



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ,
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων
ασυρμάτων αισθητήρων**

ΙΩΑΝΝΗΣ Α. ΚΑΛΑΒΡΟΣ

**Επιβλέπων: Γεώργιος Σταμούλης
Καθηγητής Π.Θ.**

Βόλος, Μάρτιος 2008



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΒΙΒΛΙΟΘΗΚΗ & ΚΕΝΤΡΟ ΠΛΗΡΟΦΟΡΗΣΗΣ
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 575/1
Ημερ. Εισ.: 10-04-2008
Δωρεά: Συγγραφέα
Ταξιθετικός Κωδικός: ΠΤ – ΜΗΥΤΔ
2008
ΚΑΛ

Η σελίδα αυτή είναι σκόπιμα λευκή



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ,
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων
ασυρμάτων αισθητήρων**

ΙΩΑΝΝΗΣ Α. ΚΑΛΑΒΡΟΣ

**Επιβλέπων: Γεώργιος Σταμούλης
Καθηγητής Π.Θ.**

Εγκρίθηκε από την διμελή εξεταστική επιτροπή την 12^η Μαρτίου 2008.

.....
Γεώργιος Σταμούλης
Καθηγητής Π.Θ.

.....
Παναγιώτης Κίκιρας
Λέκτορας ΠΔ407/80

Βόλος, Μάρτιος 2008

ΙΩΑΝΝΗΣ Α. ΚΑΛΑΒΡΟΣ

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων
Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας.

Copyright © Ιωάννης Α. Καλαβρός, 2008

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας.

Η σελίδα αυτή είναι σκόπιμα λευκή

Πρόλογος

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Γεώργιο Σταμούλη για την εμπιστοσύνη που μου έδειξε κατά τη διεξαγωγή της διπλωματικής εργασίας, αλλά κυρίως για την υποδειγματική του παρουσία και διδασκαλία όλα αυτά τα χρόνια της φοίτησής μου στη σχολή. Επίσης θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου κ. Παναγιώτη Κίκιρα για την καθοδήγηση καθ' όλη τη διάρκεια της εργασίας μου.

Ένα μεγάλο ευχαριστώ οφείλω στην οικογένειά μου για τη στήριξη που μου προσέφερε σε όλη την περίοδο των σπουδών μου.

Η σελίδα αυτή είναι σκόπιμα λευκή

Περίληψη

Στην παρούσα εργασία παρουσιάζεται ο σχεδιασμός και η ανάπτυξη ενός συστήματος αισθητήρα για μέτρηση της συγκέντρωσης αερίου αμμωνίας στον ατμοσφαιρικό αέρα. Το σύστημα αυτό αναπτύχθηκε για να μπορεί να χρησιμοποιηθεί από κόμβους ενός δικτύου ασυρμάτων αισθητήρων.

Η σελίδα αυτή είναι σκόπιμα λευκή

Κατάλογος περιεχομένων

1	Εισαγωγή.....	1
1.1	Αντικείμενο της διπλωματικής.....	1
1.2	Οργάνωση του τόμου.....	2
2	Δίκτυα ασυρμάτων αισθητήρων	4
2.1	Γενική περιγραφή.....	4
2.2	Εφαρμογές δικτύων ασυρμάτων αισθητήρων.....	4
3	Αισθητήρας ανίχνευσης αμμωνίας.....	5
3.1	Επιλογή του κατάλληλου αισθητήρα.....	5
3.2	Παρουσίαση του Figaro TGS-2444.....	6
3.2.1	Αρχή λειτουργίας αισθητήρων τύπου ημιαγωγού.....	6
3.2.2	Περιγραφή TGS-2444.....	9
3.2.3	Απαιτήσεις για το κύκλωμα διασύνδεσης του αισθητήρα.....	16
4	Εργαλεία σχεδίασης και ανάπτυξης.....	17
4.1	Σχεδίαση ηλεκτρονικού κυκλώματος.....	17
4.2	Προγραμματισμός μικροελεγκτή.....	19
4.2.1	Παρουσίαση μικροελεγκτών AVR.....	19
4.2.2	Τεχνικά χαρακτηριστικά ATmega8.....	21
4.2.3	Κατασκευή προγραμματιστή.....	25
4.2.4	Περιβάλλον προγραμματισμού.....	27
5	Σχεδιασμός ηλεκτρονικής κατασκευής.....	28
5.1	Κύκλωμα διασύνδεσης αισθητήρα.....	28
5.2	Αναλογική έξοδος μετρούμενης τιμής.....	30
5.3	Κύκλωμα τροφοδοσίας	31
5.4	Σύνδεση οθόνης.....	32
5.5	Παρουσίαση τελικών σχεδίων.....	34
5.5.1	Ηλεκτρονικό σχέδιο.....	34
5.5.2	Σχέδιο τυπωμένης πλακέτας.....	35
6	Κατασκευή πλακέτας.....	37
6.1	Εμφάνιση.....	38
6.1.1	Θάλαμος έκθεσης.....	38
6.1.2	Έκθεση σε UV.....	40
6.1.3	Αφαίρεση φωτοευαίσθητης επίστρωσης.....	40
6.2	Αποχάλκωση.....	42
6.3	Επικασσιτέρωση.....	43
6.4	Τρύπημα.....	44
7	Περιγραφή προγράμματος μικροελεγκτή.....	45

8 Λειτουργίας της κατασκευής	47
8.1 Περιγραφή πλακέτας.....	47
8.2 Διαδικασία βαθμονόμησης.....	49
8.2.1 Ρύθμιση βάσει των ηλεκτρικών χαρακτηριστικών του αισθητήρα.....	49
8.2.2 Αντιστάθμιση θερμοκρασίας και υγρασίας του περιβάλλοντος.....	50
8.3 Διασύνδεση με Tmote Sky.....	53
9 Μετρήσεις	55
9.1 Περιγραφή εξοπλισμού και διαδικασίας μετρήσεων.....	55
9.1.1 Εξοπλισμός δοκιμών.....	55
9.1.2 Μεθοδολογία μετρήσεων.....	56
9.1.3 Επεξεργασία δεδομένων.....	56
9.2 Πειραματικά δεδομένα.....	58
10 Επίλογος	61
10.1 Ιδέες για βελτιώσεις και επεκτάσεις.....	61
10.2 Συμπεράσματα.....	61
11 Παράρτημα	62
11.1 Πηγαίος κώδικας λογισμικού μικροελεγκτή.....	62
11.2 Κώδικας Matlab.....	78

1 Εισαγωγή

1.1 Αντικείμενο της διπλωματικής

Τα ασύρματα δίκτυα αισθητήρων βρίσκουν ολοένα και περισσότερες εφαρμογές σε εμπορικά και βιομηχανικά περιβάλλοντα διευκολύνοντας το έργο της επιτήρησης ενός πλήθους φυσικών μεγεθών. Όλοι οι κόμβοι αυτών των δικτύων έχουν την ίδια βασική δομή. Διαθέτουν μια μονάδα επεξεργασίας και κάποιο πομποδέκτη για ασύρματη επικοινωνία. Από εκεί και πέρα η διαφοροποίησή τους έχει να κάνει με τις επιμέρους μονάδες που διαθέτουν για να αντιλαμβάνονται και να μπορούν να μετράνε ορισμένα χαρακτηριστικά του περιβάλλοντός τους.

Αντικείμενο αυτής της εργασίας είναι η ανάπτυξη μιας τέτοιας μονάδας η οποία θα έχει τη δυνατότητα να μετράει την συγκέντρωση της αμμωνίας στον αέρα. Η μονάδα αυτή θα χρησιμοποιεί ένα αισθητήριο στοιχείο το οποίο είναι ευαίσθητο στην αμμωνία και θα πρέπει να διαθέτει τα απαραίτητα ηλεκτρονικά κυκλώματα για τη λειτουργία του. Ως έξοδο θα μας δίνει μία τιμή η οποία και θα διαβάζεται από τον κόμβο του ασύρματου δικτύου.

Ο σχεδιασμός του συστήματος γίνεται με τέτοιο τρόπο ώστε η μονάδα με τον αισθητήρα να μπορεί να λειτουργήσει ακόμα κι αυτόνομα. Χωρίς δηλαδή να είναι συνδεδεμένη σε κάποιον κόμβο θα μπορούμε να διαβάσουμε την μέτρηση συνδέοντας μια LCD οθόνη.

Στην παρούσα εργασία παρουσιάζεται αναλυτικά όλη η διαδικασία, από την επιλογή του αισθητηρίου και τον σχεδιασμό του κυκλώματος διασύνδεσης του, μέχρι και την κατασκευή της πλακέτας για το ηλεκτρονικό κύκλωμα. Επίσης υπάρχουν και στοιχεία δοκιμών που δείχνουν την συμπεριφορά της κατασκευής σε συνθήκες πραγματικής λειτουργίας σε ένα δίκτυο με ασύρματο αισθητήρα.

1.2 Οργάνωση του τόμου

Η διπλωματική εργασία οργανώνεται στα παρακάτω κεφάλαια:

Στο Κεφάλαιο 2 γίνεται μια γενική περιγραφή των δικτύων ασυρμάτων αισθητήρων και παρουσιάζονται οι δυνατότητές τους μέσα από διάφορες εφαρμογές.

Στο Κεφάλαιο 3 γίνεται σύγκριση ανάμεσα στα αισθητήρια ανίχνευσης αμμωνίας που κυκλοφορούν και επιλέγουμε αυτό που μας εξυπηρετεί. Στην συνέχεια παρουσιάζεται αυτό που τελικά επιλέξαμε και αναλύεται η αρχή λειτουργίας του και τα τεχνικά χαρακτηριστικά του. Το κεφάλαιο ολοκληρώνεται με καταγραφή των λειτουργιών που θέλουμε να διαθέτει το σύστημα που θα αναπτύξουμε.

Το Κεφάλαιο 4 περιγράφει τα εργαλεία που θα χρησιμοποιήσουμε για την ανάπτυξη της κατασκευής. Αρχικά παρουσιάζεται το λογισμικό ηλεκτρονικής σχεδίασης και το περιβάλλον προγραμματισμού που θα χρησιμοποιηθεί. Επίσης γίνεται μια παρουσίαση του μικροελεγκτή που θα αποτελέσει την καρδιά του συστήματος καθώς και της κατασκευής του προγραμματιστή που θα τον προγραμματίζει.

Το Κεφάλαιο 5 ασχολείται με τον σχεδιασμό του ηλεκτρονικού κυκλώματος. Κάθε ένα επιμέρους τμήμα του τελικού κυκλώματος αναλύεται ξεχωριστά, δείχνοντας αναλυτικά τον τρόπο υπολογισμού των εξαρτημάτων που χρησιμοποιήθηκαν. Στο τέλος παρουσιάζεται το πλήρες σχέδιο του τελικού κυκλώματος καθώς και το σχέδιο που θα χρησιμοποιηθεί για την κατασκευή της πλακέτας.

Στο Κεφάλαιο 6 παρουσιάζεται όλη η διαδικασία που ακολουθήθηκε για την παραγωγή της ηλεκτρονικής πλακέτας βάσει του σχεδίου. Εξηγείται αναλυτικά το κάθε βήμα, παρουσιάζοντας τα υλικά και τα εργαλεία που χρησιμοποιήθηκαν.

Το Κεφάλαιο 7 δίνει μια γενική περιγραφή της δομής και της λειτουργίας του προγράμματος του μικροελεγκτή.

Στο Κεφάλαιο 8 παρουσιάζεται η ολοκληρωμένη και συναρμολογημένη πλακέτα της κατασκευής και εξηγούνται όλες οι επαφές διασύνδεσης που διαθέτει. Περιγράφεται επίσης η διαδικασία που πρέπει να ακολουθηθεί για τη σωστή ρύθμιση του αισθητήρα ώστε να είναι σε θέση να παρέχει σωστές μετρήσεις. Επίσης εξηγείται και η επεξεργασία που πρέπει να γίνει από πλευράς των κόμβων του ασύρματου δικτύου αισθητήρων στα αποτελέσματα που διαβάζουν από την κατασκευή μας, ώστε να αντισταθμιστεί η αλλοίωση των μετρήσεων που προκαλείται από τις συνθήκες του περιβάλλοντος.

Το Κεφάλαιο 9 ασχολείται με την δοκιμή του συστήματος του αισθητήρα. Αρχικά γίνεται μια περιγραφή του εξοπλισμού που χρησιμοποιήθηκε για τις δοκιμές και της διαδικασίας που ακολουθήθηκε για τις μετρήσεις. Στην συνέχεια του κεφαλαίου παρουσιάζονται τα αποτελέσματα των μετρήσεων που έγιναν.

Στο Κεφάλαιο 10 γίνεται μια σύνοψη των αποτελεσμάτων της παρούσας εργασίας και περιγράφονται οι δυνατότητες βελτιώσεων και επεκτάσεων που μπορούν να γίνουν στο σύστημα που αναπτύχθηκε.

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Το Κεφάλαιο 11 έχει τον πηγαίο κώδικα του λογισμικού του μικροελεγκτή. Επίσης υπάρχουν και τα προγράμματα Matlab που χρησιμοποιήθηκαν για την επεξεργασία των δεδομένων από τις μετρήσεις.

2 Δίκτυα ασυρμάτων αισθητήρων

2.1 Γενική περιγραφή

Ένα δίκτυο ασυρμάτων αισθητήρων είναι ένα ασύρματο δίκτυο που αποτελείται από αυτόνομες συσκευές καταναμημένες στο χώρο οι οποίες χρησιμοποιούν αισθητήρες και σε συνεργασία μεταξύ τους παρακολουθούν διάφορες συνθήκες του περιβάλλοντος, όπως θερμοκρασία, ηχητική ένταση, πίεση, δονήσεις, κίνηση ή ρύπους. Ο κάθε αισθητήρας έχει δυνατότητα επεξεργασίας σήματος και ασύρματης επικοινωνίας με τους γειτονικούς του σχηματίζοντας με αυτόν τον τρόπο δίκτυα επικοινωνίας.

Η ανάπτυξη δικτύων ασυρμάτων αισθητήρων ξεκίνησε από στρατιωτικές εφαρμογές όπως η επιτήρηση του πεδίου της μάχης. Στη συνέχεια όμως χρησιμοποιήθηκαν σε πολλές άλλες εφαρμογές όπως στην παρακολούθηση του περιβάλλοντος εντός και εκτός σπιτιού, σε εφαρμογές υγείας, συστήματα αυτοματισμού και στον έλεγχο της κυκλοφορίας στους δρόμους.

Ο κάθε κόμβος ενός τέτοιου δικτύου έχει έναν ή περισσότερους αισθητήρες, έναν ασύρματο πομποδέκτη, έναν μικροελεγκτή και κάποια πηγή ενέργειας, συνήθως μια μπαταρία.

2.2 Εφαρμογές δικτύων ασυρμάτων αισθητήρων

Τα ασύρματα δίκτυα αισθητήρων έχουν πολλές εφαρμογές σε εμπορικά και βιομηχανικά περιβάλλοντα. Χρησιμοποιούνται για μετρήσεις σε διάφορα σημεία όπου θα ήταν οικονομικά ασύμφορο ή και αδύνατο να τοποθετηθούν αισθητήρες που διασυνδέονται με καλώδιο. Μπορούν να λειτουργήσουν αυτόνομα σε απομακρυσμένα σημεία για μεγάλο χρονικό διάστημα χωρίς να απαιτείται επαναφόρτιση ή αλλαγή των μπαταριών τους.

Παρακάτω παρουσιάζονται μερικά παραδείγματα εφαρμογών τέτοιων δικτύων:

- Στρατιωτικά δίκτυα αισθητήρων για επιτήρηση κινήσεων φίλιων και εχθρικών δυνάμεων, καθώς και για αποδοτικότερη υποστήριξη του μετώπου με εξοπλισμού και πυρομαχικά.
- Ανίχνευση χημικών, βιολογικών ή πυρηνικών κινδύνων στις περιοχές επιτήρησης.
- Συλλογή στοιχείων για μελέτη του φυσικού περιβάλλοντος και των αλλαγών του.
- Ανάλυση κυκλοφοριακών συνθηκών σε οδικές αρτηρίες ώστε να λαμβάνονται μέτρα για την αποδοτικότερη ρύθμιση της κυκλοφορίας.
- Ασύρματα δίκτυα αισθητήρων για επιτήρηση και παροχή ασφάλειας σε καταστήματα, χώρους στάθμευσης κλπ
- Παρακολούθηση κατάστασης ασθενών σε νοσοκομεία και κλινικές.

3 Αισθητήρας ανίχνευσης αμμωνίας

3.1 Επιλογή του κατάλληλου αισθητήρα

Πολλοί από τους αισθητήρες που κυκλοφορούν στην αγορά είναι κατασκευασμένοι για βιομηχανικές εφαρμογές και για αυτόν τον λόγο επικεντρώνονται σε μεγαλύτερες συγκεντρώσεις αμμωνίας, συνήθως μεγαλύτερες από 50ppm. Όμως για τις ανάγκες της εφαρμογής που περιγράφηκε στην προηγούμενη ενότητα, χρειαζόμαστε το αισθητήριο στοιχείο να μπορεί να ανιχνεύσει χαμηλές συγκεντρώσεις αμμωνίας.

Επίσης μας ενδιαφέρει να μην επηρεάζεται από άλλα αέρια, αλλά να παρουσιάζει υψηλή επιλεκτικότητα στην αμμωνία.

Με αυτά τα κριτήρια οι επιλογές μας περιορίστηκαν σε τέσσερις αισθητήρες των εταιρειών Nemoto, Synkera και Figaro που παρουσιάζονται συνοπτικά παρακάτω.

Κατασκευαστής	Nemoto	Synkera	Figaro	
Μοντέλο	NT-NH3	NH3	TGS-826	TGS-2444
Συγκέντρωση NH3	0-100ppm	25-300ppm	30-300ppm	10-100ppm
Ηλεκτρική κατανάλωση	-	975mW	848mW	56mW
	www.nemototech.com	www.synkera.com	www.figaro.co.jp	

Πίνακας 3.1: Συγκριτικός πίνακας αισθητήρων αμμωνίας

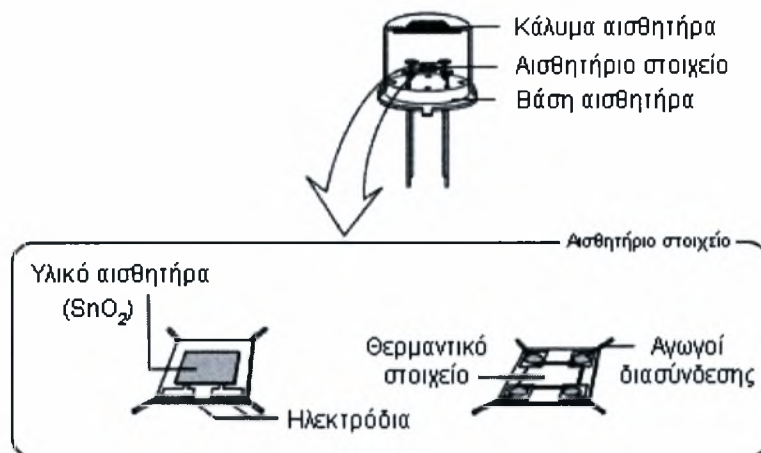
Από αυτούς επιλέχθηκε τελικώς ο TGS-2444 της Ιαπωνικής εταιρείας Figaro. Ένας σημαντικός λόγος που μας ώθησε προς αυτήν την επιλογή είναι η πολύ μικρή του κατανάλωση συγκριτικά με τους υπόλοιπους καθώς και το γεγονός ότι παρουσιάζει υψηλή ευαισθησία ακόμα και σε χαμηλές συγκεντρώσεις αμμωνίας. Ένα επιπλέον πλεονέκτημά του είναι το πολύ μικρό του μέγεθος που επιτρέπει τον περιορισμό των διατάσεων της τελικής κατασκευής.

Επίσης η τιμή του είναι ελαφρώς χαμηλότερη σε σχέση με τις ανταγωνίστριες εταιρείες.

3.2 Παρουσίαση του Figaro TGS-2444

3.2.1 Αρχή λειτουργίας αισθητήρων τύπου ημιαγωγού

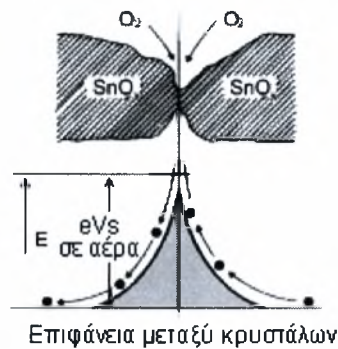
Όπως φαίνεται και στην εικόνα 3.1 ο αισθητήρας αερίου αποτελείται από το αισθητήριο στοιχείο, την βάση και το κάλυμμα. Το αισθητήριο στοιχείο αποτελείται από το ημιαγωγίμο υλικό και από το θερμαντικό στοιχείο που το θερμαίνει (πχ. στους 400°C). Το ημιαγωγίμο υλικό εξαρτάται από τον τύπο του αερίου που θέλουμε να ανιχνεύουμε. Για την περίπτωση της αμμωνίας είναι το διοξειδίου του κασσιτέρου (SnO_2).



Εικόνα 3.1: Εσωτερική δομή αισθητήρα

Όταν ένας κρύσταλλος SnO_2 θερμαίνεται σε συγκεκριμένη υψηλή θερμοκρασία παρουσία αέρα, απορροφάται οξυγόνο στην επιφάνεια των κρυστάλλων με αρνητικό φορτίο. Τότε τα ελεύθερα ηλεκτρόνια από την επιφάνεια του κρυστάλλου μεταφέρονται στο απορροφημένο οξυγόνο με αποτέλεσμα να αφήνουν θετικό φορτίο σε ένα επιφανειακό στρώμα [1]. Καθ' αυτόν τον τρόπο αναπτύσσεται ένα επιφανειακό φορτίο που λειτουργεί ως εμπόδιο στην ροή των ηλεκτρονίων (εικόνα 3.2).

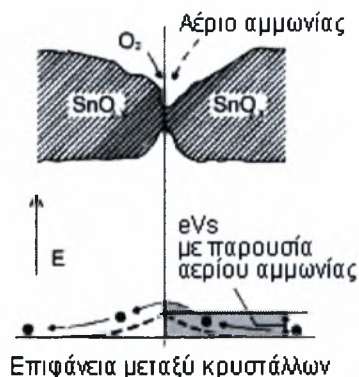
Μέσα στον αισθητήρα το ηλεκτρικό ρεύμα ρέει διαμέσου των σημείων επαφής μεταξύ των μικρο-κρυστάλλων του SnO_2 . Στα σημεία αυτά, το απορροφημένο οξυγόνο δημιουργεί ένα φραγμό που εμποδίζει την ελεύθερη κίνηση των φορτίων. Η ηλεκτρική αντίσταση του αισθητήρα οφείλεται σε αυτό ακριβώς το φαινόμενο.



- Ηλεκτρόνιο
- eVs: ενεργειακή στάθμη αντίστασης επιφανείας

Εικόνα 3.2: Μοντέλο του φραγμού στη ροή ηλεκτρονίων (απουσία αμμωνίας)

Υπό την παρουσία αμμωνίας, η επιφανειακή πυκνότητα του αρνητικά φορτισμένου οξυγόνου ελαττώνεται και έτσι χαμηλώνει και η ενεργειακή στάθμη της αντίστασης που παρουσιάζεται μεταξύ των μικρο-κρυστάλλων (εικόνα 3.3). Αυτό οδηγεί και στην μείωση της ηλεκτρικής αντίστασης του αισθητήρα.



- Ηλεκτρόνιο

Εικόνα 3.3: Μοντέλο του φραγμού στη ροή ηλεκτρονίων (παρουσία αμμωνίας)

Η σχέση μεταξύ της αντίστασης του αισθητήρα και της συγκέντρωσης του αερίου μπορεί να εκφραστεί με την μαθηματική εξίσωση: $R_s = A[C]^{-\alpha}$

όπου: R_s = ηλεκτρική αντίσταση του αισθητήρα

A = σταθερά

$[C]$ = συγκέντρωση αερίου αμμωνίας

α = κλίση της καμπύλης R_s

Λόγω της λογαριθμικής σχέσης μεταξύ της αντίστασης του αισθητήρα και της συγκέντρωσης του αερίου, οι αισθητήρες τύπου ημιαγωγού έχουν το πλεονέκτημα της υψηλής ευαισθησίας ακόμα και

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

σε χαμηλές συγκεντρώσεις αερίων.

Η εξαιρετική σταθερότητα και οι επιδόσεις των αισθητήρων αυτού του τύπου παρέχουν την δυνατότητα ανίχνευσης έχοντας με μεγάλο όριο ζωής και χαμηλό κόστος, χωρίς να χρειάζονται συντήρηση.

3.2.2 Περιγραφή TGS-2444

Το αισθητήριο στοιχείο του TGS-2444 έχει μία δομή πολλαπλών στρωμάτων. Η βάση του αισθητήρα αποτελείται από αλουμίνιο πάνω στο οποίο υπάρχει ένα θερμομονωτικό στρώμα γυαλιού το οποίο το διαχωρίζει από το θερμαντικό στοιχείο που είναι διοξείδιο του ρουθηνίου (RuO_2). Το θερμαντικό στοιχείο για την τροφοδοσία του συνδέεται με ένα ζεύγος ηλεκτροδίων από χρυσό.

Η ανίχνευση του αερίου γίνεται από ένα στρώμα διοξειδίου του κασσιτέρου (SnO_2) που βρίσκεται πάνω από μια επίστρωση μονωτικού που παρέχει την απαραίτητη ηλεκτρική απομόνωση από το θερμαντικό στοιχείο. Μεταξύ του εξωτερικού περιβλήματος και του εσωτερικού που καλύπτει το αισθητήριο στοιχείο, παρεμβάλλεται ένα φίλτρο ενεργού άνθρακα για να μειώνεται η επίδραση των αερίων που δεν μας ενδιαφέρουν.

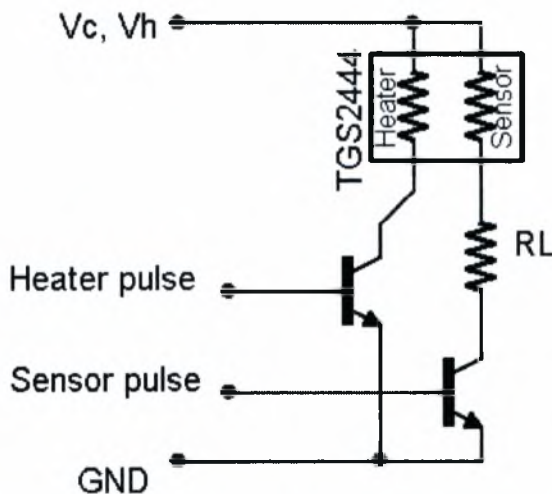
Όταν βρεθεί σε περιβάλλον με παρουσία αμμωνίας, η αγωγιμότητα του αισθητήρα αυξάνεται ανάλογα με την συγκέντρωση του αερίου στον αέρα. Ένα εξωτερικό κύκλωμα που λειτουργεί με παλμούς τάσης μπορεί να μετατρέψει αυτή την αλλαγή της αγωγιμότητας σε ένα σήμα που αντιστοιχεί στην συγκέντρωση του αερίου.

Η λειτουργία με παλμούς τροφοδοσίας απαιτείται για να εμποδίσει την πιθανή μεταφορά από υλικά του θερμαντικού στοιχείου προς την επιφάνεια του αισθητήριου στοιχείου. Κάτω από ακραίες συνθήκες υψηλής υγρασίας και θερμοκρασίας, μια συνεχής τάση τροφοδοσίας εφαρμοζόμενη στον αισθητήρα μπορεί να προκαλέσει αυτήν την μεταφορά με αποτέλεσμα μακροπρόθεσμα ολίσηση της τιμής της αντίστασης του προς μεγαλύτερες τιμές.

