαμβάνει έχει ανάλυση 14bit, με αποτέλεσμα το μέγιστο απόλυτο σφάλμα μέτρησης να είναι 0.4°C για τη θερμοκρασία και 3% για την υγρασία. Ο αισθητήρας περιλαμβάνει ακόμα θερμαντική αντίσταση που ενεργοποιείται από τον μικροελεγκτή για αποβολή της ανεπιθύμητης υγρασίας [23].

Στον αισθητήρα καταλήγει ένα επίπεδο καλώδιο τεσσάρων γραμμών: στις γραμμές 2 και 3 συνδέεται τάση 0 και 3.3Volt αντίστοιχα, ενώ στις γραμμές 1 και 4 συνδέονται τα σήματα DATA και SCK. Για να εξασφαλίζεται σωστή επικοινωνία και να μην υπάρχει κίνδυνος παρεμβολών, τα καλώδια DATA και SCK του TWI πρέπει να απέχουν τουλάχιστον 1mm [23]. Για το λόγο αυτό φροντίζουμε ώστε οι γραμμές τροφοδοσίας να παρεμβάλλονται των γραμμών δεδομένων σε όλο το μήκος του καλωδίου. Στο ακόλουθο σχήμα φαίνεται το μπλοκ διάγραμμα του αισθητήρα.



Εικόνα 37: Block διάγραμμα του αισθητήρα Sensirion

3.4 Η οθόνη υγρών κρυστάλλων

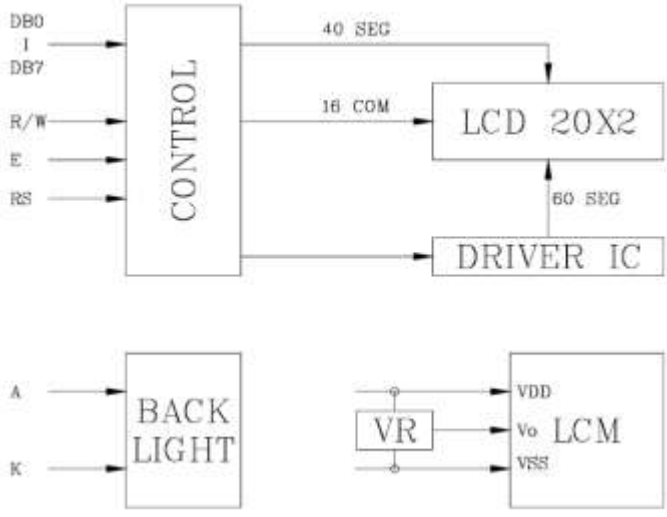
Στο σύστημά μας περιλαμβάνεται και μία οθόνη LCD 2x20 χαρακτήρων με οπίσθιο φωτισμό. Η οθόνη αυτή είναι τύπου HD44780 και έχει τα ακόλουθα χαρακτηριστικά [24]:

- Τάση τροφοδοσίας 5Volt
- Δίαυλος δεδομένων 4bit
- Μέγεθος χαρακτήρα 5x8 pixel
- Πράσινο φόντο, φωτιζόμενο από LED
- Κατανάλωση οθόνης 5mAmp
- Κατανάλωση οπίσθιου φωτισμού 110mAmp

Η οθόνη δέχεται τα ακόλουθα σήματα εισόδου:

- Σήμα ENABLE (E)
- Σήμα READ/WRITE (RW)
- Σήμα REGISTER SELECT (RS)
- Τάση τροφοδοσίας οθόνης 5Volt (VDD)
- Γείωση οθόνης 0Volt (VSS)
- Ρύθμιση αντίθεσης (Vo)
- Τέσσερα σήματα δεδομένων (DB0-DB3)
- Τάση τροφοδοσίας οπίσθιου φωτισμού 5Volt (A)
- Γείωση οπίσθιου φωτισμού 0Volt (K)

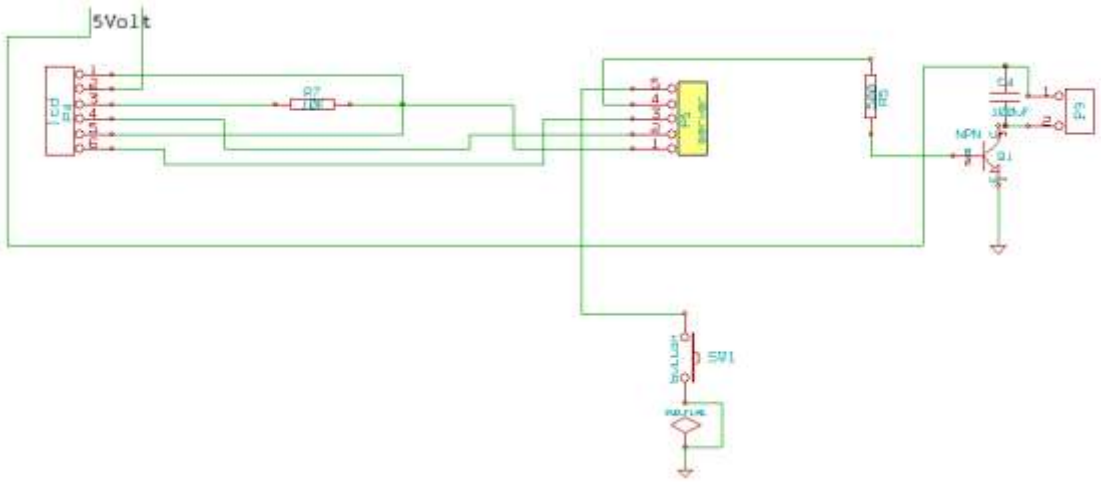
Στο ακόλουθο σχήμα φαίνεται το μπλοκ διάγραμμα της οθόνης.



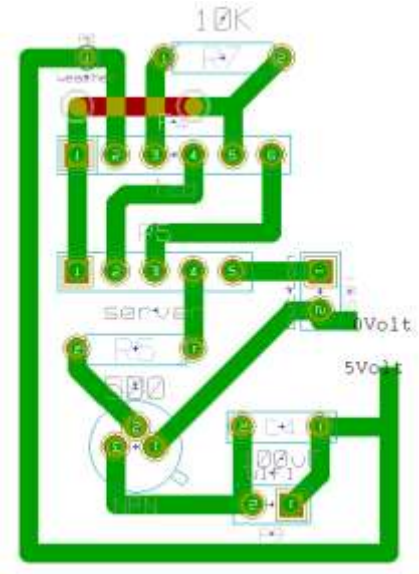
Εικόνα 38: Block διάγραμμα της οθόνης lcd

3.4.1 Η καλωδίωση της οθόνης

Η βοηθητική καλωδίωση της οθόνης βρίσκεται στην ίδια πλακέτα με τον DC-to-DC Converter των 5Volt, όπως είδαμε στο προηγούμενο κεφάλαιο. Η σχεδίαση της καλωδίωσης έγινε με το σχεδιαστικό πακέτο kicad. Στα σχήματα που ακολουθούν φαίνεται το σχέδιο του κυκλώματος και η αντίστοιχη περιοχή της τυπωμένης πλακέτας.



Εικόνα 39: Το κύκλωμα καλωδίωσης της οθόνης



Εικόνα 40: Τμήμα της τυπωμένης πλακέτας

Στο παραπάνω σχήμα με πράσινο (ασθενές) χρώμα φαίνονται οι περιοχές του τυπωμένου κυκλώματος, με κόκκινο (έντονο) χρώμα φαίνεται ένα απαραίτητο εξωτερικό καλώδιο (jumper), ενώ παρουσιάζεται και η θέση κάθε ηλεκτρονικού εξαρτήματος.

3.4.2 Συνδέσεις

Στο παραπάνω σχέδιο καλωδίωσης, οι γραμμές 1 έως 6 της θύρας P4 συνδέονται με τις γραμμές 1 έως 6 της οθόνης. Οι γραμμές 1 έως 5 της θύρας server συνδέονται με τις γραμμές 5 έως 9 της θύρας J1 στην πλακέτα του μικροελεγκτή. Επίσης οι γραμμές 11 έως 14 της οθόνης συνδέονται στις γραμμές 1 έως 4 της θύρας J1 στην πλακέτα του μικροελεγκτή. Οι γραμμές 15 και 16 της οθόνης συνδέονται στη θύρα P9 ενώ στη θύρα SW1 συνδέεται ένα push button που δίνει σήμα στον μικροελεγκτή να ανάψει τον οπίσθιο φωτισμό μέσω ενός του τρανζίστορ NPN. Παρατηρούμε ότι τα σήματα ENABLE και Vo συνδέονται με τη γείωση για λόγους απλότητας της σχεδίασης.

Κεφάλαιο 4
Network

Για την ασύρματη επικοινωνία το σύστημά μας περιλαμβάνει ένα WiFi Access Point (WiFi AP) το οποίο επιτρέπει σε διάφορες κατάλληλες συσκευές (clients) να συνδεθούν σε ένα κοινό δίκτυο. Συνήθως ένα WiFi AP συνδέεται σε κάποιο ενσύρματο δίκτυο και αναμεταδίδει πληροφορίες ανάμεσα στις ασύρματες συσκευές και το ενσύρματο δίκτυο. Πολλά WiFi AP μπορούν να συνδεθούν μεταξύ τους και να προσφέρουν στις συνδεδεμένες συσκευές τη δυνατότητα “roaming”, δηλαδή επέκτασης της συνδεσιμότητας μιας συσκευής πέρα από την περιοχή της αρχικής της εμβέλειας [25]. Όπως είδαμε η συνηθισμένη εμβέλεια ενός WiFi AP είναι 100 μέτρα σε εξωτερικούς χώρους ή 30 μέτρα σε εσωτερικούς χώρους [4]. Επίσης μια συνηθισμένη ταχύτητα μετάδοσης είναι τα 54Mbit ανά sec.

4.1 To Access Point G730

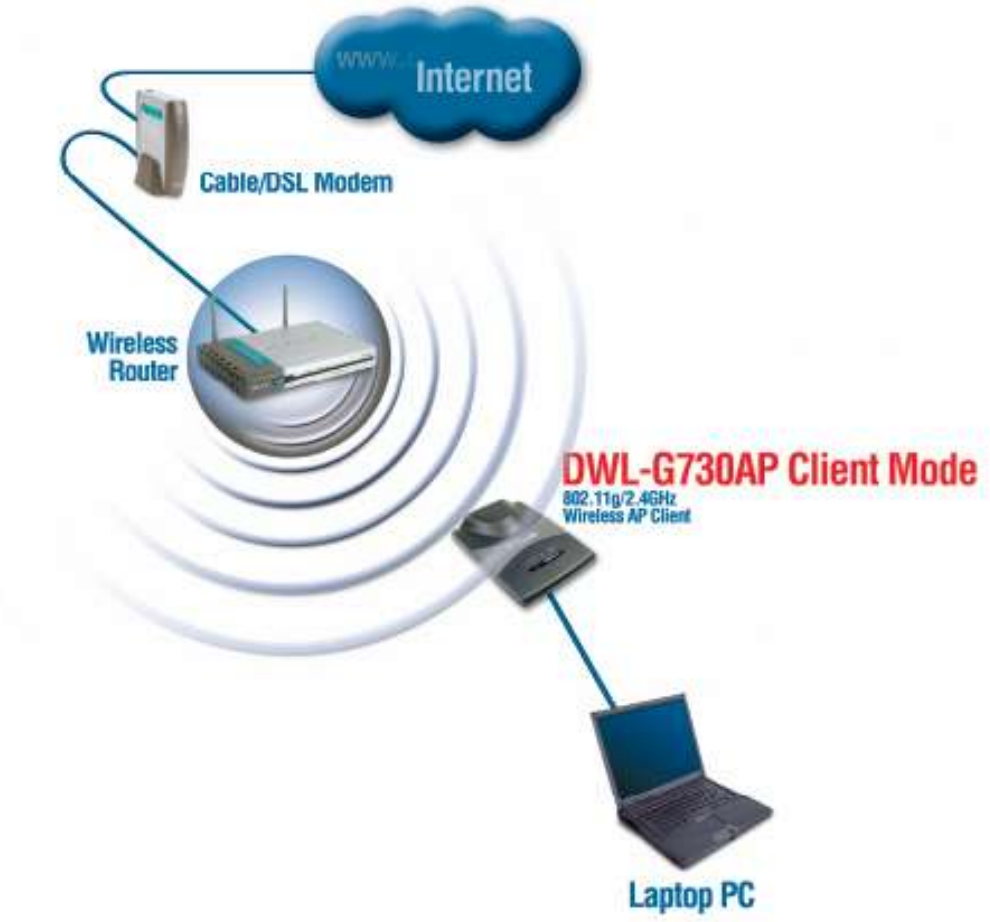
Στο σύστημά μας χρησιμοποιούμε το μοντέλο DWL-G730AP “Wireless Pocket Router” της D-Link, ονομαστικής ταχύτητας 54Mbps. Το μοντέλο αυτό είναι ειδικά σχεδιασμένο για χαμηλή κατανάλωση καθώς προορίζεται για χρήση σε φορητούς υπολογιστές και τροφοδοσία μέσω θύρας USB. Για τη λειτουργία του λοιπόν χρειάζεται τάση 5Volt και ρεύμα τουλάχιστον 0.5Amp [26]. Έτσι η συνολική ισχύς που απορροφά δεν υπερβαίνει τα 2.5Watt, πράγμα που το καθιστά ιδανική επιλογή για ένα σύστημα χαμηλής ισχύος, όπως αυτό που σχεδιάσαμε.

Παρά το μικρό του μέγεθος, το G730 προσφέρει όλες τις δυνατότητες ενός συνηθισμένου Router/Access Point. Εκτός από την εσωτερική κεραία WiFi, διαθέτει μία θύρα Ethernet, ένα κουμπί reset και τρεις ενδεικτικές λυχνίες που σχετίζονται με τη χρήση του ασύρματου δικτύου (WiFi), τη χρήση του ενσύρματου δικτύου (Ethernet) και την τροφοδοσία (Power). Επιπλέον ο χρήστης έχει τη δυνατότητα να επιλέξει μέσω ενός διακόπτη ανάμεσα σε τρεις καταστάσεις λειτουργίας:

- Λειτουργία Client
- Λειτουργία Router
- Λειτουργία Access Point

4.1.1 Η λειτουργία Client

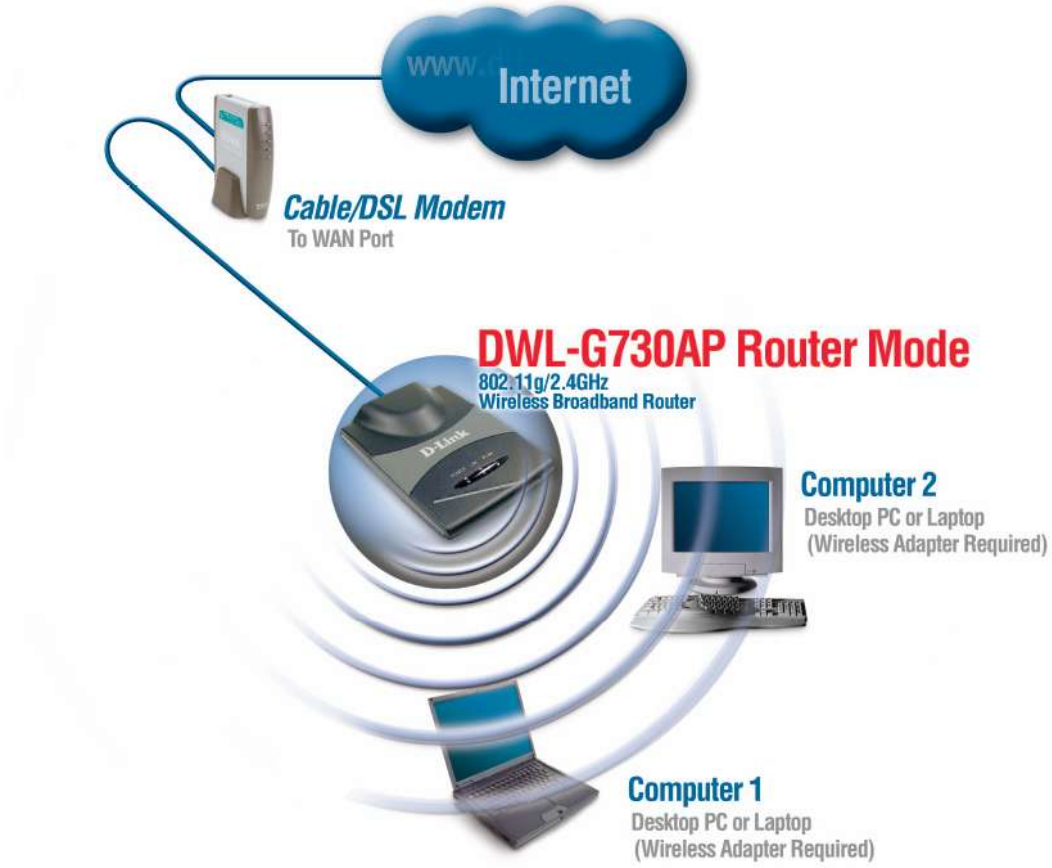
Σε αυτή την κατάσταση λειτουργίας το G730 λειτουργεί σαν διαφανής μετατροπέας των ασύρματων δεδομένων σε ενσύρματα για ένα ασύρματο δίκτυο που έχει ήδη δημιουργηθεί από κάποιο άλλο Access Point. Ένας υπολογιστής που διαθέτει θύρα Ethernet συνδέεται καλωδιακά με το G730, το οποίο στη συνέχεια συνδέεται στο ασύρματο δίκτυο. Έτσι ο υπολογιστής θεωρείται πελάτης (client) του Access Point που δημιούργησε το αρχικό δίκτυο. Για να πραγματοποιηθεί η σύνδεση το μόνο που απαιτείται από το χρήστη είναι να ανοίξει με ένα web browser την αρχική σελίδα του G730, που συνήθως βρίσκεται στη διεύθυνση 192.168.0.30 του τοπικού δικτύου Ethernet, και να ορίσει το όνομα και τον κωδικό του δικτύου στο οποίο σκοπεύει να συνδεθεί. Από την ίδια σελίδα ο χρήστης μπορεί να επέμβει και σε πιο προχωρημένες ρυθμίσεις, όπως για παράδειγμα η ισχύς της κεραίας, ο κατακερματισμός πλαισίων κλπ [27]. Στο σχήμα που ακολουθεί φαίνεται παραστατικά η συγκεκριμένη τοπολογία δικτύου.



Εικόνα 41: Τοπολογία client

4.1.2 Η λειτουργία Router

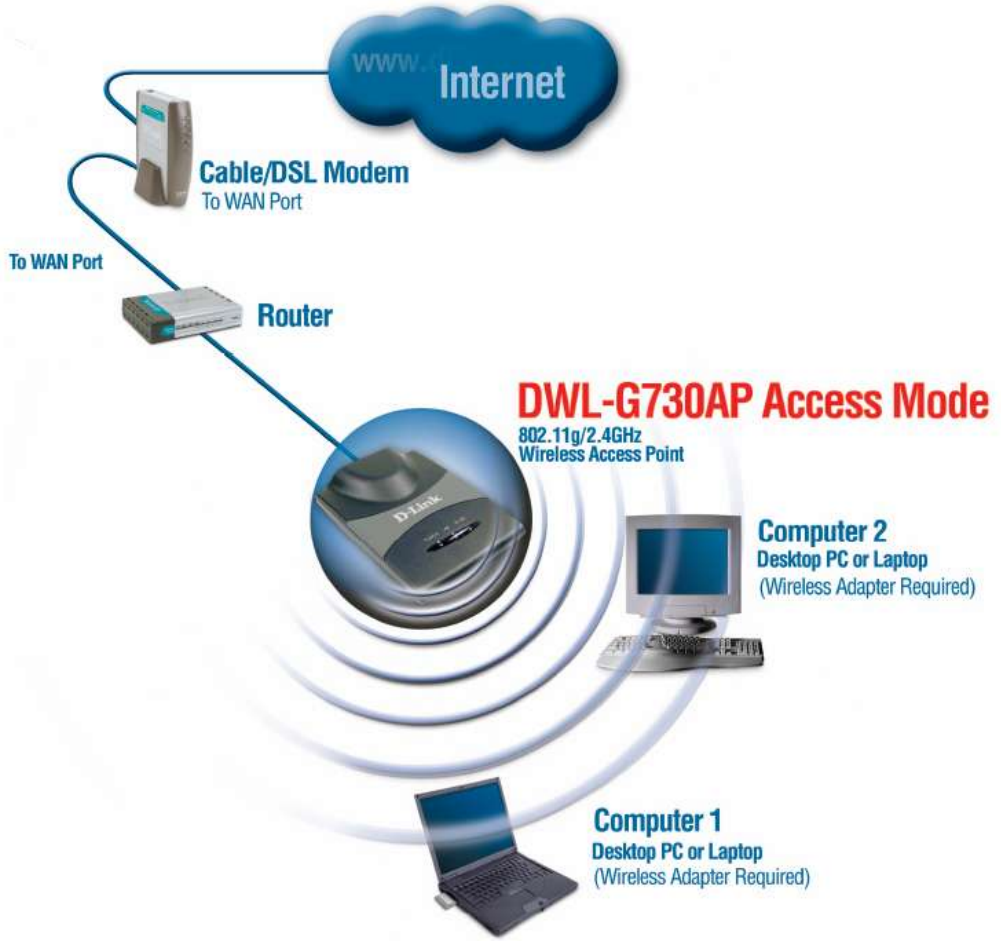
Η πιο πλήρης κατάσταση λειτουργίας είναι η κατάσταση Router. Στην περίπτωση αυτή το G730 αναλαμβάνει να δημιουργήσει ένα ασύρματο δίκτυο, να διαχειριστεί αιτήσεις πελατών για σύνδεση στο δίκτυο, να διαμοιράσει διευθύνσεις IP στους νέους πελάτες, και να αναλάβει τη δρομολόγηση όλων των πακέτων δεδομένων από και προς κάθε πελάτη, καθώς και των πακέτων που εισέρχονται και εξέρχονται από το οικείο δίκτυο προς άλλα δίκτυα. Σε αυτή την κατάσταση το G730 μπορεί να γεφυρώσει την ασύρματη σύνδεση πολλών πελατών με τη μία ενσύρματη σύνδεση που διαθέτει. Επίσης υπάρχει η δυνατότητα προηγμένων λειτουργιών, όπως φιλτράρισμα πακέτων ανάλογα με τη διεύθυνση IP ή τη διεύθυνση MAC, κρυπτογράφηση των δεδομένων, υποστήριξη UPnP, τείχος προστασίας (Firewall) και λειτουργία Network Address Translation (NAT), με την οποία ο χρήστης ορίζει σε ποιες θύρες των πελατών να δρομολογούνται ορισμένα πακέτα. Για τον καθορισμό των απαραίτητων παραμέτρων ο χρήστης πρέπει να συνδεθεί ασύρματα με τη βοήθεια ενός web browser στην αρχική σελίδα του G730 [27]. Στο ακόλουθο σχήμα φαίνεται παραστατικά η συγκεκριμένη τοπολογία δικτύου.



Εικόνα 42: Τοπολογία router

4.1.3 Η λειτουργία Access Point

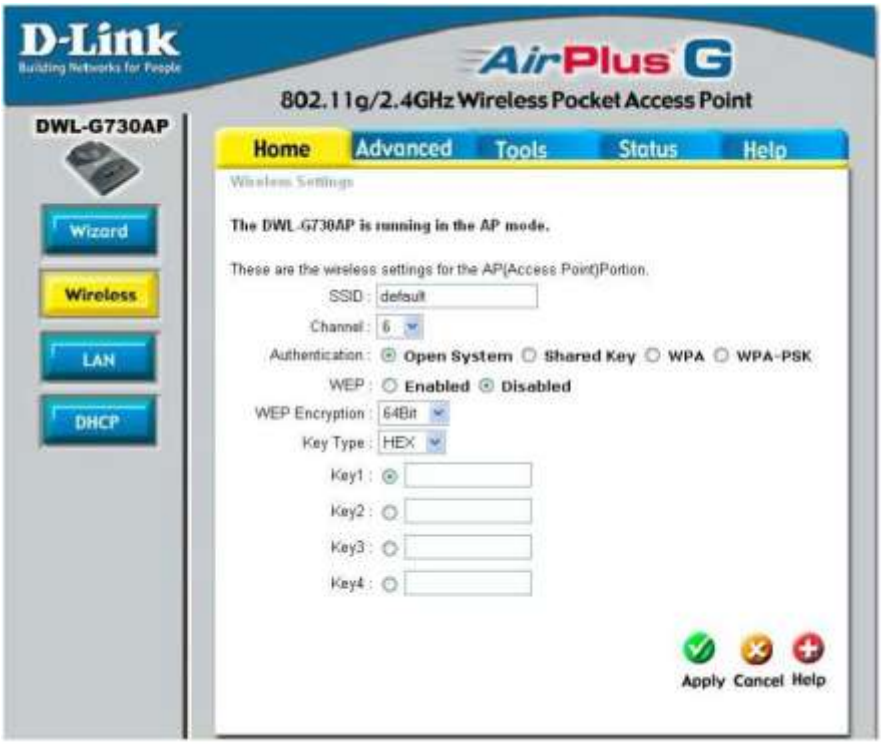
Σε αυτή την κατάσταση λειτουργίας το G730 δημιουργεί και διαχειρίζεται αυτόνομα ένα ασύρματο δίκτυο όπως και στην κατάσταση Router, αλλά δεν προσφέρει δυνατότητες δρομολόγησης στους πελάτες του. Η λειτουργία Access Point προσφέρει τις δυνατότητες γεφύρωσης του ασύρματου δικτύου με το ενσύρματο όπως και στη λειτουργία Client, με τη διαφορά ότι το ασύρματο δίκτυο το διαχειρίζεται το ίδιο το G730 και όχι κάποιο άλλο AP. Αυτό το πλεονέκτημα είναι που κάνει τη διαφορά. Μας δίνει τη δυνατότητα να δημιουργήσουμε ένα ασύρματο δίκτυο, στο οποίο μπορούν να συνδεθούν ασύρματα άλλα Access Points, με τα οποία μπορεί να ανταλλάσει δεδομένα ο ενσύρματος πελάτης του G730 [27]. Στο σχήμα που ακολουθεί παρουσιάζεται η τοπολογία Access Point.



Εικόνα 43: Τοπολογία Access point

Στο σύστημά που υλοποιήσαμε, ο Ethernet controller του server συνδέεται ενσύρματα με το G730 και δημιουργούμε ένα ασύρματο δίκτυο με την ονομασία weather. Σε αυτό το δίκτυο μπορούν πλέον να συνδεθούν άλλα access points που διαθέτουν σύνδεση στο Internet έτσι ώστε ένας οποιοσδήποτε υπολογιστής να μπορεί μέσω

διαδικτύου και Access Points, να επικοινωνήσει με τον μικροελεγκτή μας και τον server. Στο σχήμα που ακολουθεί παρουσιάζεται η κεντρική σελίδα κατασκευής ενός ασύρματου δικτύου.



Εικόνα 44: Κεντρική σελίδα κατασκευής νέου ασύρματου δικτύου

Μέρος 2ο Software

Κεφάλαιο 5
Server

Ο εξυπηρετητής διαδικτύου (server) αποτελεί τον πυρήνα λειτουργιών του συστήματός μας. Η βασικές λειτουργίες του server είναι οι ακόλουθες:

- Διασύνδεση σε τοπικό δίκτυο LAN
- Υλοποίηση πρωτοκόλλων TCP, IP, ARP, ICMP, UDP και Ethernet
- Λήψη πλαισίων Ethernet
- Δημιουργία και αποστολή πλαισίων Ethernet
- Λήψη αναλογικών μετρήσεων από αισθητήρες
- Επικοινωνία με ψηφιακό αισθητήρα
- Δημιουργία ιστοσελίδων http και XML
- Επικοινωνία και οδήγηση οθόνης LCD
- Υλοποίηση ολοκληρωτή
- Αποθήκευση μετρήσεων στην μνήμη EEPROM
- Έλεγχος της τροφοδοσίας του WiFi
- Έλεγχος του φωτισμού της οθόνης LCD μέσω push button
- Αποστολή δεδομένων σε ειδικό καταγραφέα (logger) μέσω XML
- Λήψη ημερομηνίας και ώρας από τον logger
- Υλοποίηση έξυπνης διαχείρισης ενέργειας (power profile) για περιοδική απενεργοποίηση του WiFi όταν το επίπεδο τάσης της μπαταρίας είναι χαμηλό
- Αποστολή μετρήσεων κάθε λεπτό. Όσο το WiFi είναι ανενεργό οι μετρήσεις αποθηκεύονται και αποστέλλονται όλες μαζί όταν το WiFi ενεργοποιηθεί

5.1 Ο κώδικας του μικροελεγκτή

Στην ενότητα αυτή θα περιγράψουμε τις συναρτήσεις που συνθέτουν το λογισμικό του server. Ο κώδικας είναι γραμμένος σε γλώσσα C, με τη βοήθεια της βιβλιοθήκης avr-libc, ενώ οι διάφορες συναρτήσεις ομαδοποιούνται σε ξεχωριστά αρχεία. Πιο αναλυτικά:

- Το αρχείο **analog.c** περιλαμβάνει συναρτήσεις για τη λήψη και επεξεργασία αναλογικών δεδομένων από τους αισθητήρες του συστήματος

- Το αρχείο **enc28j60.c** περιλαμβάνει τις συναρτήσεις χειρισμού του Ethernet controller ENC28J60 από τον μικροελεγκτή, μέσω του διαύλου SPI
- Το αρχείο **integrator.c** προσομοιώνει τη λειτουργία ενός ολοκληρωτή, για την μείωση των σφαλμάτων στις μετρήσεις της τάσης της μπαταρίας
- Το αρχείο **ip_arp_udp_tcp.c** περιλαμβάνει τις συναρτήσεις που υλοποιούν το TCP stack χωρίς fragmentation, την κατασκευή επικεφαλίδων των πακέτων, καθώς και τη δημιουργία νέων πλαισίων
- Το αρχείο **lcd.c** περιλαμβάνει τις συναρτήσεις οδήγησης της οθόνης LCD
- Το αρχείο **main.c** αποτελεί το κύριο αρχείο του προγράμματος και περιλαμβάνει την βασική ρουτίνα εκτέλεσης, καθώς και τις βασικές διαδικασίες λειτουργίας του συστήματος
- Το αρχείο **sensirion_protocol.c** περιλαμβάνει τις συναρτήσεις λειτουργίας του αισθητήρα θερμοκρασίας και υγρασίας
- Το αρχείο **timeout.c** περιλαμβάνει τις συναρτήσεις χρονοκαθυστέρησης
- Τα αρχεία **analog.h, enc28j60.h, integrator.h, ip_arp_udp_tcp.h, lcd.h, sensirion_protocol.h, timeout.h** είναι τα αντίστοιχα αρχεία επικεφαλίδων και περιλαμβάνουν αρχικοποιήσεις τύπου #define καθώς και τον ορισμό των external συναρτήσεων
- Το αρχείο επικεφαλίδας **avr_compat.h** περιλαμβάνει ελέγχους τύπου #ifndef και #define και χρησιμοποιείται για λόγους συμβατότητας με εντολές που πιθανόν δεν υποστηρίζονται από παλαιότερους compilers
- Το αρχείο επικεφαλίδας **config.h** χρησιμοποιείται για αρχικοποίηση του server και περιλαμβάνει αρχικές τιμές κανονικοποίησης των μετρήσεων
- Το αρχείο επικεφαλίδας **lcd_hw.h** περιλαμβάνει τις ρυθμίσεις καλωδίωσης της οθόνης με τον μικροελεγκτή
- Το αρχείο επικεφαλίδας **net.h** περιλαμβάνει απαραίτητα #define για την υλοποίηση του TCP stack από το chip ENC28J60

Στη συνέχεια παρουσιάζουμε αναλυτικά τις συναρτήσεις κάθε αρχείου.

5.1.1 analog.c

Το αρχείο analog.c χρησιμοποιεί τα αρχεία avr/io.h και config.h καθώς και την εξωτερική μεταβλητή *wifi_status* που αποθηκεύει την κατάσταση λειτουργίας του WiFi. Οι συναρτήσεις που περιλαμβάνει είναι οι ακόλουθες:

convertanalog

Η συνάρτηση αυτή επιστρέφει την αναλογική τιμή ενός καναλιού του ADC και λειτουργεί χωρίς διακοπές. Αρχικά αρχικοποιείται ο ADC με κατάλληλες τιμές σύμφωνα με το φύλλο προδιαγραφών και διαβάζεται η τιμή του χαμηλότερης αξίας byte από τον καταχωρητή μετρήσεων του ADC. Στη συνέχεια διαβάζεται το υψηλότερης αξίας byte και το αποτέλεσμα αποθηκεύεται στη μεταβλητή result. Η διαδικασία επαναλαμβάνεται ακόμα μία φορά και η νέα μέτρηση προστίθεται στο result. Τελικά διαιρούμε το result με το δύο και λαμβάνουμε το μέσο όρο των δύο διαδοχικών μετρήσεων. Η συνάρτηση αυτή είναι γενική (generic) και μπορεί να χρησιμοποιηθεί για την λήψη της ακατέργαστης μέτρησης (raw value) οποιουδήποτε καναλιού του ADC.

voltsolar

Η συνάρτηση αυτή μετασχηματίζει την παράμετρο εισόδου *adcval* σε τάση ηλιακού συλλέκτη. Η έξοδος δίνεται από τη σχέση (31·*adcval*+569)/139 που υπολογίσαμε πειραματικά, όπως θα δούμε στο 9^{ο} κεφάλαιο, ενώ ανάλογα με την κατάσταση του WiFi (μεταβλητή *wifi_status*) στο αποτέλεσμα προστίθεται το 7. Η τάση που προκύπτει τελικά είναι το δεκαπλάσιο της πραγματικής, ώστε να διατηρείται το ψηφίο των δεκάτων. Επίσης δεν γίνεται χρήση αριθμητικής κινητής υποδιαστολής, για εξοικονόμηση υπολογιστικής ισχύος και μνήμης προγράμματος (αν γινόταν χρήση αριθμητικής κινητής υποδιαστολής, ο compiler θα παρήγαγε πολλαπλάσιο κώδικα assembly). Όπως θα δούμε, στην τελική τάση προστίθεται υποδιαστολή πριν το λιγότερο σημαντικό ψηφίο.

voltbattery

Ομοίως η συνάρτηση αυτή μετασχηματίζει την παράμετρο εισόδου *adcval* σε τάση μπαταρίας. Η έξοδος δίνεται από τη σχέση 2+(30·*adcval*+669)/180 που υπολογίσαμε και πάλι πειραματικά, ενώ ανάλογα με την κατάσταση του WiFi στο αποτέλεσμα προστίθεται το 8. Ομοίως, η τάση που προκύπτει είναι δεκαπλάσια της πραγματικής.

ampsolar

Η συνάρτηση αυτή μετασχηματίζει την παράμετρο εισόδου *adcval* σε ρεύμα ηλιακού συλλέκτη. Η έξοδος δίνεται από τη σχέση (28·*adcval*-602)/37 που υπολογίστηκε πειραματικά. Αν το ρεύμα προκύψει μικρότερο από μηδέν, τότε το αποτέλεσμα είναι μηδέν. Και πάλι η τιμή εξόδου του ρεύματος είναι το δεκαπλάσια της πραγματικής.

c2f

Η συνάρτηση αυτή μετατρέπει μια θερμοκρασία από κλίμακα Κελσίου σε κλίμακα Φαρενάιτ. Η θερμοκρασία εισόδου δίνεται σε δέκατα του βαθμού Κελσίου (*celsiustimes10*) και η έξοδος υπολογίζεται με βάση τη σχέση (*celsiustimes10*·18+3200)/10 σε δέκατα του βαθμού Φαρενάιτ [28]. Η συνάρτηση αυτή είναι generic και μπορεί να χρησιμοποιηθεί οπουδήποτε χρειάζεται μετατροπή σε κλίμακα Φαρενάιτ.

pressure

Η συνάρτηση αυτή μετασχηματίζει την παράμετρο εισόδου *adcval* σε ατμοσφαιρική πίεση. Η έξοδος δίνεται από τη σχέση R+4·(*adcval*+E)·25/G, όπου E είναι το σφάλμα, G το κέρδος και R η τιμή αναφοράς όπως δίνονται στο αρχείο config.h. Η καμπύλη μετασχηματισμού προκύπτει άμεσα από την εξίσωση Βαρομετρικής Πίεσης (Barometric Formula) [29]. Το αποτέλεσμα που προκύπτει είναι δεκαπλάσιο του πραγματικού.

pressureATsealevel

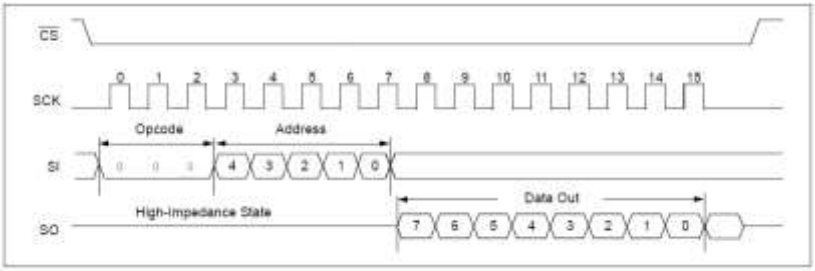
Η συνάρτηση αυτή μετασχηματίζει την τιμή εισόδου *phpa10* σε ατμοσφαιρική πίεση στην επιφάνεια της θάλασσας. Η έξοδος δίνεται από τη σχέση *phpa10*+S, όπου S η τιμή αναφοράς όπως δίνεται στο αρχείο config.h. Το αποτέλεσμα που προκύπτει είναι δεκαπλάσιο του πραγματικού.

5.1.2 enc28j60.c

Το αρχείο enc28j60.c χρησιμοποιεί τα αρχεία avr/io.h, avr_compat.h, enc28j60.h, timeout.h και util/delay.h. Αρχικά ορίζονται οι γραμμές που καλωδιώνουν το chip και ορισμένες χρήσιμες μακροεντολές. Οι συναρτήσεις που περιλαμβάνει είναι οι ακόλουθες:

enc28j60ReadOp

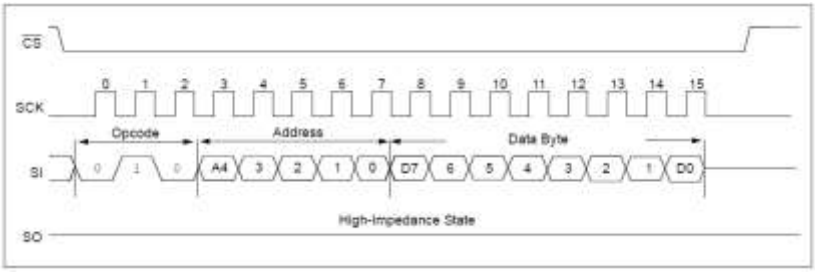
Η συνάρτηση αυτή διαβάζει ένα byte από τη μνήμη του ελεγκτή Ethernet. Αρχικά ενεργοποιείται η γραμμή chip select. Στη συνέχεια αποστέλλεται ο κωδικός εντολής και η διεύθυνση μνήμης μέσω του διαύλου SPI, και η ρουτίνα περιμένει να ολοκληρωθεί η μεταφορά. Ακολούθως μηδενίζεται ο καταχωρητής SPI data και η ρουτίνα περιμένει να διαβαστούν δεδομένα. Η ανάγνωση γίνεται ακόμα και αν δεν υπάρχουν δεδομένα. Τελικά η γραμμή chip select απελευθερώνεται και ο καταχωρητής SPDR περιέχει το byte που διαβάστηκε.



Εικόνα 45: Διαδικασία ανάγνωσης byte

enc28j60WriteOp

Η συνάρτηση αυτή εγγράφει ένα byte. Αρχικά ενεργοποιείται η γραμμή chip select. Στη συνέχεια αποστέλλεται ο κωδικός εντολής και η διεύθυνση μνήμης μέσω του διαύλου SPI, και η ρουτίνα περιμένει να ολοκληρωθεί η μεταφορά. Ακολούθως ο καταχωρητής SPI data λαμβάνει την παράμετρο εισόδου *data* και η ρουτίνα περιμένει να εγγραφούν τα δεδομένα. Τελικά η γραμμή chip select απελευθερώνεται.



Εικόνα 46: Διαδικασία εγγραφής byte

enc28j60ReadBuffer

Η συνάρτηση αυτή διαβάζει δεδομένα πλήθους *len* από τον buffer του ελεγκτή Ethernet και τα αποθηκεύει στη θέση μνήμης *data*. Αρχικά ενεργοποιείται η γραμμή chip select. Στη συνέχεια αποστέλλεται η κατάλληλη εντολή ανάγνωσης μέσω του διαύλου SPI, και η ρουτίνα περιμένει να ολοκληρωθεί η μεταφορά. Ακολούθως επαναλαμβάνεται *len* φορές ένας βρόχος ανάγνωσης που αποθηκεύει να αναγνωσμένα bytes σε διαδοχικές θέσεις μνήμης αρχίζοντας από τη θέση *data*. Τελικά ο buffer τερματίζεται με ένα κενό χαρακτήρα και η γραμμή chip select απελευθερώνεται.

enc28j60WriteBuffer

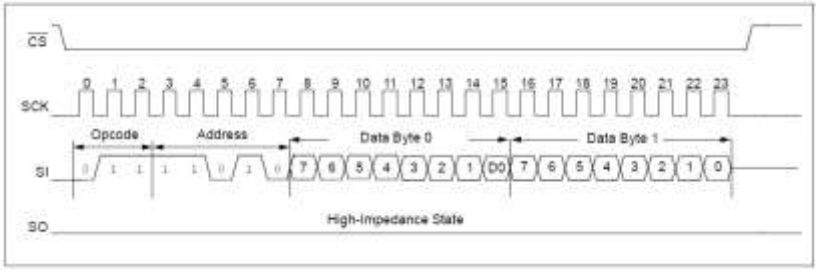
Η συνάρτηση αυτή εγγράφει στον buffer του ελεγκτή Ethernet δεδομένα πλήθους *len* που ξεκινούν από τη θέση μνήμης *data*. Αρχικά ενεργοποιείται η γραμμή chip select. Στη συνέχεια αποστέλλεται η κατάλληλη εντολή εγγραφής μέσω του διαύλου SPI, και η ρουτίνα περιμένει να ολοκληρωθεί η

μεταφορά. Ακολούθως επαναλαμβάνεται *len* φορές ένας βρόχος που διαβάζει bytes από διαδοχικές θέσεις μνήμης, αρχίζοντας από τη θέση *data,* και τα εγγράφει στο δίαυλο SPI. Τελικά η γραμμή chip select απελευθερώνεται.

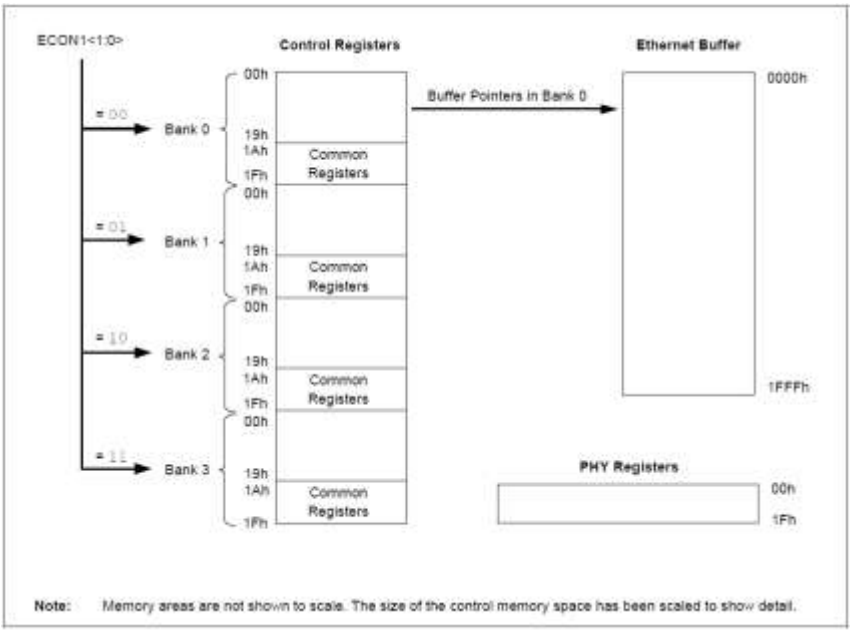
Instruction Name and Mnemonic	Byte 0		Byte 1 and Following
	Opcode	Argument	Data
Read Control Register (RCR)	0 0 0	a a a a a	N/A
Read Buffer Memory (RBM)	0 0 1	1 1 0 1 0	N/A
Write Control Register (WCR)	0 1 0	a a a a a	d d d d d d d d
Write Buffer Memory (WBM)	0 1 1	1 1 0 1 0	d d d d d d d d
Bit Field Set (BFS)	1 0 0	a a a a a	d d d d d d d d
Bit Field Clear (BFC)	1 0 1	a a a a a	d d d d d d d d
System Reset Command (Soft Reset) (SRC)	1 1 1	1 1 1 1 1	N/A

Legend: a = control register address, d = data payload

Εικόνα 47: Πίνακας εντολών



Εικόνα 48: Διαδικασία εγγραφής buffer



Εικόνα 49: Οργάνωση μνήμης του ελεγκτή Ethernet

enc28j60SetBank

Η συνάρτηση αυτή ορίζει το bank μνήμης. Αρχικά γίνεται έλεγχος της τιμής της στατικής μεταβλητής *Enc28j60Bank* που κρατά το τρέχον bank μνήμης, και ανάλογα με το αποτέλεσμα γίνονται δύο κατάλληλες εγγραφές στον καταχωρητή ECON1, ενώ η τιμή της μεταβλητής *Enc28j60Bank* ανανεώνεται.

enc28j60Read

Η συνάρτηση αυτή διαβάζει ένα control register που περιγράφεται από την παράμετρο *address*. Αρχικά ορίζεται η διεύθυνση του τρέχοντος bank μνήμης και στη συνέχεια γίνεται ανάγνωση του αντίστοιχου control register.

enc28j60Write

Η συνάρτηση αυτή εγγράφει ένα byte σε ένα control register που περιγράφεται από την παράμετρο *address*. Αρχικά ορίζεται η διεύθυνση του τρέχοντος bank μνήμης και στη συνέχεια γίνεται εγγραφή στον αντίστοιχο control register.

enc28j60PhyWrite

Η συνάρτηση αυτή εγγράφει δεδομένα στον καταχωρητή φυσικού επιπέδου που ορίζεται από την παράμετρο *address*. Αρχικά εγγράφεται η διεύθυνση του PHY register στον control register MIREGADR. Στη συνέχεια εγγράφονται το χαμηλότερο και υψηλότερο byte της μεταβλητής εισόδου *data*. Τελικά η συνάρτηση αναμένει μέχρι να ολοκληρωθεί η εγγραφή, χρησιμοποιώντας κατάλληλη ρουτίνα χρονοκαθυστέρησης.

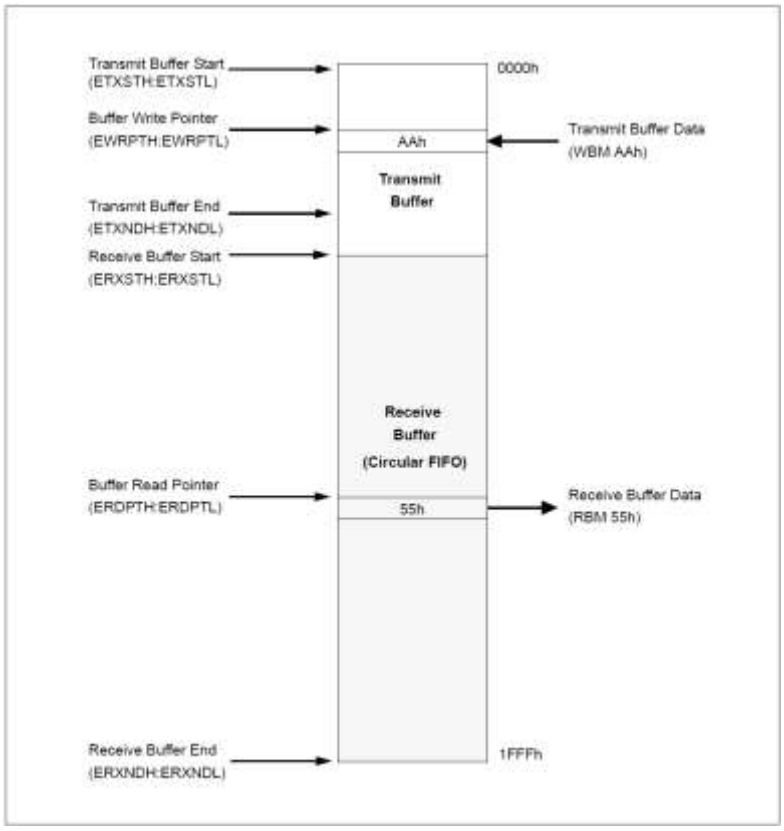
enc28j60clkout

Η συνάρτηση αυτή ορίζει τη συχνότητα παλμών του ακροδέκτη CLKOUT του ελεγκτή Ethernet. Η δοσμένη παράμετρος εισόδου εγγράφεται στον αντίστοιχο καταχωρητή για τον ορισμό της συχνότητας ρολογιού. Η τιμή εισόδου *clk=2* αντιστοιχεί σε 12.5 MHz.

enc28j60Init

Η συνάρτηση αυτή αρχικοποιεί τον ελεγκτή Ethernet. Αρχικά η γραμμές CS, MOSI και SCK ορίζονται ως έξοδοι, ενώ η γραμμή MISO ορίζεται ως είσοδος. Επίσης τα MOSI και SCK τίθενται στην κατάσταση low. Στη συνέχεια αρχικοποιούνται οι καταχωρητές SPSR και SPCR του μικροελεγκτή, ώστε το SPI να λειτουργεί σε master mode διπλής ταχύτητας. Ακολούθως γίνεται reset του ελεγκτή Ethernet, και καλείται κατάλληλη ρουτίνα χρονοκαθυστέρησης. Στη συνέχεια μηδενίζεται μετρητής πακέτων EPKTCNT και αρχικοποιείται ο buf-

fer λήψης. Αμέσως μετά ορίζεται στον buffer η αρχή της λήψης, ο pointer της διεύθυνσης λήψης και το τέλος της λήψης. Ομοίως ορίζεται η αρχή και το τέλος της μετάδοσης. Αφού αρχικοποιηθούν οι απαραίτητοι καταχωρητές μάσκας, γίνεται η ενεργοποίηση της λήψης επιπέδου MAC και ορίζεται το padding στα 60 bytes. Ακολούθως ενεργοποιείται ο έλεγχος CRC, ορίζεται το κενό μεταξύ πλαισίων καθώς και το μέγιστο μήκος πλαισίου. Τελικά εγγράφεται η επιθυμητή διεύθυνση MAC και ενεργοποιούνται οι διακοπές και η λήψη πακέτου στον ελεγκτή Ethernet.



Εικόνα 50: Η οργάνωση του buffer

enc28j60getrev

Η συνάρτηση αυτή διαβάζει την έκδοση του chip από τον καταχωρητή EREVID.

enc28j60PacketSend

Η συνάρτηση αυτή αποστέλλει ένα πακέτο. Αρχικά ορίζεται ένας δείκτης στην αρχή του buffer που περιέχει το πακέτο και αποθηκεύεται στον καταχωρητή ETXND το μέγεθος του πακέτου. Στη συνέχεια εγγράφεται το byte ελέγχου και αποστέλλεται από τον μικροελεγκτή ολόκληρος ο buffer με τα δεδομένα του πακέτου. Τελικά ο ελεγκτής Ethernet αποστέλλει τα δεδομένα

που μόλις έλαβε στο δίκτυο και η ακολουθία αποστολής αρχικοποιείται ξανά.

enc28j60PacketReceive

Η συνάρτηση αυτή λαμβάνει ένα πακέτο από το δίκτυο, εάν υπάρχει. Αρχικά ελέγχεται ο καταχωρητής EPKTCNT για την ύπαρξη νέου πακέτου. Αν δεν υπάρχει νέο πακέτο η συνάρτηση επιστρέφει μηδέν, αλλιώς ο δείκτης ανάγνωσης τοποθετείται στην αρχή του ληφθέντος πακέτου. Στη συνέχεια διαβάζεται το μήκος του πακέτου και η κατάσταση λήψης, και αν ο έλεγχος CRC επιτύχει, τότε αντιγράφεται το πακέτο στον μικροελεγκτή. Τελικά ο δείκτης λήψης δείχνει στο επόμενο πακέτο, και αφού μειωθεί ο μετρητής πακέτων, γίνεται απελευθέρωση της μνήμης.

5.1.3 integrator.c

Το αρχείο integrator.c χρησιμοποιεί τα αρχεία stdlib.h, stdio.h, integrator.h και analog.h. Οι συναρτήσεις που περιλαμβάνει είναι οι ακόλουθες:

initbat

Η συνάρτηση αυτή αρχικοποιεί ένα πίνακα δεδομένων με τις τιμές της τάσης της μπαταρίας, όπως προκύπτουν από πολλαπλές κλήσεις της συνάρτησης voltbattery. Σε κάθε κλήση της voltbattery η τιμή της τάσης προστίθεται στη μεταβλητή *bat_total*, η οποία κάθε στιγμή ισούται με το άθροισμα όλων των στοιχείων του πίνακα.

readbat

Η συνάρτηση αυτή επιστρέφει το μέσο όρο όλων των στοιχείων του πίνακα που αντιστοιχεί στην μέση τάση της μπαταρίας. Αρχικά διαβάζεται πέντε φορές η τιμή του ADC που μετρά την τάση της μπαταρίας, και λαμβάνεται η μέση τιμή των πέντε μετρήσεων. Στη συνέχεια εφαρμόζεται η συνάρτηση voltbattery στην μέση τιμή των πέντε μετρήσεων και προκύπτει μια νέα τάση μπαταρίας. Αυτή η νέα τάση αντικαθιστά την παλαιότερη τιμή τάσης του πίνακα (που κρατιέται στο δείκτη *bat_index*) και ενημερώνεται κατάλληλα η μεταβλητή *bat_total*. Τελικά υπολογίζεται ο νέος μέσος όρος της τάσης με κατάλληλη ολίσθηση της ποσότητας *bat_total* δεξιά, ώστε η διαίρεση να γίνεται αποδοτικά. Ο δείκτης *bat_index* αυξάνεται και γίνεται έλεγχος υπερχείλισης.

Με τη χρήση αυτής της συνάρτησης εξασφαλίζεται ότι κάθε στιγμή η τελική τιμή της τάσης της μπαταρίας είναι αποτέλεσμα ολοκλήρωσης πολλαπλών μετρήσεων, ώστε να μην υπάρχουν απότομες μεταβολές λόγω σφάλματος στις μετρήσεις. Ο υπολογισμός της τάσης μέσω ολοκληρωτή στηρίζεται στο γεγονός ότι η τάση της μπαταρίας του συστήματός μας μεταβάλλεται εξαιρετικά αργά και όποιες απότομες μεταβολές παρατηρούνται σε μεμονωμένες μετρήσεις είναι αποτέλεσμα σφάλματος του ADC.

