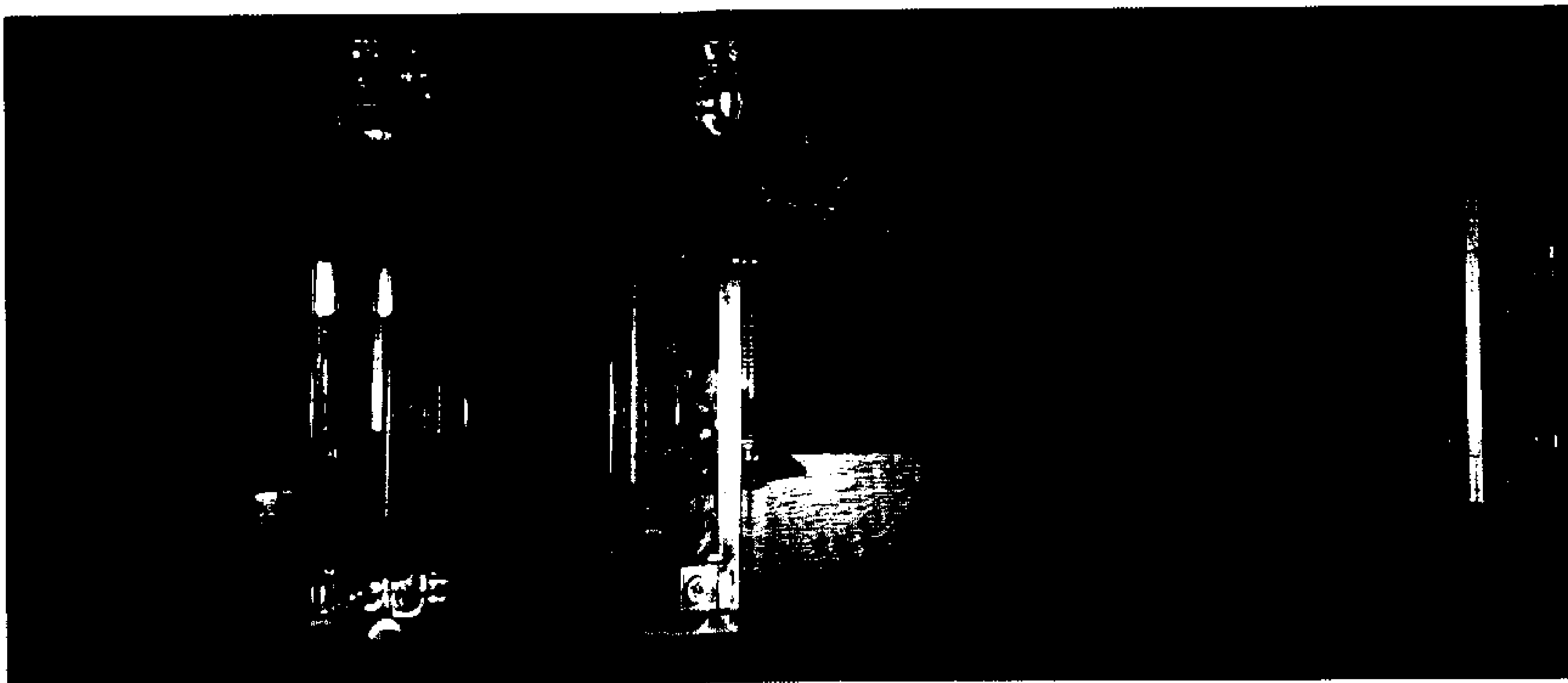


■ ■ ■ ΣΧΟΛΗ
■ ■ ΤΕΧΝΟΛΟΓΙΚΩΝ
■ ΕΦΑΡΜΟΓΩΝ

Τμήμα Ηλεκτρονικής

Πτυχιακή Εργασία

Ανάπτυξη φορητού συστήματος λήψης και επεξεργασίας ηλεκτρικών μετρήσεων βιολογικών δειγμάτων βασισμένο σε μικροελεγκτή ATxmega



Μόσχος Αναστάσιος

A.M.: 05492

Επιβλέπων: Δρ. Καλτσάς Γρηγόριος, Καθηγητής

Αθήνα, Απρίλιος 2013

Περίληψη

Στην παρούσα εργασία αναλύεται η διαδικασία σχεδιασμού και ανάπτυξης μιας φορητής υβριδικής διάταξης ελέγχου βιολογικών δειγμάτων. Η λειτουργία της βασίζεται στην τεχνολογία Bio-Electric Recognition Assay [Ref. 18-22], όπως αυτή έχει αναπτυχθεί από το τμήμα Γεωπονικής Βιοτεχνολογίας του Γεωπονικού Πανεπιστημίου Αθηνών και τα κύτταρα βιοαισθητήρων που αναπτύσσονται από την εταιρία SK Embio Diagnostics [Ref. 41-42].

Η διάταξη αποτελείται από ένα υποσύστημα που διαθέτει οκτώ ζεύγη ηλεκτροδίων από το οποίο τα ηλεκτρικά σήματα διαβιβάζονται στην κύρια μονάδα η οποία βασίζεται στον μικροελεγκτή Atmel AVR ATxmega64A3. Ο μικροελεγκτής αφού ψηφιοποιήσει τα σήματα εισόδου πραγματοποιεί φιλτράρισμα και επεξεργασία αυτών και προβάλλει τα αποτελέσματα σε οθόνη χαρακτήρων τύπου LCD 20 στηλών/4 γραμμών. Παράλληλα υπάρχει η δυνατότητα για αποστολή των αποτελεσμάτων σε Η/Υ μέσω σύνδεσης Universal Serial Bus η οποία διατίθεται στην πλειοψηφία των σημερινών Η/Υ.

Ο σχεδιασμός της διάταξης έχει γίνει με γνώμονα την εύκολη προσαρμογή καθώς οι απαιτήσεις της εφαρμογής αλλάζουν, ένας σημαντικός παράγοντας διότι η τεχνολογία BERA είναι ακόμα σε πειραματικό στάδιο, και την ευελιξία σε όλους τους τομείς. Ως εκ τούτου η συσκευή έχει σχετικά μικρές διαστάσεις και μάζα, πολλαπλές επιλογές τροφοδοσίας, δυνατότητα εύκολης αναβάθμισης και τροποποίησης του προγράμματος του μικροελεγκτή, μπορεί να λειτουργήσει αυτόνομα για μεγάλο χρονικό διάστημα, ενώ τέλος η αρθρωτή δομή της επιτρέπει την χρήση διαφόρων τύπων ηλεκτροδίων αρκεί να χρησιμοποιηθεί κατάλληλος μετατροπέας.

Abstract

This thesis presents the procedures involved in the design and development of a portable hybrid system for the evaluation of biological samples. Its operation relies on the Bio-Electric Recognition Assay technology [Ref. 18-22], developed by the department of Agricultural Biotechnology, Agricultural University of Athens while the corresponding bio-sensor cells were developed by SK Embio Diagnostics [Ref. 41-42].

The system is comprised of a subunit carrying eight pairs of electrodes, from which the electrical signals are transmitted to the main unit that is based on an Atmel ATxmega64A3 microcontroller.

The microcontroller digitizes the input signals and after filtering and processing the results are displayed on a 20 columns/4 rows character LCD. Optionally the measurements can be transmitted to a PC via a USB interface, which is available in the vast majority of modern computers.

The flexible system design can be easily adapted to the rapidly changing requirements of the application, which is an important parameter considering that the BERA technology is still experimental, and also offer versatility in every perspective. Therefore the device has relatively small dimensions and weight, can be powered by various sources and can operate for a long period of time in standalone mode, the microcontroller can be easily upgraded and reprogrammed and lastly the modular structure allows the utilization of various electrode types using a suitable adaptor.

Περιεχόμενα

Ευρετήριο εικόνων – πινάκων.....	7
Πρόλογος	8
1 Εισαγωγή	9
1.1 Βιοαισθητήρες	9
1.1.1 Η τεχνολογία BERA.....	10
1.2 Η προγενέστερη υλοποίηση.....	11
1.3 Η προτεινόμενη προσέγγιση	12
1.4 Η σειρά μικροελεγκτών ATxmega.....	13
1.4.1 Ο μικροελεγκτής ATxmega64A3.....	15
2 Σχεδιασμός - Υλοποίηση του hardware του συστήματος	16
2.1 Η κύρια πλακέτα	16
2.1.1 Κύκλωμα τροφοδοσίας	16
2.1.2 Κύκλωμα μικροελεγκτή	18
2.1.3 Κύκλωμα μεταφοράς δεδομένων σε Η/Υ	20
2.1.4 Είσοδος ηλεκτροδίων - βύσματα επέκτασης	21
2.1.5 Σχεδίαση PCB	22
2.2 Η πλακέτα διασύνδεσης με οθόνη χαρακτήρων LCD.....	23
2.3 Τα ηλεκτρόδια μέτρησης.....	24
2.3.1 Ηλεκτρόδια εγχαραγμένα σε PCB	25
2.3.2 Ηλεκτρόδια τύπου ράβδου	26
2.3.3 Ηλεκτρόδια εκτυπωμένα σε κεραμικό υπόστρωμα	29
2.4 Σχεδίαση & κατασκευή συσκευασίας.....	30
3 Ανάλυση του προγράμματος του μικροελεγκτή.....	32
3.1 Δήλωση επικεφαλίδων, μεταβλητών & σταθερών	32
3.2 Αρχικοποίηση περιφερειακών	34
3.2.1 Αρχικοποίηση πηγών χρονισμού συστήματος	34
3.2.2 Αρχικοποίηση πορτών I/O.....	36
3.2.3 Αρχικοποίηση χρονιστή/απαριθμητή (timer/counter) E0	40
3.2.4 Αρχικοποίηση σειριακής θύρας	43
3.2.5 Αρχικοποίηση A/D μετατροπέων	45
3.3 Λήψη μετρήσεων	50

3.4	Φιλτράρισμα μετρήσεων.....	50
3.5	Προσαρμογή δεδομένων.....	51
3.6	Προβολή αποτελεσμάτων.....	54
3.7	Εξοικονόμηση ενέργειας.....	55
4	Εργαστηριακές συσκευές που χρησιμοποιήθηκαν	56
4.1	Το αναπτυξιακό Atmel STK600	56
4.2	Πηγή Τάσης/Ρεύματος Keithley Sourcemeter 2612	58
4.3	Πολύμετρο πάγκου Keithley 2000 series.....	58
4.4	Παλμογράφος Rigol DS1052E.....	59
5	Το λογισμικό που χρησιμοποιήθηκε	60
5.1	HP InfoTech CodevisionAVR.....	60
5.2	Labcenter Proteus suite	61
5.3	Abacom Sprint Layout	62
5.4	National Instruments LabVIEW	63
6	Η τελική μορφή της διάταξης.....	66
6.1	Hardware – χαρακτηριστικά	66
6.2	Πρόγραμμα μικροελεγκτή.....	67
6.3	Αποτελέσματα μετρήσεων	69
7	Προτάσεις για μελλοντική βελτίωση της διάταξης	71
8	Σύνοψη – Ανασκόπηση.....	75
9	Αναφορές	77
Παραρτήματα		
A.	Quick start guide (English).....	81
B.	Κώδικας προγράμματος μικροελεγκτή.....	83
C.	Προγραμμα υπολογισμού παραμετρων Ε.Τ. σε MATLAB.....	108
D.	Σχηματικά κυκλωμάτων και σχέδια τυπωμένων πλακετών.....	109
E.	Σχέδια κουτιού	114

Ευρετήριο εικόνων – πινάκων

Εικόνα 1-I: Σχηματική απεικόνιση βιοαισθητήρα.....	9
Εικόνα 1-II: Κύτταρο τροποποιημένο ώστε να ανιχνεύει την παρουσία παθογόνου ιού.....	10
Εικόνα 1-III: Η προγενέστερη υλοποίηση βασισμένη στον ATmega2560 σε αναπτυξιακό STK600.....	12
Εικόνα 1-IV: Λειτουργικό διάγραμμα του συστήματος.....	13
Εικόνα 1-V: Τα περιφερειακά της σειράς ATxmega.....	15
Εικόνα 2-I: Κύκλωμα τροφοδοσίας.....	17
Εικόνα 2-II: Κύκλωμα μικροελεγκτή.....	18
Εικόνα 2-III: Κυμάτωση τάσης στον ακροδέκτη AV_{cc} του μικροελεγκτή.....	19
Εικόνα 2-IV: Κύκλωμα μεταφοράς δεδομένων σε Η/Υ.....	20
Εικόνα 2-V: Είσοδος ηλεκτροδίων – βύσματα επέκτασης.....	21
Εικόνα 2-VI: Σχέδιο PCB με ενδεικτικό διαχωρισμό σε διαμερίσματα.....	22
Εικόνα 2-VII: Άνω και κάτω όψη του κύριου PCB.....	23
Εικόνα 2-VIII: Κύκλωμα διασύνδεσης με οθόνη αφαριθμητικών χαρακτήρων τύπου LCD.....	24
Εικόνα 2-IX: Δισκίδιο μικροκοιλοτήτων (ELISA microtiter plate).....	25
Εικόνα 2-X: PCB που φέρει 8 ζεύγη ηλεκτροδίων εγχαραγμένα.....	25
Εικόνα 2-XI: Ηλεκτρόδια τύπου ράβδου από διάφορες οπτικές γωνίες.....	27
Εικόνα 2-XII: Εγκάρσια τομή εφαρμογής ηλεκτροδίων τυπου ραβδου σε μικροκοιλότητα.....	28
Εικόνα 2-XIII: Ηλεκτρόδια τύπου ράβδου προσαρμοσμένα σε δισκίδιο 8x12 μικροκοιλοτήτων.....	28
Εικόνα 2-XIV: Ηλεκτρόδια εκτυπωμένα σε κεραμικό υπόστρωμα.....	29
Εικόνα 2-XV: Προσαρμογέας ηλεκτροδίων Dropsens.....	29
Εικόνα 2-XVI: Ανάπτυγμα του κουτιού που σχεδιάστηκε.....	30
Εικόνα 2-XVII: Αποστάτες και σύνδεσμοι που σχεδιάστηκαν.....	31
Εικόνα 3-I: Μετρήσεις ενσωματωμένου ADC χωρίς προσαρμογή δεδομένων.....	52
Εικόνα 3-II: Μετρήσεις ενσωματωμένου ADC με προσαρμογή δεδομένων 2 περιοχών.....	53
Εικόνα 4-I: Το σύστημα διασύνδεσης του STK600.....	56
Εικόνα 4-II: Κάτοψη του αναπτυξιακού STK600 με τα επιμέρους τμήματα σημειωμένα.....	57
Εικόνα 4-III: Keithley Sourcemeter 2612.....	58
Εικόνα 4-IV: Keithley 2000 series.....	59
Εικόνα 4-V: Rigol DS1052E.....	59
Εικόνα 5-I: Το περιβάλλον εργασίας του CodevisionAVR.....	60
Εικόνα 5-II: Άποψη του CodewizardAVR για μικροελεγκτές ATxmega.....	61
Εικόνα 5-III: Το περιβάλλον εργασίας του Proteus ISIS.....	62
Εικόνα 5-IV: Περιβάλλον εργασίας του Sprint Layout.....	63
Εικόνα 5-V: Το περιβάλλον της εφαρμογής λήψης μετρήσεων μέσω LabVIEW.....	64
Εικόνα 5-VI: Μέρη του μπλοκ διαγράμματος της εφαρμογής λήψης & επεξεργασίας μετρήσεων.....	65
Εικόνα 6-I: Η τελική μορφή του hardware.....	66
Εικόνα 6-II: Διάγραμμα ροής προγράμματος μικροελεγκτή.....	67
Εικόνα 6-III: Απόκριση συστήματος σε ημιτονικό και τετραγωνικό σήμα.....	70
Εικόνα 6-IV: Ενδεικτικές μετρήσεις σε βιολογικά δείγματα.....	70
Πίνακας 4-A: Συνοπτικές προδιαγραφές Keithley 2612.....	58
Πίνακας 6-A: Χαρακτηριστικά συσκευής.....	67
Πίνακας 6-B: Ενδεικτικές μετρήσεις συστήματος.....	69

Πρόλογος

Η παρούσα εργασία αποτελεί το πρώτο στάδιο συνεργασίας του τμήματος Ηλεκτρονικής με το τμήμα Γεωπονικής Βιοτεχνολογίας του Γεωπονικού Πανεπιστημίου Αθηνών με στόχο την εξέλιξη σύγχρονων συστημάτων επιτόπιας διάγνωσης βασισμένων σε βιοαισθητήρες.

Θα ήθελα να ευχαριστήσω τον Δρ. Σ. Κίντζιο καθώς και το υπόλοιπο προσωπικό του εργαστηρίου Φυσιολογίας και Μορφολογίας Φυτών του Γ.Π.Α. για την απροβλημάτιστη συνεργασία.

Επίσης σημαντική ήταν η συμβολή της εταιρίας SK Embio Diagnostics η οποία χρηματοδότησε μερικώς την ανάπτυξη του συστήματος.

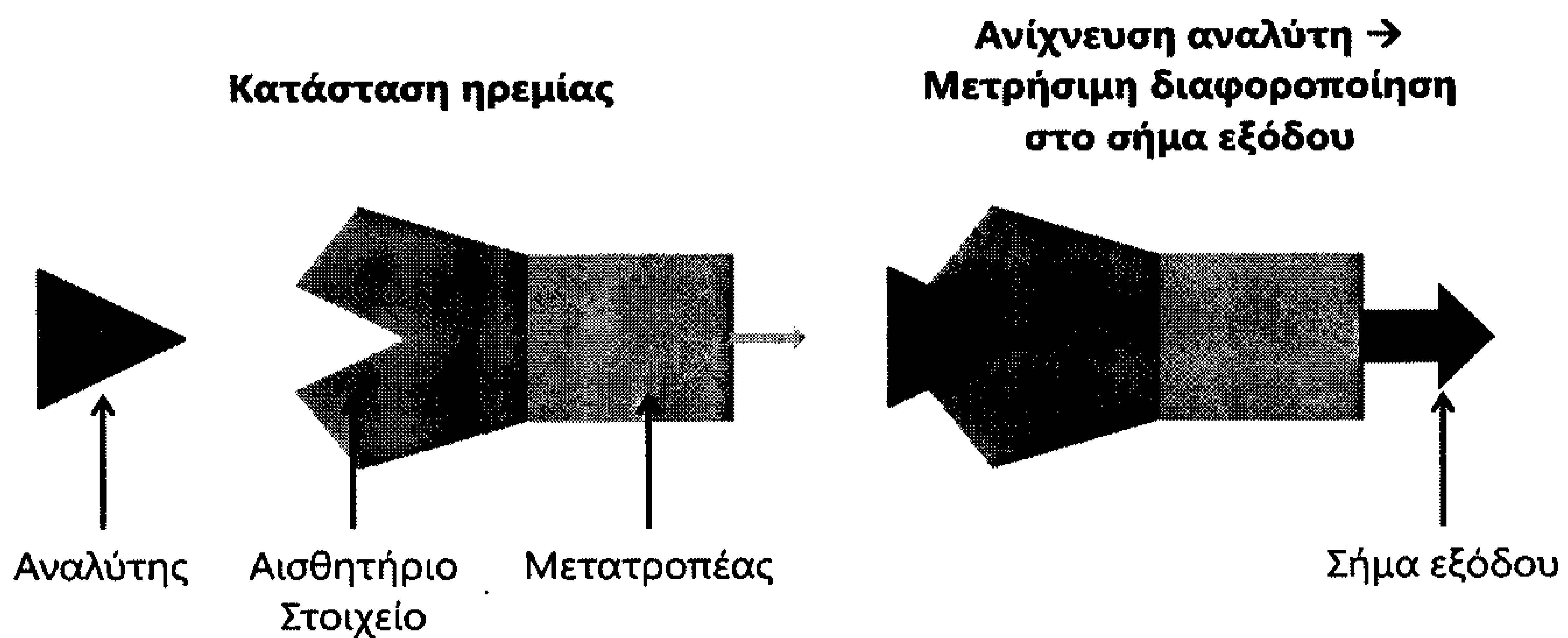
Τέλος θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους καθηγητές του τμήματος Ηλεκτρονικής για το υψηλό επίπεδο μόρφωσης που παρέχεται, παρότι οι συνθήκες δεν είναι πάντοτε ιδανικές. Ιδιαίτερως ευχαριστώ τους Δρ. Γ. Κουλούρα, Δρ. Α. Πετρόπουλο, Δρ. Ο. Τσακιρίδη και φυσικά τον επιβλέποντα Δρ. Γ. Καλτσά για την υποδειγματική καθοδήγηση, υποστήριξη και την βοήθεια που παρείχε καθ' όλη την πορεία της παρούσας εργασίας.

Μόσχος Αναστάσιος

1 Εισαγωγή

1.1 Βιοαισθητήρες

Βιοαισθητήρας σύμφωνα με έναν εκ των ευρέως αποδεκτών ορισμών είναι μια διάταξη που ενσωματώνει ένα βιολογικό αισθητήριο στοιχείο συνδεδεμένο άμεσα με ένα κατάλληλο μετατροπέα [Eggins, 1996]. Τα βιολογικά αισθητήρια ανιχνεύουν επιλεκτικά την συγκέντρωση ή την δραστηριότητα ενός αναλύτη προς ανίχνευση και μέσω του μετατροπέα παράγεται ένα μετρήσιμο σήμα στην έξοδο.



Εικόνα 1-1: Σχηματική απεικόνιση βιοαισθητήρα

Ο αναλύτης μπορεί να είναι κάποια τοξίνη, κάποιος παθογόνος μικροοργανισμός ή κάποια χημική ουσία όπως πχ φυτοφάρμακα και αντιβιοτικά.

Οι βιοαισθητήρες μπορούν να κατηγοριοποιηθούν αναλόγως του είδους του βιολογικού αισθητήριου που χρησιμοποιούν ή του τύπου μετατροπέα [Ref. 23].

Τα κύρια είδη αισθητήριων στοιχείων που χρησιμοποιούνται είναι τα ακόλουθα:

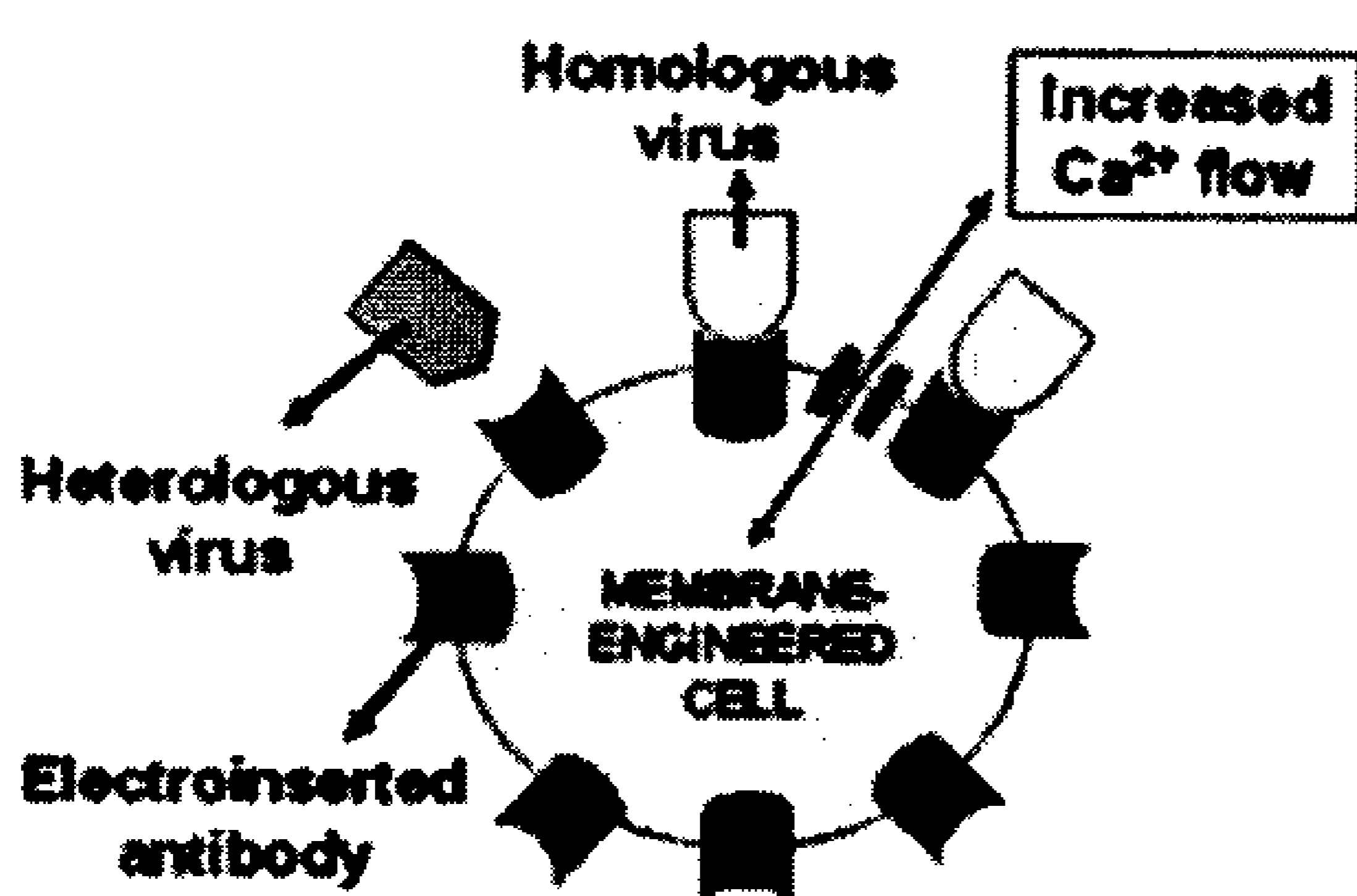
- Μικροοργανισμοί
- Αντισώματα
- Ένζυμα
- Κύτταρα
- Ιστοί
- Πρωτεΐνες
- Μembrάνες
- Κυτταρικά όργανα (μιτοχόνδρια κ.α.)
- Νουκλεϊκά οξέα

Με βάση το είδος του μετατροπέα διακρίνονται οι εξής κύριες κατηγορίες βιοαισθητήρων:

- ❖ **Ηλεκτροχημικοί:**
 - Αμπερομετρικοί: Το σήμα εξόδου είναι ηλεκτρικό ρεύμα.
 - Ποτενσιομετρικοί: Μετράται άμεσα ή έμμεσα η τάση που αναπτύσσεται.
 - Αγωγιμομετρικοί: Μεταβάλλεται η σύνθετη αντίσταση του αισθητήρα.
- ❖ **Οπτικοί:**
 - Χρωματομετρικοί: Μεταβάλλεται η απορρόφηση συγκεκριμένων μηκών κύματος ακτινοβολίας.
 - Φωτομετρικοί: Μετράται φωτοβολία ή φθορισμός.
- ❖ **Ακουστικοί:**
 - Πιεζοηλεκτρικοί: Μεταβάλλεται η συχνότητα ταλάντωσης ενός κρυσταλλικού ταλαντωτή λόγω πιέσεων στον κρύσταλλο.
 - Επιφανειακού κύματος: Μεταβάλλεται η ταχύτητα διάδοσης και η εξασθένηση ενός επιφανειακού ακουστικού κύματος.
- ❖ **Θερμιδομετρικοί:** Μεταβολές θερμοκρασίας από εξώθερμες αντιδράσεις.

Ενδεικτικά δημοφιλή εμπορικά προϊόντα που ενσωματώνουν βιοαισθητήρες είναι τα τεστ σακχάρου αίματος για άτομα που πάσχουν από διαβήτη, συσκευές τεστ εγκυμοσύνης, βιομηχανικές συσκευές έλεγχου τροφίμων κλπ.

1.1.1 Η τεχνολογία BERA



Εικόνα 1-Η: Κύτταρο τροποποιημένο κατάλληλα ώστε να ανιχνεύει επιλεκτικά την παρουσία συγκεκριμένου παθογόνου ιού

Η τεχνολογία Bio-Electric Recognition Assay αποτελεί μια σχετικά νέα μέθοδο έλεγχου βιολογικών δειγμάτων Ηλεκτροχημικού - Ποτενσιομετρικού τύπου η οποία χρησιμοποιεί μικρόβια ή κύτταρα ως αισθητήρια στοιχεία και σύμφωνα με υπάρχουσες ερευνητικές δημοσιεύσεις [Ref. 18-22] μπορεί να προσαρμοστεί σε ευρύ φάσμα εφαρμογών όπως την ανίχνευση παρουσίας φυτοφαρμάκων σε γεωργικά προϊόντα, την ανίχνευση παθογόνων ιών σε φυτικά ή ζωικά δείγματα κ.α. παρουσιάζοντας παράλληλα σημαντικά πλεονεκτήματα έναντι άλλων κλασικών μεθόδων όπως μείωση του απαιτούμενου χρόνου και κόστους ανά εξέταση.

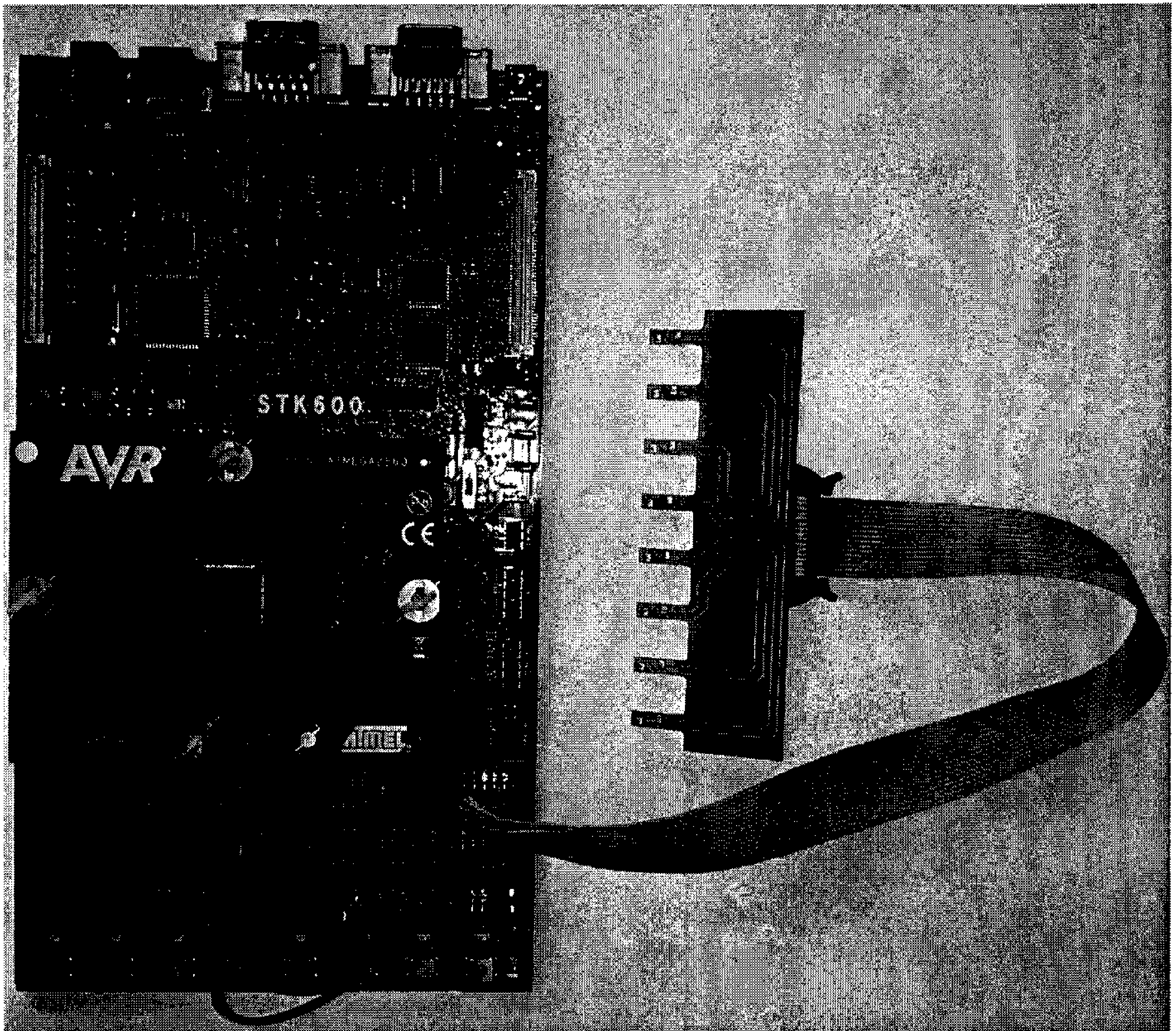
Για την συγκεκριμένη έκδοση της τεχνολογίας BERA που αξιοποιεί το σύστημα κατασκευάζονται από το εξειδικευμένο προσωπικό της Embio Diagnostics και του Τμήματος Βιοτεχνολογίας του Γεωπονικού Πανεπιστημίου Αθηνών κατάλληλα τροποποιημένα ζωντανά κύτταρα τα οποία με βάση την παρούσα θεωρία δρουν ως ευαίσθητοι βίο-αισθητήρες προσανατολισμένοι στην εκάστοτε εφαρμογή. Κατόπιν τα κύτταρα ακινητοποιούνται σε υγρό αδρανές μέσο και το διάλυμα διαχωρίζεται σε 2 φάσεις εκ των οποίων η μια περιέχει τα κύτταρα και η άλλη λειτουργεί ως σημείο αναφοράς. Καθώς τα ευαισθητοποιημένα κύτταρα αντιδρούν με το αντικείμενο προς ανίχνευση παράγεται μια ενδογενής τάση μεταξύ των δυο φάσεων του διαλύματος η οποία ανιχνεύεται με κατάλληλα τοποθετημένα ηλεκτρόδια.

1.2 Η προγενέστερη υλοποίηση

Στα πλαίσια παλαιότερης πτυχιακής εργασίας φοιτητών του τμήματος [Ref. 17] είχε αναπτυχθεί ένα σύστημα λήψης μετρήσεων βασισμένο στο αναπτυξιακό STK600 (Εικόνα 1-III) το οποίο χρησιμοποιεί τον μικροελεγκτή ATmega2560 για λήψη μετρήσεων με ηλεκτρόδια χαραγμένα σε PCB και μεταφέρει τα αποτελέσματα σε Η/Υ μέσω σειριακής θύρας RS232.

Όπως μπορεί να διαπιστωθεί η συγκεκριμένη λύση παρότι λειτουργική σε πρώτη φάση πάσχει από διάφορα προβλήματα που προκύπτουν από την φύση του STK600 και των χαρακτηριστικών του ATmega2560 όπως:

- Υψηλό κόστος
- Αυξημένες διαστάσεις
- Μη βελτιστοποιημένη σχεδίαση
- Ευαισθησία σε μετακινήσεις
- Το κύριο PCB είναι εκτεθειμένο
- Τα κανάλια μέτρησης δεν είναι πλήρως ανεξάρτητα
- Πραγματική ανάλυση ADC 9bit ($\approx 2,15\text{mV}$ χρησιμοποιώντας την εσωτερική τάση αναφοράς 1,1V) σε λειτουργία διαφορικών μετρήσεων
- Εξάρτηση από Η/Υ
- Μεταφορά δεδομένων μέσω σειριακής RS232 θύρας η οποία τείνει να εξαλειφθεί στους σύγχρονους Η/Υ
- Δυσκολία στον χειρισμό από μη εξειδικευμένο προσωπικό
- Μη βελτιστοποιημένα ηλεκτρόδια για την εφαρμογή

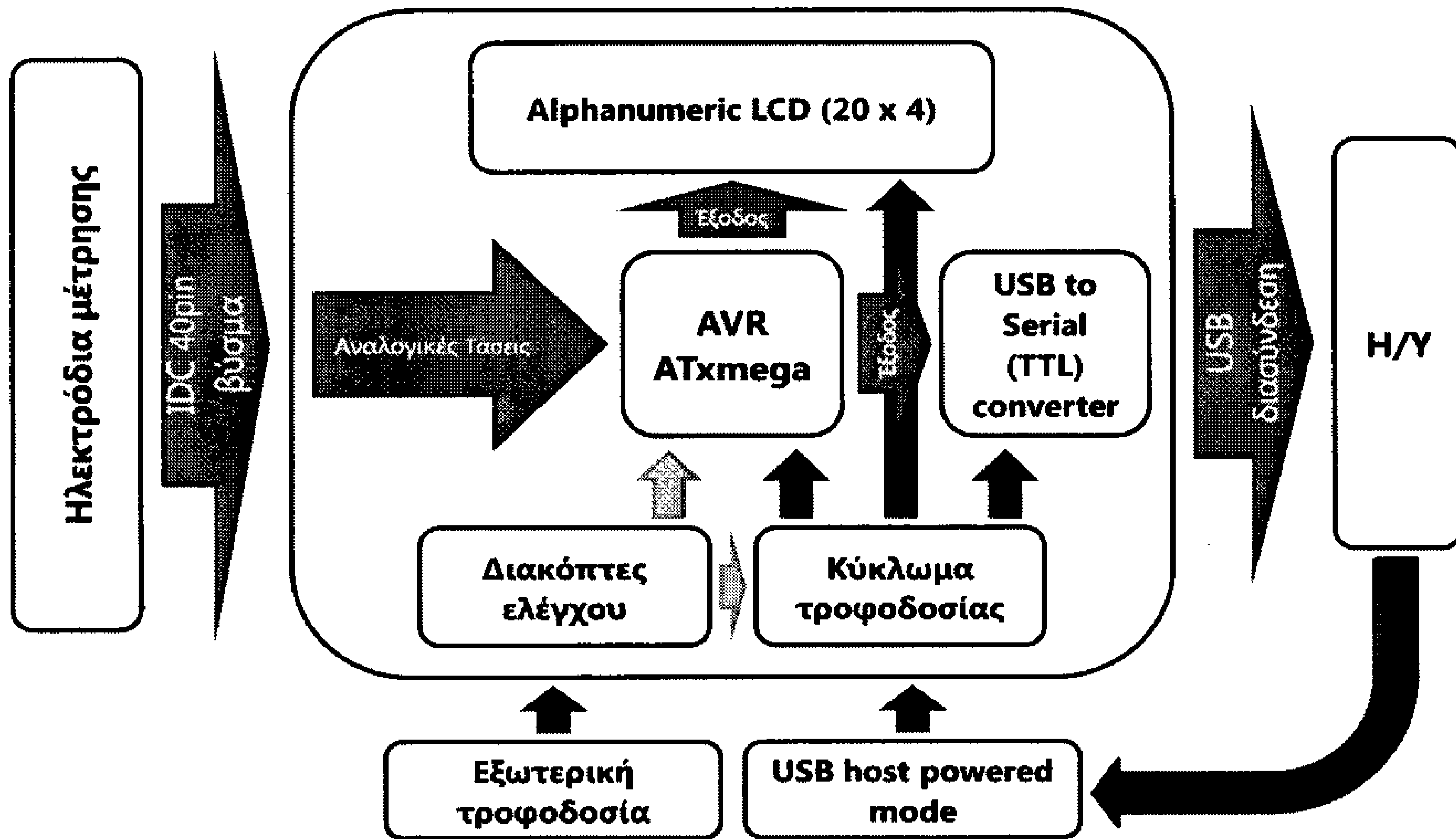


Εικόνα 1-III: Η προγενέστερη υλοποίηση, βασισμένη στον ATmega2560 σε αναπτυξιακό STK600

1.3 Η προτεινόμενη προσέγγιση

Στόχος της εργασίας είναι η σχεδίαση και υλοποίηση ενός προσαρμοσμένου συστήματος που θα εξαλείφει τα κύρια προβλήματα που υπήρχαν στο προαναφερθέν πρωτότυπο με αποτέλεσμα υψηλή φορητότητα, βελτιωμένη απόδοση, ευελιξία και ευχρηστία.

Η εικόνα 1-IV αποτελεί ένα συνοπτικό διάγραμμα των λειτουργιών που ενσωματώνει το σύστημα.



Εικόνα 1-IV: Λειτουργικό διάγραμμα του συστήματος

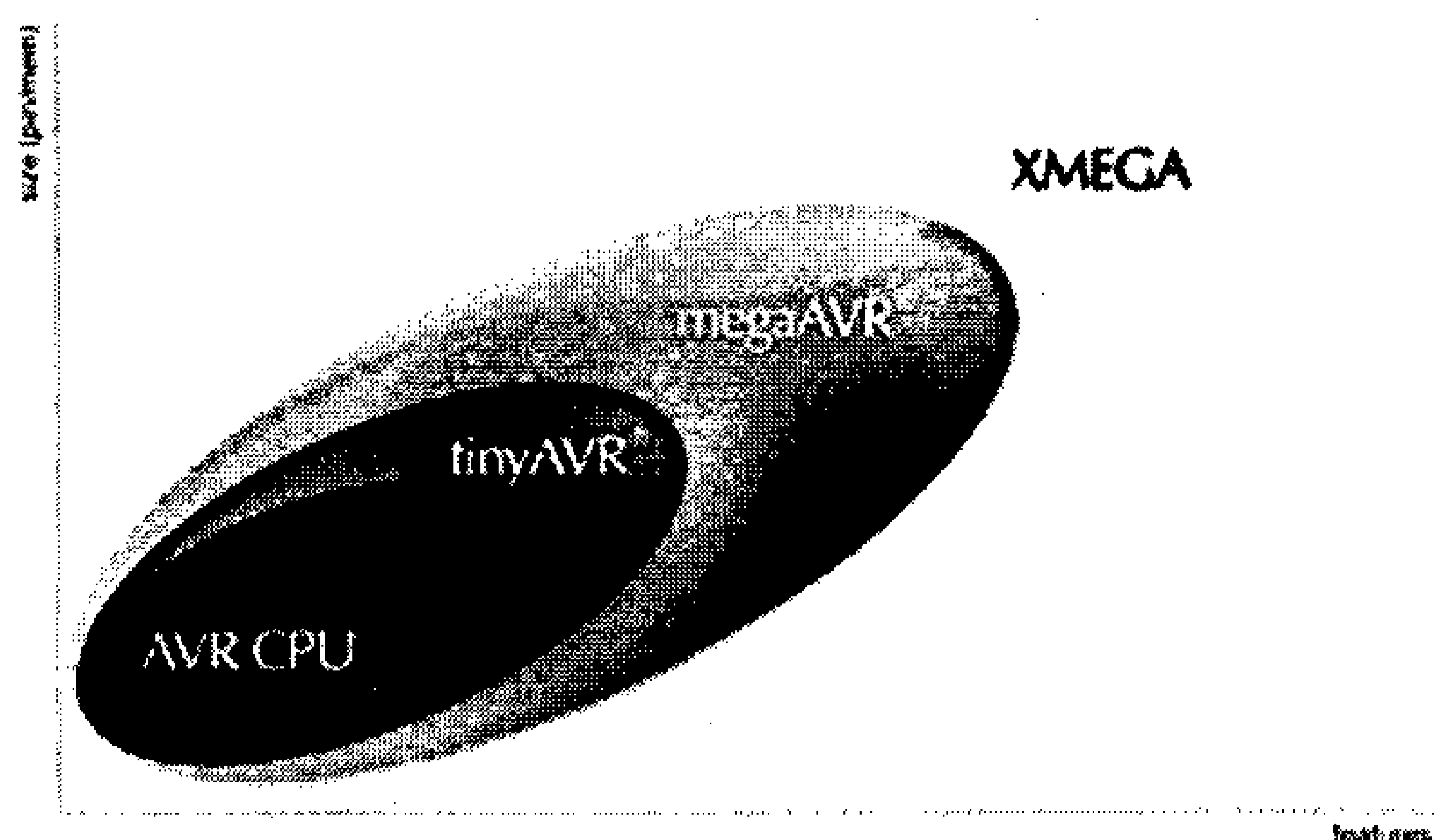
Στην καρδιά του συστήματος βρίσκεται ένας μικροελεγκτής Atmel της σειράς AVR ATxmega A3 series [Ref. 10-11] ο οποίος λαμβάνει τις αναλογικές τάσεις εισόδου από τα εξωτερικά τοποθετημένα ηλεκτρόδια μέσω ενός κοινού βύσματος τύπου IDC-40. Μετά την ψηφιοποίηση και επεξεργασία των μετρήσεων ο μικροελεγκτής προβάλλει στην ενσωματωμένη οθόνη υγρών κρυστάλλων τα αποτελέσματα ενώ παράλληλα εάν διατίθεται H/Y υπάρχει η δυνατότητα αποστολής αυτών σε σειριακό τερματικό μέσω USB σύνδεσης.

Τέλος η τροφοδοσία του κυκλώματος μπορεί να γίνει είτε απευθείας μέσω της USB θύρας ή από οποιαδήποτε εξωτερική πηγή AC/DC τάσης κατάλληλης στάθμης.

1.4 Η σειρά μικροελεγκτών ATxmega

Πρόκειται για μια σχετικά νέα σειρά μικροελεγκτών AVR η οποία παρουσιάστηκε από την Atmel στα τέλη του 2008 και αποτελεί εξέλιξη της υπάρχουσας σειράς 8bit AVR που διαχωρίζεται περαιτέρω σε ATtiny και ATmega αναλόγως του πλήθους των ενσωματωμένων χαρακτηριστικών.

Βασίζεται στην αρχιτεκτονική AVR Modified Harvard 8-bit RISC που χρησιμοποιείται και στις παλαιότερες σειρές ATtiny & ATmega αλλά προσφέρει επιπλέον προσθήκες για χειρισμό 16-bit διευθύνσεων και λειτουργιών καθώς και ένα πλήθος νέων ή βελτιωμένων περιφερειακών.

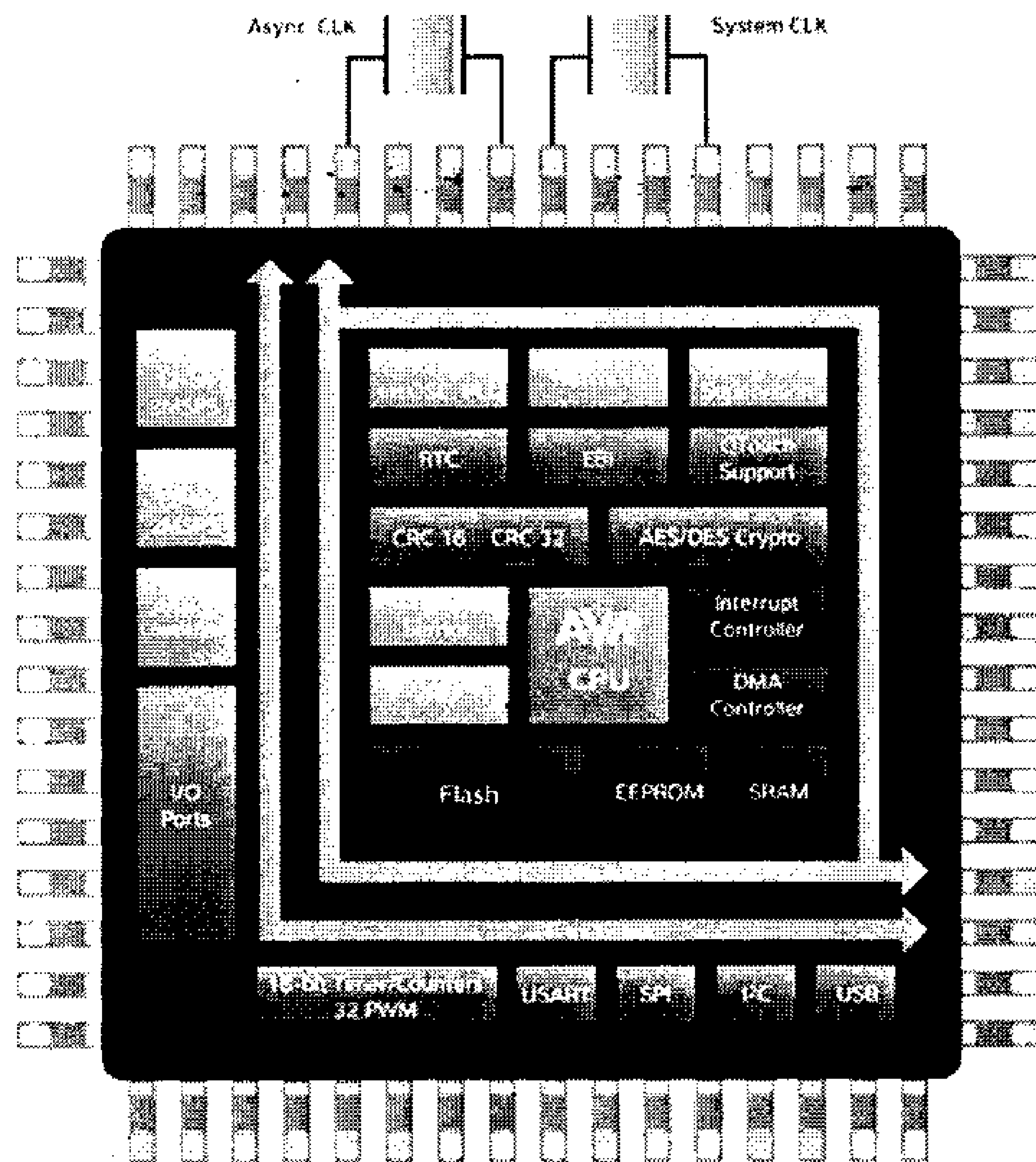


Εισαγωγή

Παρά την υψηλή ολοκλήρωση που χαρακτηρίζει την σειρά και την υψηλή απόδοση για 8-bit core μικροελεγκτές (έως 32 MIPS στα 32MHz, για τάση τροφοδοσίας $\geq 2.7V$) η κατανάλωση διατηρείται σε ιδιαίτερα χαμηλά επίπεδα καθιστώντας τους ιδανικούς για φορητές εφαρμογές.

Συνοπτικά τα χαρακτηριστικά της οικογένειας μικροελεγκτών ATxmega είναι:

- Τάση λειτουργίας 1.6V-3.6V με κατανάλωση έως 100nA
- Συχνότητα λειτουργίας 0-32MHz
- Έως 78 GPIO pins σε συσκευασία TQFP-100 ή CBGA
- Έως 384KB Flash, 32KB SRAM, 4KB EEPROM
- Ενσωματωμένοι 4 εσωτερικοί ταλαντωτές: 32MHz, 2MHz, 32.768KHz RTC & 32KHz εξαιρετικά χαμηλής κατανάλωσης καθώς και δυνατότητα λειτουργίας από ποικιλία εξωτερικών πηγών χρονισμού
- Ενσωματωμένο PLL 2-32x καθώς και prescalers 2-2048x
- Δυνατότητα εναλλαγής πηγών χρονισμού on the fly
- 32 8-bit καταχωρητές με απευθείας πρόσβαση στην ALU
- Εκτεταμένο ρεπερτόριο με υποστήριξη 138 εντολών
- 4-channel DMA controller καθώς και 8-channel event system για αυξημένη απόδοση και μείωση της απασχόλησης CPU
- Παραμετροποιήσιμη προτεραιότητα διακοπών από τον χρήστη
- Έως 8 16-bit timers με αυξημένες δυνατότητες και watchdog timer με ξεχωριστό ταλαντωτή χαμηλής κατανάλωσης
- Ενσωματωμένος Full-speed USB controller στις σειρές AU και C
- Ενσωματωμένος LCD controller πολύ χαμηλής κατανάλωσης στην σειρά B
- Έως 8 θύρες USART (με Fractional Baud Rate Generator), 4 SPI καθώς και 4 TWI modules
- Εξειδικευμένο bus για απευθείας διασύνδεση εξωτερικής μνήμης τύπου SRAM ή SDRAM
- 12-bit ADC με μέγιστη ταχύτητα 2MSPS, έως 2 hardware ADC με 8 κανάλια έκαστος, δυνατότητα διαφορικής μέτρησης με ρυθμιζόμενο κέρδος
- 12-bit DAC έως 2 καναλιών με μέγιστη ταχύτητα 1MSPS
- Έως 4 αναλογικοί συγκριτές με προχωρημένες λειτουργίες
- Ενσωματωμένη πηγή τάσης αναφοράς 1V καθώς και αισθητήρας θερμοκρασίας για αντιστάθμιση
- Ενσωματωμένη υπομονάδα κρυπτογράφησης με υποστήριξη AES και DES προτύπων
- Hardware CRC module με υποστήριξη CRC-16 και CRC-32



Εικόνα 1-V: Τα περιφερειακά της σειράς ATxmega

- Υποστήριξη Q-touch για την απλοποίηση εφαρμογών που χρησιμοποιούν χωρητικούς αισθητήρες επαφής
- Αυξημένη φορητότητα κώδικα μεταξύ συσκευών που ανήκουν στην οικογένεια ATxmega με αποτέλεσμα απλοποίηση στην ανάπτυξη εφαρμογών και εύκολη μετάβαση μεταξύ συσκευών
- Προγραμματισμός και debugging μέσω PDI ή JTAG interface
- Συμβατότητα με τα ήδη υπάρχοντα αναπτυξιακά εργαλεία (AVRISP MKII, AVR Dragon, JTAGICE MKII, JTAGICE3, STK600, AVRONE) καθώς και τους περισσότερους compilers για AVR.