Η τάση τροφοδοσίας (V_c) εφαρμόζεται στο αισθητήριο στοιχείο το οποίο εμφανίζει μια αντίσταση R_s (pins 2 και 3) και σε μία αντίσταση φορτίου R_L που συνδέεται σε σειρά με αυτό. Το θερμαντικό στοιχείο (pins 1 και 4) συνδέεται και αυτό με την τροφοδοσία.



Εικόνα 3.4:
Αισθητήρας Figaro
TGS-2444

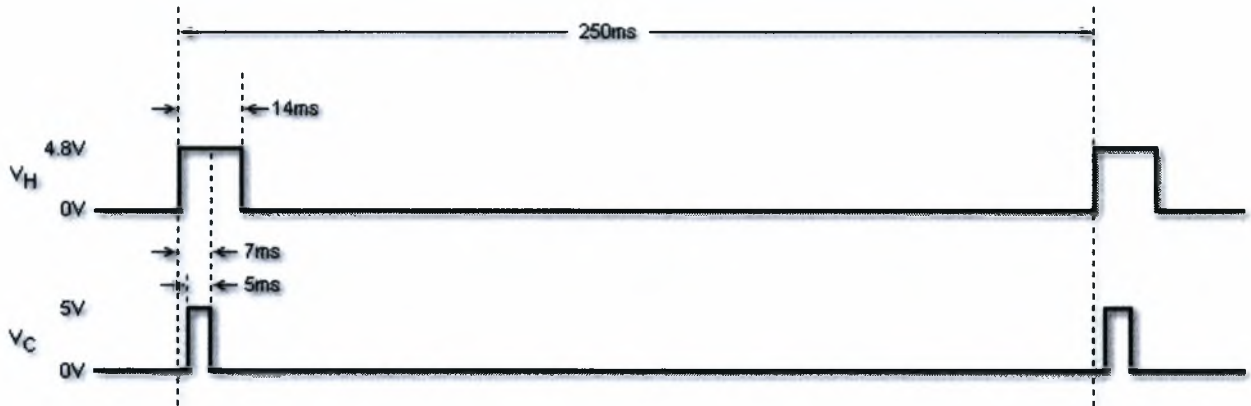


Εικόνα 3.5: Στοιχειώδες κύκλωμα για λειτουργία του αισθητήρα και μετρήσεις

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Η θέρμανση του αισθητήρα πρέπει να γίνεται περιοδικά κάθε 250ms σε συγχρονισμό με την λήψη των μετρήσεων που γίνεται και αυτή κάθε 250ms. Για τη θέρμανση απαιτείται εφαρμογή τάσης 4.8V ($\pm 0.2V$) για 14ms ακολουθούμενη με 0V για τα υπόλοιπα 236ms. Για την λήψη των μετρήσεων αρχικά εφαρμόζονται 0V για 2ms, στη συνέχεια 5V για 5ms και 0V για τα υπόλοιπα 243ms. Για την καλύτερη δυνατή μέτρηση, η δειγματοληψία πρέπει να γίνει στο δεύτερο μισό αυτού του παλμού των 5ms [2].

Όλα τα παραπάνω φαίνονται πιο παραστατικά στο διάγραμμα χρονισμού που ακολουθεί:



Εικόνα 3.6: Διάγραμμα χρονισμού τάσεων λειτουργίας του αισθητήρα

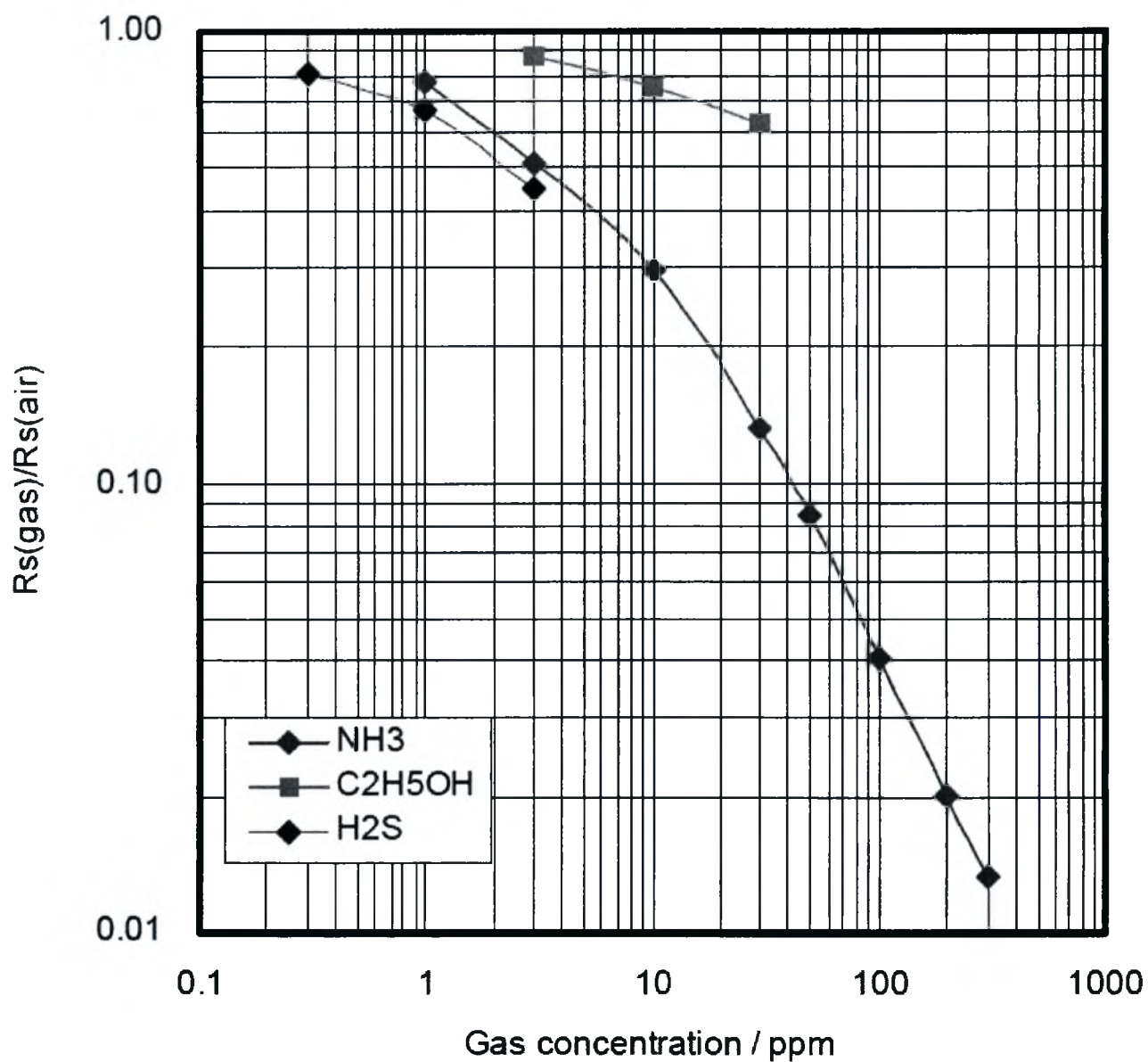
Τεχνικά χαρακτηριστικά:

Μοντέλο			TGS-2444	
Τύπος αισθητήρα			M1	
Συσκευασία			TO-5 metal can	
Ανιχνεύσιμο αέριο			NH ₃ (Αμμωνία)	
Τυπική ανιχνεύσιμη συγκέντρωση			10-100ppm	
Τυπικές συνθήκες λειτουργίας κυκλώματος	Τάση θερμαντικού στοιχείου	V _H	V _H = 4.8V ±0.2V DC για 14ms V _H = 0V για 236ms	
	Τάση λειτουργίας αισθητήρα	V _C	V _C = 0V για 245ms V _C = 5V ±0.2V DC για 5ms (βλ. εικόνα 3.6 για χρονισμό)	
	Αντίσταση φορτίου	R _L	μεταβλητή	P _S ≥ 8kΩ
Ηλεκτρικά χαρακτηριστικά υπό κανονικές συνθήκες	Αντίσταση θερμαντικού στοιχείου	R _H	17 ± 2.5Ω σε θ _{δομ.αίτ.}	
	Ρεύμα θερμαντικού στοιχείου	I _H	≈ 203mA	
	Ισχύς θερμαντικού στοιχείου	P _H	56mW	
	Αντίσταση αισθητήρα	R _S	3.63kΩ ~ 36.3kΩ σε 10ppm NH ₃	
	Ευαισθησία (λόγος μεταβολής R _S)	β	0.063 ~ 0.63	
Κανονικές συνθήκες δοκιμών	Συνθήκες αερίου δοκιμών		Αμμωνία σε αέρα θερμ. 20±2°C, 65±5% RH	
	Συνθήκες κυκλώματος		Ίδιες με Τυπικές συνθήκες λειτουργίας κυκλώματος (βλέπε παραπάνω)	
	Περίοδος προσαρμογής πριν τις δοκιμές		≥ 48 ώρες	
Συνθήκες λειτουργίας	-10°C ~ +50°C χωρίς υγραποίηση υδρατμών			
Συνθήκες αποθήκευσης	-20°C ~ +60°C χωρίς υγραποίηση υδρατμών			

Πίνακας 3.2: Τεχνικά χαρακτηριστικά αισθητήρα TGS-2444

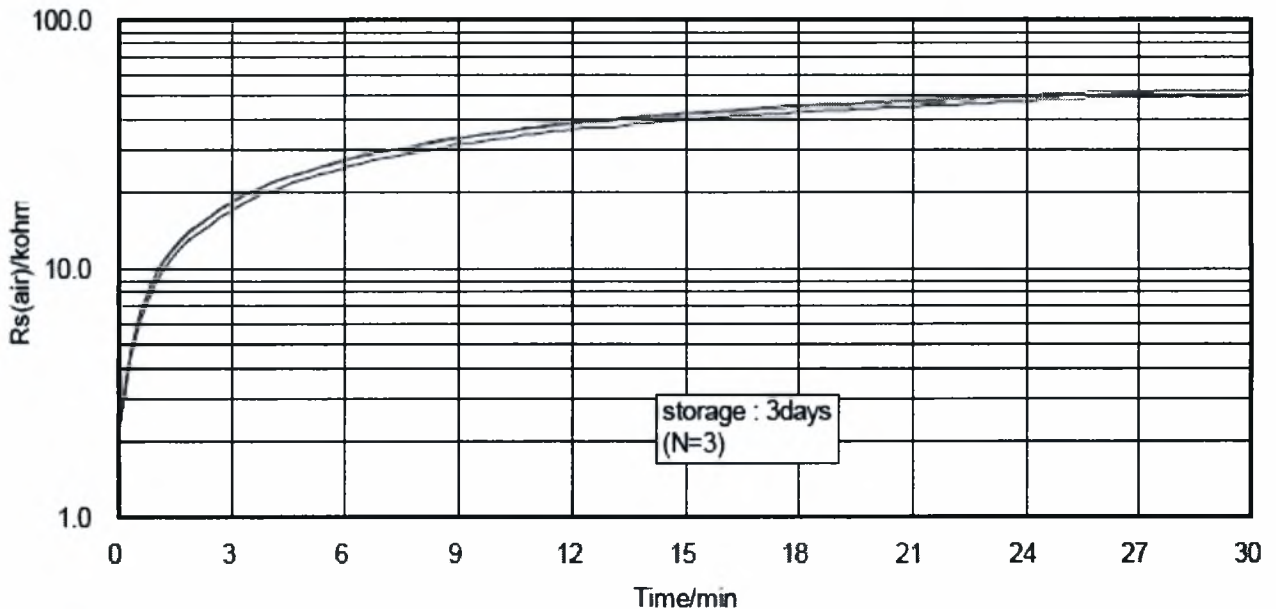
Η τιμή της αντίστασης του αισθητήρα (R_S) υπολογίζεται βάσει της μετρούμενης τιμής του V_{out} σύμφωνα με τον μαθηματικό τύπο:
$$R_S = \frac{V_C \times R_L}{V_{out}} - R_L$$

Βάσει της R_S και γνωρίζοντας την τιμή της αντίστασης που παρουσιάζει όταν βρίσκεται σε καθαρό αέρα, μπορούμε να υπολογίσουμε την συγκέντρωση της αμμωνίας στον χώρο. Η αντιστοίχιση αυτή ορίζεται στο παρακάτω διάγραμμα που μας δίνει ο κατασκευαστής:



Εικόνα 3.7: Διάγραμμα αντιστοίχισης της μεταβολής στην αντίσταση του αισθητήρα ως προς την συγκέντρωση της αμμωνίας στον αέρα

Λόγω των χημικών αντιδράσεων που γίνονται στην επιφάνεια του αισθητήριου στοιχείου, χρειάζεται να περάσει κάποιος χρόνος από την έναρξη της λειτουργίας του μέχρι τη στιγμή που θα είναι σε θέση να μας δώσει αξιόπιστες μετρήσεις. Σύμφωνα με τον κατασκευαστή, ο χρόνος αυτός που απαιτείται για να σταθεροποιηθεί η τιμή της αντίστασης του αισθητήριου είναι 30 λεπτά μετά από αποθήκευση του αισθητήρα για 3 ημέρες σε καθαρό αέρα (εικόνα 3.8).



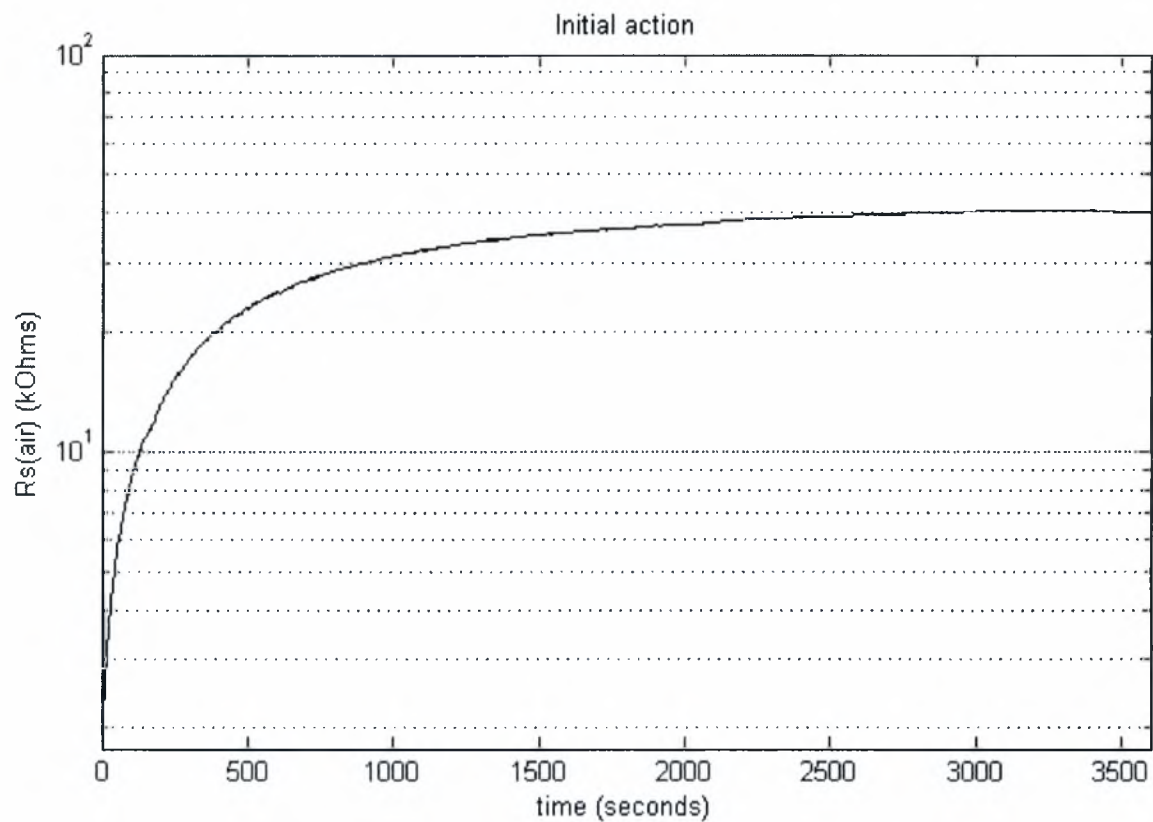
Εικόνα 3.8: Χρόνος που απαιτείται για σταθεροποίηση του TGS2444 μετά από αποθήκευση τριών ημερών σε καθαρό αέρα. Στοιχεία από Figaro.

Παρόλα αυτά, κατά την πορεία ανάπτυξης της κατασκευής που περιγράφεται στη συνέχεια, παρατηρήθηκε ότι στην πραγματικότητα αυτός ο χρόνος είναι μεγαλύτερος. Για αυτόν τον σκοπό έγιναν ορισμένες δοκιμές που συνοψίζονται στο διάγραμμα της εικόνας 3.9.

Οι συνθήκες των δοκιμών ήταν οι ίδιες με αυτές που αναφέρει η Figaro: Ο αισθητήρας, μετά από μερικές μέρες λειτουργίας, αποσυνδεόταν από το κύκλωμα και αποθηκευόταν σε μέρος με καθαρό αέρα. Μετά από 3 ημέρες τον τοποθετούσαμε πάλι στο κύκλωμα και τον λειτουργούσαμε σε καθαρό περιβάλλον. Ταυτόχρονα μετρούσαμε την τιμή της αντίστασής του μέχρι αυτή να σταθεροποιηθεί.

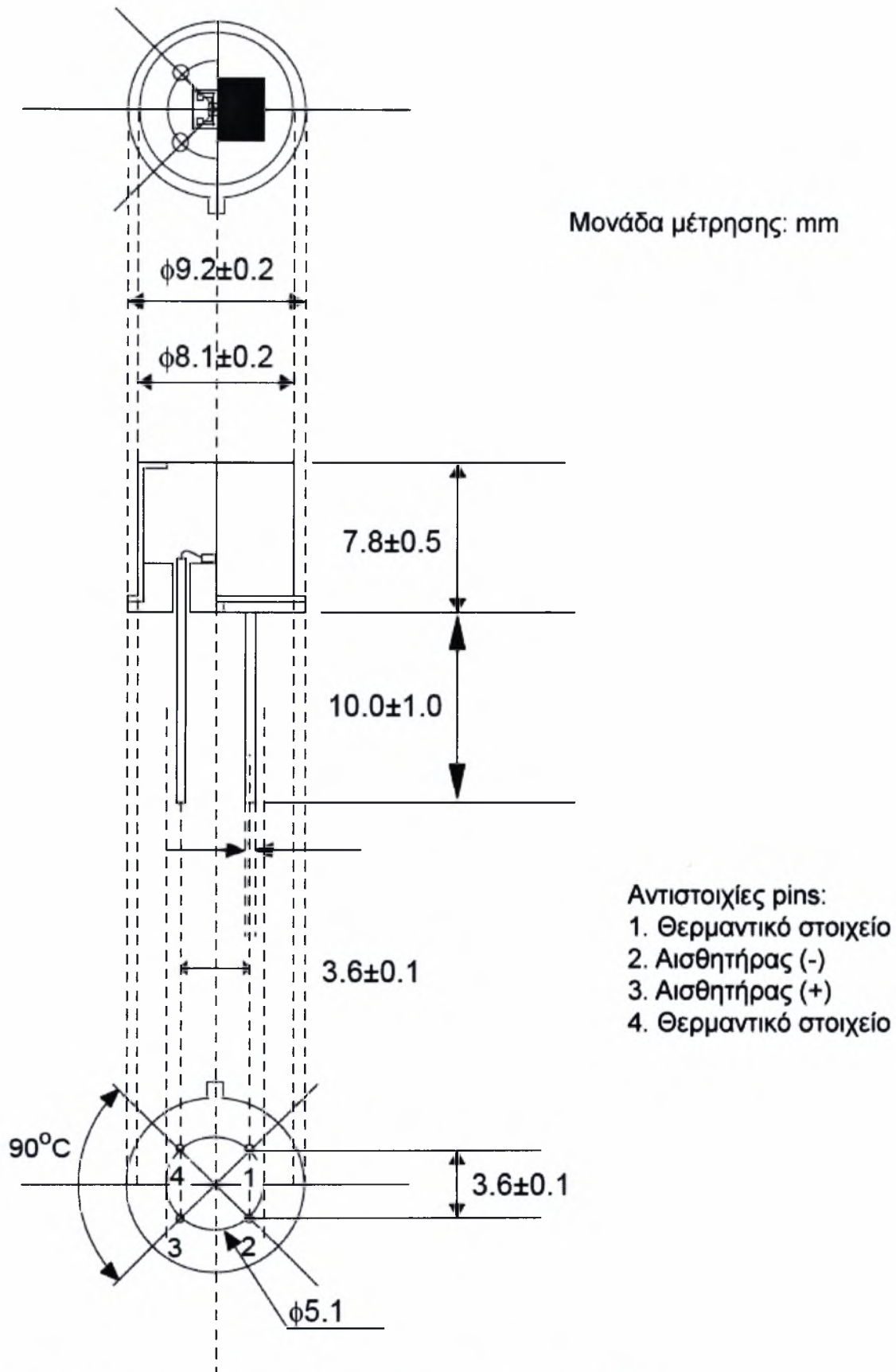
Από αυτές τις δοκιμές διαπιστώθηκε ότι ο χρόνος που χρειάζεται μέχρι να φτάσει σε κατάσταση που θα δίνει αξιόπιστες μετρήσεις είναι περίπου 50 λεπτά (σχεδόν διπλάσιος από αυτόν που δηλώνει η Figaro).

Αυτό είναι κάτι που πρέπει να λαμβάνεται υπ' όψιν κατά την λειτουργία του αισθητήρα. Όλες οι ενέργειές μας πρέπει να γίνονται μετά την πάροδο αυτού του χρονικού διαστήματος για να μην οδηγούμαστε σε λάθος συμπεράσματα.



Εικόνα 3.9: Χρόνος που απαιτείται για σταθεροποίηση του TGS2444 μετά από αποθήκευση τριών ημερών σε καθαρό αέρα. Στοιχεία από πειραματικές μετρήσεις.

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων



Εικόνα 3.10: Μορφή και φυσικές διαστάσεις αισθητήρα TGS-2444

3.2.3 Απαιτήσεις για το κύκλωμα διασύνδεσης του αισθητήρα

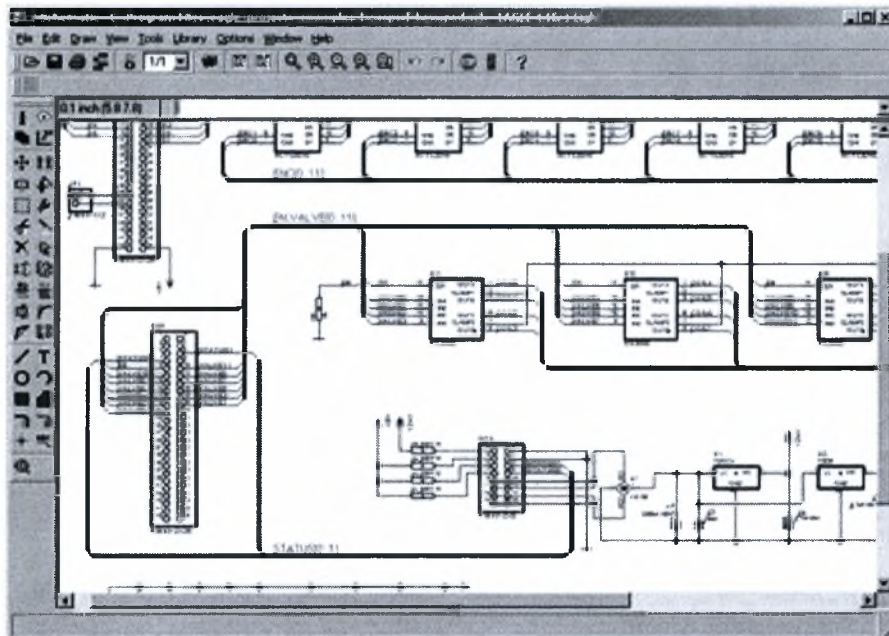
Το κύκλωμα που θα σχεδιάσουμε πρέπει να πληροί τις εξής προϋποθέσεις:

- Τήρηση με ακρίβεια του χρονισμού που ορίζει ο κατασκευαστής ώστε να έχουμε ακριβείς μετρήσεις και προστασία του αισθητήρα.
- Αναλογική έξοδο που αντιπροσωπεύει το πηλίκο R_s/R_o
- Δυνατότητα βαθμονόμησης με απλό τρόπο για σωστές μετρήσεις ακόμα και μετά από αλλαγή του αισθητήρα ή μακροχρόνια χρήση (γήρανση).
- Αποθήκευση των δεδομένων της βαθμονόμησης σε μνήμη EEPROM για διατήρησή τους ακόμα και όταν η κατασκευή είναι εκτός λειτουργίας.
- Θύρα για προαιρετική σύνδεση LCD οθόνης. Η οθόνη θα παρέχει περισσότερες πληροφορίες κατά την βαθμονόμηση και την κανονική λειτουργία ώστε να είναι δυνατόν να διαβάσουμε μετρήσεις ακόμα και όταν δεν έχει συνδεθεί σε κάποιον wireless sensor node.
- Ευρύ φάσμα τάσεων τροφοδοσίας
- Πρόβλεψη για εύκολο τρόπο επαναπρογραμματισμού του μικροελεγκτή με νεότερες εκδόσεις του λογισμικού του.

4 Εργαλεία σχεδίασης και ανάπτυξης

4.1 Σχεδίαση ηλεκτρονικού κυκλώματος

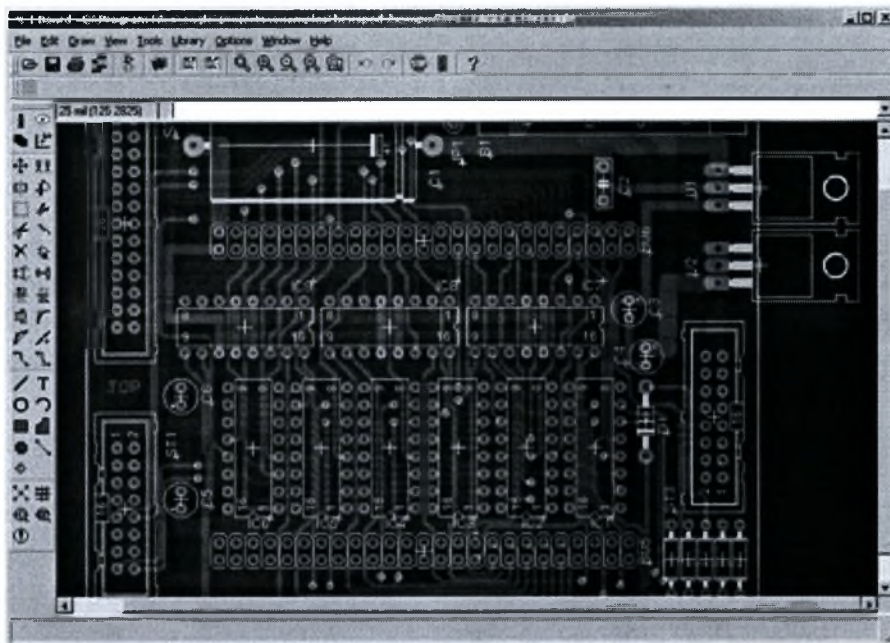
Για κατασκευή των σχηματικών και των σχεδίων για τις τυπωμένες πλακέτες θα χρειαστούμε κάποιο σχεδιαστικό εργαλείο. Ένα τέτοιο πολύ καλό στο είδος του και διαθέσιμο δωρεάν είναι το EAGLE [3] (Easily Applicable Graphical Layout Editor) της Γερμανικής εταιρείας CadSoft. Διατίθεται σε εκδόσεις για Windows, Linux και MacOS X.