5.1.4 ip_arp_udp_tcp.c

Το αρχείο ip_arp_udp_tcp.c χρησιμοποιεί τα αρχεία avr/io.h, avr/pgmspace.h, avr_compat.h, net.h και enc28j60.h. Οι συναρτήσεις που περιλαμβάνει είναι οι ακόλουθες:

checksum

Η συνάρτηση αυτή υπολογίζει το checksum της κεφαλίδας ενός πακέτου που ορίζεται από τον δείκτη *buf* [30] [31] [32]. Αν το πακέτο είναι τύπου IP, τότε το άθροισμα αρχικοποιείται στο μηδέν. Όμως αν το πακέτο είναι τύπου UDP τότε αρχικοποιείται στην τιμή 17+*len*-8. Ομοίως αν το πακέτο είναι τύπου TCP τότε το άθροισμα αρχικοποιείται στην τιμή 6+*len*-8. Στη συνέχεια στο άθροισμα προστίθεται όλες τις λέξεις 16 bit και αν χρειάζεται προστίθεται και το υπολειπόμενο byte. Όσο το άθροισμα έχει μήκος μικρότερο από 16 bit, τότε διπλασιάζεται. Τελικά η συνάρτηση επιστρέφει το συμπλήρωμα ως προς ένα του τελικού αθροίσματος.

init_ip_arp_udp_tcp

Η συνάρτηση αυτή αρχικοποιεί τις διευθύνσεις IP και MAC, καθώς και την θύρα στην οποία λαμβάνει αιτήσεις ο server.

eth_type_is_arp_and_my_ip

Η συνάρτηση αυτή ελέγχει αν το δοσμένο πακέτο που ορίζεται από τον δείκτη *buf*, είναι τύπου ARP. Αρχικά ελέγχεται το μήκος του πακέτου και στη συνέχεια συγκρίνεται το πεδίο ETH_TYPE με προκαθορισμένες τιμές. Τελικά ελέγχεται αν ταυτίζεται η διεύθυνση προορισμού του πακέτου με τη διεύθυνση IP του server.

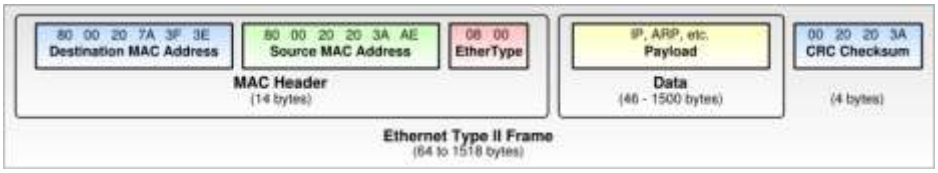
eth_type_is_ip_and_my_ip

Η συνάρτηση αυτή ελέγχει αν το δοσμένο πακέτο που ορίζεται από τον δείκτη *buf*, είναι τύπου IP. Αρχικά ελέγχεται το μήκος του πακέτου και στη συ-

νέχεια συγκρίνεται το πεδίο ETH_TYPE με προκαθορισμένες τιμές. Ακολούθως ελέγχεται αν το μήκος της επικεφαλίδας του πακέτου είναι 20 bytes, όπως προβλέπεται για τα πακέτα IP. Τελικά ελέγχεται αν ταυτίζεται η διεύθυνση προορισμού του πακέτου με τη διεύθυνση IP του server.

make_eth

Η συνάρτηση αυτή δημιουργεί μια νέα επικεφαλίδα Ethernet επάνω στο πλαίσιο που μόλις έλαβε και που ορίζεται από τον δείκτη *buf*. Η έναρξη της επικοινωνίας γίνεται πάντα από τον client που διατυπώνει ένα αίτημα στο οποίο ο server απαντά. Η υλοποίηση προβλέπει την αντιστροφή των διευθύνσεων MAC πηγής και προορισμού και την αντικατάσταση της διεύθυνσης της πηγής με τη διεύθυνση MAC του server.



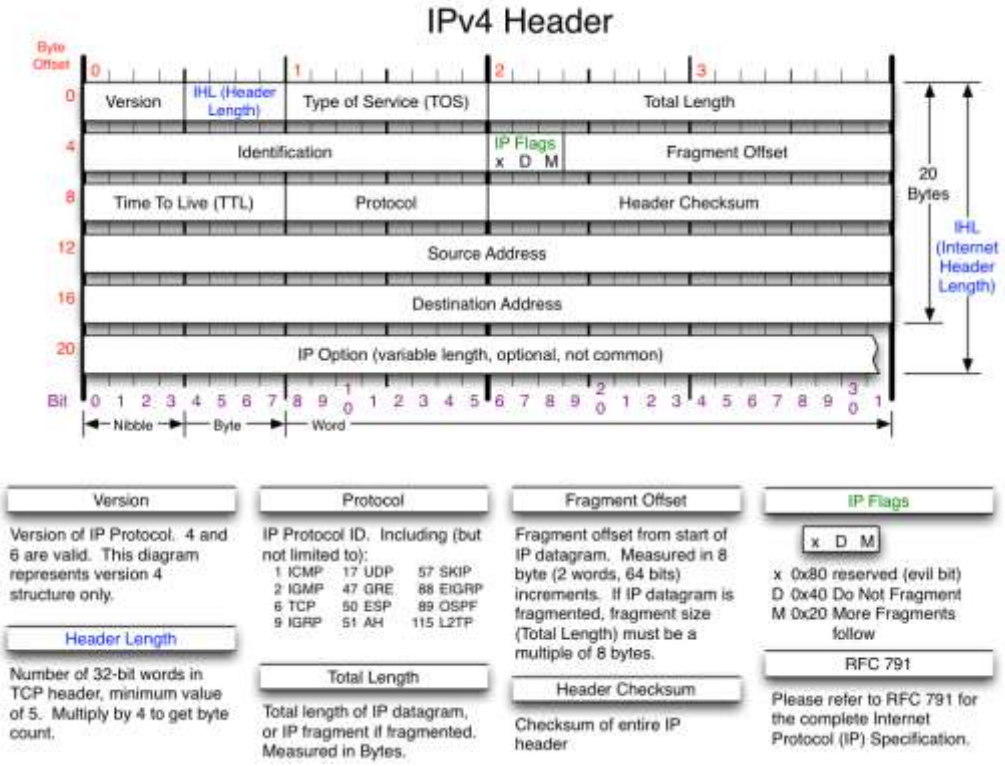
Εικόνα 51: Δομή πλαισίου Ethernet

fill_ip_hdr_checksum

Η συνάρτηση αυτή μεταβάλλει το checksum στην κεφαλίδα ενός πακέτου IP. Αρχικά εντοπίζεται η θέση των δύο bytes του checksum μέσα στο πακέτο (που δεικτοδοτείται από το δείκτη *buf*) και στη συνέχεια αρχικοποιούνται οι κατάλληλες σημαίες ώστε να μην υπάρχει κατακερματισμός. Το πεδίο TTL ορίζεται στα 64 βήματα, και αφού υπολογιστεί το checksum, αποθηκεύεται στα κατάλληλα bytes της κεφαλίδας.

make_ip

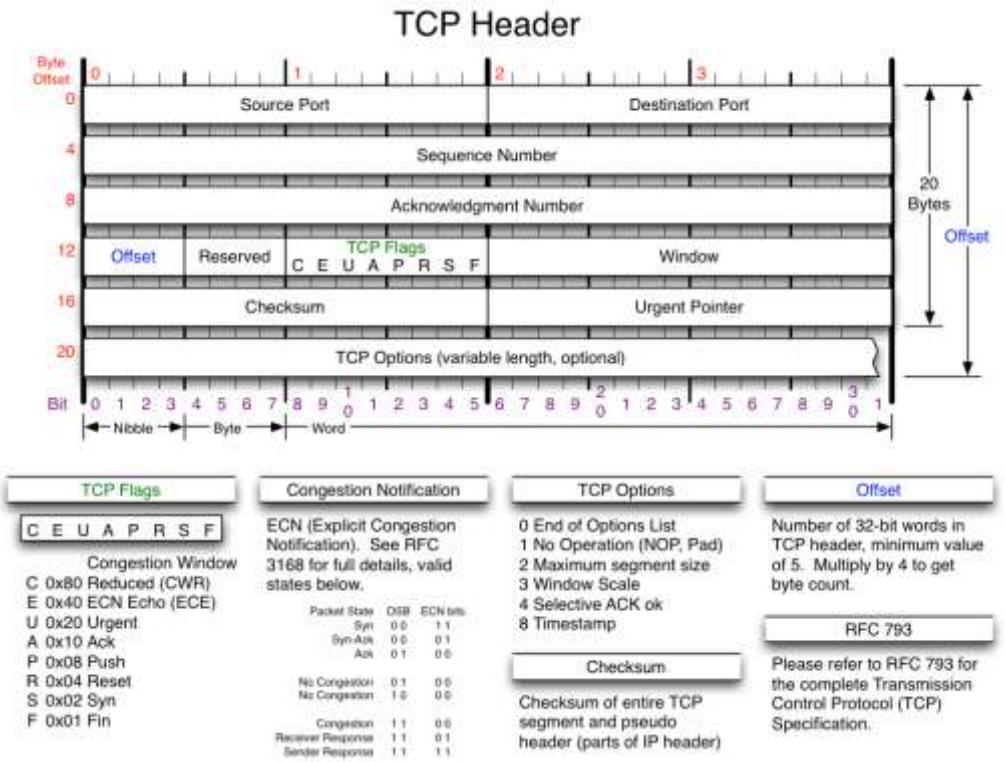
Η συνάρτηση αυτή δημιουργεί μια νέα επικεφαλίδα IP επάνω στο πακέτο που μόλις έλαβε και που ορίζεται από τον δείκτη *buf*. Για το σκοπό αυτό αντιστρέφονται οι διευθύνσεις IP πηγής και προορισμού και στη συνέχεια αντικαθίσταται η διεύθυνση της πηγής με τη διεύθυνση IP του server. Στη συνέχεια είναι απαραίτητη η ανανέωση του checksum στην επικεφαλίδα του πακέτου.



Εικόνα 52: Δομή επικεφαλίδας IP

make_tcphead

Η συνάρτηση αυτή δημιουργεί μια νέα επικεφαλίδα TCP πάνω στο πακέτο που μόλις έλαβε και που ορίζεται από τον δείκτη *buf*. Το σύστημά μας υποστηρίζει πακέτα TCP μήκους το πολύ 255 bytes χωρίς fragmentation, καθώς αυτή η υλοποίηση είναι κατάλληλη για ένα ενσωματωμένο σύστημα όπως αυτό που σχεδιάσαμε. Αρχικά αντιστρέφονται οι θύρες (ports) πηγής και προορισμού και ορίζεται η κατάλληλη θύρα πηγής του server. Στη συνέχεια αυξάνεται η μεταβλητή του sequence και αντιγράφεται ο αριθμός του acknowledgment που λήφθηκε. Αν η παράμετρος *cp_seq* είναι μηδέν, τότε η μεταβλητή του sequence αυξάνεται κατά δύο για να υπερχειλίσει, αφού τα δεδομένα είναι το πολύ 255 bytes. Στη συνέχεια μηδενίζεται το checksum και ελέγχεται η τιμή της μεταβλητής *mss*. Αν η μεταβλητή *mss* είναι 1, τότε στα options της κεφαλίδας TCP περιέχει το πεδίο MSS, το οποίο αρχικοποιείται στην τιμή 1408. Στην περίπτωση αυτή το συνολικό μήκος της κεφαλίδας είναι 24 bytes. Διαφορετικά, αν η μεταβλητή *mss* είναι μηδέν, το μήκος της κεφαλίδας είναι 20 bytes και δεν υπάρχει το πεδίο MSS.



Εικόνα 53: Δομή επικεφαλίδας TCP

make_arp_answer_from_request

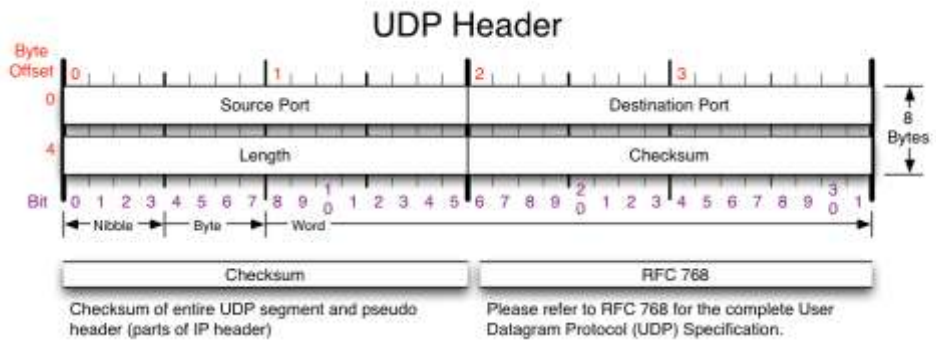
Η συνάρτηση αυτή δημιουργεί απάντηση σε ένα πακέτο ARP. Αρχικά καλείται η συνάρτηση make_eth για τη δημιουργία κεφαλίδας Ethernet από το πλαίσιο που έλαβε. Στη συνέχεια αρχικοποιούνται τα κατάλληλα πεδία του νέο πακέτου ώστε να οριστεί ο τύπος πακέτου ARP. Ακολούθως αντιστρέφονται οι διευθύνσεις MAC πηγής και προορισμού και ορίζεται ως πηγή η διεύθυνση MAC του συστήματός μας. Ομοίως αντιστρέφονται οι διευθύνσεις IP πηγής και προορισμού και ορίζεται ως πηγή η διεύθυνση IP του server. Τελικά καλείται η συνάρτηση enc28j60PacketSend που αποστέλλει το πλαίσιο (μήκους 42 bytes) στο δίκτυο.

make_echo_reply_from_request

Η συνάρτηση αυτή δημιουργεί μία απάντηση σε ένα echo request. Αρχικά καλούνται οι συναρτήσεις make_eth και make_ip για τη δημιουργία κεφαλίδων Ethernet και IP αντίστοιχα. Στη συνέχεια μεταβάλλεται ο τύπος του πακέτου από ICMP request (0x08) σε ICMP reply (0x00), και γίνεται η απαραίτητη διόρθωση στο checksum.

make_udp_reply_from_request

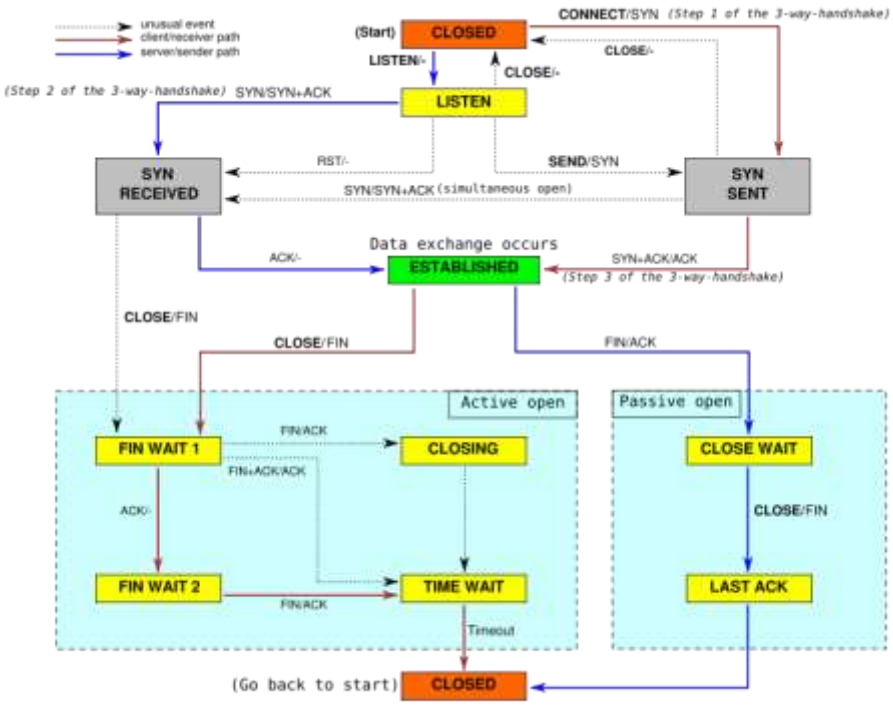
Η συνάρτηση αυτή αποστέλλει μία απάντηση UDP και το μέγιστο μήκος δεδομένων που υποστηρίζεται είναι 220 bytes. Αρχικά καλείται η συνάρτηση make_eth για τη δημιουργία κεφαλίδας Ethernet από το πλαίσιο που λήφθηκε. Στη συνέχεια ορίζεται το μέγιστο μήκος στη κεφαλίδα IP (που ισούται με το μήκος της κεφαλίδας IP, της κεφαλίδας UDP και των δεδομένων), και καλείται η συνάρτηση make_ip για τη δημιουργία κεφαλίδας IP. Ακολούθως ορίζεται η θύρα προορισμού ενώ η θύρα πηγής στην περίπτωσή μας είναι τυχαία. Στο σημείο αυτό υπολογίζεται το συνολικό μήκος του πακέτου UDP και αρχικοποιείται το checksum στο μηδέν. Στη συνέχεια αντιγράφονται τα δεδομένα (που δεικτοδοτούνται από την παράμετρο *data*) στον χώρο δεδομένων του πακέτου UDP που ορίζεται από τον δείκτη *buf*. Τελικά, αφού υπολογιστεί και αποθηκευτεί το νέο checksum, καλείται η συνάρτηση enc28j60PacketSend που αποστέλλει το πλαίσιο στο δίκτυο.



Εικόνα 54: Δομή επικεφαλίδας UDP

make_tcp_synack_from_syn

Η συνάρτηση αυτή αποστέλλει ένα μήνυμα ACK μετά την ορθή λήψη ενός πακέτου TCP. Όπως είναι γνωστό το πρωτόκολλο TCP απαιτεί τη σύναψη χειραψίας μεταξύ πηγής και προορισμού [33] [34], και η συνάρτηση αυτή εξασφαλίζει την σύναψη αυτής της χειραψίας. Οι καταστάσεις του πρωτοκόλλου TCP φαίνονται αναλυτικά στο σχήμα που ακολουθεί. Αρχικά καλείται η συνάρτηση make_eth για τη δημιουργία κεφαλίδας Ethernet από το πλαίσιο που λήφθηκε, και αποθηκεύεται το συνολικό μήκος της κεφαλίδας IP. Στη συνέχεια καλείται η συνάρτηση make_ip για τη δημιουργία κεφαλίδας IP, και αφού οριστούν οι κατάλληλες σημαίες (flags) της κεφαλίδας TCP, καλείται η συνάρτηση make_tcphead που δημιουργεί τη νέα κεφαλίδα του πακέτου. Τελικά αποθηκεύεται το νέο checksum του πακέτου TCP και καλείται η συνάρτηση enc28j60PacketSend που αποστέλλει το πλαίσιο στο δίκτυο.



Εικόνα 55: Διάγραμμα καταστάσεων TCP

get_tcp_data_pointer

Η συνάρτηση αυτή δημιουργεί ένα δείκτη στην αρχή των δεδομένων TCP και απαιτεί να έχει προηγηθεί η κλήση της συνάρτησης init_len_info. Ανάλογα με την τιμή της στατικής μεταβλητής *info_data_len*, η συνάρτηση επιστρέφει ένα νέο δείκτη, ή μηδέν αν δεν υπάρχουν δεδομένα. Ο υπολογισμός της αρχής των δεδομένων TCP γίνεται προσθέτοντας την τιμή της στατικής μεταβλητής *info_hdr_len* στη θέση (offset) της θύρας πηγής (source port) μέσα στην κεφαλίδα TCP.

init_len_info

Η συνάρτηση αυτή αρχικοποιεί τις μεταβλητές μήκους. Η στατική μεταβλητή *info_data_len* υπολογίζεται ως η διαφορά του συνολικού μήκους του πακέτου IP, από το άθροισμα του μήκους της επικεφαλίδας IP και της επικεφαλίδας TCP. Επίσης, η στατική μεταβλητή *info_hdr_len* υπολογίζεται από τον πολλαπλασιασμό επί 4 των τεσσάρων πιο σημαντικών bits του 12^{ου} byte της κεφαλίδας TCP.

fill_tcp_data_p

Η συνάρτηση αυτή μεταφέρει στατικά δεδομένα σε μία θέση του buffer δεδομένων του πακέτου TCP, απευθείας από την περιοχή προγράμματος (progmem). Με αυτό τον τρόπο γίνεται εξοικονόμηση μνήμης RAM, αφού τα

δεδομένα δεν αποθηκεύονται σε μεταβλητές, αλλά είναι διαρκώς αποθηκευμένα στην μνήμη flash του μικροελεγκτή. Η συνάρτηση διαβάζει διαδοχικά bytes από την περιοχή προγράμματος με χρήση της ειδικής συνάρτησης pgm_read_byte, και στη συνέχεια τα αποθηκεύει σε κατάλληλες θέσεις στον buffer. Ακολούθως υπολογίζεται η επόμενη κενή θέση εγγραφής και η ρουτίνα επαναλαμβάνεται μέχρι να εμφανιστεί ο χαρακτήρας τερματισμού.

fill_tcp_data

Η συνάρτηση αυτή αντιγράφει ένα string σε μία θέση του buffer δεδομένων του πακέτου TCP. Η συνάρτηση διαβάζει διαδοχικά όλα τα byte του string και τα αντιγράφει σε κατάλληλες θέσεις στον buffer μέχρι να συναντήσει τον χαρακτήρα τερματισμού.

make_tcp_ack_from_any

Η συνάρτηση αυτή δημιουργεί ένα πακέτο ACK χωρίς δεδομένα TCP. Αρχικά καλείται η συνάρτηση make_eth για τη δημιουργία επικεφαλίδας Ethernet και ορίζονται οι κατάλληλες σημαίες TCP. Στη συνέχεια ελέγχεται η στατική μεταβλητή *info_data_len* και εκτελείται η συνάρτηση make_tcphead με κατάλληλες παραμέτρους για τη δημιουργία κεφαλίδας TCP. Στη συνέχεια υπολογίζεται το συνολικό μήκος της κεφαλίδας IP, δημιουργείται η νέα κεφαλίδα IP με την εκτέλεση της συνάρτησης make_ip και υπολογίζεται το checksum. Τελικά καλείται η συνάρτηση enc28j60PacketSend που αποστέλλει το πλαίσιο στο δίκτυο.

make_tcp_ack_with_data

Η συνάρτηση αυτή δημιουργεί ένα πακέτο ACK με δεδομένα TCP. Η συνάρτηση αυτή απαιτεί προηγουμένως την κλήση της συνάρτησης init_len_info και δεν μεταβάλλει τις κεφαλίδες Ethernet, IP και TCP, με εξαίρεση το μήκος και το checksum. Αρχικά ορίζονται οι σημαίες TCP για αποστολή χωρίς fragmentation και ενεργοποιείται η σημαία FIN. Στη συνέχεια υπολογίζεται ξανά το συνολικό μήκος της κεφαλίδας IP και ανανεώνεται το checksum του πακέτου TCP. Τελικά καλείται η συνάρτηση enc28j60PacketSend που αποστέλλει το πλαίσιο στο δίκτυο.

5.1.5 lcd.c

Το αρχείο lcd.c χρησιμοποιεί τα αρχεία avr/io.h, avr/pgmspace.h, util/delay.h, lcd_hw.h και lcd.h. Οι συναρτήσεις που περιλαμβάνει είναι οι ακόλουθες:

lcd_delay_ms

Η συνάρτηση αυτή υλοποιεί χρονοκαθυστέρηση ορισμένων millisecond κάνοντας πολλαπλές κλήσεις της ειδικής μακροεντολής _delay_ms με παράμετρο 0.96.

lcd_out_low

Η συνάρτηση αυτή αποστέλλει στην οθόνη lcd τα 4 λιγότερο σημαντικά bits ενός byte. Η επικοινωνία της οθόνης με τον μικροελεγκτή γίνεται μέσω γραμμής δεδομένων των 4 bit, και για αυτό το λόγο η μετάδοση ενός byte γίνεται σε δύο βήματα. Με αυτό τον τρόπο γίνεται εξοικονόμηση στις θύρες εξόδου του μικροελεγκτή.

Η συνάρτηση αρχικά εφαρμόζει μάσκα σε καθένα από τα τέσσερα λιγότερο σημαντικά bit του byte εισόδου, και μεταβάλλει ανάλογα το επίπεδο τάσης της αντίστοιχης γραμμής μεταφοράς δεδομένων. Η μεταβολή της τάσης γίνεται με τη βοήθεια των μακροεντολών sbi (set bit) και cbi (clear bit).

lcd_out_high

Αντίστοιχα με την προηγούμενη, η συνάρτηση αυτή μεταβάλλει το επίπεδο τάσης στη γραμμή μεταφοράς δεδομένων (4 bit), εξετάζοντας καθένα από τα τέσσερα περισσότερο σημαντικά bit του byte εισόδου. Η μεταβολή της τάσης γίνεται και πάλι με τη βοήθεια των μακροεντολών sbi και cbi.

lcd_e_toggle

Η συνάρτηση αυτή προκαλεί ένα παλμό διάρκειας 4 microsecond στη γραμμή ENABLE. Η υλοποίηση γίνεται με τη βοήθεια των μακροεντολών lcd_e_high και lcd_e_low.

lcd_write

Η συνάρτηση αυτή εγγράφει ένα ολόκληρο byte στην οθόνη. Ανάλογα με την τιμή της παραμέτρου *rs* το κάθε half-byte που αποστέλλεται μπορεί να αντιστοιχεί σε εντολή ή δεδομένα. Η ρουτίνα αρχικά ορίζει κατεύθυνση εξόδου στις τέσσερις γραμμές μεταφοράς και στη συνέχεια αποστέλλει το πρώτο half-byte. Ακολούθως μεταβάλλεται το επίπεδο τάσης της γραμμής RS ανάλογα με το αν αποστείλαμε δεδομένα ή εντολές, αποστέλλεται ένας παλμός ENABLE, και η διαδικασία επαναλαμβάνεται ακόμα μία φορά για το δεύτερο half-byte.

lcd_waitcmd

Η συνάρτηση αυτή υλοποιεί χρονοκαθυστέρηση 9 microsecond ή 2 millisecond ανάλογα με την τιμή της παραμέτρου εισόδου, μέχρι η οθόνη lcd να είναι έτοιμη να δεχθεί την επόμενη εντολή. Για το σκοπό αυτό χρησιμοποιείται η μακροεντολή _delay_us και η συνάρτηση lcd_delay_ms. Η λειτουργία της ουσιαστικά υποκαθιστά τη χρήση της γραμμής ελέγχου RW, μειώνοντας τις γραμμές επικοινωνίας με τον μικροελεγκτή. Στην περίπτωση αυτή ο μικροελεγκτής περιμένει ένα εύλογο χρονικό διάστημα, μέσα στο οποίο *προβλέπεται* ότι η οθόνη έχει ολοκληρώσει την τελευταία λειτουργία της. Φυσικά η οθόνη μπορεί να έχει ολοκληρώσει νωρίτερα την τελευταία λειτουργία, όμως αυτή η μικρή απώλεια χρόνου αντισταθμίζεται από την εξοικονόμηση μίας γραμμής ελέγχου.

lcd_command

Η συνάρτηση αυτή αποστέλλει στην οθόνη την εντολή που δέχεται ως παράμετρο, χρησιμοποιώντας τη συνάρτηση lcd_write και κατάλληλες χρονοκαθυστερήσεις (lcd_waitcmd).

lcd_gotoxy

Η συνάρτηση αυτή δέχεται ως παραμέτρους δύο συντεταγμένες x, y και τοποθετεί τον δρομέα (cursor) της οθόνης στη θέση που ορίζουν οι συντεταγμένες αυτές. Ανάλογα με το πλήθος των γραμμών της οθόνης και την τιμή της συντεταγμένης y, εκτελείται η κατάλληλη εντολή lcd_command που τοποθετεί το δρομέα στη σωστή θέση. Πλέον η επόμενη εγγραφή χαρακτήρα θα γίνει στην τρέχουσα θέση του δρομέα.

lcd_putc

Η συνάρτηση αυτή εκτυπώνει ένα χαρακτήρα στην οθόνη, εκτελώντας την εντολή lcd_write σε data mode με παράμετρο τον χαρακτήρα προς εκτύπωση.

lcd_puts

Η συνάρτηση αυτή εκτυπώνει στην οθόνη ένα string, χρησιμοποιώντας πολλαπλές εκτελέσεις της εντολής lcd_putc. Η εκτύπωση τερματίζεται όταν διαβαστεί ο χαρακτήρας τερματισμού του string.

lcd_puts_p

Η συνάρτηση αυτή εκτυπώνει στην οθόνη ένα string απευθείας από την περιοχή προγράμματος (progmem). Η ρουτίνα διαβάζει ένα byte από την περι-

οχή προγράμματος με χρήση της συνάρτησης pgm_read_byte και στη συνέχεια καλείται η συνάρτηση lcd_putc για την εκτύπωσή του στην οθόνη. Η ρουτίνα επαναλαμβάνεται μέχρι να διαβαστεί ο χαρακτήρας τερματισμού.

■ **OPERATING PRINCIPLES & METHODS**

u Control and Display Command

Command	RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0	Execution Time (f_{osc} = 250kHz)	Remark
DISPLAY CLEAR	L	L	L	L	L	L	L	L	L	H	1.64ms	
RETURN HOME	L	L	L	L	L	L	L	L	H	X	1.64ms	Cursor move to first digit
ENTRY MODE SET	L	L	L	L	L	L	L	H	I/D	SH	42μs	• I/D : Set cursor move direction I/D: H Increase; L Decrease • SH : Specifies shift of display SH: H Display is shifted; L Display is not shifted
DISPLAY ON/OFF	L	L	L	L	L	L	H	D	C	B	42μs	• Display D: H Display on; L Display off • Cursor C: H Cursor on; L Cursor off • Blinking B: H Blinking on; L Blinking off
SHIFT	L	L	L	L	L	H	S/C	R/L	X	X	42μs	S/C: H Display shift; L Cursor move R/L: H Right shift; L Left shift
SET FUNCTION	L	L	L	L	H	DL	N	F	X	X	42μs	DL: H 8 bits interface; L 4 bits interface N: H 2 line display; L 1 line display F: H 5 X 10 dots; L 5 X 7 dots
SET CG RAM ADDRESS	L	L	L	H	CG RAM address (corresponds to cursor address)						42μs	CG RAM Data is sent and received after this setting
SET DD RAM ADDRESS	L	L	H	DD RAM address							42μs	DD RAM Data is sent and received after this setting
READ BUSY FLAG & ADDRESS	L	H	BF	Address Counter used for both DD & CG RAM address							0μs	BF: H Busy; L Ready – Reads BF indication internal operating is being performed – Reads address counter contents
WRITE DATA	H	L	Write Data								46μs	Write data into DD or CG RAM
READ DATA	H	H	Read Data								46μs	Read data from DD or CG RAM

Εικόνα 56: Πίνακας εντολών της οθόνης lcd

lcd_init

Η συνάρτηση αυτή αρχικοποιεί την οθόνη και τον δρομέα. Αρχικά οι γραμμές δεδομένων καθώς και οι γραμμές ENABLE και RS ορίζονται ως έξοδοι. Προς το παρόν η οθόνη βρίσκεται σε κατάσταση λειτουργίας 8 bit, οπότε εκτελούνται τρείς αρχικές εγγραφές συγχρονισμού. Στη συνέχεια εκτελείται η

εντολή αλλαγής της κατάστασης λειτουργίας από 8 bit στα 4 bit, με τη βοήθεια της συνάρτησης lcd_command και κατάλληλα ορίσματα. Ακολούθως γίνεται καθαρισμός και σβήσιμο της οθόνης, ενώ επιλέγεται και ο τρόπος παρουσίασης των χαρακτήρων στην οθόνη (entry mode). Τέλος επιλέγεται το είδος του δρομέα με βάση την παράμετρο εισόδου *dispAttr*.

5.1.6 sensirion_protocol.c

Το αρχείο sensirion_protocol.c χρησιμοποιεί τα αρχεία avr/io.h, και util/delay.h. Αρχικά ορίζονται οι γραμμές που καλωδιώνουν τον αισθητήρα καθώς και ορισμένες απαραίτητες μακροεντολές. Οι συναρτήσεις που περιλαμβάνει είναι οι ακόλουθες:

computeCRC8

Η συνάρτηση αυτή υλοποιεί τον έλεγχο CRC μήκους 8 bit. Η παράμετρος εισόδου *inData* αντιστοιχεί στο byte δεδομένων, ενώ η παράμετρος εισόδου *seed* αντιστοιχεί στην τιμή αρχικοποίησης του CRC. Αρχικά υπολογίζουμε το exclusive or (XOR) μεταξύ του λιγότερο σημαντικού bit του *seed* και του byte εισόδου, και αν το αποτέλεσμα είναι μηδέν τότε ολισθαίνουμε μία θέση δεξιά το *seed*. Σε αντίθετη περίπτωση υπολογίζουμε το exclusive or (XOR) του *seed* με το 0x18h, που είναι το χαρακτηριστικό πολυώνυμο του CRC 8 bit. Το αποτέλεσμα αποθηκεύεται πάλι στη μεταβλητή *seed*, η οποία ολισθαίνει μία θέση δεξιά και το περισσότερο σημαντικό bit τίθεται στην τιμή 1. Ακολούθως το byte εισόδου ολισθαίνει μία θέση δεξιά, και η ρουτίνα επαναλαμβάνεται για τα υπόλοιπα 7 bits [35]. Τελικά η έξοδος ισούται με το CRC 8 bit του byte εισόδου. Για τον υπολογισμό του CRC περισσότερων bytes, αρκεί να υπολογίσουμε το CRC του πρώτου και για κάθε επόμενο byte να καλούμε τη συνάρτηση computeCRC8 δίνοντας όρισμα *seed* την έξοδο της ακριβώς προηγούμενης κλήσης.

bitswapbyte

Η συνάρτηση αυτή ανταλλάσει τα 4 λιγότερο σημαντικά bit ενός byte με τα αντιδιαμετρικά περισσότερο σημαντικά bit. Η συνάρτηση αυτή χρειάζεται λόγους συμβατότητας με τη μέθοδο υπολογισμού του CRC από τον αισθητήρα sensirion. Η μετατροπή αρχίζει ολισθαίνοντας το προσωρινό αποτέλεσμα μία θέση αριστερά και θέτοντας το λιγότερο σημαντικό bit ίσο με το λιγότερο σημαντικό bit του byte εισόδου. Στη συνέχεια το byte εισόδου ολισθαίνει μία θέση δεξιά, και η μετατροπή ξεκινά από την αρχή.

s_write_byte

Η συνάρτηση αυτή εγγράφει ένα byte στο δίαυλο του αισθητήρα και περιμένει επιβεβαίωση. Αρχικά ορίζεται η ακίδα DATA ως έξοδος και με χρήση μάσκας επιλέγεται το περισσότερο σημαντικό bit του byte προς αποστολή. Ανάλογα με το bit αυτό μεταβάλλεται το επίπεδο τάσης της γραμμής DATA και ενεργοποιείται ο παλμός SCK για το απαραίτητο χρονικό διάστημα. Ακολούθως απενεργοποιείται ο παλμός SCK και μετά από την απαραίτητη χρονοκαθυστέρηση αποκατάστασης, η ρουτίνα επαναλαμβάνεται και για τα υπόλοιπα 7 bits. Στη συνέχεια η γραμμή DATA τίθεται σε κατάσταση εισόδου, ενεργοποιείται η αντίστοιχη αντίσταση pull-up, και η γραμμή SCK λαμβάνει την τιμή 1 για το απαραίτητο χρονικό διάστημα. Ανάλογα με το σήμα που λαμβάνεται στη γραμμή DATA επιβεβαιώνεται η ορθότητα της αποστολής. Τελικά ο παλμός SCK απενεργοποιείται, και η συνάρτηση επιστρέφει το bit επιβεβαίωσης [23].

s_read_byte

Η συνάρτηση αυτή διαβάζει ένα byte από το δίαυλο του αισθητήρα και αποστέλλει επιβεβαίωση. Αρχικά ορίζεται η ακίδα DATA ως είσοδος και ενεργοποιείται η αντίστοιχη αντίσταση pull-up. Στη συνέχεια ενεργοποιείται ο παλμός SCK για το απαραίτητο χρονικό διάστημα, και αποθηκεύεται το bit της γραμμής DATA στην κατάλληλη θέση μιας προσωρινής μεταβλητής, όπως ορίζεται από μια μάσκα επιλογής. Στη συνέχεια απενεργοποιείται ο παλμός SCK και μετά την απαραίτητη χρονοκαθυστέρηση η διαδικασία επαναλαμβάνεται για τα υπόλοιπα 7 bit.

Ακολούθως η γραμμής DATA ορίζεται ως έξοδος και λαμβάνει την κατάλληλη τιμή επιβεβαίωσης, όπως ορίζεται από την παράμετρο εισόδου *ack*. Τελικά αποστέλλεται ένας παλμός SCK, η γραμμή DATA ορίζεται ξανά ως είσοδος, και ενεργοποιείται η αντίσταση pull-up. Η συνάρτηση επιστρέφει το byte που διαβάστηκε.

s_transstart

Η συνάρτηση αυτή αποστέλλει τους ειδικούς παλμούς έναρξης της επικοινωνίας με τον αισθητήρα sensirion. Αρχικά η γραμμή SCK ορίζεται ως έξοδος και λαμβάνει την τιμή μηδέν. Στη συνέχεια η γραμμή DATA ορίζεται ως έξοδος και λαμβάνει την τιμή 1. Ακολούθως ενεργοποιείται η γραμμή SCK και η γραμμή DATA λαμβάνει την τιμή μηδέν. Μετά την απαραίτητη χρονοκαθυστέρηση ο παλμός SCK απενεργοποιείται για μικρό χρονικό διάστημα πριν

ενεργοποιηθεί ξανά. Μετά την απαραίτητη χρονοκαθυστέρηση η γραμμή DATA λαμβάνει την τιμή 1 και ο παλμός SCK απενεργοποιείται. Συνολικά η γραμμή SCK αποστέλλει δύο πλήρεις παλμούς όσο η γραμμή DATA βρίσκεται στο λογικό μηδέν. Τελικά η γραμμή DATA ορίζεται πάλι ως είσοδος και ενεργοποιείται η αντίσταση pull-up.

s_connectionreset

Η συνάρτηση αυτή αποστέλλει ένα μήνυμα αρχικοποίησης της επικοινωνίας (reset). Το μήνυμα αποτελείται από 9 παλμούς SCK με το επίπεδο της γραμμής DATA να βρίσκεται στο λογικό 1, ακολουθούμενους από 2 παλμούς SCK με το επίπεδο της γραμμής DATA να βρίσκεται στο λογικό μηδέν, πριν επιστρέψει πάλι στο λογικό 1.

Αρχικά ορίζεται η γραμμή SCK ως έξοδος και τίθεται στο λογικό μηδέν. Η γραμμή DATA ορίζεται ως έξοδος και τίθεται στο λογικό 1. Στη συνέχεια ενεργοποιείται ένας βρόχος 9 επαναλήψεων, σε κάθε μία από τις οποίες από αποστέλλεται ένας παλμός SCK με κατάλληλη χρονοκαθυστέρηση. Μετά την εκτέλεση του βρόχου καλείται η συνάρτηση s_connectionreset που προκαλεί την αποστολή δύο παλμών SCK με το επίπεδο της γραμμής DATA στο λογικό μηδέν, πριν επιστρέψει πάλι στο λογικό 1.

s_softreset

Η συνάρτηση αυτή αποστέλλει ένα μήνυμα soft-reset. Αρχικά καλείται η συνάρτηση s_connectionreset για αρχικοποίηση της επικοινωνίας. Στη συνέχεια καλείται η συνάρτηση s_write_byte με κατάλληλο όρισμα, προκαλώντας την αρχικοποίηση του αισθητήρα.

s_measure

Η συνάρτηση αυτή εκτελεί μετρήσεις θερμοκρασίας και υγρασίας ανάλογα με την παράμετρο εισόδου *mode*, ελέγχοντας παράλληλα το CRC. Αρχικά καλείται η συνάρτηση s_transstart για έναρξη της επικοινωνίας και η παράμετρος *mode* αποκτά την τιμή της κατάλληλης εντολής ανάλογα με τον τύπο της μέτρησης. Στη συνέχεια η τιμή του *mode* αποστέλλεται στον αισθητήρα μέσω της συνάρτησης s_write_byte και αν παρουσιαστεί σφάλμα στη μετάδοση η ρουτίνα επιστρέφει. Ακολούθως η παράμετρος *mode* που περιέχει το byte της εντολής που στείλαμε, αντιστρέφεται με τη βοήθεια της συνάρτησης bitswapbyte, και η προσωρινή μεταβλητή *crc_state* λαμβάνει την τιμή του CRC που αντιστοιχεί στην ανεστραμμένη εντολή της παραμέτρου *mode.*

Στη συνέχεια η ρουτίνα αναμένει εύλογο χρονικό διάστημα μέχρι να ολοκληρωθεί η μέτρηση, μέσω ενός βρόχου αναμονής. Ο βρόχος αυτός μπορεί να διακοπεί όταν ο αισθητήρας είναι έτοιμος και μηδενιστεί το επίπεδο της τάσης της γραμμής DATA. Αν ο αισθητήρας δεν απαντήσει στο προβλεπόμενο χρονικό διάστημα, η ρουτίνα επιστρέφει με κατάλληλο μήνυμα σφάλματος.

Η συνάρτηση είναι πλέον έτοιμη να διαβάσει το περισσότερο σημαντικό byte της μέτρησης. Για το σκοπό αυτό εκτελείται η συνάρτηση s_read_byte. Αφού διαβαστεί το πρώτο byte, η ρουτίνα ανανεώνει την προσωρινή μεταβλητή *crc_state* εκτελώντας την συνάρτηση computeCRC8 με παράμετρο δεδομένων το πρώτο byte της μέτρησης και παράμετρο *seed* την τελευταία τιμή της μεταβλητής *crc_state*. Με αυτό τον τρόπο είναι δυνατός ο υπολογισμός του CRC περισσότερων του ενός bytes.

Στη συνέχεια εκτελείται η συνάρτηση s_read_byte για να διαβαστεί το λιγότερο σημαντικό byte της μέτρησης και πλέον το ζεύγος των bytes (16 bit) δεικτοδοτείται από κατάλληλο δείκτη σε ακέραιο (16 bit) που δόθηκε ως παράμετρος. Ακολούθως η προσωρινή μεταβλητή *crc_state* ανανεώνεται και πάλι εκτελώντας την συνάρτηση computeCRC8 με παράμετρο δεδομένων το δεύτερο byte της μέτρησης και παράμετρο *seed* την τελευταία τιμή της μεταβλητής *crc_state*. Τελικά εκτελείται η συνάρτηση s_read_byte για να διαβαστεί το CRC όπως υπολογίστηκε από τον αισθητήρα και να συγκριθεί το αποτέλεσμα με την μεταβλητή *crc_state*. Ανάλογα με το αποτέλεσμα της σύγκρισης η ρουτίνα επιστρέφει το κατάλληλο μήνυμα.

calc_sth11_temp

Η συνάρτηση αυτή εφαρμόζει τον κατάλληλο μετασχηματισμό στην παράμετρο εισόδου που αντιστοιχεί στην ακατέργαστη μέτρηση *t*, για τον υπολογισμό της θερμοκρασίας. Ο μετασχηματισμός που εφαρμόζεται είναι $t \cdot 0.1 - 397$, όπως προσδιορίζεται από το φύλλο προδιαγραφών [23], ενώ το αποτέλεσμα δίνεται σε δέκατα του βαθμού Κελσίου.

rhcalc_int

Η συνάρτηση αυτή υπολογίζει τη σχετική υγρασία με χρήση γραμμικής προσέγγισης [36]. Η συνάρτηση δέχεται για είσοδο την ακατέργαστη μέτρηση *s*, η οποία βρίσκεται μεταξύ των τιμών 100 και 3340. Αν η τιμή εισόδου είναι μικρότερη από το μέσο του εύρους, τότε η σχετική υγρασία προσεγγίζεται

από τη σχέση ((36·*s*-2048)/102+5)/10, ενώ σε αντίθετη περίπτωση υπολογίζεται από τη σχέση ((14·*s*+5790)/51+5)/10. Τελικά το αποτέλεσμα στρογγυλοποιείται και η ρουτίνα επιστρέφει την τιμή της σχετικής υγρασίας.

calc_sth11_humi

Η συνάρτηση αυτή υπολογίζει την σχετική υγρασία λαμβάνοντας υπόψη την πραγματική θερμοκρασία περιβάλλοντος. Αρχικά καλείται η συνάρτηση rhcalc_int και στο αποτέλεσμα προστίθεται η παράσταση *t*/80-3, όπου η παράμετρος εισόδου *t* αντιστοιχεί στη θερμοκρασία περιβάλλοντος [23].

log10_approx

Η συνάρτηση αυτή προσεγγίζει την τιμή του δεκαδικού λογάριθμου σε εκατοστά, για τιμές εισόδου από 2 έως 100, με σφάλμα μικρότερο του 5%. Αρχικά ελέγχεται αν η παράμετρος εισόδου *x* είναι ίση με 1, οπότε η έξοδος είναι μηδέν. Στη συνέχεια ελέγχεται αν η είσοδος είναι μικρότερη του 8, οπότε η έξοδος είναι ίση με 11·*x*+11. Αν η είσοδος είναι μικρότερη του 36, τότε η έξοδος είναι 98-98/*x*+19·*x*/10-4, ενώ αν η είσοδος είναι μεταξύ 52 και 80 τότε η έξοδος είναι 98-98/*x*+67·*x*/100+42. Σε αντίθετη περίπτωση η έξοδος είναι ισούται με 98-98/*x*+67·*x*/100+39, ενώ για έξοδο μεγαλύτερη από 200, η έξοδος στρογγυλοποιείται στην τιμή 200. Όλες οι παραπάνω προσεγγίσεις υπολογίστηκαν πειραματικά.

calc_dewpoint

Η συνάρτηση αυτή υπολογίζει το σημείο δρόσου (dew point) ανάλογα με τις παραμέτρους εισόδου *t* και *rh* που αναφέρονται στη θερμοκρασία και τη σχετική υγρασία αντίστοιχα. Το σημείο δρόσου υπολογίζεται αναλυτικά με βάση τη σχέση 243.12·[(log10(*rh*) - 2) / 0.4343 + (17.62·*t*) / (243.12 + *t*)] / (17.62 - [(log10(*rh*)-2) / 0.4343 + (17.62·*t*) / (243.12 + *t*)]), την οποία και υλοποιούμε στη ρουτίνα μας. Το σημείο δρόσου αναφέρεται στη θερμοκρασία που πρέπει να ψυχθεί ο αέρας ώστε να επέλθει κορεσμός και υγροποίηση των υδρατμών. Στο σημείο δρόσου η σχετική υγρασία είναι 100% [37].

5.1.7 timeout.c

Το αρχείο timeout.c χρησιμοποιεί τα αρχεία avr/interrupt.h, avr/pgmspace.h, avr/io.h, avr_compat.h και util/delay.h. Η συνάρτηση που περιλαμβάνει είναι η delay_ms:

delay_ms

Η συνάρτηση αυτή προκαλεί χρονοκαθυστέρηση για ορισμένα millisecond, ανάλογα με την παράμετρο εισόδου *ms*. Η λειτουργία της στηρίζεται σε ένα βρόχο που περιλαμβάνει την ειδική μακροεντολή χρονοκαθυστέρησης _delay_ms με παράμετρο εισόδου την τιμή 0.96. Κάθε εκτέλεση του βρόχου προκαλεί καθυστέρηση 1 millisecond και ανάλογα με την παράμετρο *ms* η εκτέλεση επαναλαμβάνεται αναλόγως.

5.1.8 main.c

Το αρχείο main.c είναι το βασικό αρχείο του προγράμματος και χρησιμοποιεί τα αρχεία avr/io.h, stdlib.h, string.h, avr/pgmspace.h, avr/eeprom.h, avr/interrupt.h, analog.h, net.h, lcd.h, ip_arp_udp_tcp.h, enc28j60.h, timeout.h, avr_compat.h, config.h, sensirion_protocol.h, integrator.h και avr/wdt.h. Αρχικά ορίζονται και αρχικοποιούνται οι απαραίτητες global μεταβλητές. Οι συναρτήσεις που περιλαμβάνονται είναι οι ακόλουθες:

get_mcusr

Η συνάρτηση αυτή αποθηκεύει τα περιεχόμενα του καταχωρητή MCU status, και στη συνέχεια τον μηδενίζει, πριν απενεργοποιήσει τον χρονιστή watchdog. Ο χρονιστής watchdog χρησιμοποιείται για την πρόκληση γενικού reset στον μικροελεγκτή σε περίπτωση που το πρόγραμμα βρεθεί σε αόριστη κατάσταση. Αν η ροή του προγράμματος είναι η προβλεπόμενη, ο χρονιστής μηδενίζεται τακτικά μέσα από το πρόγραμμα, ενώ αν το πρόγραμμα για οποιοδήποτε λόγο κολλήσει, τότε ο χρονιστής υπερχειλίζει και προκαλείται reset. Η συνάρτηση αυτή αποθηκεύεται στην περιοχή .init3 της μνήμης flash, ενώ τα περιεχόμενα του καταχωρητή MCU status αποθηκεύονται στην περιοχή .noinit [5].

adddecimalpoint

Η συνάρτηση αυτή τοποθετεί το σημείο της υποδιαστολής σε έναν αριθμό υπό μορφή string, πριν το τελευταίο ψηφίο του. Αρχικά ελέγχεται η παρουσία αρνητικού πρόσημου, το οποίο και παρακάμπτεται, και στη συνέχεια ελέγχεται αν ο αριθμός αποτελείται μόνο από ένα ψηφίο, οπότε και τοποθετείται το ψηφίο μηδέν και η υποδιαστολή μετά από αυτό. Αν ο αριθμός αποτελείται από περισσότερα ψηφία, τότε σαρώνεται ολόκληρος μέχρι να εντοπιστεί το τελευταίο ψηφίο του πριν τον χαρακτήρα τερματισμού. Η συνάρτηση τότε αποθηκεύει το τελευταίο ψηφίο, και στη θέση του αποθηκεύει την

υποδιαστολή. Στην επόμενη θέση αποθηκεύεται το τελευταίο ψηφίο του αρχικού αριθμού και το string ολοκληρώνεται με τον χαρακτήρα τερματισμού.

clock_start

Η συνάρτηση αυτή ενεργοποιεί τον εσωτερικό χρονιστή 16 bit. Αρχικά ο χρονιστής μηδενίζεται και ορίζεται να προκαλεί διακοπή κάθε δευτερόλεπτο πριν μηδενιστεί ξανά. Στη συνέχεια ενεργοποιούνται οι διακοπές και μηδενίζονται οι αντίστοιχες μεταβλητές λεπτών και δευτερολέπτων. Ο χρονιστής αυξάνεται με συχνότητα ίση με το 1/1024 του κεντρικού ρολογιού και η τιμή του συγκρίνεται κάθε φορά με μια προκαθορισμένη μέγιστη τιμή που προβλέπεται ότι θα λάβει μετά από 1 δευτερόλεπτο. Όταν λάβει αυτή την τιμή, προκαλείται διακοπή χρονιστή, και η μέτρηση ξεκινά από την αρχή.

clock_stop

Η συνάρτηση αυτή απενεργοποιεί τον εσωτερικό χρονιστή 16 bit. Αρχικά μηδενίζονται οι σχετικοί καταχωρητές ελέγχου και στη συνέχεια απενεργοποιούνται οι διακοπές. Τελικά οι μεταβλητές λεπτών και δευτερολέπτων μηδενίζονται. Η συνάρτηση αυτή δεν καλείται στο πρόγραμμα αλλά ορίζεται για λόγους πληρότητας.

eeprom_save

Η συνάρτηση αυτή αποθηκεύει τις τιμές ενός πακέτου μετρήσεων στη μνήμη eeprom του μικροελεγκτή. Σε κάθε κλήση της συνάρτησης αυτής υπολογίζονται οι νέες θέσεις αποθήκευσης μέσα στην μνήμη eeprom και καλείται η συνάρτηση βιβλιοθήκης eeprom_write_block, η οποία εγγράφει μία μεταβλητή στην κατάλληλη θέση. Στη συνέχεια αυξάνονται οι κατάλληλοι μετρητές που ορίζουν τις επόμενες θέσεις αποθήκευσης για την επόμενη κλήση της συνάρτησης.

eeprom_load

Η συνάρτηση αυτή διαβάζει τις τιμές ενός πακέτου μετρήσεων από τη μνήμη eeprom του μικροελεγκτή. Σε κάθε κλήση της συνάρτησης αυτής υπολογίζονται οι νέες θέσεις ανάγνωσης και καλείται η συνάρτηση βιβλιοθήκης eeprom_read_block, που μεταφέρει το περιεχόμενο μιας περιοχής της μνήμης eeprom σε μια μεταβλητή. Στη συνέχεια αυξάνονται οι κατάλληλοι μετρητές που ορίζουν τις επόμενες θέσεις ανάγνωσης για την επόμενη κλήση της συνάρτησης.

eeprom_reset

Η συνάρτηση αυτή μηδενίζει τους μετρητές που ορίζουν τις επόμενες θέσεις ανάγνωσης και εγγραφής από την μνήμη eeprom.

print_lcd

Η συνάρτηση αυτή εκτυπώνει την πρώτη από τις δύο διαφάνειες μετρήσεων που παρουσιάζονται στην οθόνη lcd. Αρχικά εκτυπώνεται στην οθόνη το μήνυμα “S:” που αντιστοιχεί την τιμή της τάσης του ηλιακού συλλέκτη, με κλήση της συνάρτησης lcd_puts_p. Το string “S:” βρίσκεται αποθηκευμένο στην περιοχή μνήμης προγράμματος και όχι σε κάποια μεταβλητή για να γίνει εξοικονόμηση μνήμης. Στη συνέχεια διαβάζεται η τιμή του ADC με χρήση της συνάρτησης convertanalog, και μετατρέπεται σε τάση ηλιακού συλλέκτη με χρήση της συνάρτησης voltsolar. Η εκτύπωση στη οθόνη lcd γίνεται με χρήση της συνάρτησης lcd_puts, αφού προηγηθεί η προσθήκη υποδιαστολής με κλήση της συνάρτησης adddecimalpoint. Ακολούθως εκτυπώνεται το μήνυμα “ V ” που αντιστοιχεί στη μονάδα μέτρησης, και η μέτρηση αποθηκεύεται σε κατάλληλη μεταβλητή με σκοπό τη μεταφορά στη μνήμη eeprom αν χρειαστεί.