1.4.1 Ο μικροελεγκτής ATxmega64A3

Ο συγκεκριμένος μικροελεγκτής συγκεντρώνει τα επιθυμητά για την εφαρμογή χαρακτηριστικά αφού μεταξύ άλλων διαθέτει 64KB Flash, 4KB SRAM, 1KB EEPROM, 50 ακροδέκτες εισόδου-εξόδου γενικής χρήσης, δυο 12-bit A/D μετατροπείς με 8 διαφορεικά κανάλια συνολικά, ένα 12-bit D/A μετατροπέα, 7 θύρες USART και 7 16-bit timers.

Παράλληλα έχει σχετικά μικρό κόστος αγοράς και διατίθεται σε TQFP-64 (14x14mm) συσκευασία με απόσταση ακροδεκτών 0.8mm η οποία οδηγεί σε απλουστευμένο σχεδιασμό PCB ενώ για την κόλληση του δεν απαιτείται εξειδικευμένος εξοπλισμός.

2 Σχεδιασμός - Υλοποίηση του hardware του συστήματος

2.1 Η κύρια πλακέτα

Σε αυτή την πλακέτα ενσωματώνονται οι κύριες λειτουργίες του συστήματος. Μπορεί να θεωρηθεί ότι αποτελείται από 4 υποσυστήματα: Το κύκλωμα τροφοδοσίας, τον μικροελεγκτή με τα απαραίτητα περιφερειακά εξαρτήματα, το κύκλωμα μετατροπής από UART σε USB interface και τέλος τα βύσματα εισόδου ηλεκτροδίων και επέκτασης.

Ο σχεδιασμός των σχηματικών έγινε σε περιβάλλον Labcenter Proteus ISIS λαμβάνοντας υπ' όψιν, στο μέτρο του δυνατού, όλες τις σχετικές πληροφορίες που βρέθηκαν σε datasheets, application notes και reference designs των εξαρτημάτων που χρησιμοποιήθηκαν ή και παραπλήσιων.

Όπου αυτό ήταν δυνατό έγινε εξομοίωση χρησιμοποιώντας την ενσωματωμένη έκδοση ProSPICE που διαθέτει το Proteus. Δυστυχώς μέχρι στιγμής δεν υπάρχει κάποιο πρόγραμμα σχεδίασης/εξομοίωσης που να διαθέτει πλήρως λειτουργικά μοντέλα για το σύνολο των εξαρτημάτων που χρησιμοποιήθηκαν, οπότε δεν υπήρχε η δυνατότητα για εξομοίωση της λειτουργίας του σχεδίου ως συνόλου.

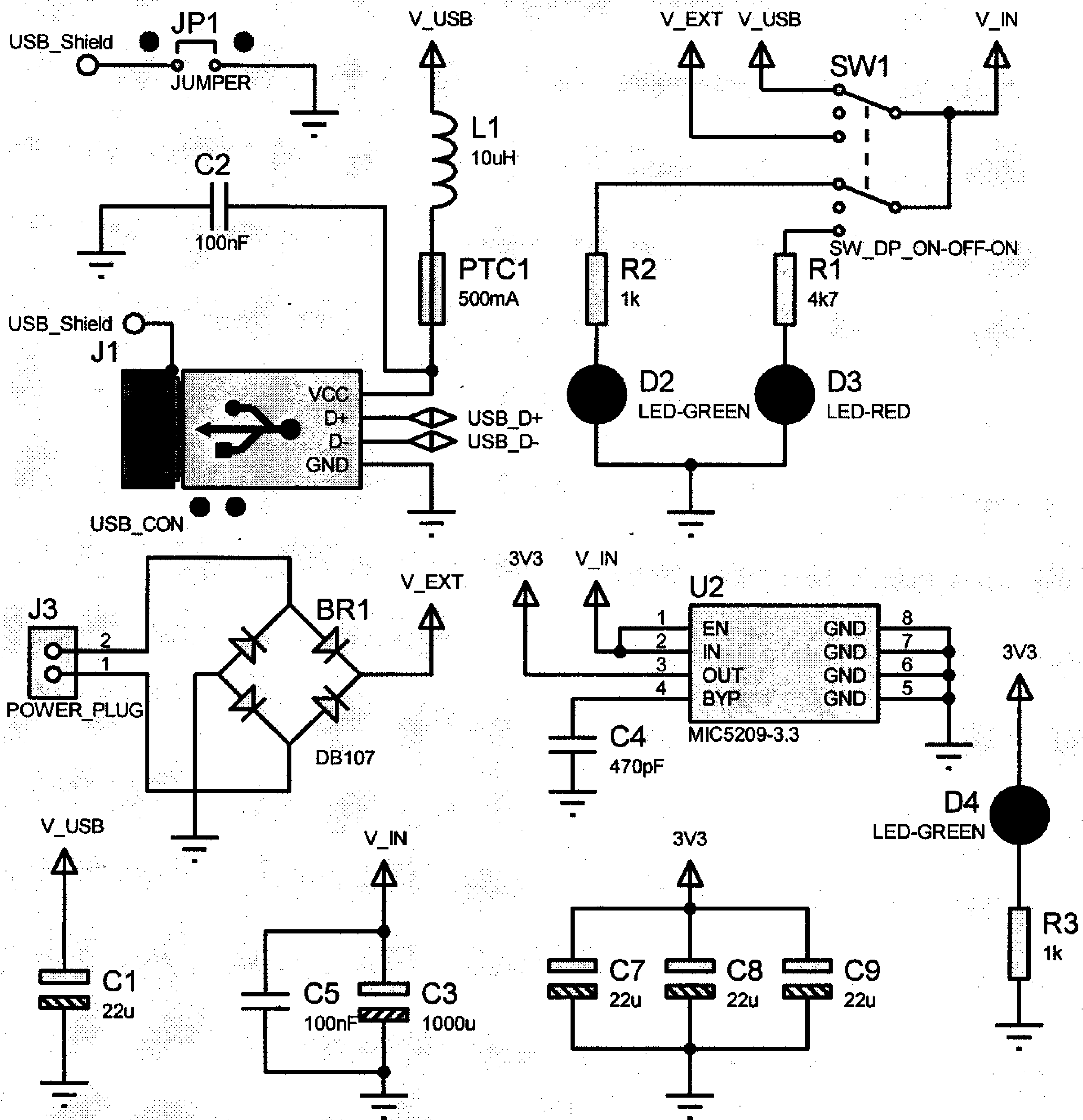
Στις περιπτώσεις που δεν βρέθηκε σχετικό part στις βιβλιοθήκες του προγράμματος τροποποιήθηκε κάποιο παρόμοιο ή σχεδιάστηκε εξ αρχής νέο part σύμφωνα με το εκάστοτε datasheet.

2.1.1 Κύκλωμα τροφοδοσίας

Το κύκλωμα τροφοδοσίας του συστήματος βασίζεται στον σταθεροποιητή MIC5209-3.3 της Micrel [Ref. 14]. Ο συγκεκριμένος σταθεροποιητής προσφέρει πολύ καλά χαρακτηριστικά με σημαντικά για την εφαρμογή πλεονεκτήματα έναντι άλλων συνηθέστερων όπως πχ LD1117.

Συγκεκριμένα έχει πολύ χαμηλό dropout voltage (500mV max) επιτρέποντας έτσι την χρήση της συσκευής με τροφοδοσία από την θύρα USB, όπου η τάση επιτρέπεται βάσει των προδιαγραφών ATX να φτάσει έως 4,75V [Ref. 9] ή ακόμα και από μια single-cell μπαταρία Li-ion με 3,7V ονομαστική τάση εάν παρακαμφθεί η γέφυρα διόδων BR1. Παρέχονται οι συνηθισμένες προστασίες ρεύματος, θερμοκρασίας και αντίστροφης πολικότητας. Παράλληλα διακρίνεται για την ποιότητα της τάσης εξόδου με σταθεροποίηση <1%, πολύ χαμηλή στάθμη θορύβου, χαμηλό θερμοκρασιακό συντελεστή και σταθερότητα σε χαμηλά φορτία. Τέλος έχει χαμηλό ρεύμα ηρεμίας συμβάλλοντας έτσι στην διατήρηση της συνολικής κατανάλωσης σε χαμηλά επίπεδα.

Σχεδιασμός - Υλοποίηση του hardware
του συστήματος



Εικόνα 2-1: Κύκλωμα τροφοδοσίας

Στην εικόνα 2-1 φαίνεται το κύκλωμα τροφοδοσίας του συστήματος. Η τροφοδοσία παρέχεται είτε από την θύρα USB του συνδεδεμένου Η/Υ μέσω του βύσματος J1 είτε από εξωτερικό τροφοδοτικό AC/DC τάσεως ή μπαταρία μέσω του βύσματος J3 (τυπικό DC jack 2,1mm).

Στην περίπτωση σύνδεσης με Η/Υ παρέχεται η δυνατότητα για σύνδεση της γείωσης του συστήματος με την γείωση του Η/Υ και κατ' επέκταση με την γείωση του δικτύου της ΔΕΗ μέσω του βραχυκυκλωτήρα JP1. Επίσης παρέχεται περιορισμός ρεύματος στα 500mA μέσω του resettable fuse PTC1 για προστασία του διασυνδεδεμένου Η/Υ καθώς και ένα επιπλέον φίλτρο αποτελούμενο από τα στοιχεία L1,C1,C2.

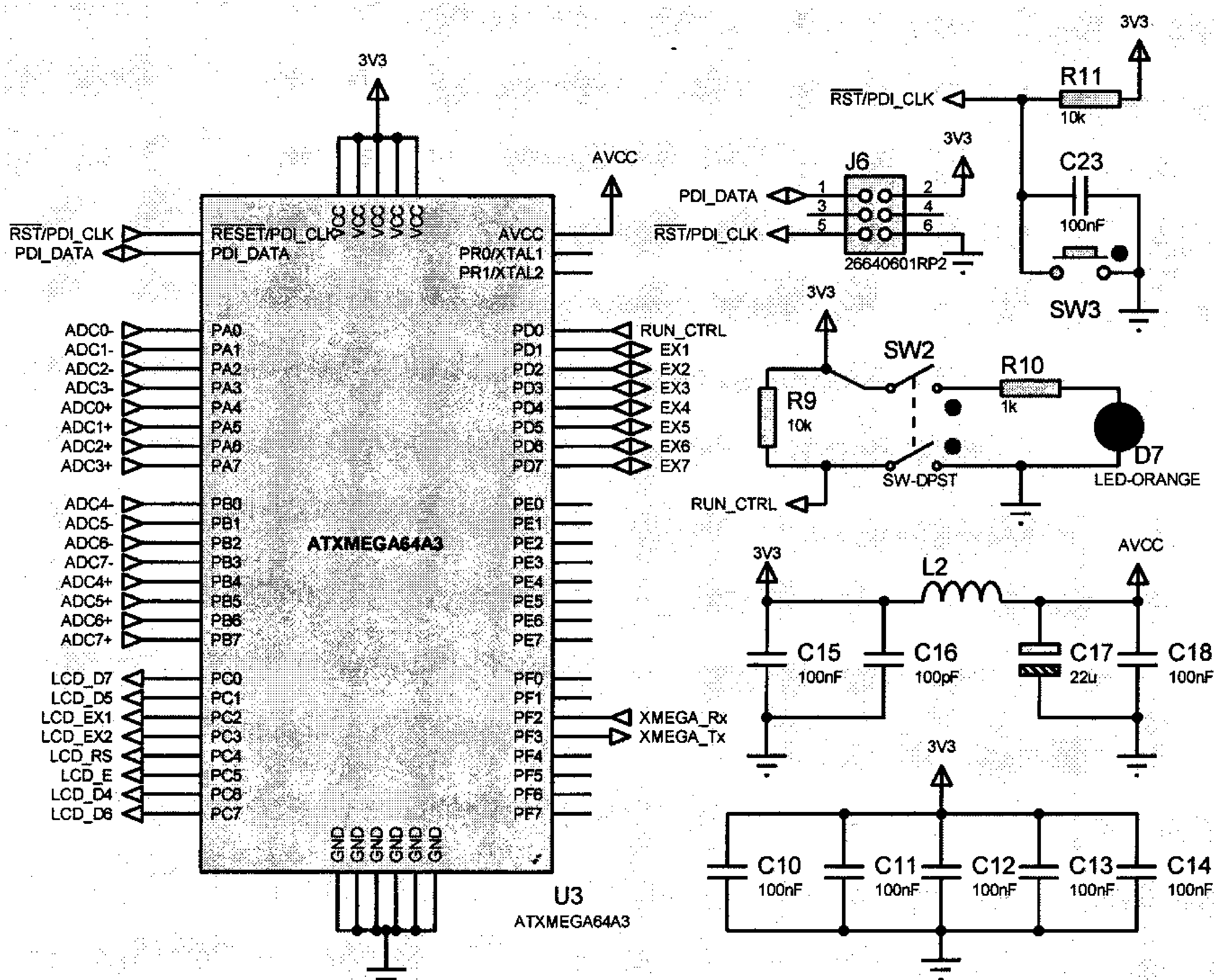
Σχεδιασμός - Υλοποίηση του hardware του συστήματος

Στην περίπτωση εξωτερικής τροφοδοσίας η γέφυρα διόδων BR1 ανορθώνει την τάση εισόδου παρέχοντας έτσι την δυνατότητα για τροφοδοσία από AC πηγή και προστασία ανάποδης πολικότητας στην περίπτωση DC πηγής.

Ο διπλός on-off-on διακόπτης SW1 επιλέγει την πηγή τροφοδοσίας. Στην θέση 1 το σύστημα τροφοδοτείται από την θύρα USB (V_USB) και φωτοβολεί αντίστοιχα με πράσινο χρώμα το ενδεικτικό LED (D2). Στην θέση 2 το σύστημα είναι εκτός λειτουργίας. Τέλος στην θέση 3 το σύστημα τροφοδοτείται από εξωτερική πηγή (V_EXT) και φωτοβολεί αντίστοιχα με κόκκινο χρώμα το ενδεικτικό LED (D3). Η σύνδεση του διακόπτη και των led με την πλακέτα γίνεται χρησιμοποιώντας ένα 4x1 polarized header.

Η τάση εισόδου (V_IN) οδηγείται στον σταθεροποιητή MIC5209 (U2) αφού φιλτραριστεί από τους πυκνωτές C3,C5 και στην έξοδο του σταθεροποιητή έχουμε την επιθυμητή τάση τροφοδοσίας των 3,3V. Το LED D4 (πράσινο) φανερώνει την λειτουργία του συστήματος. Οι πυκνωτές C7-C9 συνδέονται στην έξοδο του σταθεροποιητή για περαιτέρω φιλτράρισμα της τάσης εξόδου.

2.1.2 Κύκλωμα μικροελεγκτή



Εικόνα 2-II: Κύκλωμα μικροελεγκτή

Σχεδιασμός - Υλοποίηση του hardware του συστήματος

Στην εικόνα 2-II διακρίνονται οι διασυνδέσεις του μικροελεγκτή καθώς και τα απαραίτητα για την λειτουργία του περιφερειακά εξαρτήματα.

Στις πόρτες A & B συνδέονται τα σήματα προς μέτρηση αξιοποιώντας έτσι τους 2 ενσωματωμένους A/D μετατροπείς και παρέχοντας 4+4=8 κανάλια διαφορικής μέτρησης.

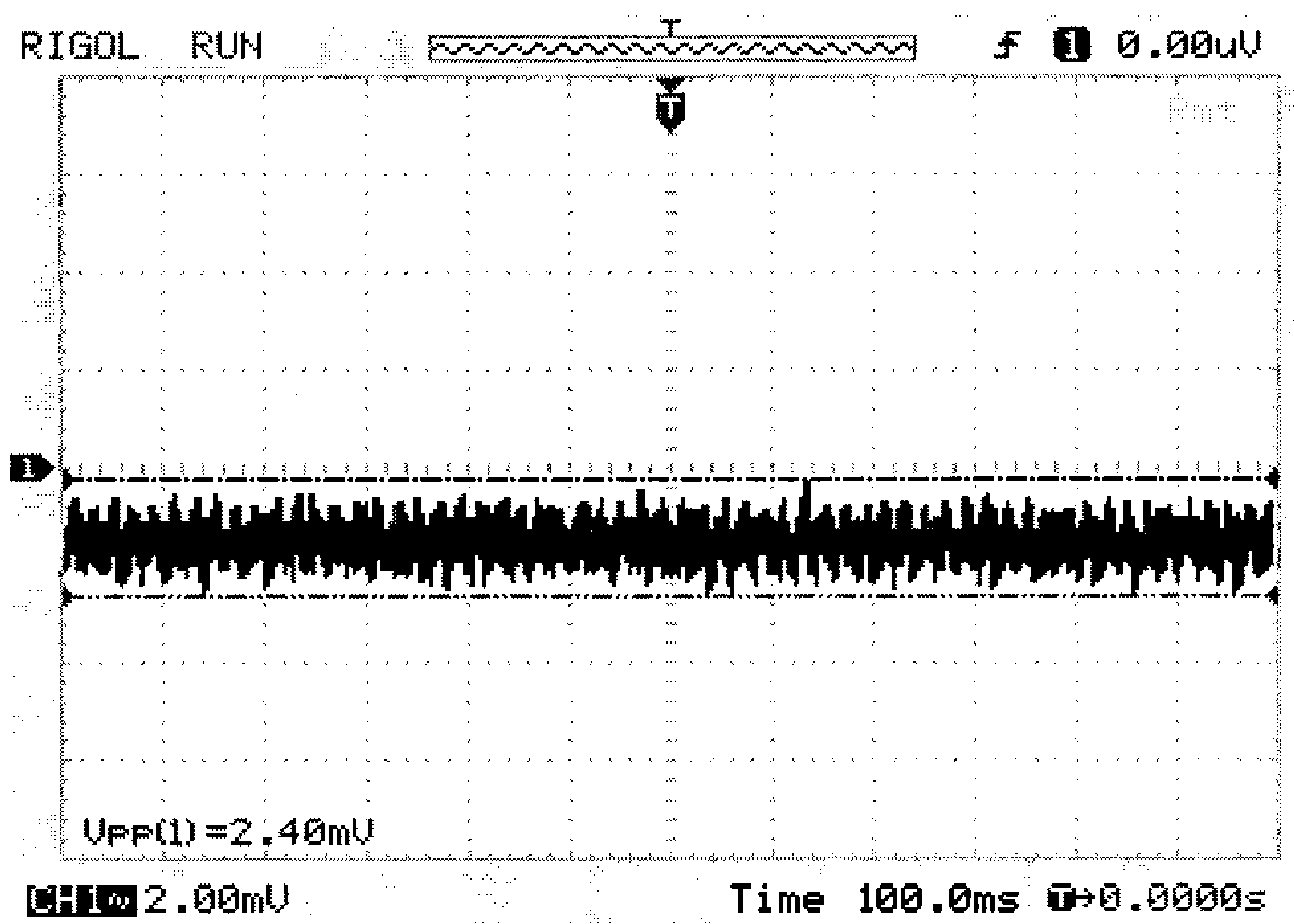
Στην πόρτα C συνδέονται τα σήματα ελέγχου και δεδομένων της οθόνης LCD (συμβατή με HD44780), αφήνοντας 2 επιπλέον I/O (C.2 & C.3) για πιθανή μελλοντική χρήση.

Το 1^ο bit της πόρτας D χρησιμοποιείται ως είσοδος για τον έλεγχο της ροής του προγράμματος. Αν ο διπλός διακόπτης SW2 είναι κλειστός το D.1 γειώνεται ενώ παράλληλα φωτοβολεί το ενδεικτικό LED D7 (πορτοκαλί), αλλιώς το D.1 αποκτά υψηλό δυναμικό μέσω της pull-up R9. Τα υπόλοιπα bit της πόρτας είναι ελεύθερα για μελλοντική χρήση.

Στα bit 2,3 της πόρτας F συνδέονται αντίστοιχα τα Receive και Transmit της σειριακής επικοινωνίας (υλοποιείται μέσω της USARTF.0), τα υπόλοιπα bits διατηρούνται ασύνδετα.

Διατίθεται βύσμα σύνδεσης για In System Programming χρησιμοποιώντας το PDI interface και ο συνηθισμένος συνδυασμός pull-up αντιστάτη (R11) και πυκνωτή απόζευξης (C23) για την απόρριψη θορύβου στην γραμμή του Reset όπως προτείνεται από το datasheet. Υπάρχει πρόβλεψη για τοποθέτηση προαιρετικού μπουτόν για την πρόκληση reset στον μικροελεγκτή (SW3).

Τέλος προκειμένου να απομονώνεται και να φιλτράρεται κατά το δυνατόν η αναλογική τροφοδοσία του μικροελεγκτή έχει χρησιμοποιηθεί το δικτύωμα αποτελούμενο από τους πυκνωτές C15-C18 και το μικρό-τσोक φερρίτη L2 με τελικό αποτέλεσμα κυμάτωση <math> < 5\text{mV}_{\text{p-p}} </math> (Εικόνα 2-III). Επίσης κοντά σε κάθε είσοδο τροφοδοσίας του μικροελεγκτή έχει συνδεθεί ένας πυκνωτής απόζευξης με τιμή 100nF (C10-C14).



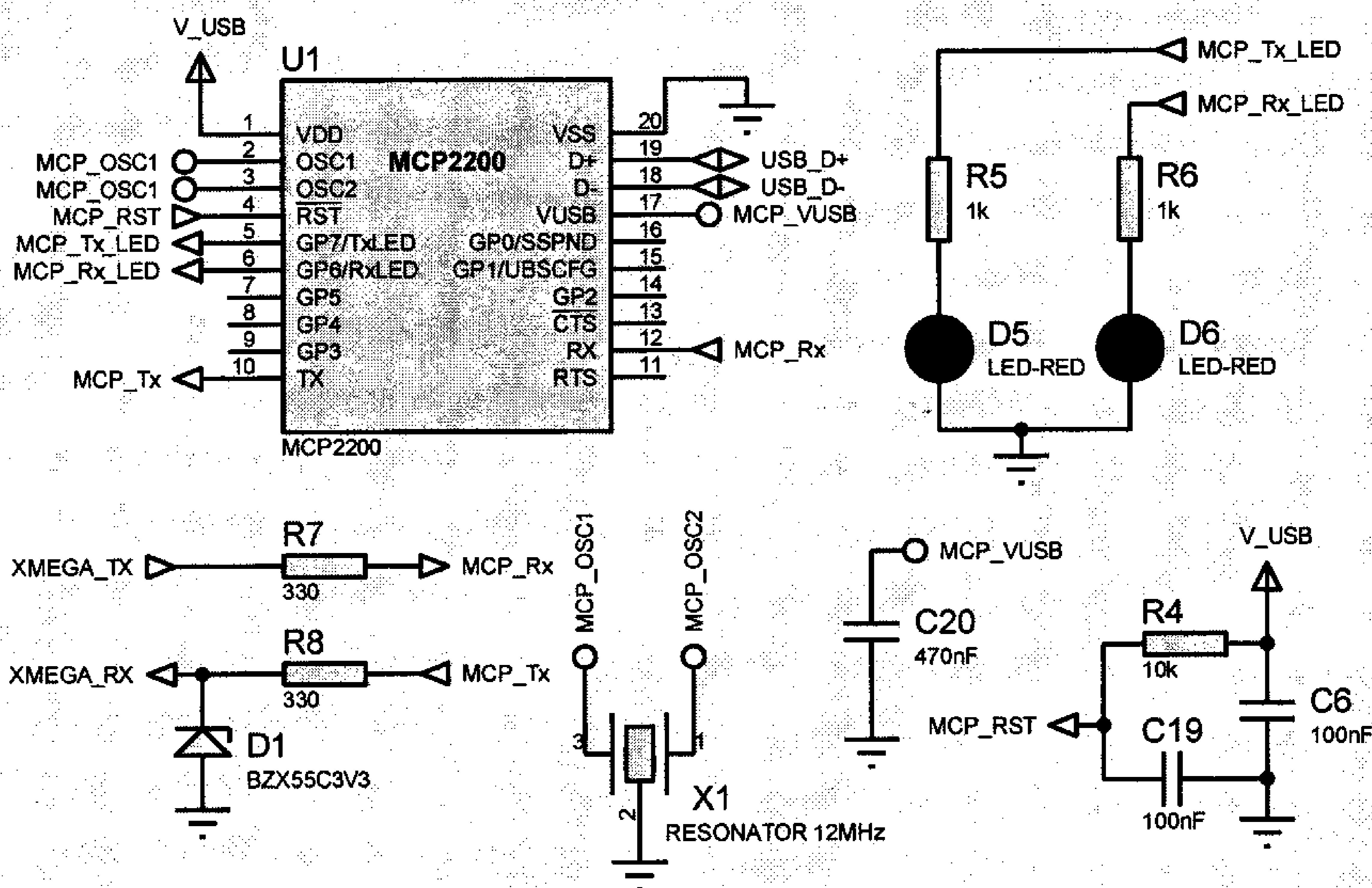
Εικόνα 2-III: Κυμάτωση τάσης στον ακροδέκτη AV_{cc} του μικροελεγκτή

Σχεδιασμός - Υλοποίηση του hardware του συστήματος

Για τον χρονισμό του μικροελεγκτή χρησιμοποιείται ο εσωτερικός ταλαντωτής λόγω του ότι η κατασκευή έγινε αρχικά χρησιμοποιώντας φωτοευαίσθητη πλακέτα με αποτέλεσμα να μην υφίσταται η δυνατότητα για δημιουργία επιμεταλλωμένων οπών και να μην υπάρχει περιθώριο χώρου για την τοποθέτηση κρυστάλλου.

Παρόλα αυτά η σχετικά υψηλή ακρίβεια ($\pm 1.5\%$ μέγιστη απόκλιση) του εσωτερικού ταλαντωτή σε συνδυασμό με το fractional baud rate generator που διατίθεται στην σειρά xmega για τον χρονισμό των θυρών UART έχει ως αποτέλεσμα την επίτευξη ικανοποιητικά χαμηλού ποσοστού σφαλμάτων κατά την σειριακή επικοινωνία.

2.1.3 Κύκλωμα μεταφοράς δεδομένων σε H/Y



Εικόνα 2-IV: Κύκλωμα μεταφοράς δεδομένων σε H/Y

Για την αμφίδρομη επικοινωνία της συσκευής με H/Y μέσω USB χρησιμοποιείται ο USB→UART controller MCP2200 της Microchip [Ref. 8]. Η επιλογή αυτή έγινε κυρίως λόγω της αρκετά χαμηλότερης τιμής αγοράς αλλά και της «φιλικότερης» SOIC συσκευασίας, που διατίθεται σε σχέση με τα ανταγωνιστικά ολοκληρωμένα που κυκλοφορούν αποκλειστικά σε tssop ή μικρότερες συσκευασίες.

Η τροφοδοσία του ολοκληρωμένου γίνεται απευθείας από το USB bus με αποτέλεσμα απλούστερο κύκλωμα αλλά και μεγαλύτερη αυτονομία αφού εάν δεν υπάρχει σύνδεση USB τίθεται εκτός λειτουργίας το ολοκληρωμένο.

Σχεδιασμός - Υλοποίηση του hardware του συστήματος

Οι πυκνωτές απόζευξης C6 και C20 συνδέονται κοντά στους αντίστοιχους ακροδέκτες τροφοδοσίας, όπου MCP_V_{USB} η έξοδος του ενσωματωμένου σταθεροποιητή τάσης.

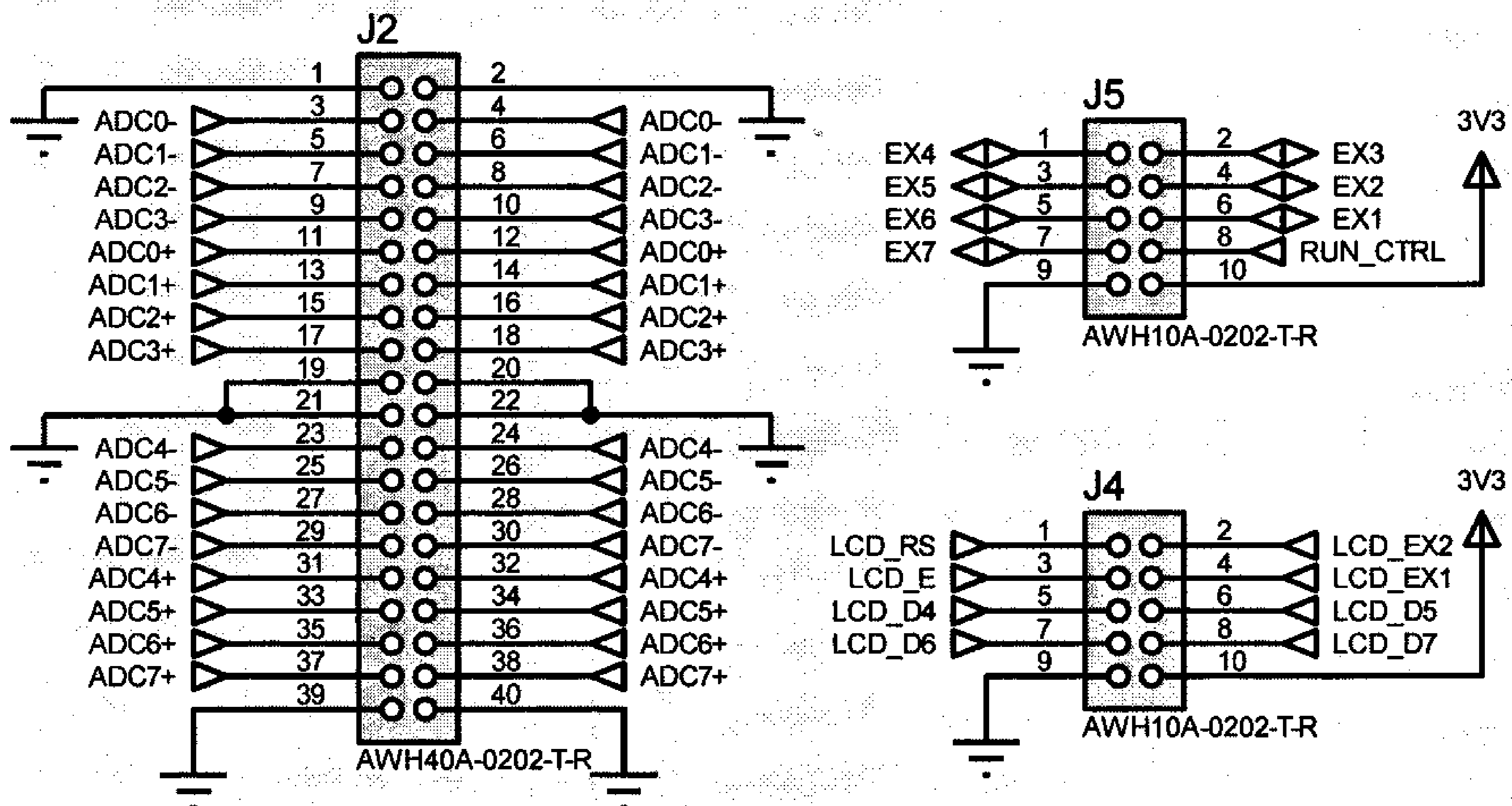
Στην γραμμή reset του MCP2200 συνδέεται ο pull-up αντιστάτης R4 και ο πυκνωτής απόζευξης C19 για την απόρριψη θορύβου και την διατήρηση σταθερής στάθμης τάσης.

Ο χρονισμός γίνεται με τον εξωτερικό κεραμικό ταλαντωτή X1 στα 12MHz.

Τα LED D5,D6 με τις αντίστοιχες αντιστάσεις περιορισμού ρεύματος παρέχουν μια οπτική ένδειξη της κυκλοφορίας δεδομένων μεταξύ H/Y και συσκευής.

Τέλος οι αντιστάσεις R7,R8 περιορίζουν το ρεύμα μεταξύ των ακροδεκτών R_x, T_x του MCP2200 και των αντίστοιχων ακροδεκτών του μικροελεγκτή ενώ η διάοδος ζένερ D1 ψαλιδίζει το σήμα που φτάνει στο R_x του xmega στα 3.3V μιας που οι ακροδέκτες του δεν είναι 5V tolerant [Ref. 11].

2.1.4 Είσοδος ηλεκτροδίων - βύσματα επέκτασης



Εικόνα 2-V: Είσοδος ηλεκτροδίων - βύσματα επέκτασης

Για την διασύνδεση των ηλεκτροδίων στο κυρίως σώμα της συσκευής μετά από αναζήτηση για πιθανές λύσεις και σχετικές δοκιμές επιλέχτηκε η χρήση απλών βυσμάτων IDC-40pin.

Αυτή η λύση προσφέρει χαμηλό κόστος, καλή ηλεκτρική αγωγιμότητα, εύκολη προσαρμογή σε διαφορετικούς τύπους ηλεκτροδίων και διασύνδεση μέσω καλωδιοταινίας ή απευθείας με PCB-mounted βύσμα. Για την εξασφάλιση καλής ηλεκτρικής επαφής χρησιμοποιείται ένα ζεύγος παρακείμενων ακροδεκτών για το κάθε σήμα εισόδου.

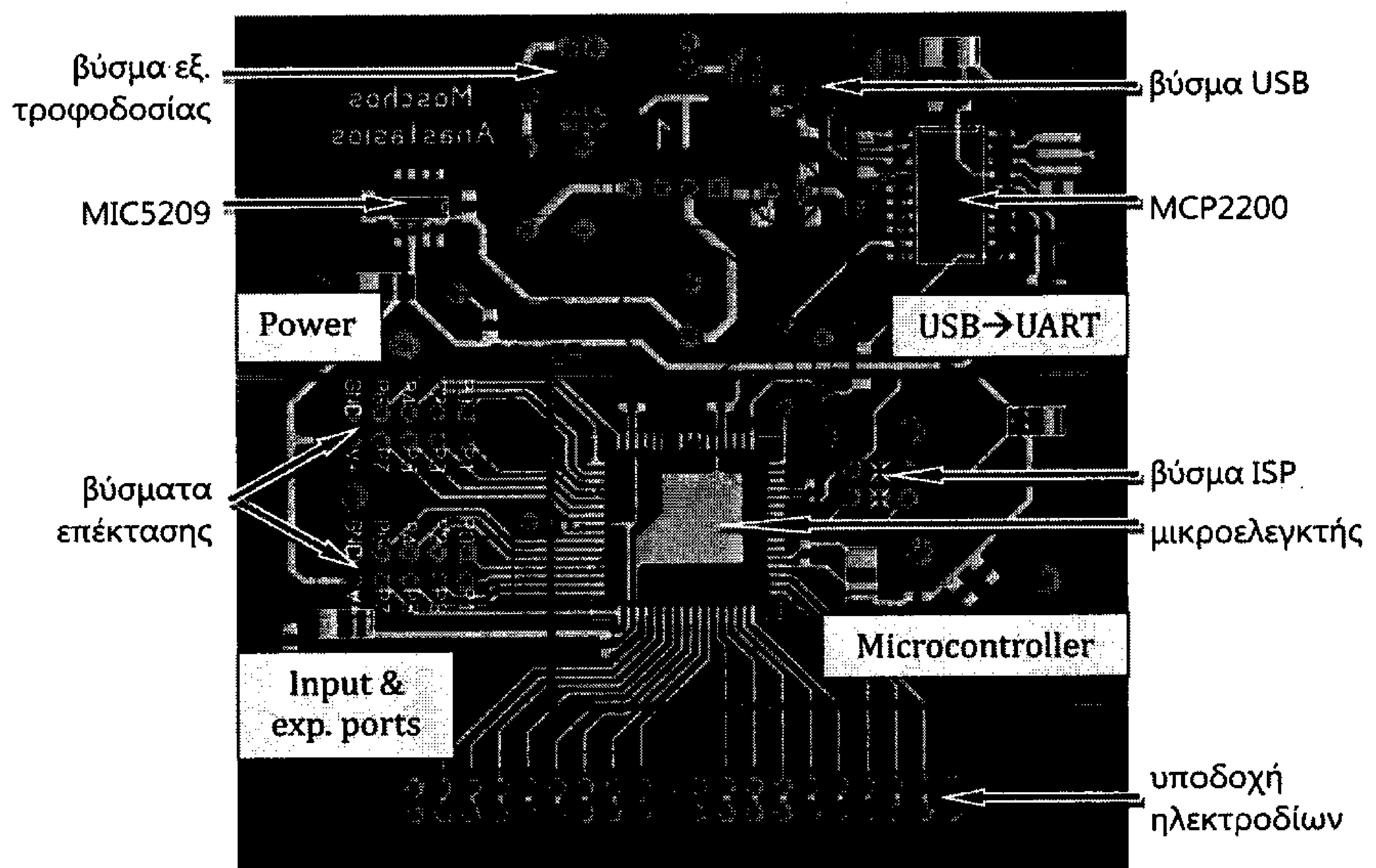
Σχεδιασμός - Υλοποίηση του hardware του συστήματος

Προκειμένου να μειωθούν οι παρεμβολές από εξωτερικές πηγές και οι αλληλοπαρεμβολές μεταξύ των 2 ADC οι οποίοι λαμβάνουν τα δείγματα σχεδόν ταυτόχρονα έχουν γειωθεί οι επαφές που βρίσκονται στο κέντρο, ανάμεσα στους ακροδέκτες που δρομολογούνται στον ADCA και τον ADCB, καθώς και οι επαφές στα άκρα του βύσματος.

Παρότι δεν υπάρχει πλήρης θωράκιση των σημάτων εισόδου αυτό σύμφωνα με μετρήσεις δεν επηρεάζει σημαντικά τα αποτελέσματα, εν μέρει λόγω του ότι έχουν υλοποιηθεί ψηφιακά φίλτρα στο λογισμικό της συσκευής και επίσης διότι τα φαινόμενα που παρατηρούνται κατά κανόνα εξελίσσονται ομαλά και αργά σε σχέση με την συχνότητα δειγματοληψίας.

Εκτός από το βύσμα ηλεκτροδίων στην πλακέτα υπάρχουν και 2 βύσματα IDC-10rip τα οποία εξυπηρετούν για την σύνδεση της οθόνης LCD και του διακόπτη ελέγχου ροής του προγράμματος. Έχει προβλεφθεί η ύπαρξη επιπλέον επαφών για την περίπτωση μελλοντικής χρήσης όπως πχ αν αντικατασταθεί η οθόνη χαρακτήρων με οθόνη γραφικών που απαιτεί περισσότερα I/O.

2.1.5 Σχεδίαση PCB



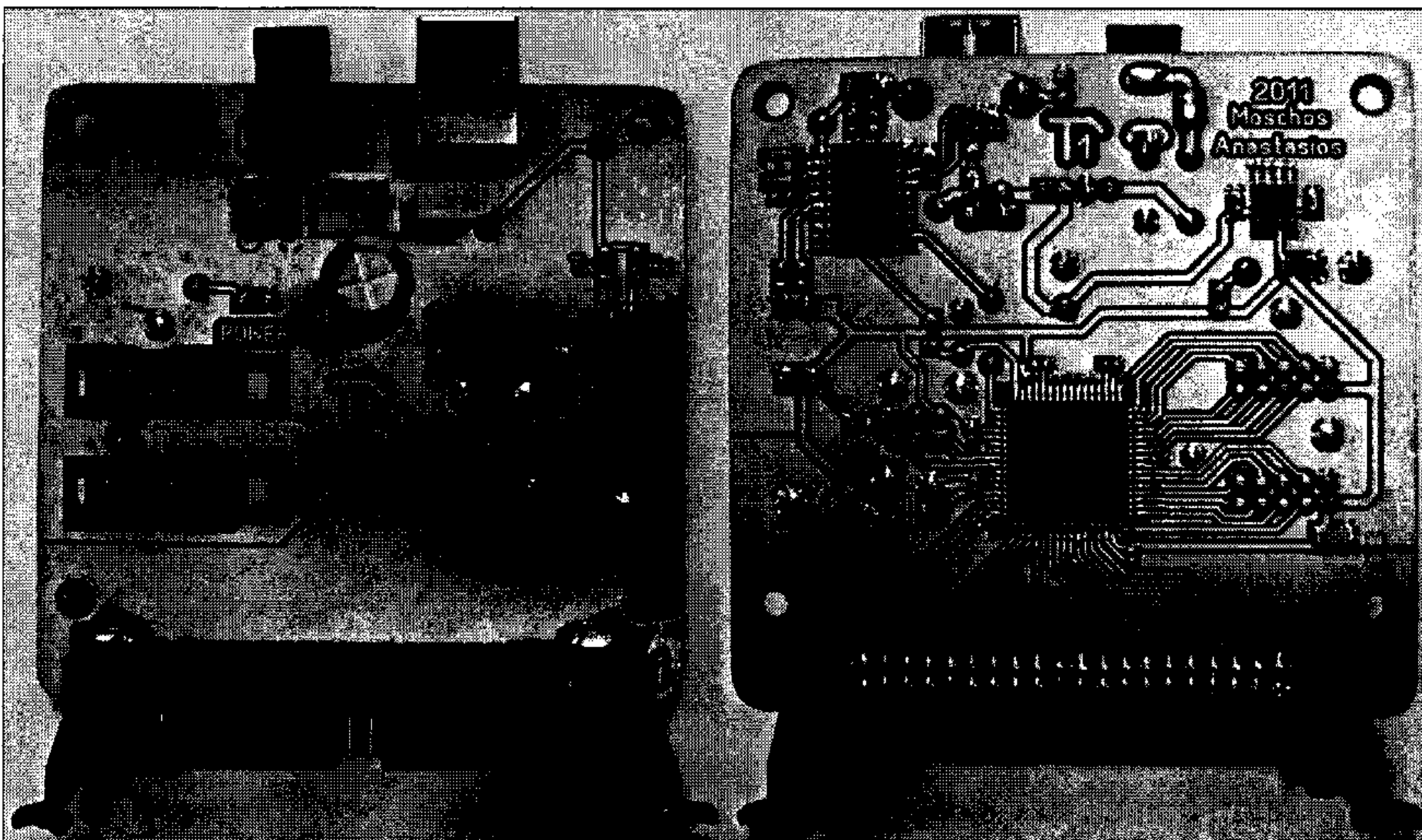
Εικόνα 2-VI: Σχέδιο PCB με ενδεικτικό διαχωρισμό σε διαμερίσματα και τα κυριότερα εξαρτήματα σημειωμένα

Τα κυκλώματα που παρουσιάστηκαν στις προηγούμενες παραγράφους (2.1.1-2.1.4) υλοποιήθηκαν σε PCB χρησιμοποιώντας το Sprint Layout. Για τα εξαρτήματα που δεν υπήρχαν στις βιβλιοθήκες του προγράμματος σχεδιάστηκε κατάλληλο footprint με βάση το εκάστοτε datasheet.

Σχεδιασμός - Υλοποίηση του hardware του συστήματος

Η πλακέτα που σχεδιάστηκε είναι διπλής όψεως με διαστάσεις 80x77mm, με ελάχιστο πλάτος διαδρόμου 0,3mm και ελάχιστη απόσταση μεταξύ γειτονικών τμημάτων χαλκού 0,25mm ώστε να είναι εύκολη η κατασκευή της με συμβατική τεχνολογία φωτοευαίσθητων πλακετών.

Κατά τον σχεδιασμό διαχωρίστηκαν κατά το δυνατόν τα επιμέρους τμήματα σε διαφορετικές περιοχές όπως φαίνεται στην εικόνα 2-VI ούτως ώστε να ελαχιστοποιηθούν οι παρεμβολές μεταξύ των υψίσυχων ψηφιακών και των ευαίσθητων αναλογικών τμημάτων. Επίσης καλύφθηκε όλη η ελεύθερη επιφάνεια με ground plane και στις δυο όψεις με διαχωρισμό αυτού σε αναλογικό και ψηφιακό τμήμα προς περιορισμό του θορύβου [Ref. 36-39].



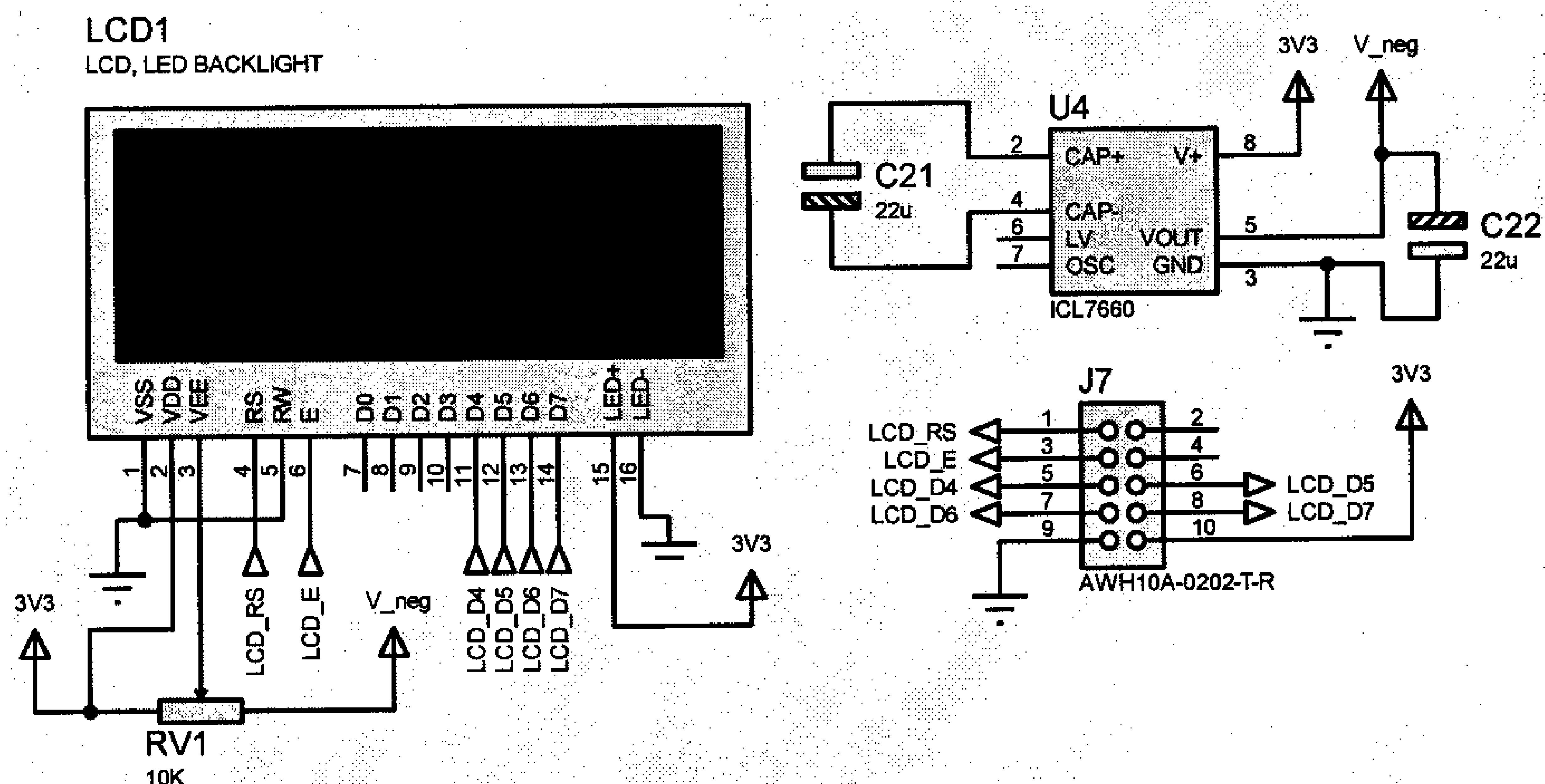
Εικόνα 2-VII: Άνω και κάτω όψη του κύριου PCB

2.2 Η πλακέτα διασύνδεσης με οθόνη χαρακτήρων LCD

Για την απεικόνιση των αποτελεσμάτων χρησιμοποιείται μια κοινή οθόνη 20x4 αλφαριθμητικών χαρακτήρων βασισμένη σε ολοκληρωμένο συμβατό με Hitachi HD44780 [Ref. 12].

Δεδομένου ότι οι οθόνες τέτοιου τύπου με προδιαγραφές για χρήση σε τροφοδοσία 3,3V από τον κατασκευαστή είναι δυσεύρετες και αρκετά ακριβότερες προτιμήθηκε να χρησιμοποιηθεί μια κοινή 5V έκδοση με ένα επιπλέον κύκλωμα προσαρμογής αποτελούμενο από ένα ολοκληρωμένο ICL7660 (U4), δυο πυκνωτές τανταλίου 22μF (C21,C22) και ένα τρίμμερ 10KΩ (RV1) για την ρύθμιση της αντίθεσης όπου και υπάρχει το κυρίως πρόβλημα.

Σχεδιασμός - Υλοποίηση του hardware του συστήματος



Εικόνα 2-VIII: Κύκλωμα διασύνδεσης με οθόνη αλφαριθμητικών χαρακτήρων τύπου LCD

Το κύκλωμα παράγει μια αρνητική τάση περίπου $-3,3V$ η οποία εφαρμόζεται στο ένα άκρο του μεταβλητού αντιστάτη και αντισταθμίζει την μειωμένη τάση τροφοδοσίας, παρέχοντας το δυναμικό εύρος που είναι απαραίτητο με αποτέλεσμα ικανοποιητικά επίπεδα αντίθεσης. Όσον αφορά τον ελεγκτή της LCD και το backlight όπως διαπιστώθηκε μετά από εκτεταμένες δοκιμές δεν υπάρχει πρόβλημα κατά την λειτουργία με τάση τροφοδοσίας $3,3V$.

Η LCD λειτουργεί σε τυπική συνδεσμολογία 4 bit mode για απλούστευση και εξοικονόμηση ακροδεκτών του μικροελεγκτή με εξαίρεση το R/W pin που είναι γειωμένο αφού δεν χρειάζεται να διαβάζονται δεδομένα από την LCD λόγω του αρκετά χαμηλού ρυθμού ανανέωσης.

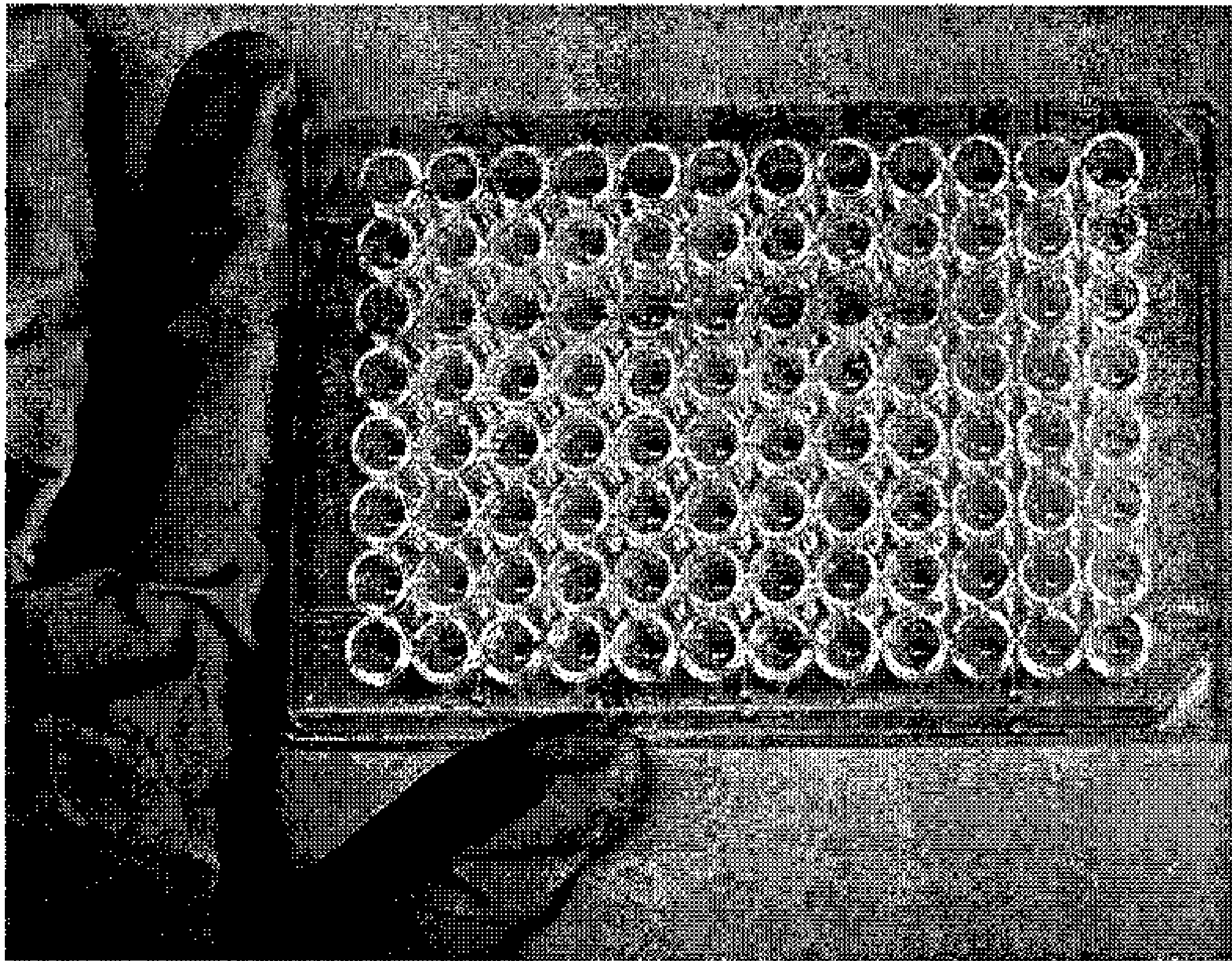
Τέλος στο βύσμα σύνδεσης με τον μικροελεγκτή υπάρχουν 2 ασύνδετοι ακροδέκτες που μπορούν να χρησιμοποιηθούν μελλοντικά για τον έλεγχο του φωτισμού της LCD ή για κάποια άλλη λειτουργία.