Εικόνα 4.1: Το περιβάλλον του EAGLE για σχεδίαση σχηματικού

Το EAGLE παρέχει ένα περιβάλλον για κατασκευή σχηματικών το οποίο διασυνδέεται με το εργαλείο σχεδιασμού τυπωμένων πλακετών. Τα εξαρτήματα τοποθετούνται χειροκίνητα στην πλακέτα με τη βοήθεια χρωματιστών γραμμών που δείχνουν τις διασυνδέσεις μεταξύ τους βάσει του σχηματικού διαγράμματος, βοηθώντας έτσι στην εξεύρεση της βέλτιστης θέσης ώστε να χαρακτηθούν αποδοτικότερα οι χαλκοδιάδρομοι.

Παρέχει επίσης και έναν πολύ καλό autorouter, ένα εργαλείο αυτόματης χάραξης χαλκοδιάδρομων. Δεν καταφέρνει βέβαια πάντα να κάνει την βέλτιστη χάραξη για αυτό επιτρέπει πρώτα την χειροκίνητη χάραξη ορισμένων κρίσιμων σημάτων όπως τροφοδοσίας και υψηλών συχνοτήτων, ώστε στη συνέχεια να αναλάβει ο autorouter.



Εικόνα 4.2: Το περιβάλλον του EAGLE για σχεδίαση πλακέτας

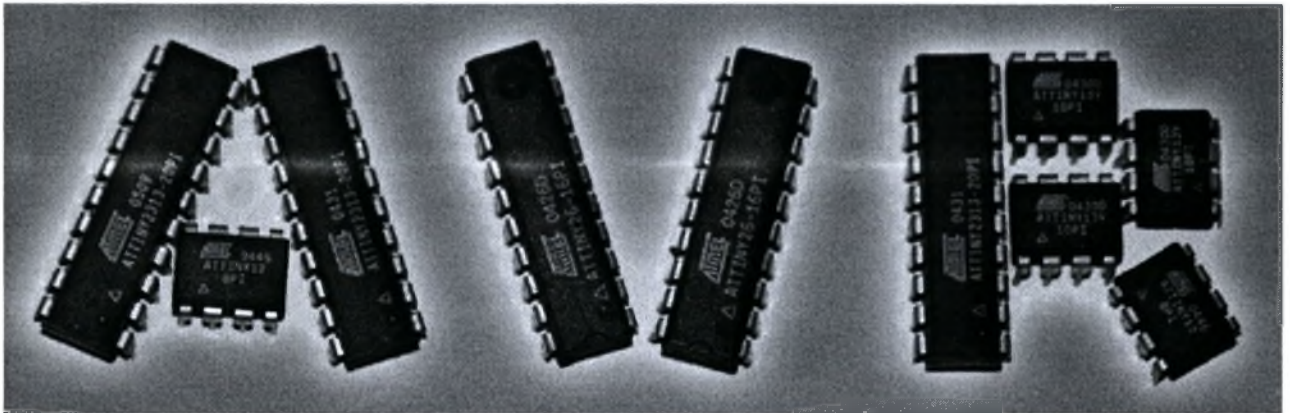
Η ελεύθερη έκδοση του EAGLE θέτει κάποιους περιορισμούς: μόνο 2 επίπεδα χαλκού και ένα μέγιστο μέγεθος 80x100mm αλλά για τους σκοπούς που θα χρησιμοποιηθεί στην παρούσα εργασία είναι αρκετό.

4.2 Προγραμματισμός μικροελεγκτή

4.2.1 Παρουσίαση μικροελεγκτών AVR

Η κατασκευή που θα αναπτυχθεί για τον έλεγχο του αισθητήρα θα ελέγχεται από έναν AVR μικροελεγκτή της Atmel.

Ο AVR είναι ένας 8-bit μικροεπεξεργαστής RISC αρχιτεκτονικής Harvard που αναπτύχθηκε από την Atmel το 1996. Η αρχιτεκτονική Harvard αναφέρεται στο γεγονός ότι η CPU διαθέτει ξεχωριστή μνήμη για το πρόγραμμα και για τα δεδομένα. Η οικογένεια των AVR ήταν από τις πρώτες οικογένειες μικροελεγκτών που είχαν ενσωματωμένη μέσα στο chip μνήμη flash για αποθήκευση του προγράμματος, σε αντίθεση με τις προγραμματιζόμενες μνήμες μιας εγγραφής (ROM, EPROM ή EEPROM) που είχαν οι περισσότεροι μικροελεγκτές εκείνη την εποχή.



Εικόνα 4.3: Διάφοροι τύποι μικροελεγκτών AVR

Τα βασικά χαρακτηριστικά της οικογένειας των AVR μικροελεγκτών είναι τα παρακάτω:

- Ενσωματωμένη μνήμη Flash, με δυνατότητα για In System Programming (ISP) που χρησιμοποιείται για την αποθήκευση του προγράμματος. Αυτό σημαίνει ότι δεν χρειάζεται η κατασκευή να έχει εξωτερική EPROM ή ROM όπου θα είναι αποθηκευμένο το εκτελέσιμο πρόγραμμα του μικροελεγκτή. Επίσης η μνήμη αυτή μπορεί να προγραμματιστεί χωρίς να χρειαστεί να τον αποσυνδέσουμε από την πλακέτα της κατασκευής. Ο προγραμματισμός της μνήμης μπορεί να γίνει με δύο τρόπους: σειριακά και παράλληλα.
- 32x8bit καταχωρητές γενικού σκοπού
- Ενσωματωμένη μνήμη EEPROM για αποθήκευση δεδομένων και RAM για τις μεταβλητές κατά την εκτέλεση του λογισμικού.
- Συχνότητα ρολογιού από 0 – 40MHz. Οι περισσότερες εντολές εκτελούνται σε ένα κύκλο ρολογιού με αποτέλεσμα βελτίωση των επιδόσεων μέχρι και 10 φορές σε σχέση με άλλους μικροελεγκτές (πχ 8051) που λειτουργούν στην ίδια συχνότητα.

Οι AVR σε γενικές γραμμές διαχωρίζονται σε τρεις ομάδες:

- **tinyAVR**
 - 1 – 8 kB μνήμης προγράμματος
 - 8 – 32 ακροδέκτες στο σώμα chip
 - περιορισμένο πλήθος περιφερειακών

- **megaAVR**
 - 4 – 256 kB μνήμης προγράμματος
 - 28 – 100 ακροδέκτες στο σώμα του chip
 - Διευρυμένο σετ εντολών
 - Διευρυμένο αριθμό περιφερειακών

- **AVR συγκεκριμένων εφαρμογών**
 - πρόκειται για megaAVR με συγκεκριμένα χαρακτηριστικά που δεν τα βρίσκουμε σε άλλους AVR. Για παράδειγμα ορισμένοι από αυτούς διαθέτουν ελεγκτές LCD, ελεγκτές USB, βελτιωμένο PWM κλπ
 - FPSLIC (Field Programmable System Level Integrated Circuit), ένας πυρήνας AVR μαζί με ένα FPGA μέσα στο ίδιο chip. Το FPSLIC χρησιμοποιεί SRAM για το πρόγραμμα του AVR, σε αντίθεση με όλους τους άλλους AVR. Η αυξημένη ταχύτητα πρόσβασης στις SRAM σε σχέση με τις μνήμες flash δίνει την δυνατότητα στον AVR πυρήνα του FPSLIC να λειτουργεί με ταχύτητες μέχρι και 50MHz.

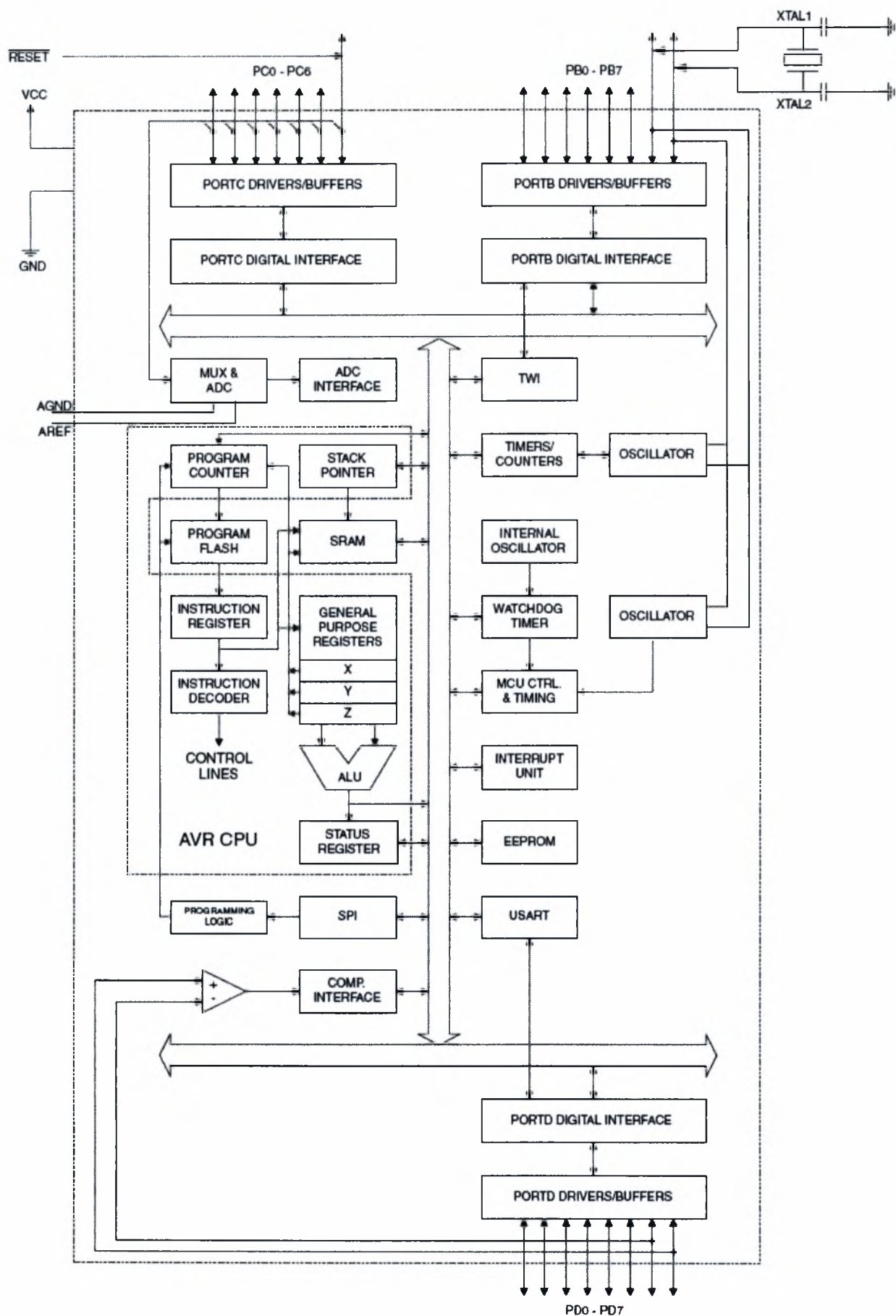
Για τους σκοπούς αυτής της εργασίας επιλέχθηκε ένας μικροελεγκτής megaAVR, και πιο συγκεκριμένα ο ATmega8.

4.2.2 Τεχνικά χαρακτηριστικά ATmega8

- 8-bit AVR μικροελεγκτής
- Αρχιτεκτονική RISC
 - 130 εντολές, οι περισσότερες απαιτούν 1 κύκλο για εκτέλεση
 - 32 x 8 καταχωρητές γενικού σκοπού
 - Έως 16MIPS ταχύτητα στα 16MHz
- Υψηλής αντοχής non-volatile τμήματα μνήμης
 - 8KBytes In System Programmable μνήμης Flash
 - 512Bytes EEPROM
 - 1KByte SRAM
 - Κύκλοι εγγραφής/ανάγνωσης: 10.000 για Flash και 100.000 για EEPROM
 - Αποθήκευση δεδομένων: 20 χρόνια στους 85°C / 100 χρόνια στους 25°C
 - Προαιρετικό τμήμα με boot code, In System προγραμματισμός από το boot πρόγραμμα
 - Κλείδωμα για προστασία λογισμικού
- Χαρακτηριστικά περιφερειακών
 - Δύο 8-bit χρονιστές/μετρητές (timer/counters) με ξεχωριστούς διαιρέτες συχνότητας
 - Ένα 16-bit χρονιστή/μετρητή (timer/counter) με ξεχωριστό διαιρέτη συχνότητας
 - Μετρητής πραγματικού χρόνου με ξεχωριστό ταλαντωτή
 - 3 κανάλια PWM (Pulse Width Modulation)
 - 6 κανάλια ADC (Analog/Digital Converter) με 10-bit ακρίβεια
 - Σειριακή διασύνδεση 2 αγωγών
 - Προγραμματιζόμενο σειριακό USART (Universal Asynchronous Receiver/Transmitter)
 - Διεπαφή master/slave για σειριακό SPI (Serial Peripheral Interface bus)
 - Προγραμματιζόμενο χρονιστή watchdog με ενσωματωμένο ξεχωριστό ταλαντωτή
 - Ενσωματωμένο αναλογικό συγκριτή
- Ειδικές δυνατότητες
 - Power-on Reset και προγραμματιζόμενη ανίχνευση Brown-out
 - Εσωτερικό ρυθμισμένο RC ταλαντωτή
 - Εξωτερικές και εσωτερικές πηγές interrupts
 - Πέντε επίπεδα sleep: Idle, λειτουργία μείωσης θορύβου ADC, Power-save, Power-down και Standby
- Γραμμές εισόδου/εξόδου
 - 23 προγραμματιζόμενες γραμμές I/O
- Τάση λειτουργίας
 - 2.7V – 5.5V για τον ATmega8L
 - 4.8V – 5.5V για τον ATmega8
- Συχνότητα λειτουργίας
 - 0 – 8 MHz για τον ATmega8L
 - 0 – 16 MHz για τον ATmega8
- Κατανάλωση ενέργειας στα 4MHz, 3V, 25°C
 - Active: 3.6mA
 - Idle mode: 1.0mA
 - Power down mode: 0.5μA

Η πλήρης και αναλυτική περιγραφή όλων των τεχνικών χαρακτηριστικών και των υπομονάδων του Atmega8 υπάρχει στο datasheet του [4].

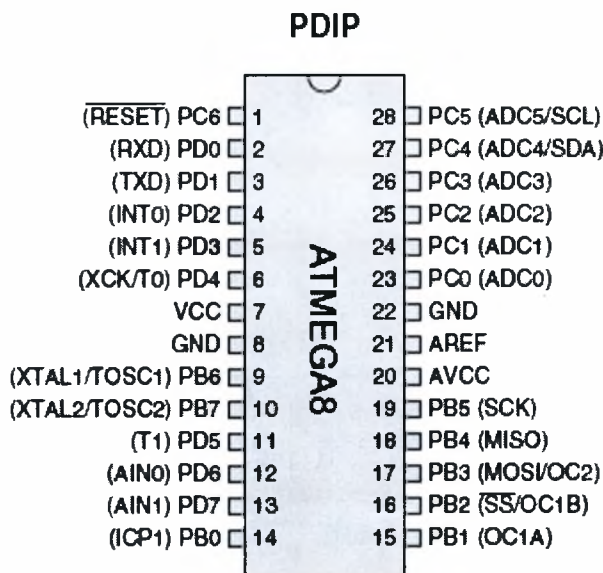
Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων



Εικόνα 4.4: Block διάγραμμα του ATmega8

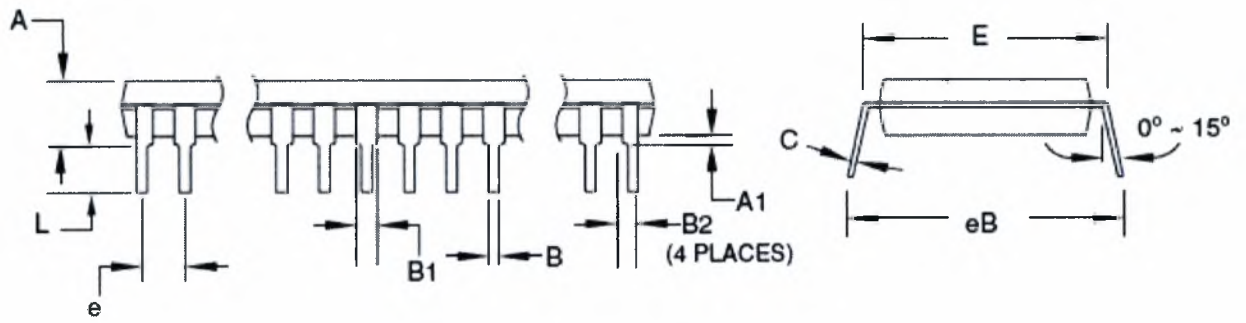
Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Η διάταξη των ακροδεκτών του μικροελεγκτή που θα χρησιμοποιήσουμε φαίνεται στην εικόνα 4.5. Είναι σε συσκευασία (PDIP) Plastic Dual Inline Package (εικόνα 4.6).

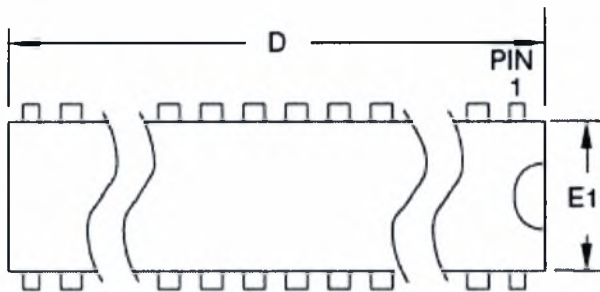


Εικόνα 4.5: Διάταξη ακροδεκτών ATmega8 σε συσκευασία PDIP

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων



COMMON DIMENSIONS
(Unit of Measure = mm)



SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	4.5724	
A1	0.508	-	-	
D	34.544	-	34.798	Note 1
E	7.620	-	8.255	
E1	7.112	-	7.493	Note 1
B	0.381	-	0.533	
B1	1.143	-	1.397	
B2	0.762	-	1.143	
L	3.175	-	3.429	
C	0.203	-	0.356	
eB	-	-	10.160	
e	2.540 TYP			

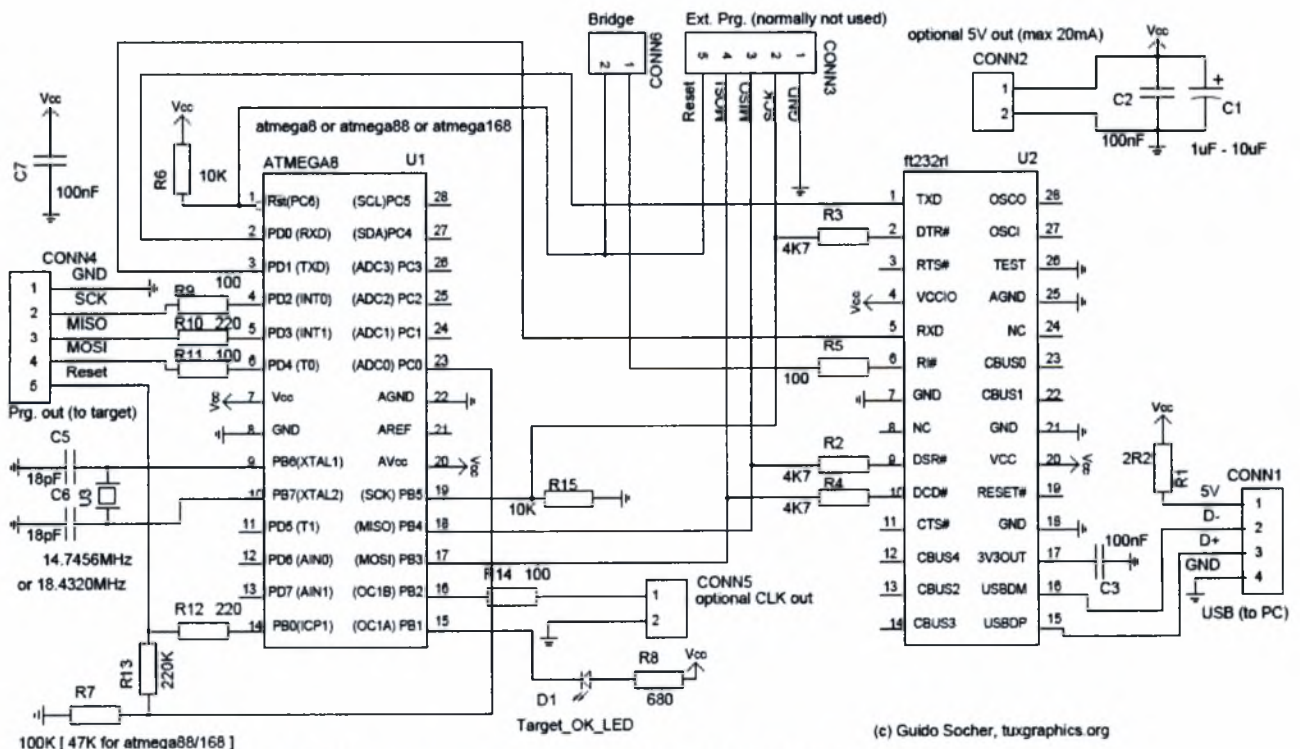
Εικόνα 4.6: Συσκευασία 28 επαφών Plastic Dual In-line Package (PDIP) 7.62mm

4.2.3 Κατασκευή προγραμματιστή

Για τον προγραμματισμό του μικροελεγκτή θα χρησιμοποιήσουμε την μέθοδο ISP (In-System Programming) που μας επιτρέπει τον τον προγραμματισμό του χωρίς να χρειάζεται να τον αποσυνδέσουμε από την πλακέτα της κατασκευής. Με αυτό τον τρόπο μπορούμε να δοκιμάζουμε πολύ πιο εύκολα και γρήγορα τον κώδικα σε κάθε φάση της ανάπτυξής του.

Η Atmel έχει κατασκευάσει και διαθέτει προς πώληση μια πλακέτα κατάλληλη για ανάπτυξη εφαρμογών βασισμένων σε AVR και προγραμματισμό μικροελεγκτών. Η πλακέτα αυτή έχει την ονομασία STK500 και γι αυτό το λόγο συχνά χρησιμοποιείται το ίδιο όνομα και για το πρωτόκολλο επικοινωνίας του προγραμματιστή με τον υπολογιστή. Η πρώτη έκδοση του πρωτοκόλλου αυτού (STK500 v1) ήταν εξαρτώμενη σε μεγάλο βαθμό από τον εκάστοτε προγραμματιστή με αποτέλεσμα να χρειάζεται συνεχείς ενημερώσεις κάθε φορά που αυτός άλλαζε. Τελικώς διατέθηκε η δεύτερη έκδοση του πρωτοκόλλου το οποίο περιέγραφε μερικές πιο γενικές διαδικασίες με αποτέλεσμα να υποστηρίζει ποικιλία προγραμματιστών. Ονομάστηκε STK500 v2 και η ακριβής προδιαγραφή περιγράφεται στο AVR068 [5]. Υπάρχει επίσης και μια νεότερη έκδοσή της με την ονομασία AVR060 [6]. Με το STK500 AVR068 δεν χρειάζεται πλέον να υπάρχουν συγκεκριμένες δομές δεδομένων στις συσκευές καθώς όλες οι ενέργειες που εξαρτώνται από τον εκάστοτε προγραμματιστή γίνονται σε επίπεδο λογισμικού στο πρόγραμμα που τρέχει στον υπολογιστή.

Αντί να αγοράσουμε κάποιον έτοιμο προγραμματιστή, προτιμήθηκε η κατασκευή κάποιου που να είναι ελεύθερα διαθέσιμος. Ο προγραμματιστής που επιλέχθηκε είναι ο AvrUsb500v2 [7] του οποίου το σχέδιο και το λογισμικό διανέμονται υπό άδεια GPLv2. Είναι ένας In-System Programmer, συμβατός με το πρωτόκολλο STK500 v2 και η φυσική διασύνδεσή του με τον υπολογιστή γίνεται μέσω USB.

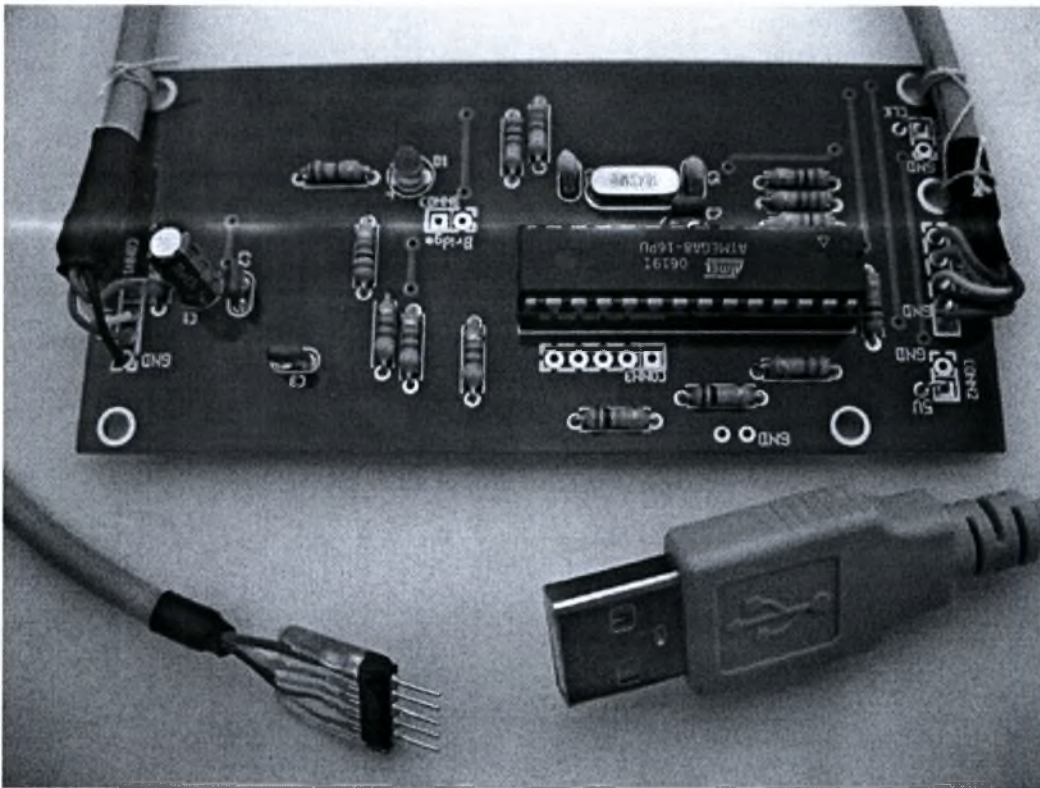


Εικόνα 4.7: Σχηματικό διάγραμμα προγραμματιστή AvrUsb500v2

Παρατηρώντας το σχηματικό του (εικόνα 4.7) βλέπει κανείς ότι ο προγραμματιστής βασίζεται και ο ίδιος σε έναν ATmega μικροελεγκτή ο οποίος και υλοποιεί σε software την επικοινωνία μέσω SPI (Serial Peripheral Interface) με την προγραμματιζόμενη συσκευή. Αυτό μοιάζει να είναι αδιέξοδο εφόσον για να λειτουργήσει ο προγραμματιστής χρειάζεται έναν μικροελεγκτή από αυτούς που καλείται να προγραμματίζει, αλλά υπάρχει λύση.

