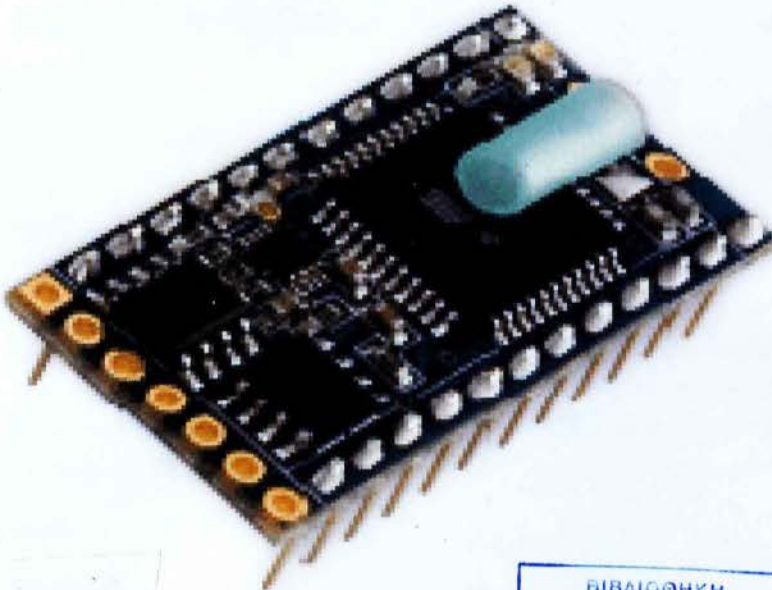


Πτυχιακή Εργασία του σπουδαστή
Παναγιώτη Παναγόπουλου

Υπεύθυνος καθηγητής: Νικόλαος Καλοκάσης

Θέμα: Αυτοματοποιημένο σύστημα πυρόσβεσης με την
χρήση μικροεπεξεργαστή(BX-24p)

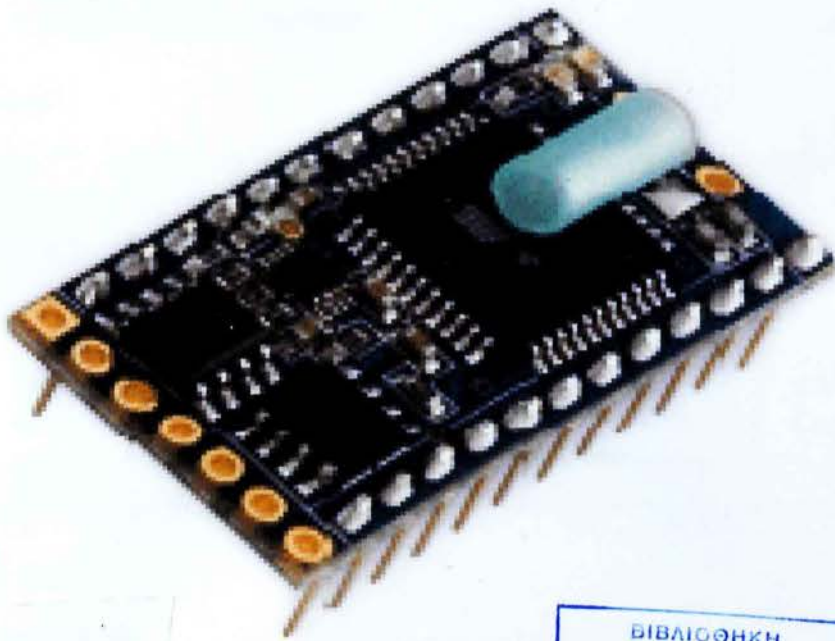


ΒΙΒΛΙΟΘΗΚΗ
ΤΕΙ ΠΕΙΡΑΙΑ

Πτυχιακή Εργασία του σπουδαστή Παναγιώτη Παναγόπουλου

Υπεύθυνος καθηγητής: Νικόλαος Καλοκάσης

Θέμα: Αυτοματοποιημένο σύστημα πυρόσβεσης με την
χρήση μικροεπεξεργαστή(BX-24p)



ΒΙΒΛΙΟΘΗΚΗ
ΤΕΙ ΠΕΙΡΑΙΑ

Περιεχόμενα

Κεφάλαιο 1

- Γενικές πληροφορίες για την πυρόσβεση.....σελ.6
- Καθήκοντα πυροσβέστη και ατυχήματα πυρκαγιών.....σελ.7
- Στοιχεία φωτιάς.....σελ.8
- Μέθοδοι καταπολέμησης πυρκαγιών.....σελ.9

Κεφάλαιο 2

- Περιγραφή Πτυχιακής.....σελ. 14
- Κύκλωμα Πτυχιακής.....σελ.15
- Περιγραφή εξαρτημάτων.....σελ.16

Κεφάλαιο 3

- Πληροφορίες BX-24r και Atmel AVR ATmega8535.....σελ.18
- Μπλοκ διάγραμμα Atmel AVR ATmega8535.....σελ.21
- Διαμόρφωση pin Atmel AVR ATmega8535.....σελ.22
- SPI EEPROM.....σελ.23
- Τεχνικά χαρακτηριστικά BX-24r.....σελ.26
- BX-24r αρίθμηση pin.....σελ.27
- BX-24r ορισμοί pin.....σελ.28
- BX-24r DC χαρακτηριστικά.....σελ.29

Κεφάλαιο 4

- Αισθητήριο LM-35.....σελ.31
- Τυπικές εφαρμογές LM-35.....σελ.32
- Απώλυτες τιμές και ηλεκτρικά χαρακτηριστικά LM-35.....σελ.34

Κεφάλαιο 5

- Εκτέλεση και κώδικας πειράματος.....σελ.38
- Επεξήγηση κώδικα προγράμματος.....σελ.41
- Μεταφορά κώδικα από υπολογιστή στον BX-24r.....σελ.43
- Λειτουργία του προγράμματος.....σελ.46

Κεφάλαιο 6

- EEPROM.....σελ.51
- Λειτουργίες των EEPROM.....σελ.52
- Σύγκριση μεταξύ EPROM και EEPROM/Flash.....σελ.54

Κεφάλαιο 7

- Αισθητήρια (sensors).....σελ.56
- Ταξινόμηση μετρήσιμων λαθών και αποκλίσεις.....σελ.57

Κεφάλαιο 8

- Χρησιμοποιώντας ADC για να διαβάσουμε ένα ποτενσιόμετρο...σελ.60
- Χρησιμοποιώντας το ADXL202 επιταχυνσιόμετρο με τον BX-24p.....σελ.62
- Μπλοκ δεδομένων κλάσεων.....σελ.66
- Αλληλεπίδραση πλήκτρων και διακόπτων.....σελ.70
- Μετρώντας παλμούς με Hardware διακοπές.....σελ.73
- ADC Ατζέντα δεδομένων 8 καναλιών.....σελ.75
- Προγραμματίζοντας τον Timer1 για διπλό παλμό διαμόρφωσης πλάτους.....σελ.78
- Προγραμματίζοντας τον Timer1 για σύλληψη αλλαγής παλμού...σελ.81
- Διασύνδεση υπέρυθρων αισθητηρίων με τον BX-24p.....σελ.86
- Διασύνδεση πληκτρολογίων με τον BX-24p.....σελ.88
- Ελέγχοντας Servos με τον BX-24p.....σελ.91
- Χρησιμοποιώντας την Rctime για να μετρήσουμε αντίσταση.....σελ.94

Βιβλιογραφία.....σελ.100

Περίληψη

Σε αυτήν την μελέτη ο αναγνώστης θα εμβαθύνει στον κόσμο των μικρο-επεξεργαστών και των αισθητήριων, εξερευνώντας τις δυνατότητές τους, κατανοώντας των τρόπο λειτουργίας τους και μαθαίνοντας τρόπους συνεργασίας αυτών των δύο εξαρτημάτων. Η παρούσα πτυχιακή εργασία πραγματεύεται την μελέτη, ανάπτυξη και κατασκευή ενός αυτοματοποιημένου συστήματος πυρόσβεσης με την χρήση μικρο-επεξεργαστή. Παρέχεται πλήρης θεωρητική και τεχνική μελέτη με την χρήση πινάκων τεχνικών χαρακτηριστικών, επεξήγηση του κώδικα λειτουργίας του προγράμματος, υπολογισμοί κατά την διάρκεια λειτουργίας του κυκλώματος, φωτογραφίες πριν και μετά την ολοκλήρωση της κατασκευής και ηλεκτρονικό σχέδιο.

Επίσης παρέχονται γενικές πληροφορίες όσον αφορά τη διαδικασία της πυρόσβεσης σε διάφορες καταστάσεις και περιβάλλοντα, τις μνήμες EEPROM και τα αισθητήρια σε μια μεγαλύτερη κλίμακα εφαρμογών. Τέλος, παρέχεται ένας αριθμός εναλλακτικών εφαρμογών με την χρήση του μικρο-επεξεργαστή BX-24p.

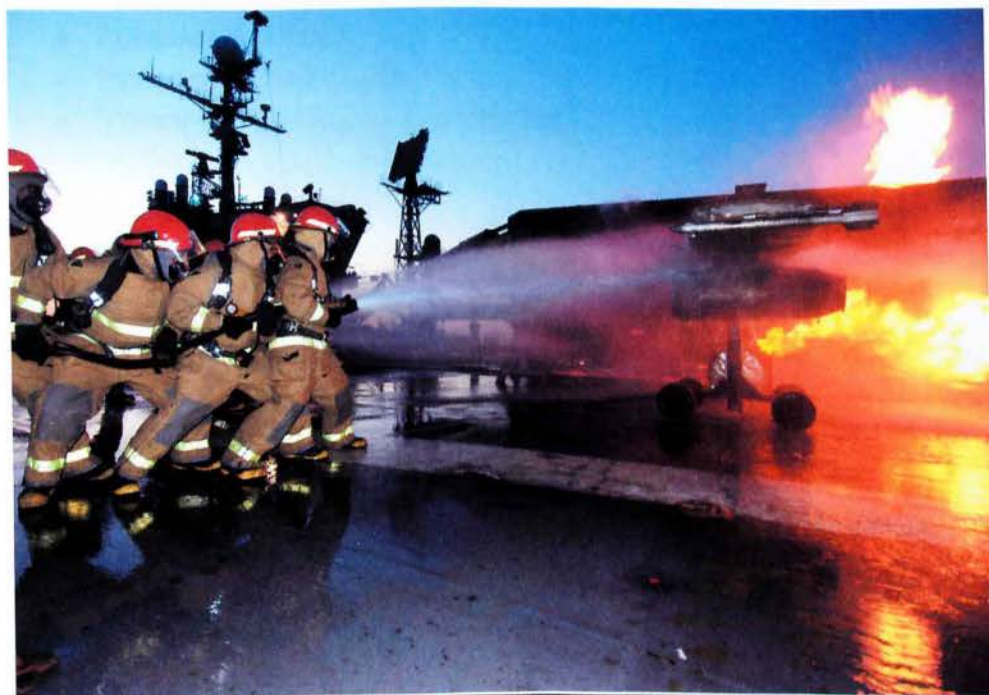
Κεφάλαιο 1

Γενικές πληροφορίες για την πυρόσβεση και τις φωτιές

Πυρόσβεση

Γενικές πληροφορίες

Η πυρόσβεση είναι η ενέργεια κατά την οποία εξολοθρεύουμε καταστροφικές φωτιές. Ο πυροσβέστης αντιμετωπίζει αυτές τις φωτιές για να αποτρέψει την καταστροφή της ζωής και της προστασίας του περιβάλλοντος. Η πυρόσβεση είναι μια πολύ τεχνική ιδιότητα που απαιτεί από τους επαγγελματίες χρόνια εκπαίδευσης και προετοιμασίας για να γίνουν αποτελεσματικοί.



Ομάδα πυρόσβεσης πάνω στο
αεροπλανοφόρο USS John F. Kennedy
αντιμετωπίζοντας μια κατάσταση
πυρκαγιάς στις 2 Μαΐου 2006

Καθήκοντα πυροσβέστη

Ο σκοπός ενός πυροσβέστη είναι η προστασία της ζωής, της ιδιοκτησίας και του περιβάλλοντος. Μια φωτιά μπορεί να επεκταθεί ταχύτατα και να απειλήσει πολλές ζωές. Με τις σύγχρονες μεθόδους πυροπροστασίας, η καταστροφή συνήθως αποφεύγεται. Για να αποτρέψουμε την εκκίνηση των πυρκαγιών, τα καθήκοντα ενός πυροσβέστη περιλαμβάνουν την εκπαίδευση του κοινού και συχνές επιθεωρήσεις για εστίες πυρκαγιών.

Επειδή οι πυροσβέστες είναι συχνά οι πρώτοι που απαντούν στις εκκλήσεις των ανθρώπων σε κρίσιμες καταστάσεις, προσφέρουν αρκετές άλλες χρήσιμες υπηρεσίες στην κοινότητα, όπως επείγον ιατρικές υπηρεσίες, σαν επείγοντες ιατροί ή επικυρωμένοι βοηθοί γιατρού ή προσωπικό ασθενοφόρων.

- Μετριασμός επικίνδυνων υλικών
- Δύσκολες διασώσεις
- Εντοπισμός και διάσωση
- Υποστήριξη καταστροφής κοινότητας

Οι πυροσβέστες επίσης χρησιμοποιούνται στα παρακάτω πεδία

- Αεροπλάνα/διάσωση αερομεταφορών
- Καταστολή φωτιάς εδάφους
- Φωτιά σε πλοία
- Καταπολέμηση φωτιάς στρατού
- Τακτική υποστήριξη βοηθών γιατρού
- Ανύψωση με εργαλεία

Ατυχήματα που προκαλούνται από τις πυρκαγιές

Ο πιο πρωτεύον κίνδυνος σε μια φωτιά για τους ανθρώπους είναι η εισπνοή επικίνδυνων ουσιών. Οι πιο πολλοί άνθρωποι που πεθαίνουν σε μια φωτιά είναι από αυτή την αιτία και όχι από τα εγκαύματα. Οι κίνδυνοι αυτοί περιλαμβάνουν:

- Ασφυξία από εισπνοή φωτιάς ή ανεπάρκεια οξυγόνου
- Δηλητηριώδη αέρια που παράγονται από την φωτιά
- Αναρρόφηση καπνού και δηλητηρίαση στους πνεύμονες

Η θερμότητα αναγκάζει την ανθρώπινη σάρκα να καεί σαν καύσιμο προκαλώντας σοβαρά ιατρικά προβλήματα. Ανάλογα με την ένταση της φωτιάς, τα εγκαύματα μπορούν να προκληθούν μέσα σε δευτερόλεπτα. Ένα έγκαυμα πρώτου βαθμού (στην επιφάνεια του δέρματος) είναι εξαιρετικά επώδυνο. Ένα έγκαυμα δεύτερου βαθμού διαπερνάει το δέρμα και προκαλεί σοκ, μολύνσεις και αφυδάτωση και εάν δεν αντιμετωπιστεί, συχνά οδηγεί τον θάνατο. Εγκαύματα τρίτου βαθμού καταστρέφουν τον νευρικό ιστό και δεν είναι επώδυνα, ωστόσο οι άκρες τέτοιων εγκαυμάτων είναι δεύτερου και πρώτου βαθμού και είναι αρκετά επώδυνα. Εάν το άτομο αποφύγει το σοκ και την έκθεση σε μικροοργανισμούς, η ιατρική βοήθεια είναι πολύ δύσκολη.

Περεταίρω κίνδυνοι της πυρόσβεσης είναι:

- Η όραση ελαττώνεται με την φωτιά. Ένα άτομο μέσα σε ένα κτήριο μπορεί αν μην είναι ικανό να δει, μπορεί να πέσει ή να αποπροσανατολιστεί. Έτσι παγιδεύεται και πεθαίνει από τον καπνό και την φωτιά
- Το κτήριο μπορεί να καταρρεύσει στα θεμέλια του

Στοιχεία φωτιάς

Είναι τέσσερα τα στοιχεία που απαιτούνται για να ξεκινήσει μια πυρκαγιά και να διατηρηθεί. Αυτά τα στοιχεία είναι ιεραρχημένα στο λεγόμενο “Τετράεδρο Φωτιάς”. Τα τέσσερα αυτά στοιχεία είναι:

- Καύσιμο
- Θερμότητα
- Αυτόνομη χημική αλυσιδωτή αντίδραση
- Οξυγόνο

Το καύσιμο είναι η ουσία που καίγεται κατά την διάρκεια της καύσης. Τα πιο συνήθη καύσιμα περιέχουν άνθρακα σε συνδυασμό με υδρογόνο και οξυγόνο. Η θερμότητα είναι το ενεργειακό κομμάτι του τετράεδρου. Όταν η θερμότητα έρχεται σε επαφή με το καύσιμο, παρέχει την απαραίτητη ενέργεια για ανάφλεξη, προκαλώντας την συνεχή παραγωγή και ανάφλεξη των ατμών του καύσιμου ώστε να καθιστά δυνατή την συνεχόμενη αντίδραση αυτών των στοιχείων. Μια συνεχόμενη αντίδραση είναι μια σειρά γεγονότων, κατά την οποία τα αποτελέσματα κάθε ξεχωριστής πράξης προστίθενται στα υπόλοιπα. Αυτό συμβαίνει στην επιστήμη της φωτιάς, αλλά είναι αυτόνομη, συνεχίζει χωρίς διακοπή. Ένας παράγοντας οξειδωσης είναι μια ουσία που στις

κατάλληλες συνθήκες απελευθερώνει οξυγόνο. Αυτό είναι κρίσιμο για την διατήρηση της φωτιάς.

Μια φωτιά σβήνει όταν εξαλείψουμε έστω ένα από τα στοιχεία του “τετράεδρου”. Μια μέθοδο για να σβήσουμε μια φωτιά είναι η χρήση του νερού. Ο πρώτος τρόπος που το νερό αντιμετωπίζει μια φωτιά είναι με την ψύξη, η οποία αφαιρεί θερμότητα από την φωτιά. Αυτό είναι δυνατόν εξαιτίας της ιδιότητας του νερού να απορροφά μεγάλα ποσά θερμότητας διαμέσου της μετατροπής του σε ατμό. Χωρίς θερμότητα, η φωτιά δεν μπορεί να συνεχίζει να οξειδώνεται για να διατηρηθεί. Ο δεύτερος τρόπος που το νερό αντιμετωπίζει μια φωτιά είναι με την μέθοδο της κατάπνιξης. Όταν το νερό θερμαίνεται στο σημείο βρασμού του, μετατρέπεται σε ατμό. Όταν γίνεται αυτή η μετατροπή, εξασθενίζει το οξυγόνο στην ατμόσφαιρα με ατμό, δηλαδή αφαιρεί ένα από τα κυρίαρχα στοιχεία για την διατήρηση της φωτιάς. Αυτό μπορεί να επιτευχθεί επίσης με αφρό.

Μέθοδοι καταπολέμησης φωτιάς

Πυροσβεστήρας

Ο πυροσβεστήρας (σχήμα 1) είναι μια συσκευή σχεδιασμένη να αντιμετωπίζει μικρές φωτιές, συχνά σε καταστάσεις έκτακτης ανάγκης. Δεν είναι σχεδιασμένοι να λειτουργούν σε φωτιές που είναι εκτός ελέγχου, όπως αυτές που η φωτιά έχει φτάσει στο ταβάνι, έχουν βάλει σε κίνδυνο τον χρήστη ή απαιτούν την αντιμετώπισή τους από πυροσβέστη. Τυπικά, ένας πυροσβεστήρας αποτελείται από κυλινδρικό δοχείο υπό πίεση το οποίο περιέχει μια ουσία για την καταπολέμηση της φωτιάς.

Υπάρχουν δύο κύριοι τύποι πυροσβεστήρων: αποθηκευμένης πίεσης και εξωτερικής θήκης. Στους πυροσβεστήρες αποθηκευμένης πίεσης, το απωθητικό υλικό είναι στον ίδιο θάλαμο με το υλικό κατάσβεσης. Ανάλογα με το υλικό κατάσβεσης, χρησιμοποιείται και το κατάλληλο υλικό προώθησης. Με πυροσβεστήρες ξερών χημικών συνήθως χρησιμοποιείται άζωτο. Πυροσβεστήρες νερού και αφρού συνήθως χρησιμοποιούν αέρα. Οι πυροσβεστήρες αποθηκευμένης πίεσης είναι οι πιο συνηθισμένοι.

Οι πυροσβεστήρες εξωτερικής θήκης περιέχουν το υλικό προώθησης σε ξεχωριστό θάλαμο από αυτόν που βρίσκεται το υλικό κατάσβεσης. Τα δύο αυτά υλικά αναμιγνύονται στην έξοδο για να δουλέψει ο πυροσβεστήρας.

Αυτός ο τύπος πυροσβεστήρα δεν είναι συνηθισμένος και χρησιμοποιείται κυρίως σε περιοχές όπως βιομηχανικές εγκαταστάσεις όπου λαμβάνουν υψηλότερη από συνήθως χρήση. Έχουν το πλεονέκτημα της απλής επαναφόρτισης, επιτρέποντας στον χρήστη να τον χρησιμοποιήσει και να τον επαναφορτίσει στη συνέχεια μέσα σε ένα σύντομο χρονικό διάστημα ώστε να συνεχίσει την κατάσβεση. Σε αντίθεση με τους πυροσβεστήρες αποθηκευμένης πίεσης, οι πυροσβεστήρες ξεχωριστής θήκης χρησιμοποιούν συμπιεσμένο διοξείδιο του άνθρακα αντί για άζωτο, αν και άζωτο χρησιμοποιείται σε θήκες πολύ χαμηλής θερμοκρασίας (-60°C). Οι πυροσβεστήρες εξωτερικής θήκης είναι διαθέσιμοι σε εκδόσεις ξηρού χημικού και ξηρής πούδρας στην Αμερική και σε νερού, υγρού στοιχείου, αφρού, ξηρών χημικών (κλάσεις ABC και D) στον υπόλοιπο κόσμο.



Σχήμα 1
Πυροσβεστήρας κυλινδρικού
τύπου

Πυρόσβεση αέρος

Το Bombardier (πρώην Canadair) CL-415 (SuperScooper) είναι ένα Καναδικό αμφίβιο αεροσκάφος, ειδικά σχεδιασμένο για ρόλους αεροπυρόσβεσης. Είναι το μόνο αεροσκάφος σχεδιασμένο και κατασκευασμένο αποκλειστικά για αυτόν το ρόλο και βασίζεται στο CL-215 της ίδιας εταιρίας.

Το CL-415 (σχήμα 2) έκανε την πρώτη του πτήση το Δεκέμβριο του 1993 και οι πρώτες παραδόσεις άρχισαν τον Νοέμβριο του 1994. Εφοδιάζεται από δύο νέους τούρμπο κινητήρες Pratt & Whitney Canada PW123AF, αντικαθιστώντας τους εμβολοφόρους κινητήρες του CL-215, ενώ έχει και αναβαθμισμένο κόκπιτ υγρών κρυστάλλων. Επίσης έγιναν πολλές αεροδυναμικές βελτιώσεις με την προσθήκη πτερυγίων και τον επανασχεδιασμό μερών της ατράκτου.

Το CL-415 έχει ικανότητα υδροληψίας μέχρι 6,140 λίτρα νερού από υδάτινες επιφάνειες (θάλασσες, λίμνες) που βρίσκονται κοντά στην περιοχή που έχει πιάσει φωτιά, το οποίο μπορεί να εμπλουτίσει με επιβραδυντικό αφρό και να ρίξει στη φωτιά, επαναλαμβάνοντας τη διαδικασία χωρίς να χρειάζεται να επιστρέψει στην βάση του για ανεφοδιασμό. Επίσης σχεδιάστηκε έτσι ώστε να μπορεί να ρίχνει μεγάλες ποσότητες επιβραδυντικού υγρού, στα πρώτα στάδια της φωτιάς, αποτρέποντας την από το να βγει εκτός ελέγχου. Παράλληλα έγινε διαθέσιμη και μία έκδοση πολλαπλών ρόλων του CL-415 (CL-415MP) για χρήση ως αεροσκάφος έρευνας και διάσωσης ή ελαφρύ μεταγωγικό.



Σχήμα 2

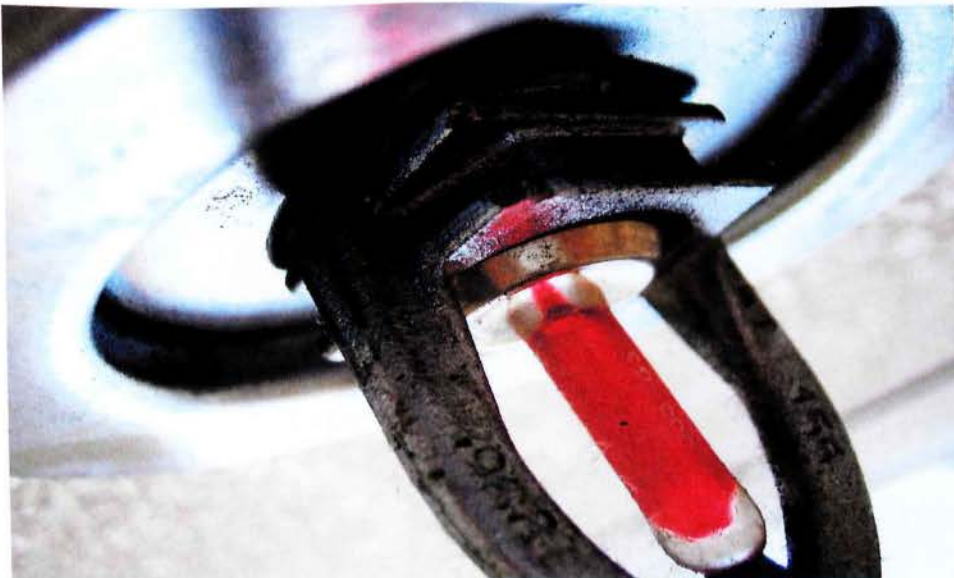
Ψεκαστήρας Φωτιάς

Ο ψεκαστήρας φωτιάς (σχήμα 3) είναι ένα τμήμα ενός αυτοματοποιημένου συστήματος που ρίχνει νερό όταν εντοπίζονται τα στοιχεία μιας φωτιάς, όπως είναι ο έντονος καπνός ή η υπέρβαση μιας προκαθορισμένης τιμής θερμοκρασίας.

Λειτουργία

Κάθε κεφαλή του ψεκαστήρα είναι ερμητικά κλειστή είτε από έναν γυάλινο βολβό ευαίσθητο στην θερμοκρασία, είτε από μεταλλικό σύνδεσμο δύο μερών που συγκρατούνται μεταξύ τους με εύηκτο κράμα μετάλλων. Ο γυάλινος βολβός ασκεί πίεση σε ένα καπάκι που εμποδίζει το νερό από το να τρέξει μέχρι η περιβάλλον θερμοκρασία γύρω από τον ψεκαστήρα να φτάσει την θερμοκρασία ενεργοποίησής του. Επειδή κάθε ψεκαστήρας ενεργοποιείται μεμονωμένα όταν το προκαθορισμένο όριο θερμοκρασίας επιτευχθεί, ο αριθμός των ψεκαστήρων που θα ενεργοποιηθούν ελαχιστοποιείται μόνο σε αυτούς που είναι γύρω από την φωτιά ώστε να έχουμε μέγιστη πίεση νερού στο συγκεκριμένο σημείο.

Ο γυάλινος βολβός σπάει σαν αποτέλεσμα της θερμικής διαστολής του υγρού που έχει μέσα του. Ο χρόνος που απαιτείται για τον βολβό να σπάσει εξαρτάται από την θερμοκρασία γύρω του. Κάτω από την θερμοκρασία ενεργοποίησης δεν συμβαίνει τίποτα ενώ πάνω από αυτήν απαιτείται λιγότερος χρόνος για τις υψηλότερες θερμοκρασίες. Ο χρόνος αντίδρασης εκφράζεται σαν Response Time Index (RTI), και τυπικά παίρνει τιμές από 35 έως $250 \text{ m}^{1/2}\text{s}^{1/2}$.



