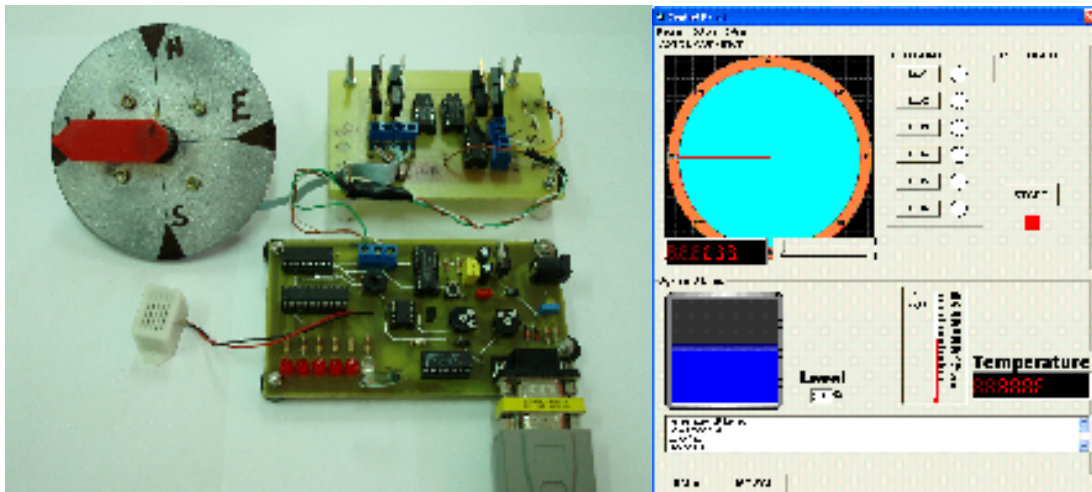




ΤΕΙ ΚΡΗΤΗΣ – ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

# Σχεδίαση και κατασκευή συστήματος επιτήρησης και ελέγχου βιομηχανικού εξοπλισμού μέσω Η/Υ.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ



Κασιμάτη Κωνσταντίνου

Επιβλέπων : Δρ. Μηχ. Νικόλαος Σ. Πετράκης  
Καθηγητής Εφαρμογών

Χανιά,  
Σεπτέμβρης 2008

## Περίληψη

Ο στόχος της παρούσας πτυχιακής εργασίας ήταν η μελέτη συστημάτων επιτήρησης και ελέγχου με την χρήση Η/Υ. Για τον λόγο αυτό σχεδιάστηκαν και κατασκευάστηκαν δύο κυκλώματα. Το πρώτο κομμάτι της κατασκευής είναι μια λειτουργική κάρτα η οποία θα δέχεται και θα στέλνει δεδομένα από και προς τον ηλεκτρονικό υπολογιστή. Η κατασκευή αποτελείται από καταχωρητές, αισθητήρα θερμοκρασίας, αναλογικό σε ψηφιακό μετατροπέα (A/D converter) καθώς και διάφορα άλλα ψηφιακά κυκλώματα με λογικές πύλες. Η δεύτερη κατασκευή είναι ένα κύκλωμα οδήγησης βηματικού κινητήρα πάλι με ολοκληρωμένα κυκλώματα λογικών πυλών. Το κύκλωμα αυτό συνδέεται με τη λειτουργική κάρτα και δέχεται σήματα από αυτήν και έτσι θα μπορεί να ελέγχεται ο βηματικός κινητήρας ως προς τη ταχύτητα του, τη θέση του και τη φορά περιστροφής του. Άρα λοιπόν θα ελέγχεται μέσω του ηλεκτρονικού υπολογιστή. Η σύνδεση της λειτουργικής κάρτας θα γίνεται μέσω της σειριακής θύρας του υπολογιστή. Φυσικά για να υπάρξει επικοινωνία της κατασκευής μας με τον ηλεκτρονικό υπολογιστή θα πρέπει να υπάρξει και το κατάλληλο λογισμικό. Μια φόρμα η οποία θα είναι εύκολη στη χρήση και μέσω αυτής θα μπορούμε να ελέγξουμε τις συσκευές οι οποίες θα είναι συνδεδεμένες πάνω στη λειτουργική κάρτα. Εκτός από ενεργοποίηση των συσκευών που θα γίνεται μέσω ηλεκτρονικού υπολογιστή θα έχουμε και άλλες ενέργειες όπως μέτρηση θερμοκρασίας και έλεγχος φωτεινότητας ενός χώρου με τη βοήθεια φωτοκύτταρου. Όλα αυτά θα απεικονίζονται στη φόρμα και ο χρήστης θα τα επεξεργάζεται μέσω του ηλεκτρονικού υπολογιστή. Το λογισμικό λοιπόν αυτό κατασκευάστηκε με τη βοήθεια της γλώσσας Visual Basic. Μια γλώσσα η οποία είναι εύκολη στη δημιουργία φόρμας με πολλά και διαφορετικά εργαλεία. Πίσω απ' όλα αυτά τρέχει ο απαραίτητος κώδικας.

## Abstract

This dissertation focuses on the study and the construction of the digital circuits. The first part of the construction is a development board witch will receive and send data from and to the computer. The construction consists of registers, temperature sensor, A/D converter and as well as various other digital circuits with logic gates. The second construction is a stepper motor driving circuit, also with integrated circuits with logic gates. This circuit connects with the development board and receives signals from it, so in this way stepper motor will be controlled as far as its speed, position and rotation. Consequently it will be controlled through the computer. The connection of the development board will be occurred through the serial port. In order to take place a communication between our construction and the computer an appropriate software is required. A form, witch will be easy to use and though it will be able to control the devises witch will be connected to the development board. Except for the activation of the devises, witch will be occur through the computer, we will have other activities such as the calculation of the temperature and the control of the brightness of a place with the help of a photocell. All these will be shown on the form and the user will work on them through the computer. This software was created with the help of the Visual Basic language. A Language witch is easy for the creation of a form with various different tools. Behind all these, the necessary code is running.

## 1. Εισαγωγή

Η παρούσα πτυχιακή εργασία η οποία εκπονήθηκε για την ολοκλήρωση των σπουδών μου στο Τμήμα Ηλεκτρονικής του Τ.Ε.Ι. Κρήτης συνίσταται από δύο διακριτά τμήματα, τόσο σχεδίαση και κατασκευή κυκλωμάτων, όσο και ανάπτυξη προγράμματος σε γλώσσα υψηλού επιπέδου προγραμματισμού. Το πρώτο της τμήμα ήταν κατασκευή πλακέτας με διάφορους τύπους ολοκληρωμένων κυκλωμάτων που θα αναλυθούν κατά τη διάρκεια της πτυχιακής εργασίας λεπτομερώς. Το δεύτερο κομμάτι της εργασίας είναι προγραμματισμός με χρήση της γλώσσας προγραμματισμού Visual Basic (από το πακέτο Visual Studio V6.0). Αυτό το κομμάτι είναι απαραίτητο για την επικοινωνία της πλακέτας με τον ηλεκτρονικό υπολογιστή. Άλλωστε το νόημα της πτυχιακής εργασίας είναι η μεταφορά δεδομένων από τον ηλεκτρονικό υπολογιστή προς τη λειτουργική κάρτα όπως την ονομάζουμε και αντίστροφα. Θύρα επικοινωνίας του υπολογιστή επιλέχτηκε η σειριακή θύρα.

Σαν σύνολο της υλικού (hardware) και του λογισμικού (software) έχουμε μια κατασκευή η οποία μπορεί να επικοινωνεί με τον ηλεκτρονικό υπολογιστή όπως είπαμε προηγουμένως με τη χρήση της σειριακής θύρας. Έχοντας σε εκτέλεση τη διεπαφή (interface) και συνδεδεμένη τη κατασκευή στη σειριακή θύρα (μαζί με τη τάση τροφοδοσίας των πλακετών) του υπολογιστή τότε μπορούμε να ελέγξουμε κάποια bit εισόδου και κάποια bit εξόδου. Σαν εισόδους έχουμε ένα bit για την εισαγωγή σειριακά της θερμοκρασίας και άλλο ένα bit για εισαγωγή της καταστάσεως του φωτοκύτταρου. Ως εξόδους έχουμε έξι bit για έλεγχο των έξι led και δυο bit εξόδου για τον έλεγχο των βημάτων και της κατεύθυνσης του βηματικού κινητήρα αντίστοιχα. Άλλο ένα τελευταίο bit εξόδου χρησιμοποιείται για την ενεργοποίηση ενός βομβητή (buzzer). Αντί για led στη θέση τους θα μπορούμε να έχουμε τοποθετήσει κάποια ρελέ και με αυτά να γίνεται ενεργοποίηση φωτισμού ή και ακόμη μιας ολόκληρης μονάδας σε ένα βιομηχανικό χώρο. Ένα τελευταίο bit εξόδου (το οποίο ελέγχεται και αυτό από τη διεπαφή) χρησιμοποιείται για το χρονοισμό των ψηφιακών κυκλωμάτων (σήμα ρολογιού). Αυτή είναι σε γενικές γραμμές η ιδιότητα της κατασκευής της πτυχιακής εργασίας. Παρακάτω ακολουθεί μια μικρή περιγραφή του τι περιέχει κάθε κεφάλαιο. Η πτυχιακή εργασία αποτελείται από άλλα πέντε βασικά κεφάλαιο. Κάθε κεφάλαιο έχει γραφτεί για διαφορετικό κομμάτι της εργασίας.

