Τεχνολογικό
Εκπαιδευτικό
Ίδρυμα Κρήτης
Παράρτημα Χανίων
Τμήμα Ηλεκτρονικής

Φορητό σύστημα συλλογής δεδομένων από πολλούς σταθμούς

TEMP CONTROL SYSTEM

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Πέντες Γιώργος

Επιβλέπων: Ρηγάκης Ηρακλής
Εργαστηριακός συνεργάτης

Χανιά 2012

ΠΕΡΙΛΗΨΗ

Η ανάπτυξη της τεχνολογίας έχει δημιουργήσει την ανάγκη για ακόμα πιο «έξυπνα», αυτόνομα και ευκολότερα στη χρήση ηλεκτρονικά συστήματα. Θέλοντας ένας μηχανικός να υλοποιήσει ένα δικό του σύστημα, θα πρέπει να ακολουθήσει την ανάπτυξη αυτή, να μελετήσει και να ασχοληθεί με όλες αυτές τις δυνατότητες που του δίνει η σημερινή τεχνολογία ώστε το σύστημα που θα υλοποιήσει να είναι καινοτόμο και ανταγωνιστικότερο όλων των υπολοίπων ομοειδών συστημάτων του.

Ένα από τα βασικότερα χαρακτηριστικά που χρειάζεται πολλές φορές και για διάφορους λόγους να έχει ένα ηλεκτρονικό σύστημα είναι αυτό της ασύρματης επικοινωνίας αλλά και της επικοινωνίας με έναν ηλεκτρονικό υπολογιστή.

Στα πλαίσια της πτυχιακής αυτής, έγινε η θεωρητική μελέτη ενός μικροελεγκτή (μC), ο μικροελεγκτής είναι ένα στοιχείο που χρησιμοποιείται ευρέως στα ηλεκτρονικά συστήματα, ενός ολοκληρωμένου κυκλώματος (IC) για ασύρματες επικοινωνίες καθώς και η μελέτη αυτών στη πράξη μέσω της ανάπτυξης ενός συστήματος. Το σύστημα αυτό έχει την δυνατότητα να κάνει καταγραφή θερμοκρασίας μέσω ενός αισθητήρα και να στέλνει τις μετρήσεις στο δέκτη του συστήματος. Ο δέκτης με την σειρά του στέλνει μέσω της σειριακής θύρας του υπολογιστή τις μετρήσεις θερμοκρασίας και ο υπολογιστής έπειτα μέσω του κατάλληλου λογισμικού που αναπτύχθηκε αναπαριστά και αποθηκεύει τις μετρήσεις αυτές.

Έτσι μπορεί να δημιουργηθεί ένα ασύρματο δίκτυο ανταλλαγής δεδομένων που να είναι αυτόνομο, να μην επιβαρύνει υπάρχοντα δίκτυα (τοπικά δίκτυα υπολογιστών, ασύρματα δίκτυα) και να μπορεί να παραμετροποιηθεί ανάλογα με τις απαιτήσεις του εκάστοτε δικτύου.

Τέλος θέλοντας να πάρει η πτυχιακή αυτή ένα χαρακτήρα ενός σχεδόν πραγματικού συστήματος, πραγματοποιήθηκε έτσι ώστε να χρησιμοποιηθεί για τις ανάγκες θερμομέτρησης ασθενών που έχει ένα νοσοκομείο.

Λέξεις κλειδιά: Ηλεκτρονικό σύστημα, Μικροελεγκτής (μC), ασύρματη επικοινωνία, μέτρηση θερμοκρασίας.

Technological
Educational
Institute of Crete
Branch of Chania
Department of electronics

Portable system of data collection from many stations

Thesis of:
Pentes George

Supervisor professor:
Rigakis Iraklis

Chania 2012

RESUME

The growth of technology has created a need for more "intelligent", autonomous and easier to use electronic systems. If an engineer wants to develop his own system, he should follow this growth. He has to study and deal with all these features that give today's technology so the system will be developed innovative and more competitive to the other similar.

One of the key features needed by an electronic system is that of the wireless communication and communication with a computer.

A part of this thesis was the theoretical study of a microcontroller (uC), the microcontroller is an element commonly used in electronics, an integrated circuit (IC) for wireless communications and the study of these in practice through the development of a system. This system has the ability to make temperature recording through a sensor and then to send the measurements to the receiver system. The receiver sends temperature measurements via the serial port of the computer and the computer then through the appropriate software developed represents and stores these measurements.

This can create a wireless network for data exchange that is autonomous, not to overload existing networks (local computer networks LANs, wireless networks) and can be customized depending on the requirements of the network.

Finally, to give this thesis almost a character of a real system, it was used for the needs of patients' thermometrisis in a hospital.

Keywords: Electronic system, microcontroller (μC), wireless communication, temperature measuring.

ΕΥΧΑΡΙΣΤΙΕΣ

Είμαι ευγνώμων πρώτα από όλα στην οικογένεια μου για την ηθική και οικονομική στήριξη που μου παρείχε κατά την διάρκεια των σπουδών μου. Καθώς επίσης και στους συμφοιτητές φίλους που μου στάθηκαν δίπλα μου σε κάθε δυσκολία των φοιτητικών μου χρόνων.

Θερμά ευχαριστώ τον εισηγητή κι επιβλέποντα καθηγητή της πτυχιακής μου εργασίας, κ. Ριγάκη Ηρακλή, ο οποίος με τις υποδείξεις του, τις εύστοχες παρατηρήσεις του και τη συνεχή και αμέριστη υποστήριξή του, με βοήθησε να πραγματοποιήσω την παρούσα πτυχιακή εργασία.

Τέλος θα ήθελα να ευχαριστήσω τους καθηγητές του Τμήματος Ηλεκτρονικής του Τ.Ε.Ι. Κρήτης, για τις πολύ σημαντικές και εποικοδομητικές γνώσεις που μας προσέφεραν σε ολόκληρη την πορεία προς το πτυχίο.

Χανιά, Ιούλιος 2012
Πέντες Γιώργος

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ

Ένα από τα μεγαλύτερα επιτεύγματα του ανθρώπου είναι η επιμήκυνση του προσδόκιμου επιβίωσης το οποίο εν πολλοίς επιτεύχθηκε χάρη στην τεχνολογία. Όμως αποτελεί αίτημα των σύγχρονων οικονομικών δομών η επιδίωξη ολοένα καλύτερων υπηρεσιών περίθαλψης να συνδυάζονται με την εξοικονόμηση πόρων που θα διατεθούν σε άλλες κοινωνικές ανάγκες.

Σε κλινήρεις ασθενείς συνανθρώπους μας, απαιτείται πολλές φορές καθημερινά, η μέτρηση της θερμοκρασίας του σώματος τους. Για το σκοπό αυτό αναπτύχθηκε το σύστημα μέτρησης θερμοκρασίας "Temp Control System" το οποίο προσφέρει συνεχή ροή αυτής της πολύτιμης για την νοσηλεία πληροφορίας, χωρίς την παρουσία νοσηλευτή και χωρίς να απαιτείται χειρόγραφη καταγραφή των μετρήσεων.

Η συσκευή "Sensor" τοποθετείται στη μασχάλη του ασθενή με τη βοήθεια μίας ειδικής θήκης. Ασύρματα η συσκευή μεταδίδει μετρήσεις σε μία συσκευή – "Station" - συνδεδεμένη με υπολογιστή. Μία εφαρμογή λογισμικού παρακολουθεί και επεξεργάζεται τις τιμές αποδίδοντας διάφορα αποτελέσματα όπως: τις θερμοκρασίες του κάθε ασθενή, σήμανση συναγερμού πάνω από κάποια τιμή, καταγραφή και αναφορά δεδομένων για μελέτη ιστορικού.

Το Temp Control System μπορεί να βρει εφαρμογή σε φορείς παροχής υγειονομικής περίθαλψης όπως είναι τα νοσοκομεία και οι κάθε είδους κλινικές, γηροκομεία ή άλλα εξειδικευμένα ιδρύματα. Ακόμη στην κατ' οίκον νοσηλεία και σε υπερήλικες που ζουν μόνοι τους στο σπίτι.

Πρόκειται για ένα σύστημα, ο στόχος του οποίου έγκειται στο να προάγει και να καθιερώσει μία περισσότερο ευέλικτη καθημερινότητα για τους μελλοντικούς χρήστες αυτού αλλά και περισσότερη ασφάλεια στα άτομα τα οποία θα παρακολουθούνται.

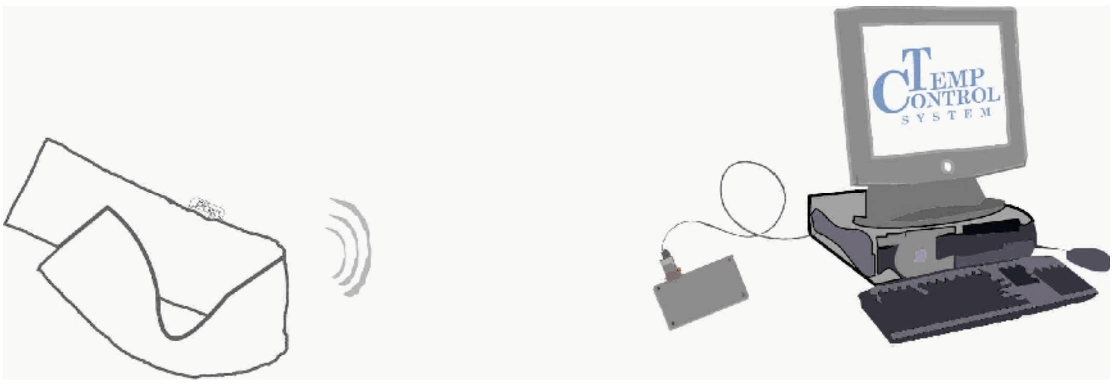
Στα κεφάλαια που ακολουθούν, περιγράφονται με τη σειρά το υλικό μέρος του συστήματος, ο προγραμματισμός του, η λειτουργία του και κάποιες προτάσεις για την περαιτέρω βελτίωση κι αξιοποίησή του.

ΚΕΦΑΛΑΙΟ 1

Το υλικό μέρος του συστήματος

1.1 Η βασική λειτουργία του συστήματος

Ο αισθητήρας θερμοκρασίας στην συσκευή “Sensor” λαμβάνει τις μετρήσεις από την μασχάλη του ασθενούς και τις στέλνει ασύρματα στην συσκευή “Station”. Στη συνέχεια, η συσκευή Station η οποία είναι συνδεδεμένη με τον ηλεκτρονικό υπολογιστή, μέσω του κατάλληλου λογισμικού, ενημερώνει με τις λαμβανόμενες τιμές την βάση δεδομένων των ασθενών. Ο χρήστης του συστήματος έχει την δυνατότητα να παραμετροποιήσει στην εφαρμογή διάφορα στοιχεία όπως, τα στοιχεία των θαλάμων του νοσοκομείου, τα στοιχεία των ασθενών που βρίσκονται σε κάθε θάλαμο, τα όρια που το σύστημα θα εκπέμψει ηχητική ειδοποίηση. Στην πλήρη εξέλιξη του το σύστημα θα μπορεί να συνδεθεί με συστήματα τηλεειδοποίησης μέσω διαδικτύου.



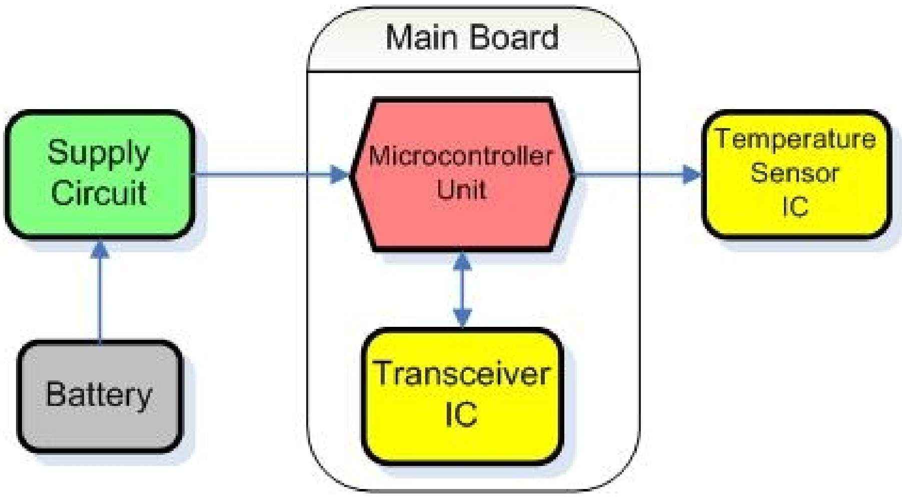
Σχήμα 1.1.1: Σκίτσο τρόπου λειτουργίας της συσκευής

Το σύστημα είναι σχεδιασμένο με τρόπο τέτοιο, ώστε να έχει πολύ χαμηλή κατανάλωση και να μπορεί να τροφοδοτηθεί ακόμα και από μικρή μπαταρία.

Η ασύρματη επικοινωνία πραγματοποιείται με ένα πομποδέκτη στην περιοχή των 2.4Ghz. Η περιοχή αυτή συχνοτήτων είναι ελεύθερα διαθέσιμη στις επικοινωνίες. Χρησιμοποιείται κυρίως για βιομηχανικούς, επιστημονικούς και ιατρικούς σκοπούς (ISM radio band, Industrial Scientific and Medicine).

Στο κομμάτι του λογισμικού, χρησιμοποιώντας την γλώσσα C# καθώς και την γλώσσα SQL υλοποιήθηκε ένα πρόγραμμα για να αναπαριστά και να αποθηκεύει τις μετρήσεις θερμοκρασίας. Το πρόγραμμα αυτό για την λήψη των θερμοκρασιών, επικοινωνεί μέσω της σειριακής θύρας του υπολογιστή με την συσκευή Station.

1.2 Το κύκλωμα της συσκευής «Sensor»

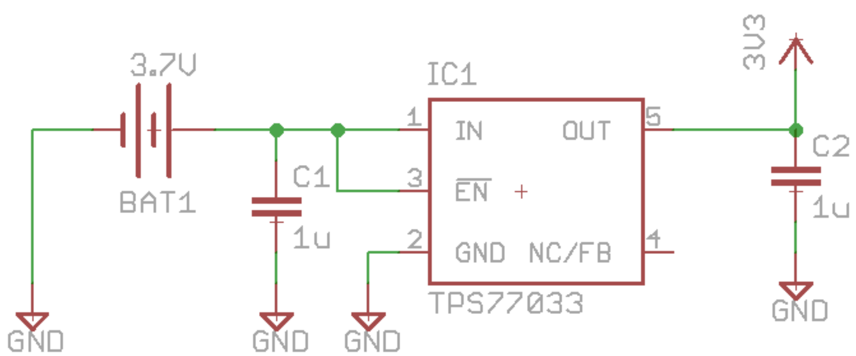


Σχήμα 1.2.1: Το διάγραμμα βαθμίδων της συσκευής Sensor.

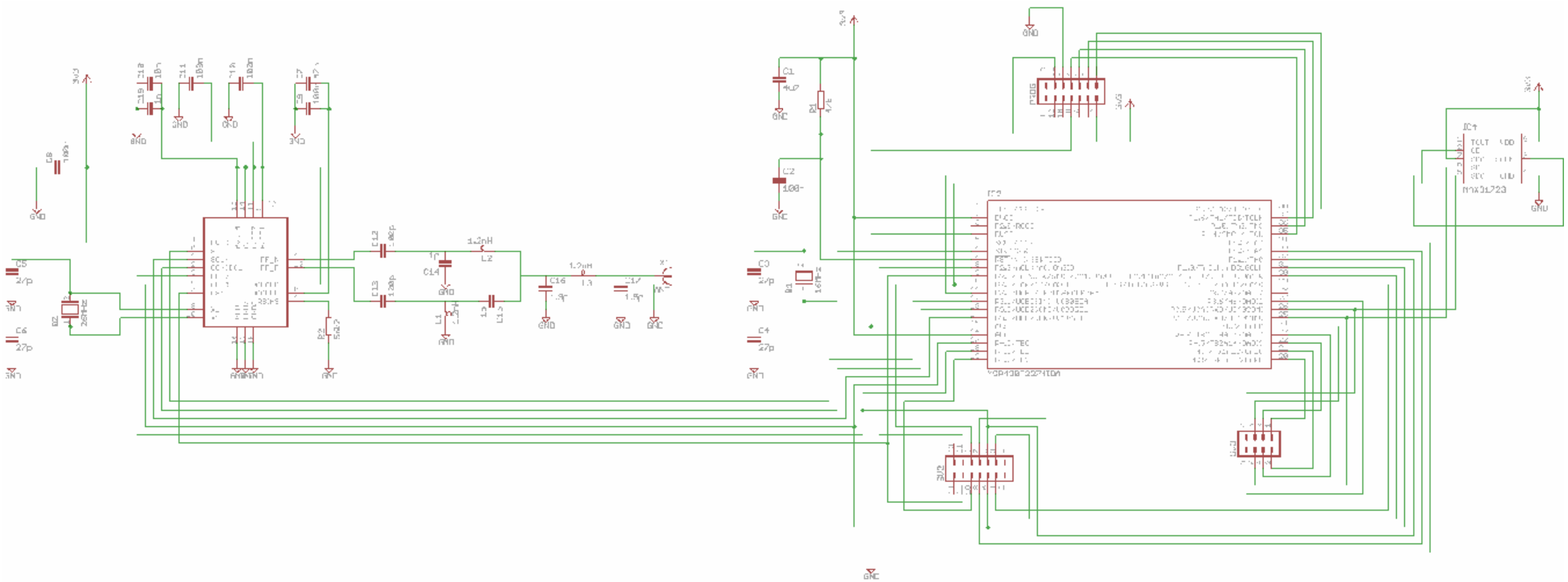
Το ηλεκτρονικό κύκλωμα της συσκευής Sensor αποτελείται από:

1. Μια τροφοδοτική διάταξη με μπαταρία.
2. Μια βαθμίδα ελέγχου με μικροελεγκτή (MSP430F2274).
3. Ένα ολοκληρωμένο κύκλωμα για αποστολή και λήψη δεδομένων στην ζώνη συχνοτήτων ISM 2,4GHz.
4. Ένα ολοκληρωμένο κύκλωμα για μέτρηση θερμοκρασίας.

Το κυκλωματικό διάγραμμα της συσκευής Sensor φαίνεται στο Σχ.1.2.2 και Σχ.1.2.3. Οι επιμέρους βαθμίδες της αναλύονται στις παραγράφους που ακολουθούν.

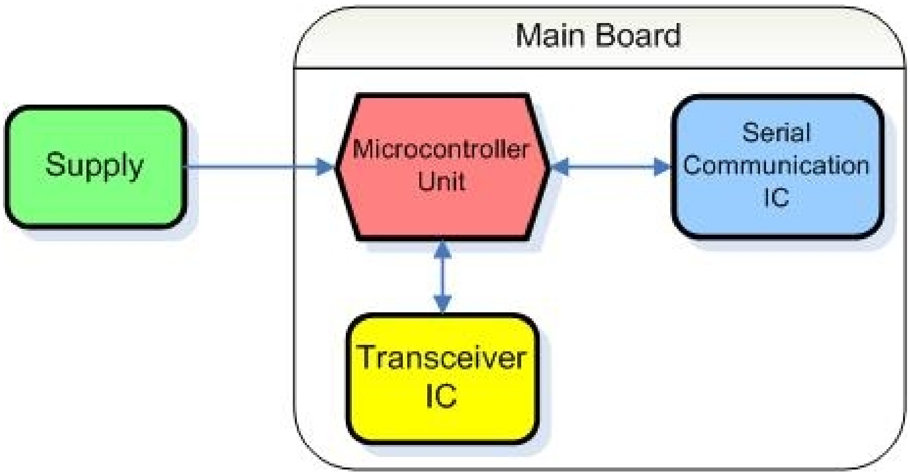


Σχήμα 1.2.2: Κυκλωματικό διάγραμμα τροφοδοσίας.



Σχήμα 1.2.3: Κυκλωματικό διάγραμμα της συσκευής Sensor.