2.3 Τα ηλεκτρόδια μέτρησης

Δεδομένης της φύσεως της εφαρμογής είναι εύκολα αντιληπτό ότι τα ηλεκτρόδια μέτρησης είναι ένα κρίσιμο μέρος του συστήματος.

Τα ηλεκτρόδια ιδανικά θα πρέπει να είναι πλήρως αδρανή, ήτοι να μην αλληλεπιδρούν με τα δείγματα προς μέτρηση και να μην επηρεάζουν τις μετρούμενες τάσεις.

Σχεδιασμός - Υλοποίηση του hardware του συστήματος

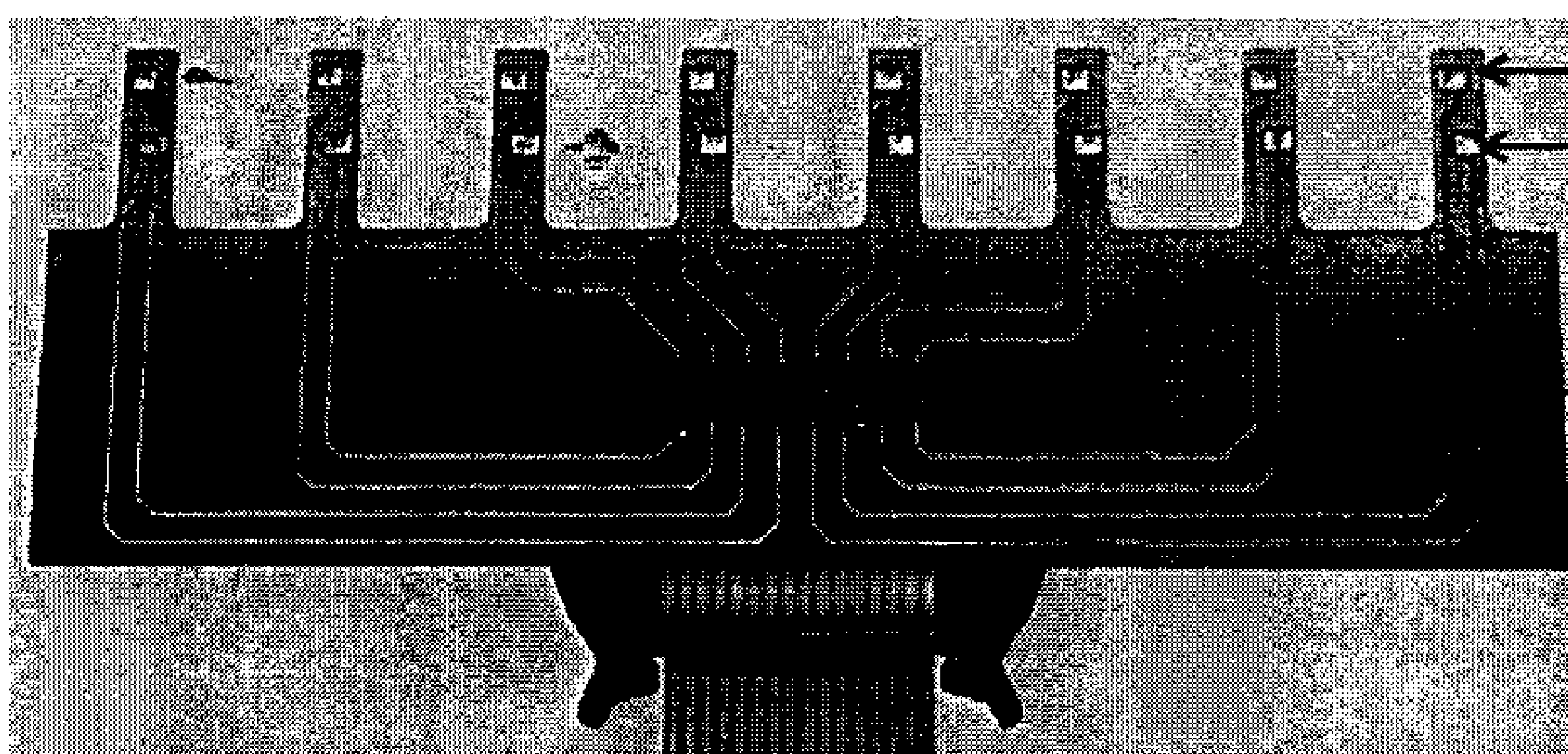


Εικόνα 2-ΙΧ: Δισκίδιο μικροκοιλοτήτων
(ELISA microtiter plate) [Φωτ.: Jeffrey M. Vinocur]

Άλλα επιθυμητά χαρακτηριστικά είναι η εύκολη προσαρμογή των ηλεκτροδίων σε κάποια κοινή δομή κατάλληλη για εργαστηριακές δοκιμές όπως τα καθιερωμένα δισκίδια μικροκοιλοτήτων (ELISA microtiter plate) (Εικόνα 2-ΙΧ), η εύκολη παραγωγή σε μεγάλη κλίμακα, η δυνατότητα επαναχρησιμοποίησης ή ανακατασκευής και φυσικά το χαμηλό κόστος κατασκευής.

Στις επόμενες παραγράφους θα παρουσιαστούν οι τύποι ηλεκτροδίων που έχουν δοκιμαστεί πειραματικά.

2.3.1 Ηλεκτρόδια εγχαραγμένα σε PCB



Εικόνα 2-Χ: PCB που φέρει 8 ζεύγη ηλεκτροδίων εγχαραγμένα

Αυτός ήταν ο πρώτος τύπος ηλεκτροδίων που χρησιμοποιήθηκε για την εφαρμογή (Αρχική σχεδίαση: Οδυσσέας Τσακιρίδης). Αποτελείται από ένα PCB με FR-4 υπόστρωμα στο οποίο έχουν δημιουργηθεί κατάλληλες εγκοπές με CNC router ώστε να σχηματίζεται μια οκτάδα προεξεχόντων ενδιάμεσων τμημάτων που φέρουν τα ηλεκτρόδια ώστε να προσαρμόζεται απευθείας σε συνηθισμένα δισκίδια 8x12 μικροκοιλοτήτων.

Συγκεκριμένα στην επιφάνεια κάθε ενδιάμεσου τμήματος έχει σχεδιαστεί ένα ζεύγος επαφών με διαστάσεις 2x1,5mm και απόσταση μεταξύ τους 3,5mm οι οποίες αποτελούν τα ηλεκτρόδια, τα οποία είναι επικασσιτερωμένα με HASL (Hot Air Solder Leveling) μέθοδο. Η ελεύθερη επιφάνεια καλύπτεται πλήρως με κλασικό soldermask.

Σχεδιασμός - Υλοποίηση του hardware του συστήματος

Βυθίζοντας πλήρως την πλακέτα ηλεκτροδίων σε ένα δισκίδιο μικροκοιλοτήτων που περιέχει τα δείγματα προς μέτρηση τα ηλεκτρόδια τοποθετούνται στο κατάλληλο ύψος, τα ηλεκτρόδια μέτρησης περίπου στο μέσο του πήγματος που περιέχει τα κύτταρα και τα ηλεκτρόδια αναφοράς στο άνω μέρος του διαλύματος που λειτουργεί ως σημείο αναφοράς. Η πλακέτα συνδέεται με το σύστημα μέτρησης μέσω καλωδιοταινίας χρησιμοποιώντας κατάλληλα βύσματα τύπου IDC.

Τα πλεονεκτήματα αυτού του τύπου ηλεκτροδίων είναι προφανή αφού στηρίζεται αποκλειστικά στην τυποποιημένη διαδικασία παραγωγής PCB όπως αυτή χρησιμοποιείται από την βιομηχανία ηλεκτρονικών. Είναι δυνατή η εύκολη και φτηνή μαζική παραγωγή τους και μπορούν να τροποποιηθούν εύκολα εάν χρειαστεί να προσαρμοστούν σε κάποια άλλη διάταξη.

Δυστυχώς η κατασκευή με τεχνολογία γενικής χρήσης επιβάλλει κάποιους σημαντικούς περιορισμούς οι οποίοι τελικά καθιστούν τα ηλεκτρόδια αυτού του τύπου ακατάλληλα για χρήση. Καταρχάς ο χαλκός και ο κασσίτερος που χρησιμοποιούνται ως αγωγοί δεν είναι ευγενή μέταλλα οπότε επηρεάζουν τις μετρήσεις καθώς οξειδώνονται ή αντιδρούν με τα διαλύματα. Το soldermask που χρησιμοποιείται ως μέσο μόνωσης πιθανόν να μην είναι απόλυτα συμπαγές και ομοιόμορφο, ειδικά εάν σχηματίζεται με μεθόδους μεταξοτυπίας ή ψεκασμού με αποτέλεσμα ανεπαρκή μόνωση των διαδρόμων. Τέλος η υλοποίηση των ηλεκτροδίων στην επιφάνεια της πλακέτας σε συνδυασμό με την μειωμένη συνάφεια και το σχετικά μεγάλο ιξώδες του διαλύματος οδηγούν σε μη εξασφαλισμένη ηλεκτρική επαφή μεταξύ των ηλεκτροδίων και του διαλύματος.

Υπάρχουν προοπτικές για βελτιωμένη έκδοση αυτού του τύπου ηλεκτροδίων όπου θα χρησιμοποιούνται επιχρυσωμένες επαφές και soldermask τύπου φιλμ το οποίο παρέχει ομοιόμορφη κάλυψη αλλά επί του παρόντος δεν έχει προχωρήσει η ανάπτυξη της.

2.3.2 Ηλεκτρόδια τύπου ράβδου

Αυτός ο τύπος ηλεκτροδίων αναπτύχθηκε εξ ολοκλήρου στα πλαίσια της παρούσας εργασίας κι ουσιαστικά αποτελεί εξέλιξη των ηλεκτροδίων PCB αντιμετωπίζοντας τα κυριότερα μειονεκτήματα τους.

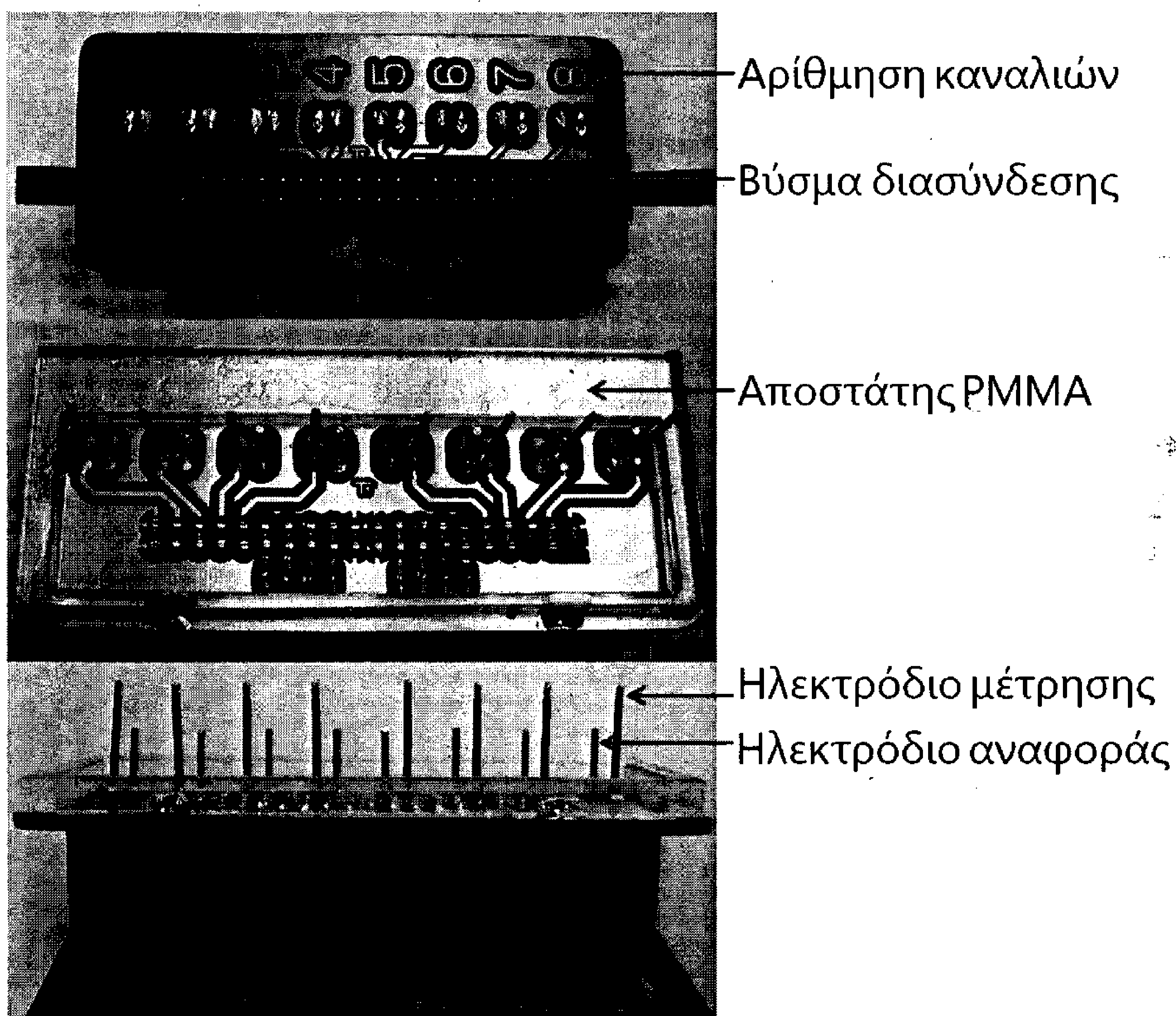
Χρησιμοποιώντας ράβδους αργύρου διαμέτρου 1mm ως ηλεκτρόδια εξασφαλίζεται καλή ηλεκτρική επαφή με το μετρούμενο διάλυμα και δεδομένου ότι ο άργυρος είναι ευγενές μέταλλο με καλύτερη βιοσυμβατότητα από τον χαλκό μειώνεται η επίδραση των ηλεκτροδίων στις μετρήσεις.

Σχεδιασμός - Υλοποίηση του hardware του συστήματος

Οκτώ ζεύγη ράβδων, με την μία ράβδο σε κάθε ζεύγος να δρα ως ηλεκτρόδιο αναφοράς και την δεύτερη ως ηλεκτρόδιο μέτρησης, κολλώνται πάνω σε κατάλληλα σχεδιασμένο PCB με τις ράβδους κατακόρυφα ως προς το επίπεδο της πλακέτας.

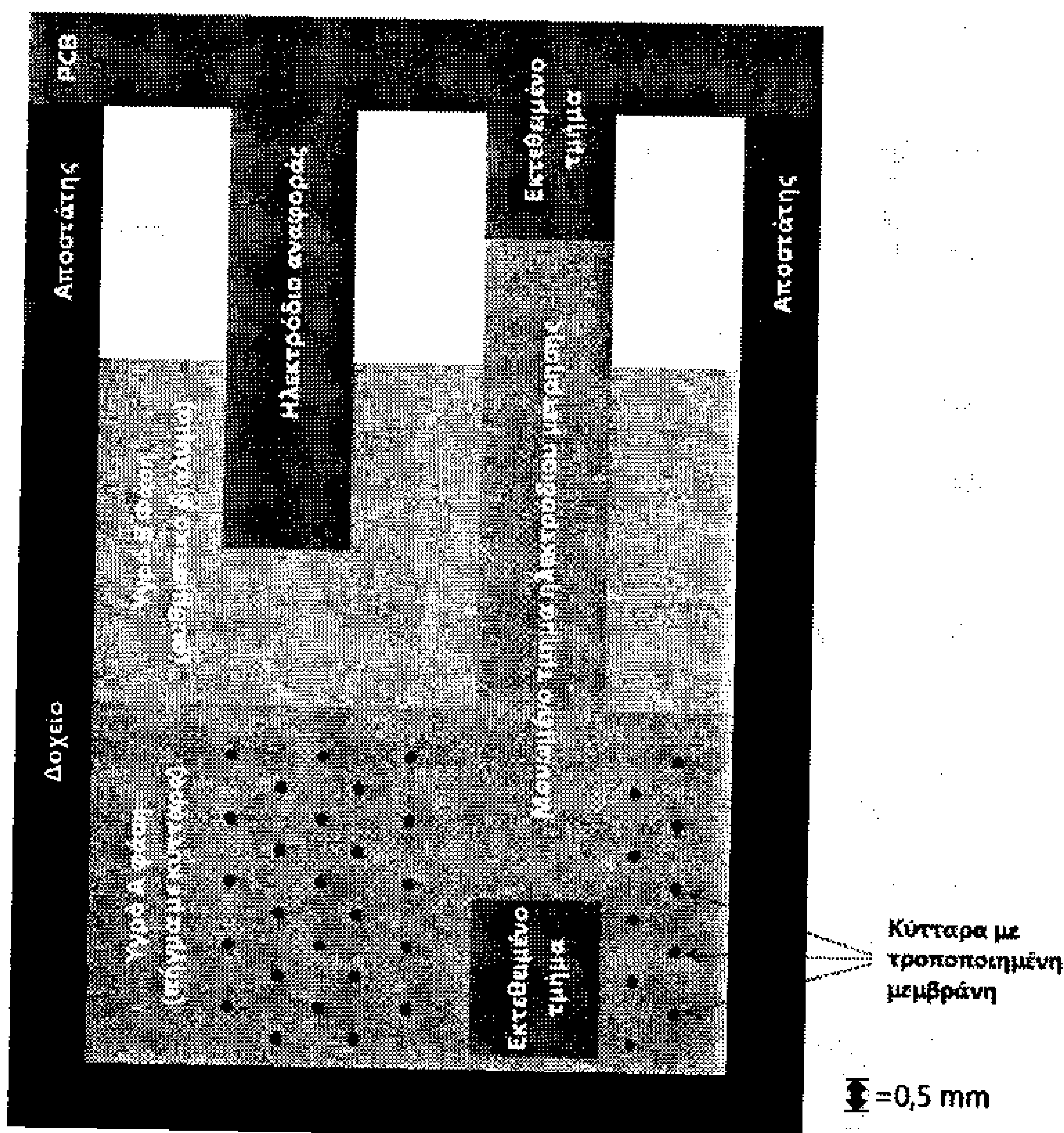
Για τα ηλεκτρόδια μέτρησης χρησιμοποιούνται ειδικά κατασκευασμένες εμαγιέ ράβδοι με μόνωση από βερνίκι πολυουρεθάνης που σχηματίζει ένα ομοιόμορφο και ανθεκτικό μονωτικό φιλμ στην κυλινδρική επιφάνεια, με εκτεθειμένα τμήματα σε αμφοτέρωτα τα άκρα της ράβδου.

Στο κάτω μέρος της πλακέτας επικολλάται κατάλληλα κατασκευασμένος αποστάτης από PMMA (πλεξιγκλάς) προκειμένου να προσαρμόζονται επακριβώς τα ηλεκτρόδια σε συνηθισμένα δισκίδια 8x12 μικροκοιλοτήτων και να αποφεύγεται η άμεση επαφή του PCB με το δισκίδιο.



Εικόνα 2-ΧΙ: Ηλεκτρόδια τύπου ράβδου από διάφορες οπτικές γωνίες

Σχεδιασμός - Υλοποίηση του hardware του συστήματος



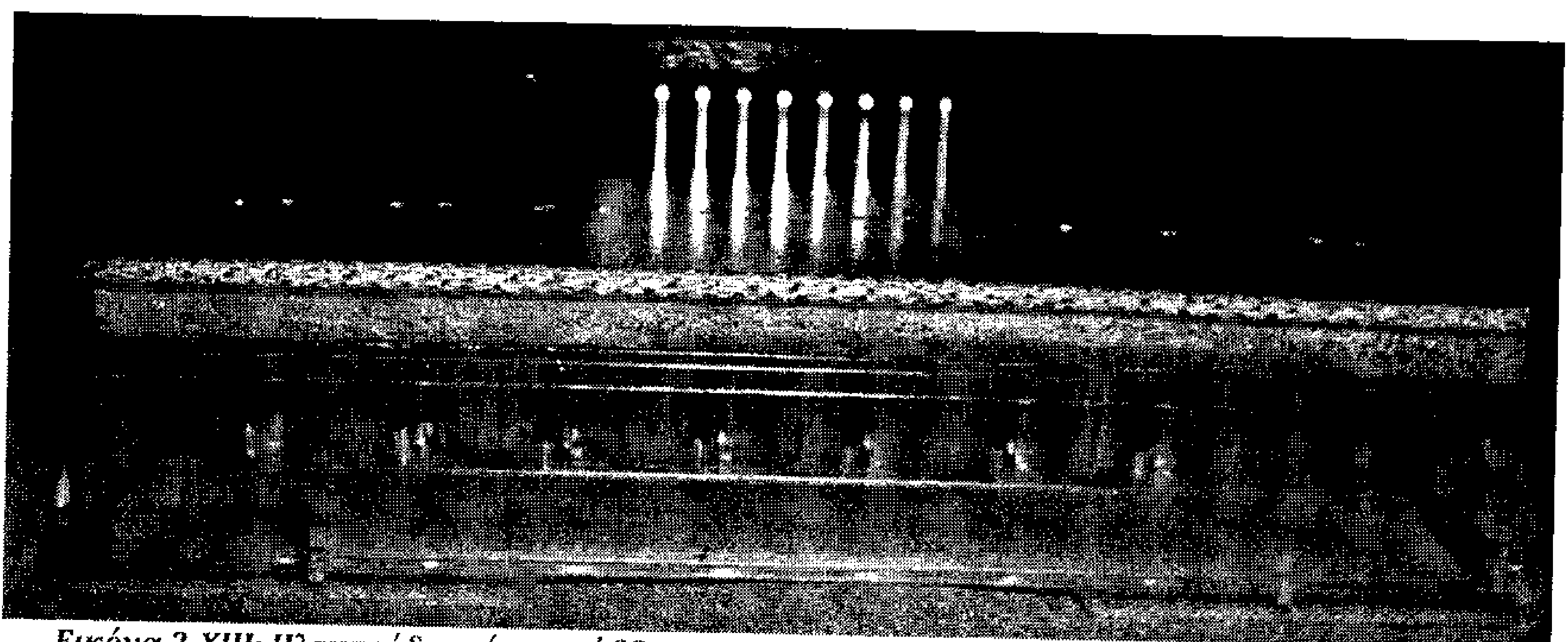
Παρότι ο εν λόγω τύπος ηλεκτροδίων δεν κατασκευάζεται αποκλειστικά με τεχνολογία παραγωγής PCB το κόστος κατασκευής διατηρείται σε χαμηλά επίπεδα και είναι δυνατή η μαζική παραγωγή χρησιμοποιώντας επιπλέον μόνο μηχανολογικό εξοπλισμό χαμηλής τεχνολογίας.

Σημαντικό πλεονέκτημα αυτού του τύπου είναι η προοπτική για πολλαπλές χρήσεις αξιοποιώντας κατάλληλη μέθοδο καθαρισμού ή μερικής ανακατασκευής.

Εικόνα 2-XII: Εγκάρσια τομή εφαρμογής ηλεκτροδίων τύπου ράβδου σε μικροκοιλότητα.

Σημ: Η οριζοντια διάσταση και το μέγεθος των κυττάρων δεν είναι σε κλίμακα.

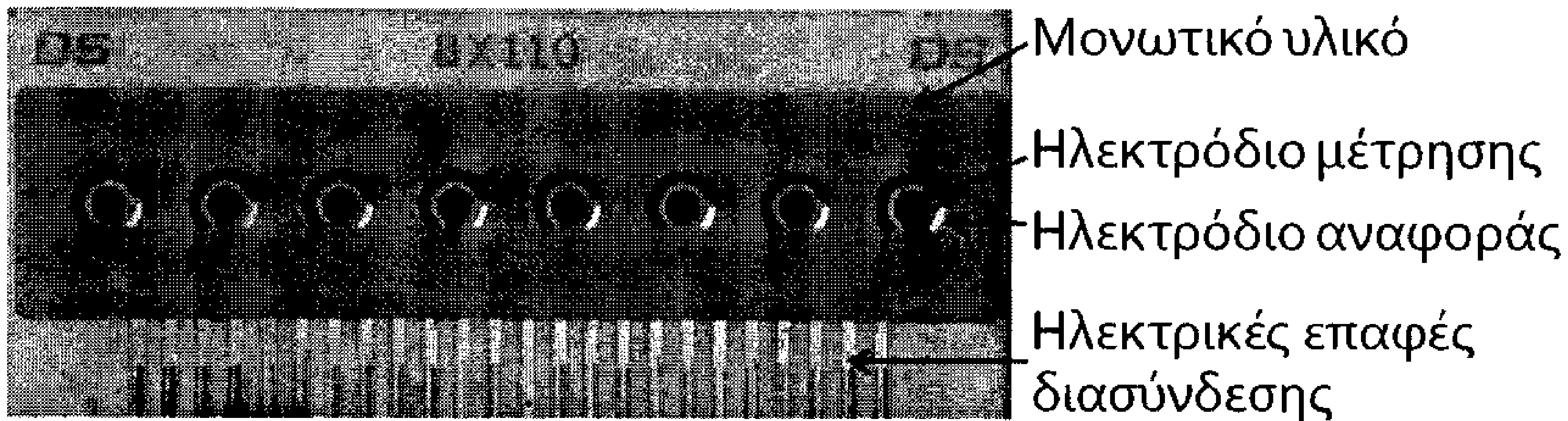
Δυστυχώς σύμφωνα με πειραματικά δεδομένα υπάρχει μεν σημαντική βελτίωση έναντι του προηγούμενου τύπου ωστόσο εξακολουθούν να υπάρχουν προβλήματα με την ποιότητα του σήματος που λαμβάνεται, πιθανόν διότι και τα δυο ηλεκτρόδια είναι από το ίδιο υλικό ή λόγω κάποιου άλλου παράγοντα σχετικού με την αλληλεπίδραση των ηλεκτροδίων με τα βιολογικά δείγματα.



Εικόνα 2-XIII: Ηλεκτρόδια τύπου ράβδου προσαρμοσμένα σε δισκίο 8x12 μικροκοιλοτήτων

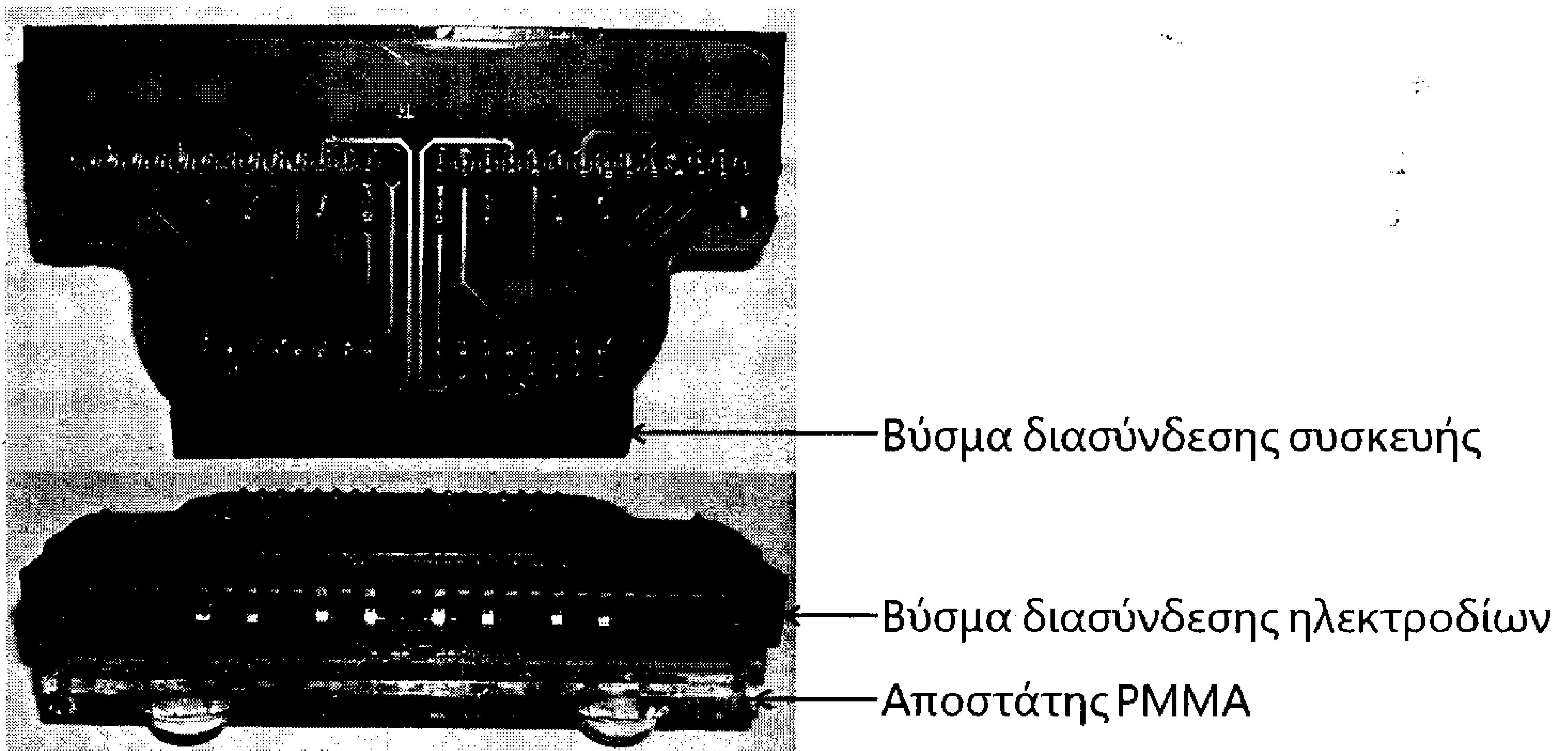
2.3.3 Ηλεκτρόδια εκτυπωμένα σε κεραμικό υπόστρωμα

Τα ηλεκτρόδια αυτού του τύπου έχουν σχεδιαστεί και παράγονται από την εταιρία Dropsens [ref.15] η οποία εξειδικεύεται στον χώρο των screen-printed ηλεκτροδίων.



Εικόνα 2-XIV: Ηλεκτρόδια εκτυπωμένα σε κεραμικό υπόστρωμα

Αποτελούνται από ένα κεραμικό υπόστρωμα πάχους 1mm και διαστάσεων 78,5x33mm στο οποίο έχουν εναποτεθεί αγωγάμοι διάδρομοι από άργυρο καθώς και 8 τριάδες ηλεκτροδίων εκ των οποίων δυο επαφές άνθρακα και μια επαφή αναφοράς από άργυρο με τεχνολογία screen-printing. Κατόπιν η επιφάνεια δοκιμής μονώνεται με φιλμ από πολυμερές υλικό.



Εικόνα 2-XV: Προσαρμογέας ηλεκτροδίων Dropsens

Για την διασύνδεση των ηλεκτροδίων αυτού του τύπου με το κυρίως σύστημα σχεδιάστηκε η πλακέτα προσαρμογής της εικόνας 2-XV χρησιμοποιώντας ένα κοινό βύσμα card-edge 2x30 επαφών το οποίο με ελαφρά τροποποίηση ταιριάζει άψογα με τις επαφές εξόδου που διαθέτουν τα ηλεκτρόδια. Η σύνδεση με την συσκευή γίνεται απευθείας, χωρίς την ανάγκη για χρήση καλωδιοταινίας.

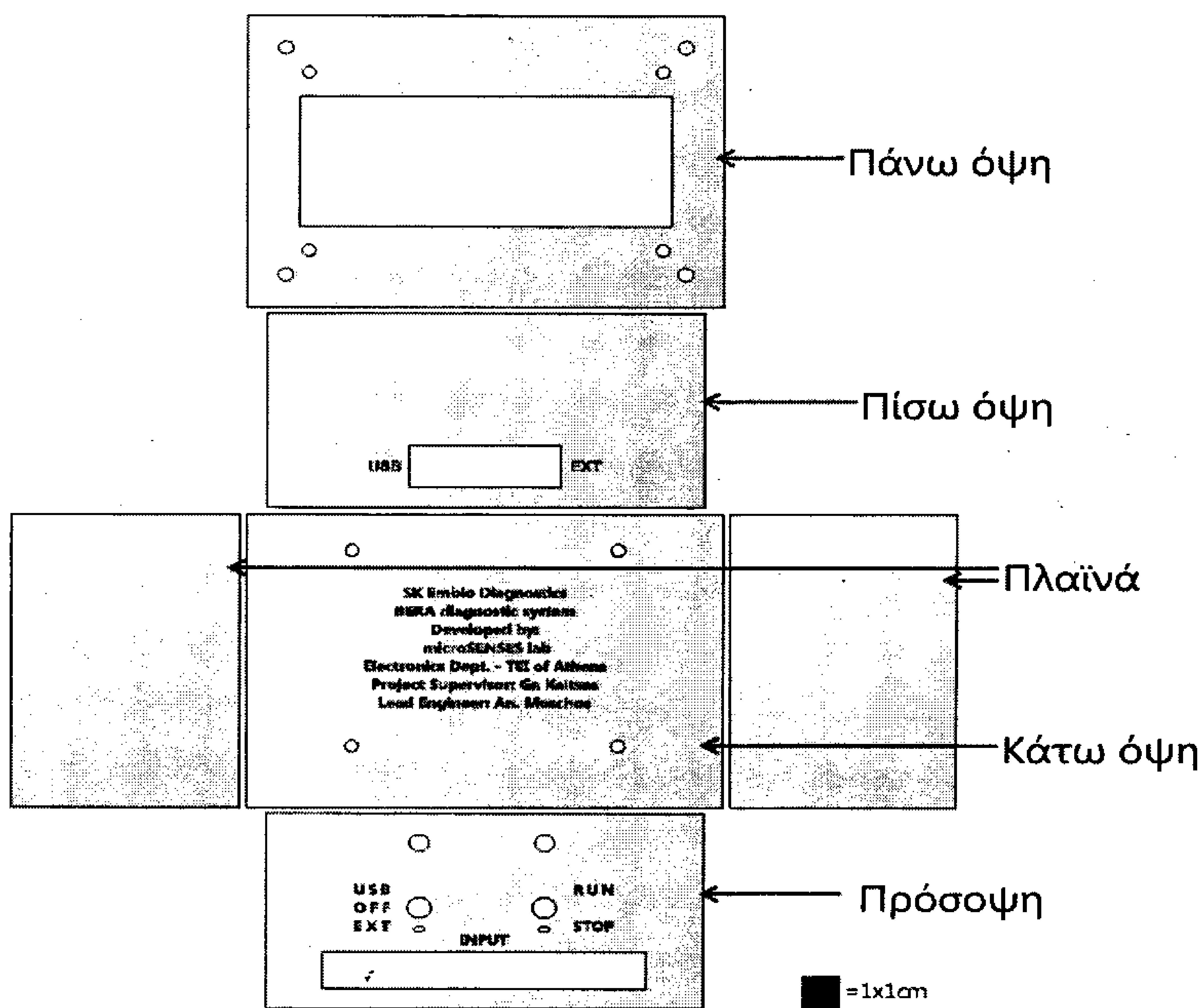
Σχεδιασμός - Υλοποίηση του hardware του συστήματος

Αυτός ο τύπος ηλεκτροδίων σύμφωνα με πειραματικά δεδομένα θεωρείται ότι παρέχει την σταθερότερη συμπεριφορά και τα πιο αξιόπιστα αποτελέσματα αλλά το υψηλό κόστος αγοράς σε συνδυασμό με το γεγονός ότι μπορούν να χρησιμοποιηθούν μόνο μια φορά αποτελεί πηγή προβληματισμού για την καθιέρωση της χρήσης τους.

2.4 Σχεδίαση & κατασκευή συσκευασίας

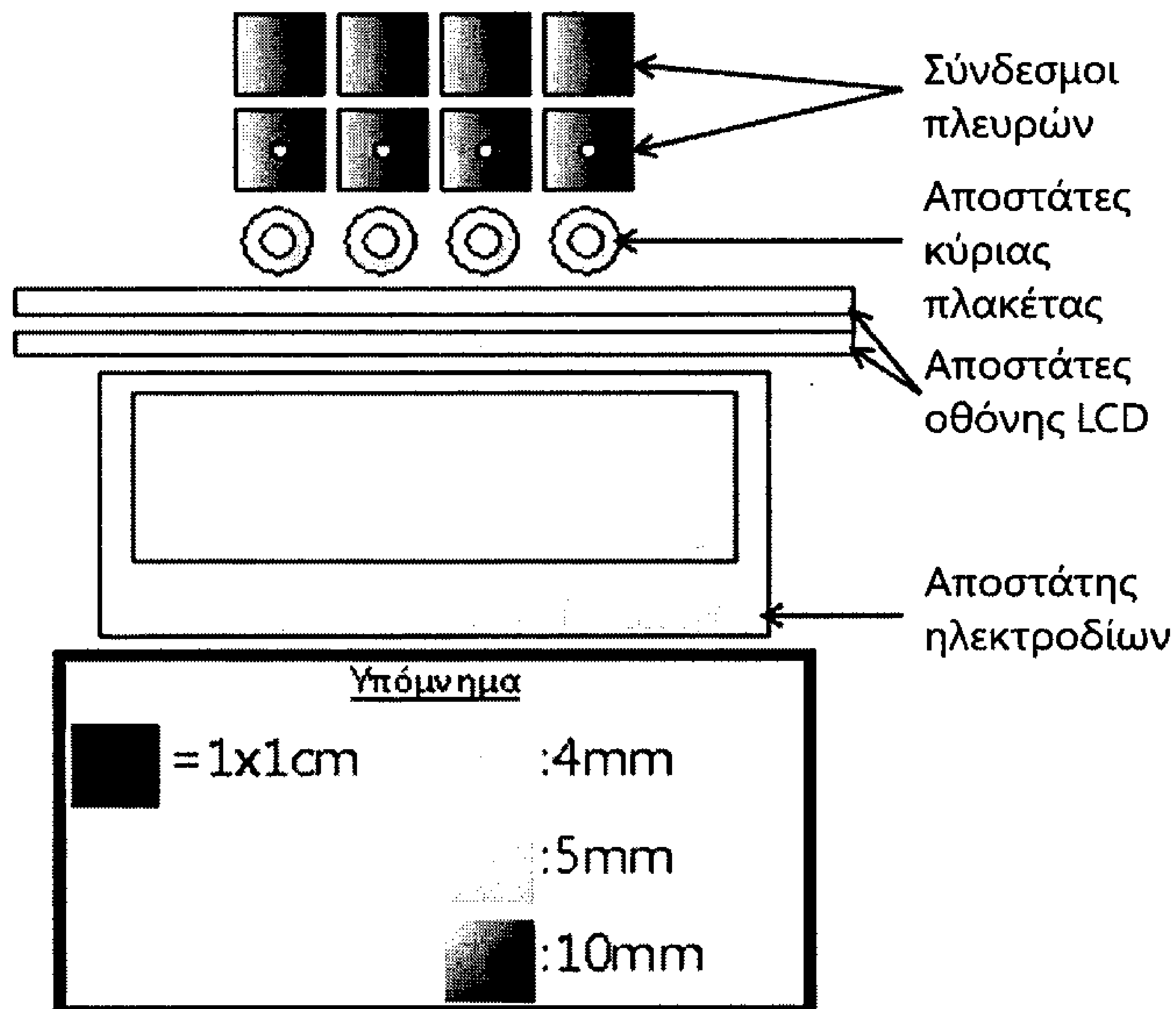
Για την στέγαση του κυρίου μέρους της συσκευής σχεδιάστηκε και κατασκευάστηκε κουτί από διαφανές πλεξιγκλάς (Εικόνα 2-XVI). Η επιλογή αυτή οφείλεται κυρίως στο γεγονός ότι υπήρχε πρόσβαση σε κατάλληλο εξοπλισμό CNC κοπής και εγχάραξης με χρήση λέιζερ, το οποίο διευκόλυνε σημαντικά την διαδικασία κατασκευής, παράγοντας παράλληλα ικανοποιητικό αισθητικά αποτέλεσμα.

Η σχεδίαση πραγματοποιήθηκε σε περιβάλλον CorelDraw, μιας που η μηχανή επεξεργασίας συνεργάζεται απευθείας με το εν λόγω λογισμικό. Για την υλοποίηση χρησιμοποιήθηκε κυρίως υλικό πάχους 5mm, καθώς και 4 ή 10mm για κάποιους αποστάτες και συνδέσμους που χρειάστηκαν. Αφού κόπηκαν τα επιμέρους τμήματα η συναρμολόγηση έγινε συγκολλώντας τα τμήματα με μίγμα διχλωρομεθάνιου / μικρορινισμάτων πλεξιγκλάς εξασφαλίζοντας έτσι άριστη εφαρμογή και υψηλή ποιότητα συγκόλλησης.



Εικόνα 2-XVI: Ανάπτυγμα του κουτιού που σχεδιάστηκε

Σχεδιασμός - Υλοποίηση του hardware
του συστήματος



Εικόνα 2-XVII: Αποστάτες και σύνδεσμοι που σχεδιάστηκαν.

Μειονέκτημα της χρήσης πλεξιγκλάς για την κατασκευή είναι ότι προφανώς το υλικό δεν παρέχει Η/Μ θωράκιση, η οποία φυσικά θα ήταν επιθυμητή για την μείωση των παρεμβολών κατά την λήψη μετρήσεων. Το συνολικό κόστος διατηρείται σε σχετικά χαμηλά επίπεδα δεδομένου ότι πρόκειται για κατασκευή πρωτοτύπου προσαρμοσμένου επακριβώς στις ανάγκες που υπήρχαν.

3 Ανάλυση του προγράμματος του μικροελεγκτή

Το πρόγραμμα του μικροελεγκτή αναπτύχθηκε σε περιβάλλον CodevisionAVR, το οποίο βασίζεται στην γλώσσα προγραμματισμού ANSI C, τροποποιημένη κατάλληλα ώστε να συνεργάζεται με μικροελεγκτές AVR.

Ο μικροελεγκτής χρονίζεται χρησιμοποιώντας τον εσωτερικό ταλαντωτή 2MHz προκειμένου να επιτευχθεί χαμηλή κατανάλωση ρεύματος και μείωση του θορύβου στις μετρήσεις.

Σε γενικές γραμμές αποφεύχθηκε η χρήση των εξελιγμένων λειτουργιών που διαθέτει ο μικροελεγκτής λόγω του ότι τα ολοκληρωμένα που υπήρχαν ανήκαν σε πρώιμες εκδόσεις πυριτίου κι ως εκ τούτου υπάρχει ένας σχετικά μεγάλος αριθμός προβλημάτων στην λειτουργία αρκετών υποσυστημάτων όπως μπορεί να διαπιστώσει κανείς ανατρέχοντας στο αντίστοιχο παράρτημα του datasheet. Επιπλέον η απλούστερη υλοποίηση διευκολύνει την πραγματοποίηση αλλαγών και αναπροσαρμογών, το οποίο είναι επιθυμητό δεδομένου ότι η τεχνολογία BERA βρίσκεται ακόμα σε πειραματικό στάδιο και δεν έχει καθοριστεί επακριβώς η συσχέτιση ανάμεσα στην είσοδο και την έξοδο καθώς και άλλες λειτουργικές παράμετροι.

Στις επόμενες ενότητες παρουσιάζονται τα επιμέρους τμήματα του κώδικα που αντιστοιχεί στην πιο πρόσφατη έκδοση του προγράμματος.

3.1 Δήλωση επικεφαλίδων, μεταβλητών & σταθερών

Η δήλωση των απαραίτητων αρχείων επικεφαλίδων αποτελείται από τις παρακάτω γραμμές κώδικα:

```
// I/O Registers definitions
#include <io.h> (δηλώσεις καταχωρητών εισόδου-εξόδου)

// Standard Input/Output functions
#include <stdio.h> (Συναρτήσεις I/O σειριακής, strings & αρχείων)
// Delay functions
#include <delay.h> (Συναρτήσεις δημιουργίας καθυστερήσεων)

// Alphanumeric LCD interface functions
#include <alcd.h> (Συναρτήσεις χειρισμού αλφαριθμητικών οθονών LCD)
```

```
//Sleep functions
#include <sleep.h> (Συναρτήσεις εξοικονόμησης ενέργειας)

//Stdlib functions
#include <stdlib.h> (Συναρτήσεις μετατροπής τύπων, διαχείρισης μνήμης κα)
```

Σημειώνεται ότι μέσω του io.h συμπεριλαμβάνονται και άλλα αρχεία επικεφαλίδων, δηλαδή το <xmega64a3.h> που περιέχει δηλώσεις περιφερειακών, αντιστοιχήσεις καταχωρητών και διανυσμάτων διακοπών, το <xmstruct.h> που περιέχει δομές για τον χειρισμό των περιφερειακών και το <xmbits_a3.h> που περιέχει ονομασίες των bits των καταχωρητών του μικροελεγκτή, έτοιμες μάσκες bits κ.λπ.

Ακολουθούν οι δηλώσεις των σταθερών που χρησιμοποιούνται στο πρόγραμμα:

```
// Declare constants
(Παραμετροι ευθείας E.T. για την προσαρμογή δεδομένων)
#define LS_SLOPE_B 0.98395
#define LS_YP_B -0.48669
#define LS_SLOPE_S 1.00359
#define LS_YP_S -0.51164

#define INTRO_DELAY 500 (Ορισμός καθυστέρησης για τις οθόνες εκκίνησης)
#define DELAY_CNT 250 (Ορισμός περιόδου λήψης μετρήσεων σε ms)
#define MEAN_LENGTH 3 (Ορισμός # μετρήσεων για φίλτρο μέσου όρου)
```

Τέλος οι δηλώσεις των γενικών μεταβλητών:

```
// Declare global variables
#if (MEAN_LENGTH>=15)
signed long int averaging_sum[8]; (Πίνακας με το άθροισμα των μετρήσεων)
#else
signed int averaging_sum[8];
#endif

#if (DELAY_CNT>=255)
unsigned int counter; (Μετρητής για τον timer)
#else
unsigned char counter;
#endif

char i,j,k,str[15],temp[15]; (Μετρητές & προσωρινές μεταβλητές/πίνακες)
```



```
signed int data[8][3]; (Πίνακας προσωρινής αποθήκευσης μετρήσεων)
unsigned long t_count; (Μετρητής συνολικού χρόνου)
signed long t_sum[8]; (Πίνακας αποθήκευσης συσσωρευτικών αθροισμάτων)
```

Για την εξοικονόμηση χώρου στην SRAM και αποφυγή περιττών πράξεων με μεγάλους τύπους μεταβλητών έχει χρησιμοποιηθεί η δομή μεταγλωττισμού υπό όρους (`#if...#endif`). Αναλόγως της τιμής που έχει δοθεί στις αντίστοιχες σταθερές από τον προγραμματιστή επιλέγεται από τον μεταγλωττιστή ο τύπος που θα χρησιμοποιηθεί για τον πίνακα `averaging_sum` και την μεταβλητή `counter`.

3.2 Αρχικοποίηση περιφερειακών

Για την αρχικοποίηση των περιφερειακών χρησιμοποιήθηκε ο κώδικας που παράγεται από το Codewizard με μικρές τροποποιήσεις.