Στη συνέχεια εκτυπώνεται το μήνυμα “B:” που αντιστοιχεί στην τιμή της τάσης της μπαταρίας. Το string αυτό διαβάζεται από την περιοχή μνήμης προγράμματος, όπως συμβαίνει και με όλα τα στατικά string. Στη συνέχεια καλείται η συνάρτηση readbat του ολοκληρωτή και αφού το αποτέλεσμα αποθηκευτεί προσωρινά, εκτυπώνεται στην οθόνη με προσθήκη υποδιαστολής. Ακολούθως εκτυπώνεται το μήνυμα “ V ” που αντιστοιχεί στη μονάδα μέτρησης και η μέτρηση αποθηκεύεται σε κατάλληλη μεταβλητή με σκοπό τη μεταφορά στη μνήμη eeprom αν χρειαστεί.

Στη συνέχεια καλείται η συνάρτηση lcd_gotoxy και ο δρομέας της οθόνης τοποθετείται στην επόμενη γραμμή, όπου και εκτυπώνεται το μήνυμα “S:” υποδηλώνοντας τη τιμή ρεύματος του ηλιακού συλλέκτη. Η τιμή του ρεύματος λαμβάνεται από την πτώση τάσης πάνω σε γνωστή αντίσταση με χρήση της συνάρτησης convertanalog. Το αποτέλεσμα μετατρέπεται σε ρεύμα ηλιακού συλλέκτη με χρήση της συνάρτησης ampsolar, πριν εκτυπωθεί στην οθόνη μαζί με το μήνυμα “mA”. Η μέτρηση αποθηκεύεται σε κατάλληλη μεταβλητή με σκοπό τη μεταφορά στη μνήμη eeprom αν χρειαστεί.

print_lcd1

Η συνάρτηση αυτή εκτυπώνει την δεύτερη διαφάνεια μετρήσεων στην οθόνη lcd. Αρχικά η οθόνη καθαρίζεται με κλήση της συνάρτησης lcd_clrscr, και αν χρειάζεται, αρχικοποιείται η επικοινωνία του μικροελεγκτή με τον αισθητήρα sensirion μέσω της συνάρτησης s_connectionreset. Η αρχικοποίηση γίνεται μία φορά για κάθε 9 κλήσεις της συνάρτησης. Στη συνέχεια λαμβάνεται η μέτρηση της θερμοκρασίας με κλήση της συνάρτησης s_measure με κατάλληλο όρισμα. Αν δεν παρουσιαστεί σφάλμα, λαμβάνεται η μέτρηση της υγρασίας με κλήση της συνάρτησης s_measure, και πάλι με κατάλληλο όρισμα. Σε περίπτωση σφάλματος η επικοινωνία αρχικοποιείται στην επόμενη κλήση της συνάρτησης.

Ακολούθως εκτυπώνεται στην οθόνη το μήνυμα “T:” και μετά από κλήση της συνάρτησης calc_sth11_temp εκτυπώνεται στην οθόνη η θερμοκρασία περιβάλλοντος με προσθήκη υποδιαστολής. Στη συνέχεια εκτυπώνεται το μήνυμα “’C ” που αντιστοιχεί στις μονάδες, και η μέτρηση αποθηκεύεται σε κατάλληλη μεταβλητή με σκοπό τη μεταφορά στη μνήμη eeprom αν χρειαστεί.

Αμέσως μετά στην οθόνη εκτυπώνεται το μήνυμα “D:” που αντιστοιχεί στο σημείο δρόσου. Αφού μετατραπεί η ακατέργαστη τιμή της υγρασίας με χρήση της συνάρτησης calc_sth11_humi, και αποθηκεύτει σε κατάλληλη μεταβλητή, υπολογίζεται το σημείο δρόσου με κλήση της συνάρτησης calc_dewpoint και ορίσματα την θερμοκρασία και την υγρασία. Ακολούθως γίνεται η εκτύπωση του αποτελέσματος με προσθήκη υποδιαστολής και του μηνύματος “’C” που αντιστοιχεί στις μονάδες. Τελικά και η μέτρηση αποθηκεύεται σε κατάλληλη μεταβλητή με σκοπό τη μεταφορά στη μνήμη eeprom αν χρειαστεί.

Στη συνέχεια ο δρομέας μετακινείται στην επόμενη γραμμή και εκτυπώνεται το μήνυμα “P:” που αντιστοιχεί στην ατμοσφαιρική πίεση στην επιφάνεια της θάλασσας. Αφού διαβαστεί με μετατραπεί η τιμή του ADC πίεσης με χρήση των συναρτήσεων convertanalog και pressure, στη μέτρηση προστίθεται υποδιαστολή πριν αποθηκευτεί σε κατάλληλη μεταβλητή. Ακολούθως καλείται η συνάρτηση pressureATsealevel που υπολογίζει την πίεση στην επιφάνεια της θάλασσας, και το αποτέλεσμα εκτυπώνεται στην οθόνη με προσθήκη υποδιαστολής. Στη συνέχεια εκτυπώνεται το μήνυμα “ hPa nrm” και η μέτρηση αποθηκεύεται σε κατάλληλη μεταβλητή με σκοπό τη μεταφορά στη μνήμη eeprom αν χρειαστεί.

print_xmlpage

Η συνάρτηση αυτή δημιουργεί μια σελίδα xml με δεδομένα μετρήσεων. Αρχικά εγγράφονται στον buffer αποστολής οι απαραίτητες επικεφαλίδες μιας σελίδας xml με κλήση της συνάρτησης fill_tcp_data_p, και αρχικοποιείται η επικοινωνία με τον αισθητήρα θερμοκρασίας και υγρασίας, αν χρειάζεται. Στη συνέχεια διαβάζεται η τιμή της θερμοκρασίας και της υγρασίας, ενώ σε περίπτωση σφάλματος στον buffer προστίθεται το κατάλληλο μήνυμα και τα απαραίτητα xml tags που οριοθετούν τις μετρήσεις. Αν δεν παρουσιαστεί σφάλμα η τιμή της θερμοκρασίας και της υγρασίας τοποθετούνται στον buffer συνοδευόμενες από τα κατάλληλα xml tags, με τη βοήθεια των συναρτήσεων fill_tcp_data και fill_tcp_data_p. Ακολούθως υπολογίζεται το σημείο δρόσου και τοποθετείται και αυτό στον buffer. Στη συνέχεια μετριέται η τάση ηλιακού συλλέκτη και αποθηκεύεται στον buffer μαζί με τα κατάλληλα xml tags. Η τάση της μπαταρίας διαβάζεται από μια προσωρινή μεταβλητή και αποθηκεύεται με τη σειρά της στον buffer του αρχείου xml που δημιουργούμε. Τις μετρήσεις συμπληρώνουν η πραγματική πίεση και η πίεση στην επιφάνεια της θάλασσας, που αποθηκεύονται στον buffer μαζί με τα απαραίτητα xml tags. Τελικά στον buffer προστίθεται το xml tag λήξης, και η συνάρτηση επιστρέφει το μήκος του buffer.

print_xmlpage_eeprom

Η συνάρτηση αυτή δημιουργεί μία σελίδα xml διαβάζοντας κάθε φορά δεδομένα μετρήσεων από τη μνήμη eeprom. Αρχικά ο buffer αποστολής γεμίζει με τις απαραίτητες επικεφαλίδες ενός αρχείου xml, και στη συνέχεια καλείται η συνάρτηση eeprom_load που μεταφέρει ένα πακέτο μετρήσεων από την μνήμη eeprom σε μεταβλητές της μνήμης RAM. Για κάθε μέτρηση, τοποθετείται στον buffer η τιμή της αντίστοιχης μεταβλητής, συνοδευόμενη από τα αντίστοιχα xml tags, που υποδηλώνουν το είδος της μέτρησης. Έτσι αποθηκεύονται στον buffer οι μετρήσεις θερμοκρασίας, υγρασίας, σημείου δρόσου, τάσης ηλιακού συλλέκτη, τάσης μπαταρίας, ρεύματος ηλιακού συλλέκτη, πραγματικής ατμοσφαιρικής πίεσης και πίεσης στην επιφάνεια της θάλασσας. Στη συνέχεια στον buffer τοποθετείται ένα xml tag με την ονομασία <more> που υποδηλώνει αν στην μνήμη eeprom υπάρχουν κι άλλα πακέτα μετρήσεων. Σε περίπτωση που υπάρχουν αποθηκεύεται στον buffer ο αριθμός των υπολειπόμενων πακέτων και η ημερομηνία και ώρα του server, μαζί με τα κατάλληλα xml tags. Τελικά στον buffer προστίθεται το xml tag λήξης, και η συνάρτηση επιστρέφει το μήκος του buffer.

print_webpage

Η συνάρτηση αυτή δημιουργεί στον buffer αποστολής μια σελίδα html με δεδομένα μετρήσεων. Αρχικά ο buffer γεμίζει με τις απαραίτητες επικεφαλίδες ενός αρχείου html, και στη συνέχεια αποθηκεύεται ο τίτλος της ιστοσελίδας με χρήση κατάλληλων html tags. Όπως πάντα όλα τα στατικά string διαβάζονται από την μνήμη προγράμματος. Στη συνέχεια αρχικοποιείται η επικοινωνία με τον αισθητήρα θερμοκρασίας και υγρασίας, αν χρειάζεται, και διαβάζεται η τιμή της θερμοκρασίας και της υγρασίας. Σε περίπτωση σφάλματος στον buffer προστίθεται το κατάλληλο μήνυμα και τα απαραίτητα html tags, αλλιώς εγγράφονται οι τιμές των μετρήσεων θερμοκρασίας, υγρασίας και σημείου δρόσου, μαζί με τις αντίστοιχες μονάδες, επικεφαλίδες και html tags. Όλες οι θερμοκρασίες παρουσιάζονται σε βαθμούς Κελσίου και Φαρενάιτ με ένα δεκαδικό ψηφίο.

Στη συνέχεια προστίθεται στον buffer η μετρούμενη τάση του ηλιακού συλλέκτη, μαζί με την κατάλληλη επικεφαλίδα, τις μονάδες, την υποδιαστολή, και τα αντίστοιχα html tags. Ακολούθως διαβάζεται η τάση της μπαταρίας από μια προσωρινή μεταβλητή, και τοποθετείται και αυτή στον buffer αποστολής με την επικεφαλίδα της, τις μονάδες, την υποδιαστολή, και τα αντίστοιχα html tags. Αμέσως μετά μετριέται το ρεύμα ηλιακού συλλέκτη και η ατμοσφαιρική πίεση, ενώ υπολογίζεται και η πίεση στην επιφάνεια της θάλασσας. Οι τιμές αυτές τοποθετούνται με τη σειρά τους στον buffer, συνοδευόμενες από τις μονάδες τους, τις επικεφαλίδες και τα απαραίτητα html tags. Όλες οι πιέσεις συνοδεύονται από υποδιαστολή.

Τον buffer αποστολής συμπληρώνουν ένα link που πραγματοποιεί refresh στην ιστοσελίδα, και ένα μήνυμα που αναφέρει την έκδοση του προγράμματος του server.

ISR(TIMER1_COMPA_vect)

Η συνάρτηση αυτή υλοποιεί την ρουτίνα εξυπηρέτησης της διακοπής που προκαλείται κάθε δευτερόλεπτο από τον χρονιστή. Η ρουτίνα ελέγχει την τιμή της μεταβλητής *seconds*, και αν δεν ξεπερνά ένα ορισμένο κατώφλι, την αυξάνει κατά 1. Στη συνέχεια θέτει την τιμή 1 στην μεταβλητή *tick* που υποδηλώνει ότι η αύξηση των δευτερολέπτων θα διαβαστεί μόνο μια φορά από το πρόγραμμα.

main

Η συνάρτηση αυτή αποτελεί τη βασική ρουτίνα εκτέλεσης του server. Στη συνέχεια περιγράφουμε αναλυτικά τη λειτουργία της.

Αρχικοποιήσεις

Αρχικά ενεργοποιείται ο χρονιστής watchdog με διάρκεια 1 δευτερόλεπτο, και ορίζεται το εσωτερικό ρολόι του μικροελεγκτή σε συχνότητα 8 MHz. Στη συνέχεια η ακίδα PD2 ορίζεται σαν είσοδος, ενώ η ακίδα PD7 ορίζεται σαν έξοδος. Στην ακίδα PD7 συνδέεται το τρανζίστορ που ελέγχει τον ηλεκτρονόμο, ο οποίος τροφοδοτεί το WiFi. Ακολούθως το WiFi ενεργοποιείται και η μεταβλητή *wifi_status* που αποθηκεύει την κατάσταση του WiFi λαμβάνει την τιμή 1. Αμέσως μετά η ακίδα PB7 που ελέγχει το τρανζίστορ του φωτισμού της οθόνης LCD, ορίζεται σαν έξοδος, και ο φωτισμός απενεργοποιείται.

Στη συνέχεια εκτελείται η ρουτίνα αρχικοποίησης του Ethernet controller με παράμετρο την επιθυμητή διεύθυνση MAC, και το ρολόι του enc28j60 ορίζεται στην συχνότητα 12.5 MHz. Ακολούθως η ακίδα PB6 στην οποία συνδέεται το push button, ορίζεται ως είσοδος και ενεργοποιείται η αντίσταση pull-up. Αμέσως μετά ενεργοποιούνται τα LED λειτουργίας της θύρας Magnetics, και εκτελείται η ρουτίνα αρχικοποίησης της οθόνης lcd. Στη συνέχεια καλείται η ρουτίνα καθαρισμού της οθόνης, και εκτυπώνεται το μήνυμα “=ok=”. Η επόμενη εντολή αρχικοποιεί τα επίπεδα TCP/IP και Ethernet, και αμέσως μετά ενεργοποιείται ο ολοκληρωτής της τάσης της μπαταρίας. Πριν περάσουμε στον κύριο βρόχο, ενεργοποιείται ο χρονιστής διακοπών.

Κύριος βρόχος - έναρξη

Στο σημείο αυτό του προγράμματος ελέγχεται αν έχει ληφθεί κάποιο πακέτο. Αν δεν έχει ληφθεί πακέτο τότε ελέγχεται αν έχει πατηθεί το push button. Αν έχει πατηθεί τότε εκτελείται αρχικοποίηση της οθόνης και ενεργοποιείται ο οπίσθιος φωτισμός της. Η αρχικοποίηση είναι απαραίτητη σε περίπτωση που η οθόνη έχει βρεθεί σε αόριστη κατάσταση. Αν το push button δεν είναι πατημένο, ο φωτισμός απενεργοποιείται.

Κύριος βρόχος – power profile

Στο σημείο αυτό εξετάζεται αν το επίπεδο της τάσης της μπαταρίας είναι μεγαλύτερο από ένα ορισμένο κατώφλι, και ταυτόχρονα η εγγραφή στην μνήμη eeprom είναι απενεργοποιημένη. Αν ισχύει αυτή η συνθήκη πρέπει το

WiFi να είναι ενεργοποιημένο, γι' αυτό ελέγχεται επιπλέον η μεταβλητή κατάστασης *wifi_status*. Αν το WiFi βρεθεί απενεργοποιημένο, τότε ενεργοποιείται και η μεταβλητή *wifi_status* λαμβάνει την τιμή 1.

Στη συνέχεια ελέγχεται αν η τάση της μπαταρίας είναι χαμηλή και ανάμεσα σε δύο κρίσιμα όρια, ενώ ταυτόχρονα η εγγραφή στη μνήμη eeprom είναι απενεργοποιημένη. Στην περίπτωση αυτή χρειάζεται να γίνει εξοικονόμηση ενέργειας. Έτσι η εγγραφή στην μνήμη eeprom ενεργοποιείται και αρχικοποιούνται οι μεταβλητές που ορίζουν τις θέσεις εγγραφής και ανάγνωσης. Στη συνέχεια ελέγχεται αν η ημερομηνία και ώρα που υπάρχει αποθηκευμένη στον server έχει ενημερωθεί τα τελευταία 2 λεπτά. Αν η τελευταία συνθήκη είναι αληθής τότε η μεταβλητή *time_is_valid* λαμβάνει την τιμή 1.

Ακολούθως αρχικοποιούνται στο μηδέν οι μετρητές λεπτών και δευτερολέπτων και ελέγχεται αν το WiFi είναι ενεργοποιημένο. Αν είναι ενεργοποιημένο, τότε απενεργοποιείται για εξοικονόμηση ενεργείας. Στη συνέχεια ελέγχεται αν η εγγραφή στη μνήμη eeprom είναι ενεργοποιημένη. Αν η συνθήκη αυτή ισχύει τότε ελέγχεται αν ο μετρητής δευτερολέπτων έχει λάβει την τιμή 60 για πρώτη φορά. Αυτός ο έλεγχος είναι σημαντικός γιατί ο κώδικας που ελέγχει αυτή τη συνθήκη θέλουμε να εκτελεστεί μόνο μία φορά. Έτσι ελέγχεται επιπλέον η σημαία *tick* που απενεργοποιείται την πρώτη φορά που γίνεται ο έλεγχος, ώστε να μην επαναληφθεί ξανά. Αν λοιπόν τα δευτερόλεπτα έχουν την τιμή 60 και είναι η πρώτη φορά που το ελέγχουμε, τότε απενεργοποιείται η σημαία *tick*, μηδενίζονται τα δευτερόλεπτα και αυξάνονται τα λεπτά. Επίσης, αν η μεταβλητή *time_is_valid* είναι ίση με 1 και το WiFi απενεργοποιημένο, τότε αποθηκεύεται το τελευταίο πακέτο μετρήσεων στη μνήμη eeprom. Η μνήμη eeprom μπορεί να αποθηκεύσει συνολικά 14 πακέτα μετρήσεων, και η αποθήκευση είδαμε ότι μπορεί να γίνει όταν συμπληρώνεται ένα λεπτό.

Στη συνέχεια ελέγχεται αν τα λεπτά έχουν λάβει την τιμή 14. Στην περίπτωση αυτή η μνήμη eeprom δεν μπορεί να αποθηκεύσει άλλες μετρήσεις και επομένως χρειάζεται ενεργοποίηση του WiFi για αποστολή των αποθηκευμένων μετρήσεων στον logger. Έτσι, ενεργοποιείται το WiFi, μηδενίζεται ο δείκτης ανάγνωσης δεδομένων στη μνήμη eeprom και η μεταβλητή *time_is_valid* λαμβάνει την τιμή μηδέν, αφού πλέον η ημερομηνία και ώρα μπορούν να αλλάξουν. Η ημερομηνία και ώρα του server ανανεώνονται κάθε φορά που λαμβάνεται ένα νέο πακέτο από τον logger, και πρέπει να διατηρείται στα-

θερή όσο εγγράφονται οι μετρήσεις στη μνήμη eeprom. Η ενεργοποίηση του WiFi λοιπόν επιτρέπει την αλλαγή της ημερομηνίας και ώρας, και επομένως η εγγραφή στη μνήμη eeprom, που ελέγχεται από τη μεταβλητή *time_is_valid*, πρέπει να σταματήσει. Ο logger έχει τώρα περιθώριο 4 λεπτά για να ανακτήσει τις αποθηκευμένες μετρήσεις. Τελικά ελέγχεται αν τα λεπτά έχουν λάβει την τιμή 18, οπότε και απενεργοποιείται η εγγραφή στη μνήμη eeprom.

Στη συνέχεια ελέγχεται αν το επίπεδο τάσης είναι χαμηλότερο από μία ελάχιστη τιμή. Στην περίπτωση αυτή το WiFi απενεργοποιείται μέχρι η μπαταρία να φορτιστεί ξανά μέσω του ηλιακού συλλέκτη. Παρόλο που το WiFi είναι απενεργοποιημένο, ο server λειτουργεί κανονικά και οι τιμές των μετρήσεων μπορούν να λαμβάνονται από την οθόνη lcd. Στην οριακή περίπτωση που η τάση της μπαταρίας είναι οριακά χαμηλή, τότε επεμβαίνει ο ηλιακός φορτιστής και αποκόπτει συνολικά της τροφοδοσία του συστήματος, για προστασία της μπαταρίας από υπερβολική εκφόρτιση.

Μετά την ολοκλήρωση των ελέγχων τάσης, το πρόγραμμα μηδενίζει τον χρονιστή watchdog. Αυτός ο μηδενισμός συμβαίνει μόνο αν το πρόγραμμα δεν έχει παρουσιάσει δυσλειτουργία μέχρι αυτό το σημείο. Σε αντίθετη περίπτωση ο μηδενισμός του watchdog δεν θα πραγματοποιηθεί και ο μικροελεγκτής θα επανεκκινήσει.

Στις επόμενες εντολές ανανεώνεται η οθόνη. Αν έχει γίνει κατάλληλος αριθμός επαναλήψεων του κύριου βρόχου, τότε ελέγχεται μια μεταβλητή που καθορίζει πιο slide θα παρουσιαστεί στην οθόνη. Στην πρώτη περίπτωση η οθόνη καθαρίζει και καλείται η ρουτίνα παρουσίασης του πρώτου slide μετρήσεων. Επίσης στην κάτω δεξιά γωνία της οθόνης παρουσιάζεται το περιεχόμενο του μετρητή λεπτών του server. Στην δεύτερη περίπτωση παρουσιάζεται το δεύτερο slide μετρήσεων.

Κύριος βρόχος – αποστολή πακέτου

Αν έχει ληφθεί κάποιο πακέτο, τότε απαιτείται η αποστολή απάντησης. Το σύστημά μας είναι σχεδιασμένο μόνο για να απαντά σε αιτήσεις, και δεν έχει τη δυνατότητα να αποστείλει πακέτο αν πρώτα δεν έχει λάβει αίτημα. Αρχικά ελέγχεται αν το πακέτο που λάβαμε είναι πακέτο ARP και προορίζεται για εμάς. Τα πακέτα ARP χρησιμοποιούνται για την εύρεση της διεύθυνσης MAC ενός κόμβου του δικτύου αν γνωρίζουμε την διεύθυνση IP. Έτσι στο

πακέτο αυτό θα απαντήσουμε αποστέλλοντας ένα νέο πακέτο ARP που περιλαμβάνει τη δική μας διεύθυνση MAC. Στη συνέχεια το πρόγραμμα κάνει άλμα στην αρχή του κύριου βρόχου.

Αν το πακέτο που λάβαμε είναι πακέτο IP και απευθύνεται σε εμάς, τότε γίνεται απευθείας άλμα στην αρχή του κύριου βρόχου. Όμως, αν το πακέτο που λάβαμε είναι πακέτο ICMP και περιέχει ένα αίτημα ping, τότε καλείται η κατάλληλη ρουτίνα αποστολής echo reply, και στη συνέχεια γίνεται άλμα στην αρχή του κύριου βρόχου.

Κύριος βρόχος – απάντηση TCP

Αν το πακέτο που λάβαμε είναι τύπου TCP, τότε ελέγχεται αν είναι πακέτο έναρξης της χειραψίας (πακέτο SYN). Σε αυτή τη περίπτωση στέλνουμε απάντηση SYN-ACK, που αποτελεί το δεύτερο στάδιο της χειραψίας, και γίνεται άλμα στην αρχή του κύριου βρόχου, περιμένοντας να λάβουμε το πακέτο τύπου ACK που θα ολοκληρώσει τη χειραψία.

Αν το πακέτο που λάβαμε είναι τύπου ACK, τότε αρχικοποιείται ο buffer και λαμβάνεται ένας δείκτης στην περιοχή δεδομένων του πακέτου TCP. Αν το πακέτο δεν περιέχει δεδομένα τότε είναι τύπου FIN-ACK για τερματισμό της χειραψίας, και πρέπει να απαντήσουμε με ένα πακέτο τύπου ACK. Αφού αποσταλεί το πακέτο ACK για τερματισμό της επικοινωνίας, γίνεται άλμα στην αρχή του κύριου βρόχου, αναμένοντας νέα πακέτα.

Όμως αν το πακέτο τύπου ACK που λάβαμε περιέχει δεδομένα, τότε ελέγχουμε αν αυτά ξεκινούν με την εντολή GET του πρωτοκόλλου HTTP. Στην περίπτωση που δεν ξεκινούν με την εντολή GET, δημιουργούμε στον buffer την κατάλληλη απάντηση HTTP και αποστέλλουμε πρώτα ένα πακέτο τύπου ACK χωρίς δεδομένα, και στη συνέχεια ένα πακέτο τύπου ACK, με τα περιεχόμενα του buffer για δεδομένα.

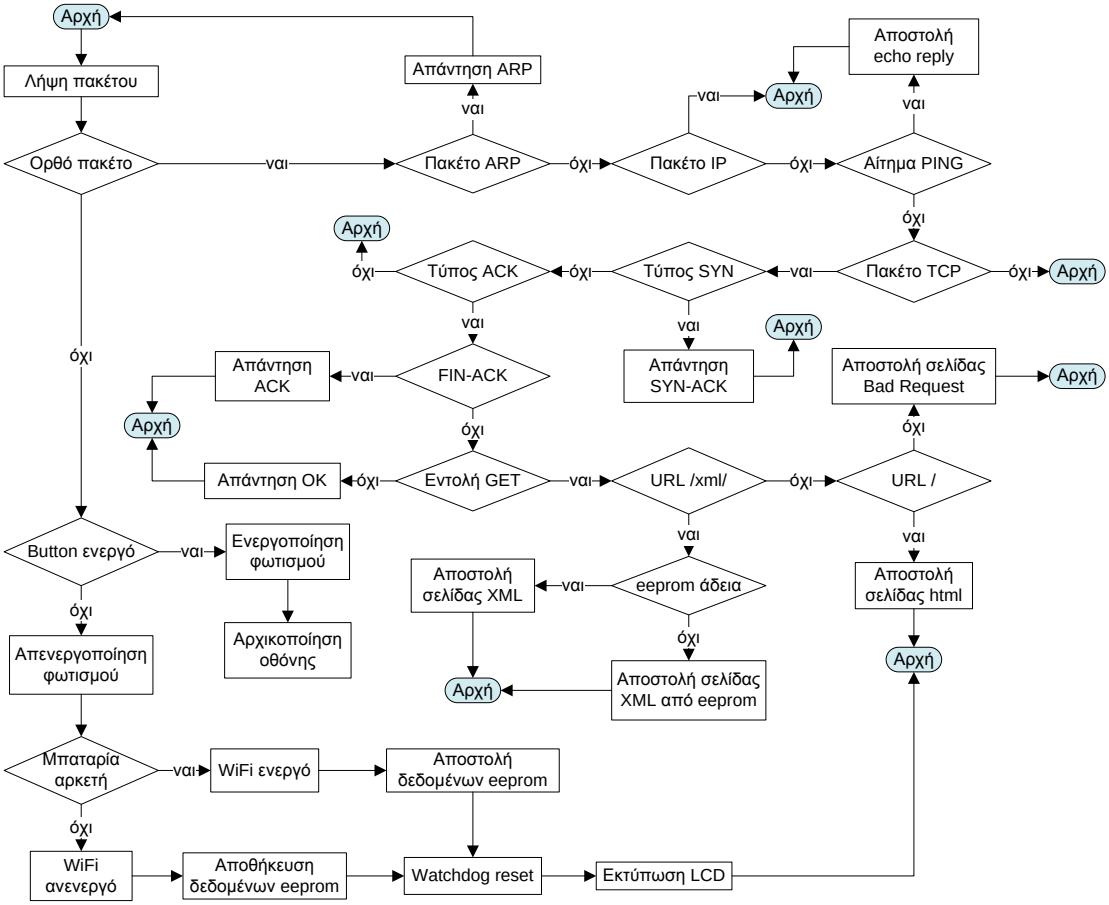
Αν τα δεδομένα του πακέτου ξεκινούν με την εντολή GET, τότε ελέγχουμε το URL που ζητήθηκε. Αν ζητήθηκε το “/xml/” σημαίνει ότι ο logger θέλει να διαβάσει τη σελίδα XML με τα δεδομένα των μετρήσεων. Στην περίπτωση αυτή αναμένεται ότι το URL περιλαμβάνει και ένα string με την τρέχουσα ημερομηνία και ώρα. Έτσι, αφού πρώτα διαβάσουμε την ημερομηνία και ώρα από το κατάλληλο τμήμα του URL, ελέγχουμε αν η εγγραφή στη μνήμη eeprom είναι ενεργοποιημένη. Αν δεν είναι ενεργοποιημένη τότε μηδενίζουμε τη μεταβλητή δευτερολέπτων, αφού η ώρα του server ανανεώθηκε.

Ακολούθως ελέγχεται αν η μνήμη eeprom έχει πακέτα μετρήσεων που δεν έχουν αποσταλεί στον logger, και αν υπάρχουν δημιουργεί στον buffer μια σελίδα XML με μετρήσεις από τη μνήμη eeprom. Διαφορετικά η σελίδα XML που δημιουργείται στον buffer περιέχει τις τρέχουσες τιμές μετρήσεων.

Αν το URL που ζητήθηκε είναι το πηγαίο (root directory "/"), τότε στον buffer δημιουργείται μια κανονική ιστοσελίδα html, που περιέχει όλες τις μετρήσεις. Αν το URL που ζητήθηκε δεν ανήκει σε καμία από τις δύο προηγούμενες περιπτώσεις, τότε στον buffer δημιουργείται μια ιστοσελίδα html τύπου "Bad Request".

Τελικά αποστέλλουμε πρώτα ένα πακέτο τύπου ACK χωρίς δεδομένα, και στη συνέχεια ένα πακέτο τύπου ACK, με τα περιεχόμενα του buffer για δεδομένα. Στη συνέχεια ο κύριος βρόχος επαναλαμβάνεται διαρκώς.

Στο σχήμα που ακολουθεί παρουσιάζεται ένα block διάγραμμα του κύριου βρόχου του προγράμματος.



Εικόνα 57: Διάγραμμα καταστάσεων του κύριου βρόχου

Κεφάλαιο 6
Logger

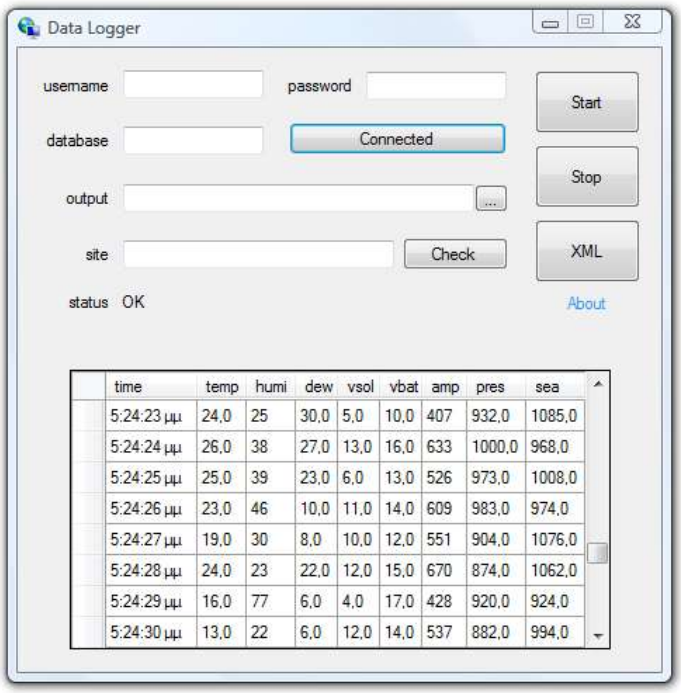
Ο καταγραφέας δεδομένων (logger) εξυπηρετεί στη διαρκή καταγραφή και αποθήκευση των μετρήσεων του server. Αναλυτικά οι βασικές λειτουργίες του logger είναι οι ακόλουθες:

- Σύνδεση με βάση δεδομένων
- Λήψη μετρήσεων από τον server κάθε λεπτό σε μορφή XML
- Λήψη μετρήσεων αποθηκευμένων στην μνήμη eeprom του server
- Αποθήκευση μετρήσεων σε βάση δεδομένων
- Δημιουργία αρχείων XML για αποθήκευση του πίνακα μετρήσεων ώστε να είναι προσπελάσιμος από τα διαγράμματα παρουσίασης των δεδομένων
- Αποστολή στον server της τρέχουσας ημερομηνίας και ώρας

6.1 Ο κώδικας του logger

Στην ενότητα αυτή θα περιγράψουμε τις υπορουτίνες που συνθέτουν το λογισμικό του logger. Ο κώδικας είναι γραμμένος σε γλώσσα Visual Basic και η εφαρμογή αποτελείται συνολικά από δύο φόρμες, οι οποίες παρουσιάζονται στα ακόλουθα σχήματα.

Εικόνα 58: Το βοηθητικό παράθυρο του logger



Εικόνα 59: Το κύριο παράθυρο του logger

Στην κεντρική φόρμα αρχικά βλέπουμε τα πεδία username, password και database. Σε αυτά τα πεδία ο χρήστης καλείται να τοποθετήσει το όνομα χρήστη, τον κωδικό και τη διεύθυνση IP της βάσης δεδομένων αντίστοιχα. Στο πεδίο output ο χρήστης επιλέγει τη διεύθυνση που θα αποθηκεύονται τα αρχεία XML για να διαβαστούν από τα διαγράμματα παρουσίασης των δεδομένων, όπως θα δούμε στο επόμενο κεφάλαιο. Επίσης στο πεδίο site πρέπει να τοποθετηθεί το URL του server και συγκεκριμένα το URL που επιστρέφει τις μετρήσεις με μορφή XML.

Πατώντας το κουμπί Connect ο logger συνδέεται με τη βάση δεδομένων, ενώ πατώντας το κουμπί Check ο logger ελέγχει τη σύνδεση με τον server. Το κουμπί Start ξεκινά την αυτόματη καταγραφή και αποθήκευση των μετρήσεων στη βάση δεδομένων και στα αρχεία XML, ενώ το κουμπί Stop σταματά την καταγραφή. Το κουμπί XML αποθηκεύει τον τρέχοντα πίνακα μετρήσεων στα αρχεία XML, ενώ στο κατώτερο τμήμα της φόρμας, αφού γίνει η σύνδεση με τη βάση, παρουσιάζεται ο πίνακας των μετρήσεων. Όταν λαμβάνονται νέες μετρήσεις ο πίνακας ανανεώνεται.

Πατώντας ο χρήστης στην επιγραφή About, εμφανίζεται η δεύτερη φόρμα που περιέχει οδηγίες χρήσης του προγράμματος. Επίσης αφήνοντας τον δείκτη του mouse πάνω από τα κουμπιά ή τα κενά πεδία της κύριας φόρμας, εμφανίζονται οδηγίες με μορφή tooltip.

Στη συνέχεια παρουσιάζονται αναλυτικά οι υπορουτίνες του logger.

cmdConnect_Click

Η υπορουτίνα αυτή υλοποιεί τη σύνδεση με τη βάση δεδομένων MySQL και τη δημιουργία του πίνακα μετρήσεων. Αρχικά αποθηκεύεται το connection string και δημιουργείται ένα νέο αντικείμενο τύπου MySqlConnection με παράμετρο το δεδομένο connection string. Στη συνέχεια καλείται η μέθοδος open του αντικειμένου που μόλις δημιουργήθηκε, και η σύνδεση ολοκληρώνεται. Με την επιτυχία της σύνδεσης το κουμπί Connect αλλάζει σε Connected και η επιγραφή status λαμβάνει την τιμή OK. Σε περίπτωση σφάλματος σύνδεσης, το κουμπί Connect δεν μεταβάλλεται ενώ η επιγραφή status αναγράφει το σφάλμα που προέκυψε.

Στη συνέχεια δημιουργείται ένα νέο αντικείμενο τύπου MySqlDataAdapter με παράμετρο ένα κατάλληλο query που επιλέγει όλες τις μετρήσεις, και περνά ως παράμετρος στον κατασκευαστή ενός νέου αντικειμένου τύπου MySqlCommandBuilder. Ακολούθως οι εγγραφές που προκύπτουν από την εφαρμογή του query στη βάση δεδομένων παρουσιάζονται στο datagrid που υπάρχει στην κεντρική φόρμα της εφαρμογής. Στη συνέχεια εκτελούνται οι κατάλληλες εντολές ορισμού του πλάτους κάθε στήλης του πίνακα του datagrid, και ορίζεται ποιες εγγραφές θα είναι ορατές. Αν δεν παρουσιαστεί σφάλμα η επιγραφή status αναγράφει το μήνυμα OK, διαφορετικά αναγράφεται το μήνυμα σφάλματος που παρουσιάστηκε.

writeTableToXML

Η υπορουτίνα αυτή εγγράφει τα δεδομένα του datagrid της κεντρικής φόρμας της εφαρμογής, σε ένα αρχείο XML με ορισμένο τρόπο, ώστε να διαβάζεται από τα διαγράμματα παρουσίασης των δεδομένων. Αρχικά ελέγχεται αν το πλήθος των εγγραφών στο datagrid ξεπερνά το πλήθος των ζητούμενων εγγραφών στο αρχείο XML, και σε αντίθετη περίπτωση το ζητούμενο πλήθος εξισώνεται με το πλήθος των εγγραφών που υπάρχουν.

Το κάθε αρχείο XML κατασκευάζεται με τη βοήθεια ενός νέου αντικειμένου τύπου SteamWriter και αποθηκεύεται στη θέση που δίνεται στο πεδίο output με τη επέκταση .xml. Στη συνέχεια αποθηκεύονται στο κάθε νέο αρχείο οι απαραίτητες επικεφαλίδες, και αμέσως μετά αποθηκεύονται με κατάλληλο τρόπο τα δεδομένα των μετρήσεων. Τα δεδομένα που αποθηκεύονται αφορούν την θερμοκρασία, την υγρασία, το σημείο δρόσου, την τάση ηλιακού συλλέκτη, την τάση μπαταρίας, την ισχύ του ηλιακού συλλέκτη, την ατμοσφαιρική πίεση και την πίεση στην επιφάνεια της θάλασσας. Η παράμε-

τρος εισόδου *samples* ορίζει πόσες εγγραφές, ξεκινώντας από το τέλος, θα αποθηκευτούν στο αρχείο XML, ενώ η παράμετρος εισόδου *period* ορίζει το βήμα με το οποίο επιλέγονται οι εγγραφές. Έτσι για παράμετρο εισόδου *period* ίση με 1 επιλέγονται διαδοχικές εγγραφές, ενώ για τιμές μεγαλύτερες του 1 δεν επιλέγονται διαδοχικές εγγραφές.

cmdXML_Click

Η υπορουτίνα αυτή καλεί την υπορουτίνα writeTableToXML με κατάλληλες παραμέτρους εισόδου. Η πρώτη κλήση αφορά τα δεδομένα των τελευταίων 30 λεπτών και αποθηκεύεται μία εγγραφή για κάθε λεπτό. Η δεύτερη κλίση αφορά τα δεδομένα της τελευταίας ημέρας, και αποθηκεύεται μία εγγραφή για κάθε 20 λεπτά. Η τρίτη κλίση αφορά τα δεδομένα των τριών τελευταίων ημερών, και αποθηκεύεται μία εγγραφή για κάθε μία ώρα.

cmdPath_Click

Η υπορουτίνα αυτή ανοίγει ένα παράθυρο διαλόγου και δίνει τη δυνατότητα στο χρήστη να επιλέξει εύκολα το φάκελο αποθήκευσης των αρχείων XML. Ο νέος φάκελος προορισμού τοποθετείται στο πεδίο output.

cmdStart_Click

Η υπορουτίνα αυτή ενεργοποιεί τον χρονιστή του συστήματος, που προκαλεί μία διακοπή κάθε λεπτό, ώστε να διαβάζεται ένα νέο πακέτο μετρήσεων από το server.

Timer1_Tick

Η υπορουτίνα αυτή καλείται σε κάθε κτύπο του χρονιστή του συστήματος, και αποθηκεύει νέα δεδομένα μετρήσεων στη βάση δεδομένων. Αρχικά καλείται η υπορουτίνα cmdCheck_Click, η οποία ελέγχει την επικοινωνία με τον server, διαβάζει τη σελίδα XML με τις μετρήσεις και μεταφέρει τα δεδομένα σε κατάλληλες μεταβλητές. Στη συνέχεια δημιουργείται ένα νέο string που περιέχει το query εισαγωγής δεδομένων στη βάση με όλες τις νέες τιμές που διαβαστήκαν. Ακολούθως δημιουργείται ένα νέο αντικείμενο τύπου MySqlCommand με παράμετρο κατασκευαστή το query εισαγωγής δεδομένων. Αφού το νέο αντικείμενο συνδεθεί με τη βάση δεδομένων μέσω της μεθόδου Connect, γίνεται η εισαγωγή των δεδομένων σε μια νέα εγγραφή μέσω της μεθόδου ExecuteNonQuery. Στη συνέχεια καλείται η υπορουτίνα cmdXML_Click για την ανανέωση των περιεχομένων των αρχείων XML, και αμέσως μετά ανανεώνονται και τα περιεχόμενα του datagrid. Αν δεν παρου-

σιαστεί σφάλμα η επιγραφή status λαμβάνει την τιμή OK, ενώ σε αντίθετη περίπτωση εκτυπώνεται το μήνυμα σφάλματος.

Σε περίπτωση που η σελίδα XML που διαβάστηκε από τον server περιέχει μετρήσεις που προέρχονται από τη μνήμη eeprom, τότε ο χρονιστής του συστήματος σταματά και ενεργοποιείται ένα βοηθητικός χρονιστής με συχνότητα διακοπών πολύ μεγαλύτερη από αυτή του κεντρικού χρονιστή. Αυτός ο βοηθητικός χρονιστής θα χρησιμεύσει για την γρήγορη λήψη των μετρήσεων που προέρχονται από την μνήμη eeprom, σε αντίθεση με τον κεντρικό χρονιστή που χρησιμεύει για την τακτική λήψη των άμεσων μετρήσεων κάθε ένα λεπτό.

cmdStop_Click

Η υπορουτίνα αυτή απενεργοποιεί το χρονιστή του συστήματος, και σταματά η λήψη μετρήσεων από το server.

cmdCheck_Click

Η υπορουτίνα αυτή ελέγχει την επικοινωνία με τον server, διαβάζει τη σελίδα XML με τις μετρήσεις και μεταφέρει τα δεδομένα σε κατάλληλες μεταβλητές. Αρχικά δημιουργείται ένα νέο αντικείμενο τύπου XmlReader με κατάλληλα settings. Στη συνέχεια καλείται η μέθοδος Create του νέου μας αντικειμένου, με παράμετρο εισόδου το URL της σελίδας XML με τις μετρήσεις του server, και την τρέχουσα ημερομηνία και ώρα. Ακολούθως το αντικείμενό μας διαβάζει τη σελίδα XML και αποθηκεύει την τιμή κάθε μέτρησης στην αντίστοιχη μεταβλητή. Επίσης εξετάζεται αν η σελίδα XML περιλαμβάνει xml tags που αναφέρονται σε δεδομένα αποθηκευμένα στη μνήμη eeprom. Στην περίπτωση αυτή διαβάζεται η ημερομηνία και ώρα του server που σχετίζεται με τις μετρήσεις, καθώς και ο αριθμός των υπολειπόμενων πακέτων μετρήσεων στη μνήμη eeprom.

Αν δεν παρουσιαστεί σφάλμα το κουμπί Check αλλάζει σε Checked και η επιγραφή status εμφανίζει το μήνυμα OK. Σε αντίθετη περίπτωση το κουμπί Check δεν μεταβάλλεται, και η επιγραφή status τυπώνει το αντίστοιχο μήνυμα σφάλματος.

lblAbout_Click

Η υπορουτίνα αυτή εμφανίζει τη δεύτερη φόρμα της εφαρμογής μας, με χρήσιμες πληροφορίες και οδηγίες.

Timer2_Tick

Η υπορουτίνα αυτή εξυπηρετεί την ταχεία αποθήκευση των δεδομένων της μνήμης eeprom που στέλνει ο server. Αρχικά καλείται η υπορουτίνα cmdCheck_Click, η οποία ελέγχει την επικοινωνία με τον server, διαβάζει τη σελίδα XML με τις μετρήσεις της μνήμης eeprom και μεταφέρει τα δεδομένα σε κατάλληλες μεταβλητές. Στη συνέχεια δημιουργείται το query εισαγωγής δεδομένων στη βάση με όλες τις νέες τιμές που διαβαστήκαν και με το αντίστοιχο timestamp ανάλογα με την ημερομηνία και ώρα που έστειλε πίσω ο server, ενώ η θέση της κάθε μέτρησης καθορίζεται από την χρονική αλληλουχία. Όταν ο server αποθηκεύει τις μετρήσεις στη μνήμη eeprom κρατά μια σταθερή ημερομηνία και ώρα για όλες. Η ημερομηνία και ώρα της κάθε μίας ξεχωριστά μπορεί να υπολογιστεί, αφού γνωρίζουμε ότι η συχνότητα των μετρήσεων είναι κάθε ένα λεπτό. Έτσι όταν διαβάζεται η σελίδα XML του server, η πρώτη σειρά μετρήσεων αντιστοιχεί στην ημερομηνία και ώρα που περιέχεται στη σελίδα XML, ενώ κάθε επόμενη σειρά μετρήσεων συνέβη τόσα λεπτά αργότερα όσα και ο αύξων αριθμός της σειράς μετρήσεων.

Αφού υπολογιστεί το query εισαγωγής, δημιουργείται ένα νέο αντικείμενο τύπου MySqlCommand, το οποίο και συνδέεται με τη βάση δεδομένων μέσω της μεθόδου Connect. Η εισαγωγή των δεδομένων γίνεται μέσω της μεθόδου ExecuteNonQuery, ενώ αμέσως μετά καλείται η υπορουτίνα cmdXML_Click για την ανανέωση των περιεχομένων των αρχείων XML. Στη συνέχεια ανανεώνονται τα περιεχόμενα του datagrid και αν δεν έχει παρουσιαστεί σφάλμα η επιγραφή status λαμβάνει την τιμή OK. Σε αντίθετη περίπτωση η επιγραφή status εκτυπώνει το μήνυμα σφάλματος. Επίσης, αν έχουν διαβαστεί όλα τα περιεχόμενα της μνήμης eeprom του server, ο βοηθητικός χρονιστής απενεργοποιείται, και ενεργοποιείται ξανά ο βασικός χρονιστής της εφαρμογής μας.

Κεφάλαιο 7
MySQL και homepage

7.1 MySQL και πακέτο xampp

Ο εξυπηρετητής MySQL αποτελεί ένα ακόμα κομμάτι του συστήματός μας, αφού εξασφαλίζει την αποθήκευση και ανάκτηση όλων μετρήσεων που καταγράφηκαν. Στο σύστημά μας χρησιμοποιούμε το ολοκληρωμένο πακέτο xampp, που εκτός από τον MySQL server, περιλαμβάνει τη γλώσσα προγραμματισμού php καθώς και τον πολύ χρήσιμο Apache web server, που θα χρησιμοποιήσουμε για την παρουσίαση μιας ιστοσελίδας με συγκεντρωτικές μετρήσεις, όπως θα δούμε στη συνέχεια. Ένα ακόμα πολύ χρήσιμο εργαλείο του πακέτου xampp είναι ο διαχειριστής βάσεων δεδομένων phpMyAdmin. Με τον διαχειριστή phpMyAdmin μπορούμε να έχουμε πρόσβαση στον MySQL server μέσω ενός εύχρηστου περιβάλλοντος διαχείρισης. Στη συνέχεια παρουσιάζονται τα χαρακτηριστικά της βάσης δεδομένων που υλοποιήσαμε για το σύστημά μας.

Πεδίο	Τύπος	Χαρακτηριστικά	Κενό	Προκαθορισμένο
time	timestamp	ON UPDATE CURRENT_TIMESTAMP	Όχι	CURRENT_TIMESTAMP
temp	decimal(3,1)		Ναι	NULL
humi	tinyint(3)		Ναι	NULL
dew	decimal(3,1)		Ναι	NULL
vsol	decimal(3,1)		Ναι	NULL
vbat	decimal(3,1)		Ναι	NULL
amp	smallint(3)		Ναι	NULL
pres	decimal(5,1)		Ναι	NULL
sea	decimal(5,1)		Ναι	NULL

Δηλώσεις	Τιμή
Μορφοποίηση	προκαθορισμένου μήκους
Collation	ascii_general_ci
Μέγεθος Γραμμής	23
Μέγεθος Εγγραφής	35 Bytes

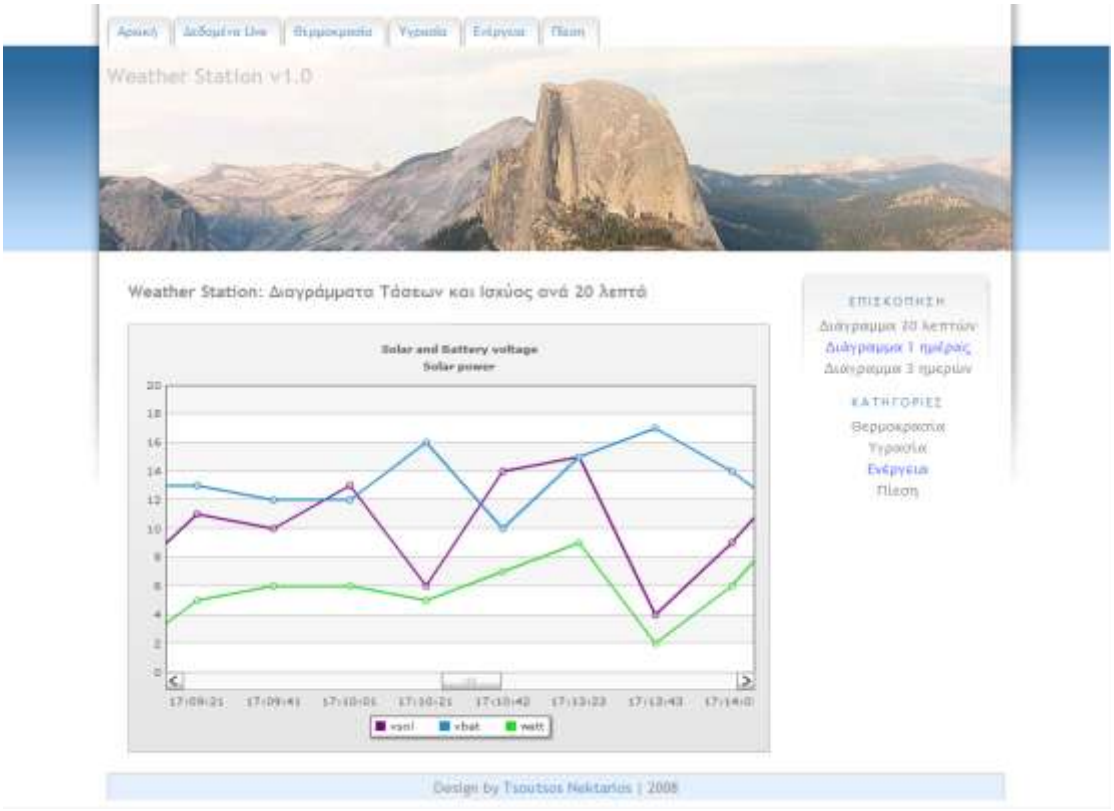
Η βάση δεδομένων λέγεται *weather* και ο πίνακας που αποθηκεύονται οι μετρήσεις λέγεται *webdata*. Το πρωτεύον κλειδί είναι το πεδίο time, και είναι τύπου timestamp. Το πεδίο αυτό έχει το χαρακτηριστικό ότι όταν ανανεώνεται ή δημιουργείται μία συγκριμένη εγγραφή και δεν δίνεται τιμή στο πεδίο time, η τιμή του πεδίο ισούται με την τρέχουσα ημερομηνία και ώρα. Παρόλα αυτά όταν δίνεται τιμή στο πεδίο time, τότε υπερισχύει η δοσμένη τιμή, και όχι η τρέχουσα. Με αυτό τον τρόπο είναι δυνατή η αποθήκευση των δεδομένων της μνήμη eeprom με διαφορετικές τιμές στο πεδίο time από την τρέχουσα ώρα. Επίσης, επειδή το πεδίο αυτό είναι και το πρωτεύον κλειδί του πίνακα, κάθε τιμή πρέπει να είναι μοναδική. Η ακρίβεια του πεδίου είναι ίση με ακρίβεια δευτερολέπτου, και έτσι δεν είναι δυνατή η μεταβολή ή αποθήκευση δύο εγγραφών ταυτόχρονα αν δεν δίνεται διαφορετική τιμή στα πεδία time της κάθε μίας.

Επίσης αξίζει να σημειωθεί η ιδιαίτερη έμφαση που έχει δοθεί στο μήκος του κάθε πεδίου. Μια εγγραφή έχει μέγεθος 35 byte και έτσι υπάρχει η δυνατότητα αποθήκευσης χιλιάδων εγγραφών χωρίς ιδιαίτερη επιβάρυνση του αποθηκευτικού χώρου.

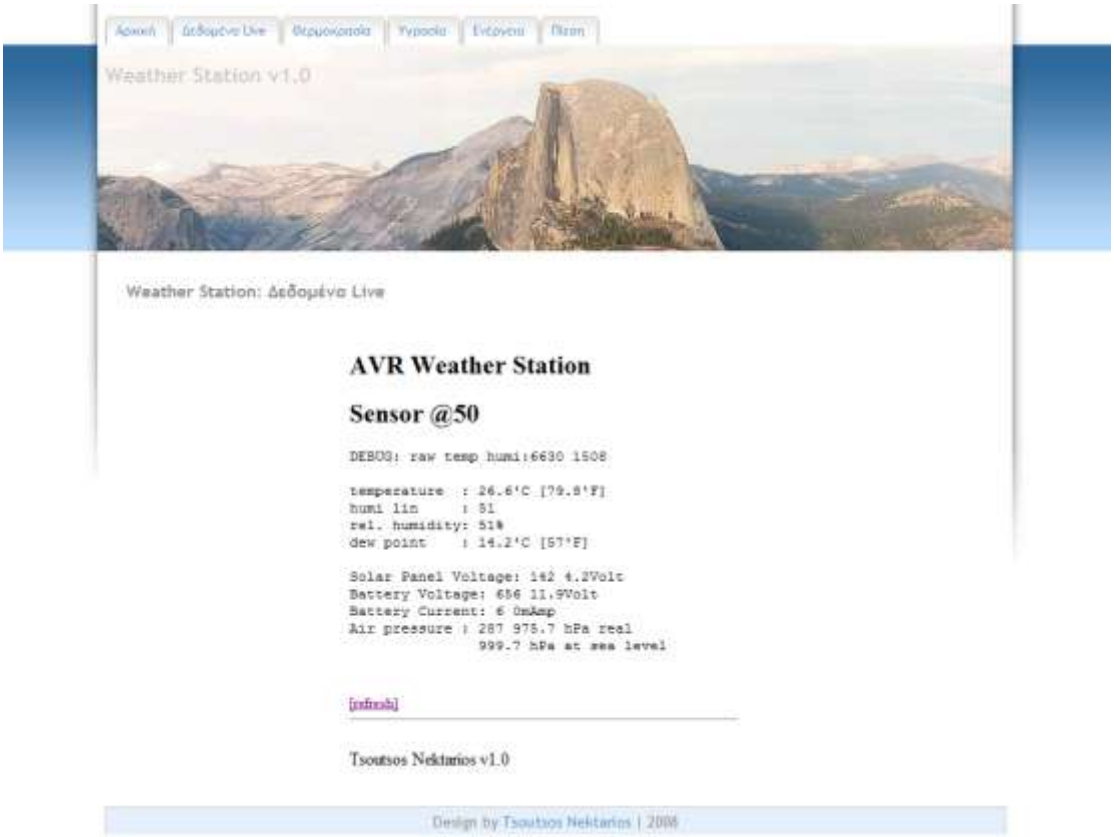
Στη συνέχεια θα ασχοληθούμε με την ιστοσελίδα παρουσίασης των αποτελεσμάτων με μορφή γραφημάτων.