1.3 Το κύκλωμα της συσκευής «Station»

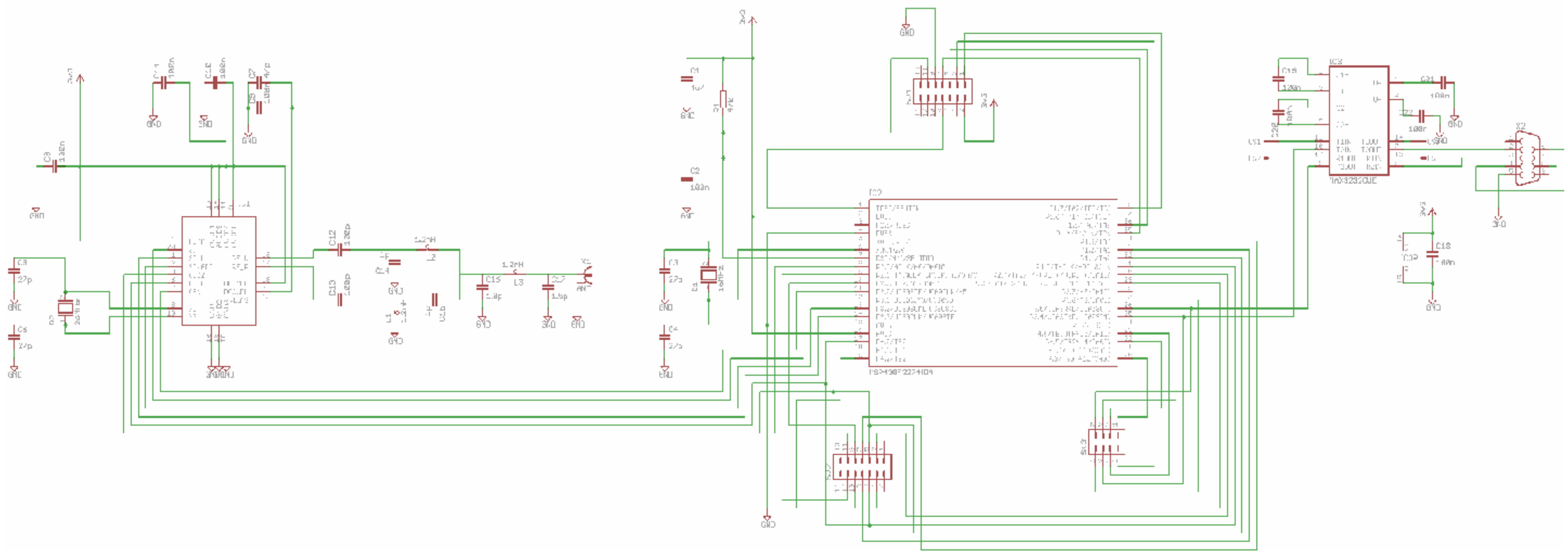


Σχήμα 1.3.1: Το διάγραμμα βαθμίδων της συσκευής Station.

Το ηλεκτρονικό κύκλωμα της συσκευής Station αποτελείται από:

1. Μια βαθμίδα ελέγχου με μικροελεγκτή (MSP430F2274).
2. Ένα ολοκληρωμένο κύκλωμα για αποστολή και λήψη δεδομένων στην ζώνη συχνοτήτων ISM 2,4GHz.
3. Ένα ολοκληρωμένο κύκλωμα για επικοινωνία μέσω σειριακής (RS-232) με υπολογιστή.

Το κυκλωματικό διάγραμμα της συσκευής Sensor φαίνεται στο Σχ.1.3.2. Οι επιμέρους βαθμίδες της αναλύονται στις παραγράφους που ακολουθούν.



Σχ.1.3.2: Κυκλωματικό διάγραμμα της συσκευής Station.

1.4 Περιγραφή των μονάδων των συσκευών “Sensor” – “Station”

Στις παραγράφους που ακολουθούν περιγράφονται τα βασικά χαρακτηριστικά και η λειτουργία των ολοκληρωμένων κυκλωμάτων και των βασικών εξαρτημάτων που χρησιμοποιήθηκαν για την υλοποίηση των συσκευών. Και στις δυο συσκευές χρησιμοποιήθηκε το ίδιο τυπωμένο κύκλωμα (PCB) με κάποιες μικρές αλλαγές.

Τόσο στη συσκευή Sensor όσο και στη συσκευή Station χρησιμοποιήθηκε ο ίδιος μικροελεγκτής και το ίδιο ολοκληρωμένο κύκλωμα πομποδέκτη. Επιπλέον στη συσκευή Sensor χρησιμοποιήθηκε ολοκληρωμένο κύκλωμα αισθητήρα θερμοκρασίας και στη συσκευή Station χρησιμοποιήθηκε ολοκληρωμένο κύκλωμα πομποδέκτη επικοινωνίας RS-232.

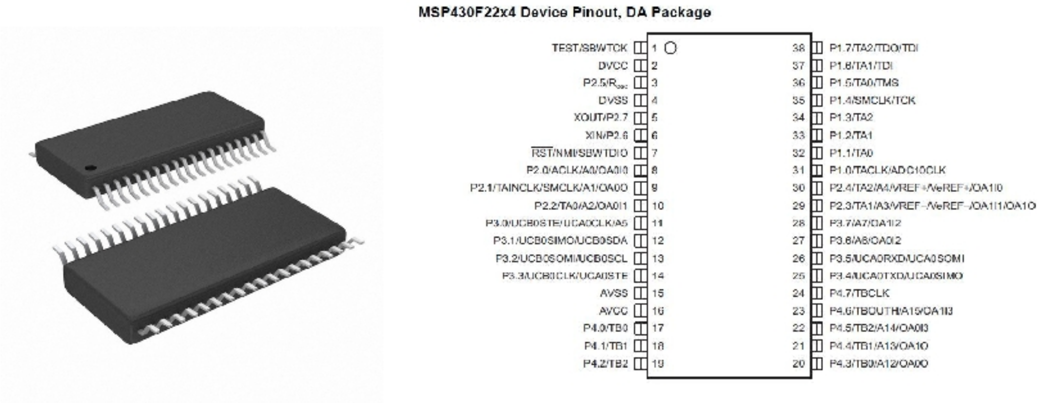
Το πρόγραμμα του κάθε μικροελεγκτή γράφτηκε σε γλώσσα assembly με την βοήθεια του περιβάλλοντος εργασίας IAR Embedded Workbench. Τα προγράμματα αυτά περιγράφονται στις παραγράφους 2.2 – 2.3 του Κεφ. 2.

1.4.1 Η κεντρική μονάδα επεξεργασίας και ελέγχου

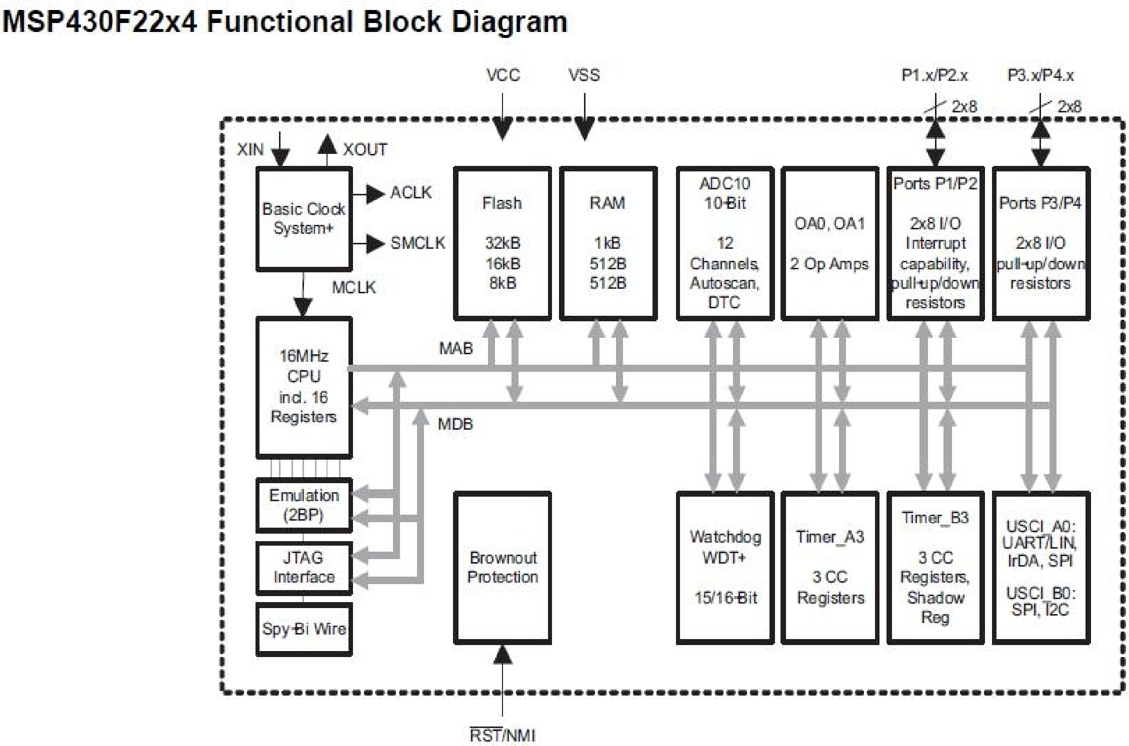
Στα πλαίσια της πτυχιακής αυτής, αρχικά μελετήθηκε ο μικροελεγκτής MSP430F5438 της εταιρίας Texas Instruments[1]. Κατασκευάστηκε αναπτυξιακό όπου μελετήθηκε θεωρητικά και πρακτικά η κάθε βαθμίδα του μικροελεγκτή ξεχωριστά. Όμως θέλοντας να γίνει εξοικονόμηση χώρου στις συσκευές και καθώς δεν υπήρχαν ιδιαίτερες απαιτήσεις από πλευρά βαθμίδων και ακροδεκτών για την υλοποίηση της κεντρικής μονάδας ελέγχου και επεξεργασίας τους, χρησιμοποιήθηκε ένας απλούστερος μικροελεγκτής μίας άλλης οικογένειας, της ίδιας εταιρίας, ο MSP430F2274. Πρόκειται για ένα μικροελεγκτή μικρότερο από τον MSP430F5438, χαμηλού κόστους και πολύ χαμηλής κατανάλωσης, ιδανικό για εφαρμογές που τροφοδοτούνται από μπαταρία. Στο κέλυφός του ενσωματώνεται μία κεντρική μονάδα επεξεργασίας των 16-bit, καταχωρητές των 16-bit, ένας ψηφιακά ελεγχόμενος ταλαντωτής (DCO), 2 μετρητές των 16-bit, ένας μετρητής ελέγχου σε περίπτωση σφάλματος στο πρόγραμμα του μικροελεγτή (watchdog timer), σειριακή επικοινωνία (UART/IrDA/SPI), ένας μετατροπέας από αναλογικό σε ψηφιακό (A/D), 2 τελεστικοί ενισχυτές γενικής χρήσης και 32 ακροδέκτες εισόδου-εξόδου. Η μέγιστη συχνότητα χρονισμού είναι 16MHz. Η μνήμη του μικροελεγκτή είναι 32KB, 256B flash memory και 1KB RAM. Ο μικροελεγκτής είναι αρχιτεκτονικής RISC (*Reduced Instruction Set Controller)* και το set εντολών του αποτελείται από 27 εντολές και 7 διαφορετικές λειτουργίες διευθυνσιοδότησης.

Ο μικροελεγκτής μπορεί να τροφοδοτηθεί από 1,8V μέχρι 3,6V. Η κατανάλωση του σε πλήρη λειτουργία είναι 390μA χωρίς να καταναλώνεται φορτίο στις εξόδους του και σε λειτουργίες χαμηλής κατανάλωσης, η κατανάλωση κυμαίνεται από 90μA μέχρι και 0.1μA.

[1] Texas Instruments: http://www.ti.com



Σχήμα 1.4.1.1: Ο μικροελεγκτής MSP430F2274 σε κέλυφος TSSOP-38 και το διάγραμμα των ακροδεκτών του.



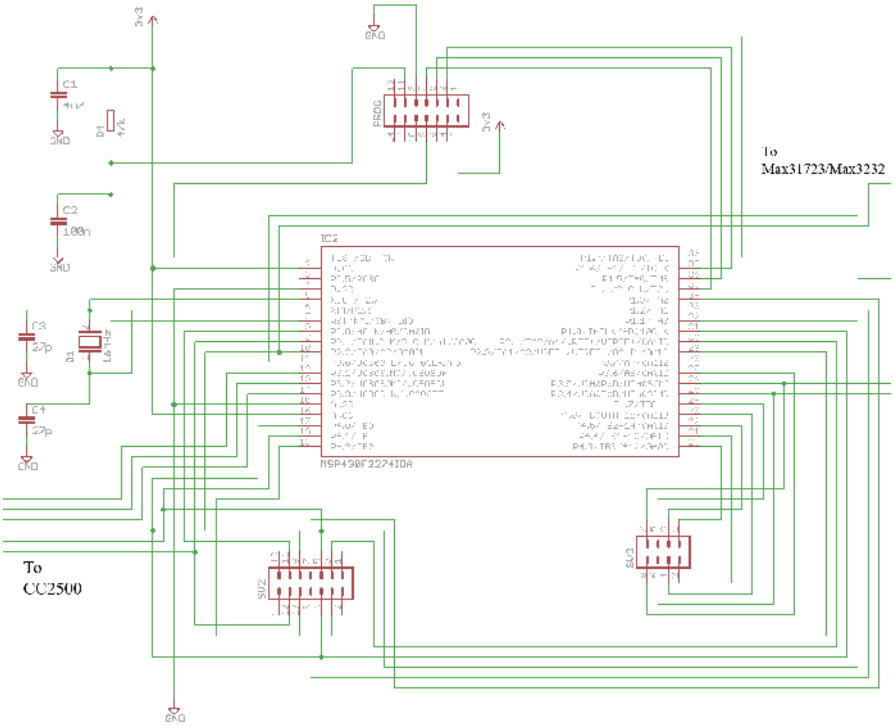
Σχήμα 1.4.1.2: Το εσωτερικό διάγραμμα βαθμίδων του μικροελεγκτή MSP430F2274.

Σύνδεση και λειτουργία των ακροδεκτών του MSP430F2274

- Τάση τροφοδοσίας 3V3: DVCC, AVCC
- Γείωση: DVSS, AVSS
- Σύνδεση κρυστάλλου 16 MHz προς υλοποίηση ταλαντωτή: XTALIN, XTALOUT
- Σήματα για τον προγραμματισμό του μικροελεγκτή: SBWTCK, RST, SMCLK, TMS, TDI, TDO
- SPI (Serial Peripheral Interface) επικοινωνία με πομποδέκτη: UCB0SIMO, UCB0SOMI, UCB0CLK

- SPI (Serial Peripheral Interface) επικοινωνία με αισθητήρα θερμοκρασίας, συσκευή Sensor: UCA0SIMO, UCA0SOMI, UCA0CLK
- UART (Universal Asynchronous Receiver/Transmitter) σειριακή επικοινωνία RS-232, συσκευή Station: UCA0RX, UCA0TX

Σχετικά με την χρήση των υπόλοιπων ακροδεκτών εισόδου εξόδου θα γίνει αναφορά στη κάθε βαθμίδα ξεχωριστά.



Σχήμα 1.4.1.3: Το σχηματικό διάγραμμα συνδεσμολογίας του μικροελεγκτή MSP430F2274.

Τόσο στη συσκευή Sensor όσο και στη συσκευή Station χρησιμοποιήθηκε η βαθμίδα επικοινωνίας SPI UCB0 για την επικοινωνία του μικροελεγκτή με τον πομποδέκτη. Τα σήματα για την επικοινωνία SPI είναι το σήμα ρολογιού CLK, το σήμα δεδομένων εισόδου στο μικροελεγκτή SOMI, το σήμα δεδομένων εξόδου από τον μικροελεγκτή SIMO και το σήμα επιλογής του ολοκληρωμένου με το οποίο επικοινωνεί ο μικροελεγκτής STE. Επίσης στη συσκευή Sensor χρησιμοποιήθηκε η βαθμίδα επικοινωνίας SPI UCA0 για την επικοινωνία του μικροελεγκτή με το ολοκληρωμένο κύκλωμα του αισθητήρα θερμοκρασίας. Τέλος στη συσκευή Station χρησιμοποιήθηκε η ίδια πόρτα επικοινωνίας σε UART λειτουργία (UCA0RXD, UCA0TXD) για την επικοινωνία του μικροελεγκτή μέσω σειριακής θύρας με τον υπολογιστή, παρεμβάλλοντας και το ολοκληρωμένο κύκλωμα MAX3232.

1.4.2 Ο πομποδέκτης δεδομένων

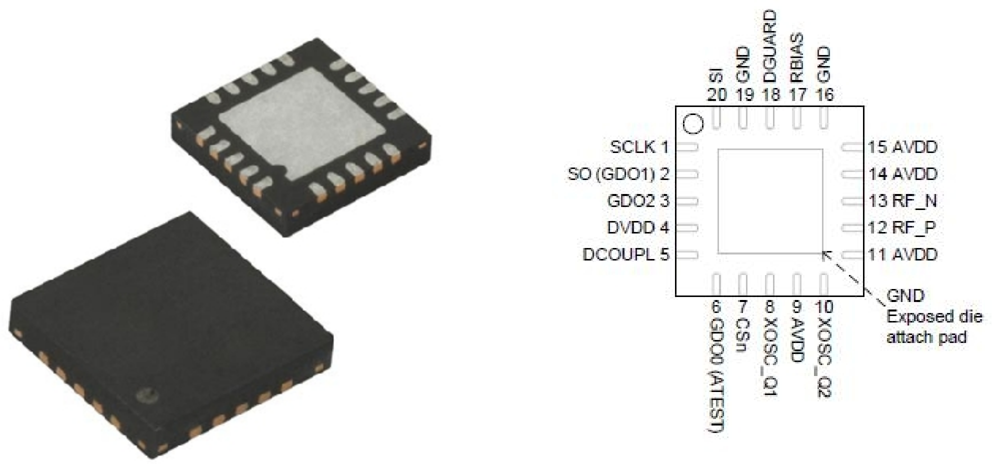
Για την βαθμίδα του πομποδέκτη δεδομένων χρησιμοποιήθηκε το ολοκληρωμένο κύκλωμα CC2500 της εταιρίας Texas Instruments. Πρόκειται για ένα ολοκληρωμένο πολύ μικρού μεγέθους, πολύ χαμηλού κόστους και κατανάλωσης αλλά πολλών δυνατοτήτων.

Η συχνότητα εκπομπής του είναι 2.4GHz. Η συχνότητα αυτή είναι ελεύθερη συχνότητα για επικοινωνίες και ανήκει στη ζώνη συχνοτήτων ISM, Industrial Scientific and Medicine. Η επικοινωνία με την μονάδα ελέγχου γίνεται μέσω SPI και για την λειτουργία του χρειάζονται μόνο λίγα παθητικά εξαρτήματα επιπλέον.

Το ολοκληρωμένο είναι εφοδιασμένο με βαθμίδα διαμορφωτή (modem) που δίνει την δυνατότητα διαμόρφωσης του προς εκπεμπόμενου σήματος με διάφορους τρόπους OOK (on – off Keying), 2-FSK (Frequency Shift Keying για δυαδικά σήματα δεδομένων), GFSK (Gaussian Frequency Shift Keying), MSK (Minimum - Shift Keying). Η μέγιστη ταχύτητα μετάδοσης είναι 500 kBaud. Λόγω των εκτεταμένων βαθμίδων που περιέχονται στο ολοκληρωμένο κύκλωμα δίνεται η δυνατότητα για διαχείριση πακέτων, προσωρινή αποθήκευση δεδομένων (64 byte RX and TX FIFOs), συνεχόμενες μεταδόσεις, εκτίμηση καναλιού για το αν είναι ελεύθερο ή όχι το κανάλι μετάδοσης, εκτίμηση ποιότητας της ζεύξης και δυνατότητα λειτουργίας σε αναμονή και επαναλειτουργίας σε κανονική κατάσταση την στιγμή λήψης σήματος (Wake on Radio). Αναλυτικότερα, για τα πακέτα δεδομένων παρέχονται οι δυνατότητες ελέγχου λέξης συγχρονισμού, διεύθυνσης, μεγέθους πακέτου, σφαλμάτων και αποσφαλμάτωσης. Εκτός από την δυνατότητα ελέγχου στα πακέτα δίνεται και η δυνατότητα ελέγχου του εύρους ζώνης του φίλτρου πομπού και δέκτη και ο έλεγχος ύπαρξης φέρων κύματος στο δέκτη.