Σχήμα 3

Κεφάλαιο 2

Περιγραφή πτυχιακής

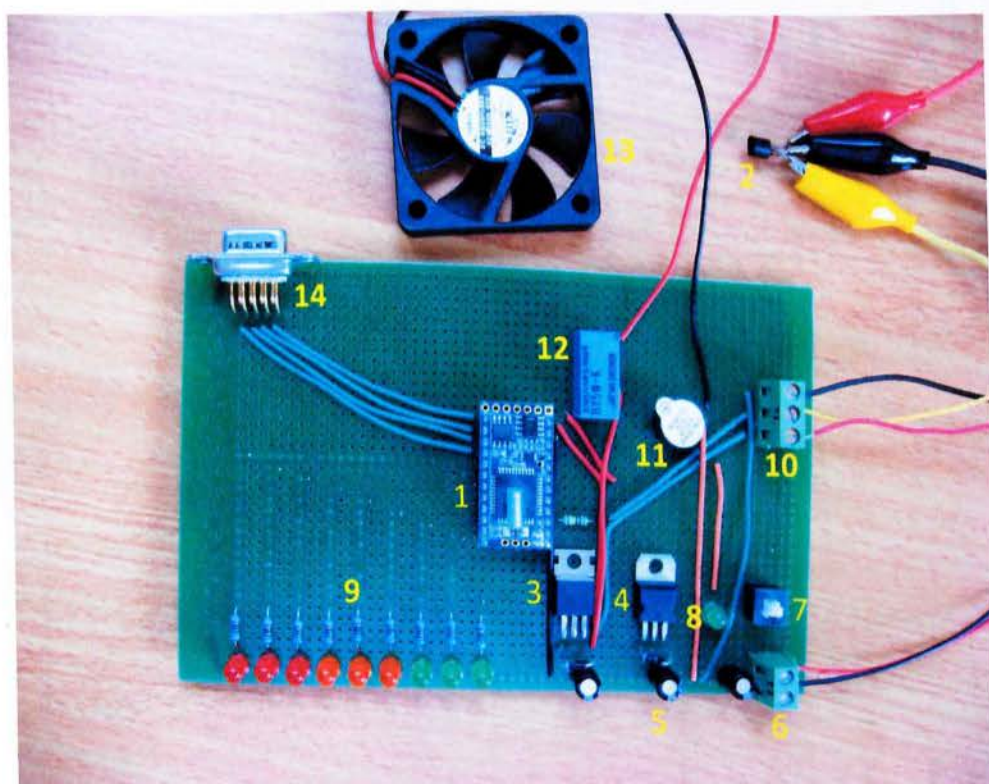
Πρόγραμμα



Περιγραφή: Με την χρήση του μικροεπεξεργαστή και συγκεκριμένα του BX-24p και των ADC εισόδων του, θα γίνει η κατασκευή ενός αυτοματοποιημένου συστήματος πυρόσβεσης (σχήμα 1) .

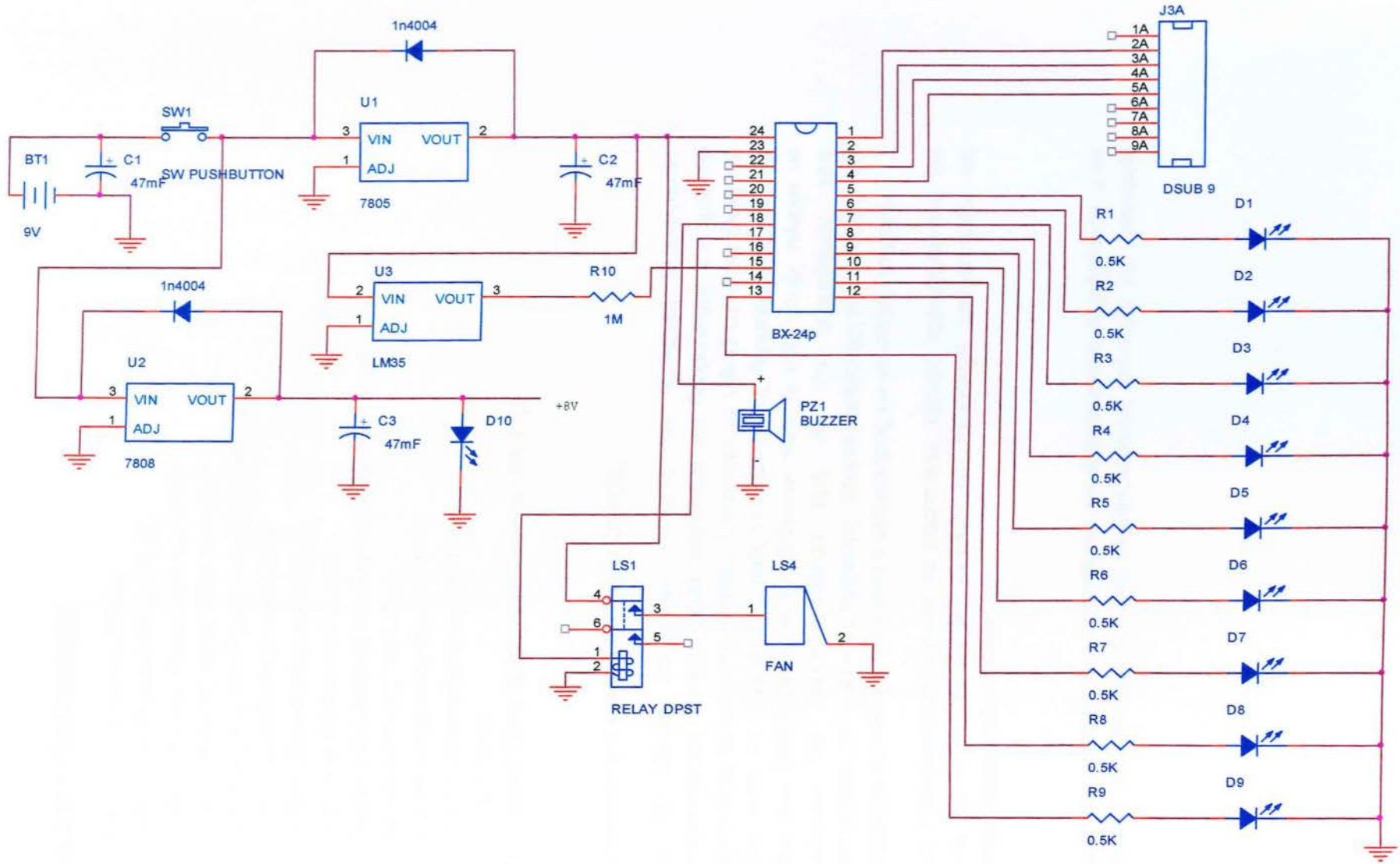
Θα γίνει χρήση πηγής θερμότητας (κερί). Με την βοήθεια του αισθητηρίου LM-35, θα διεξάγονται συνεχείς μετρήσεις της θερμοκρασίας, οι οποίες θα οδηγούνται στην ADC είσοδο του μικροεπεξεργαστή BX-24p. Μόλις οι τιμές αυτές θα ξεπερνούν μια συγκεκριμένη μέγιστη τιμή η οποία θα προσδιορίζεται εκ των προτέρων, ο μικροεπεξεργαστής θα αναλαμβάνει να σβήνει το κερί, με την ενεργοποίηση ενός ανεμιστήρα. Παράλληλα θα φροντίζει να ενημερώνει για την παραβίαση της συγκεκριμένης τεθείσας μέγιστης τιμής με την χρήση οπτικών και ακουστικών σημάτων.

Κύκλωμα



Σχήμα 1

Ολοκληρωμένο κύκλωμα πτυχιακής



Title		
Automated fire-extinguisher system with the use of the BX-24p microcontroller		
Size	Document Number	Rev
A		

Περιγραφή εξαρτημάτων

1. Μικροεπεξεργαστής BX-24p
2. Αισθητήριο θερμοκρασίας LM-35
3. Σταθεροποιητής τάσης 7805 των 5Volt
4. Σταθεροποιητής τάσης 7808 των 8Volt
5. Πυκνωτές των 47mF μαζί με διόδους IN4004
6. Κλέμα τροφοδοσίας κυκλώματος
7. Button τροφοδοσίας κυκλώματος
8. Led τροφοδοσίας κυκλώματος
9. Σειρά LED ένδειξης της θερμοκρασίας του LM-35
10. Κλέμα σύνδεσης του LM-35
11. Buzzer υπέρβασης θερμοκρασίας
12. Ρελιέ ενεργοποίησης ανεμιστήρα
13. Ανεμιστήρας
14. Σειριακή θύρα προγραμματισμού του BX-24p

Η λειτουργία του κυκλώματος έχει ως εξής:

Μέσω του LM-35, διαβάζουμε συνεχώς την εξωτερική θερμοκρασία. Χρησιμοποιώντας 2 κεριά ρεσό, αρχίζουμε και θερμαίνουμε ένα χάλκινο σωλήνα(καλός αγωγός θερμότητας). Εισάγουμε το LM-35 στην μία άκρη του σωλήνα ώστε να πάρουμε τις τιμές μεταβολής της θερμοκρασίας. Με την βοήθεια των LED ένδειξης καταλαβαίνουμε πότε οι τιμές αυτές τείνουν να πλησιάσουν την μέγιστη δοθείσα τιμή. Αν την ξεπεράσουν, τότε ενεργοποιείται το buzzer υπέρβασης θερμοκρασίας(ηχητική ένδειξη) και αναλαμβάνει να τεθεί σε λειτουργία ο ανεμιστήρας για να σβήσει τα ρεσό.

Για να πραγματοποιηθούν όλα τα παραπάνω, έχουμε προγραμματίσει τον BX-24p με το κατάλληλο set εντολών και το έχουμε εισάγει μέσω της σειριακής θύρας ενός υπολογιστή.

Σε αυτό το σημείο, είναι απαραίτητο να εμβαθύνουμε περισσότερο στον σχεδιασμό του κυκλώματος ώστε να καταλάβουμε καλύτερα την λειτουργία του.

Κεφάλαιο 3

Πληροφορίες και τεχνικά χαρακτηριστικά για τον ΒΧ-24p



Εικόνα 1

Μεταλλοτεχνείο

ΒΧ-24p

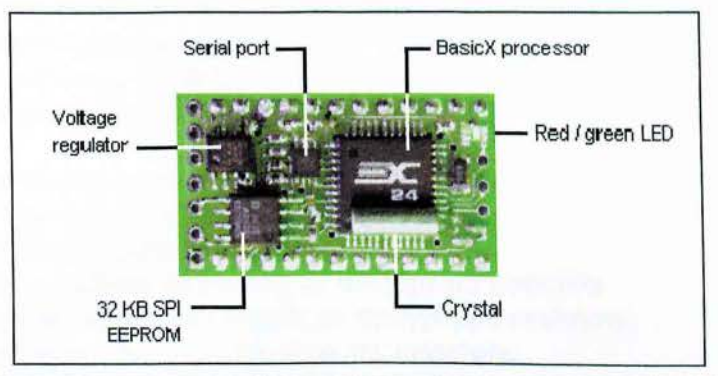
Περιγραφή

Ο ΒΧ-24p είναι ένα μικρό, ελαφρύ, και εύκολο στην χρήση όχημα. Είναι κατάλληλο για χρήση σε αστικές περιοχές και σε χώρους με περιορισμένο χώρο. Το όχημα είναι εξοπλισμένο με όλα τα απαραίτητα εξαρτήματα για να μπορεί να λειτουργήσει ως αυτοκίνητο ή ως φορτηγό. Η ταχύτητα του οχήματος είναι 24 km/h.

Το όχημα είναι κατασκευασμένο από ανθεκτικό πλαστικό και έχει μια ισχυρή κινητήρα. Η μπαταρία του οχήματος είναι φορτιζόμενη και μπορεί να φορτιστεί σε οποιοδήποτε σημείο που υπάρχει ηλεκτρικό ρεύμα. Η διάρκεια ζωής της μπαταρίας είναι 24 ώρες. Το όχημα είναι επίσης εξοπλισμένο με φώτα και κωδωνάκι.

Όπως γίνεται κατανοητό, δύο είναι τα πιο σημαντικά στοιχεία του συγκεκριμένου κυκλώματος και τα οποία χρειάζονται περισσότερη θεωρητική μελέτη. Ο μικροεπεξεργαστής BX-24p και το αισθητήριο LM-35.

BX-24p



Σχήμα 1

Μικροεπεξεργαστής BX-24p

BasicX επεξεργαστής

Ο πυρήνας του BX-24p (σχήμα 1) είναι ο επεξεργαστής BasicX ο οποίος είναι βασισμένος στο Atmel AVR ATmega8535 τσιπ. Αυτό το custom-programmed 44 chip διαβάζει και εκτελεί τα προγράμματα που είναι αποθηκευμένα στην 32 KB EEPROM που διαθέτει.

Ο BX-24p έχει 16 εισόδους I/O γενικής χρήσης που είναι TTL and CMOS compatible. Όταν χρησιμοποιούνται για ψηφιακές I/O, κάθε είσοδος μπορεί να βρεθεί σε τέσσερα states – output high, output low, input tri-state and input with pull-up. Μέχρι 8 από τις 16 εισόδους μπορούν εναλλακτικά να χρησιμοποιηθούν σαν 10-bit analog to digital converters(ADCs) για να διαβάζουν αναλογικά voltage.

Atmel AVR ATmega8535

Ο πυρήνας του BX-24p (σχήμα 2) είναι ένας AVR 8-bit επεξεργαστής χαμηλής κατανάλωσης.

Χαρακτηριστικά του:

- Αναβαθμισμένη RISC αρχιτεκτονική
 - 130 εντολές – περισσότερες επεξεργασίες μονού ρολογιού
 - 32 x 8 καταχωρητές γενικού
 - Πλήρης στατική επεξεργασία
 - Μέχρι 16 MIPS στα 16 MHz
 - Ενσωματωμένος 2-cycle πολλαπλασιαστής

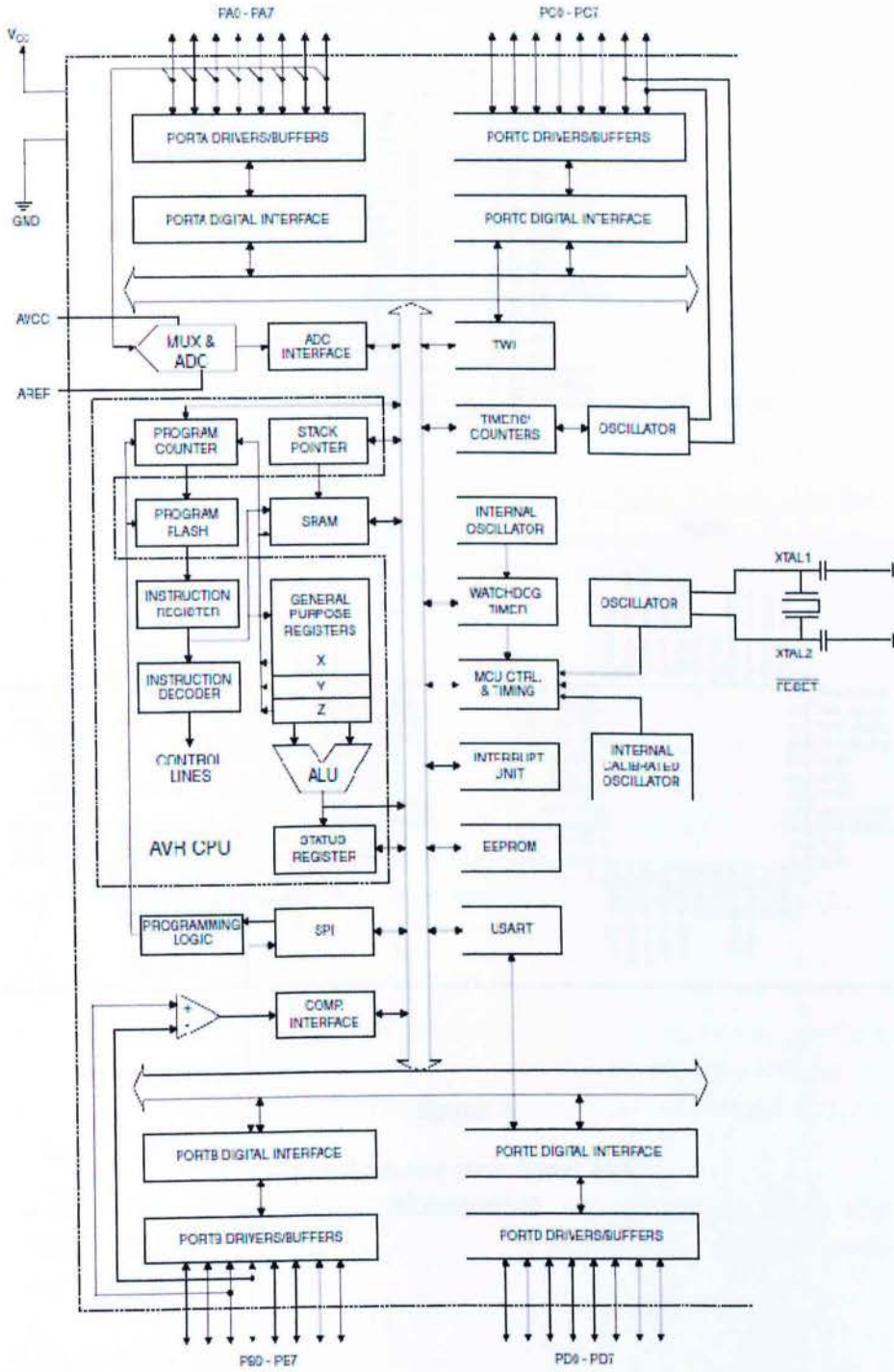
- Μνήμες μη-τάσης για προγράμματα και δεδομένα
 - 8K Bytes προγραμματιζόμενης μνήμης flash
 - Διάρκεια: 10,000 εγγραφές/διαγραφές
 - Επιλεγμένος κώδικας φόρτωσης με ανεξάρτητα Lock Bits
 - Ενσωματωμένος προγραμματισμός με πρόγραμμα εκκίνησης
 - Δυνατότητα ανάγνωσης κατά την ώρα της εγγραφής
 - 512 Bytes EEPROM
 - Διάρκεια: 10,000 εγγραφές/διαγραφές
 - 512 Bytes εσωτερική SRAM
 - Προγραμματιζόμενο πρόγραμμα ασφαλείας

- Περιφερειακά χαρακτηριστικά
 - Δύο 8-bit χρονιστές/μετρητές με ξεχωριστά mode σύγκρισης
 - Ένας 16-bit χρονιστής/μετρητής με ξεχωριστό mode σύγκρισης και σύλληψης
 - Ρολόι αληθινού χρόνου
 - Τέσσερα PWM κανάλια
 - 8-κάναλος, 10-bit ADC
 - 8 Single-ended κανάλια
 - 7 διαφορεικά κανάλια για QFP
 - 2 διαφορεικά κανάλια με προγραμματιζόμενο κέρδος στα 1x, 10x, or 200x για TQFP
 - Byte-αρχιτεκτονικής δύο -καλωδίων σειριακό περιβάλλον
 - Προγραμματιζόμενο σειριακό USART
 - Master/Slave SPI σειριακό περιβάλλον
 - Προγραμματιζόμενο ρολόι ελέγχου με ξεχωριστό ενσωματωμένο ταλαντωτή
 - Ενσωματωμένο αναλογικός συγκριτής

- Ειδικά χαρακτηριστικά μικροελεγκτή Special
 - Power-on Reset and προγραμματιζόμενο Brown-out ανιχνευτής
 - Ενσωματωμένος ρυθμισμένος RC ταλαντωτής
 - Εξωτερικές και εσωτερικές πηγές διακοπής σήματος
 - Έξη Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby
 - και Extended Standby

- I/O και πακέτα
 - 32 προγραμματιζόμενες I/O γραμμές
 - 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad QFN/MLF
- Τάσεις λειτουργίας
 - 2.7 - 5.5V για ATmega8535L
 - 4.5 - 5.5V για ATmega8535
- Βαθμοί ταχύτητας
 - 0 - 8 MHz για ATmega8535L
 - 0 - 16 MHz για ATmega8535

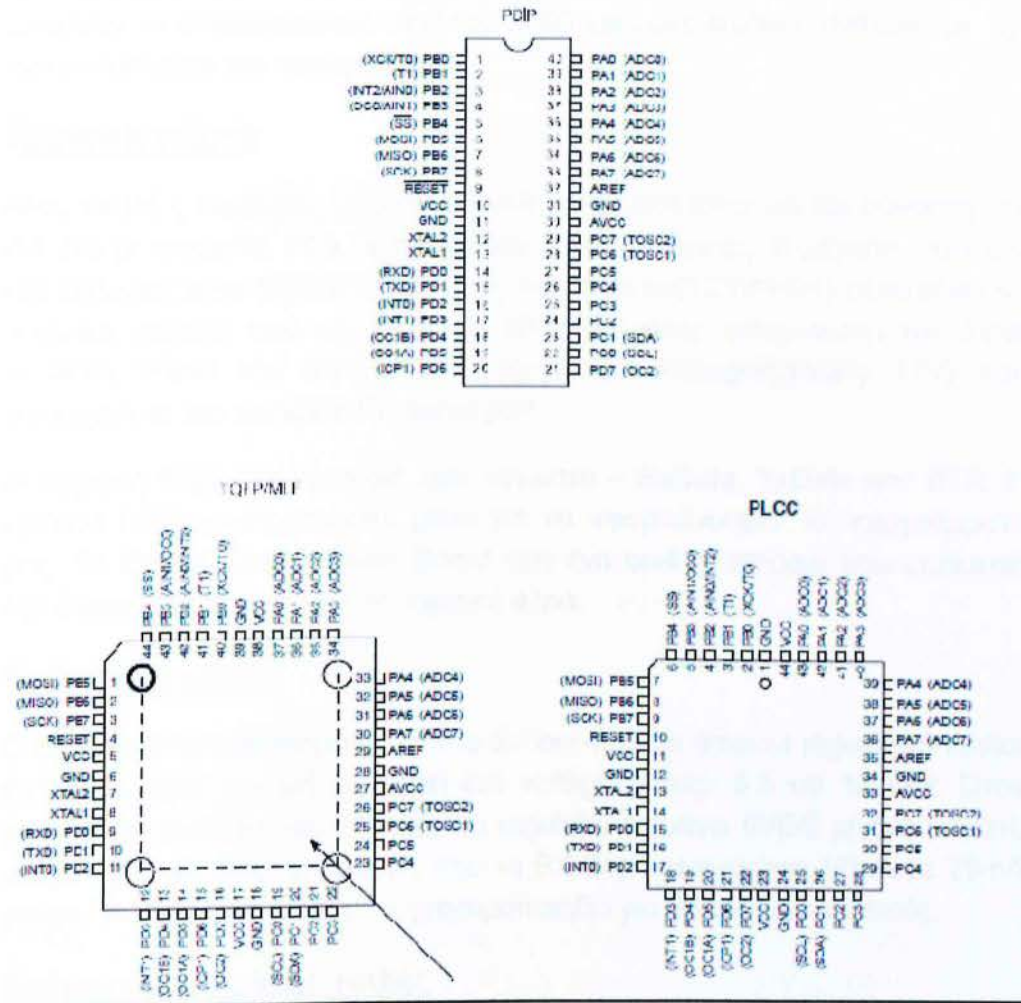
Μπλοκ Διάγραμμα



Σχήμα 2

Μπλοκ διάγραμμα Atmel AVR
ATmega8535

Διαμόρφωση Pin



Σχήμα 3

Διαμόρφωση pins Atmel AVR
ATmega8535

SPI EEPROM τσιπ

Όταν γράφεται ένα πρόγραμμα, αποθηκεύεται στο SPI EEPROM chip. Όταν ο BasicX εκτελείται, παίρνει οδηγίες από το chip. Τα 32 KB EEPROM(AT25656) μπορούν να αποθηκεύσουν περίπου 8000 γραμμές κώδικα, ανάλογα με την πολυπλοκότητα του προγράμματος.

Σειριακή πόρτα

Μιας υψηλής ταχύτητας θύρα των 5Volt χρησιμοποιείται για την σύνδεση του BX-24p με modems, PCs, terminals και άλλους ελεγκτές. Η μέγιστη ταχύτητα επικοινωνίας είναι 460800 baud. Ένας hex inverter(TC7WH04) μετατρέπει να σειριακά σήματα από και προς το BX-24p. Επίσης απομονώνει την 5Volt σειριακή πόρτα του BX-24p από υψηλότερα voltage(typically 12V) που υπάρχουν σε μια standard PC serial port.

Η σειριακή θύρα χρησιμοποιεί τρία σύρματα – RxData, TxData and DTR. Η γραμμή DTR χρησιμοποιείται μόνο για να «φορτώνουμε» τα προγράμματά μας. Το BasicX Development Board έχει ένα built-in window που επιτρέπει την 2-way επικοινωνία με την σειριακή θύρα.

Ρυθμιστής τάσης

Ο BX-24p είναι εφοδιασμένος με ένα 5V low-voltage dropout regulator/monitor (LP2951). Αυτό μπορεί να δεχτεί ένα voltage μεταξύ 5.5 και 12 Volt. Όταν συνδέουμε τάση στο pin 24 (Vin), το regulator παράγει 5VDC με ένα μέγιστο ρεύμα 100 mA. Από την στιγμή που το BX-24p καταναλώνει 17mA με 25mA ρεύμα, το υπόλοιπο μπορεί να χρησιμοποιηθεί για δικούς μας σκοπούς.

Ρυθμιστής χαμηλής τάσης

Για να αποτρέψουμε το BX-24p από το κλειδώσει ή να εκτελεί λανθασμένα κατά την ενεργοποίησή του ή ατά οποιαδήποτε περίοδο low-voltage, το BX-24p διαθέτει ένα low-voltage monitor. Αυτό είναι ένα εσωτερικό κομμάτι του on-board regulator chip.

Ελέγχει συνέχεια το system voltage level. Αν αυτό πέσει κάτω από τα 4.75Volt, τότε θέτει τον BX-24p σε reset μέχρις ότου το voltage να ανέβει σε κανονικές τιμές.

Αναλογικός σε ψηφιακός μετατροπέας

Ο BX-24p διαθέτει 8 εισόδους, 10-bit analog to digital converters (ADC). Οι εισοδοί αυτοί είναι δεσμευμένες στα pin 13-20. Και οι 8 είσοδοί μπορούν να χρησιμοποιηθούν σαν analog or digital inputs.

Οι ADC είσοδοι είναι της τάξης των 0V to 5V και δεν μπορούν να ανεχθούν μεγαλύτερες ή αρνητικές τιμές. Για αξιόπιστες ADC μετατροπές, συνιστάται η κοινή γείωση του voltage που μετράμε με την γείωση του BX-24p.

Χάρτης Μνήμης

Ram – Αποθηκευμένη μέσα στο τσιπ του επεξεργαστή. Χρησιμοποιούμε Ram peek, RAMpoke για απευθείας πρόσβαση.

Αρχική διεύθυνση: 207

Τελική διεύθυνση: 607

Μέγεθος: 401 bytes

Μόνιμη μνήμη -- Αποθηκευμένη μέσα στο τσιπ του επεξεργαστή. PersistentPeek, PersistentPoke για απευθείας πρόσβαση.

Αρχική διεύθυνση: 32

Τελική διεύθυνση: 511

Μέγεθος: 480 bytes

EEPROM memory – αποθηκευμένη σε ένα ξεχωριστό SPI EEPROM chip. Χρησιμοποιούμε GetEEPROM, PutEEPROM για απευθείας πρόσβαση. Ο κώδικας του προγράμματος είναι αποθηκευμένος εδώ.

Αρχική διεύθυνση: 0

Τελική διεύθυνση: 32767

Μέγεθος: 32768 bytes

Καταχωρητές ώρας και ημερομηνίας

The BX-24p έχει ένα ενσωματωμένο ρολόι/ημερολόγιο που λειτουργεί στα 512 Hz. Κάθε παλμός του ρολογιού αυξάνει το Register.RTCTick, ο οποίος είναι ένας 32 bit ακέραιος. Η τιμή αλλάζει κάθε 24 ώρες.

Η ημερομηνία είναι αποθηκευμένη σαν ένας αριθμός στο Register.RTCDay. Ημέρα 0 is 1 January 1999. Ο αριθμός είναι ένας 16 bit μη μαρκαραρισμένος ακέραιος, που αυξάνει κάθε φορά που το Register.RTCTick αλλάζει.