7.2 homepage

Όπως είδαμε προηγουμένως, ο server διαθέτει τη δυνατότητα να δημιουργεί και να αποστέλλει τις μετρήσεις σε μορφή ιστοσελίδας. Το μειονέκτημα όμως του server είναι ότι μπορεί να υποστηρίξει ιστοσελίδες πολύ μικρού μεγέθους (που χωρούν σε ένα μόλις πακέτο TCP, αφού δεν υποστηρίζεται fragmentation) και χωρίς γραφικά. Για να υπερνικήσουμε αυτό τον περιορισμό δημιουργήσαμε μία πλήρη ιστοσελίδα παρουσίασης των μετρήσεων με μορφή γραφημάτων, που εξυπηρετείται από τον Apache web server που περιλαμβάνεται στο πακέτο xampp. Η ιστοσελίδα αυτή (homepage) επεκτείνει τη λειτουργία της ιστοσελίδας του server, αφού δεν περιορίζεται από την ανάγκη εξοικονόμησης ενέργειας και απενεργοποίησης του WiFi. Επίσης δίνει τη δυνατότητα παρουσίασης ενός μεγάλου αριθμού μετρήσεων, όπως το σύνολο των μετρήσεων των τελευταίων 30 λεπτών, της τελευταίας ημέρας (μία μέτρηση κάθε 20 λεπτά), καθώς και των τελευταίων τριών ημερών (μία μέτρηση κάθε 60 λεπτά). Με αυτό τον τρόπο λαμβάνεται μια πλήρης εικόνα των μετρήσεων και της εξέλιξής τους στο χρόνο.



Εικόνα 60: Screenshot της ιστοσελίδας



Εικόνα 61: Screenshot της ιστοσελίδας

FusionCharts

Η παρουσίαση των μετρήσεων όπως προαναφέραμε γίνεται με χρήση διαγραμμάτων τεχνολογίας flash, που εκτός από τη δυνατότητα animation στις καμπύλες, προσφέρουν τη δυνατότητα διαδραστικότητας με το χρήστη. Τα διαγράμματα αυτά διαβάζουν τα δεδομένα από κατάλληλα αρχεία XML, τα οποία δημιουργεί και ανανεώνει διαρκώς ο logger. Επίσης μία από τις ιστοσελίδες που προσφέρονται περιλαμβάνει ένα εσωτερικό παράθυρο που παρουσιάζει την ιστοσελίδα που στέλνει ο server, όταν είναι ενεργοποιούμενο το WiFi.

Τα διαγράμματα που χρησιμοποιήσαμε ονομάζονται FusionCharts, και είναι σχεδιασμένα ώστε να παραμετροποιούνται σε μεγάλο βαθμό από το χρήστη, μέσω του ιδίου αρχείου XML που περιλαμβάνει και τα δεδομένα που παρουσιάζουν. Στην υλοποίησή μας χρησιμοποιήσαμε τις εξής παραμέτρους:

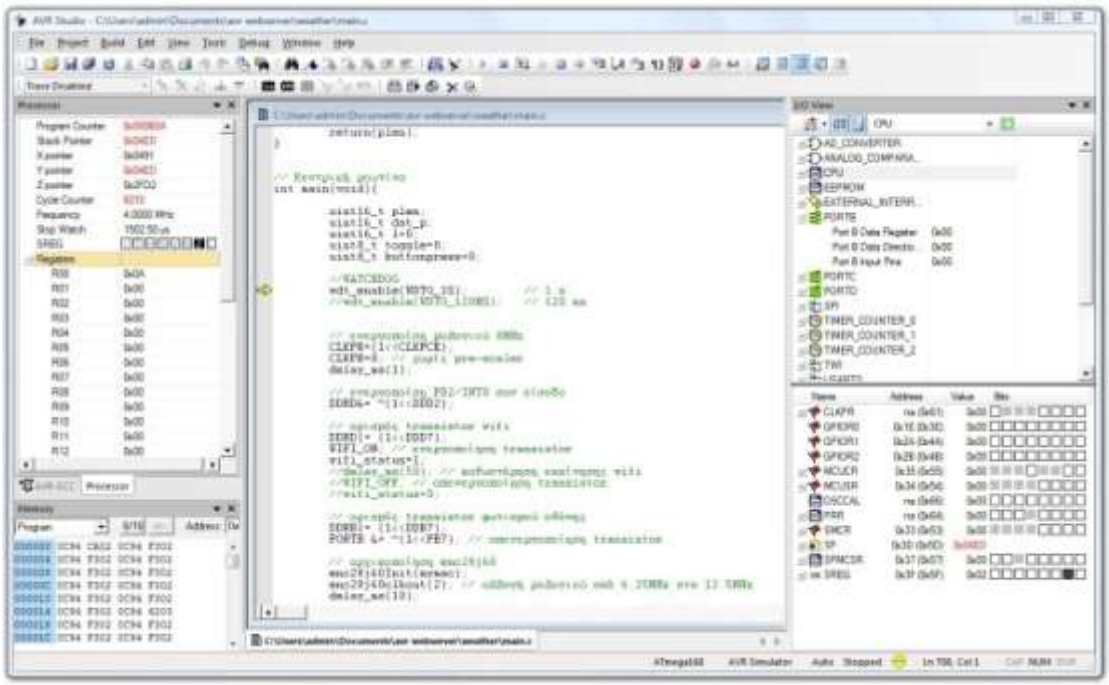
- *caption,* για τον τίτλο του διαγράμματος
- *subCaption,* για τον υπότιτλο του διαγράμματος
- *numdivlines,* για τον αριθμό των οριζόντιων υποδιαιρέσεων
- *lineThickness,* που ορίζει το πάχος της καμπύλης
- *showValues,* που ορίζει αν θα εκτυπώνονται οι τιμές της καμπύλης
- *animation,* για τον ορισμό κίνησης στις καμπύλες
- *inDecimalSeparator,* για τον ορισμό του συμβόλου υποδιαστολής
- *setAdaptiveYMin,* για την προσαρμογή του ελάχιστου στον κάθετο άξονα
- *decimals,* για το πλήθος των δεκαδικών ψηφίων
- *seriesName,* που ορίζει το όνομα της καμπύλης
- *color,* που ορίζει το χρώμα της καμπύλης

Η εισαγωγή των διαγραμμάτων σε κάθε ιστοσελίδα γίνεται παρόμοια με την εισαγωγή ενός απλού αρχείου flash. Η αποθήκευση των αρχείων XML γίνεται στον ίδιο φάκελο που αποθηκεύονται τα αρχεία των ιστοσελίδων, και η ανανέωση των διαγραμμάτων γίνεται αυτόματα με την χρήση του refresh του browser.

Κεφάλαιο 8
Software tools

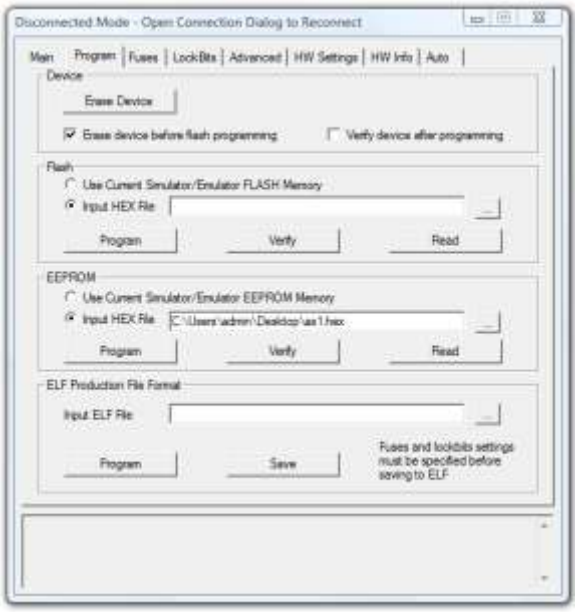
8.1 AVRstudio

Το περιβάλλον AVRstudio της Atmel είναι ένα IDE που προορίζεται για την ανάπτυξη λογισμικού σε μικροελεγκτές τύπου AVR. Εκτός από τον εξελιγμένο editor, περιλαμβάνει και εξομοιωτή (emulator) για την διόρθωση πιθανών σφαλμάτων στα προγράμματα. Ο εξομοιωτής δίνει στο χρήστη τη δυνατότητα εκτέλεσης του προγράμματός του σε βήματα μίας εντολής, ενώ ταυτόχρονα δίνει πρόσβαση σε όλες τις συσκευές που περιλαμβάνει ο μικροελεγκτής. Έτσι ο χρήστης μπορεί να αλλάζει για παράδειγμα ένα σήμα εισόδου ή ακόμα και τα περιεχόμενα της μνήμης κατά τη διάρκεια της βηματικής εκτέλεσης του κώδικα. Επίσης παρουσιάζονται όλες οι πληροφορίες για την κατάσταση του επεξεργαστή, καθώς και τα περιεχόμενα των καταχωρητών του συστήματος. Φυσικά η εξομοίωση γίνεται εξολοκλήρου στον υπολογιστή, σε εικονικό περιβάλλον, και η επικοινωνία με άλλες συσκευές, όπως ο Ethernet controller, δεν είναι εφικτή στο εξομοιωτή.

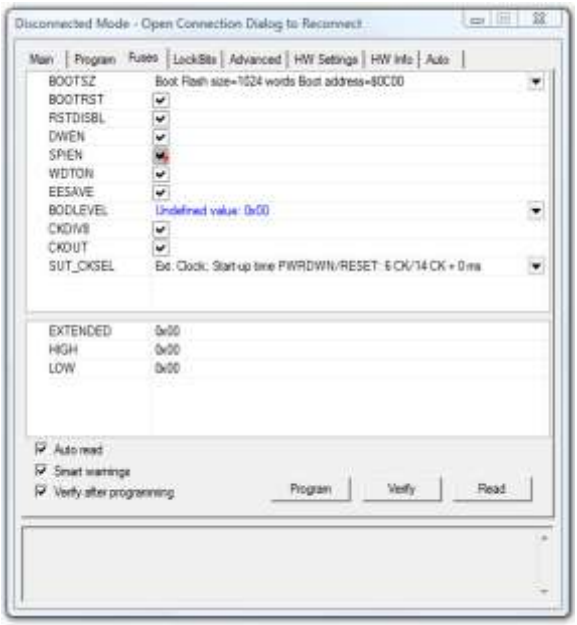


Εικόνα 62: Το περιβάλλον εργασίας του AVRstudio

Μια άλλη πολύ σημαντική λειτουργία του AVRstudio, είναι η μεταγλώττιση των προγραμμάτων. Ο χρήστης έχει τη δυνατότητα να μεταγλωττίσει τον κώδικα του σε γλώσσα μηχανής και να τον μεταφορτώσει στον αντίστοιχο μικροελεγκτή AVR. Η μεταφορά του κώδικα γίνεται με χρήση μια ειδικής αναπτυξιακής πλακέτας που ονομάζεται προγραμματιστής (programmer). Ο προγραμματιστής επιτρέπει την πρόσβαση στα fuses, στη μνήμη flash και τη μνήμη eeprom για σειριακό ή παράλληλο προγραμματισμό. Όπως θα δούμε παρακάτω, ο πιο συνηθισμένος προγραμματιστής για AVR είναι ο προγραμματιστής STK500 της Atmel.



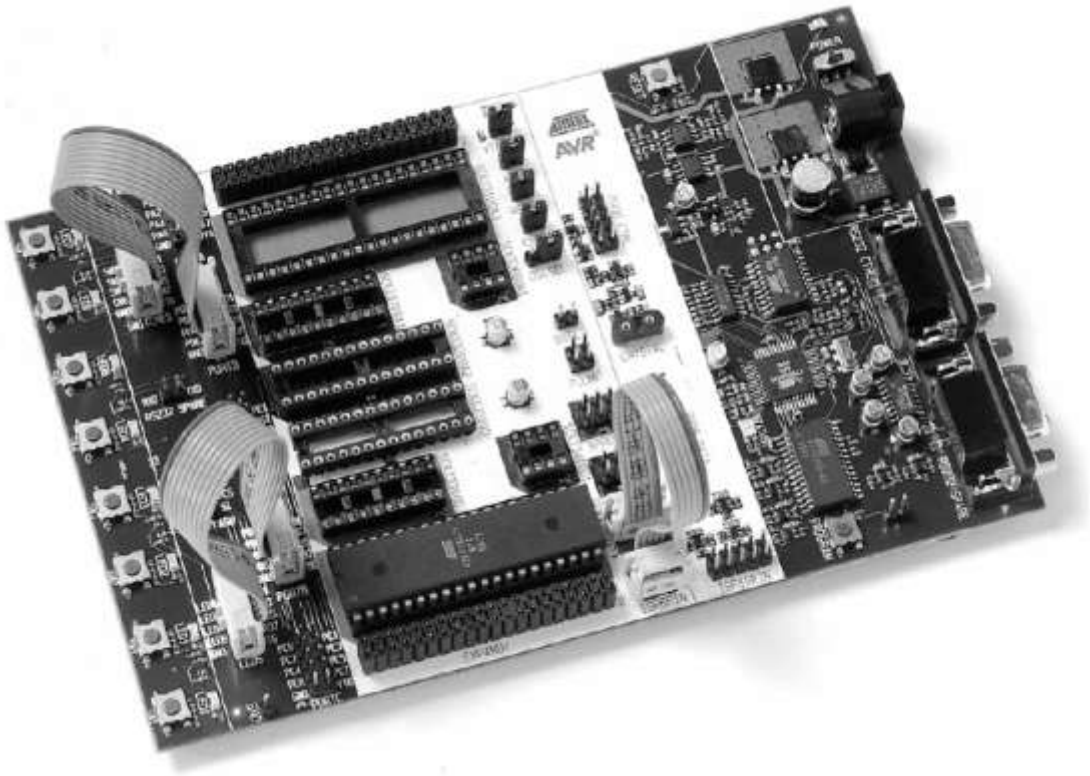
Εικόνα 63: Το μενού προγραμματισμού



Εικόνα 64: Το μενού ελέγχου των fuses

8.2 STK500

Το STK500 είναι μια αναπτυξιακή πλακέτα που μπορεί να χρησιμοποιηθεί με πολλούς διαφορετικούς μικροελεγκτές AVR. Περιλαμβάνει ειδικές θύρες σύνδεσης με μικροελεγκτές 20, 28 και 40 ακίδων, ενώ μέσω της θύρας επέκτασης που διαθέτει, μπορεί να συνδεθεί με άλλες πλακέτες της οικογένειας STK που υποστηρίζουν διαφορετικούς τύπους μικροελεγκτών. Η σύνδεση με τον υπολογιστή γίνεται μέσω σειριακής θύρας, ενώ ο προγραμματισμός του μικροελεγκτή συνήθως γίνεται με σειριακό προγραμματισμό ISP (In System Programming). Το αναπτυξιακό συμπληρώνουν 8 διακόπτες που μπορούν να συνδεθούν με μια θύρα εισόδου ενός μικροελεγκτή, καθώς και 8 led που μπορούν να συνδεθούν με μια θύρα εξόδου. Έτσι ανάλογα με το πρόγραμμα που εκτελείται ο χρήστης μπορεί να ελέγχει τα σήματα εισόδου και να διαβάζει τα σήματα εξόδου από τις συγκεκριμένες θύρες. Τέλος το αναπτυξιακό διαθέτει ταλαντωτή ρυθμιζόμενης συχνότητας για σύνδεση με το ρολόι του μικροελεγκτή, ενώ και η τάση τροφοδοσίας του συνδεδεμένου chip μπορεί να μεταβληθεί.

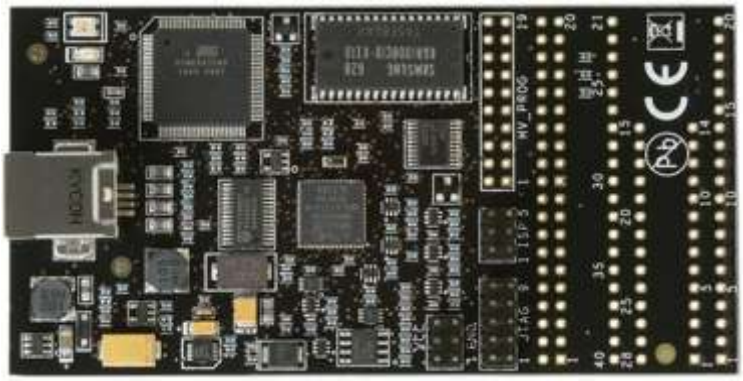


Εικόνα 65: Η πλακέτα του STK500

8.3 AVRdragon

Ένα ακόμα πολύ χρήσιμο εργαλείο από την Atmel, είναι το AVRdragon, που συνδυάζει ένα programmer με έναν on-chip debugger. Ένας on-chip debugger προσφέρει

στο χρήστη τη δυνατότητα να εκτελέσει και να διορθώσει αν χρειαστεί το πρόγραμμά του απευθείας στον μικροελεγκτή του συστήματος, και όχι σε κάποιον εξομοιωτή. Έτσι είναι εφικτή η βηματική εκτέλεση του κώδικα καθώς και η αλλαγή σημαντικών παραμέτρων, όπως τα περιεχόμενα ενός καταχωρητή, απευθείας στο chip.



Εικόνα 66: Η πλακέτα του AVRdragon

Το σημαντικό πλεονέκτημα σε σχέση με τον εξομοιωτή παρουσιάζεται όταν ο μικροελεγκτής που προγραμματίζουμε συνδέεται με άλλες συσκευές, όπως ένας Ethernet controller, και η μελέτη της επικοινωνίας των δύο αυτών συσκευών μπορεί να πραγματοποιηθεί μόνο απευθείας στο τελικό κύκλωμα. Στο σύστημά μας χρησιμοποιήσαμε το AVRdragon για τον έλεγχο του προγράμματος μας, και την ανάλυση της επικοινωνίας με όλες τις περιφερειακές συσκευές.

8.4 WinAVR

Όπως είδαμε το AVRstudio προσφέρει ένα περιβάλλον μεταγλώττισης και διόρθωσης προγραμμάτων για μικροελεγκτές AVR. Όμως η μόνη γλώσσα προγραμματισμού που υποστηρίζει είναι η γλώσσα assembly. Η υλοποίηση μεγάλων προγραμμάτων σε γλώσσα assembly δεν είναι πρακτική και για αυτό το λόγο συχνά προτιμάται η γλώσσα C. Η γλώσσα C διευκολύνει τον προγραμματιστή στην αποθήκευση μεταβλητών στη μνήμη RAM, αλλά και στην χρήση συναρτήσεων με παραμέτρους. Η δυνατότητα προγραμματισμού σε γλώσσα C προσφέρεται στο AVRstudio από το πακέτο WinAVR.

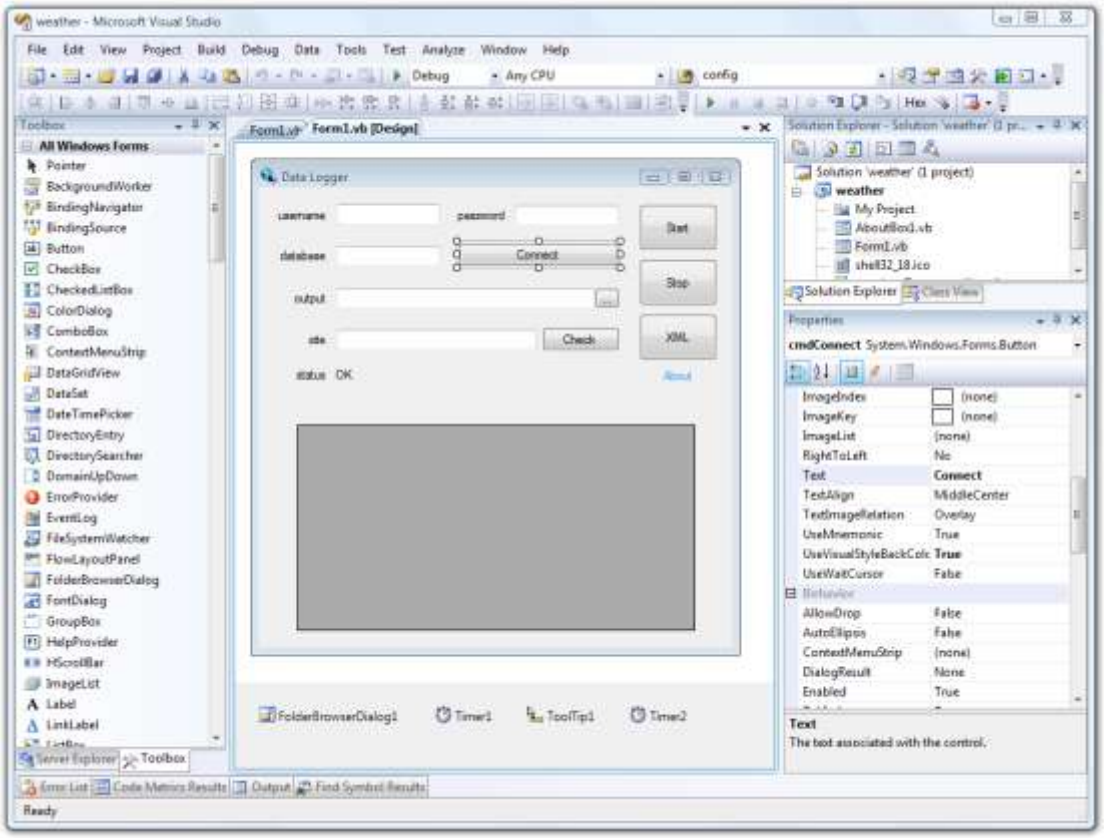
Το WinAVR περιλαμβάνει τη βιβλιοθήκη avr-libc που επιτρέπει τον προγραμματισμό μικροελεγκτών AVR σε γλώσσα C. Επίσης περιέχει και μια πλούσια βιβλιοθήκη συναρτήσεων που διευκολύνουν τον προγραμματιστή στην υλοποίηση τετριμμένων λειτουργιών του συστήματος, όπως η εγγραφή στη μνήμη eeprom και η αποθήκευ-

ση δεδομένων την περιοχή προγράμματος. Στη συνέχεια παρουσιάζονται τα αρχεία επικεφαλίδων της βιβλιοθήκης avr-libc που χρησιμοποιήσαμε στο πρόγραμμά μας.

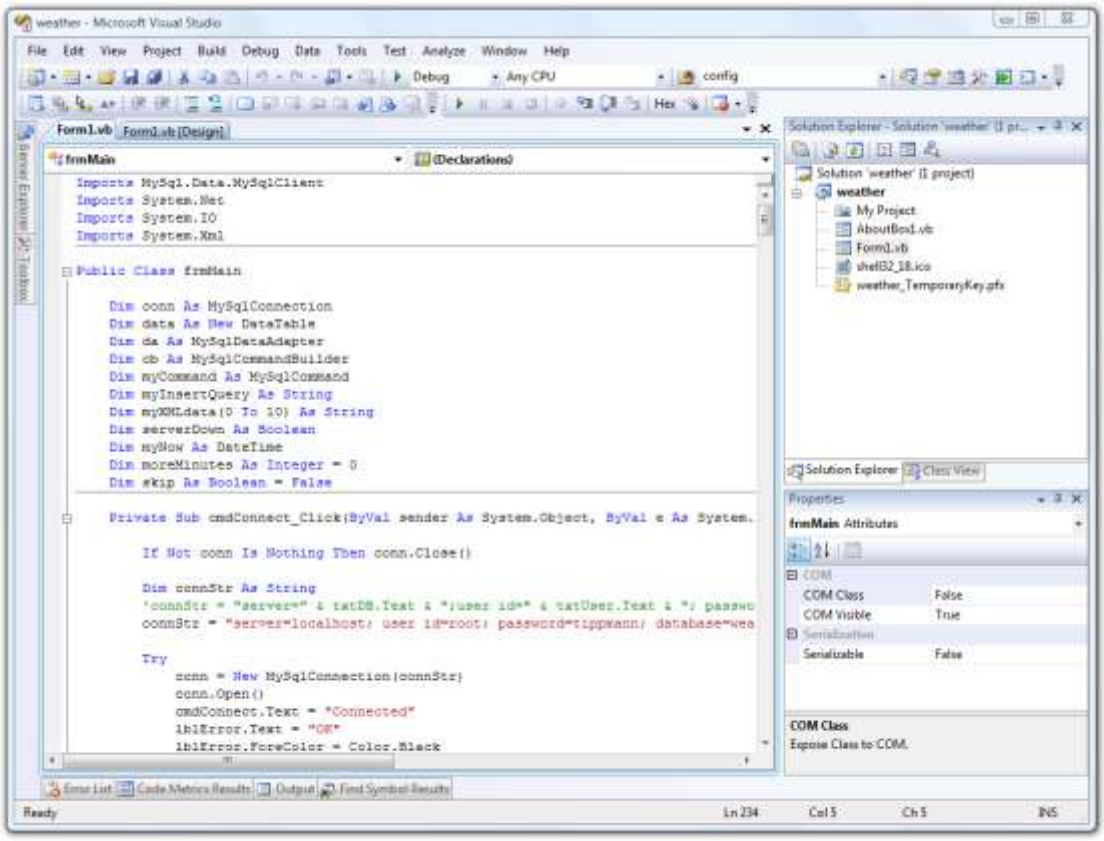
- <inttypes.h>, μετατροπές ακεραίων
- <stdio.h>, συναρτήσεις εισόδου-εξόδου
- <stdlib.h>, γενικά εργαλεία
- <string.h>, συναρτήσεις string
- <avr/eeprom.h>, χειρισμός μνήμης eeprom
- <avr/interrupt.h>, διακοπές
- <avr/io.h>, ορισμοί εισόδου-εξόδου για συσκευές AVR
- <avr/pgmspace.h>, εργαλεία περιοχής προγράμματος
- <avr/wdt.h>, χειρισμός χρονιστή watchdog
- <util/delay.h>, συναρτήσεις χρονοκαθυστέρησης

8.5 Visual Studio

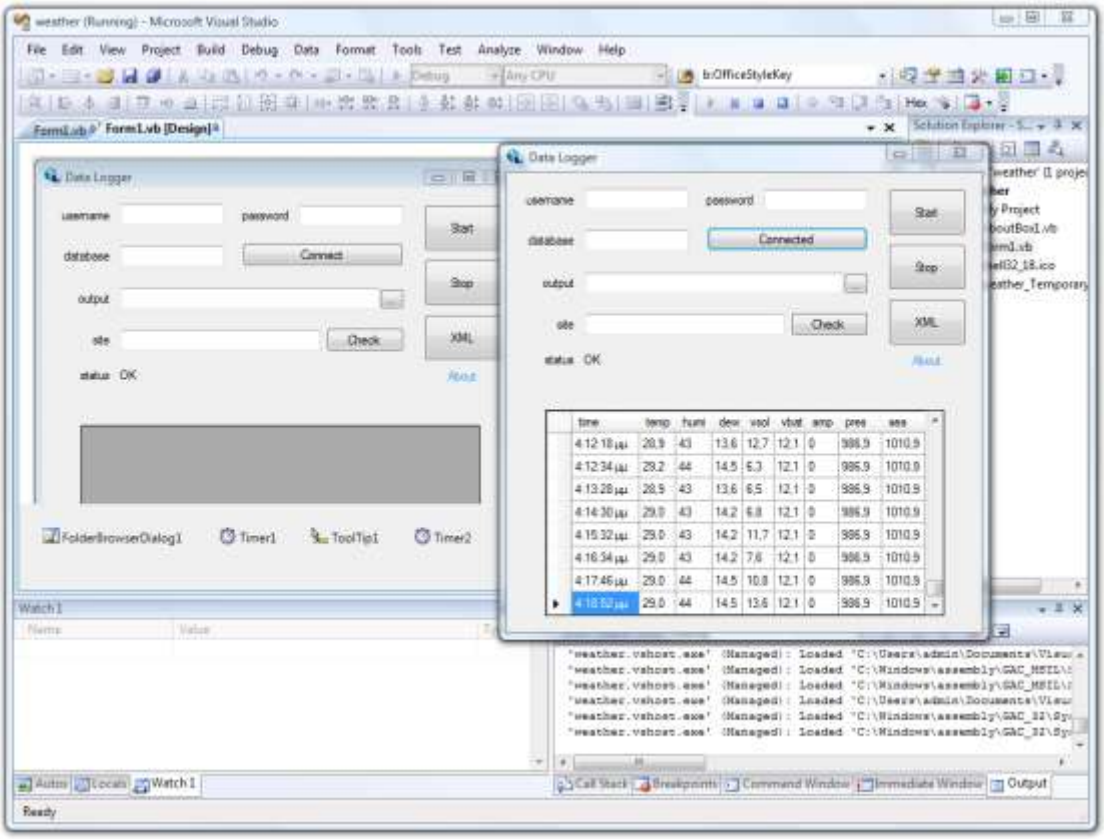
Το Visual Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον που επιτρέπει τη δημιουργία εφαρμογών για Windows. Το περιβάλλον αυτό προσφέρει τη δυνατότητα προγραμματισμού στις γλώσσες Visual Basic, Visual C++ και Visual C#. Η υλοποίηση εφαρμογών σε Visual Basic δίνει την δυνατότητα στο προγραμματιστή να κατασκευάσει εύκολα ένα γραφικό περιβάλλον από φόρμες, με τις οποίες ο τελικός χρήστης αλληλεπιδρά. Η βιβλιοθήκη συναρτήσεων του Visual Studio είναι κοινή και για τις τρείς γλώσσες προγραμματισμού που αναφέραμε και περιλαμβάνεται στο πακέτο .NET framework. Όλες οι παραπάνω δυνατότητες της γλώσσας Visual Basic και του περιβάλλοντος Visual Studio, τα καθιστούν ιδανική λύση για τον προγραμματισμό και έλεγχο του logger στο σύστημα που υλοποιήσαμε.



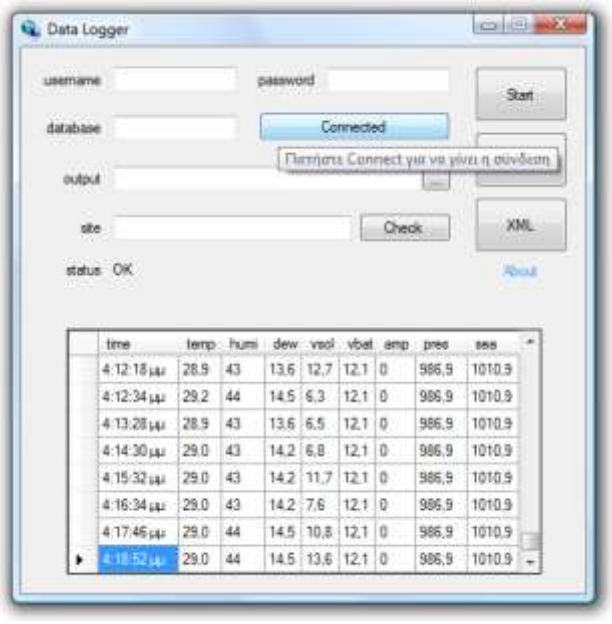
Εικόνα 67: Το περιβάλλον εργασίας του Visual Studio



Εικόνα 68: Το περιβάλλον εργασίας του Visual Studio



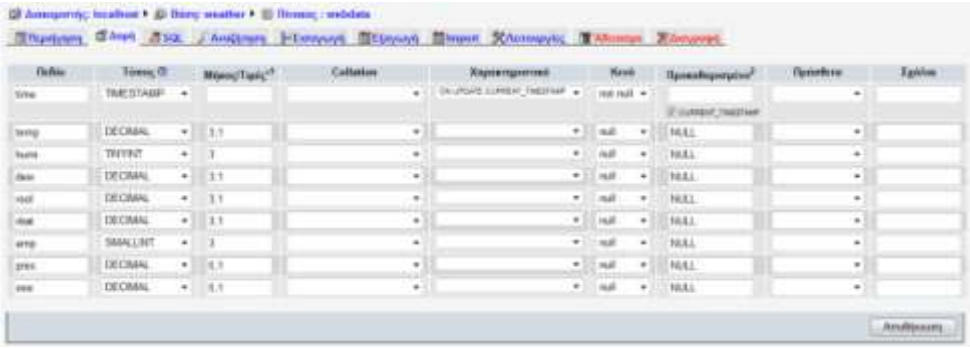
Εικόνα 69: Δοκιμαστική εκτέλεση του logger



Εικόνα 70: Μήνυμα βοήθειας tooltip

8.6 phpMyAdmin

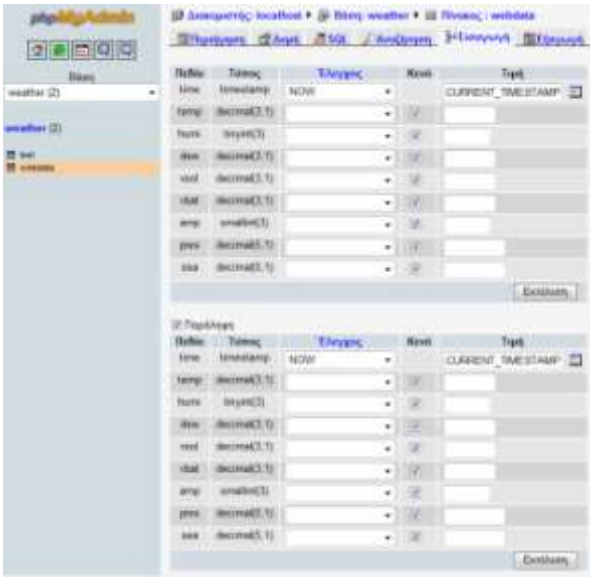
Το εργαλείο phpMyAdmin περιέχεται στο πακέτο xampp και προσφέρει υπηρεσίες διαχείρισης της βάσης δεδομένων MySQL. Το εργαλείο αυτό είναι γραμμένο σε γλώσσα php και εκτελείται στον Apache web server του πακέτου xampp. Στο σύστημά μας χρησιμοποιήσαμε το phpMyAdmin για τη σχεδίαση και βελτιστοποίηση της βάσης δεδομένων. Οι δυνατότητες που προσφέρονται περιλαμβάνουν δημιουργία νέας βάσης δεδομένων, δημιουργία νέου πίνακα στη βάση, εισαγωγή πεδίων στον πίνακα, ορισμό πρωτεύοντος κλειδιού και φυσικά εισαγωγή δεδομένων. Η απλότητα της χρήσης εντοπίζεται στο γραφικό περιβάλλον, το οποίο μπορεί να προσπελαστεί μέσω ενός απλού web browser. Στις εικόνες που ακολουθούν παρουσιάζεται το περιβάλλον του εργαλείου phpMyAdmin.



Εικόνα 71: Δημιουργία πίνακα στο phpMyAdmin

Εικόνα 72: Επισκόπηση πίνακα στο phpMyAdmin

Εικόνα 73: Προβολή περιεχομένων πίνακα στο phpMyAdmin

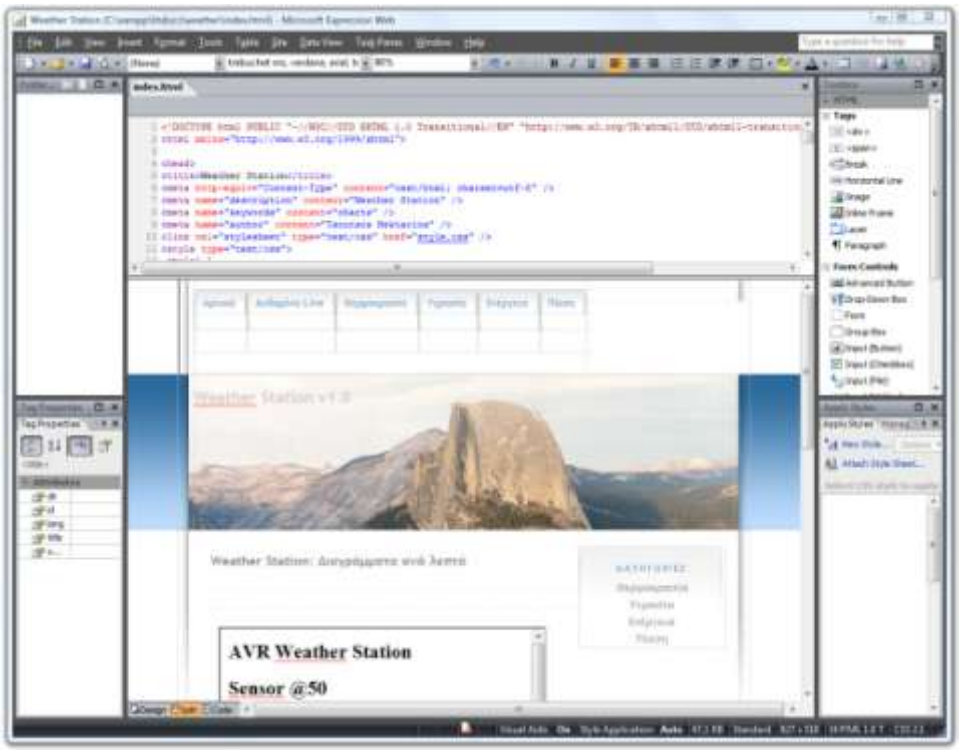


Εικόνα 74: Εισαγωγή εγγραφής στο phpMyAdmin

8.7 Expression Web

Η υλοποίηση του homepage έγινε με τη βοήθεια του σχεδιαστικού εργαλείου Expression Web της Microsoft, που δίνει τη δυνατότητα στον προγραμματιστή να σχεδιάζει σελίδες πλούσιες σε γραφικά. Ο προγραμματιστής μπορεί επίσης να εισαγάγει ένα υποπαράθυρο που λειτουργεί ξεχωριστά από τη μητρική ιστοσελίδα. Αυτή την δυνατότητα εφαρμόσαμε για την παρουσίαση της ιστοσελίδας που στέλνει ο server, μέσα από τη σελίδα του homepage. Ένα ακόμα πολύ σημαντικό χαρακτηριστικό του εργαλείου Expression Web είναι ο έλεγχος προτύπων που προσφέρει. Κάθε ιστοσελίδα που σχεδιάζεται, ελέγχεται αν υποστηρίζει τα απαιτούμενα

standards, καθώς και αν υποστηρίζεται από διάφορους web browsers. Επίσης ο προγραμματιστής έχει τη δυνατότητα να ενσωματώσει στις ιστοσελίδες που σχεδιάζει ένα πλήθος από αντικείμενα multimedia, όπως βίντεο, μουσική, αρχεία flash, εικόνες και JavaScript.



Εικόνα 75: Το περιβάλλον εργασίας του Expression Web

Μέρος 3ο Ανάλυση

Κεφάλαιο 9
Calculations

Στην ενότητα αυτή θα παρουσιάσουμε τις μετρήσεις και τους υπολογισμούς που πραγματοποιήσαμε για τον εκτίμηση της κατανάλωσης ενέργειας του συστήματος. Επίσης θα παρουσιάσουμε τους υπολογισμούς που αφορούν στις καμπύλες μετασχηματισμού των μετρήσεων του ADC, καθώς και τους υπολογισμούς της κυμάτωσης του DC-to-DC converter.

9.1 Μετρήσεις ενέργειας

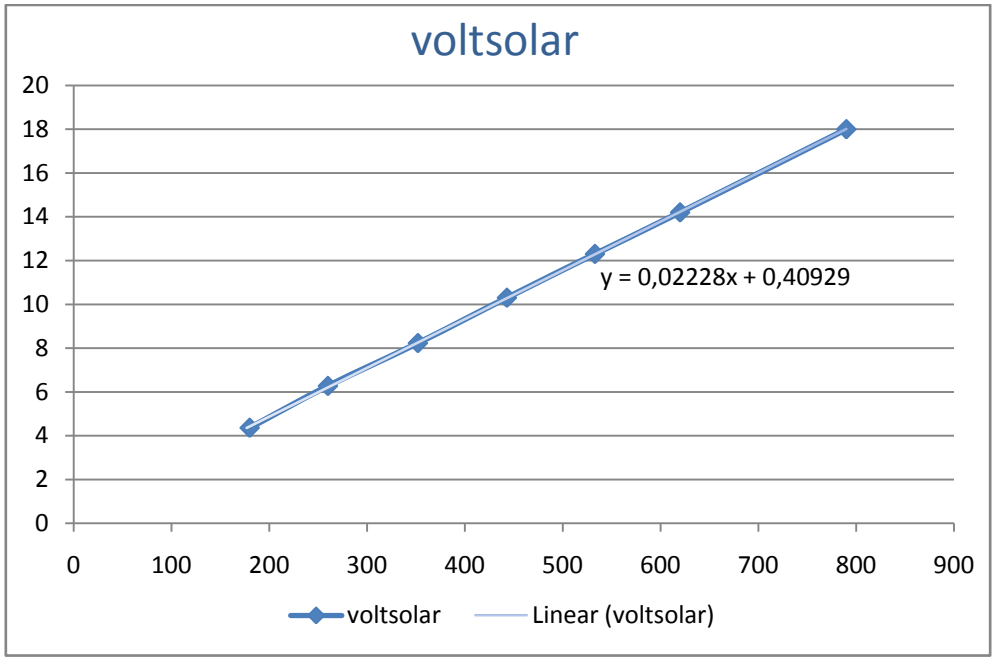
Αρχικά θα μελετήσουμε την μετρούμενη κατανάλωση του συστήματος και την ενέργεια που προσφέρεται από τον ηλιακό συλλέκτη. Από τις μετρήσεις που πραγματοποιήσαμε προκύπτει ότι η κατανάλωση του WiFi είναι 2.2Watt και κατανάλωση του server είναι 0.6Watt. Οι υπολογισμοί αυτοί έγιναν μετρώντας το ρεύμα τροφοδοσίας κάθε συσκευής υπό την ονομαστική τάση 5.3Volt. Η κατανάλωση μετά τον DC-to-DC converter από 12 σε 5.3Volt μετρήθηκε ίση με 2.9Watt, ενώ η κατανάλωση πριν τον DC-to-DC converter μετρήθηκε ίση με 3.6Watt, και αντιστοιχεί στην ισχύ που τροφοδοτεί η μπαταρία. Όπως βλέπουμε η απώλειες του DC-to-DC converter είναι περίπου 0.7Watt, που αντιστοιχεί σε απόδοση 80%.

Εξάρτημα	**Ρεύμα I**	**Τάση V**	**Ισχύς W**
DC-to-DC converter	0.27A	13.13V	3.55W
Server + WiFi	0.54A	5.33V	2.86W
Μπαταρία	0.33A	13.13V	4.33W
Ηλιακός συλλέκτης	0.62A	13.9V	8.62W

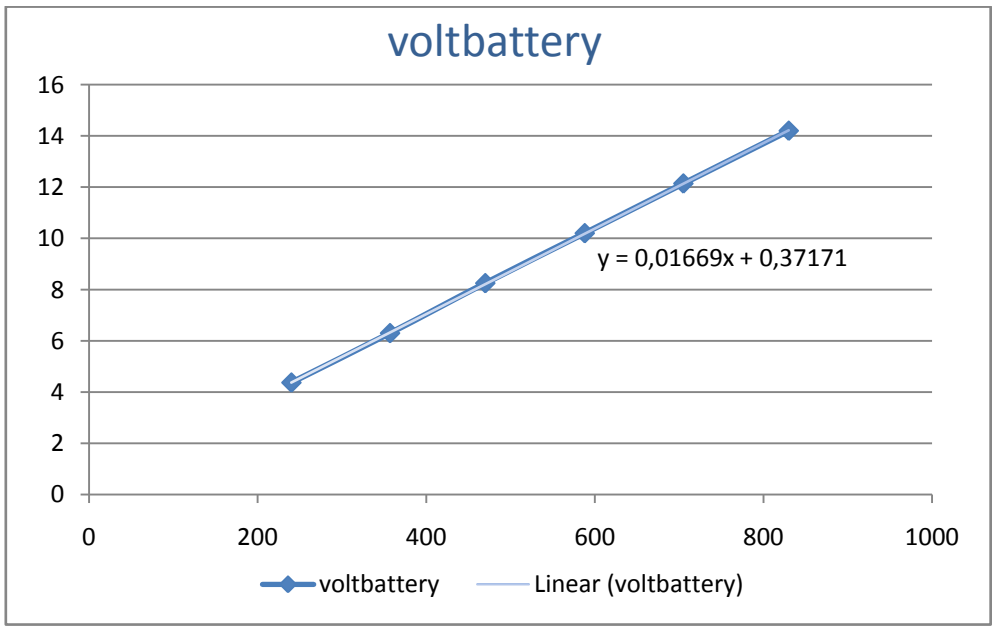
Η ισχύς του ηλιακού συλλέκτη σε πλήρη ηλιοφάνεια τις μεσημβρινές ώρες υπολογίστηκε 8.6Watt. Αν υποθέσουμε ότι μια ηλιόλουστη μέρα ο ηλιακός συλλέκτης λειτουργεί κατά μέσο όρο 10 ώρες, η ενέργεια που παρέχεται είναι περίπου 86 Wh. Η ενέργεια που καταναλώνεται από την μπαταρία κατά τη διάρκεια μίας ημέρας είναι και αυτή περίπου 86 Wh. Έτσι συμπεραίνουμε ότι υπό κανονικές συνθήκες η ενέργεια του ηλιακού συλλέκτη επαρκεί για την κάλυψη των ενεργειακών αναγκών μίας ημέρας. Όμως η μπαταρία δεν έχει απόδοση 100%, οπότε είναι απαραίτητη η χρήση του ενεργειακού προφίλ που υλοποιήσαμε στον server.

9.2 Καμπύλες ADC

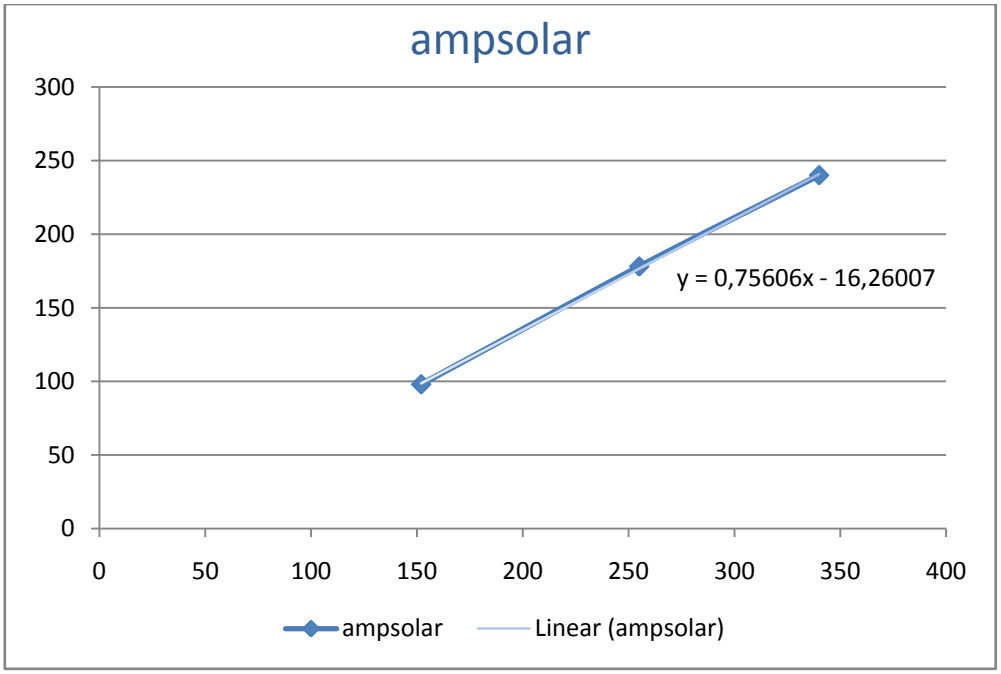
Στην παράγραφο αυτή θα μελετήσουμε τις καμπύλες μετασχηματισμού των ακαθάριστων (raw) μετρήσεων του ADC, σε τιμές τάσης και ρεύματος. Ο υπολογισμός κάθε καμπύλης έγινε πειραματικά, αντιστοιχίζοντας διαφορετικές τιμές του ADC με τις πραγματικές τιμές που μετρώνται στο πολύμετρο. Όπως φαίνεται στις γραφικές παραστάσεις που ακολουθούν, οι μετρήσεις προσεγγίζονται πολύ καλά από ευθείες.



Εικόνα 76: Καμπύλη μετασχηματισμού τάσης ηλιακού συλλέκτη



Εικόνα 77: Καμπύλη μετασχηματισμού τάσης μπαταρίας



Εικόνα 78: Καμπύλη μετασχηματισμού ρεύματος ηλιακού συλλέκτη

9.3 DC-to-DC ripple

Στην παράγραφο αυτή θα μελετήσουμε την θεωρητική τιμή της κυμάτωσης του DC-to-DC converter. Στον πίνακα που ακολουθεί παρουσιάζονται οι εξισώσεις που επιλύουν τα στοιχεία του κυκλώματος του μετατροπέα που χρησιμοποιούμε.

CALCULATION	STEP DOWN
t_{on}/t_{off}	$\frac{V_{out} + V_F}{V_{in(min)} - V_{sat} - V_{out}}$
$(t_{on} + t_{off})$	$\frac{1}{f}$
t_{off}	$\frac{t_{on} + t_{off}}{\frac{t_{on}}{t_{off}} + 1}$
t_{on}	$(t_{on} + t_{off}) - t_{off}$
C_T	$4 \times 10^{-5} t_{on}$
$I_{pk(switch)}$	$2I_{out(max)}$
R_{SC}	$\frac{0.3}{I_{pk(switch)}}$
$L_{(min)}$	$\left(\frac{(V_{in(min)} - V_{sat} - V_{out})}{I_{pk(switch)}}\right) t_{on(max)}$
C_O	$\frac{I_{pk(switch)}(t_{on} + t_{off})}{8V_{ripple(pp)}}$
V_{out}	$1.25(1 + \frac{R2}{R1})$

Εικόνα 79: Πίνακας παραμέτρων του υποβιβαστή τάσης

Στον πίνακα που ακολουθεί παρουσιάζονται οι τιμές των κρίσιμων παραμέτρων για κάθε ένα από τους μετατροπείς τάσης που χρησιμοποιούμε.

V_{in}	V_{out}	t_{on}/t_{off}	t_{on}	t_{off}	I_{pk}	I_{out}	V_{ripple}	C_O
12.5V	5.3V	0.92	$1.18 \cdot 10^{-5}$s	$1.28 \cdot 10^{-5}$s	1.6A	0.8A	4.91mV	1000μF
5.3V	3.3V	3.7	$1.18 \cdot 10^{-5}$s	$3.18 \cdot 10^{-6}$s	0.5A	0.25A	0.93mV	1000μF

Όπως φαίνεται και από τον πίνακα, η τιμή της κυμάτωσης είναι περίπου 5mV για τον πρώτο μετατροπέα και περίπου 1mV για τον δεύτερο.

Κεφάλαιο 10
Προβλήματα

Στην ενότητα αυτή θα παρουσιάσουμε τα προβλήματα που συναντήσαμε κατά την υλοποίηση του συστήματος. και θα αναλύσουμε τις λύσεις που δόθηκαν. Το προβλήματα αυτά αφορούν:

- την εξοικονόμηση ενέργειας
- την ακριβή μέτρηση της τάσης
- την διατήρηση ημερομηνίας στον server
- τον διακόπτη τροφοδοσίας του WiFi
- την μέτρηση του ρεύματος
- την αρχικοποίηση της οθόνης lcd
- την υλοποίηση του ελέγχου CRC
- και τη γραμμική προσέγγιση του δεκαδικού λογάριθμου

Επίσης παρουσιάζεται η προβλεπόμενη λειτουργία του συστήματος σε περίπτωση σφάλματος σε κάποιο υποσύστημα.

10.1 Power profile

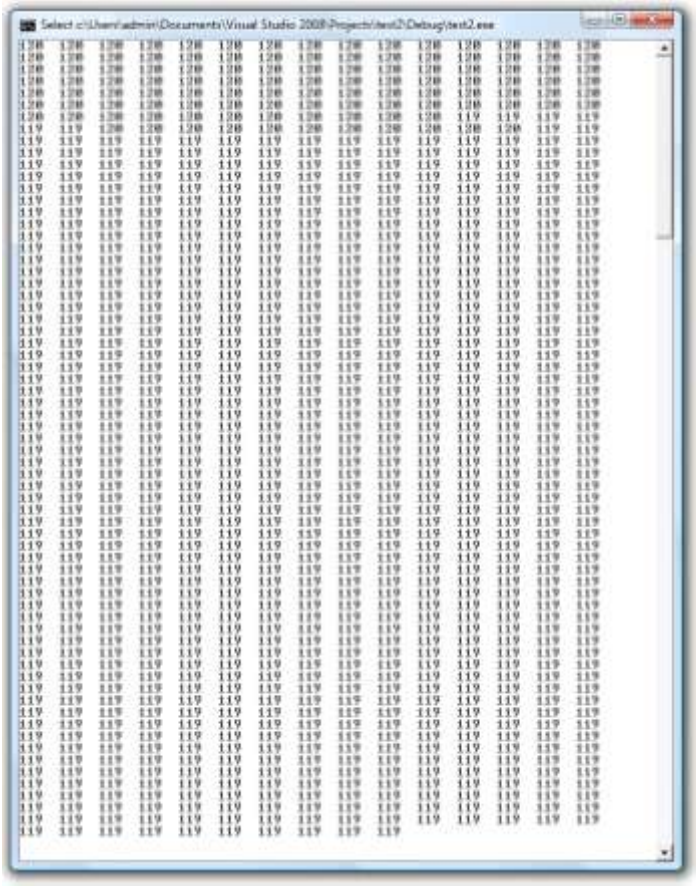
Στην παράγραφο αυτή θα αναλύσουμε το πρόβλημα της διαχείρισης της ενέργειας. Το σύστημά μας τροφοδοτείται από ηλιακή ενέργεια η οποία δεν αποτελεί αξιόπιστη πηγή, αφού λόγω των καιρικών συνθηκών η ηλιοφάνεια μεταβάλλεται σημαντικά σε ορισμένες περιπτώσεις. Αυτό έχει σαν αποτέλεσμα η φόρτιση της μπαταρίας να μην επαρκεί για την κάλυψη των ημερήσιων ενεργειακών αναγκών. Στην περίπτωση αυτή το σύστημα απενεργοποιεί τις ενεργοβόρες συσκευές του, και τίθεται σε κατάσταση αναμονής μέχρι να βρεθεί η απαιτούμενη ενέργεια. Όμως ακόμα και στην περίπτωση πλήρους απενεργοποίησης του server, το τμήμα του συστήματος που διαβάζει τις μετρήσεις από τη βάση και τις παρουσιάζει σε μορφή διαγραμμάτων, εξακολουθεί να λειτουργεί. Έτσι ο χρήστης μπορεί να μελετήσει τις μετρήσεις που καταγράφηκαν πριν την απενεργοποίηση του server. Σε κάθε περίπτωση πάντως, η πλήρης απενεργοποίηση του server δεν προβλέπεται να συμβαίνει συχνά, δίνοντας μεγάλη πρακτική αξία στο σύστημα.