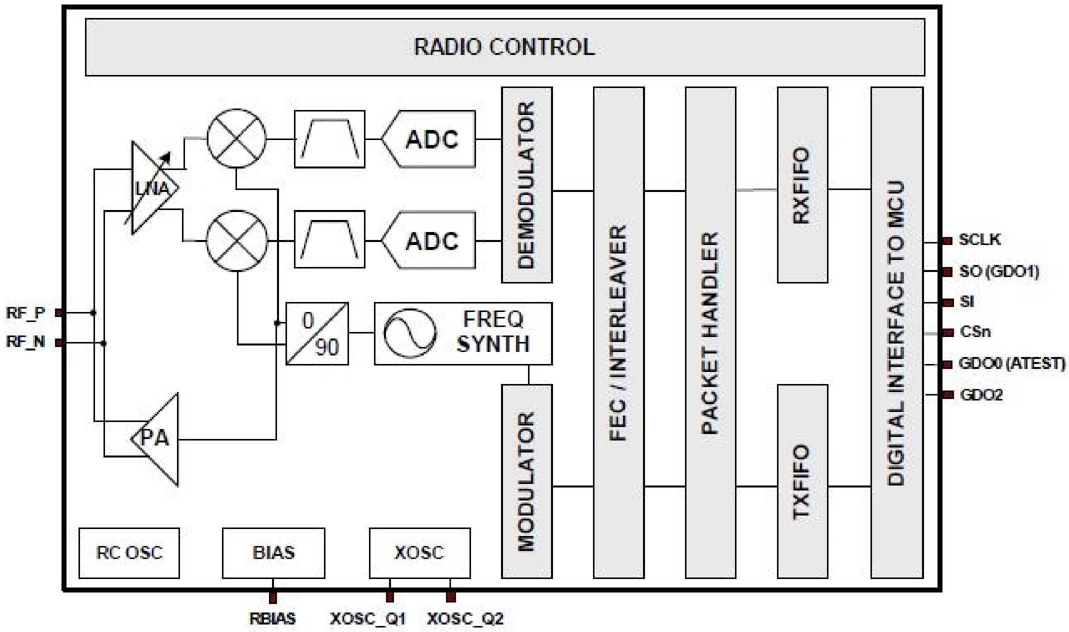
Ο δέκτης του ολοκληρωμένου έχει ευαισθησία -100dBm και υψηλή επιλεκτικότητα και η ισχύς εκπομπής του πομπού είναι προγραμματιζόμενη μέχρι και +1dBm.

Η κατανάλωση του ολοκληρωμένου είναι 400nA σε αναμονή (SLEEP mode), 17mA σε κατάσταση λήψης (RX mode) και 21mA σε κατάσταση εκπομπής (TX mode) για την μέγιστη ισχύ εκπομπής. Ο χρόνος που χρειάζεται για να μεταβεί από την μια κατάσταση στην άλλη είναι μερικά μSec.



Σχήμα 1.4.2.1: Ο πομποδέκτης CC2500 σε κέλυφος QFN-20 και το διάγραμμα των ακροδεκτών του.

CC2500 Functional Block Diagram

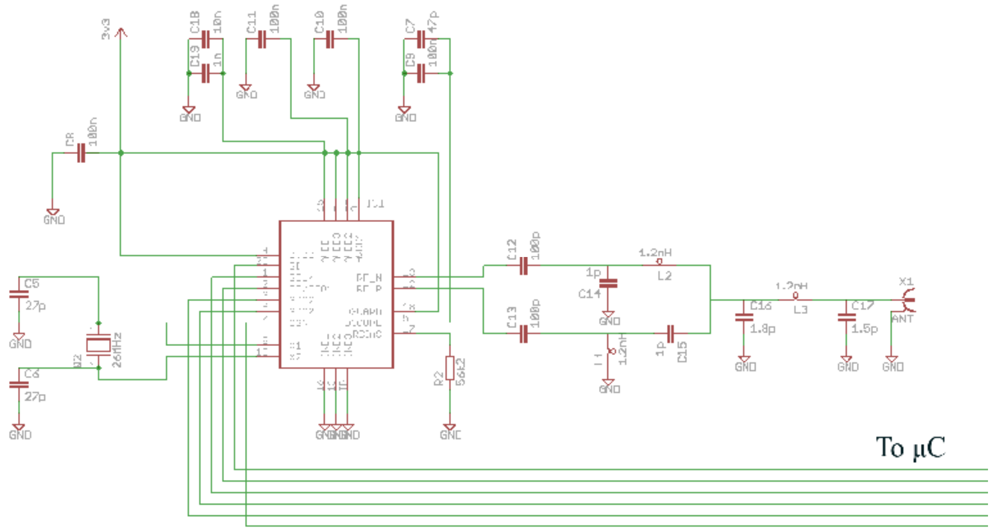


Σχήμα 1.4.2.2: Το εσωτερικό διάγραμμα βαθμίδων του πομποδέκτη CC2500.

Σύνδεση και λειτουργία των ακροδεκτών του πομποδέκτη CC2500

- Τάση τροφοδοσίας 3V3: DVDD, AVDD1, AVDD2, AVDD3, AVDD4, DGUARD
- Γείωση: GND1, GND2, GND3 (TP)
- Σύνδεση κρυστάλλου 26 MHz προς υλοποίηση ταλαντωτή: X1, X2
- SPI (Serial Peripheral Interface) επικοινωνία με κεντρική μονάδα ελέγχου: SCLK, SI, SO
- Σήμα επιλογής ολοκληρωμένου: CSN
- Σήματα ελέγχου πομποδέκτη: GDO0, GDO2
- Πυκνωτές απόζευξης: DCOUPL
- Αντίσταση πόλωσης: RBIAS
- Θετική έξοδος /είσοδος RF σήματος: RF_P
- Αρνητική έξοδος/είσοδος RF σήματος: RF_N

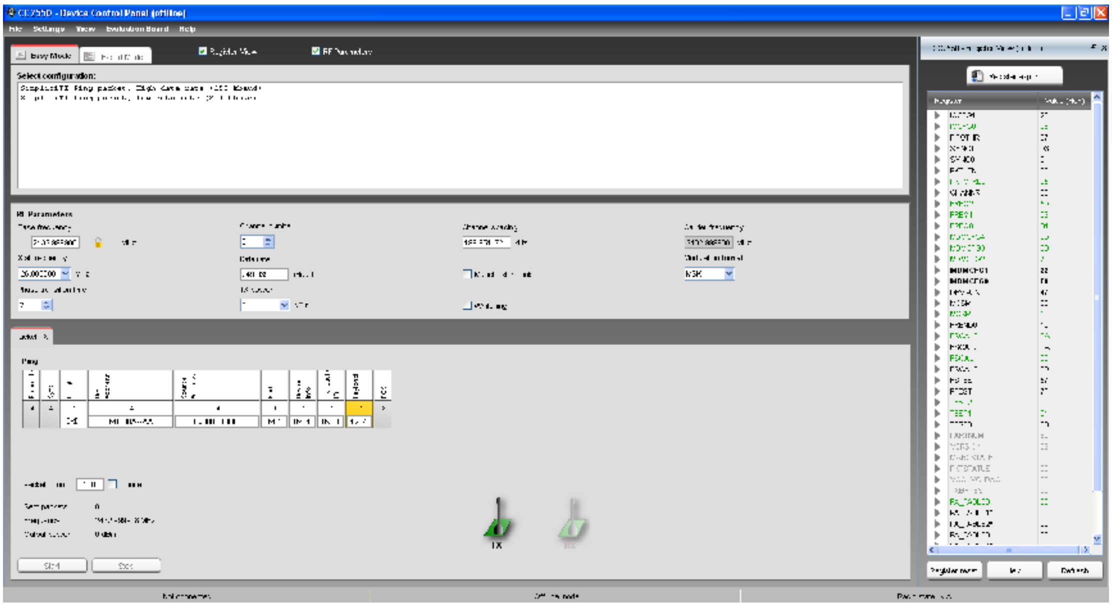
Μεταξύ θετικής και αρνητικής εξόδου/εισόδου χρησιμοποιήθηκε φίλτρο τύπου balun που μετατρέπει το διαφορικό RF σήμα σε κοινό (single – ended). Επίσης χρησιμοποιήθηκαν πυκνωτές απόρριψης DC συνιστώσας. Το φίλτρο balun μαζί με το L – C δικτύωμα που χρησιμοποιείται έπειτα δημιουργούν μια προσαρμογή τιμής 50Ω για την σύνδεση της κεραίας. Η κεραία που χρησιμοποιήθηκε και στις δυο συσκευές επιλέχθηκε από τα τεχνικά φύλλα του κατασκευαστή έτσι ώστε να ταιριάζει στις προδιαγραφές του συστήματος. Είναι κεραία τυπωμένη στην πλακέτα της συσκευής, με μέγιστη ενίσχυση +3,3 dB και εύρος ζώνης συχνοτήτων 280 MHz.



Σχήμα 1.4.2.3: Το σχηματικό διάγραμμα συνδεσμολογίας του πομποδέκτη CC2500.

Για την επικοινωνία των συσκευών προγραμματίστηκαν οι πομποδέκτες έτσι ώστε η συχνότητα εκπομπής και λήψης τους να είναι 2433 MHz, η ισχύς εξόδου του πομπού 0 dBm, το εύρος ζώνης του φίλτρου λήψης 540 kHz, η ταχύτητα μετάδοσης δεδομένων 250 kBps και η διαμόρφωση δεδομένων τύπου MSK. Επίσης η απόσταση μεταξύ του κάθε καναλιού να είναι 199,95 kHz, η εκπομπή να γίνεται στο κανάλι 0 και η λειτουργία του πομποδέκτη να είναι βελτιστοποιημένη ως προς την ευαισθησία του δέκτη. Για την διαμόρφωση των δεδομένων επιλέχτηκε αυτό το είδος επειδή το απαιτούμενο εύρος ζώνης καναλιού είναι μικρό, σε σχέση με τις απαιτήσεις των υπόλοιπων διαμορφώσεων και επίσης υπάρχει ανεκτικότητα στο θόρυβο. Η ταχύτητα μετάδοσης επιλέχτηκε έτσι ώστε να υπάρχει ικανοποιητική σχέση μεταξύ ταχύτητας μετάδοσης και σφάλματος κατά την λήψη BER (Bit Error Rate), σύμφωνα με τα τεχνικά φύλλα του κατασκευαστή, καθώς επίσης και οι υπόλοιπες τιμές, εύρος ζώνης φίλτρου λήψης, απόσταση καναλιών επιλέχθηκαν και αυτές με βάση τα τεχνικά φύλλα αλλά και το πρόγραμμα υπολογισμού των τιμών αυτών Smart RF Studio Σχ. 1.4.2.4 που παρέχει ο κατασκευαστής.

Σε ότι αφορά τις ρυθμίσεις στον τομέα των πακέτων έγινε προγραμματισμός να γίνεται έλεγχος έτσι ώστε τα 30 από τα 32 bits της λέξης συγχρονισμού να ταυτίζονται, να χρησιμοποιούνται οι λίστες FIFOS για προσωρινή αποθήκευση των δεδομένων που αποστέλλονται προς το πομπό και λαμβάνονται από το δέκτη, να γίνεται έλεγχος σφαλμάτων, να υπάρχει η δυνατότητα για μεταβλητό μέγεθος πακέτων με μέγιστο μέγεθος πακέτου 255 bytes και να προστίθενται στο πακέτο δεδομένων κατά την λήψη 2 bytes που θα περιγράφονται σε αυτό πληροφορίες σχετικά με την ισχύ του σήματος, με την ποιότητα της ζεύξης και τον έλεγχο σφάλματος. Τέλος ο ακροδέκτης γενικής χρήσης GDO0 του πομποδέκτη χρησιμοποιήθηκε έτσι ώστε να προκαλεί διακοπές στο πρόγραμμα του μικροελεγκτή όταν η λέξη συγχρονισμού έχει μεταδοθεί ή ληφθεί.



Σχήμα 1.4.2.4: Το πρόγραμμα υπολογισμού των καταχωρυτών λειτουργίας (control registers) του πομποδέκτη CC2500.

1.4.3 Ο αισθητήρας μέτρησης θερμοκρασίας

Για την μέτρηση της θερμοκρασίας των ασθενών χρησιμοποιήθηκε το ολοκληρωμένο κύκλωμα MAX31723 της εταιρίας Maxim[2]. Πρόκειται για ένα ολοκληρωμένο πολύ μικρού μεγέθους και πολύ χαμηλής κατανάλωσης.

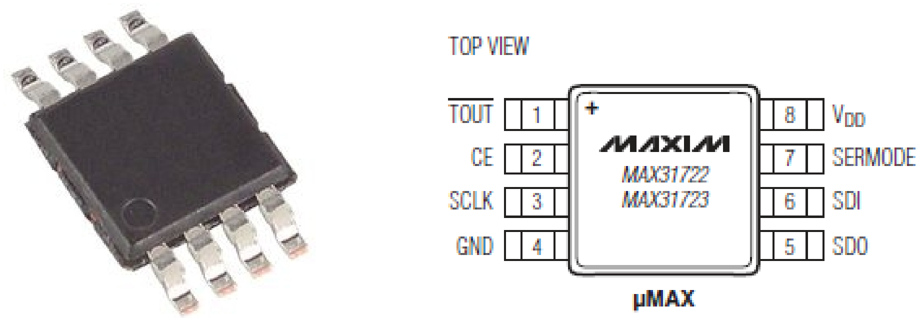
Ο συγκεκριμένος αισθητήρας μπορεί να μετρήσει θερμοκρασίες από -55°C μέχρι +125°C με ακρίβεια ±0,5°C. Το πόσο αναλυτική θα είναι η τιμή της μέτρησης μπορεί να ρυθμιστεί με τον κατάλληλο προγραμματισμό, από 0,5°C μέχρι 0,0625°C.

Ο αισθητήρας μπορεί να λειτουργήσει με δυο διαφορετικούς τρόπους είτε σαν θερμόμετρο είτε σαν θερμοστάτης. Στην περίπτωση του θερμομέτρου μπορεί να παίρνει μετρήσεις θερμοκρασίας συνεχόμενα κάθε φορά που ενεργοποιείται μέσω του σήματος CS (Chip Select), είτε να παίρνει μια μέτρηση και να την αποθηκεύει προσωρινά στη μνήμη EEPROM και να γίνεται ανάκτηση της μέτρησης από την μνήμη. Στη περίπτωση του θερμοστάτη γίνεται καταχώρηση των ορίων θερμοκρασίας, για τα οποία θέλουμε να έχουμε ειδοποίηση, στην μνήμη EEPROM του ολοκληρωμένου και κάθε φορά που γίνεται μέτρηση συγκρίνεται η τιμή της μέτρησης με αυτή των ορίων. Σε περίπτωση που είναι εκτός των ορίων η τιμή που μετρήθηκε, η έξοδος TOUT αλλάζει κατάσταση μέχρι η τιμή της θερμοκρασίας να γίνει χαμηλότερη από αυτή του κάτω ορίου θερμοκρασίας που αποθηκεύτηκε, λειτουργία θερμοστάτη σαν συγκριτή (comparator mode). Αν η λειτουργία του θερμοστάτη είναι σε λειτουργία διακοπής (interrupt mode) τότε η έξοδος TOUT αλλάζει κατάσταση σε περίπτωση που η μετρούμενη θερμοκρασία έχει υπερβεί το αποθηκευμένο όριο και αλλάζει πάλι όταν γίνει ανάγνωση της μέτρησης από της αντίστοιχες θέσεις μνήμης του ολοκληρωμένου.

Η επικοινωνία του αισθητήρα με την μονάδα ελέγχου γίνεται μέσω SPI. Η τροφοδοσία του αισθητήρα μπορεί να είναι από 1,7V έως 3,7V και η κατανάλωση του είναι 1,1mA κατά την διάρκεια της μέτρησης και της εγγραφής στην μνήμη

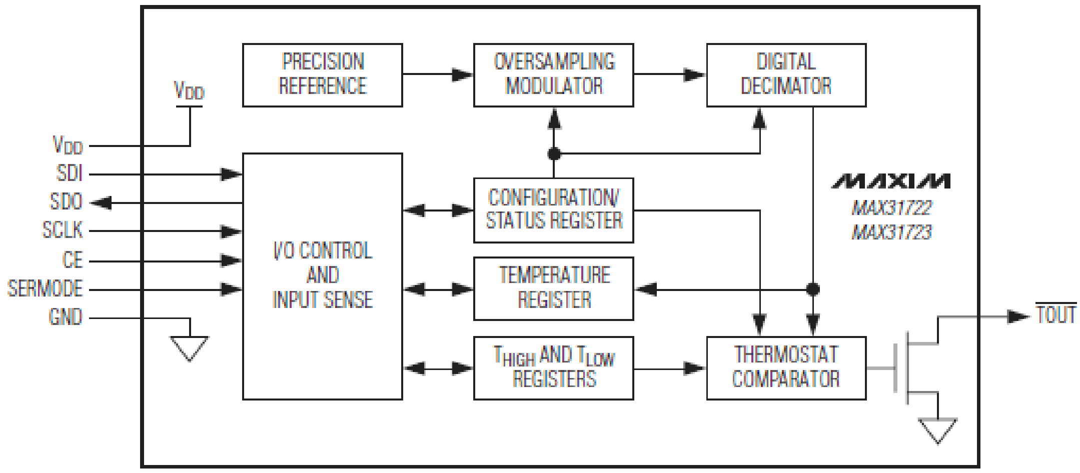
[2] Maxim: http://www.maxim-ic.com

EEPROM, 100μA κατά την διάρκεια της SPI επικοινωνίας και 2μA κατά την διάρκεια που το ολοκληρωμένο δεν είναι ενεργοποιημένο από το σήμα CS.



Σχήμα 1.4.3.1: Ο αισθητήρας θερμοκρασίας MAX31723 σε κέλυφος μMAX-8 και το διάγραμμα των ακροδεκτών του.

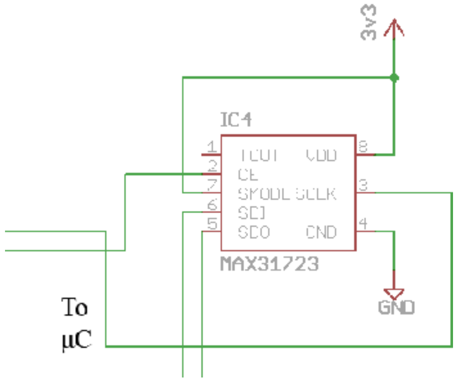
MAX31723 Functional Block Diagram



Σχήμα 1.4.3.2: Το εσωτερικό διάγραμμα βαθμίδων του αισθητήρα MAX31723.

Σύνδεση και λειτουργία των ακροδεκτών του αισθητήρα MAX31723

- Τάση τροφοδοσίας 3V3: VDD
- Γείωση: GND
- Σήμα επιλογής ολοκληρωμένου: CE
- Σήμα επιλογής λειτουργίας SPI / 3Wire: SERMODE
- Έξοδος απαγωγού (Drain) του Mosfet που υπάρχει μέσα στο ολοκληρωμένο και οδηγείται από αυτό κατά την λειτουργία του αισθητήρα σαν θερμοστάτη: TOUT
- SPI (Serial Peripheral Interface) επικοινωνία με κεντρική μονάδα ελέγχου: SCLK, SDI, SDO



Σχήμα 1.4.3.3: Το σχηματικό διάγραμμα συνδεσμολογίας του αισθητήρα MAX31723.