Στο επόμενο (αριθμημένο ως δεύτερο) κεφάλαιο έχουμε τη περιγραφή της λειτουργικής κάρτας. Συνοδεύεται με αναλυτική περιγραφή κάθε εξαρτήματος μαζί με τις ιδιότητές του και τα τεχνικά χαρακτηριστικά του. Η λειτουργική κάρτα αποτελείται από διάφορων τύπων καταχωρητές, καθώς και διάφορες συσκευές που επικοινωνούν με τη λειτουργική κάρτα όπως αισθητήρα θερμοκρασίας για ψηφιακή μέτρηση της θερμοκρασίας, καθώς και διάφορα άλλα ψηφιακά κυκλώματα με διάφορους τύπους πυλών ούτως ώστε να κάνουμε τις επιθυμητές εργασίες π.χ. χρονοισμό των υπόλοιπων κυκλωμάτων. Ανεξάρτητη μονάδα είναι και η κατασκευή φωτοκύτταρου για έλεγχο της φωτεινότητας ενός χώρου. Αυτά είναι τα βασικά τμήματα της λειτουργικής κάρτας. Επίσης υπάρχουν και άλλα συνηθισμένα εξαρτήματα όπως πυκνωτές για διάφορες εργασίες όπως εξομάλυνση του συνεχούς ρεύματος κ.α. Ακόμη δίοδοι Zener για προστασία σε ανάστροφες τάσεις.

Το τρίτο κεφάλαιο ξεκινά με μια θεωρητική αναφορά πάνω στους βηματικούς κινητήρες. Γίνεται αναλυτική επεξήγηση πάνω στο τρόπο λειτουργίας τους καθώς και μελέτη των διάφορων τύπων βηματικών κινητήρων. Κάθε τύπος βηματικού κινητήρα χρειάζεται διαφορετικό κύκλωμα οδήγησης όπως θα δούμε. Το κεφάλαιο τελειώνει με τη περιγραφή κατασκευής κυκλώματος οδήγησης βηματικού κινητήρα. Ο βηματικός κινητήρας που επιλέχτηκε ήταν διπολικός και φτιάχτηκε κύκλωμα οδήγησης για αυτόν. Το κύκλωμα αυτό φυσικά σχεδιάστηκε για να μπορεί να συνδεθεί με τη λειτουργική κάρτα και δέχεται τον

έλεγχου μέσω του ηλεκτρονικού υπολογιστή. Το κύκλωμα όπως θα δούμε αποτελείται κατά βάση από ψηφιακά κυκλώματα που έχουν διάφορες πύλες στο εσωτερικό τους και η οδήγηση του βηματικού κινητήρα γίνεται με χρήση των MOSFET.

Το τέταρτο κεφάλαιο γράφτηκε για τη περιγραφή του κομματιού του προγραμματισμού. Με τη γλώσσα Visual Basic σχεδιάσαμε την απαραίτητη διεπαφή (interface) η οποία έχει κάποια κουμπιά ελέγχου. Με όλα αυτά τα κουμπιά μπορούμε να ελέγχουμε κάθε τι πάνω στη λειτουργική κάρτα. Οτιδήποτε δεδομένο εισόδου – εξόδου θα μπορεί να ελέγχεται από τη διεπαφή. Για κάθε κουμπί ελέγχου υπάρχει ο κώδικας ο οποίος εκτελείται και με αυτόν θα γίνεται μεταφορά δεδομένων από τον ηλεκτρονικό υπολογιστή προς τη λειτουργική κάρτα και το αντίστροφο. Στο κεφάλαιο αυτό γίνεται αναλυτική περιγραφή του κώδικα. Δηλαδή κάθε γραμμή κώδικα γίνεται επεξήγηση του τι γίνεται κατά την εκτέλεσή της. Έτσι κάποιος ο οποίος δε γνωρίζει καθόλου από τη γλώσσα προγραμματισμού Visual Basic θα μπορέσει να καταλάβει ακριβώς το τι κάνει το πρόγραμμα και με ποια λογική επιλέχθηκαν οι εντολές. Τέλος η φόρμα που φτιάχτηκε είναι απλή και εύκολη στη χρήση για οποιονδήποτε.

Το πέμπτο κεφάλαιο γράφτηκε για τη χρήση της κατασκευής αυτής. Δηλαδή όλα τα βήματα που πρέπει να κάνει κάποιος ο οποίος δεν έχει γνώση για τη συγκεκριμένη κατασκευή ούτως ώστε να μπορεί να θέσει σε λειτουργία τη κατασκευή σε οποιονδήποτε υπολογιστή. Εργασίες όπως εγκατάσταση του απαραίτητου λογισμικού στον ηλεκτρονικό υπολογιστή κ.α. Με αυτά τελειώνει ο κορμός της πτυχιακής εργασίας αφήνοντας και δυνατότητες εξέλιξης της κατασκευής για περισσότερες λειτουργίες ελέγχου. Επιπρόσθετες λειτουργίες όπως εισαγωγή στο υπολογιστή περισσότερων δεδομένων. Για παράδειγμα όπως παρακάτω θα δούμε τον τρόπο με το οποίο γίνεται η μέτρηση της θερμοκρασίας με χρήση μεταβαλλόμενης τάσης έτσι αναλόγως θα μπορούσαμε να μετρήσουμε και τη στάθμη υγρών μιας δεξαμενής.

Στο έκτο και τελευταίο κεφάλαιο έχουμε τη συγκέντρωση όλων των συμπερασμάτων. Δηλαδή συμπεράσματα που βγήκαν κατά τη διάρκεια της εκτέλεσης της πτυχιακής εργασίας. Κάθε πείραμα μέχρι τελικής κατασκευής της λειτουργικής κάρτας είχε ως αποτέλεσμα τη συγκέντρωση συμπερασμάτων για τη συμπεριφορά τόσο των ψηφιακών κυκλωμάτων όσο και των αναλογικών κυκλωμάτων. Τα συμπεράσματα λοιπόν είναι το σύνολο των γνώσεων που αποκομίστηκαν κατά τη διάρκεια της πτυχιακής εργασίας.

Στο τέλος όλου του εγγράφου της πτυχιακής εργασίας υπάρχει η αντίστοιχη βιβλιογραφία. Όλο το υλικό τόσο για το θεωρητικό κομμάτι των βηματικών κινητήρων όσο και για το πρακτικό κομμάτι που ήταν η κατασκευή των πλακετών χρησιμοποιήθηκαν βιβλία και site από το Διαδίκτυο. Από το διαδίκτυο έγινε εύρεση μέσω μηχανών αναζήτησης κομμάτια θεωρίας που χρειάστηκαν καθώς ακόμη και οτιδήποτε στοιχεία (datasheet) πάνω σε ψηφιακά κυκλώματα και διάφορα ηλεκτρονικά κυκλώματα. Τέλος πληροφορίες για την εγγραφή του απαραίτητου κώδικα που συγκροτεί τη διεπαφή. Τα βιβλία ήταν χρήσιμα για θεωρία πάνω σε ψηφιακά κυκλώματα, η οποία θεωρία ήταν απαραίτητη για τη σύνθεση των εξαρτημάτων της λειτουργικής κάρτας. Τέλος ένα βιβλίο χρειάστηκε για την εκμάθηση της γλώσσας Visual Basic που χρησιμοποιήθηκε για τη κατασκευή της διεπαφής.

## 2. Σχεδίαση Κάρτας – Πλήρης ανάλυση κάθε βαθμίδας

Η λειτουργική αυτή κάρτα κατασκευάστηκε για να ελέγχουμε τη λειτουργία συσκευών από μακριά και με τη χρήση του ηλεκτρονικού υπολογιστή. Μέσω της διεπαφής που κατασκευάστηκε μπορούμε να στείλουμε και να λάβουμε δεδομένα στη λειτουργική κάρτα. Όπως για παράδειγμα, θέλουμε να ενεργοποιήσουμε ένα φωτιστικό θα μπορούμε να το ανάψουμε μέσω ηλεκτρονικού υπολογιστή. Ακόμη θέλουμε να στρέφουμε μια κάμερα καταγραφής σε άλλη κατεύθυνση θα μπορούμε πάλι μέσω υπολογιστή. Τέτοιες αυτοματοποιημένες λειτουργίες και διάφορες άλλες τέτοιου τύπου θα μπορούν να γίνουν πράξη μέσω της λειτουργικής αυτής κάρτας.

### 2.1. Γενικά περί της κάρτας

Η λειτουργική αυτή κάρτα αποτελείται από βαθμίδες από τις οποίες κάθε μια προορίζεται για διαφορετική λειτουργία. Το πρώτο κομμάτι της κάρτας είναι το κομμάτι το οποίο είναι υπεύθυνο για τη σταθεροποίηση της τάσης στα 5 Volt. Από την έξοδο αυτής της βαθμίδας τροφοδοτούμε όλες τις επιμέρους βαθμίδες με σταθερή DC τάση στα 5 Volt. Το ολοκληρωμένο που χρησιμοποιούμε είναι το LM7805 το οποίο όπως είπαμε σταθεροποιεί τη τάση στα + 5 Volt.

Το επόμενο κομμάτι της κάρτας είναι οι καταχωρητές που χρησιμοποιούνται για την αποθήκευση των δεδομένων. Στη συγκεκριμένη κατασκευή χρησιμοποιούνται δυο ειδών καταχωρητές. Ένας είναι ο 8-bit σειριακής φόρτωσης δεδομένων και παράλληλη έξοδο δεδομένων (TC74HC4094) και ο άλλος είναι παράλληλης φόρτωσης και παράλληλης εξόδου δεδομένων (MC74HC374A).

Μια άλλη βαθμίδα είναι το ολοκληρωμένο M54HC4022 το οποίο είναι ένας οκταδικός μετρητής. Το νόημα αυτού του ολοκληρωμένου είναι να συγχρονίζει τη μεταφορά των δεδομένων. Δηλαδή καθώς τα δεδομένα γεμίζουν το καταχωρητή σειριακής φόρτωσης να εισέρχονται στο καταχωρητή παράλληλης φόρτωσης. Αναλυτικότερα τη λειτουργία του θα τη δούμε στη συνέχεια του κεφαλαίου.