Για την επικοινωνία μέσω USB χρησιμοποιείται το chip FT232r1 [8] της εταιρείας FTDI. Αυτό το ολοκληρωμένο έχει μια λειτουργία που λέγεται bit-bang mode και μας επιτρέπει να μεταφέρουμε bit προς bit το αρχικό πρόγραμμα από το PC προς τον μικροελεγκτή του προγραμματιστή.

Αυτή η διαδικασία είναι πολύ αργή διότι, παρόλο που η USB επικοινωνία είναι γρήγορη κατά την μεταφορά μεγάλων τμημάτων δεδομένων, όταν στέλνουμε 1 bit κάθε φορά καθυστερεί πάρα πολύ. Αυτό όμως δεν αποτελεί πρόβλημα γιατί η καθυστέρηση αυτή θα υπάρχει μόνο μέχρι να προγραμματίσουμε τον μικροελεγκτή του προγραμματιστή. Από εκεί και έπειτα η επικοινωνία μέσω USB θα γίνεται σε γρήγορους ρυθμούς.

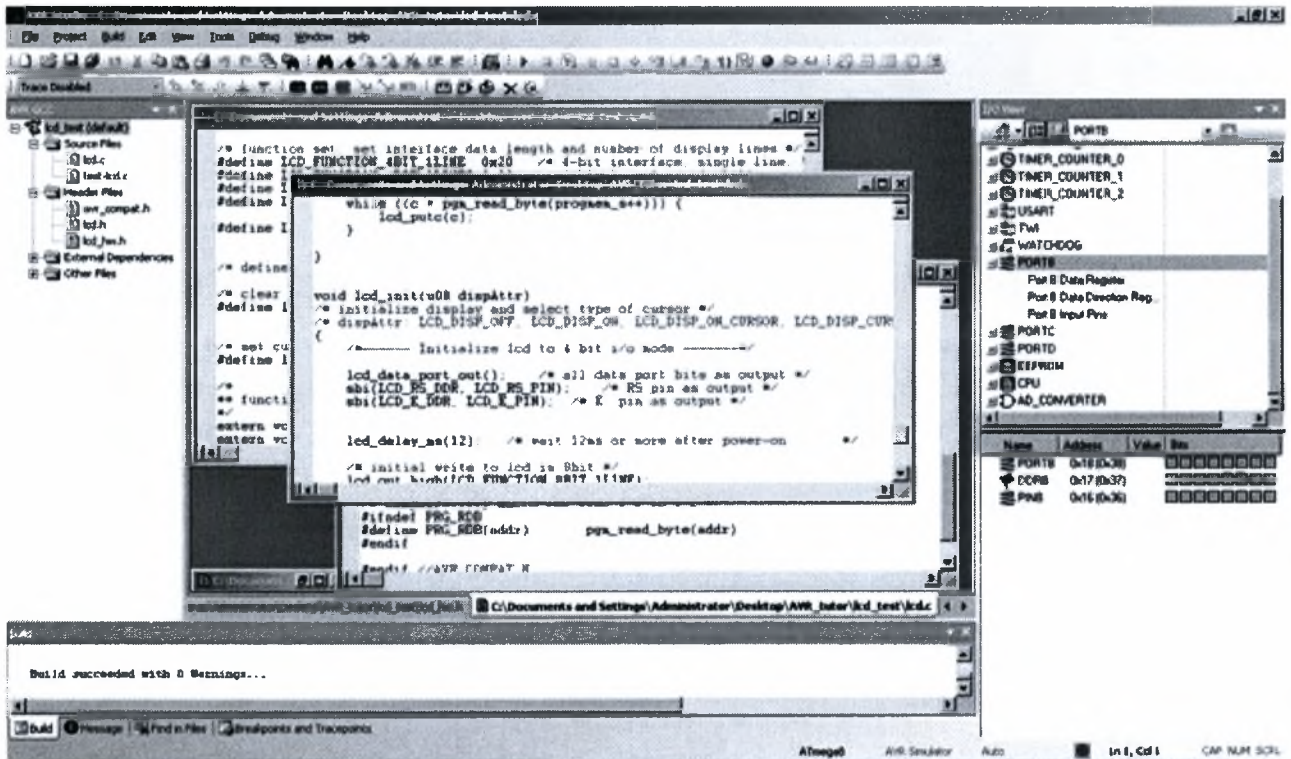


Εικόνα 4.8: Ο προγραμματιστής ολοκληρωμένος και έτοιμος προς χρήση

4.2.4 Περιβάλλον προγραμματισμού

Η Atmel προσφέρει το AVR Studio [9], ένα γραφικό περιβάλλον ανάπτυξης IDE (Integrated Development Environment) για έλεγχο, προγραμματισμό και debugging για τους AVR. Ο compiler που ενσωματώνει είναι για assembly αλλά το AVR Studio είναι έτσι φτιαγμένο ώστε να μπορεί να συνεργάζεται και με άλλους compilers/assemblers. Έτσι μπορεί κανείς να χρησιμοποιήσει γλώσσα C, Basic ή Pascal.

Υποστηρίζει όλους τους μικροελεγκτές AVR καθώς και πολλές αναπτυξιακές πλατφόρμες και εργαλεία προσομοίωσης.



Εικόνα 4.9: Το περιβάλλον εργασίας του AVR Studio

Για τον προγραμματισμό σε γλώσσα C θα χρησιμοποιήσουμε το WinAVR [10], μια σουίτα αναπτυξιακών εργαλείων ανοικτού κώδικα για την οικογένεια των AVR. Ενσωματώνει τον GNU GCC (avr-gcc) compiler για C και C++, τον προγραμματιστή avrdude και τον debugger avr-gdb.

5 Σχεδιασμός ηλεκτρονικής κατασκευής

5.1 Κύκλωμα διασύνδεσης αισθητήρα

Όπως είδαμε στο κεφάλαιο της περιγραφής του αισθητήρα, για να λειτουργήσει απαιτεί την εφαρμογή παλμών τροφοδοσίας 5V και διάρκειας μερικών milliseconds.

Οι έξοδοι του μικροελεγκτή έχουν την δυνατότητα να παρέχουν τάση ίση με την τάση τροφοδοσίας και ρεύμα μερικών mA. Πιο συγκεκριμένα, σύμφωνα με τα τεχνικά χαρακτηριστικά του Atmega8, η κάθε έξοδος του δίνει μέχρι 10mA.

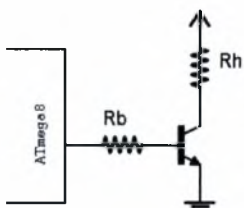
Αυτό σημαίνει ότι δεν μπορούμε να συνδέσουμε απευθείας στις εξόδους του κυκλώματα που απαιτούν μεγαλύτερο ρεύμα. Για να μπορεί να διαχειρίζεται μεγαλύτερα φορτία θα πρέπει να παρεμβάλλουμε κάποιο κύκλωμα με transistor που να λειτουργεί ως διακόπτης.

Η επιλογή του κατάλληλου transistor για τις ανάγκες της κατασκευής απαιτεί τον προσδιορισμό των εξής παραμέτρων του:

- Μέγιστο ρεύμα συλλέκτη $I_{C(max)}$
Είναι το μέγιστο ρεύμα που θα τροφοδοτεί το φορτίο.
- Μέγιστη τάση λειτουργίας $V_{CE(max)}$
Είναι η μέγιστη τάση που παρουσιάζεται στο transistor μεταξύ συλλέκτη και εκπομπού όταν αυτό βρίσκεται σε αποκοπή. Στην περίπτωση του κυκλώματός μας είναι ίση με την τάση τροφοδοσίας, δηλαδή 5V.
- Ελάχιστο απαιτούμενο κέρδος $h_{FE(min)}$
Θα πρέπει να είναι τουλάχιστον 5 φορές μεγαλύτερο από το πηλίκο του ρεύματος φορτίου δια το μέγιστο ρεύμα που μπορεί να μας δώσει σε κάποια έξοδό του ο μικροελεγκτής.

Δηλαδή:
$$h_{FE(min)} > 5 \times \frac{\text{ρεύμα φορτίου} (I_C)}{\text{μέγιστο ρεύμα μικροελεγκτή} (I_B)}$$

Κύκλωμα θερμαντικού στοιχείου



Το θερμαντικό στοιχείο του αισθητήρα απαιτεί ρεύμα 203mA.
Άρα πρέπει: $I_C > 203mA$

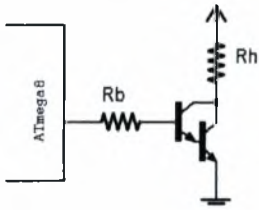
Επίσης προκύπτει ότι για ρεύμα φορτίου $I_C = 203mA$ χρειαζόμαστε:

$$h_{FE(min)} > 5 \times \frac{203mA}{10mA} \Rightarrow h_{FE(min)} > 101,5$$

Ψάχνοντας με αυτά τα στοιχεία θα δούμε ότι δεν υπάρχει κάποιο απλό NPN transistor που να έχει τόσο μεγάλο $h_{FE(min)}$ και παράλληλα να μπορεί να αντέχει στο I_C που θέλουμε. Αυτό σημαίνει ότι θα πρέπει να αλλάξουμε την σχεδίασή μας.

Ένας τρόπος είναι να χρησιμοποιήσουμε transistor σε συνδεσμολογία Darlington. Με αυτόν τον τρόπο πετυχαίνουμε πολύ μεγάλο κέρδος και ταυτόχρονα μπορούμε να διαχειριστούμε μεγαλύτερο ρεύμα στον συλλέκτη.

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων



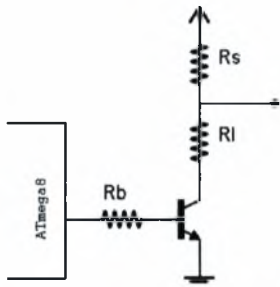
Το μόνο μειονέκτημα αυτής της συνδεσμολογίας είναι ότι για να φτάσει στον κορεσμό απαιτείται η διπλάσια τάση μεταξύ βάσης-εκπομπού (περίπου 1,4V) σε σχέση με το αν είχαμε ένα απλό transistor. Αυτό όμως δεν αποτελεί πρόβλημα καθώς ο μικροελεγκτής δίνει 5V στις εξόδους του.

Βάσει αυτών των κριτηρίων επιλέχθηκε το **BC517** NPN Darlington transistor με $I_C = 500\text{mA}$ και $h_{FE(\min)} = 30000$.

Η αντίσταση R_b συνδέεται στην έξοδο του μικροελεγκτή. Ο σκοπός της είναι να περιορίζει το ρεύμα που ρέει προς την βάση του transistor.

$$\text{Η τιμή της πρέπει να ισούται με: } R_b = \frac{V_C \times h_{FE}}{5 \times I_C} \Rightarrow R_b = \frac{5\text{V} \times 30000}{5 \times 203\text{mA}} \Rightarrow R_b = 147,8\text{k}\Omega$$

Κύκλωμα αισθητήριου στοιχείου



Το αισθητήριο στοιχείο R_s συνδέεται σε σειρά με μια εξωτερική αντίσταση R_L σχηματίζοντας έτσι έναν διαιρέτη τάσης.

Σύμφωνα με το datasheet του TGS2444, η τιμή της R_s κυμαίνεται μεταξύ 0,063kΩ και 0,63kΩ στα 10ppm και ελαττώνεται σε μεγαλύτερες συγκεντρώσεις. Επίσης πρέπει $R_L > 8\text{k}\Omega$.

Για τον υπολογισμό του μέγιστου ρεύματος του συλλέκτη θα θεωρήσουμε την μικρότερη τιμή της R_s και $R_L = 20\text{k}\Omega$.

$$I_C = \frac{V_C}{R_s + R_L} \Rightarrow I_C = \frac{5\text{V}}{0,063\text{k}\Omega + 20\text{k}\Omega} \Rightarrow I_C \approx 0,3\text{mA}$$

$$\text{Άρα: } h_{FE(\min)} > 5 \times \frac{0,3\text{mA}}{10\text{mA}} \Rightarrow h_{FE(\min)} > 0,15 \text{ δηλαδή θέλουμε πολύ μικρό κέρδος.}$$

Σχεδόν όλα τα transistor γενικού σκοπού ικανοποιούν αυτά τα χαρακτηριστικά.

Από αυτά επιλέγουμε το **BC237** NPN transistor με $I_C = 100\text{mA}$ και $h_{FE(\min)} = 200$.

$$\text{Τέλος υπολογίζουμε και την } R_b = \frac{V_C \times h_{FE}}{5 \times I_C} \Rightarrow R_b = \frac{5\text{V} \times 200}{5 \times 0,3\text{mA}} \Rightarrow R_b \approx 667\text{k}\Omega$$

5.2 Αναλογική έξοδος μετρούμενης τιμής

Ο μικροελεγκτής θα υπολογίζει και θα δίνει ως αναλογική έξοδο τάσης την τιμή της μεταβολής της αντίστασης του αισθητήρα.

Πιο συγκεκριμένα θα υπολογίζει το πηλίκο:

$$\frac{R_s}{R_o} = \frac{\text{αντίσταση αισθητήρα}}{\text{αντίσταση αισθητήρα σε καθαρό αέρα}} \quad \text{με} \quad \frac{R_s}{R_o} \leq 1$$

και θα μας επιστρέφει μια τάση V_{out} από 0 μέχρι 2,5V δηλαδή: $V_{out} = \frac{R_s}{R_o} \times 2,5V$

Για την δημιουργία της αναλογικής εξόδου θα χρησιμοποιηθεί διαμόρφωση πλάτους παλμού (pulse width modulation, PWM). Θα χρησιμοποιήσουμε δηλαδή παλμούς σταθερής συχνότητας αλλά μεταβλητού πλάτους για να δημιουργήσουμε την επιθυμητή τάση.

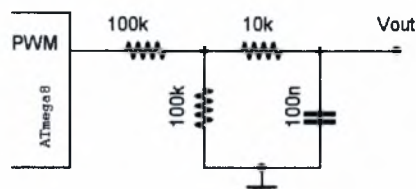
Ο Atmega8 έχει τη δυνατότητα να παράγει τέτοιους παλμούς τάσης 5V και υψηλής συχνότητας τους οποίους στην συνέχεια μπορούμε να φιλτράρουμε με ένα χαμηλοπερατό φίλτρο παίρνοντας έτσι την ζητούμενη σταθερή τάση εξόδου. Η ακρίβεια που μπορούμε να πετύχουμε με αυτόν τον τρόπο είναι 8bit. Έχουμε δηλαδή 256 διαβαθμίσεις μεταξύ 0 και 5V.

Προκειμένου να επιστρέφουμε τιμές μέχρι 2,5V έχουμε δύο επιλογές: Είτε μέσω μικροελεγκτή ρίχνοντας την ανάλυση στα 7bit, είτε με κάποιο αναλογικό κύκλωμα ώστε να εξακολουθήσουμε να έχουμε την μέγιστη δυνατή ανάλυση. Ο τρόπος που τελικά επιλέχθηκε είναι ο δεύτερος και υλοποιήθηκε με ένα δικτύωμα δύο αντιστάσεων που δημιουργούν έναν διαιρέτη τάσης.

Η συχνότητα του PWM που θα υλοποιηθεί στο πρόγραμμα του μικροελεγκτή είναι 31,25kHz οπότε θέλουμε ένα φίλτρο που να την αποκόπτει. Θα χρησιμοποιήσουμε ένα RC φίλτρο με $R=16k\Omega$ και $C=100nF$ το οποίο έχει συχνότητα αποκοπής:

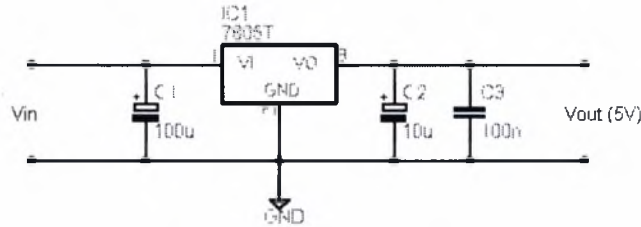
$$f_c = \frac{1}{2 \times \pi \times R \times C} \Rightarrow f_c = \frac{1}{2 \times \pi \times 1,6 \cdot 10^3 \times 10^{-6}} \Rightarrow f_c = 99,47 \text{ Hz}$$

Το τελικό κύκλωμα που προκύπτει για την δημιουργία της τάσης εξόδου φαίνεται παρακάτω.



5.3 Κύκλωμα τροφοδοσίας

Η τάση λειτουργίας του αισθητήρα και του μικροελεγκτή είναι 5V. Προκειμένου να λαμβάνουν πάντα την σωστή τάση, θα προσθέσουμε στην κατασκευή ένα voltage regulator. Επιπλέον, επειδή η κατασκευή μας χρησιμοποιεί παλμούς ρεύματος, πρέπει να δοθεί ιδιαίτερη μέριμνα στην όσο το δυνατόν αποτελεσματικότερη εξάλειψή τους για να μπορούμε να έχουμε σωστές μετρήσεις.



Ως voltage regulator επιλέχθηκε το ολοκληρωμένο LM7805 [11] που έχει τη δυνατότητα να παρέχει σταθεροποιημένη τάση 5V και μέχρι 1A. Επίσης διαθέτει κυκλώματα προστασίας από υπερφόρτωση και υπερθέρμανση.

Για την προστασία από απότομες διακυμάνσεις της τάσης υπάρχουν οι κατάλληλοι πυκνωτές τόσο κοντά στο LM7805 όσο και στην τροφοδοσία του μικροελεγκτή.

5.4 Σύνδεση οθόνης

Η κατασκευή θα είναι έτσι φτιαγμένη ώστε να μπορεί προαιρετικά να συνδεθεί σε αυτήν κάποια οθόνη υγρών κρυστάλλων (liquid crystal display, LCD) συμβατή με το πρότυπο HD44780 [12]. Αυτές οι οθόνες μπορούν να εμφανίζουν αλφαριθμητικούς χαρακτήρες και συνήθως τις βρίσκει κανείς σε συγκεκριμένα μεγέθη 8x1, 16x2, και 20x4 χαρακτήρων. Μπορεί να είναι με οπίσθιο φωτισμό, είτε χωρίς.



Εικόνα 5.1: LCD οθόνη 16x2 χαρακτήρων με ελεγκτή HD44780

Όλα τα απαραίτητα ηλεκτρονικά κυκλώματα για την λειτουργία τους (RAM, γεννήτρια χαρακτήρων, ελεγκτής LCD) είναι ενσωματωμένα στην οθόνη οπότε για την λειτουργία τους δεν απαιτείται κάτι επιπλέον. Η επικοινωνία, για αποστολή εντολών και χαρακτήρων προς εμφάνιση, γίνεται μέσω ενός διαύλου 14 καναλιών (με 8 data bits). Αν επιπλέον υπάρχει και φωτισμός χρησιμοποιούνται συνολικά 16 επαφές.

Η επεξήγηση της λειτουργίας της κάθε μιας από αυτές τις επαφές φαίνεται στον πίνακα 5.1.

pin #	Όνομασία και περιγραφή σήματος	pin #	Όνομασία και περιγραφή σήματος
1	VSS τροφοδοσία οθόνης (GND)	9	DB2 γραμμή δεδομένων 2
2	VCC τροφοδοσία οθόνης (+5V)	10	DB3 γραμμή δεδομένων 3
3	VEE ρύθμιση αντίθεσης	11	DB4 γραμμή δεδομένων 4
4	RS 0 = εισαγωγή εντολών 1 = εισαγωγή δεδομένων	12	DB5 γραμμή δεδομένων 5
5	R/W 0 = εγγραφή σε LCD 1 = ανάγνωση από LCD	13	DB6 γραμμή δεδομένων 6
6	E σήμα ενεργοποίησης	14	DB7 γραμμή δεδομένων 7 (MSB)
7	DB0 γραμμή δεδομένων 0 (LSB)	15	LED - τροφοδοσία φωτισμού (GND)
8	DB1 γραμμή δεδομένων 1	16	LED + τροφοδοσία φωτισμού (+5V)

Πίνακας 5.1: Επεξήγηση επαφών διασύνδεσης για LCD οθόνη συμβατή με HD44780

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Οι οθόνες αυτές εκτός από την λειτουργία σε 8bit mode μπορούν να λειτουργήσουν και με 4bit, στέλνοντας κάθε φορά μόνο μισό byte στις γραμμές DB4 έως DB7. Με αυτόν τον τρόπο μπορούμε να περιορίσουμε το πλήθος των επαφών που απαιτούνται για την διασύνδεση τους με τον μικροελεγκτή.

Στην τελική πλακέτα της κατασκευής, λόγω περιορισμένου χώρου προσπαθήσαμε να περιορίσουμε όσο το δυνατόν περισσότερο το πλήθος των συνδέσεων που θα καταλήγουν στην οθόνη. Τελικά καταλήξαμε στο να υπάρχει μια διπλή ακιδοσειρά με 8 επαφές (2x4). Η μείωση αυτή κατέστη δυνατή λόγω των συγκεκριμένων δυνατοτήτων που θα χρησιμοποιήσουμε από την οθόνη μας. Πιο συγκεκριμένα, οι επαφές που διαθέτει η πλακέτα είναι με την εξής διάταξη:

pin #	Περιγραφή επαφής πλακέτας		pin #	Περιγραφή επαφής πλακέτας	
1	VCC	τροφοδοσία οθόνης (+5V)	2	GND	τροφοδοσία οθόνης (GND)
3	E	σήμα ενεργοποίησης	4	RS	0 = εισαγωγή εντολών 1 = εισαγωγή δεδομένων
5	DB4	γραμμή δεδομένων 4	6	DB5	γραμμή δεδομένων 5
7	DB6	γραμμή δεδομένων 6	8	DB7	γραμμή δεδομένων 7

Πίνακας 5.2: Διάταξη επαφών πλακέτας για σύνδεση εξωτερικής LCD

Οι επαφές της οθόνης (όπως αυτές παρουσιάστηκαν στον πίνακα 5.1) που λείπουν από την πλακέτα, είναι μόνιμα συνδεδεμένες είτε στην τάση τροφοδοσίας είτε στην γείωση. Αυτό μπορεί να γίνει απευθείας πάνω στην οθόνη και γι' αυτό δεν απαιτήθηκαν επιπλέον ακίδες για αυτές. Στον πίνακα 5.3 φαίνονται αναλυτικά οι συνδέσεις αυτές.

pin #	Περιγραφή συνδέσεων που γίνονται απευθείας πάνω στην LCD οθόνη			pin #	
3	VEE	ρύθμιση αντίθεσης	συνδέεται με γείωση για να έχουμε πάντα μέγιστη αντίθεση	VSS	1
5	R/W	0 = εγγραφή σε LCD 1 = ανάγνωση από LCD	συνδέεται με γείωση εφόσον θα στέλνουμε μόνο εντολές εγγραφής	VSS	1
15	LED -	τροφοδοσία φωτισμού (GND)	συνδέεται στην γείωση τροφοδοσίας της LCD	VSS	1
16	LED +	τροφοδοσία φωτισμού (+5V)	συνδέεται στην θετική τάση τροφοδοσίας της LCD	VCC	2

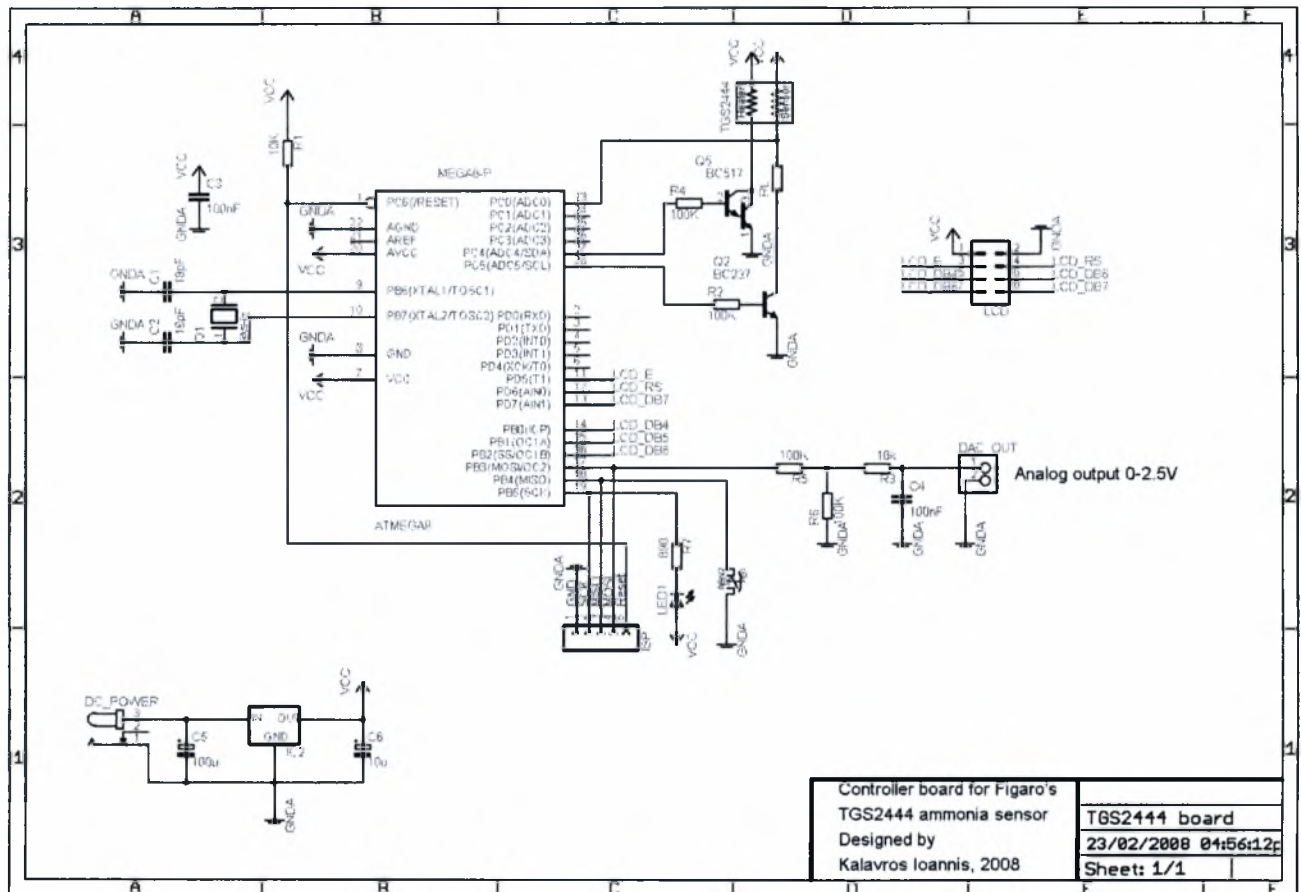
Πίνακας 5.3: Συνδέσεις που απαιτούνται πάνω στην πλακέτα LCD

Για την σωστή εμφάνιση των πληροφοριών κατάστασης του αισθητήρα, θα πρέπει να χρησιμοποιηθούν οθόνες που έχουν τουλάχιστον δύο γραμμές των δεκαέξι χαρακτήρων.