Για την συσκευή Sensor μέσω του κατάλληλου προγραμματισμού ρυθμίστηκε ο αισθητήρας σε συνεχή λειτουργία μέτρησης. Η ανάλυση της μέτρησης ρυθμίστηκε 0,0625°C. Επίσης ο ακροδέκτης SMODE συνδέθηκε στην τάση τροφοδοσίας για να επιλεχτεί το SPI πρωτόκολλο επικοινωνίας σύμφωνα με τα τεχνικά φύλλα του κατασκευαστή. Ο ακροδέκτης TOUT έμεινε αποσυνδεδεμένος καθώς δεν χρησιμοποιήθηκε ο αισθητήρας σε λειτουργία θερμοστάτη.

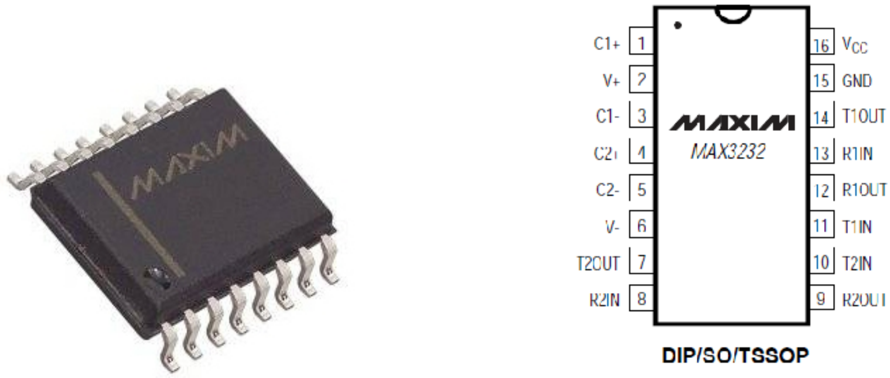
Για τις θερμοκρασίες που θα λαμβάνει ο αισθητήρας η ανοχή στην μέτρηση σύμφωνα με τα αντίστοιχα γραφήματα στα τεχνικά φύλλα του κατασκευαστή είναι της τάξης των ±0,02°C. Επειδή όμως η γενική ανοχή του αισθητήρα είναι ±0,5°C, το συγκεκριμένο ολοκληρωμένο δεν ενδείκνυται για την χρήση σαν θερμόμετρο σώματος. Θα ήταν προτιμότερο να χρησιμοποιηθεί ένα άλλο ολοκληρωμένο αντίστοιχων χαρακτηριστικών με αυτό αλλά με ανοχή πολύ χαμηλότερη. Ο συγκεκριμένος αισθητήρας επιλέχθηκε επειδή τα τεχνικά χαρακτηριστικά του αρκούν για τις απαιτήσεις της πτυχιακής αυτής, καθώς επίσης επειδή παραχωρείται από την κατασκευάστρια εταιρία σαν δείγμα και αποστέλλεται δωρεάν.

1.4.4 Ο πομποδέκτης σειριακής επικοινωνίας RS-232

Για την επικοινωνία της συσκευής Station με υπολογιστή, μέσω της σειριακής θύρας, χρησιμοποιήθηκε το πρωτόκολλο επικοινωνίας RS-232. Το ολοκληρωμένο κύκλωμα MAX3232 της εταιρίας Maxim, χρησιμοποιήθηκε για το σκοπό αυτό.

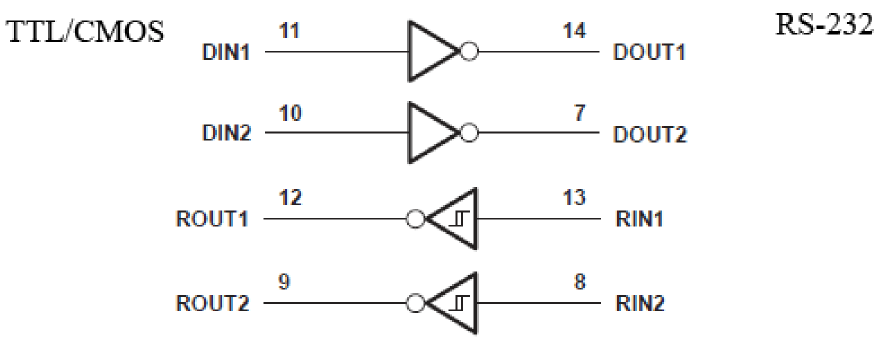
Πρόκειται για ένα πομποδέκτη για σειριακή επικοινωνία RS-232, μικρού μεγέθους και χαμηλής κατανάλωσης. Η βασική λειτουργία του είναι να μετατρέπει τις στάθμες TTL/CMOS που προέρχονται από την κεντρική μονάδα επεξεργασίας και ελέγχου σε στάθμες RS-232, για την σειριακή θύρα UART του υπολογιστή και το αντίστροφο. Η λειτουργία του στηρίζεται στους εξωτερικούς πυκνωτές που συνδέονται στο ολοκληρωμένο έτσι ώστε να παρέχονται μέσω αυτών οι τάσεις +5,5V και -5,5V που χρειάζονται για την επικοινωνία και στους αντιστροφείς τάσεων που υπάρχουν στις βαθμίδες εκπομπής και λήψης μέσα στο ολοκληρωμένο. Η ταχύτητα επικοινωνίας που μπορεί να υποστηρίξει είναι μέχρι 120kbps.

Η τάση τροφοδοσίας του ολοκληρωμένου μπορεί να πάρει τιμές από +3V μέχρι και +5,5V. Η κατανάλωση του κατά τη λειτουργία του σε κατάσταση αναμονής είναι 0,3mA και ±35mA σε κατάσταση εκπομπής.



Σχήμα 1.4.4.1: Ο πομποδέκτης επικοινωνίας RS-232 MAX31723 σε κέλυφος TSSOP-16 και το διάγραμμα των ακροδεκτών του.

MAX3232 Logic Diagram

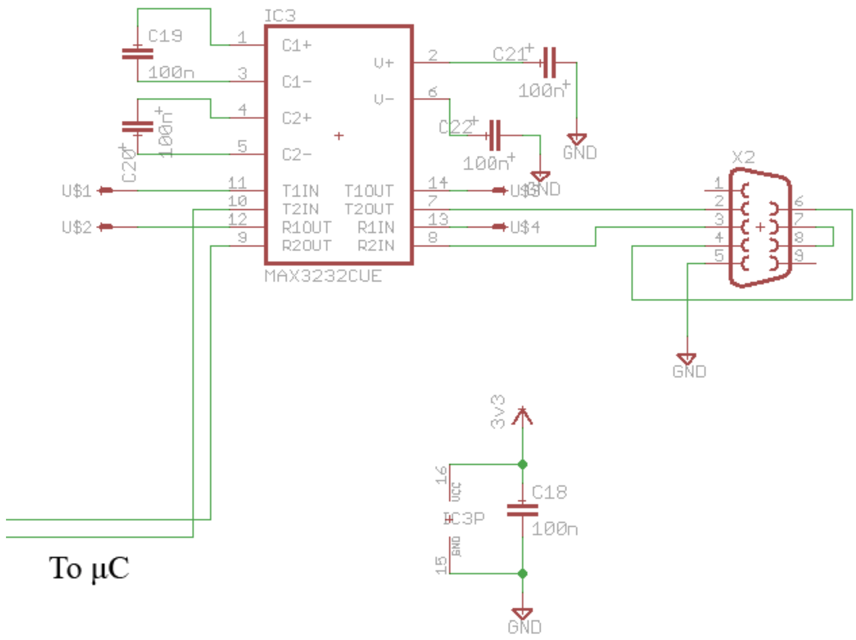


Σχήμα 1.4.4.2: Το λογικό διάγραμμα βαθμίδων εκπομπής και λήψης του πομποδέκτη RS-232 MAX3232.

Σύνδεση και λειτουργία των ακροδεκτών του πομποδέκτη MAX3232

- Τάση τροφοδοσίας 3V3: VCC
- Γείωση: GND
- Θετικός ακροδέκτης για σύνδεση πυκνωτή θετικής παροχής τάσης: C1+
- Αρνητικός ακροδέκτης για σύνδεση πυκνωτή θετικής παροχής τάσης: C1-
- Θετικός ακροδέκτης για σύνδεση πυκνωτή αρνητικής παροχής τάσης: C2+
- Αρνητικός ακροδέκτης για σύνδεση πυκνωτή αρνητικής παροχής τάσης: C2-
- Τάση +5,5V από πυκνωτές παροχής τάσης: V+
- Τάση -5,5V από πυκνωτές παροχής τάσης: V-
- Είσοδος του προς εκπομπή σήματος TX από κεντρική μονάδα ελέγχου και επεξεργασίας: T2IN
- Έξοδος του ληφθέντος RS-232 σήματος RX προς κεντρική μονάδα ελέγχου και επεξεργασίας: R2OUT

- Έξοδος του προς εκπομπή RS-232 σήματος: T2OUT
- Είσοδος του ληφθέντος RS-232 σήματος: R2IN



Σχήμα 1.4.4.3: Το σχηματικό διάγραμμα συνδεσμολογίας του πομποδέκτη επικοινωνίας RS-232 MAX3232.

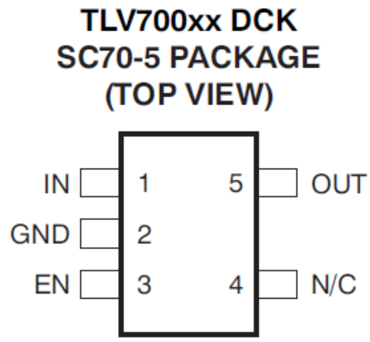
Η επιλογή των πυκνωτών του ολοκληρωμένου έγινε με βάση τα τεχνικά φύλλα του κατασκευαστή σε σχέση με την τάση 3,3V που τροφοδοτήθηκε. Επίσης στην έξοδο των σημάτων RS-232 συνδέθηκε ακροδέκτης σειριακής θύρας τύπου DSUB-9 και έγιναν οι κατάλληλες καλωδιώσεις σε αυτόν ώστε να γίνεται σωστά η επικοινωνία.. Τέλος για πειραματικούς σκοπούς οι ακροδέκτες του 1^{ου} πομποδέκτη, T1IN, R1OUT, T1OUT, R1IN συνδέθηκαν σε ακιδοσειρά ώστε να υπάρχει η δυνατότητα χρήσης τους.

1.4.5 Η τροφοδοσία των συσκευών

Για την τροφοδοσία της συσκευής Sensor υπήρχε η απαίτηση για τροφοδοσία μέσω μπαταρίας υψηλής απόδοσης αλλά μικρού μεγέθους, έτσι ώστε η συσκευή να μπορεί να έχει μεγάλη αυτονομία αλλά ταυτόχρονα να είναι εύχρηστη και να μην ενοχλεί τον ασθενή. Γι' αυτό το λόγο επιλέχθηκε μπαταρία από συσκευή κινητού τηλεφώνου χωρητικότητας 860mAh και τάσης 3,7V. Επειδή η απαιτούμενη τάση της συσκευής είναι 3,3V και η τάση της μπαταρίας από 3,7V μέχρι 4,1V χρησιμοποιήθηκε μετατροπέας – σταθεροποιητής τάσης.

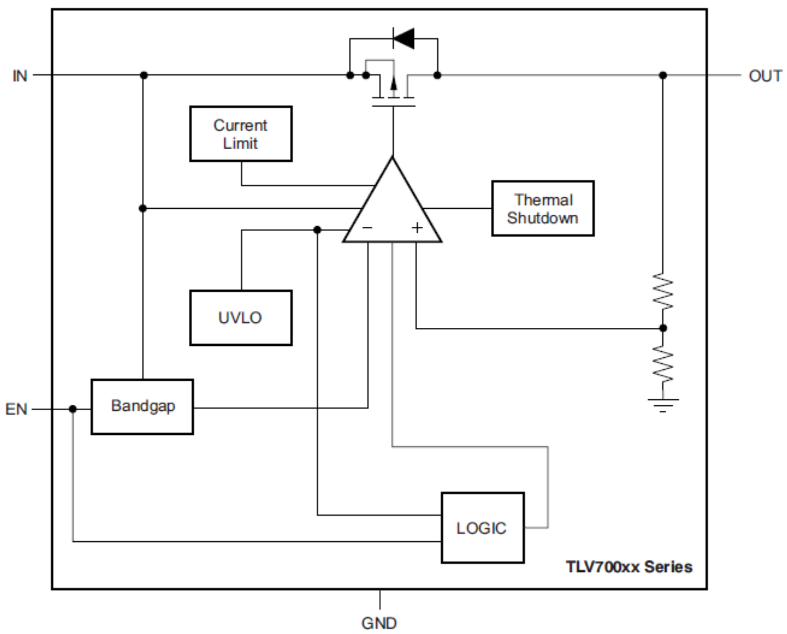
Ο μετατροπέας που χρησιμοποιήθηκε είναι ο TLV70033 της εταιρίας Texas Instruments. Πρόκειται για έναν μετατροπέα τάσης που ενδείκνυται για τα κυκλώματα τροφοδοσίας φορητών συσκευών. Ο συγκεκριμένος μετατροπέας μπορεί να δεχτεί τάσεις εισόδου από 3,5V μέχρι 5,5V και η τάση εξόδου του είναι 3,3V. Επίσης παρέχει τη δυνατότητα διακοπής της λειτουργίας του σε περίπτωση που η θερμοκρασία του υπερβεί τους +160°C και ακόμα παρέχει την δυνατότητα διακοπής σε περίπτωση που η απαίτηση ρεύματος από το κύκλωμα υπερβεί τη μέγιστη τιμή ρεύματος που παρέχει το ολοκληρωμένο.

Για την αποφυγή κυμάτωσης των τάσεων, χρησιμοποιήθηκαν στις τάσεις εισόδου και εξόδου πυκνωτές τιμής 1μF, σύμφωνα με τα τεχνικά φύλλα του κατασκευαστή. Το σήμα EN συνδέθηκε στη τάση τροφοδοσίας έτσι ώστε το ολοκληρωμένο να είναι συνέχεια ενεργοποιημένο.



Σχήμα 1.4.5.1: Ο μετατροπέας – σταθεροποιητής τάσης TLV7033 σε κέλυφος SC70-5 και το διάγραμμα των ακροδεκτών του.

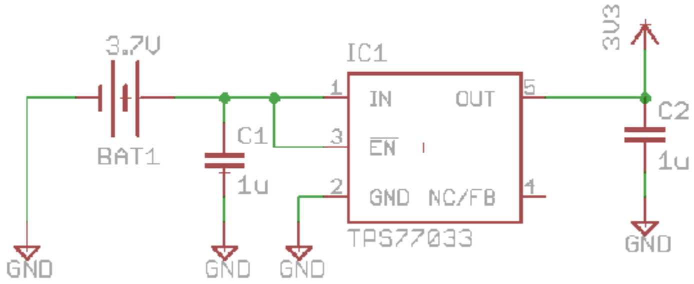
TLV70033 Functional Block Diagram



Σχήμα 1.4.5.2: Το εσωτερικό διάγραμμα βαθμίδων του μετατροπέα – σταθεροποιητή τάσης TLV70033.

Σύνδεση και λειτουργία των ακροδεκτών του μετατροπέα – σταθεροποιητή τάσης TLV70033

- Τάση εισόδου: IN
- Τάση εξόδου 3,3V: OUT
- Γείωση: GND
- Σήμα ενεργοποίησης ολοκληρωμένου: EN



Σχήμα 1.4.5.3: Το σχηματικό διάγραμμα συνδεσμολογίας του μετατροπέα – σταθεροποιητή τάσης TPS770033

Για την τροφοδοσία της συσκευής Station χρησιμοποιήθηκε κοινό τροφοδοτικό εμπορίου. Το συγκεκριμένο τροφοδοτικό που επιλέχτηκε είναι της εταιρίας Mean Well[3] και έχει την δυνατότητα επιλογής τάσης. Η τάση που επιλέχτηκε ήταν 3V με μέγιστη δυνατότητα παροχής ρεύματος τα 1400mA, ρεύμα υπεραρκετό για την τροφοδοσία της συσκευής.

Σχήμα 1.4.5.4: Το τροφοδοτικό MWS8211GS που χρησιμοποιήθηκε για την τροφοδοσία της συσκευής Station.

[3] MEAN WELL: http://www.meanwell.com

1.5 Η υλοποίηση του συστήματος

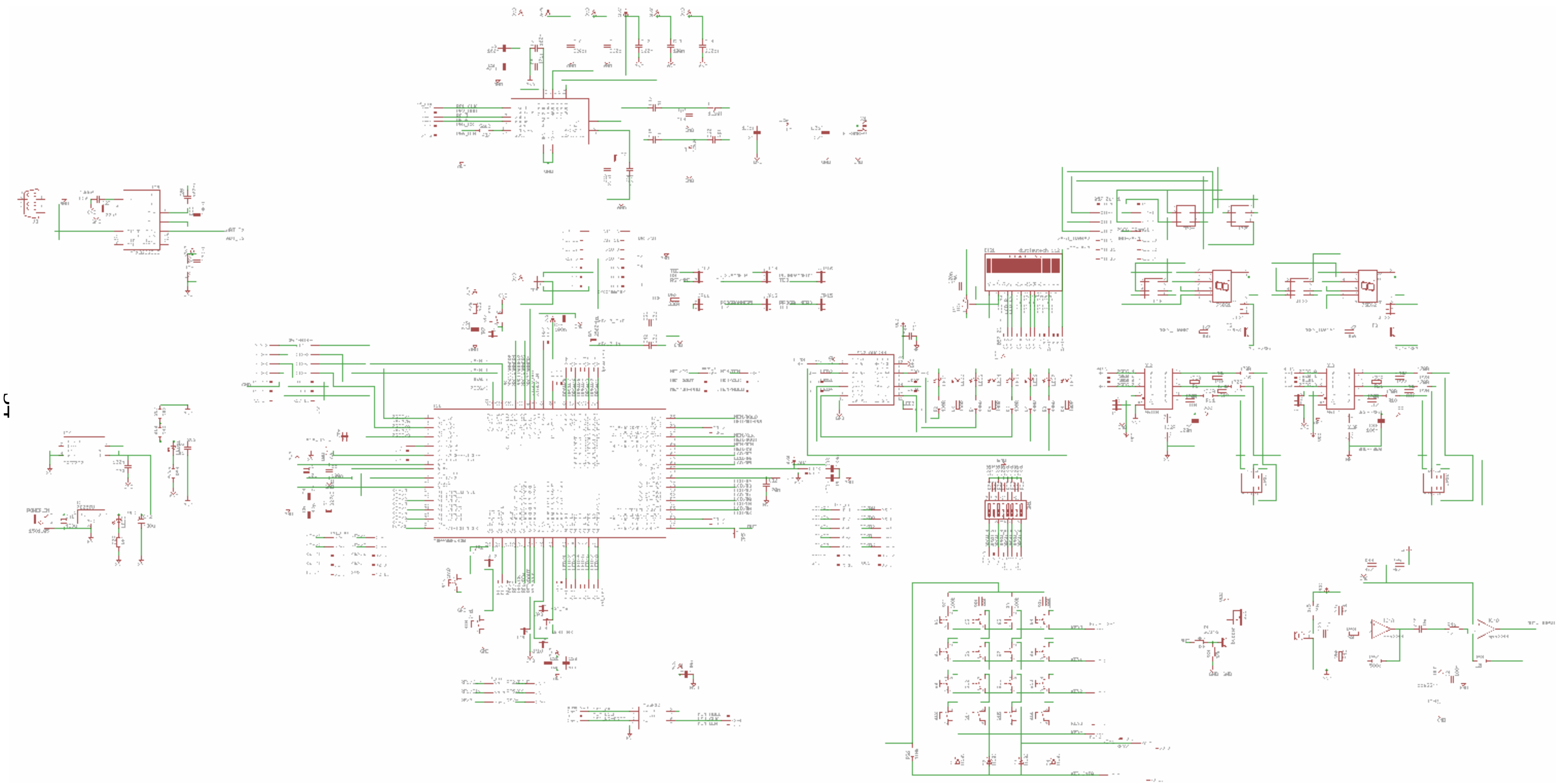
Η μελέτη ξεκίνησε σχεδιάζοντας και υλοποιώντας αναπτυξιακό για τον πρώτο μικροελεγκτή MSP430F5438. Η σχεδίαση έγινε χρησιμοποιώντας το λογισμικό Eagle 5.10.0. Με το αναπτυξιακό αυτό δίνεται η δυνατότητα πειραματισμού με τις βαθμίδες του μικροελεγκτή καθώς επίσης και με διάφορα περιφερειακά τα οποία υπάρχουν στην πλακέτα του αναπτυξιακού και σε μια δεύτερη πλακέτα.