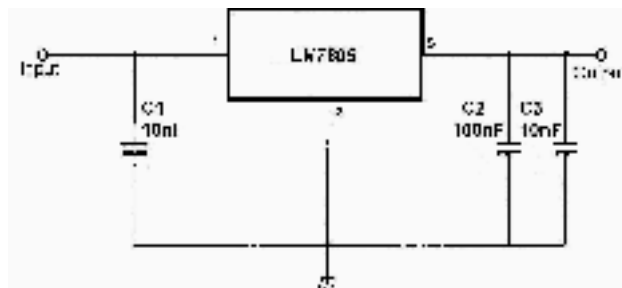
Από την έξοδο του καταχωρητή παράλληλης φόρτωσης μπορούμε να ενεργοποιήσουμε 8 διαφορετικές συσκευές. Στη θέση αυτών εμείς έχουμε βάλει led. Αν θέλουμε να ενεργοποιήσουμε κάποια συσκευή (που απαιτεί μεγαλύτερη ποσότητα ρεύματος) τότε θα χρησιμοποιήσουμε κάποιο ρελέ και με αυτό θα ενεργοποιείται η αντίστοιχη συσκευή. Μια συσκευή από αυτές θα μπορούσε να είναι και μια λάμπα φωτισμού. Όσον αφορά τη λάμπα φωτισμού πάνω στη κάρτα έχουμε μια ακόμη βαθμίδα η οποία είναι υπεύθυνη στο να ελέγχει το αν η λάμπα φωτισμού έχει ανάψει (φυσικά αντί για λάμπα έχουμε τοποθετήσει ένα led ισχύος λευκού φωτισμού). Αυτή η βαθμίδα αποτελείται από μια φωτοαντίσταση συνδεδεμένη σε ένα διαιρέτη τάσης. Ένα σήμα από τη φωτοαντίσταση οδηγείται διαδοχικά σε δυο πύλες NOT Schmitt Trigger (από το ολοκληρωμένο HEF 40106) ούτως ώστε με την ενεργοποίηση του φωτιστικού θα έχουμε μια μεταβολή από λογικό '0' σε λογικό '1' στην έξοδο της δεύτερης NOT και από λογικό '1' σε λογικό '0' σε αντίστροφη διαδικασία. Το σήμα αυτό θα εισέρχεται μέσω σειριακής στον υπολογιστή σαν είσοδος. Έτσι λοιπόν ο Ηλεκτρονικός Υπολογιστής θα αναγνωρίζει ότι το φωτιστικό έχει ανάψει. Ο λόγος που γίνεται αυτός ο έλεγχος είναι ότι μπορεί να μην το σήμα ενεργοποίησης του φωτιστικού να σταλεί μέσω σειριακής αλλά η λάμπα να είναι εκτός λειτουργίας και εάν το φωτιστικό βρίσκεται μακριά από τον ηλεκτρονικό Υπολογιστή πρέπει με κάποιο τρόπο να ελεγχθεί η ενεργοποίησή του.

Η επόμενη βαθμίδα είναι για τη μέτρηση της θερμοκρασίας του περιβάλλοντος και απεικόνιση της τιμής της στον υπολογιστή. Δυο ολοκληρωμένα χρησιμοποιούνται. Το ένα είναι το LM 335 το οποίο είναι αισθητήρας θερμοκρασίας και ανάλογα τη μεταβολή της έχουν και αντίστοιχη τάση στην έξοδο του ολοκληρωμένου. Το δεύτερο ολοκληρωμένο είναι το TLC549IP οποίο είναι μετατροπέας αναλογικού σήματος σε ψηφιακό (analog to digital converter). Λαμβάνει την αντίστοιχη τάση από την έξοδο του LM 335 και την κάνει μια συστοιχία ψηφιακών δεδομένων. Το ψηφιακά δεδομένα από το TLC549IP εισέρχονται σαν είσοδος στον υπολογιστή και επεξεργάζονται από το πρόγραμμα. Κάθε βαθμός Κελσίου αντιστοιχεί σε διαφορετική τάση και επομένως σε αντίστοιχη συστοιχία ψηφιακών δεδομένων.

Τελευταία βαθμίδα η οποία θα αναλυθεί σε επόμενο κεφάλαιο είναι το κύκλωμα οδήγησης βηματικού κινητήρα το οποίο συνδέεται με αυτή τη κάρτα και μπορεί να δημιουργεί διαφορετικές λειτουργίες στο βηματικό κινητήρα. Φυσικά δε θα μπορούσε να παραληφθεί και η βαθμίδα σταθεροποίησης της τάσης τροφοδοσίας η οποία τροφοδοτεί όλα τα ολοκληρωμένα κυκλώματα της κάρτας μας.

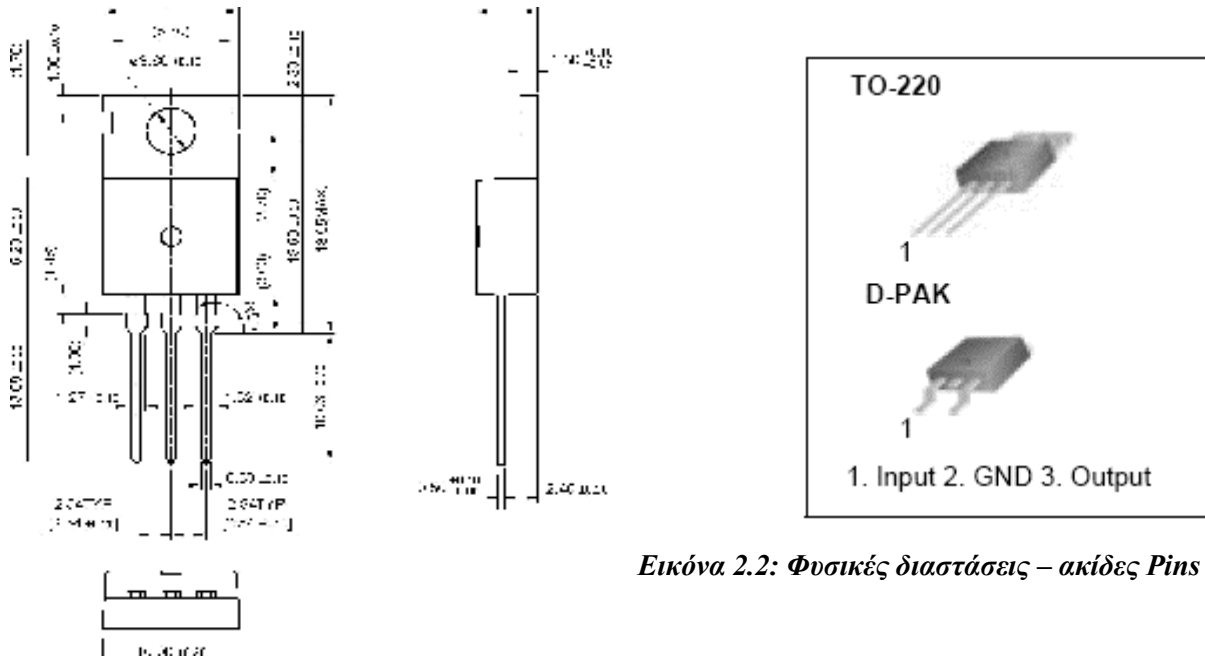
## 2.2. Σταθεροποιητής τάσεως με χρήση του ολοκληρωμένου LM7805

Βασικό στο κύκλωμά μας είναι η σταθεροποίηση της τάσης προς αποφυγήν καταστροφής των ολοκληρωμένων κυκλωμάτων της κάρτας. Επίσης μη σταθεροποιημένη τάση θα μπορούσε να δημιουργήσει και λανθασμένα δεδομένα στην επικοινωνία της κάρτας με τον ηλεκτρονικό υπολογιστή. Έτσι λοιπόν το ολοκληρωμένο LM7805 λαμβάνει σαν είσοδο τη τάση από το τροφοδοτικό (στη συγκεκριμένη κάρτα 12 Volt DC) και τη σταθεροποιεί στα + 5 Volt. Οι πυκνωτές που φαίνονται στην εικόνα 2.1 είναι για πλήρη εξομάλυνση της τάσης μην αφήνοντας κάποια συνιστώσα εναλλασσόμενη (AC) να εμφανιστεί στην έξοδο. Όπως ξέρουμε και θεωρητικά το εναλλασσόμενο ρεύμα προτιμά τη διέλευσή του μέσω του πυκνωτή. Άρα στην έξοδο του σταθεροποιητή θα έχουμε καθαρή συνεχή τάση.