10.2 integrator

Αρχικά θα εξετάσουμε τη μέτρηση της τάσης της μπαταρίας. Η τάση της μπαταρίας μετριέται μέσω του ADC, του οποίου η τυπική τιμή σφάλματος είναι της τάξης του 2%. Έτσι για τη μέτρηση της τάσης της μπαταρίας, η οποία κυμαίνεται στα 13Volt, η αναμενόμενη απόκλιση είναι περίπου 0.25Volt. Όμως η ακρίβεια στον υπολογισμό της τάσης της μπαταρίας είναι πολύ σημαντική, αφού με βάση αυτό το επίπεδο τάσης γίνεται η διαχείριση της ενέργειας από τον server. Για να βελτιώσουμε την ακρίβεια της μέτρησης χρησιμοποιούμε την υπόθεση ότι το επίπεδο της τάσης της μπαταρίας μεταβάλλεται πολύ αργά, και το γεγονός ότι διαδοχικές μετρήσεις σε μικρό χρονικό διάστημα διαφέρουν μέχρι και 0.5Volt, όπως παρατηρήθηκε, είναι αποτέλεσμα σφάλματος του ADC. Υλοποιώντας μία συνάρτηση που λαμβάνει διαρκώς μετρήσεις και επιστρέφει τον μέσο όρο των πιο πρόσφατων από αυτές, επιτυγχάνουμε την απόρριψη μεγάλου μέρους του τυχαίου σφάλματος που υπεισέρχεται στις μετρήσεις. Ο ολοκληρωτής μεταβάλλει την τιμή της εξόδου του μόνο όταν ένα μεγάλο ποσοστό των μετρήσεων έχει μεταβληθεί, πράγμα που υποθέσαμε ότι συμβαίνει μόνο όταν η τάση έχει πραγματικά μεταβληθεί, και όχι λόγω τυχαίων σφαλμάτων.

Η υλοποίηση της συνάρτησης παρουσιάστηκε σε προηγούμενο κεφάλαιο. Στην ενότητα αυτή θα παρουσιάσουμε μία μέθοδο που επιβεβαιώνει ότι όταν υπάρχει ικανός αριθμός δειγμάτων, το τυχαίο σφάλμα μπορεί σχεδόν να εξαλειφθεί. Για το λόγο αυτό σχεδιάσαμε μια συνάρτηση παρόμοια με αυτή που υπάρχει στο server, με τη διάφορα ότι οι μετρήσεις δεν λαμβάνονται από κάποιον ADC, αλλά από μια άλλη συνάρτηση που προσομοιώνει τον ADC και εισαγάγει τυχαίο σφάλμα στα αποτελέσματα. Πιο συγκεκριμένα ο εικονικός ADC χρησιμοποιεί πάντα μια σταθερή τιμή στην οποία προσθέτει ή αφαιρεί ένα τυχαίο αριθμό πεπερασμένου εύρους. Έτσι η έξοδος του εικονικού ADC προσομοιώνει τον πραγματικό ADC όπου το σφάλμα προστίθεται ή αφαιρείται από την πραγματική τιμή. Στη συνέχεια εξετάζουμε αν ο ολοκληρωτής που σχεδιάσαμε μπορεί να εξαλείψει το τυχαίο σφάλμα και να επιτρέψει μια τιμή πολύ κοντά στην ορθή.

Το πρόγραμμα εξομοίωσης λειτουργεί σε περιβάλλον command prompt και λαμβάνει ως πραγματική τιμή το 120 στην οποία προστίθεται ή αφαιρείται τυχαίο σφάλμα μέγιστης τιμής 7. Η έξοδος προκύπτει από τον μέσο όρο των τελευταίων 128 τιμών. Στη συνέχεια παρουσιάζεται η έξοδος του προσομοιωτή για τις πρώτες 1000 μετρήσεις.



Εικόνα 80: Η έξοδος του εξομοιωτή

Όπως φαίνεται καθαρά στο σχήμα, ο ολοκληρωτής λειτουργεί επιτυχημένα, αφού απορρίπτει σχεδόν ολοκληρωτικά το σφάλμα, και συγκλίνει σε μία τιμή πολύ κοντά στην πραγματική. Αξίζει να σημειωθεί ότι και σε μεγαλύτερη προσομοίωση 10000 μετρήσεων, η έξοδος ήταν σχεδόν πάντα 119, και μόνο σε ελάχιστες περιπτώσεις ήταν 120.

10.3 Timestamp

Για την αντιστοίχηση ενός πακέτου μετρήσεων με την χρονική στιγμή που αποθηκεύτηκαν στην μνήμη eeprom, ο server πρέπει να γνωρίζει την ημερομηνία και την ώρα. Όμως στην περίπτωση που ο server απενεργοποιηθεί λόγω χαμηλής τάσης της μπαταρίας, ο μικροελεγκτής θα αρχικοποιηθεί και οποιαδήποτε ημερομηνία και ώρα κρατείται στην μνήμη, θα χαθεί ή θα διαφέρει από την πραγματική. Τη λύση σε αυτό το πρόβλημα δίνει ο logger, που σε κάθε επικοινωνία του με τον server, αποστέλλει την τρέχουσα ημερομηνία και ώρα (timestamp). Η πληροφορία αυτή κρατείται σαν offset και σε αυτή προστίθεται ο χρόνος που πέρασε από την στιγμή της

τελευταίας ανανέωσης. Έτσι ο server μπορεί να ανακτά την ημερομηνία και την ώρα αυτόματα, χωρίς την ανάγκη προγραμματισμού του από το χρήστη.

10.4 Διακόπτης WiFi

Η εξοικονόμηση ενέργειας που περιγράψαμε προϋποθέτει την απενεργοποίηση των ενεργοβόρων συσκευών, όπως ο WiFi router. Τυπικά ένα τρανζίστορ μπορεί να χρησιμοποιηθεί σαν ηλεκτρονικός διακόπτης για σχετικά μικρά φορτία. Όμως ο WiFi router σε πλήρη λειτουργία είναι ένα φορτίο της τάξης των 420mA, που όταν τροφοδοτείται με ρεύμα, εκκινεί μια διαδικασία αρχικοποίησης η οποία καταναλώνει περίπου 120mA, πριν ενεργοποιήσει την κεραία WiFi μετά από μερικά δευτερόλεπτα. Είναι λοιπόν φανερό ότι το φορτίο μας μεταβάλλεται σημαντικά στο χρόνο. Όταν για τον έλεγχο της τροφοδοσίας του WiFi router χρησιμοποιείται τρανζίστορ τότε εμφανίζονται τα ακόλουθα προβλήματα:

- Τυπικά ένα τρανζίστορ εμφανίζει πτώση τάσης από 0.6Volt ως και 1.0Volt, η οποία είναι μεγαλύτερη όσο μεγαλύτερο είναι το ρεύμα του εκπομπού
- Αν η τάση τροφοδοσίας του WiFi router δεν είναι σταθερά μεγαλύτερη από 4.8Volt, τότε αποτυγχάνει η εκκίνηση της κεραίας WiFi και ο router πέφτει σε αόριστη κατάσταση
- Ένα τυπικό τρανζίστορ μεγάλης ενίσχυσης και μικρής πτώσης τάσης μπορεί να οδηγήσει ρεύματα της τάξεως των 100mA
- Ένα τυπικό τρανζίστορ μεγάλης ενίσχυσης που μπορεί να οδηγήσει ρεύματα της τάξεων των 500mA, εμφανίζει μη αποδεκτή πτώση τάσης της τάξεως των 0.7Volt όταν ενεργοποιείται η κεραία WiFi, και η εκκίνηση αποτυγχάνει.
- Ένα τυπικό τρανζίστορ ισχύος μπορεί να οδηγήσει μεγάλα ρεύματα της τάξεως των 10Amp, παρουσιάζοντας πτώση τάσης της τάξεως των 1.2Volt, η οποία όμως δεν μεταβάλλεται με το φορτίο
- Η ακίδα του μικροελεγκτή που οδηγεί το τρανζίστορ αντέχει μέγιστο ρεύμα 40mA (ρεύμα βάσης), και επομένως το τρανζίστορ πρέπει να έχει μεγάλο λόγο ενίσχυσης (ρεύμα εκπομπού 420mA), πράγμα που δεν ισχύει για ένα τυπικό τρανζίστορ ισχύος
- Αν αυξήσουμε την τάση τροφοδοσίας τότε, κατά την εκκίνηση και πριν εκκινήσει η κεραία, η πτώση τάσης στο τρανζίστορ θα είναι μικρή (μικρό ρεύμα) και ο WiFi router θα βρίσκεται σε τάση πολύ πάνω από τα αποδεκτά όρια λειτουργίας ώστε να κινδυνεύει να καεί, ενώ μετά την εκκίνηση της κεραίας

η τάση τροφοδοσίας θα είναι φυσιολογική καθώς υπάρχει μεγάλη πτώση τάσης στο τρανζίστορ (μεγάλο ρεύμα)

- Αυξάνοντας την τάση τροφοδοσίας πάνω από τα 5.3Volt τότε αυξάνονται οι απώλειες ενέργειας συνολικά στο σύστημα

Από τα παραπάνω γίνεται φανερό ότι λόγο της πολυπλοκότητας του φορτίου, και του γεγονότος ότι διαθέτει ένα στάδιο εκκίνησης με διαφορετικές απαιτήσεις σε ρεύμα, η χρήση τρανζίστορ είναι προβληματική. Έτσι επιλέγεται να χρησιμοποιηθεί ο μηχανικός διακόπτης ενός ηλεκτρονόμου. Αυτή η υλοποίηση αν και μηχανικής φύσεως, δεν παρουσιάζει κανένα πρόβλημα, αφού δεν παρουσιάζεται πτώση τάσης, και δεν υπάρχει ουσιαστικός περιορισμός στο ρεύμα που διέρχεται.

10.5 Αμπερόμετρο

Για τον υπολογισμό του ρεύματος του ηλιακού συλλέκτη έπρεπε να υλοποιηθεί ένα αμπερόμετρο. Μια αποδεκτή υλοποίηση είναι να μετρήσουμε το ρεύμα υπολογίζοντας την πτώση τάσης που προκαλεί σε γνωστή αντίσταση. Για αυτό το λόγο το ο αρνητικός πόλος του ηλιακού συλλέκτη συνδέεται με μία αντίσταση 0,1 Ohm πριν συνδεθεί με το υπόλοιπο κύκλωμα. Ένας εύκολος τρόπος υπολογισμού της πτώσης τάσης θα ήταν να χρησιμοποιήσουμε δύο κανάλια του ADC και να υπολογίζουμε τη διαφορά των δύο μετρήσεων. Ο περιορισμός όμως που υπάρχει είναι ότι διαθέτουμε μόνο ένα κανάλι ADC.

Μια λανθασμένη λύση που χρησιμοποιεί μόνο ένα κανάλι του ADC θα ήταν να υποθέσουμε ότι ο αρνητικός πόλος του ηλιακού συλλέκτη έχει δυναμικό 0Volt, και επομένως ο ακροδέκτης της αντίστασης 0.1Ohm που βρίσκεται από την μεριά του υπόλοιπου κυκλώματος θα έχει δυναμικό ίσο με την πτώση τάσης στην αντίσταση. Τότε θα αρκούσε να υπολογίζουμε αυτό το δυναμικό με τον ADC. Όμως στην πραγματικότητα αυτός ο υπολογισμός είναι αδύνατος γιατί ο ADC λαμβάνει αναφορά την γείωση του μικροελεγκτή η οποία πάντα βρίσκεται σε υψηλότερο δυναμικό από τον ακροδέκτη της αντίστασης 0.1Ohm από την οποία λαμβάνουμε την μέτρηση.

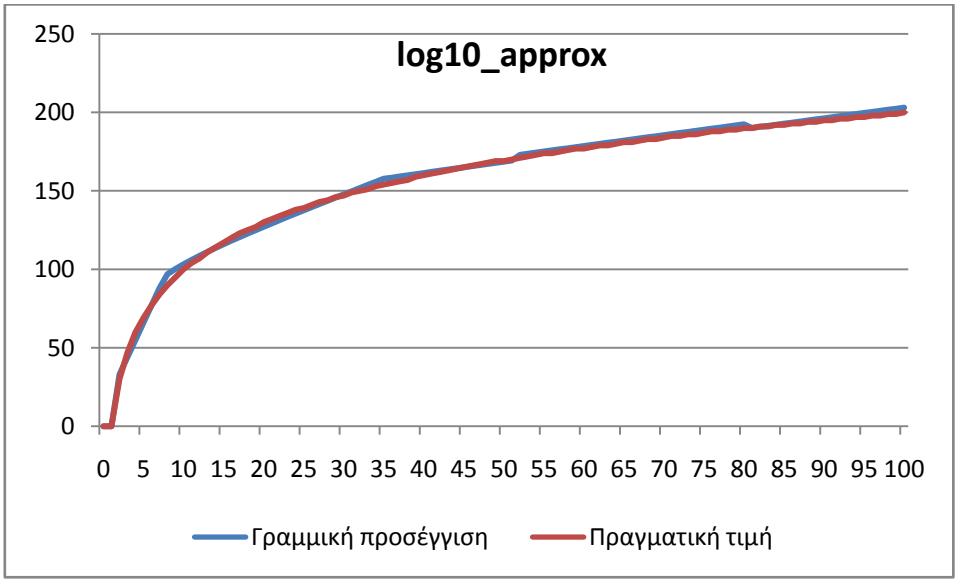
Τελικά η ορθή υλοποίηση είναι να χρησιμοποιήσουμε ένα τελεστικό ενισχυτή σε συνδεσμολογία διαφορικού ενισχυτή, με εισόδους τους δύο ακροδέκτες της αντίστασης 0.1Ohm. Ο τελεστικός ενισχυτής και ο μικροελεγκτής έχουν πρακτικά ίδιο δυναμικό στον ακροδέκτη γείωσης, και επομένως αρκεί ο ADC να μετρήσει την τάση στην έξοδο του διαφορικού ενισχυτή. Με αυτό τον τρόπο επιτυγχάνεται αποτελεσματικά η μέτρηση του ρεύματος ηλιακού συλλέκτη.

10.6 Reset LCD

Αν κατά τη χρήση του συστήματος η οθόνη lcd αποσυνδεθεί από το σύστημα, τότε σε περίπτωση επανασύνδεσης θα πρέπει να είναι δυνατή η αρχικοποίησή της χωρίς να απαιτείται επανεκκίνηση του συστήματος. Για το λόγο αυτό χρησιμοποιείται ένα push button το οποίο κάθε φορά που πατιέται ενεργοποιεί τη συνάρτηση αρχικοποίησης της οθόνης lcd. Επίσης με το ίδιο push button ενεργοποιείται και ο οπίσθιος φωτισμός της οθόνης.

10.7 Γραμμική προσέγγιση λογαρίθμου

Όπως είδαμε σε προηγούμενο κεφάλαιο, ο υπολογισμός του σημείου δρόσου προϋποθέτει τον υπολογισμό του δεκαδικού λογάριθμου της σχετικής υγρασίας. Για να αποφύγουμε τη χρήση μαθηματικής βιβλιοθήκης και αριθμητική κινητή υποδιαστολής, είδαμε ότι ο υπολογισμός του λογαρίθμου έγινε με χρήση γραμμικής προσέγγισης. Στο σημείο λοιπόν αυτό θα παρουσιάσουμε τη γραφική παράσταση της γραμμικής προσέγγισης που υλοποιήσαμε, και θα την συγκρίνουμε με την αντίστοιχη πραγματική τιμή. Στο σχήμα που ακολουθεί φαίνεται η πραγματική και η προσεγγιστική καμπύλη.



Εικόνα 81: Γραμμική προσέγγιση του λογαρίθμου

Όπως είναι προφανές η γραμμική προσέγγιση είναι πολύ κοντά στην πραγματική τιμή, και το σφάλμα στον υπολογισμό του σημείου δρόσου μπορεί να θεωρηθεί αμελητέο.

10.8 Υλοποίηση ελέγχου CRC

Όπως είδαμε και σε προηγούμενο κεφάλαιο, η λήψη δεδομένων από τον αισθητήρα Sensirion συνοδεύεται από τον απαραίτητο έλεγχο CRC. Ο έλεγχος αυτό γίνεται στον ίδιο τον αισθητήρα και η τιμή του CRC αποστέλλεται στον server μαζί με τα δεδομένα των μετρήσεων. Στη συνέχεια ο server υπολογίζει το CRC των δεδομένων που έλαβε, ώστε να το συγκρίνει με την τιμή που υπολόγισε ο αισθητήρας, για να διαπιστωθεί αν συνέβη σφάλμα κατά την μετάδοση. Το πρόβλημα που παρουσιάζεται είναι ότι ο αισθητήρας Sensirion υλοποιεί τον έλεγχο CRC αναστρέφοντας το κάθε byte που εξετάζει, και επομένως το αποτέλεσμα δεν είναι συμβατό με τον συνηθισμένο τρόπο υπολογισμού. Στην περίπτωση αυτή, προτιμήθηκε η υλοποίηση ενός γενικευμένου αλγόριθμου υπολογισμού του CRC, ο οποίος θα εφαρμόζεται σε ανεστραμμένα bytes κάθε φορά. Με αυτό τον τρόπο η συνάρτηση υπολογισμού του CRC δεν περιορίζεται μόνο στα δεδομένα του αισθητήρα Sensirion. Έτσι, με τη χρήση της συνάρτησης bitswapbyte, ο υπολογισμός του ελέγχου CRC στον αισθητήρα και τον μικροελεγκτή συμπίπτει.

10.9 Σταθερότητα του συστήματος

Στην παράγραφο αυτή εξετάζεται η αντίδραση του συστήματος όταν συμβεί σφάλμα σε κάποιο από τα υποσυστήματα που το αποτελούν. Αρχικά θα μελετήσουμε την περίπτωση που ο server βρεθεί σε αόριστη κατάσταση. Σε αυτή την περίπτωση αναμένεται ότι μετά από σύντομο χρονικό διάστημα ο watchdog timer θα επανεκκινήσει το σύστημα. Μέχρι ο server να εκκινήσει ξανά, ο WiFi router λειτουργεί κανονικά και μπορεί να δέχεται μηνύματα χωρίς κανένα απολύτως πρόβλημα, ενώ όταν ο server αρχικοποιηθεί ξανά, η επικοινωνία με τον router συνεχίζει ανεπηρέαστη. Ομοίως, μια αστοχία του server δεν επηρεάζει καθόλου τη βάση MySQL, ούτε τον WiFi router που υπάρχει στον υπολογιστή που εκτελείται ο logger. Επίσης ο Apache web server δεν αντιμετωπίζει πρόβλημα, αφού στις σελίδες που εμφανίζεται η ιστοσελίδα που δημιουργεί ο server, θα τυπωθεί απλά ένα μήνυμα “not found”. Ο logger θα πάψει να λαμβάνει μετρήσεις από τον server, αλλά αυτό δεν αποτελεί πρόβλημα, αφού στον logger θα τεθεί σε κατάσταση αναμονής μέχρι να ενεργοποιηθεί ξανά ο server. Επίσης ο logger μπορεί να αντιμετωπίσει χωρίς πρόβλημα και την περίπτωση που ο server καταρρεύσει χωρίς να έχει ολοκληρωθεί η αποστολή όλων των περιεχομένων της μνήμης eeprom.

Μια άλλη περίπτωση είναι να μην λειτουργεί σωστά ο WiFi router που συνδέεται με τον server. Και σε αυτή την περίπτωση ο logger λειτουργεί χωρίς πρόβλημα όπως και στην πρώτη περίπτωση, ενώ δεν επηρεάζεται καθόλου η βάση MySQL, ο Apache web server και ο WiFi router στον υπολογιστή που εκτελείται ο logger. Αν ο WiFi router έχει αποσυνδεθεί, ο server περιμένει χωρίς κανένα πρόβλημα μέχρι να ανακτηθεί η σύνδεση, διαφορετικά η επανεκκίνηση του router θα πραγματοποιηθεί όταν ισχύουν οι συνθήκες που προβλέπονται από το προφίλ διαχείρισης της ενέργειας.

Μια τρίτη περίπτωση είναι αυτή κατά την οποία έχει αποσυνδεθεί ο WiFi router στον υπολογιστή που εκτελείται ο logger. Σε αυτή την περίπτωση ο server λειτουργεί κανονικά και αυτόνομα, χωρίς όμως να μπορεί να αποθηκεύσει δεδομένα στην μνήμη eeprom, αφού η ημερομηνία και η ώρα δεν θα μπορούν να ανανεωθούν από τον logger. Από τα υπόλοιπα υποσυστήματα, ο logger δεν παρουσιάζει πρόβλημα αφού όπως είδαμε προηγουμένως μπορεί να χειριστεί άνετα καταστάσεις αποσύνδεσης από τον server. Ομοίως ο Apache web server και η βάση MySQL λειτουργούν χωρίς πρόβλημα. Παρόλα αυτά το πρόβλημα του WiFi router που βρίσκεται στον υπολογιστή που εκτελείται ο logger πρέπει να διορθωθεί από τον χρήστη, αφού δεν είναι δυνατή η αυτόματη αποκατάσταση όπως στην περίπτωση του server.

Στην περίπτωση που ο logger παρουσιάσει δυσλειτουργία, αυτή δεν διορθώνεται αυτόματα, και πρέπει να επιληφθεί ο χρήστης. Όσο ο logger βρίσκεται εκτός λειτουργίας, ο server και ο WiFi router με τον οποίο επικοινωνεί, λειτουργούν κανονικά, αλλά όπως είδαμε δεν είναι δυνατή η αποθήκευση δεδομένων στην μνήμη eeprom. Η βάση MySQL δεν επηρεάζεται, ενώ όλες οι σελίδες του Apache web server λειτουργούν κανονικά, αλλά παρουσιάζουν παλαιότερα δεδομένα.

Αν εμφανιστεί σφάλμα στη βάση MySQL, όλα τα συστήματα λειτουργούν κανονικά, με τη διαφορά ότι ο logger παρακάμπτει την αποθήκευση στη βάση, μέχρι να αποκατασταθεί από τον χρήστη το σφάλμα. Τέλος αν παρουσιαστεί σφάλμα στον apache web server, όλα τα υπόλοιπα υποσυστήματα δεν επηρεάζονται, ενώ και πάλι ο χρήστης οφείλει να αποκαταστήσει το σφάλμα.

Επίλογος

Στο σημείο αυτό ολοκληρώνεται η παρουσίαση του ενσωματωμένου συστήματος που υλοποιήσαμε. Συμπερασματικά αξίζει να αναφέρουμε ότι οι στόχοι της διπλωματικής εργασίας επιτεύχθηκαν και οι γνώσεις και εμπειρίες που αποκτήθηκαν είναι ανεκτίμητες. Πολύ σημαντικό κομμάτι της διπλωματικής αποτελεί σίγουρα η κατασκευή του ίδιου του κυκλώματος, που από μόνη της ήταν μια μεγάλη πρόκληση. Από τη σχεδίαση στον υπολογιστή και την εκτύπωση και εμφάνιση του κυκλώματος με λάμπα ατμών υδραργύρου και καυστική σόδα, μέχρι την βηματική εκτέλεση των εντολών με χρήση του AVRdragon. Κάθε στάδιο της κατασκευής προσφέρει στον νέο μηχανικό τα απαραίτητα εφόδια για ενασχόληση με ακόμα πιο επίπονα και φιλόδοξα σχέδια.

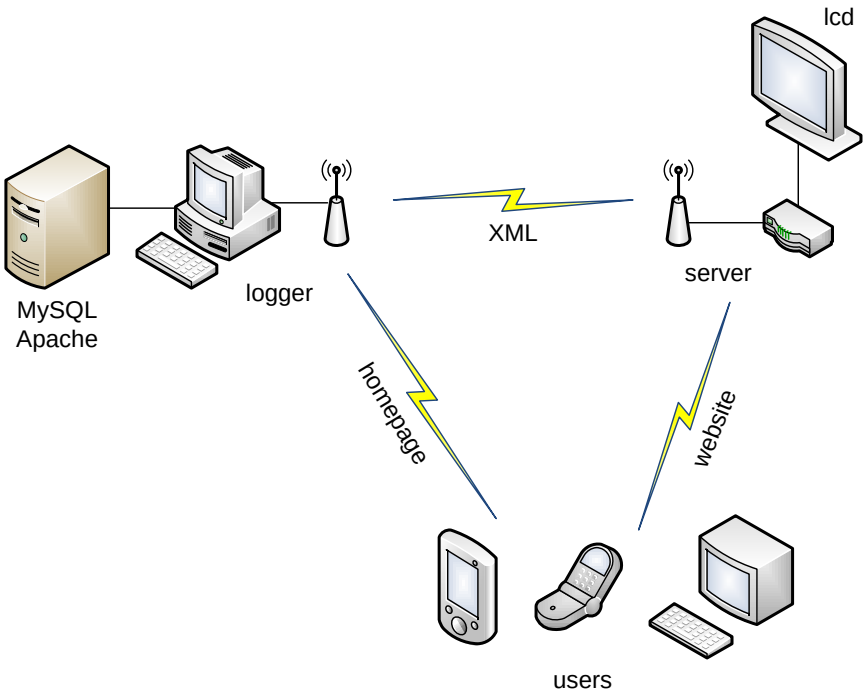
Το πραγματικό κύκλωμα δοκιμάστηκε σε πραγματικές συνθήκες λειτουργίας για ένα εύλογο χρονικό διάστημα, και τα αποτελέσματα ήταν περισσότερο από ικανοποιητικά. Το σύστημα διαχείρισης της ενέργειας πέτυχε την βέλτιστη αξιοποίηση των φυσικών πόρων, ενώ η ακρίβεια των μετρήσεων ήταν θεαματική. Έτσι είναι πλέον δυνατή η μόνιμη εγκατάσταση του κυκλώματος για τη ουσιαστική αξιοποίησή του.

Έχοντας ολοκληρωθεί πλέον το σύστημα, μπορούμε να αναφέρουμε και ορισμένες προτάσεις για μελλοντικές βελτιώσεις. Σε μελλοντικές εκδόσεις του λογισμικού μπορεί να προστεθεί η δυνατότητα ελέγχου από τον server της κατάστασης του router και αν χρειάζεται να τον επανεκκινεί. Μια ακόμα πρόταση θα ήταν η ιστοσελίδα του server να αποθηκεύεται από τον logger και να στέλνεται έμμεσα στον Apache. Τέλος, σε περίπτωση νέο σχεδιασμού του συστήματος, κρίνεται απαραίτητη η χρήση μεγαλύτερου μικροελεγκτή, ο οποίος να επικοινωνεί με εξωτερική μνήμη flash για αποθήκευση των μετρήσεων.

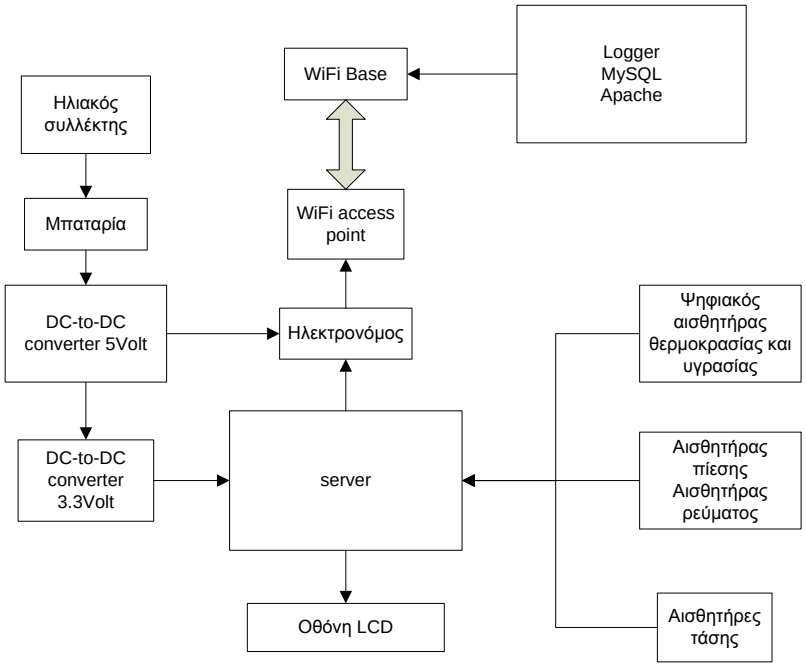
Σε κάθε περίπτωση η δεδομένη υλοποίηση καταφέρνει να κεντρίσει το ενδιαφέρον του μηχανικού και να προκαλέσει τις απαραίτητες ζυμώσεις που τον οδηγούν στην αναζήτηση ακόμα πιο πολύπλοκων συστημάτων.

Παράρτημα Α

Ο χάρτης του συστήματος

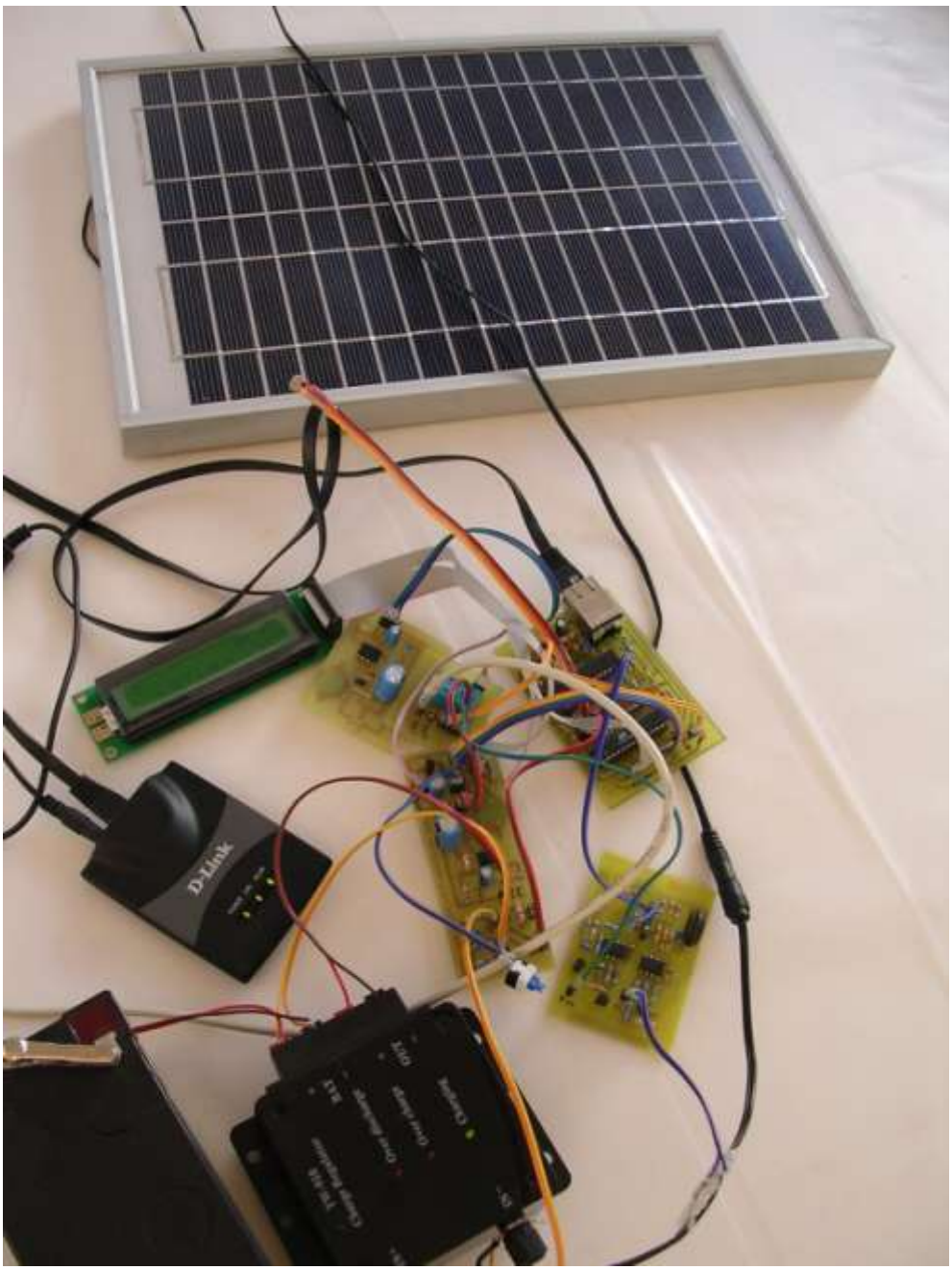


Εικόνα 82: Ο χάρτης του συστήματος

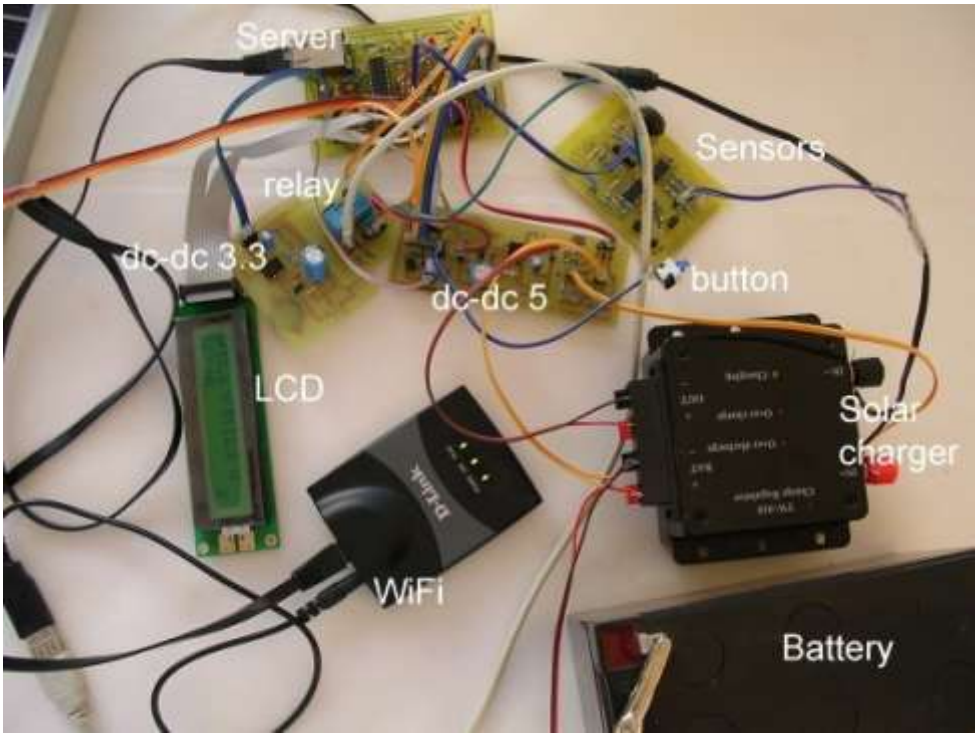


Εικόνα 83: Οι διασυνδεδεμένες συσκευές

Φωτογραφίες του συστήματος



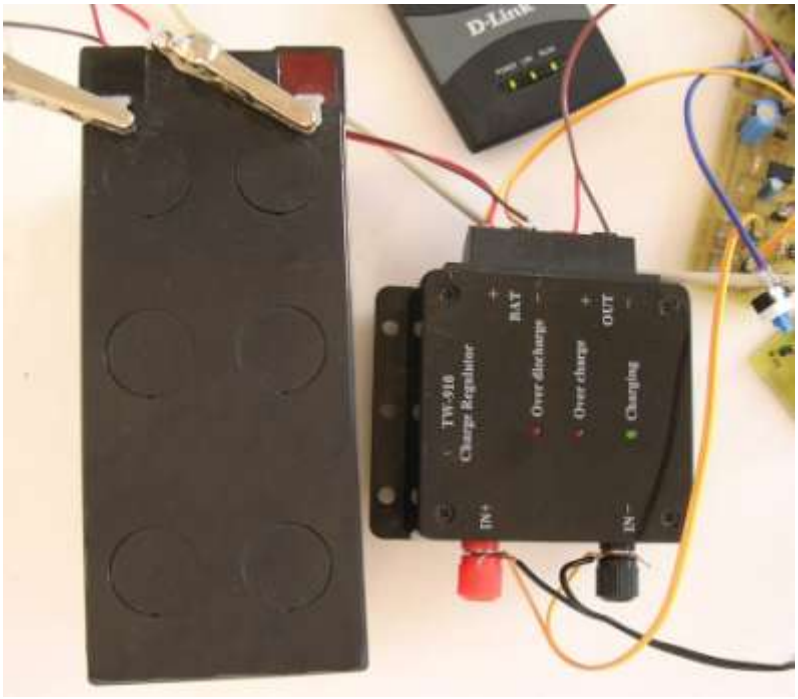
Εικόνα 84: Φωτογραφία του συστήματος



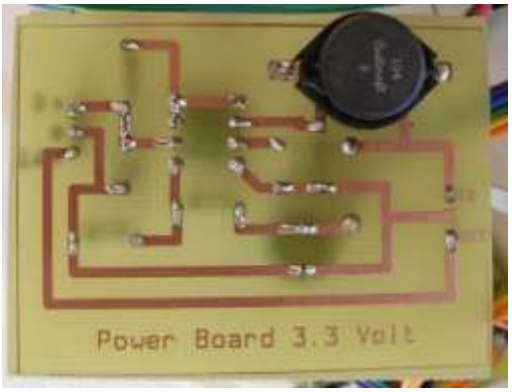
Εικόνα 85: Υποσυστήματα

Εικόνα 86: Η οθόνη lcd

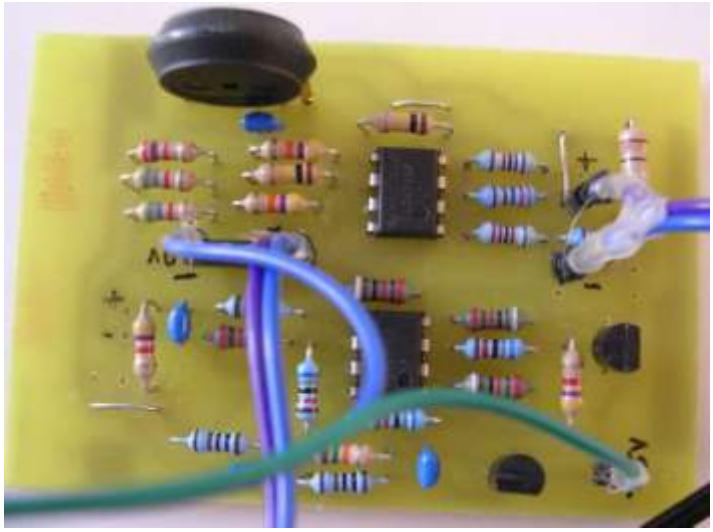
Εικόνα 87: Ο ηλιακός συλλέκτης



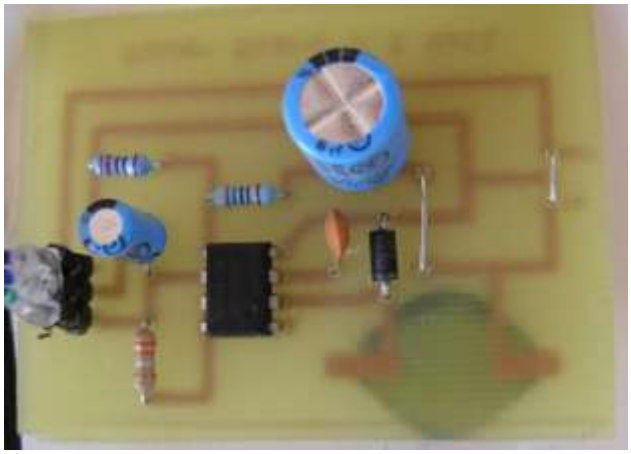
Εικόνα 88: Η μπαταρία και ο φορτιστής



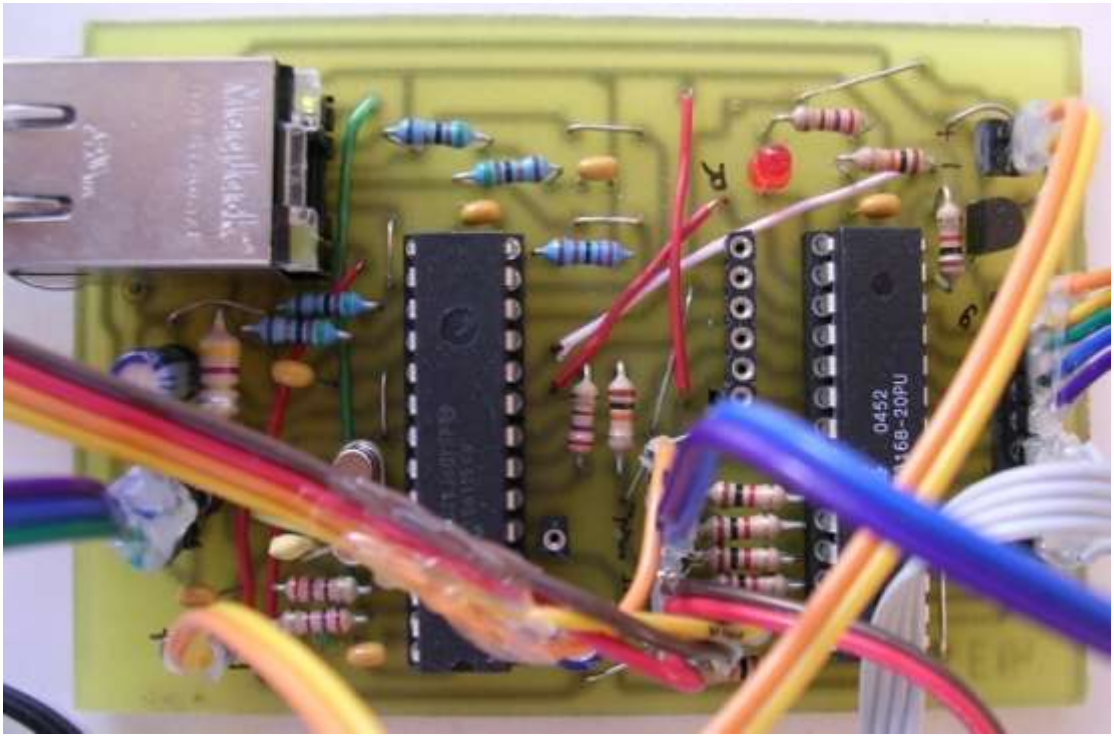
Εικόνα 89: Το πηνίο συσκευασίας SMD



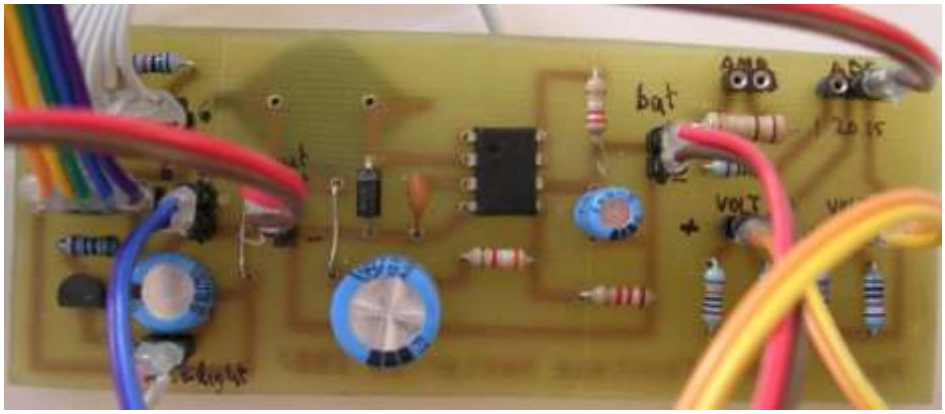
Εικόνα 90: Αναλογικός αισθητήρας πίεσης



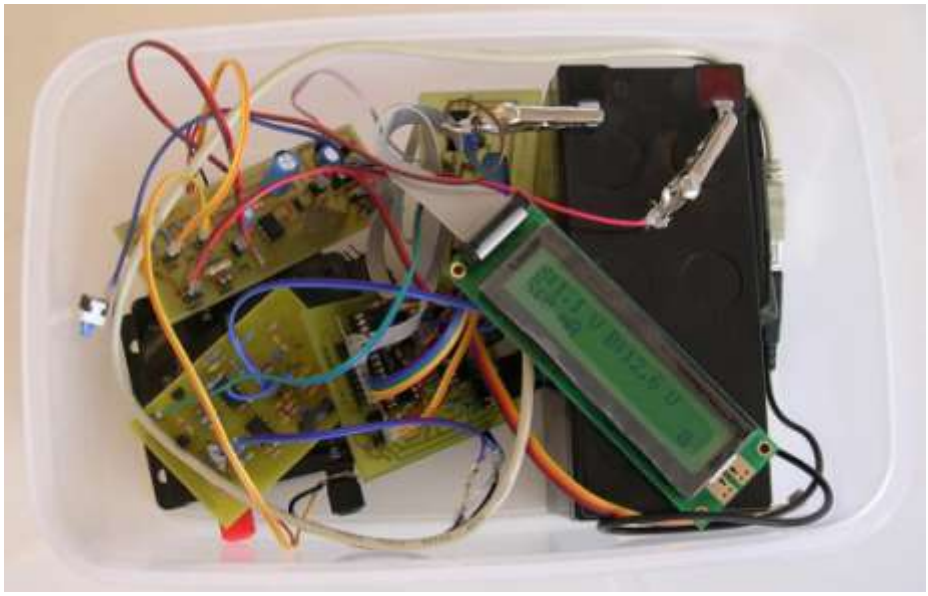
Εικόνα 91: DC-to-DC converter 3.3Volt



Εικόνα 92: O server



Εικόνα 93: DC-to-DC converter 5Volt



Εικόνα 94: Το σύστημα εντός του δοχείου προστασίας

Εικόνα 95: Το σύστημα σφραγισμένο

Εικόνα 96: Συσκευασίες σιλικόνης κατά της υγρασίας

Παράρτημα Β

Ο κώδικας του server

analog.c

```
// Επεξεργασία αναλογικών δεδομένων από αισθητήρες

#include <avr/io.h>
#include "config.h"

extern char wifi_status;   // κατάσταση wifi

// Επιστρέφει την αναλογική τιμή ενός καναλιού του ADC
// Λειτουργεί χωρίς Διακοπές
int convertanalog(unsigned char channel)
{
	unsigned char adlow,adhigh;
	int result;

/*	Λειτουργία single run χωρίς noise canceler

	Ονομασία bits
	ADEN: Analog Digital Converter Enable
	ADIE: ADC Interrupt Enable  (0=no interrupt)
	ADIF: ADC Interrupt Flag
	ADCSR: ADC Control and Status Register
	ADPS2..ADPS0: ADC Prescaler Select Bits
	REFS: Reference Selection Bits
	*/

#ifdef USEAVCCREF
	// αναφορά AVcc
	ADMUX=(1<<REFS0)|(channel & 0x0f);
#else
	// χρήση εσωτερικής τάσης αναφοράς 1.1Volt
	ADMUX=(1<<REFS1)|(1<<REFS0)|(channel & 0x0f);
#endif
	// διαιρέτης ρολογιού 128 για συχνότητες 50-200kHz
	ADCSRA=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
	// απενεργοποίηση ψηφιακής γραμμής εισόδου
	//DIDR0=(1<<channel)& 0x1f;

	// έναρξη μετατροπής
	ADCSRA|= (1<<ADSC);
	while(bit_is_set(ADCSRA,ADSC));  // αναμονή αποτελέσματος
	adlow=ADCL;                      // διάβασμα low byte
	adhigh=ADCH;
	result=(adhigh<<8)|(adlow & 0xFF);

	// δημιουργία μέσης τιμής
	ADCSRA|= (1<<ADSC);
	while(bit_is_set(ADCSRA,ADSC));  // αναμονή αποτελέσματος
	adlow=ADCL;                      // διάβασμα low byte
	adhigh=ADCH;
	result+=(adhigh<<8)|(adlow & 0xFF);
	return(result/2);
}

// Αισθητήρας τάσης ηλιακού συλλέκτη, αποτελέσματα σε δέκατα του Volt
int voltsolar(int adcval){
```

```
	int i;
	i=adcval;
	//καμπύλη μετασχηματισμού
	i=31*i+569;
	i/=139;
	i=i+(7*wifi_status);
	return(i);
}

// Αισθητήρας τάσης μπαταρίας, αποτελέσματα σε δέκατα του Volt
int voltbattery(int adcval){
	int i;
	i=adcval;
	//καμπύλη μετασχηματισμού
	i=30*i+669;
	i/=180;
	i=i+(8*wifi_status)+2;
	return(i);
}

// Αισθητήρας ρεύματος ηλιακού συλλέκτη, αποτελέσματα σε milli amps
int ampsolar(int adcval){
	int i;
	i=adcval;
	//καμπύλη μετασχηματισμού
	i=28*i-602;
	i/=37;
	if (i<0)
		i=0;
	return(i);
}


// Αισθητήρας θερμοκρασίας 0, αποτελέσματα σε δέκατα του βαθμού Κελσίου
int tempcelsius0(int adcval){
	int i;
	//καμπύλη μετασχηματισμού
	i=(adcval *10);
	i/=KELVINPERADC0;
	i=i*10 - OFFSET0;
	return(i);
}

// Αισθητήρας θερμοκρασίας 1, αποτελέσματα σε δέκατα του βαθμού Κελσίου
int tempcelsius1(int adcval){
	int i;
	//καμπύλη μετασχηματισμού
	i=(adcval *10);
	i/=KELVINPERADC1;
	i=i*10 - OFFSET1;
	return(i);
}

// Μετατροπή δεκαπλάσιου θερμοκρασίας Κελσίου
// στο δεκαπλάσιο θερμοκρασίας Φαρενάιτ
int c2f(int celsiustimes10){
	return((celsiustimes10 * 18 + 3200)/10);
}

// Αισθητήρας πίεσης, αποτελέσματα σε δέκατα του hPa
int pressure(int adcval){
	int i;
	//καμπύλη μετασχηματισμού
	i=(adcval+(ERROR_A))*25/GAIN_A;
	i=i*4;
	return(i+REFTERM_A);
}
```

```
// Μετατροπή πίεσης στο επίπεδο της θάλασσας
int pressureATsealevel(int phpa10){
	return(phpa10+SEALEVELOFFSET);
}
```