5.5 Παρουσίαση τελικών σχεδίων

5.5.1 Ηλεκτρονικό σχέδιο

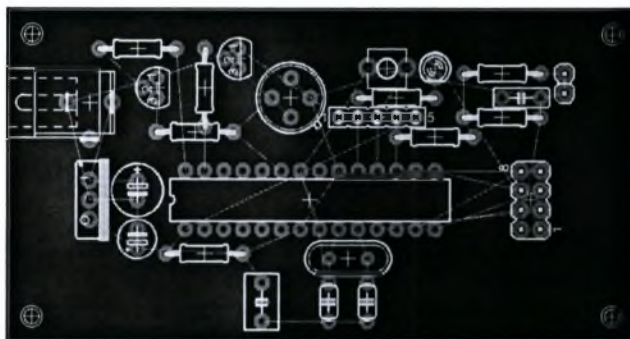
Συνδυάζοντας όλα τα επιμέρους κυκλώματα που παρουσιάστηκαν παραπάνω, καταλήγουμε στο παρακάτω σχέδιο:



Εικόνα 5.2: Ηλεκτρονικό σχέδιο κυκλώματος διασύνδεσης του αισθητήρα TGS2444

5.5.2 Σχέδιο τυπωμένης πλακέτας

Με βάση το ηλεκτρονικό σχέδιο, ξεκινάμε την δημιουργία της πλακέτας. Αρχικά τοποθετούμε τα ηλεκτρονικά εξαρτήματα στην περιοχή σχεδίασης με τέτοιο τρόπο ώστε τα εξαρτήματα που διασυνδέονται μεταξύ τους να μην απέχουν πολύ. Αυτό βοηθάει στην χάραξη των χαλκοδιαδρομών με τον πιο αποδοτικό τρόπο.



Εικόνα 5.3: Τοποθέτηση εξαρτημάτων στην πλακέτα

Το σχεδιαστικό πρόγραμμα Eagle έχει ένα εργαλείο autorouter το οποίο μπορεί να σχεδιάσει αυτόματα τις χαλκοδιαδρομές. Λειτουργεί όμως αποδοτικότερα για πλακέτες πολλαπλών επιπέδων. Στην περίπτωσή μας, που θέλουμε να δημιουργήσουμε μια πλακέτα μονής όψης με όσο το δυνατόν πιο πυκνή τοποθέτηση εξαρτημάτων, αναγκάζομαστε να κάνουμε την χάραξη χειροκίνητα. Με αυτόν τον τρόπο καταφέρνουμε να εξαλείψουμε όσο το δυνατόν περισσότερο την ανάγκη χρήσης γεφυρώσεων στην επάνω πλευρά της πλακέτας.

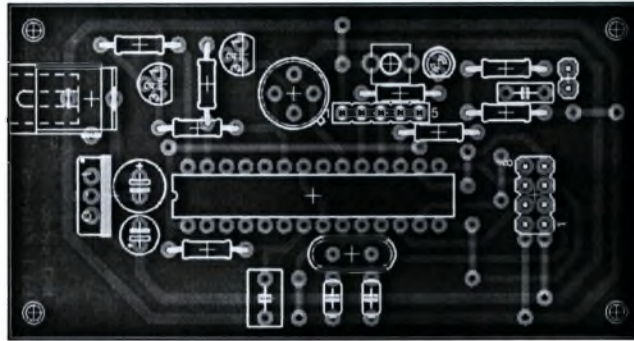
Ένας παράγοντας που πρέπει να λάβουμε υπόψιν σε αυτήν την φάση έχει να κάνει σχετικά με το πλάτος των διαδρομών και την διάμετρο των τρυπών. Επειδή η πλακέτα δεν θα κατασκευαστεί σε κάποιο εξειδικευμένο εργαστήριο, πρέπει να φροντίσουμε ώστε να μπορεί να γίνει ακόμα και με απλά εργαλεία.

Έτσι προκύπτουν οι εξής περιορισμοί:

- Οι διαμέτροι των τρυπών πρέπει να είναι μεγαλύτερες από 0,8mm. Αυτό είναι το ελάχιστο μέγεθος που μπορεί να τρυπηθεί με ακρίβεια με τον εξοπλισμό που είχα διαθέσιμο.
- Το πλάτος των διαδρομών θα είναι 32mil (0,8mm περίπου). Αυτό είναι αρκετό ώστε η αποχάλκωση να μπορεί να γίνει πιο απλή, χωρίς κίνδυνο να υπάρξουν ασυνέχειες στους αγωγούς.
- Η ελάχιστη απόσταση μεταξύ γειτονικών διαδρομών θα είναι 40mil (1mm περίπου) ώστε να μην υπάρχουν κίνδυνοι δημιουργίας βραχυκυκλωμάτων.

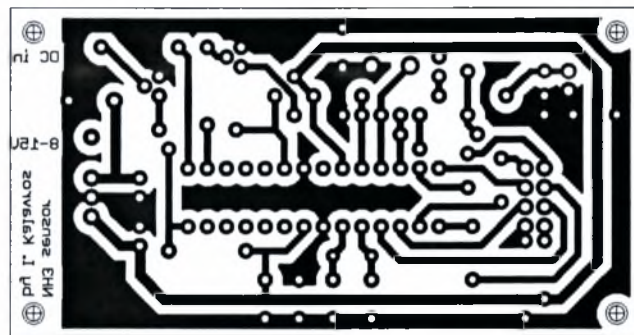
Αυτοί οι περιορισμοί μας οδηγούν στη σχεδίαση που φαίνεται στην εικόνα 5.4. Με μπλε χρώμα παρουσιάζονται οι συνδέσεις στην πλευρά του χαλκού και με κόκκινο χρώμα οι γεφυρώσεις της πάνω πλευράς.

Οι διαστάσεις της πλακέτας είναι 42x80mm.



Εικόνα 5.4: Χαλκοδιαδρομές και γεφυρώσεις

Η διαφάνεια που θα χρειαστεί για την φωτοχημική χάραξη της πλακέτας προκύπτει από το παραπάνω σχέδιο. Θα έχει μαύρο χρώμα μόνο στις περιοχές που θέλουμε να μείνει χαλκός. Προκειμένου να διευκολυνθεί η αποχάλκωση, καλό είναι να μειώσουμε όσο το δυνατόν περισσότερο την επιφάνεια που θα πρέπει να αφαιρεθεί. Για αυτόν τον λόγο επιλέγουμε να διευρύνουμε τις χαλκοδιαδρομές που είναι συνδεδεμένες με την γείωση ώστε να καλύψουν όσο το δυνατόν μεγαλύτερη επιφάνεια (εικόνα 5.5).



Εικόνα 5.5: Διαφάνεια για εμφάνιση φωτοευαίσθητης πλακέτας

6 Κατασκευή πλακέτας

Η μέθοδος που θα ακολουθήσουμε για την κατασκευή της πλακέτας είναι η φωτοχημική. Η πλακέτα αρχικά είναι ολόκληρη καλυμμένη με ένα φύλλο χαλκού. Αυτό έχει μια προστατευτική επίστρωση η οποία είναι ανθεκτική στα χημικά της αποχάλκωσης αλλά καταστρέφεται κατά την έκθεσή της σε υπεριώδη ακτινοβολία. Έτσι μπορούμε να χαράξουμε ότι σχέδιο θέλουμε, ανάλογα με το ποιες περιοχές εκθέτονται στην υπεριώδη ακτινοβολία.

Η διαδικασία χωρίζεται σε τρία βασικά στάδια:

- **Εμφάνιση**

Το σχέδιο μεταφέρεται στην φωτοευαίσθητη επιφάνεια ώστε τελικώς να μείνει η επίστρωση μόνο στα σημεία που θα υπάρχει χαλκός

- **Αποχάλκωση**

Απομάκρυνση του χαλκού με χημικά μέσα.

- **Επικασσιτέρωση**

Επικάλυψη του χαλκού με λεπτό στρώμα κασσίτερου που τον προστατεύει από τη διάβρωση και διευκολύνει το κόλλημα των ηλεκτρονικών εξαρτημάτων.

Πριν από όλα αυτά όμως θα ξεκινήσουμε κατασκευάζοντας τον κατάλληλο θάλαμο έκθεσης.

6.1 Εμφάνιση

6.1.1 Θάλαμος έκθεσης

Για να έχουμε σωστή έκθεση θα πρέπει να δημιουργήσουμε κάποιον θάλαμο όπου θα έχουμε σταθερό και ομοιόμορφο φωτισμό υπεριώδους ακτινοβολίας.

Αρχικά θα πρέπει να αποφασίσουμε για την πηγή της υπεριώδους ακτινοβολίας. Οι επιλογές που έχουμε για λάμπες είναι οι εξής:

- Ατμών υδραργύρου
- Φθορίου
- Cold-cathode fluorescent lamp (CCFL)

Στον πίνακα 6.1 γίνεται μια σύγκριση αυτών των λύσεων.

Τύπος λάμπας	Προτερήματα	Μειονεκτήματα
Ατμών υδραργύρου	Υψηλή ένταση φωτός Μικροί χρόνοι έκθεσης	Μεγάλο μέγεθος Αναπτύσσουν μεγάλη θερμοκρασία Απαιτούν κύκλωμα μετασχηματιστή Χρειάζονται προθέρμανση για να αποδώσουν
Φθορίου	Υψηλή ένταση φωτός Μικροί χρόνοι έκθεσης	Μεγάλο μέγεθος Απαιτούν κύκλωμα ballast και starter
Cold-cathode	Μικρό μέγεθος Ελάχιστη έκλυση θερμότητας Μικρό κύκλωμα οδήγησης Χαμηλή τιμή	Μικρότερη ένταση φωτός Μεγαλύτεροι χρόνοι έκθεσης

Πίνακας 6.1: Σύγκριση λαμπτήρων υπεριώδους ακτινοβολίας

Οι λάμπες που τελικώς επιλέχθηκαν είναι τύπου cold-cathode. Για ομοιόμορφο φωτισμό σε όλο το πλάτος της πλακέτας θα χρησιμοποιήσουμε δύο τέτοιες λάμπες.



Εικόνα 6.1: CCFL λάμπες υπεριώδους ακτινοβολίας και κύκλωμα οδήγησης (πολλαπλασιαστής)

Ο θάλαμος κατασκευάστηκε από ένα κουτί αρκετά μεγάλο ώστε να μπορεί να χωρέσει τις λάμπες. Εσωτερικά επικαλύφθηκε με φύλλο αλουμινίου ώστε να αντανακλά την ακτινοβολία και να πετυχαίνει ομοιόμορφο φωτισμό.

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Οι λάμπες στερεώθηκαν στο καπάκι του, σε ίσες αποστάσεις μεταξύ τους. Τα καλώδια για την τροφοδοσία τους οδηγήθηκαν στο εξωτερικό του κουτιού όπου συνδέθηκαν με τον πολλαπλασιαστή.

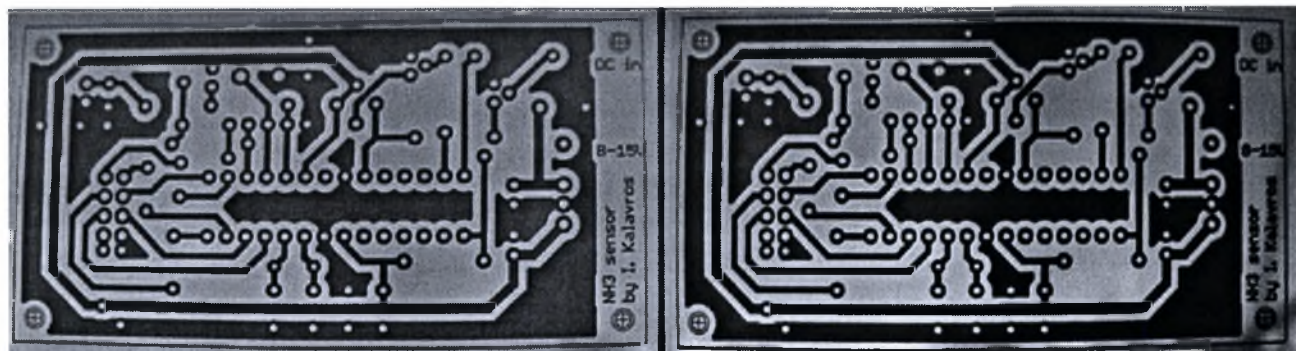


Εικόνα 6.2: Άποψη του εσωτερικού του θαλάμου έκθεσης

6.1.2 Έκθεση σε UV

Αρχικά πρέπει να ξεκινήσουμε με την προετοιμασία της διαφάνειας που έχει το σχέδιο προς εμφάνιση. Για καλύτερα αποτελέσματα πρέπει οι περιοχές της διαφάνειας με μαύρο χρώμα να είναι όσο το δυνατόν πιο αδιαπέραστες από την υπεριώδη ακτινοβολία.

Αυτό το επιτυγχάνουμε χρησιμοποιώντας δύο διαφάνειες, βάζοντας την μία πάνω από την άλλη και φροντίζοντας ώστε να είναι απόλυτα ευθυγραμμισμένες μεταξύ τους. Όπως φαίνεται και από την σύγκριση στην εικόνα 6.3 όταν χρησιμοποιούμε δύο διαφάνειες το αποτέλεσμα είναι πολύ καλύτερο.



Εικόνα 6.3: Σύγκριση μονής και διπλής διαφάνειας

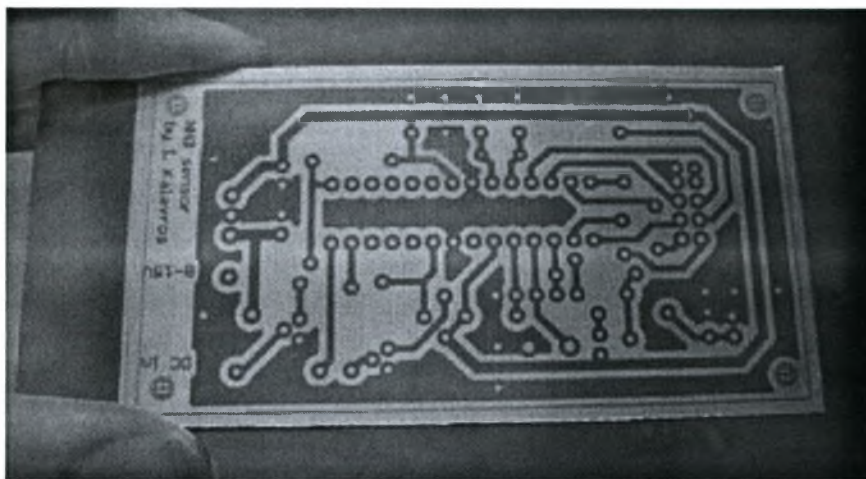
Οι διαφάνειες τοποθετούνται πάνω στην φωτοευαίσθητη πλακέτα και φροντίζουμε ώστε να εφάπτονται πλήρως με αυτήν πιέζοντάς τες με ένα κομμάτι τζαμιού. Η έκθεση στον θάλαμο που κατασκευάσαμε ολοκληρώνεται σε 15 λεπτά.

6.1.3 Αφαίρεση φωτοευαίσθητης επίστρωσης

Για την αφαίρεση της επίστρωσης που εκτέθηκε στην UV ακτινοβολία βυθίζουμε την πλακέτα σε διάλυμα καυστικής σόδας (NaOH). Την καυστική σόδα μπορούμε να την προμηθευτούμε εύκολα καθώς υπάρχει σε διάφορα καθαριστικά προϊόντα.

Η διαδικασία αυτή ολοκληρώνεται σε 1-2 λεπτά. Το σχέδιο αρχίζει σχεδόν αμέσως να διαγράφεται αποκαλύπτοντας την επιφάνεια του χαλκού στα σημεία που εκτέθηκαν (εικόνα 6.4).

Αμέσως μετά ξεπλένουμε την πλακέτα σε καθαρό νερό για να την καθαρίσουμε από την καυστική σόδα. Αν δεν το κάνουμε θα αρχίσει σιγά σιγά να καταστρέφεται και η επίστρωση από τα υπόλοιπα σημεία.



Εικόνα 6.4: Η πλακέτα είναι έτοιμη προς αποχάλκωση

Ιδιαίτερη προσοχή πρέπει να δοθεί καθώς η καυστική σόδα δεν πρέπει να έρχεται σε επαφή με το δέρμα γιατί προκαλεί χημικά εγκαύματα.

6.2 Αποχάλκωση

Η αποχάλκωση μπορεί να γίνει με διάφορα χημικά. Οι πιο συνηθισμένες μέθοδοι είναι οι παρακάτω:

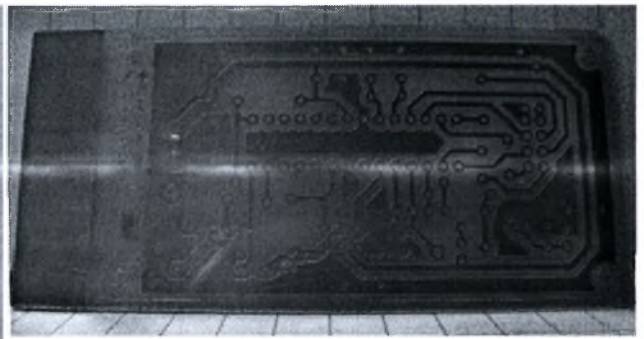
- διάλυμα ένυδρου χλωριούχου σιδήρου σε αναλογία 1:1 σε νερό
- διάλυμα άνυδρου χλωριούχου σιδήρου σε αναλογία 1:2 σε νερό
- διάλυμα υπερθεϊκού άλατος αμμωνίας σε αναλογία 4:10 σε νερό θερμοκρασίας 40°C
- διάλυμα υδροχλωρικού οξέως και υπεροξειδίου του υδρογόνου σε αναλογία 2:1:4 σε νερό

Η κάθε μία μέθοδος έχει και τα δικά της προτερήματα και μειονεκτήματα. Από αυτές τελικά επιλέχθηκε η τελευταία διότι ήταν πιο εύκολο να προμηθευτούμε τα απαραίτητα χημικά.

Ετοιμάζουμε το διάλυμα σε ένα πλαστικό σκεύος και βυθίζουμε σε αυτό την πλακέτα που εμφανίσαμε προηγουμένως. Ο χαλκός αρχίζει σιγά σιγά να απομακρύνεται (εικόνα 6.5). Για να επιταχυνθεί η διαδικασία καλό είναι να ανακινούμε την πλακέτα μέσα στο διάλυμα.



Εικόνα 6.5: Η πλακέτα μέσα στο διάλυμα αποχάλκωσης



Εικόνα 6.6: Η αποχάλκωμένη πλακέτα

Όλη αυτή η διαδικασία πρέπει να γίνει σε καλά αεριζόμενο χώρο γιατί εκλύει βλαβερά αέρια. Επίσης τόσο το υδροχλωρικό οξύ όσο και το υπεροξείδιο του υδρογόνου δεν πρέπει να έρθουν σε επαφή με το δέρμα διότι είναι ισχυρά οξειδωτικά και προκαλούν χημικά εγκαύματα.

Η αποχάλκωση ολοκληρώνεται σε 30-60 λεπτά αναλόγως με το πόσο ισχυρό είναι το διάλυμα που φτιάξαμε. Η πλακέτα θα πρέπει να έχει καθαρίσει τελείως από τον χαλκό στις περιοχές που εκτέθηκαν και θα φαίνεται καθαρά το σχέδιο που τυπώσαμε (εικόνα 6.6). Την καθαρίζουμε με καθαρό νερό και την αφήνουμε να στεγνώσει.

Αυτό που πρέπει να κάνουμε τώρα είναι να απομακρύνουμε την φωτοευαίσθητη επίστρωση πάνω από τους χαλκοδιαδρόμους. Για να γίνει αυτό πρέπει να την ξαναβάλουμε στον θάλαμο έκθεσης για μερικά λεπτά ώστε να καεί όλη η επίστρωση και μετά να την καθαρίσουμε στο διάλυμα της καυστικής σόδας.

6.3 Επικασσιτέρωση

Η επιφάνεια του χαλκού είναι πολύ σημαντικό να μην αφηθεί εκτεθειμένη επειδή είναι ευάλωτη στην οξείδωση από τον ατμοσφαιρικό αέρα. Η μέθοδοι που χρησιμοποιούνται συνήθως για την προστασία της είναι τρεις:

- Επικάλυψη με υλικό κόλλησης (καλάι)
- Ψεκασμός με ειδικό spray βερνικιού
- Επικασσιτέρωση

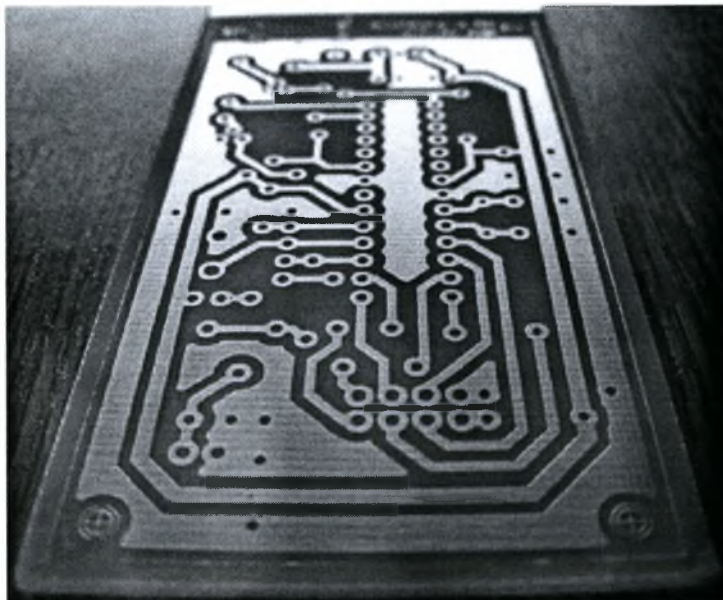
Η επικασσιτέρωση παρέχει καλύτερα αποτελέσματα καθότι δημιουργεί μια πιο ανθεκτική και ομοιόμορφη στρώση. Χρησιμοποιείται όμως σπάνια για ερασιτεχνική χρήση επειδή είναι πιο πολύπλοκη διαδικασία. Χρειάζεται να γίνει είτε με ειδικό μηχάνημα (roller tinning machine) είτε με χημική μέθοδο που είναι ακριβότερη και με χημικά που χαλάνε εύκολα.

Η μέθοδος επικασσιτέρωσης που χρησιμοποιήθηκε για την συγκεκριμένη πλακέτα είναι ασυνήθιστη, αλλά παράγει πάρα πολύ καλά αποτελέσματα. Έγινε χρησιμοποιώντας μια αλοιφή που προορίζεται για συγκόλληση σωλήνων χαλκού.

Πρόκειται για την αλοιφή χαλκού με κόλληση Rosol 3 της εταιρείας Rothenberger. Περιέχει το συστατικό $S-Sn_{97}Cu_3$ το οποίο με θέρμανση εναποθέτει κασσίτερο πάνω στον χαλκό.

Η εφαρμογή του απαιτεί την επάλειψη της πλακέτας με ένα λεπτό στρώμα από αυτό το υλικό και στην συνέχεια ομοιόμορφη θέρμανσή του. Αυτό μπορεί να γίνει βάζοντας την πλακέτα σε έναν προθερμασμένο φούρνο στους $200^{\circ}C$ για 2-3 λεπτά.

Μόλις κρυώσει, απομακρύνουμε το επιπλέον υλικό με ελαφρύ τρίψιμο και νερό.

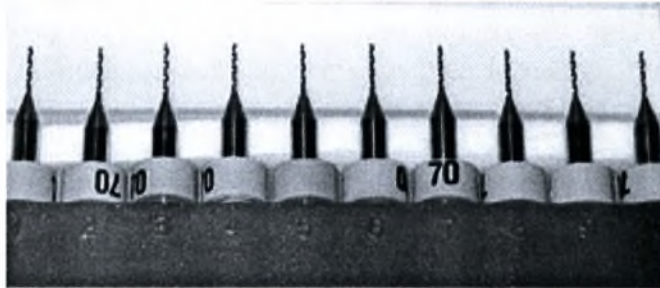


Εικόνα 6.7: Επικασσιτερωμένη πλακέτα

6.4 Τρύπημα

Μετά από όλες τις προηγούμενες διεργασίες, ήρθε η ώρα να ανοιχτούν οι τρύπες από όπου θα περάσουν οι ακροδέκτες των εξαρτημάτων και οι διασυνδέσεις των γεφυρώσεων της επάνω πλευράς. Θα χρειαστούμε ένα δράπανο που να έχει πάνω από 2000 στροφές/λεπτό καθώς και τρυπάνια στις κατάλληλες διατομές.

Επειδή η πλακέτα είναι κατασκευασμένη από fiberglass τα απλά τρυπάνια δεν είναι κατάλληλα επειδή φθείρονται πολύ εύκολα. Αντί αυτών πρέπει να χρησιμοποιήσουμε τρυπάνια κατασκευασμένα από καρβίδιο βολφραμίου (tungsten carbide).



Εικόνα 6.8: Τρυπάνια καρβιδίου

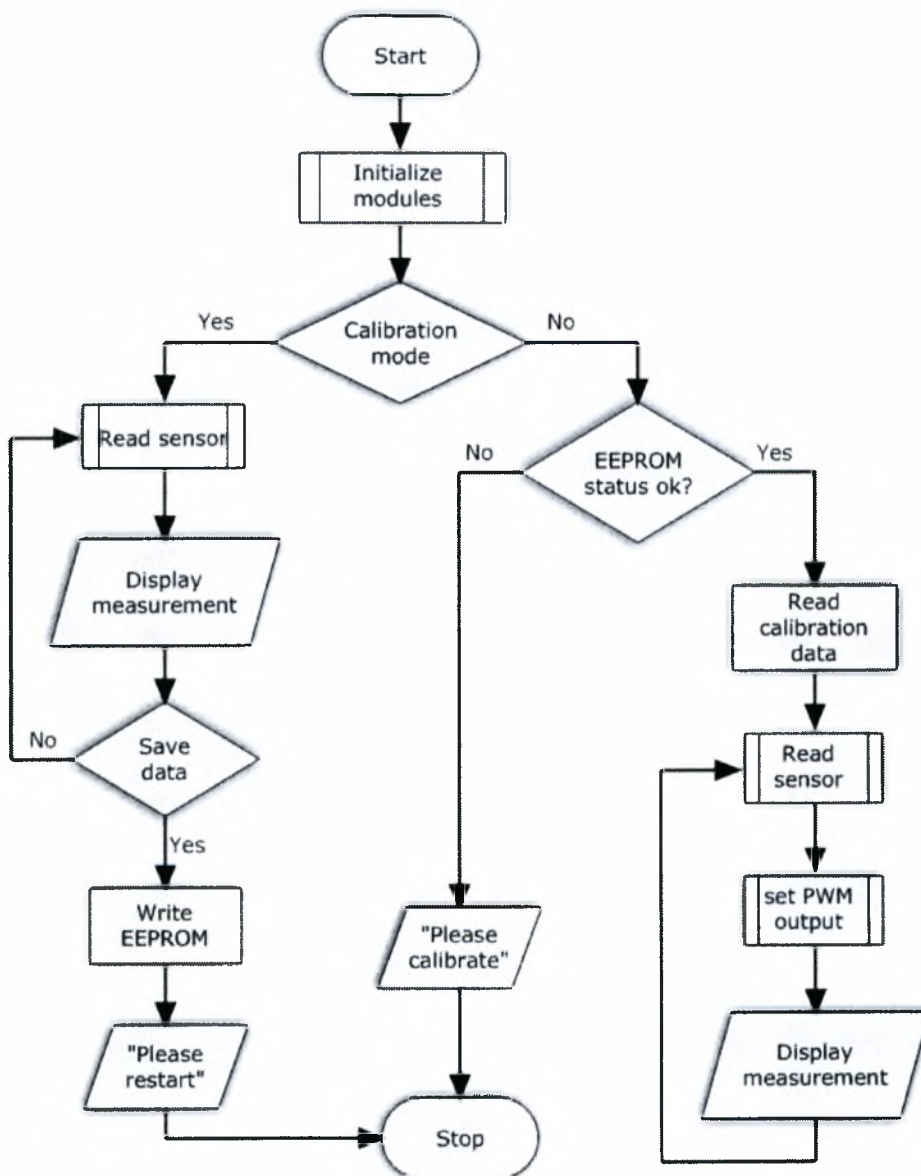
Οι διαμέτροι των τρυπών που πρέπει να ανοίξουμε στην κατασκευή μας είναι 0,8mm για όλα τα εξαρτήματα που θα έχει και 2,5mm για τις οπές στήριξης της πλακέτας

Το τρύπημα πρέπει να γίνει με χρήση κάποια σταθερής βάσης επειδή τα τρυπάνια αυτά είναι εξαιρετικά ευαίσθητα και σπάνε.