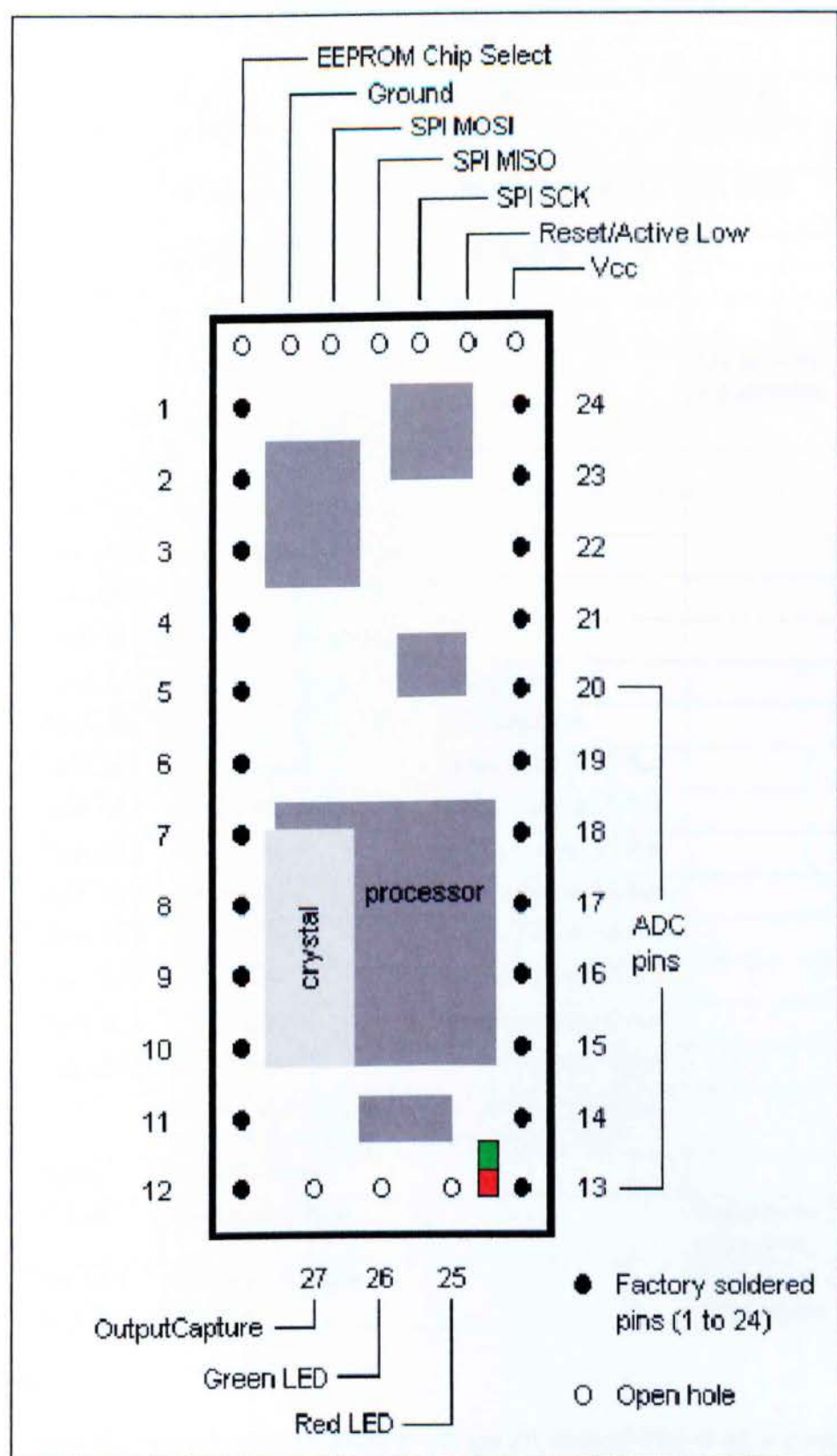
Τεχνικά Χαρακτηριστικά

BX-24 technical specifications

General

I/O Lines	16 total; 8 digital plus 8 lines that can be ADC or digital
EEPROM for program and data storage	On-board 32 KB EEPROM Largest executable user program size is 32 KBytes
RAM	400 bytes
Analog to digital converter	8 channels of 10 bit ADC, can also be used as regular digital (TTL level) I/O
ADC sample rate	6 k samples/s maximum
On-chip LEDs	Has a 2-color surface mount LED (red/green), fully user programmable, not counted as I/O line
Program execution speed	60 microseconds per 16 bit integer add/subtract
Serial I/O speed	2400 baud to 460.8 Kbaud on Com1 300 baud to 19 200 baud on any I/O pin (Com3)
Operating voltage range Min/Max	4.8 VDC to 12.0 VDC
Current requirements	20 mA plus I/O loads, if any
I/O output source current	10 mA @ 5 V (I/O pin driven high)
I/O output sink current	20 mA @ 5 V (I/O pin pulled low)
Combined maximum current load allowed across all I/Os	80 mA sink or source
I/O internal pull-up resistors	120 k Ω maximum
Floating point math	Yes
On-chip multitasking	Yes
On-chip clock/calendar	Yes
Built-in SPI interface	Yes
PC programming interface	Parallel or serial downloads
Package type	24 pin PDIP carrier board
Environmental specifications Absolute maximum ratings	Operating temperature: 0 °C to +70 °C Storage temperature: -85 °C to +150 °C

BX-24 pin numbering



Σχήμα 4

Αρίθμηση pins BX-24p

BX-24 Pin Definitions

Each pin on the BX-24 computer has a primary and alternate function as shown in the table below. The primary function describes how the pin can be configured. The alternate function describes how the pin is configured when BasicX built-in options are selected.

Pin #	Primary Function	Primary Description	Alternate Function	Alternate Description
1	Com1 Transmit	High speed Com port	Serial download transmit	See note 1
2	Com1 Receive	High speed Com port	Serial download receive	
3	ATN line	For serial downloading		
4	Ground	Common with pin 23		See below for DC characteristics
5	PortC, Bit 7	General Purpose I/O Port		
6	PortC, Bit 6	General Purpose I/O Port		
7	PortC, Bit 5	General Purpose I/O Port		
8	PortC, Bit 4	General Purpose I/O Port		
9	PortC, Bit 3	General Purpose I/O Port		
10	PortC, Bit 2	General Purpose I/O Port		
11	PortC, Bit 1	General Purpose I/O Port	Interrupt pin	
12	PortC, Bit 0	General Purpose I/O Port	Input capture pin	
13	PortA, Bit 7	ADC channel 7	General Purpose I/O Port	
14	PortA, Bit 6	ADC channel 6	General Purpose I/O Port	
15	PortA, Bit 5	ADC channel 5	General Purpose I/O Port	
16	PortA, Bit 4	ADC channel 4	General Purpose I/O Port	
17	PortA, Bit 3	ADC channel 3	General Purpose I/O Port	
18	PortA, Bit 2	ADC channel 2	General Purpose I/O Port	
19	PortA, Bit 1	ADC channel 1	General Purpose I/O Port	
20	PortA, Bit 0	ADC channel 0	General Purpose I/O Port	
21	VCC	5 V output from regulator (when powered at Vin)	4.8 V to 5.5 V power input (when Vin not used)	
22	Reset	Low Active Reset		
23	Ground	Common with Pin 4		See below for DC characteristics
24	Vin, 5.5 V to 12 V	Input to 5 V regulator (see note 2)		See below for DC characteristics

Notes:

[1] If regulated 5 V power is connected directly to VCC (pin 21), Vin (pin 24) can be left unconnected.

[2] The Com1 serial port transmits 0 V or 5 V signals, which technically does not conform to the EIA/RS-232 standard. In practice, this is rarely a problem with PC serial ports over short distances, but additional components would be required if you need to meet the EIA/RS-232 standard.

BX-24 DC characteristics

Parameter	Condition	Min	Typ	Max
Pin 24 power input to voltage regulator	Cannot be used as input if pin 21 is connected	5.5 V	6.0 V	12.0 V
Pin 21 regulated input/output	Cannot be used as input if pin 24 is connected	4.8 V	5.0 V	5.5 V
Ground is common to pins 4 and 23				
Output low voltage	$V_{CC} = 5.0\text{ V}$ Output current = 10 mA			0.5 V
Output high voltage	$V_{CC} = 5.0\text{ V}$ Output current = 10 mA	4.5 V		
Output source current	$V_{CC} = 5.0\text{ V}$ $V_{CC} = 2.7\text{ V}$			10 mA 5 mA
Output sink current	$V_{CC} = 5.0\text{ V}$ $V_{CC} = 2.7\text{ V}$			15 mA 10 mA
Maximum total	for all output pins			70 mA
Analog comparator input offset voltage	$V_{CC} = 5.0\text{ V}$			40 mV
Analog comparator input leakage A	$V_{CC} = 5.0\text{ V}$ $V_{in} = V_{CC}/2$	-50 nA	50 nA	
Analog comparator propagation delay	$V_{CC} = 5.0\text{ V}$		500 ns	
I/O pin programmable pull-up resistor		35 k Ω		120 k Ω

Κεφάλαιο 4

Πληροφορίες, τυπικές εφαρμογές και χαρακτηριστικά για το αισθητήριο LM- 35

Αισθητήριο LM-35

Η σειρά LM-35 είναι μια συλλογή αισθητηρίων θερμοκρασίας ενσωματωμένου κυκλώματος υψηλής ακρίβειας, η τάση των οποίων είναι γραμμικά αναλογική με την μονάδα μέτρησης θερμοκρασίας Celsius.

Το LM-35 έχει τον πλεονέκτημα έναντι άλλων γραμμικών αισθητηρίων θερμοκρασίας που μετράνε σε Kelvin στο ότι δεν απαιτεί από τον χρήστη την αφαίρεση μιας μεγάλης σταθεράς από την μετρήσιμη τιμή για να λάβει μια βολική Centigrade τιμή. Το LM-35 δεν απαιτεί κάποια εξωτερική ρύθμιση για να δώσει τις τυπικές ακρίβειες του $\pm 1/4^\circ\text{C}$ σε θερμοκρασία δωματίου και του $\pm 3/4^\circ\text{C}$ πάνω σε ένα εύρος από -55 έως $+150^\circ\text{C}$. Χαμηλό κόστος επιτυγχάνεται με την ρύθμιση του αισθητήρα στο αρχικό επίπεδο δίσκου πυριτίου. Χαρακτηριστικά όπως χαμηλή αντίσταση εξόδου, γραμμική έξοδος και ακριβής έμφυτος ρυθμιστής κάνουν τον έλεγχο κυκλωμάτων ιδιαίτερα εύκολη διαδικασία.

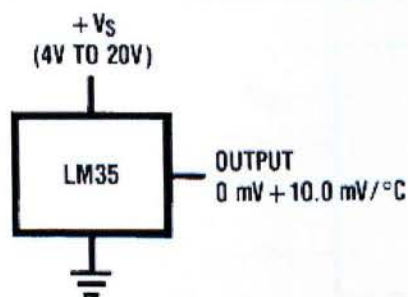
Το LM-35 μπορεί να χρησιμοποιηθεί με μονή πηγή ενέργειας ή με \pm πηγές ενέργειας. Καθώς το ρεύμα που απαιτεί είναι της τάξης των μόλις $60\mu\text{A}$ έχει πολύ χαμηλό επίπεδο αυτό-θέρμανσης, λιγότερο από 0.1°C σε κενό αέρος. Το εύρος λειτουργίας του LM-35 είναι μεταξύ -55 και $+150^\circ\text{C}$, ενώ του LM-35C είναι μεταξύ -40 και $+110^\circ\text{C}$ (-10°C με βελτιωμένη ακρίβεια). Η σειρά LM-35 είναι διαθέσιμη σε ερμητικά πακέτα TO-46 τρανζίστορ, ενώ τα LM-35C, LM-35CA και LM-35D είναι επίσης διαθέσιμα σε πλαστικά πακέτα TO-92 τρανζίστορ. Το LM-35D είναι επίσης διαθέσιμο σε μια 8 ακροδεκτών βάση και σε ένα TO-220 τρανζίστορ.

Χαρακτηριστικά

- Ρυθμισμένο απευθείας σε $^\circ\text{C}$
- Γραμμικό με σταθερό όρο κλίμακας $+ 10.0 \text{ mV}/^\circ\text{C}$
- Εγγυημένη ακρίβεια 0.5°C στους 25°C
- Εύρος λειτουργίας από -55 έως $+150^\circ\text{C}$
- Ιδανικό για τηλεκατευθυνόμενες εφαρμογές
- Χαμηλό κόστος λόγω ρύθμισης στο αρχικό επίπεδο δίσκου πυριτίου
- Λειτουργία από 4 έως 30 Volts
- Λιγότερο από $60\mu\text{A}$ κατανάλωση ρεύματος
- Χαμηλή αυτό θέρμανση, 0.08°C σε κενό αέρος
- Μη γραμμικότητα μόνο $\pm 1/4^\circ\text{C}$ τυπικό
- Χαμηλή αντίσταση εξόδου, 0.1 W για 1 mA φορτίο

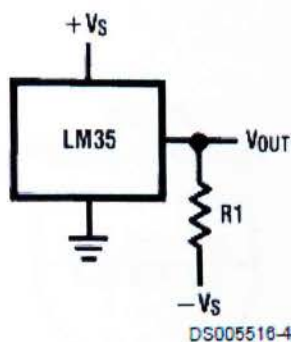
Τυπικές εφαρμογές

1.



Σχήμα 1. Βασικός αισθητήρας θερμοκρασίας (+2 έως +150°C)

2.



$$R1 = -VS/50 \mu A$$

$$V_{OUT} = +1,500 \text{ mV στα } +150^{\circ}\text{C}$$

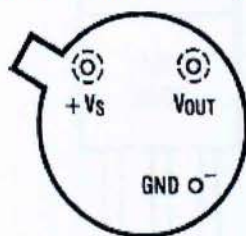
$$= +250 \text{ mV στα } +25^{\circ}\text{C}$$

$$= -550 \text{ mV στα } -55^{\circ}\text{C}$$

Σχήμα 2. Πλήρης κλίμακας αισθητήρας θερμοκρασίας

Διαγράμματα Σύνδεσης

TO-46 Metal Can Package*

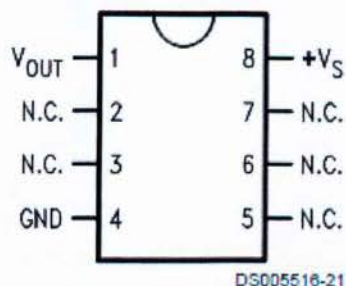


DS005516-1

*Case is connected to negative pin (GND)

Σχήμα σύνδεσης 1

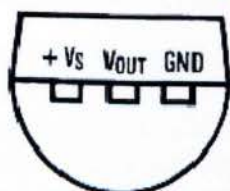
SO-8
Small Outline Molded Package



Σχήμα σύνδεσης 2

N.C. = No Connection

TO-92
Plastic Package

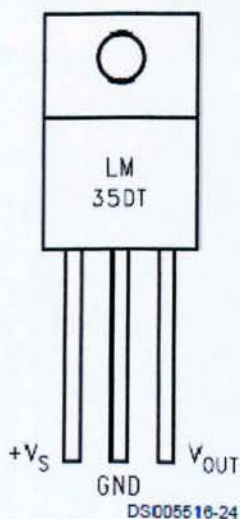


BOTTOM VIEW

DS005516-2

Σχήμα σύνδεσης 3

TO-220
Plastic Package*



*Tab is connected to the negative pin (GND).

Note: The LM35DT pinout is different than the discontinued LM35DP.

Σχήμα σύνδεσης 4

Απόλυτες τιμές

Παρεχόμενη τάση τροφοδοσίας +35V to -0.2V

Τάση εξόδου +6V to -1.0V

Ρεύμα εξόδου 10 mA

Θερμοκρασία αποθήκευσης:

TO-46, -60°C to +180°C

TO-92, -60°C to +150°C

SO-8, -65°C to +150°C

TO-220, -65°C to +150°C

Θερμοκρασία συγκόλλησης:

TO-46(για 10sec), 300°C

TO-92 και TO-220(για 10sec), 260°C

Θερμοκρασία καταστροφής (για 60sec) 215°C

Υπέρυθρες ακτίνες (15sec) 220°C

Ηλεκτρική επιδεκτικότητα 2500V

Προβλεπόμενο εύρος θερμοκρασίας λειτουργίας T_{MIN} to T_{MAX}

LM35, LM35A -55°C έως +150°C

LM35C, LM35CA -40°C έως +110°C

LM35D 0°C έως +100°C

Ηλεκτρικά χαρακτηριστικά οικογένειας LM-35

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	T _A =+25°C	±0.2	±0.5		±0.2	±0.5		°C
	T _A =-10°C	±0.3			±0.3		±1.0	°C
	T _A =T _{MAX}	±0.4	±1.0		±0.4	±1.0		°C
	T _A =T _{MIN}	±0.4	±1.0		±0.4		±1.5	°C
Nonlinearity (Note 8)	T _{MIN} ≤ T _A ≤ T _{MAX}	±0.18		±0.35	±0.15		±0.3	°C
Sensor Gain (Average Slope)	T _{MIN} ≤ T _A ≤ T _{MAX}	+10.0	+9.9, +10.1		+10.0		+9.9, +10.1	mV/°C
Load Regulation (Note 3) 0 ≤ I _L ≤ 1 mA	T _A =+25°C	±0.4	±1.0		±0.4	±1.0		mV/mA
	T _{MIN} ≤ T _A ≤ T _{MAX}	±0.5		±3.0	±0.5		±3.0	mV/mA
Line Regulation (Note 3)	T _A =+25°C	±0.01	±0.05		±0.01	±0.05		mV/V
	4V ≤ V _o ≤ 30V	±0.02		±0.1	±0.02		±0.1	mV/V
Quiescent Current (Note 9)	V _o =+5V, +25°C	56	67		56	67		μA
	V _o =+5V	105		131	91		114	μA
	V _o =+30V, +25°C	56.2	68		56.2	68		μA
	V _o =+30V	105.5		133	91.5		116	μA
Change of Quiescent Current (Note 3)	4V ≤ V _o ≤ 30V, +25°C	0.2	1.0		0.2	1.0		μA
	4V ≤ V _o ≤ 30V	0.5		2.0	0.5		2.0	μA
Temperature Coefficient of Quiescent Current		+0.39		+0.5	+0.39		+0.5	μA/°C
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, I _L =0	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	T _J =T _{MAX} , for 1000 hours	±0.08			±0.08			°C

Electrical Characteristics								
(Notes 1, 6)								
Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	$T_A = +25^\circ\text{C}$	± 0.4	± 1.0		± 0.4	± 1.0		$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	± 0.5			± 0.5		± 1.5	$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$	± 0.8	± 1.5		± 0.8		± 1.5	$^\circ\text{C}$
	$T_A = T_{\text{MIN}}$	± 0.8		± 1.5	± 0.8		± 2.0	$^\circ\text{C}$
Accuracy, LM35D (Note 7)	$T_A = +25^\circ\text{C}$				± 0.6	± 1.5		$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$				± 0.9		± 2.0	$^\circ\text{C}$
	$T_A = T_{\text{MIN}}$				± 0.9		± 2.0	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.3		± 0.5	± 0.2		± 0.5	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	$+10.0$	$+9.8, +10.2$		$+10.0$		$+9.8, +10.2$	mV/°C
Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$	$T_A = +25^\circ\text{C}$	± 0.4	± 2.0		± 0.4	± 2.0		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.5		± 5.0	± 0.5		± 5.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	± 0.01	± 0.1		± 0.01	± 0.1		mV/V
	$4V \leq V_O \leq 30V$	± 0.02		± 0.2	± 0.02		± 0.2	mV/V
Quiescent Current (Note 9)	$V_O = +5V, +25^\circ\text{C}$	56	80		56	80		μA
	$V_O = +5V$	105		158	91		138	μA
	$V_O = +30V, +25^\circ\text{C}$	56.2	82		56.2	82		μA
	$V_O = +30V$	105.5		161	91.5		141	μA
Change of Quiescent Current (Note 3)	$4V \leq V_O \leq 30V, +25^\circ\text{C}$	0.2	2.0		0.2	2.0		μA
	$4V \leq V_O \leq 30V$	0.5		3.0	0.5		3.0	μA
Temperature Coefficient of Quiescent Current		$+0.39$		$+0.7$	$+0.39$		$+0.7$	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	$+1.5$		$+2.0$	$+1.5$		$+2.0$	$^\circ\text{C}$
Long Term Stability	$T_J = T_{\text{MAX}}$, for 1000 hours	± 0.08			± 0.08			$^\circ\text{C}$

Note 1: Εκτός και δεν υπάρχει άλλη σημείωση, οι προδιαγραφές αυτές έχουν ως εξής: $-55^\circ\text{C} \leq T_J \leq +150^\circ\text{C}$ για το LM35 και LM35A; $-40^\circ \leq T_J \leq +110^\circ\text{C}$ για το LM35C και LM35CA; και

$0^\circ \leq T_J \leq +100^\circ\text{C}$ για το LM35D. $V_S = +5V_{\text{dc}}$ και $I_{\text{LOAD}} = 50 \mu\text{A}$, στο κύκλωμα του σχήματος 2. Αυτές τα χαρακτηριστικά έχουν επίσης εφαρμογή από $+2^\circ\text{C}$ έως T_{MAX} στο κύκλωμα του σχήματος 1.

Χαρακτηριστικά με έντονη γραφή έχουν εφαρμογή σε πλήρη κλίμακα θερμοκρασίας.

Note 2: Η θερμική αντίσταση του TO-46 είναι $400^\circ\text{C}/\text{W}$, junction to ambient, και $24^\circ\text{C}/\text{W}$ junction to case. Η θερμική αντίσταση του TO-92 είναι $180^\circ\text{C}/\text{W}$ junction to ambient. Η θερμική αντίσταση του είναι $220^\circ\text{C}/\text{W}$ junction to ambient. Η θερμική αντίσταση του TO-220 είναι $90^\circ\text{C}/\text{W}$ junction to ambient.

Note 3: Ο συντονισμός είναι μετρημένος σε θερμοκρασία ένωσης, χρησιμοποιώντας παλμικό τεστ σε χαμηλό κύκλο λειτουργίας. Αλλαγές στην έξοδο εξαιτίας θερμικών απωλειών μπορούν υπολογιστούν πολλαπλασιάζοντας την απώλεια με την θερμική αντίσταση.

Note 4: Τα δοκιμασμένα όρια είναι εγγυημένα και 100% δοκιμασμένα στην παραγωγή.

Note 5: Τα όρια σχεδίασης είναι εγγυημένα (αλλά όχι 100% δοκιμασμένα στην παραγωγή) πάνω στις προβλεπόμενες τιμές θερμοκρασίας και τάσης. Αυτά

τα όρια δεν έχουν χρησιμοποιηθεί για να υπολογιστούν επίπεδα ποιότητας εξόδου.

Note 6: Χαρακτηριστικά με έντονη γραφή έχουν εφαρμογή σε πλήρη κλίμακα θερμοκρασίας.

Note 7: Ακρίβεια ορίζεται σαν το λάθος μεταξύ της εξόδου τάσης και $10\text{mV}/^{\circ}\text{C}$ φορές την θερμοκρασία της θήκης της συσκευής, σε συγκεκριμένες συνθήκες τάσης, ρεύματος και θερμοκρασίας (εκφρασμένη σε $^{\circ}\text{C}$).

Note 8: Μη-γραμμικότητα ορίζεται σαν η απόκλιση της εξόδου-τάσης/θερμοκρασίας συνάρτησης από την βέλτιστη ευθεία γραμμή, πάνω από την βαθμονομημένο εύρος θερμοκρασίας της συσκευής.

Note 9: Αδρανή ρεύμα έχει σαν ορισμό το σχήμα 1.

Note 10: Απόλυτες μέγιστες τιμές επιδεικνύουν τα όρια μεταξύ ποιων ζημιών μπορούν να προκληθούν στην συσκευή. DC και AC ηλεκτρικές προδιαγραφές δεν μπορούν να εφαρμοστούν όταν η συσκευή λειτουργεί πέρα των συνθηκών λειτουργίας της. Βλέπε Note 1.

Note 11: Ανθρώπινο μοντέλο, 100 pF αποφορτίστηκε διαμέσου αντίστασης 1.5 kW.

Note 12: Βλέπε AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National

Semiconductor Linear Data Book for other methods of soldering surface mount devices.

Κεφάλαιο 5

Εκτέλεση πτυχιακής και επεξήγηση του κώδικα λειτουργίας της

Εκτέλεση πτυχιακής

Ο BX-24p για να λειτουργήσει έχει κάποια pins τα οποία πρέπει να είναι συνδεδεμένα πάντα, ανεξαρτήτως πειράματος. Αυτά τα pins είναι τα 24 και 23 και προσφέρουν αντίστοιχα τάση και γείωση.

Για να είναι επιτυχής η εκτέλεση όλου της πτυχιακής, θα πρέπει να έχουμε φορτώσει στον BX-24p το κατάλληλο σετ εντολών προγραμματισμού, τον λεγόμενο και κώδικα του προγράμματος. Με αυτό τον κώδικα, μέσω του BX-24p ελέγχουμε όλα τα περιφερειακά συστήματα του κυκλώματος όπως led, ηλεκτρονόμους, buzzer και ανεμιστήρα. Για να είναι επιτυχής η εκτέλεση, ο κώδικας θα πρέπει να ορίζει με σαφήνεια την λειτουργία κάθε εξαρτήματος.

Σημείωση: όποτε θα αναφερόμαστε στον όρο θερμοκρασία για το πείραμα, οι τιμές που θα σημειώνουμε δεν θα είναι τιμές πραγματικής θερμοκρασίας αλλά τιμές που επιστρέφει το LM-35 και θα τις αντιμετωπίζουμε σαν ακέραιους αριθμούς.

Ο κώδικας του προγράμματος είναι ο εξής:

1. Option Explicit

2. Public Sub Main()

3. Dim Temperature As Integer

4. Const MyPin As Byte = 15

5. Const LEDPin1 As Byte = 13

6. Const LEDPin2 As Byte = 12

7. Const LEDPin3 As Byte = 11

8. Const LEDPin4 As Byte = 10

9. Const LEDPin5 As Byte = 9

10. Const LEDPin6 As Byte = 8

11. Const LEDPin7 As Byte = 7

12. Const LEDPin8 As Byte = 6

13. Const LEDPin9 As Byte = 5

14. Const Fan As Byte = 17

15. Const Buzz As Byte = 18

16. Const LEDOn As Byte = 1

17. Const LEDOff As Byte = 0

18. Const Rithmisi as Integer = -10

19. Do

20. Temperature = GetAdc(MyPin)

Call PutPin(LEDPin1,LEDOff)

Call PutPin(LEDPin2,LEDOff)

Call PutPin(LEDPin3,LEDOff)

Call PutPin(LEDPin4,LEDOff)

```
Call PutPin(LEDpin5,LEDOff)
Call PutPin(LEDpin6,LEDOff)
Call PutPin(LEDpin7,LEDOff)
Call PutPin(LEDpin8,LEDOff)
Call PutPin(LEDpin9,LEDOff)
```

```
21.If (Temperature > (96+Rithmisi)) then
    Call PutPin(Fan,LEDOff)
    Call PutPin(Buzz,LEDOff)
End If
```

```
22.If (Temperature < (120+Rithmisi)) then
    Call PutPin(LEDpin1,LEDOff)
End If
```

```
23.If (Temperature < (112+Rithmisi)) then
    Call PutPin(LEDpin1,LEDOff)
    Call PutPin(LEDpin2,LEDOff)
End If
```

```
24.If (Temperature < (104+Rithmisi)) then
    Call PutPin(LEDpin1,LEDOff)
    Call PutPin(LEDpin2,LEDOff)
    Call PutPin(LEDpin3,LEDOff)
End If
```

```
25.If (Temperature < (96+Rithmisi)) then
    Call PutPin(LEDpin1,LEDOff)
    Call PutPin(LEDpin2,LEDOff)
    Call PutPin(LEDpin3,LEDOff)
    Call PutPin(LEDpin4,LEDOff)
```

.End If

```
26.If (Temperature < (88+Rithmisi)) then
    Call PutPin(LEDpin1,LEDOff)
    Call PutPin(LEDpin2,LEDOff)
    Call PutPin(LEDpin3,LEDOff)
    Call PutPin(LEDpin4,LEDOff)
    Call PutPin(LEDpin5,LEDOff)
End If
```



```
27.If (Temperature < (80+Rithmisi)) then
    Call PutPin(LEDpin1,LEDon)
    Call PutPin(LEDpin2,LEDon)
    Call PutPin(LEDpin3,LEDon)
    Call PutPin(LEDpin4,LEDon)
    Call PutPin(LEDpin5,LEDon)
    Call PutPin(LEDpin6,LEDon)
```

End If

```
28.If (Temperature < (72+Rithmisi)) then
    Call PutPin(LEDpin1,LEDon)
    Call PutPin(LEDpin2,LEDon)
    Call PutPin(LEDpin3,LEDon)
    Call PutPin(LEDpin4,LEDon)
    Call PutPin(LEDpin5,LEDon)
    Call PutPin(LEDpin6,LEDon)
    Call PutPin(LEDpin7,LEDon)
```

End If

```
29.If (Temperature < (64+Rithmisi)) then
    Call PutPin(LEDpin1,LEDon)
    Call PutPin(LEDpin2,LEDon)
    Call PutPin(LEDpin3,LEDon)
    Call PutPin(LEDpin4,LEDon)
    Call PutPin(LEDpin5,LEDon)
    Call PutPin(LEDpin6,LEDon)
    Call PutPin(LEDpin7,LEDon)
    Call PutPin(LEDpin8,LEDon)
```

End If

```
30.If (Temperature < (56+Rithmisi)) then
    Call PutPin(LEDpin1,LEDon)
    Call PutPin(LEDpin2,LEDon)
    Call PutPin(LEDpin3,LEDon)
    Call PutPin(LEDpin4,LEDon)
    Call PutPin(LEDpin5,LEDon)
    Call PutPin(LEDpin6,LEDon)
    Call PutPin(LEDpin7,LEDon)
    Call PutPin(LEDpin8,LEDon)
    Call PutPin(LEDpin9,LEDon)
    Call PutPin(Fan,LEDon)
    Call PutPin(Buzz,LEDon)
```

End If

31.Call Delay(0.5)

32.Debug.Print CStr (Temperature)

33.Loop

34.End Sub

Επεξήγηση του κώδικα

Ο κώδικας είναι χωρισμένος σε γραμμές ώστε να μπορούμε να τον επεξηγήσουμε τμηματικά.