3.2.1 Αρχικοποίηση πηγών χρονισμού συστήματος

Ενεργοποίηση του εσωτερικού ταλαντωτή στα 2MHz και ρύθμιση διαιρετών συχνότητας για τα περιφερειακά:

```
// System Clocks initialization
void system_clocks_init(void)
{
    unsigned char n,s;
    (προσωρινές μεταβλητές, s→αποθήκευση καταχωρητή κατάστασης, n→ Με
    την αλλαγή του CCP καταχωρητή επιτρέπεται η πρόσβαση στους
    καταχωρητές που προστατεύει μόνο για 4 clock cycles, γι' αυτό
    χρησιμοποιείται η μεταβλητή n για την προσωρινή αποθήκευση ρυθμίσεων)

    // Optimize for speed
    #pragma optimize-
    // Save interrupts enabled/disabled state
    s=SREG;
    // Disable interrupts
    #asm("cli")

    // Internal 2 MHz RC oscillator initialization
```

Ανάλυση του προγράμματος του μικροελεγκτή

```
// Enable the internal 2 MHz RC oscillator
OSC.CTRL |= OSC_RC2MEN_bm;

// System Clock prescaler A division factor: 1
// System Clock prescalers B & C division factors: B:1, C:1
// ClkPer4: 2000,000 kHz
// ClkPer2: 2000,000 kHz
// ClkPer: 2000,000 kHz
// ClkCPU: 2000,000 kHz
n=(CLK.PSCTRL & ~(CLK_PSADIV_gm | CLK_PSBCDIV1_bm |
CLK_PSBCDIV0_bm))) | CLK_PSADIV_1_gc | CLK_PSBCDIV_1_1_gc;
CCP=CCP_IOREG_gc;
CLK.PSCTRL=n;
// Disable the autocalibration of the internal 2 MHz RC oscillator
DFLLRC2M.CTRL&= ~DFLL_ENABLE_bm;

// Wait for the internal 2 MHz RC oscillator to stabilize
while ((OSC.STATUS & OSC_RC2MRDY_bm)==0);

// Select the system clock source: 2 MHz Internal RC Osc.
n=(CLK.CTRL & (~CLK_SCLKSEL_gm)) | CLK_SCLKSEL_RC2M_gc;
CCP=CCP_IOREG_gc;
CLK.CTRL=n;

// Disable the unused oscillators: 32 MHz, 32 kHz, external clock/crystal
oscillator, PLL
OSC.CTRL&= ~(OSC_RC32MEN_bm | OSC_RC32KEN_bm | OSC_XOSCEN_bm
| OSC_PLEN_bm);

// Peripheral Clock output: Disabled
PORTCFG.CLKEVOUT=(PORTCFG.CLKEVOUT & (~PORTCFG_CLKOUT_gm)) |
PORTCFG_CLKOUT_OFF_gc;

// Restore interrupts enabled/disabled state
SREG=s;
// Restore optimization for size if needed
#pragma optimize_default
}
```

3.2.2 Αρχικοποίηση πορτών I/O

Στις πόρτες A και B που χρησιμοποιούνται για την λήψη των αναλογικών μετρήσεων απενεργοποιήθηκαν τα ψηφιακά buffer ενώ οι υπόλοιπες πόρτες ρυθμίστηκαν ως είσοδοι και ενεργοποιήθηκαν οι εσωτερικές pull-up προκειμένου να ελαχιστοποιηθεί η κατανάλωση ισχύος από τον μικροελεγκτή:

```
// Ports initialization
void ports_init(void)
{
// PORTA initialization (ομοίως αρχικοποιείται η πόρτα B)
// OUT register
PORTA.OUT=0x00;
// Bit0: Input
// Bit1: Input
// Bit2: Input
// Bit3: Input
// Bit4: Input
// Bit5: Input
// Bit6: Input
// Bit7: Input
PORTA.DIR=0x00;
// Bit0 Output/Pull configuration: Totempole/No
// Bit0 Input/Sense configuration: Input buffer disabled
// Bit0 inverted: Off
// Bit0 slew rate limitation: Off
PORTA.PIN0CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit1 Output/Pull configuration: Totempole/No
// Bit1 Input/Sense configuration: Input buffer disabled
// Bit1 inverted: Off
// Bit1 slew rate limitation: Off
PORTA.PIN1CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit2 Output/Pull configuration: Totempole/No
// Bit2 Input/Sense configuration: Input buffer disabled
// Bit2 inverted: Off
// Bit2 slew rate limitation: Off
PORTA.PIN2CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit3 Output/Pull configuration: Totempole/No
// Bit3 Input/Sense configuration: Input buffer disabled
// Bit3 inverted: Off
// Bit3 slew rate limitation: Off
PORTA.PIN3CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
```

```
// Bit4 Output/Pull configuration: Totempole/No
// Bit4 Input/Sense configuration: Input buffer disabled
// Bit4 inverted: Off
// Bit4 slew rate limitation: Off
PORTA.PIN4CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit5 Output/Pull configuration: Totempole/No
// Bit5 Input/Sense configuration: Input buffer disabled
// Bit5 inverted: Off
// Bit5 slew rate limitation: Off
PORTA.PIN5CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit6 Output/Pull configuration: Totempole/No
// Bit6 Input/Sense configuration: Input buffer disabled
// Bit6 inverted: Off
// Bit6 slew rate limitation: Off
PORTA.PIN6CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit7 Output/Pull configuration: Totempole/No
// Bit7 Input/Sense configuration: Input buffer disabled
// Bit7 inverted: Off
// Bit7 slew rate limitation: Off
PORTA.PIN7CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Interrupt 0 level: Disabled
// Interrupt 1 level: Disabled
PORTA.INTCTRL=(PORTA.INTCTRL & ~(PORT_INT1LVL_gm | PORT_INT0LVL_gm))
| PORT_INT1LVL_OFF_gc | PORT_INT0LVL_OFF_gc;
// Bit0 pin change interrupt 0: Off
// Bit1 pin change interrupt 0: Off
// Bit2 pin change interrupt 0: Off
// Bit3 pin change interrupt 0: Off
// Bit4 pin change interrupt 0: Off
// Bit5 pin change interrupt 0: Off
// Bit6 pin change interrupt 0: Off
// Bit7 pin change interrupt 0: Off
PORTA.INTOMASK=0x00;
// Bit0 pin change interrupt 1: Off
// Bit1 pin change interrupt 1: Off
// Bit2 pin change interrupt 1: Off
// Bit3 pin change interrupt 1: Off
// Bit4 pin change interrupt 1: Off
// Bit5 pin change interrupt 1: Off
// Bit6 pin change interrupt 1: Off
// Bit7 pin change interrupt 1: Off
```

```
PORTA.INT1MASK=0x00;

// PORTD initialization (ομοίως αρχικοποιούνται οι πόρτες C,E,F,R)
// OUT register
PORTD.OUT=0xFF;
// Bit0: Input
// Bit1: Input
// Bit2: Input
// Bit3: Input
// Bit4: Input
// Bit5: Input
// Bit6: Input
// Bit7: Input
PORTD.DIR=0x00;
// Bit0 Output/Pull configuration: Totempole/No
// Bit0 Input/Sense configuration: Sense both edges
// Bit0 inverted: Off
// Bit0 slew rate limitation: Off
PORTD.PIN0CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit1 Output/Pull configuration: Totempole/No
// Bit1 Input/Sense configuration: Sense both edges
// Bit1 inverted: Off
// Bit1 slew rate limitation: Off
PORTD.PIN1CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit2 Output/Pull configuration: Totempole/No
// Bit2 Input/Sense configuration: Sense both edges
// Bit2 inverted: Off
// Bit2 slew rate limitation: Off
PORTD.PIN2CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit3 Output/Pull configuration: Totempole/No
// Bit3 Input/Sense configuration: Sense both edges
// Bit3 inverted: Off
// Bit3 slew rate limitation: Off
PORTD.PIN3CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit4 Output/Pull configuration: Totempole/No
// Bit4 Input/Sense configuration: Sense both edges
// Bit4 inverted: Off
// Bit4 slew rate limitation: Off
PORTD.PIN4CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit5 Output/Pull configuration: Totempole/No
// Bit5 Input/Sense configuration: Sense both edges
// Bit5 inverted: Off
```

Ανάλυση του προγράμματος του μικροελεγκτή

```
// Bit5 slew rate limitation: Off
PORTD.PIN5CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit6 Output/Pull configuration: Totempole/No
// Bit6 Input/Sense configuration: Sense both edges
// Bit6 inverted: Off
// Bit6 slew rate limitation: Off
PORTD.PIN6CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit7 Output/Pull configuration: Totempole/No
// Bit7 Input/Sense configuration: Sense both edges
// Bit7 inverted: Off
// Bit7 slew rate limitation: Off
PORTD.PIN7CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Interrupt 0 level: Disabled
// Interrupt 1 level: Disabled
PORTD.INTCTRL=(PORTD.INTCTRL & ~(PORT_INT1LVL_gm | PORT_INT0LVL_gm))
| PORT_INT1LVL_OFF_gc | PORT_INT0LVL_OFF_gc;
// Bit0 pin change interrupt 0: Off
// Bit1 pin change interrupt 0: Off
// Bit2 pin change interrupt 0: Off
// Bit3 pin change interrupt 0: Off
// Bit4 pin change interrupt 0: Off
// Bit5 pin change interrupt 0: Off
// Bit6 pin change interrupt 0: Off
// Bit7 pin change interrupt 0: Off
PORTD.INTOMASK=0x00;
// Bit0 pin change interrupt 1: Off
// Bit1 pin change interrupt 1: Off
// Bit2 pin change interrupt 1: Off
// Bit3 pin change interrupt 1: Off
// Bit4 pin change interrupt 1: Off
// Bit5 pin change interrupt 1: Off
// Bit6 pin change interrupt 1: Off
// Bit7 pin change interrupt 1: Off
PORTD.INT1MASK=0x00;
```

Αντιστοίχιση πορτών A,B,C,D σε εικονικές για άμεση πρόσβαση με εντολές που μπορούν να χρησιμοποιηθούν μόνο στην I/O περιοχή της μνήμης:

```
// Virtual Ports initialization
```

```
void vports_init(void)
{
```

```
// PORTA mapped to VPORT0
// PORTB mapped to VPORT1
PORTCFG.VPCTRLA=PORTCFG_VP1MAP_PORTB_gc |
PORTCFG_VP0MAP_PORTA_gc;
// PORTC mapped to VPORT2
// PORTD mapped to VPORT3
PORTCFG.VPCTRLB=PORTCFG_VP3MAP_PORTD_gc |
PORTCFG_VP2MAP_PORTC_gc;
}
```

Σημ: Μόνο η πόρτα D χρησιμοποιείται ως ψηφιακή είσοδος στο πρόγραμμα.

3.2.3 Αρχικοποίηση χρονιστή/απαριθμητή (timer/counter) E0

Ο χρονιστής E0 ρυθμίστηκε σε 8bit mode, μιας που δεν είναι απαραίτητη η χρήση 16-bit ανάλυσης, με περίοδο υπερχείλισης 1msec και πραγματοποίηση διακοπής χαμηλού επιπέδου όποτε γίνεται υπερχείλιση του καταχωρητή.

```
// Timer/Counter TCE0 initialization
void tce0_init(void)
{
    unsigned char s;
    unsigned char n;

    // Note: the correct PORTE direction for the Compare Channels outputs
    // is configured in the ports_init function

    // Save interrupts enabled/disabled state
    s=SREG;
    // Disable interrupts
    #asm("cli")

    // Disable and reset the timer/counter just to be sure
    tc0_disable(&TCE0);
    // Clock source: Peripheral Clock/8      (2MHz/8→250KHz timer clock)
    TCE0.CTRLA=(TCE0.CTRLA & (~TC0_CLKSEL_gm)) | TC_CLKSEL_DIV8_gc;
    // Mode: Normal Operation, Overflow Int./Event on TOP
    // Compare/Capture on channel A: Off
    // Compare/Capture on channel B: Off
    // Compare/Capture on channel C: Off
    // Compare/Capture on channel D: Off
}
```

```
TCE0.CTRLB=(TCE0.CTRLB & ~(TC0_CCAEN_bm | TC0_CCBEN_bm |
TC0_CCCEN_bm | TC0_CCDEN_bm | TC0_WGMODE_gm))) |
    TC_WGMODE_NORMAL_gc;

// Capture event source: None
// Capture event action: None
TCE0.CTRLD=(TCE0.CTRLD & ~(TC0_EVACT_gm | TC0_EVSEL_gm))) |
    TC_EVACT_OFF_gc | TC_EVSEL_OFF_gc;

// Timer/counter in 8bit mode: On
TCE0.CTRLE|=TC0_BYTEM_bm;

// Overflow interrupt: Low Level
// Error interrupt: Disabled
TCE0.INTCTRLA=(TCE0.INTCTRLA & ~(TC0_ERRINTLVL_gm |
TC0_OVFINTLVL_gm))) |
    TC_ERRINTLVL_OFF_gc | TC_OVFINTLVL_LO_gc;

// Compare/Capture channel A interrupt: Disabled
// Compare/Capture channel B interrupt: Disabled
// Compare/Capture channel C interrupt: Disabled
// Compare/Capture channel D interrupt: Disabled
TCE0.INTCTRLB=(TCE0.INTCTRLB & ~(TC0_CCDINTLVL_gm |
TC0_CCCINTLVL_gm | TC0_CCBINTLVL_gm | TC0_CCAINTLVL_gm))) |
    TC_CCDINTLVL_OFF_gc | TC_CCCINTLVL_OFF_gc |
TC_CCBINTLVL_OFF_gc | TC_CCAINTLVL_OFF_gc;

// High resolution extension: Off
HIRESE.CTRL&= ~HIRES_HREN0_bm;

// Advanced Waveform Extension initialization
// Optimize for speed
#pragma optimize-
// Disable locking the AWEX configuration registers just to be sure
n=MCU.AWEXLOCK & (~MCU_AWEXELOCK_bm);
CCP=CCP_IOREG_gc;
MCU.AWEXLOCK=n;
// Restore optimization for size if needed
#pragma optimize_default

// Pattern generation: Off
// Dead time insertion: Off
```


Ανάλυση του προγράμματος του μικροελεγκτή

```
AWEXE.CTRL&= ~(AWEX_PGM_bm | AWEX_CWCM_bm |
AWEX_DTICCDEN_bm | AWEX_DTICCCEN_bm | AWEX_DTICCBEN_bm |
AWEX_DTICCAEN_bm);

// Fault protection initialization
// Fault detection on OCD Break detection: On
// Fault detection restart mode: Latched Mode
// Fault detection action: None (Fault protection disabled)
AWEXE.FDCTRL=(AWEXE.FDCTRL & ~(AWEX_FDDBD_bm |
AWEX_FDMODE_bm | AWEX_FDACT_gm))) |
    AWEX_FDACT_NONE_gc;
// Fault detect events:
// Event channel 0: Off
// Event channel 1: Off
// Event channel 2: Off
// Event channel 3: Off
// Event channel 4: Off
// Event channel 5: Off
// Event channel 6: Off
// Event channel 7: Off
AWEXE.FDEVMASK=0b00000000;
// Make sure the fault detect flag is cleared
AWEXE.STATUS |= AWEXE.STATUS & AWEX_FDF_bm;

// Clear the interrupt flags
TCE0.INTFLAGS=TCE0.INTFLAGS;
// Set counter register
TCE0.CNT=0x0000;
// Set period register
TCE0.PER=0x00F9; (F9→250 counts, δηλαδή 1msec περίοδος)
// Set channel A Compare/Capture register
TCE0.CCA=0x0000;
// Set channel B Compare/Capture register
TCE0.CCB=0x0000;
// Set channel C Compare/Capture register
TCE0.CCC=0x0000;
// Set channel D Compare/Capture register
TCE0.CCD=0x0000;

// Restore interrupts enabled/disabled state
SREG=s;
}
```

Σε κάθε υπερχείλιση του timer E0 εκτελείται η παρακάτω ρουτίνα εξυπηρέτησης της διακοπής με αποτέλεσμα να αυξάνεται η μεταβλητή counter κάθε 1msec ενώ παράλληλα επαναφέρεται ο μικροελεγκτής από κατάσταση εξοικονόμησης ενέργειας σε κανονική λειτουργία.

```
// Timer/counter TCE0 Overflow/Underflow interrupt service routine
interrupt [TCE0_OVF_vect] void tce0_overflow_isr(void)
{
  counter++;
}
```

3.2.4 Αρχικοποίηση σειριακής θύρας

Για την μεταφορά δεδομένων από/προς Η/Υ χρησιμοποιείται η σειριακή θύρα F0, που αντιστοιχεί στα pin 2,3 (Rx,Tx) της θύρας F.

Η μεταφορά δεδομένων πραγματοποιείται με ρυθμό 115200bps, με πλαίσιο 8bit δεδομένων, 1 bit τερματισμού και χωρίς bit ισοτιμίας.

Η πραγματοποίηση σειριακών επικοινωνιών χωρίς την χρήση κρυσταλλικού ταλαντωτή κατάλληλης συχνότητας, όπως συνηθίζεται προκειμένου να επιτυγχάνεται ικανοποιητικά χαμηλό ποσοστό σφαλμάτων μετάδοσης, είναι εφικτή λόγω της υψηλής ακριβείας του ενσωματωμένου RC ταλαντωτή στους μικροελεγκτές της σειράς xmega καθώς και του κλασματικού διαιρέτη που χρησιμοποιείται για τον χρονισμό των σειριακών θυρών (fractional baud rate generator), χαρακτηριστικό που μέχρι πρότινος υπήρχε μόνο σε μικροεπεξεργαστές διαφορετικής κατηγορίας και υψηλότερου κόστους.

Το μέσο σφάλμα μετάδοσης δεδομένων διατηρείται κάτω από το όριο του 2% που επιβάλλεται για να θεωρείται η σειριακή επικοινωνία επαρκούς αξιοπιστίας.

```
// USARTF0 initialization
void usartf0_init(void)
{
  // Note: the correct PORTF direction for the RxD, TxD and XCK signals
  // is configured in the ports_init function

  // Transmitter is enabled
  // Set TxD=1
  PORTF.OUTSET=0x08;

  // Communication mode: Asynchronous USART
  // Data bits: 8
```

Ανάλυση του προγράμματος του μικροελεγκτή

```
// Stop bits: 1
// Parity: Disabled
USARTF0.CTRLA=USART_CMODE_ASYNCHRONOUS_gc |
USART_PMODE_DISABLED_gc | USART_CHSIZE_8BIT_gc;

// Receive complete interrupt: Disabled
// Transmit complete interrupt: Disabled
// Data register empty interrupt: Disabled
USARTF0.CTRLA=(USARTF0.CTRLA & ~(USART_RXCINTLVL_gm |
USART_TXCINTLVL_gm | USART_DREINTLVL_gm))) |
    USART_RXCINTLVL_OFF_gc | USART_TXCINTLVL_OFF_gc |
USART_DREINTLVL_OFF_gc;

// Required Baud rate: 115200
// Real Baud Rate: 115107,9 (x1 Mode), Error: 0,1 %
USARTF0.BAUDCTRLA=0x0B;
USARTF0.BAUDCTRLB=((0x09 << USART_BSCALE_bp) & USART_BSCALE_gm) |
0x00;

// Receiver: On
// Transmitter: On
// Double transmission speed mode: Off
// Multi-processor communication mode: Off
USARTF0.CTRLB=(USARTF0.CTRLB & ~(USART_RXEN_bm | USART_TXEN_bm |
USART_CLK2X_bm | USART_MPCM_bm | USART_TXB8_bm))) |
    USART_RXEN_bm | USART_TXEN_bm;
}

// Receive a character from USARTF0
// USARTF0 is used as the default input device by the 'getchar' function
#define _ALTERNATE_GETCHAR_

#pragma used+
char getchar(void)
{
char data;
unsigned char status;

while (1)
{
while (((status=USARTF0.STATUS) & USART_RXCIF_bm) == 0);
data=USARTF0.DATA;
```

```
    if ((status & (USART_FERR_bm | USART_PERR_bm | USART_BUFOVF_bm)) ==
0) return data;
    }
}
#pragma used-

// Write a character to the USARTF0 Transmitter
// USARTF0 is used as the default output device by the 'putchar' function
#define _ALTERNATE_PUTCHAR_

#pragma used+
void putchar(char c)
{
while ((USARTF0.STATUS & USART_DREIF_bm) == 0);
USARTF0.DATA=c;
}
#pragma used-
```

3.2.5 Αρχικοποίηση A/D μετατροπέων

Οι μετατροπείς A/D είναι ρυθμισμένοι για λήψη διαφορικών μετρήσεων με 12bit ανάλυση χρησιμοποιώντας την εσωτερική τάση αναφοράς 1,0V με αποτέλεσμα δυναμικό εύρος μετρήσεων $\pm 1V$. Η συχνότητα λειτουργίας των ADC έχει ρυθμιστεί σε 3,906 kHz προκειμένου να διατηρηθεί ο θόρυβος σε χαμηλά επίπεδα.

Κάθε ένας από τους A/D μετατροπείς παρέχει 4 διαφορικά κανάλια μέτρησης τα όποια υλοποιούνται μετρώντας την διαφορά τάσης ανάμεσα στους ακροδέκτες 4-0, 5-1, 6-2, 7-3 αντίστοιχα. Ακροδέκτης αναφοράς θεωρείται εκείνος με τον μικρότερο αριθμό για κάθε κανάλι. Ακολουθεί η αρχικοποίηση του ADCA.

```
// Function used to read the calibration byte from the
// signature row, specified by 'index'
#pragma optsize-
unsigned char read_calibration_byte(unsigned char index)
{
unsigned char r;
NVM.CMD=NVM_CMD_READ_CALIB_ROW_gc;
r=((flash unsigned char*) index);
// Clean up NVM command register
NVM.CMD=NVM_CMD_NO_OPERATION_gc;
```

```
return r;
}
#pragma optimize_default

// ADCA initialization

// Variable used to store the ADC offset
// for 12 Bit Signed conversion mode
signed char adca_offset;

void adca_init(void)
{
unsigned char i;
int ofs;

// ADCA is enabled
// Resolution: 12 Bits
// Load the calibration value for 12 Bit resolution
// from the signature row
ADCA.CALL=read_calibration_byte(PROD_SIGNATURES_START+ADCACAL0_of
fset);
ADCA.CALH=read_calibration_byte(PROD_SIGNATURES_START+ADCACAL1_of
fset);

// Free Running mode: Off
// Conversion mode: Signed
ADCA.CTRLB=(ADCA.CTRLB & ~(ADC_CONMODE_bm | ADC_FREERUN_bm |
ADC_RESOLUTION_gm))) |
    ADC_CONMODE_bm | ADC_RESOLUTION_12BIT_gc;

// Clock frequency: 3,906 kHz
ADCA.PRESCALER=(ADCA.PRESCALER & (~ADC_PRESCALER_gm)) |
ADC_PRESCALER_DIV512_gc;

// Reference: Internal 1.00 V
// Temperature reference: Off
ADCA.REFCTRL=(ADCA.REFCTRL & ((~(ADC_REFSEL_gm | ADC_TEMPREF_bm))
| ADC_BANDGAP_bm)) |
    ADC_REFSEL_INT1V_gc | ADC_BANDGAP_bm;
// Read and save the ADC offset using channel 0
ADCA.CH0.CTRL=(ADCA.CH0.CTRL & ~(ADC_CH_START_bm |
ADC_CH_GAINFAC_gm | ADC_CH_INPUTMODE_gm))) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
```

```
ADCA.CH0.MUXCTRL=(ADCA.CH0.MUXCTRL & (~{ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm})) |
    ADC_CH_MUXPOS_PIN0_gc | ADC_CH_MUXNEG_PIN0_gc;
// Enable the ADC in order to read the offset
ADCA.CTRLA |=ADC_ENABLE_bm;
// Insert a delay to allow the ADC common mode voltage to stabilize
delay_us(2);
// Perform several offset measurements and store the mean value
offs=0;
for (i=0; i<16; i++)
{
    // Start the AD conversion on channel 0
    ADCA.CH0.CTRL |=ADC_CH_START_bm;
    // Wait for the AD conversion to complete
    while ((ADCA.CH0.INTFLAGS & ADC_CH_CHIF_bm)==0);
    // Clear the interrupt flag
    ADCA.CH0.INTFLAGS=ADC_CH_CHIF_bm;
    // Read the offset
    offs+=(signed char) ADCA.CH0.RESL;
}
// Disable the ADC
ADCA.CTRLA&= ~ADC_ENABLE_bm;
// Store the mean value of the offset
adca_offset=(signed char) (offs/16);

// Initialize the ADC Compare register
ADCA.CMPL=0x00;
ADCA.CMPH=0x00;

// ADC channel 0 gain: 1
// ADC channel 0 input mode: Differential input signal
ADCA.CH0.CTRL=(ADCA.CH0.CTRL & (~{ADC_CH_START_bm |
ADC_CH_GAINFAC_gm | ADC_CH_INPUTMODE_gm})) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;

// ADC channel 0 positive input: ADC4 pin
// ADC channel 0 negative input: ADC0 pin
ADCA.CH0.MUXCTRL=(ADCA.CH0.MUXCTRL & (~{ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm})) |
    ADC_CH_MUXPOS_PIN4_gc | ADC_CH_MUXNEG_PIN0_gc;

// ADC channel 1 gain: 1
```

Ανάλυση του προγράμματος του μικροελεγκτή

```
// ADC channel 1 input mode: Differential input signal
ADCA.CH1.CTRL=(ADCA.CH1.CTRL & ~(ADC_CH_START_bm |
ADC_CH_GAINFAC_gm | ADC_CH_INPUTMODE_gm))) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;

// ADC channel 1 positive input: ADC5 pin
// ADC channel 1 negative input: ADC1 pin
ADCA.CH1.MUXCTRL=(ADCA.CH1.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm))) |
    ADC_CH_MUXPOS_PIN5_gc | ADC_CH_MUXNEG_PIN1_gc;

// ADC channel 2 gain: 1
// ADC channel 2 input mode: Differential input signal
ADCA.CH2.CTRL=(ADCA.CH2.CTRL & ~(ADC_CH_START_bm |
ADC_CH_GAINFAC_gm | ADC_CH_INPUTMODE_gm))) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;

// ADC channel 2 positive input: ADC6 pin
// ADC channel 2 negative input: ADC2 pin
ADCA.CH2.MUXCTRL=(ADCA.CH2.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm))) |
    ADC_CH_MUXPOS_PIN6_gc | ADC_CH_MUXNEG_PIN2_gc;

// ADC channel 3 gain: 1
// ADC channel 3 input mode: Differential input signal
ADCA.CH3.CTRL=(ADCA.CH3.CTRL & ~(ADC_CH_START_bm |
ADC_CH_GAINFAC_gm | ADC_CH_INPUTMODE_gm))) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;

// ADC channel 3 positive input: ADC7 pin
// ADC channel 3 negative input: ADC3 pin
ADCA.CH3.MUXCTRL=(ADCA.CH3.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm))) |
    ADC_CH_MUXPOS_PIN7_gc | ADC_CH_MUXNEG_PIN3_gc;

// AD conversion is started by software
ADCA.EVCTRL=ADC_EVACT_NONE_gc;

// Channel 0 interrupt: Disabled
ADCA.CH0.INTCTRL=(ADCA.CH0.INTCTRL & ~(ADC_CH_INTMODE_gm |
ADC_CH_INTLVL_gm))) |
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Channel 1 interrupt: Disabled
```

```
ADCA.CH1.INTCTRL=(ADCA.CH1.INTCTRL & ~(ADC_CH_INTMODE_gm |
ADC_CH_INTLVL_gm)) |
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Channel 2 interrupt: Disabled
ADCA.CH2.INTCTRL=(ADCA.CH2.INTCTRL & ~(ADC_CH_INTMODE_gm |
ADC_CH_INTLVL_gm)) |
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Channel 3 interrupt: Disabled
ADCA.CH3.INTCTRL=(ADCA.CH3.INTCTRL & ~(ADC_CH_INTMODE_gm |
ADC_CH_INTLVL_gm)) |
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;

// Enable the ADC
ADCA.CTRLA |= ADC_ENABLE_bm;
// Insert a delay to allow the ADC common mode voltage to stabilize
delay_us(2);
}

// ADCA channel data read function using polled mode
signed int adca_read(unsigned char channel)
{
    ADC_CH_t *pch=&ADCA.CH0+channel;
    signed int data;

    // Start the AD conversion
    pch->CTRL |= ADC_CH_START_bm;
    // Wait for the AD conversion to complete
    while ((pch->INTFLAGS & ADC_CH_CHIF_bm)==0);
    // Clear the interrupt flag
    pch->INTFLAGS=ADC_CH_CHIF_bm;
    // Read the AD conversion result
    ((unsigned char *) &data)[0]=pch->RESL;
    ((unsigned char *) &data)[1]=pch->RESH;
    // Compensate the ADC offset
    data-=adca_offset;
    return data;
}
```

Ομοίως αρχικοποιείται και ο ADCB και ορίζεται η συνάρτηση adcb_read η οποία επιστρέφει την μετρούμενη τιμή για το κανάλι που επιλέγεται.

3.3 Λήψη μετρήσεων

Η λήψη των μετρήσεων πραγματοποιείται από τον παρακάτω βρόχο επανάληψης. Λαμβάνονται μετρήσεις παράλληλα από τους 2 ενσωματωμένους A/D μετατροπείς και χρησιμοποιώντας κατάλληλα ορισμένους δείκτες αποθηκεύονται σε έναν κοινό πίνακα δεδομένων, με τα στοιχεία 0-3 του πίνακα να αντιστοιχούν στις μετρήσεις που λαμβάνονται από τον ADCA και τα στοιχεία 4-7 του πίνακα να αποτελούνται από τις μετρήσεις του ADCB αντίστοιχα. Σημειώνεται ότι ο πίνακας αποθήκευσης δεδομένων είναι δισδιάστατος [j][i] προκειμένου να υλοποιείται το φιλτράρισμα δεδομένων, για περισσότερες πληροφορίες βλ. §3.4.

```
for (j=0;j<=3;j++) //ch scan loop
{
    data[j][i]=adca_read(j); //ch 0-3
    data[j+4][i]=adcb_read(j); //ch 4-7
} //ch scan loop
```

3.4 Φιλτράρισμα μετρήσεων

Για το φιλτράρισμα των μετρήσεων και την απόρριψη θορύβου χρησιμοποιείται ένας συνδυασμός δυο απλών φίλτρων πεπερασμένης κρουστικής απόκρισης (FIR), ενός φίλτρου διάμεσης τιμής 3 εισόδων (median₃) και ενός φίλτρου μέσης τιμής (mean value) με μήκος φίλτρου που μπορεί να οριστεί αλλάζοντας την αντίστοιχη σταθερά (MEAN_LENGTH) στο πρόγραμμα του μικροελεγκτή.

Για την εφαρμογή δεν κρίθηκε απαραίτητο να υλοποιηθεί φίλτρο διάμεσου μεγαλύτερης τάξης αφού θα κλιμακώνονταν οι υπολογιστικές ανάγκες και θα έπρεπε να αυξηθεί η συχνότητα λειτουργίας του μικροελεγκτή για να μπορεί να ανταπεξέλθει.

Το φίλτρο διάμεσης τιμής υλοποιείται από την ακόλουθη συνάρτηση:

```
signed int median(signed int a, signed int b, signed int c)
{
    signed int med;
    if ((a <= b) && (a <= c))
    {
        med = (b <= c) ? b : c;
    }
    else if ((b <= a) && (b <= c))
```

```
{
  med = (a <= c) ? a : c;
}
else
{
  med = (a <= b) ? a : b;
}
return med; (στην μεταβλητή med περιέχεται η διάμεσος)
}
```

Το φίλτρο μέσης τιμής υλοποιείται με ένα απλό βρόχο ο οποίος αθροίζει το ζητούμενο πλήθος τιμών για το κάθε κανάλι αφού έχει βρεθεί η διάμεσος καλώντας την προαναφερθείσα συνάρτηση και αφού έχει γίνει προσαρμογή των δεδομένων καλώντας την συνάρτηση LS_fit (βλ. §3.5).

Δεδομένου ότι στην παρούσα φάση οι μετρήσεις ούτως η άλλως υπόκεινται σε περαιτέρω επεξεργασία σε H/Y αφού αποθηκευθούν προτιμήθηκε η απλούστερη δυνατή υλοποίηση έναντι ενός πραγματικού κινούμενου μέσου όρου προκειμένου να ελαχιστοποιηθούν οι απαραίτητες πράξεις και να εξοικονομηθεί μνήμη sram.

```
for (k=0;k<MEAN_LENGTH;k++)
(Λήψη μετρήσεων, βλ. §3.3)
  for (j=0;j<=7;j++) //find median for each ch and add
  {
    averaging_sum[j] += LS_fit(median(data[j][0],data[j][1],data[j][2]));
  } //ch scan loop
} //averaging loop
(προβολή αποτελεσμάτων και μηδενισμός του πίνακα averaging_sum[], βλ. §3.6)
```

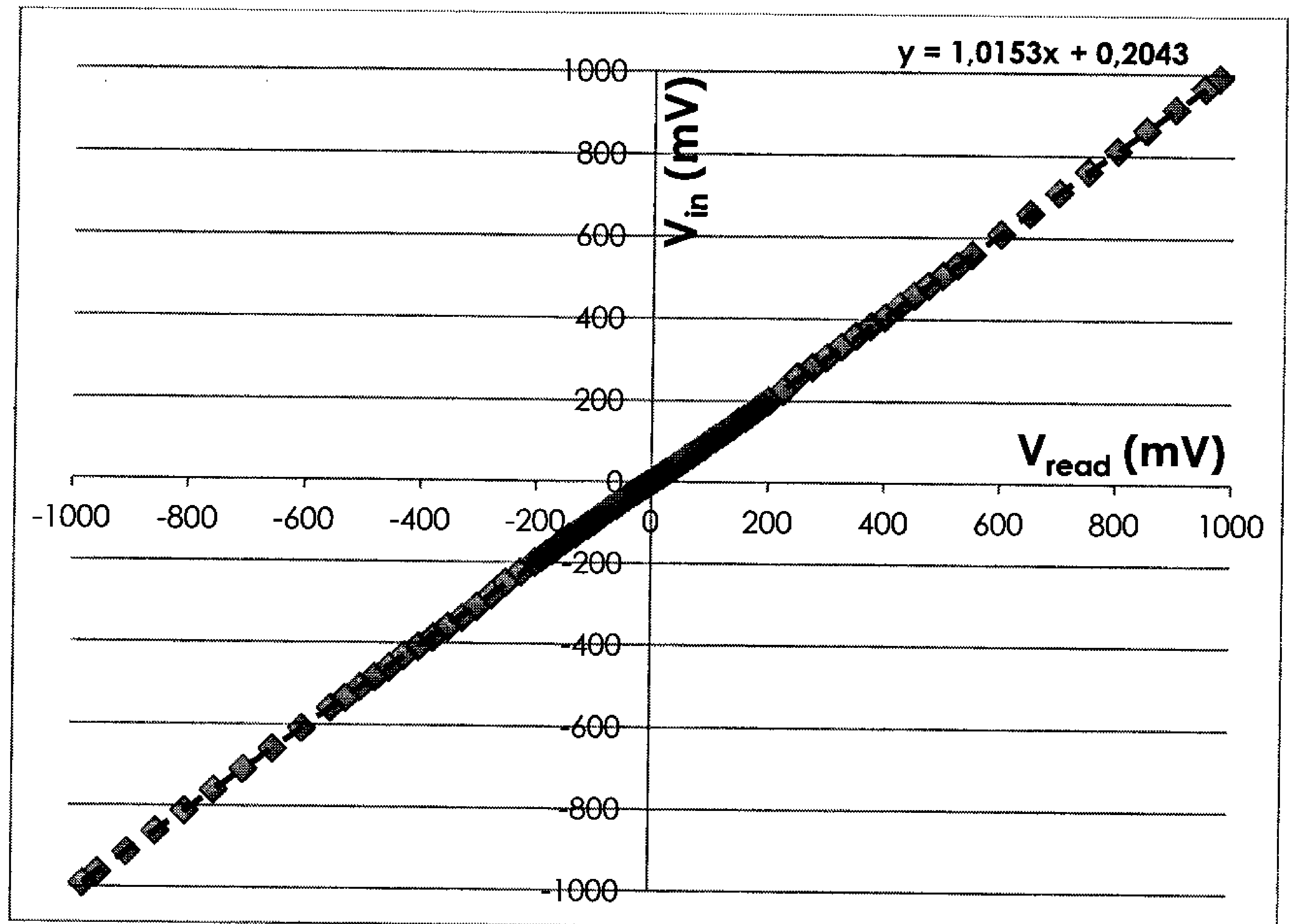
3.5 Προσαρμογή δεδομένων

Προκειμένου να διορθωθούν σφάλματα γραμμικότητας και μετατόπισης στις μετρήσεις που λαμβάνονται από τους A/D μετατροπείς χρησιμοποιείται η μέθοδος ελαχίστων τετραγώνων για την προσαρμογή των μετρήσεων.

Πραγματοποιήθηκε μια σειρά μετρήσεων παρέχοντας στην είσοδο του συστήματος μια τάση αναφοράς και σημειώνοντας τον μέσο όρο 500 μετρήσεων. Η διαδικασία επαναλήφθηκε μεταβάλλοντας την τάση αναφοράς ώστε να καλυφθεί σχεδόν πλήρως το εύρος τιμών εισόδου του συστήματος [-975mV έως +975mV] δίνοντας έμφαση στις κεντρικότερες περιοχές του συνόλου που είναι σημαντικότερες για την εφαρμογή.

Ανάλυση του προγράμματος του μικροελεγκτή

Αφού αποθηκεύτηκαν οι μετρήσεις γράφτηκε ένα απλό πρόγραμμα υπολογισμού της ευθείας ελαχίστων τετραγώνων και απεικόνισης των αποτελεσμάτων σε περιβάλλον MATLAB (παράρτημα C).



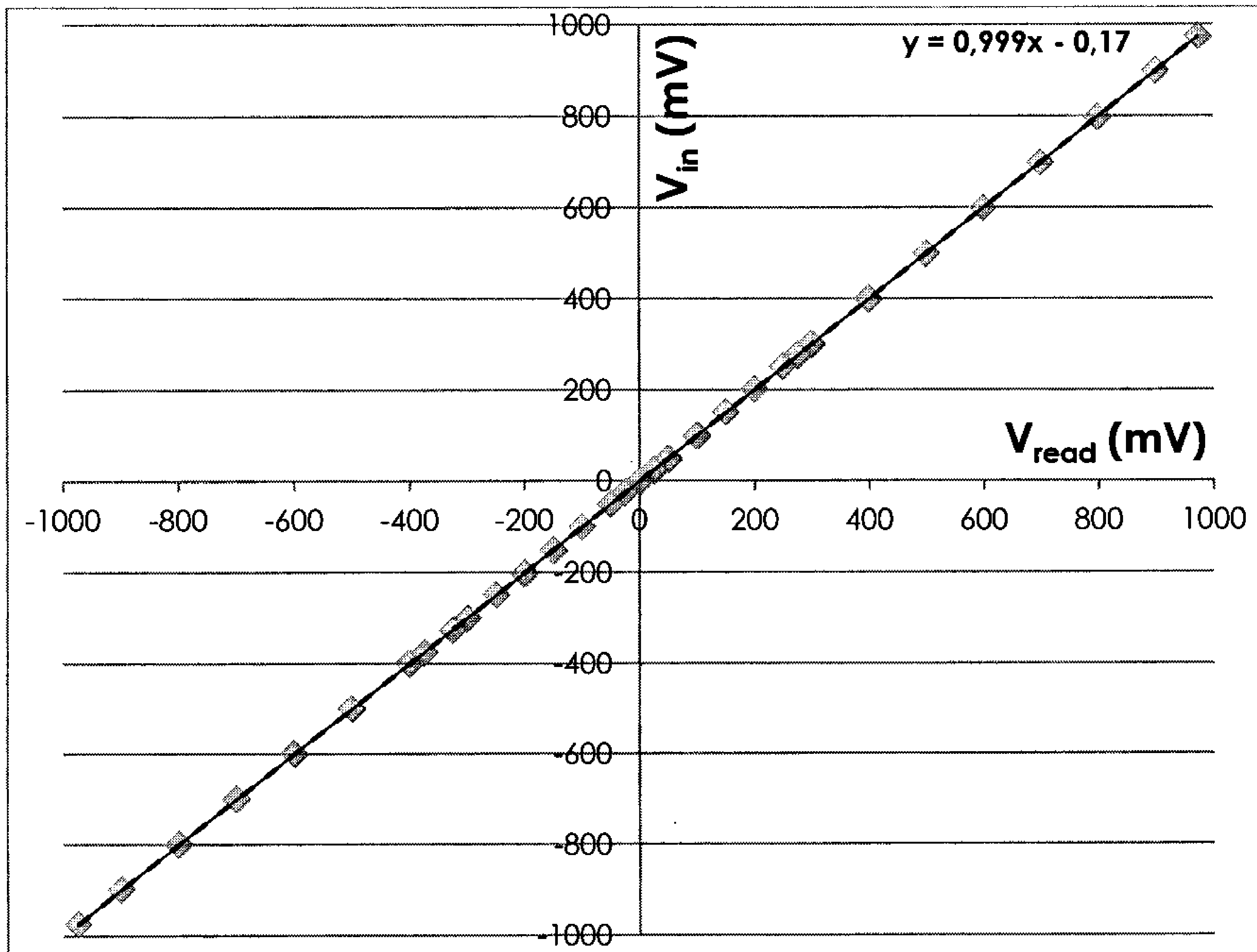
Εικόνα 3-I: Μετρήσεις ενσωματωμένου ADC χωρίς προσαρμογή δεδομένων

Όπως είναι προφανές από το παραπάνω διάγραμμα οι ενσωματωμένοι A/D μετατροπείς έχουν πολύ καλή γραμμικότητα και αρκετά χαμηλό μέσο σφάλμα, εξ' ου και ότι επιλέχτηκε η απλή μέθοδος ελαχίστων τετραγώνων. Ωστόσο για την βελτιστοποίηση της προσαρμογής κρίθηκε αναγκαίο να διαχωριστεί η περιοχή μέτρησης σε δυο τομείς, έναν για τάση εισόδου με $|V_{in}| < 250\text{mV}$ και ένα για $|V_{in}| \geq 250\text{mV}$.

Οι παράμετροι που υπολογίστηκαν δίνονται ως σταθερές στο πρόγραμμα του μικροελεγκτή, όπου LS_SLOPE η κλίση της ευθείας E.T. και LS_YP το σημείο τομής με τον Y άξονα. Το επίθεμα $_B$ δηλώνει ότι πρόκειται για τις παραμέτρους της περιοχής $|V_{in}| \geq 250\text{mV}$ ενώ το επίθεμα $_S$ αντιστοιχεί στην περιοχή $|V_{in}| < 250\text{mV}$.

Χρησιμοποιώντας τις τιμές που υπολογίστηκαν για προσαρμογή των δεδομένων επαναλήφθηκε η διαδικασία μέτρησης και τα αποτελέσματα παρουσιάζονται στο διάγραμμα 3-II.

Ανάλυση του προγράμματος του μικροελεγκτή



Εικόνα 3-II: Μετρήσεις ενσωματωμένου ADC με προσαρμογή δεδομένων 2 περιοχών

Ο συντελεστής προσδιορισμού R^2 και το μέσο απόλυτο σχετικό σφάλμα % (mare %) για τα παραπάνω δεδομένα επιβεβαιώνουν ότι η προσαρμογή των μετρήσεων είναι άριστη.

mare %	0,2407239
R^2	0,9999946

Σημειώνεται ότι προκειμένου να εξοικονομηθεί χώρος στην sram και να αποφευχθούν κατά το δυνατόν οι πράξεις κινητής υποδιαστολής για την αποθήκευση των μετρήσεων χρησιμοποιούνται πίνακες τύπου signed int (16bit) και ως εκ τούτου η προσαρμογή από τον μικροελεγκτή γίνεται με τα δεδομένα σε δυαδική μορφή (εύρος τιμών -2048...2047) και όχι σε κανονική αναπαράσταση τάσης (-1...+1V). Φυσικά οι παράμετροι E.T. που χρησιμοποιούνται υπολογίστηκαν μετατρέποντας τα δεδομένα σε ισοδύναμες δυαδικές αναπαραστάσεις κατά την επεξεργασία.

Η προσαρμογή των μετρήσεων από τον μικροελεγκτή με βάση τις δηλωθέντες παραμέτρους (Βλ. §3,1) πραγματοποιείται από την παρακάτω απλή συνάρτηση:

```
// Fit ADC data using linear least squares method
signed int LS_fit(signed int x)
```

```
{
if ((x>512) || (x<-512)) return (float)x*LS_SLOPE_B+LS_YP_B; ( |Vin| > 250mV)

else return (float)x*LS_SLOPE_S+LS_YP_S; (-250mV ≤ Vin ≤ 250mV)
}
```

3.6 Προβολή αποτελεσμάτων

Τα αποτελέσματα των μετρήσεων στέλνονται μετά την ολοκλήρωση κάθε επανάληψης σε Η/Υ μέσω της σειριακής θύρας F.0 με την μορφή:

[αριθμός καναλιού (1-8)];[αύξων αριθμός μέτρησης];[μέτρηση (σε mV)]

Κατόπιν προστίθεται η παρούσα μέτρηση στον πίνακα `t_sum[]` που περιέχει τα συσσωρευτικά αθροίσματα για κάθε κανάλι και μηδενίζεται ο πίνακας `averaging_sum[]`. Παρακάτω παρατίθεται το αντίστοιχο τμήμα κώδικα:

```
// print to uart & zero averaging_sum array
for (j=0;j<=7;j++) //Ch scan loop
{
ftoa(((float) averaging_sum[j]*1000/(MEAN_LENGTH*2048.)),1,str);
printf("ch%d;%d;%s\n",j+1,t_count,str);
t_sum[j] += averaging_sum[j];
averaging_sum[j]=0;
} //Ch scan loop
```

Κάθε 10 επαναλήψεις ανανεώνεται η οθόνη LCD με τον συσσωρευτικό μέσο όρο για κάθε κανάλι με την μορφή:

[αριθμός καναλιού (1-8)];[μέτρηση (σε mV)]

Τα κανάλια 1-4 (ADCA) εμφανίζονται στο αριστερό ήμισυ της οθόνης στοιχισμένα κατά αύξοντα αριθμό καναλιού ενώ τα κανάλια 5-8 (ADCB) εμφανίζονται αντίστοιχα στην δεξιά μεριά.

```
// update LCD with accumulative average
if (t_count%10==0) //update LCD
{
lcd_clear();
for (j=0;j<=3;j++) //ch scan loop
{
```

```
itoa(((float)t_sum[j]*1000/(t_count*MEAN_LENGTH*2048.)),str); //0-3
sprintf(temp,"%d:%smV", (j+1),str);
lcd_gotoxy(0,j);
lcd_puts(temp);
itoa(((float) t_sum[j+4]*1000/(t_count*MEAN_LENGTH*2048.)),str); //4-7
sprintf(temp,"%d:%smV", (j+5),str);
lcd_gotoxy(10,j);
lcd_puts(temp);
} //ch scan loop
} //update LCD
```

3.7 Εξοικονόμηση ενέργειας

Προκειμένου το σύστημα να διαθέτει ικανοποιητική αυτονομία όταν τροφοδοτείται από μπαταρία είναι απαραίτητη η εξοικονόμηση ενέργειας. Αυτός είναι και ένας από τους λόγους που χρησιμοποιείται ο εσωτερικός ταλαντωτής στα 2MHz αντί του ταλαντωτή 32MHz για τον χρονισμό του συστήματος.

Επιπροσθέτως όποτε δεν εκτελείται κάποια λειτουργία ο μικροελεγκτής μεταβαίνει σε κατάσταση εξοικονόμησης ενέργειας. Αυτό πραγματοποιείται με τις ακόλουθες εντολές που συμπεριλαμβάνονται στην Sleep.h του Codevision τοποθετημένες στο τέλος του κύριου βρόχου επανάληψης:

```
sleep_enable(); // allow entering sleep modes
idle(); //enter sleep mode
```

Ο μικροελεγκτής επιστρέφει σε ενεργή κατάσταση από την διακοπή υπερχειλίσης του timer E0 και επανέρχεται στον κύριο βρόχο όπου ελέγχεται εάν ο counter έχει φτάσει την επιθυμητή τιμή που ορίζεται από την σταθερά DELAY_CNT και εάν ο διακόπτης ελέγχου στην πρόσοψη της συσκευής είναι κλειστός.

Εάν ισχύουν οι συνθήκες απενεργοποιείται η δυνατότητα εισόδου σε κατάσταση εξοικονόμησης ενέργειας με την εντολή:

```
sleep_disable(); // prevent entering sleep modes
```

Ακολούθως εκτελείται η διαδικασία λήψης-επεξεργασίας-προβολής μετρήσεων.

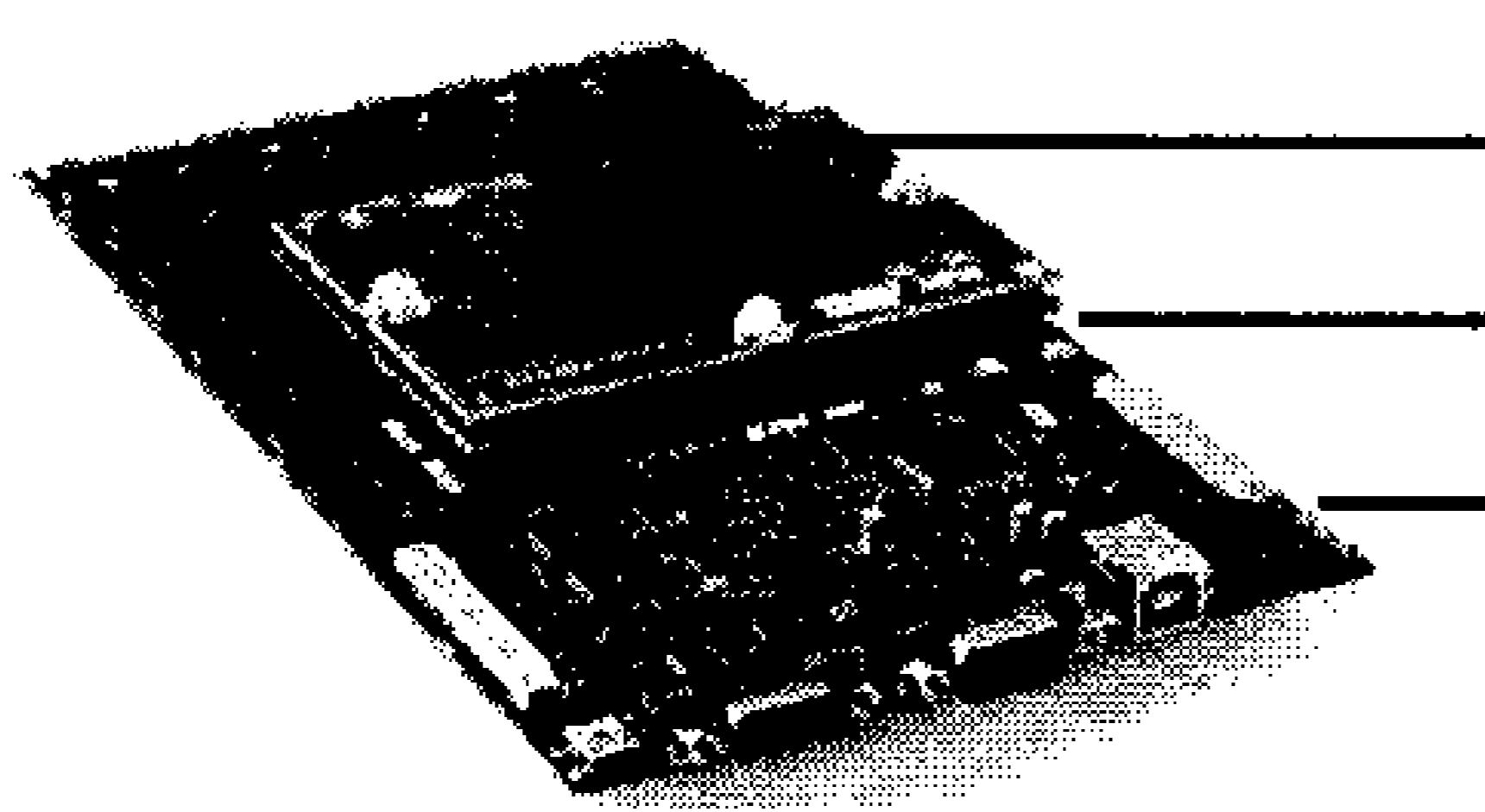
Εάν δεν ισχύουν ταυτόχρονα και οι δύο συνθήκες τότε ο μικροελεγκτής εισέρχεται και πάλι σε κατάσταση εξοικονόμησης ενέργειας μέχρι την επομένη υπερχειλίση του timer E0.

4 Εργαστηριακές συσκευές που χρησιμοποιήθηκαν

Στο παρόν κεφάλαιο παρουσιάζονται συνοπτικά οι κυριότερες συσκευές του εργαστηρίου που χρησιμοποιήθηκαν κατά την ανάπτυξη - αξιολόγηση της διάταξης.

4.1 Το αναπτυξιακό Atmel STK600

Το STK600 είναι ένα πλήρες αναπτυξιακό σύστημα με υποστήριξη για το σύνολο των σύγχρονων 8-bit και 32-bit μικροελεγκτών AVR το οποίο είναι εξαιρετικά χρήσιμο για τις πρώτες φάσεις δοκιμών ενός πρωτοτύπου βασισμένο σε μικροελεγκτές AVR αφού μεταξύ άλλων παρέχει απλοποιημένη σύνδεση με Η/Υ μέσω USB για έλεγχο και προγραμματισμό, απευθείας πρόσβαση σε όλα τα I/O pins του μικροελεγκτή, τροφοδοσία μέσω USB ή από εξωτερικό τροφοδοτικό κατάλληλης τάσης, ενσωματωμένο προγραμματιστή ISP/PDI/TPI/JTAG/aWire/HVSerial/HVParallel, ρυθμιζόμενες τάσεις τροφοδοσίας και αναφοράς, ρυθμιζόμενο ταλαντωτή και on-board LEDs, μπουτόν, μνήμη flash και κυκλώματα διασύνδεσης με RS-232,USB,CAN κ.α.



Εικόνα 4-1: Το σύστημα διασύνδεσης του STK600

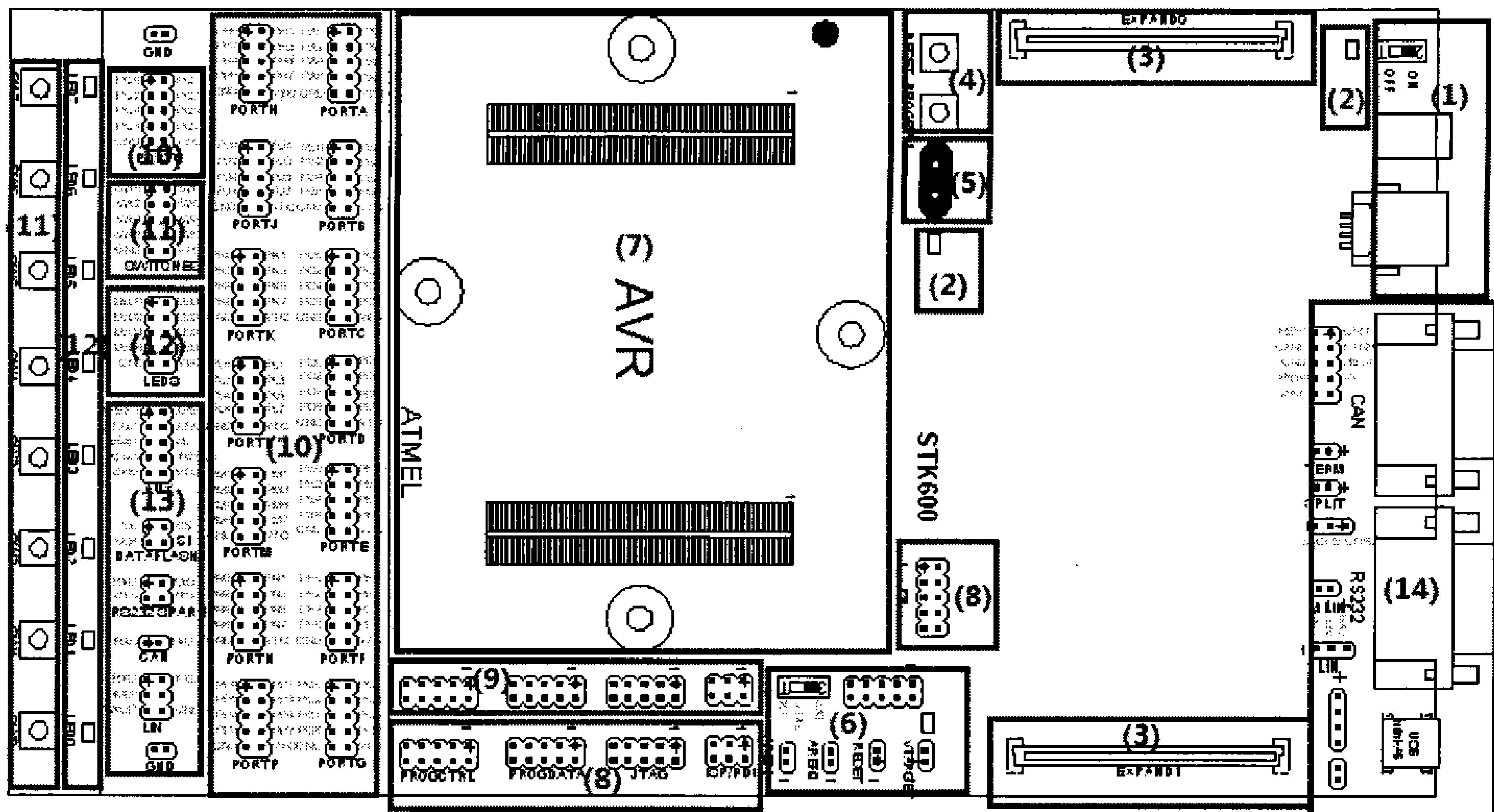
Socket board
Routing card
STK600

Η κάθε κάρτα διασύνδεσης αναγνωρίζεται αυτόματα από την μητρική πλακέτα του STK600 με βάση ένα απλό σύστημα ψηφιακής λογικής επιτρέποντας έτσι την ενημέρωση του χρήστη σε περίπτωση λάθους συνδυασμού και την επιβολή περιορισμών τάσης τροφοδοσίας αναλόγως του μικροελεγκτή που χρησιμοποιείται.