Τα περιφερειακά που υπάρχουν για πειραματισμό στην πλακέτα του αναπτυξιακού είναι:

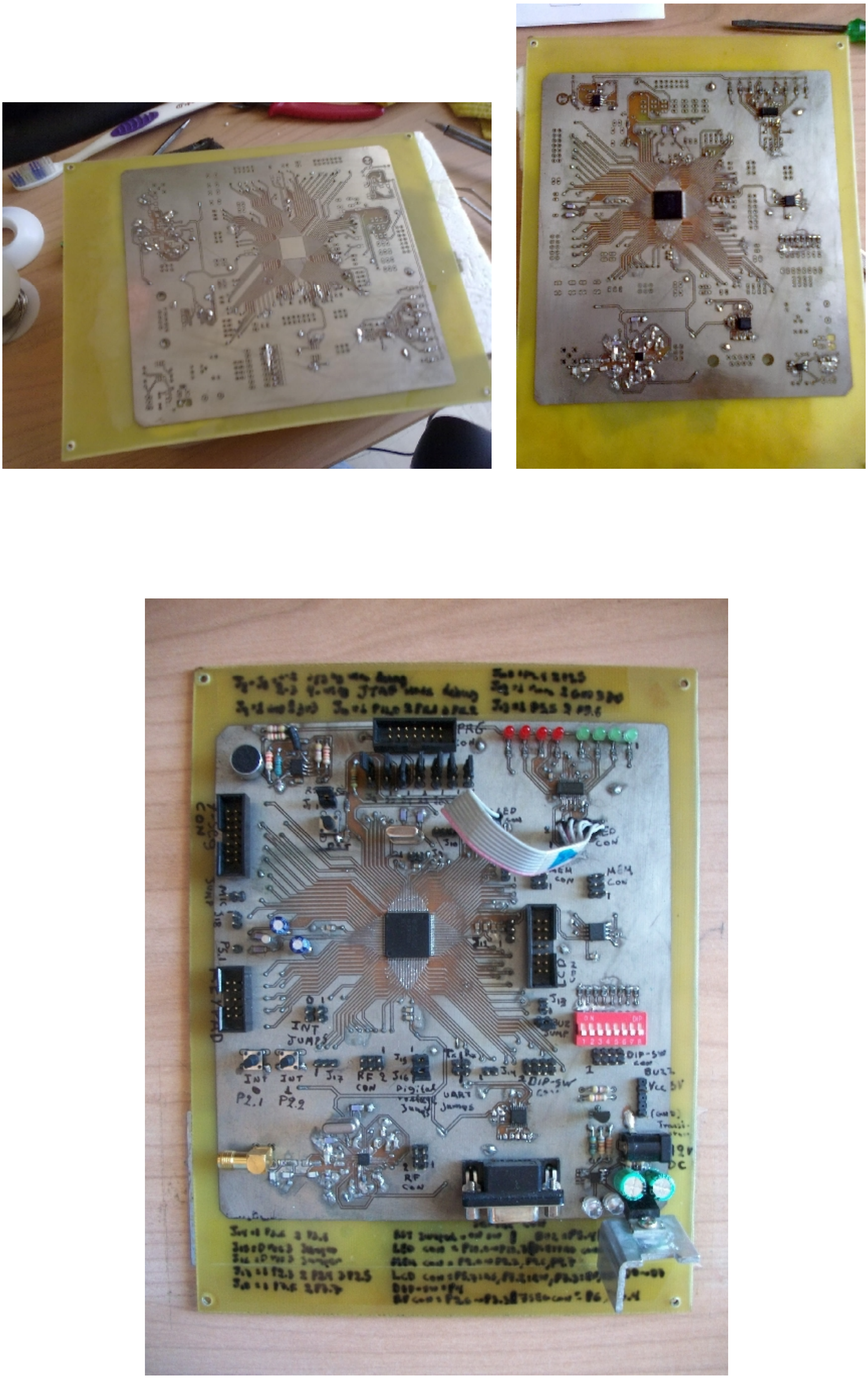
- Μικρόφωνο
- LED
- Μνήμη Flash
- Διακόπτες τύπου Dip-Switches
- Σειριακή επικοινωνία τύπου UART
- Πομποδέκτης δεδομένων (CC2500)
- Διακόπτες διακοπής του προγράμματος που εκτελείται στον μικροελεγκτή (Interrupts)
- Βαθμίδα ελέγχου τάσης με την χρήση transistor
- Ακιδοσειρές για χρήση των ακροδεκτών του μικροελεγκτή για οποιοδήποτε σκοπό

Τα περιφερειακά που υπάρχουν για πειραματισμό στην δεύτερη πλακέτα είναι:

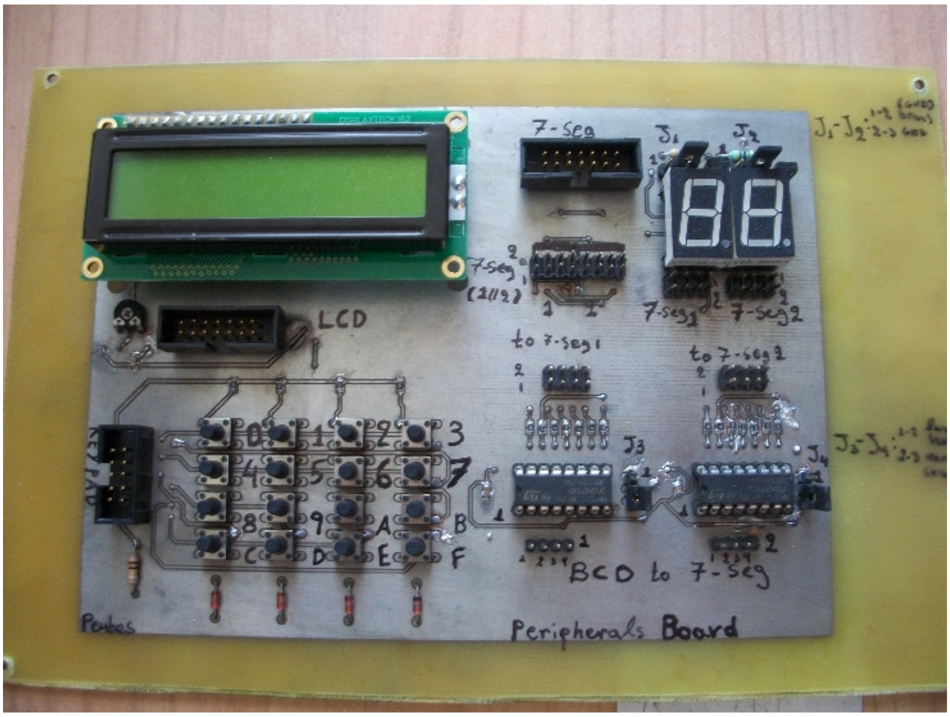
- Οθόνη LCD 2*16 γραμμών
- Πληκτρολόγιο (Push button 4*4)
- 7-segment και ολοκληρωμένα οδήγησης αυτών



Σχήμα 1.5.1: Το σχηματικό διάγραμμα του αναπτυξιακού και της δεύτερης πλακέτας περιφερειακών

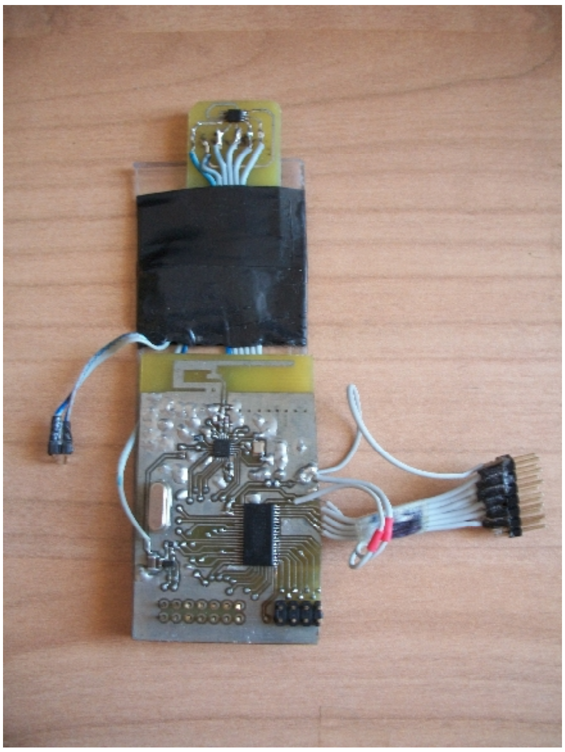


Σχήμα 1.5.2: Τα στάδια υλοποίησης του αναπτυξιακού

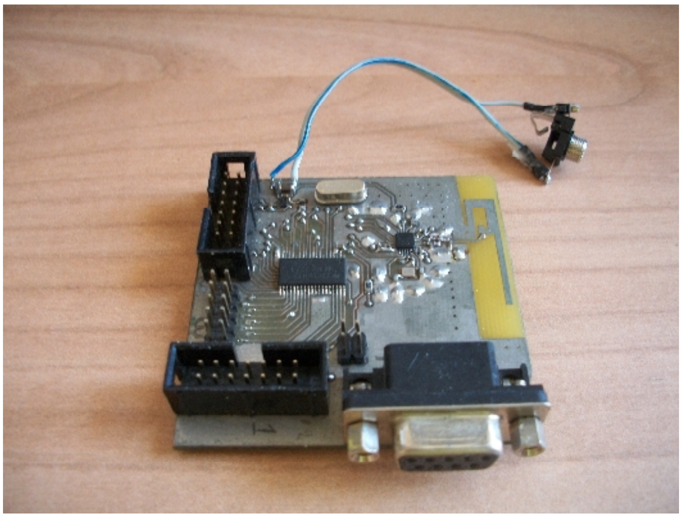


Σχήμα 1.5.3: Η δεύτερη πλακέτα περιφερειακών

Στη συνέχεια, επειδή οι συσκευές έπρεπε να είναι μικρές και εργονομικές, επιλέχθηκε ο μικροελεγκτής MSP430F2274.

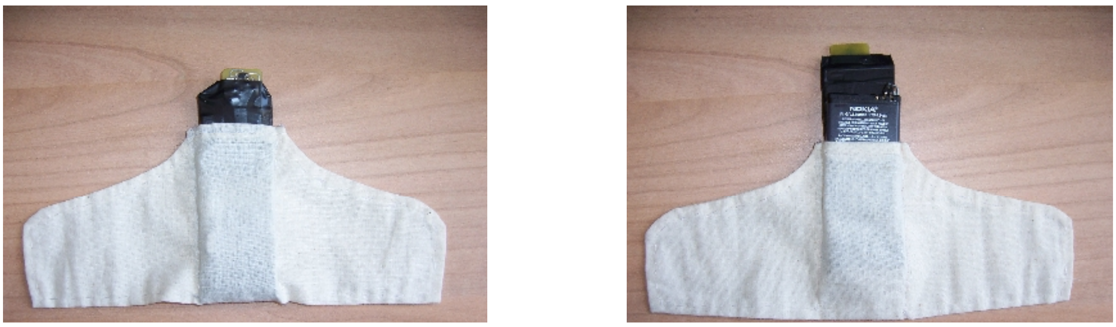


Σχήμα 1.5.4: Η πλακέτα της συσκευής Sensor. Διακρίνονται ο αισθητήρας στο πάνω μέρος της συσκευής, ο πομποδέκτης και η τυπωμένη κεραία, ο μικροελεγκτής και το καλώδιο προγραμματισμού.

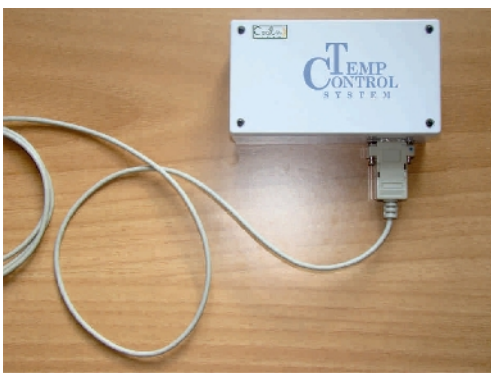


Σχήμα 1.5.5: Η πλακέτα της συσκευής Station. Διακρίνονται ο πομποδέκτης και η τυπωμένη κεραία, ο μικροελεγκτής και ο συνδετήρας για την σειριακή επικοινωνία.

Οι συσκευές Sensor και Station υλοποιήθηκαν σαν συσκευές επίδειξης. Για την συσκευή Sensor δημιουργήθηκε πάνινη θήκη σαν μια πρώτη ιδέα για το που μπορεί να τοποθετηθεί η πλακέτα της συσκευής. Η πάνινη θήκη μπορεί να κολληθεί στο μπράτσο του ασθενούς καθώς επίσης και να πλυθεί και να αποστειρωθεί. Μια άλλη ιδέα σε περίπτωση που το σύστημα γίνει τελική συσκευή είναι η συσκευή Sensor να τοποθετηθεί μέσα σε υλικό καουτσούκ και να υπάρχει ένα καπάκι έτσι ώστε να μπορεί να αλλαχτεί η μπαταρία της συσκευής.



Σχήμα 1.5.6: Η συσκευή Sensor στη πάνινη θήκη και η πίσω όψη αυτής μαζί με την μπαταρία.



Σχήμα 1.5.7: Η συσκευή Station

1.6 Ανάλυση του κόστους των υλικών

Στον πίνακα που ακολουθεί καταγράφονται τα υλικά που απαιτούνται για την κατασκευή των συσκευών.

			Συσκευή Sensor		**Συσκευή Station**	
α/α	**Είδος**	**Τιμή μονάδας (€)**	**Ποσότητα**	**Κόστος (€)**	**Ποσότητα**	**Κόστος (€)**
1	Τυπωμένο κύκλωμα (PCB)	3.5	1	3.5	1	3.5
2	Μικροελεγκτής MSP430F2274	4.64	1	4.64	1	4.64
3	Πομποδέκτης δεδομένων CC2500	3.19	1	3.19	1	3.19
4	Αισθητήρας θερμοκρασίας MAX31723	2.99	1	2.99	-	-
5	Πομποδέκτης επικοινωνίας RS-232 MAX3232	3.60	-	-	1	3.60
6	Διάφορα εξαρτήματα	3.5	1	3.5	1	3.5
7	Συσσωρευτής NOKIA BL-4C 3.7V	2.15	1	2.15	-	-
8	Τροφοδοτικό Universal MWS8211GS	8.60	-	-	1	8.60
9	Πάνινη θήκη	2	1	2	-	-
10	Πλαστικό κουτί	3,5	-	-	1	3,5
	Συνολικό κόστος (€)			**21.97**		**30.53**

Πίνακας 1.6.1: Το κόστος κατασκευής των συσκευών.

Στα πλαίσια της πτυχιακής αυτής υλοποιήθηκε μια συσκευή Sensor και μια συσκευή Station.

ΚΕΦΑΛΑΙΟ 2

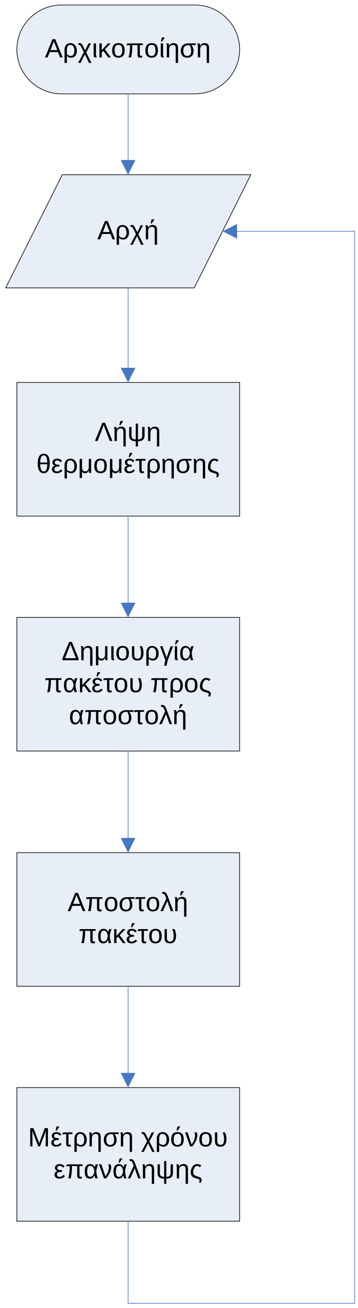
Ο προγραμματισμός των μικροελεγκτών

2.1 Οι διαδικασίες προγραμματισμού του συστήματος

Για τον προγραμματισμό των μικροελεγκτών του συστήματος χρησιμοποιήθηκε το περιβάλλον IAR Embedded Workbench. Ο προγραμματισμός έγινε σε γλώσσα assembly. Πλεονέκτημα αυτής ήταν ότι ο αλγόριθμος του προγράμματος εκτελείται πολύ πιο γρήγορα σε σχέση με τις υπόλοιπες γλώσσες προγραμματισμού. Επίσης ακόμα ένα πλεονέκτημα είναι ότι μπορεί να υπάρχει ακριβής έλεγχος στο τι κάνει το πρόγραμμα σε κάθε εντολή.

Το πρόγραμμα της κάθε συσκευής μαζί με τα διαγράμματα ροής και την περιγραφή τους αναλύονται στις παραγράφους που ακολουθούν. Οι συσκευές Sensor θα πρέπει να προγραμματιστούν έτσι ώστε να έχουν ένα μοναδικό αναγνωριστικό ID και αντίστοιχα οι συσκευές Station να προγραμματιστούν έτσι ώστε να γνωρίζουν από ποιες συσκευές Sensor θα λαμβάνουν δεδομένα. Επίσης στη συσκευή Sensor θα πρέπει να προγραμματιστεί κάθε πότε θα στέλνετε θερμομέτρηση.

2.2 Το πρόγραμμα στο μικροελεγκτή της συσκευής Sensor

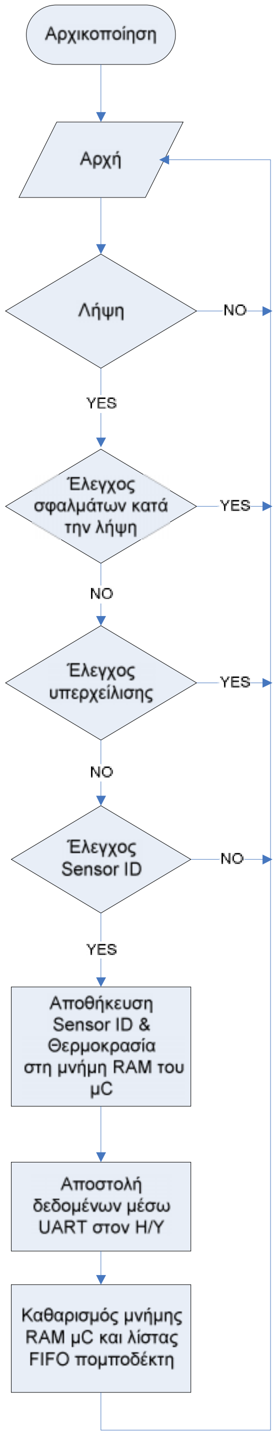


Σχήμα 2.2.1: Το διάγραμμα ροής του προγράμματος στο μικροελεγκτή της συσκευής Sensor.