Γραμμές 1 και 2

Είναι η αρχικοποίηση κάθε προγράμματος. Δημιουργούμε μια Public Sub Main η οποία είναι η κυρίως ρουτίνα του κώδικα.

Γραμμές 3 έως 15

Θέτουμε την θερμοκρασία σαν ακέραιο αριθμό που μπορεί να πάρει τιμές από -32768 έως 32767. Στην συνέχεια ορίζουμε ποιες θα είναι η έξοδοι του BX-24r που θα είναι ενεργοί και τις θέτουμε σαν μεταβλητές. Από το pin 5 έως το 13 θα είναι τα led, pin 15 θα είναι η επικοινωνία με το LM-35, το pin 17 θα είναι ο ανεμιστήρας και το pin 18 θα είναι το buzzer.

Γραμμές 16 έως 18

Θέτουμε δύο μεταβλητές LEDoff και LEDon, προσδιορίζοντας τις καταστάσεις 0 και 1 αντίστοιχα. Η μεταβλητή Rithmisi μας επιτρέπει να προσθέτουμε μια μικρή ποσότητα καλιμπραρίσματος της θερμοκρασίας ανάλογα με το κλίμα που επικρατεί.

Γραμμή 19

Με την εντολή Do ξεκινάει το πρόγραμμα την λειτουργία του. Είναι το κομμάτι από την εντολή Do...Loop η οποία δημιουργεί ένα βρόγχο ο οποίος εκτελείται συνέχεια.

Γραμμή 20

Εδώ πέρα γράφουμε τις αρχικοποιήσεις του κώδικα. Διαβάζει την θερμοκρασία από τον αισθητήρα και θέτει όλα τα led σε κατάσταση off.

Γραμμή 21

Θέτουμε τον ανεμιστήρα και το megáfwno σε κατάσταση off.

Γραμμή 22

Αν η θερμοκρασία είναι κάτω από την τιμή 120, θέτει σε λογικό 1 το πρώτο πράσινο led

Γραμμή 23

Αν η θερμοκρασία είναι κάτω από την τιμή 112, θέτει σε λογικό 1 τα πρώτα δύο πράσινα led

Γραμμή 24

Αν η θερμοκρασία είναι κάτω από την τιμή 104, θέτει σε λογικό 1 τα πρώτα τρία πράσινα led

Γραμμή 25

Αν η θερμοκρασία είναι κάτω από την τιμή 96, θέτει σε λογικό 1 τα πρώτα τρία πράσινα led και το πρώτο πορτοκαλί led.

Γραμμή 26

Αν η θερμοκρασία είναι κάτω από την τιμή 88, θέτει σε λογικό 1 τα πρώτα τρία πράσινα led και τα πρώτα δύο πορτοκαλί led.

Γραμμή 27

Αν η θερμοκρασία είναι κάτω από την τιμή 80, θέτει σε λογικό 1 τα πρώτα τρία πράσινα led και τα πρώτα τρία πορτοκαλί led.

Γραμμή 28

Αν η θερμοκρασία είναι κάτω από την τιμή 72, θέτει σε λογικό 1 τα πρώτα τρία πράσινα led, τα πρώτα τρία πορτοκαλί led και το πρώτο κόκκινο led.

Γραμμή 29

Αν η θερμοκρασία είναι κάτω από την τιμή 64, θέτει σε λογικό 1 τα πρώτα τρία πράσινα led, τα πρώτα τρία πορτοκαλί led και τα πρώτα δύο κόκκινα led.

Γραμμή 30

Αν η θερμοκρασία είναι κάτω από την τιμή 56, θέτει σε λογικό 1 τα πρώτα τρία πράσινα led, τα πρώτα τρία πορτοκαλί led και τα πρώτα τρία κόκκινα led. Επίσης θέτει σε λογικό 1 τον ανεμιστήρα και το μεγάφωνο.

Γραμμή 31

Προκαλεί καθυστέρηση μισού δευτερολέπτου. Αυτό μας επιτρέπει να προλαβαίνουμε να βλέπουμε τις τιμές στην οθόνη του υπολογιστή.

Γραμμή 32

Με αυτήν την εντολή τυπώνουμε τις τιμές που μας επιστρέφει το αισθητήριο θερμοκρασίας στην οθόνη του υπολογιστή.

Γραμμή 33

Με την εντολή Loop δημιουργούμε έναν κλειστό βρόγχο ο οποίος εκτελείται συνέχεια.

Γραμμή 34

Με την εντολή End Sub κλείνουμε τον κώδικα του προγράμματος μας.

Μεταφορά του κώδικα από τον υπολογιστή στον BX-24p

Ο κώδικας του BX-24p δημιουργείται με την κατάλληλη εφαρμογή (BasicX) σε ηλεκτρονικό υπολογιστή. Η μεταφορά του από τον υπολογιστή στον BX-24p γίνεται μέσω της σειριακής θύρας του υπολογιστή στα κατάλληλα pins του BX-24p (σχήματα 3 και 4). Το καλώδιο υπεύθυνο για την μεταφορά των δεδομένων, αν και υπάρχει στο εμπόριο, μπορούμε να το φτιάξουμε εμείς. Η συνδεσμολογία μεταξύ της σειριακής θύρας και του BX-24p είναι η εξής.

Από τον BX-24p θα χρειαστούμε τα pins 1 (Tx), 2 (Rx), 3 (ATN), 4 (GND).

DB-9

Connector BX-24 Chip

Pin 2 = TX >-----< Pin 1 of the BX-24

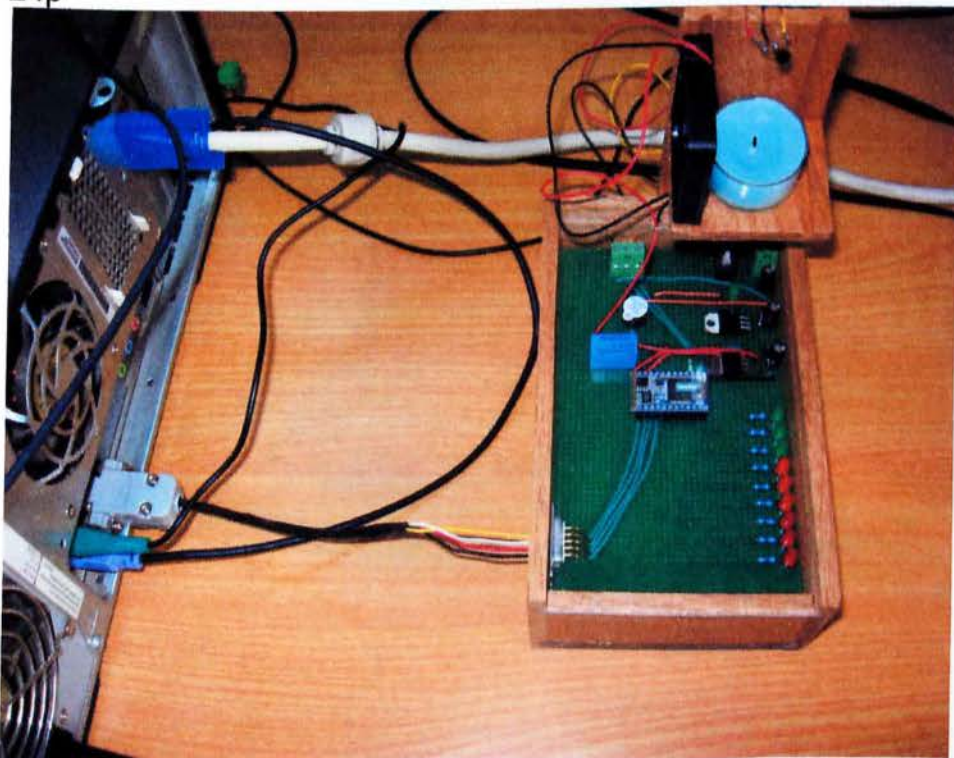
Pin 3 = RX >-----< Pin 2 of the BX-24

Pin 4 = ATN >-----< Pin 3 of the BX-24

Pin 5 = Ground >-----< Pins 4 or 23 of the BX-24

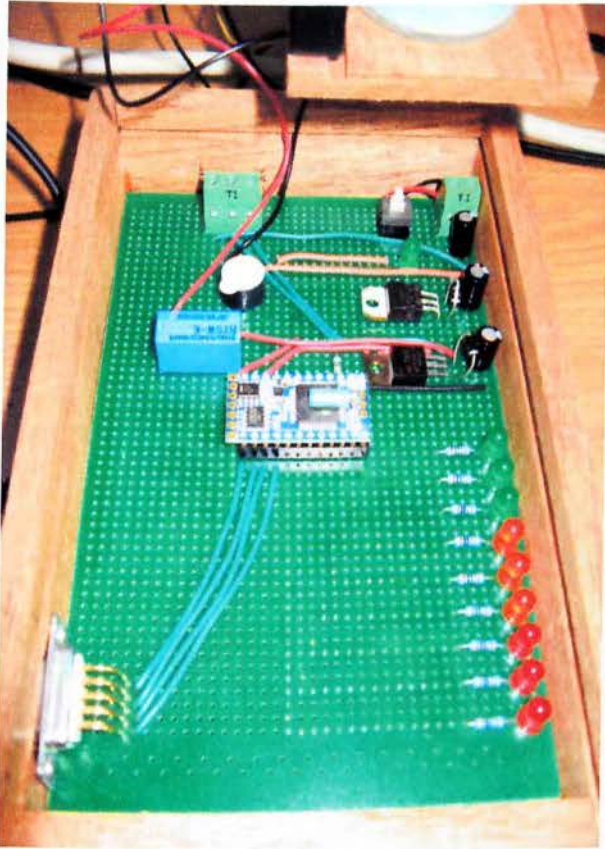
Ακολούθως, πρέπει να συνδέσουμε σωστά τον LM-35 ώστε να μπορούμε να παίρνουμε τις κατάλληλες μετρήσεις. Η σύνδεση του LM-35 με τον BX-24p γίνεται μέσω του pin 15.

Ας δούμε με ποιον τρόπο γίνεται η σύνδεση μέσω του υπολογιστή και του BX-24p

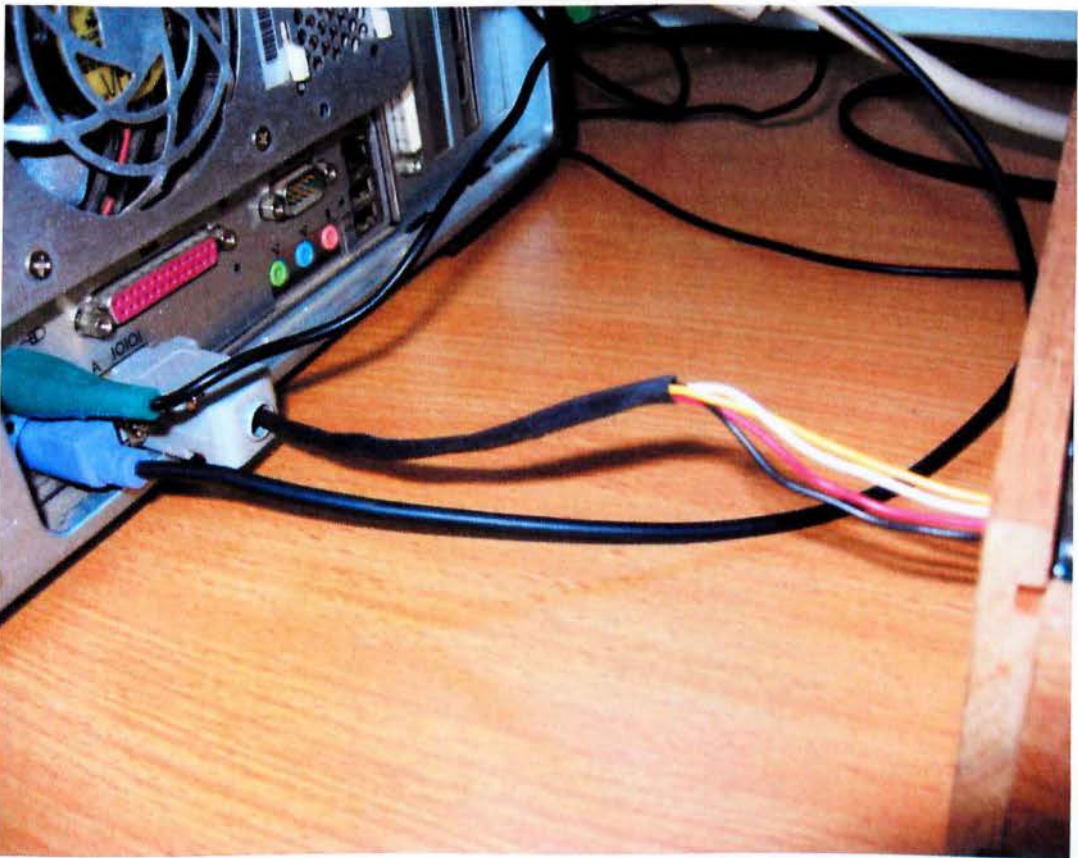


Σχήμα 1

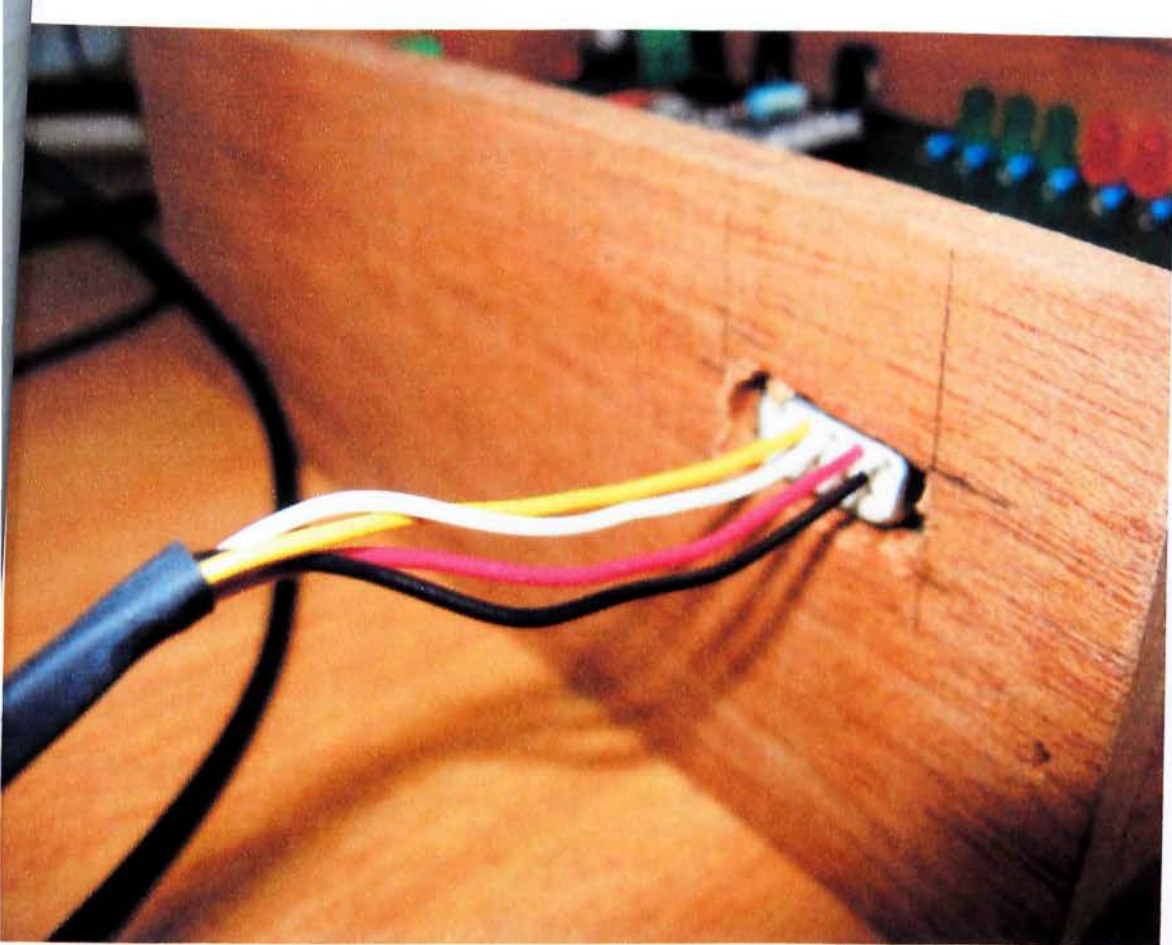
Ολοκληρωμένη σύνδεση πτυχιακής στον Η/Υ



Σχήμα 2
Θήκη πτυχιακής εργασίας μαζί
το ολοκληρωμένο κύκλωμα



Σχήμα 3
Σύνδεση σειριακού καλωδίου στον Η/Υ



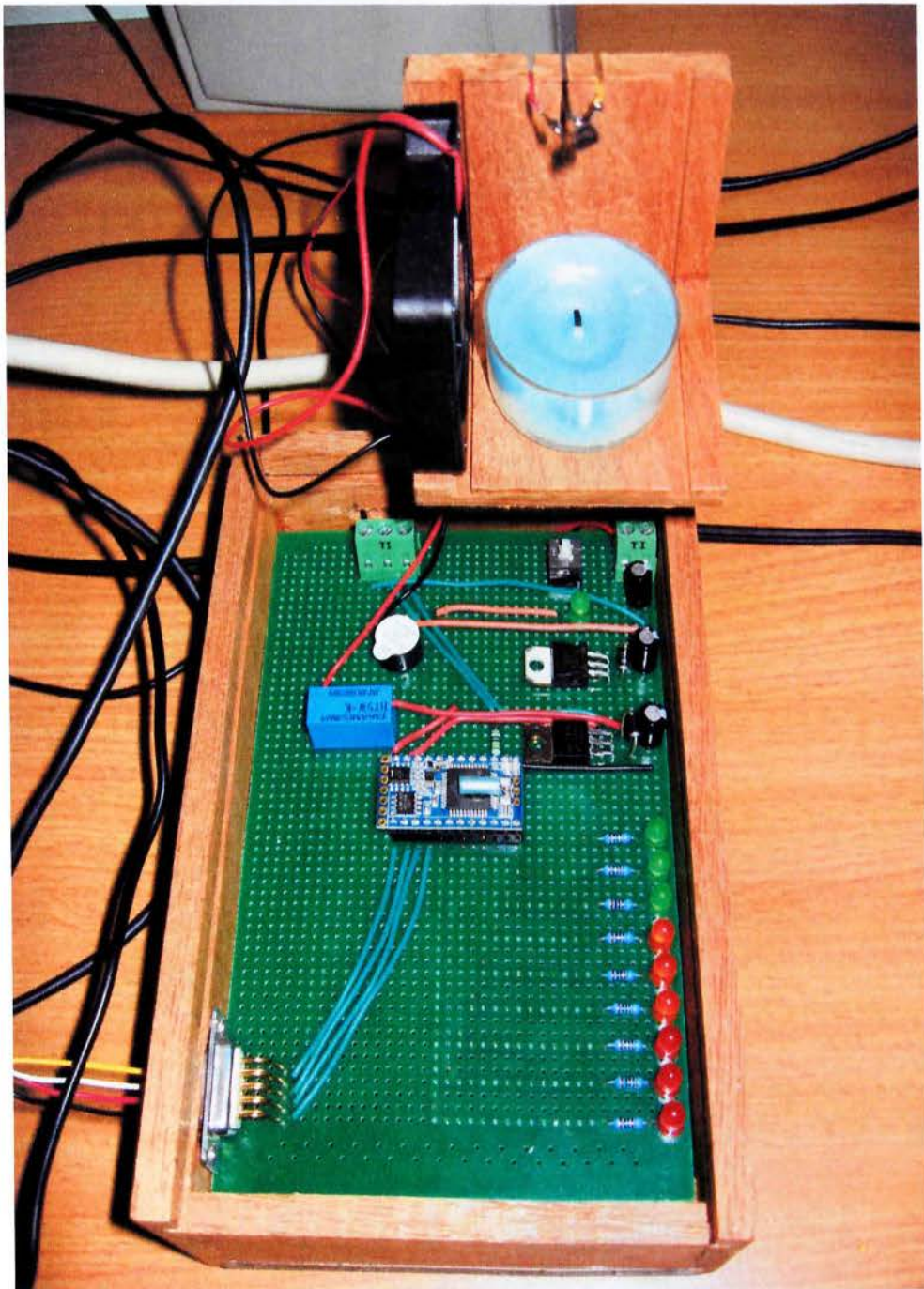
Σχήμα 4

**Σύνδεση σειριακού καλωδίου στην σειριακή
θύρα της πτυχιακής**

Πρέπει να προσέξουμε οι συνδέσεις να γίνουν όπως ακριβώς είναι οδηγίες και να αντιστοιχήσουμε τα σωστά pin της σειριακής με τα σωστά pin του BX-24p.

Λειτουργία της πτυχιακής

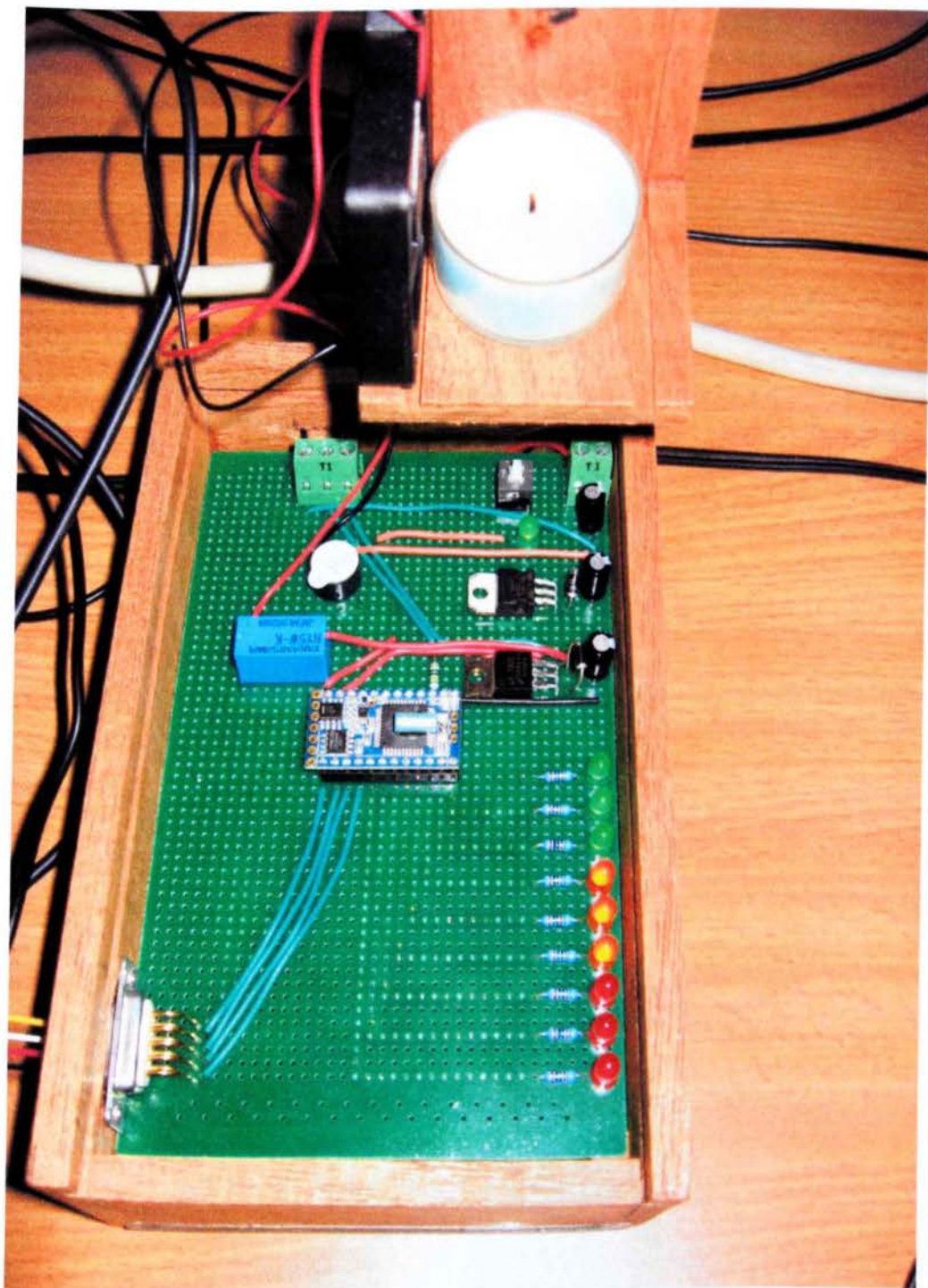
Ενεργοποιούμε τον διακόπτη λειτουργίας (σχήμα 5). Αυτομάτως τα led ενεργοποιούνται ανάλογα με την θερμοκρασία του περιβάλλοντος. Στην συγκεκριμένη περίπτωση έχουμε αναμμένα τα πρώτα τρία πράσινα led. Ο ανεμιστήρας και το megάφωνο φυσικά είναι σε κατάσταση off.



Σχήμα 5

Ενεργοποίηση κυκλώματος

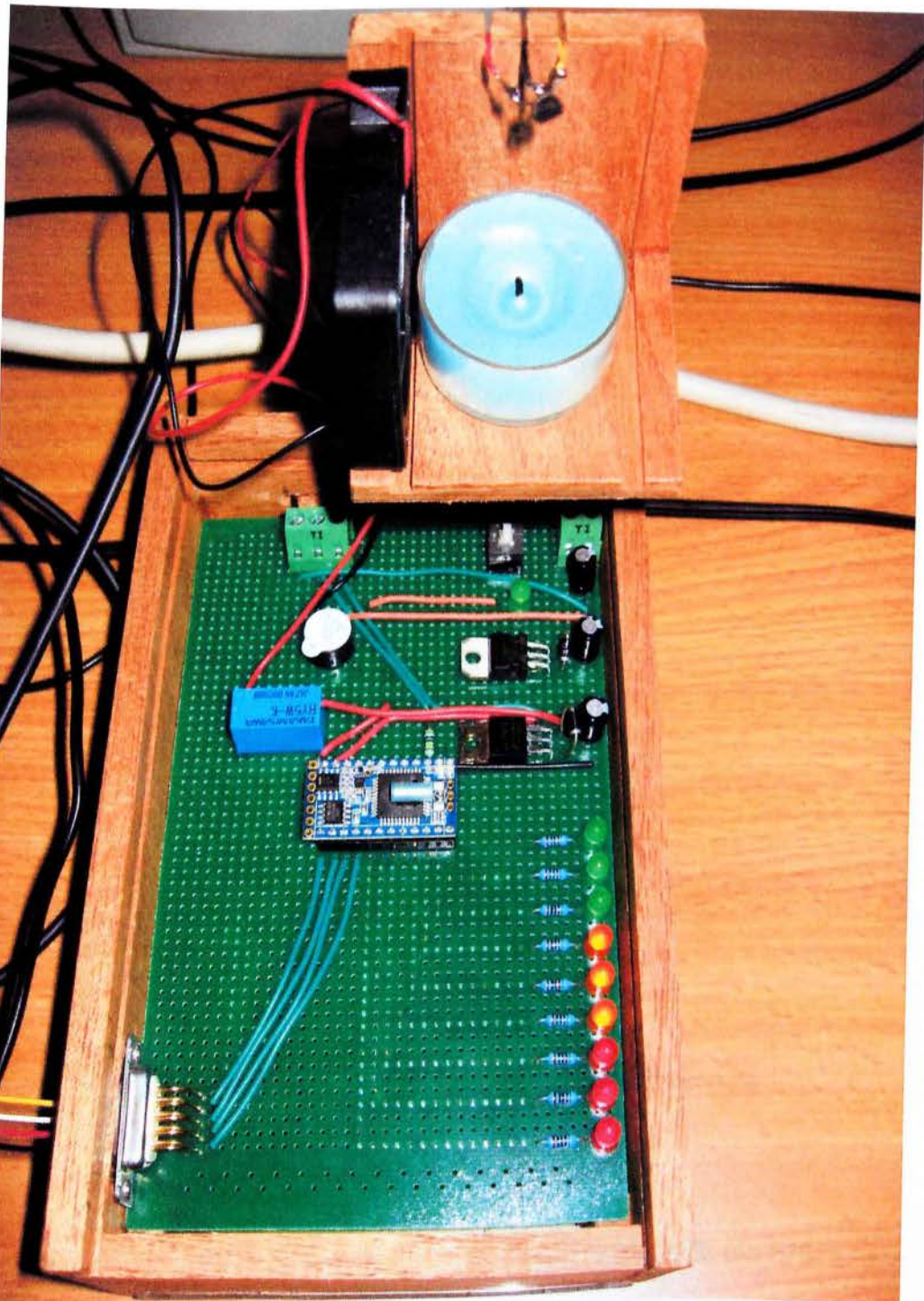
Στην συνέχεια ανάβουμε το κερί. Καθώς αρχίζει η θερμότητα να επιδράει στο αισθητήριο, αρχίζουν τα πορτοκαλί led να ενεργοποιούνται (σχήμα 5).