Για την διασύνδεση του μικροελεγκτή προορισμού (target microcontroller) χρησιμοποιείται ένα αρθρωτό σύστημα με κάρτες υποδοχής και κάρτες δρομολόγησης προκειμένου να παρέχεται συμβατότητα με ένα πλήθος συσκευών που διατίθενται σε διαφορετικές συσκευασίες και με διαφορετική διάταξη ακροδεκτών διατηρώντας παράλληλα μικρό μέγεθος και χαμηλό αρχικό κόστος.

Στην εικόνα 4-II φαίνεται η διάταξη του STK600 όπου:

- 1) Βύσματα USB & τροφοδοσίας, γενικός διακόπτης τροφοδοσίας
- 2) Ενδεικτικά LEDs
- 3) Βύσματα διασύνδεσης με εξωτερικές διατάξεις
- 4) Μπουτόν RESET και αναβάθμισης firmware
- 5) Θέση κρυστάλλου
- 6) Επιλογείς σύνδεσης τροφοδοσίας, γραμμής RESET, τάσεων αναφοράς και πηγής χρονισμού για τον μικροελεγκτή
- 7) Θέση μικροελεγκτή προς προγραμματισμό (μέσω καρτών διασύνδεσης)
- 8) Βύσματα προγραμματισμού (από τον ενσωματωμένο προγραμματιστή)
- 9) Βύσματα προγραμματισμού (προς τον μικροελεγκτή)
- 10) Βύσματα πρόσβασης στα I/O pins του μικροελεγκτή
- 11) Μπουτόν και αντίστοιχο βύσμα σύνδεσης τους
- 12) LEDs και αντίστοιχο βύσμα σύνδεσης τους
- 13) Βύσματα διασύνδεσης με τάσεις αναφοράς, ταλαντωτές, μνήμη flash και κυκλώματα RS232/USB/CAN/LIN
- 14) Θύρες διασύνδεσης με RS232/USB/CAN/LIN



Εικόνα 4-II: Κάτοψη του αναπτυξιακού STK600 με τα επιμέρους τμήματα σημειωμένα

Το STK600 χρησιμοποιήθηκε κυρίως για τις πρώτες δοκιμές με την πλατφόρμα xmega καθώς και κατά την ανάπτυξη του προγράμματος του μικροελεγκτή αφού προσφέρει ευκολίες οι οποίες δεν μπορούν να ενσωματωθούν σε μια συσκευή ειδικής χρήσης όπως άμεση πρόσβαση σε όλα τα I/O pins, on-board προγραμματιστή και δυνατότητα άμεσης αλλαγής μικροελεγκτή χωρίς εργαλεία.

4.2 Πηγή Τάσης/Ρεύματος Keithley Sourcemeeter 2612



Εικόνα 4-III: Keithley Sourcemeeter 2612

Το Keithley Sourcemeeter 2612 ενσωματώνει μια πηγή τάσης/ρεύματος αναφοράς 2 καναλιών που μπορεί να λειτουργήσει και ως ηλεκτρονικό φορτίο, πολύμετρο 5½ ψηφίων καθώς και όλα τα απαραίτητα κυκλώματα ελέγχου - χρονισμού. Προαιρετικά μπορεί να διασυνδεθεί με Η/Υ μέσω GPIB (IEEE488.1), RS232 και με άλλες συσκευές ίδιου τύπου μέσω TSP-Link.

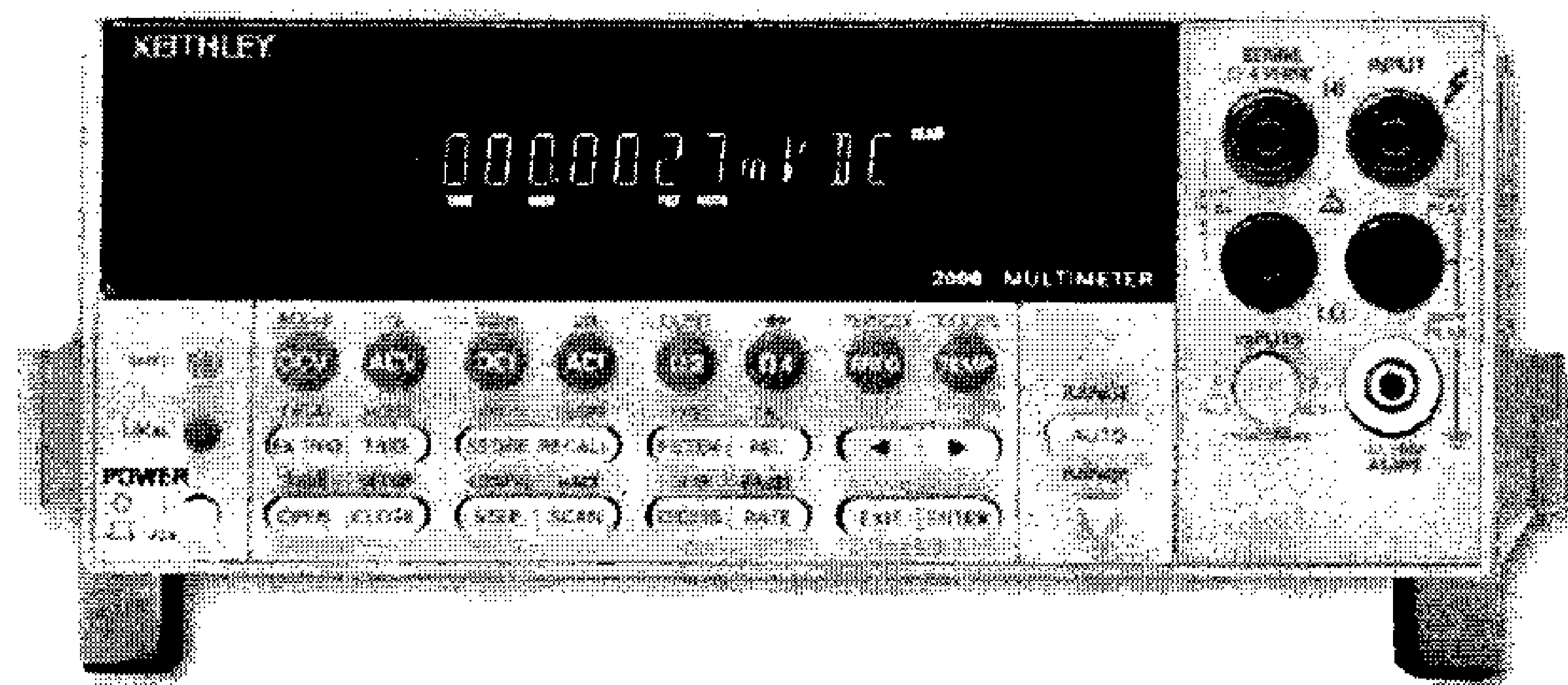
Το Sourcemeeter 2612 χρησίμευσε για τον έλεγχο ορθής λειτουργίας καθώς και για την μελέτη και διόρθωση της συμπεριφοράς του ενσωματωμένου A/D μετατροπέα του μικροελεγκτή [Ενότητα 3.5] αφού μπορεί να παρέχει μεταβαλλόμενη τάση εξόδου με υψηλή ακρίβεια και χαμηλό θόρυβο.

Source/Measure Accuracy	2601, 2602	2611, 2612
• Current measure range (per channel)	1pA to 3A	DC, 1pA to 1.5A; pulsed, 1pA to 10A
• Current source range (per channel)	5pA to 3A	DC, 5pA to 1.5A; pulsed, 5pA to 10A
• Basic current programming accuracy	0.03%	0.03%
• Basic current measurement accuracy	0.02%	0.02%
• Voltage measure range (per channel)	1μV to 40V	1μV to 200V
• Voltage source range (per channel)	5μV to 40V	50μV to 200V
• Basic voltage programming accuracy	0.02%	0.02%
• Basic voltage measurement accuracy	0.015%	0.015%
Speed		
• Readings to memory		10,000/s
• Sweep source change and reading to memory		5,500/s
• Sweep source change and reading to GPIB		3,600/s
• Spot readings to GPIB		1,100/s
• Spot source change and reading to GPIB		880/s
• Pulse capability		200μs minimum source/measure pulse

Πίνακας 4-A: Συνοπτικές προδιαγραφές Keithley 2612

4.3 Πολύμετρο πάγκου Keithley 2000 series

Το Keithley 2000 είναι ένα ψηφιακό Πολύμετρο υψηλής ακριβείας 6½ ψηφίων το οποίο χρησιμοποιείται ευρέως από επιστημονικούς φορείς λόγω των δυνατοτήτων και της αξιοπιστίας του.

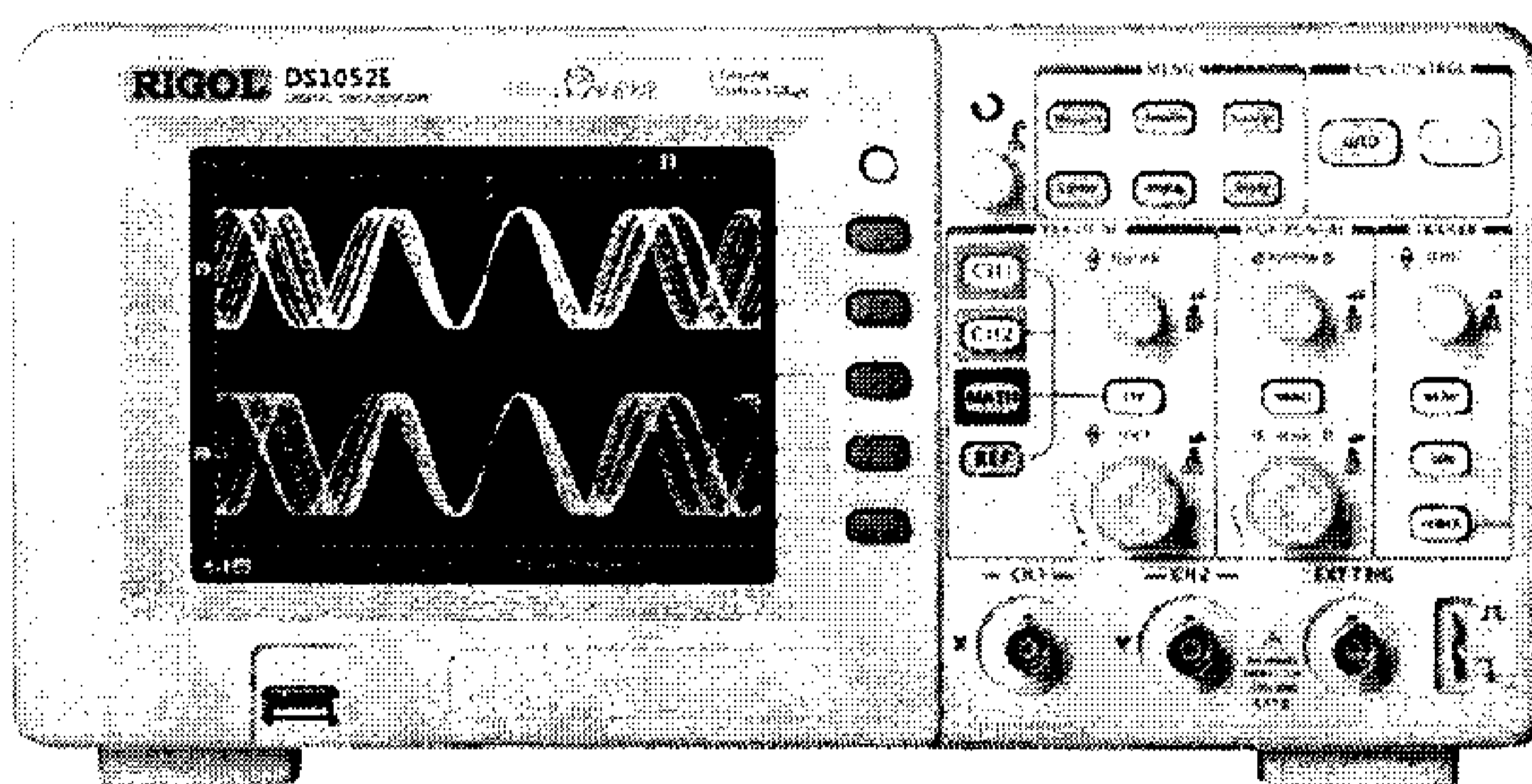


Εικόνα 4-IV: Keithley 2000 series

Ως εκ τούτου είναι ιδανικό για την ανάπτυξη συσκευών μετρήσεων όπου απαιτούνται μετρήσεις αναφοράς για σύγκριση με την συσκευή υπό έλεγχο καθώς και για τις διάφορες φάσεις ηλεκτρονικών κατασκευών προκειμένου να διαπιστωθεί η άριστη αγωγιμότητα των συνδέσεων, η καλή συμπεριφορά του κυκλώματος τροφοδοσίας και γενικότερα να γίνουν μετρήσεις ακριβείας όπου απαιτείται.

Παρέχονται 13 λειτουργίες μέτρησης με βασική ακρίβεια μέτρησης σε DC τάση 100ppm, ταχύτητα λήψης μετρήσεων έως 2000 μετρήσεις/sec, πλήθος επιλογών παραμετροποίησης και δυνατότητα διασύνδεσης με Η/Υ μέσω GPIB (IEEE488.1) ή RS232.

4.4 Παλμογράφος Rigol DS1052E



Εικόνα 4-V: Rigol DS1052E

Μια ακόμα συσκευή που ήταν εξαιρετικά χρήσιμη κατά την ανάπτυξη της διάταξης είναι ο ψηφιακός παλμογράφος DS1052E, κατασκευασμένος από την εταιρία Rigol technologies. Παρά το χαμηλό κόστος αγοράς παρέχονται όλες οι συνηθισμένες λειτουργίες ενός σύγχρονου ψηφιακού παλμογράφου

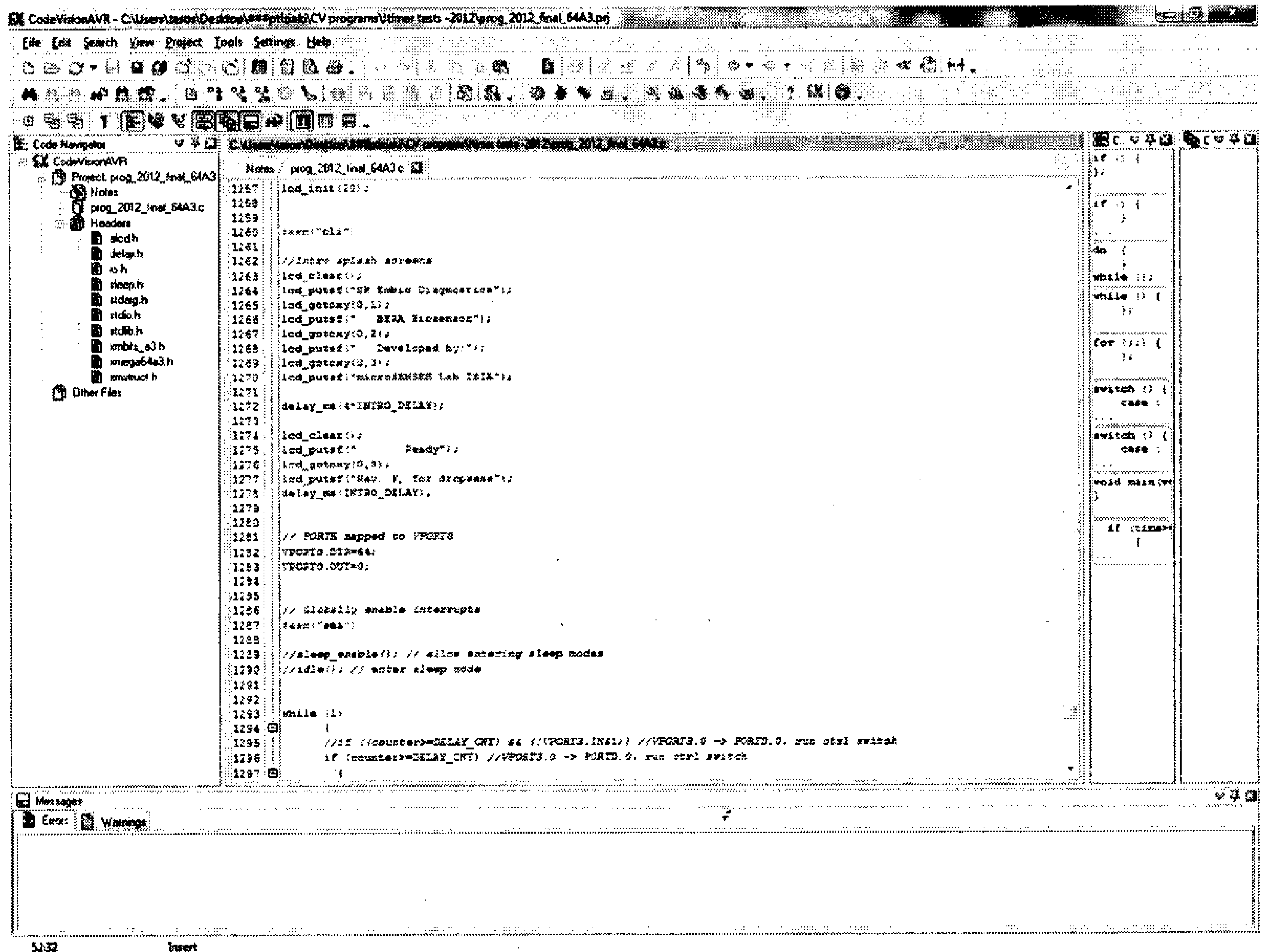
όπως δυνατότητες εκτέλεσης μετρήσεων, πράξεων, ελέγχων και αποθήκευσης κυματομορφών από τα δυο διαθέσιμα αναλογικά κανάλια, πλήθος επιλογών σκανδαλισμού, απομακρυσμένος έλεγχος και λήψη μετρήσεων από Η/Υ μέσω USB/RS232. Χρησιμοποιήθηκε κατά κύριο λόγο κατά την ανάπτυξη του κυκλώματος τροφοδοσίας, για την μέτρηση κυμάτωσης τάσης σε διάφορα σημεία ενδιαφέροντος και τον έλεγχο της επίδρασης εξωτερικού θορύβου στο κύκλωμα.

5 Το λογισμικό που χρησιμοποιήθηκε

Παρουσιάζονται συνοπτικά τα πακέτα λογισμικού που χρησιμοποιήθηκαν για τον προγραμματισμό του μικροελεγκτή, την σχεδίαση & κατασκευή των ηλεκτρονικών κυκλωμάτων και την λήψη μετρήσεων.

5.1 HP InfoTech CodevisionAVR

Το CodevisionAVR είναι ένα περιβάλλον προγραμματισμού βασισμένο στην ANSI C με τροποποιήσεις και εμπλουτισμό των ενσωματωμένων λειτουργιών ώστε να συνεργάζεται με τους μικροελεγκτές AVR της εταιρίας Atmel και να μεταγλωττίζει τα προγράμματα που γράφονται από C σε αντίστοιχη συμβολική γλώσσα (AVR Assembly). Αφού γίνει μεταγλώττιση υπάρχει η επιλογή για παραγωγή κώδικα μηχανής και την μεταφορά του στον μικροελεγκτή ή την έναρξη εκσφαλμάτωσης μέσω του AVR Studio.

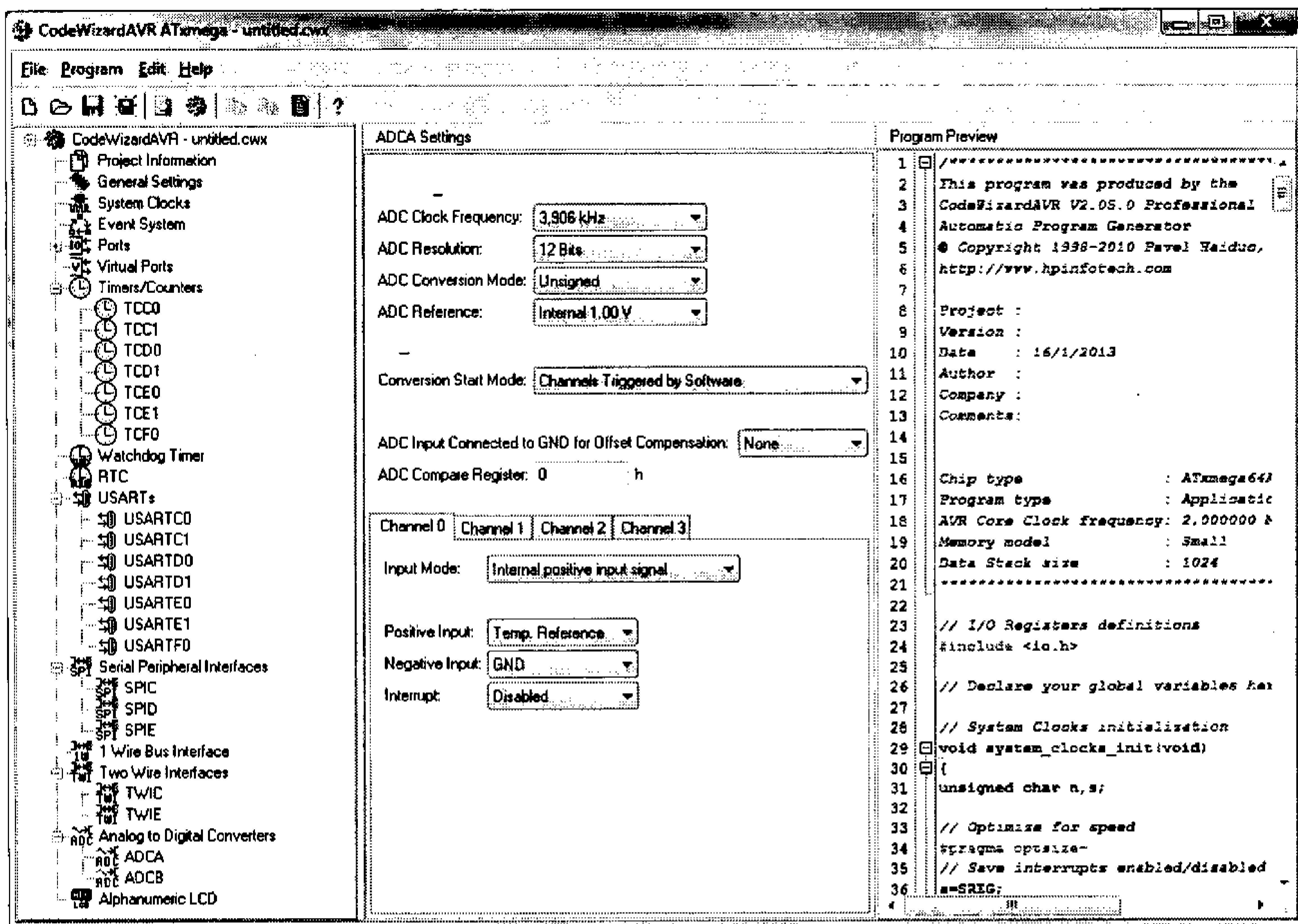


Εικόνα 5-1: Το περιβάλλον εργασίας του CodevisionAVR

Παρέχεται πλήθος έτοιμων συναρτήσεων για εύκολη χρήση των ενσωματωμένων περιφερειακών του μικροελεγκτή καθώς και τον χειρισμό κοινών εξωτερικών συσκευών όπως πχ οθονών LCD, αισθητήρων θερμοκρασίας, RTC κλπ.

Το λογισμικό που χρησιμοποιήθηκε

Φυσικά υπάρχουν όλες οι τυπικές λειτουργίες ενός σύγχρονου προγραμματιστικού περιβάλλοντος όπως αυτόματη χρωματική μορφοποίηση του κώδικα για την εύκολη διάκριση σχολίων κτλ, εντοπισμό σφαλμάτων κατά την μεταγλώττιση, προβολή πληροφοριών για τον κώδικα και δυνατότητα για εισαγωγή εντολών σε assembly ενώ υπάρχει και ενσωματωμένο τερματικό που χρησιμεύει για την ανάπτυξη πρωτοτύπων που χρησιμοποιούν σειριακές επικοινωνίες. Ένα σημαντικό πλεονέκτημα του CodevisionAVR έναντι ανταγωνιστικών πακέτων είναι η ενσωμάτωση του "CodewizardAVR", ενός αυτοματοποιημένου εργαλείου για την εύκολη και γρήγορη παραγωγή κώδικα για τα βασικά περιφερειακά του μικροελεγκτή και μερικές από τις συνηθέστερες εξωτερικές συσκευές.



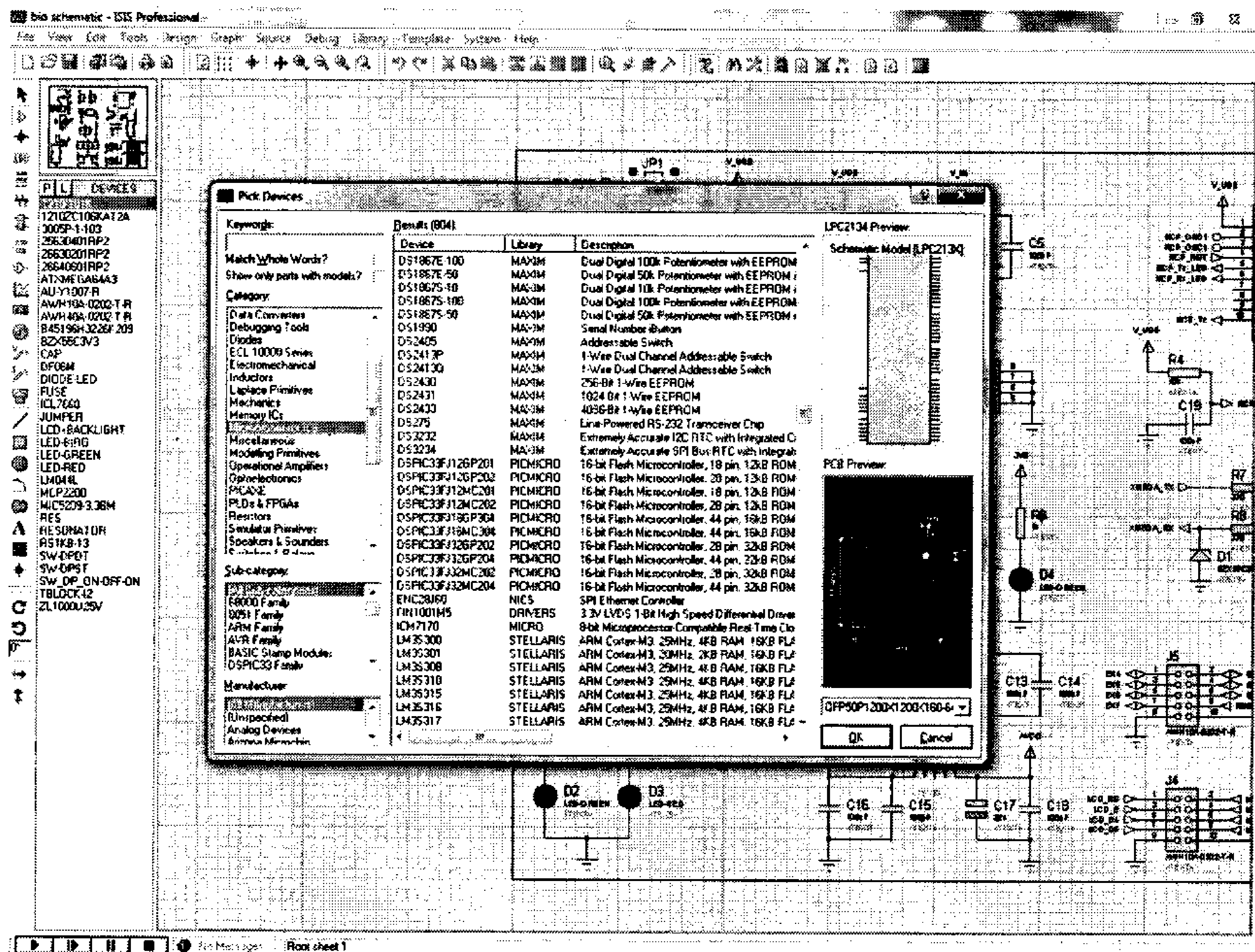
Εικόνα 5-II: Άποψη του CodewizardAVR για μικροελεγκτές ATxmega

Οι δυνατότητες του σε συνδυασμό με το χαμηλό κόστος αγοράς και το ικανοποιητικό αποτέλεσμα καθιστούν το CodevisionAVR ένα εκ των δημοφιλέστερων C compilers για AVR.

5.2 Labcenter Proteus suite

Το Proteus 7 είναι ένα δημοφιλές πακέτο λογισμικού για την σχεδίαση και εξομοίωση ηλεκτρονικών κυκλωμάτων. Είναι δυνατή η εξομοίωση αναλογικών, ψηφιακών ή μικτών κυκλωμάτων καθώς και μικροελεγκτών με τον συνδυασμό μιας προσαρμοσμένης έκδοσης ProSPICE και κατάλληλων μοντέλων. Παρέχεται

πληθώρα έτοιμων βιβλιοθηκών με σύγχρονα εξαρτήματα, εκ των οποίων τα περισσότερα συνοδεύονται από αντίστοιχο μοντέλο για εξομοίωση.

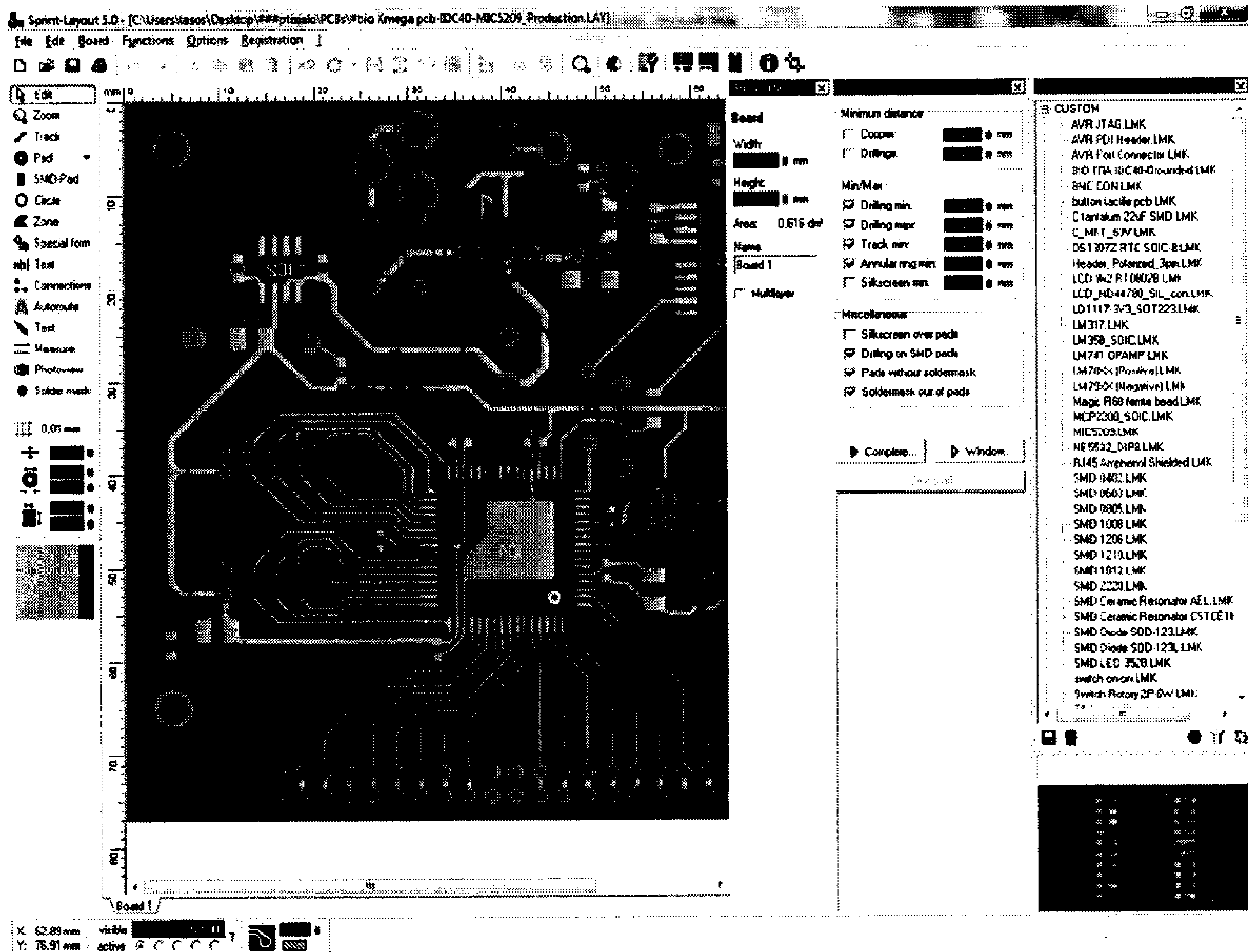


Εικόνα 5-III: Το περιβάλλον εργασίας του Proteus ISIS

Εκτός από τις κλασικές λειτουργίες ενός πακέτου σχεδίασης ηλεκτρονικών κυκλωμάτων ενσωματώνονται και κάποιες πρωτότυπες λειτουργίες όπως η δυνατότητα για ανάλυση θορύβου και παραμόρφωσης, προβολή ανάλυσης Fourier κλπ, η προβολή τρισδιάστατων μοντέλων για τα PCB που σχεδιάζονται και ο προγραμματισμός scripts σε γλώσσα EasyHDL.

5.3 Abacom Sprint Layout

Το Sprint Layout είναι ένα εξειδικευμένο πρόγραμμα σχεδιασμού τυπωμένων κυκλωμάτων. Δεν υπάρχει άμεση σύνδεση με κάποια εφαρμογή παραγωγής σχηματικών όπως συνηθίζεται, αντιθέτως ο χρήστης επιλέγει απευθείας τα εξαρτήματα υπό την μορφή footprints και τα τοποθετεί όπου επιθυμεί, συνδέοντας τα στην συνέχεια και δημιουργώντας το PCB layout. Παρότι δεν παρέχονται προχωρημένες λειτουργίες αφού ακόμα ο autorouter του προγράμματος είναι απολύτως βασικός το πολύ χαμηλό κόστος αγοράς και η απλότητα του προγράμματος το καθιστούν δημοφιλές για ερασιτεχνική και επαγγελματική χρήση στην περίπτωση απλών σχεδίων τα οποία υλοποιούνται εύκολα σε πλακέτες ενός έως τεσσάρων επιπέδων.



Εικόνα 5-IV: Περιβάλλον εργασίας του Sprint Layout

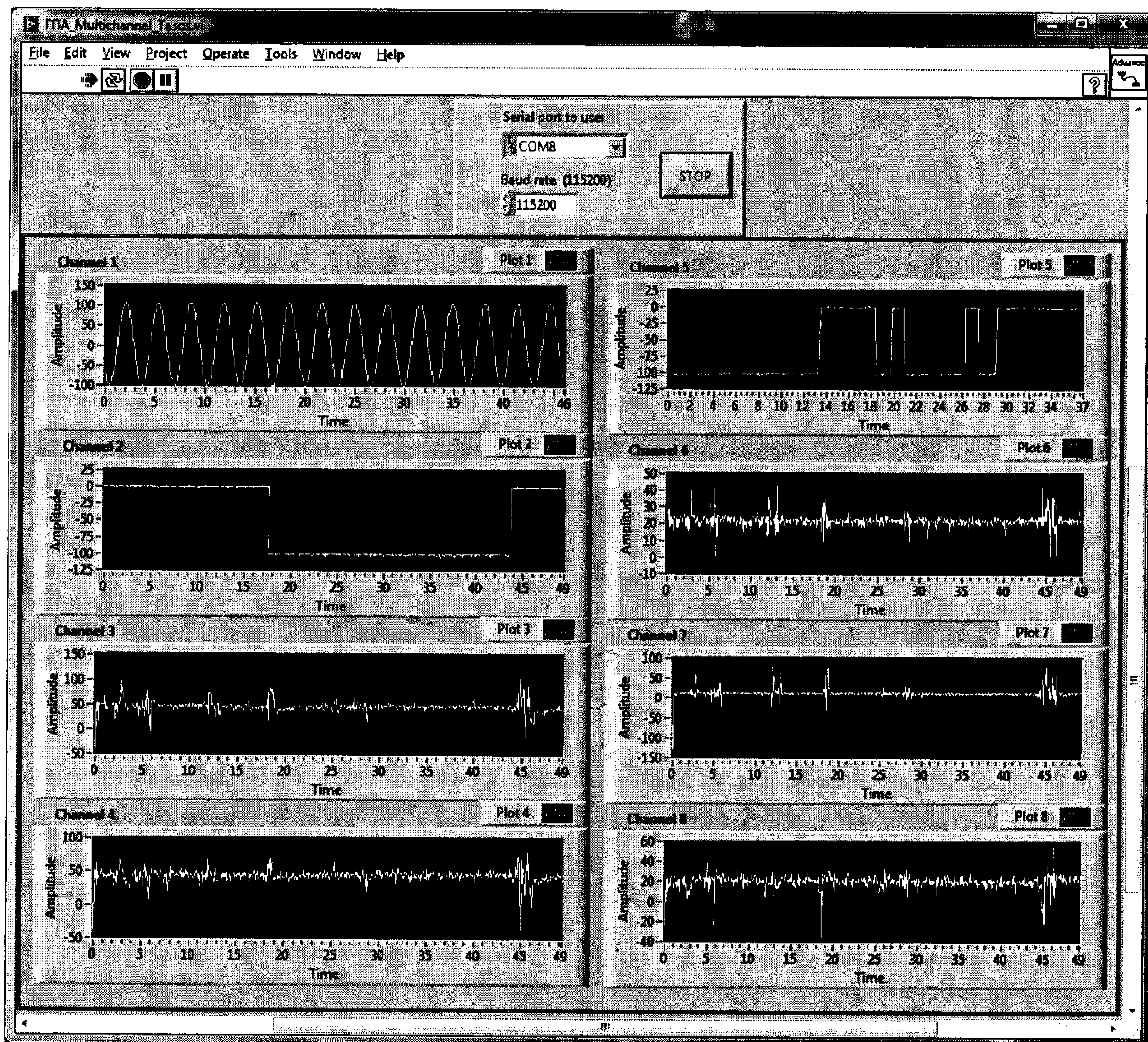
Δεδομένου ότι για τέτοιου επιπέδου σχέδια συχνά οι έμπειροι μηχανικοί σχεδιάζουν εξ αρχής το τυπωμένο κύκλωμα αγνοώντας αυτοματοποιημένα εργαλεία τοποθέτησης εξαρτημάτων και διαδρόμων προκειμένου το αποτέλεσμα να είναι βελτιστοποιημένο από άποψη απόδοσης / λειτουργικότητας και προσαρμοσμένο στις ανάγκες που υπάρχουν το sprint layout μπορεί να αντικαταστήσει πιο εξελιγμένα σχεδιαστικά πακέτα με εξίσου ικανοποιητικό αποτέλεσμα και πιθανώς συντομότερο χρόνο σχεδίασης μιας που είναι εξ αρχής προσανατολισμένο σε μη αυτόματο σχεδιασμό τυπωμένων.

5.4 National Instruments LabVIEW

Το LabVIEW είναι ένα αναπτυξιακό περιβάλλον που χρησιμοποιείται ευρέως από ερευνητικά ιδρύματα και εταιρίες για την λήψη και επεξεργασία μετρήσεων, τον έλεγχο οργάνων μέτρησης και την διεξαγωγή δοκιμών.

Το λογισμικό που χρησιμοποιήθηκε

Βασίζεται σε μια γραφική γλώσσα προγραμματισμού η οποία έχει σχεδιαστεί εξ αρχής ώστε να διευκολύνει την διεπαφή με συσκευές και την πραγματοποίηση μετρήσεων. Παρέχεται πλήθος έτοιμων λειτουργιών, διαδικασιών, drivers για συσκευές και πρωτόκολλα διεπαφής, ευπροσάρμοστα στοιχεία μέσω οδηγών και έτοιμα παραδείγματα εφαρμογών.



Εικόνα 5-V: Το περιβάλλον της εφαρμογής λήψης μετρήσεων μέσω LabVIEW

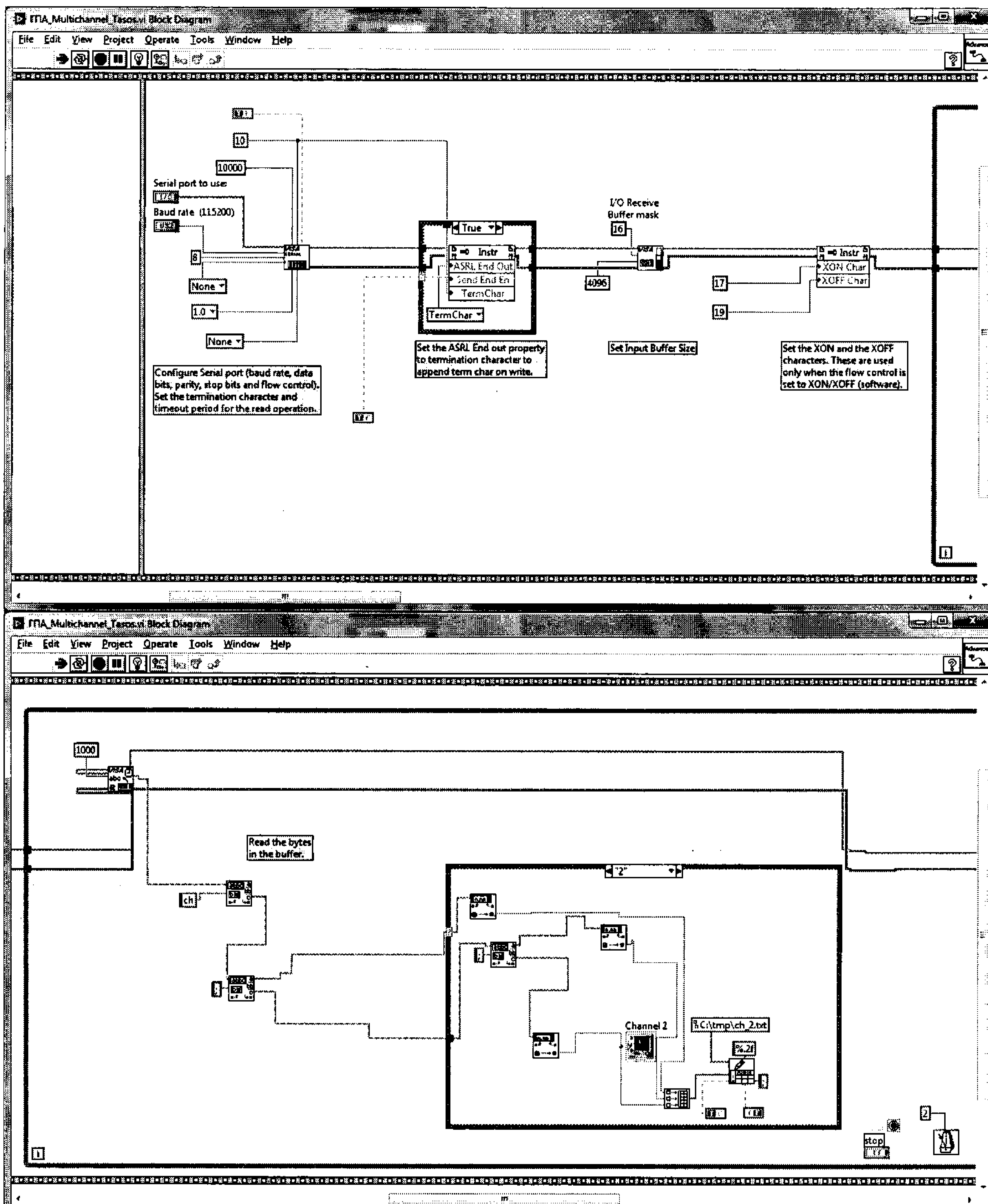
Για την διευκόλυνση της λήψης και επεξεργασίας των μετρήσεων που λαμβάνονται από το σύστημα χρησιμοποιήθηκε η εφαρμογή της εικόνας 5-V η οποία βασίζεται σε παλαιότερη πτυχιακή εργασία [Ref. 17].

Η εφαρμογή λαμβάνει τα δεδομένα που αποστέλλονται μέσω της εικονικής σειριακής θύρας που δημιουργείται από τον usb>serial controller και διαχωρίζει τις μετρήσεις ανά κανάλι.

Το λογισμικό που χρησιμοποιήθηκε

Ακολουθως προβάλλονται τα αποτελέσματα σε γραφήματα ενώ ταυτοχρόνως αποθηκεύονται οι μετρήσεις σε κατάλληλη μορφή αρχείου για περαιτέρω επεξεργασία.

Στην εικόνα 5-VI διακρίνονται τα βασικά δομικά μέρη της εφαρμογής.



Εικόνα 5-VI: Μέρη του μπλοκ διαγράμματος της εφαρμογής λήψης & επεξεργασίας των μετρήσεων σε LabVIEW. Στο άνω τμήμα η ρύθμιση της σειριακής επικοινωνίας, στο κάτω τμήμα η λήψη, ο διαχωρισμός και η προβολή των δεδομένων.

6 Η τελική μορφή της διάταξης

6.1 Hardware – χαρακτηριστικά

Στην εικόνα 6-1 παρουσιάζεται η τελική μορφή της κύριας συσκευής. Αφού συναρμολογήθηκε το κουτί στερεώθηκε η οθόνη LCD και η κύρια πλακέτα στις θέσεις τους χρησιμοποιώντας πλαστικές βίδες M3. Κατόπιν τοποθετήθηκαν οι διακόπτες ελέγχου και τα ενδεικτικά LED στην πρόσοψη, με τα τελευταία να στερεώνονται χρησιμοποιώντας διάφανη εποξική κόλλα 2 συστατικών. Αφού έγιναν όλες οι απαραίτητες συνδέσεις στο εσωτερικό της συσκευής σφραγίστηκε το κουτί και βιδώθηκε το άνω μέρος στους συνδέσμους που έχουν τοποθετηθεί στο εσωτερικό των ακμών του κουτιού χρησιμοποιώντας μεταλλικές βίδες.



Εικόνα 6-1: Η τελική μορφή του hardware (Άνω αριστερά: Άνω όψη, Άνω δεξιά: Κάτω όψη, Κάτω αριστερά: Πρόσοψη, Κάτω δεξιά: Πίσω όψη)

Στον πίνακα 6-A συνοψίζονται τα κύρια χαρακτηριστικά της συσκευής, όπως αυτά μετρήθηκαν στην παρούσα μορφή της και με βάση τις προδιαγραφές - περιορισμούς των εξαρτημάτων που χρησιμοποιήθηκαν.

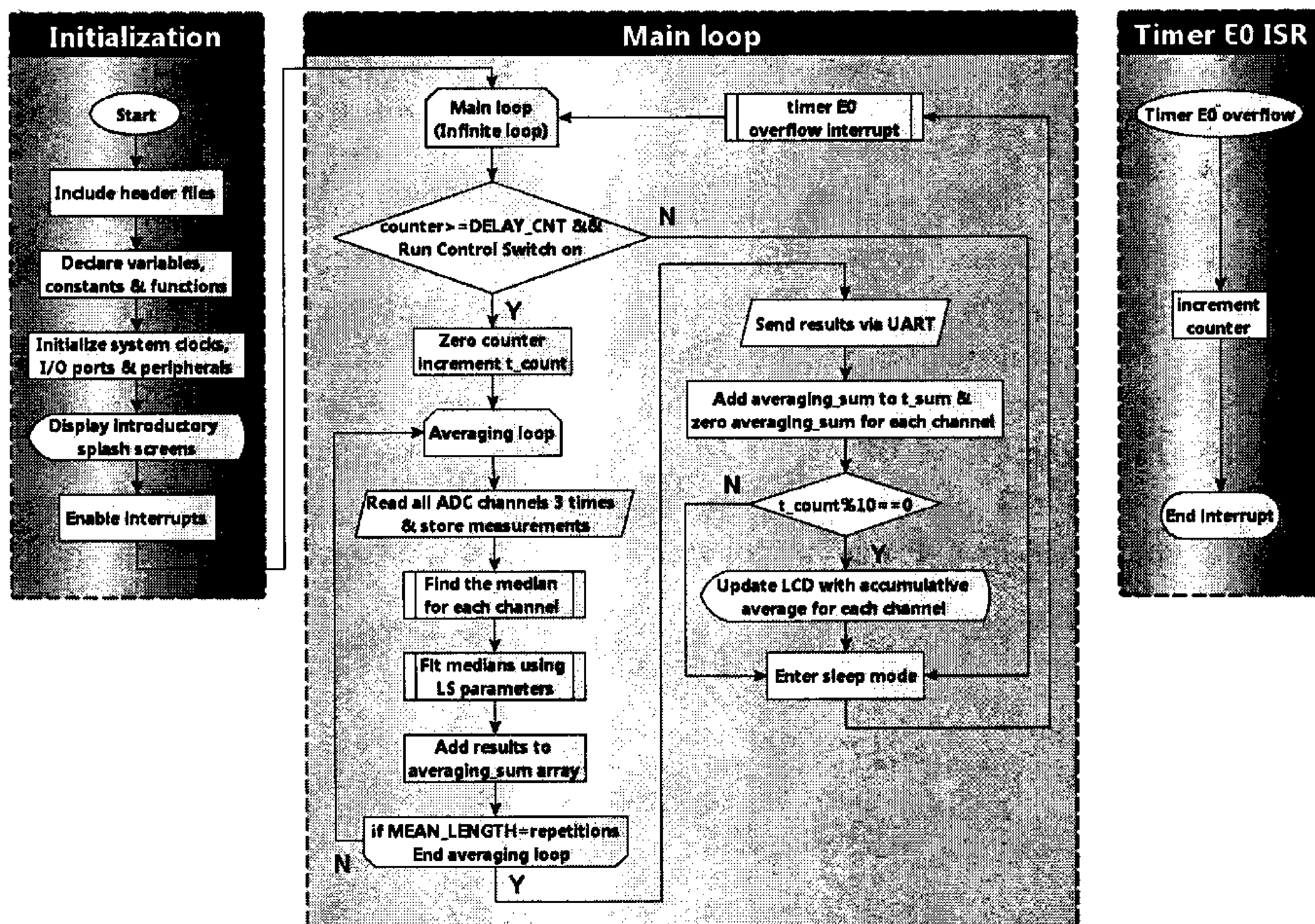
Η τελική μορφή της διάταξης

Τάση τροφοδοσίας	4,5...12V DC ή 5...9V AC
Μέση κατανάλωση ρεύματος	45mA
Αυτονομία με μπαταρία (9V PP3)	10 ώρες (κατά προσέγγιση)
Οθόνη	20*4 χαρακτήρων, φωτιζόμενη
Χρόνος αρχικοποίησης	3 sec (κατά προσέγγιση)
Κανάλια μέτρησης	8 διαφορεικά κανάλια
Ρυθμός μέτρησης	4 Δείγματα/sec (ανά κανάλι, ±5%)
Δυναμικό εύρος μέτρησης	-850mV ~ 850mV
Ανάλυση	1mV
Μέγιστο απόλυτο σφάλμα	±5mV
Διαστάσεις (κυρίως συσκευή)	125*103*77mm (Π*Μ*Υ)
Βάρος (κυρίως συσκευή)	405gr (κατά προσέγγιση)
Θερμοκρασία λειτουργίας	5 ~ 50 °C

Πίνακας 6-A: Χαρακτηριστικά συσκευής

6.2 Πρόγραμμα μικροελεγκτή

Το πρόγραμμα του μικροελεγκτή αποτελείται από τα επιμέρους τμήματα που έχουν αναλυθεί στο κεφάλαιο 3, η διασύνδεση των οποίων παρουσιάζεται στο διάγραμμα ροής της εικόνας 6-II.