Στο διάγραμμα ροής του προγράμματος που τρέχει στο μικροελεγκτή της συσκευής Sensor, Σχ. 2.2.1, διακρίνουμε τις εξής διαδικασίες:

1. **Αρχικοποίηση:** Κατά την διαδικασία της αρχικοποίησης, ρυθμίζεται η βαθμίδα της SPI επικοινωνίας για την επικοινωνία μικροελεγκτή και πομποδέκτη. Έπειτα ρυθμίζεται ο τρόπος λειτουργίας του πομποδέκτη. Στη συνέχεια ρυθμίζονται οι ακροδέκτες εισόδου και εξόδου του μικροελεγκτή. Τέλος, ρυθμίζεται η 2^{η} βαθμίδα SPI επικοινωνίας του μικροελεγκτή με τον αισθητήρα θερμοκρασίας και ο τρόπος λειτουργίας του αισθητήρα.
2. **Αρχή:** Αφού τελειώσουν οι απαραίτητες αρχικοποιήσεις ξεκινάει το κυρίως πρόγραμμα.
3. **Λήψη θερμομέτρησης:** Αρχικά γίνεται μέτρηση της θερμοκρασίας και η μέτρηση αποθηκεύεται προσωρινά στη μνήμη RAM του μικροελεγκτή.
4. **Δημιουργία πακέτου:** Στη συνέχεια δημιουργείται το πακέτο δεδομένων που θα αποσταλεί στον δέκτη. Το πακέτο περιλαμβάνει πληροφορίες σχετικά με το μέγεθος του, την διεύθυνση του, το ID της συσκευής Sensor και τα 2 bytes της μέτρησης.
5. **Αποστολή πακέτου:** Αφού το πακέτο έχει δημιουργηθεί καλείται η ρουτίνα αποστολής και αποστέλλεται στο δέκτη.
6. **Μέτρηση χρόνου επανάληψης:** Αφού το πακέτο έχει αποσταλεί, ακολουθεί μια χρονοκαθυστέρηση, η διάρκεια της οποίας μπορεί να ρυθμιστεί ανάλογα με τις ανάγκες του χρήστη. Αφού λήξει η χρονοκαθυστέρηση η διαδικασία εκτελείται πάλι από την αρχή.

2.3 Το πρόγραμμα στο μικροελεγκτή της συσκευής Station



Σχήμα 2.3.1: Το διάγραμμα ροής του προγράμματος στο μικροελεγκτή της συσκευής Station.

Στο διάγραμμα ροής του προγράμματος που τρέχει στο μικροελεγκτή της συσκευής Station, Σχ. 2.3.1, διακρίνουμε τις εξής διαδικασίες:

1. **Αρχικοποίηση:** Κατά την διαδικασία της αρχικοποίησης, ρυθμίζεται η βαθμίδα της SPI επικοινωνίας για την επικοινωνία μικροελεγκτή και πομποδέκτη. Έπειτα ρυθμίζεται ο τρόπος λειτουργίας του πομποδέκτη. Στη συνέχεια ρυθμίζονται οι ακροδέκτες εισόδου και εξόδου του μικροελεγκτή. Τέλος, ρυθμίζεται η βαθμίδα σειριακής επικοινωνίας UART για την επικοινωνία του μικροελεγκτή με Η/Υ.
2. **Αρχή:** Εδώ ξεκινάει το κυρίως πρόγραμμα. Ο μικροελεγτής είναι ρυθμισμένος σε λειτουργία αναμονής και ο πομποδέκτης είναι σε λειτουργία λήψης.
3. **Λήψη:** Την στιγμή που ο πομποδέκτης λαμβάνει RF σήμα προκαλεί διακοπή (Interrupt) στο πρόγραμμα του μικροελεγκτή και ξεκινάει η διαδικασία λήψης του πακέτου.
4. **Έλεγχος σφαλμάτων κατά την λήψη:** Η πρώτη διαδικασία που γίνεται κατά την λήψη του πακέτου είναι να γίνει έλεγχος CRC. Πρόκειται για ένα έλεγχο σχετικά με το αν έγινε σωστή λήψη δεδομένων. Στο ολοκληρωμένο κύκλωμα του πομποδέκτη υπάρχει η βαθμίδα που αναλαμβάνει αυτή την διαδικασία και δίνει το αποτέλεσμα κάνοντας ένα bit ελέγχου “0” ή “1” ανάλογα με το αποτέλεσμα της διαδικασίας. Αν η λήψη του πακέτου είχε σφάλματα τότε η διαδικασία δεν συνεχίζεται και επιστρέφει το σύστημα στην αρχική κατάσταση αναμονής.
5. **Έλεγχος υπερχείλισης:** Η δεύτερη διαδικασία ελέγχου είναι να γίνει έλεγχος υπερχείλισης. Κατά την διαδικασία αυτή ελέγχεται αν τα δεδομένα που έλαβε ο πομποδέκτης ήταν περισσότερα σε σχέση με αυτά που μπορεί να αποθηκεύσει στη λίστα δεδομένων του, FIFO. Αν τα δεδομένα ήταν περισσότερα τότε ένα bit ελέγχου γίνεται “1”, το πακέτο δεν έχει ληφθεί σωστά επειδή έχει αποθηκευτεί μόνο ένα μέρος του, οπότε η διαδικασία δεν συνεχίζεται και επιστρέφει το σύστημα στην αρχική κατάσταση αναμονής.
6. **Έλεγχος Sensor ID:** Η τρίτη διαδικασία είναι να γίνει έλεγχος του Sensor ID. Κατά την διαδικασία αυτή εξετάζεται η πληροφορία που περιέχεται μέσα στο πακέτο δεδομένων σχετικά με το ποια συσκευή Sensor έστειλε την μέτρηση. Αν η συγκεκριμένη συσκευή Station είναι ρυθμισμένη να επικοινωνεί με την συγκεκριμένη συσκευή Sensor τότε η διαδικασία συνεχίζεται, διαφορετικά η διαδικασία τερματίζεται και το σύστημα επιστρέφει στην αρχική κατάσταση αναμονής.
7. **Αποθήκευση Sensor ID και θερμοκρασίας:** Σε περίπτωση που ο παραπάνω έλεγχος είναι αληθής, το επόμενο βήμα του προγράμματος είναι να αποθηκευτούν οι πληροφορίες. Το πακέτο που λήφθηκε περιλαμβάνει την πληροφορία ποια συσκευή Sensor έστειλε την μέτρηση (Sensor ID) και την πληροφορία της μέτρησης. Οι πληροφορίες αυτές αποθηκεύονται προσωρινά στην μνήμη RAM του μικροελεγκτή.
8. **Αποστολή δεδομένων μέσω UART στον Η/Υ:** Αφού αποθηκευτούν τα δεδομένα στη μνήμη RAM του μικροελεγκτή, η επόμενη διαδικασία είναι να αποσταλούν τα δεδομένα αυτά στον Η/Υ. Η αποστολή των δεδομένων γίνεται μέσω της σειριακής θύρας του υπολογιστή και στην συνέχεια αποθηκεύονται σε αυτόν μέσω κατάλληλης εφαρμογής. Η εφαρμογή αυτή αναλύεται στο Κεφ. 3.
9. **Καθαρισμός μνήμης RAM μικροελεγκτή και λίστας δεδομένων FIFO πομποδέκτη:** Αφού ολοκληρωθεί και η διαδικασία αποστολής των

δεδομένων στον Η/Υ το τελευταίο βήμα της διαδικασίας είναι να καθαριστούν η μνήμη RAM του μικροελεγκτή και η λίστα προσωρινής αποθήκευσης δεδομένων FIFO του πομποδέκτη. Μετά την διαδικασία αυτή τα ολοκληρωμένα είναι έτοιμα να αποθηκεύσουν εκ νέου πληροφορίες. Αφού ολοκληρωθεί και αυτό το βήμα, το σύστημα επιστρέφει στην αρχική κατάσταση αναμονής.

ΚΕΦΑΛΑΙΟ 3

Η εφαρμογή

3.1 Η γλώσσα προγραμματισμού C#

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής του συστήματος ήταν η C#. Πρόκειται για μια αντικειμενοστραφής γλώσσα προγραμματισμού, κατασκευασμένη από την Microsoft, με εγγενή υποστήριξη πολλών χαρακτηριστικών της πλατφόρμας .NET. Παρέχει της δυνατότητες της C/C++ με την ευκολία της Visual Basic. Βασικό χαρακτηριστικό της είναι ότι δεν παράγει απευθείας κώδικα μηχανής όπως η C++, άλλα ένα ενδιάμεσο κώδικα που στοχεύει το .NET. Επίσης παρέχει τις δυνατότητες συναρτησιακού αλλά και δυναμικού προγραμματισμού. Πρόκειται για μια γλώσσα πολύ φιλική και εύκολη στη χρήση για τον προγραμματιστή αλλά ταυτόχρονα πολλών δυνατοτήτων λόγω του .NET.

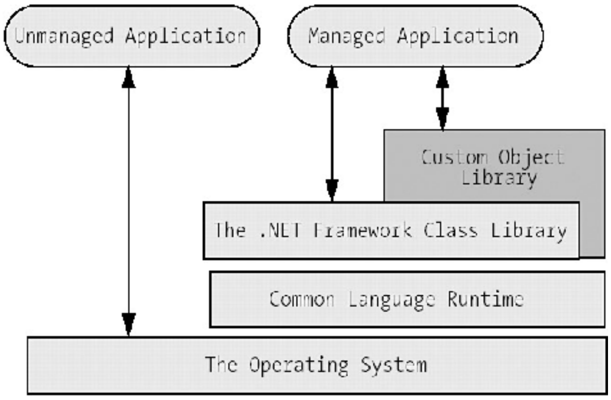
3.2 Η πλατφόρμα .NET

Το .NET είναι μια πλατφόρμα στην οποία μπορούμε να δημιουργήσουμε κάθε είδους αντικειμενοστραφής εφαρμογές. Παρέχει μια πληθώρα βιβλιοθηκών με έτοιμες λειτουργίες που μπορεί να χρησιμοποιήσει ο χρήστης για την ανάπτυξη των εφαρμογών. Χαρακτηρίζεται ως διαχειρίσιμη πλατφόρμα με την έννοια ότι δημιουργεί ένα ελεγχόμενο και ασφαλές περιβάλλον μέσα στο οποίο μπορεί να εκτελείται μια εφαρμογή. Η ασφάλεια έγκειται για παράδειγμα στον έλεγχο, στην δέσμευση και προσπέλαση της μνήμης, στο τύπο των μεταβλητών και δεδομένων ή στην αυτόματη υλοποίηση δικλείδων ασφαλείας. Επίσης περιλαμβάνει μηχανισμούς (CLR) και βιβλιοθήκες (BSL).

Υπάρχουν δυο τύποι εφαρμογών οι διαχειριζόμενες (managed) και οι μη διαχειριζόμενες (unmanaged). Managed εφαρμογές είναι οι εφαρμογές που είναι γραμμένες σε κάποια γλώσσα του .NET. Η εκτέλεση τους διαχειρίζεται από την υποδομή του .NET, το Common Language Runtime ή CLR.

Το .NET Framework περιέχει επίσης την Base Class Library ή BCL. Πρόκειται για μια βασική συλλογή από κλάσεις οι οποίες είναι απαραίτητες για τη λειτουργία του .NET. Μάλιστα η ίδια η εφαρμογή μπορεί να χρησιμοποιήσει πολλές από αυτές για την ίδια τη λειτουργία της αν χρειάζεται, π.χ. για ανάγνωση ή εγγραφή αρχείων.

Unmanaged εφαρμογές είναι οι εφαρμογές που είναι γραμμένες σε C, C++, VB6 (κ.α.) και δεν εκτελούνται κάτω από την ‘επίβλεψη’ του .NET Framework. Όπως είναι φανερό το μεγάλο πλεονέκτημα των managed εφαρμογών είναι ότι το .NET Framework προσφέρει πολλές ευκολίες όπως εργαλεία, έτοιμες βιβλιοθήκες αυτόματη διαχείριση μνήμης, εξαιρέσεων (Exceptions), ασφάλεια, φορητότητα κ.λ.π.



Σχήμα 3.2.1: Συσχετισμός λειτουργικού, μεταγλωττιστών, βιβλιοθηκών σε managed και unmanaged εφαρμογές.

Το Common Language Infrastructure (CLI) είναι ένα ανοιχτό πρότυπο που έχει αναπτυχθεί από τη Microsoft και περιγράφει τον εκτελέσιμο κώδικα αλλά και το περιβάλλον εκτέλεσης που σχηματίζουν τον πυρήνα του .NET Framework. Το πρότυπο προσδιορίζει ένα περιβάλλον που επιτρέπει σε πολλαπλές γλώσσες υψηλού επιπέδου να χρησιμοποιηθούν σε διαφορετικές υπολογιστικές πλατφόρμες χωρίς να χρειάζεται να ξαναγραφούν/επαναμεταγλωττιστούν για συγκεκριμένες αρχιτεκτονικές. Το CLI περιγράφει τις εξής λειτουργίες:

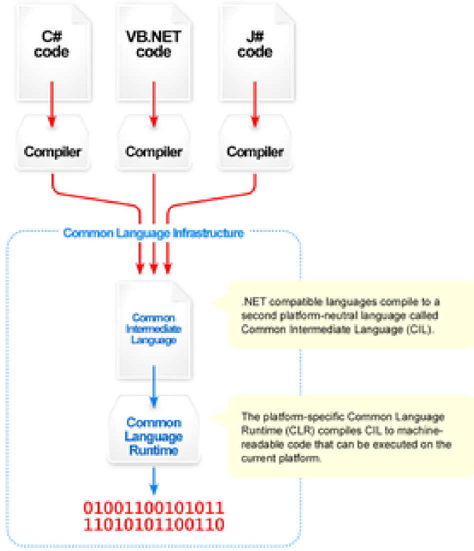
1. **Common Type System (CTS):** Ένα σύνολο από τύπους δεδομένων και λειτουργίες που μοιράζονται από όλες τις CTS-συμβατές γλώσσες.

2. **Metadata:** Οι πληροφορίες για την δομή του προγράμματος/βιβλιοθήκες είναι ανεξάρτητες της γλώσσας προγραμματισμού έτσι ώστε να μπορούν να είναι ορατές από άλλες γλώσσες και εργαλεία, επιτυγχάνοντας την επαναχρησιμοποίηση μεταγλωττισμένου κώδικα ο οποίος μπορεί να είναι γραμμένος σε διαφορετική γλώσσα από την οποία εργαζόμαστε.

3. **Common Language Specification:** Ένα σύνολο κανόνων στους οποίους πρέπει να υπακούει κάθε γλώσσα που είναι γραμμένη για το CLI. Με αυτό τον τρόπο θα μπορεί να διαλειτουργήσει με άλλες CLS-συμβατές γλώσσες. Οι κανόνες του CLS προσδιορίζουν ένα υποσύνολο του CTS.

4. **Virtual Execution System:** Το VES φορτώνει και εκτελεί CLI-συμβατά προγράμματα χρησιμοποιώντας τα metadata, προκειμένου να συνδυάσει κομμάτια κώδικα γραμμένα σε διαφορετικές γλώσσες, κατά την εκτέλεση των προγραμμάτων.

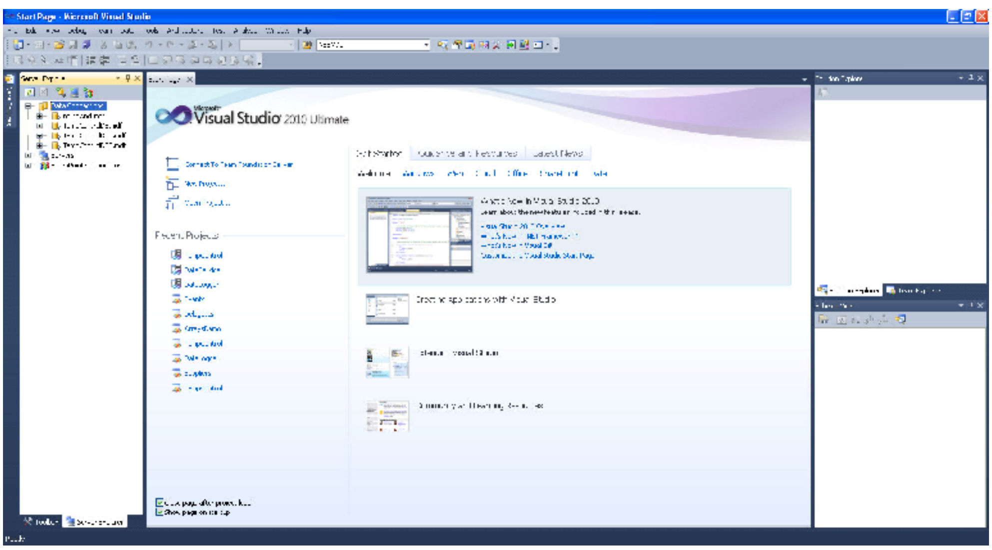
Οι .NET γλώσσες μεταγλωττίζονται σε μια 2η platform-neutral γλώσσα, την Intermediate Language (IL). Το CLR, platform-specific, μεταγλωττίζει την IL σε κώδικα μηχανής για να εκτελεστεί. Με αυτόν τον τρόπο, .NET εφαρμογές μπορούν να γραφούν σε πάρα πολλές γλώσσες, όμως όλες, θα μεταγλωττιστούν από το CLR σε γλώσσα μηχανής για να εκτελεστούν από τον υπολογιστή. Κατά μια έννοια κρύβει το δύσχρηστο Win32 API που χρησιμοποιείται συχνά για προγραμματισμό εφαρμογών Windows και παρουσιάζει στο χρήστη ένα απλούστερο και περισσότερο εύχρηστοWin32 API.



Σχήμα 3.2.2: Η διαδικασία μετατροπής του κώδικα από γλώσσα ανώτερου επιπέδου σε γλώσσα μηχανής. Στο σχήμα εμφανίζονται και τα στάδια IL και CLR.

3.3 Το περιβάλλον προγραμματισμού Visual Studio 2010

Για την ανάπτυξη της εφαρμογής και του κώδικα C# χρησιμοποιήθηκε το περιβάλλον εργασίας Visual Studio 2010. Το περιβάλλον αυτό μπορεί να υποστηρίξει πολλές γλώσσες προγραμματισμού όπως C++, C#, F#, Visual Basic .NET και άλλες. Είναι ένα περιβάλλον εργασίας πολύ φιλικό και εύχρηστο για τον προγραμματιστή όπου σε αυτό μπορεί να αναπτύξει εφαρμογές για PC, για Windows Phone και για ιστοσελίδες. Προσφέρει εργαλεία έτσι ώστε η συγγραφή του κώδικα να γίνεται με πολύ εύκολο τρόπο καθώς και δυνατότητες εντοπισμού σφαλμάτων. Το Visual Studio διατίθεται δωρεάν στους φοιτητές και μπορούν να το κατεβάσουν μέσω της ιστοσελίδας της σχολής τους ή από την σελίδα Microsoft Dream Spark[4].