Σχήμα 6

Ενεργοποίηση πορτοκαλί led

Η αύξηση της θερμότητας στο αισθητήριο σε πολύ υψηλά επίπεδα, οδήγησε στην ενεργοποίηση των κόκκινων led και την είσοδο στην επικίνδυνη ζώνη ενεργοποίησης του ανεμιστήρα και του μεγάρωνου (σχήμα 6). Με το που άναψαν όλα τα led, ενεργοποιήθηκε αυτόματα ο ανεμιστήρας ώστε να εξουδετερώσει την αιτία της αύξησης της θερμοκρασίας που στην προκειμένη περίπτωση είναι το κερί. Το μεγάρφωνο ενεργοποιήθηκε και αυτό ώστε να μας ενημερώσει ηχητικά ότι η θερμοκρασία ξεπέρασε το όριο την μέγιστης θεθείσας αρχικής τιμής.



Σχήμα 6

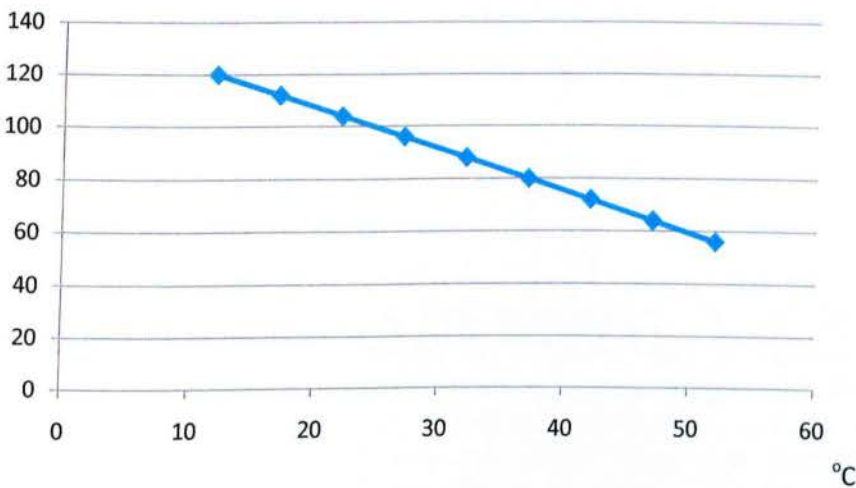
Η διαδικασία αυτή θα επαναλαμβάνεται συνέχεια όποτε ανάβουμε το κερι.

Για να κατανοήσουμε καλύτερα την συμπεριφορά του κυκλώματος κατά την διάρκεια της λειτουργίας του, μπορούμε να κοιτάξουμε την παρακάτω αντιστοιχία μεταξύ $^{\circ}\text{C}/\text{CPU units}$ και το γράφημα που το συνοδεύει.

Degrees Celcius	CPU Units
12	120
17	112
22	104
27	96
32	88
37	80
42	72
47	64
52	56

Γραφική απεικόνιση $^{\circ}\text{C}/\text{CPU units}$

CPU units



Σχήμα 7

Γραφική παράσταση CPU units σε συνάρτηση με την θερμοκρασία

Με έναν απλό μαθηματικό υπολογισμό βλέπουμε ότι για κάθε μία CPU unit, η θερμοκρασία αυξάνεται κατά $0,625^{\circ}\text{C}$.

Κεφάλαιο 6

Πληροφορίες για τις μνήμες EEPROM και σύγκρισή τους με τις μνήμες EPROM/Flash

EEPROM

Η λέξη EEPROM είναι τα αρχικά της φράσης **E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory, δηλαδή "ηλεκτρικά διαγραφόμενη προγραμματιζόμενη μνήμη ανάγνωσης. Είναι μια μνήμη τύπου non-volatile, δηλαδή μπορεί να έχει αποθηκευμένα τα αρχεία της ακόμα και χωρίς να έχει ρεύμα. Η χρήση της είναι πολύ συνηθισμένη σε ηλεκτρονικούς υπολογιστές και ηλεκτρονικές συσκευές για να αποθηκεύονται μικρής έκτασης πληροφορίες και δεδομένα τα οποία πρέπει να διατηρούνται ακόμα και όταν η ενέργεια χαθεί π.χ. πίνακες ρυθμίσεων και δεδομένα συσκευών.

Όταν χρειάζεται να αποθηκευτούν μεγάλα δεδομένα πληροφοριών (όπως σε USB sticks), τότε χρησιμοποιείται ένας συγκεκριμένος τύπος EEPROM ο οποίος ονομάζεται flash memory και είναι πιο οικονομικός από τον παραδοσιακό τύπο EEPROM.

Τα EEPROM ορίζονται σαν πίνακες floating-gate τρανζίστορ.

Τα EEPROM είναι μνήμες που χρησιμοποιούνται από τον χρήστη και μπορούν να σβηστούν και να ξαναγραφτούν συνέχεια με μια εφαρμογή τάσης λίγο υψηλότερη από την συνήθη τάση συσκευής. Σε αντίθεση με τα EPROM chips, τα EEPROM δεν χρειάζεται να αφαιρεθούν από την συσκευή για να ρυθμιστούν. Όμως, ένα EEPROM chip μπορεί να σβηστεί και να ξαναγραφτεί σε όλο του το φάσμα και όχι τμηματικά. Επίσης, έχει ένα πεπερασμένο όριο που μπορεί να ξαναγραφτεί, το οποίο μπορεί να είναι μερικές δεκάδες ή εκατοντάδες χιλιάδες φορές. Σε ένα EEPROM το οποίο αναπρογραμματίζεται ενώ η συσκευή βρίσκεται σε λειτουργία, μελέτες δείχνουν ότι η ζωή του chip μειώνεται σημαντικά.

Ιστορία EEPROM

Το 1978 ο Γεώργιος Περγλέος στην Intel κατασκεύασε τον Intel 2816, ο οποίος βασιζόταν σε μια πρώιμη έκδοση του EPROM αλλά χρησιμοποιούσε ένα λεπτό στρώμα οξειδίου ώστε να chip να μπορούσε να σβήσει τα δεδομένα του χωρίς να χρειάζεται μια πηγή UV. Αργότερα, ο Περγλέος μαζί με άλλα μέλη της Intel δημιούργησαν μια δικιά τους εταιρία ονόματι Seeq Technology κατασκευάζοντας chips στα οποία ενσωμάτωσαν αντλίες φόρτισης υψηλού ρεύματος απαραίτητες για τον προγραμματισμό τους. Έτσι δημιουργήθηκαν τα EEPROM.

Λειτουργίες των EEPROM

Υπάρχουν διάφοροι τύποι ηλεκτρικής αλληλεπίδρασης με EEPROM. Οι κυριότερες κατηγορίες είναι

- Σειριακός δίαυλος
- Παράλληλος δίαυλος

Το πώς η συσκευή λειτουργεί εξαρτάται από την ηλεκτρική αλληλεπίδραση.

Συσκευές σειριακού δίαυλου

Οι πιο συνηθισμένοι τύποι σειριακού δίαυλου είναι οι SPI, I²C, Microwire, UNI/O και 1-Wire. Αυτοί οι τύποι απαιτούν μεταξύ 1 και 4 σημάτων ελέγχου για επεξεργασία, με αποτέλεσμα η συσκευή να είναι 8 pin ή λιγότερα.

Η σειριακή EEPROM τυπικά λειτουργεί σε τρεις φάσεις: OP-Code Phase, Address Phase και Data Phase. Η OP-Code Phase είναι συνήθως η 8-bit είσοδος στην σειριακή είσοδο του EEPROM ακολουθούμενη από 8 έως 24 bits διευθύνσεων ανάλογα με το βάθος της συσκευής και τα δεδομένα που πρέπει να γραφτούν ή να διαβαστούν.

Κάθε EEPROM συσκευή τυπικά έχει το δικό της σετ εντολών OP-Code Phase για διάφορες συναρτήσεις.

Μερικές από τις πιο συνηθισμένες εργασίες σε SPI EEPROM συσκευές είναι:

- Write Enable (Wren)
- Write Disable (WRDI)
- Read Status Register (RDSR)
- Write Status Register (WRSR)
- Read Data (READ)
- Write Data (WRITE)

Άλλες εργασίες από EEPROM συσκευές είναι:

- Program
- Sector Erase
- Chip Erase Commands

Συσκευές παράλληλου δίαυλου

Οι EEPROM συσκευές παράλληλου δίαυλου τυπικά έχουν έναν 8-bit δίαυλο δεδομένων και ένα δίαυλο διευθύνσεων αρκετά μεγάλο ώστε να μπορεί να αλύψει ολόκληρη την μνήμη. Οι περισσότερες συσκευές έχουν chip select και write protect pins. Μερικοί μικροεπεξεργαστές έχουν επίσης ανσωματωμένα παράλληλα EEPROM.

Η λειτουργία ενός παράλληλου EEPROM είναι απλή και γρήγορη σε σύγκριση με ένα σειριακό EEPROM αλλά αυτές οι συσκευές είναι μεγαλύτερες σε μέγεθος εξαιτίας του μεγαλύτερου αριθμού pin (28 pin ή και περισσότερα) και έχουν χάσει σε προτίμηση και δημοτικότητα έναντι των σειριακών EEPROM ή των flash memory.

Άλλες συσκευές

Η EEPROM χρησιμοποιείται για να ενεργοποιήσει χαρακτηριστικά σε συσκευές οι οποίες δεν είναι αναγκαστικά συσκευές μνήμης. Τέτοιες συσκευές όπως ρολόγια, ψηφιακά ποτενσιόμετρα, ψηφιακοί σένσορες θερμοκρασίας μπορούν να έχουν μικρές ποσότητες EEPROM για να αποθηκεύουν ρυθμίσεις και δεδομένα σε περίπτωση που υπάρχει απώλεια ενέργειας.

Καταστάσεις αποτυχίας

Υπάρχουν δύο περιορισμοί όσον αφορά τις αποθηκευμένες πληροφορίες: διάρκεια και διατήρηση δεδομένων.

Κατά την διάρκεια των εγγραφών, η πύλη οξειδίου στα τρανζίστορ σταδιακά συλλαμβάνει παγιδευμένα ηλεκτρόνια. Το ενεργειακό πεδίο των παγιδευμένων ηλεκτρονίων προστίθεται σε αυτό των ηλεκτρονίων της πύλης, κατεβάζοντας το παράθυρο μεταξύ της τάσης κατωφλίου μηδέν – ένα. Μετά από επαρκή αριθμό εγγραφών, η διαφορά αυτή γίνεται πολύ μικρή για να γίνει αντιληπτή, ο πυρήνας κολλάει σε συνθήκη εγγραφής και παρουσιάζεται σφάλμα εγγραφής. Οι κατασκευαστές συνήθως ορίζουν τον μέγιστο αριθμό εγγραφών σε 10^6 .

Κατά την διάρκεια αποθήκευσης, τα ηλεκτρόνια που έχουν εισέλθει στην πύλη μπορούν μέσω του μονωτικού υλικού, ειδικά σε υψηλές θερμοκρασίες και απώλειες φόρτισης, να προκαλέσουν διαγραφή δεδομένων. Οι κατασκευαστές συνήθως εγγυώνται την ασφάλεια των δεδομένων για 10 χρόνια ή περισσότερα.

Σχετικοί τύποι

Η μνήμη flash είναι μια μεταγενέστερη μορφή της EEPROM. Στην βιομηχανία υπάρχει μια συμφωνία να διατηρείται ο όρος EEPROM των byte-wise διαγραφόμενων μνημών έναντι των block-wise διαγραφόμενων μνημών flash. Η EEPROM παίρνει περισσότερη περιοχή θανάτου σε σχέση με την μνήμη flash για την ίδια ποσότητα δεδομένων διότι κάθε πυρήνας συνήθως χρειάζεται μαζί τρανζίστορ ανάγνωσης, εγγραφής και διαγραφής ενώ η μνήμη flash το κύκλωμα διαγραφής το έχει μοιρασμένο σε μεγάλα block πυρήνων (συχνά 512x8).

Σύγχρονες μνήμες όπως οι FeRAM και οι MRAM σιγά-σιγά αντικαθιστούν τις EEPROM σε μερικές εφαρμογές αλλά δύναται να παραμείνει μια μικρή ποσότητα EEPROM για μελλοντικές εφαρμογές.

Σύγκριση μεταξύ EPROM και EEPROM/Flash

Η διαφορά μεταξύ EPROM και EEPROM έγκειται στο πώς η κάθε μνήμη προγραμματίζεται και διαγράφεται. Η EEPROM μπορεί να προγραμματιστεί και να σβηστεί εφαρμόζοντας μια εκπομπή ηλεκτρονίων (περισσότερο γνωστή σαν “Fowler – Nordheim tunneling”).

Οι EPROM δεν μπορούν να διαγραφούν ηλεκτρικά και προγραμματίζονται διαμέσου εισαγωγής ρεύματος στην πύλη του τρανζίστορ. Η διαγραφή γίνεται με υπεριώδης πηγή ακτινοβολίας, αν και στην πράξη πολλές EPROM είναι περικλειούνται σε πλαστικές θήκες που δεν επιτρέπουν την έκθεση σε υπεριώδη ακτινοβολίες και είναι προγραμματιζόμενες μόνο για μία φορά.

Οι περισσότερες NOR Flash μνήμες είναι ένα υβρίδιο, προγραμματίζονται με εισαγωγή ρεύματος και διαγράφονται με το φαινόμενο “Fowler – Nordheim tunneling”.

Κατασκευαστές EEPROM

- ON Semiconductor
- Mitsubishi
- Atmel
- Hitachi
- Infineon
- Maxwell Technologies
- Microchip Technology
- NXP Technology
- ROHM
- Samsung Electronics
- Renesas Technology
- STMicroelectronics
- Seiko Instruments
- Winbond

Κεφάλαιο 7

Πληροφορίες για τα αισθητήρια (sensors)

Αισθητήρια (sensors)

Ο σένσορας (αισθητήριο) είναι μια συσκευή που μετράει μια φυσική ποσότητα και την μετατρέπει σε ένα σήμα από έναν παρατηρητή ή ένα όργανο. Για παράδειγμα, ένα θερμόμετρο υδραργύρου μετατρέπει την θερμοκρασία σε επιμήκυνση και διαστολή ενός υγρού το οποίο μπορεί να διαβαστεί μέσω ενός βαθμονομημένου γυάλινου σωλήνα. Ένα θερμοζεύγος μετατρέπει την θερμοκρασία σε τάση η οποία μπορεί να διαβαστεί με ένα βολτόμετρο. Για να έχουμε ακρίβεια, όλοι οι σένσορες πρέπει να ρυθμιστούν βάσει γνωστών τιμών.

Χρήση

Σένσορες χρησιμοποιούνται σε καθημερινές συσκευές όπως σε πλήκτρα ασανσέρ ευαίσθητα στην αφή και σε λάμπες οι οποίες αλλάζουν την φωτεινότητα τους ανάλογα με το φως του δωματίου. Υπάρχουν αμέτρητες εφαρμογές για σένσορες τις οποίες οι περισσότεροι άνθρωποι αγνοούν. Αυτές περιλαμβάνουν αυτοκίνητα, μηχανές, αεροπλάνα, φάρμακα, κατασκευές και ρομπότ.

Ο σένσορας είναι μια συσκευή που λαμβάνει και αντιδράει σε σήματα ή ερεθίσματα. Σε αυτή την περίπτωση, με τον όρο "ερέθισμα" εννοούμε μια ποσότητα ή μια μεταβλητή που πρέπει να μετατραπεί σε ηλεκτρική μορφή. Έτσι, σένσορας μπορεί να οριστεί σαν μια συσκευή σαν μια συσκευή η οποία λαμβάνει ένα σήμα και το μετατρέπει σε ηλεκτρική μορφή η οποία μπορεί μετά να χρησιμοποιηθεί από ηλεκτρικές συσκευές. Ένας σένσορας διαφέρει από έναν μετασχηματιστή με την έννοια ότι ένας μετασχηματιστής μετατρέπει μια μορφή ενέργειας σε μια άλλη ενώ ένας σένσορας μετατρέπει ένα σήμα εισόδου σε ηλεκτρική μορφή μόνο.

Η ευαισθησία ενός σένσορα δείχνει κατά πόσο η έξοδος του αλλάζει όταν η τιμή της μετρούμενης ποσότητας μεταβάλλεται. Για παράδειγμα, αν ο υδράργυρος σε ένα θερμόμετρο ανέβει κατά 1εκ. όταν η θερμοκρασία ανέβει και αυτή κατά 1 °C, τότε η ευαισθησία είναι 1εκ./ °C. Σένσορες οι οποίες μετράνε πολύ μικρές αλλαγές, παρουσιάζουν μεγάλη ευαισθησία. Επίσης έχουν αντίκτυπο σε κάθε τι που μετράνε. Για παράδειγμα, ένα θερμόμετρο δωματίου το οποίο το τοποθετούμε μέσα σε καυτό ποτήρι με υγρό, ψύχει το υγρό την ίδια ώρα που το υγρό θερμαίνει το θερμόμετρο. Ο σένσορας χρειάζεται να σχεδιαστεί κατάλληλα ώστε να έχει μια μικρή επίδραση σε αυτό που μετράει. Κάνοντας τον πιο μικρό, συχνά βελτιώνουμε αυτό το χαρακτηριστικό και μπορεί να παρουσιαστούν και άλλα πλεονεκτήματα. Η τεχνολογική πρόοδος επιτρέπει σε όλο και πιο πολλούς σένσορες να κατασκευάζονται σε μικρο-κλίμακες σαν μικρο-σένσορες χρησιμοποιώντας MEMS τεχνολογία. Στις πιο πολλές περιπτώσεις, ένας μικρο-σένσορας επιτυγχάνει μεγάλες ταχύτητες και αυξημένη ευαισθησία.

Ταξινόμηση μετρήσιμων λαθών

Ένας σωστός σένσορας υπακούει στους παρακάτω κανόνες:

- Ευαισθησία στην μετρήσιμη ιδιότητα
- Μη-ευαισθησία σε οτιδήποτε άλλη ιδιότητα που μπορεί να επιδράσει στην μέτρηση
- Δεν επηρεάζει την μετρήσιμη ιδιότητα

Οι ιδανικοί σένσορες είναι σχεδιασμένοι να είναι γραμμικοί ή γραμμικοί σε μια απλή μαθηματική συνάρτηση της μέτρησης, συνήθως λογαριθμική. Το σήμα εξόδου ενός τέτοιου σένσορα πρέπει να είναι γραμμικά ανάλογο με την μετρήσιμη τιμή ή μια συνάρτηση της μετρήσιμης ιδιότητας. Η ευαισθησία σε αυτήν την περίπτωση ορίζεται σαν την αναλογία μεταξύ του σήματος εξόδου και της μετρήσιμης ιδιότητας. Για παράδειγμα, αν ένας σένσορας μετράει θερμοκρασία και έχει σαν έξοδο τάση, η ευαισθησία είναι μια σταθερά με μονάδα $[V/K]$. Ο σένσορας αυτός είναι γραμμικός επειδή η αναλογία είναι σταθερή σε όλα τα σημεία μέτρησης.

Αποκλίσεις Σένσορα

Εάν ο σένσορας δεν είναι ιδανικός, αρκετοί τύποι αποκλίσεων μπορούν να παρατηρηθούν:

- Η ευαισθησία μπορεί στην πράξη να διαφέρει από την δοθείσα τιμή. Αυτό ονομάζεται λάθος ευαισθησίας αλλά ο σένσορας παραμένει γραμμικός.
- Επειδή το εύρος του σήματος εξόδου είναι πάντα πεπερασμένο, το σήμα εξόδου θα φθάσει ένα μίνιμουμ ή ένα μέγιστουμ όταν η μετρήσιμη ιδιότητα ξεπεράσει τα όρια. Το εύρος της πλήρους κλίμακας ορίζει τις ελάχιστες και τις μέγιστες τιμές της μετρήσιμης ιδιότητας.
- Εάν το σήμα εξόδου δεν είναι μηδενικό όταν η μετρήσιμη τιμή είναι μηδέν, ο σένσορας έχει ένα αντιστάθμισμα ή μια κλίση. Αυτό ορίζεται σαν την έξοδο του σένσορα σε μηδενική είσοδο.
- Εάν η ευαισθησία δεν είναι σταθερή πάνω στο εύρος του σένσορα, αυτό ονομάζεται μη-γραμμικότητα. Συνήθως αυτό ορίζεται από το πόσο η έξοδος διαφέρει από την ιδανική συμπεριφορά πάνω στο πλήρες εύρος του σένσορα, συχνά αναφέρεται σαν ένα ποσοστό από το πλήρες εύρος.
- Εάν η απόκλιση έχει προκληθεί από μια βίαιη αλλαγή της μετρήσιμης ιδιότητας κατά την διάρκεια του χρόνου, τότε έχουμε ένα δυναμικό λάθος. Συχνά, αυτή η συμπεριφορά περιγράφεται με ένα γράφημα του λάθους ευαισθησίας σε συνάρτηση με την διαφορά φάσης πάνω στην συχνότητα του περιοδικού σήματος εισόδου.
- Εάν το σήμα εξόδου αλλάζει με χαμηλό ρυθμό και ανεξάρτητα από την μετρήσιμη ιδιότητα, αυτό ορίζεται σαν ολίσθημα (τηλεπικοινωνίες).
- Ολίσθημα μακράς διάρκειας συνήθως δείχνει μια χαμηλή υποβάθμιση των ιδιοτήτων του σένσορα μετά από μια μακρά περίοδο χρήσης.

- Ο “θόρυβος” είναι μια τυχαία απόκλιση του σήματος που αλλάζει με τον χρόνο.
- Η “υστέρηση” είναι ένα λάθος που προκαλείται όταν η μετρήσιμη ιδιότητα αλλάζει κατεύθυνση αλλά υπάρχει μια πεπερασμένη καθυστέρηση για τον σένσορα να απαντήσει, δημιουργώντας ένα διαφορετικό αντιστάθμισμα λάθους προς την μία κατεύθυνση.
- Εάν ο σένσορας έχει ψηφιακή έξοδο, η έξοδος είναι στην ουσία μια εκτίμηση της μετρήσιμης ιδιότητας. Το λάθος εκτίμησης είναι επίσης λέγεται ψηφιακό λάθος.
- Εάν το σήμα παρακολουθείται ψηφιακά, το όριο της συχνότητας δειγματοληψίας επίσης μπορεί να προκαλέσει ένα δυναμικό λάθος, ή εάν ο μεταβλητός ή πρόσθετος θόρυβος μεταβάλλεται περιοδικά σε μια συχνότητα κοντά σε πολλαπλάσιο της τιμής δειγματοληψίας μπορεί να προκαλέσει ενοχλητικά λάθη.
- Ο σένσορας μπορεί σε κάποια έκταση να είναι ευαίσθητος σε ιδιότητες διαφορετικές από την μετρήσιμη ιδιότητα. Για παράδειγμα, οι περισσότεροι σένσορες επηρεάζονται από την θερμοκρασία του περιβάλλοντος τους.

Όλες αυτές οι αποκλίσεις μπορούν να καταχωρηθούν σαν λάθη συστήματος ή τυχαία λάθη. Τα λάθη συστήματος κάποιες φορές χρησιμοποιούνται σαν κάποια μορφή στρατηγικής ρύθμισης. Ο θόρυβος είναι ένα τυχαίο λάθος που μπορεί να μειωθεί με επεξεργασία σήματος, όπως το φιλτράρισμα, συνήθως με κόστος την δυναμική συμπεριφορά του σένσορα.

Ανάλυση

Η ανάλυση ενός σένσορα ορίζεται σαν την μικρότερη αλλαγή που μπορεί να παρατηρηθεί στην ποσότητα η οποία μετρείται. Συχνά σε μια ψηφιακά οθόνη, το λιγότερο σημαντικό ψηφίο θα διακυμανθεί, δείχνοντας ότι μια αλλαγή αυτού του μεγέθους έχει μόλις επιλυθεί. Η ανάλυση είναι σχετική με την ακρίβεια με την οποία γίνεται η μέτρηση. Για παράδειγμα, ένα ακροσωλήνιο σάρωσης (μια μικρή άκρη κοντά στην επιφάνεια συλλέγει ένα ηλεκτρόνιο ρεύματος) μπορεί να ξεχωρίσει άτομα και μόρια.

Κεφάλαιο 8

Χρήσιμες εφαρμογές με την χρήση του BX-24p



Χρησιμοποιώντας ADC για να διαβάσουμε ένα ποτενσιόμετρο

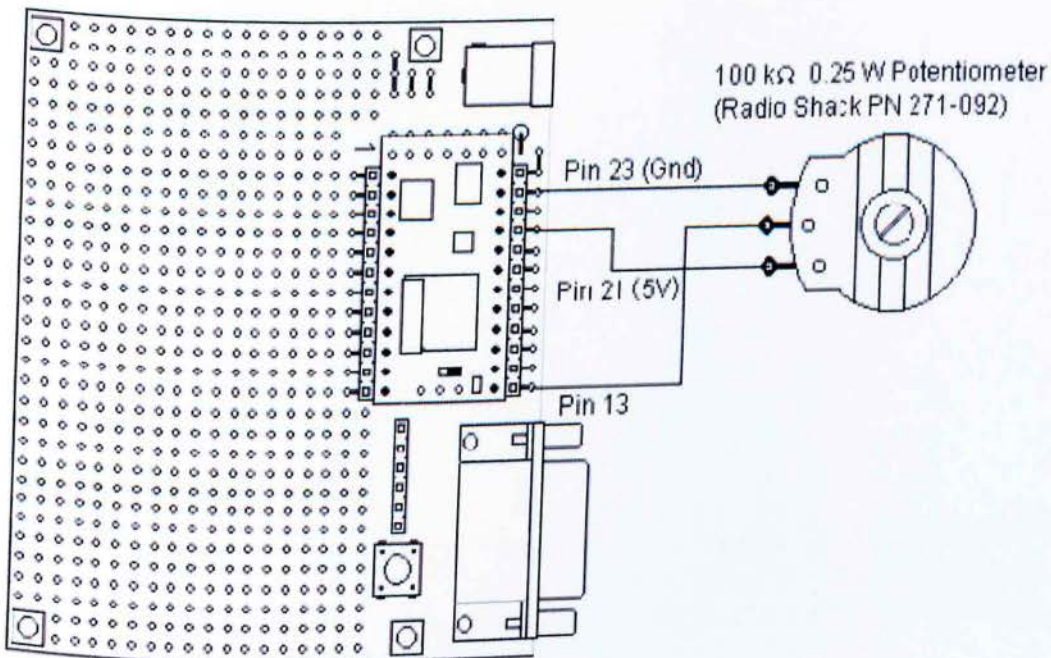
Συνδέοντας έναν αναλογικό-σε-ψηφιακό μετατροπέα στον BX-24p

Η εφαρμογή αυτή περιγράφει τις μεθόδους hardware και software για την επέμβαση ενός ποτενσιόμετρου στον ADC μετατροπέα ενός BX-24p (Σχήμα 1).