Εικόνα 6-II: Διάγραμμα ροής προγράμματος μικροελεγκτή

Η τελική μορφή της διάταξης

Αρχικά δηλώνονται σταθερές, μεταβλητές και συναρτήσεις καθώς επίσης αρχικοποιούνται τα περιφερειακά του μικροελεγκτή ενώ παρουσιάζονται κάποιες εισαγωγικές πληροφορίες στην οθόνη LCD.

Αφού ολοκληρωθεί το στάδιο αρχικοποίησης ενεργοποιείται ο χρονιστής E0 ο οποίος υπερχειλίζει κάθε 1msec, παράγοντας μια διακοπή. Η συνάρτηση εξυπηρέτησης της διακοπής αυξάνει την γενική μεταβλητή counter κατά μία μονάδα.

Στον κύριο βρόχο του προγράμματος (ατέρμονας βρόχος) ελέγχεται εάν η μεταβλητή counter έχει φτάσει την ορισμένη μέσω της σταθεράς DELAY_CNT καθυστέρηση καθώς και εάν είναι κλειστός ο διακόπτης έλεγχου εκτέλεσης του προγράμματος που βρίσκεται στην πρόσοψη της συσκευής.

Εάν έστω μια από τις συνθήκες δεν ισχύει τότε ο μικροελεγκτής εισέρχεται σε κατάσταση εξοικονόμησης ενέργειας μέχρι την επόμενη διακοπή του timer E0.

Εάν οι συνθήκες ισχύουν ταυτόχρονα τότε μηδενίζεται η μεταβλητή counter και αυξάνεται κατά μια μονάδα η μεταβλητή t_count η οποία αντιστοιχεί στον συνολικό αριθμό μετρήσεων που έχουν ληφθεί ανά κανάλι στην τρέχουσα περίοδο χρήσης της συσκευής.

Κατόπιν λαμβάνεται ένα σύνολο 3 μετρήσεων ανά κανάλι, οι οποίες χρησιμοποιούνται για την υλοποίηση ενός φίλτρου διαμέσου τιμής. Η διαδικασία λήψης μετρήσεων και εξαγωγής διάμεσου επαναλαμβάνεται όσες φορές ορίζεται από την σταθερά MEAN_LENGTH προκειμένου να εξαχθεί ένας μέσος όρος των μετρήσεων για κάθε κανάλι ο οποίος αποθηκεύεται στον πίνακα averaging_sum.

Ο υπολογισθείς μέσος όρος για κάθε κανάλι αποστέλλεται στον μετατροπέα UART>USB κι από εκεί στον συνδεδεμένο Η/Υ. Αφού προστεθεί η τρέχουσα τιμή στον πίνακα t_sum για κάθε κανάλι μηδενίζεται ο πίνακας averaging_sum.

Κάθε 10 επαναλήψεις ανανεώνεται η οθόνη LCD με τον συσσωρευτικό μέσο όρο ανά κανάλι ο οποίος εξάγεται χρησιμοποιώντας τον πίνακα t_sum και την μεταβλητή t_count.

Τέλος αφού ολοκληρωθεί ο βρόχος ο μικροελεγκτής εισέρχεται σε κατάσταση εξοικονόμησης ενέργειας μέχρι την επομένη διακοπή του timer E0.

6.3 Αποτελέσματα μετρήσεων

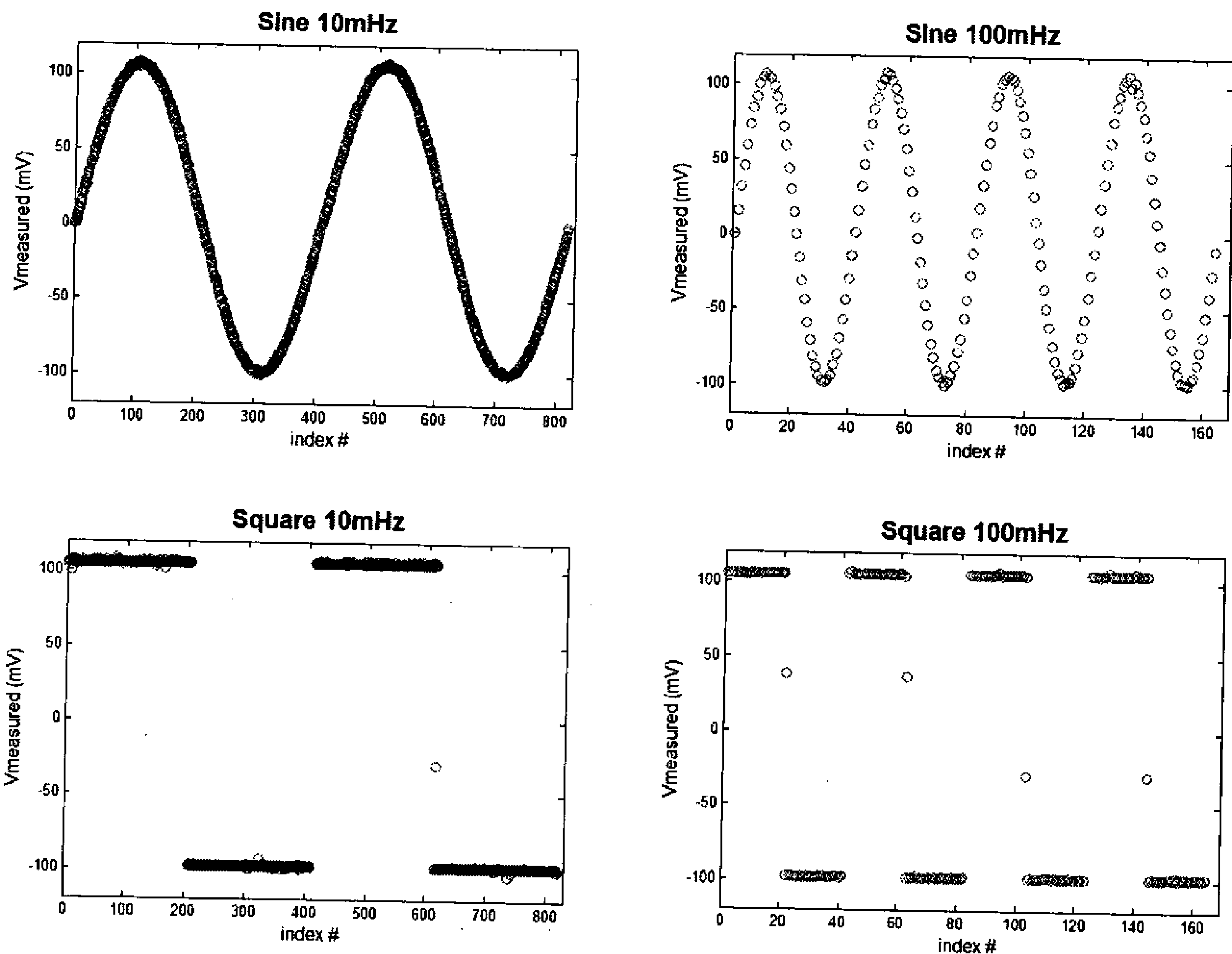
Στον πίνακα 6-B απεικονίζεται ο μέσος όρος 100 μετρήσεων του συστήματος για κάθε τιμή της μεταβαλλόμενης τάση αναφοράς που δόθηκε ως είσοδος. Το μέσο απόλυτο σφάλμα σύμφωνα με τις μετρήσεις είναι χαμηλότερο από 5 mV σε όλες τις περιπτώσεις.

V_{in} (mV)	Vread (mV)	V_{in} (mV)	Vread (mV)
-975	-974	25	24
-900	-898	50	48
-800	-799	100	99
-700	-699	150	151
-600	-599	200	201
-500	-499	250	252
-400	-400	300	300
-300	-301	400	400
-250	-249	500	499
-200	-202	600	599
-150	-152	700	699
-100	-100	800	798
-50	-50	900	899
-25	-25	975	974
0	0		

Πίνακας 6-B: Ενδεικτικές μετρήσεις συστήματος

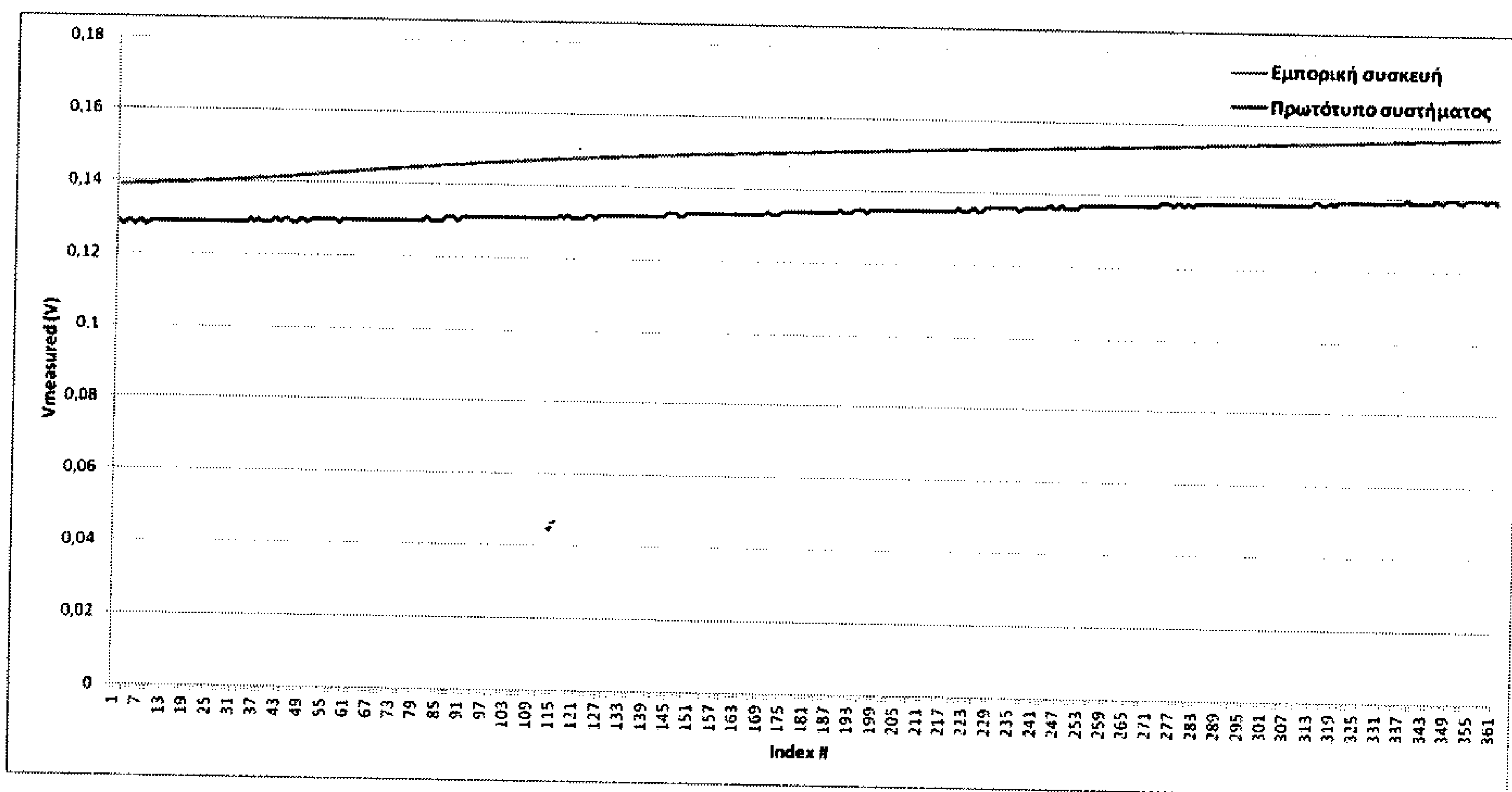
Στο διάγραμμα 6-III φαίνεται η απόκριση του συστήματος για είσοδο ημιτονικού και τετραγωνικού σήματος συχνότητας 10 και 100mHz και πλάτους κατά προσέγγιση 200mV peak-peak. Σημειώνεται ότι οι μετρήσεις αυτές δεν πραγματοποιήθηκαν με χρήση πλήρως θωρακισμένου καλωδίου.

Η τελική μορφή της διάταξης



Εικόνα 6-III: Απόκριση συστήματος σε ημιτονικό και τετραγωνικό σήμα

Στο διάγραμμα 6-IV απεικονίζονται ενδεικτικά αποτελέσματα μετρήσεων που έχουν διεξαχθεί με την συσκευή από το Γεωπονικό Πανεπιστήμιο Αθηνών σε σύγκριση με αποτελέσματα μετρήσεων χρησιμοποιώντας εξειδικευμένη εμπορική συσκευή.



Εικόνα 6-IV: Ενδεικτικές μετρήσεις σε βιολογικά δείγματα

7 Προτάσεις για μελλοντική βελτίωση της διάταξης

Όπως σε κάθε άλλο σύστημα έτσι και στο παρόν υπάρχουν περιθώρια βελτίωσης, είτε με την τροποποίηση των υπάρχοντων λειτουργιών είτε προσθέτοντας / αντικαθιστώντας κάποια τμήματα. Παρακάτω παρατίθενται ενδεικτικές προτάσεις βελτίωσης του συστήματος συνοδευόμενες από μια συνοπτική περιγραφή.

➤ **Συγγραφή εξειδικευμένου λογισμικού επεξεργασίας & προβολής δεδομένων σε Η/Υ**

Στην παρούσα φάση οι ανάγκες σε αυτό τον τομέα καλύπτονται από το πρόγραμμα που έχει γραφεί σε περιβάλλον LabVIEW ωστόσο θα ήταν επιθυμητό να υπάρχει μια εξειδικευμένη εφαρμογή η οποία θα ήταν βελτιστοποιημένη για το σύστημα διότι για την εκτέλεση ενός προγράμματος LabVIEW είναι απαραίτητη η εγκατάσταση των σχετικών βιβλιοθηκών με αποτέλεσμα αυξημένες απαιτήσεις επεξεργαστικής ισχύος και αποθηκευτικού χώρου.

➤ **Απευθείας προβολή στην LCD της συγκέντρωσης παθογόνου στο εξεταζόμενο δείγμα**

Αποτελεί το κύριο βήμα για την μετατροπή του συστήματος σε πραγματική συσκευή επιτόπιας διάγνωσης χαμηλού κόστους, την χρήση για την οποία προορίζεται.

Απαραίτητη προϋπόθεση φυσικά για την πραγματοποίηση αυτής της πρότασης είναι να προσδιοριστεί επακριβώς η συσχέτιση μεταξύ των μετρούμενων τάσεων και της συγκέντρωσης των παθογόνων. Επί της παρούσας φάσης λαμβάνονται μετρήσεις από την ερευνητική ομάδα του τμήματος Βιοτεχνολογίας, Γεωπονικού Πανεπιστημίου Αθηνών και επεξεργάζονται τα αποτελέσματα με τελικό σκοπό τον προσδιορισμό της σχέσης που συνδέει τις μετρούμενες τάσεις με την συγκέντρωση παθογόνου σε ποικιλία εξεταζόμενων δειγμάτων.

➤ **Δυνατότητα επιλογής παθογόνου προς ανίχνευση**

Με την ολοκλήρωση του προηγούμενου βήματος θα διευκρινιστεί και η μεταβολή της απόκρισης σε σχέση με το είδος του παθογόνου που αναζητείται. Υποθέτοντας ότι η συσχέτιση συγκέντρωσης - μετρήσεων ακολουθεί ένα συγκεκριμένο μοτίβο με την μεταβολή μόνο κάποιων λειτουργικών παραμέτρων θα είναι απλή η υλοποίηση ενός περιβάλλοντος διεπαφής με τον χρήστη μέσω του οποίου θα επιλέγεται ο τύπος παθογόνου και θα ορίζονται δυναμικά οι παράμετροι ανίχνευσης από το σύστημα.

➤ **Ενσωμάτωση αναλογικού front-end για τα κανάλια μέτρησης**

Αν και στο σύστημα ενσωματώνονται ψηφιακά φίλτρα στο πρόγραμμα του μικροελεγκτή είναι βέβαιο πως μπορεί να υπάρξει αισθητή βελτίωση στην ποιότητα των μετρήσεων με την υλοποίηση ενός κυκλώματος που θα περιλαμβάνει αναλογικά φίλτρα και δικτυώματα προσαρμογής για τους ADC. Ωστόσο υπάρχουν και μειονεκτήματα που οφείλονται στην σημαντική αύξηση της πολυπλοκότητας όπως η αύξηση του κόστους κατασκευής, η αύξηση του όγκου και βάρους της συσκευής καθώς και η ανάγκη για δημιουργία επιπλέον γραμμών τροφοδοσίας.

➤ **Χρήση εξωτερικού ολοκληρωμένου A/D μετατροπέα**

Παρότι η χρήση των ενσωματωμένων στον μικροελεγκτή ADC απλοποιεί την διαδικασία λήψης μετρήσεων και φυσικά τον σχεδιασμό του συστήματος με άμεσο επακόλουθο την διατήρηση του κόστους κατασκευής σε χαμηλά επίπεδα υπάρχουν κάποιοι περιορισμοί που προκύπτουν από την επιλογή αυτή.

Η χρήση εξωτερικών ολοκληρωμένων ADC μπορεί να επιφέρει σημαντικές βελτιώσεις οι σημαντικότερες εκ των οποίων είναι:

Δυνατότητα για υψηλότερη ανάλυση/ακρίβεια μετρήσεων, πχ με χρήση ευρέως διαδεδομένων 16bit ADC και διατηρώντας την τάση αναφοράς στο 1V το LSB αντιστοιχεί σε $\sim 0,03\text{mV}$ έναντι $\sim 0,49\text{mV}$ για τον ενσωματωμένο 12bit A/D.

Καλύτερος διαχωρισμός του αναλογικού τμήματος από τα θορυβώδη ψηφιακά. Προτείνεται να δημιουργηθεί ένα επιπλέον PCB στο οποίο θα συνδέονται απευθείας τα ηλεκτρόδια μέτρησης και θα περιέχει εξ ολοκλήρου το αναλογικό τμήμα του συστήματος μαζί με το απαραίτητο κύκλωμα τροφοδοσίας.

Άμεσο επακόλουθο της αλλαγής αυτής θα είναι και η μείωση της επίδρασης του θορύβου περιβάλλοντος στις μετρήσεις αφού θα ελαχιστοποιηθεί το μήκος διαδρομής των αναλογικών σημάτων και οι μετρήσεις θα μεταφέρονται στην κυρίως συσκευή ως ψηφιακά σήματα χρησιμοποιώντας καλώδια με χαμηλότερο αριθμό αγωγών με τα προφανή πλεονεκτήματα σε ευκαμψία, ευκολία σύνδεσης και θωράκισης που επιφέρει αυτό. πχ για μια κλασική υλοποίηση με χρήση SPI δίαυλου απαιτούνται μόνο 4 αγωγοί για την μεταφορά των σημάτων έναντι 16 αγωγών για την απευθείας μεταφορά των αναλογικών σημάτων (\pm για κάθε κανάλι, διαφορική μέτρηση).

➤ **Εξέλιξη των ηλεκτροδίων λήψης μετρήσεων**

Υποθέτοντας ότι τα αποτελέσματα από την πρώτη φάση εφαρμογής είναι θετικά θα πρέπει να επανεξεταστεί το θέμα των ηλεκτροδίων λήψης μετρήσεων μιας που τα ηλεκτρόδια της Dropsens έχουν πρακτικά απαγορευτικό κόστος για γενική χρήση.

Είτε θα πρέπει να βρεθεί κάποιος εναλλακτικός προμηθευτής που θα προσφέρει παρόμοιο προϊόν σε χαμηλότερο κόστος είτε να εξελιχθεί κάποιος από τους άλλους τύπους. Πιο πρόσφορος για εξέλιξη είναι ο τύπος εγχαραγμένων σε PCB ηλεκτροδίων λόγω της εύκολης μαζικής παραγωγής και του χαμηλότερου κόστους κατασκευής. Έχει σχεδιαστεί ήδη ένα πρότυπο για δοκιμές με επιχρυσωμένες επαφές το οποίο θεωρείται ότι μπορεί να έχει θετικά αποτελέσματα.

➤ **Μετάβαση σε μικροελεγκτές xmega -U series**

Η καινούργια σειρά μικροελεγκτών ATxmega -U series στηρίζεται στην ίδια πλατφόρμα με την -A σειρά που χρησιμοποιείται στην τρέχουσα έκδοση της συσκευής παρέχοντας τα ίδια ή ελαφρώς βελτιωμένα χαρακτηριστικά και διατηρώντας πλήρη συμβατότητα όσον αφορά την διάταξη ακροδεκτών ενώ παράλληλα έχουν διορθωθεί αρκετά από τα προβλήματα που εντοπίστηκαν στις πρώτες γενιές των μικροελεγκτών xmega.

Επιπλέον ενσωματώνεται ελεγκτής Full Speed USB παρέχοντας έτσι την δυνατότητα να καταργηθεί εξ ολοκλήρου το ενδιάμεσο κύκλωμα μετατροπής USB→serial και να πραγματοποιείται πιο αξιόπιστα και ταχύτερα η διαμεταγωγή των δεδομένων προς H/Y.

Τέλος λόγω αυξημένης παραγωγής η νέα σειρά διατίθεται σε χαμηλότερο κόστος από τους αντίστοιχους -A series xmega ενώ για την προσαρμογή του προγράμματος στην νέα σειρά απαιτούνται ελάχιστες αλλαγές.

Έχει ήδη γίνει προμήθεια μικροελεγκτών της νέας σειράς και αναμένεται συσκευές που θα κατασκευαστούν μελλοντικά να βασίζονται σε αυτή την σειρά αποκλειστικά μιας που δεν συντρέχει λόγος για το αντίθετο.

➤ **Κατασκευή μεταλλικού κουτιού**

Όπως είναι προφανές η αντικατάσταση του κουτιού με μεταλλικό θα παρέχει καλύτερη H/M θωράκιση και φυσική προστασία της συσκευής αλλά η παραγωγή ενός μεταλλικού κουτιού κατά παραγγελία σε μικρή κλίμακα έχει αρκετά αυξημένο κόστος μιας που απαιτείται εξειδικευμένος εξοπλισμός για την κοπή των πλαισίων και την διάνοιξη οπών με ακρίβεια.

➤ **Χρονισμός με κρυσταλλικό ταλαντωτή**

Ο χρονισμός με χρήση κρυσταλλικού ταλαντωτή έναντι του ενσωματωμένου RC ταλαντωτή που χρησιμοποιείται στην παρούσα έκδοση του συστήματος επιτρέπει την ελαχιστοποίηση των σφαλμάτων στις σειριακές επικοινωνίες και γενικώς παρέχει σημαντική βελτίωση της ακρίβειας κατά την τήρηση χρονικών διαστημάτων με χρήση timers κλπ.

Ο μόνος λόγος που δεν υπήρχε θέση για κρυσταλλικό ταλαντωτή στην παρούσα σχεδίαση του PCB είναι ότι λόγω της θέσης των ακροδεκτών τοποθέτησης κρυστάλλου είναι απαραίτητη η χρήση επιμεταλλωμένων vias για την μεταφορά των συνδέσεων στην άλλη πλευρά του PCB αφού δεν υπάρχει χώρος για τοποθέτηση κανονικών pads στην περιοχή.

Μιας που η παραγωγή επιμεταλλωμένων οπών απαιτεί ειδικές μεθόδους και εξοπλισμό με αποτέλεσμα αυτή η τεχνολογία να μην είναι διαθέσιμη σε ερασιτεχνική παραγωγή PCB δεν ήταν εφικτό να τοποθετηθεί θέση κρυστάλλου χωρίς να γίνουν άλλες υποχωρήσεις. Σε επόμενη έκδοση του PCB που θα παραχθεί από βιομηχανία με αντίστοιχες δυνατότητες έχει ενσωματωθεί θέση για κρύσταλλο στην άνω όψη του τυπωμένου.

Βελτίωση φορητότητας

Δεδομένου ότι εάν υλοποιηθούν η 2^η και η 3^η πρόταση από τις παραπάνω θα είναι ενδεχομένως περιττή η ύπαρξη επικοινωνίας με Η/Υ είναι δυνατή η σχεδίαση μιας τροποποιημένης, πιο εστιασμένης έκδοσης του συστήματος η οποία θα παρέχει μόνο τα απολύτως απαραίτητα με αποτέλεσμα μια εκτιμώμενη μείωση της τάξης του 75% σε όγκο και 50% σε βάρος σε σχέση με την τρέχουσα έκδοση. Επίσης είναι εφικτή η μείωση της κατανάλωσης ισχύος και η τροφοδοσία αποκλειστικά από ένα ζεύγος κοινών μπαταριών τύπου AAA.

➤ **Ενσωμάτωση οθόνης γραφικών**

Η αντικατάσταση της οθόνης χαρακτήρων με οθόνη γραφικών θα παρέχει την δυνατότητα για απεικόνιση διαγραμμάτων, πινάκων και γραφημάτων παράλληλα με το κείμενο βελτιώνοντας έτσι την παρουσίαση των δεδομένων και την διεπαφή με τον χρήστη.

Προτείνεται η ενσωμάτωση οθόνης τύπου TFT, αφού το αυξημένο σχετικά κόστος αγοράς αντισταθμίζεται από τα θετικά χαρακτηριστικά που διακρίνουν αυτό τον τύπο όπως πολύ καλή χρωματική απεικόνιση, ικανοποιητική αντίθεση, απευθείας λειτουργία με 3,3V τάση τροφοδοσίας και αρκετά χαμηλή κατανάλωση ισχύος.

➤ **Ενσωμάτωση Bluetooth module**

Η ενσωμάτωση ενός απλού Bluetooth-serial bridge μπορεί να παρέχει ασύρματη επικοινωνία και μετάδοση δεδομένων σε Η/Υ καταργώντας έτσι την ανάγκη για ενσύρματη σύνδεση μέσω USB.

Επιπλέον παρέχεται η δυνατότητα για διασύνδεση του συστήματος με smartphones βασισμένα σε λειτουργικό android και η αποστολή δεδομένων σε αυτά προκειμένου να αποθηκευτούν χρησιμοποιώντας ευρέως διαθέσιμες εφαρμογές που εξομοιώνουν σειριακό τερματικό.

Εάν συγγράφει κατάλληλη εφαρμογή φυσικά είναι επίσης δυνατή η άμεση επεξεργασία των δεδομένων από τον ενσωματωμένο επεξεργαστή του κινητού τηλέφωνου και η προβολή των αποτελεσμάτων στην οθόνη του.

Επί του παρόντος γίνονται οι πρώτες δοκιμές με ένα Bluetooth→serial module βασισμένο στο ολοκληρωμένο BC417 της CSR.

8 Σύνοψη – Ανασκόπηση

Ανακεφαλαιώνοντας, το σύστημα που αναπτύχθηκε σε γενικές γραμμές ικανοποιεί τους στόχους και τις απαιτήσεις που τέθηκαν, καλύπτοντας τις τρέχουσες ανάγκες της εφαρμογής και παρέχει την δυνατότητα για εύκολη εξέλιξη και προσαρμογή, μια βασική προτεραιότητα κατά την σχεδίαση του συστήματος, δεδομένου ότι η τεχνολογία BERA είναι ακόμα σε πρώιμο στάδιο και το σύστημα πρέπει να είναι σε θέση να την υποστηρίξει καθώς προκύπτουν νέα δεδομένα και ανάγκες.

Σε σχέση με το προγενέστερο σύστημα που υπήρχε έχει επιτευχθεί βελτίωση σε όλους τους βασικούς τομείς καθώς παρά τα βελτιωμένα χαρακτηριστικά το κόστος υλοποίησης είναι περίπου 50% χαμηλότερο, η χρήση του είναι απλούστατη χωρίς να απαιτούνται ιδιαίτερες γνώσεις, η κατασκευή διασφαλίζει επαρκώς την προστασία των ηλεκτρονικών κυκλωμάτων ενώ τέλος παρότι προσαρμοσμένο στην συγκεκριμένη εφαρμογή το σύστημα διατηρεί έναν ικανοποιητικό βαθμό ευελιξίας.

Ένας τομέας στον οποίο με απόλυτα μέτρα πάσχει το σύστημα είναι το συνολικό κόστος κατασκευής το οποίο δεν μπορεί να χαρακτηριστεί χαμηλό, όμως εάν λάβουμε υπ' όψιν ότι πρόκειται για πρωτότυπη διάταξη με έμφαση στην ευελιξία και όχι αυστηρά καθορισμένων προδιαγραφών εμπορικό προϊόν μαζικής παραγωγής καθώς και ότι τα μέρη του κουτιού είναι κατασκευασμένα κατά παραγγελία δικαιολογείται πλήρως το κόστος κατασκευής. Εάν γίνει παραγωγή σε μεγαλύτερη κλίμακα σε συνδυασμό με κάποιες τροποποιήσεις όπως πχ χρήση κάποιου εμπορικά διαθέσιμου έτοιμου κουτιού εκτιμάται ότι το κόστος κατασκευής μπορεί να μειωθεί στο 40-60% του αρχικού ακόμα κι αν διατηρηθεί αναλλοίωτος ο χαρακτήρας της συσκευής.

Όσον αφορά την σειρά μικροελεγκτών xmega που χρησιμοποιήθηκε για την πραγματοποίηση αυτής της εργασίας παρά το γεγονός ότι προσφέρονται προχωρημένα χαρακτηριστικά και πληθώρα ενσωματωμένων περιφερειακών σε σχέση με ανταγωνιστικούς 8bit μικροελεγκτές αρκετά από αυτά τα χαρακτηριστικά περιορίζονται από σχεδιαστικά και κατασκευαστικά προβλήματα στις πρώτες εκδόσεις πυριτίου. Ως εκ τούτου κρίνεται επιτακτικό σε μελλοντικές κατασκευές να χρησιμοποιηθεί η νέα σειρά -U στην οποία έχουν γίνει διορθώσεις ενώ διαθέτει επιπλέον ενσωματωμένο ελεγκτή USB, διατίθεται σε χαμηλότερες τιμές για τα ίδια χαρακτηριστικά και πλέον υποστηρίζεται θερμότερα από την κατασκευάστρια εταιρία σε σχέση με την βασική σειρά.

Γενικότερα αν και η μετάβαση από τους tiny/megaAVR στην σειρά xmega μπορεί αρχικά να ξενίσει λόγω των διαφοροποιήσεων σε ονομασίες καταχωρητών και τον τρόπο λειτουργίας μερικών περιφερειακών ένας σχετικά έμπειρος προγραμματιστής σύντομα θα προσαρμοστεί άψογα καθότι οι xmega έχουν σχεδιαστεί εξ αρχής ώστε να αποτελούν μια ενιαία πλατφόρμα με κοινά

χαρακτηριστικά και πιο αυστηρά δομημένο τρόπο χειρισμού καταχωρητών και περιφερειακών, ακολουθώντας σε γενικές γραμμές τα πρότυπα που χρησιμοποιούνται σε μικροεπεξεργαστές τύπου ARM.

Τέλος σημειώνεται ότι το σύστημα που αναπτύχθηκε στην παρούσα εργασία χρησιμοποιήθηκε από το τμήμα Βιοτεχνολογίας του Γ.Π.Α για την λήψη μετρήσεων σε επιστημονική εργασία που παρουσιάστηκε σε διεθνές συνέδριο βιοαισθητήρων [Ref. 39] ενώ παράλληλα έχουν αποσταλεί αντίγραφα του συστήματος για δοκιμή και αξιολόγηση στους εξής φορείς:

- The Macfarlane Burnet Institute for Medical Research and Public Health
Ερευνητικό ινστιτούτο με έμφαση στην ιατρική και την διαγνωστική, Αυστραλία (www.burnet.edu.au).
- Tasting Room
Εταιρία οινολογίας με εξειδίκευση στην διεξαγωγή ελέγχων ποιότητας σε δείγματα οίνου, Η.Π.Α. (www.tastingroom.com).

9 Αναφορές

Βιβλία

- [1] Barnett, R.H. et al., "Embedded C Programming And The Atmel AVR 2nd Edition, Delmar Publishers Inc, 2006, ISBN: 978-1418039592
- [2] Gadre, D.V., "Programming and customizing the AVR Microcontroller", McGraw-Hill/TAB Electronics, 2000, ISBN: 978-0071346665
- [3] Πεκμεστζή, Κ., "Συστήματα μικροϋπολογιστών II", Εκδόσεις Συμμετρία, Αθήνα 2009, ISBN: 978-960-266-269-4

Datasheets – Manuals

- [4] HP Infotech S.R.L., "CodeVisionAVR User Manual", (<http://hpinfotech.ro>)
- [5] Keithley Inc., "Keithley 2000 Multimeter User's Manual", (<http://www.keithley.com>)
- [6] Keithley Inc., "Keithley Series 2600 System Sourcemeter User's Manual", (<http://www.keithley.com>)
- [7] Rigol Inc., "Rigol DS1000D/E user's Guide", (<http://www.rigolna.com>)
- [8] Microchip Inc., "MCP2200 datasheet", 2010, (<http://www.microchip.com>)
- [9] Intel Corp. "ATX Specification Version 2.1", 2002, (<http://www.intel.com>)
- [10] Atmel Corp., "XMEGA A manual", (<http://www.atmel.com>)
- [11] Atmel Corp., "ATxmega64A3 Datasheet", (<http://www.atmel.com>)
- [12] WINSTAR Display Corp., "WH2004B 20x4 Character LCD datasheet", (<http://www.winstar.com.tw>)
- [13] Maxim Integrated Corp., "ICL7660 Datasheet", (<http://www.maximintegrated.com>)
- [14] Micrel Inc, "MIC5209 datasheet", 2006 (<http://www.micrel.com>)
- [15] Dropsens S.L., "8x110 Screen-printed Carbon electrodes datasheet" http://www.dropsens.com/en/pdfs_productos/new_brochures/8x110.pdf

Σημειώσεις - Άρθρα - Δημοσιεύσεις

- [16] Καλτσάς, Γ, "Μικροελεγκτές & Ενσωματωμένα Συστήματα, παρουσίαση μαθήματος", Αθήνα 2011, (<https://education.teiath.gr>)
- [17] Λαϊνάς Α., Σεφερλή, Χ., "Ανάπτυξη διάταξης ελέγχου βιολογικών δειγμάτων με χρήση μικροελεγκτή", Πτυχιακή εργασία, Τμήμα Ηλεκτρονικής, ΤΕΙ Αθήνας, 2012
- [18] Kintzios, S. et al., "Bioelectric recognition assay (BERA)", Biosensors & Bioelectronics 16, 2001, pp 325-336

- [19] Kintzios, S. et al., "Study on the mechanism of Bioelectric Recognition Assay: evidence for immobilized cell membrane interactions with viral fragments", *Biosensors and Bioelectronics* 20, 2004, pp 907-916
- [20] Voumvouraki, E. et al., "Differential screening of the neurotoxicity of insecticides by means of a novel electrophysiological biosensor", *Euroensors XXV*, September 4-7, 2011, Athens, Greece [*Procedia Engineering* 25, 2011, pp 964 - 967]
- [21] Varelas, V. et al., "Development of a cellular biosensor for the detection of 2,4,6-trichloroanisole (TCA)", *Talanta* 84, 2011, pp 936-940
- [22] Perdikaris, P. et al., "Development of a portable, high throughput biosensor system for rapid plant virus detection", *Journal of Virological Methods* 177, 2011, pp 94- 99
- [23] Reyes De Corcuera J.I., Cavalieri R.P. "Biosensors", *Encyclopedia of Agricultural, Food, and Biological Engineering*, Heldman, D. Editor. Marcel Dekker Inc., 2003, pp 119-123
- [24] Αλεξανδριδης, Α., "Ευφυή Συστήματα, παρουσίαση μαθήματος", Αθήνα 2011, (<https://education.teiath.gr>)
- [25] Atmel Corp., "AVR042: AVR Hardware Design Considerations", Application Note, (<http://www.atmel.com>)
- [26] Atmel Corp., "AVR120: Characterization and Calibration of the ADC on an AVR", Application Note, (<http://www.atmel.com>)
- [27] Atmel Corp., "AVR127: Understanding ADC parameters", Application Note, (<http://www.atmel.com>)
- [28] Atmel Corp., "AVR1000: Getting Started Writing C-code for XMEGA", Application Note, (<http://www.atmel.com>)
- [29] Atmel Corp., "AVR1005: Getting started with XMEGA", Application Note, (<http://www.atmel.com>)
- [30] Atmel Corp., "AVR1010: Minimizing the power consumption of Atmel AVR XMEGA devices", Application Note, (<http://www.atmel.com>)
- [31] Atmel Corp., "AVR1012: XMEGA A Schematic Checklist", Application Note, (<http://www.atmel.com>)
- [32] Atmel Corp., "AVR1300: Using the XMEGA ADC", Application Note, (<http://www.atmel.com>)
- [33] Atmel Corp., "AVR1306: Using the XMEGA Timer/Counter", Application Note, (<http://www.atmel.com>)
- [34] Jones, D.L., "PCB Design Tutorial", 2004, (<http://alternatzone.com/electronics/files/PCBDesignTutorialRevA.pdf>)
- [35] Williams, J., Owen, T., "Performance Verification of Low Noise, Low Dropout Regulators", *Linear Technology Application Note*, March 2000 (<http://www.linear.com>)
- [36] Kester, W., "A Grounding Philosophy for Mixed-Signal Systems", *Electronic Design Analog Applications Issue*, June 23, 1997, p. 29.

- [37] Kester, W, Bryant, J., "GROUNDING IN MIXED SIGNAL SYSTEMS", Analog Devices Corp, Hardware design techniques (<http://www.analog.com>)
- [38] Ott, H.W., "Partitioning and layout of a mixed signal PCB", Printed Circuit Design, CMP Media, June 2001
([www.hottconsultants.com/pdf files/june2001pcd_mixedsignal.pdf](http://www.hottconsultants.com/pdf_files/june2001pcd_mixedsignal.pdf))
- [39] Brooks, D., "Splitting Planes For Speed and Power", Printed Circuit Design, CMP Media, December 2000
- [40] Blouchos, P., Voumnouraki, E., **Moschos, A.**, Kaltsas, G., Dimitrakis, A., Kintzios, S., "Membrane-engineered bacterial cells as promising biorecognition elements for the detection of 2,4,6-trichloroanisole in wine" Biosensors 2012, Cancun, Mexico, May 15-18, 2012.

Ιστοσελίδες

- [41] Γεωπονικό πανεπιστήμιο Αθηνών, Τμήμα Γεωπονικής Βιοτεχνολογίας (<http://gbt.aua.gr/el>)
- [42] SK Embio Diagnostics, cell-based biosensors research & production (<http://www.embiodiagnostics.com>)
- [43] AVR freaks, επίσημο forum υποστήριξης AVR μικροελεγκτών (<http://www.avrfreaks.net>)
- [44] Cornell University ECE 4760 Designing with Microcontrollers (<http://people.ece.cornell.edu/land/courses/ece4760/>)
- [45] Embedded Gurus, Blog περί ενσωματωμένων συστημάτων (<http://embeddedgurus.com>)
- [46] Hlektronika.gr, Ελληνικό forum Ηλεκτρονικών (<http://www.hlektronika.gr>)
- [47] Abacom, Electronics design software (<http://www.abacom-online.de/>)
- [48] Labcenter Electronics Ltd., Electronics design & emulation software (<http://www.labcenter.com>)
- [49] National Instruments Corp., Measurement and control hardware & software (<http://www.ni.com>)

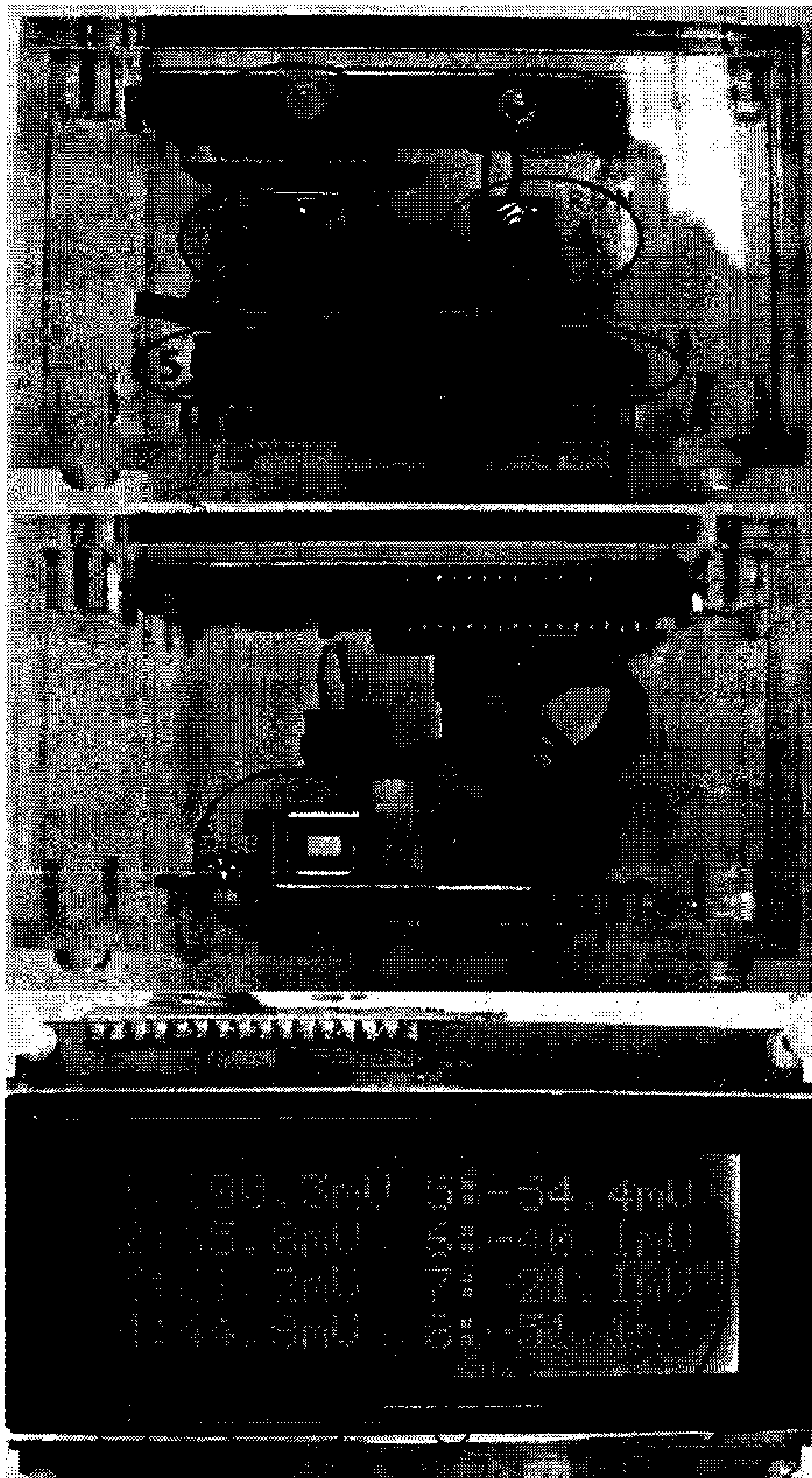
Παραρτήματα

A. Quick start guide (English)

Introduction

The BERA diagnostic device is a portable, low cost multichannel measurement system, designed to simplify the process of examining biological samples using the BERA biosensor cells developed by Embio Diagnostics and allow for either completely autonomous (results are displayed on integrated LCD) or computer assisted operation (measurements are also transmitted to computer for further processing).

Hardware description



1. Power indication LED
Green → USB powered
Red → External power
2. Operation LED
Flashing → Running
Off → Halted
3. Power control switch
4. Operation control switch
5. Electrodes input connector
(polarized)
6. USB cable connector
7. External power connector
(2,1mm diameter DC jack)
8. Measurement channels
9. Results

Installation - Operation

- Place the device in a dry location, away from sources of electromagnetic radiation.
- Make sure that the power switch (3) is set to "OFF" position and the operation switch (4) to "STOP" position.
- If you intent to collect the data using a computer connect the device via the USB connector (6) using a USB type A→B cable.
- The computer should identify the device as "MCP2200 USB Serial Port Emulator" and prompt you to install the drivers.
- Look for the "Drivers" folder in the supplied disc and select the folder that matches your OS.
- If you want to use the device independently connect a 9V battery to the EXT plug (7) using the supplied adaptor or connect a suitable AC/DC power source using a 2,1mm jack.
- Prepare the biological samples to be examined.
- Connect the electrodes PCB to the main device input (5) and place the electrodes in the samples.
- Power up the device, turning switch (3) to the respective position according to the power source you have connected.
- If you are using a computer start the supplied measurements logging software now.
- After the initialization is complete the message "Ready" will be displayed on the LCD.
- Turn switch (4) to the RUN position
- As soon as examination is completed turn off the device, disconnect and rinse electrodes with distilled water.

Specifications

Power supply	4,5-12V DC / 5-9V AC
Current consumption (typ.)	45mA (approx.)
Battery run-time (9V PP3)	10h (approx.)
Display	20*4 Character LCD, blue BL
Initialization time	3 sec (approx.)
Measurement channels	8 differential channels
Sampling Rate	4 SPS (output per Ch, ±5%)
Input voltage range	-850mV ~ 850mV
Resolution	1mV
Max abs. error	±5mV
Dimensions (main unit)	125*103*77mm (W*L*H)
Weight (main unit)	405gr (approx.)
Working temperature	5 ~ 50 °C

Β. Κώδικας προγράμματος μικροελεγκτή

```
/*  
Project : BERA diagnostic system  
Version : F  
Date   : 31/12/2012  
Author : Moschos Anastasios  
Company : TEI of Athens, microSENSES Lab  
Comments:  
Chip type      : ATxmega64A3  
Program type   : Application  
AVR Core Clock frequency: 2,000000 MHz  
Memory model   : Small  
Data Stack size : 1024  
*/  
  
// I/O Registers definitions  
#include <io.h>  
// Standard Input/Output functions  
#include <stdio.h>  
// Delay functions  
#include <delay.h>  
// Alphanumeric LCD interface functions  
#include <alcd.h>  
// Sleep functions  
#include <sleep.h>  
// Stdlib functions  
#include <stdlib.h>  
// Declare constants  
#define LS_SLOPE_B 0.98395  
#define LS_YP_B   -0.48669  
#define LS_SLOPE_S 1.00359  
#define LS_YP_S   -0.51164  
#define INTRO_DELAY 500  
#define DELAY_CNT 250  
#define MEAN_LENGTH 3  
// Declare global variables  
#if (MEAN_LENGTH>=15)  
signed long int averaging_sum[8];  
#else  
signed int averaging_sum[8];  
#endif  
#if (DELAY_CNT>=255)  
unsigned int counter;  
#else  
unsigned char counter;  
#endif  
char i,j,k,str[15],temp[15];  
signed int data[8][3];  
unsigned long t_count;
```

Κώδικας προγράμματος μικροελεγκτή

```
signed long t_sum[8];
// System Clocks initialization
void system_clocks_init(void)
{
  unsigned char n,s;
  // Optimize for speed
  #pragma optimize-
  // Save interrupts enabled/disabled state
  s=SREG;
  // Disable interrupts
  #asm("cli")
  // Internal 2 MHz RC oscillator initialization
  // Enable the internal 2 MHz RC oscillator
  OSC.CTRL|=OSC_RC2MEN_bm;
  // System Clock prescaler A division factor: 1
  // System Clock prescalers B & C division factors: B:1, C:1
  // ClkPer4: 2000,000 kHz
  // ClkPer2: 2000,000 kHz
  // ClkPer: 2000,000 kHz
  // ClkCPU: 2000,000 kHz
  n=(CLK.PSCTRL & ~(CLK_PSADIV_gm | CLK_PSBCDIV1_bm | CLK_PSBCDIV0_bm))) |
    CLK_PSADIV_1_gc | CLK_PSBCDIV_1_1_gc;
  CCP=CCP_IOREG_gc;
  CLK.PSCTRL=n;
  // Disable the autocalibration of the internal 2 MHz RC oscillator
  DFLLRC2M.CTRL&= ~DFLL_ENABLE_bm;
  // Wait for the internal 2 MHz RC oscillator to stabilize
  while ((OSC.STATUS & OSC_RC2MRDY_bm)!=0);
  // Select the system clock source: 2 MHz Internal RC Osc.
  n=(CLK.CTRL & (~CLK_SCLKSEL_gm)) | CLK_SCLKSEL_RC2M_gc;
  CCP=CCP_IOREG_gc;
  CLK.CTRL=n;
  // Disable the unused oscillators: 32 MHz, 32 kHz, external clock/crystal oscillator, PLL
  OSC.CTRL&= ~(OSC_RC32MEN_bm | OSC_RC32KEN_bm | OSC_XOSCEN_bm |
  OSC_PLEN_bm);
  // Peripheral Clock output: Disabled
  PORTCFG.CLKEVOUT=(PORTCFG.CLKEVOUT & (~PORTCFG_CLKOUT_gm)) |
  PORTCFG_CLKOUT_OFF_gc;
  // Restore interrupts enabled/disabled state
  SREG=s;
  // Restore optimization for size if needed
  #pragma optimize_default
}
// Ports initialization
void ports_init(void)
{
  // PORTA initialization
  // OUT register
  PORTA.OUT=0x00;
  // Bit0: Input
```

Κώδικας προγράμματος μικροελεγκτή

```
// Bit1: Input
// Bit2: Input
// Bit3: Input
// Bit4: Input
// Bit5: Input
// Bit6: Input
// Bit7: Input
PORTA.DIR=0x00;
// Bit0 Output/Pull configuration: Totempole/No
// Bit0 Input/Sense configuration: Input buffer disabled
// Bit0 inverted: Off
// Bit0 slew rate limitation: Off
PORTA.PIN0CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit1 Output/Pull configuration: Totempole/No
// Bit1 Input/Sense configuration: Input buffer disabled
// Bit1 inverted: Off
// Bit1 slew rate limitation: Off
PORTA.PIN1CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit2 Output/Pull configuration: Totempole/No
// Bit2 Input/Sense configuration: Input buffer disabled
// Bit2 inverted: Off
// Bit2 slew rate limitation: Off
PORTA.PIN2CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit3 Output/Pull configuration: Totempole/No
// Bit3 Input/Sense configuration: Input buffer disabled
// Bit3 inverted: Off
// Bit3 slew rate limitation: Off
PORTA.PIN3CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit4 Output/Pull configuration: Totempole/No
// Bit4 Input/Sense configuration: Input buffer disabled
// Bit4 inverted: Off
// Bit4 slew rate limitation: Off
PORTA.PIN4CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit5 Output/Pull configuration: Totempole/No
// Bit5 Input/Sense configuration: Input buffer disabled
// Bit5 inverted: Off
// Bit5 slew rate limitation: Off
PORTA.PIN5CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit6 Output/Pull configuration: Totempole/No
// Bit6 Input/Sense configuration: Input buffer disabled
// Bit6 inverted: Off
// Bit6 slew rate limitation: Off
PORTA.PIN6CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit7 Output/Pull configuration: Totempole/No
// Bit7 Input/Sense configuration: Input buffer disabled
// Bit7 inverted: Off
// Bit7 slew rate limitation: Off
PORTA.PIN7CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Interrupt 0 level: Disabled
```

Κώδικας προγράμματος μικροελεγκτή

```
// Interrupt 1 level: Disabled
PORTA.INTCTRL=(PORTA.INTCTRL & ~(PORT_INT1LVL_gm | PORT_INT0LVL_gm))
| PORT_INT1LVL_OFF_gc | PORT_INT0LVL_OFF_gc;
// Bit0 pin change interrupt 0: Off
// Bit1 pin change interrupt 0: Off
// Bit2 pin change interrupt 0: Off
// Bit3 pin change interrupt 0: Off
// Bit4 pin change interrupt 0: Off
// Bit5 pin change interrupt 0: Off
// Bit6 pin change interrupt 0: Off
// Bit7 pin change interrupt 0: Off
PORTA.INT0MASK=0x00;
// Bit0 pin change interrupt 1: Off
// Bit1 pin change interrupt 1: Off
// Bit2 pin change interrupt 1: Off
// Bit3 pin change interrupt 1: Off
// Bit4 pin change interrupt 1: Off
// Bit5 pin change interrupt 1: Off
// Bit6 pin change interrupt 1: Off
// Bit7 pin change interrupt 1: Off
PORTA.INT1MASK=0x00;
// PORTB initialization
// OUT register
PORTB.OUT=0x00;
// Bit0: Input
// Bit1: Input
// Bit2: Input
// Bit3: Input
// Bit4: Input
// Bit5: Input
// Bit6: Input
// Bit7: Input
PORTB.DIR=0x00;
// Bit0 Output/Pull configuration: Totempole/No
// Bit0 Input/Sense configuration: Input buffer disabled
// Bit0 inverted: Off
// Bit0 slew rate limitation: Off
PORTB.PIN0CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit1 Output/Pull configuration: Totempole/No
// Bit1 Input/Sense configuration: Input buffer disabled
// Bit1 inverted: Off
// Bit1 slew rate limitation: Off
PORTB.PIN1CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit2 Output/Pull configuration: Totempole/No
// Bit2 Input/Sense configuration: Input buffer disabled
// Bit2 inverted: Off
// Bit2 slew rate limitation: Off
PORTB.PIN2CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit3 Output/Pull configuration: Totempole/No
// Bit3 Input/Sense configuration: Input buffer disabled
```