Σχήμα 3.3.1: Το περιβάλλον εργασίας Visual Studio 2010

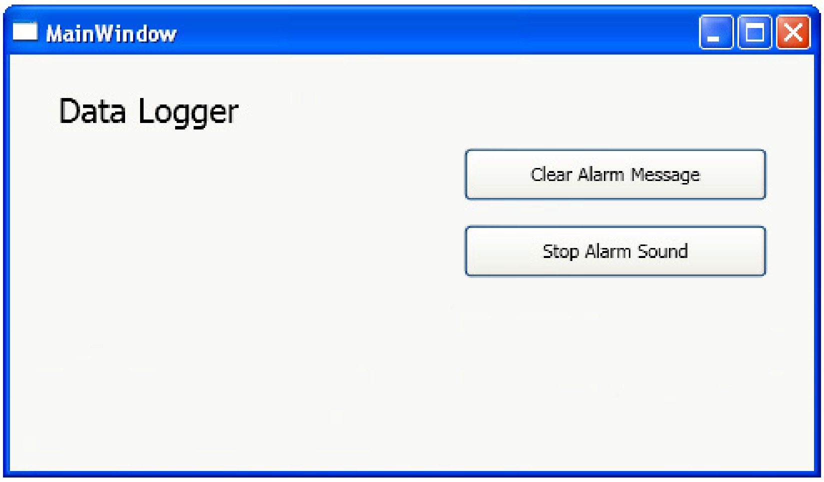
[4] Microsoft Dream Spark: http://www.dreamspark.com

3.4 Η εφαρμογή του συστήματος

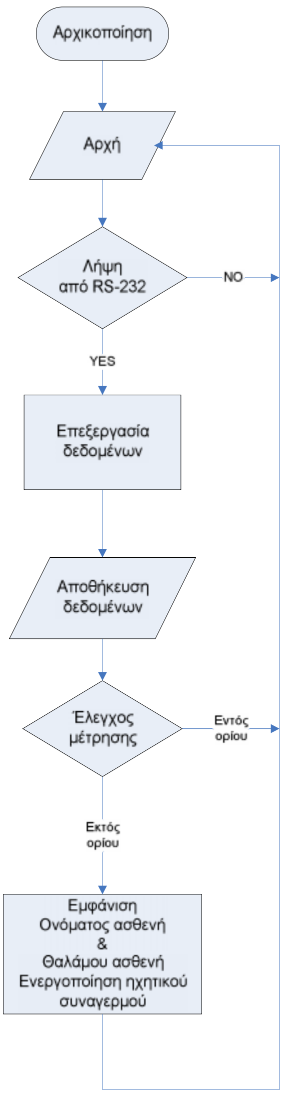
Για την εφαρμογή εκτός από την γλώσσα C# χρησιμοποιήθηκε και η γλώσσα SQL για την αποθήκευση και διαχείριση των δεδομένων. Για την δημιουργία του παραθυρικού περιβάλλοντος της εφαρμογής χρησιμοποιήθηκε το υποσύστημα του .NET, WPF. Πρόκειται για το πιο πρόσφατο υποσύστημα του .NET, που ανέπτυξε η Microsoft για την ανάπτυξη παραθυρικών εφαρμογών. Για τις ανάγκες του συστήματος αναπτύχθηκαν δυο εφαρμογές. Η “DataLogger” για να αποθηκεύει τα δεδομένα και να κάνει κάποιες διεργασίες ελέγχου και η “TempControl” για να γίνεται αναπαράσταση των μετρήσεων και διαχείριση των δεδομένων από τον χρήστη. Επίσης και σε αυτή την εφαρμογή εκτελούνται διεργασίες ελέγχου. Η κάθε εφαρμογή αναλύεται στις παραγράφους που ακολουθούν.

3.4.1 Η πρώτη εφαρμογή συλλογής δεδομένων “DataLogger”

Η εφαρμογή DataLogger είναι υπεύθυνη για την αποθήκευση και τον έλεγχο των θερμομετρήσεων που αποστέλλονται από τις συσκευές Sensor. Οι διαδικασίες που εκτελούνται στην εφαρμογή ξεκινάν με το που ανοίξει ο χρήστης την εφαρμογή.



Σχήμα 3.4.1.1: Το παράθυρο της εφαρμογής DataLogger.



Σχήμα 3.4.1.2: Το διάγραμμα ροής της εφαρμογής DataLogger: Αποθήκευση και έλεγχος θερμομετρήσεων

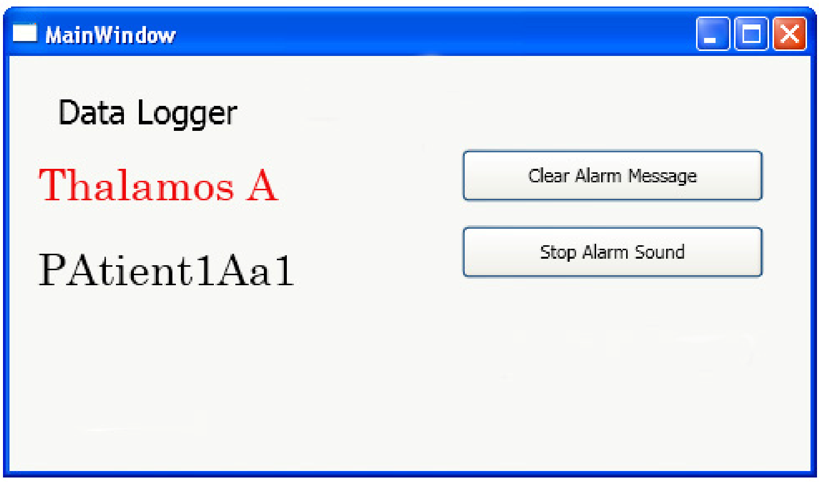
Η πρώτη εφαρμογή DataLogger είναι υπεύθυνη για την αποθήκευση και τον έλεγχο των θερμομετρήσεων. Τα βήματα που εκτελούνται κατά την λειτουργία αυτή είναι τα εξής:

1. **Αρχικοποίηση:** Αρχικά δηλώνονται οι κατάλληλες βιβλιοθήκες που θα χρησιμοποιηθούν και γίνονται οι απαραίτητες αρχικοποιήσεις των μεταβλητών και της σειριακής θύρας, με την οποία επικοινωνεί ο υπολογιστής με την συσκευή Station.
2. **Αρχή:** Η εφαρμογή βρίσκεται σε αναμονή για λήψη από την σειριακή θύρα.
3. **Λήψη:** Σε περίπτωση που ληφθεί πακέτο δεδομένων από την σειριακή θύρα γίνονται οι παρακάτω εργασίες. Διαφορετικά η εφαρμογή παραμένει σε αναμονή.
4. **Επεξεργασία δεδομένων:** Το πακέτο δεδομένων που έλαβε ο υπολογιστής αποτελείται από 3 bytes. Το πρώτο byte είναι το Sensor ID, δηλαδή ποια συσκευή Sensor έστειλε την μέτρηση. Το δεύτερο byte είναι το ακέραιο κομμάτι, της τιμής της μέτρησης και το τρίτο byte είναι το δεκαδικό κομμάτι,

της τιμής της μέτρησης. Τα δυο bytes αφού μετατραπούν από δεκαεξαδικές τιμές σε δεκαδικές, προστίθενται.

5. **Αποθήκευση δεδομένων:** Αφού ολοκληρωθεί η διαδικασία της επεξεργασίας των δεδομένων, στη συνέχεια, αποθηκεύονται στη βάση SQL το ID του Sensor που έστειλε την μέτρηση, η μέτρηση στη τελική της δεκαδική μορφή και η ημερομηνία και ώρα που έγινε η λήψη της μέτρησης.
6. **Έλεγχος μέτρησης:** Τελευταία διαδικασία της λειτουργίας της εφαρμογής αυτής είναι να ελέγχει αν η τιμή της μέτρησης είναι εντός του ορίου θερμοκρασίας που έχει ορίσει ο χρήστης για το συγκεκριμένο SensorID, στην δεύτερη εφαρμογή "TempControl". Σε περίπτωση που η μέτρηση έχει ξεπεράσει την τιμή του ορίου η εφαρμογή ειδοποιεί τον χρήστη εμφανίζοντας στο παράθυρο διαλόγου το όνομα του ασθενή που χρειάζεται προσοχή και το θάλαμο του καθώς επίσης και ένα ηχητικό σήμα ειδοποίησης ακούγεται από τα ηχεία του υπολογιστή. Αφού γίνει η ειδοποίηση η λειτουργία επιστρέφει στην αρχική κατάσταση αναμονής. Ο χρήστης έχει την δυνατότητα μέσω κουμπιών να απενεργοποιήσει το ηχητικό σήμα αλλά και να καθαρίσει το παράθυρο των μηνυμάτων ειδοποίησης. Εάν όλες οι μετρήσεις από τις συσκευές Sensor είναι εντός των ορίων τότε δεν εμφανίζεται κάτι και η λειτουργία επιστρέφει στην αρχική κατάσταση αναμονής.

Στο σχήμα 3.4.1.3 απεικονίζεται το παράθυρο της εφαρμογής σε περίπτωση ειδοποίησης. Κατά την ειδοποίηση εμφανίζεται το όνομα (Patient1Aa1) και ο θάλαμος (Thalamos A) του ασθενούς. Δεξιά διακρίνονται τα κουμπιά ελέγχου με τα οποία ο χρήστης μπορεί να καθαρίσει τα μηνύματα του παραθύρου καθώς επίσης και να απενεργοποιήσει την ηχητική ειδοποίηση.



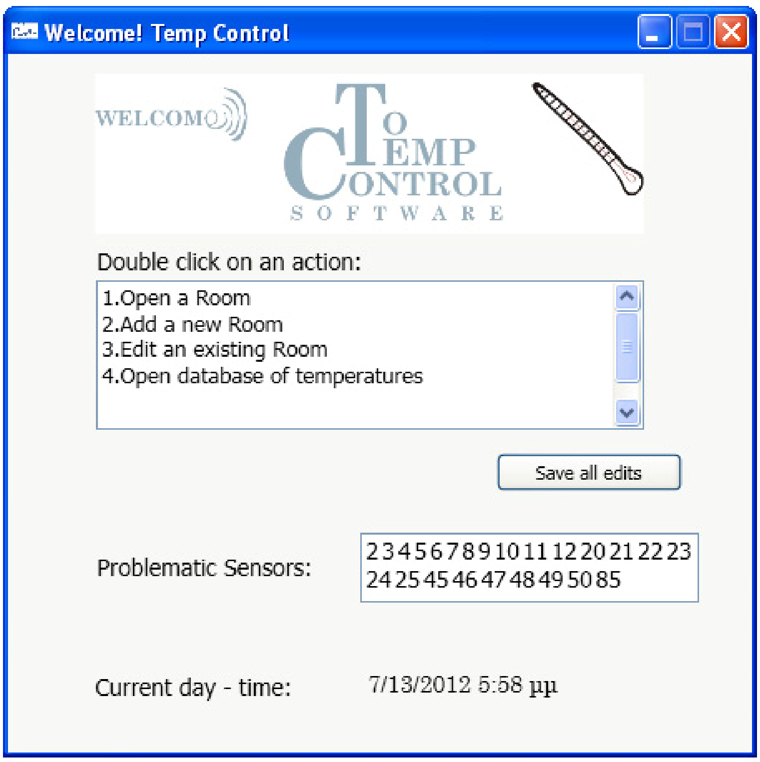
Σχήμα 3.4.1.3: Ειδοποίηση χρήστη για θερμοκρασία εκτός ορίου.

3.4.2 Η δεύτερη εφαρμογή εμφάνισης και επεξεργασίας δεδομένων “TempControl”

Η εφαρμογή TempControl παρέχει τις εξής δυνατότητες στο χρήστη:

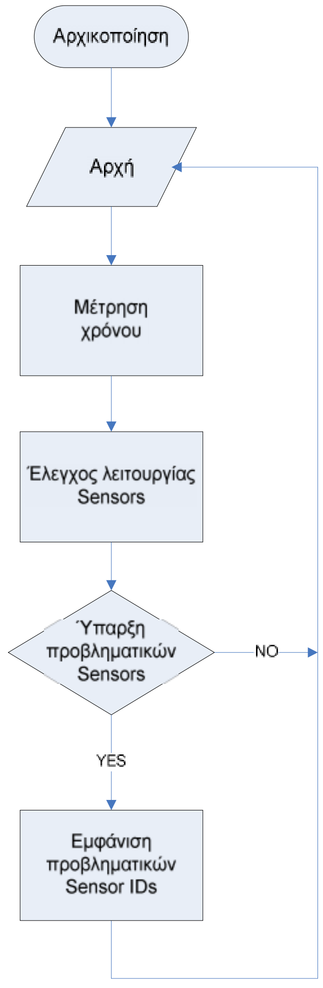
- Εμφάνιση ασθενών και των θερμομετρήσεών τους ανά θάλαμο.
- Εισαγωγή νέου θαλάμου και εισαγωγή στοιχείων για τον νέο θάλαμο (όνομα θαλάμου, ονοματεπώνυμα ασθενών, το ID της συσκευής sensor του κάθε ασθενή, το όριο θερμοκρασίας στο οποίο θα γίνεται ειδοποίηση του χρήστη και ο αριθμός κρεβατιού του).
- Επεξεργασία στα υπάρχοντα δεδομένα του κάθε θαλάμου.
- Εμφάνιση όλων των θερμομετρήσεων ανά ασθενή σε λίστα .
- Έλεγχος προβληματικών συσκευών Sensor.

Στο Σχ.3.4.2.1 απεικονίζεται το πρώτο παράθυρο που εμφανίζεται όταν ανοίξει ο χρήστης την εφαρμογή TempControl. Σε αυτό φαίνονται οι 4 λειτουργίες που αναφέρθηκαν παραπάνω. Ο χρήστης μπορεί να επιλέξει την κάθε λειτουργία κάνοντας διπλό κλίκ πάνω σε αυτή. Κάτω από το πλαίσιο των λειτουργιών υπάρχει το κουμπί “Save all edits” το οποίο αποθηκεύει όλες τις αλλαγές που έχει κάνει ο χρήστης στα δεδομένα του προγράμματος. Κάτω από το κουμπί της αποθήκευσης υπάρχει ένα πλαίσιο όπου αναγράφονται σε αυτό τα ID των συσκευών Sensor που έχουν εμφανίσει κάποιο τεχνικό πρόβλημα και δεν έχουν στείλει μέτρηση. Ο τρόπος με τον οποίο ελέγχονται οι συσκευές Sensor θα περιγραφεί παρακάτω. Τέλος για λόγους ενημέρωσης του χρήστη εμφανίζεται στο κάτω μέρος του παραθύρου η τρέχουσα ημερομηνία και ώρα.



Σχήμα 3.4.2.1: Αρχική οθόνη της εφαρμογής TempControl.

Ο τρόπος με τον οποίο γίνεται ο έλεγχος των συσκευών Sensor περιγράφεται στο Σχ. 3.4.2.2.



Σχήμα 3.4.2.2: Έλεγχος προβληματικών συσκευών Sensor

Η διαδικασία που εκτελείται είναι η εξής:

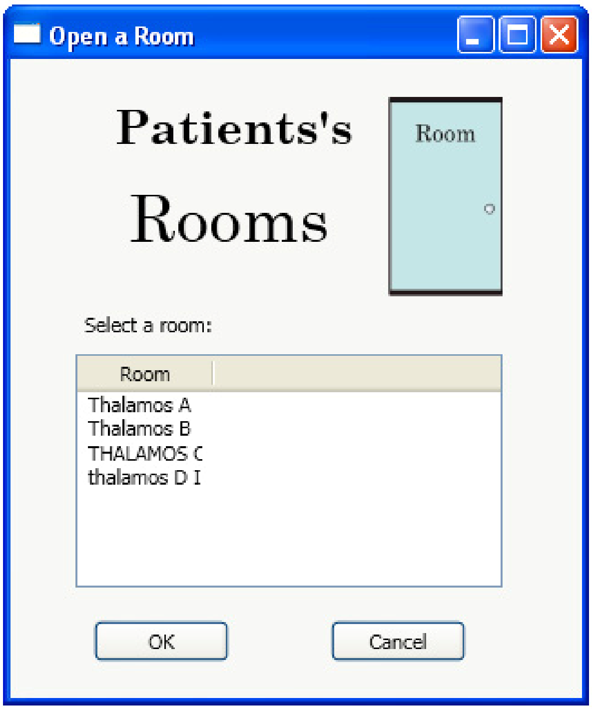
1. **Αρχικοποίηση:** Δηλώνονται οι κατάλληλες βιβλιοθήκες και αρχικοποιούνται οι μεταβλητές που θα χρησιμοποιηθούν καθώς και ένας μετρητής χρόνου.
2. **Μέτρηση χρόνου:** Ο μετρητής χρόνου μετράει ένα χρονικό διάστημα με διάρκεια λίγο μεγαλύτερη από αυτή που στέλνονται οι μετρήσεις. Αφού ολοκληρωθεί αυτή η μέτρηση συνεχίζεται η διαδικασία.
3. **Έλεγχος λειτουργίας συσκευών Sensors:** Στο βήμα αυτό γίνεται η εξής διαδικασία: Στο χρονικό διάστημα που έχει περάσει από την στιγμή που ξεκίνησε να μετράει ο μετρητής οι συσκευές Sensor αν δουλεύουν σωστά θα πρέπει να έχουν στείλει μέτρηση. Για να γίνει αυτός ο έλεγχος, αποθηκεύεται η τρέχουσα ώρα του συστήματος και αφαιρείται από αυτήν το χρονικό διάστημα με το οποίο στέλνονται οι μετρήσεις. Στη συνέχεια η νέα αυτή ώρα συγκρίνεται με την ώρα που αποθηκεύτηκε η τελευταία μέτρηση. Αν το αποτέλεσμα της σύγκρισής είναι ότι η επεξεργασμένη ώρα του συστήματος είναι πιο πρόσφατη από την ώρα της μέτρησης τότε η συγκεκριμένη συσκευή Sensor δεν έχει στείλει μέτρηση και πρέπει να ελεγχθεί από τον χρήστη. Η

διαδικασία αυτή εκτελείται για όλες τις τελευταίες μετρήσεις που έχουν στείλει όλες οι συσκευές Sensor.

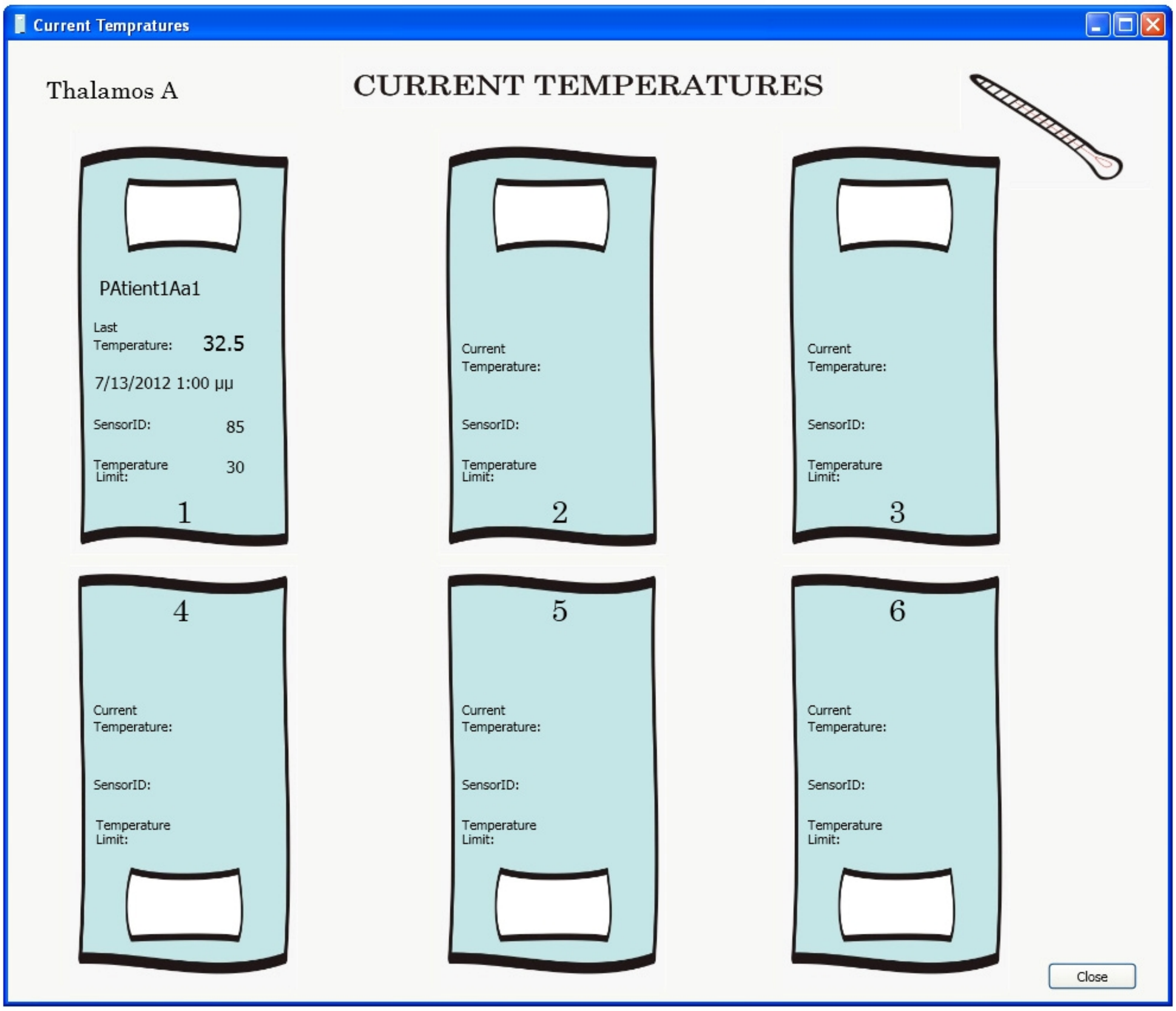
4. **Ύπαρξη προβληματικών Sensors:** Σε περίπτωση που υπάρχουν προβληματικές συσκευές Sensor γίνεται εμφάνιση των ID αυτών στο παράθυρο του προγράμματος, έτσι ώστε να ενημερωθεί ο χρήστης. Αφού γίνει η εμφάνιση των προβληματικών Sensor ID, η λειτουργία επιστρέφει στην αρχική της κατάσταση όπου η διαδικασία ξεκινάει πάλι από την αρχή. Σε περίπτωση που δεν υπάρχουν προβληματικές συσκευές Sensor, δεν εμφανίζεται τίποτα και η λειτουργία επιστρέφει στην αρχική της κατάσταση για να ξεκινήσει η διαδικασία από την αρχή.