**Εικόνα 2.1: Σταθεροποίηση τάσης με ολοκληρωμένο LM7805**

Στην εικόνα 2.2 βλέπουμε τις φυσικές διαστάσεις του ολοκληρωμένου καθώς και η αρίθμηση των ακίδων (pins) και η αντιστοιχία τους στη κάθε λειτουργία



Εικόνα 2.2: Φυσικές διαστάσεις – ακίδες Pins

Το ολοκληρωμένο LM7805 έχει τα ακόλουθα χαρακτηριστικά:

- Ρεύμα εξόδου έως και 1A
- Τάση εξόδου +5Volt
- Θερμική προστασία υπερφόρτωσης ρεύματος (θερμοκρασία λειτουργίας  $-65 \dots 150^{\circ}\text{C}$ )
- Προστασία για βραχυκυκλώματα
- Τάση εισόδου για σωστή λειτουργία του σταθεροποιητή από 5 Volt έως 20 Volt

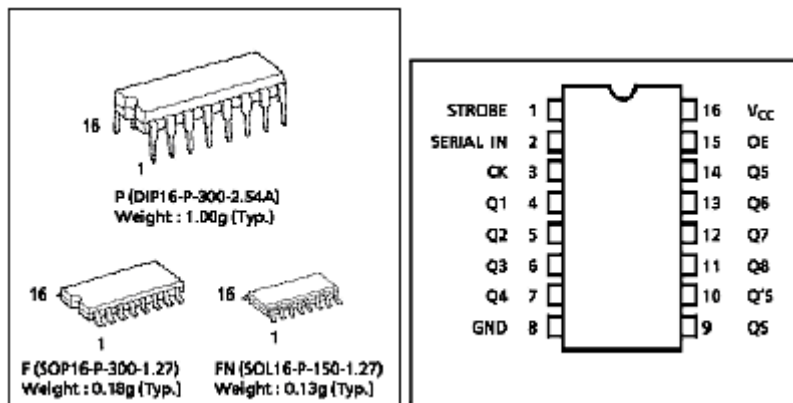
### 2.3. Καταχωρητές για την αποθήκευση των δεδομένων

Στη συγκεκριμένη κατασκευή χρησιμοποιήσαμε δυο τύπους καταχωρητών. Ο πρώτος τύπος είναι το ολοκληρωμένο κύκλωμα TC74HC4094. Ο καταχωρητής αυτός λαμβάνει τα δεδομένα σειριακά και τα εμφανίζει στην έξοδο παράλληλα. Ο δεύτερος τύπος είναι το ολοκληρωμένο κύκλωμα MC74HC374A το οποίο λαμβάνει τα δεδομένα παράλληλα και τα εμφανίζει στην έξοδο επίσης παράλληλα. Αναλυτικότερα θα τα δούμε παρακάτω:

#### 2.3.1. Ολοκληρωμένο κύκλωμα TC74HC4094 καταχωρητής ολίσθησης (Shift register)

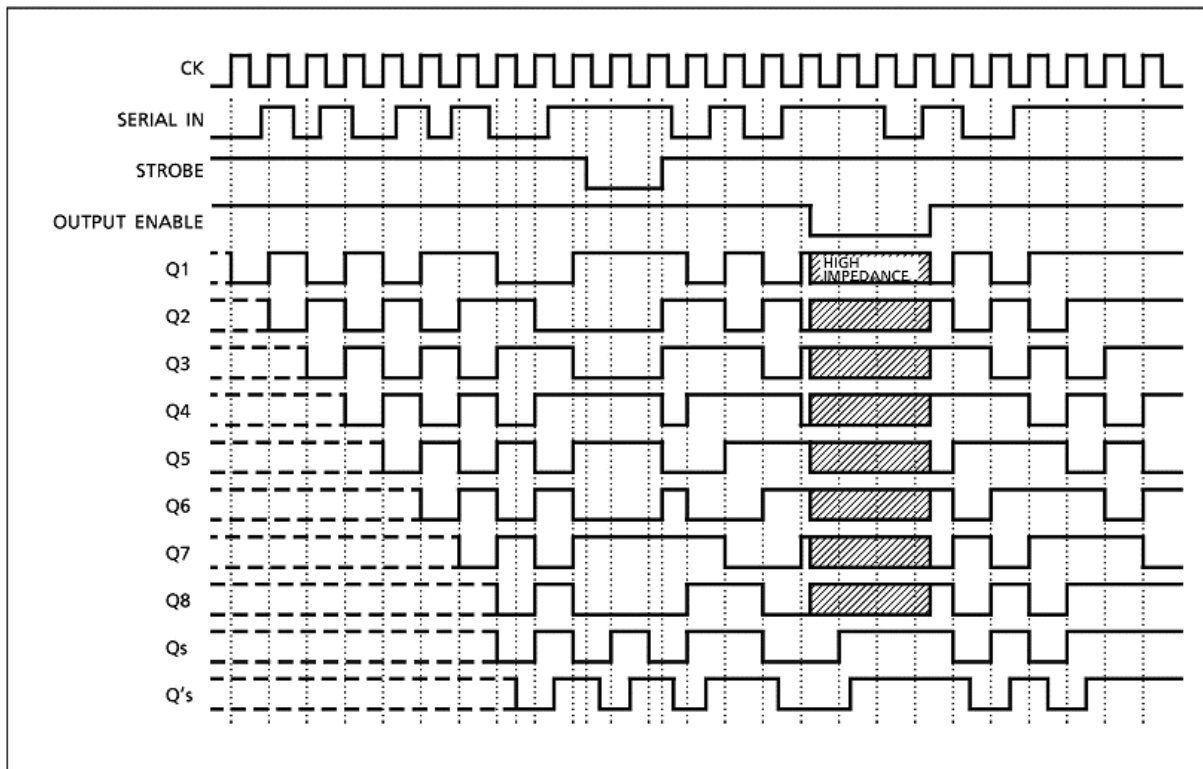
Το συγκεκριμένο ολοκληρωμένο κύκλωμα το χρησιμοποιήσαμε για να μεταφέρει τα δεδομένα από τον υπολογιστή στη λειτουργική κάρτα μέσω της σειριακής θύρας. Μια συστοιχία από 8 bits θέλουμε να μεταφερθεί από τον υπολογιστή στις εξόδους του καταχωρητή. Και τα 8 bits μεταφέρονται μέσω του ενός καλωδίου τις σειριακής σειριακά (το ένα μετά το άλλο) και εισέρχονται διαδοχικά στην είσοδο του ολοκληρωμένου. Μετά από οκτώ διαδοχικές ολισθήσεις τα δεδομένα βγαίνουν στις οκτώ δυαδικές εξόδους (bits) του καταχωρητή. Όπως όλα τα σύγχρονα ακολουθιακά κυκλώματα και το συγκεκριμένο ολοκληρωμένο χρειάζεται χρονισμό για τη μεταφορά των δεδομένων. Έτσι λοιπόν στη Τρίτη ακίδα (pin) συνδέουμε το ρολόι το οποίο είναι κοινό και για τα υπόλοιπα ολοκληρωμένα

κυκλώματα. Τα δεδομένα εισέρχονται στο δεύτερο pin. Στην εικόνα 2.3 βλέπουμε την εικονική περιγραφή της θήκης του ολοκληρωμένου και την επεξήγηση των ακίδων (Pins).



Εικόνα 2.3: θήκη και ακίδες Pins του ολοκληρωμένου 4094 shift register

Κάθε λοιπόν bit των εξόδων του ολοκληρωμένου δέχεται δεδομένα διαδοχικά το ένα μετά το άλλο. Άρα λοιπόν για να στείλουμε μια ολόκληρη οκτάδα δεδομένων στο ολοκληρωμένο θα πρέπει να ενεργήσουν οκτώ παλμοί. Οι ακόλουθη εικόνα δείχνει το διάγραμμα καταστάσεων σε κάθε ακίδα ξεχωριστά



Εικόνα 2.4: διάγραμμα καταστάσεων

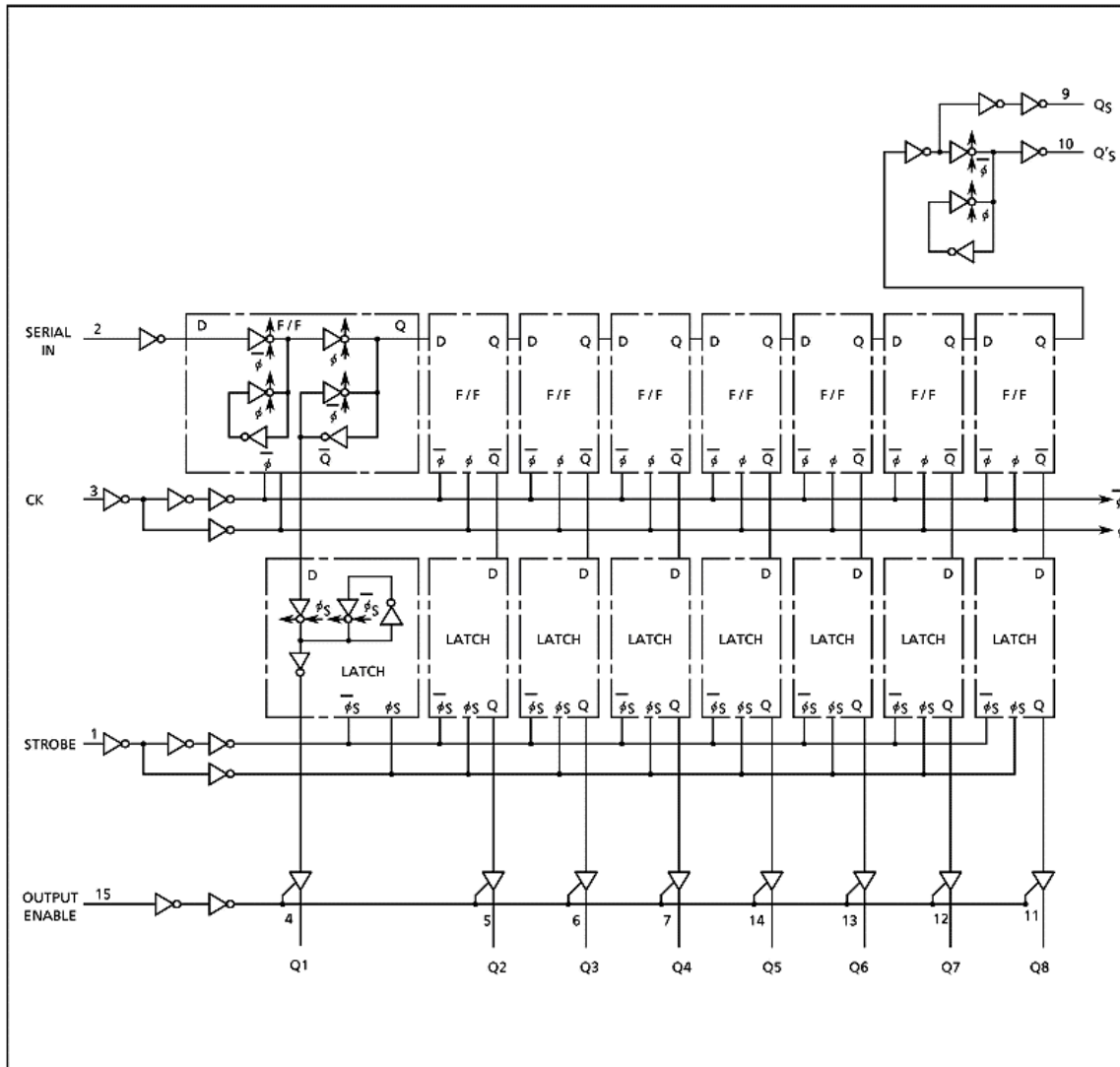
Όπως βλέπουμε τα δεδομένα εισέρχονται σειριακά στο ολοκληρωμένο. Στο πρώτο παλμό του ρολογιού ένα bit δεδομένων από τον υπολογιστή εμφανίζεται στη έξοδο Q1 του ολοκληρωμένου. Στον επόμενο παλμό η κατάσταση του Q1 μεταφέρεται στο Q2 (ολίσθηση)



και το επόμενο bit δεδομένων του υπολογιστή εμφανίζεται πάλι στο Q1. Με αυτό τον τρόπο με το πέρας των 8 παλμών του ρολογιού οι εξόδοι Q1-Q8 θα έχουν λάβει τα δεδομένα από τον υπολογιστή και θα έχουν εμφανιστεί στις εξόδους του ολοκληρωμένου.