Ένας ADC μετατρέπει μια αναλογική τάση με ψηφιακή μορφή. Ο BX-24p περιέχει 8 κανάλια από 10-bit ADCs, τα οποία είναι συνδεδεμένα στα I/O pins 13 έως 20. Οι ADCs έχουν τάση εισόδου από 0.0 V έως 5.0 VDC (σε 5 V συστήματα) και θα αποδώσουν μετρημένες τάσεις σε μορφή γραμμικών 10 bit ψηφιακών τιμών (0 to 1023).

Η ανάλυση είναι περίπου 4.9 mV. Η μέγιστη ταχύτητα δειγματοληψίας είναι 6000 δείγματα το δευτερόλεπτο.

Το παρακάτω σχήμα δείχνει τις συνδέσεις που είναι απαραίτητες για την επέμβαση ενός ποτενσιόμετρου στον BX-24p χρησιμοποιώντας το pin 13 σαν παράδειγμα.



Σχήμα 1

Σύνδεση ποτενσιόμετρου στον BX-24p

Σύνταξη προγράμματος

Το σύστημα καλεί την εντολή GetADC για να διαβάσει τον ADC μετατροπέα.

Υπάρχουν δύο εκδόσεις της GetADC. Η ακέραια έκδοση επιστρέφει την ακριβή 10-bit ακέραια τάση. Η επιπλέον έκδοση επιστρέφει μια μη-διαστασιολογημένη τιμή τάσης στο εύρος 0.0 to 1.0.

Const PinNumber As Byte = 13

Dim iV As Integer, V As Single

' Integer version; iV ranges from 0 to 1023.

iV = GetADC(PinNumber)

' Float version; V ranges from 0.0 to 1.0.

Call GetADC(PinNumber, V)

Χρησιμοποιώντας ΤΟ ADXL202 Επιταχυνσιόμετρο με τον BX-24p

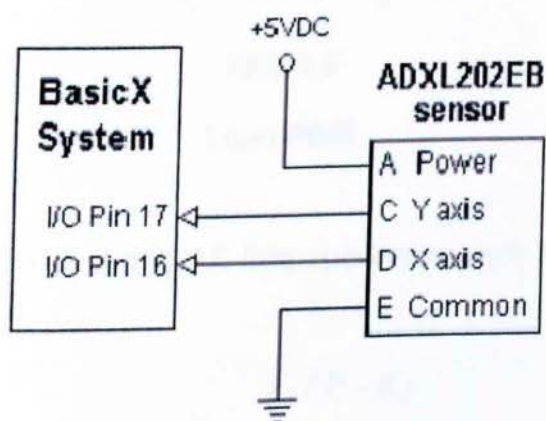
Εισαγωγή

Η αναλογική συσκευή ADXL202 είναι ένα επιταχυνσιόμετρο 2 αξόνων χαμηλού κόστους το οποίο μπορεί εύκολα να συνδεθεί με 8-bit μικροεπεξεργαστές. Είναι ικανό να διαβάσει τόσο στατικές όσο και δυναμικές επιταχύνσεις με ένα εύρος μετρήσεων $\pm 19.6 \text{ m/s}^2$ ($\pm 2 \text{ g}$). Η ανάλυση εξαρτάται από το εύρος και είναι 49 mm/s^2 (5 mg) στα 60 Hz.

Ο αισθητήρας παράγει ένα παλμικού εύρους διαμορφωμένο σήμα (PWM) σε καθένα από τα 2 κανάλια εξόδου του. Το εξάρτημα επιτάχυνσης σε κάθε άξονα είναι εύκολα εξαγωγίμο μετρώντας τον κύκλο εργασιών κάθε σήματος PWM το οποίο σημαίνει ότι ένας ADC μετατροπέας δεν είναι αναγκαίος.

Hardware interface

Στο σχήμα 1 βλέπουμε πώς να συνδέσουμε ένα ADXL202EB, το οποίο είναι μια έκδοση του ADXL202:



Σχήμα 1

Σύνδεση ADXL202EB στον BX-24p

Είναι πιθανό να έχουμε τον BX-24p και τον αισθητήρα να μοιράζονται την ίδια +5 VDC τάση, αλλά αυτή δεν είναι η επιλογή με τον λιγότερο δυνατό «θόρυβο». Μια ξεχωριστή πηγή ενέργειας θα πρέπει να χρησιμοποιηθεί αν θέλουμε να έχουμε τον λιγότερο δυνατό θόρυβο.

Περιβάλλον Software

Σύμβολα που χρησιμοποιούνται σε αυτό το τμήμα:

A_x = X συνιστώσα του διανύσματος επιτάχυνσης, μη διαστασιολογημένες μονάδες βαρύτητας.

A_y = Y συνιστώσα του διανύσματος επιτάχυνσης, μη διαστασιολογημένες μονάδες βαρύτητας.

K_1 = σταθερά, μη διαστασιολογημένη

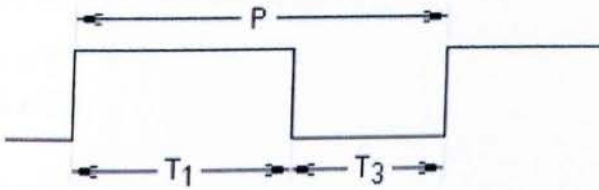
K_2 = σταθερά, μη διαστασιολογημένη

P = περίοδος του σήματος PWM, σε δευτερόλεπτα

T_1 = διάρκεια του παλμού υψηλής λειτουργίας, σε δευτερόλεπτα

T_3 = διάρκεια του παλμού χαμηλής λειτουργίας, σε δευτερόλεπτα

Το παρακάτω σχήμα δείχνει το σήμα PWM που παράγεται από το ένα κανάλι:



Σχήμα 2

Σήμα PWM

Η X συνιστώσα επιτάχυνσης A_x διαμορφώνεται από το ακόλουθο (A_y είναι ίδιο):

$$A_x = \frac{T_1/P - K_1}{K_2}$$

Οι σταθερές K_1 και K_2 διαφέρουν μεταξύ των αισθητηρίων και πρέπει να εκτιμηθούν εμπειρικά.

Συνήθεις τιμές είναι 0.5 για K_1 και 0.125 για K_2 .

Το πρώτο βήμα όταν μετράμε επιτάχυνση είναι να προσδιορίσουμε την περίοδο P. Μπορούμε να καλέσουμε την εντολή PulseIn για να μετρήσουμε τα πλάτη των υψηλής και χαμηλής λειτουργίας παλμών. Αρκετά δείγματα μπορούν να ελεγχτούν για να βρεθεί η περίοδος. Εδώ παίρνουμε 20 δείγματα:

```
Const PinX As Byte = 16, PinY As Byte = 17
Dim SumT1 As Single, SumT3 As Single, Period As Single
Dim T1 As Single, T3 As Single, i As Integer
Dim AvgT1 As Single, AvgT3 As Single
Const NSamples As Integer = 20
```

```
SumT1 = 0.0
SumT3 = 0.0
For i = 1 to NSamples
Call PulseIn(PinX, 1, T1) ' High-going pulse.
Call PulseIn(PinX, 0, T3) ' Low-going pulse.
```

```
SumT1 = SumT1 + T1
SumT3 = SumT3 + T3
Next
```

```
AvgT1 = SumT1 / CSng(NSamples)
AvgT3 = SumT3 / CSng(NSamples)
```

```
Period = AvgT1 + AvgT3 ' Units are in seconds.
```

Η περίοδος εξελίσσεται σχετικά αργά, οπότε δεν χρειάζεται η διαδικασία αυτή να γίνει περισσότερα από μια φορά το λεπτό. Επίσης, μόνο ένας άξονας χρειάζεται να μετρηθεί από την στιγμή που η περίοδος είναι ίδια και για τους δύο άξονες.

Το PulseIn επιστρέφει μονάδες σε επιπλέοντα δευτερόλεπτα, τα οποία κάνουν τις εκφράσεις πιο εύκολες να γραφτούν. PulseIn έχει ένα όριο – η ακέραια έκδοση μπορεί να χρησιμοποιηθεί αν απαιτείται υψηλότερη επίδοση.

Αν η περίοδος είναι γνωστή, είναι απλό να μετρήσουμε κάθε άξονα. Η επόμενη ακολουθία επιστρέφει την φιλτραρισμένη επιτάχυνση σε κάθε άξονα:

```
Sub GetAccelerations( _  
ByVal Period As Single, _  
ByRef Ax As Single, _  
ByRef Ay As Single)
```

```
' This procedure reads each axis of an ADXL202 accelerometer. Both  
' components of a 2D acceleration vector are returned. Units are  
' in gravities.
```

```
Dim T1X as Single, T1Y As Single  
Dim SumX As Single, SumY As Single  
Dim i As Byte, T1 As Single  
Const NSamples As Byte = 20  
Const PinX As Byte = 16, PinY As Byte = 17
```

```
SumX = 0.0
```

```
SumY = 0.0
```

```
For i = 1 to NSamples
```

```
Call PulseIn(PinX, 1, T1)
```

```
SumX = SumX + T1
```

```
Call PulseIn(PinY, 1, T1)
```

```
SumY = SumY + T1
```

```
Next
```

```
' Take averages for each axis.
```

```
T1X = SumX / CSng(NSamples)
```

```
T1Y = SumY / CSng(NSamples)
```

```
' Determine the acceleration components.
```

```
Ax = ((T1X / Period) - 0.5) / 0.125
```

```
Ay = ((T1Y / Period) - 0.5) / 0.125
```

```
End Sub
```

Μπλοκ δεδομένων κλάσεων

Εισαγωγή

Η εφαρμογή αυτή περιγράφει πώς να χρησιμοποιούμε μπλοκ δεδομένων κλάσεων σε έναν BX-24p.

Πίνακας τοποθέτησης αρχικών τιμών

Ένα πρόβλημα με τους συμβατικούς πίνακες είναι ότι δεν υπάρχει εύκολος τρόπος να τους αρχικοποιήσεις χωρίς να έχεις προβλήματα στην απόδοση. Οι συνηθισμένες γλώσσες Basic χρησιμοποιούν DATA δηλώσεις για πανομοιότυπους σκοπούς, αλλά οι DATA δηλώσεις είναι δύσκολες στο να χρησιμοποιηθούν και δεν είναι συμβατές με την VB.

Ο BX-24p λύνει αυτό το πρόβλημα παρέχοντας system-defined μπλοκ δεδομένων κλάσεων, τα οποία είναι ισοδύναμα των αρχικά τοποθετημένων τιμών στην μνήμη:

Σε αυτό το παράδειγμα, θα χρησιμοποιήσουμε έναν 1-D πίνακα bytes. Τα πρώτο βήμα είναι να δηλώσουμε ένα μπλοκ δεδομένων αντικείμενο, το οποίο πρέπει να είναι σε κώδικα module-level:

```
Private Vector As New ByteVectorDataRW
```

Το επόμενο βήμα είναι να ορίσουμε την πηγή των δεδομένων:

```
Call Vector.Source("ByteVector.txt")
```

Η μέθοδος πηγής ορίζει ένα text αρχείο αποθηκευμένο στον υπολογιστή. Πρέπει να την καλέσουμε πριν κάποια άλλη πρόσβαση στον μπλοκ πίνακα δεδομένων:

Μόλις το αντικείμενο οριστεί, μπορούμε να διαβάσουμε και να γράψουμε σε αυτό σαν να ήταν πίνακας. Σε αυτήν την περίπτωση, ο πίνακας συμβαίνει να τοποθετείται στην μνήμη EEPROM, στην οποία είναι επίσης αποθηκευμένο το πρόγραμμα. Σύνταξη:

```
Dim B As Byte
```

```
' Write to element 1.  
Vector(1) = B
```

```
' Read element 1.  
B = Vector(1)
```



Προειδοποίηση – ένα μπλοκ αντικείμενο δεδομένων είναι παρόμοιο με μία επίμονη μεταβλητή όσο αφορά στο γράψιμο ορίων κύκλων και στον χρόνο που χρειάζεται να γράψεις το αντικείμενο.

Πρόγραμμα παραδείγματος

Αυτό το πρόγραμμα διαβάζει και γράφει τα πρώτα πέντε στοιχεία ενός byte διανύσματος αντικειμένου:

```
Private Vector As New ByteVectorDataRW ' Read-write.
```

```
Public Sub Main()
```

```
Dim N As Byte
```

```
Dim B As Byte
```

```
Call Vector.Source("ByteVector.txt")
```

```
Debug.Print
```

```
Debug.Print " Data using object access:"
```

```
For N = 1 To 5
```

```
Debug.Print " Vector("; CStr(N); ") = ";
```

```
Debug.Print Cstr(Vector(N))
```

```
Next
```

```
Debug.Print
```

```
Debug.Print " EEPROM address of Vector: ";
```

```
Debug.Print CStr(Vector.DataAddress)
```

```
Debug.Print
```

```
Debug.Print " Data using direct EEPROM access:"
```

```
For N = 0 To 4 ' Note shift to 0-base.
```

```
Call GetEEPROM( Vector.DataAddress + CLng(N), B, 1 )
```

```
Debug.Print " Vector("; CStr(N+1); ") = "; CStr(B)
```

```
Next
```

```
' Modify the second element upon first download.
```

```
If ( FirstTime() ) Then
```

```
Vector(2) = 255
```

```
End If
```

```
End Sub
```

Το αρχείο ByteVector.txt περιέχει πέντε γραμμές κειμένου:

10

20

30

40

50

Αυτή είναι η έξοδος του προγράμματος μετά το κατέβασμα:

Data using object access:

```
Vector(1) = 10  
Vector(2) = 20  
Vector(3) = 30  
Vector(4) = 40  
Vector(5) = 50
```

EEPROM address of Vector: 1495

Data using direct EEPROM access:

```
Vector(1) = 10  
Vector(2) = 20  
Vector(3) = 30  
Vector(4) = 40  
Vector(5) = 50
```

Η λειτουργία FirstTime προκαλεί στο πρόγραμμα να τρέχει διαφορετικά την πρώτη φορά μετά το κατέβασμα. Την δεύτερη φορά που θα τρέξει το πρόγραμμα μετά το κατέβασμα, η έξοδος θα είναι η εξής:

Data using object access:

```
Vector(1) = 10  
Vector(2) = 255  
Vector(3) = 30  
Vector(4) = 40  
Vector(5) = 50
```

EEPROM address of Vector: 1495

Data using direct EEPROM access:

```
Vector(1) = 10  
Vector(2) = 255  
Vector(3) = 30  
Vector(4) = 40  
Vector(5) = 50
```

Βλέπουμε ότι η τιμή του στοιχείου 5 άλλαξε από 20 σε 255.

Επίσης παρατηρούμε ότι μπορούμε να έχουμε πρόσβαση στην διεύθυνση EEPROM του αντικειμένου. Σε αυτήν την περίπτωση, το αντικείμενο βρίσκεται στην διεύθυνση 1495. Μπορούμε να τσεκάρουμε την διεύθυνση συμβουλευόντας τον χάρτη MPP, ο οποίος έχει την τοποθεσία του αντικειμένου, όπως επίσης και το πλήρες pathname του αρχείου ByteVector.txt.

Άλλα μπλοκ δεδομένων κλάσεις

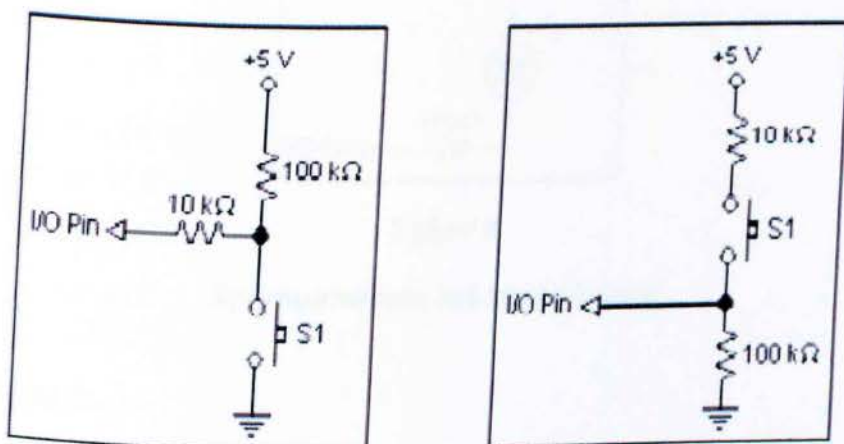
Το παραπάνω παράδειγμα μας δείχνει πώς να χρησιμοποιήσουμε έναν 1-D Byte πίνακα. Μπορούμε επίσης να δουλέψουμε με 2-D πίνακες του τύπου Byte, Integer, Long and Single. Επιπρόσθετα, μπορούμε να ελέγξουμε εάν το αντικείμενο θα είναι μόνο-ανάγνωση ή μόνο-εγγραφή.

Αλληλεπίδραση πλήκτρων και διακόπτων

Εισαγωγή

Η εφαρμογή αυτή περιγράφει έναν αριθμό hardware και software μεθόδων αλληλεπίδρασης πλήκτρων και διακόπτων με τον BX-24p.

Η ανάγκη να λαμβάνουμε πληροφορίες ή δεδομένα εισαγωγής από τον έξω κόσμο είναι ένα σημαντικό κομμάτι στις περισσότερες εφαρμογές μικρο-επεξεργαστών. Για παράδειγμα, ένας συνδυασμός πλήκτρων μπορεί να χρησιμοποιηθεί για να ξεκινήσει να λειτουργεί ένας τόνος που είναι βασισμένος στον BX-24p. Την ίδια στιγμή, διακόπτες ασφαλείας μπορούν να επιτρέψουν στον BX-24p να αντιληφθεί ότι ένα προστατευτικό κάλυμμα έχει αφαιρεθεί, με συνέπεια ο μικρο-επεξεργαστής να σταματήσει την μηχανή για λόγους ασφαλείας.



Σχήμα 1 και 2

Μέθοδοι ανάγνωσης πλήκτρων και διακόπτων

Hardware αλληλεπίδραση

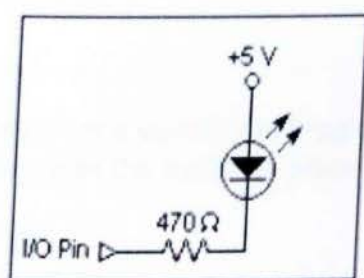
Υπάρχουν αρκετοί τρόποι που μπορούμε να διαβάσουμε ένα πλήκτρο ή έναν διακόπτη. Σε αυτή την εφαρμογή θα εστιάσουμε στις δύο πιο συνηθισμένες μεθόδους. Η πρώτη μέθοδο (σχήμα 1) περιέχει το να παραπέμψουμε την I/O pin high διαμέσου μιας υψηλής τιμής αντίστασης. Ένας διακόπτης τραβάει την I/O pin διαμέσου μιας δεύτερης μικρότερης αντίστασης. Αυτή η μέθοδος δουλεύει ισάξια καλά τόσο για normally-open όσο και normally-closed διακόπτες.

Η δεύτερη μέθοδος (σχήμα 2) περιέχει το να παραπέμψουμε την I/O pin στην είωση η στο σημείο low διαμέσου μιας σχετικά υψηλής τιμής αντίστασης. Μια αντίσταση 100 kΩ χρησιμοποιείται στο διάγραμμα παραδείγματος. Το

πλήκτρο χρησιμοποιείται για να τραβήξει την I/O pin high διαμέσου μιας μικρότερης αντίστασης. Και πάλι, αυτή η μέθοδος δουλεύει ισάξια καλά τόσο για normally-open όσο και normally-closed διακόπτες.

Παράδειγμα προγράμματος

Αυτό το πρώτο παράδειγμα υποθέτει ότι χρησιμοποιούμε μια I/O pin 16 ενός BX-24p και ότι έχουμε έναν διακόπτη συνδεδεμένο σύμφωνα με το διάγραμμα στο σχήμα 1. Το πρόγραμμα διαβάζει την κατάσταση κάθε διακόπτη συνδεδεμένο στο pin και γράφει αυτές τις τιμές σε ένα δεύτερο pin (I/O pin 17, σε αυτήν την περίπτωση). Η λογική κατάσταση του δεύτερου I/O pin μπορεί να επαληθευτεί με έναν παλμογράφο ή ένα λογικό κύκλωμα. Ένα LED μπορεί επίσης να χρησιμοποιηθεί (σχήμα 3).



Σχήμα 3

Χρησιμοποίηση led επαλήθευσης

Παράδειγμα 1:

```
Sub Main()
```

```
Dim State as Byte
```

```
Do
```

```
' Read I/O pin 16.  
State = GetPin(16)
```

```
' Copy the state to pin 17.
```

```
Call PutPin(17, State)
```

```
Loop
```

```
End Sub
```

Αυτό το πρόγραμμα λειτουργεί καλά με διακόπτες, αλλά αν δοκιμάσουμε αυτό το παράδειγμα χρησιμοποιώντας στιγμιαίο πλήκτρο θα παρατηρήσουμε ότι το pin 17 μένει μόνο στην στιγμιαία κατάσταση για όσο το πλήκτρο είναι πατημένο. Αυτό το πρόγραμμα δεν λειτουργεί καλά εάν θέλουμε να κρατάμε πατημένο η κατάσταση ενός I/O pin και να μένει εκεί χωρίς να χρειάζεται να κρατάμε πατημένο το πλήκτρο.

Σε περίπτωση που θέλουμε να κάνουμε χρήση στιγμιαίων πλήκτρων ή διακοπών, χρειάζεται μια διαφορετική προσέγγιση. Το παράδειγμα αυτό υποθέτει ότι χρησιμοποιούμε το pin 16 για την είσοδο και το pin 17 για την έξοδο και έχουμε ένα στιγμιαίο πλήκτρο συνδεδεμένο σύμφωνα με το διάγραμμα του σχήματος 1. Το παράδειγμα προγράμματος 2 θα χρησιμοποιήσει την bitwise binary XOR λειτουργία για να μεταπίπτει την κατάσταση του pin 17 κάθε φορά που το στιγμιαίο πλήκτρο πιέζεται.

Παράδειγμα προγράμματος 2:

```
Sub Main()
```

```
' This program reads the state of a switch and toggles the  
' state of an output pin whenever the switch is pressed.
```

```
Const InputPin As Byte = 16  
Const OutputPin As Byte = 17
```

```
' Configure pins.  
Call PutPin(InputPin, bxInputTristate)  
Call PutPin(OutputPin, bxOutputHigh)
```

```
Dim State as Byte
```

```
State = 0
```

```
Do  
' Toggle State if switch is pressed.  
If GetPin(InputPin) = 0 Then
```

```
State = State Xor 1
```

```
' Pause a quarter-second for button de-bounce.  
Call Delay(0.25)  
End If
```

```
' Write State to output pin.  
Call PutPin(OutputPin, State)  
Loop
```

```
End Sub
```

Μετρώντας παλμούς με Hardware διακοπές

Εισαγωγή

Αυτή η εφαρμογή δείχνει πώς μπορούμε να χρησιμοποιήσουμε τις διακοπές του BX-24p για να μετρήσουμε παλμούς. Η WaitForInterrupt εντολή χρησιμοποιείται για να έχουμε πρόσβαση στο INT1 pin BX-24p. Μόλις ένα πρόγραμμα καλέσει την WaitForInterrupt, το πρόγραμμα μπλοκάρεται μέχρι να συμβεί μια διακοπή, το οποίο σημαίνει ότι τυπικά χρησιμοποιείται πολύ λίγη ισχύ από την CPU, σε σύγκριση με εναλλακτικές προσεγγίσεις όπως τα polling loops.

Κάλεσμα WaitForInterrupt

Η WaitForInterrupt επιτρέπει στα προγράμματα να ανταποκρίνονται ταχύτερα στις διακοπές σε μια I/O pin. Στο παράδειγμα που ακολουθεί, η εφαρμογή PulseCountTask τρέχει σε ένα διαρκές loop και καλεί την WaitForInterrupt. Η εφαρμογή μπλοκάρεται όταν ένα υψηλό άκρο εμφανίζεται στο INT1 pin, το οποίο είναι το pin 11 στον BX-24p.

Κάθε φορά που συμβαίνει η διακοπή, ο μετρητής PulseCount αυξάνεται και το κόκκινο LED του BX-24p ανάβει με σκοπό να έχουμε οπτική ένδειξη.

Μια ακόμη εφαρμογή του προγράμματος μετατρέπει τον αριθμό των παλμών σε μια string μεταβλητή και την στέλνει στον υπολογιστή μέσω της Com1 σειριακής θύρας. Αυτό γίνεται κάθε ένα δευτερόλεπτο.

```
Private Const StackSize As Integer = 25  
Private PulseCountStack(1 To StackSize) As Byte  
Private PulseCount As Integer
```

```
Public Sub Main()
```

```
Call InitializeBX24
```

```
PulseCount = 0
```

```
CallTask "PulseCountTask", PulseCountStack
```

```
Do  
Debug.Print CStr(PulseCount)  
Call Delay(1.0)  
Loop
```

```
End Sub
```

```
Private Sub PulseCountTask()
```

```
Const LEDpin As Byte = 25 ' Red LED
```

```
Const LEDon As Byte = 0
```

```
Const LEDoff As Byte = 1
```

```
Do
```

```
Call WaitForInterrupt(bxPinRisingEdge)
```

```
PulseCount = PulseCount + 1
```

```
Call PutPin(LEDpin, LEDon)
```

```
' Debounce.
```

```
Call Sleep(0.1)
```

```
Call PutPin(LEDpin, LEDoff)
```

```
Loop
```

```
End Sub
```

```
Private Sub InitializeBX24()
```

```
' BX-24 only -- configure INTO as input-pullup in order  
' to prevent it from causing an unwanted interrupt.  
' INTO is (internal) pin 11 on the 8535 chip, and is not  
' connected to an external pin.
```

```
Register.DDRD = Register.DDRD And bx1111_1011
```

```
Register.PORTD = Register.PORTD Or bx0000_0100
```

```
End Sub
```

Αν και η κύρια εφαρμογή μεταδίδει τον αριθμό παλμών σε έναν αργό ρυθμό, η εντολή PulseCountTask, τρέχοντας στο παρασκήνιο, είναι ικανή να ανταποκρίνεται ταχύτερα σε ένα υψηλό μέτωπο στο pin διακοπής.

Ανάλογα με το πόσο συχνά ο παλμός εφαρμόζεται, η PulseCountTask μπορεί να περνάει τον περισσότερο χρόνο στην WaitForInterrupt δίχως να καταναλώνει τον υψηλότερο χρόνο CPU εναλλακτικών προσεγγίσεων όπως τα polling loops. Αυτό συνήθως αυξάνει την ποσότητα CPU διαθέσιμου χρόνου για άλλες εφαρμογές, εκτός και αν ο παλμικός ρυθμός είναι τόσο υψηλός ώστε η PulseCountTask να μονοπωλεί την CPU.

Ξημειώνουμε ότι η WaitForInterrupt παράμετρος μπορεί να ορίσει 3 διαφορετικούς τύπους διακοπών – χαμηλού μετώπου, υψηλού μετώπου, αμυγλής λογικής.

ADC Ατζέντα δεδομένων 8 καναλιών

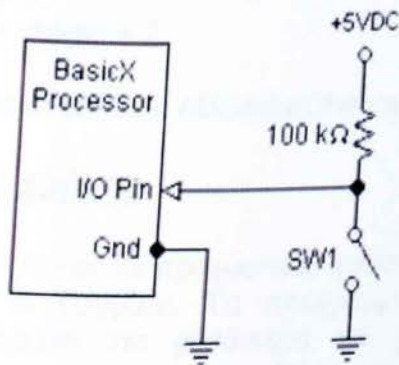
Εισαγωγή

Η εφαρμογή αυτή δείχνει πώς μπορούμε να χρησιμοποιήσουμε τον BX-24p σαν μια ατζέντα δεδομένων 8 καναλιών (σχήμα 1). Κάθε πέντε λεπτά το σύστημα διαβάζει και καταχωρεί δεδομένα από κάθε μία από τις 8 ADC εισόδους του BX-24p. Ο 10 bit ακέραιος αριθμός δεδομένων καταχωρείται σε μια EEPROM μνήμη.