7 Περιγραφή προγράμματος μικροελεγκτή

Η βασική δομή του κυρίως προγράμματος του μικροελεγκτή παρουσιάζεται στο παρακάτω διάγραμμα ροής (εικόνα 7.1). Ξεκινώντας, το πρόγραμμα αρχικοποιεί τις επιμέρους μονάδες του AVR (θύρες επικοινωνίας, ADC, PWM, χρονοιστές), καθώς και την εξωτερική LCD οθόνη (σε περίπτωση που είναι συνδεδεμένη).

Οι χρονοιστές έχουν την δυνατότητα να εκτελούνται εντελώς ανεξάρτητα από το κυρίως πρόγραμμα. Έτσι μπορούν να μετρούν με ακρίβεια τον χρόνο που περνάει και μέσω ενός σήματος διακοπής να καλούν την διαδικασία εκείνη που φροντίζει για την ενεργοποίηση των σημάτων ελέγχου του αισθητήρα καθώς και για την λήψη των μετρήσεων την κατάλληλη στιγμή.



Εικόνα 7.1: Απλοποιημένο διάγραμμα ροής για το λογισμικό του μικροελεγκτή

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Κατά την εκκίνηση του προγράμματος, αναλόγως με το αν έχει πατηθεί το κουμπί που βρίσκεται πάνω στην πακέτα, ο μικροελεγκτής εισέρχεται σε λειτουργία βαθμονόμησης ή στην κανονική λειτουργία.

Η βαθμονόμηση είναι ένας κύκλος συνεχόμενων μετρήσεων της ηλεκτρικής αντίστασης που παρουσιάζει ο αισθητήρας. Ολοκληρώνεται με το πάτημα του κουμπιού της πλακέτας, οπότε και αποθηκεύεται στην μνήμη EEPROM η τιμή της τελευταίας μέτρησης που έγινε.

Η κανονική λειτουργία ξεκινάει αρχικά με τον έλεγχο της κατάστασης της μνήμης EEPROM. Αν δεν είναι στην σωστή κατάσταση (όπως για παράδειγμα μετά από προγραμματισμό του μικροελεγκτή) το πρόγραμμα σταματάει προτρέποντάς μας να κάνουμε εκ νέου βαθμονόμηση. Αυτή η κατάσταση υποδηλώνεται με γρήγορες αναλαμπές του ενδεικτικού LED και σχετικό μήνυμα στην LCD οθόνη.

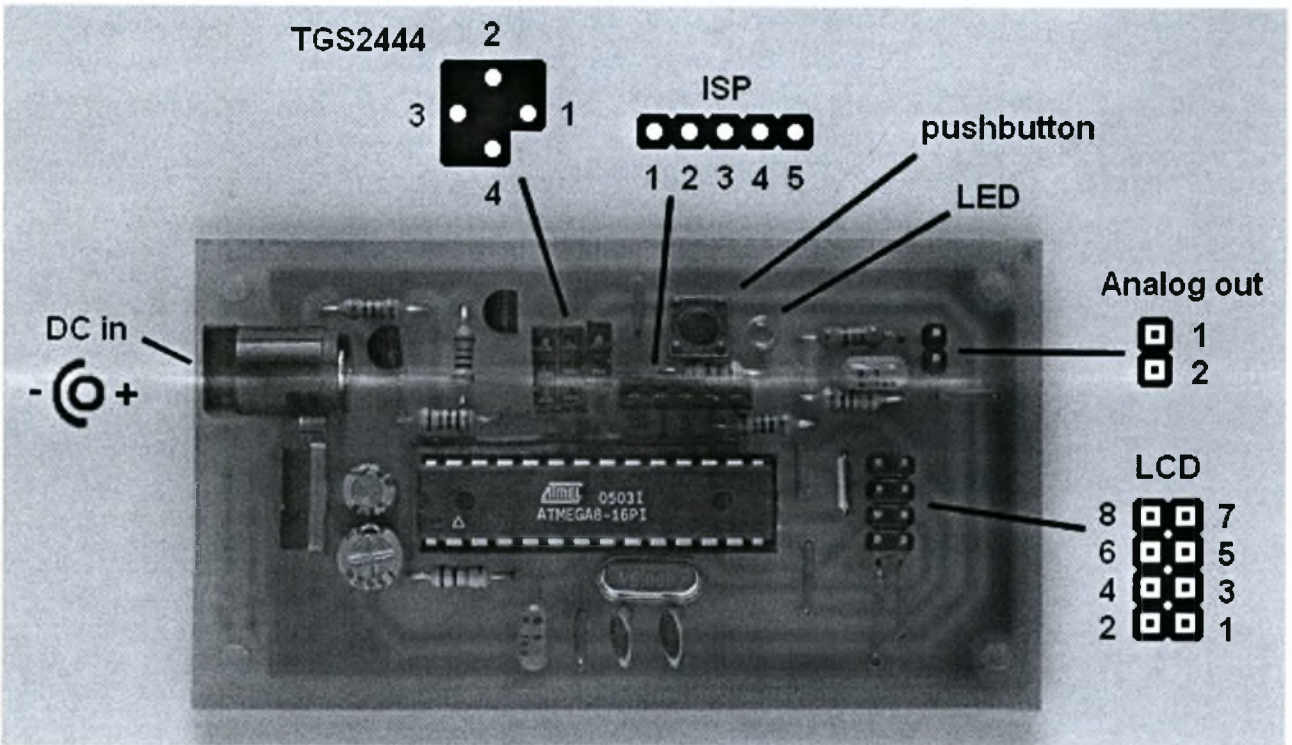
Αν η μνήμη είναι εντάξει, διαβάζονται τα στοιχεία αυτά και ξεκινάει ο κύκλος των μετρήσεων. Σε κάθε κύκλο γίνεται μία μέτρηση και η τιμή του αποτελέσματος υπολογίζεται συναρτήσει των δεδομένων από την τελευταία βαθμονόμηση που έχει γίνει . Τέλος, δίνεται η εντολή στην μονάδα PWM για την δημιουργία της αναλογικής τάσης που θα πάρουμε ως έξοδο.

Ο πηγαίος κώδικας του μικροελεγκτή δίνεται στο Παράρτημα.

8 Λειτουργίας της κατασκευής

8.1 Περιγραφή πλακέτας

Στην εικόνα 8.1 υπάρχει μια φωτογραφία της πλακέτας όπου έχουν σημειωθεί όλες οι υποδοχές και επαφές διασύνδεσης που διαθέτει. Όταν γίνεται κάποια σύνδεση, πρέπει να δοθεί ιδιαίτερη προσοχή στην σωστή πολικότητα των επαφών που συνδέουμε.



Εικόνα 8.1: Η πλακέτα με επεξήγηση όλων των επαφών διασύνδεσης που διαθέτει

Ο TGS2444 συνδέεται στη βάση που έχει σχήμα τετραγώνου. Ο προσανατολισμός του πρέπει να είναι τέτοιος ώστε η ακίδα που έχει στην περιφέρεια του κελύφους του να ευθυγραμμίζεται με την εσοχή που υπάρχει στη μία γωνία της βάσης. Στην διπλανή εικόνα φαίνεται καθαρά αυτή η ακίδα που μας βοηθάει να διακρίνουμε μεταξύ τους τα pins. (η αρίθμηση των pins εξηγείται στην εικόνα 3.10 της σελίδας 15)



Η αναλογική έξοδος της μετρούμενης τιμής γίνεται από δύο ακίδες που βρίσκονται στην πάνω δεξιά γωνία της πλακέτας.

pin #	Περιγραφή επαφής πλακέτας	
1	Vout	Αναλογική έξοδος (DC τάση 0-2,5V)
2	GND	Γείωση

Πίνακας 8.1: Επεξήγηση των ακίδων για αναλογική έξοδο διασύνδεσης

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Η LCD οθόνη συνδέεται στην επαφή με τις 8 ακίδες που βρίσκεται στο δεξιό μέρος της πλακέτας. Η αρίθμηση των επαφών φαίνεται στην εικόνα 8.1 και η ερμηνεία τους στον παρακάτω πίνακα.

pin #	Περιγραφή επαφής πλακέτας		pin #	Περιγραφή επαφής πλακέτας	
8	DB7	γραμμή δεδομένων 7	7	DB6	γραμμή δεδομένων 6
6	DB5	γραμμή δεδομένων 5	5	RS	γραμμή δεδομένων 4
4	DB4	0 = εισαγωγή εντολών 1 = εισαγωγή δεδομένων	3	E	σήμα ενεργοποίησης
2	GND	τροφοδοσία οθόνης (GND)	1	VCC	τροφοδοσία οθόνης (+5V)

Πίνακας 8.2: Επεξήγηση των ακίδων για σύνδεση οθόνης LCD

Στα δεξιά της βάσης του αισθητήρα υπάρχει μία σειρά με πέντε υποδοχές. Εκεί μπορεί να συνδεθεί ένας ISP προγραμματιστής για τον προγραμματισμό του μικροελεγκτή AVR. Η αντιστοιχία των επαφών δίνεται παρακάτω.

pin #	Περιγραφή επαφής πλακέτας	
1	GND	Γείωση
2	SCK	Ρολόι για την σειριακή επικοινωνία
3	MISO	Master In Slave Out. Για αποστολή δεδομένων από τον AVR (slave) προς το PC (master). Τα δεδομένα εκπέμπονται κατά τις ακμές του SCK.
4	MOSI	Master Out Slave In. Για αποστολή δεδομένων από το PC (master) προς τον AVR (slave). Τα δεδομένα εκπέμπονται κατά τις ακμές του SCK.
5	Reset	Χρησιμοποιείται για επανεκκίνηση του AVR κάτω από τον έλεγχο του PC

Πίνακας 8.3: Επεξήγηση των ακίδων για ISP προγραμματισμό του μικροελεγκτή

Η τροφοδοσία της πλακέτας με ρεύμα γίνεται από το βύσμα που υπάρχει στην αριστερή πλευρά της πλακέτας. Απαιτεί ένα ομοαξονικό φως 2,5mm, με το περίβλημα να φέρει την γείωση και την κεντρική επαφή με την θετική τάση.

Η τάση τροφοδοσίας πρέπει να είναι σταθεροποιημένη από 7V μέχρι 12V. Η κατασκευή μπορεί να αντέξει ακόμα και υπερτάσεις έως 20V χωρίς να υπάρχει κίνδυνος πρόκλησης βλάβης, αλλά επειδή δεν διαθέτει ψύκτρα πρέπει να χρησιμοποιείται η χαμηλότερη επιτρεπτή τάση.

8.2 Διαδικασία βαθμονόμησης

8.2.1 Ρύθμιση βάσει των ηλεκτρικών χαρακτηριστικών του αισθητήρα

Οι αισθητήρες TGS2444 λόγω της κατασκευής τους δεν έχουν όλοι ακριβώς τα ίδια ηλεκτρικά χαρακτηριστικά. Μάλιστα σύμφωνα με την κατασκευάστρια εταιρεία η ηλεκτρική αντίσταση που παρουσιάζουν όταν βρίσκονται σε περιβάλλον με 10ppm NH₃ κυμαίνεται μεταξύ 3,63kΩ και 36,3kΩ. Επίσης μακροπρόθεσμα και μετά από πολλές ώρες χρήσης μπορεί να μεταβληθούν σε κάποιο βαθμό τα ηλεκτρικά τους χαρακτηριστικά. Για αυτόν τον λόγο, η πλακέτα διασύνδεσης παρέχει την δυνατότητα ρύθμισης ώστε να έχουμε αξιόπιστες μετρήσεις με τον εκάστοτε χρησιμοποιούμενο αισθητήρα.

Η βαθμονόμηση του αισθητήρα θα πρέπει να γίνει τοποθετώντας την κατασκευή σε χώρο με καθαρό αέρα και με θερμοκρασία 20°C. Η τιμή που θα υπολογιστεί με αυτόν τον τρόπο είναι σημαντική γιατί θα αποτελεί την τιμή αναφοράς βάσει της οποίας θα γίνονται όλες οι μετρήσεις στο μέλλον. Για αυτόν τον λόγο πρέπει να δοθεί προσοχή ώστε να γίνει σε χώρο χωρίς αναθυμιάσεις αμμωνίας και με τη θερμοκρασία να πλησιάζει όσο το δυνατόν περισσότερο τους 20°C.

Παρακάτω θα περιγραφεί η διαδικασία που πρέπει να ακολουθήσουμε για την βαθμονόμηση του αισθητήρα. Κατά τη διαδικασία αυτή δεν είναι απαραίτητη η σύνδεση της εξωτερικής LCD οθόνης καθώς μπορούμε να καταλάβουμε την κατάσταση στην οποία βρισκόμαστε απλά παρακολουθώντας το LED. Σε περίπτωση που υπάρχει η οθόνη μπορούμε να βλέπουμε σε πραγματικό χρόνο την τιμή R_s της αντίστασης του αισθητήρα.

Τα βήματα που πρέπει να ακολουθήσουμε είναι τα εξής:

1. Με την κατασκευή αποσυνδεδεμένη από την τροφοδοσία, βεβαιωνόμαστε ότι ο αισθητήρας είναι σωστά τοποθετημένος (με σωστή πολικότητα) πάνω στην βάση του.
2. Πατάμε και κρατάμε πατημένο το κουμπί που είναι πάνω στην πλακέτα.
3. Χωρίς να αφήσουμε το κουμπί συνδέουμε την τροφοδοσία και εξακολουθούμε να το πατάμε (για περίπου 5 δευτερόλεπτα) έως ότου το ενδεικτικό LED πάψει να αναβοσβήνει και μείνει αναμμένο.

Όλο αυτό το διάστημα η οθόνη γράφει “Wait 5sec for Calibration Mode”

Αν κατά λάθος αφήσουμε το κουμπί νωρίτερα, το σύστημα εισέρχεται σε κανονική λειτουργία. Για να επανεκινήσουμε την διαδικασία ξεκινάμε από το βήμα 1.

4. Μόλις το LED σταματήσει να αναβοσβήνει απελευθερώνουμε το κουμπί. Η οθόνη γράφει “Calibration Mode starting now!”. Όσο το LED είναι αναμμένο, βρισκόμαστε σε διαδικασία βαθμονόμησης.
5. Τοποθετούμε την κατασκευή σε μέρος με καθαρό αέρα και θερμοκρασία 20°C. Την αφήνουμε έτσι για τουλάχιστον 50 λεπτά ώστε να μπορέσει να φτάσει ο αισθητήρας στην σωστή κατάσταση λειτουργίας.
Αν έχουμε συνδεδεμένη την οθόνη μπορούμε να βλέπουμε τις μεταβολές στην αντίσταση. Η τιμή θα αυξάνεται μέχρι να σταθεροποιηθεί σε περίπου 50 λεπτά.
6. Για να αποθηκεύσουμε την τιμή αυτή ως την αντίσταση αναφοράς πιέζουμε το κουμπί. Μόλις το LED σβήσει, η τιμή έχει αποθηκευτεί στην μνήμη του μικροελεγκτή.
Η LCD γράφει “Calibration done! Please restart!”
7. Αποσυνδέουμε την τροφοδοσία και μόλις την ξανασυνδέσουμε η κατασκευή ξεκινά να λειτουργεί κανονικά.

8.2.2 Αντιστάθμιση θερμοκρασίας και υγρασίας του περιβάλλοντος

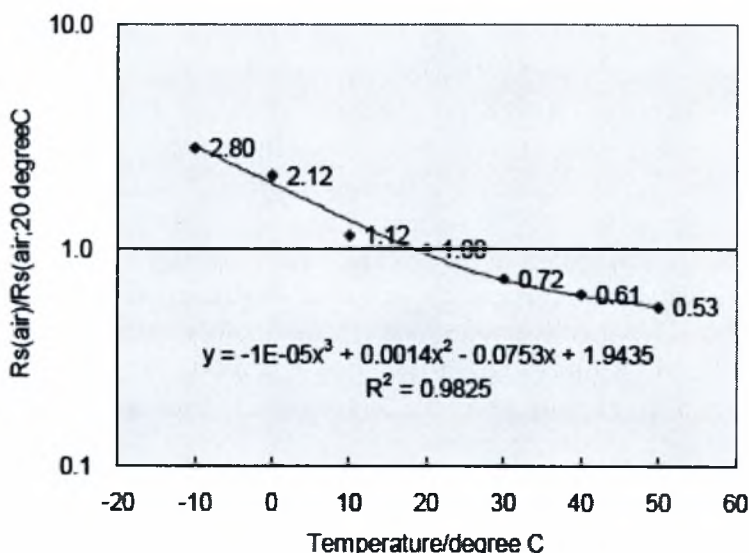
Στην περιγραφή της αρχής λειτουργίας του αισθητήριου στοιχείου είδαμε πως μια αντίδραση οξείδωσης είναι υπεύθυνη για την μεταβολή της αγωγιμότητάς του. Όπως είναι γνωστό, οι χημικές αντιδράσεις επηρεάζονται από την θερμοκρασία του περιβάλλοντος. Αυτό συμβαίνει και με το αισθητήριο στοιχείο του TGS2444. Έτσι η ένδειξη που παίρνουμε για συγκεκριμένη περιεκτικότητα αμμωνίας στον αέρα θα είναι διαφορετική ανάλογα με την θερμοκρασία του αέρα. Η βαθμονόμηση του αισθητήρα γίνεται σε θερμοκρασία 20°C, οπότε οποιαδήποτε μέτρηση σε διαφορετική θερμοκρασία θα έχει ένα σφάλμα το οποίο και πρέπει να εξαλείψουμε.

Αυτή η διόρθωση θα πρέπει να γίνει από το λογισμικό του συστήματος που θα διασυνδέεται με την πλακέτα, δηλαδή στο Tmote Sky. Ο λόγος για αυτό είναι ότι η κατασκευή που αναπτύχθηκε δεν έχει κάποιον ενσωματωμένο αισθητήρα θερμοκρασίας σε αντίθεση με το Tmote Sky που διαθέτει τον SHT11 αισθητήρα θερμοκρασίας/υγρασίας της εταιρείας Sensirion.

Ακόμη και αν προσθέταμε κάποιον αισθητήρα θερμοκρασίας στην πλακέτα μας θα είχαμε πρόβλημα με τους υπολογισμούς στον μικροελεγκτή. Ο ATmega8 είναι ένας 8bit μικροελεγκτής. Αυτό σημαίνει ότι οι πράξεις κινητής υποδιαστολής που θα έπρεπε να γίνουν θα μεγάλωναν πολύ το μεταγλωττισμένο πρόγραμμα (ενδεχομένως να μην χωρούσε και στον AVR) και θα το έκαναν πάρα πολύ αργό.

Παρακάτω εξηγείται ο τρόπος με τον οποίο θα πρέπει να συνυπολογιστεί η θερμοκρασία στο αποτέλεσμα της μέτρησης.

Ο κατασκευαστής του αισθητήρα μας δίνει ένα διάγραμμα με πειραματικές μετρήσεις σε περιβάλλον καθαρού αέρα και για διάφορες θερμοκρασίες. Το διάγραμμα αυτό φαίνεται στην εικόνα 8.2.



Εικόνα 8.2: Μεταβολή τιμής του αισθητήρα λόγω θερμοκρασίας (στοιχεία από Figaro)

Στο διάγραμμα αυτό, εκτός από τα σημεία με τα πειραματικά δεδομένα, είναι σχεδιασμένη και η καμπύλη που τα προσεγγίζει. Δίνεται μάλιστα και ένας μαθηματικός τύπος για την καμπύλη

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

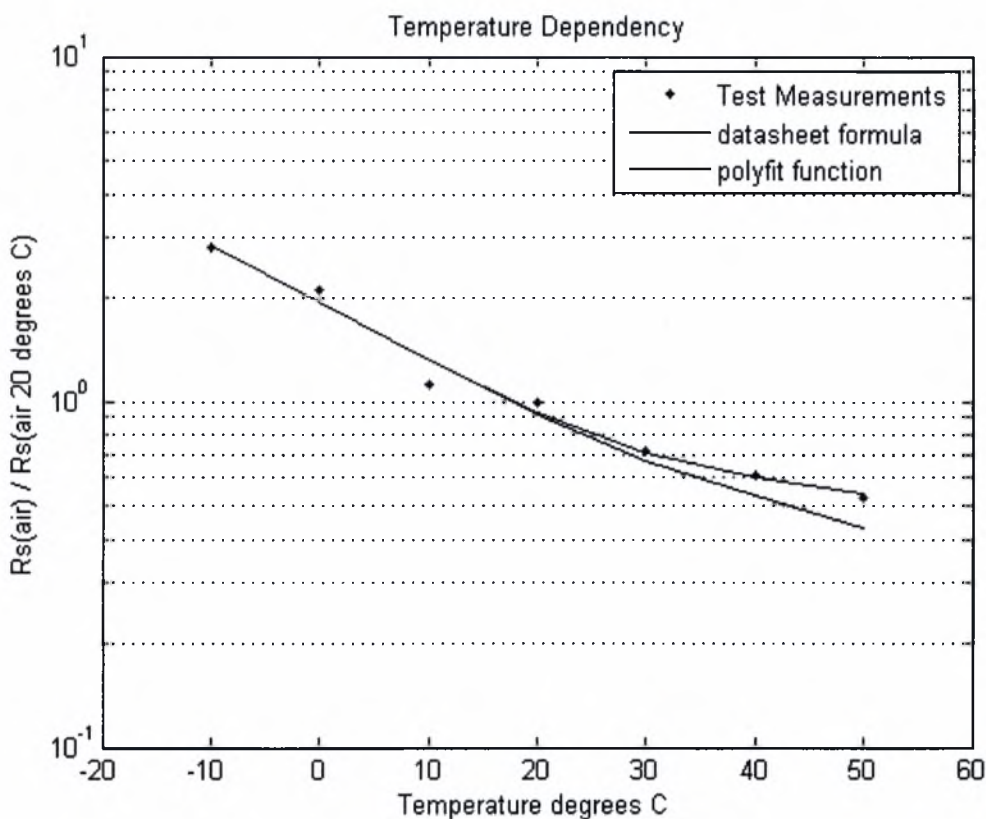
αυτή: $Y = -1 \cdot 10^{-5} \cdot x^3 + 0.0014 \cdot x^2 - 0.0753 \cdot x + 1,9435$, όπου x η θερμοκρασία σε °C

Όμως αυτός ο μαθηματικός τύπος είναι λάθος!

Δεν ανταποκρίνεται στην σχεδιασμένη καμπύλη, καθώς οι συντελεστές του πολυωνύμου έχουν λανθασμένες στρογγυλοποιήσεις! Η απόκλιση του τύπου αυτού είναι μεγαλύτερη στις υψηλότερες θερμοκρασίες και το σφάλμα φτάνει μέχρι και το 10%.

Αυτό γίνεται εύκολα κατανοητό στο διάγραμμα της εικόνας 8.3. Με μαύρο χρώμα έχει σχεδιαστεί η καμπύλη σύμφωνα με τον μαθηματικό τύπο που δίνει η Figaro και με μπλε χρώμα είναι ένα πολυώνυμο 3ου βαθμού που υπολογίσαμε με την βοήθεια του Matlab.

(Ο κώδικας του προγράμματος Matlab που χρησιμοποιήθηκε δίνεται στο Παράρτημα)



Εικόνα 8.3: Σφάλμα του πολυωνύμου προσέγγισης που δίνει η Figaro

Ο τύπος του πολυωνύμου που υπολογίσαμε είναι:

$$Y = -1 \cdot 10^{-5} \cdot x^3 + 0.001449 \cdot x^2 - 0.075488 \cdot x + 1,942143$$
 , όπου x η θερμοκρασία σε °C

Όπως φαίνεται και από το διάγραμμα, το Y μας δίνει την σχέση μεταξύ της αντίστασης του αισθητήριου στον αέρα ως προς την αντίσταση του αισθητήριου στον αέρα θερμοκρασίας 20°C.

Δηλαδή: $Y = \frac{R_{S(air)}}{R_{S(air 20^\circ C)}} \quad (1)$

Η κατασκευή που αναπτύξαμε μας επιστρέφει τον λόγο K της τιμής της αντίστασης στην δεδομένη

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

συγκέντρωση αμμωνίας και στην τρέχουσα θερμοκρασία, ως προς την τιμή της αντίστασης που αποθηκεύτηκε κατά την βαθμονόμηση, δηλαδή σε καθαρό αέρα 20°C.

$$\text{Δηλαδή: } K = \frac{R_S}{R_{S(\text{air } 20^\circ\text{C})}} \quad (2)$$

Από τους τύπους (1) και (2) προκύπτει ότι σε οποιαδήποτε θερμοκρασία, ο λόγος της αντίστασης R_S προς την αντίσταση $R_{S(\text{air})}$ σε καθαρό αέρα σε αυτή τη θερμοκρασία θα ισούται με:

$$\frac{R_S}{R_{S(\text{air})}} = \frac{K}{Y}$$

Άρα το Tmote Sky θα πρέπει όταν διαβάζει μια τιμή K από την πλακέτα με τον αισθητήρα να μετράει με το ενσωματωμένο θερμόμετρό του και την θερμοκρασία x (°C) του περιβάλλοντος και να υπολογίζει τον τύπο:

$$\frac{R_S}{R_{S(\text{air})}} = \frac{K}{-1 \cdot 10^{-5} \cdot x^3 + 0.001449 \cdot x^2 - 0.075488 \cdot x + 1,942143}$$

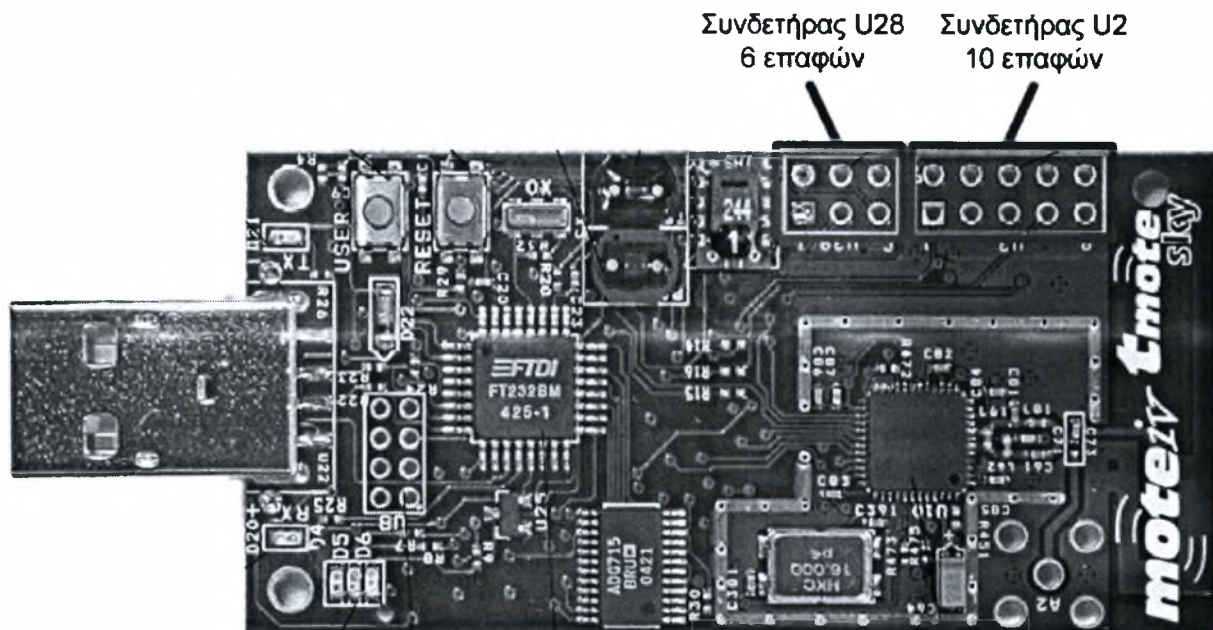
Βάσει αυτής της τιμής και του διαγράμματος της εικόνας 3.7 (σελίδα 12) θα εξάγεται η συγκέντρωση της αμμωνίας στον αέρα.