enc28j60.c

```
// Συναρτήσεις λειτουργιών του chip ENC28J60

#include <avr/io.h>
#include "avr_compat.h"
#include "enc28j60.h"
#include "timeout.h"

#ifndef F_CPU
#define F_CPU 8000000UL
#endif
#ifndef ALIBC_OLD
#include <util/delay.h>
#else
#include <avr/delay.h>
#endif

static uint8_t Enc28j60Bank;
static uint16_t NextPacketPtr;
// γραμμές επικοινωνίας
#define ENC28J60_CONTROL_PORT	PORTB
#define ENC28J60_CONTROL_DDR	DDRB
#define ENC28J60_CONTROL_CS		2
// ορισμός CS στο 0 = active
#define CSACTIVE ENC28J60_CONTROL_PORT&=~(1<<ENC28J60_CONTROL_CS)
// ορισμός CS στο 1 = passive
#define CSPASSIVE ENC28J60_CONTROL_PORT|=(1<<ENC28J60_CONTROL_CS)
//μακροεντολή
#define waitspi() while(!(SPSR&(1<<SPIF)))


//Συναρτήσεις
//datasheet page 28+

// ανάγνωση byte από τη μνήμη του enc28j60
uint8_t enc28j60ReadOp(uint8_t op, uint8_t address)
{
	CSACTIVE;	// ενεργοποίηση γραμμής chip select

	SPDR = op | (address & ADDR_MASK);	// αποστολή εντολής ανάγνωσης
	waitspi();

	SPDR = 0x00;  // ανάγνωση byte
	waitspi();

	if(address & 0x80)  // ανάγνωση ακόμα και αν δεν υπάρχουν δεδομένα
	{
		SPDR = 0x00;
		waitspi();
	}

	CSPASSIVE;	// απελευθέρωση γραμμής chip select
	return(SPDR);
}

// εγγραφή byte από τη μνήμη του enc28j60
void enc28j60WriteOp(uint8_t op, uint8_t address, uint8_t data)
{
	CSACTIVE;	// ενεργοποίηση γραμμής chip select

	SPDR = op | (address & ADDR_MASK);	// αποστολή εντολής εγγραφής
```

```
	waitspi();

	SPDR = data;  // εγγραφή byte
	waitspi();
	CSPASSIVE;		// απελευθέρωση γραμμής chip select
}

// ανάγνωση buffer memory του enc28j60
void enc28j60ReadBuffer(uint16_t len, uint8_t* data)
{
	CSACTIVE;	// ενεργοποίηση γραμμής chip select

	SPDR = ENC28J60_READ_BUF_MEM;	// αποστολή εντολής ανάγνωσης
	waitspi();
	while(len)
	{
		len--;

		SPDR = 0x00;  // ανάγνωση buffer
		waitspi();
		*data = SPDR;
		data++;
	}
	*data='\0';
	CSPASSIVE;	// απελευθέρωση γραμμής chip select
}

// εγγραφή buffer memory του enc28j60
void enc28j60WriteBuffer(uint16_t len, uint8_t* data)
{
	CSACTIVE;	// ενεργοποίηση γραμμής chip select

	SPDR = ENC28J60_WRITE_BUF_MEM;	// αποστολή εντολής εγγραφής
	waitspi();
	while(len)
	{
		len--;

		SPDR = *data;// εγγραφή buffer
		data++;
		waitspi();
	}
	CSPASSIVE;	// απελευθέρωση γραμμής chip select
}

// Ορισμός bank
void enc28j60SetBank(uint8_t address)
{
	if((address & BANK_MASK) != Enc28j60Bank)
	{
		enc28j60WriteOp(ENC28J60_BIT_FIELD_CLR, ECON1,
(ECON1_BSEL1|ECON1_BSEL0));
		enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, ECON1, (address &
BANK_MASK)>>5);
		Enc28j60Bank = (address & BANK_MASK);
	}
}

// Λειτουργία ανάγνωσης control register του enc28j60
uint8_t enc28j60Read(uint8_t address)
{
	// ορισμός bank
	enc28j60SetBank(address);
	// ανάγνωση
	return enc28j60ReadOp(ENC28J60_READ_CTRL_REG, address);
}

// Λειτουργία εγγραφής control register του enc28j60
```

```
void enc28j60Write(uint8_t address, uint8_t data)
{
	// ορισμός bank
	enc28j60SetBank(address);
	// εγγραφή
	enc28j60WriteOp(ENC28J60_WRITE_CTRL_REG, address, data);
}

// Λειτουργία εγγραφής PHY data
void enc28j60PhyWrite(uint8_t address, uint16_t data)
{
	// ορισμός διεύθυνσης PHY register
	enc28j60Write(MIREGADR, address);
	// εγγραφή PHY data
	enc28j60Write(MIWRL, data);
	enc28j60Write(MIWRH, data>>8);
	// αναμονή μέχρι να ολοκληρωθεί η εγγραφή
	while(enc28j60Read(MISTAT) & MISTAT_BUSY){
		_delay_us(15);
	}
}

// Ορισμός συχνότητας ρολογιού
void enc28j60clkout(uint8_t clk)
{
	//clk=2 αντιστοιχεί σε 12.5MHz
	enc28j60Write(ECOCON, clk & 0x7);
}

void enc28j60Init(uint8_t* macaddr)
{
	// αρχικοποίηση I/O

	// ορισμός cs σαν έξοδος
	ENC28J60_CONTROL_DDR |= 1<<ENC28J60_CONTROL_CS;
	CSPASSIVE; // cs=0

	DDRB  |= 1<<PB3 | 1<<PB5;  // mosi, sck είναι έξοδος
	cbi(DDRB,PINB4);    // MISO είναι είσοδος
	//
	cbi(PORTB,PB3);     // MOSI κατάσταση low
	cbi(PORTB,PB5);     // SCK κατάσταση low

	// αρχικοποίηση SPI interface
	SPCR = (1<<SPE)|(1<<MSTR); // master mode
	SPSR |= (1<<SPI2X); //ρολόι Fosc/2

	// reset συστήματος
	enc28j60WriteOp(ENC28J60_SOFT_RESET, 0, ENC28J60_SOFT_RESET);
	delay_ms(50);

	// εγγραφή μηδέν στο EPKTCNT
	enc28j60Write(EPKTCNT, 0);

	// έλεγχος CLKRDY bit για ολοκλήρωση reset
	//while(!(enc28j60Read(ESTAT) & ESTAT_CLKRDY));

	// αρχικοποίηση buffer λήψης
	NextPacketPtr = RXSTART_INIT;
	// έναρξη λήψης
	enc28j60Write(ERXSTL, RXSTART_INIT&0xFF);
	enc28j60Write(ERXSTH, RXSTART_INIT>>8);
	// ορισμός pointer διέυθυνσης λήψης
	enc28j60Write(ERXRDPTL, RXSTART_INIT&0xFF);
	enc28j60Write(ERXRDPTH, RXSTART_INIT>>8);
	// τέλος λήψης
	enc28j60Write(ERXNDL, RXSTOP_INIT&0xFF);
	enc28j60Write(ERXNDH, RXSTOP_INIT>>8);
```

```
        // ένρξη μετάδοσης
        enc28j60Write(ETXSTL, TXSTART_INIT&0xFF);
        enc28j60Write(ETXSTH, TXSTART_INIT>>8);
        // τέλος μετάδοσης
        enc28j60Write(ETXNDL, TXSTOP_INIT&0xFF);
        enc28j60Write(ETXNDH, TXSTOP_INIT>>8);

        enc28j60Write(ERXFCON, ERXFCON_UCEN|ERXFCON_CRCEN|ERXFCON_PMEN);
        enc28j60Write(EPMM0, 0x3f);
        enc28j60Write(EPMM1, 0x30);
        enc28j60Write(EPMCSL, 0xf9);
        enc28j60Write(EPMCSH, 0xf7);

        // ενεργοποίηση λήψης MAC
        enc28j60Write(MACON1, MACON1_MARXEN|MACON1_TXPAUS|MACON1_RXPAUS);
        enc28j60Write(MACON2, 0x00);
        // ενεργοποίηση αυτόματου padding στα 60 bytes και λειτουργίες CRC
        enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, MACON3,
MACON3_PADCFG0|MACON3_TXCRCEN|MACON3_FRMLNEN);
        // ορισμός κενού μεταξύ πλαισίων (όχι back-to-back)
        enc28j60Write(MAIPGL, 0x12);
        enc28j60Write(MAIPGH, 0x0C);
        // ορισμός κενού μεταξύ πλαισίων (back-to-back)
        enc28j60Write(MABBIPG, 0x12);
        // ορισμός μέγιστου πλαισίου
        enc28j60Write(MAMXFLL, MAX_FRAMELEN&0xFF);
        enc28j60Write(MAMXFLH, MAX_FRAMELEN>>8);
        // εγγραφή διεύθυνσης MAC
        enc28j60Write(MAADR5, macaddr[0]);
        enc28j60Write(MAADR4, macaddr[1]);
        enc28j60Write(MAADR3, macaddr[2]);
        enc28j60Write(MAADR2, macaddr[3]);
        enc28j60Write(MAADR1, macaddr[4]);
        enc28j60Write(MAADR0, macaddr[5]);

        enc28j60PhyWrite(PHCON2, PHCON2_HDLDIS);

        enc28j60SetBank(ECON1);
        // ενεργοποίηση διακοπών
        enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, EIE, EIE_INTIE|EIE_PKTIE);
        // ενεργοποίηση λήψης πακέτου
        enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, ECON1, ECON1_RXEN);
}

// ανάγνωση έκδοσης chip
uint8_t enc28j60getrev(void)
{
        return(enc28j60Read(EREVID));
}

// αποστολή πακέτου
void enc28j60PacketSend(uint16_t len, uint8_t* packet)
{
        // ορισμός pointer στην αρχή του buffer
        enc28j60Write(EWRPTL, TXSTART_INIT&0xFF);
        enc28j60Write(EWRPTH, TXSTART_INIT>>8);
        // ορισμός του TXND pointer στο ορισμένο μέγεθος πακέτου
        enc28j60Write(ETXNDL, (TXSTART_INIT+len)&0xFF);
        enc28j60Write(ETXNDH, (TXSTART_INIT+len)>>8);
        // εγγραφή byte ελέγχου
        enc28j60WriteOp(ENC28J60_WRITE_BUF_MEM, 0, 0x00);
        // αντιγραφή πακέτου στον buffer αποστολής
        enc28j60WriteBuffer(len, packet);
        // αποστολή των δεδομένων του buffer στο δίκτυο
        enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, ECON1, ECON1_TXRTS);
        // reset της ακολουθίας αποστολής
        if( (enc28j60Read(EIR) & EIR_TXERIF) ){
                enc28j60WriteOp(ENC28J60_BIT_FIELD_CLR, ECON1, ECON1_TXRTS);
```

```
	}
}

// αποδοχή πακέτου από τον buffer λήψης δικτύου αν υπάρχει
// το πακέτο έχει επικεφαλίδα ethernet:
//           maxlen  το μέγιστο μήκος πακέτου
//           packet  δείκτης στα δεδομένα του πακέτου
// επιστρέφει: μέγεθος πακέτου που λήφθηκε, ή μηδέν
uint16_t enc28j60PacketReceive(uint16_t maxlen, uint8_t* packet)
{
	uint16_t rxstat;
	uint16_t len;
	// έλεγχος αν λήφθηκε το πακέτο
	if( enc28j60Read(EPKTCNT) ==0 ){
		return(0);
	}

	// τοποθέτηση του pointer ανάγνωσης στην αρχή του ληφθέντως πακέτου
	enc28j60Write(ERDPTL, (NextPacketPtr));
	enc28j60Write(ERDPTH, (NextPacketPtr)>>8);
	// ανάγνωση του pointer του επόμενου πακέτου
	NextPacketPtr  = enc28j60ReadOp(ENC28J60_READ_BUF_MEM, 0);
	NextPacketPtr |= enc28j60ReadOp(ENC28J60_READ_BUF_MEM, 0)<<8;
	// ανάγνωση του μήκου πακέτου
	len  = enc28j60ReadOp(ENC28J60_READ_BUF_MEM, 0);
	len |= enc28j60ReadOp(ENC28J60_READ_BUF_MEM, 0)<<8;
	len-=4; //αφαίρεση του CRC
	// άνάγνωση της κατάστασης λήψης
	rxstat  = enc28j60ReadOp(ENC28J60_READ_BUF_MEM, 0);
	rxstat |= enc28j60ReadOp(ENC28J60_READ_BUF_MEM, 0)<<8;
	// περιορισμός του ορίου ανάκτησης
	if (len>maxlen-1){
		len=maxlen-1;
	}
	// έλεγχος CRC
	if ((rxstat & 0x80)==0){
		// invalid
		len=0;
	}else{
		// αντιγραφή πακέτου από τον buffer λήψης
		enc28j60ReadBuffer(len, packet);
	}
	// μετακίνηση του pointer λήψης στην αρχή του επόμενου πακέτου
	// και γίνεται απελευθέρωση μνήμης
	enc28j60Write(ERXRDPTL, (NextPacketPtr));
	enc28j60Write(ERXRDPTH, (NextPacketPtr)>>8);
	// μείωση του μετρητή πακέτων
	enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, ECON2, ECON2_PKTDEC);
	return(len);
}
```

integrator.c

```
#include <stdlib.h>
#include <stdio.h>
#include "integrator.h"
#include "analog.h"

static uint8_t battery[SAMPLES];
static int bat_total=0;
static uint8_t bat_index=0;      //oldest value
static uint8_t bat_mean=0;

// γέμισμα του πίνακα δεδομένων
void initbat() {

	for (bat_index=0; bat_index < SAMPLES; bat_index++) {
```

```
                battery[bat_index] = voltbattery(convertanalog(3))-6;
                bat_total += battery[bat_index];
        }
        bat_index=0;
}

// επιστρέφει το νέο μέσο όρο τάσης της μπαταρίας
// αντικαθηστά την παλιά τιμή
uint8_t readbat() {

        int temp;

        temp=convertanalog(3);
        temp+=convertanalog(3);
        temp+=convertanalog(3);
        temp+=convertanalog(3);
        temp+=convertanalog(3);
        temp/=5;

        temp=voltbattery(temp);

        bat_total-=battery[bat_index];
        battery[bat_index]=temp;
        bat_total+=temp;

        bat_mean=bat_total >> SAMPLES_BITS;

        bat_index++;
        if (bat_index == SAMPLES)
                bat_index=0;

        return bat_mean;
}
```

ip_arp_udp_tcp.c

```
// Συναρτήσεις TCP, ARP, UDP και IP
// Υλοποίηση του TCP stack χωρίς fragmentation (single data packet TCP)
// όλα τα δεδομένα στέλνονται σε ένα πακέτο

#include <avr/io.h>
#include <avr/pgmspace.h>
#include "avr_compat.h"
#include "net.h"
#include "enc28j60.h"

static uint8_t wwwport=80;
static uint8_t macaddr[6];
static uint8_t ipaddr[4];
static int16_t info_hdr_len=0;
static int16_t info_data_len=0;
static uint8_t seqnum=0xa; // αρχικός αριθμός για ακολουθία tcp

// Συνάρτηση checksum

// Το IP checksum υπολογίζεται στην κεφαλίδα IP σε μήκος 20 bytes
// και αρχικοποιείται στο μηδέν
// Στις περιπτώσεις των πακέτων UDP/TCP δεν δημιουργούμε ψευδοεπικεφαλίδα
// αλλά χρησιμοποιούμε τα πεδία ip.src και ip.dst του πραγματικού πακέτου
// Συνολικά:
// Μήκος UDP: 4+4 + 8  + μήκος data
// Μήκος TCP: 4+4 + 20 + μήκος options + μήκος data
uint16_t checksum(uint8_t *buf, uint16_t len,uint8_t type){
        // τύπος     0=ip
        //                 1=udp
        //                 2=tcp
        uint32_t sum = 0;
```

```
        //if(type==0){
        // τίποτα δεν προστίθεται
        //}
        if(type==1){
                sum+=IP_PROTO_UDP_V;        // πρωτόκολλο udp
                // το συνολικό μήκος είναι το μήκος του udp (μήκος data+header)
                // = μήκος εισόδου - (μήκος IP.scr+IP.dst)
                sum+=len-8;   // = πραγματικό μήκος udp
        }
        if(type==2){
                sum+=IP_PROTO_TCP_V;
                // το συνολικό μήκος είναι το μήκος του tcp (μήκος data+header)
                // = μήκος εισόδου - (μήκος IP.scr+IP.dst)
                sum+=len-8;   // = πραγματικό μήκος tcp
        }
        // άθροισμα από λέξεις 16bit
        while(len >1){
                sum += 0xFFFF & (*buf<<8|*(buf+1));
                buf+=2;
                len-=2;
        }
        // προσθήκη υπολοιπόμενου byte
        if (len){
                sum += (0xFF & *buf)<<8;
        }
        // υπολογισμός του συνολικού αθροίσματος
        // μέχρι το αποτέλεσμα να είναι 16 bit
        while (sum>>16){
                sum = (sum & 0xFFFF)+(sum >> 16);
        }
        // συμπλήρωμα ως προς 1
        return( (uint16_t) sum ^ 0xFFFF);
}

// συνάρτηση αρχικοποίησης
void init_ip_arp_udp_tcp(uint8_t *mymac,uint8_t *myip,uint8_t wwwp){
        uint8_t i=0;
        wwwport=wwwp;
        while(i<4){
                ipaddr[i]=myip[i];
                i++;
        }
        i=0;
        while(i<6){
                macaddr[i]=mymac[i];
                i++;
        }
}

// Έλεγχος τύπου πακέτων
uint8_t eth_type_is_arp_and_my_ip(uint8_t *buf,uint16_t len){
        uint8_t i=0;

        if (len<41){
                return(0);
        }
        if(buf[ETH_TYPE_H_P] != ETHTYPE_ARP_H_V ||
                buf[ETH_TYPE_L_P] != ETHTYPE_ARP_L_V){
                        return(0);
        }
        while(i<4){
                if(buf[ETH_ARP_DST_IP_P+i] != ipaddr[i]){
                        return(0);
                }
                i++;
        }
        return(1);
```

```
}

// Έλεγχος τύπου πακέτων
uint8_t eth_type_is_ip_and_my_ip(uint8_t *buf,uint16_t len){
	uint8_t i=0;
	//eth+ip+κεφαλίδα udp = 42
	if (len<42){
		return(0);
	}
	if(buf[ETH_TYPE_H_P]!=ETHTYPE_IP_H_V ||
		buf[ETH_TYPE_L_P]!=ETHTYPE_IP_L_V){
			return(0);
	}
	if (buf[IP_HEADER_LEN_VER_P]!=0x45){
		// επικεφαλίδα IP v4 των 20 bytes
		return(0);
	}
	while(i<4){
		if(buf[IP_DST_P+i]!=ipaddr[i]){
			return(0);
		}
		i++;
	}
	return(1);
}

// δημιουργία επικεφαλίδας eth από το πακέτο eth που λήφθηκε
void make_eth(uint8_t *buf)
{
	uint8_t i=0;

	//αντιστροφή διευθύνσεων mac πηγής και προορισμού
	while(i<6){
		buf[ETH_DST_MAC +i]=buf[ETH_SRC_MAC +i];
		buf[ETH_SRC_MAC +i]=macaddr[i];
		i++;
	}
}

// προσθήκη checksum στην κεφαλίδα
void fill_ip_hdr_checksum(uint8_t *buf)
{
	uint16_t ck;
	// καθαρισμός 2 byte του checksum
	buf[IP_CHECKSUM_P]=0;
	buf[IP_CHECKSUM_P+1]=0;
	buf[IP_FLAGS_P]=0x40;	// χωρίς κατακερματισμό
	buf[IP_FLAGS_P+1]=0;	// offset κατακερματισμού
	buf[IP_TTL_P]=64; // TTL
	// υπολογισμός checksum:
	ck=checksum(&buf[IP_P], IP_HEADER_LEN,0);
	buf[IP_CHECKSUM_P]=ck>>8;
	buf[IP_CHECKSUM_P+1]=ck& 0xff;
}

// δημιουργία κεφαλίδας IP από το πακέτο IP που λήφθηκε
void make_ip(uint8_t *buf)
{
	uint8_t i=0;
	while(i<4){
		buf[IP_DST_P+i]=buf[IP_SRC_P+i];
		buf[IP_SRC_P+i]=ipaddr[i];
		i++;
	}
	fill_ip_hdr_checksum(buf);
}

// δημιουργία κεφαλίδας TCP από το πακέτο TCP που λήφθηκε
```

```
// το rel_ack_num καθορίζει το βήμα αύξησης του sequence
// το πακέτο TCP περιέχει το πολύ 255 bytes δεδομένων
// αν η σημαία mss=1 τότε το mss περιέχεται στα options
// η θέση του πρώτου byte δεδομένων είναι TCP_OPTIONS_P+4
// αν cp_seq=0 τότε γίνεται αρχικοποίηση του sequence (synack)
// διαφορετικά χρησιμοποιείται η τιμή του πακέτου που λήφθηκε
void make_tcphead(uint8_t *buf,uint16_t rel_ack_num,uint8_t mss,uint8_t
cp_seq)
{
	uint8_t i=0;
	uint8_t tseq;
	while(i<2){
		buf[TCP_DST_PORT_H_P+i]=buf[TCP_SRC_PORT_H_P+i];
		buf[TCP_SRC_PORT_H_P+i]=0; // clear source port
		i++;
	}
	// ορισμός θύρας αποστολής (http)
	buf[TCP_SRC_PORT_L_P]=wwwport;
	i=4;
	// αριθμοί sequence
	// πρόσθεση του rel_ack_num στο SEQACK
	while(i>0){
		rel_ack_num=buf[TCP_SEQ_H_P+i-1]+rel_ack_num;
		tseq=buf[TCP_SEQACK_H_P+i-1];
		buf[TCP_SEQACK_H_P+i-1]=0xff&rel_ack_num;
		if (cp_seq){
			// αντιγραφή του acknum που λήφθηκε στο sequence
			buf[TCP_SEQ_H_P+i-1]=tseq;
		}else{
			buf[TCP_SEQ_H_P+i-1]= 0;   // προεπιλεγμένη τιμή
		}
		rel_ack_num=rel_ack_num>>8;
		i--;
	}
	if (cp_seq==0){
		// ορισμός αρχικού αριθμού sequence
		buf[TCP_SEQ_H_P+0]= 0;
		buf[TCP_SEQ_H_P+1]= 0;
		// αυξηση μόνο του δεύτερου byte
		buf[TCP_SEQ_H_P+2]= seqnum;
		buf[TCP_SEQ_H_P+3]= 0;
		// αύξηση κατά 2, αφού τα δεδομένα είναι το πολύ 255
		// δεν θα σταλεί επόμενο πακέτο (fragmentation)
		seqnum+=2;
	}
	// μηδενισμός checksum
	buf[TCP_CHECKSUM_H_P]=0;
	buf[TCP_CHECKSUM_L_P]=0;

	if (mss){
		// θέτουμε το MSS στο 1408:
		// 1408 = 0x580
		buf[TCP_OPTIONS_P]=2;
		buf[TCP_OPTIONS_P+1]=4;
		buf[TCP_OPTIONS_P+2]=0x05;
		buf[TCP_OPTIONS_P+3]=0x80;
		// 24 bytes:
		buf[TCP_HEADER_LEN_P]=0x60;
	}else{
		// χωρίς options:
		// 20 bytes:
		buf[TCP_HEADER_LEN_P]=0x50;
	}
}

// απάντηση σε πακέτο ARP
void make_arp_answer_from_request(uint8_t *buf)
```

```
{
	uint8_t i=0;

	make_eth(buf);
	buf[ETH_ARP_OPCODE_H_P]=ETH_ARP_OPCODE_REPLY_H_V;
	buf[ETH_ARP_OPCODE_L_P]=ETH_ARP_OPCODE_REPLY_L_V;
	// εγγραφή διευθύνσεων mac
	while(i<6){
		buf[ETH_ARP_DST_MAC_P+i]=buf[ETH_ARP_SRC_MAC_P+i];
		buf[ETH_ARP_SRC_MAC_P+i]=macaddr[i];
		i++;
	}
	i=0;
	while(i<4){
		buf[ETH_ARP_DST_IP_P+i]=buf[ETH_ARP_SRC_IP_P+i];
		buf[ETH_ARP_SRC_IP_P+i]=ipaddr[i];
		i++;
	}
	// eth+arp = 42 bytes
	enc28j60PacketSend(42,buf);
}

// απάντηση σε ECHO
void make_echo_reply_from_request(uint8_t *buf,uint16_t len)
{
	make_eth(buf);
	make_ip(buf);
	buf[ICMP_TYPE_P]=ICMP_TYPE_ECHOREPLY_V;
	// αλλαγή ου πεδίου icmp.type από request(=8) σε reply(=0).
	// διορθώνουμε το checksum
	if (buf[ICMP_CHECKSUM_P] > (0xff-0x08)){
		buf[ICMP_CHECKSUM_P+1]++;
	}
	buf[ICMP_CHECKSUM_P]+=0x08;
	//
	enc28j60PacketSend(len,buf);
}

// αποστολή απάντησης UDP
// μέγιστο μήκος data: 220 bytes
void make_udp_reply_from_request(uint8_t *buf,char *data,uint8_t
datalen,uint16_t port)
{
	uint8_t i=0;
	uint16_t ck;
	make_eth(buf);
	if (datalen>220){
		datalen=220;
	}
	// ορισμός μέγιστου μήκους στη κεφαλίδα IP
	buf[IP_TOTLEN_H_P]=0;
	buf[IP_TOTLEN_L_P]=IP_HEADER_LEN+UDP_HEADER_LEN+datalen;
	make_ip(buf);
	buf[UDP_DST_PORT_H_P]=port>>8;
	buf[UDP_DST_PORT_L_P]=port & 0xff;
	// η source port είναι η θύρα του αποστολέα και είναι τυχαία
	// υπολογισμός μήκους udp:
	buf[UDP_LEN_H_P]=0;
	buf[UDP_LEN_L_P]=UDP_HEADER_LEN+datalen;
	// μηδενισμός του checksum
	buf[UDP_CHECKSUM_H_P]=0;
	buf[UDP_CHECKSUM_L_P]=0;
	// αντιγραφή δεδομένων
	while(i<datalen){
		buf[UDP_DATA_P+i]=data[i];
		i++;
	}
	ck=checksum(&buf[IP_SRC_P], 16 + datalen,1);
```

```
	buf[UDP_CHECKSUM_H_P]=ck>>8;
	buf[UDP_CHECKSUM_L_P]=ck& 0xff;
	enc28j60PacketSend(UDP_HEADER_LEN+IP_HEADER_LEN+ETH_HEADER_LEN+datalen
,buf);
}

// αποστολή acknowledge
void make_tcp_synack_from_syn(uint8_t *buf)
{
	uint16_t ck;
	make_eth(buf);
	// συνολικό μήκος καφαλίδας IP:
	// 20 bytes IP + 24 bytes (20tcp+4tcp options)
	buf[IP_TOTLEN_H_P]=0;
	buf[IP_TOTLEN_L_P]=IP_HEADER_LEN+TCP_HEADER_LEN_PLAIN+4;
	make_ip(buf);
	buf[TCP_FLAGS_P]=TCP_FLAGS_SYNACK_V;
	make_tcphead(buf,1,1,0);
	// υπολογισμός checksum
	ck=checksum(&buf[IP_SRC_P], 8+TCP_HEADER_LEN_PLAIN+4,2);
	buf[TCP_CHECKSUM_H_P]=ck>>8;
	buf[TCP_CHECKSUM_L_P]=ck& 0xff;
	// πρόσθεση mss (4 bytes)
	enc28j60PacketSend(IP_HEADER_LEN+TCP_HEADER_LEN_PLAIN+4+ETH_HEADER_LEN
,buf);
}

// δημιουργία δείκτη στην αρχή των δεδομένων TCP

// επιστρέφει 0 αν δεν υπάρχουν δεδομένα
// απαιτείται η κλήση της init_len_info πρώτα
uint16_t get_tcp_data_pointer(void)
{
	if (info_data_len){
		return((uint16_t)TCP_SRC_PORT_H_P+info_hdr_len);
	}else{
		return(0);
	}
}

// αρχικοποίηση μεταβλητών μήκους
void init_len_info(uint8_t *buf)
{
	info_data_len=(buf[IP_TOTLEN_H_P]<<8)|(buf[IP_TOTLEN_L_P]&0xff);
	info_data_len-=IP_HEADER_LEN;
	info_hdr_len=(buf[TCP_HEADER_LEN_P]>>4)*4;	// υπολογισμός μήκους
σε bytes
	info_data_len-=info_hdr_len;
	if (info_data_len<=0){
		info_data_len=0;
	}
}

// εγγραφή δεδομένων TCP σε ορισμένη θέση από progmem
// επιστρέφει την νέα ελεύθερη θέση
uint16_t fill_tcp_data_p(uint8_t *buf,uint16_t pos, const prog_char
*progmem_s)
{
	char c;
	// εγγραφή σε ορισμένη θέση

	// χωρίς options τα δεδομένα ξεκινούν μετά το checksum + 2 θέσεις
	while ((c = pgm_read_byte(progmem_s++))) {
		buf[TCP_CHECKSUM_L_P+3+pos]=c;
		pos++;
	}
	return(pos);
}
```

```
// εγγραφή δεδομένων TCP σε ορισμένη θέση από string
// επιστρέφει την νέα ελεύθερη θέση
uint16_t fill_tcp_data(uint8_t *buf,uint16_t pos, const char *s)
{
	// εγγραφή σε ορισμένη θέση

	// χωρίς options τα δεδομένα ξεκινούν μετά το checksum + 2 θέσεις
	while (*s) {
		buf[TCP_CHECKSUM_L_P+3+pos]=*s;
		pos++;
		s++;
	}
	return(pos);
}

// δημιουργία πακέτου ack χωρίς tcp δεδομένα
// μεταβάλλεται η κεφαλίδα eth/ip/tcp
void make_tcp_ack_from_any(uint8_t *buf)
{
	uint16_t j;
	make_eth(buf);
	// δημιουργία καφαλίδας
	buf[TCP_FLAGS_P]=TCP_FLAGS_ACK_V;
	if (info_data_len==0){
		// αποστολή ack ακόμα και σε άδειο πακέτο
		make_tcphead(buf,1,0,1);   // χωρίς options
	}else{
		make_tcphead(buf,info_data_len,0,1);    // χωρίς options
	}

	// ορισμός συνολικού μήκους στην κεφαλίδα IP
	// 20 bytes IP + 20 bytes tcp (χωρίς options)
	j=IP_HEADER_LEN+TCP_HEADER_LEN_PLAIN;
	buf[IP_TOTLEN_H_P]=j>>8;
	buf[IP_TOTLEN_L_P]=j& 0xff;
	make_ip(buf);
	// υπολογισμός του checksum
	j=checksum(&buf[IP_SRC_P], 8+TCP_HEADER_LEN_PLAIN,2);
	buf[TCP_CHECKSUM_H_P]=j>>8;
	buf[TCP_CHECKSUM_L_P]=j& 0xff;
	enc28j60PacketSend(IP_HEADER_LEN+TCP_HEADER_LEN_PLAIN+ETH_HEADER_LEN,b
uf);
}


// δημιουργία ACK από πακέτο TCP με δεδομένα

// απαιτείται η κλήση της init_len_info
// dlen = μέγεθος δεδομένων TCP (HTTP) που στέλνουμε
// μπορεί να χρησιμοποιηθεί απευθείας μετά την make_tcp_ack_from_any αφού
// δεν ματαβάλλεται η κεφαλίδα eth/ip/tcp εκτός από το μήκος και το checksum
void make_tcp_ack_with_data(uint8_t *buf,uint16_t dlen)
{
	uint16_t j;
	// δημιουργία κεφαλίδας
	// στέλνουμε μόνο ένα πακέτο (χωρίς fragmentation)
	// και ορίζουμε FIN
	buf[TCP_FLAGS_P]=TCP_FLAGS_ACK_V|TCP_FLAGS_PUSH_V|TCP_FLAGS_FIN_V;

	// ορισμός συνολικού μήκους στην κεφαλίδα IP
	// 20 bytes IP + 20 bytes TCP (χωρίς options) + μήκος δεδομένων
	j=IP_HEADER_LEN+TCP_HEADER_LEN_PLAIN+dlen;
	buf[IP_TOTLEN_H_P]=j>>8;
	buf[IP_TOTLEN_L_P]=j& 0xff;
	fill_ip_hdr_checksum(buf);
	// μηδενισμός checksum
	buf[TCP_CHECKSUM_H_P]=0;
```

```
        buf[TCP_CHECKSUM_L_P]=0;
        // υπολογισμός checksum
        j=checksum(&buf[IP_SRC_P], 8+TCP_HEADER_LEN_PLAIN+dlen,2);
        buf[TCP_CHECKSUM_H_P]=j>>8;
        buf[TCP_CHECKSUM_L_P]=j& 0xff;
        enc28j60PacketSend(IP_HEADER_LEN+TCP_HEADER_LEN_PLAIN+dlen+ETH_HEADER_
LEN,buf);
}
```

lcd.c

```
// Βιβλιοθήκη οθόνης LCd τύπου HD44780

#include <avr/io.h>
#include <avr/pgmspace.h>
#ifndef F_CPU
#define F_CPU 8000000UL
#endif
#include <util/delay.h>
#include "lcd_hw.h"
#include "lcd.h"

/* μακροεντολές συμβατότητας */
#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif

#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif


/* μακροεντολές και σταθερές */
#define lcd_e_high()    sbi(LCD_E_PORT, LCD_E_PIN)
#define lcd_e_low()     cbi(LCD_E_PORT, LCD_E_PIN)
/* RS=0 command mode */
#define lcd_cmd_mode()     cbi(LCD_RS_PORT, LCD_RS_PIN)
/* RS=1 data mode */
#define lcd_data_mode()    sbi(LCD_RS_PORT, LCD_RS_PIN)
/* ορίζει όλα τα data pins σαν έξοδο */
#define lcd_data_port_out() {                   \
        sbi(LCD_DATA_DDR_D7,LCD_DATA_PIN_D7);   \
        sbi(LCD_DATA_DDR_D6,LCD_DATA_PIN_D6);   \
        sbi(LCD_DATA_DDR_D5,LCD_DATA_PIN_D5);   \
        sbi(LCD_DATA_DDR_D4,LCD_DATA_PIN_D4);   \
}

#if LCD_LINES==1
#define LCD_FUNCTION_DEFAULT    LCD_FUNCTION_4BIT_1LINE
#else
#define LCD_FUNCTION_DEFAULT    LCD_FUNCTION_4BIT_2LINES
#endif


/* δηλώσεις συναρτήσεων */
static void lcd_e_toggle(void);
static void lcd_out_high(u08 d);
static void lcd_out_low(u08 d);


/* συναρτήσεις */

// καθυστέρηση ορισμένα <ms>
void lcd_delay_ms(unsigned int ms)
{
        // χρησιμοποιούμε ειδική μακροεντολή
        while(ms){
```

```
                _delay_ms(0.96);
                ms--;
        }
}

/* αποστολή δεύτερου half-byte */
static void lcd_out_low(u08 d)
{
        if (d&0x08)  sbi(LCD_DATA_PORT_D7,LCD_DATA_PIN_D7);
                else cbi(LCD_DATA_PORT_D7,LCD_DATA_PIN_D7);
        if (d&0x04)  sbi(LCD_DATA_PORT_D6,LCD_DATA_PIN_D6);
                else cbi(LCD_DATA_PORT_D6,LCD_DATA_PIN_D6);
        if (d&0x02)  sbi(LCD_DATA_PORT_D5,LCD_DATA_PIN_D5);
                else cbi(LCD_DATA_PORT_D5,LCD_DATA_PIN_D5);
        if (d&0x01)  sbi(LCD_DATA_PORT_D4,LCD_DATA_PIN_D4);
                else cbi(LCD_DATA_PORT_D4,LCD_DATA_PIN_D4);
}

/* αποστολή πρώτου half-byte */
static void lcd_out_high(u08 d)
{
        if (d&0x80)  sbi(LCD_DATA_PORT_D7,LCD_DATA_PIN_D7);
                else cbi(LCD_DATA_PORT_D7,LCD_DATA_PIN_D7);
        if (d&0x40)  sbi(LCD_DATA_PORT_D6,LCD_DATA_PIN_D6);
                else cbi(LCD_DATA_PORT_D6,LCD_DATA_PIN_D6);
        if (d&0x20)  sbi(LCD_DATA_PORT_D5,LCD_DATA_PIN_D5);
                else cbi(LCD_DATA_PORT_D5,LCD_DATA_PIN_D5);
        if (d&0x10)  sbi(LCD_DATA_PORT_D4,LCD_DATA_PIN_D4);
                else cbi(LCD_DATA_PORT_D4,LCD_DATA_PIN_D4);
}

/* αλλαγή στάθμης Enable */
static void lcd_e_toggle(void)

{
        lcd_e_high();
        _delay_us(4);
        lcd_e_low();
}

/* εγγραφή byte */
static void lcd_write(u08 data, u08 rs)
{
        /* ορισμός data pins σε έξοδο */
        lcd_data_port_out();
        _delay_us(4);

        /* αποστολή πρώτου half-byte */
        lcd_out_high(data);
        _delay_us(2);

        if (rs)
                lcd_data_mode(); //rs=1: δεδομένα
        else
                lcd_cmd_mode();     //rs=0: εντολές
        lcd_e_toggle();

        /* αποστολή δεύτερου half-byte */
        lcd_out_low(data);
        _delay_us(2);

        if (rs)
                lcd_data_mode(); //rs=1: δεδομένα
        else
                lcd_cmd_mode();     //rs=0: εντολές

        lcd_e_toggle();
}
```

```
// αναμονή μέχρι η LCD να είναι έτοιμη

// δεν χρησιμοποιείται η γραμμή RW και απλά περιμένουμε
// αρκετά ώστε να ετοιμαστεί η οθόνη
static unsigned char lcd_waitcmd(unsigned char cmdwait)
{
	_delay_us(9);
	/* η οθόνη χρειάζεται επιπλέον χρόνο */
	if (cmdwait){
		lcd_delay_ms(2);
	}
	return (0);
}



/* public συναρτήσεις */

/* αποστολή εντολής <cmd> στην οθόνη */
void lcd_command(u08 cmd)
{
	lcd_waitcmd(0);
	lcd_write(cmd, 0);
	lcd_waitcmd(1);
}

/* τοποθέτηση του cursor στη θέση (x,y) */
void lcd_gotoxy(u08 x, u08 y)
{
#if LCD_LINES==1
	lcd_command((1 << LCD_DDRAM) + LCD_START_LINE1 + x);
#endif
#if LCD_LINES==2
	if (y == 0)
		lcd_command((1 << LCD_DDRAM) + LCD_START_LINE1 + x);
	else
		lcd_command((1 << LCD_DDRAM) + LCD_START_LINE2 + x);
#endif
#if LCD_LINES==3
	if (y == 0)
		lcd_command((1 << LCD_DDRAM) + LCD_START_LINE1 + x);
	else if (y == 1)
		lcd_command((1 << LCD_DDRAM) + LCD_START_LINE2 + x);
	else if (y == 2)
		lcd_command((1 << LCD_DDRAM) + LCD_START_LINE3 + x);
#endif
#if LCD_LINES==4
	if (y == 0)
		lcd_command((1 << LCD_DDRAM) + LCD_START_LINE1 + x);
	else if (y == 1)
		lcd_command((1 << LCD_DDRAM) + LCD_START_LINE2 + x);
	else if (y == 2)
		lcd_command((1 << LCD_DDRAM) + LCD_START_LINE3 + x);
	else                /* y==3 */
		lcd_command((1 << LCD_DDRAM) + LCD_START_LINE4 + x);
#endif

}


/* εκτύπωση χαρακτήρα στη θέση του cursor */
void lcd_putc(char c)
{
	lcd_waitcmd(0);
	lcd_write((unsigned char)c, 1);
	lcd_waitcmd(0);
}
```

```
/* εκτύπωση string στην οθόνη */
void lcd_puts(const char *s)
{
	while (*s) {
		lcd_putc(*s);
		s++;
	}
}


/* εκτύπωση string από την progmem στην οθόνη */
void lcd_puts_p(const prog_char *progmem_s)
{
	register char c;

	while ((c = pgm_read_byte(progmem_s++))) {
		lcd_putc(c);
	}
}


/* αρχικοποίηση οθόνης και επιλογή cursor
dispAttr=
	LCD_DISP_OFF
	LCD_DISP_ON
	LCD_DISP_ON_CURSOR
	LCD_DISP_CURSOR_BLINK */
void lcd_init(u08 dispAttr)
{
	/* αρχικοποίηση καναλιών */
	lcd_data_port_out();		// ορισμός θυρών δεδομένων σε έξοδο
	sbi(LCD_RS_DDR, LCD_RS_PIN);	// ορισμός καναλιού RS σε έξοδο
	sbi(LCD_E_DDR, LCD_E_PIN); // ορισμόε καναλιού Enable σε έξοδο

	lcd_delay_ms(12);	// αναμονή 12ms

	/* αρχική εγγραφή 8 bit */
	lcd_out_high(LCD_FUNCTION_8BIT_1LINE);
	lcd_e_toggle();
	lcd_delay_ms(2);	// αναμονή 2ms

	lcd_out_high(LCD_FUNCTION_8BIT_1LINE);
	lcd_e_toggle();
	lcd_delay_ms(2);	// αναμονή 2ms

	lcd_out_high(LCD_FUNCTION_8BIT_1LINE);
	lcd_e_toggle();
	lcd_delay_ms(2);	// αναμονή 2ms

	/* ενεργοποίηση λειτουργίας 4 bit */
	lcd_out_high(LCD_FUNCTION_4BIT_1LINE);
	lcd_e_toggle();

	/* χρήση εντολής lcd_command() σε λειτουργία 4 bit */
	lcd_command(LCD_FUNCTION_DEFAULT);
	lcd_command(LCD_DISP_OFF);		// σβήσιμο οθόνης
	lcd_clrscr();				// καθαρισμός οθόνης
	lcd_command(LCD_MODE_DEFAULT);	// επιλογή entry mode
	lcd_command(dispAttr);			// επιλογή cursor
	lcd_waitcmd(1);
}
```

sensirion_protocol.c

```
// Πρωτόκολλο επικοινωνίας αισθητήρα sentirion SHT11

#include <avr/io.h>
#ifndef F_CPU
#define F_CPU 8000000UL
#endif
#include <util/delay.h>

// διευθύνσεις
#define STATUS_REG_W 0x06  //000 0011 0
#define STATUS_REG_R 0x07  //000 0011 1
#define MEASURE_TEMP 0x03  //000 0001 1
#define MEASURE_HUMI 0x05  //000 0010 1
#define RESET 0x1e                //000 1111 0

// κανάλια επικοινωνίας
#define SETSCK1 PORTC|=(1<<PC5)
#define SETSCK0 PORTC&=~(1<<PC5)
#define SCKOUTP DDRC|=(1<<DDC5)

#define SETDAT1 PORTC|=(1<<PC4)
#define SETDAT0 PORTC&=~(1<<PC4)
#define GETDATA (PINC&(1<<PINC4))

#define DMODEIN DDRC&=~(1<<DDC4)
#define PULLUP1 PORTC|=(1<<PINC4)
#define DMODEOU DDRC|=(1<<DDC4)

//pulswitch
#define S_PULSLONG _delay_us(3.0)
#define S_PULSSHORT _delay_us(1.0)


// Υπολογισμός CRC
// Για τον υπολογισμό του CRC περισσότερων bytes,
// αρκεί να υπολογίσουμε το CRC του πρώτου και
// για κάθε επόμενο byte να καλούμε τη συνάρτηση
// computeCRC8, δίνοντας όρισμα seed την έξοδο της
// ακριβώς προηγούμενης κλήσης
unsigned char computeCRC8(unsigned char inData, unsigned char seed)
{
	unsigned char bitsLeft;
	unsigned char tmp;

	for (bitsLeft = 8; bitsLeft > 0; bitsLeft--)
	{
		tmp = ((seed ^ inData) & 0x01);   // XOR στα LSB
		if (tmp == 0)
		{
			seed >>= 1;
		}
		else
		{
			seed ^= 0x18; // χαρακτηριστικό πολυώνυμο
			seed >>= 1;
			seed |= 0x80; // set MSB
		}
		inData >>= 1;
	}
	return seed;
}


// αντιστροφή των bits
// για συμβατότητα με την
// υλοποίηση του CRC στον αισθητήρα
```

```
// (bit7->bit0, bit6->bit1 ...)
unsigned char bitswapbyte(unsigned char byte)
{
	unsigned char i=8;
	unsigned char result=0;
	while(i){
		result=(result<<1);
		if (1 & byte) {
			result=result | 1;
		}
		i--;
		byte=(byte>>1);
	}
	return(result);
}

// εγγραφή byte και επιβεβαίωση
char s_write_byte(unsigned char value)
{
	unsigned char i=0x80;
	unsigned char error=0;
	DMODEOU;
	while(i){	// μάσκα επιλογής
		if (i & value) {
			SETDAT1;	// pull-up
		}else{
			SETDAT0;	// pull-down
		}
		SETSCK1;	// παλμός
		S_PULSLONG;
		SETSCK0;
		S_PULSSHORT;
		i=(i>>1);
	}
	DMODEIN;	// απελευθέρωση γραμμής DATA
	PULLUP1;
	SETSCK1;	// παλμός για επιβεβαίωση
	S_PULSLONG;
	if (GETDATA){// επιβεβαίωση
		error=1;
	}
	S_PULSSHORT;
	SETSCK0;
	return(error);	// error=1 αν δεν γίνει επιβεβαίωση
}

// ανάγνωση byte και επιβεβαίωση
unsigned char s_read_byte(unsigned char ack)
{
	unsigned char i=0x80;
	unsigned char val=0;
	DMODEIN;	// απελευθέρωση γραμμής DATA
	PULLUP1;
	while(i){	// μάσκα επιλογής
		SETSCK1;	// παλμός
		S_PULSSHORT;
		if (GETDATA){
			val=(val | i);	// ανάγνωση bit
		}
		SETSCK0;
		S_PULSSHORT;
		i=(i>>1);
	}
	DMODEOU;
	if (ack){
		SETDAT0;	// pull-down
	}else{
		SETDAT1;	// pull-up
```

```
        }
        SETSCK1;      // παλμός για επιβεβαίωση
        S_PULSLONG;
        SETSCK0;
        S_PULSSHORT;
        DMODEIN;      // απελευθέρωση γραμμής DATA
        PULLUP1;
        return (val);
}

// αποστολή σήματος έναρξης επικοινωνίας
//        _____         ____
// DATA:      |_________|
//           ___     ___
// SCK : ___|   |___|   |___
void s_transstart(void)
{
        // αρχική κατάσταση
        SCKOUTP;
        SETSCK0;
        DMODEOU;
        SETDAT1;

        S_PULSSHORT;
        SETSCK1;
        S_PULSSHORT;
        SETDAT0;
        S_PULSSHORT;
        SETSCK0;
        S_PULSLONG;
        SETSCK1;
        S_PULSSHORT;
        SETDAT1;
        S_PULSSHORT;
        SETSCK0;
        S_PULSSHORT;

        DMODEIN;      // απελευθέρωση γραμμής DATA
        PULLUP1;
}

// αποστολή μηνύματος reset επικοινωνίας
//        _____________________________________________________         ___
// DATA:                                                      |_______|
//          _    _    _    _    _    _    _    _    _    _    __   __
// SCK : __| |__| |__| |__| |__| |__| |__| |__| |__| |__| |___| |___| |__
void s_connectionreset(void)
{
        unsigned char i;
        // αρχική κατάσταση
        SCKOUTP;
        SETSCK0;
        DMODEOU;
        SETDAT1;
        for(i=0;i<9;i++){    //9 παλμοί
                SETSCK1;
                S_PULSLONG;
                SETSCK0;
                S_PULSLONG;
        }
        s_transstart();      // ακολουθία έναρξης
}

// αποστολή μηνύματος soft-reset
char s_softreset(void)
{
        s_connectionreset();       // ακολουθία reset
```

```
	// αποστολή εντολής RESET
	// επιστρέφει 1 αν ο αισθητήρας δεν απαντήσει
	return (s_write_byte(RESET));
}


// μετρήσεις θερμοκρασίας και υγρασίας με checksum
// mode: 1=υγρασία, 0=θερμοκρασία
// επιστρέφει 1=λάθος εγγραφής, 2=λάθος crc, 3=τέλος χρόνου
char s_measure(unsigned int *p_value, unsigned char mode)
{
	unsigned char i=0;
	unsigned char msb,lsb;
	unsigned char checksum;
	// υπολογισμός CRC από την αρχή ως το τέλος της επικοινωνίας
	unsigned char crc_state=0;
	*p_value=0;
	s_transstart();		// έναρξη μετάδοσης
	if(mode){
		mode=MEASURE_HUMI;
	}else{
		mode=MEASURE_TEMP;
	}
	if (s_write_byte(mode)){
		return(1);
	}
	crc_state=computeCRC8(bitswapbyte(mode),crc_state);
	// αναμενόμενες καθυστερήσεις: θερμοκρασία i=70, υγρασία i=20
	while(i<240){
		_delay_ms(3.0);
		if (GETDATA==0){
			i=0;
			break;
		}
		i++;
	}
	if(i){
		// τέλος χρόνου
		return(3);
	}
	msb=s_read_byte(1); // ανάγνωση πρώτου byte (MSB)
	crc_state=computeCRC8(bitswapbyte(msb),crc_state);
	lsb=s_read_byte(1); // ανάγνωση δεύτερου byte (LSB)
	*p_value=(msb<<8)|(lsb);
	crc_state=computeCRC8(bitswapbyte(lsb),crc_state);
	checksum =s_read_byte(0);  // ανάγνωση checksum
	if (crc_state != checksum ) {
		return(2);
	}
	return(0);
}


// υπολογισμός θερμοκρασίας
// επιστρέφει την θερμοκρασία σε δέκατα του βαθμού Κελσίου
int calc_sth11_temp(unsigned int t)
{
	// t = 10*(t *0.01 - 39.7);
	// t = t *0.1 - 397;
	t/=10;
	t-=397;
	return(t);
}


// υπολογισμός σχετικής υγρασίας
// γίνεται χρήση γραμμικής προσέγγισης
unsigned char rhcalc_int(unsigned int s)
```

```
{
	unsigned int rh;
	// το s είναι από 100 έως 3340
	// αν s < 1712 τοτε η σχετική υγρασία είναι (143*s - 8192)/4096
	// αν s >= 1712 τοτε η σχετική υγρασία είναι (111*s + 46288)/4096
	// rh = σχετική υγρασία * 10

	if (s<1712){
		// διαίρεση με 4
		rh=(36*s - 2048)/102;
	}else{
		// διαίρεση με 8
		rh=(14*s + 5790)/51;
	}
	// στρογγυλοποίηση
	rh+=5;
	rh/=10;
	if (rh>98){
		rh=100;
	}
	return((unsigned char)(rh));
}

// υπολογισμός υγρασίας ανάλογα με τη θερμοκρασία
unsigned char calc_sth11_humi(unsigned int h, int t)
{
	int rh;
	rh=rhcalc_int(h);
	// rh=(t/10-25)*(0.01+0.00008*(sensor_val))+rh;
	// sensor_val ~= rh*30
	rh=rh + (t/80 - 3);
	return((unsigned char)rh);
}

// προσσέγγιση της 100*log10(x)
// δεν απαιτείται χρήση της βιβλιοθήκης math
// σφάλμα < 5% για 2<x<100
int log10_approx(unsigned char x)
{
	int l,log;
	if (x==1){
		return(0);
	}
	if (x<8){
		return(11*x+11);
	}

	log=980-980/x;
	log/=10;
	if (x<36){
		l=19*x;
		l=l/10;
		log+=l-4;
	}else{
		l=67*x;
		l=l/100;
		if (x>51 && x<81){
			log+=l +42;
		}else{
			log+=l +39;
		}
	}
	if (log>200) log=200;
	return(log);
}


// υπολογισμός σημείου δρόσου
```

```
int calc_dewpoint(unsigned char rh,int t)
{
	int k,tmp;
	k = (100*log10_approx(rh)-20000)/43;
	tmp=t/10;
	tmp=881*tmp;
	tmp=tmp/(243+t/10);
	k+=tmp*2;
	tmp=1762-k;
	tmp=24310/tmp;
	tmp*=k;
	// στρογγυλοποίηση
	if (tmp<0){
		tmp-=51;
	}else{
		tmp+=51;
	}
	return (tmp/10);
}
```

timeout.c

```
// Συναρτήσεις χρονοκαθυστέρησης

#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <avr/io.h>
#include "avr_compat.h"
#ifndef F_CPU
#define F_CPU 8000000UL
#endif
#ifndef ALIBC_OLD
#include <util/delay.h>
#else
#include <avr/delay.h>
#endif

void delay_ms(unsigned int ms)
/* καθυστέρηση ορισμένα <ms> */
{
	//χρησιμοποιούμε ειδική μακροεντολή
	while(ms){
		_delay_ms(0.96);
		ms--;
	}
}
```