Κώδικας προγράμματος μικροελεγκτή

```
// Bit3 inverted: Off
// Bit3 slew rate limitation: Off
PORTB.PIN3CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit4 Output/Pull configuration: Totempole/No
// Bit4 Input/Sense configuration: Input buffer disabled
// Bit4 inverted: Off
// Bit4 slew rate limitation: Off
PORTB.PIN4CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit5 Output/Pull configuration: Totempole/No
// Bit5 Input/Sense configuration: Input buffer disabled
// Bit5 inverted: Off
// Bit5 slew rate limitation: Off
PORTB.PIN5CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit6 Output/Pull configuration: Totempole/No
// Bit6 Input/Sense configuration: Input buffer disabled
// Bit6 inverted: Off
// Bit6 slew rate limitation: Off
PORTB.PIN6CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Bit7 Output/Pull configuration: Totempole/No
// Bit7 Input/Sense configuration: Input buffer disabled
// Bit7 inverted: Off
// Bit7 slew rate limitation: Off
PORTB.PIN7CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_INPUT_DISABLE_gc;
// Interrupt 0 level: Disabled
// Interrupt 1 level: Disabled
PORTB.INTCTRL=(PORTB.INTCTRL & ~(PORT_INT1LVL_gm | PORT_INT0LVL_gm))
| PORT_INT1LVL_OFF_gc | PORT_INT0LVL_OFF_gc;
// Bit0 pin change interrupt 0: Off
// Bit1 pin change interrupt 0: Off
// Bit2 pin change interrupt 0: Off
// Bit3 pin change interrupt 0: Off
// Bit4 pin change interrupt 0: Off
// Bit5 pin change interrupt 0: Off
// Bit6 pin change interrupt 0: Off
// Bit7 pin change interrupt 0: Off
PORTB.INTOMASK=0x00;
// Bit0 pin change interrupt 1: Off
// Bit1 pin change interrupt 1: Off
// Bit2 pin change interrupt 1: Off
// Bit3 pin change interrupt 1: Off
// Bit4 pin change interrupt 1: Off
// Bit5 pin change interrupt 1: Off
// Bit6 pin change interrupt 1: Off
// Bit7 pin change interrupt 1: Off
PORTB.INT1MASK=0x00;
// PORTC initialization
// OUT register
PORTC.OUT=0xFF;
// Bit0: Input
// Bit1: Input
```

Κώδικας προγράμματος μικροελεγκτή

```
// Bit2: Input
// Bit3: Input
// Bit4: Input
// Bit5: Input
// Bit6: Input
// Bit7: Input
PORTC.DIR=0x00;
// Bit0 Output/Pull configuration: Totempole/No
// Bit0 Input/Sense configuration: Sense both edges
// Bit0 inverted: Off
// Bit0 slew rate limitation: Off
PORTC.PIN0CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit1 Output/Pull configuration: Totempole/No
// Bit1 Input/Sense configuration: Sense both edges
// Bit1 inverted: Off
// Bit1 slew rate limitation: Off
PORTC.PIN1CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit2 Output/Pull configuration: Totempole/No
// Bit2 Input/Sense configuration: Sense both edges
// Bit2 inverted: Off
// Bit2 slew rate limitation: Off
PORTC.PIN2CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit3 Output/Pull configuration: Totempole/No
// Bit3 Input/Sense configuration: Sense both edges
// Bit3 inverted: Off
// Bit3 slew rate limitation: Off
PORTC.PIN3CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit4 Output/Pull configuration: Totempole/No
// Bit4 Input/Sense configuration: Sense both edges
// Bit4 inverted: Off
// Bit4 slew rate limitation: Off
PORTC.PIN4CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit5 Output/Pull configuration: Totempole/No
// Bit5 Input/Sense configuration: Sense both edges
// Bit5 inverted: Off
// Bit5 slew rate limitation: Off
PORTC.PIN5CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit6 Output/Pull configuration: Totempole/No
// Bit6 Input/Sense configuration: Sense both edges
// Bit6 inverted: Off
// Bit6 slew rate limitation: Off
PORTC.PIN6CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit7 Output/Pull configuration: Totempole/No
// Bit7 Input/Sense configuration: Sense both edges
// Bit7 inverted: Off
// Bit7 slew rate limitation: Off
PORTC.PIN7CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Interrupt 0 level: Disabled
// Interrupt 1 level: Disabled
```

Κώδικας προγράμματος μικροελεγκτή

```
PORTC.INTCTRL=(PORTC.INTCTRL & ~(PORT_INT1LVL_gm | PORT_INT0LVL_gm))
| PORT_INT1LVL_OFF_gc | PORT_INT0LVL_OFF_gc;
// Bit0 pin change interrupt 0: Off
// Bit1 pin change interrupt 0: Off
// Bit2 pin change interrupt 0: Off
// Bit3 pin change interrupt 0: Off
// Bit4 pin change interrupt 0: Off
// Bit5 pin change interrupt 0: Off
// Bit6 pin change interrupt 0: Off
// Bit7 pin change interrupt 0: Off
PORTC.INT0MASK=0x00;
// Bit0 pin change interrupt 1: Off
// Bit1 pin change interrupt 1: Off
// Bit2 pin change interrupt 1: Off
// Bit3 pin change interrupt 1: Off
// Bit4 pin change interrupt 1: Off
// Bit5 pin change interrupt 1: Off
// Bit6 pin change interrupt 1: Off
// Bit7 pin change interrupt 1: Off
PORTC.INT1MASK=0x00;
// PORTD initialization
// OUT register
PORTD.OUT=0xFF;
// Bit0: Input
// Bit1: Input
// Bit2: Input
// Bit3: Input
// Bit4: Input
// Bit5: Input
// Bit6: Input
// Bit7: Input
PORTD.DIR=0x00;
// Bit0 Output/Pull configuration: Totempole/No
// Bit0 Input/Sense configuration: Sense both edges
// Bit0 inverted: Off
// Bit0 slew rate limitation: Off
PORTD.PIN0CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit1 Output/Pull configuration: Totempole/No
// Bit1 Input/Sense configuration: Sense both edges
// Bit1 inverted: Off
// Bit1 slew rate limitation: Off
PORTD.PIN1CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit2 Output/Pull configuration: Totempole/No
// Bit2 Input/Sense configuration: Sense both edges
// Bit2 inverted: Off
// Bit2 slew rate limitation: Off
PORTD.PIN2CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit3 Output/Pull configuration: Totempole/No
// Bit3 Input/Sense configuration: Sense both edges
// Bit3 inverted: Off
```


Κώδικας προγράμματος μικροελεγκτή

```
// Bit3 slew rate limitation: Off
PORTD.PIN3CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit4 Output/Pull configuration: Totempole/No
// Bit4 Input/Sense configuration: Sense both edges
// Bit4 inverted: Off
// Bit4 slew rate limitation: Off
PORTD.PIN4CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit5 Output/Pull configuration: Totempole/No
// Bit5 Input/Sense configuration: Sense both edges
// Bit5 inverted: Off
// Bit5 slew rate limitation: Off
PORTD.PIN5CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit6 Output/Pull configuration: Totempole/No
// Bit6 Input/Sense configuration: Sense both edges
// Bit6 inverted: Off
// Bit6 slew rate limitation: Off
PORTD.PIN6CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit7 Output/Pull configuration: Totempole/No
// Bit7 Input/Sense configuration: Sense both edges
// Bit7 inverted: Off
// Bit7 slew rate limitation: Off
PORTD.PIN7CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Interrupt 0 level: Disabled
// Interrupt 1 level: Disabled
PORTD.INTCTRL=(PORTD.INTCTRL & ~(PORT_INT1LVL_gm | PORT_INT0LVL_gm))
| PORT_INT1LVL_OFF_gc | PORT_INT0LVL_OFF_gc;
// Bit0 pin change interrupt 0: Off
// Bit1 pin change interrupt 0: Off
// Bit2 pin change interrupt 0: Off
// Bit3 pin change interrupt 0: Off
// Bit4 pin change interrupt 0: Off
// Bit5 pin change interrupt 0: Off
// Bit6 pin change interrupt 0: Off
// Bit7 pin change interrupt 0: Off
PORTD.INT0MASK=0x00;
// Bit0 pin change interrupt 1: Off
// Bit1 pin change interrupt 1: Off
// Bit2 pin change interrupt 1: Off
// Bit3 pin change interrupt 1: Off
// Bit4 pin change interrupt 1: Off
// Bit5 pin change interrupt 1: Off
// Bit6 pin change interrupt 1: Off
// Bit7 pin change interrupt 1: Off
PORTD.INT1MASK=0x00;
// PORTE initialization
// OUT register
PORTE.OUT=0xFF;
// Bit0: Input
// Bit1: Input
// Bit2: Input
```

Κώδικας προγράμματος μικροελεγκτή

```
// Bit3: Input
// Bit4: Input
// Bit5: Input
// Bit6: Input
// Bit7: Input
PORTE.DIR=0x00;
// Bit0 Output/Pull configuration: Totempole/No
// Bit0 Input/Sense configuration: Sense both edges
// Bit0 inverted: Off
// Bit0 slew rate limitation: Off
PORTE.PIN0CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit1 Output/Pull configuration: Totempole/No
// Bit1 Input/Sense configuration: Sense both edges
// Bit1 inverted: Off
// Bit1 slew rate limitation: Off
PORTE.PIN1CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit2 Output/Pull configuration: Totempole/No
// Bit2 Input/Sense configuration: Sense both edges
// Bit2 inverted: Off
// Bit2 slew rate limitation: Off
PORTE.PIN2CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit3 Output/Pull configuration: Totempole/No
// Bit3 Input/Sense configuration: Sense both edges
// Bit3 inverted: Off
// Bit3 slew rate limitation: Off
PORTE.PIN3CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit4 Output/Pull configuration: Totempole/No
// Bit4 Input/Sense configuration: Sense both edges
// Bit4 inverted: Off
// Bit4 slew rate limitation: Off
PORTE.PIN4CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit5 Output/Pull configuration: Totempole/No
// Bit5 Input/Sense configuration: Sense both edges
// Bit5 inverted: Off
// Bit5 slew rate limitation: Off
PORTE.PIN5CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit6 Output/Pull configuration: Totempole/No
// Bit6 Input/Sense configuration: Sense both edges
// Bit6 inverted: Off
// Bit6 slew rate limitation: Off
PORTE.PIN6CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit7 Output/Pull configuration: Totempole/No
// Bit7 Input/Sense configuration: Sense both edges
// Bit7 inverted: Off
// Bit7 slew rate limitation: Off
PORTE.PIN7CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Interrupt 0 level: Disabled
// Interrupt 1 level: Disabled
PORTE.INTCTRL=(PORTE.INTCTRL & ~(PORT_INT1LVL_gm | PORT_INT0LVL_gm)))
| PORT_INT1LVL_OFF_gc | PORT_INT0LVL_OFF_gc;
```

Κώδικας προγράμματος μικροελεγκτή

```
// Bit0 pin change interrupt 0: Off
// Bit1 pin change interrupt 0: Off
// Bit2 pin change interrupt 0: Off
// Bit3 pin change interrupt 0: Off
// Bit4 pin change interrupt 0: Off
// Bit5 pin change interrupt 0: Off
// Bit6 pin change interrupt 0: Off
// Bit7 pin change interrupt 0: Off
PORTE.INT0MASK=0x00;
// Bit0 pin change interrupt 1: Off
// Bit1 pin change interrupt 1: Off
// Bit2 pin change interrupt 1: Off
// Bit3 pin change interrupt 1: Off
// Bit4 pin change interrupt 1: Off
// Bit5 pin change interrupt 1: Off
// Bit6 pin change interrupt 1: Off
// Bit7 pin change interrupt 1: Off
PORTE.INT1MASK=0x00;
// PORTF initialization
// OUT register
PORTF.OUT=0xFF;
// Bit0: Input
// Bit1: Input
// Bit2: Input
// Bit3: Output
// Bit4: Input
// Bit5: Input
// Bit6: Input
// Bit7: Input
PORTF.DIR=0x08;
// Bit0 Output/Pull configuration: Totempole/No
// Bit0 Input/Sense configuration: Sense both edges
// Bit0 inverted: Off
// Bit0 slew rate limitation: Off
PORTF.PIN0CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit1 Output/Pull configuration: Totempole/No
// Bit1 Input/Sense configuration: Sense both edges
// Bit1 inverted: Off
// Bit1 slew rate limitation: Off
PORTF.PIN1CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit2 Output/Pull configuration: Totempole/No
// Bit2 Input/Sense configuration: Sense both edges
// Bit2 inverted: Off
// Bit2 slew rate limitation: Off
PORTF.PIN2CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit3 Output/Pull configuration: Totempole/No
// Bit3 Input/Sense configuration: Sense both edges
// Bit3 inverted: Off
// Bit3 slew rate limitation: Off
```

Κώδικας προγράμματος μικροελεγκτή

```
PORTF.PIN3CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit4 Output/Pull configuration: Totempole/No
// Bit4 Input/Sense configuration: Sense both edges
// Bit4 inverted: Off
// Bit4 slew rate limitation: Off
PORTF.PIN4CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit5 Output/Pull configuration: Totempole/No
// Bit5 Input/Sense configuration: Sense both edges
// Bit5 inverted: Off
// Bit5 slew rate limitation: Off
PORTF.PIN5CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit6 Output/Pull configuration: Totempole/No
// Bit6 Input/Sense configuration: Sense both edges
// Bit6 inverted: Off
// Bit6 slew rate limitation: Off
PORTF.PIN6CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit7 Output/Pull configuration: Totempole/No
// Bit7 Input/Sense configuration: Sense both edges
// Bit7 inverted: Off
// Bit7 slew rate limitation: Off
PORTF.PIN7CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Interrupt 0 level: Disabled
// Interrupt 1 level: Disabled
PORTF.INTCTRL=(PORTF.INTCTRL & ~(PORT_INT1LVL_gm | PORT_INT0LVL_gm))
| PORT_INT1LVL_OFF_gc | PORT_INT0LVL_OFF_gc;
// Bit0 pin change interrupt 0: Off
// Bit1 pin change interrupt 0: Off
// Bit2 pin change interrupt 0: Off
// Bit3 pin change interrupt 0: Off
// Bit4 pin change interrupt 0: Off
// Bit5 pin change interrupt 0: Off
// Bit6 pin change interrupt 0: Off
// Bit7 pin change interrupt 0: Off
PORTF.INT0MASK=0x00;
// Bit0 pin change interrupt 1: Off
// Bit1 pin change interrupt 1: Off
// Bit2 pin change interrupt 1: Off
// Bit3 pin change interrupt 1: Off
// Bit4 pin change interrupt 1: Off
// Bit5 pin change interrupt 1: Off
// Bit6 pin change interrupt 1: Off
// Bit7 pin change interrupt 1: Off
PORTF.INT1MASK=0x00;
// PORTR initialization
// OUT register
PORTR.OUT=0x03;
// Bit0: Input
// Bit1: Input
PORTR.DIR=0x00;
// Bit0 Output/Pull configuration: Totempole/No
```

Κώδικας προγράμματος μικροελεγκτή

```
// Bit0 Input/Sense configuration: Sense both edges
// Bit0 inverted: Off
// Bit0 slew rate limitation: Off
PORTR.PIN0CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Bit1 Output/Pull configuration: Totempole/No
// Bit1 Input/Sense configuration: Sense both edges
// Bit1 inverted: Off
// Bit1 slew rate limitation: Off
PORTR.PIN1CTRL=PORT_OPC_TOTEM_gc | PORT_ISC_BOTHEDGES_gc;
// Interrupt 0 level: Disabled
// Interrupt 1 level: Disabled
PORTR.INTCTRL=(PORTR.INTCTRL & ~(PORT_INT1LVL_gm | PORT_INT0LVL_gm))
| PORT_INT1LVL_OFF_gc | PORT_INT0LVL_OFF_gc;
// Bit0 pin change interrupt 0: Off
// Bit1 pin change interrupt 0: Off
PORTR.INT0MASK=0x00;
// Bit0 pin change interrupt 1: Off
// Bit1 pin change interrupt 1: Off
PORTR.INT1MASK=0x00;
}
// Virtual Ports initialization
void vports_init(void)
{
// PORTA mapped to VPORT0
// PORTB mapped to VPORT1
PORTCFG.VPCTRLA=PORTCFG_VP1MAP_PORTB_gc | PORTCFG_VP0MAP_PORTA_gc;
// PORTC mapped to VPORT2
// PORTD mapped to VPORT3
PORTCFG.VPCTRLB=PORTCFG_VP3MAP_PORTD_gc | PORTCFG_VP2MAP_PORTC_gc;
}
// Disable a Timer/Counter type 0
void tc0_disable(TCO_t *ptc)
{
// Timer/Counter off
ptc->CTRLA=(ptc->CTRLA & (~TC0_CLKSEL_gm)) | TC_CLKSEL_OFF_gc;
// Issue a reset command
ptc->CTRLFSET=TC_CMD_RESET_gc;
}
// Timer/Counter TCE0 initialization
void tce0_init(void)
{
unsigned char s;
unsigned char n;
// Note: the correct PORTE direction for the Compare Channels outputs
// is configured in the ports_init function
// Save interrupts enabled/disabled state
s=SREG;
// Disable interrupts
#asm("cli")
// Disable and reset the timer/counter just to be sure
```

Κώδικας προγράμματος μικροελεγκτή

```
tc0_disable(&TCE0);
// Clock source: Peripheral Clock/8
TCE0.CTRLA=(TCE0.CTRLA & (~TC0_CLKSEL_gm)) | TC_CLKSEL_DIV8_gc;
// Mode: Normal Operation, Overflow Int./Event on TOP
// Compare/Capture on channel A: Off
// Compare/Capture on channel B: Off
// Compare/Capture on channel C: Off
// Compare/Capture on channel D: Off
TCE0.CTRLB=(TCE0.CTRLB & (~(TC0_CCAEN_bm | TC0_CCBEN_bm | TC0_CCCEN_bm |
TC0_CCDEN_bm | TC0_WGMODE_gm))) |
    TC_WGMODE_NORMAL_gc;
// Capture event source: None
// Capture event action: None
TCE0.CTRLD=(TCE0.CTRLD & (~(TC0_EVACT_gm | TC0_EVSEL_gm))) |
    TC_EVACT_OFF_gc | TC_EVSEL_OFF_gc;
// Timer/counter in 8bit mode: On
TCE0.CTRLE |= TC0_BYTEM_bm;
// Overflow interrupt: Low Level
// Error interrupt: Disabled
TCE0.INTCTRLA=(TCE0.INTCTRLA & (~(TC0_ERRINTLVL_gm | TC0_OVFINTLVL_gm))) |
    TC_ERRINTLVL_OFF_gc | TC_OVFINTLVL_LO_gc;
// Compare/Capture channel A interrupt: Disabled
// Compare/Capture channel B interrupt: Disabled
// Compare/Capture channel C interrupt: Disabled
// Compare/Capture channel D interrupt: Disabled
TCE0.INTCTRLB=(TCE0.INTCTRLB & (~(TC0_CCDINTLVL_gm | TC0_CCCINTLVL_gm |
TC0_CCBINTLVL_gm | TC0_CCAINTLVL_gm))) |
    TC_CCDINTLVL_OFF_gc | TC_CCCINTLVL_OFF_gc | TC_CCBINTLVL_OFF_gc |
    TC_CCAINTLVL_OFF_gc;
// High resolution extension: Off
HIRESE.CTRL&= ~HIRES_HREN0_bm;
// Advanced Waveform Extension initialization
// Optimize for speed
#pragma optsize-
// Disable locking the AWEX configuration registers just to be sure
n=MCU.AWEXLOCK & (~MCU_AWEXELOCK_bm);
CCP=CCP_IOREG_gc;
MCU.AWEXLOCK=n;
// Restore optimization for size if needed
#pragma optsize_default
// Pattern generation: Off
// Dead time insertion: Off
AWEXE.CTRL&= ~(AWEX_PGM_bm | AWEX_CWCM_bm | AWEX_DTICCDEN_bm |
AWEX_DTICCCEN_bm | AWEX_DTICCBEN_bm | AWEX_DTICCAEN_bm);
// Fault protection initialization
// Fault detection on OCD Break detection: On
// Fault detection restart mode: Latched Mode
// Fault detection action: None (Fault protection disabled)
AWEXE.FDCTRL=(AWEXE.FDCTRL & (~(AWEX_FDDBD_bm | AWEX_FDMODE_bm |
AWEX_FDACT_gm))) | AWEX_FDACT_NONE_gc;
// Fault detect events:
```

Κώδικας προγράμματος μικροελεγκτή

```
// Event channel 0: Off
// Event channel 1: Off
// Event channel 2: Off
// Event channel 3: Off
// Event channel 4: Off
// Event channel 5: Off
// Event channel 6: Off
// Event channel 7: Off
AWEXE.FDEVMASK=0b00000000;
// Make sure the fault detect flag is cleared
AWEXE.STATUS |= AWEXE.STATUS & AWEX_FDF_bm;
// Clear the interrupt flags
TCE0.INTFLAGS=TCE0.INTFLAGS;
// Set counter register
TCE0.CNT=0x0000;
// Set period register
TCE0.PER=0x00F9;
// Set channel A Compare/Capture register
TCE0.CCA=0x0000;
// Set channel B Compare/Capture register
TCE0.CCB=0x0000;
// Set channel C Compare/Capture register
TCE0.CCC=0x0000;
// Set channel D Compare/Capture register
TCE0.CCD=0x0000;

// Restore interrupts enabled/disabled state
SREG=s;
}
// Timer/counter TCE0 Overflow/Underflow interrupt service routine
interrupt [TCE0_OVF_vect] void tce0_overflow_isr(void)
{
// write your code here
counter++;
}
// USARTF0 initialization
void usartf0_init(void)
{
// Note: the correct PORTF direction for the RxD, TxD and XCK signals
// is configured in the ports_init function
// Transmitter is enabled
// Set TxD=1
PORTF.OUTSET=0x08;
// Communication mode: Asynchronous USART
// Data bits: 8
// Stop bits: 1
// Parity: Disabled
USARTF0.CTRLA=USART_CMODE_ASYNCHRONOUS_gc | USART_PMODE_DISABLED_gc |
USART_CHSIZE_8BIT_gc;
// Receive complete interrupt: Disabled
```

Κώδικας προγράμματος μικροελεγκτή

```
// Transmit complete interrupt: Disabled
// Data register empty interrupt: Disabled
USARTF0.CTRLA=(USARTF0.CTRLA & ~(USART_RXCINTLVL_gm | USART_TXCINTLVL_gm |
USART_DREINTLVL_gm))) |
    USART_RXCINTLVL_OFF_gc | USART_TXCINTLVL_OFF_gc | USART_DREINTLVL_OFF_gc;
// Required Baud rate: 115200
// Real Baud Rate: 115107,9 (x1 Mode), Error: 0,1 %
USARTF0.BAUDCTRLA=0x0B;
USARTF0.BAUDCTRLB=((0x09 << USART_BSCALE_bp) & USART_BSCALE_gm) | 0x00;
// Receiver: On
// Transmitter: On
// Double transmission speed mode: Off
// Multi-processor communication mode: Off
USARTF0.CTRLB=(USARTF0.CTRLB & ~(USART_RXEN_bm | USART_TXEN_bm | USART_CLK2X_bm |
USART_MPCM_bm | USART_TXB8_bm))) |
    USART_RXEN_bm | USART_TXEN_bm;
}
// Receive a character from USARTF0
// USARTF0 is used as the default input device by the 'getchar' function
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
unsigned char status;
while (1)
{
while (((status=USARTF0.STATUS) & USART_RXCIF_bm) == 0);
data=USARTF0.DATA;
if ((status & (USART_FERR_bm | USART_PERR_bm | USART_BUFOVF_bm)) == 0) return data;
}
}
#pragma used-
// Write a character to the USARTF0 Transmitter
// USARTF0 is used as the default output device by the 'putchar' function
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while ((USARTF0.STATUS & USART_DREIF_bm) == 0);
USARTF0.DATA=c;
}
#pragma used-
// Function used to read the calibration byte from the
// signature row, specified by 'index'
#pragma optsize-
unsigned char read_calibration_byte(unsigned char index)
{
unsigned char r;
NVM.CMD=NVM_CMD_READ_CALIB_ROW_gc;
```


Κώδικας προγράμματος μικροελεγκτή

```
r*((flash unsigned char*) index);
// Clean up NVM command register
NVM.CMD=NVM_CMD_NO_OPERATION_gc;
return r;
}
#pragma optsize_default
// ADCA initialization
// Variable used to store the ADC offset
// for 12 Bit Signed conversion mode
signed char adca_offset;
void adca_init(void)
{
unsigned char i;
int offs;
// ADCA is enabled
// Resolution: 12 Bits
// Load the calibration value for 12 Bit resolution
// from the signature row
ADCA.CALL=read_calibration_byte(PROD_SIGNATURES_START+ADCACAL0_offset);
ADCA.CALH=read_calibration_byte(PROD_SIGNATURES_START+ADCACAL1_offset);
// Free Running mode: Off
// Conversion mode: Signed
ADCA.CTRLB=(ADCA.CTRLB & ~(ADC_CONMODE_bm | ADC_FREERUN_bm |
ADC_RESOLUTION_gm)) |
ADC_CONMODE_bm | ADC_RESOLUTION_12BIT_gc;
// Clock frequency: 3,906 kHz
ADCA.PRESCALER=(ADCA.PRESCALER & (~ADC_PRESCALER_gm)) |
ADC_PRESCALER_DIV512_gc;
// Reference: Internal 1.00 V
// Temperature reference: Off
ADCA.REFCTRL=(ADCA.REFCTRL & ((~(ADC_REFSEL_gm | ADC_TEMPREF_bm)) |
ADC_BANDGAP_bm)) | ADC_REFSEL_INT1V_gc | ADC_BANDGAP_bm;
// Read and save the ADC offset using channel 0
ADCA.CH0.CTRL=(ADCA.CH0.CTRL & ~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm)) |
ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
ADCA.CH0.MUXCTRL=(ADCA.CH0.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm)) | ADC_CH_MUXPOS_PIN0_gc | ADC_CH_MUXNEG_PIN0_gc;
// Enable the ADC in order to read the offset
ADCA.CTRLA |= ADC_ENABLE_bm;
// Insert a delay to allow the ADC common mode voltage to stabilize
delay_us(2);
// Perform several offset measurements and store the mean value
offs=0;
for (i=0; i<16; i++)
{
// Start the AD conversion on channel 0
ADCA.CH0.CTRL |= ADC_CH_START_bm;
// Wait for the AD conversion to complete
while ((ADCA.CH0.INTFLAGS & ADC_CH_CHIF_bm)==0);
// Clear the interrupt flag
```

Κώδικας προγράμματος μικροελεγκτή

```
ADCA.CH0.INTFLAGS=ADC_CH_CHIF_bm;
// Read the offset
offs+=(signed char) ADCA.CH0.RESL;
}
// Disable the ADC
ADCA.CTRLA&= ~ADC_ENABLE_bm;
// Store the mean value of the offset
adca_offset=(signed char) (offs/16);
// Initialize the ADC Compare register
ADCA.CMPL=0x00;
ADCA.CMPH=0x00;
// ADC channel 0 gain: 1
// ADC channel 0 input mode: Differential input signal
ADCA.CH0.CTRL=(ADCA.CH0.CTRL & ~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm)) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
// ADC channel 0 positive input: ADC4 pin
// ADC channel 0 negative input: ADC0 pin
ADCA.CH0.MUXCTRL=(ADCA.CH0.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm)) |
    ADC_CH_MUXPOS_PIN4_gc | ADC_CH_MUXNEG_PIN0_gc;
// ADC channel 1 gain: 1
// ADC channel 1 input mode: Differential input signal
ADCA.CH1.CTRL=(ADCA.CH1.CTRL & ~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm)) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
// ADC channel 1 positive input: ADC5 pin
// ADC channel 1 negative input: ADC1 pin
ADCA.CH1.MUXCTRL=(ADCA.CH1.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm)) |
    ADC_CH_MUXPOS_PIN5_gc | ADC_CH_MUXNEG_PIN1_gc;
// ADC channel 2 gain: 1
// ADC channel 2 input mode: Differential input signal
ADCA.CH2.CTRL=(ADCA.CH2.CTRL & ~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm)) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
// ADC channel 2 positive input: ADC6 pin
// ADC channel 2 negative input: ADC2 pin
ADCA.CH2.MUXCTRL=(ADCA.CH2.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm)) |
    ADC_CH_MUXPOS_PIN6_gc | ADC_CH_MUXNEG_PIN2_gc;
// ADC channel 3 gain: 1
// ADC channel 3 input mode: Differential input signal
ADCA.CH3.CTRL=(ADCA.CH3.CTRL & ~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm)) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
// ADC channel 3 positive input: ADC7 pin
// ADC channel 3 negative input: ADC3 pin
ADCA.CH3.MUXCTRL=(ADCA.CH3.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm)) |
    ADC_CH_MUXPOS_PIN7_gc | ADC_CH_MUXNEG_PIN3_gc;
```

Κώδικας προγράμματος μικροελεγκτή

```
// AD conversion is started by software
ADCA.EVCTRL=ADC_EVACT_NONE_gc;
// Channel 0 interrupt: Disabled
ADCA.CH0.INTCTRL=(ADCA.CH0.INTCTRL & ~(ADC_CH_INTMODE_gm | ADC_CH_INTLVL_gm)))
|
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Channel 1 interrupt: Disabled
ADCA.CH1.INTCTRL=(ADCA.CH1.INTCTRL & ~(ADC_CH_INTMODE_gm | ADC_CH_INTLVL_gm)))
|
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Channel 2 interrupt: Disabled
ADCA.CH2.INTCTRL=(ADCA.CH2.INTCTRL & ~(ADC_CH_INTMODE_gm | ADC_CH_INTLVL_gm)))
|
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Channel 3 interrupt: Disabled
ADCA.CH3.INTCTRL=(ADCA.CH3.INTCTRL & ~(ADC_CH_INTMODE_gm | ADC_CH_INTLVL_gm)))
|
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Enable the ADC
ADCA.CTRLA |= ADC_ENABLE_bm;
// Insert a delay to allow the ADC common mode voltage to stabilize
delay_us(2);
}
// ADCA channel data read function using polled mode
signed int adca_read(unsigned char channel)
{
    ADC_CH_t *pch=&ADCA.CH0+channel;
    signed int data;
    // Start the AD conversion
    pch->CTRL |= ADC_CH_START_bm;
    // Wait for the AD conversion to complete
    while ((pch->INTFLAGS & ADC_CH_CHIF_bm)==0);
    // Clear the interrupt flag
    pch->INTFLAGS=ADC_CH_CHIF_bm;
    // Read the AD conversion result
    ((unsigned char *) &data)[0]=pch->RESL;
    ((unsigned char *) &data)[1]=pch->RESH;
    // Compensate the ADC offset
    data-=adca_offset;
    return data;
}
// ADCA swept channel(s) data read function
// for software triggered mode
void adca_sweep_read(unsigned char nch, signed int *pdata)
{
    ADC_CH_t *pch=&ADCA.CH0;
    unsigned char i,j,m;
    // Sweep starts with channel 0
    j=ADC_CH0START_bm;
    // Prepare the AD conversion start mask for the swept channel(s)
```

Κώδικας προγράμματος μικροελεγκτή

```
m=0;
i=0;
do
{
  m|=j;
  j<<=1;
}
while (++i<nch);
// Ensure the interrupt flags are cleared
ADCA.INTFLAGS=ADCA.INTFLAGS;
// Start the AD conversion for the swept channel(s)
ADCA.CTRLA=(ADCA.CTRLA & (ADC_DMASEL_gm | ADC_FLUSH_bm | ADC_ENABLE_bm)) | m;
// Read and store the AD conversion results for all the swept channels
for (i=0; i<nch; i++)
{
  // Wait for the AD conversion to complete
  while ((pch->INTFLAGS & ADC_CH_CHIF_bm)==0);
  // Clear the interrupt flag
  pch->INTFLAGS=ADC_CH_CHIF_bm;
  // Read the AD conversion result
  ((unsigned char *)pdata)[0]=pch->RESL;
  ((unsigned char *)pdata)[1]=pch->RESH;
  // Compensate the ADC offset
  *pdata-=adca_offset;
  pdata++;
  pch++;
}
}
// ADCB initialization
// Variable used to store the ADC offset
// for 12 Bit Signed conversion mode
signed char adcb_offset;
void adcb_init(void)
{
  unsigned char i;
  int offs;
  // ADCB is enabled
  // Resolution: 12 Bits
  // Load the calibration value for 12 Bit resolution
  // from the signature row
  ADCB.CALL=read_calibration_byte(PROD_SIGNATURES_START+ADCB_CAL0_offset);
  ADCB.CALH=read_calibration_byte(PROD_SIGNATURES_START+ADCB_CAL1_offset);
  // Free Running mode: Off
  // Conversion mode: Signed
  ADCB.CTRLB=(ADCB.CTRLB & ~(ADC_CONMODE_bm | ADC_FREERUN_bm |
  ADC_RESOLUTION_gm))) |
  ADC_CONMODE_bm | ADC_RESOLUTION_12BIT_gc;
  // Clock frequency: 3,906 kHz
  ADCB.PRESCALER=(ADCB.PRESCALER & (~ADC_PRESCALER_gm)) |
  ADC_PRESCALER_DIV512_gc;
```

Κώδικας προγράμματος μικροελεγκτή

```
// Reference: Internal 1.00 V
// Temperature reference: Off
ADCB.REFCTRL=(ADCB.REFCTRL & ((~(ADC_REFSEL_gm | ADC_TEMPREF_bm)) |
ADC_BANDGAP_bm)) |
    ADC_REFSEL_INT1V_gc | ADC_BANDGAP_bm;
// Read and save the ADC offset using channel 0
ADCB.CH0.CTRL=(ADCB.CH0.CTRL & (~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm))) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
ADCB.CH0.MUXCTRL=(ADCB.CH0.MUXCTRL & (~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm))) |
    ADC_CH_MUXPOS_PIN0_gc | ADC_CH_MUXNEG_PIN0_gc;
// Enable the ADC in order to read the offset
ADCB.CTRLA |= ADC_ENABLE_bm;
// Insert a delay to allow the ADC common mode voltage to stabilize
delay_us(2);
// Perform several offset measurements and store the mean value
offs=0;
for (i=0; i<16; i++)
{
    // Start the AD conversion on channel 0
    ADCB.CH0.CTRL |= ADC_CH_START_bm;
    // Wait for the AD conversion to complete
    while ((ADCB.CH0.INTFLAGS & ADC_CH_CHIF_bm)==0);
    // Clear the interrupt flag
    ADCB.CH0.INTFLAGS=ADC_CH_CHIF_bm;
    // Read the offset
    offs+=(signed char) ADCB.CH0.RESL;
}
// Disable the ADC
ADCB.CTRLA&= ~ADC_ENABLE_bm;
// Store the mean value of the offset
adcb_offset=(signed char) (offs/16);
// Initialize the ADC Compare register
ADCB.CMPL=0x00;
ADCB.CMPH=0x00;
// ADC channel 0 gain: 1
// ADC channel 0 input mode: Differential input signal
ADCB.CH0.CTRL=(ADCB.CH0.CTRL & (~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm))) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
// ADC channel 0 positive input: ADC4 pin
// ADC channel 0 negative input: ADC0 pin
ADCB.CH0.MUXCTRL=(ADCB.CH0.MUXCTRL & (~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm))) |
    ADC_CH_MUXPOS_PIN4_gc | ADC_CH_MUXNEG_PIN0_gc;
// ADC channel 1 gain: 1
// ADC channel 1 input mode: Differential input signal
ADCB.CH1.CTRL=(ADCB.CH1.CTRL & (~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm))) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
```

Κώδικας προγράμματος μικροελεγκτή

```
// ADC channel 1 positive input: ADC5 pin
// ADC channel 1 negative input: ADC1 pin
ADCB.CH1.MUXCTRL=(ADCB.CH1.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm)) |
    ADC_CH_MUXPOS_PIN5_gc | ADC_CH_MUXNEG_PIN1_gc;
// ADC channel 2 gain: 1
// ADC channel 2 input mode: Differential input signal
ADCB.CH2.CTRL=(ADCB.CH2.CTRL & ~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm)) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
// ADC channel 2 positive input: ADC6 pin
// ADC channel 2 negative input: ADC2 pin
ADCB.CH2.MUXCTRL=(ADCB.CH2.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm)) |
    ADC_CH_MUXPOS_PIN6_gc | ADC_CH_MUXNEG_PIN2_gc;
// ADC channel 3 gain: 1
// ADC channel 3 input mode: Differential input signal
ADCB.CH3.CTRL=(ADCB.CH3.CTRL & ~(ADC_CH_START_bm | ADC_CH_GAINFAC_gm |
ADC_CH_INPUTMODE_gm)) |
    ADC_CH_GAIN_1X_gc | ADC_CH_INPUTMODE_DIFF_gc;
// ADC channel 3 positive input: ADC7 pin
// ADC channel 3 negative input: ADC3 pin
ADCB.CH3.MUXCTRL=(ADCB.CH3.MUXCTRL & ~(ADC_CH_MUXPOS_gm |
ADC_CH_MUXNEG_gm)) |
    ADC_CH_MUXPOS_PIN7_gc | ADC_CH_MUXNEG_PIN3_gc;
// AD conversion is started by software
ADCB.EVCTRL=ADC_EVACT_NONE_gc;
// Channel 0 interrupt: Disabled
ADCB.CH0.INTCTRL=(ADCB.CH0.INTCTRL & ~(ADC_CH_INTMODE_gm | ADC_CH_INTLVL_gm))
|
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Channel 1 interrupt: Disabled
ADCB.CH1.INTCTRL=(ADCB.CH1.INTCTRL & ~(ADC_CH_INTMODE_gm | ADC_CH_INTLVL_gm))
|
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Channel 2 interrupt: Disabled
ADCB.CH2.INTCTRL=(ADCB.CH2.INTCTRL & ~(ADC_CH_INTMODE_gm | ADC_CH_INTLVL_gm))
|
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Channel 3 interrupt: Disabled
ADCB.CH3.INTCTRL=(ADCB.CH3.INTCTRL & ~(ADC_CH_INTMODE_gm | ADC_CH_INTLVL_gm))
|
    ADC_CH_INTMODE_COMPLETE_gc | ADC_CH_INTLVL_OFF_gc;
// Enable the ADC
ADCB.CTRLA |= ADC_ENABLE_bm;
// Insert a delay to allow the ADC common mode voltage to stabilize
delay_us(2);
}
// ADCB channel data read function using polled mode
signed int adcb_read(unsigned char channel)
{
```

Κώδικας προγράμματος μικροελεγκτή

```
ADC_CH_t *pch=&ADCB.CH0+channel;
signed int data;
// Start the AD conversion
pch->CTRL|=ADC_CH_START_bm;
// Wait for the AD conversion to complete
while ((pch->INTFLAGS & ADC_CH_CHIF_bm)==0);
// Clear the interrupt flag
pch->INTFLAGS=ADC_CH_CHIF_bm;
// Read the AD conversion result
((unsigned char *) &data)[0]=pch->RESL;
((unsigned char *) &data)[1]=pch->RESH;
// Compensate the ADC offset
data-=adcb_offset;
return data;
}
// ADCB swept channel(s) data read function
// for software triggered mode
void adcb_sweep_read(unsigned char nch, signed int *pdata)
{
ADC_CH_t *pch=&ADCB.CH0;
unsigned char i,j,m;
// Sweep starts with channel 0
j=ADC_CH0START_bm;
// Prepare the AD conversion start mask for the swept channel(s)
m=0;
i=0;
do
{
m|=j;
j<<=1;
}
while (++i<nch);
// Ensure the interrupt flags are cleared
ADCB.INTFLAGS=ADCB.INTFLAGS;
// Start the AD conversion for the swept channel(s)
ADCB.CTRLA=(ADCB.CTRLA & (ADC_DMASEL_gm | ADC_FLUSH_bm | ADC_ENABLE_bm)) | m;
// Read and store the AD conversion results for all the swept channels
for (i=0; i<nch; i++)
{
// Wait for the AD conversion to complete
while ((pch->INTFLAGS & ADC_CH_CHIF_bm)==0);
// Clear the interrupt flag
pch->INTFLAGS=ADC_CH_CHIF_bm;
// Read the AD conversion result
((unsigned char *) pdata)[0]=pch->RESL;
((unsigned char *) pdata)[1]=pch->RESH;
// Compensate the ADC offset
*pdata-=adcb_offset;
pdata++;
}
```

Κώδικας προγράμματος μικροελεγκτή

```
pch++;
}
}
// Fit ADC data using linear least squares method
signed int LS_fit(signed int x)
{
if ((x>512) || (x<-512)) return (float)x*LS_SLOPE_B+LS_YP_B;
else return (float)x*LS_SLOPE_S+LS_YP_S;
}
// Find median of 3 numbers a,b,c
signed int median(signed int a, signed int b, signed int c)
{
signed int med;

if ((a <= b) && (a <= c))
{
med = (b <= c) ? b : c;
}
else if ((b <= a) && (b <= c))
{
med = (a <= c) ? a : c;
}
else
{
med = (a <= b) ? a : b;
}
return med;
}
void main(void)
{
// Declare your local variables here
unsigned char n;
// Interrupt system initialization
// Optimize for speed
#pragma optsize-
// Make sure the interrupts are disabled
#pragma asm("cli")
// Low level interrupt: On
// Round-robin scheduling for low level interrupt: Off
// Medium level interrupt: Off
// High level interrupt: Off
// The interrupt vectors will be placed at the start of the Application FLASH section
n=(PMIC.CTRL & ~(PMIC_RREN_bm | PMIC_IVSEL_bm | PMIC_HILVLEN_bm |
PMIC_MEDLVLEN_bm | PMIC_LOLVLEN_bm)) | PMIC_LOLVLEN_bm;
CCP=CCP_IJOREG_gc;
PMIC.CTRL=n;
// Set the default priority for round-robin scheduling
PMIC.INTPRI=0x00;
// Restore optimization for size if needed
#pragma optsize_default
```


Κώδικας προγράμματος μικροελεγκτή

```
// System clocks initialization
system_clocks_init();
// Ports initialization
ports_init();
// Virtual Ports initialization
vports_init();
// Timer/Counter TCE0 initialization
tce0_init();
// USARTF0 initialization
usartf0_init();
// ADCA initialization
adca_init();
// ADCB initialization
adcb_init();
// Alphanumeric LCD initialization
// Connections specified in the
// Project | Configure | C Compiler | Libraries | Alphanumeric LCD menu:
// RS - PORTC Bit 4
// RD - PORTC Bit 3
// EN - PORTC Bit 5
// D4 - PORTC Bit 6
// D5 - PORTC Bit 1
// D6 - PORTC Bit 7
// D7 - PORTC Bit 0
// Characters/line: 20
lcd_init(20);
#asm("cli")
//Intro splash screens
lcd_clear();
lcd_putsf("SK Embio Diagnostics");
lcd_gotoxy(0,1);
lcd_putsf(" BERA Biosensor");
lcd_gotoxy(0,2);
lcd_putsf(" Developed by:");
lcd_gotoxy(0,3);
lcd_putsf("microSENSES Lab TEIA");
delay_ms(4*INTRO_DELAY);
lcd_clear();
lcd_putsf(" Ready");
lcd_gotoxy(0,3);
lcd_putsf("Rev. F, for dropsens");
delay_ms(INTRO_DELAY);
// Globally enable interrupts
#asm("sei")
sleep_enable(); // allow entering sleep modes
idle(); // enter sleep mode
while (1)
{
    if ((counter>=DELAY_CNT) && (!VPORT3.IN&1)) //VPORT3.0 -> PORTD.0, run ctrl switch
```

Κώδικας προγράμματος μικροελεγκτή

```
{
sleep_disable(); // prevent entering sleep modes
counter=0;
t_count++;
for (k=0;k<MEAN_LENGTH;k++) //averaging loop
{
for (i=0;i<=2;i++) //median loop
{
for (j=0;j<=3;j++) //ch scan loop
{
data[j][i]=adca_read(j); //ch 0-3
data[j+4][i]=adcb_read(j); //ch 4-7
} //ch scan loop
} //median loop
for (j=0;j<=7;j++) //find median for each ch and add
{
averaging_sum[j] += LS_fit(median(data[j][0],data[j][1],data[j][2]));
}
} //averaging loop
// print to uart & zero averaging_sum array
for (j=0;j<=7;j++) //Ch scan loop
{
ftoa(((float) averaging_sum[j]*1000/(MEAN_LENGTH*2048.)),1,str);
printf("ch%d;%d;%s\n",j+1,t_count,str);
t_sum[j] += averaging_sum[j];
averaging_sum[j]=0;
} //Ch scan loop
// update LCD with accumulative average
if (t_count%10==0) //update LCD
{
lcd_clear();
for (j=0;j<=3;j++) //ch scan loop
{
itoa(((float)t_sum[j]*1000/(t_count*MEAN_LENGTH*2048.)),str); //ch 0-3
sprintf(temp,"%d:%smV", (j+1),str);
lcd_gotoxy(0,j);
lcd_puts(temp);
itoa(((float)t_sum[j+4]*1000/(t_count*MEAN_LENGTH*2048.)),str); //ch 4-7
sprintf(temp,"%d:%smV", (j+5),str);
lcd_gotoxy(10,j);
lcd_puts(temp);
} //ch scan loop
} //update LCD
} //if (run control switch)
sleep_enable(); // allow entering sleep modes
idle(); //enter sleep mode
} //while(1)
} //main
```

Σ. Προγραμμα υπολογισμού παραμετρων E.T. σε MATLAB

```
clc
clear
data=xlsread('\ADC_measurements.xlsx');
%Y=round(data(:,1)*2047/1000); bin
%X=round(data(:,2)*2047/1000); bin

Y=data(:,1);
X=data(:,2);
X(:,end+1)=1;

A=inv(X'*X)*X'*Y;

Y_proj=round(X*A);

mare_percent=100*sum(abs(Y-Y_proj)./Y)/length(Y);
y_mean=sum(Y)/length(Y);
SSE=sum((Y-Y_proj).^2);
SST=sum((Y-y_mean).^2);

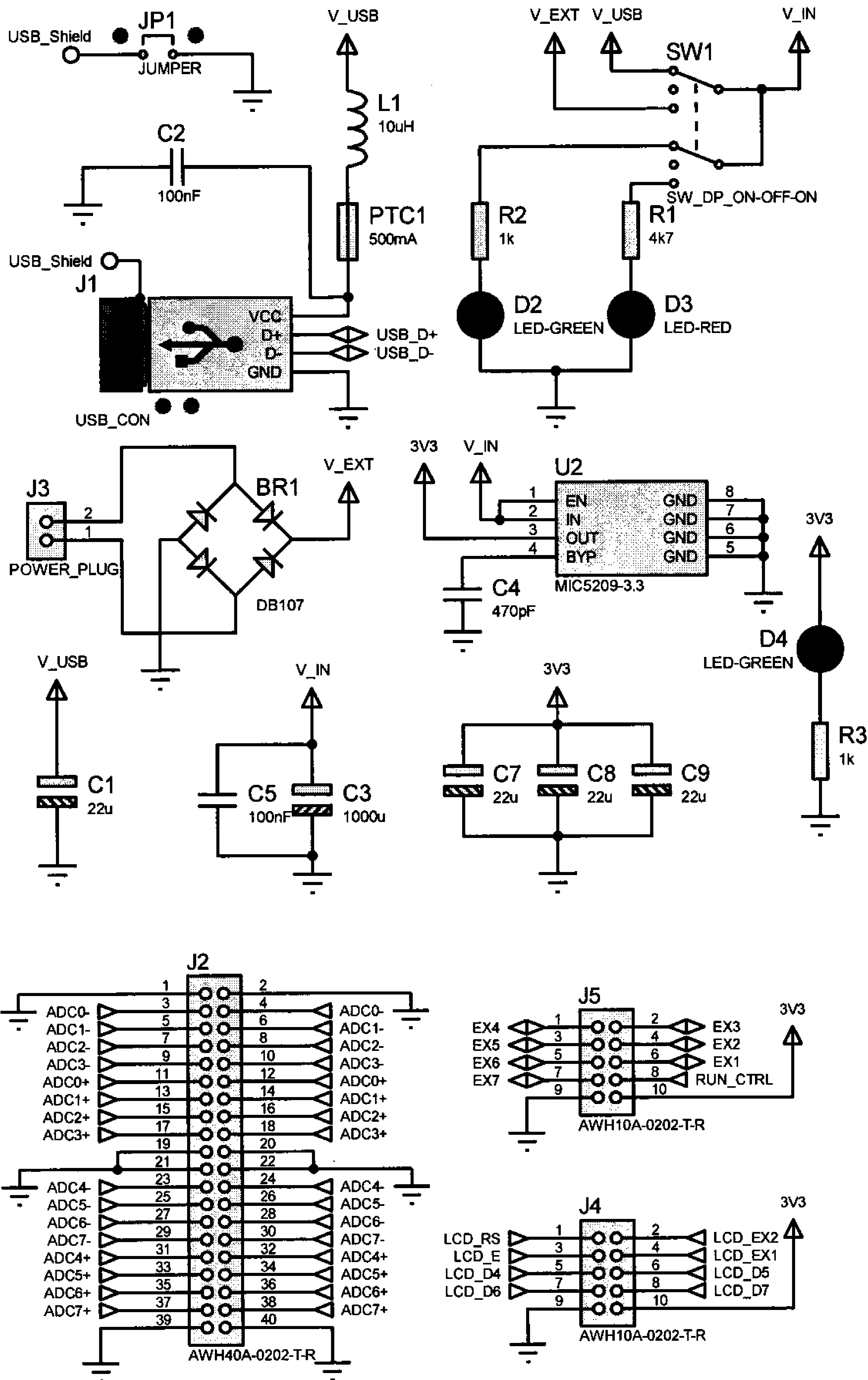
R_squared=1-(SSE/SST);

plot(X(:,1),Y,'+r')
hold
plot(X(:,1),Y_proj,'b')

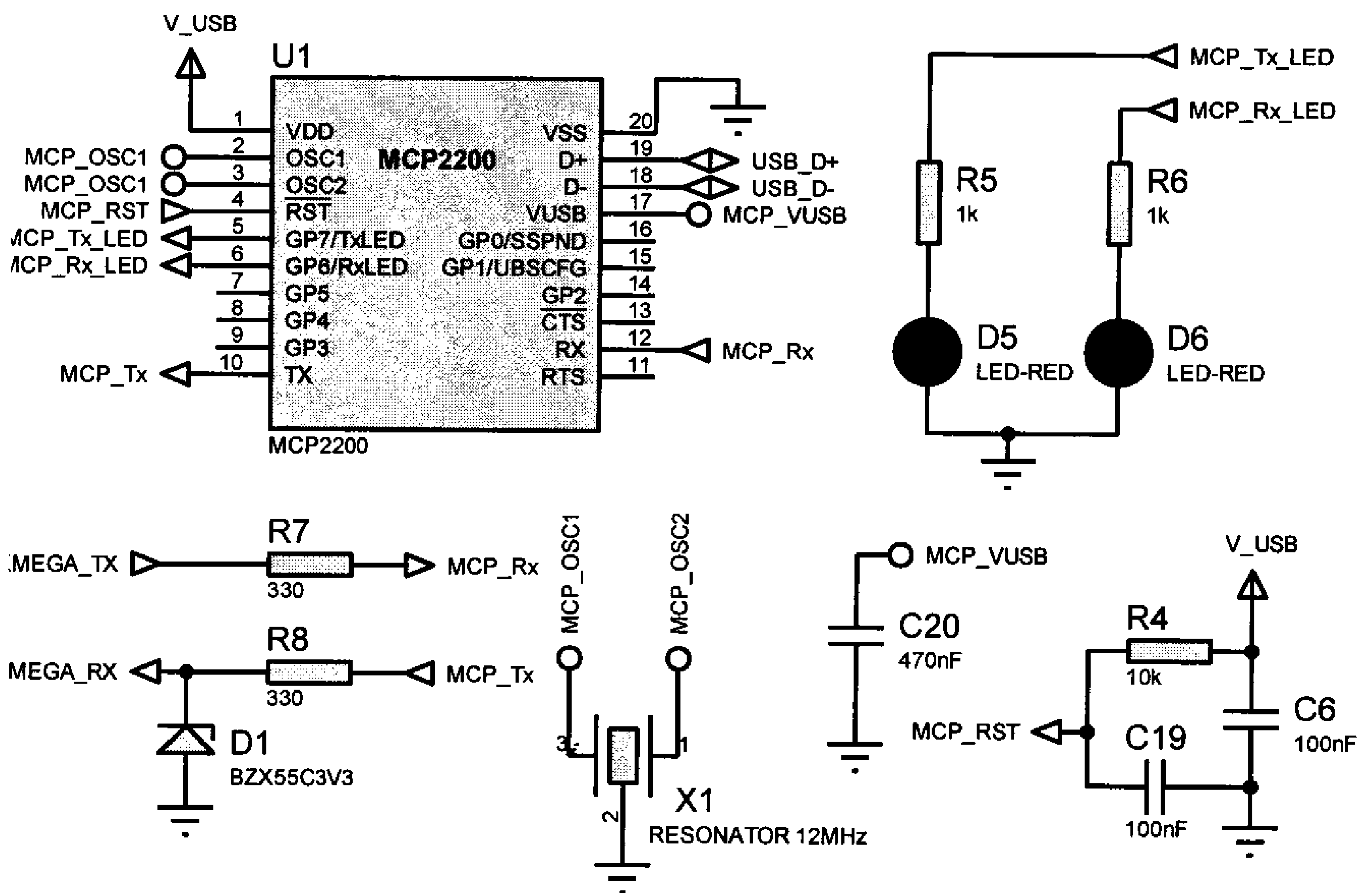
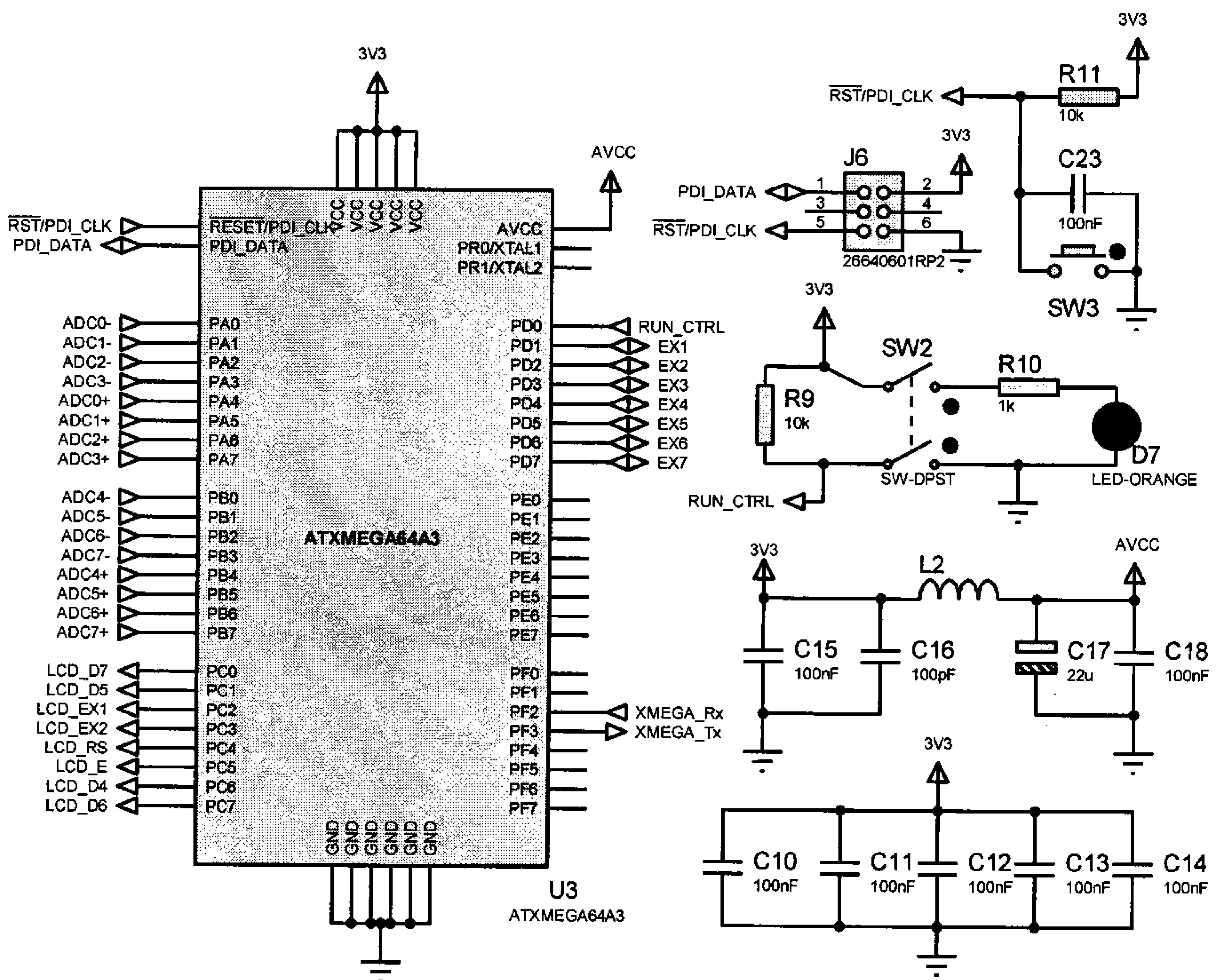
legend('measurements',' LS line');
xlabel('index #','FontSize',12)
```