Τροφοδοσία στο ολοκληρωμένο έχουμε +5 Volt στο 16<sup>ο</sup> pin και γείωση στο 8<sup>ο</sup> pin. Το σήμα output enable του 15<sup>ου</sup> pin είναι τοποθετημένο σε υψηλό δυναμικό, δηλαδή ενεργό στο "1". Όπως επίσης και το σήμα strobe του 1<sup>ου</sup> pin.

Όπως τα περισσότερα ολοκληρωμένα κυκλώματα έτσι και το 4094 εσωτερικά αποτελείται από ένα σύνολο πυλών. Στην εικόνα 2.5 βλέπουμε αναλυτικά το εσωτερικό του.

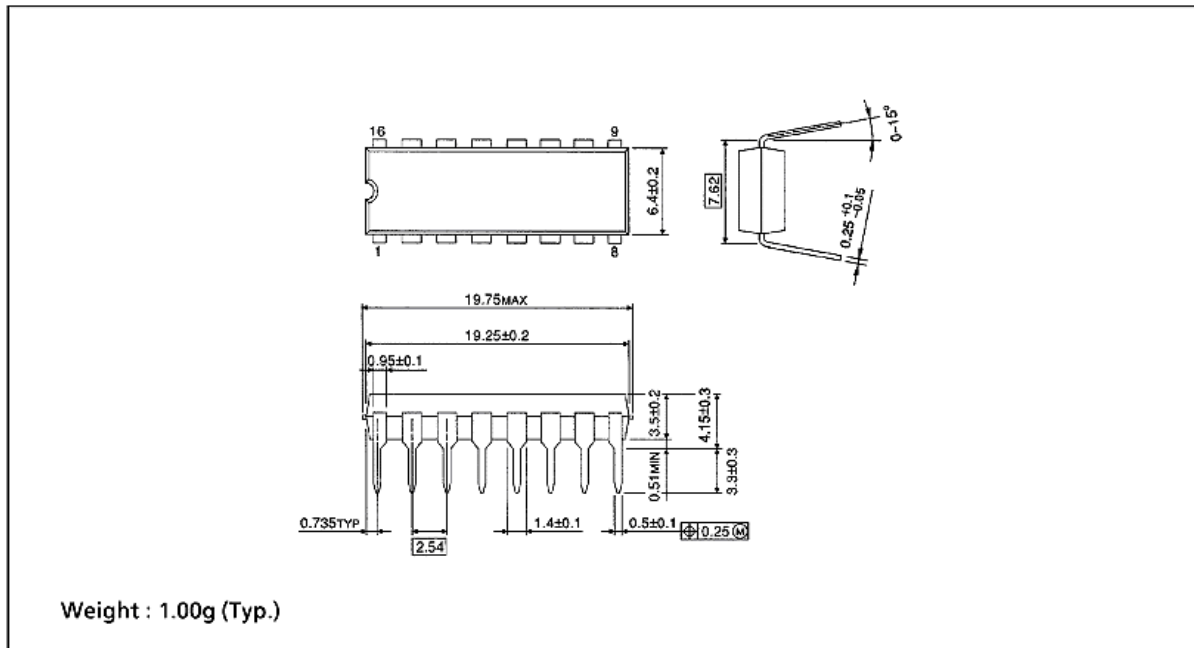


**Εικόνα 2.5: Διάγραμμα πυλών εσωτερικού ολοκληρωμένου 4094**

Το ολοκληρωμένο 4094 έχει τα ακόλουθα χαρακτηριστικά:

- Τάση σωστής λειτουργίας από 2 Volt έως περίπου 6 Volt
- Τάση εισόδων από -0.5 Volt έως  $V_{cc}+0.5\text{Volt}$
- Τάση εξόδων από -0.5 Volt έως  $V_{cc}+0.5\text{Volt}$
- Θερμοκρασία στο εσωτερικό του για σωστή λειτουργία -65 έως 150°C
- Ρεύμα εξόδου 20 mA
- Απώλεια ισχύος 500 mW (για θήκη PDIP) και 180 mW (για θήκη SOP)

Στο σχήμα που ακολουθεί βλέπουμε τις φυσικές διαστάσεις του ολοκληρωμένου 4094. Ο τύπος της θήκης την οποία είχε το ολοκληρωμένο που χρησιμοποιήσαμε είναι PDIP.

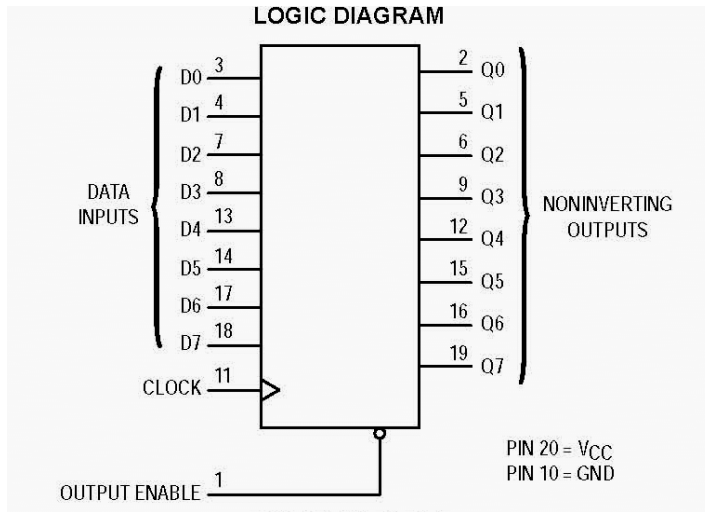


Εικόνα 2.6: Φυσικές διαστάσεις ολοκληρωμένου 4094

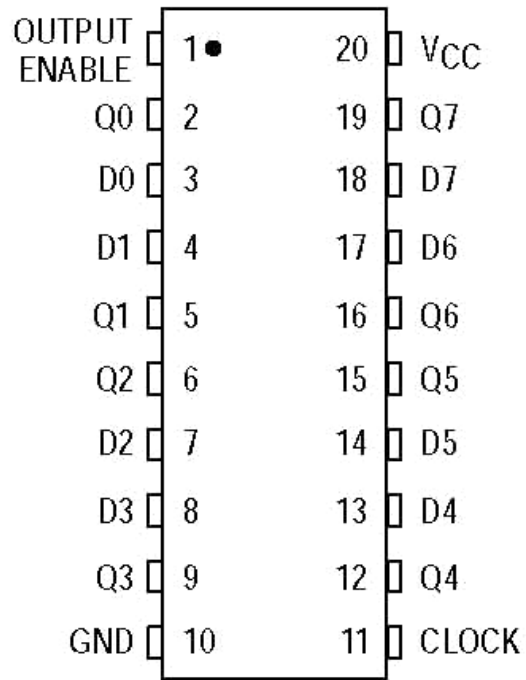
### 2.3.2. Ολοκληρωμένο κύκλωμα MC74HC374A

Το συγκεκριμένο ολοκληρωμένο κύκλωμα έχει χρησιμοποιηθεί σαν τελικό στάδιο των δεδομένων μας. Λαμβάνει τα δεδομένα παράλληλα από τον καταχωρητή που μελετήσαμε προηγουμένως και τα βγάζει στην έξοδο παράλληλα σε μορφή μιας ψηφιακής λέξης των 8-bits. Ο λόγος που χρησιμοποιήθηκε ήταν για να μένουν τα δεδομένα ανεπηρέαστα στις εξόδους. Αιτία είναι ο καταχωρητής που περιγράψαμε προηγουμένως. Καθώς κάνει ολίσθηση δεδομένων η κάθε έξοδος Q1-Q8 αλλάζει τιμή μέχρι η οκταδική λέξη να τοποθετηθεί στο καταχωρητή. Αν εμείς λοιπόν θέλουμε να ενεργοποιήσουμε μόνο ένα φωτισμό για ελάχιστο χρόνο η τιμή που θέλουμε να δώσουμε στην αντίστοιχη έξοδο θα περάσει και από άλλες συσκευές. Έτσι λοιπόν μόλις τα δεδομένα φορτωθούν στο καταχωρητή 4094 δίνεται εντολή να μεταφερθούν παράλληλα στο καταχωρητή MC74HC374A. Και έτσι οι συσκευές όπως κάποιος φωτισμός συνδέεται στις εξόδους του καταχωρητή MC74HC374A και έτσι η κατάσταση τους μένει ανεπηρέαστη από αλλαγή κατάστασης σε διπλανή έξοδο.