main.c

```
// Βασικό πρόγραμμα λειτουργίας

#include <avr/io.h>
#include <stdlib.h>
#include <string.h>
#include <avr/pgmspace.h>
#include <avr/eeprom.h>
#include <avr/interrupt.h>
#include "ip_arp_udp_tcp.h"
#include "enc28j60.h"
#include "timeout.h"
#include "avr_compat.h"
#include "analog.h"
#include "net.h"
#include "lcd.h"
#include "config.h"
#include "sensirion_protocol.h"
```

```
#include "integrator.h"
#include <avr/wdt.h>

// εμφάνιση ακατέργαστων τιμών
//#define SHOWADCRAW 1
//#define SHOW_RAW 1

// Μακροεντολές wifi
#define WIFI_ON     PORTD|= (1<<PD7)
#define WIFI_OFF    PORTD &= ~(1<<PD7)

// Όρια τάσης μπαταρίας
#define ALWAYS_ON_UPPER 122
#define ALWAYS_ON_LOWER 121
#define ALWAYS_OFF_UPPER 114
#define ALWAYS_OFF_LOWER 113

// Ενεργοποίηση διαχείρησης ενέργειας
#define ENABLE_POWER_MANAGER 1

// Διευθύνσεις MAC/IP συστήματος
static uint8_t mymac[6] = {0x51,0x52,0x53,0x54,0x55,0x56};
static uint8_t myip[4] = {192,168,1,50};
static char baseurl[]="http://192.168.1.50/";
static uint16_t mywwwport = 80; // θύρα HTTP

// Μέγεθος buffer
#define BUFFER_SIZE 600
static uint8_t buf[BUFFER_SIZE+1];

static uint8_t s_conreset_cnt=0; // μετρητής reset αισθητήρα sentirion

unsigned char battlevel=ALWAYS_ON_UPPER;       // αρχικοποίηση τάσης
μπαταρίας
char wifi_status;   // κατάσταση λειτουργίας wifi

// αρχικοποίηση στο μέγιστο μήκος string
char * ts=   "2008-03-10.05:24:48";
char * tsp=  "2008-04-10.05:24:48";     // παλία τιμή
char * temp_="-19.3";
char * humi_="100";
char * dew_  ="-19.2";
char * vsol_="12.5";
char * vbat_="10.3";
char * amp_  ="507";
char * pres_="1007.6";
char * sea_  ="1328.3";

// βοηθητικές μεταβλητές
unsigned char seconds=0;
unsigned char minutes=0;
char eeprom_counter=0;     // δείκτης στην eeprom
char eeprom_pointer=0;     // μετρητής εγγραφών
int  eeprom_data=0;        // θέση επόμενου byte στην eeprom
char eeprom_save_enabled=0;
char time_is_valid=0;
char tick=0;


//WATCHDOG
uint8_t mcusr_mirror __attribute__ ((section (".noinit")));

void get_mcusr(void) \
__attribute__((naked)) \
__attribute__((section(".init3")));
void get_mcusr(void)
{
       mcusr_mirror = MCUSR;
```

```
	MCUSR = 0;
	wdt_disable();
} /* WATCHDOG */


// εισαγωγή υποδιαστολής
void adddecimalpoint(char *str){
	char c;
	if (*str == '-'){
		str++;
	}
	if (*(str+1) == '\0'){
		// just one digit
		c=*str;
		*str='0';
		*(str+1)='.';
		*(str+2)=c;
		*(str+3)='\0';
		return;
	}
	while(*str){
		c=*str;
		str++;
		if (!*str){
			// τελευταίος χαρακτήρας
			*(str-1)='.';
			*str=c;
			str++;
			*str='\0';
		}
	}
}

// έναρξη εσωτερικού ρολογιού
void clock_start(void)
{

	TCNT1H=0;
	TCNT1L=0;

	// λειτουργία output compare και
	// μηδενισμός μετά από κάθε σύγκριση
	TCCR1A=(0<<COM1B1)|(0<<COM1B0)|(0<<WGM11)|(0<<WGM10);
	TCCR1B=(1<<CS12)|(1<<CS10)|(1<<WGM12)|(0<<WGM13);

	// τιμή μετά από 1 second
	OCR1AH=0x1e;
	OCR1AL=0x91;

	TIMSK1 = (1 << OCIE1A);

	sei(); // ενεργοποίηση διακοπών
	seconds=0;
	minutes=0;

}

// διακοπή εσωτερικού ρολογιού
void clock_stop(void)
{

	TCNT1H=0;
	TCNT1L=0;

	TCCR1A=(0<<COM1B1)|(0<<COM1B0)|(0<<WGM11)|(0<<WGM10);
	TCCR1B=(0<<CS12)|(0<<CS10)|(0<<WGM12)|(0<<WGM13);

	OCR1AH=0x00;
```

```
        OCR1AL=0x00;

        TIMSK1 = (0 << OCIE1A);

        cli(); // απενεργοποίηση διακοπών
        seconds=0;
        minutes=0;

}

// εγγραφή δεδομένων eeprom
void eeprom_save(void){

        eeprom_data = 36 * eeprom_counter;

        eeprom_write_block(temp_,  (void*)eeprom_data,        5);
        eeprom_write_block(humi_,  (void*)eeprom_data+5,      3);
        eeprom_write_block(dew_,   (void*)eeprom_data+8,      5);
        eeprom_write_block(vsol_,  (void*)eeprom_data+13,     4);
        eeprom_write_block(vbat_,  (void*)eeprom_data+17,     4);
        eeprom_write_block(amp_,   (void*)eeprom_data+21,     3);
        eeprom_write_block(pres_,  (void*)eeprom_data+24,     6);
        eeprom_write_block(sea_,   (void*)eeprom_data+30,     6);

        eeprom_counter++;
        eeprom_pointer++;

}

// ανάγνωση δεδομένων eeprom
void eeprom_load(void){

        eeprom_data = 36 * eeprom_counter;

        eeprom_read_block(temp_,   (void*)eeprom_data,        5);
        eeprom_read_block(humi_,   (void*)eeprom_data+5,      3);
        eeprom_read_block(dew_,    (void*)eeprom_data+8,      5);
        eeprom_read_block(vsol_,   (void*)eeprom_data+13,     4);
        eeprom_read_block(vbat_,   (void*)eeprom_data+17,     4);
        eeprom_read_block(amp_,    (void*)eeprom_data+21,     3);
        eeprom_read_block(pres_,   (void*)eeprom_data+24,     6);
        eeprom_read_block(sea_,    (void*)eeprom_data+30,     6);

        eeprom_counter++;
        eeprom_pointer--;

}

// αρχικοποίηση eeprom
void eeprom_reset(void){

        eeprom_counter=0;
        eeprom_pointer=0;
        eeprom_data=0;

}

// ρουτίνα εκτύπωσης πρώτου slide στην οθόνη
void print_lcd()
{

        char numstr[8];
        int i;
        lcd_clrscr();

        // αισθητήρας τάσης ηλιακού συλλέκτη
        lcd_puts_p(PSTR("S:"));
        i=convertanalog(2);        // μέσος όρος
```

```
        i+=convertanalog(2);
        i+=convertanalog(2);
        i+=convertanalog(2);
        i+=convertanalog(2);
        i/=5;
        i=voltsolar(i);
        itoa(i,numstr,10);
        adddecimalpoint(numstr);
        lcd_puts(numstr);
        lcd_puts_p(PSTR(" V "));
        // αποθήκευση
        strcpy(vsol_,numstr);

        // αισθητήρας τάσης μπαταρίας
        lcd_puts_p(PSTR("B:"));
        i=readbat();

        battlevel=i;  // αποθήκευση τάσης μπαταρίας

        itoa(i,numstr,10);
        adddecimalpoint(numstr);
        lcd_puts(numstr);
        lcd_puts_p(PSTR(" V "));
        // αποθήκευση
        strcpy(vbat_,numstr);

        lcd_gotoxy(0,1); //next line

        // αισθητήρας ρεύματος ηλιακού συλλέκτη
        lcd_puts_p(PSTR("S:"));
        i=convertanalog(1);        // μέσος όρος
        i+=convertanalog(1);
        i+=convertanalog(1);
        i+=convertanalog(1);
        i+=convertanalog(1);
        i/=5;
        i=ampsolar(i);
        itoa(i,numstr,10);

        lcd_puts(numstr);
        lcd_puts_p(PSTR(" mA "));
        // αποθήκευση
        strcpy(amp_,numstr);

}

// ρουτίνα εκτύπωσης δεύτερου slide στην οθόνη
void print_lcd1()
{
        char numstr[8];
        int i;
        unsigned char error;
        unsigned char rh;
        unsigned int humival_raw,tempval_raw;
        int dew,temp;

        lcd_clrscr();

        // αισθητήρας θερμοκρασίας και υγρασίας
        if (s_conreset_cnt==0){
                s_connectionreset();
                s_conreset_cnt++;
        }
        if (s_conreset_cnt>9){
                s_conreset_cnt=0;
        }
        error=s_measure( &tempval_raw,0); // μέτρηση θερμοκρασίας
        if (error==0){
```

```
			error=s_measure( &humival_raw,1); // μέτρηση υγρασίας
		}
		if(error!=0){ // περίπτωση λάθους
			s_conreset_cnt=0;
		}

		lcd_puts_p(PSTR("T:"));
		temp=calc_sth11_temp(tempval_raw);
		itoa(temp,numstr,10);
		adddecimalpoint(numstr);
		lcd_puts(numstr);
		lcd_puts_p(PSTR("'C "));
		// αποθήκευση
		strcpy(temp_,numstr);

		lcd_puts_p(PSTR("D:"));
		rh=calc_sth11_humi(humival_raw,temp);
		// αποθήκευση
		itoa(rh,numstr,10);
		strcpy(humi_,numstr);

		dew=calc_dewpoint(rh,temp);
		itoa(dew,numstr,10);
		adddecimalpoint(numstr);
		lcd_puts(numstr);
		lcd_puts_p(PSTR("'C"));
		// αποθήκευση
		strcpy(dew_,numstr);


		lcd_gotoxy(0,1);	// επόμενη γραμμή

		// αισθητήρας πίεσης
		lcd_puts_p(PSTR("P:"));
		i=convertanalog(0);		// μέσος όρος
		i+=convertanalog(0);
		i+=convertanalog(0);
		i+=convertanalog(0);
		i+=convertanalog(0);
		i/=5;
		i=pressure(i);
		// αποθήκευση
		itoa(i,numstr,10);
		adddecimalpoint(numstr);
		strcpy(pres_,numstr);

		itoa(pressureATsealevel(i),numstr,10);
		adddecimalpoint(numstr);
		lcd_puts(numstr);
		lcd_puts_p(PSTR(" hPa nrm"));	// στην επιφάνεια της θάλασσας
		// αποθήκευση
		strcpy(sea_,numstr);

}

// δημιουργία σελίδας XML με live δεδομένα
uint16_t print_xmlpage(uint8_t *buf)
{
		uint16_t plen;
		char numstr[8];
		int i;
		unsigned char error;
		unsigned char rh;
		unsigned int humival_raw,tempval_raw;
		int dew,temp;
		char str[15];

		// εγγραφή στον buffer από την progmem
```

```
	plen=fill_tcp_data_p(buf,0,PSTR("HTTP/1.0 200 OK\r\nContent-Type:
application/xml\r\nPragma: no-cache\r\n\r\n"));
	plen=fill_tcp_data_p(buf,plen,PSTR("<?xml version=\"1.0\"
encoding=\"iso-8859-7\" ?>\n"));
	plen=fill_tcp_data_p(buf,plen,PSTR("<root>\n<temp>"));

	if (s_conreset_cnt==0){
		s_connectionreset();
		s_conreset_cnt++;
	}
	if (s_conreset_cnt>9){
		s_conreset_cnt=0;
	}
	error=s_measure( &tempval_raw,0); // μέτρηση θερμοκρασίας
	if (error==0){
		error=s_measure( &humival_raw,1); // μέτρηση υγρασίας
	}
	//
	if(error!=0){ // περίπτωση λάθους
		if (error==1){
			plen=fill_tcp_data_p(buf,plen,PSTR("sensor communication
error"));
		}
		if (error==2){
			plen=fill_tcp_data_p(buf,plen,PSTR("sensor crc error"));
		}
		if (error==3){
			plen=fill_tcp_data_p(buf,plen,PSTR("sensor timeout
error"));
		}
		plen=fill_tcp_data_p(buf,plen,PSTR("</temp>\n</root>\n"));
		s_conreset_cnt=0;
		return(plen);
	}

	temp=calc_sth11_temp(tempval_raw);
	itoa(temp,str,10);
	adddecimalpoint(str);
	plen=fill_tcp_data(buf,plen,str);
	plen=fill_tcp_data_p(buf,plen,PSTR("</temp>\n<humi>"));

	rh=calc_sth11_humi(humival_raw,temp);
	itoa(rh,str,10);
	plen=fill_tcp_data(buf,plen,str);
	plen=fill_tcp_data_p(buf,plen,PSTR("</humi>\n<dew>"));
	dew=calc_dewpoint(rh,temp);
	itoa(dew,str,10);
	adddecimalpoint(str);
	plen=fill_tcp_data(buf,plen,str);
	plen=fill_tcp_data_p(buf,plen,PSTR("</dew>\n<vsol>"));

	// αισθητήρας τάσης ηλιακού συλλέκτη
	i=convertanalog(2);		// μέσος όρος
	i+=convertanalog(2);
	i+=convertanalog(2);
	i+=convertanalog(2);
	i+=convertanalog(2);
	i/=5;

	i=voltsolar(i);
	itoa(i,numstr,10);
	adddecimalpoint(numstr);
	plen=fill_tcp_data(buf,plen,numstr);
	plen=fill_tcp_data_p(buf,plen,PSTR("</vsol>\n<vbat>"));

	// αισθητήρας τάσης μπαταρίας
	//i=readbat();
	i=battlevel;
```

```
        itoa(i,numstr,10);
        adddecimalpoint(numstr);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR("</vbat>\n<amp>"));

        // αισθητήρας ρεύματος ηλιακού συλλέκτη
        i=convertanalog(1);        // μέσος όρος
        i+=convertanalog(1);
        i+=convertanalog(1);
        i+=convertanalog(1);
        i+=convertanalog(1);
        i/=5;

        i=ampsolar(i);
        itoa(i,numstr,10);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR("</amp>\n<pres>"));


        // αισθητήρας πίεσης
        i=convertanalog(0);        // μέσος όρος
        i+=convertanalog(0);
        i+=convertanalog(0);
        i+=convertanalog(0);
        i+=convertanalog(0);
        i/=5;

        i=pressure(i);
        itoa(i,numstr,10);
        adddecimalpoint(numstr);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR("</pres>\n<sea>"));
        itoa(pressureATsealevel(i),numstr,10);
        adddecimalpoint(numstr);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR("</sea>\n<more>no</more>\n"));


        plen=fill_tcp_data_p(buf,plen,PSTR("</root>"));
        return(plen);
}

// δημιουργία σελίδας XML με δεδομένα eeprom
uint16_t print_xmlpage_eeprom(uint8_t *buf)
{
        uint16_t plen;
        char numstr[8];

        // ανάγνωση δεδομένων
        eeprom_load();

        // εγγραφή στον buffer
        plen=fill_tcp_data_p(buf,0,PSTR("HTTP/1.0 200 OK\r\nContent-Type:
application/xml\r\nPragma: no-cache\r\n\r\n"));
        plen=fill_tcp_data_p(buf,plen,PSTR("<?xml version=\"1.0\"
encoding=\"iso-8859-7\" ?>\n"));
        plen=fill_tcp_data_p(buf,plen,PSTR("<root>\n<temp>"));

        plen=fill_tcp_data(buf,plen,temp_);
        plen=fill_tcp_data_p(buf,plen,PSTR("</temp>\n<humi>"));

        plen=fill_tcp_data(buf,plen,humi_);
        plen=fill_tcp_data_p(buf,plen,PSTR("</humi>\n<dew>"));

        plen=fill_tcp_data(buf,plen,dew_);
        plen=fill_tcp_data_p(buf,plen,PSTR("</dew>\n<vsol>"));
```

```
        plen=fill_tcp_data(buf,plen,vsol_);
        plen=fill_tcp_data_p(buf,plen,PSTR("</vsol>\n<vbat>"));

        plen=fill_tcp_data(buf,plen,vbat_);
        plen=fill_tcp_data_p(buf,plen,PSTR("</vbat>\n<amp>"));

        plen=fill_tcp_data(buf,plen,amp_);
        plen=fill_tcp_data_p(buf,plen,PSTR("</amp>\n<pres>"));

        plen=fill_tcp_data(buf,plen,pres_);
        plen=fill_tcp_data_p(buf,plen,PSTR("</pres>\n<sea>"));

        plen=fill_tcp_data(buf,plen,sea_);
        plen=fill_tcp_data_p(buf,plen,PSTR("</sea>\n<more>yes</more>\n<time>")
);

        //plen=fill_tcp_data(buf,plen,ts);
        plen=fill_tcp_data(buf,plen,tsp);
        plen=fill_tcp_data_p(buf,plen,PSTR("</time>\n<remain>"));

        itoa(eeprom_pointer,numstr,10);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR("</remain>\n"));


        plen=fill_tcp_data_p(buf,plen,PSTR("</root>"));
        return(plen);
}


// δημιουργία ιστοσελίδας στον buffer
uint16_t print_webpage(uint8_t *buf)
{
        uint16_t plen;
        char numstr[8];
        int i;
        unsigned char error;
        unsigned char rh;
        unsigned int humival_raw,tempval_raw;
        int dew,temp;
        char str[15];

        plen=fill_tcp_data_p(buf,0,PSTR("HTTP/1.0 200 OK\r\nContent-Type:
text/html\r\nPragma: no-cache\r\n\r\n"));
        plen=fill_tcp_data_p(buf,plen,PSTR("<h2>AVR Weather Station</h2>\n"));
        plen=fill_tcp_data_p(buf,plen,PSTR("<h2>Sensor @"));
        itoa(myip[3],str,10);
        plen=fill_tcp_data(buf,plen,str);
        plen=fill_tcp_data_p(buf,plen,PSTR("</h2>\n"));
        plen=fill_tcp_data_p(buf,plen,PSTR("<pre>\n"));
        if (s_conreset_cnt==0){
                s_connectionreset();
                s_conreset_cnt++;
        }
        if (s_conreset_cnt>9){
                s_conreset_cnt=0;
        }
        error=s_measure( &tempval_raw,0); // μέτρηση θερμοκρασίας
        if (error==0){
                error=s_measure( &humival_raw,1); // μέτρηση υγρασίας
        }
        //
        if(error!=0){ // περίπτωση λάθους
                if (error==1){
                        plen=fill_tcp_data_p(buf,plen,PSTR("sensor communication
error\n"));
                }
                if (error==2){
```

```
                    plen=fill_tcp_data_p(buf,plen,PSTR("sensor crc
error\n"));
             }
             if (error==3){
                    plen=fill_tcp_data_p(buf,plen,PSTR("sensor timeout
error\n"));
             }
             itoa(humival_raw,str,10);
             plen=fill_tcp_data(buf,plen,str);
             plen=fill_tcp_data_p(buf,plen,PSTR("\n\n"));
             plen=fill_tcp_data_p(buf,plen,PSTR("</pre>\n"));
             s_conreset_cnt=0;
             return(plen);
      }
#ifdef SHOW_RAW            // ακατέργαστα δεδομένα
      plen=fill_tcp_data_p(buf,plen,PSTR("DEBUG: raw temp humi:"));
      itoa(tempval_raw,str,10);
      plen=fill_tcp_data(buf,plen,str);
      plen=fill_tcp_data_p(buf,plen,PSTR(" "));
      itoa(humival_raw,str,10);
      plen=fill_tcp_data(buf,plen,str);
      plen=fill_tcp_data_p(buf,plen,PSTR("\n\n"));
#endif

      plen=fill_tcp_data_p(buf,plen,PSTR("temperature  : "));
      temp=calc_sth11_temp(tempval_raw);
      itoa(temp,str,10);
      adddecimalpoint(str);
      plen=fill_tcp_data(buf,plen,str);
      plen=fill_tcp_data_p(buf,plen,PSTR("'C ["));
      itoa(c2f(temp),str,10);
      adddecimalpoint(str);
      plen=fill_tcp_data(buf,plen,str);
      plen=fill_tcp_data_p(buf,plen,PSTR("'F]"));
#ifdef SHOW_RAW            // ακατέργαστα δεδομένα
      plen=fill_tcp_data_p(buf,plen,PSTR("\nhumi lin     : "));
      rh=rhcalc_int(humival_raw);
      itoa(rh,str,10);
      plen=fill_tcp_data(buf,plen,str);
#endif
      plen=fill_tcp_data_p(buf,plen,PSTR("\nrel. humidity: "));
      rh=calc_sth11_humi(humival_raw,temp);
      itoa(rh,str,10);
      plen=fill_tcp_data(buf,plen,str);
      plen=fill_tcp_data_p(buf,plen,PSTR("%\ndew point    : "));
      dew=calc_dewpoint(rh,temp);
      itoa(dew,str,10);
      adddecimalpoint(str);
      plen=fill_tcp_data(buf,plen,str);
      plen=fill_tcp_data_p(buf,plen,PSTR("'C ["));
      // στρογγυλοποίηση
      itoa(c2f(dew)/10,str,10);
      plen=fill_tcp_data(buf,plen,str);
      plen=fill_tcp_data_p(buf,plen,PSTR("'F]\n"));

      // αισθητήρας τάσης ηλιακού συλλέκτη
      plen=fill_tcp_data_p(buf,plen,PSTR("\nSolar Panel Voltage: "));
      i=convertanalog(2);        // μέσος όρος
      i+=convertanalog(2);
      i+=convertanalog(2);
      i+=convertanalog(2);
      i+=convertanalog(2);
      i/=5;
#ifdef SHOWADCRAW   // ακατέργαστα δεδομένα
      itoa(i,numstr,10); // μετατροπή ακέραιου σε string
      plen=fill_tcp_data(buf,plen,numstr);
      plen=fill_tcp_data_p(buf,plen,PSTR(" "));
#endif
```

```
        i=voltsolar(i);
        itoa(i,numstr,10);
        adddecimalpoint(numstr);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR("Volt"));

        // αισθητήρας τάσης μπαταρίας
        plen=fill_tcp_data_p(buf,plen,PSTR("\nBattery Voltage: "));

#ifdef SHOWADCRAW   // ακατέργαστα δεδομένα
        i=convertanalog(3);        // μέσος όρος
        i+=convertanalog(3);
        i+=convertanalog(3);
        i+=convertanalog(3);
        i+=convertanalog(3);
        i/=5;
        itoa(i,numstr,10); // μετατροπή ακέραιου σε string
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR(" "));
#endif
        //i=readbat();
        i=battlevel;
        itoa(i,numstr,10);
        adddecimalpoint(numstr);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR("Volt"));

        // αισθητήρας ρεύματος ηλιακού συλλέκτη
        plen=fill_tcp_data_p(buf,plen,PSTR("\nSolar Current: "));
        i=convertanalog(1);        // μέσος όρος
        i+=convertanalog(1);
        i+=convertanalog(1);
        i+=convertanalog(1);
        i+=convertanalog(1);
        i/=5;
#ifdef SHOWADCRAW   // ακατέργαστα δεδομένα
        itoa(i,numstr,10); // μετατροπή ακέραιου σε string
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR(" "));
#endif
        i=ampsolar(i);
        itoa(i,numstr,10);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR("mAmp"));


        // αισθητήρας πίεσης
        plen=fill_tcp_data_p(buf,plen,PSTR("\nAir pressure : "));
        i=convertanalog(0);        // μέσος όρος
        i+=convertanalog(0);
        i+=convertanalog(0);
        i+=convertanalog(0);
        i+=convertanalog(0);
        i/=5;
#ifdef SHOWADCRAW
        itoa(i,numstr,10); // μετατροπή ακέραιου σε string
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR(" "));
#endif
        i=pressure(i);
        itoa(i,numstr,10);
        adddecimalpoint(numstr);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR(" hPa real\n                "));
        itoa(pressureATsealevel(i),numstr,10);
        adddecimalpoint(numstr);
        plen=fill_tcp_data(buf,plen,numstr);
        plen=fill_tcp_data_p(buf,plen,PSTR(" hPa at sea level"));
```

```
        plen=fill_tcp_data_p(buf,plen,PSTR("</pre>\n<br>"));
        plen=fill_tcp_data_p(buf,plen,PSTR("<small><a href=\""));
        plen=fill_tcp_data(buf,plen,baseurl);
        plen=fill_tcp_data_p(buf,plen,PSTR("\">[refresh]</a></small><br>\n<hr>
<br>"));
        plen=fill_tcp_data_p(buf,plen,PSTR(VERSIONSTR));
        return(plen);
}


// Κεντρική ρουτίνα
int main(void){

        uint16_t plen;
        uint16_t dat_p;
        uint16_t l=0;
        uint8_t toggle=0;
        uint8_t buttonpress=0;

        //WATCHDOG
        wdt_enable(WDTO_1S);              // 1 s
        //wdt_enable(WDTO_120MS);  // 120 ms


        // ενεργοποίση ρολογιού 8MHz
        CLKPR=(1<<CLKPCE);
        CLKPR=0;      // χωρίς pre-scaler
        delay_ms(1);

        // ενεργοποίση PD2/INT0 σαν είσοδο
        DDRD&= ~(1<<DDD2);

        // ορισμός transistor wifi
        DDRD|= (1<<DDD7);
        WIFI_ON;      // ενεργοποίηση transistor
        wifi_status=1;
        //delay_ms(50);     // καθυστέρηση εκκίνησης wifi
        //WIFI_OFF;  // απενεργοποίηση transistor
        //wifi_status=0;

        // ορισμός transistor φωτισμού οθόνης
        DDRB|= (1<<DDB7);
        PORTB &= ~(1<<PB7); // απενεργοποίηση transistor

        // αρχικοποίηση enc28j60
        enc28j60Init(mymac);
        enc28j60clkout(2);  // αλλαγή ρολογιού από 6.25MHz στα 12.5MHz
        delay_ms(10);

        // Push button
        // ενεργοποίηση PB6 σαν είσοδο
        DDRB&= ~(1<<DDB6);
        // ενεργοποίηση αντίστασης pull-up
        PORTB|= (1<<PINB6);

        // αρχικοποίηση των LEDS του Magjack
        enc28j60PhyWrite(PHLCON,0x476);
        delay_ms(20);


        // αρχικοποίηση οθόνης
        lcd_init(LCD_DISP_ON);
        lcd_clrscr();
        lcd_puts_P("=ok=");
        delay_ms(20);
```

```
	// αρχικοποίηση επιπεδου ethernet/ip
	init_ip_arp_udp_tcp(mymac,myip,mywwwport);

	// αρχικοποίηση ολοκληρωτή τάσης μπαταρίας
	initbat();

	// έναρξη ρολογιού
	clock_start();

	while(1){
		// λήψη του επόμενου πακέτου
		plen = enc28j60PacketReceive(BUFFER_SIZE, buf);

		// το plen είναι μη μηδενικό αν λάβαμε πακέτο χωρίς λάθος CRC
		if(plen==0){
			l++;
			if (bit_is_clear(PINB,PINB6)){
				// πατήθηκε το button
				//l=0; // το 'l' λειτουργεί και σαν
χρονοκαθυστέρηση
				// που εξασφαλίζει την απελευθέρωση του button
				PORTB|= (1<<PB7);   // ενεργοποίηση transistor
				if (buttonpress==0)
					lcd_init(LCD_DISP_ON);     // επανεκκίνηση
οθόνης αν είναι ασταθής
				buttonpress=1;
			}
			else {
				PORTB &= ~(1<<PB7); //απενεργοποίηση transistor
				buttonpress=0;
			}

//=========================
#ifdef ENABLE_POWER_MANAGER
//=========================

			// διαχείρησης ενέργειας
			if ((battlevel>ALWAYS_ON_UPPER) &&
(eeprom_save_enabled==0))
				if (wifi_status==0) {
					WIFI_ON;
					wifi_status=1;
				}

			if ((battlevel<ALWAYS_ON_LOWER) &&
(battlevel>ALWAYS_OFF_UPPER) && (eeprom_save_enabled==0)) {
				// ενεργοποίηση εγγραφής eeprom
				// απενεργοποίηση διαχείρησης ενέργειας
				eeprom_save_enabled=1;
				eeprom_reset();

				// έλεγχος ώρας
				if (seconds<120)
					time_is_valid=1;

				// έναρξη ρολογιού
				//clock_start();
				minutes=0;
				seconds=0;

				// αλλαγή κατάστασης wifi
				if (wifi_status==1) {
					WIFI_OFF;
					wifi_status=0;
				}
			}

			if (eeprom_save_enabled==1) {
```

```
                        if ((seconds==60) && (tick==1)) {

                                // ανανέωση ρολογιού
                                tick=0;             // το βήμα αυτό δεν
πρέπει να επαναληφθεί
                                seconds=0;
                                minutes++;

                                // αποθήκευση eeprom μόνο αν η ώρα είναι
valid,
                                // αλλιώς γίνεται εξοικονόμηση ενέργειας
                                if ((time_is_valid==1) && (wifi_status==0))
                                        eeprom_save();

                                //termination condition
                                if (minutes==14) {
                                        strcpy(tsp,ts);            //
ανανέωση ώρας
                                        WIFI_ON;
                                        wifi_status=1;
                                        eeprom_counter=0;   // έτοιμο για
ανάγνωση δεδομένων
                                        time_is_valid=0;    // με ενεργό
wifi η ώρα μπορεί να ανανεωθεί
                                }

                                // περιθώριο 4 λεπτά για ενεργοποίηση wifi
                                // αν ο logger αποτύχει, η τροφοδοσία θα
κοπεί
                                if (minutes==18) {
                                        eeprom_save_enabled=0;
                                        //clock_stop();
                                }

                        }

                }

                if ((battlevel<ALWAYS_OFF_LOWER) &&
(eeprom_save_enabled==0))
                        if (wifi_status==1) {
                                WIFI_OFF;
                                wifi_status=0;
                        }

//============================
#endif
//============================


                //WATCHDOG
                wdt_reset();

                if (l==100){
                        // ανανέωση οθόνης
                        //if (buttonpress){
                        //      buttonpress=0;
                        //      celsius=celsius ^ 1;       // xor
                        //}
                        if (toggle==0){
                                lcd_clrscr();
                                print_lcd();
                                //lcd_gotoxy(19,1);
                                //lcd_putc('+');
                                char ttext[2];
                                lcd_gotoxy(18,1);
                                itoa(minutes,ttext,10);
```

```
                                        lcd_puts(ttext);
                                }
                                else {
                                        //print_lcd();
                                        print_lcd1();
                                }
                                toggle=toggle ^ 1; // xor
                        }
                        continue;
                }

                // δημοσιοποίηση διεύθυνσης MAC
                if(eth_type_is_arp_and_my_ip(buf,plen)){
                        make_arp_answer_from_request(buf);
                        continue;
                }

                // έλεγχος προορισμού πακέτων IP
                if(eth_type_is_ip_and_my_ip(buf,plen)==0){
                        continue;
                }

                // απάντηση σε PING
                if(buf[IP_PROTO_P]==IP_PROTO_ICMP_V &&
buf[ICMP_TYPE_P]==ICMP_TYPE_ECHOREQUEST_V){
                        make_echo_reply_from_request(buf,plen);
                        continue;
                }

                // απάντηση TCP
                if
(buf[IP_PROTO_P]==IP_PROTO_TCP_V&&buf[TCP_DST_PORT_H_P]==0&&buf[TCP_DST_PORT
_L_P]==mywwwport){
                        if (buf[TCP_FLAGS_P] & TCP_FLAGS_SYN_V){
                                make_tcp_synack_from_syn(buf);
                                // η συνάρτηση make_tcp_synack_from_syn στέλνει
synack
                                continue;
                        }
                        if (buf[TCP_FLAGS_P] & TCP_FLAGS_ACK_V){
                                init_len_info(buf); // αρχικοποίηση
                                // αποστολή ACK χωρίς δεδομένα
                                dat_p=get_tcp_data_pointer();
                                if (dat_p==0){
                                        if (buf[TCP_FLAGS_P] & TCP_FLAGS_FIN_V){
                                                // finack, answer with ack
                                                make_tcp_ack_from_any(buf);
                                        }
                                        // αποστολή ACK χωρίς δεδομένα και αναμονή
νέου πακέτου
                                        continue;
                                }
                                if (strncmp("GET ",(char *)&(buf[dat_p]),4)!=0){

                                        plen=fill_tcp_data_p(buf,0,PSTR("HTTP/1.0
200 OK\r\nContent-Type: text/html\r\n\r\n<h1>200 OK</h1>"));
                                        goto SENDTCP;
                                }

                                if (strncmp("/xml/",(char *)&(buf[dat_p+4]),5)==0)
{

                                        // Ανάγνωση ώρας
                                        strncpy(ts,(char *)&(buf[dat_p+9]),19);
                                        strncpy(&ts[10]," ",1);
                                        //time_is_valid=1;
                                        // μηδενισμός ρολογιού
                                        //minutes=0;
```

```
                                        if (eeprom_save_enabled==0)
                                                seconds=0;    // δεν μηδενίζουμε τα
seconds αν ο logger διαβάζει δεδομένα eeprom

                                        if (eeprom_pointer==0)
                                                plen=print_xmlpage(buf);
                                        else
                                                plen=print_xmlpage_eeprom(buf);
                                        // ΠΡΟΣΟΧΗ eeprom_counter=0;
                                }

                                // βασικό URL
                                else if (strncmp("/ ",(char
*)&(buf[dat_p+4]),2)==0){
                                        // τυχαίο URL
                                        //if (buf[dat_p+4] == '/'){
                                        plen=print_webpage(buf);
                                }
                                else
                                {
                                        // τυχαία δεδομένα:
                                        //plen=print_xmlpage(buf);
                                        plen=fill_tcp_data_p(buf,0,PSTR("HTTP/1.0
400 Bad Request\r\nContent-Type: text/html\r\n\r\n<h1>400 Bad
Request</h1>"));
                                }
SENDTCP:
                                make_tcp_ack_from_any(buf); // αποστολή ACK στην
αίτηση GET (HTTP)
                                make_tcp_ack_with_data(buf,plen); // αποστολή
δεδομένων
                                continue;
                        }

                }

                }
                return (0);
}


// διακοπή σε κάθε χτύπο του ρολογιού
ISR(TIMER1_COMPA_vect){
        if (seconds<250)
                seconds++;
        tick=1;
}
```

analog.h

```
/* Αναλογική μετατροπή */
#ifndef ANALOG_H
#define ANALOG_H

// Συναρτήσεις
extern int convertanalog(unsigned char channel);

extern int tempcelsius0(int adcval);
extern int tempcelsius1(int adcval);

extern int c2f(int celsiustimes10);

extern int voltsolar(int adcval);
extern int voltbattery(int adcval);
extern int ampsolar(int adcval);

extern int pressure(int adcval);
```

```
extern int pressureATsealevel(int phpa10);

#endif
```

avr_compat.h

```
/* Συναρτήσεις συμβατότητας compilers*/
#ifndef AVR_COMPAT_H
#define AVR_COMPAT_H

// Χρήση define για βιβλιοθήκη avr-1.2.x ή παλαιότερη
//#define ALIBC_OLD 1

#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif

#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif

#ifndef inb
#define inb(sfr) _SFR_BYTE(sfr)
#endif

#ifndef outb
#define outb(sfr, val) (_SFR_BYTE(sfr) = (val))
#endif

#ifndef inw
#define inw(sfr) _SFR_WORD(sfr)
#endif

#ifndef outw
#define outw(sfr, val) ( _SFR_WORD(sfr) = (val))
#endif

#ifndef outp
#define outp(val, sfr) outb(sfr, val)
#endif

#ifndef inp
#define inp(sfr) inb(sfr)
#endif

#ifndef BV
#define BV(bit) _BV(bit)
#endif

/* Συμβατότητα με libc 1.0 έως 1.2 */
#ifndef PRG_RDB
#define PRG_RDB(addr)      pgm_read_byte(addr)
#endif

#endif
```

config.h

```
// Αρχείο ρυθμίσεων

#ifndef CONFIG_H
#define CONFIG_H

//Εκδοση server
#define VERSIONSTR "Tsoutsos Nektarios v1.0\n"
```

```
// Συντελεστής μέτρησης Ampere
#define AMPCALIB 1

// Τιμές ADC ανά KELVIN
#define KELVINPERADC0 127
#define KELVINPERADC1 127

// Offset σε δέκατα του βαθμού Κελσίου
#define OFFSET0 305
#define OFFSET1 305

// Κέρδος αισθητήρα πίεσης
#define GAIN_A 88

// Περιθώριο σφάλματος αισθητήρα πιεσης
#define ERROR_A 0

// Offset θερμοκρασίας αναφοράς, ανάλογα με την τάση
// και το κέρδος του αισθητήρα
#define REFTERM_A 9433

// Πίεση αναφοράς στην επιφάνεια της θάλασσας σε δέκατα του hPa
#define SEALEVELOFFSET 240

#endif
```

enc28j60.h

```
// Driver για το chip ENC28J60 της Microchip
// Προσφέρει λειτουργίες αρχικοποίησης, αποστολής και λήψης

#ifndef ENC28J60_H
#define ENC28J60_H
#include <inttypes.h>

// Control Registers
// Αποτελούνται από ένα συνδυασμό register address,
// bank number και Ethernet/MAC/PHY indicator bits.
// Register address   (bits 0-4)
// Bank number        (bits 5-6)
// MAC/PHY indicator  (bit 7)
#define ADDR_MASK        0x1F
#define BANK_MASK        0x60
#define SPRD_MASK        0x80
// All-bank registers
#define EIE              0x1B
#define EIR              0x1C
#define ESTAT            0x1D
#define ECON2            0x1E
#define ECON1            0x1F
// Bank 0 registers
#define ERDPTL           (0x00|0x00)
#define ERDPTH           (0x01|0x00)
#define EWRPTL           (0x02|0x00)
#define EWRPTH           (0x03|0x00)
#define ETXSTL           (0x04|0x00)
#define ETXSTH           (0x05|0x00)
#define ETXNDL           (0x06|0x00)
#define ETXNDH           (0x07|0x00)
#define ERXSTL           (0x08|0x00)
#define ERXSTH           (0x09|0x00)
#define ERXNDL           (0x0A|0x00)
#define ERXNDH           (0x0B|0x00)
#define ERXRDPTL         (0x0C|0x00)
#define ERXRDPTH         (0x0D|0x00)
#define ERXWRPTL         (0x0E|0x00)
```

```
#define ERXWRPTH          (0x0F|0x00)
#define EDMASTL           (0x10|0x00)
#define EDMASTH           (0x11|0x00)
#define EDMANDL           (0x12|0x00)
#define EDMANDH           (0x13|0x00)
#define EDMADSTL          (0x14|0x00)
#define EDMADSTH          (0x15|0x00)
#define EDMACSL           (0x16|0x00)
#define EDMACSH           (0x17|0x00)
// Bank 1 registers
#define EHT0              (0x00|0x20)
#define EHT1              (0x01|0x20)
#define EHT2              (0x02|0x20)
#define EHT3              (0x03|0x20)
#define EHT4              (0x04|0x20)
#define EHT5              (0x05|0x20)
#define EHT6              (0x06|0x20)
#define EHT7              (0x07|0x20)
#define EPMM0             (0x08|0x20)
#define EPMM1             (0x09|0x20)
#define EPMM2             (0x0A|0x20)
#define EPMM3             (0x0B|0x20)
#define EPMM4             (0x0C|0x20)
#define EPMM5             (0x0D|0x20)
#define EPMM6             (0x0E|0x20)
#define EPMM7             (0x0F|0x20)
#define EPMCSL            (0x10|0x20)
#define EPMCSH            (0x11|0x20)
#define EPMOL             (0x14|0x20)
#define EPMOH             (0x15|0x20)
#define EWOLIE            (0x16|0x20)
#define EWOLIR            (0x17|0x20)
#define ERXFCON           (0x18|0x20)
#define EPKTCNT           (0x19|0x20)
// Bank 2 registers
#define MACON1            (0x00|0x40|0x80)
#define MACON2            (0x01|0x40|0x80)
#define MACON3            (0x02|0x40|0x80)
#define MACON4            (0x03|0x40|0x80)
#define MABBIPG           (0x04|0x40|0x80)
#define MAIPGL            (0x06|0x40|0x80)
#define MAIPGH            (0x07|0x40|0x80)
#define MACLCON1          (0x08|0x40|0x80)
#define MACLCON2          (0x09|0x40|0x80)
#define MAMXFLL           (0x0A|0x40|0x80)
#define MAMXFLH           (0x0B|0x40|0x80)
#define MAPHSUP           (0x0D|0x40|0x80)
#define MICON             (0x11|0x40|0x80)
#define MICMD             (0x12|0x40|0x80)
#define MIREGADR          (0x14|0x40|0x80)
#define MIWRL             (0x16|0x40|0x80)
#define MIWRH             (0x17|0x40|0x80)
#define MIRDL             (0x18|0x40|0x80)
#define MIRDH             (0x19|0x40|0x80)
// Bank 3 registers
#define MAADR1            (0x00|0x60|0x80)
#define MAADR0            (0x01|0x60|0x80)
#define MAADR3            (0x02|0x60|0x80)
#define MAADR2            (0x03|0x60|0x80)
#define MAADR5            (0x04|0x60|0x80)
#define MAADR4            (0x05|0x60|0x80)
#define EBSTSD            (0x06|0x60)
#define EBSTCON           (0x07|0x60)
#define EBSTCSL           (0x08|0x60)
#define EBSTCSH           (0x09|0x60)
#define MISTAT            (0x0A|0x60|0x80)
#define EREVID            (0x12|0x60)
#define ECOCON            (0x15|0x60)
```

```
#define EFLOCON          (0x17|0x60)
#define EPAUSL           (0x18|0x60)
#define EPAUSH           (0x19|0x60)
// PHY registers
#define PHCON1           0x00
#define PHSTAT1          0x01
#define PHHID1           0x02
#define PHHID2           0x03
#define PHCON2           0x10
#define PHSTAT2          0x11
#define PHIE             0x12
#define PHIR             0x13
#define PHLCON           0x14

// ERXFCON Register Bits
#define ERXFCON_UCEN     0x80
#define ERXFCON_ANDOR    0x40
#define ERXFCON_CRCEN    0x20
#define ERXFCON_PMEN     0x10
#define ERXFCON_MPEN     0x08
#define ERXFCON_HTEN     0x04
#define ERXFCON_MCEN     0x02
#define ERXFCON_BCEN     0x01
// EIE Register Bits
#define EIE_INTIE        0x80
#define EIE_PKTIE        0x40
#define EIE_DMAIE        0x20
#define EIE_LINKIE       0x10
#define EIE_TXIE         0x08
#define EIE_WOLIE        0x04
#define EIE_TXERIE       0x02
#define EIE_RXERIE       0x01
// EIR Register Bits
#define EIR_PKTIF        0x40
#define EIR_DMAIF        0x20
#define EIR_LINKIF       0x10
#define EIR_TXIF         0x08
#define EIR_WOLIF        0x04
#define EIR_TXERIF       0x02
#define EIR_RXERIF       0x01
// ESTAT Register Bits
#define ESTAT_INT        0x80
#define ESTAT_LATECOL    0x10
#define ESTAT_RXBUSY     0x04
#define ESTAT_TXABRT     0x02
#define ESTAT_CLKRDY     0x01
// ECON2 Register Bits
#define ECON2_AUTOINC    0x80
#define ECON2_PKTDEC     0x40
#define ECON2_PWRSV      0x20
#define ECON2_VRPS       0x08
// ECON1 Register Bits
#define ECON1_TXRST      0x80
#define ECON1_RXRST      0x40
#define ECON1_DMAST      0x20
#define ECON1_CSUMEN     0x10
#define ECON1_TXRTS      0x08
#define ECON1_RXEN       0x04
#define ECON1_BSEL1      0x02
#define ECON1_BSEL0      0x01
// MACON1 Register Bits
#define MACON1_LOOPBK    0x10
#define MACON1_TXPAUS    0x08
#define MACON1_RXPAUS    0x04
#define MACON1_PASSALL   0x02
#define MACON1_MARXEN    0x01
// MACON2 Register Bits
#define MACON2_MARST     0x80
```

```
#define MACON2_RNDRST    0x40
#define MACON2_MARXRST   0x08
#define MACON2_RFUNRST   0x04
#define MACON2_MATXRST   0x02
#define MACON2_TFUNRST   0x01
// MACON3 Register Bits
#define MACON3_PADCFG2   0x80
#define MACON3_PADCFG1   0x40
#define MACON3_PADCFG0   0x20
#define MACON3_TXCRCEN   0x10
#define MACON3_PHDRLEN   0x08
#define MACON3_HFRMLEN   0x04
#define MACON3_FRMLNEN   0x02
#define MACON3_FULDPX    0x01
// MICMD Register Bits
#define MICMD_MIISCAN    0x02
#define MICMD_MIIRD      0x01
// MISTAT Register Bits
#define MISTAT_NVALID    0x04
#define MISTAT_SCAN      0x02
#define MISTAT_BUSY      0x01
// PHY PHCON1 Register Bits
#define PHCON1_PRST      0x8000
#define PHCON1_PLOOPBK   0x4000
#define PHCON1_PPWRSV    0x0800
#define PHCON1_PDPXMD    0x0100
// PHY PHSTAT1 Register Bits
#define PHSTAT1_PFDPX    0x1000
#define PHSTAT1_PHDPX    0x0800
#define PHSTAT1_LLSTAT   0x0004
#define PHSTAT1_JBSTAT   0x0002
// PHY PHCON2 Register Bits
#define PHCON2_FRCLINK   0x4000
#define PHCON2_TXDIS     0x2000
#define PHCON2_JABBER    0x0400
#define PHCON2_HDLDIS    0x0100

// Packet Control-Byte Bits
#define PKTCTRL_PHUGEEN  0x08
#define PKTCTRL_PPADEN   0x04
#define PKTCTRL_PCRCEN   0x02
#define PKTCTRL_POVERRIDE 0x01

// SPI operation codes
#define ENC28J60_READ_CTRL_REG       0x00
#define ENC28J60_READ_BUF_MEM        0x3A
#define ENC28J60_WRITE_CTRL_REG      0x40
#define ENC28J60_WRITE_BUF_MEM       0x7A
#define ENC28J60_BIT_FIELD_SET       0x80
#define ENC28J60_BIT_FIELD_CLR       0xA0
#define ENC28J60_SOFT_RESET          0xFF


// Το RXSTART_INIT πρέπει να είναι μηδέν
// εκχωρείται ολόκληρη η εσωτερική μνήμη packet buffer

// αρχή στο 0
#define RXSTART_INIT     0x0
// τέλος του buffer λήψης
#define RXSTOP_INIT      (0x1FFF-0x0600-1)
// αρχή buffer μετάδοσης στο 0x1FFF-0x0600
// (ρυθμός κατάλληλος για ένα ολόκληρο πλαίσιο ethernet)
#define TXSTART_INIT     (0x1FFF-0x0600)
// τέλος buffer λήψης στο τέλος της μνήμης
#define TXSTOP_INIT      0x1FFF
//
// μέγιστο μήκος πλαισίου που δέχεται ο controller
// (το μέγιστο μήκος πλαισίου ethernet είναι 1518)
```

```
#define MAX_FRAMELEN        1500
//#define MAX_FRAMELEN      600


// Συναρτήσεις
extern uint8_t enc28j60ReadOp(uint8_t op, uint8_t address);
extern void enc28j60WriteOp(uint8_t op, uint8_t address, uint8_t data);
extern void enc28j60ReadBuffer(uint16_t len, uint8_t* data);
extern void enc28j60WriteBuffer(uint16_t len, uint8_t* data);
extern void enc28j60SetBank(uint8_t address);
extern uint8_t enc28j60Read(uint8_t address);
extern void enc28j60Write(uint8_t address, uint8_t data);
extern void enc28j60PhyWrite(uint8_t address, uint16_t data);
extern void enc28j60clkout(uint8_t clk);
extern void enc28j60Init(uint8_t* macaddr);
extern void enc28j60PacketSend(uint16_t len, uint8_t* packet);
extern uint16_t enc28j60PacketReceive(uint16_t maxlen, uint8_t* packet);
extern uint8_t enc28j60getrev(void);

#endif
```

integrator.h

```
// Ολοκληρωτής

#ifndef INTEGRATOR_H
#define INTEGRATOR_H

#define SAMPLES128         //comment out for 64 samples

#ifdef SAMPLES128
#define SAMPLES 128
#define SAMPLES_BITS 7
#else
#define SAMPLES 64
#define SAMPLES_BITS 6
#endif

// Συναρτήσεις
extern void initbat();
extern uint8_t readbat();


#endif
```

ip_arp_udp_tcp.h

```
// Συναρτήσεις IP/ARP/UDP/TCP

#ifndef IP_ARP_UDP_TCP_H
#define IP_ARP_UDP_TCP_H
#include <avr/pgmspace.h>

// Συνάρτηση αρχικοποίησης
extern void init_ip_arp_udp_tcp(uint8_t *mymac,uint8_t *myip,uint8_t wwwp);
// Συναρτήσεις λειτουργιών
extern uint8_t eth_type_is_arp_and_my_ip(uint8_t *buf,uint16_t len);
extern uint8_t eth_type_is_ip_and_my_ip(uint8_t *buf,uint16_t len);
extern void make_arp_answer_from_request(uint8_t *buf);
extern void make_echo_reply_from_request(uint8_t *buf,uint16_t len);
extern void make_udp_reply_from_request(uint8_t *buf,char *data,uint8_t
datalen,uint16_t port);
extern void make_tcp_synack_from_syn(uint8_t *buf);
extern void init_len_info(uint8_t *buf);
extern uint16_t get_tcp_data_pointer(void);
```

```
extern uint16_t fill_tcp_data_p(uint8_t *buf,uint16_t pos, const prog_char
*progmem_s);
extern uint16_t fill_tcp_data(uint8_t *buf,uint16_t pos, const char *s);
extern void make_tcp_ack_from_any(uint8_t *buf);
extern void make_tcp_ack_with_data(uint8_t *buf,uint16_t dlen);


#endif
```

lcd.h

```
// Βιβλιοθήκη οθόνης LCD τύπου HD44780

#ifndef LCD_H
#define LCD_H
#include <avr/pgmspace.h>

typedef unsigned char  u08;
typedef unsigned short u16;

/* θέσεις bits του instruction register */
#define LCD_CLR             0      /* DB0: καθαρισμός οθόνης */
#define LCD_HOME            1      /* DB1: επιστροφή στην αρχική θέση */
#define LCD_ENTRY_MODE      2      /* DB2: ορισμός σε entry mode */
#define LCD_ENTRY_INC       1      /* DB1: 1=αύξηση, 0=μείωση */
#define LCD_ENTRY_SHIFT     0      /* DB2: 1=display shift on */
#define LCD_ON              3      /* DB3: άναμα display/cursor */
#define LCD_ON_DISPLAY      2      /* DB2: άναμα display */
#define LCD_ON_CURSOR       1      /* DB1: άναμα cursor */
#define LCD_ON_BLINK        0      /* DB0: λειτουργία blink του cursor */
#define LCD_MOVE            4      /* DB4: μετακίνηση cursor/display */
#define LCD_MOVE_DISP       3      /* DB3: μετακίνηση display (0-> cursor)
*/
#define LCD_MOVE_RIGHT      2      /* DB2: μετακίνηση δεξιά (0-> αριστερά)
*/
#define LCD_FUNCTION        5      /* DB5: ορισμός λειτουργίας */
#define LCD_FUNCTION_8BIT   4      /* DB4: ορισμός 8BIT mode (0->4BIT mode)
*/
#define LCD_FUNCTION_2LINES 3      /* DB3: δύο γραμμές (0->one line) */
#define LCD_FUNCTION_10DOTS 2      /* DB2: 5x10 σύμβολα (0->5x7 font) */
#define LCD_CGRAM           6      /* DB6: ορισμός CG RAM address */
#define LCD_DDRAM           7      /* DB7: ορισμός DD RAM address */
#define LCD_BUSY            7      /* DB7: η οθόνη είναι απασχολημένη */

// ορισμός entry mode: κατάσταση μετακίνησης display on/off
// μείωση/αύξηση κατεύθυνσης μετακίνησης του cursor */

/* μετακίνηση display off, μείωση κατεύθυνσης μετακίνησης του cursor */
#define LCD_ENTRY_DEC            0x04
/* μετακίνηση display on, μείωση κατεύθυνσης μετακίνησης του cursor */
#define LCD_ENTRY_DEC_SHIFT      0x05
/* μετακίνηση display off, αύξηση κατεύθυνσης μετακίνησης του cursor */
#define LCD_ENTRY_INC_           0x06
/* μετακίνηση display on, αύξηση κατεύθυνσης μετακίνησης του cursor */
#define LCD_ENTRY_INC_SHIFT      0x07

/* display on/off, cursor on/off, blinking char στη θέση του cursor */
#define LCD_DISP_OFF             0x08 /* display off */
#define LCD_DISP_ON              0x0C /* display on, cursor off */
#define LCD_DISP_ON_BLINK        0x0D /* display on, cursor off, blink char
*/
#define LCD_DISP_ON_CURSOR       0x0E /* display on, cursor on */
#define LCD_DISP_ON_CURSOR_BLINK 0x0F /* display on, cursor on, blink char
*/

/* μετακίνηση cursor/display */
#define LCD_MOVE_CURSOR_LEFT     0x10 /* μετακίνηση cursor αριστερά */
```

```
#define LCD_MOVE_CURSOR_RIGHT    0x14 /* μετακίνηση cursor δεξιά */
#define LCD_MOVE_DISP_LEFT       0x18 /* μετακίνηση display αριστερά */
#define LCD_MOVE_DISP_RIGHT      0x1C /* μετακίνηση display δεξιά */

// ορισμός λειτουργίας: ορισμός μήκους δεδομένων interface
// και αριθμού γραμμών display

/* 4-bit interface, μία γραμμή, 5x7 σημεία */
#define LCD_FUNCTION_4BIT_1LINE  0x20
/* 4-bit interface, δύο γραμμές, 5x7 σημεία */
#define LCD_FUNCTION_4BIT_2LINES 0x28
/* 8-bit interface, μία γραμμή, 5x7 σημεία */
#define LCD_FUNCTION_8BIT_1LINE  0x30
/* 8-bit interface, δύο γραμμές, 5x7 σημεία */
#define LCD_FUNCTION_8BIT_2LINES 0x38

#define LCD_MODE_DEFAULT     ((1<<LCD_ENTRY_MODE) | (1<<LCD_ENTRY_INC) )


// Μακροεντολές

/* μετακίνηση cursor στην αρχική θέση */
#define lcd_home() lcd_command(1 << LCD_HOME)
/* καθαρισμός οθόνης και μετακίνηση cursor στην αρχική θέση */
#define lcd_clrscr() lcd_command(1 << LCD_CLR)

// Συναρτήσεις
extern void lcd_command(u08 cmd);
extern void lcd_gotoxy(u08 x, u08 y); /* γραμμή 1 y=0, γραμμή 2 y=1 */
extern void lcd_putc(char c); /* εκτύπωση χαρακτήρα στη θέση του cursor */
extern void lcd_puts(const char *s); /* εκτύπωση string στη οθόνη (χωρίς LF)
*/

// Αρχικοποίηση της LCD
extern void lcd_init(u08 dispAttr);
// εκτύπωση string από την program memory (χωρίς LF)
// η αποθήκευση strings γίνεται πχ char *str =PSTR("hello world");
// και η εμφάνηση στην οθόνη με lcd_puts_p(str);*/
extern void lcd_puts_p(const prog_char *progmem_s);


// Μακροεντολές για αυτόματη αποθήκευση
// string constants στην program memory
#ifndef P
#define P(s) ({static const char c[] __attribute__ ((progmem)) = s;c;})
#endif
#define lcd_puts_P(__s)          lcd_puts_p(P(__s))


#endif
```

lcd_hw.h

```
// Βιβλιοθήκη hardware για την οθόνη LCD τύπου HD44780

#ifndef LCD_HW_H
#define LCD_HW_H


/* ορισμοί ανάλογα με την σύνδεση του hardware */

#define LCD_DATA_PIN_D7  PD0     /* θέση γραμμής D7 */
#define LCD_DATA_PIN_D6  PD1     /* θέση γραμμής D6 */
#define LCD_DATA_PIN_D5  PD3     /* θέση γραμμής D5 */
#define LCD_DATA_PIN_D4  PD4     /* θέση γραμμής D4 */

#define LCD_DATA_DDR_D7  DDRD    /* κατεύθυνση γραμμής D7 */
```

```
#define LCD_DATA_DDR_D6  DDRD     /* κατεύθυνση γραμμής D6 */
#define LCD_DATA_DDR_D5  DDRD     /* κατεύθυνση γραμμής D5 */
#define LCD_DATA_DDR_D4  DDRD     /* κατεύθυνση γραμμής D4 */

#define LCD_DATA_PORT_D7 PORTD    /* θύρα γραμμής D7 */
#define LCD_DATA_PORT_D6 PORTD    /* θύρα γραμμής D6 */
#define LCD_DATA_PORT_D5 PORTD    /* θύρα γραμμής D5 */
#define LCD_DATA_PORT_D4 PORTD    /* θύρα γραμμής D4 */

#define LCD_RS_DDR       DDRD     /* κατεύθυνση γραμμής RS */
#define LCD_RS_PORT      PORTD    /* θύρα γραμμής RS */
#define LCD_RS_PIN       PD6      /* θέση γραμμής RS */
#define LCD_E_DDR        DDRD     /* κατεύθυνση γραμμής Enable */
#define LCD_E_PORT       PORTD    /* θύρα γραμμής Enable */
#define LCD_E_PIN        PD5      /* θέση γραμμής Enable */

#define LCD_LINES           2     /* αριθμός γραμμών */

/* διεύθυνση DDRAM του πρώτου χαρακτήρα κάθε γραμμής */
#define LCD_START_LINE1  0x00
#define LCD_START_LINE2  0x40
#define LCD_START_LINE3  0x14
#define LCD_START_LINE4  0x54

#endif
```

net.h

```
// Βιβλιοθήκη IP/ARP/UDP/TCP για το chip ENC28J60

#ifndef NET_H
#define NET_H

/* συμβολισμοί:
_P = θέση πεδίου
_V = τιμή πεδίου */

// ******* ETH *******
#define ETH_HEADER_LEN     14
// τιμές ορισμένων bytes:
#define ETHTYPE_ARP_H_V 0x08
#define ETHTYPE_ARP_L_V 0x06
#define ETHTYPE_IP_H_V  0x08
#define ETHTYPE_IP_L_V  0x00

// θέσεις bytes στο πλαίσιο ethernet

#define ETH_TYPE_H_P 12    // πεδίο τύπου ethernet (2 bytes)
#define ETH_TYPE_L_P 13

#define ETH_DST_MAC 0
#define ETH_SRC_MAC 6

// ******* ARP *******
#define ETH_ARP_OPCODE_REPLY_H_V 0x0
#define ETH_ARP_OPCODE_REPLY_L_V 0x02

#define ETHTYPE_ARP_L_V 0x06
// arp.dst.ip
#define ETH_ARP_DST_IP_P 0x26
// arp.opcode
#define ETH_ARP_OPCODE_H_P 0x14
#define ETH_ARP_OPCODE_L_P 0x15
// arp.src.mac
#define ETH_ARP_SRC_MAC_P 0x16
#define ETH_ARP_SRC_IP_P 0x1c
#define ETH_ARP_DST_MAC_P 0x20
```

```
#define ETH_ARP_DST_IP_P 0x26

// ******* IP *******
#define IP_HEADER_LEN      20
// ip.src
#define IP_SRC_P 0x1a
#define IP_DST_P 0x1e
#define IP_HEADER_LEN_VER_P 0xe
#define IP_CHECKSUM_P 0x18
#define IP_TTL_P 0x16
#define IP_FLAGS_P 0x14
#define IP_P 0xe
#define IP_TOTLEN_H_P 0x10
#define IP_TOTLEN_L_P 0x11

#define IP_PROTO_P 0x17

#define IP_PROTO_ICMP_V 1
#define IP_PROTO_TCP_V 6
#define IP_PROTO_UDP_V 17  // 17=0x11
// ******* ICMP *******
#define ICMP_TYPE_ECHOREPLY_V 0
#define ICMP_TYPE_ECHOREQUEST_V 8

#define ICMP_TYPE_P 0x22
#define ICMP_CHECKSUM_P 0x24

// ******* UDP *******
#define UDP_HEADER_LEN     8

#define UDP_SRC_PORT_H_P 0x22
#define UDP_SRC_PORT_L_P 0x23
#define UDP_DST_PORT_H_P 0x24
#define UDP_DST_PORT_L_P 0x25

#define UDP_LEN_H_P 0x26
#define UDP_LEN_L_P 0x27
#define UDP_CHECKSUM_H_P 0x28
#define UDP_CHECKSUM_L_P 0x29
#define UDP_DATA_P 0x2a

// ******* TCP *******
#define TCP_SRC_PORT_H_P 0x22
#define TCP_SRC_PORT_L_P 0x23
#define TCP_DST_PORT_H_P 0x24
#define TCP_DST_PORT_L_P 0x25

#define TCP_SEQ_H_P 0x26   // ο αριθμός tcp seq είναι 4 bytes (0x26-0x29)
#define TCP_SEQACK_H_P 0x2a

#define TCP_FLAGS_P 0x2f   // σημαίες: SYN=2
#define TCP_FLAGS_SYN_V 2
#define TCP_FLAGS_FIN_V 1
#define TCP_FLAGS_PUSH_V 8
#define TCP_FLAGS_SYNACK_V 0x12
#define TCP_FLAGS_ACK_V 0x10
#define TCP_FLAGS_PSHACK_V 0x18

#define TCP_HEADER_LEN_PLAIN 20  // μήκος header χωρίς options
#define TCP_HEADER_LEN_P 0x2e
#define TCP_CHECKSUM_H_P 0x32
#define TCP_CHECKSUM_L_P 0x33
#define TCP_OPTIONS_P 0x36

#endif
```

sensirion_protocol.h

```
// Πρωτόκολλο επικοινωνίας αισθητήρα sentirion SHT11

#ifndef SENSIRION_PROTOCOL_H
#define SENSIRION_PROTOCOL_H

// Συναρτήσεις
extern void s_connectionreset(void);
extern char s_softreset(void);
extern char s_measure(unsigned int *p_value, unsigned char mode);
extern int calc_sth11_temp(unsigned int t);
extern unsigned char rhcalc_int(unsigned int s);
extern unsigned char calc_sth11_humi(unsigned int h, int t);
extern int calc_dewpoint(unsigned char rh,int t);
extern int log10_approx(unsigned char x);

#endif
```

timeout.h

```
// Επικεφαλίδα timeout
#ifndef TOUT_H
#define TOUT_H

// Συνάρτηση
extern void delay_ms(unsigned int ms);

#endif
```