Με κάποιον ανάλογο τρόπο θα πρέπει να λαμβάνεται υπόψιν και η επίδραση της υγρασίας στην μετρούμενη τιμή. Οι πληροφορίες όμως από το τεχνικό εγχειρίδιο που μας έδωσε η Figaro είναι ελλιπείς. Έτσι ο μόνος τρόπος για να το μάθουμε είναι μέσω πειραματικών δοκιμών σε περιβάλλον ελεγχόμενης υγρασίας. Δυστυχώς όμως αυτές οι δοκιμές δεν μπόρεσαν να γίνουν λόγω έλλειψης του κατάλληλου τεχνικού εξοπλισμού.

8.3 Διασύνδεση με Tmote Sky

Το Tmote Sky διαθέτει δύο σειρές επαφών για σύνδεση εξωτερικών συσκευών. Πρόκειται για τις επαφές U2 (10 ακίδων) και U28 (6 ακίδων) που φαίνονται στην εικόνα 8.4. Σε αυτές μεταξύ άλλων υπάρχουν διαθέσιμα τα 6 κανάλια του μετατροπέα από αναλογικό σε ψηφιακό (ADC) που διαθέτει ο επεξεργαστής του Tmote Sky (πίνακας 8.4).

Η καρδιά του Tmote Sky είναι ένας 8MHz 16bit RISC μικροελεγκτής της Texas Instruments, ο MSP430F1611 [13]. Ο ADC του μικροελεγκτή αυτού μπορεί να διαβάσει με ακρίβεια 12bit αναλογικές τάσεις από 0V μέχρι την τάση τροφοδοσίας 3.6V (max).



Εικόνα 8.4: Συνδετήρες επέκτασης TmoteSky

Η πλακέτα του αισθητήρα που περιγράφηκε στην προηγούμενη ενότητα έχει τη δυνατότητα να μας δίνει μια αναλογική έξοδο η τιμή της οποίας είναι ανάλογη της συγκέντρωσης της αμμωνίας στον αέρα. Αυτήν την αναλογική τιμή θα την διαβάσει το Tmote Sky διαμέσου του ADC του.

pin #	Ονομασία και περιγραφή σήματος	pin #	Ονομασία και περιγραφή σήματος
1	AVcc Τάση τροφοδοσίας Vcc	2	UART0RX UART receive
3	ADC0 Αναλογική είσοδος 0	4	UART0TX UART transmit
5	ADC1 Αναλογική είσοδος 1	6	I2C_SCL Ρολόι I2C
7	ADC2 Αναλογική είσοδος 2	8	I2C_SDA Δεδομένα I2C
9	GND Γείωση τροφοδοσίας	10	ADC3 Αναλογική είσοδος 3

Πίνακας 8.4: Επεξήγηση σημάτων του συνδετήρα 10 επαφών U2 του Tmote Sky

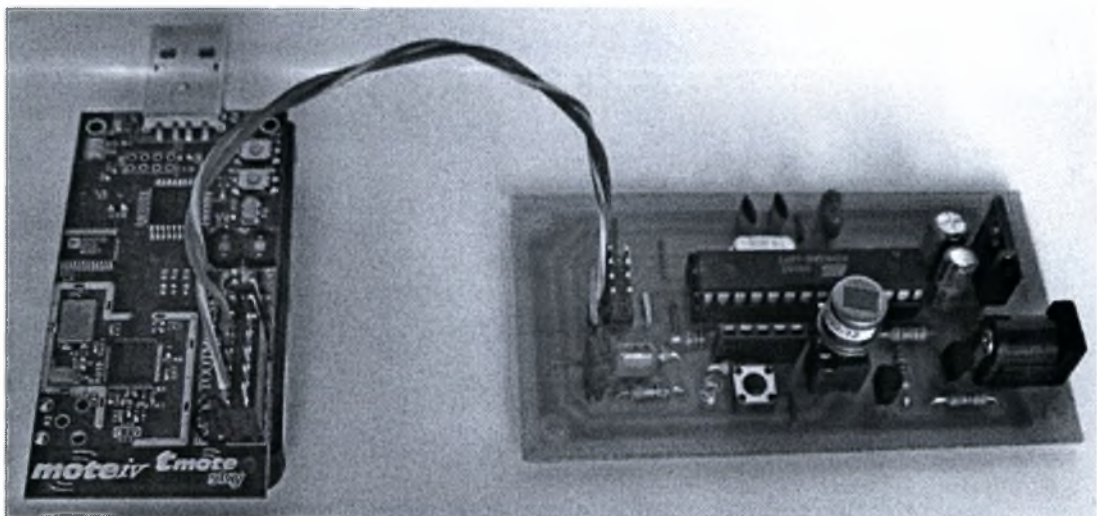
Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Για τις δοκιμές μας επιλέξαμε να χρησιμοποιήσουμε το πρώτο κανάλι του μετατροπέα από αναλογικό σε ψηφιακό, δηλαδή το ADC0 που είναι στην επαφή 3. Επίσης πρέπει να συνδέσουμε και την γείωση του Tmote Sky με την γείωση της πλακέτας μας ώστε οι μετρήσεις να έχουν κοινό δυναμικό αναφοράς.

Η σύνδεση αυτή γίνεται με ένα απλό καλώδιο ως εξής:

pin #	Tmote Sky (connector U2)		πλακέτα αισθητήρα		pin #
3	ADC0	Πρώτο κανάλι μετατροπέα από αναλογικό σε ψηφιακό	Αναλογική έξοδος	Vout	1
9	GND	Γείωση	Γείωση	GND	2

Πίνακας 8.5: Συνδέσεις μεταξύ Tmote Sky και πλακέτας αισθητήρα



Εικόνα 8.5: Σύνδεση μεταξύ Tmote Sky και πλακέτας με αισθητήρα

9 Μετρήσεις

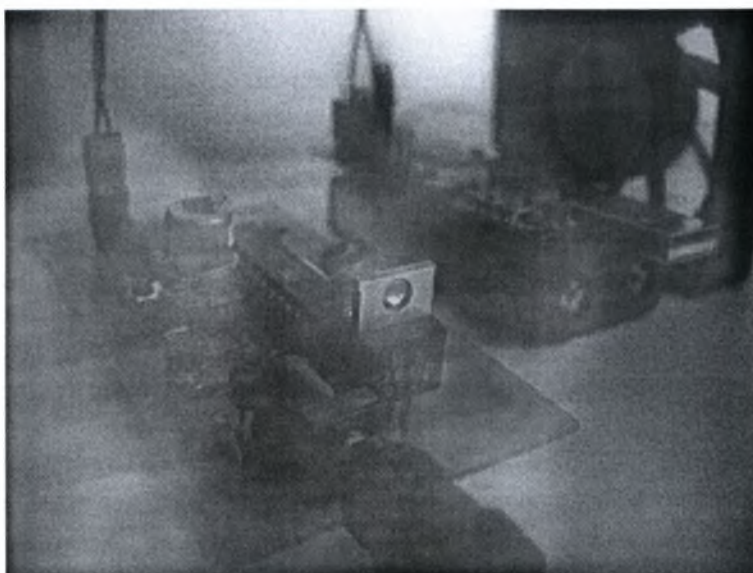
9.1 Περιγραφή εξοπλισμού και διαδικασίας μετρήσεων

Οι δοκιμές του αισθητήρα δεν έγιναν σε κάποιο ειδικό εργαστήριο. Για αυτόν το λόγο έπρεπε αρχικά να κατασκευαστεί ένα αξιόπιστο σύστημα μετρήσεων υπό από ελεγχόμενες συνθήκες. Το σύστημα αυτό πρέπει να επιτρέπει την έκχυση σταθερών ποσοτήτων μίγματος αερίου αμμωνίας με παράλληλη παρακολούθηση της θερμοκρασίας του περιβάλλοντος ώστε να έχουμε συγκρίσιμα αποτελέσματα.

Το αέριο αμμωνίας που χρησιμοποιήθηκε προερχόταν από αναθυμιάσεις φαρμακευτικής αμμωνίας. Η αμμωνία ήταν σε ένα κλειστό δοχείο το οποίο είχε αρκετό διαθέσιμο χώρο ώστε να γεμίζει με αναθυμιάσεις. Η ακριβής τους συγκέντρωση δεν ήταν γνωστή. Αυτό όμως δεν αποτελούσε πρόβλημα για τις μετρήσεις καθώς, σε κάθε επανάληψη του πειράματος, παίρναμε συγκεκριμένους όγκους από αυτό το σταθερής συγκέντρωσης μείγμα. Με αυτόν τον τρόπο η συγκέντρωση της αερίου αμμωνίας που βάζαμε στον θάλαμο των δοκιμών ήταν ανάλογη της ποσότητας του μείγματος που χρησιμοποιούσαμε. Για την ακριβή μέτρηση του μείγματος χρησιμοποιήθηκε μια βαθμονομημένη σύριγγα.

9.1.1 Εξοπλισμός δοκιμών

Ο θάλαμος όπου γινόντουσαν οι μετρήσεις ήταν ένα πλαστικό δοχείο αρκετά μεγάλο ώστε να χωράει την πλακέτα του αισθητήρα και το Tmote Sky για ασύρματη συλλογή δεδομένων. Στο εσωτερικό του διέθετε ένα μικρό ανεμιστήρα ο οποίος φρόντιζε για την ομοιόμορφη ανάμιξη της αμμωνίας με τον αέρα του δοχείου (εικόνα 9.1). Μέσα στο δοχείο υπήρχε επίσης και ένα θερμόμετρο για να μπορούμε να επιβεβαιώνουμε ότι η θερμοκρασία διατηρείται σταθερή καθ' όλη την διάρκεια των μετρήσεων.



Εικόνα 9.1: Φωτογραφία του εσωτερικού του θαλάμου δοκιμών. Στο βάθος διακρίνεται ο ανεμιστήρας ανάμιξης.

9.1.2 Μεθοδολογία μετρήσεων

Η συλλογή των μετρήσεων γινόταν σε πραγματικό χρόνο από υπολογιστή. Η πλακέτα του αισθητήρα συνδεόταν με ένα Tmote που εξέπεμπε ασύρματα τις μετρήσεις σε ένα δεύτερο Tmote το οποίο ήταν στον υπολογιστή. Το Tmote που έπαιρνε τις μετρήσεις είχε εγκατεστημένη μια τροποποιημένη έκδοση του προγράμματος OscilloscopeDemo που υπάρχει στο TinyOS. Ο ρυθμός δειγματοληψίας ορίστηκε στις 4 μετρήσεις ανά δευτερόλεπτο. Η εκπομπή των δεδομένων γινόταν σε πακέτα με 10 μετρήσεις το κάθε ένα. Ο υπολογιστής αποθήκευε αυτούσια αυτά τα πακέτα ώστε στη συνέχεια να είναι δυνατή η περαιτέρω επεξεργασία τους με οποιονδήποτε τόπο.

Τα βήματα που ακολουθήθηκαν σε όλες τις επαναλήψεις του πειράματος ήταν τα εξής:

1. Αερισμός για 2-3 λεπτά του δοχείου σε περιβάλλον καθαρού αέρα (πχ σε ένα ανοιχτό παράθυρο) με παράλληλη λειτουργία του ανεμιστήρα ανάμειξης ώστε να βεβαιωθούμε ότι δεν υπάρχει καθόλου αμμωνία από προηγούμενα πειράματα.
2. Τοποθέτηση της πλακέτας στο κουτί και σύνδεσή της με το Tmote.
3. Κλείσιμο δοχείου.
4. Γέμισμα της σύριγγας με την επιθυμητή ποσότητα.
5. Έναρξη ασύρματης συλλογής μετρήσεων.
6. Έκχυση του περιεχομένου της σύριγγας μέσω μιας μικρής οπής στο δοχείο.
7. Λειτουργία του ανεμιστήρα ανάμειξης για 30 δευτερόλεπτα.
8. Αναμονή μέχρι να σταθεροποιηθεί η μετρούμενη τιμή.
9. Άνοιγμα του δοχείου και λειτουργία του ανεμιστήρα ανάμειξης μέχρι η μετρούμενη τιμή να επανέλθει στα αρχικά της επίπεδα.

9.1.3 Επεξεργασία δεδομένων

Τα πακέτα που είχαν αποθηκευτεί στον υπολογιστή ήταν σε μορφή κειμένου και αποτελούνταν από τις δεκαεξαδικές τιμές όλων των bytes του κάθε μηνύματος. Στον παρακάτω πίνακα φαίνεται η μορφή ενός τέτοιου πακέτου.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...	34	35
type	dest.addr.		src.addr.		size	grp	hnd	version		interval		ID		count		μετρ.1		...	μετρ.10	
00	FF	FF	FF	FF	1C	00	93	00	00	01	00	00	02	12	34	11	11	...	11	11

Το κάθε πακέτο έχει συνολικό μήκος 36 bytes. Τα πρώτα 16 bytes αποτελούν την επικεφαλίδα του μηνύματος και τα υπόλοιπα 20 έχουν τις τιμές των 10 μετρήσεων (2 bytes ανά μέτρηση). Για όσα πεδία καταλαμβάνουν πάνω από 1 byte χρησιμοποιείται κωδικοποίηση big endian.

Για τις ανάγκες των μετρήσεων, μπορούμε να αγνοήσουμε τα πρώτα 14 bytes τα οποία έχουν και πάντα τις ίδιες τιμές όπως φαίνονται στον πίνακα.

Το πεδίο count περιέχει έναν δεκαεξαδικό αριθμό μεγέθους 2 byte ο οποίος διαρκώς αυξάνεται σε κάθε ένα καινούριο πακέτο που δημιουργείται.

Το αρχείο που προκύπτει από το κάθε πείραμα το επεξεργάζεται ένα πρόγραμμα Matlab που αναπτύχθηκε για αυτόν τον σκοπό.

Ο κώδικας αυτού του προγράμματος Matlab, δίνεται στο Παράρτημα.

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Οι μετρήσεις εξάγονται και μετατρέπονται σε δεκαδικές τιμές με τρόπο ώστε να αντιστοιχούν στην μεταβολή της αντίστασης του αισθητήρα $R_{S(GAS)}$ ως προς την τιμή $R_{S(AIR)}$ που είχε όταν ήταν σε καθαρό αέρα. Αυτό γίνεται με τον τρόπο που περιγράφεται παρακάτω.

Η κατασκευή εμφανίζει στην έξοδό της μια τάση V_{out} μεταξύ 0-2,5V που αντιπροσωπεύει τον λόγο $R_{S(gas)}/R_{S(air)}$.

Δηλαδή ισχύει:
$$\frac{R_{S(gas)}}{R_{S(air)}} = \frac{V_{out}}{2,5 V}$$

Ο μετατροπέας από αναλογικό σε ψηφιακό (ADC) του Tmote Sky μπορεί να μετράει τάσεις από 0V μέχρι την τάση αναφοράς 3V. Η τιμή `adc_value` που επιστρέφει είναι 4096 για 0V και 0 για 3V.

Από αυτήν μπορούμε να υπολογίσουμε τον λόγο $R_{S(gas)}/R_{S(air)}$ ως εξής:

$$\frac{R_{S(gas)}}{R_{S(air)}} = \frac{\frac{4096 - \text{adc_value}}{4096} \times 3V}{2,5 V} \Leftrightarrow \frac{R_{S(gas)}}{R_{S(air)}} = \frac{(4096 - \text{adc_value}) \times 3V}{4096 \times 2,5 V}$$

9.2 Πειραματικά δεδομένα

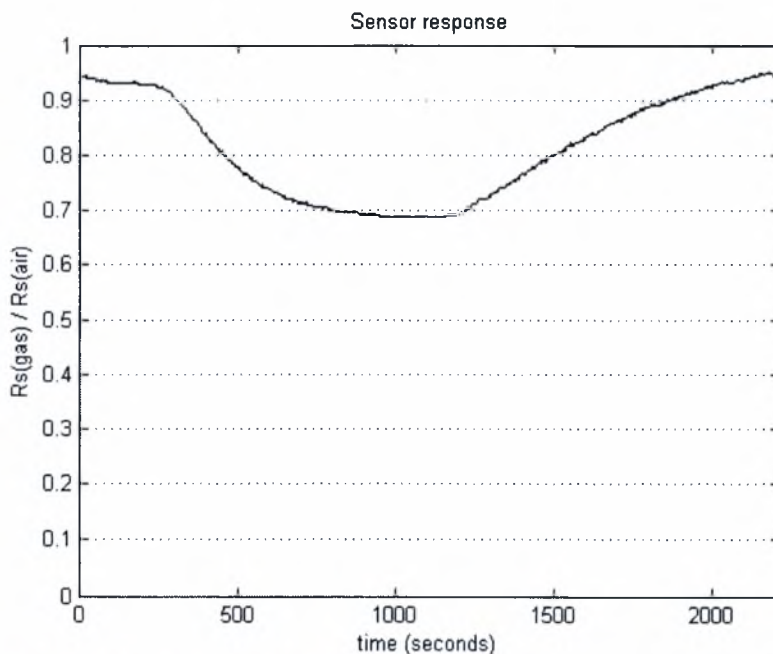
Παρακάτω παρουσιάζονται τα αποτελέσματα πέντε δοκιμών με διαφορετικές συγκεντρώσεις αερίου ώστε να διαπιστωθεί αξιοπιστία των μετρήσεων του αισθητήρα.

Με βάση την τιμή της μέτρησης από τον αισθητήρα γίνεται και ο προσεγγιστικός υπολογισμός της συγκέντρωσης σε ppm με χρήση του διαγράμματος που δίνει ο κατασκευαστής του αισθητήρα (εικόνα 3.7 σελ. 12).

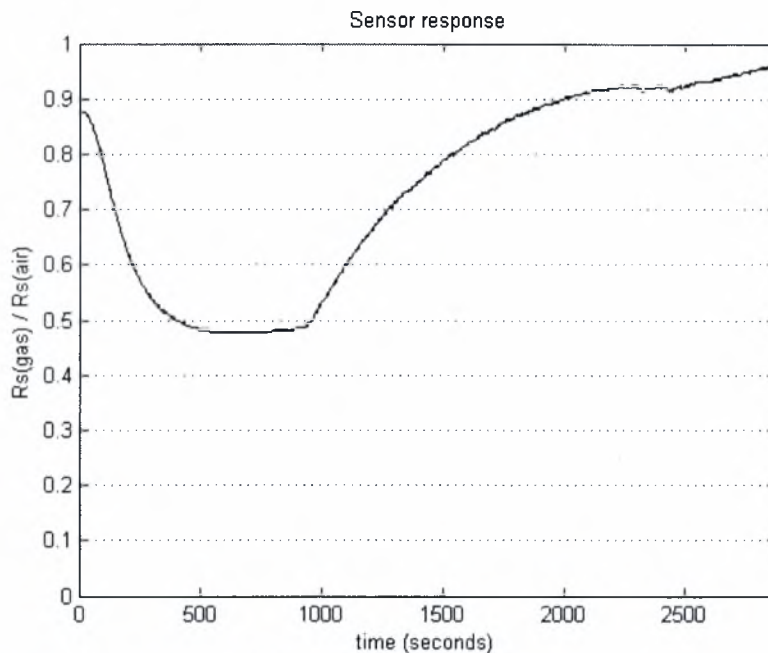
Για το κάθε ένα από τα πειράματα που έγιναν, έχει μετρηθεί ο χρόνος απόκρισης του αισθητήρα. Δηλαδή ο χρόνος που χρειάστηκε από της στιγμή της εισαγωγής του μίγματος αμμωνίας μέχρι την σταθεροποίηση της μετρούμενης τιμής, καθώς και ο χρόνος από την στιγμή του εξαερισμού του θαλάμου μέχρι την επαναφορά της τιμής του αισθητήρα στα φυσιολογικά επίπεδα.

	Πείραμα 1	Πείραμα 2	Πείραμα 3	Πείραμα 4	Πείραμα 5
Όγκος μείγματος NH ₃	0,2ml	0,4ml	0,8ml	2,5ml	7ml
Τιμή Rs/Ro που μετρήθηκε	0,667	0,467	0,385	0,188	0,059
Υπολογιζόμενη συγκέντρωση από datasheet βάσει Rs/Ro	1,2ppm	4ppm	7ppm	25ppm	60ppm
Χρόνος για σταθεροποίηση σε μετρούμενη τιμή	795sec	539sec	531sec	385sec	310sec
Χρόνος για επαναφορά μετά από εξαερισμό	903sec	1041sec	1425sec	2020sec	1551sec

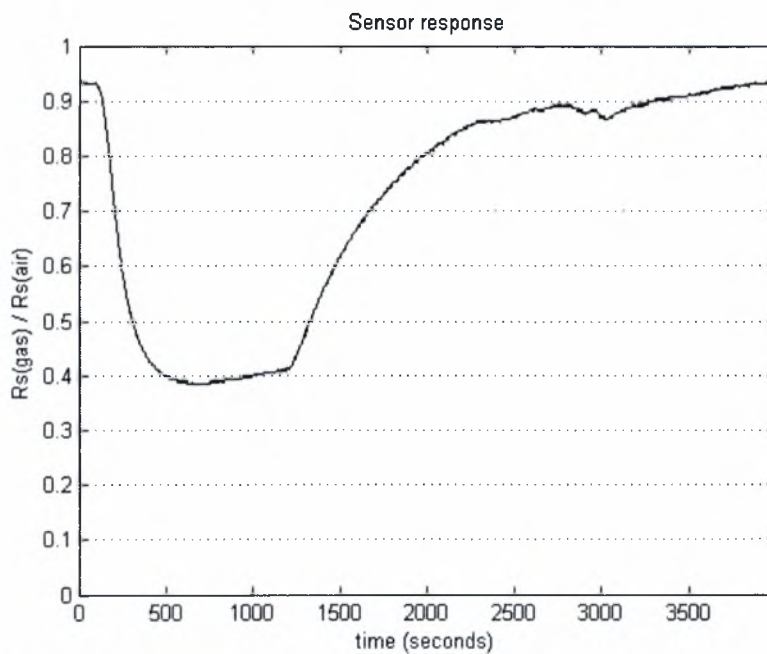
Πίνακας 9.1: Αποτελέσματα πειραματικών μετρήσεων



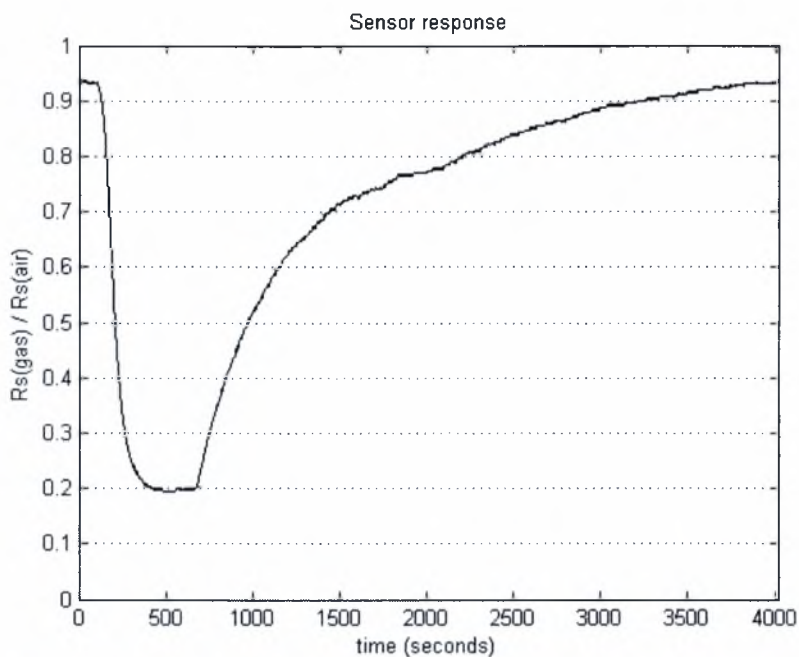
Εικόνα 9.2: Πείραμα 1 (1,2ppm NH₃)



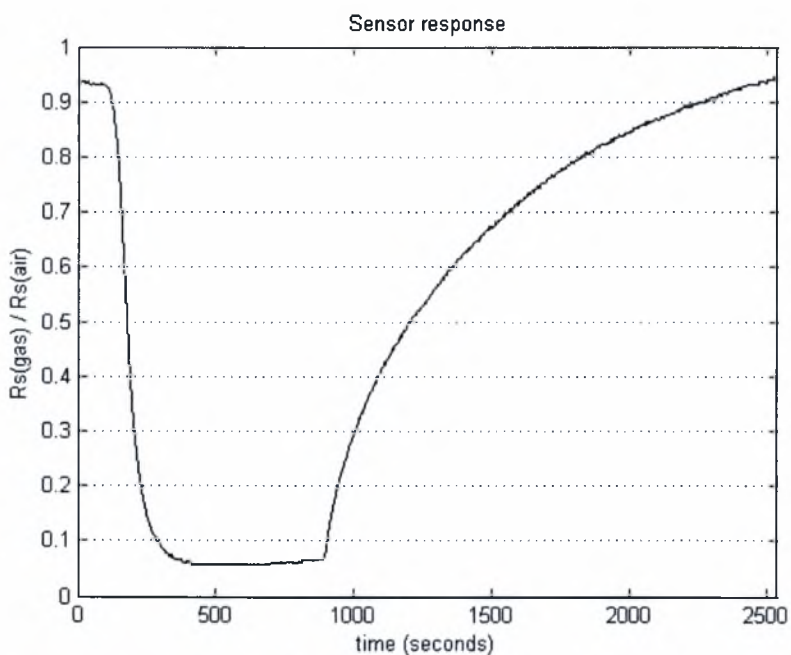
Εικόνα 9.3: Πείραμα 2 (4ppm NH_3)



Εικόνα 9.4: Πείραμα 3 (7ppm NH_3)



Εικόνα 9.5: Πείραμα 4 (25ppm NH₃)



Εικόνα 9.6: Πείραμα 5 (60ppm NH₃)

10 Επίλογος

10.1 Ιδέες για βελτιώσεις και επεκτάσεις

Η σχεδίαση της πλακέτας της κατασκευής όπως παρουσιάστηκε σε αυτή την εργασία έχει γίνει με συμβατικά ηλεκτρονικά εξαρτήματα τα οποία όμως καταλαμβάνουν μεγαλύτερο χώρο πάνω στην πλακέτα. Μια περαιτέρω μείωση του μεγέθους της κατασκευής μπορεί να γίνει χρησιμοποιώντας πλακέτα δύο όψεων και με χρήση εξαρτημάτων επιφανειακής στήριξης (SDM, Surface Mount Device). Αυτό βέβαια αυξάνει τις δυσκολίες και την πολυπλοκότητα κατασκευής της πλακέτας, αλλά μπορεί να μας δώσει μια κατασκευή αρκετά πιο μικρή από την υπάρχουσα.