D. Σχηματικά κυκλωμάτων και σχέδια τυπωμένων πλακετών

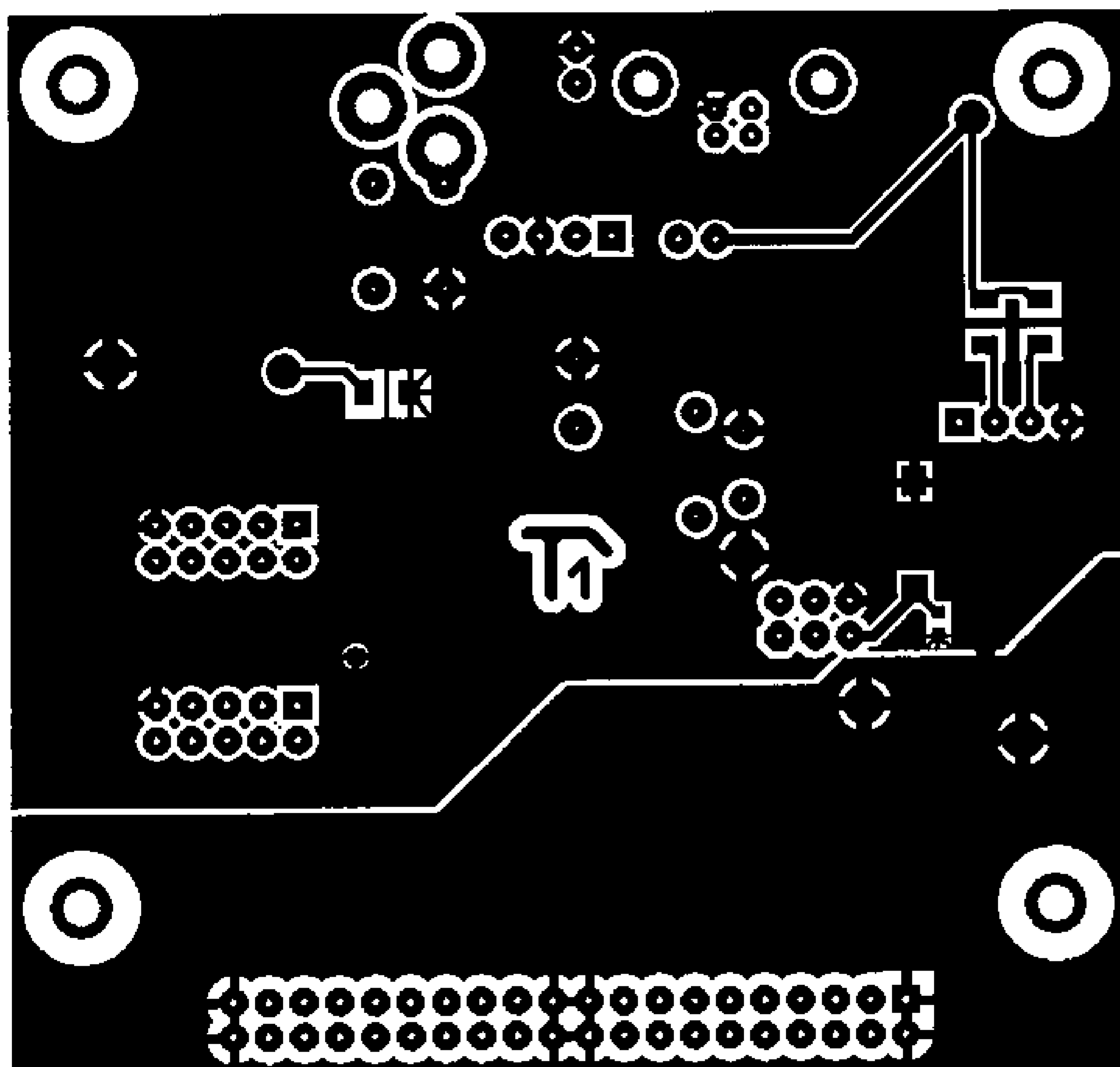
Σχηματικό κύριας πλακέτας:



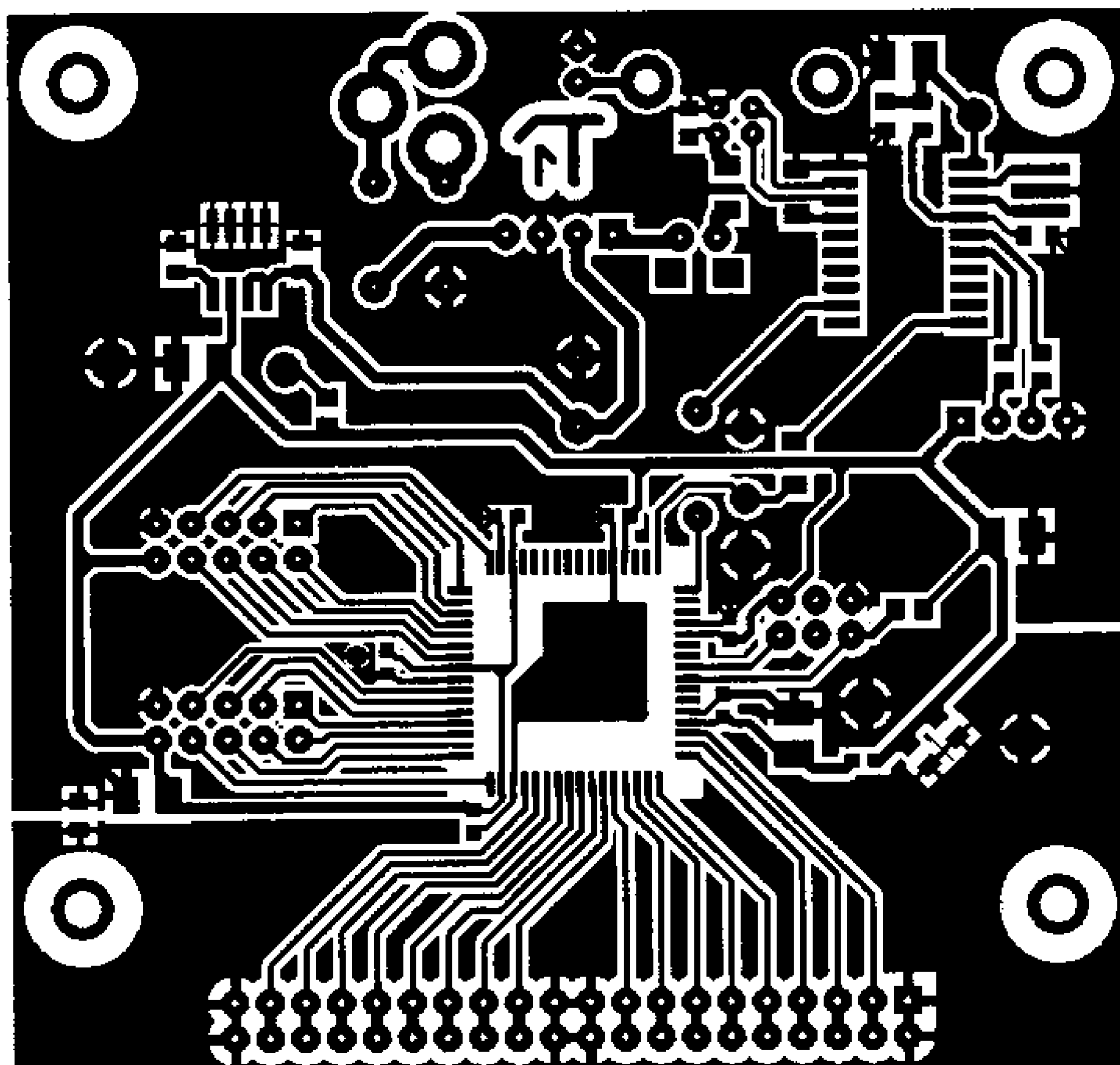
Σχηματικά κυκλωμάτων και σχέδια τυπωμένων πλακετών



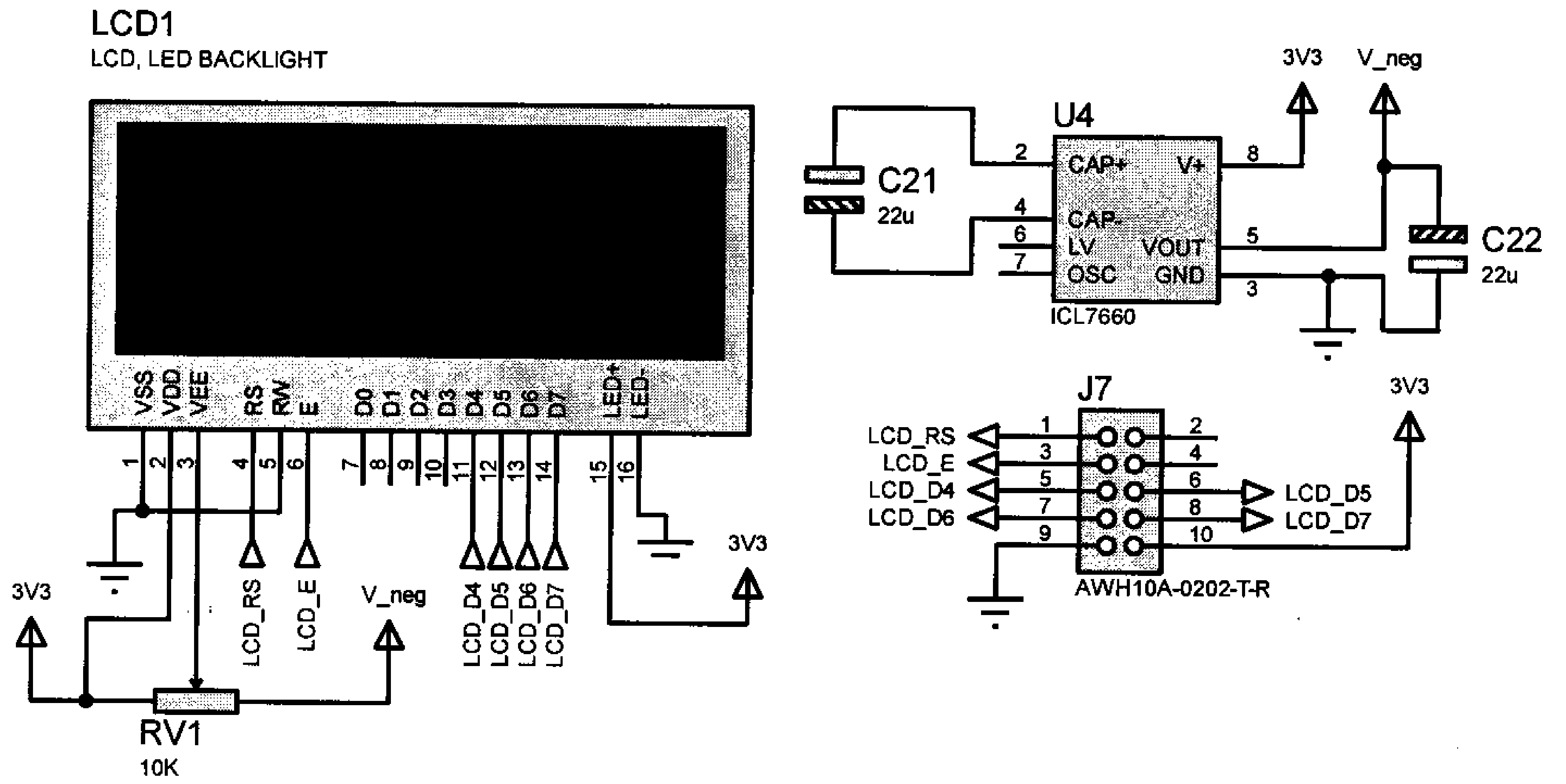
PCB layout κύριας πλακέτας (άνω όψη), 1:1 μέγεθος:



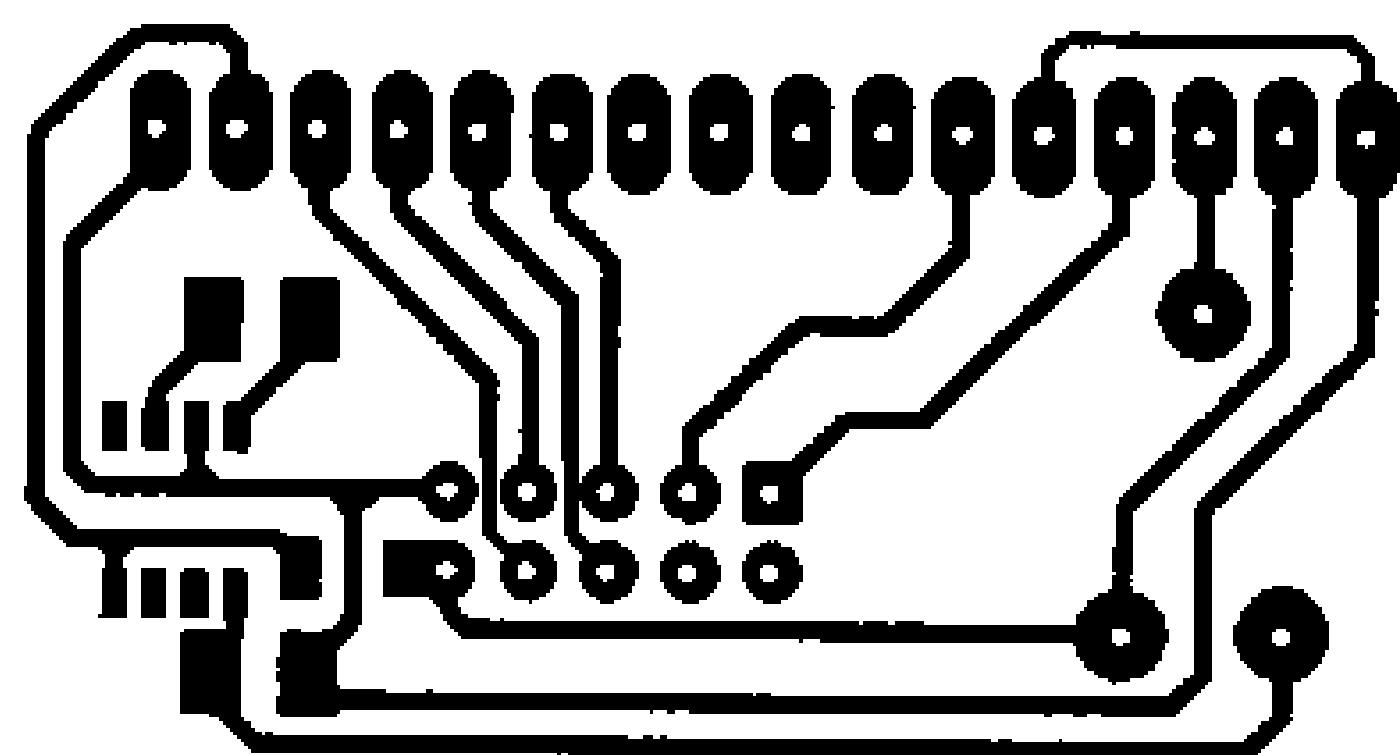
PCB layout κύριας πλακέτας (κάτω όψη), 1:1 μέγεθος:



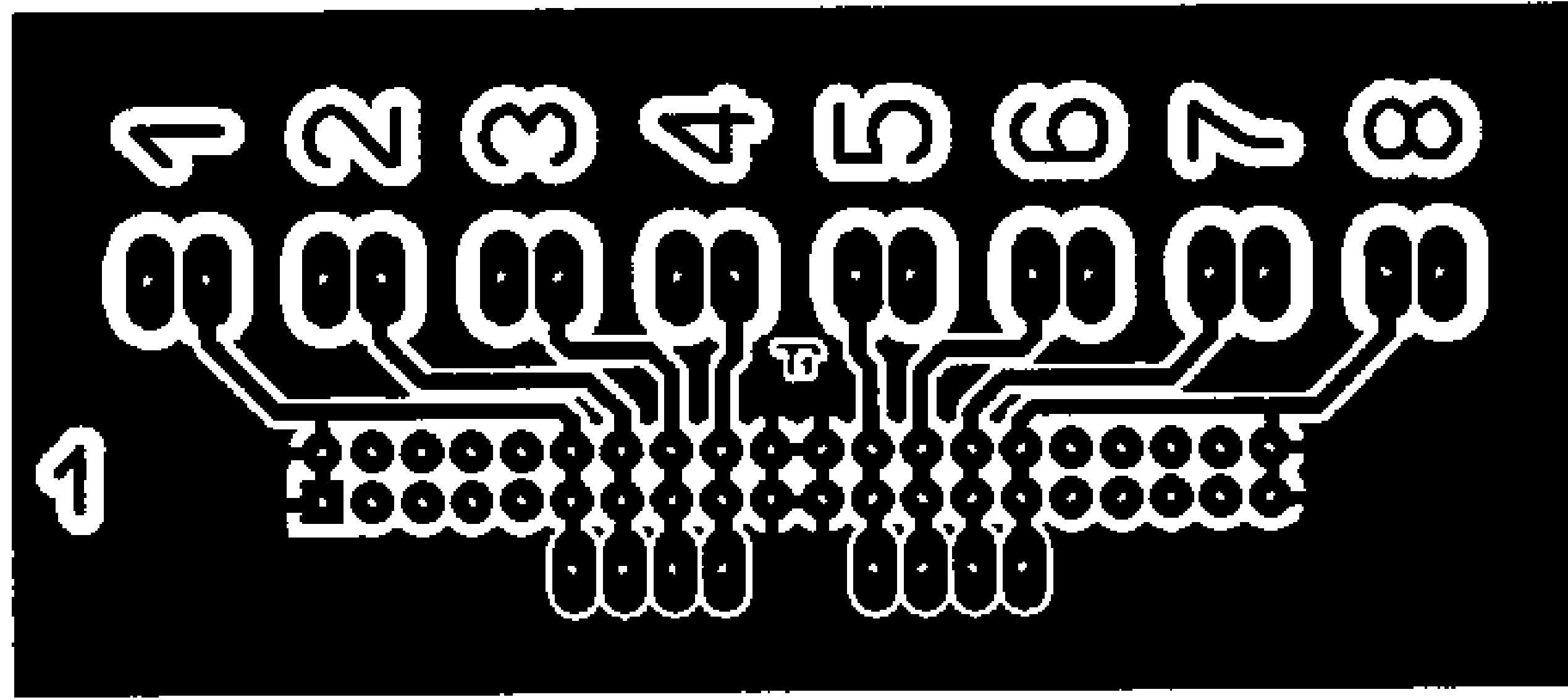
Σχηματικό πλακέτας διασύνδεσης με LCD:



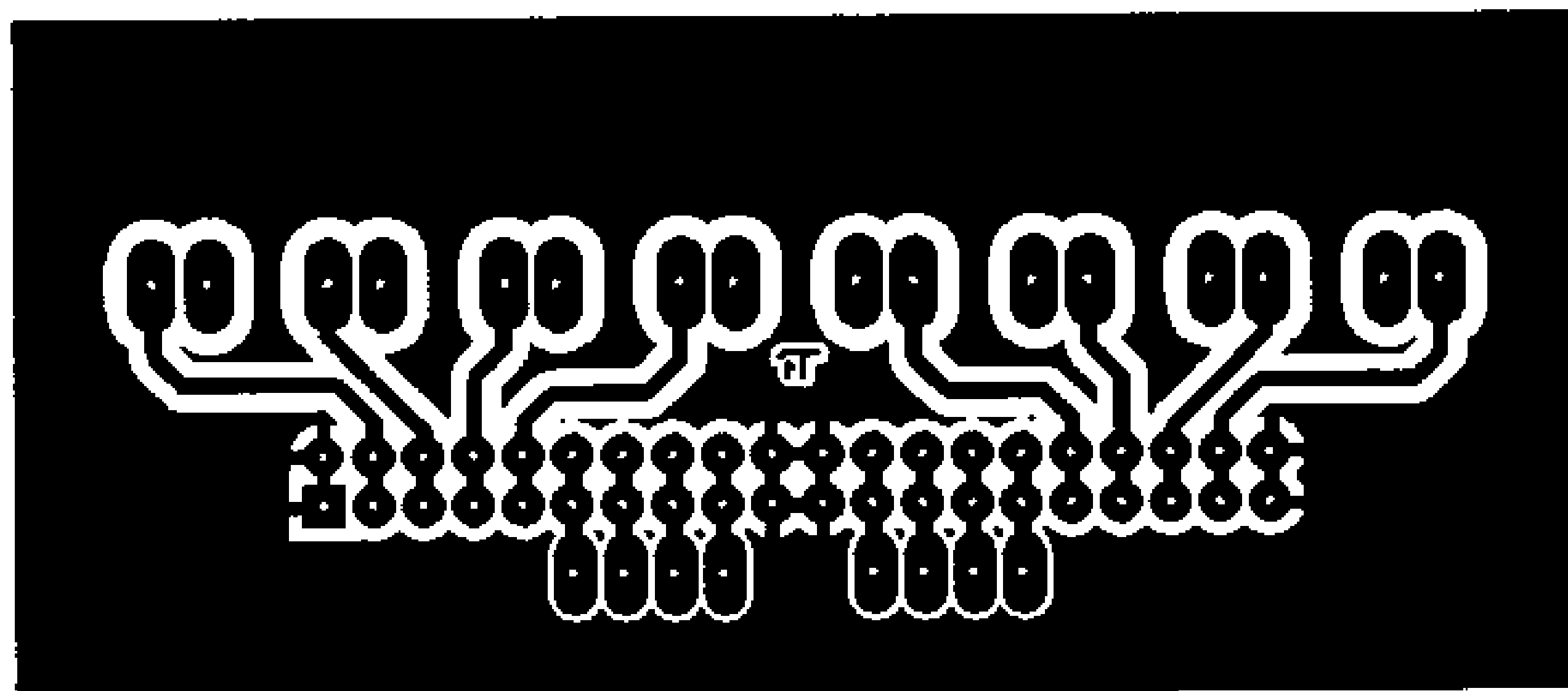
PCB layout πλακέτας διασύνδεσης με LCD, 1:1 μέγεθος:



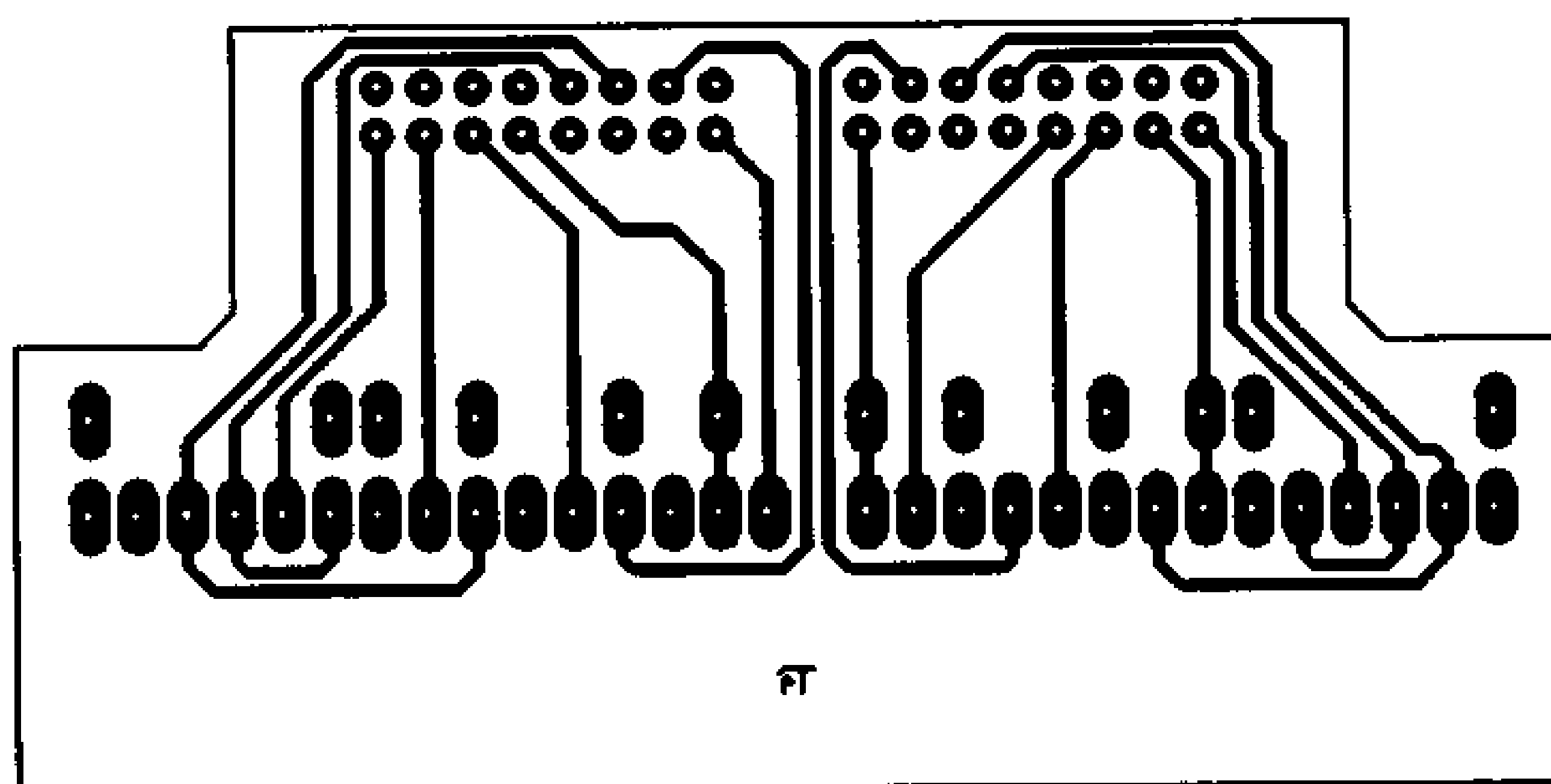
PCB layout πλακέτας ηλεκτροδίων ράβδου (άνω όψη), 1:1 μέγεθος:



PCB layout πλακέτας ηλεκτροδίων ράβδου (κάτω όψη), 1:1 μέγεθος:

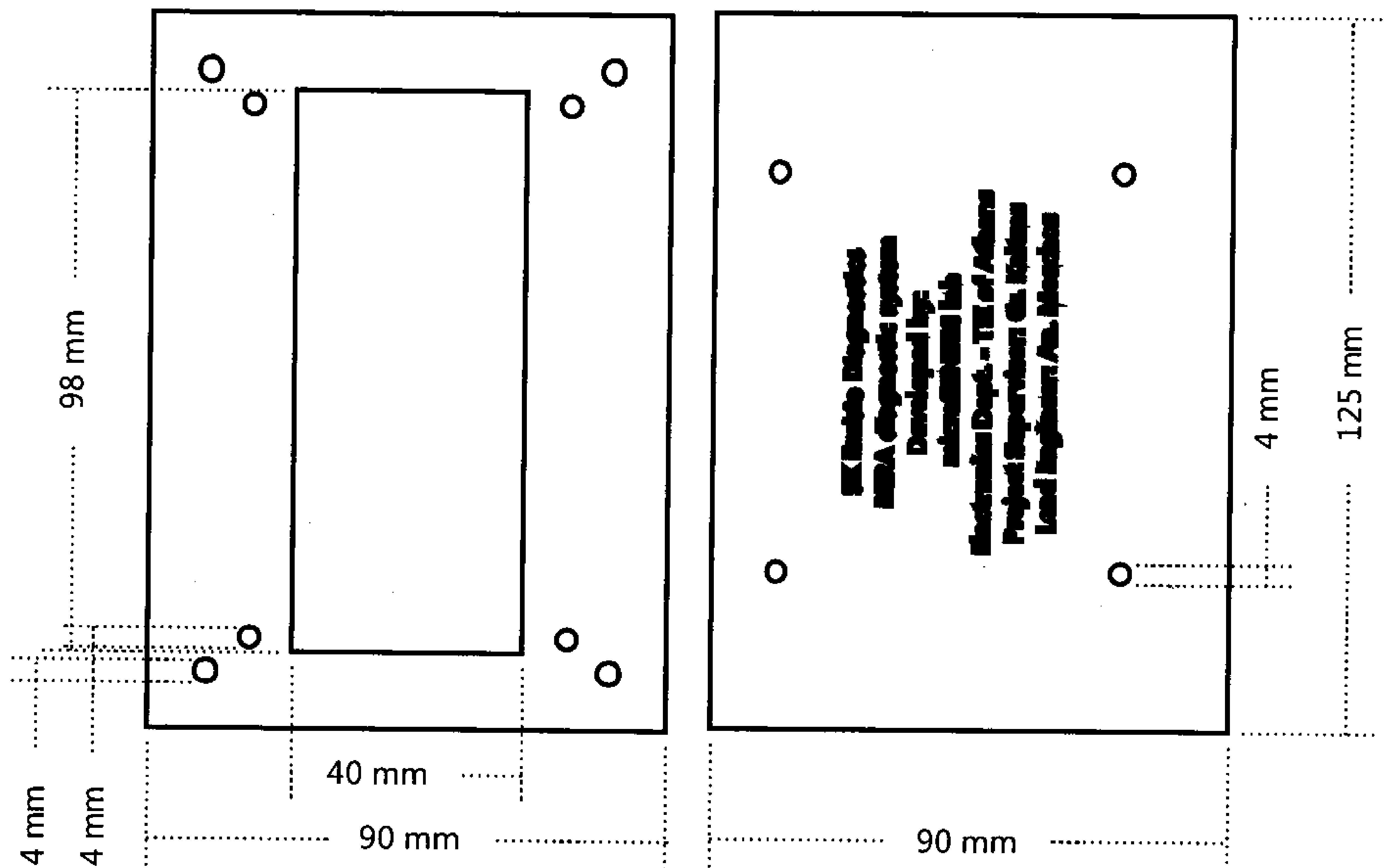


PCB layout πλακέτας προσαρμογής ηλεκτροδίων Dropsens, 1:1 μέγεθος:

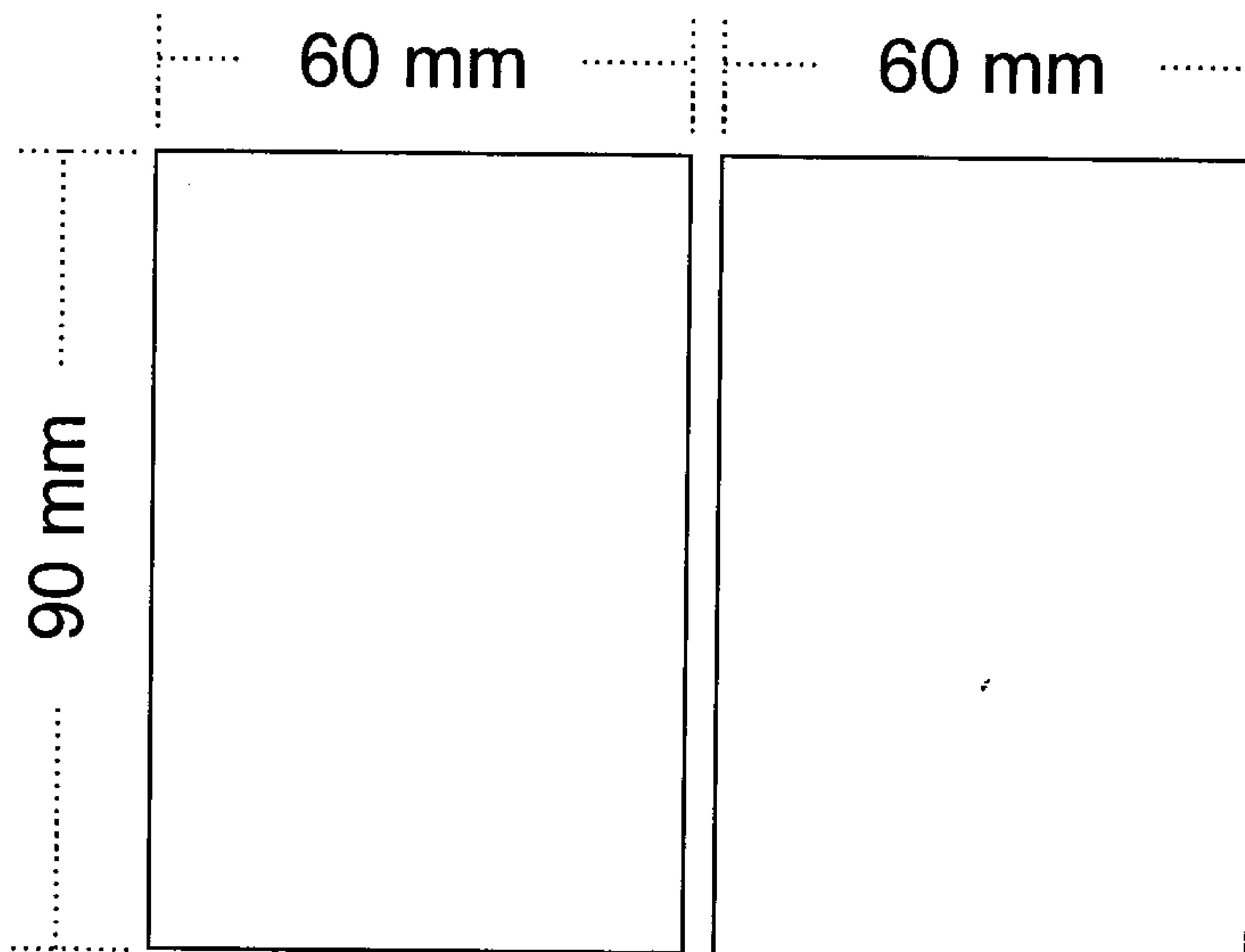


Ε. Σχέδια κουτιού

Άνω και κάτω όψη (υλικό 5mm πάχος):

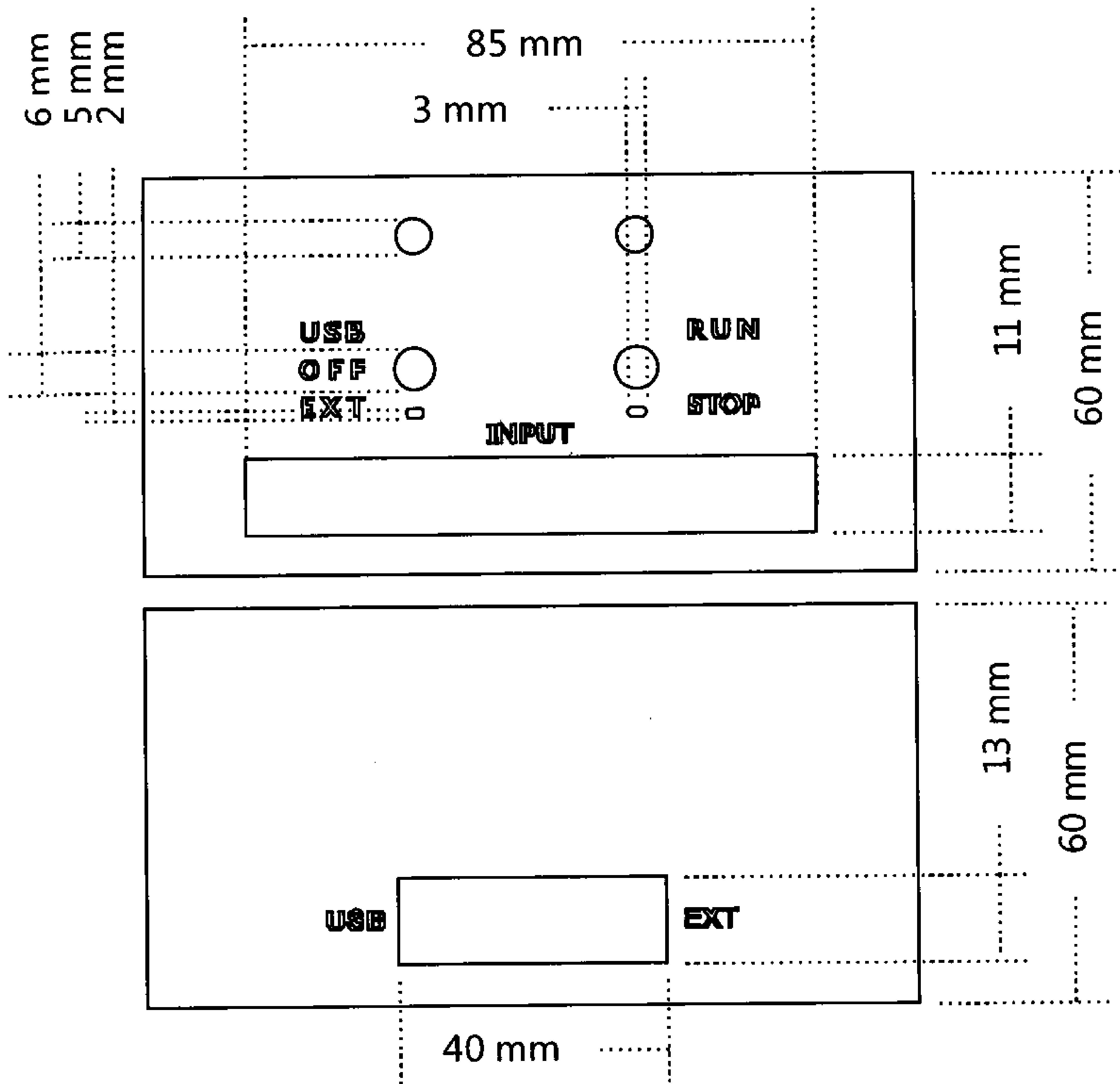


Δεξιά και αριστερά πλαϊνά τμήματα (υλικό 5mm πάχος):

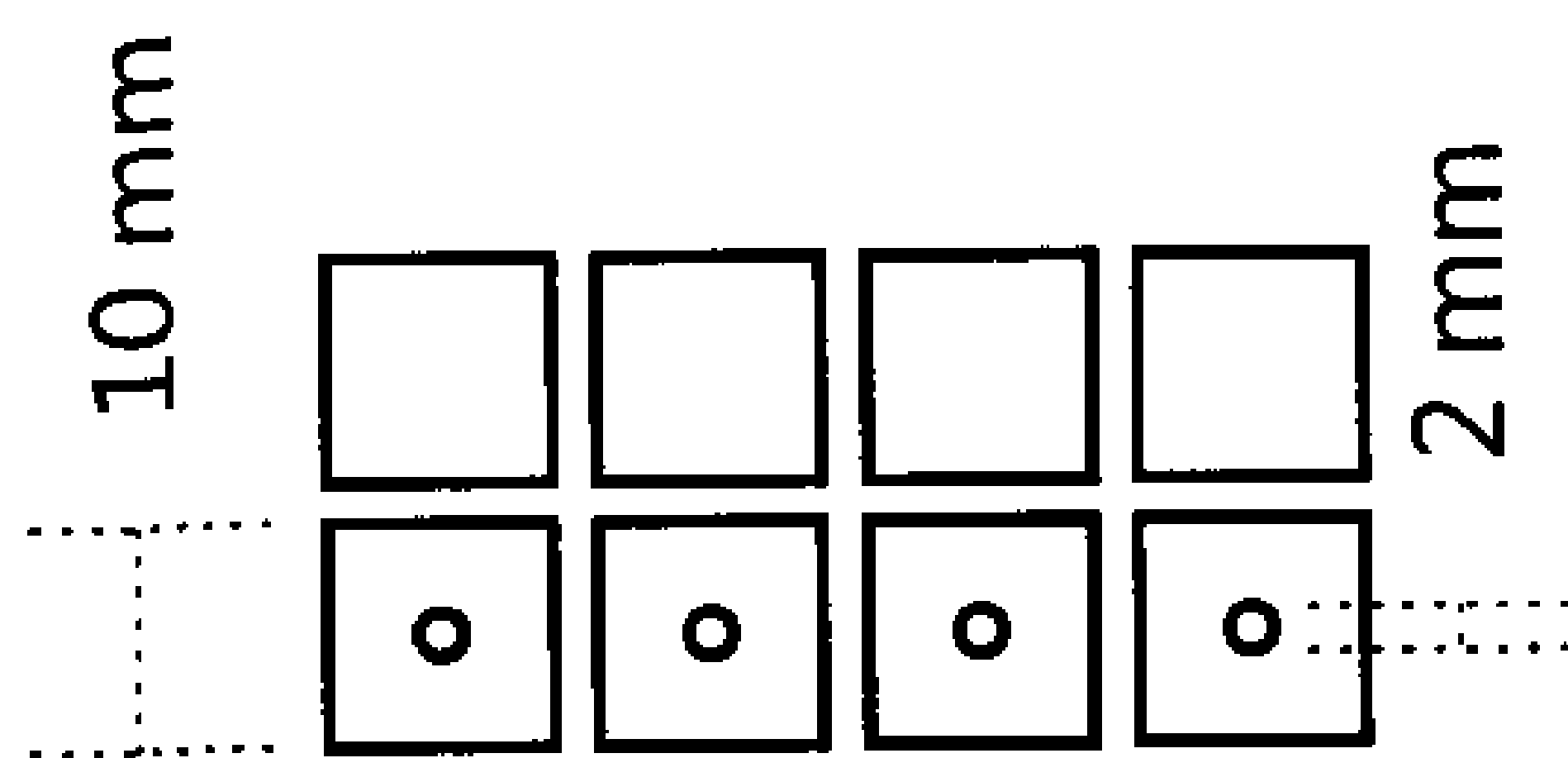


Σχέδια κουτιού

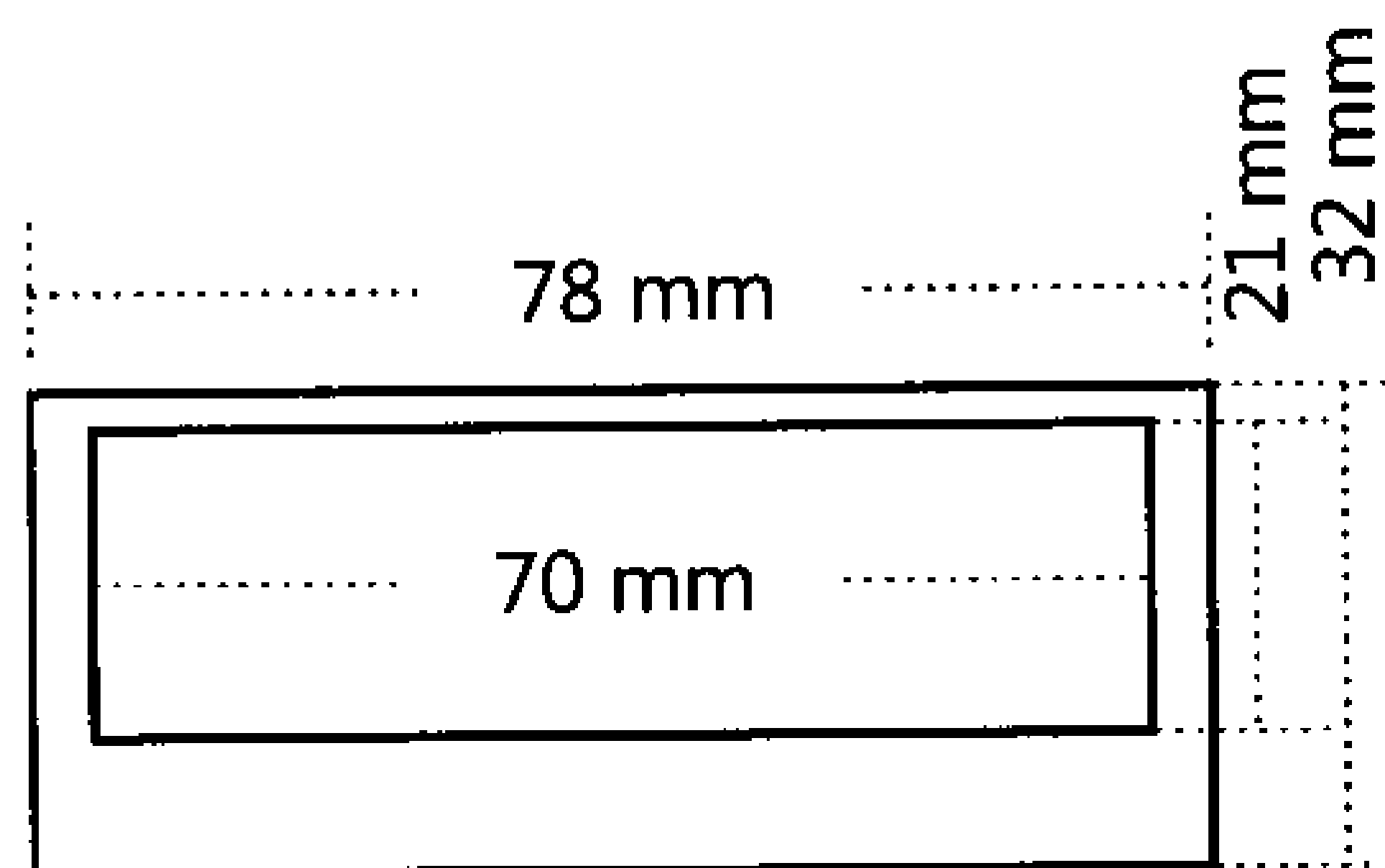
Εμπρός και πίσω όψη(υλικό 5mm πάχος):



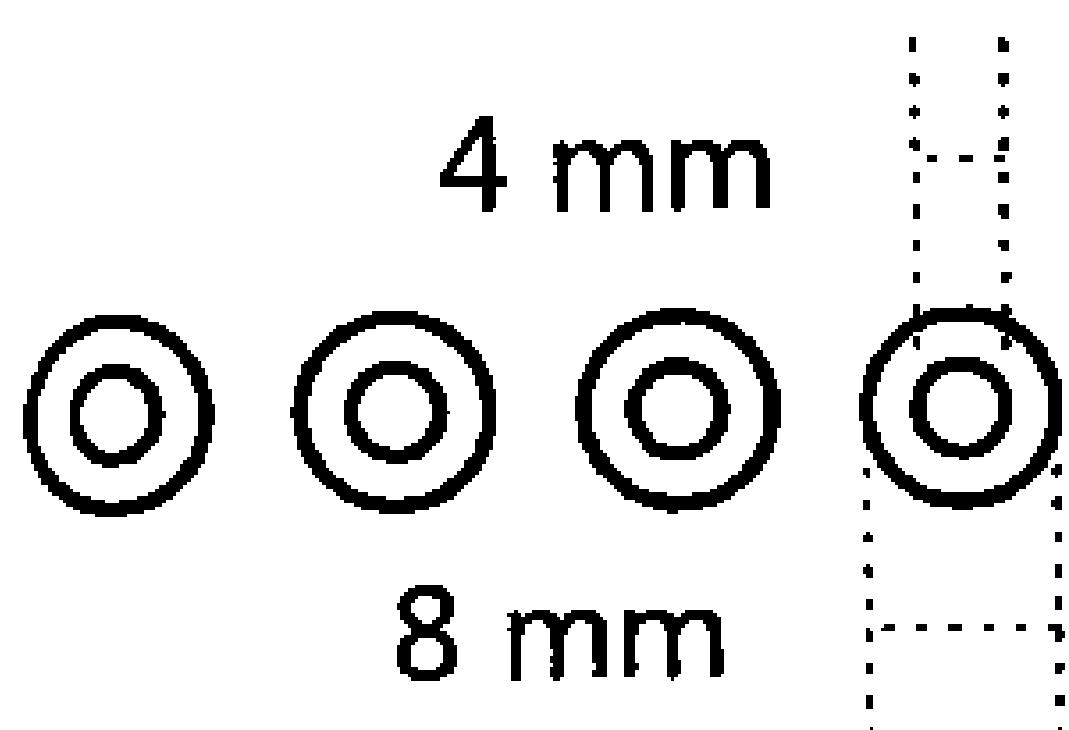
Σύνδεσμοι (υλικό 10mm πάχος):



Αποστάτες ηλεκτροδίων (υλικό 4mm πάχος):



Αποστάτες PCB (υλικό 5mm πάχος):



Αποστάτες LCD (υλικό 4mm πάχος):