3.4.2.1 Η πρώτη λειτουργία της εφαρμογής "TempControl"

Ο χρήστης επιλέγοντας την πρώτη λειτουργία της εφαρμογής εμφανίζεται το παράθυρο του Σχ. 3.4.2.1.1 όπου από αυτό επιλέγει ποιόν θάλαμο θα ανοίξει ώστε να βλέπει σε πραγματικό χρόνο τις θερμομετρήσεις των ασθενών Σχ. 3.4.2.1.2.



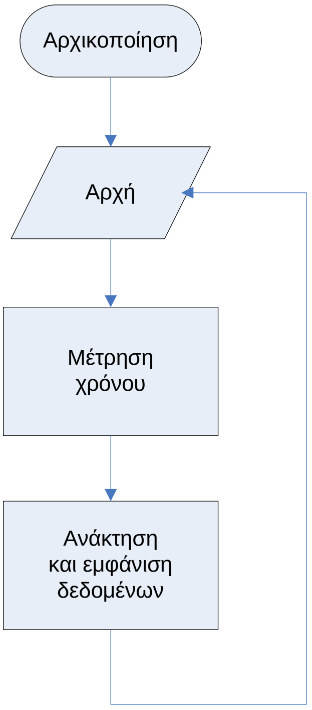
Σχήμα 3.4.2.1.1: Επιλογή θαλάμου προς εμφάνιση



Σχήμα 3.4.2.1.2: Προβολή θερμομετρήσεων ασθενών του θαλάμου Α σε πραγματικό χρόνο.

Στο Σχ. 3.4.2.1.2 εμφανίζονται οι θερμομετρήσεις των ασθενών για τον θάλαμο Α, σε πραγματικό χρόνο. Σε κάθε κρεβάτι εμφανίζεται το ονοματεπώνυμο του ασθενή, η τελευταία μέτρηση της θερμοκρασίας του, η ημερομηνία και ώρα της τελευταίας μέτρησης, το ID της συσκευής Sensor που είναι τοποθετημένη στον ασθενή και το όριο θερμοκρασίας για το οποίο θα έχουμε συναγερμό σε περίπτωση που ξεπεραστεί.

Ο τρόπος με τον οποίο γίνεται ανάκτηση των δεδομένων από την βάση SQL περιγράφεται στο διάγραμμα ροής του Σχ. 3.4.2.1.3.



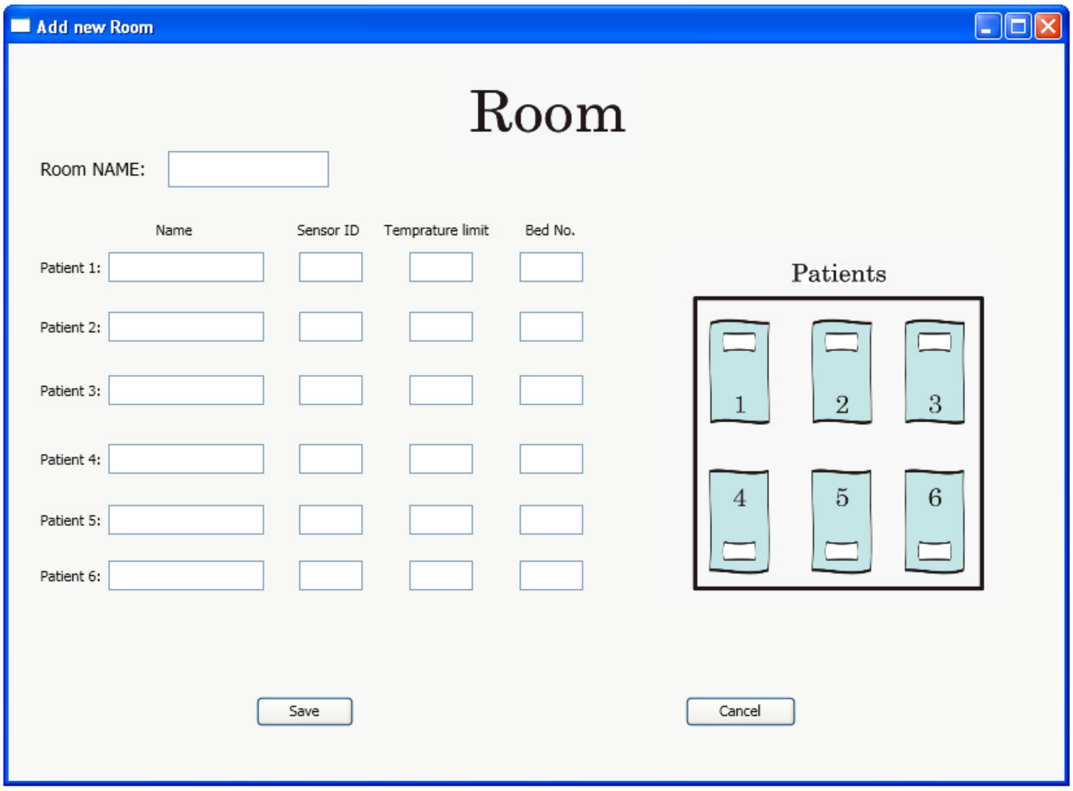
Σχήμα 3.4.2.1.3: Διαδικασία εμφάνισης θερμομετρήσεων

Η διαδικασία εμφάνισης των θερμομετρήσεων δουλεύει ως εξής:

1. **Αρχικοποίηση:** Στο προηγούμενο παράθυρο έχει γίνει ανάκτηση όλων των εγγραφών των θαλάμων. Ο θάλαμος που θα επιλεχθεί από το παράθυρο αυτό αποθηκεύεται προσωρινά σε μια μεταβλητή.
2. **Αρχή:** Από εδώ ξεκινάει η διαδικασία.
3. **Μέτρηση χρόνου:** Στο βήμα αυτό μετριέται χρόνος λίγο μεγαλύτερος από αυτόν που αποστέλλονται οι μετρήσεις, έτσι ώστε να έχει γίνει λήψη των μετρήσεων.
4. **Ανάκτηση και εμφάνιση δεδομένων:** Με βάση το όνομα του θαλάμου γίνεται ανάκτηση και εμφάνιση των δεδομένων που υπάρχουν για αυτόν τον θάλαμο (ονοματεπώνυμα ασθενών, τελευταίες θερμομετρήσεις, ημερομηνία και ώρα για αυτές, ID συσκευών Sensor και όρια θερμοκρασιών συναγερμού κάθε ασθενή). Αφού ολοκληρωθεί η εμφάνιση των δεδομένων η διαδικασία επιστρέφει στην αρχική της κατάσταση.

3.4.2.2 Η δεύτερη λειτουργία της εφαρμογής “TempControl”

Ο χρήστης επιλέγοντας την δεύτερη λειτουργία της εφαρμογής TempControl μπορεί να προσθέσει ένα νέο θάλαμο.

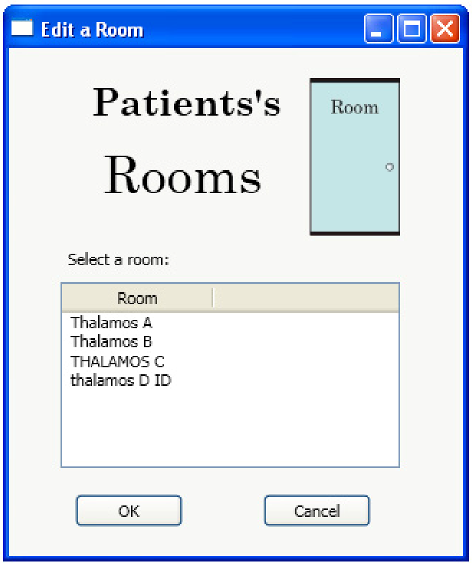


Σχήμα 3.4.2.2.1: Παράθυρο προσθήκης νέου θαλάμου

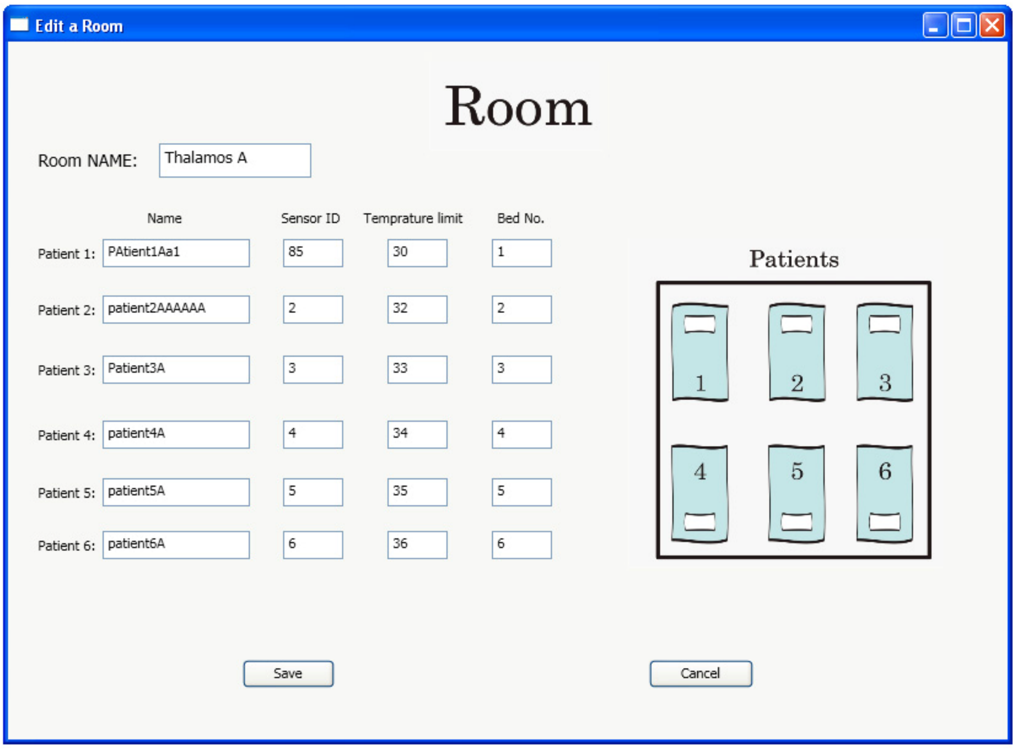
Στο Σχ. 3.4.2.2.1 εμφανίζονται τα πεδία όπου μπορεί να συμπληρώσει ο χρήστης σε περίπτωση που θέλει να προσθέσει στη βάση δεδομένων του προγράμματος έναν νέο θάλαμο. Τα στοιχεία που χρειάζεται να προσθέσει είναι το όνομα του θαλάμου, τα ονοματεπώνυμα των ασθενών, τα ID των συσκευών Sensor που έχει ο κάθε ασθενής, το όριο θερμοκρασίας για το οποίο ο χρήστης θέλει να ειδοποιείται, σε περίπτωση που ξεπεραστεί η θερμοκρασία του ασθενούς και ο αριθμός κρεβατιού του. Τα στοιχεία αποθηκεύονται προσωρινά στο πρόγραμμα όταν ο χρήστης πατήσει το κουμπί “Save” και έπειτα αποθηκεύονται στη βάση δεδομένων του προγράμματος όταν πατήσει το “Save all edits” στο αρχικό παράθυρο.

3.4.2.3 Η τρίτη λειτουργία της εφαρμογής "TempControl"

Ο χρήστης επιλέγοντας την Τρίτη επιλογή του προγράμματος TempControl έχει την δυνατότητα να επεξεργαστεί τα δεδομένα ενός θαλάμου. Πρώτα πρέπει να επιλέξει ποιο θάλαμο θα επεξεργαστεί Σχ. 3.4.2.3.1 και έπειτα μπορεί να επεξεργαστεί το θάλαμο που επέλεξε Σχ. 3.4.2.3.2.



Σχήμα 3.4.2.3.1: Παράθυρο επιλογής θαλάμου προς επεξεργασία.

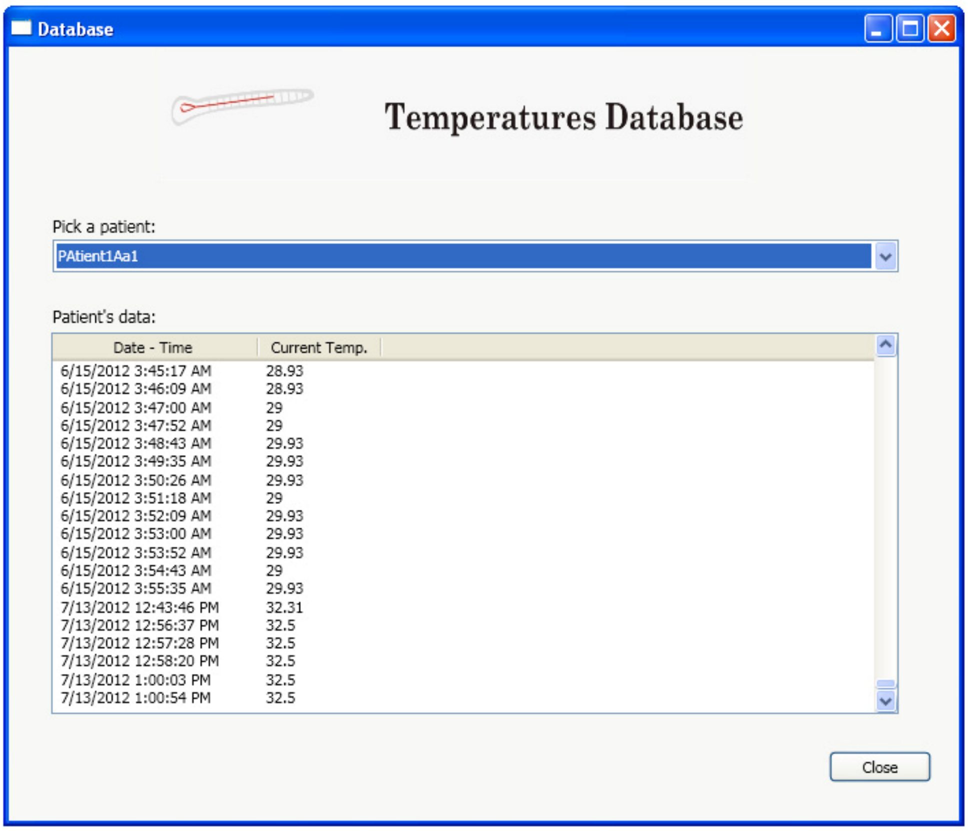


Σχήμα 3.4.2.3.2: Επεξεργασία δεδομένων θαλάμου

Στο Σχ. 3.4.2.3.2 φαίνονται τα δεδομένα του θαλάμου A τα οποία μπορεί να επεξεργαστεί ο χρήστης. Οι αλλαγές αποθηκεύονται προσωρινά στο πρόγραμμα όταν ο χρήστης πατήσει το κουμπί “Save” και στη συνέχεια αποθηκεύονται στη βάση δεδομένων του προγράμματος όταν πατήσει το κουμπί “Save all edits” στο αρχικό παράθυρο της εφαρμογής. Ο τρόπος με τον οποίο εμφανίζονται τα δεδομένα στο παράθυρο του Σχ. 3.4.2.3.2 είναι ακριβώς ίδιος με αυτόν που περιγράφεται στο διάγραμμα ροής στο Σχ. 3.4.2.1.3 χωρίς να χρησιμοποιείται η χρονική καθυστέρηση.

3.4.2.4 Η τέταρτη λειτουργία της εφαρμογής “TempControl”

Η τέταρτη λειτουργία της εφαρμογής TempControl είναι η εμφάνιση των θερμομετρήσεων ανά ασθενή σε μορφή λίστας.



Σχήμα 3.4.2.4.1: Εμφάνιση θερμομετρήσεων ασθενή σε μορφή λίστας.

Στο Σχ. 3.4.2.4.1 εμφανίζονται οι θερμομετρήσεις σε μορφή λίστας. Ο χρήστης επιλέγει τον ασθενή για τον οποίο θέλει να δει τις θερμομετρήσεις του από το ειδικό μενού. Έπειτα αυτόματα εμφανίζονται όλες οι θερμομετρήσεις του και η ημερομηνία και ώρα που λήφθηκε η κάθε μέτρηση.

ΚΕΦΑΛΑΙΟ 4

Η εξέλιξη

4.1 Η εξέλιξη του συστήματος

Το σύστημα λόγω του ότι έχει σχεδιαστεί και αναπτυχθεί από το μηδέν δίνει την δυνατότητα να τροποποιηθεί και να βελτιστοποιηθεί ανάλογα με τις απαιτήσεις του χρήστη. Θα μπορούσε να συνδεθούν και άλλου είδους αισθητήρες και να γίνεται μέτρηση σφυγμών, παλμών και ότι άλλο τίθεται ανάγκη. Επίσης θα μπορούσε να συνδεθεί GSM module και να γίνεται χειρισμός του συστήματος και επικοινωνία μέσω μηνύματος SMS. Επίσης θα μπορούσε να συνδεθεί GPS module και να υπάρχει συνεχής ενημέρωση σχετικά με την θέση του προς επιτήρηση ατόμου. Ένας άλλος τρόπος εξέλιξης είναι να συνδεθεί στην συσκευή Station κατάλληλο module ώστε να παρέχεται σε αυτήν η δυνατότητα internet και όλα τα δεδομένα να αποστέλλονται σε μια διαδικτυακή βάση. Αυτό θα καταστήσει το σύστημα αυτόνομο και δεν θα υπάρχει η ανάγκη για σύνδεση του, με υπολογιστή. Έπειτα μέσω του κατάλληλου λογισμικού που θα αναπτυχθεί, η βάση θα μπορεί να είναι προσπελάσιμη είτε από κάποιον φυλλομετρητή (browser) είτε από κάποια εφαρμογή για smartphone (app). Τέλος η δυνατότητα αυτή μπορεί να υπάρξει παρεμβαίνοντας μόνο στο ήδη υπάρχον πρόγραμμα αλλά σε αυτή την περίπτωση η χρήση του υπολογιστή θα είναι αναγκαστική.

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ

Βιβλιογραφία

1. Predko M., *Προγραμματίζοντας το μικροελεγκτή 8051*, Τζιόλα 1999
2. Σεφερίδης Β., *C για αρχαρίους*, Κλειδάριθμος 1995, Κλειδάριθμος 1995
3. Miles R., *C# Programming*, Department of Computer Science The University of Hull 2011
4. Watson B., *C# 4.0 HOW – TO Real Solutions for C# 4.0 Programmers,* SAMS 2010
5. Sharp J., *Visual C# 2010,* Microsoft Press 2010

Σύνδεσμοι

1. http://www.ti.com/
2. http://www.maxim-ic.com/
3. http://www.iar.com/
4. http://msdn.microsoft.com/
5. http://studentguru.gr/
6. http://www.wikipedia.org/