Ο κώδικας του logger

Form1.vb

```
Imports MySql.Data.MySqlClient
Imports System.Net
Imports System.IO
Imports System.Xml

Public Class frmMain

    Dim conn As MySqlConnection
    Dim data As New DataTable
    Dim da As MySqlDataAdapter
    Dim cb As MySqlCommandBuilder
    Dim myCommand As MySqlCommand
    Dim myInsertQuery As String
    Dim myXMLdata(0 To 10) As String
    Dim serverDown As Boolean
    Dim myNow As DateTime
    Dim moreMinutes As Integer = 0
    Dim skip As Boolean = False

    Private Sub cmdConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdConnect.Click

        If Not conn Is Nothing Then conn.Close()

        Dim connStr As String
        connStr = "server=" & txtDB.Text & ";user id=" & txtUser.Text & ";
password=" & txtPass.Text & "; database=weather"


        Try
            conn = New MySqlConnection(connStr)
            conn.Open()
            cmdConnect.Text = "Connected"
            lblError.Text = "OK"
            lblError.ForeColor = Color.Black
        Catch ex As MySqlException
            lblError.Text = "Error: " + ex.Message
            lblError.ForeColor = Color.Red
            cmdConnect.Text = "Connect"
        End Try

        Try
            da = New MySqlDataAdapter("(SELECT
time,temp,humi,dew,vsol,vbat,amp,pres,sea FROM webdata " _
& "ORDER BY time DESC limit 4320) ORDER BY time ASC", conn)
            cb = New MySqlCommandBuilder(da)
            data.Clear()
            da.Fill(data)
            myDataGrid.DataSource = data

            'Format Table
            myDataGrid.Width = 385
            myDataGrid.Columns("time").Width = 65
            myDataGrid.Columns("temp").Width = 35
            myDataGrid.Columns("humi").Width = 35
            myDataGrid.Columns("dew").Width = 30
            myDataGrid.Columns("vsol").Width = 30
            myDataGrid.Columns("vbat").Width = 30
            myDataGrid.Columns("amp").Width = 32
            myDataGrid.Columns("pres").Width = 42
            myDataGrid.Columns("sea").Width = 42
```

```
            myDataGrid.Columns(0).DefaultCellStyle.Format = "T"

            myDataGrid.CurrentCell =
myDataGrid.Rows(myDataGrid.Rows.GetRowCount(DataGridViewElementStates.Visibl
e) - 1).Cells(0)
            lblError.Text = "OK"
            lblError.ForeColor = Color.Black
        Catch ex As Exception
            lblError.Text = "Error: " + ex.Message
            lblError.ForeColor = Color.Red
        End Try

    End Sub

    Private Sub writeTableToXML(ByVal samples As Integer, ByVal period As
Integer, ByVal keyCode As String)
        Dim entries As Integer = data.Rows.Count - 1

        'txtPath.Text = "C:\xampp\htdocs\weather\Data" '-----------------
debug

        If entries < samples Then samples = entries
        'XML file 1
        Using myWriter As System.IO.StreamWriter = New
System.IO.StreamWriter(txtPath.Text & "\TempData" & keyCode & ".xml")

            myWriter.WriteLine("<chart caption='Temperature' numdivlines='9'
lineThickness='2' " _
& "showValues='0' anchorRadius='3' anchorBgAlpha='50'
showAlternateVGridColor='1' " _
& "numVisiblePlot='8' animation='1' inDecimalSeparator=',' >")
            myWriter.WriteLine("<categories>")

            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<category label='" &
data.Rows.Item(counter).Item(0).TimeOfDay.ToString & "' />")

            Next counter
            myWriter.WriteLine("</categories>")
            myWriter.WriteLine("<dataset seriesName='temp' color='800080'
anchorBorderColor='800080'>")

            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<set value='" &
data.Rows.Item(counter).Item(1).ToString & "' />")

            Next counter

            myWriter.WriteLine("</dataset>")

            myWriter.WriteLine("</chart>")

        End Using

        'XML file 2
        Using myWriter As System.IO.StreamWriter = New
System.IO.StreamWriter(txtPath.Text & "\HumiData" & keyCode & ".xml")

            myWriter.WriteLine("<chart caption='Humidity and Dew point'
subCaption='versus Time' numdivlines='9' lineThickness='2' " _
& "showValues='0' anchorRadius='3' anchorBgAlpha='50'
showAlternateVGridColor='1' " _
& "numVisiblePlot='8' animation='1' inDecimalSeparator=',' >")
            myWriter.WriteLine("<categories>")
```

```
            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<category label='" &
data.Rows.Item(counter).Item(0).TimeOfDay.ToString & "' />")

            Next counter
            myWriter.WriteLine("</categories>")

            'first diagram
            myWriter.WriteLine("<dataset seriesName='humi' color='800080'
anchorBorderColor='800080'>")

            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<set value='" &
data.Rows.Item(counter).Item(2).ToString & "' />")

            Next counter
            myWriter.WriteLine("</dataset>")

            'second diagram
            myWriter.WriteLine("<dataset seriesName='dew' color='1D8BD1'
anchorBorderColor='1D8BD1'>")

            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<set value='" &
data.Rows.Item(counter).Item(3).ToString & "' />")

            Next counter

            myWriter.WriteLine("</dataset>")

            myWriter.WriteLine("</chart>")

        End Using

        'XML file 3
        Using myWriter As System.IO.StreamWriter = New
System.IO.StreamWriter(txtPath.Text & "\VoltData" & keyCode & ".xml")

            myWriter.WriteLine("<chart caption='Solar and Battery voltage'
subCaption='Solar power' numdivlines='9' lineThickness='2' " _
& "showValues='0' anchorRadius='3' anchorBgAlpha='50'
showAlternateVGridColor='1' " _
& "numVisiblePlot='8' animation='1' inDecimalSeparator=',' >")
            myWriter.WriteLine("<categories>")

            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<category label='" &
data.Rows.Item(counter).Item(0).TimeOfDay.ToString & "' />")

            Next counter
            myWriter.WriteLine("</categories>")

            'first diagram
            myWriter.WriteLine("<dataset seriesName='vsol' color='800080'
anchorBorderColor='800080'>")

            For counter As Integer = (entries - samples) To (entries) Step
period
```

```
                myWriter.WriteLine("<set value='" &
data.Rows.Item(counter).Item(4).ToString & "' />")

            Next counter
            myWriter.WriteLine("</dataset>")

            'second diagram
            myWriter.WriteLine("<dataset seriesName='vbat' color='1D8BD1'
anchorBorderColor='1D8BD1'>")

            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<set value='" &
data.Rows.Item(counter).Item(5).ToString & "' />")

            Next counter
            myWriter.WriteLine("</dataset>")

            'third diagram
            myWriter.WriteLine("<dataset seriesName='watt' color='2AD62A'
anchorBorderColor='2AD62A'>")

            For counter As Integer = (entries - samples) To (entries) Step
period
                Dim watts As Integer = data.Rows.Item(counter).Item(6) *
data.Rows.Item(counter).Item(4) / 1000
                myWriter.WriteLine("<set value='" & watts.ToString & "' />")

            Next counter
            myWriter.WriteLine("</dataset>")

            myWriter.WriteLine("</chart>")

        End Using

        'XML file 4
        Using myWriter As System.IO.StreamWriter = New
System.IO.StreamWriter(txtPath.Text & "\PresData" & keyCode & ".xml")

            myWriter.WriteLine("<chart caption='Air Pressure'
subCaption='Pressure at sea level' numdivlines='9' lineThickness='2' " _
& "showValues='0' anchorRadius='3' anchorBgAlpha='50'
showAlternateVGridColor='1' formatNumber='0' formatNumberScale='0' " _
& "numVisiblePlot='8' animation='1' inDecimalSeparator=','
setAdaptiveYMin='1' decimals='4'> ")
            myWriter.WriteLine("<categories>")

            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<category label='" &
data.Rows.Item(counter).Item(0).TimeOfDay.ToString & "' />")

            Next counter
            myWriter.WriteLine("</categories>")

            'first diagram
            myWriter.WriteLine("<dataset seriesName='pres' color='800080'
anchorBorderColor='800080'>")

            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<set value='" &
data.Rows.Item(counter).Item(7).ToString & "' />")

            Next counter
```

```
            myWriter.WriteLine("</dataset>")

            'second diagram
            myWriter.WriteLine("<dataset seriesName='sea' color='1D8BD1'
anchorBorderColor='1D8BD1'>")

            For counter As Integer = (entries - samples) To (entries) Step
period

                myWriter.WriteLine("<set value='" &
data.Rows.Item(counter).Item(8).ToString & "' />")

            Next counter
            myWriter.WriteLine("</dataset>")

            myWriter.WriteLine("</chart>")

        End Using

    End Sub


    Private Sub cmdXML_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdXML.Click

        '30 minutes data
        Call writeTableToXML(30, 1, "")

        '1 day data
        Call writeTableToXML(1440, 20, "1day")

        '3 day data
        Call writeTableToXML(4320, 60, "3day")

    End Sub

    Private Sub cmdPath_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdPath.Click

        If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then
            txtPath.Text = FolderBrowserDialog1.SelectedPath
        End If

    End Sub

    Private Sub cmdStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdStart.Click

        Timer1.Start()
        'myXMLdata(10) = 2 '---------------debug

    End Sub

    Private Sub rndData() '----------------------------debug
        myXMLdata(0) = CInt(Int((40 - 10 + 1) * Rnd() + 10))
        myXMLdata(1) = CInt(Int((80 - 10 + 1) * Rnd() + 10))
        myXMLdata(2) = CInt(Int((30 - 5 + 1) * Rnd() + 5))
        myXMLdata(3) = CInt(Int((15 - 4 + 1) * Rnd() + 4))
        myXMLdata(4) = CInt(Int((18 - 10 + 1) * Rnd() + 10))
        myXMLdata(5) = CInt(Int((800 - 200 + 1) * Rnd() + 200))
        myXMLdata(6) = CInt(Int((1000 - 850 + 1) * Rnd() + 850))
        myXMLdata(7) = CInt(Int((1100 - 900 + 1) * Rnd() + 900))
        myXMLdata(8) = "yes"
        myXMLdata(9) = Now() '"2008-03-24 23:57:32"
        myXMLdata(10) = myXMLdata(10) - 1
        skip = True

    End Sub
```

```
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

        Try
            Call cmdCheck_Click(sender, e)
            'Call rndData() '----------------------------------------debug

            'Save to database
            If (serverDown = False) And (skip = False) Then
                myInsertQuery = "INSERT INTO webdata
(time,temp,humi,dew,vsol,vbat,amp,pres,sea) VALUES" _
& "(NOW( ) , " & myXMLdata(0) & ", " & myXMLdata(1) & ", " & myXMLdata(2) &
", " _
& myXMLdata(3) & ", " & myXMLdata(4) & ", " & myXMLdata(5) & ", " _
& myXMLdata(6) & ", " & myXMLdata(7) & ");"

                da = New MySqlDataAdapter("(SELECT
time,temp,humi,dew,vsol,vbat,amp,pres,sea FROM webdata " _
& "ORDER BY time DESC limit 4320) ORDER BY time ASC", conn)
                cb = New MySqlCommandBuilder(da)

                myCommand = New MySqlCommand(myInsertQuery)
                myCommand.Connection = conn
                myCommand.ExecuteNonQuery()
                Call cmdXML_Click(sender, e)
                data.Clear()
                da.Fill(data)
                myDataGrid.DataSource = data
                myDataGrid.Columns(0).DefaultCellStyle.Format = "T"
                myDataGrid.CurrentCell =
myDataGrid.Rows(myDataGrid.Rows.GetRowCount(DataGridViewElementStates.Visibl
e) - 1).Cells(0)
                lblError.Text = "OK"
                lblError.ForeColor = Color.Black
            End If

            If skip = True Then
                Timer1.Stop()
                Timer2.Start()
            End If

        Catch ex As Exception
            lblError.Text = "Error: " + ex.Message
            lblError.ForeColor = Color.Red
            If Not (ex.Message.StartsWith("You have an error in your SQL
syntax")) Then
                Call cmdConnect_Click(sender, e)
            End If

        End Try

    End Sub

    Private Sub cmdStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdStop.Click

        Timer1.Stop()

    End Sub

    Private Sub cmdCheck_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCheck.Click

        Dim settings As XmlReaderSettings = New XmlReaderSettings()
        settings.IgnoreComments = True
        settings.IgnoreProcessingInstructions = True
        settings.IgnoreWhitespace = True
```

```
        Dim resolver As XmlUrlResolver = New XmlUrlResolver()
        resolver.Credentials = System.Net.CredentialCache.DefaultCredentials
        ' Set the reader settings object to use the resolver.
        settings.XmlResolver = resolver

        ' Create the XmlReader object.
        Dim myXMLreader As XmlReader

        Try

            myXMLreader = XmlReader.Create("txtSite.Text + "/xml/" +
Format(Now, "yyyy-MM-dd.hh:mm:ss"), settings)
            'myXMLreader = XmlReader.Create("http://192.168.1.50/xml/" +
Format(Now, "yyyy-MM-dd.HH:mm:ss"), settings)
            'myXMLreader =
XmlReader.Create("http://127.0.0.1:8080/weather/data.xml", settings) '------
-debug
            'myXMLreader =
XmlReader.Create("http://users.ntua.gr/el01018/data.xml", settings) '-------
debug

            myXMLreader.Read()
            myXMLreader.ReadStartElement("root")
            myXMLreader.ReadStartElement("temp")
            myXMLdata(0) = myXMLreader.ReadString()
            myXMLreader.ReadEndElement()

            myXMLreader.ReadStartElement("humi")
            myXMLdata(1) = myXMLreader.ReadString()
            myXMLreader.ReadEndElement()

            myXMLreader.ReadStartElement("dew")
            myXMLdata(2) = myXMLreader.ReadString()
            myXMLreader.ReadEndElement()

            myXMLreader.ReadStartElement("vsol")
            myXMLdata(3) = myXMLreader.ReadString()
            myXMLreader.ReadEndElement()

            myXMLreader.ReadStartElement("vbat")
            myXMLdata(4) = myXMLreader.ReadString()
            myXMLreader.ReadEndElement()

            myXMLreader.ReadStartElement("amp")
            myXMLdata(5) = myXMLreader.ReadString()
            myXMLreader.ReadEndElement()

            myXMLreader.ReadStartElement("pres")
            myXMLdata(6) = myXMLreader.ReadString()
            myXMLreader.ReadEndElement()

            myXMLreader.ReadStartElement("sea")
            myXMLdata(7) = myXMLreader.ReadString()
            myXMLreader.ReadEndElement()

            myXMLreader.ReadStartElement("more")
            myXMLdata(8) = myXMLreader.ReadString()
            myXMLreader.ReadEndElement()

            If (myXMLdata(8) = "yes") Then
                myXMLreader.ReadStartElement("time")
                myXMLdata(9) = myXMLreader.ReadString()
                myXMLreader.ReadEndElement()

                myXMLreader.ReadStartElement("remain")
                myXMLdata(10) = myXMLreader.ReadString()
                myXMLreader.ReadEndElement()
```

```
                skip = True
            End If

            myXMLreader.ReadEndElement()

            cmdCheck.Text = "Checked"
            lblError.Text = "OK"
            lblError.ForeColor = Color.Black
            serverDown = False
        Catch ex As XmlException
            lblError.Text = "XML string error. Not well-formed XML"
            lblError.ForeColor = Color.Red
            cmdCheck.Text = "Check"
        Catch ex As WebException
            serverDown = True
            lblError.Text = "Web site down. XML not found"
            lblError.ForeColor = Color.Red
            cmdCheck.Text = "Check"
        Catch ex As Exception
            lblError.Text = "Error: " + ex.Message
            lblError.ForeColor = Color.Red
            cmdCheck.Text = "Check"
        End Try

    End Sub

    Private Sub lblAbout_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lblAbout.Click
        frmAboutBox.Show()
    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick

        Try
            'skip first time
            If skip = False Then
                Call cmdCheck_Click(sender, e)
                'Call rndData() '----------------------------------------
debug
            End If
            skip = False

            'Save to database
            If serverDown = False Then

                'Fix record time for SQL
                moreMinutes += 1
                myNow = DateTime.Parse(myXMLdata(9)).AddMinutes(moreMinutes)

                myInsertQuery = "INSERT INTO webdata
(time,temp,humi,dew,vsol,vbat,amp,pres,sea) VALUES (" _
& Format(myNow, "\'yyyy-MM-dd HH:mm:ss\'") & " , " & myXMLdata(0) & ", " &
myXMLdata(1) & ", " & myXMLdata(2) & ", " _
& myXMLdata(3) & ", " & myXMLdata(4) & ", " & myXMLdata(5) & ", " _
& myXMLdata(6) & ", " & myXMLdata(7) & ");"

                da = New MySqlDataAdapter("(SELECT
time,temp,humi,dew,vsol,vbat,amp,pres,sea FROM webdata " _
& "ORDER BY time DESC limit 4320) ORDER BY time ASC", conn)
                cb = New MySqlCommandBuilder(da)

                myCommand = New MySqlCommand(myInsertQuery)
                myCommand.Connection = conn
                myCommand.ExecuteNonQuery()
                Call cmdXML_Click(sender, e)
                data.Clear()
                da.Fill(data)
```

```
                myDataGrid.DataSource = data
                myDataGrid.Columns(0).DefaultCellStyle.Format = "T"
                myDataGrid.CurrentCell =
myDataGrid.Rows(myDataGrid.Rows.GetRowCount(DataGridViewElementStates.Visibl
e) - 1).Cells(0)
                lblError.Text = "OK"
                lblError.ForeColor = Color.Black
            Else
                myXMLdata(10) = 0   'Restart timer 1
            End If

            If myXMLdata(10) <= 0 Then
                Timer2.Stop()
                Timer1.Start()
                moreMinutes = 0
            End If

        Catch ex As Exception
            lblError.Text = "Error: " + ex.Message
            lblError.ForeColor = Color.Red
            If Not (ex.Message.StartsWith("You have an error in your SQL
syntax")) Then
                Call cmdConnect_Click(sender, e)
            End If

        End Try

    End Sub

End Class
```

Form1.Designer.vb

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class frmMain
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container
        Dim resources As System.ComponentModel.ComponentResourceManager =
New System.ComponentModel.ComponentResourceManager(GetType(frmMain))
        Me.cmdConnect = New System.Windows.Forms.Button
        Me.cmdXML = New System.Windows.Forms.Button
        Me.txtUser = New System.Windows.Forms.TextBox
        Me.txtPass = New System.Windows.Forms.TextBox
        Me.lblUser = New System.Windows.Forms.Label
        Me.lblPass = New System.Windows.Forms.Label
        Me.txtDB = New System.Windows.Forms.TextBox
```

```
        Me.cmdStart = New System.Windows.Forms.Button
        Me.lblDB = New System.Windows.Forms.Label
        Me.FolderBrowserDialog1 = New
System.Windows.Forms.FolderBrowserDialog
        Me.txtPath = New System.Windows.Forms.TextBox
        Me.cmdPath = New System.Windows.Forms.Button
        Me.Timer1 = New System.Windows.Forms.Timer(Me.components)
        Me.lblPath = New System.Windows.Forms.Label
        Me.lblSite = New System.Windows.Forms.Label
        Me.txtSite = New System.Windows.Forms.TextBox
        Me.cmdStop = New System.Windows.Forms.Button
        Me.cmdCheck = New System.Windows.Forms.Button
        Me.lblStatus = New System.Windows.Forms.Label
        Me.lblError = New System.Windows.Forms.Label
        Me.myDataGrid = New System.Windows.Forms.DataGridView
        Me.lblAbout = New System.Windows.Forms.Label
        Me.ToolTip1 = New System.Windows.Forms.ToolTip(Me.components)
        Me.Timer2 = New System.Windows.Forms.Timer(Me.components)
        CType(Me.myDataGrid,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        '
        'cmdConnect
        '
        Me.cmdConnect.Location = New System.Drawing.Point(193, 53)
        Me.cmdConnect.Name = "cmdConnect"
        Me.cmdConnect.Size = New System.Drawing.Size(155, 23)
        Me.cmdConnect.TabIndex = 0
        Me.cmdConnect.Text = "Connect"
        Me.ToolTip1.SetToolTip(Me.cmdConnect, "Πατήστε Connect για να γίνει
η σύνδεση")
        Me.cmdConnect.UseVisualStyleBackColor = True
        '
        'cmdXML
        '
        Me.cmdXML.Location = New System.Drawing.Point(368, 122)
        Me.cmdXML.Name = "cmdXML"
        Me.cmdXML.Size = New System.Drawing.Size(75, 45)
        Me.cmdXML.TabIndex = 3
        Me.cmdXML.Text = "XML"
        Me.ToolTip1.SetToolTip(Me.cmdXML, "Πατήστε εδώ για να γίνει
χειροκίνητη αποθήκευση του πίνακα σε αρχεία XML.")
        Me.cmdXML.UseVisualStyleBackColor = True
        '
        'txtUser
        '
        Me.txtUser.Location = New System.Drawing.Point(75, 16)
        Me.txtUser.Name = "txtUser"
        Me.txtUser.Size = New System.Drawing.Size(100, 20)
        Me.txtUser.TabIndex = 4
        Me.ToolTip1.SetToolTip(Me.txtUser, "Δώστε το όνομα χρήστη για τη
βάση mySQL")
        '
        'txtPass
        '
        Me.txtPass.Location = New System.Drawing.Point(248, 17)
        Me.txtPass.Name = "txtPass"
        Me.txtPass.PasswordChar = Global.Microsoft.VisualBasic.ChrW(42)
        Me.txtPass.Size = New System.Drawing.Size(100, 20)
        Me.txtPass.TabIndex = 5
        Me.ToolTip1.SetToolTip(Me.txtPass, "Δώστε τον κωδικό χρήστη για τη
βάση mySQL")
        '
        'lblUser
        '
        Me.lblUser.AutoSize = True
        Me.lblUser.Location = New System.Drawing.Point(16, 20)
        Me.lblUser.Name = "lblUser"
```

```
        Me.lblUser.Size = New System.Drawing.Size(53, 13)
        Me.lblUser.TabIndex = 6
        Me.lblUser.Text = "username"
        '
        'lblPass
        '
        Me.lblPass.AutoSize = True
        Me.lblPass.Location = New System.Drawing.Point(190, 20)
        Me.lblPass.Name = "lblPass"
        Me.lblPass.Size = New System.Drawing.Size(52, 13)
        Me.lblPass.TabIndex = 7
        Me.lblPass.Text = "password"
        '
        'txtDB
        '
        Me.txtDB.Location = New System.Drawing.Point(75, 56)
        Me.txtDB.Name = "txtDB"
        Me.txtDB.Size = New System.Drawing.Size(100, 20)
        Me.txtDB.TabIndex = 9
        Me.ToolTip1.SetToolTip(Me.txtDB, "Δώστε την διεύθυνση IP του mySQL
server. (πχ localhost)")
        '
        'cmdStart
        '
        Me.cmdStart.Location = New System.Drawing.Point(368, 16)
        Me.cmdStart.Name = "cmdStart"
        Me.cmdStart.Size = New System.Drawing.Size(75, 45)
        Me.cmdStart.TabIndex = 10
        Me.cmdStart.Text = "Start"
        Me.ToolTip1.SetToolTip(Me.cmdStart, "Πατήστε Start για να ξεκινήσει
η αυτόματη καταγραφή")
        Me.cmdStart.UseVisualStyleBackColor = True
        '
        'lblDB
        '
        Me.lblDB.AutoSize = True
        Me.lblDB.Location = New System.Drawing.Point(18, 59)
        Me.lblDB.Name = "lblDB"
        Me.lblDB.Size = New System.Drawing.Size(51, 13)
        Me.lblDB.TabIndex = 8
        Me.lblDB.Text = "database"
        '
        'txtPath
        '
        Me.txtPath.Location = New System.Drawing.Point(75, 97)
        Me.txtPath.Margin = New System.Windows.Forms.Padding(3, 3, 0, 3)
        Me.txtPath.Name = "txtPath"
        Me.txtPath.Size = New System.Drawing.Size(250, 20)
        Me.txtPath.TabIndex = 11
        Me.ToolTip1.SetToolTip(Me.txtPath, "Δώστε τον φάκελο προορισμού για
τα αρχεία XML με τα δεδομένα")
        '
        'cmdPath
        '
        Me.cmdPath.Font = New System.Drawing.Font("Microsoft Sans Serif",
8.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(161, Byte))
        Me.cmdPath.Location = New System.Drawing.Point(325, 97)
        Me.cmdPath.Margin = New System.Windows.Forms.Padding(0, 3, 3, 3)
        Me.cmdPath.Name = "cmdPath"
        Me.cmdPath.Size = New System.Drawing.Size(24, 20)
        Me.cmdPath.TabIndex = 12
        Me.cmdPath.Text = "..."
        Me.cmdPath.TextAlign = System.Drawing.ContentAlignment.BottomCenter
        Me.ToolTip1.SetToolTip(Me.cmdPath, "Πατήστε εδώ για να επιλέξετε
φάκελο προορισμού")
        Me.cmdPath.UseVisualStyleBackColor = True
        '
```

```
        'Timer1
        '
        Me.Timer1.Interval = 60000
        '
        'lblPath
        '
        Me.lblPath.AutoSize = True
        Me.lblPath.Location = New System.Drawing.Point(32, 100)
        Me.lblPath.Name = "lblPath"
        Me.lblPath.Size = New System.Drawing.Size(37, 13)
        Me.lblPath.TabIndex = 13
        Me.lblPath.Text = "output"
        '
        'lblSite
        '
        Me.lblSite.AutoSize = True
        Me.lblSite.Location = New System.Drawing.Point(46, 140)
        Me.lblSite.Name = "lblSite"
        Me.lblSite.Size = New System.Drawing.Size(23, 13)
        Me.lblSite.TabIndex = 14
        Me.lblSite.Text = "site"
        '
        'txtSite
        '
        Me.txtSite.Location = New System.Drawing.Point(75, 137)
        Me.txtSite.Name = "txtSite"
        Me.txtSite.Size = New System.Drawing.Size(193, 20)
        Me.txtSite.TabIndex = 15
        Me.ToolTip1.SetToolTip(Me.txtSite, "Δώστε την διεύθυνση που
βρίσκονται τα δεδομένα XML. (πχ http://weather.com)")
        '
        'cmdStop
        '
        Me.cmdStop.Location = New System.Drawing.Point(368, 69)
        Me.cmdStop.Name = "cmdStop"
        Me.cmdStop.Size = New System.Drawing.Size(75, 45)
        Me.cmdStop.TabIndex = 16
        Me.cmdStop.Text = "Stop"
        Me.ToolTip1.SetToolTip(Me.cmdStop, "Πατήστε Stop για να σταματήσει η
αυτόματη καταγραφή")
        Me.cmdStop.UseVisualStyleBackColor = True
        '
        'cmdCheck
        '
        Me.cmdCheck.Location = New System.Drawing.Point(274, 135)
        Me.cmdCheck.Name = "cmdCheck"
        Me.cmdCheck.Size = New System.Drawing.Size(75, 23)
        Me.cmdCheck.TabIndex = 17
        Me.cmdCheck.Text = "Check"
        Me.ToolTip1.SetToolTip(Me.cmdCheck, "Πατήστε Check για έλεγχο των
δεδομένων")
        Me.cmdCheck.UseVisualStyleBackColor = True
        '
        'lblStatus
        '
        Me.lblStatus.AutoSize = True
        Me.lblStatus.Location = New System.Drawing.Point(34, 174)
        Me.lblStatus.Name = "lblStatus"
        Me.lblStatus.Size = New System.Drawing.Size(35, 13)
        Me.lblStatus.TabIndex = 18
        Me.lblStatus.Text = "status"
        '
        'lblError
        '
        Me.lblError.Location = New System.Drawing.Point(72, 174)
        Me.lblError.Name = "lblError"
        Me.lblError.Size = New System.Drawing.Size(276, 42)
        Me.lblError.TabIndex = 19
```

```
        Me.lblError.Text = "OK"
        Me.ToolTip1.SetToolTip(Me.lblError, "Εδώ εμφανίζονται όλα τα
διαγνωστικά μηνύματα")
        '
        'myDataGrid
        '
        Me.myDataGrid.AllowUserToAddRows = False
        Me.myDataGrid.AllowUserToDeleteRows = False
        Me.myDataGrid.AllowUserToResizeColumns = False
        Me.myDataGrid.AllowUserToResizeRows = False
        Me.myDataGrid.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize
        Me.myDataGrid.Location = New System.Drawing.Point(37, 229)
        Me.myDataGrid.Name = "myDataGrid"
        Me.myDataGrid.RowHeadersWidth = 25
        Me.myDataGrid.Size = New System.Drawing.Size(385, 199)
        Me.myDataGrid.TabIndex = 20
        Me.ToolTip1.SetToolTip(Me.myDataGrid, "Εδώ εμφανίζονται τα δεδομένα
του weather station")
        '
        'lblAbout
        '
        Me.lblAbout.AutoSize = True
        Me.lblAbout.ForeColor = System.Drawing.SystemColors.Highlight
        Me.lblAbout.Location = New System.Drawing.Point(388, 175)
        Me.lblAbout.Name = "lblAbout"
        Me.lblAbout.Size = New System.Drawing.Size(35, 13)
        Me.lblAbout.TabIndex = 22
        Me.lblAbout.Text = "About"
        '
        'ToolTip1
        '
        Me.ToolTip1.AutoPopDelay = 8000
        Me.ToolTip1.InitialDelay = 1000
        Me.ToolTip1.ReshowDelay = 100
        '
        'Timer2
        '
        Me.Timer2.Interval = 1500
        '
        'frmMain
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.ClientSize = New System.Drawing.Size(458, 444)
        Me.Controls.Add(Me.lblAbout)
        Me.Controls.Add(Me.myDataGrid)
        Me.Controls.Add(Me.lblError)
        Me.Controls.Add(Me.lblStatus)
        Me.Controls.Add(Me.cmdCheck)
        Me.Controls.Add(Me.cmdStop)
        Me.Controls.Add(Me.txtSite)
        Me.Controls.Add(Me.lblSite)
        Me.Controls.Add(Me.lblPath)
        Me.Controls.Add(Me.cmdPath)
        Me.Controls.Add(Me.txtPath)
        Me.Controls.Add(Me.cmdStart)
        Me.Controls.Add(Me.txtDB)
        Me.Controls.Add(Me.lblDB)
        Me.Controls.Add(Me.lblPass)
        Me.Controls.Add(Me.lblUser)
        Me.Controls.Add(Me.txtPass)
        Me.Controls.Add(Me.txtUser)
        Me.Controls.Add(Me.cmdXML)
        Me.Controls.Add(Me.cmdConnect)
        Me.Icon = CType(resources.GetObject("$this.Icon"),
System.Drawing.Icon)
        Me.MaximizeBox = False
```

```
        Me.MaximumSize = New System.Drawing.Size(474, 480)
        Me.MinimumSize = New System.Drawing.Size(474, 480)
        Me.Name = "frmMain"
        Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "Data Logger"
        CType(Me.myDataGrid,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
        Me.PerformLayout()

    End Sub
    Friend WithEvents cmdConnect As System.Windows.Forms.Button
    Friend WithEvents cmdXML As System.Windows.Forms.Button
    Friend WithEvents txtUser As System.Windows.Forms.TextBox
    Friend WithEvents txtPass As System.Windows.Forms.TextBox
    Friend WithEvents lblUser As System.Windows.Forms.Label
    Friend WithEvents lblPass As System.Windows.Forms.Label
    Friend WithEvents txtDB As System.Windows.Forms.TextBox
    Friend WithEvents cmdStart As System.Windows.Forms.Button
    Friend WithEvents lblDB As System.Windows.Forms.Label
    Friend WithEvents FolderBrowserDialog1 As
System.Windows.Forms.FolderBrowserDialog
    Friend WithEvents txtPath As System.Windows.Forms.TextBox
    Friend WithEvents cmdPath As System.Windows.Forms.Button
    Friend WithEvents Timer1 As System.Windows.Forms.Timer
    Friend WithEvents lblPath As System.Windows.Forms.Label
    Friend WithEvents lblSite As System.Windows.Forms.Label
    Friend WithEvents txtSite As System.Windows.Forms.TextBox
    Friend WithEvents cmdStop As System.Windows.Forms.Button
    Friend WithEvents cmdCheck As System.Windows.Forms.Button
    Friend WithEvents lblStatus As System.Windows.Forms.Label
    Friend WithEvents lblError As System.Windows.Forms.Label
    Friend WithEvents myDataGrid As System.Windows.Forms.DataGridView
    Friend WithEvents lblAbout As System.Windows.Forms.Label
    Friend WithEvents ToolTip1 As System.Windows.Forms.ToolTip
    Friend WithEvents Timer2 As System.Windows.Forms.Timer

End Class
```

AboutBox1.vb

```
Public NotInheritable Class frmAboutBox

    Private Sub AboutBox1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        Me.LabelProductName.Text = My.Application.Info.ProductName
        Me.LabelVersion.Text = String.Format("Version {0}",
My.Application.Info.Version.ToString)
        Me.LabelCopyright.Text = My.Application.Info.Copyright
        Me.LabelCompanyName.Text = My.Application.Info.Description
    End Sub

    Private Sub OKButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OKButton.Click
        Me.Close()
    End Sub

End Class
```

AboutBox1.Designer.vb

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class frmAboutBox
    Inherits System.Windows.Forms.Form
```

```
    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    Friend WithEvents TableLayoutPanel As
System.Windows.Forms.TableLayoutPanel
    Friend WithEvents LogoPictureBox As System.Windows.Forms.PictureBox
    Friend WithEvents LabelProductName As System.Windows.Forms.Label
    Friend WithEvents LabelVersion As System.Windows.Forms.Label
    Friend WithEvents LabelCompanyName As System.Windows.Forms.Label
    Friend WithEvents TextBoxDescription As System.Windows.Forms.TextBox
    Friend WithEvents OKButton As System.Windows.Forms.Button
    Friend WithEvents LabelCopyright As System.Windows.Forms.Label

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Dim resources As System.ComponentModel.ComponentResourceManager =
New System.ComponentModel.ComponentResourceManager(GetType(frmAboutBox))
        Me.TableLayoutPanel = New System.Windows.Forms.TableLayoutPanel
        Me.LogoPictureBox = New System.Windows.Forms.PictureBox
        Me.LabelProductName = New System.Windows.Forms.Label
        Me.LabelVersion = New System.Windows.Forms.Label
        Me.LabelCopyright = New System.Windows.Forms.Label
        Me.LabelCompanyName = New System.Windows.Forms.Label
        Me.TextBoxDescription = New System.Windows.Forms.TextBox
        Me.OKButton = New System.Windows.Forms.Button
        Me.TableLayoutPanel.SuspendLayout()
        CType(Me.LogoPictureBox,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        '
        'TableLayoutPanel
        '
        Me.TableLayoutPanel.ColumnCount = 2
        Me.TableLayoutPanel.ColumnStyles.Add(New
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
33.0!))
        Me.TableLayoutPanel.ColumnStyles.Add(New
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
67.0!))
        Me.TableLayoutPanel.Controls.Add(Me.LogoPictureBox, 0, 0)
        Me.TableLayoutPanel.Controls.Add(Me.LabelProductName, 1, 0)
        Me.TableLayoutPanel.Controls.Add(Me.LabelVersion, 1, 1)
        Me.TableLayoutPanel.Controls.Add(Me.LabelCopyright, 1, 2)
        Me.TableLayoutPanel.Controls.Add(Me.LabelCompanyName, 1, 3)
        Me.TableLayoutPanel.Controls.Add(Me.TextBoxDescription, 1, 4)
        Me.TableLayoutPanel.Controls.Add(Me.OKButton, 1, 5)
        Me.TableLayoutPanel.Dock = System.Windows.Forms.DockStyle.Fill
        Me.TableLayoutPanel.Location = New System.Drawing.Point(9, 9)
        Me.TableLayoutPanel.Name = "TableLayoutPanel"
        Me.TableLayoutPanel.RowCount = 6
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 10.0!))
```

```
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 10.0!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 10.0!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 10.0!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 50.0!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 10.0!))
        Me.TableLayoutPanel.Size = New System.Drawing.Size(396, 258)
        Me.TableLayoutPanel.TabIndex = 0
        '
        'LogoPictureBox
        '
        Me.LogoPictureBox.Dock = System.Windows.Forms.DockStyle.Fill
        Me.LogoPictureBox.Image =
CType(resources.GetObject("LogoPictureBox.Image"), System.Drawing.Image)
        Me.LogoPictureBox.Location = New System.Drawing.Point(3, 3)
        Me.LogoPictureBox.Name = "LogoPictureBox"
        Me.TableLayoutPanel.SetRowSpan(Me.LogoPictureBox, 6)
        Me.LogoPictureBox.Size = New System.Drawing.Size(124, 252)
        Me.LogoPictureBox.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.LogoPictureBox.TabIndex = 0
        Me.LogoPictureBox.TabStop = False
        '
        'LabelProductName
        '
        Me.LabelProductName.Dock = System.Windows.Forms.DockStyle.Fill
        Me.LabelProductName.Font = New System.Drawing.Font("Cambria",
12.75!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(161, Byte))
        Me.LabelProductName.ForeColor =
System.Drawing.Color.FromArgb(CType(CType(128, Byte), Integer),
CType(CType(64, Byte), Integer), CType(CType(0, Byte), Integer))
        Me.LabelProductName.Location = New System.Drawing.Point(136, 0)
        Me.LabelProductName.Margin = New System.Windows.Forms.Padding(6, 0,
3, 0)
        Me.LabelProductName.MaximumSize = New System.Drawing.Size(0, 20)
        Me.LabelProductName.Name = "LabelProductName"
        Me.LabelProductName.Size = New System.Drawing.Size(257, 20)
        Me.LabelProductName.TabIndex = 0
        Me.LabelProductName.Text = "Product Name"
        Me.LabelProductName.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft
        '
        'LabelVersion
        '
        Me.LabelVersion.Dock = System.Windows.Forms.DockStyle.Fill
        Me.LabelVersion.Font = New System.Drawing.Font("Cambria", 9.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(161, Byte))
        Me.LabelVersion.Location = New System.Drawing.Point(136, 25)
        Me.LabelVersion.Margin = New System.Windows.Forms.Padding(6, 0, 3,
0)
        Me.LabelVersion.MaximumSize = New System.Drawing.Size(0, 17)
        Me.LabelVersion.Name = "LabelVersion"
        Me.LabelVersion.Size = New System.Drawing.Size(257, 17)
        Me.LabelVersion.TabIndex = 0
        Me.LabelVersion.Text = "Version"
        Me.LabelVersion.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft
        '
        'LabelCopyright
        '
        Me.LabelCopyright.Dock = System.Windows.Forms.DockStyle.Fill
```

```
        Me.LabelCopyright.Font = New System.Drawing.Font("Cambria", 9.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(161, Byte))
        Me.LabelCopyright.Location = New System.Drawing.Point(136, 50)
        Me.LabelCopyright.Margin = New System.Windows.Forms.Padding(6, 0, 3,
0)
        Me.LabelCopyright.MaximumSize = New System.Drawing.Size(0, 17)
        Me.LabelCopyright.Name = "LabelCopyright"
        Me.LabelCopyright.Size = New System.Drawing.Size(257, 17)
        Me.LabelCopyright.TabIndex = 0
        Me.LabelCopyright.Text = "Copyright"
        Me.LabelCopyright.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft
        '
        'LabelCompanyName
        '
        Me.LabelCompanyName.Dock = System.Windows.Forms.DockStyle.Fill
        Me.LabelCompanyName.Font = New System.Drawing.Font("Cambria", 9.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(161, Byte))
        Me.LabelCompanyName.Location = New System.Drawing.Point(136, 75)
        Me.LabelCompanyName.Margin = New System.Windows.Forms.Padding(6, 0,
3, 0)
        Me.LabelCompanyName.MaximumSize = New System.Drawing.Size(0, 17)
        Me.LabelCompanyName.Name = "LabelCompanyName"
        Me.LabelCompanyName.Size = New System.Drawing.Size(257, 17)
        Me.LabelCompanyName.TabIndex = 0
        Me.LabelCompanyName.Text = "Company Name"
        Me.LabelCompanyName.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft
        '
        'TextBoxDescription
        '
        Me.TextBoxDescription.Dock = System.Windows.Forms.DockStyle.Fill
        Me.TextBoxDescription.Font = New System.Drawing.Font("Cambria",
9.75!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(161, Byte))
        Me.TextBoxDescription.Location = New System.Drawing.Point(136, 103)
        Me.TextBoxDescription.Margin = New System.Windows.Forms.Padding(6,
3, 3, 3)
        Me.TextBoxDescription.Multiline = True
        Me.TextBoxDescription.Name = "TextBoxDescription"
        Me.TextBoxDescription.ReadOnly = True
        Me.TextBoxDescription.ScrollBars =
System.Windows.Forms.ScrollBars.Both
        Me.TextBoxDescription.Size = New System.Drawing.Size(257, 123)
        Me.TextBoxDescription.TabIndex = 0
        Me.TextBoxDescription.TabStop = False
        Me.TextBoxDescription.Text =
resources.GetString("TextBoxDescription.Text")
        '
        'OKButton
        '
        Me.OKButton.Anchor = CType((System.Windows.Forms.AnchorStyles.Bottom
Or System.Windows.Forms.AnchorStyles.Right),
System.Windows.Forms.AnchorStyles)
        Me.OKButton.DialogResult = System.Windows.Forms.DialogResult.Cancel
        Me.OKButton.Location = New System.Drawing.Point(318, 232)
        Me.OKButton.Name = "OKButton"
        Me.OKButton.Size = New System.Drawing.Size(75, 23)
        Me.OKButton.TabIndex = 0
        Me.OKButton.Text = "&OK"
        '
        'frmAboutBox
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.CancelButton = Me.OKButton
```

```
        Me.ClientSize = New System.Drawing.Size(414, 276)
        Me.Controls.Add(Me.TableLayoutPanel)
        Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedDialog
        Me.Icon = CType(resources.GetObject("$this.Icon"),
System.Drawing.Icon)
        Me.MaximizeBox = False
        Me.MinimizeBox = False
        Me.Name = "frmAboutBox"
        Me.Padding = New System.Windows.Forms.Padding(9)
        Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterParent
        Me.Text = "About"
        Me.TableLayoutPanel.ResumeLayout(False)
        Me.TableLayoutPanel.PerformLayout()
        CType(Me.LogoPictureBox,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)

    End Sub

End Class
```

Ο κώδικας MySQL

Κατασκευή πίνακα

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

CREATE TABLE webdata (
  `time` timestamp NOT NULL default CURRENT_TIMESTAMP on update
CURRENT_TIMESTAMP,
  temp decimal(3,1) default NULL,
  humi tinyint(3) default NULL,
  dew decimal(3,1) default NULL,
  vsol decimal(3,1) default NULL,
  vbat decimal(3,1) default NULL,
  amp smallint(3) default NULL,
  pres decimal(5,1) default NULL,
  sea decimal(5,1) default NULL,
  PRIMARY KEY  (`time`)
) ENGINE=MyISAM DEFAULT CHARSET=ascii;
```

Βιβλιογραφία

[1] **Barr, Michael.** Embedded Systems Glossary. [Ηλεκτρονικό] http://www.netrino.com/Embedded-Systems/Glossary.

[2] **Howe, Walt.** A Brief History of the Internet. [Ηλεκτρονικό] http://www.walthowe.com/navnet/history.html.

[3] Netcraft Web Server Survey. [Ηλεκτρονικό] http://survey.netcraft.com/Reports/200804/index.html.

[4] Range of a typical Wi-Fi access point. [Ηλεκτρονικό] http://help.unc.edu/6009.

[5] Atmel AVR mega168 datasheet. [Ηλεκτρονικό] http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf.

[6] Microchip ENC28J60 Ethernet controller datasheet. [Ηλεκτρονικό] http://ww1.microchip.com/downloads/en/DeviceDoc/39662b.pdf.

[7] **Pryor, Steve.** 1000BASE-T Basics and Startup. [Ηλεκτρονικό] http://grouper.ieee.org/groups/802/3/minutes/july98/E2_0798.pdf.

[8] **Lenardic, Denis.** History of photovoltaics. [Ηλεκτρονικό] http://www.pvresources.com/en/history.php.

[9] Solar Cells. [Ηλεκτρονικό] http://www.4offsets.com/solar-cells.php.

[10] CSB 12Volt 7.2A SLA Battery datasheet. [Ηλεκτρονικό] http://www.csb-battery.com/upfiles/dow01204794623.pdf.

[11] Sealed Lead Acid Battery Charging Basics. [Ηλεκτρονικό] http://www.powerstream.com/SLA.htm.

[12] **Buchmann, Isidor.** Battery performance as a function of cycling. [Ηλεκτρονικό] http://www.batteryuniversity.com/parttwo-36.htm.

[13] Overcharging Batteries. [Ηλεκτρονικό] http://bugclub.org/beginners/batteries/overcharge.html.

[14] **Taylor, Robert.** Lead-Acid battery hazards. [Ηλεκτρονικό] http://www.calicorp.com/articles/batteries-hazards.html.

[15] **Ledwich, G.** DC-DC Converter Basics. [Ηλεκτρονικό] http://powerdesigners.com/InfoWeb/design_center/articles/DC-DC/converter.shtm.

[16] Texas Instruments MC34063A switching regulator datasheet. [Ηλεκτρονικό] http://focus.ti.com/lit/ds/symlink/mc34063a.pdf.

[17] Understanding SAR ADCs. [Ηλεκτρονικό] http://www.maxim-ic.com/appnotes.cfm/appnote_number/1080/CMP/WP-50.

[18] Successive approximation ADC. [Ηλεκτρονικό] http://www.allaboutcircuits.com/vol_4/chpt_13/6.html.

[19] Freescale MPX4115A Pressure sensor datasheet. [Ηλεκτρονικό] http://www.freescale.com/files/sensors/doc/data_sheet/MPX4115A.pdf.

[20] Microcontroller Voltmeter / Ammeter with LCD. [Ηλεκτρονικό] http://electronics-diy.com/AVR_LCD_Voltmeter_Ammeter.php.

[21] Understanding Ground Loops. [Ηλεκτρονικό] http://www.bapihvac.com/CatalogPDFs/I_App_Notes/Understanding_Ground_Loops.pdf.

[22] Avoiding Ground Loops. [Ηλεκτρονικό] http://www.bapihvac.com/CatalogPDFs/I_App_Notes/Avoiding_Ground_Loops.pdf.

[23] Sensistion SHT1x Humidity and Temperature Sensor datasheet. [Ηλεκτρονικό] http://www.sensirion.com/en/pdf/product_information/Datasheet-humidity-sensor-SHT1x.pdf.

[24] SDEC LMC-SSC2A20DLYY Dot Matrix LCD Module datasheet. [Ηλεκτρονικό] http://www.100y.com.tw/pdf_file/LMC-SSC2A20DLYY.PDF.

[25] Article about Roaming. [Ηλεκτρονικό] http://en.wikipedia.org/w/index.php?title=Roaming&oldid=236502310.

[26] Charging Batteries from USB. [Ηλεκτρονικό] http://www.maxim-ic.com/appnotes.cfm/appnote_number/3607.

[27] D-Link DWL-G730AP product manual (compressed). [Ηλεκτρονικό] ftp://ftp.dlink.com/Wireless/dwlg730AP/Manual/dwlg730AP_manual_06302005.zip.

[28] Fahrenheit and Celsius Conversion Formulas. [Ηλεκτρονικό] http://www.albireo.ch/temperatureconverter/formula.htm.

[29] The Barometric Formula. [Ηλεκτρονικό] http://hyperphysics.phy-astr.gsu.edu/hbase/kinetic/barfor.html.

[30] UDP and TCP Checksums. [Ηλεκτρονικό] http://www.msc.uky.edu/ken/cs471/notes/chap3.htm.

[31] RFC1071 - Computing the Internet checksum. [Ηλεκτρονικό] http://www.faqs.org/rfcs/rfc1071.html.

[32] Short description of the Internet checksum. [Ηλεκτρονικό] http://www.netfor2.com/checksum.html.

[33] RFC793 - Transmission Control Protocol. [Ηλεκτρονικό] http://www.faqs.org/rfcs/rfc793.html.

[34] Explanation of the Three-Way Handshake via TCP/IP. [Ηλεκτρονικό] http://support.microsoft.com/kb/172983.

[35] CRC computation. [Ηλεκτρονικό] http://www.atmel.com/dyn/resources/prod_documents/doc2579.pdf.

[36] Sensirion SHTxx Non-Linearity compensation. [Ηλεκτρονικό] http://www.sensirion.com/pdf/product_information/Non-Linearity_Compensation_Humidity_Sensors_E.pdf.

[37] Sensirion SHTxx Dew-point Calculation. [Ηλεκτρονικό] http://www.sensirion.com/pdf/product_information/Dewpoint_Calculation_Humidity_Sensor_E.pdf.

[38] Theory and Applications of the MC34063 Switching Regulator Control Circuits. [Ηλεκτρονικό] http://www.onsemi.com/pub_link/Collateral/AN920-D.PDF.

[39] **Aydin, Dincer.** LCD commands. [Ηλεκτρονικό] http://www.geocities.com/dinceraydin/lcd/commands.htm.

[40] Microcontroller Tutorial - LCD HD44780. [Ηλεκτρονικό] http://web.tampabay.rr.com/hazer/hd44780_part_1.htm.

[41] HD44780 instruction set. [Ηλεκτρονικό] http://www.mil.ufl.edu/4744/docs/lcdmanual/commands.html.

[42] AVR Hardware Design Considerations. [Ηλεκτρονικό] http://www.atmel.com/dyn/resources/prod_documents/doc2521.pdf.

[43] Sensirion SHTxx Relative Humidity Calculation - Application Note. [Ηλεκτρονικό] http://www.sensirion.com/en/pdf/product_information/Relative_Humidity_Calculation_E.pdf.

[44] Sensirion SHT1x Soldering Procedure - Application Note. [Ηλεκτρονικό] http://www.sensirion.com/images/getFile?id=65.

[45] Sensirion SHTxx ESD, Latch-Up and EMC application note. [Ηλεκτρονικό] http://www.sensirion.com/pdf/product_information/ESD_Latch-up_and_EMC_E.pdf.

[46] SMT Power Inductors – DO5022P Series datasheet. [Ηλεκτρονικό] http://www.coilcraft.com/pdfs/do5022p.pdf.

[47] Simple solar charger circuit. [Ηλεκτρονικό] http://www.talkingelectronics.com/projects/SolarCharger/SolarCharger.html.

[48] **Uijl, Oscar den.** Solar charger for lead-acid batteries. [Ηλεκτρονικό] http://www.electronics-lab.com/projects/power/031/.

[49] Solar Nicad charger. [Ηλεκτρονικό] http://www.uoguelph.ca/~antoon/gadgets/solar1g.htm.

[50] Nicad Battery Charger. [Ηλεκτρονικό] http://www.mitedu.freeserve.co.uk/Circuits/Power/nicad.htm.

[51] **Hayles, Peter.** Intelligent NiCd/NiMH Battery Charger. [Ηλεκτρονικό] http://www.angelfire.com/electronic/hayles/charge1.html.

[52] Battery Voltage and Current - Tech Note. [Ηλεκτρονικό] http://www.xantrex.com/web/id/329/DocServe.aspx.

[53] Sealed Lead Acid batteries - Tech Manual. [Ηλεκτρονικό] http://www.jlktechnicalsales.com/downloads/power-sonic/techman.pdf.

[54] Internet Protocol header description. [Ηλεκτρονικό] http://www.networksorcery.com/enp/protocol/ip.htm.

[55] **White, Fredric.** webACE Server - World's Smallest Web Server. [Ηλεκτρονικό] http://d116.com/ace/.

[56] **Dunkels, Adam.** The uIP Embedded TCP/IP Stack. [Ηλεκτρονικό] http://www.sics.se/~adam/download/uip-1.0-refman.pdf.

[57] **Stevens, John και Corey, Garth.** A Study of Lead-Acid Battery Efficiency Near Top-of-Charge and the Impact on PV System Design. [Ηλεκτρονικό] http://www.localenergy.org/pdfs/Document%20Library/Lead%20Acid%20Battery%20Efficiency.pdf.

Βιβλία

[58] **Hoenig, Stuart.** *How to build and use electronic devices without frustration, panic, mountains of money, or an engineering degree.* Boston : Little, Brown and Company (Inc.), 1980. ISBN 0-316-36808-3.

[59] **Μανιάς, Στέφανος και Κολετσάνος, Αθανάσιος.** *Βιομηχανικά Ηλεκτρονικά.* Αθήνα : Εκδόσεις Συμεών, 2001. ISBN 960-7888-25-1.

[60] **Sedra, Adel και Smith, Kenneth.** *Μικροηλεκτρονικά Κυκλώματα.* Αθήνα : Παπασωτηρίου, 1994. ISBN 960-85334-5-7.

[61] **Kurose, James και Ross, Keith.** *Δικτύωση Υπολογιστών.* Αθήνα : Εκδόσεις Μ. Γκιούργας, 2004. ISBN 960-512-387-8.

[62] **Silberschatz, Abraham και Korth, Henry.** *Συστήματα Βάσεων Δεδομένων.* Αθήνα : Εκδόσεις Μ. Γκιούρδας, 2004. ISBN 960-512-384-3.

[63] **Tanenbaum, Andrew.** *Δίκτυα Υπολογιστών.* Αθήνα : Παπασωτηρίου, 2000. ISBN 960-7510-70-4.

[64] **Kernighan, Brian και Ritchie, Dennis.** *Η Γλώσσα Προγραμματισμού C.* Αθήνα : Κλειδάριθμος, 1990. ISBN 960-209-053-7.

[65] **Πεκμεστζή, Κιαμάλ.** *Συστήματα Μικροϋπολογιστών.* Αθήνα : Εκδόσεις Συμμετρία, 1995.

[66] **Πεκμεστζή, Κιαμάλ.** *Μικροελεγκτές AVR και PIC.* Αθήνα : Εκδόσεις ΕΜΠ, 2007.