Παρακάτω θα δούμε κάποιες εικονικές σχεδιάσεις του ολοκληρωμένου. Αποτελείται από 20 ακίδες (pins), δηλαδή 8 ακίδες εισόδων και 8 εξόδων, μια ακίδα τροφοδοσίας ένα για τη γείωση, μια ακίδα γενικής ενεργοποίησης καθώς και μια ακίδα για το σήμα του ρολογιού (clock).



Σχήμα 2.7: Λογικό διάγραμμα



Σχήμα 2.8: Προσδιορισμός των ακίδων

Ο παρακάτω πίνακας που ακολουθεί μας δίνει το πίνακα αληθείας των εξόδων συναρτήσει των εισόδων, output enable και clock.

Είσοδοι			Έξοδοι
Output Enable	Clock	D	Q
L		H	H
L		L	L
L	L, H,	X	No Change
H	X	X	Z

X = Αδιάφορος όρος

Z = Υψηλή σύνθετη αντίσταση

Πίνακας 2.1: πίνακας αληθείας

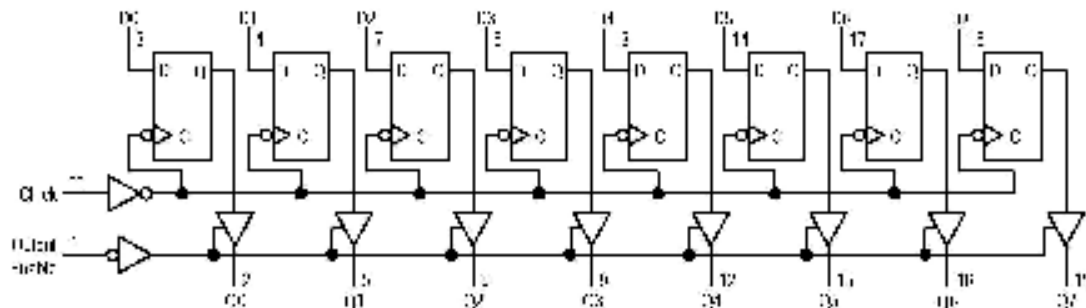
Παρακάτω στον επόμενο πίνακα έχουν συγκεντρωθεί όλα τα απαραίτητα στοιχεία για τη μελέτη του συγκεκριμένου καταχωρητή.

Σύμβολο	Παράμετρος	Τιμή	Μονάδα
V <sub>cc</sub>	DC τάση τροφοδοσίας (σε αναφορά με τη γείωση)	- 0.5 έως + 7.0	Volt
V <sub>in</sub>	DC τάση εισόδου (σε αναφορά με τη γείωση)	- 0.5 έως V <sub>cc</sub> + 0.5	Volt
V <sub>out</sub>	DC τάση εξόδου (σε αναφορά με τη γείωση)	- 0.5 έως V <sub>cc</sub> + 0.5	Volt
I <sub>in</sub>	DC ρεύμα εισόδου ανά pin	± 20	mA
I <sub>out</sub>	DC ρεύμα εξόδου ανά pin	± 35	mA
I <sub>cc</sub>	DC ρεύμα τροφοδοσίας στα pin VCC και GND	± 75	mA
PD	Απώλεια ισχύος	750	mW
T <sub>stg</sub>	Θερμοκρασία αντοχής (Storage Temperature)	- 65 έως + 150	°C

Πίνακας 2.2: Χαρακτηριστικά στοιχεία καταχωρητή

**Σημείωση:** Το συγκεκριμένο κύκλωμα προστατεύεται από το στατικό ηλεκτρισμό καθώς και από επιδράσεις εξωτερικών ηλεκτρικών πεδίων.

Στο επόμενο σχήμα βλέπουμε το λογικό διάγραμμα του εσωτερικού του ολοκληρωμένου κυκλώματος MC74HC374A. Όπως θα παρατηρήσουμε κάθε flip-flop είναι ανεξάρτητο από τα υπόλοιπα. Οπότε εμείς θα μπορούσαμε να χρησιμοποιήσουμε όσα από αυτά θέλουμε. Δηλαδή δεν είναι απαραίτητο να χρησιμοποιηθούν όλα τα flip-flops.



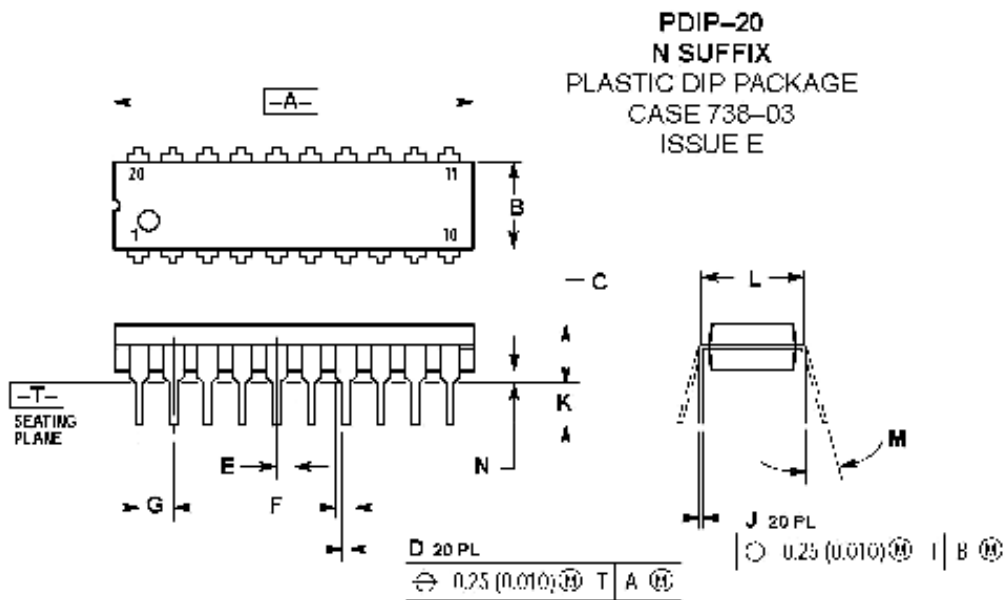
Σχήμα 2.9: Λογικό διάγραμμα

Οι ακίδες (pins) του καταχωρητή έχουν ως εξής: στο pin 10 & 20 συνδέουμε τη γείωση και τη τάση τροφοδοσίας αντίστοιχα. Το output enable (pin 1) το συνδέουμε στη γείωση διότι το ολοκληρωμένο είναι ενεργό όταν το output enable είναι στο μηδέν δηλαδή σε χαμηλό δυναμικό (βλ. πίνακας 2.1). Το σήμα του ρολογιού για την ακίδα 11 θα το πάρουμε από εξωτερικό χρονοισμό λόγω του ότι δεν είναι κοινό με το ρολόι των άλλων ολοκληρωμένων.

Τελειώνοντας με αυτό το κύκλωμα θα δώσουμε παρακάτω το σχέδιο της θήκης του ολοκληρωμένου κυκλώματος με τις φυσικές διαστάσεις του.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	1.010	1.070	25.66	27.17
B	0.240	0.260	6.10	6.60
C	0.150	0.180	3.81	4.57
D	0.015	0.022	0.39	0.55
E	0.050 BSC		1.27 BSC	
F	0.050	0.070	1.27	1.77
G	0.100 BSC		2.54 BSC	
J	0.008	0.015	0.21	0.38
K	0.110	0.140	2.80	3.55
L	0.300 BSC		7.62 BSC	
M	0°	15°	0°	15°
N	0.020	0.040	0.51	1.01

**Πίνακας 2.3:**  
Οι τιμές των διαστάσεων του Σχ. 2.10



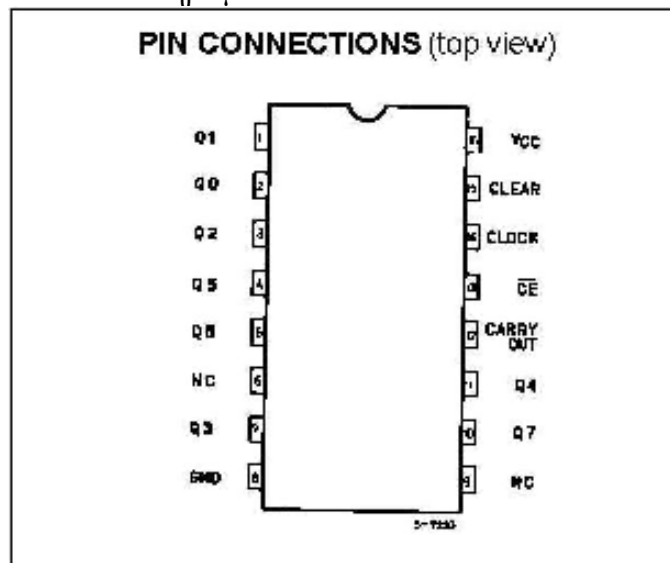
Σχήμα 2.10: Φυσικές διαστάσεις του MC74HC374A

## 2.4. Ολοκληρωμένο κύκλωμα 4022 για χρονοισμό του καταχωρητή MC74HC374A

Όπως είπαμε ο προηγούμενος καταχωρητής παράλληλης φόρτωσης MC74HC374A δέχεται σήμα ρολογιού από εξωτερικό ολοκληρωμένο κύκλωμα. Αυτό είναι ο οκταδικός μετρητής 4022. Ο λόγος είναι διότι το ρολόι του πρέπει να είναι διαφορετικό από τα υπόλοιπα κυκλώματα. Μόλις η οκτάδα bits εισαχθούν στο καταχωρητή σειριακής φόρτωσης τότε πρέπει να δοθεί παλμός στο καταχωρητή παράλληλης φόρτωσης να βγάλει τα δεδομένα στις εξόδους του. Άρα λοιπόν στο ολοκληρωμένο 4022 συνδέουμε σήμα ρολογιού αυτό που έχουν και τα υπόλοιπα ολοκληρωμένα κυκλώματα. Το 4022 έχει 8 εξόδους. Μια μόνο από αυτές μεταβαίνει σε υψηλό δυναμικό σε κάθε παλμό του ρολογιού. Άρα αν συνδέσουμε μια έξοδο στο ρολόι του 4022 του καταχωρητή παράλληλης φόρτωσης τότε θα παίρνει παλμό