Τα καταχωρημένα δεδομένα μπορούν να καλεστούν οποιαδήποτε στιγμή απλώς βάζοντας την ατζέντα σε κατάσταση upload. Σε αυτή την κατάσταση, όλα τα αποθηκευμένα δεδομένα μεταδίδονται από την σειριακή θύρα σε μορφή ASCII text strings.

Περίληψη Προγράμματος

Ο κύριος κώδικας που συνοδεύει την εφαρμογή έχει δύο καταστάσεις λειτουργίας – (1) data logging, και (2) upload. Η κατάσταση καθορίζεται από ένα διακόπτη SW1 και την θέση στην οποία αυτός βρίσκεται κατά την ενεργοποίηση. Αν ο διακόπτης SW1 είναι σε ανοιχτή θέση (υψηλή λογική), το πρόγραμμα θα τρέχει σε κατάσταση data logging. Σε αυτήν την κατάσταση ο BX-24p θα πάρει την τιμή από κάθε ADC κανάλι μία φορά κάθε πέντε λεπτά.



Σχήμα 1

Διακόπτης κατάστασης upload

Ο διακόπτης SW1 είναι συνδεδεμένος με μια I/O pin του BX-24p. Ο αριθμός αυτού του pin εξαρτάται από τον τύπο του επεξεργαστή. Με το διακόπτη SW1 σε κλειστή θέση, το I/O pin είναι στα 0V και το πρόγραμμα θα τρέχει σε κατάσταση upload. Με τον διακόπτη σε ανοιχτή θέση, το pin είναι στα +5 V, και το πρόγραμμα θα τρέχει σε κατάσταση data logging.

Εάν ο διακόπτης SW1 είναι σε κλειστή θέση (χαμηλή λογική) κατά την ενεργοποίηση, το πρόγραμμα θα τρέχει σε κατάσταση upload. Σε αυτή την κατάσταση, το πρόγραμμα επανακτά τα αποθηκευμένα ADC δεδομένα από

την μνήμη EEPROM, ένα δείγμα την φορά. Κάθε μπλοκ 8 ADC τιμών μετατρέπεται σε ASCII string και μεταδίδεται από την σειριακή θύρα Com1. Σημείωση – με τον διακόπτη SW1 στην θέση δεδομένων, εάν πατήσουμε το reset, ή έχουμε απώλεια ενέργειας, το πρόγραμμα θα αρχίσει από την αρχή. Με το που γίνει η επανεκκίνηση, το πρόγραμμα θα γράψει πάνω στις προηγούμενες τιμές.

Μπορούμε σαν δυνατότητα να αλλάξουμε τα πρόγραμμα να καθαρίζει αυτόματα την μνήμη EEPROM στην οποία είναι τα δεδομένα αποθηκευμένα. Η εντολή ClearMemoryBlock μπορεί να καλεστεί εάν η εντολή FirstTime επιστρέψει τιμή true, το οποίο σημαίνει ότι αυτή ήταν η πρώτη φορά που η συνάρτηση καλέστηκε από την στιγμή που άρχισε το πρόγραμμα να λειτουργεί. Η αρχικοποίηση μπορεί να πάρει αρκετά λεπτά και μπορεί να παρακαμφθεί εάν δεν μας ενδιαφέρει για αρχικοποίηση της μνήμης.

Κώδικας χαμηλού επιπέδου

Για να αποθηκεύσουμε και να καλέσουμε δεδομένα σε μια EEPROM, το σύστημα καλεί τις εντολές PutEEPROM και GetEEPROM.

Παράδειγμα σύνταξης PutEEPROM:

```
Dim CurrentAddress As Long
Dim ADCdata(1 To 8) As Byte
Dim Channel As Integer
Const ElementSize As Integer = 2
```

```
Call PutEEPROM(CurrentAddress, ADCdata(Channel), ElementSize)
```

Κρατώντας αρχείο χρόνου

Η ατζέντα χρησιμοποιεί ένα ρολόι πραγματικού χρόνου για να καθορίσει κάθε πότε θα παίρνονται τα δείγματα. Τα πρόγραμμα το καταφέρνει αυτό αρχικοποιώντας τον χρόνο στα μεσάνυχτα και μετά χρησιμοποιεί την συνάρτηση Timer για να κρατάει αρχείο των δευτερολέπτων.

Όταν ο χρόνος που περνάει είναι ίδιος με την επιθυμητό χρόνο δείγματος, ο οποίος στη περίπτωση μας είναι 5 λεπτά (300 δευτερόλεπτα), ένα νέο δείγμα λαμβάνεται. Σε αυτό το σημείο, το RTC ξαναγυρνάει στα μεσάνυχτα και η διαδικασία αρχίζει από την αρχή.

Αν και ο BX-24p δεν κρατάει αρχείο για την κάθε χρονική στιγμή που έχουμε ένα δείγμα, οι ρυθμοί δειγματοληψίας είναι ξεκάθαροι. Από την στιγμή που κάθε δείγμα λαμβάνεται κάθε 5 λεπτά, ο χρόνος του μπορεί να υπολογιστεί από την ημέρα που έχει παρθεί, προσθέτοντας στην συνέχεια πέντε λεπτά για κάθε δείγμα. Για παράδειγμα, εάν η ατζέντα άρχισε στις 1:00 MM, το πρώτο δείγμα πρέπει να καταχωρήθηκε στις 1:05 MM και το δέκατο δείγμα πρέπει τότε να καταχωρήθηκε στις 1:50 MM.

Καθορίζοντας το μπλοκ της EEPROM

Κάθε ADC δείγμα απαιτεί 2 bytes, και για να αποθηκεύσουμε όλα τα δεδομένα χρειαζόμαστε 16 bytes. Με ένα ρυθμό δειγματοληψίας των 16 bytes κάθε 300 s (5 λεπτά), και υποθέτοντας περίπου 28 KB EEPROM διαθέσιμα, αυτό μεταφράζεται σε 6 ημέρες αποθήκευσης.

Από την στιγμή που το πρόγραμμα μοιράζεται την EEPROM με τα αποθηκευμένα δεδομένα, πρέπει να προσέξουμε να αποφύγουμε συγκρούσεις στην μνήμη. Ο MPP χάρτης αρχείων μπορεί να χρησιμοποιηθεί για να καθορίσουμε την διαθέσιμη μνήμη αποθήκευσης των δεδομένων. Ο χάρτης MPP δημιουργείται κάθε φορά που αρχίζουμε ένα πρόγραμμα.

Σαν εναλλακτική, μπορούμε να χρησιμοποιήσουμε μπλοκ δεδομένων για να αποθηκεύσουμε δεδομένα στην EEPROM. Το πλεονέκτημα με τα μπλοκ δεδομένων είναι ότι μοιάζουν αρκετά με τους πίνακες, το οποίο σημαίνει ότι δεν χρειάζεται να κάνουμε κλήσεις χαμηλού επιπέδου όπως GetEEPROM και PutEEPROM. Επίσης, το πρόγραμμα αυτόματα καταχωρεί τα δεδομένα στην μνήμη και ενημερώνει αν υπάρχουν συγκρούσεις.

Ένα άλλο πλεονέκτημα είναι ότι το πρόγραμμα μπορεί αν αρχικοποιήσει την μνήμη EEPROM ταχύτερα από την ατζέντα – αρκετά δευτερόλεπτα για το πρόγραμμα εναντίον αρκετών λεπτών για την ατζέντα.

Προγραμματίζοντας τον Timer1 για διπλό παλμό διαμόρφωσης πλάτους

Εισαγωγή

Ο BX-24p περιέχει έναν ενσωματωμένο χρονιστή, τον Timer1. Ο χρονιστής μπορεί αν χρησιμοποιηθεί για αρκετές λειτουργίες, μία από τις οποίες είναι να παράγει διπλές hardware PWM εξόδους. Τα Pins OC1A και OC1B χρησιμοποιούνται σαν εξόδοι, τα οποία είναι τα pins 26 και 27. Έχουμε την επιλογή από 8-, 9- or 10-bit PWM εξόδων, όπως και επιλογή διάφορων συχνοτήτων.

Ο χρονιστής είναι ικανός να λειτουργεί σε πέντε διακριτές συχνότητες με εύρος από 7.20 kHz έως 7.37 MHz. Μετά την επεξεργασία, η συχνότητα μεταφράζεται σε PWM παλμό βαθμολογημένο από 3.52 Hz (για 10-bit εξόδους) έως 14.4 kHz (για 8-bit εξόδους).

Το επόμενο είναι μια λίστα με τις κυριότερες καταχωρήσεις απαραίτητες για να έχουμε πρόσβαση στον Timer1:

Type	Name	Description
Byte	TCNT1L	Timer/Counter1 Low Byte
Byte	TCNT1H	Timer/Counter1 High Byte
Byte	TCCR1B	Timer/Counter1 Control Register B
Byte	TCCR1A	Timer/Counter1 Control Register A

Προσοχή – ο Timer1 πρέπει να χρησιμοποιείται μόνο σε περίπτωση που δεν υπάρχει σύγκρουση με άλλες πηγές συστήματος, όπως τα InputCapture και OutputCapture, όλα τα οποία εξαρτώνται από τον Timer1.

Προγραμματίζοντας τον Timer1 για διπλό PWM

Αρχικοποιήσεις

Το παράδειγμα δείχνει πως χρησιμοποιούμε τον Timer1 για να παράγουμε 8-bit PWM εξόδους στα pins OC1A and OC1B, τα οποία είναι τα pins 26 και 27. Σημειώνουμε ότι το pin 26 είναι μοιρασμένο με το πράσινο LED, το οποίο το καθιστά ιδανικό για να έχουμε μια οπτική επιβεβαίωση της λειτουργίας PWM σε αυτό το pin.

Η πρώτη αρχικοποίηση είναι να σταματήσουμε τον χρονιστή:

```
register.TCCR1B = 0
```

Το επόμενο βήμα είναι να θέσουμε τον Timer1 σε 8-bit PWM mode (9- και 10-bit modes είναι παρόμοια):

Const PWMmode8bit As Byte = bx0000_0001
 Const PWMmode9bit As Byte = bx0000_0010
 Const PWMmode10bit As Byte = bx0000_0011
 Const PWMmodeOff As Byte = bx0000_0000

Register.TCCR1A = PWMmode8bit

Τώρα χρειαζόμαστε να καθορίσουμε τον κύκλο εργασίας για κάθε pin, το οποίο είναι φορτωμένο στα OCR1A και OCR1B. Ο κύκλος εργασιών μπορεί αν είναι 8, 9 ή 10 bit αριθμός. Σε αυτό το παράδειγμα, θα χρησιμοποιήσουμε 8-bit τιμές για 25% και 75% κύκλο εργασιών (64 και 191). Σημειώνουμε ότι τα υψηλά bytes πρέπει να γραφτούν για αυτούς τους καταχωρητές:

Register.OCR1AH = 0

Register.OCR1AL = 64 ' = 25 %

Register.OCR1BH = 0

Register.OCR1BL = 191 ' = 75 %

Ξεκινώντας τον Timer1

Τώρα μπορούμε να ξεκινήσουμε τον χρονιστή γράφοντας μία από τις πέντε αριθμημένες τιμές στον καταχωρητή TCCR1B.

Οι παρακάτω τιμές είναι:

TCCR1B Value	Tick Frequency (Hz)	8-bit PWM Pulse Rate (Hz)
1	7 372 800	14 456
2	921 600	1 807
3	115 200	225.9
4	28 800	56.47
5	7 200	14.12

Σε αυτό το παράδειγμα θα χρησιμοποιήσουμε την χαμηλότερη συχνότητα των 7.2 kHz:

Register.TCCR1B = 5

Εδώ η συχνότητα παλμών είναι ισότιμη με την συχνότητα του Timer1 διαιρεμένη κατά 510, η οποία σε αυτήν την περίπτωση είναι περίπου 14.1 Hz. Για 9-bit εξόδους, ο διαιρέτης είναι 1022, και για 10-bit εξόδους ο διαιρέτης είναι 2046.

Το τελευταίο βήμα είναι να συνδέσουμε τον χρονιστή με τα pins OC1A και OC1B:

```
Const MaskOC1A As Byte = bx1000_0000  
Const MaskOC1B As Byte = bx0010_0000
```

```
' Initialize both pins to output-low.  
Call PutPin(PinOC1A, bxOutputLow)  
Call PutPin(PinOC1B, bxOutputLow)
```

```
' Enable PWM for both pins.  
Register.TCCR1A = Register.TCCR1A Or MaskOC1A  
Register.TCCR1A = Register.TCCR1A Or MaskOC1B
```

Μετά από αυτό το σημείο μπορούμε να μεταβάλλουμε συνέχεια τον κύκλο εργασιών PWM γράφοντας καινούργιες τιμές στους καταχωρητές OCR1A και OCR1B. Οι έξοδοι PWM παράγονται στο παρασκήνιο.

Προγραμματίζοντας τον Timer1 για σύλληψη αλλαγής παλμού

Ο Timer1 και το pin εισόδου

Ο BX-24p περιλαμβάνει έναν ενσωματωμένο χρονιστή που ονομάζεται Timer1. Ο χρονιστής μπορεί να χρησιμοποιηθεί για αρκετές λειτουργίες, μία από τις οποίες είναι να μετράει τον χρόνο καθυστέρησης της αλλαγής ενός παλμού από υψηλό σε χαμηλό ή το αντίστροφο σε ένα pin.

Γιατί να χρησιμοποιήσουμε τον χρονιστή? Το κύριο πλεονέκτημα είναι ότι ο επεξεργαστής δεν χρειάζεται να περιμένει για να αλλάξει η κατάσταση σε ένα pin. Σε αντίθεση, μπορούσε το σύστημα να καλέσει την Rctime, η οποία εκτελεί μια παρόμοια ενέργεια. Η Rctime επιτρέπει να καθορίσουμε μια γενικού σκοπού I/O pin σε input-tristate (υψηλή σύνθετη αντίσταση) και να μετρήσουμε τον χρόνο καθυστέρησης της μετάβασης. Το πρόβλημα το την Rctime είναι ότι εμπλέκει τον επεξεργαστή στην μέτρηση. Το ρολόι πραγματικού χρόνου, η αλλαγή εφαρμογής και το φόρτο του δικτύου "παγώνουν" κατά την διάρκεια εκτέλεσης της συνάρτησης. Καθώς ο Timer1 περιμένει για την μετάβαση μετώπου, ο επεξεργαστής μπορεί να συνεχίσει να ασχολείται με άλλες εφαρμογές.

Υπάρχουν όμως και μειονεκτήματα σε αυτή την μέθοδο. Ο Timer1 πρέπει μόνο να χρησιμοποιείται μόνο όταν δεν υπάρχουν συγκρούσεις με άλλα συστήματα, όπως οι Com2 ή Com3 σειριακές πόρτες, όπως και οι λειτουργίες InputCapture και OutputCapture, όλες οι οποίες εξαρτώνται από τον Timer1.

Θα αναρωτιέστε γιατί δεν μπορούμε να χρησιμοποιήσουμε την λειτουργία InputCapture για μετάβαση μετώπου. Υπάρχουν δύο προβλήματα – πρώτον, η InputCapture υπάρχει στην πραγματικότητα για να μετράει παλμούς πλάτους, και δεν καταγράφει την καθυστέρηση χρόνου της πρώτης μετάβασης. Δεύτερον, η διαδικασία θα κολλήσει εάν η μετάβαση δεν συμβεί ποτέ.

Δυνατότητες Timer1

Μπορούμε να προσπεράσουμε αυτά τα προβλήματα χρησιμοποιώντας τον Timer1. Ο χρονιστής είναι ικανός να επεξεργάζεται σε πέντε διακριτές συχνότητες από 7.20 kHz έως 7.37 MHz, το οποίο σημαίνει ότι μπορούμε να χρησιμοποιήσουμε τον Timer1 για να μετρήσουμε τις διακοπές χρόνου με εύρος από 136 ns έως 139 μ s.

Οι παρακάτω μεταβλητές χρησιμοποιούνται για την μετάβαση μετώπου:

Type	Name	Description
Byte	TCNT1L	Timer/Counter1 Low Byte
Byte	TCNT1H	Timer/Counter1 High Byte
Byte	TCCR1B	Timer/Counter1 Control Register B
Byte	TCCR1A	Timer/Counter1 Control Register A
Byte	TIFR	Timer/Counter Interrupt Flag register
Byte	ICR1L	T/C 1 Input Capture Register High Byte
Byte	ICR1H	T/C 1 Input Capture Register High Byte

Προσοχή -- Ο Timer1 πρέπει μόνο να χρησιμοποιείται μόνο όταν δεν υπάρχουν συγκρούσεις με άλλα συστήματα, όπως οι Com2 ή Com3 σειριακές πόρτες, όπως και οι λειτουργίες InputCapture και OutputCapture, όλες οι οποίες εξαρτώνται από τον Timer1.

Programming Timer1 for edge capture

Αρχικοποιήσεις

Πρέπει να αρχικοποιήσουμε τον χρονοστή πριν τον χρησιμοποιήσουμε. Το πρώτο βήμα είναι να καθαρίσουμε τον καταχωρητή του Timer1 TCCR1A. Αυτό αποσυνδέει τον Timer1 από τα pins OC1A and OC1B, και απενεργοποιεί την λειτουργία PWM:

```
Register.TCCR1A = 0
```

Το επόμενο βήμα είναι να σταματήσουμε τον χρονοστή:

```
Register.TCCR1B = 0
```

Τώρα πρέπει να καθαρίσουμε τα 2 bytes του μετρητή Timer1. Το πρώτο υψηλό byte πρέπει να γραφτεί πρώτα, ακολουθούμενο από το χαμηλό:

```
Register.TCNT1H = 0
```

```
Register.TCNT1L = 0
```

Χρειάζεται επίσης να καθαρίσουμε το Input Capture Flag 1 (ICF1), το οποίο γίνεται γράφοντας 1 σε ένα bit στον καταχωρητή TIFR:

```
Const ICF1 As Byte = bx00001000 ' BX-01
```

```
Const ICF1 As Byte = bx00100000 ' BX-24, BX-35
```

```
Register.TIFR = ICF1
```

Ο Timer/Counter1 Overflow Flag (TOV1) πρέπει να καθαριστεί, πάλι γράφοντας στο κατάλληλο bit του TIFR:

```
Const TOV1 As Byte = bx10000000 ' BX-01  
Const TOV1 As Byte = bx00000100 ' BX-24, BX-35
```

```
Register.TIFR = TOV1
```

Η τελευταία αρχικοποίηση καθορίζει για το αν αναζητούμε για ένα μέτωπο ανεβάσματος ή κατεβάσματος. Η είσοδος Capture1 Edge Select (ICES1) bit αναλαμβάνει αυτήν την δουλειά. Το ICES πρέπει να έχει ρυθμιστεί στο 1 για μέτωπο ανεβάσματος ή στο 0 για ένα μέτωπο κατεβάσματος:

```
Const ICES1 As Byte = bx01000000
```

```
Register.TCCR1B = ICES1 ' For rising edge.
```

Starting Timer1

Μόλις ο Timer1 έχει αρχικοποιηθεί, αρχίζει γράφοντας μία από τις πέντε τιμές στα χαμηλότερα τρία bits του καταχωρητή TCCR1B. Οι τιμές φαίνονται στο παρακάτω πίνακα:

TCCR1B Value	Timer Resolution (μs)	Tick Frequency (Hz)	Maximum Time Range (s)
1	0.135 633 7	7 372 800	0.008 889
2	1.085 069	921 600	0.071 11
3	8.680 555	115 200	0.568 9
4	34.722 22	28 800	2.276
5	138.888 9	7 200	9.102

Σε αυτό το παράδειγμα θα χρησιμοποιήσουμε συχνότητα 28.8 kHz:

```
Register.TCCR1B = Register.TCCR1B + 4
```

Περιμένοντας για μετάβαση

Μόλις ο χρονιστής έχει αρχικοποιηθεί και ξεκινήσει, το πρόγραμμα χρειάζεται να κοιτάξει στην τιμή του flag ICF1, το οποίο αυτόματα θέτεται όταν η επιθυμητή μετάβαση στο pin συμβεί. Μόλις το flag τοποθετηθεί, ο επεξεργαστής επίσης αντιγράφει την τιμή του χρονιστή στους καταχωρητές ICR1L και ICR1H. Εάν η μετάβαση δεν συμβεί πριν ο χρονιστής υπερχειλίσει, το flag TOV1 τοποθετείται.

Ο ακόλουθος κώδικας περιμένει είτε για μια μετάβαση είτε για μια υπερχειλίση, ότι έρθει πρώτο:

```
Dim HasOverflowed As Boolean, TIFRcopy As Byte
```

```
HasOverflowed = False
```

```
Do
```

```
TIFRcopy = Register.TIFR
```

```
' Bail out if timer overflows.
```

```
If ((TIFRcopy And TOV1) = TOV1) Then
```

```
HasOverflowed = True
```

```
Exit Do
```

```
End If
```

```
Loop Until ((TIFRcopy And ICF1) = ICF1) ' Stop if edge is detected.
```

Κατά την διάρκεια αυτής της επανάληψης, άλλες διαδικασίες μπορούν να τρέχουν, περιλαμβάνοντας χειρισμός δικτύων, σειριακές θύρες και ρολόγια πραγματικού χρόνου. Ο επεξεργαστής δεν χρειάζεται να προσέχει για να δει αν υπάρχει μετάβαση στο pin.

Σημειώνουμε ότι κάνουμε ένα αντίγραφο του καταχωρητή TIFR από το να τσεκάρουμε τον καταχωρητή κατευθείαν. Αυτό γίνεται επειδή πρέπει να ελέγξουμε δύο διαφορετικά bits στον καταχωρητή. Κάνοντας ένα αντίγραφο, παγώνουμε το υπόδειγμα του καταχωρητή και δεν έχουμε να ανησυχούμε για αλλαγές των bits μεταξύ της μετάβασης και της υπερχειλίσης.

Εάν δεν συμβεί καμία υπερχειλίση, η τιμή του χρονιστή μπορεί να αντιγραφεί στους καταχωρητές ICR1L και ICR1H. Είναι σημαντικό να διαβάζουμε πρώτα το χαμηλό byte και μετά το υψηλό byte:

```
Dim LowByte As Byte, HighByte As Byte, Count As New UnsignedInteger
```

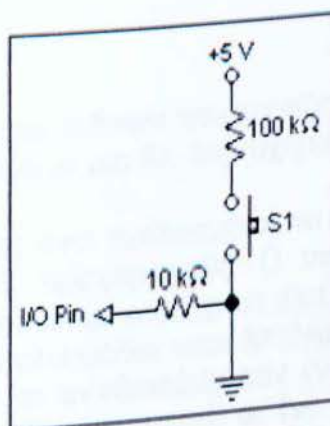
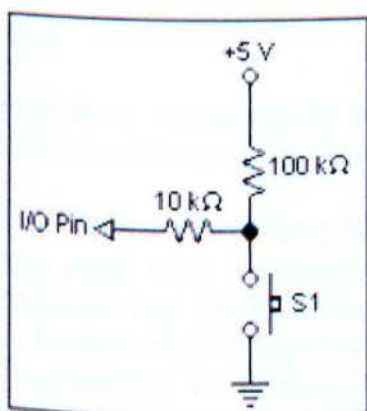
```
LowByte = Register.ICR1L
```

```
HighByte = Register.ICR1H
```

```
Count = CuiInt(HighByte) * 256 + CuiInt(LowByte)
```

Το τελικό αποτέλεσμα *Count* είναι μια τιμή με ένα εύρος από 0 έως 65 535 μονάδες. Η μετατροπή μονάδας σε δευτερόλεπτα εξαρτάται από τον Timer. Στα 28.8 kHz, ο παράγοντας κλίμακας είναι περίπου 3.47×10^{-5} , το οποίο μας δίνει ένα όριο περίπου $(65\ 535 \text{ μονάδες})(3.47 \times 10^{-5} \text{ s/unit}) = 2.27$ δευτερόλεπτα πριν την υπερχειλίση

Κώδικας παραδείγματος



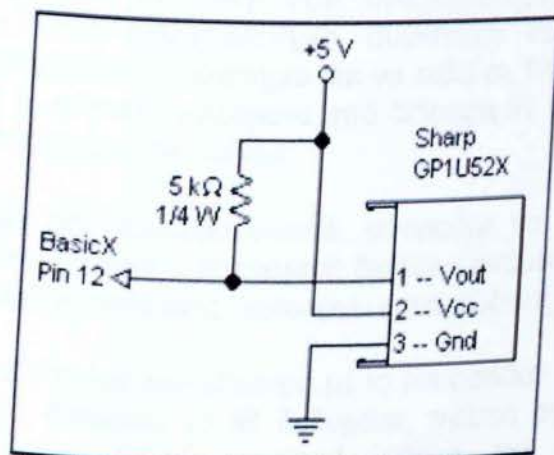
Σχήμα 1 και 2

Διασύνδεση υπέρυθρων αισθητήριων με τον BX-24p

Εισαγωγή

Η εφαρμογή αυτή περιγράφει το hardware και software που χρειάζεται για να συνδέσουμε έναν Sharp υπέρυθρο αισθητήρα με ένα BX-24p (σχήμα 1).

Ο Sharp GP1U52X υπέρυθρος αισθητήρας είναι σχεδιασμένος να λαμβάνει IR σήματα από τους περισσότερους IR τηλεχειρισμούς. Ο αισθητήρας αναδιαμορφώνει ένα 40 kHz IR σήμα και το μετατρέπει σε ένα σειριακό δεδομένο. Συνδέοντας την συσκευή σε ένα pin εισόδου στον BX-24p (pin 12), μπορεί αν συλλάβει τα δεδομένα και να τα αποθηκεύσει σαν ένα πίνακα ακεραίων. Ο πίνακας στην συνέχεια μπορεί να μετατραπεί σε ένα μοναδικό σετ εντολών.



Σχήμα 1

Σύνδεση Sharp GP1U52X στον BX-24p

Hardware διασύνδεση

Όπως φαίνεται και στο σχήμα 1, συνδέοντας ένα Sharp IR ανιχνευτή σε ένα BX-24p απαιτεί μόνο μια αντίσταση 5 kΩ και τρία καλώδια.. Μπορούμε να συνδέσουμε την αντίσταση στο πίσω μέρος της IR μονάδας και να εξαλείψουμε τον κίνδυνο να χρειαστούμε ένα ξεχωριστό κύκλωμα. Η συσκευή χρειάζεται επίσης μια σταθερή τάση 5 volt για αξιόπιστη λειτουργία. Για ρυθμισμένη τάση, μπορούμε να χρησιμοποιήσουμε τον pin 21 BX-24p. Το pin 23 μπορεί επίσης να χρησιμοποιηθεί σαν γείωση.

Παράδειγμα προγράμματος

Υπάρχουν τρία παραδείγματα που ακολουθούν την συγκεκριμένη τεχνική. Κάθε πρόγραμμα διαβάζει ένα παλμό και χρησιμοποιεί την σειριακή θύρα του BX-24p για να στείλει τα δεδομένα στον υπολογιστή.

IRcapture.bas

Displays the pulse widths of an InputCapture data stream

IRbinary.bas

Converts IR code to sequence of 15 binary digits

IRcodes.bas

Converts IR code to 16-bit number

Το πρόγραμμα IRcapture είναι χρήσιμο αν έχουμε άγνωστη πηγή IR. Το IRcapture δεν εξαρτάται από κάποιο συγκεκριμένο σύστημα κωδικοποίησης – το πρόγραμμα απλώς δείχνει τα παλμικά πλάτη τα οποία διαβάζει, των οποίων οι μονάδες είναι σε microseconds.

Τα προγράμματα IRbinary και IRcodes είναι σχεδιασμένα να διαβάζουν IR κώδικες δεδομένων από τα Sharp VCR τηλεχειριστήρια. Εάν δεν υπάρχει κάποιο συμβατό Sharp τηλεχειριστήριο, συνιστάται να αγοραστεί κάποιο γενικά προγραμματιζόμενο χειριστήριο και να τεθεί σε IR code 048. Με αυτό τον τρόπο έχουμε άμεσα αποτελέσματα από διάφορα IR προγράμματα χωρίς να χρειαστεί να αλλάξουμε τον κώδικα.

Εάν το πρόγραμμα δεν δουλεύει σωστά, μπορούμε να χρησιμοποιήσουμε διαφορετικούς κώδικες, αφού οι κατασκευή διαφέρει ανάμεσα στα διαφορετικά προϊόντα. Σε μερικές περιπτώσεις, λειτουργεί και ο κώδικας 012.