Μια άλλη πιθανή βελτίωση έχει να κάνει με το κύκλωμα τροφοδοσίας της κατασκευής. Αντί της ύπαρξης ξεχωριστής τροφοδοσίας για τον αισθητήρα και ξεχωριστής για το Tmote Sky, μπορεί να φτιαχτεί ένα κοινό κύκλωμα τροφοδοσίας το οποίο θα είναι πιο αποδοτικό. Τα κυκλώματα στην πλακέτα με τον αισθητήρα λειτουργούν με 5V και το ίδιο γίνεται και με το Tmote Sky όταν τροφοδοτείται από της USB θύρα. Έτσι ένα κοινό κύκλωμα τροφοδοσίας με switching λειτουργία θα μπορούσε να χρησιμοποιηθεί και για τα δύο με αποτέλεσμα να ελαχιστοποιήσουμε στο μεγαλύτερο δυνατό βαθμό τη σπατάλη ενέργειας. Με ένα τέτοιο σύστημα θα μπορούσαμε να έχουμε απόδοση της τάξης του 90%.

Τέλος, αξίζει να αναφέρουμε ότι η κατασκευή που παρουσιάσαμε παραπάνω δεν αξιοποιεί πλήρως όλες τις διαθέσιμες γραμμές εισόδου/εξόδου του μικροελεγκτή. Έτσι με το κατάλληλο λογισμικό θα μπορούσαν να υποστηριχθούν επιπλέον λειτουργίες.

Για παράδειγμα, το σύστημα θα μπορούσε να αυτοματοποιηθεί ώστε ο μικροελεγκτής να έχει τη δυνατότητα αναλόγως με την μετρούμενη συγκέντρωση αμμωνίας να αποφασίζει για τις απαραίτητες ενέργειες (πχ να ενεργοποιεί κάποιο σύστημα εξαερισμού όταν η συγκέντρωση υπερβεί ένα καθορισμένο όριο).

Με αυτόν τον τρόπο οι του κόμβοι ασύρματου δικτύου αισθητήρων απαλλάσσονται από αυτές τις εργασίες και επικεντρώνονται μόνο στην συλλογή των μετρήσεων και την ασύρματη επικοινωνία.

10.2 Συμπεράσματα

Η κατασκευή που παρουσιάστηκε σε αυτήν την εργασία μας έδωσε τη δυνατότητα να δοκιμάσουμε το αισθητήριο TGS2444 το οποίο είναι ένα νέο προϊόν στην οικογένεια των αισθητήρων αμμωνίας. Η σχεδίασή του είναι πρωτοποριακή καθώς καταφέρνει να έχει κατανάλωση πολύ μικρότερη από τα υπόλοιπα προϊόντα που κυκλοφορούν στην αγορά. Οι δυνατότητές του αξιοποιούνται πλήρως από την πλακέτα της κατασκευής μας.

Η σχεδίαση του συστήματος γύρω από έναν ισχυρό μικροελεγκτή όπως είναι ο Atmega8, μας επιτρέπει να κάνουμε εύκολα μελλοντικές αλλαγές και προσθήκες νέων λειτουργιών.

11 Παράρτημα

11.1 Πηγαίος κώδικας λογισμικού μικροελεγκτή

```
main.c
/*****
Author           : Ioannis Kalavros (ioannis.kalavros@gmail.com)
Copyright        : 2008 Ioannis Kalavros
License          : GPLv3

This file is part of Ammonia Sensor.

Ammonia Sensor is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

Ammonia Sensor is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Ammonia Sensor. If not, see <http://www.gnu.org/licenses/>.
*****/

/*****
Chip type        : ATMEGA8
Clock frequency  : External crystal 8 Mhz
*****/

#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 8000000UL // 8 MHz crystal frequency
#include <util/delay.h>
#include <stdlib.h>
#include <avr/eeprom.h>

#include "board_hw.h"
#include "avr_compat.h"
#include "ADC_module.h"
#include "LCD_module.h"
#include "PWM_module.h"
#include "sensor_module.h"
#include "EEPROM_module.h"

unsigned char calibration_mode;
unsigned int Ro_from_eeprom;
unsigned int adc_value;
unsigned char duty;
```

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

```

/*****
Function: static int disp_u_to_adc(int disp)
Purpose:  convert voltage values to adc values, disp=100 is 1.0V
Input:   int adc value
Returns: integer representation of the adc value in volts
*****/
static int disp_u_to_adc(int disp){
    return( (int) (disp*ADC_REF)/10.23 );
}

/*****
Function: void int_to_ascii(int inum, char *outbuf,
                          signed char decimalpoint_pos,
                          signed char spacepadd)
Purpose:  Convert a integer which is representing a float into a string.
          The integer may not be larger than 10000.
          The integer must be a positive number.
Input:   decimalpoint_pos sets the decimal point after 2 pos: e.g 74
          becomes "0.74"
          spacepadd can be used to add a leading space if number is
          less than 10
Returns: none
*****/
static void int_to_ascii(int inum,char *outbuf,signed char
decimalpoint_pos,signed char spacepadd){
    signed char i,j;
    char chbuf[8];
    j=0;
    while(inum>9 && j<7){
        // zero is ascii 48:
        chbuf[j]=(char)48+ inum-((inum/10)*10);
        inum=inum/10;
        j++;
        if(decimalpoint_pos==j){
            chbuf[j]='.';
            j++;
        }
    }
    chbuf[j]=(char)48+inum; // most significant digit
    decimalpoint_pos--;
    while(j<decimalpoint_pos){
        j++;
        chbuf[j]='0';
    }
    if (spacepadd && j > (decimalpoint_pos+2)){
        // no leading space padding needed
        spacepadd=0;
    }
    if(decimalpoint_pos==j){
        j++;
        chbuf[j]='.';
        j++;
        chbuf[j]='0'; // leading zero
    }
    if (spacepadd){

```

```

        j++;
        chbuf[j]=' '; // leading space padding: "9.50" becomes " 9.50"
    }
    // now reverse the order
    i=0;
    while(j>=0){
        outbuf[i]=chbuf[j];
        j--;
        i++;
    }
    outbuf[i]='\0';
}

/*****
Function: void delay_ms(void)
Purpose:  delay for a minimum of given ms (milliseconds) using a
          more accurate calibrated macro
Input:    number of ms
Returns:  none
*****/
void delay_ms(unsigned int ms)
{
    while(ms){
        _delay_ms(1);
        ms--;
    }
}

/*****
Function: void delay_100us(unsigned int us)
Purpose:  delay for a minimum of given 100us (microseconds) using a
          more accurate calibrated macro
Input:    number of 100us
Returns:  none
*****/
void delay_100us(unsigned int us)
{
    while(us){
        _delay_ms(0.1);
        us--;
    }
}

/*****
Function: void toggle_led(void)
Purpose:  Toggles the state of the LED
Input:    none
Returns:  none
*****/
void toggle_led(void)
{
    LED_PORT ^= _BV(LED_PIN);
}

```

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

```

/*****
Function: void toggle_led_on(void)
Purpose: Turns on the LED
Input: none
Returns: none
*****/
void toggle_led_on(void)
{
    cbi(LED_PORT, LED_PIN);
}

/*****
Function: void toggle_led_off(void)
Purpose: Turns off the LED
Input: none
Returns: none
*****/
void toggle_led_off(void)
{
    sbi(LED_PORT, LED_PIN);
}

/*****
Function: void ports_init(void)
Purpose: Initialize ports for Sensor, LED and Button
Input: none
Returns: none
*****/
void ports_init(void)
{
    sbi(SENSOR_HEATER_DDR, SENSOR_HEATER_PIN);
    sbi(SENSOR_SENSOR_DDR, SENSOR_SENSOR_PIN);

    sbi(LED_DDR, LED_PIN); // set LED pin as digital output
    sbi(LED_PORT, LED_PIN); // led is OFF initially (set pin high)

    // turn on internal pull-up resistor for the switch
    sbi(BUTTON_PORT, BUTTON_BIT);
}

/*****
Function: void timer_init(void)
Purpose: Initialize Timer1 for generating interrupts every 250ms
        (this is the sensor operating cycle according to the datasheet)
Input: none
Returns: none
*****/
void timer_init(void)
{
    TCCR1B |= (1 << WGM12); // Configure timer 1 for CTC mode
    TIMSK |= (1 << OCIE1A); // Enable CTC interrupt
    sei(); // Enable global interrupts

    /* Set CTC compare value to 250ms

```

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

```
For 8MHz AVR clock, with a prescaler of 64 */
OCR1A = 31250;

// Start timer at Fcpu/64
TCCR1B |= ((1 << CS10) | (1 << CS11));
}

int main(void)
{
    char out_buf[16+1];

    calibration_mode = 0;

    lcd_init(LCD_DISP_ON);
    ports_init();
    adc_init();
    pwm_init();
    timer_init();

    if (bit_is_clear(BUTTON_PIN, BUTTON_BIT)) // if button pressed
    {
        lcd_puts_p(PSTR("Wait 5sec for"));
        lcd_gotoxy(0, 1); // line 1 y=0, line 2 y=1
        lcd_puts_p(PSTR("Calibration Mode"));

        // button must be pressed continually during this period
        for(char i=9; i>0 && bit_is_clear(BUTTON_PIN, BUTTON_BIT); i--)
        {
            toggle_led();
            delay_ms(500);
        }

        if (bit_is_clear(BUTTON_PIN, BUTTON_BIT))
        {
            //Button pressed for 5sec. Now we are in calibration mode!
            lcd_clrscr();
            lcd_puts_p(PSTR("Calibration Mode"));
            lcd_gotoxy(0, 1); // line 1 y=0, line 2 y=1
            lcd_puts_p(PSTR("starting now!"));

            calibration_mode = 1;

            delay_ms(3000);
        }
    }

    if(calibration_mode == 1)
    {
        /***** We are in CALIBRATION mode *****/

        unsigned long Rs;
```

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

```
//EEPROM init. is not needed. It is done later by writing the Rs
//eeprom_init(); //mark EEPROM as ok and erase data

for(;;)
{
    Rs = ((unsigned long)sensor_Rs_value2(adc_value));

    /* sensor_Rs_value2() returns the Rs divided by 2
    (see also sensor_module.c)
    This is done to avoid overflows in calculations because we
    use 2bytes. Here, in order to display the correct value we
    multiply by 2 (left shifting) */
    ultoa((Rs<<1), out_buf, 10);

    lcd_clrscr();
    lcd_puts_p(PSTR("Rs: "));
    lcd_puts(out_buf);
    lcd_puts_p(PSTR(" Ohm"));
    lcd_gotoxy(0, 1); // line 1 y=0, line 2 y=1
    lcd_puts_p(PSTR("Press to save Rs"));

    if (bit_is_clear(BUTTON_PIN, BUTTON_BIT)) //if button pressed
    {
        // store Rs in EEPROM
        eeprom_write_Rs_default(Rs);

        lcd_clrscr();
        lcd_puts_p(PSTR("Calibration Done"));
        lcd_gotoxy(0, 1); // line 1 y=0, line 2 y=1
        lcd_puts_p(PSTR("Please restart"));

        // turn off LED to indicate that we finished
        toggle_led_off();
        return(0);
    }

    delay_ms(500);
}

/***** end of NORMAL operation mode *****/
/*****

}
else
{
/*****
/***** We are in NORMAL operation mode *****/

// check if it's the very first power-up (EEPROM not initialized)
if (eeprom_read_byte((uint8_t*)E2PROM_INIT_ADDR) != 0xAA)
{
    lcd_clrscr();
    lcd_puts_p(PSTR("EEPROM not ready"));
    lcd_gotoxy(0, 1); // line 1 y=0, line 2 y=1
    lcd_puts_p(PSTR("Please calibrate"));

    /* blink LED to indicate calibration needed */
    for(;;)
    {
```

```
        toggle_led();
        delay_ms(100);
    }

}

/* Read Ro (resistance in clean air from calibration) from EEPROM
The variable is accessed from sensor_module.c
for the calculations */
Ro_from_eeprom = eeprom_read_Rs_default();

/* LCD startup message */
lcd_clrscr();
lcd_puts_p(PSTR("Ammonia Sensor"));
lcd_gotoxy(0, 1); // line 1 y=0, line 2 y=1
lcd_puts_p(PSTR(" by I. Kalavros"));

// light LED for a second to indicate that everything is ok
toggle_led_on();

//wait for some cycles to get a few measurements
delay_ms(1000);

toggle_led_off();

for(;;) {

    /* Convert ADC value to Volts, and then to string */
    int_to_ascii(dis_u_to_adc adc_value),out_buf,2,0);

    lcd_clrscr();
    lcd_puts_p(PSTR("Rs volts: "));
    lcd_puts(out_buf);
    lcd_puts_p(PSTR(" V"));

    lcd_gotoxy(0, 1); // line 1 y=0, line 2 y=1
    lcd_puts_p(PSTR("PWM out: "));
    int_to_ascii(duty,out_buf,0,0);
    lcd_puts(out_buf);
    lcd_puts_p(PSTR("/255"));

    delay_ms(500);
}

/***** end of NORMAL operation mode *****/
/*****

}

return(0); // avoid gcc warning
}
```


Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

```
ISR(TIMER1_COMPA_vect)
{
    cli();          // disable interrupts because timing is critical

    sbi(SENSOR_HEATER_PORT, SENSOR_HEATER_PIN);    // heater on
    delay_ms(2);
    sbi(SENSOR_SENSOR_PORT, SENSOR_SENSOR_PIN);    // sensor on
    delay_ms(4);
    delay_100us(8);

    /* ADC start. The analog value measured 104us after this
    moment according to the datasheet */
    adc_startconversion();

    delay_100us(2);
    cbi(SENSOR_SENSOR_PORT, SENSOR_SENSOR_PIN);    // sensor off
    delay_ms(7);
    cbi(SENSOR_HEATER_PORT, SENSOR_HEATER_PIN);    // heater off

    sei();          // enable interrupts

    adc_value = adc_getresult(); // get the measured ADC value

    if(calibration_mode == 0)
    {
        duty = sensor_value(adc_value);
        pwm_setduty( duty );    // set the PWM duty cycle
    }
}
```

ADC_module.c

```
/******  
Author       : Ioannis Kalavros (ioannis.kalavros@gmail.com)  
Copyright    : 2008 Ioannis Kalavros  
License      : GPLv3
```

This file is part of Ammonia Sensor.

Ammonia Sensor is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Ammonia Sensor is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Ammonia Sensor. If not, see <<http://www.gnu.org/licenses/>>.

```
#include "LCD_module.h"
```

```
void adc_init(void)
```

```
{  
    /* ADC input frequency must be between 50KHz and 200KHz so  
    the prescalar must be set accordingly (datasheet page 208)  
    For 8MHz clock, prescalar must be 64 (110) for sample rate 125KHz */  
    ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (0 << ADPS0);  
    ADMUX |= (0 << REFS0) | (1 << REFS0); // Set ADC reference to AVCC  
    ADMUX |= (0 << ADLAR); // Right (0) adjusted ADC result  
  
    /* Select ADC0 as input (datasheet page 206)  
    All zeros so this line can be omitted*/  
    //ADMUX |= (0 << MUX3) | (0 << MUX2) | (0 << MUX1) | (0 << MUX0);  
  
    ADCSRA |= (1 << ADEN); // Enable ADC  
}
```

```
/******
```

```
Function: void adc_startconversion(void)  
Purpose: orders the ADC unit to start an AD conversion  
Input: none  
Returns: none
```

```
*****
```

```
void adc_startconversion(void)
```

```
{  
    ADCSRA |= (1 << ADSC); // Start A2D Conversion  
}
```

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

```
/******  
Function: uint16_t adc_getresult(void)  
Purpose:  orders the ADC unit to start an ADC conversion  
Input:    none  
Returns:  result of ADC as a 16bit value (000000XX XXXXXXXX)  
*****/  
uint16_t adc_getresult(void)  
{  
    uint16_t adc_value;  
  
    // we read last the ADCLH register because this ensures that they  
    // don't get changed in meanwhile  
    adc_value = ADCL;        // get the measured ADC value  
    adc_value |= (ADCH << 8);  
  
    return adc_value;  
}
```

```
EEPROM_module.c
/*****
Author      : Ioannis Kalavros (ioannis.kalavros@gmail.com)
Copyright   : 2008 Ioannis Kalavros
License     : GPLv3

This file is part of Ammonia Sensor.

Ammonia Sensor is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

Ammonia Sensor is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Ammonia Sensor. If not, see <http://www.gnu.org/licenses/>.
*****/

#include <avr\io.h>
#include <avr\interrupt.h>
#include <avr/eeprom.h>
#include "EEPROM_module.h"

/*****
Function: void eeprom_init(void)
Purpose:  the very first time the device is turned on it erases the eeprom
          area to be used for data storing.
Input:    none
Returns:  none
*****/
void eeprom_init(void)
{
    //erase data area
    //eeprom_write_word((uint16_t*)RS_DEFAULT_ADDR, 0xFFFF);

    // mark eeprom as ok
    eeprom_write_byte((uint8_t*)E2PROM_INIT_ADDR, 0xAA);

    return;
}

/*****
Function: void eeprom_write_Rs_default(unsigned int Ro)
Purpose:  writes the default sensor resistance during calibration
          it also writes the value which means that the EEPROM is
          initialized
Input:    Data to be written
Returns:  none
*****/
```

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

```
void eeprom_write_Rs_default(unsigned int Ro)
{
    eeprom_write_word((uint16_t *)RS_DEFAULT_ADDR, (uint16_t)Ro);

    // mark eeprom as ok by writing the correct value to the
E2PROM_INIT_ADDR
    eeprom_write_byte((uint8_t*)E2PROM_INIT_ADDR, 0xAA); //eeprom init done
}

/*****
Function: unsigned int eeprom_read_Rs_default(void)
Purpose:  read the default sensor resistance saved from calibration
Input:    none
Returns:  sensor resistance
*****/
unsigned int eeprom_read_Rs_default(void)
{
    uint16_t rs;

    rs = eeprom_read_word((uint16_t*)RS_DEFAULT_ADDR);
    return ((unsigned int)rs);
}
```

```
PWM_module.c
/*****
Author       : Ioannis Kalavros (ioannis.kalavros@gmail.com)
Copyright    : 2008 Ioannis Kalavros
License      : GPLv3

This file is part of Ammonia Sensor.

Ammonia Sensor is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

Ammonia Sensor is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Ammonia Sensor. If not, see <http://www.gnu.org/licenses/>.
*****/

#include <avr/io.h>
#include "board_hw.h"
#include "PWM_module.h"

/*****
Function: void pwm_init(void)
Purpose:  initialize the PWM module (select output pin and start timer)
Input:   none
Returns: none
*****/
void pwm_init(void)
{
    PWM_OUT_DDR |= (1<<PWM_OUT_PIN);

    TCNT2=0x00;      // set timer counter initial value
    pwm_setduty(0); // Set timer output compare register

    /* start timer (datasheet page 117):
       without prescaler ( CS22_CS21_CS20 = 001 )
       Inverted mode in OC2 pin ( COM21_COM20 = 10 )
       phase corect PWM ( WGM20 = 1 )
    */
    TCCR2 |= (1<<COM21) | (1<<WGM20) | (1<<CS20);
}

/*****
Function: void pwm_setduty(void)
Purpose:  set the PWM duty cycle
Input:   duty cycle 0-255
*****/
```

Κατασκευή αισθητήρα αμμονίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

Returns: none

*****/

```
void pwm_setduty(unsigned char duty) {  
    OCR2 = (unsigned) duty;  
}
```

```
sensor_module.h
/*****
Author      : Ioannis Kalavros (ioannis.kalavros@gmail.com)
Copyright   : 2008 Ioannis Kalavros
License     : GPLv3

This file is part of Ammonia Sensor.

Ammonia Sensor is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

Ammonia Sensor is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Ammonia Sensor. If not, see <http://www.gnu.org/licenses/>.
*****/

#include "sensor_module.h"
#include "board_hw.h"
#include "EEPROM_module.h"

/*****
Function: unsigned char sensor_value(unsigned int adc_vout)
Purpose:  returns an 8bit number corresponding the Rs/Ro
          Rs is sensor's resistance in gas, Ro_from_eeprom is resistance
          in clean air
          That means 255 in clean air, arround 10 in 100ppm NH3
Input:    ADC measured voltage 0-1023 (corresponding 0-5Volts)
Returns:  returns an 8bit number corresponding 255*Rs/Ro
*****/
unsigned char sensor_value(unsigned int adc_vout)
{
    extern unsigned int Ro_from_eeprom;

    unsigned long RsRo;

    /* Rs = (Vc*Rl)/Vout - Rl
    RsRo = Rs/Ro
    Vc is 5Volts which means 1023 in 10bit representation (like ADC values)
    We must return 255*RsRo because we want values 255-0
    be carefull to do calculations with correct order to avoid overflow!! */

    //RsRo = 255 * ( ((1023*(unsigned long)R_LOAD)/adc_vout) - R_LOAD ) /
Ro ;

    unsigned int Rs = sensor_Rs_value2(adc_vout);

    if(Rs >= Ro_from_eeprom)
    {
```


Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

```
        /* if measured Rs is bigger than Ro we return the
        maximum value (avoiding overflow) */
        RsRo = 255;
    }
    else
    {
        RsRo = 255 * (unsigned long)Rs / Ro_from_eeprom;
    }

    return ((unsigned char)(RsRo));
}

/*****
Function: unsigned int sensor_Rs_value2(unsigned int adc_vout)
Purpose:  calculates the measured sensors resistance Rs
          In order to avoid overflows the return vale is the measured
          Rs divided by 2 (Rs/2)
Input:    ADC measured voltage 0-1023 (corresponding 0-5Volts)
Returns:  Rs/2 (half the value of Rs) as 16bit integer
*****/
unsigned int sensor_Rs_value2(unsigned int adc_vout)
{
    unsigned long Rs;

    /* In order to avoid overflow (return value is 2bytes), we divide by 2
    the resistor values in our calculation
    That means that e.g. instead of 20000 Ohms we suppose 10000
    The final calculation in sensor_value() is a division between resistor
    values so the result is not affected */

    Rs = ( ( 1023*(unsigned long)(R_LOAD>>1) )/adc_vout ) - (R_LOAD>>1);

    return ((unsigned int)Rs);
}
```

11.2 Κώδικας Matlab

```

parse_mesages.m
clear;

filename = 'SerialMessages_test.txt';
column_count_MSB = 15; %count value is in columns 15 and 16
column_count_LSB = 16;
column_data = 17;      %data are in 17th column until the end of line

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

adc_count = 0;          %counter for the measurements
adc_temp = 0;

fid = fopen(filename, 'r');      %open file

while ~feof(fid)
    line = fgetl(fid);
    Values = sscanf(line, '%x');

    d_count = 256*Values(column_count_MSB) + Values(column_count_LSB);

    for i=0:9
        position = column_data + 2*i;
        d_adcvalue = 256*Values(position) + Values(position+1);
        %fprintf('%d ', d_adcvalue);

        adc_count = adc_count + 1;
        adc_values(adc_count) = (3 * (4096 - d_adcvalue)/4096) / 2.5;

        % The tmotesky returns 4096 for 0V and 0 for 3V, so we invert and
        % then scale it to 3V
        % The NH3 sensor returns values from 0-2,5V. We divide by 2,5 to
        % get the sensor's change ratio
    end
    %fprintf('\n');
end

fclose(fid);      %close file

mv = 4;           % plots the mean value every 'mv' measurements
n = 1;
temp=0;
for t=1:adc_count
    if ( mod(t, mv) == 0 )
        temp = temp / mv;
        plot_values(n) = temp;
        n = n + 1;
    else
        temp = temp + adc_values(t);
    end
end
end

```

Κατασκευή αισθητήρα αμμωνίας για εφαρμογές δικτύων ασυρμάτων αισθητήρων

```
plot(plot_values);
title('Sensor response');
%legend('Test Measurements' , 'datasheet formula', 'polyfit function');
xlabel('time (seconds)');
ylabel('Rs (gas) / Rs (air)');

set(gca, 'XScale', 'linear', 'YScale', 'linear');
set(gca, 'xlim', [0,n]);
set(gca, 'ylim', [0,1]);
set(gca, 'XGrid', 'off', 'YGrid', 'on');
set(gca, 'XColor', [0.3,0.3,0.3], 'YColor', [0.3,0.3,0.3]);
set(gca, 'LineWidth', 1);
```

TGS2444_temperature.m

```

clc;
clear;

% These are the actual data from the test measurements
% x is the temperature in degrees Celcius
% y is Rs(air)/Rs(air at 20 degrees Celcius)
x = -10:10:50;
y = [2.8, 2.12, 1.12, 1.0, 0.72, 0.61, 0.53];
semilogy(x, y, 'r.');
```

$$yy(i) = -1 \cdot 10^{(-5)} \cdot x(i)^3 + 0.0014 \cdot x(i)^2 - 0.0753 \cdot x(i) + 1.9435;$$

```

axis([-20, 60, 0.1, 10]);
hold on;

% yy holds the values calculated by the formula given in datasheet
[sr,sc] = size(x);
for i=1:sc,
    yy(i) = -1*10^(-5)*x(i)^3 + 0.0014*x(i)^2 - 0.0753*x(i) + 1.9435;

% These below are the polyfit coefficients rounded to 6 decimal digits
% yy(i) = -1*10^(-5)*x(i)^3 + 0.001449*x(i)^2 - 0.075488*x(i) + 1.942143;
end

semilogy(x, yy, 'k-');
```

$$p = \text{polyfit}(x, y, 3);$$

```

fprintf('Polyfit coefs: %2.6f %2.6f %2.6f %2.6f \n', p(1), p(2), p(3), p(4) );
yp = polyval(p,x);
plot(x, yp, 'b-');
```

```

%grid on;
title('Temperature Dependency');
legend('Test Measurements', 'datasheet formula', 'polyfit function');
xlabel('Temperature degrees C');
ylabel('Rs(air) / Rs(air 20 degrees C)');
```

```

set(gca, 'XScale', 'linear', 'YScale', 'log');
set(gca, 'XGrid', 'off', 'YGrid', 'on');
set(gca, 'XColor', [0.0,0.0,0.0], 'YColor', [0.0,0.0,0.0]);
set(gca, 'LineWidth', 1);
```

Βιβλιογραφία

- [1] Figaro, Operating principle of semiconductor type gas sensors, <http://www.figaro.co.jp/en/item2.html>
- [2] Figaro, TGS2444 datasheet, <http://www.figaro.co.jp/en/pdf/2444ProductInfo0507.pdf>
- [3] CadSoft, EAGLE, <http://www.cadsoft.de/>
- [4] Atmel, ATmega8 datasheet, http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf
- [5] Atmel, AVR068: STK500 Communication Protocol, http://www.atmel.com/dyn/resources/prod_documents/doc2591.pdf
- [6] Atmel, AVR069: AVRISP mkII Communication Protocol, http://www.atmel.com/dyn/resources/prod_documents/doc8015.pdf
- [7] Guido Socher, AvrUsb500v2 - an open source Atmel AVR Programmer stk500 v2 compatible with USB interface, <http://tuxgraphics.com/electronics/200705/article07052.shtml>
- [8] FTDI, FT232R USB UART I.C., http://www.ftdichip.com/Documents/DataSheets/DS_FT232R.pdf
- [9] Atmel, AVR studio 4, http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725
- [10] WinAVR, AVR-GCC, <http://winavr.sourceforge.net/>
- [11] Fairchild Semiconductor, LM78XX/LM78XXA, <http://www.fairchildsemi.com/ds/LM/LM7805.pdf>
- [12] Hitachi, Dot Matrix Liquid Crystal Display Controller/Driver, <http://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- [13] Texas Instruments, MSP430F1611, <http://focus.ti.com/lit/ds/symlink/msp430f1611.pdf>



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ



004000091498