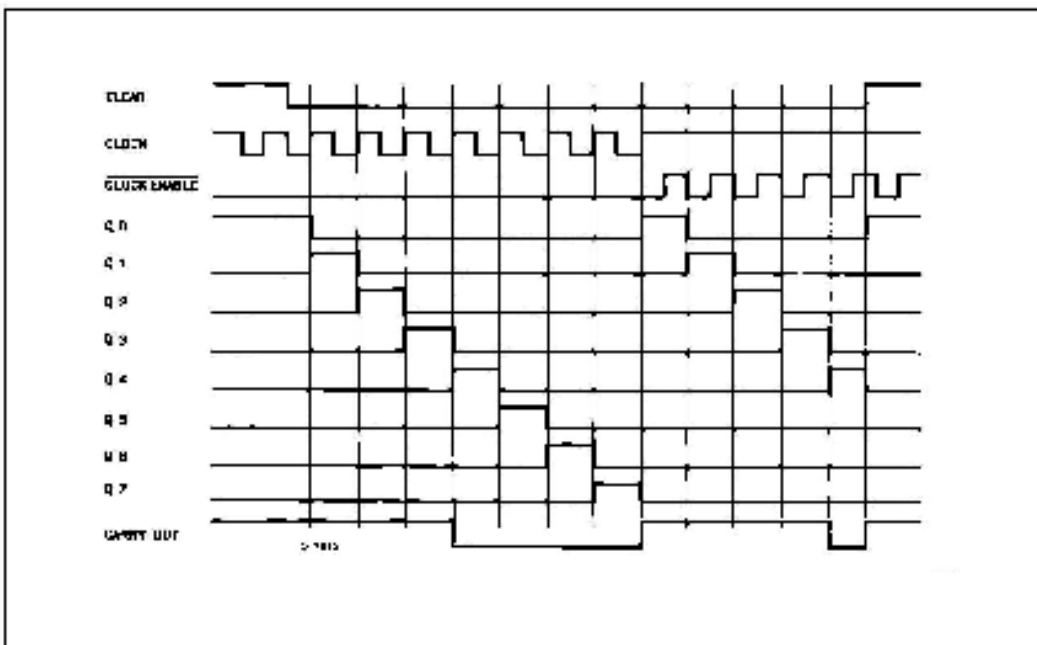
μετά από την εισαγωγή και το 8 bits από τον υπολογιστή στο καταχωρητή σειριακής φόρτωσης. Έτσι λοιπόν ο μετρητής 4022 μετράει 8 παλμούς και μετά δίνει ένα παλμό στο καταχωρητή παράλληλης φόρτωσης

Στην εικόνα 2.11 βλέπουμε τη θήκη του ολοκληρωμένου κυκλώματος 4022. Στα pin 8 & 16 έχουμε τη σύνδεση της γείωσης και της τάσης τροφοδοσίας αντίστοιχα. Τα pin 2 έχουμε την έξοδο Q2 του 4022 και τη συνδέουμε ως ρολόι στο καταχωρητή παράλληλης φόρτωσης. Χρησιμοποιήσαμε την έξοδο 2 διότι αυτή μας έδινε υψηλό δυναμικό μετά από 8 παλμούς ρολογιού. Οι υπόλοιπες εξοδοι μεταβαίνουν σε υψηλό δυναμικό διαδοχικά σε επόμενους παλμούς. Οι υπόλοιποι εξοδοι δε συνδέθηκαν καθόλου. Επίσης το pin 13 (clock enable) το συνδέουμε στη γείωση. Τέλος το Pin 15 είναι αναγκαίο για σήμα Reset του ολοκληρωμένου. Το συνδέσαμε στη γείωση μέσω μιας αντίστασης 100K και με ειδικό κουμπί (button) η ακίδα (pin) 15 μεταβαίνει είτε στη γείωση είτε σε υψηλό δυναμικό. Άρα όταν μεταβαίνει σε υψηλό δυναμικό έχουμε reset στο ολοκληρωμένο.



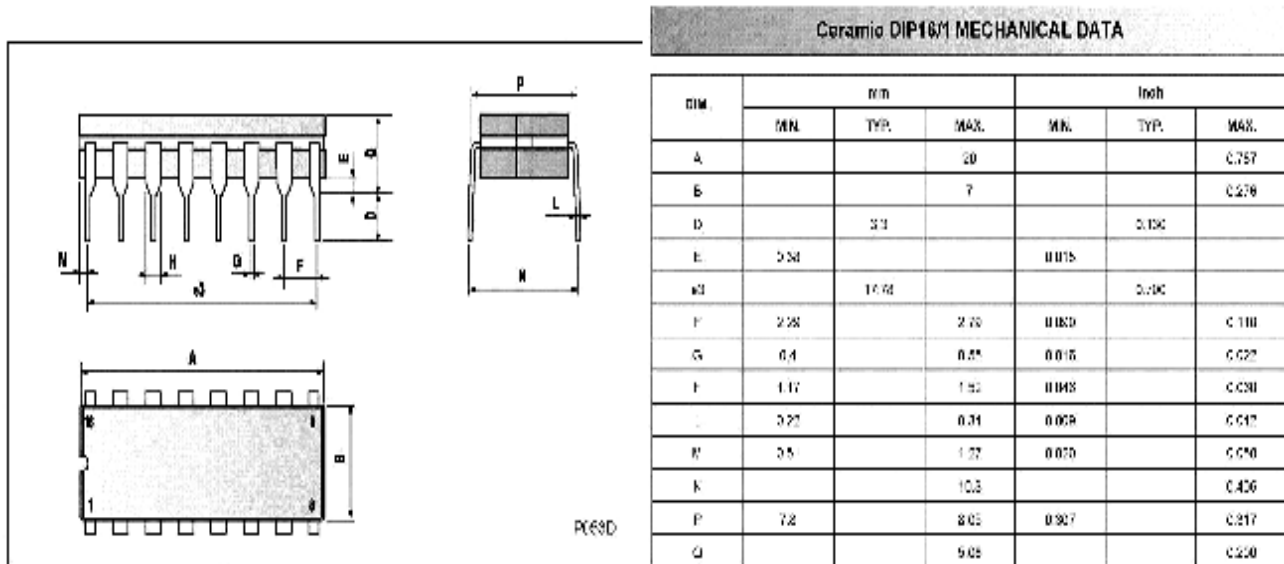
**Εικόνα 2.11:** Επεξήγηση ακίδων ολοκληρωμένου 4022

Στην εικόνα 2.12 βλέπουμε το διάγραμμα καταστάσεων των εξόδων σε σχέση με το σήμα του ρολογιού και του σήματος clock enable και clear.



**Εικόνα 2.12:** διάγραμμα καταστάσεων ακίδων

- Το ολοκληρωμένο 4022 έχει τα ακόλουθα χαρακτηριστικά:
- Τάση σωστής λειτουργίας από 2 Volt έως περίπου 6 Volt
- Τάση εισόδου από 0 Volt έως Vcc
- Τάση εξόδων από 0 Volt έως Vcc
- Θερμοκρασία στο εσωτερικό του για σωστή λειτουργία -55 έως 125°C
- Ρεύμα εξόδου 25 mA



Εικόνα 2.13: Φυσικές διαστάσεις ολοκληρωμένου 4022

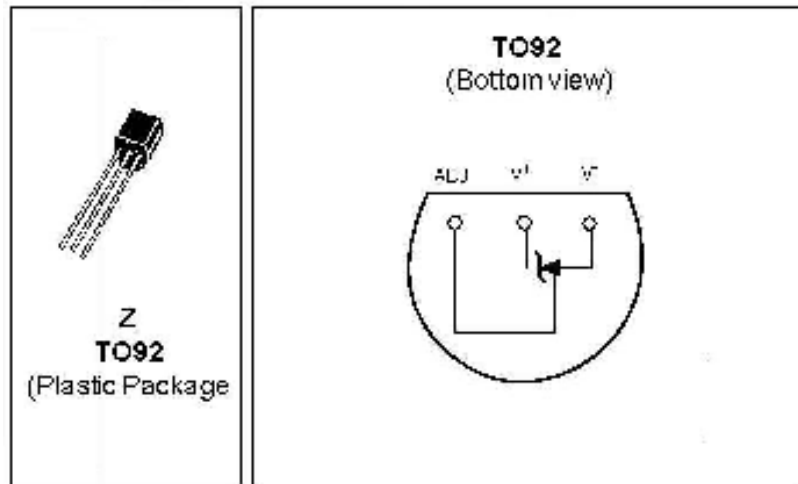
Τέλος στην εικόνα 2.13 βλέπουμε το σχέδιο της θήκης του ολοκληρωμένου κυκλώματος 4022 και οι φυσικές του διαστάσεις σε χιλιοστά και ίντσες (mm και inch)

## 2.5. Ψηφιακό θερμομέτρο με τη χρήση ολοκληρωμένου LM335

Αυτή η βαθμίδα της λειτουργικής κάρτας είναι υπεύθυνη για την μέτρηση της θερμοκρασίας και απεικόνιση αυτής στη διεπαφή (interface) του ηλεκτρονικού υπολογιστή. Η βαθμίδα αυτή αποτελείται από δυο ολοκληρωμένα κυκλώματα. Το ένα είναι το LM335 το οποίο έχει αισθητήρα θερμοκρασίας και ανάλογα τη μεταβολή της θερμοκρασίας μεταβάλλει και τη τάση στους ακροδέκτες στην έξοδό του. Το δεύτερο ολοκληρωμένο είναι το TLC549IP το οποίο λαμβάνει τη τάση από την έξοδο του LM335 και κάνει μετατροπή από αναλογικό σε ψηφιακό. Δηλαδή κάθε τάση αντιστοιχεί σε διαφορετική συστοιχία δυαδικών ψηφίων. Παρακάτω θα δούμε αναλυτικότερα και τα δυο ολοκληρωμένα.