Προσοχή – εάν συνδέσουμε τον σένσορα με το pin εισόδου και καλέσουμε την InputCapture για να διαβάσει τα IR δεδομένα, πρέπει αν αποφύγουμε να χρησιμοποιήσουμε την Com2 σειριακή πόρτα για να αποφύγουμε συγκρούσεις και με τον Timer1. Αυτό ισχύει και για τα τρία προγράμματα.

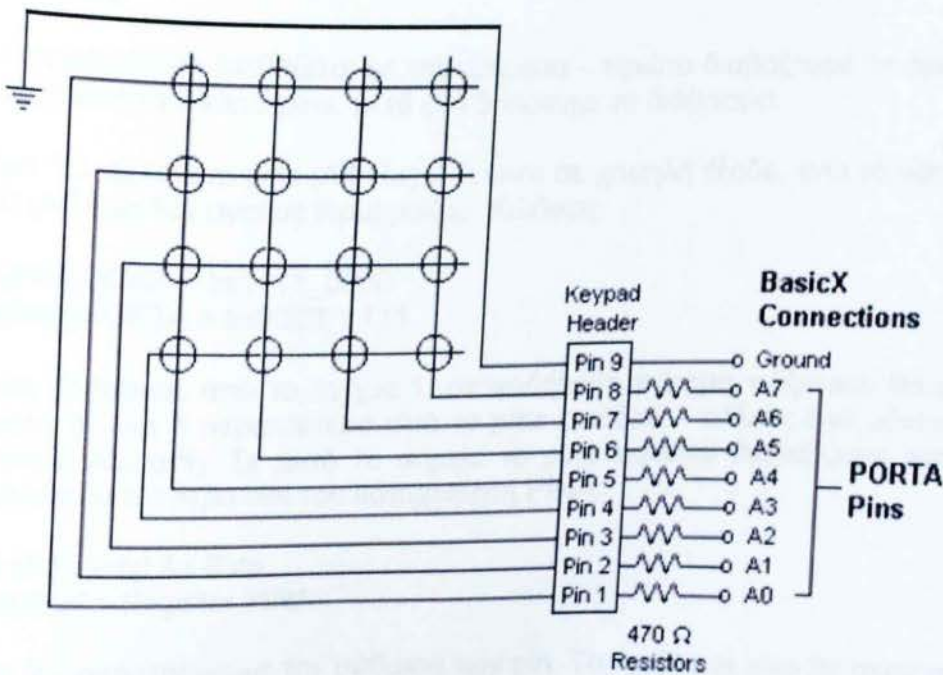
Διασύνδεση πληκτρολογίων με τον BX-24p

Εισαγωγή

Η εφαρμογή αυτή περιγράφει τις hardware και software μεθόδους που απαιτούνται για να υπάρξει διασύνδεση πληκτρολογίου με τον BX-24p (σχήμα 1).

Πληκτρολόγιο

Το μέγεθος ενός πληκτρολογίου καθορίζεται από τον αριθμό των πλήκτρων που διαθέτει και από το πώς αυτά είναι διαμοιρασμένα. Τυπικά ο πρώτος αριθμός μιας περιγραφής πληκτρολογίου (όπως 6 x 4) δηλώνει τον αριθμό των πλήκτρων που έχει σε σειρά και ο δεύτερος τον αριθμό των πλήκτρων που έχει σε στήλες. Πολλαπλασιάζοντας αυτούς τους δύο αριθμούς παίρνουμε τον συνολικό αριθμό των πλήκτρων. Όταν προσθέτουμε αυτούς τους δύο αριθμούς παίρνουμε τον αριθμό των I/O γραμμών που χρειαζόμαστε για να διαβάσουμε το πληκτρολόγιο.



Σχήμα 1

Διασύνδεση πληκτρολογίου με τον BX-24p

Hardware

Το σχήμα 1 μας δείχνει πώς να συνδέσουμε ένα τυπικό 4 x 4 πληκτρολόγιο στον BX-24p. Οι αντιστάσεις 470 Ω στις I/O γραμμές A0 - A7 δεν είναι απαραίτητες για να λειτουργήσει η συσκευή αλλά συστήνονται για να προστατευτούν οι γραμμές από στατικό ηλεκτρισμό.

Σε αυτό το παράδειγμα, ο καταχωρητής PORTA χρησιμοποιείται για να διαβάσει το πληκτρολόγιο. Όπως φαίνεται στο Figure 1, οι πρώτες τέσσερις I/O γραμμές από την PORTA (A0 έως A3) χρησιμοποιούνται για τις οριζόντιες συνδέσεις. Οι εναπομείναντες τέσσερις γραμμές (A4 έως A7) είναι δεσμευμένες στις κάθετες συνδέσεις.

BX-01 PORTA pin numbers: 32 to 39 (32 is MSbit, 39 is LSbit -- note reversed ordering)

BX-24 PORTA pin numbers: 13 to 20 (13 is MSbit, 20 is LSbit -- note reversed ordering)

Software

Το πληκτρολόγιο διαβάζεται σε τρία βήματα – πρώτα διαβάζουμε τα οριζόντια pins, έπειτα τα κάθετα pins, μετά συνδυάζουμε το διάβασμα.

Βήμα 1 – τα κάθετα pins (A4 έως A7) είναι σε χαμηλή έξοδο, ενώ τα οριζόντια pins (A0 έως A3) είναι σε input-pullup. Κώδικας:

```
Register.DDRA = bx1111_0000  
Register.PORTA = bx0000_1111
```

Όπως βλέπουμε από το σχήμα 1, οποιοδήποτε πάτημα πλήκτρου θα φέρει χαμηλό σε ένα ή περισσότερα από τα pins εισόδου – αλλιώς όλα μένουν σε υψηλή κατάσταση. Σε αυτό το σημείο τα pins εισόδου διαβάζονται από τα χαμηλότερα τέσσερα bits του καταχωρητή PINA:

```
Dim Horizontal As Byte  
Horizontal = Register.PINA
```

Βήμα 2 – αντιστρέφουμε την ρύθμιση των pin. Τα οριζόντια pins βρίσκονται σε χαμηλή έξοδο, ενώ τα κάθετα σε input-pullup. Τότε τα υψηλότερα 4 bits του PINA διαβάζουν:

```
Register.DDRA = bx0000_1111  
Register.PORTA = bx1111_0000
```

```
Dim Vertical As Byte  
Vertical = Register.PINA
```

Βήμα 3 – παίρνουμε και τα δύο 4-bit διαβάσματα του PINA και τα συνδυάζουμε σε μια 8-bit τιμή:

```
Dim KeypadValue As Byte  
KeypadValue = Vertical Or Horizontal
```

Ελέγχοντας Servos με τον BX-24p

Εισαγωγή

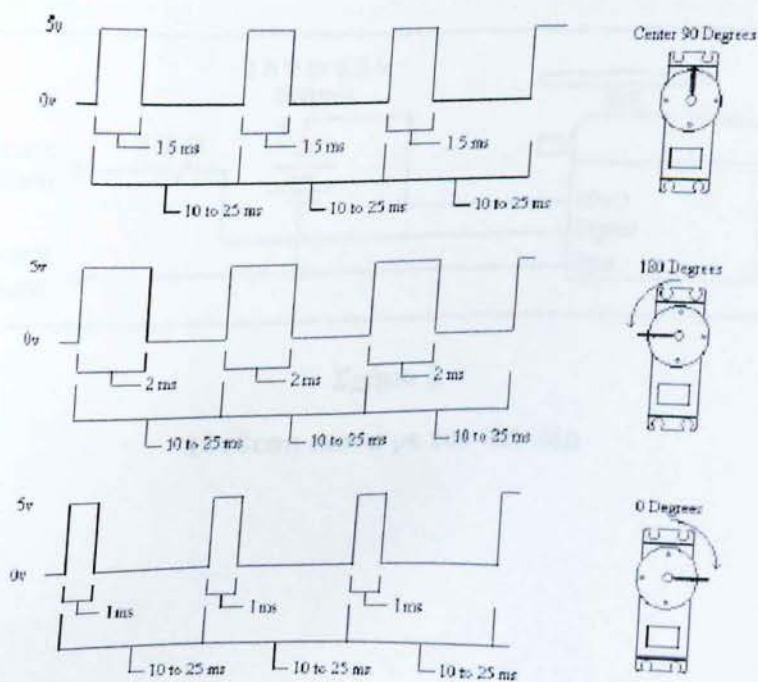
Η εφαρμογή αυτή περιγράφει τις hardware και software μεθόδους για την σύνδεση R/C type hobby servos στον BX-24p.

R/C hobby servos γρήγορα γίνονται πολύ δημοφιλή κομμάτια εφαρμογών ελεγχόμενων από υπολογιστές. Τα hobby servos αρχικά ήταν σχεδιασμένα και πωλούνταν με σκοπό τον έλεγχο μικρών μοντέλων (αεροπλάνα, αμάξια, βάρκες).

Hardware Σύνδεση

Τα Servos απαιτούν ηλεκτρική ενέργεια και ένα σήμα θέσης για να λειτουργήσουν. Τυπικά απαιτούν 4.8 VDC έως 6.5 VDC τάση. Οι απαιτήσεις σε ρεύμα ποικίλουν. Συχνά πρέπει να αναφερόμαστε στον κατασκευαστή για να μαθαίνουμε τις ακριβείς τιμές.

Το σήμα θέσης συνήθως αποτελείται από έναν 1 ms έως 2 ms παλμό σε συχνότητα 50 Hz. Ένα παράδειγμα ενός τέτοιου σήματος και τις κινήσεις που προκαλεί στο servo φαίνονται παρακάτω:



Σχήμα 1

Κινήσεις των servo σε συνάρτηση με την συχνότητα λειτουργίας

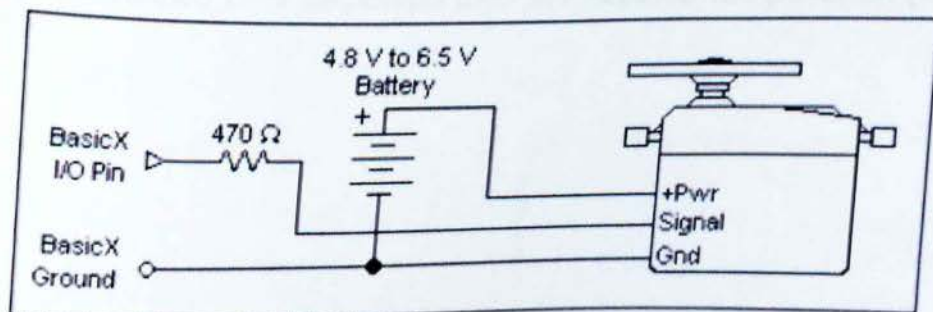
Το πλάτος παλμού μπορεί να διαφέρει ανάλογα το μοντέλο του servo. Μερικά servos έχουν μεγαλύτερη ακτίνα κίνησης και μπορούν να απαιτούν παλμούς των 0.5 ms ή των 2.2 ms για να επιτύχουν πλήρη κίνηση. Πρέπει όμως να προσέχουμε να μην υπερβούμε ένα ασφαλές εύρος -- ανάλογα με το servo, εάν ο παλμός είναι πολύ μεγάλος ή πολύ μικρός μπορούμε να προκαλέσουμε στο servo δυσλειτουργία και κίνδυνο για σφάλμα λειτουργίας.

Το βέλτιστο παλμικό εύρος μπορεί να ποικίλει. Εάν η συχνότητα ανανέωσης είναι πολύ αργή ή πολύ γρήγορη, το servo μπορεί να προκαλέσει έναν δυνατό θόρυβο. Εάν αυτό συμβεί, προσπαθούμε να ανεβάσουμε την κόρνα εξόδου -- εάν ο θόρυβος γίνει δυνατότερος, η συχνότητα ανανέωσης είναι πολύ χαμηλή. Αλλιώς είναι πολύ υψηλή.

Σύνδεση Servo

Το σχήμα 2 δείχνει μια προτεινόμενη μέθοδο για την διασύνδεση ενός servo με τον BX-24p. Τα κόκκινα και μαύρα καλώδια είναι συνήθως για τάση και το άσπρο καλώδιο είναι για το σήμα.

Σημείωση ότι το servo παίρνει ενέργεια από ξεχωριστή μπαταρία σε αυτό το παράδειγμα. Τα Servos τυπικά τραβάνε περισσότερη ενέργεια από όση το BasicX development board μπορεί να δώσει. Είναι επίσης σημαντικό να σημειώσουμε ότι τόσο η γείωση από το servo όσο και από τον BX-24p είναι δεμένες μαζί -- αλλιώς το servo μπορεί να μην λειτουργήσει σωστά.



Σχήμα 2

Σύνδεση servo με τον BX-24p

Κώδικας παραδείγματος

Χρησιμοποιώντας την λειτουργία PulseOut, ο BX-24p μπορεί να παράγει τα απαιτούμενα servo σήματα θέσης. Ανάλογα με το πώς το πρόγραμμα είναι γραμμένο, 10 ή περισσότερα servos μπορούν να τρέξουν από τον BX-24p. Ο κώδικας για ένα servo είναι:

```
Sub Main()
```

```
' This program will move the servo to its middle position.
```

```
Const ServoPin As Byte = 16
```

```
Do
```

```
' Generate a high-going 1.5 ms pulse.
```

```
Call PulseOut(ServoPin, 0.0015, 1)
```

```
' This is to produce a pulse rate of about 50 Hz.
```

```
Call Delay(0.02)
```

```
Loop
```

```
End Sub
```

Προσοχή – εάν χρησιμοποιήσουμε την PulseOut για να παράγουμε σήματα servo, το ρολόι πραγματικού χρόνου μπορεί να χάσει χρόνο εκτός και αν το πλάτος του παλμού είναι μικρότερο από την περίοδο του ρολογιού (περίπου 1.94 ms).



Χρησιμοποιώντας την RCtime για να μετρήσουμε αντίσταση

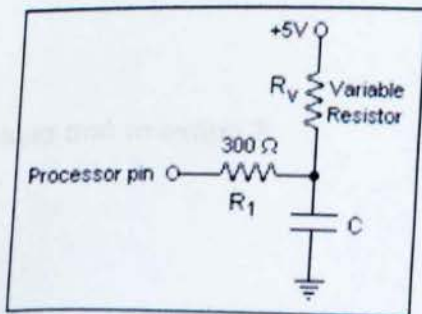
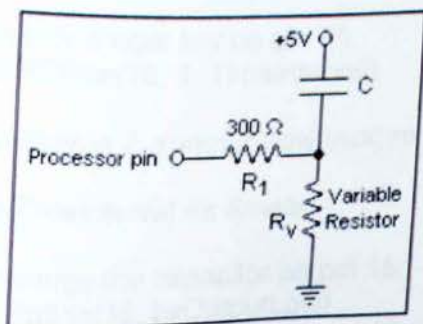
Εισαγωγή

Μια κοινή χρήση των I/O pins είναι να μετράνε αναλογικές τιμές μιας μεταβλητής αντίστασης. Αν και ένα ενσωματωμένος ADC (αναλογικός σε ψηφιακός μετατροπέας) είναι πιθανόν ο ευκολότερος τρόπος να το πετύχουμε αυτό, είναι επίσης πιθανό να χρησιμοποιήσουμε ένα I/O pin. Αυτό μπορεί αν είναι χρήσιμο σε περιπτώσεις που δεν έχουμε αρκετά ADC κανάλια ή αν ένας συγκεκριμένος επεξεργαστής δεν έχει ADC δυνατότητα.

Διαδικασία RCtime

Ο BX-24p προσφέρει μια ειδική διαδικασία που ονομάζεται RCtime για αυτό τον σκοπό. Η RCtime μετράει τον χρόνο που απαιτείται για ένα pin για να αλλάξει την κατάσταση του σε μια προκαθορισμένη τιμή. Συνδέοντας ένα συγκεκριμένο πυκνωτή και μια μεταβλητή αντίσταση σε ένα κύκλωμα RC μπορούμε να χρησιμοποιήσουμε ένα I/O pin για να μετρήσουμε την τιμή της μεταβλητής αντίστασης, η οποία μπορεί αν είναι μια συσκευή όπως ένα ποτενσιόμετρο.

Υπάρχουν δύο συνήθεις τρόποι για να συνδέσεις ένα RCtime σύστημα. Ο πρώτος είναι να δέσεις την μεταβλητή αντίσταση στην γείωση. Το σχήμα 1 δείχνει αυτή τη σύνδεση. Το πλεονέκτημα είναι ότι υπάρχει μικρότερος κίνδυνος σφάλματος από στατικό ηλεκτρισμό:



Σχήματα 1 και 2

Σύνδεση RCtime συστήματος στον BX-24p

Η δεύτερη σύνδεση φαίνεται στο σχήμα 2. Εδώ χρησιμοποιούμε την αντίθετη σύνδεση, όπου ο πυκνωτής C είναι δεμένος στην γείωση και η μεταβλητή αντίσταση RV είναι δεμένη στα 5 volts:

Και στα δύο κυκλώματα η αντίσταση $R1$ είναι εκεί για να προστατεύσει τον BX-24p από μεγάλη ένταση ρεύματος όταν φορτίζεται ο πυκνωτής.

Για να πάρουμε ένα δείγμα, ο πυκνωτής πρώτα αποφορτίζεται παίρνοντας το pin στην σωστή κατάσταση. Στην περίπτωση του σχήματος 1, το pin χρειάζεται να πάει ψηλά (+5 V) για να παράγει 0 volts διαμέσου του πυκνωτή, το οποίο και προκαλεί την αποφόρτιση. Στο σχήμα 2 η επιθυμητή κατάσταση είναι χαμηλό ή γείωση. Σε κάθε περίπτωση, ο πυκνωτής αποφορτίζεται για τουλάχιστον τέσσερις χρονικές σταθερές, οι οποίες είναι $4 R1C$. Μόλις αυτός ο χρόνος περάσει, ο πυκνωτής πρέπει να είναι αποφορτισμένος επιτυχώς.

Σε αυτό το σημείο, η RCtime διαδικασία καλείται. Η RCtime απελευθερώνει τον πυκνωτή κάνοντας το pin μια tristate είσοδο. Η διαδικασία τότε μετράει πόσο χρόνο παίρνει για τον πυκνωτή να φορτιστεί στο trip point του pin εισόδου. Αυτός ο χρόνος διακοπής είναι μια συνάρτηση του RV . Εάν η $R1$ είναι αμελητέα σε σύγκριση με την RV , ο χρόνος διακοπής είναι ανάλογος με την RV .

Ίσως χρειαστεί να πειραματιστούμε με διάφορες τιμές πυκνωτών για να βελτιστοποιήσουμε την διαδικασία.

Παράδειγμα 1, χρησιμοποιώντας το κύκλωμα από το σχήμα 1:

Dim TimeInterval As Single

```
' Discharge the capacitor on pin 15.  
Call PutPin(15, bxOutputHigh)  
Call Delay(20.0E-3) ' Wait about 20 ms.
```

```
' Wait for a logic low on pin 15.  
Call Rctime(15, 1, TimeInterval)
```

Παράδειγμα 2, χρησιμοποιώντας το κύκλωμα από το σχήμα 2:

Dim TimeInterval As Single

```
' Discharge the capacitor on pin 15.  
Call PutPin(15, bxOutputLow)  
Call Delay(20.0E-3) ' Wait about 20 ms.
```

```
' Wait for a logic high on pin 15.  
Call Rctime(15, 0, TimeInterval)
```

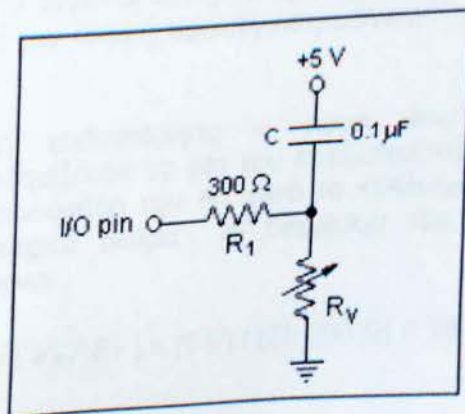

Παράδειγμα ποτενσιόμετρου

Σύμβολα σε αυτό το τμήμα

τ	Time constant
C	Capacitance
I	Current through pot and capacitor
I_0	Initial current through component
R_1	Resistance of fixed resistor
R_V	Resistance of pot
t	Time
V_0	Initial voltage across component
V_c	Voltage across capacitor
V_{R1}	Voltage across fixed resistor R_1
V_{RV}	Voltage across pot
V_{trip}	Voltage trip point for the input pin

Συμπεριφορά κυκλώματος

Σε αυτό το παράδειγμα, θα χρησιμοποιήσουμε τις ίδιες ρυθμίσεις όπως στο σχήμα 1. Ένα ποτενσιόμετρο χρησιμοποιείται για την μεταβλητή αντίσταση R_V και έχει μέγιστη αντίσταση $50\text{ k}\Omega$ (σχήμα 3 κάτω):



Σχήμα 3

Η ακόλουθη παραγωγή υποθέτει ότι η R_V είναι μεγάλη σε σύγκριση με την R_1 . Δηλαδή, $R_1 / R_V \ll 1$.

Αποφορτίζοντας τον πυκνωτή – το πρώτο βήμα είναι να αποφορτίσουμε τον πυκνωτή ανυψώνοντας την I/O pin. Το pin χρειάζεται να κρατηθεί πατημένο αρκετά ώστε να επιτρέψει στην V_c να εξασθενήσει σε μια αμελητέα τιμή.

Για αρχικές συνθήκες, υποθέτουμε ότι η τάση στα άκρα του πυκνωτή έχει φτάσει ένα steady-state 5 V. Επίσης υποθέτουμε ότι το pin αντιδρά σαν ένα ιδανικό διακόπτη. Στον χρόνο $t = 0$, το pin είναι σε υψηλό επίπεδο και το αρχικό ρεύμα στην $R1$ είναι

$$I_0 = (V_0 / R_1) = (5 \text{ V}) / (300 \Omega) = 16.7 \text{ mA}$$

Η τάση V_C στα άκρα του πυκνωτή στο $t > 0$ είναι

$$V_C = \frac{I_0 R_1}{e^{(t/\tau)}}$$

Εδώ, η χρονική σταθερά $\tau = (300 \Omega) (0.1 \mu\text{F}) = 30.0 \mu\text{s}$. Ένας κανόνας είναι να αποφορτιστεί ο πυκνωτής τουλάχιστον τέσσερις φορές ($120.0 \mu\text{s}$, σε αυτή την περίπτωση), το οποίο σημαίνει ότι η τάση θα εξασθενήσει όπως:

$$V_{R1} = \frac{V_0}{e^{(4\tau/\tau)}}$$

$$= (5 \text{ V}) / e^4 = 91.6 \text{ mV} \approx 1.8 \% \text{ of maximum}$$

Εδώ παραμελούμε το γεγονός ότι η επίδραση της R_V στην αποφόρτιση του πυκνωτή. Αυτή είναι μια λογική προσέγγιση μόνο αν η $R1$ είναι αμελητέα σε σύγκριση με την R_V .

Μετρώντας χρονική καθυστέρηση – τώρα που ο πυκνωτής είναι αποφορτισμένος, καθορίζουμε το pin του επεξεργαστή σε κατάσταση input-tristate, το οποίο αποσυνδέει την $R1$ από το κύκλωμα. Σε έναν καινούργιο χρόνο $t = 0$, το αρχικό ρεύμα I_0 διαμέσου του πυκνωτή C και του ποτενσιόμετρου R_V είναι

$$I_0 = (V_0 / R_V) = (5 \text{ V}) / (50\,000 \Omega) = 100 \mu\text{A}$$

Στον χρόνο $t > 0$, το ρεύμα είναι

$$I = \frac{I_0}{e^{(t/\tau)}}$$

Η χρονική σταθερά $\tau = R_V C = (50\,000 \Omega) (0.1 \mu\text{F}) = 5.00 \text{ ms}$. Αυτό αναπαριστάει τον χρόνο που χρειάζεται το ρεύμα να πέσει διαμέσου του κυκλώματος για $(1 - 1/e)$ ή περίπου 63 % κάτω από την μέγιστη τιμή.

Η τάση VRV στα άκρα του ποτενσιόμετρου στον χρόνο $t > 0$ είναι

$$V_{RV} = \frac{I_0 R_Y}{e^{(t/\tau)}}$$

Τώρα πρέπει να γνωρίζουμε το trip point V_{trip} για το pin εισόδου, το οποίο είναι η τάση για την οποία το pin πηγαίνει από χαμηλό λογικό σε υψηλό λογικό. Το σημείο αυτό μπορεί να διαφέρει από pin σε pin και χρειάζεται να καθοριστεί εμπειρικά.

Εάν γνωρίζουμε την V_{trip} , η αντίσταση του ποτενσιόμετρου γίνεται

$$R_Y = \frac{t}{C \ln(V_0 / V_{trip})}$$

Για το Atmel chip που χρησιμοποιείται στον BX-24p, V_{trip} είναι στα 2.5 V και δεν φαίνεται να επηρεάζεται από αλλαγές στην θερμοκρασία. Υποθέτοντας $V_0 = 5$ V και $V_{trip} = 2.5$ V, η αντίσταση είναι

$$R_Y = 1.4427 (t / C)$$

Με άλλα λόγια, για μια σταθερή χωρητικότητα, η αντίσταση είναι ανάλογη σε σχέση με τον χρόνο.

Σε αυτό το παράδειγμα, ο χρόνος που χρειάζεται ο πυκνωτής για να φορτιστεί στο $V_{trip} = 2.5$ V είναι

$$t = (0.1 \mu\text{F}) (50 \text{ k}\Omega) / 1.4427 = 3.47 \text{ ms.}$$

Βελτιστοποίηση – σε αυτή την συγκεκριμένη εφαρμογή, υπάρχει μια ανταλλαγή μεταξύ ακρίβειας και χρόνου απαιτούμενου για τις μετρήσεις. Η RCtime έχει μια Υψηλότερη ακρίβεια απαιτεί μεγαλύτερο χρόνο μέτρησης. Η RCtime έχει μια πλήρη κλίμακα τιμής περίπου 71.1 ms και μια ανάλυση περίπου 1.085 μs (1 / 65 535 σε πλήρη κλίμακα), το οποίο πρέπει να παίρνετε υπόψη.

Για μια δοσμένη αντίσταση, μόλις καθορίσουμε τον βέλτιστο trip time t , έτσι καθορίζουμε τον μέγεθος του πυκνωτή:

$$C = \frac{t}{R_Y \ln(V_0 / V_{trip})}$$

Χαμηλές τιμές της R_V – οι πηγές σε αυτό το τμήμα υποθέτουν ότι η R_1 είναι αμελητέα σε σύγκριση με την R_V . Εάν δεν είναι αυτή η περίπτωση, οι εξισώσεις είναι πιο περίπλοκες. Συγκεκριμένα, καθώς η σχέση R_1 / R_V αυξάνεται, γίνεται λιγότερο έγκυρο να υποθέσουμε ότι ο πυκνωτής είναι πλήρως φορτισμένος στην αρχή κάθε μέτρησης. Στην οριακή περίπτωση όπου $R_V = 0$, ο πυκνωτής δεν αποφορτίζεται καθόλου.

Παράδειγμα κώδικα

```
Sub GetResistance( _  
ByRef Resistance As Single)
```

```
' Measures the resistance of a pot attached to an I/O pin.
```

```
Const PotPin As Byte = 15
```

```
Const DischargeTime As Single = 120.0E-6 ' Seconds
```

```
Const Capacitance As Single = 0.1E-6 ' Farads
```

```
Const TripVoltage As Single = 2.5 ' Volts
```

```
Const InitialVoltage As Single = 5.0 ' Volts
```

```
Dim TimeInterval As Single, K As Single
```

```
' Raise the pin and discharge the capacitor.
```

```
Call PutPin(PotPin, bxOutputHigh)
```

```
Call Delay(DischargeTime)
```

```
' Set the pin to input-tristate, then measure how long the pin stays  
' at logic-high.
```

```
Call Rctime(PotPin, 1, TimeInterval)
```

```
K = 1.0 / (Capacitance * Log(InitialVoltage / TripVoltage))
```

```
Resistance = TimeInterval * K
```

```
End Sub
```

Το παράδειγμα αυτό χρησιμοποιεί το κύκλωμα του σχήματος 3.

Βιβλιογραφία

BX-24p και εναλλακτικές εφαρμογές

Manual του BX-24p της NetMedia Corp. για Hardware Reference και system library.

LM-35

Manual του LM-35 της National Semiconductor για τιμές και τεχνικά χαρακτηριστικά.

Πυρόσβεση, αισθητήρια, EEPROM

Πηγή Wikipedia.

ΒΙΒΛΙΟΘΗΚΗ
ΤΕΙ ΠΕΙΡΑΙΑ