### 2.5.1. Αισθητήρας θερμοκρασίας LM335

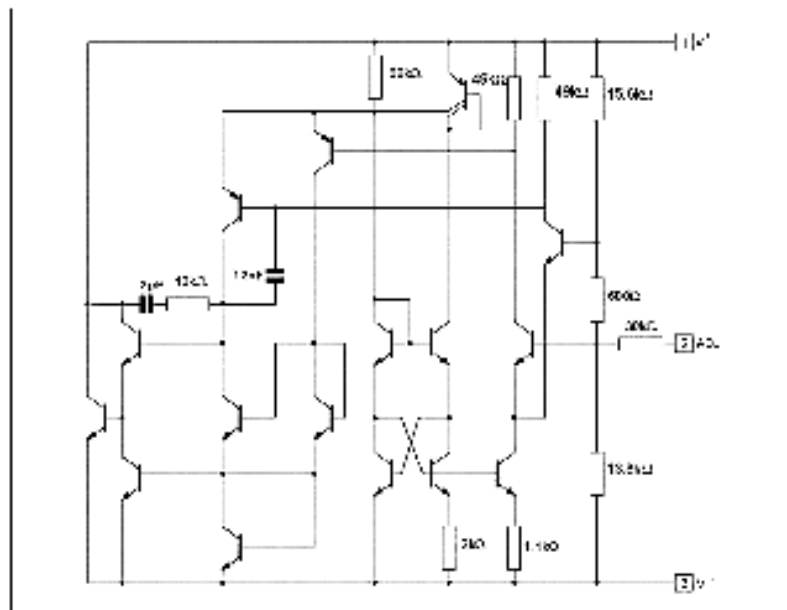
Όπως είπαμε προηγουμένως το LM335 αποδίδει τη θερμοκρασία του περιβάλλοντος σε μεταβαλλόμενη τάση στην έξοδό του. Η παρακάτω εικόνα μας δείχνει τη θήκη του και οι ακίδες (pins):



Εικόνα 2.14: Θήκη και επεξήγηση των ακίδων (Pins)

Όπως βλέπουμε στη παραπάνω εικόνα έχουμε τρεις ακροδέκτες στο LM335. Ο πρώτος έχει ονομασία adjust και τον συνδέουμε σε ένα ποτενσιόμετρο. Με το ποτενσιόμετρο αυτό μεταβάλλουμε τη τάση που πέφτει στον ακροδέκτη. Με τη μεταβολή της τάσης αυτής καλιμπράρουμε τον αισθητήρα θερμοκρασίας έτσι ώστε το θερμοκρασία που θα μετράει να είναι και η πραγματική. Από το δεύτερο ακροδέκτη λαμβάνουμε τη DC τάση και την οδηγούμε στον αναλογικό σε ψηφιακό μετατροπέα (A/D converter). Ο τρίτος ακροδέκτης οδηγείται στη γείωση ως σημείο αναφοράς.

Στην εικόνα 2.15 βλέπουμε το σχηματικό διάγραμμα του εσωτερικού του LM335.



Εικόνα 2.15: Σχηματικό διάγραμμα LM335

Το ολοκληρωμένο LM335 έχει τα ακόλουθα χαρακτηριστικά:

- Μπορεί να ρυθμιστεί να καταμετρά και σε  $^{\circ}\text{C}$  και σε  $^{\circ}\text{K}$
- Ακρίβεια  $1^{\circ}\text{C}$
- Λειτουργεί από  $450\mu\text{A}$  έως  $5\text{mA}$





μας εξυπηρετεί να βρίσκεται στη γείωση. Στο Pin 7 συνδέουμε το σήμα του ρολογιού που όπως είπαμε είναι κοινό και για τα υπόλοιπα κυκλώματα. Το pin 6 το συνδέουμε με έναν ακροδέκτη εισόδου δεδομένων της σειριακής θύρας. Τέλος ο ακροδέκτης 2 είναι η είσοδος στο ολοκληρωμένο της αναλογικής τάσης από το LM335.

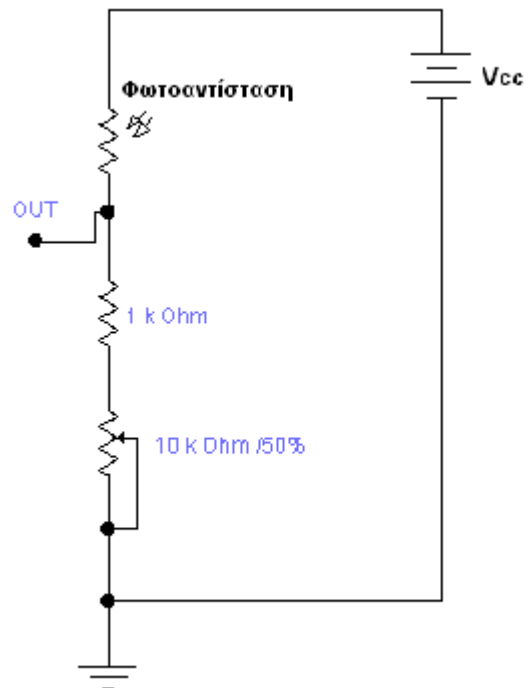
Μερικές ιδιότητες του ολοκληρωμένου TLC549IP:

- 8-bit analog to digital converter
- Τάση λειτουργίας από 3 έως 6 Volt
- Χρόνος μετατροπής 17μS maximum
- Απώλεια ισχύος 15mW
- CMOS τεχνολογία
- 4 MHz εσωτερικό ταλαντωτή

## 2.6. Ανίχνευση φωτεινότητας

Ένα κομμάτι της εργασίας όπως έχουμε αναφέρει και σε προηγούμενα κεφάλαια είναι η ενεργοποίηση ενός φωτιστικού και κατόπιν η ανίχνευση αυτού ότι όντως έχει ενεργοποιηθεί. (και αυτό γιατί ας υποθέσουμε ότι φώτα βρίσκονται μακριά και θέλουμε να βεβαιωθούμε ότι ενεργοποιήθηκαν). Ο φωτισμός θα ενεργοποιείται από τον καταχωρητή MC74HC374A που περιγράψαμε σε προηγούμενο υποκεφάλαιο. Δηλαδή ενεργοποιείται με μια ακίδα εξόδου του καταχωρητή. Θα χρειαστεί φυσικά και ένα ρελέ.

Για να γίνει λοιπόν η ανίχνευση του φωτισμού θα βάλουμε κοντά στο led της πλακέτας (το οποίο led θα μπορούσε να αναπαριστά και ένα φωτιστικό κύκλωμα ενός κτιρίου) μια φωτοαντίσταση

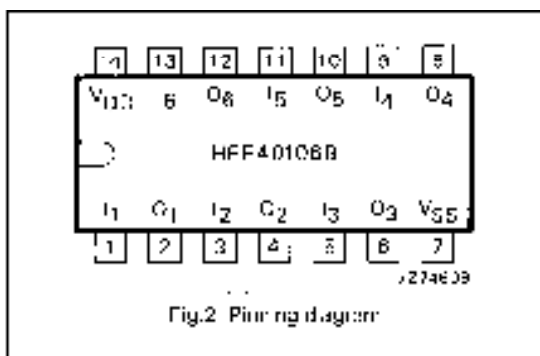


Σχήμα 2.18  
Κύκλωμα ανίχνευσης φωτεινότητας

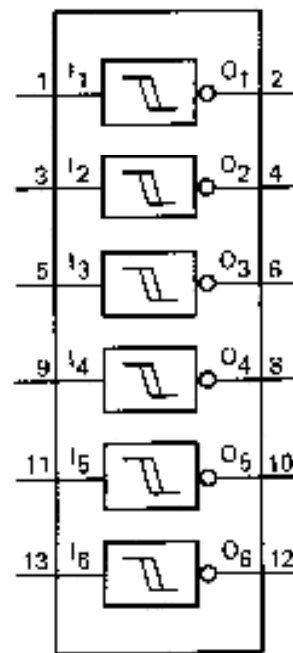
Στα άκρα λοιπόν της φωτοαντίστασης με τη μεταβολή της φωτεινότητας θα έχουμε και μεταβολή της τάσης. Με τη βοήθεια του ποτενσιόμετρου (το οποίο είναι σε σειρά σε ένα διαιρέτη τάσης) μεταβάλλουμε τη τάση στο σημείο OUT του σχήματος 2.18. Οπότε το ιδανικό που θέλουμε είναι ένα υψηλό δυναμικό στο σημείο OUT όταν ανάβει το led και ένα χαμηλό δυναμικό όταν δεν ανάβει το led στα άκρα της φωτοαντίστασης. Όμως πρακτικά αυτό δεν γινόταν διότι τα πειράματα γίνονταν και στο φως της ημέρας οπότε οι μεταβολές στη τάση ήταν ελάχιστες. Οπότε χρειαζόταν κάποιο ακόμη κύκλωμα που θα ήταν ευαίσθητο στις ελάχιστες μεταβολές της τάσης. Και έτσι λοιπόν επιλέχθηκε το ολοκληρωμένο κύκλωμα HEF 40106 (Hex Inverting Schmitt trigger). Έτσι λοιπόν στην έξοδο των πυλών αυτών έχουμε το επιθυμητό αποτέλεσμα. Τέλος όπως είπαμε προηγουμένως γυρίζουμε το ποτενσιόμετρο του σχήματος 2.18 και έτσι εκτός τη τάση μεταβάλλουμε και την ευαισθησία του φωτοκύτταρου. Σε κάποια τιμή της αντίστασης του ποτενσιόμετρου το φωτοκύτταρο λειτουργεί ιδανικά.

### 2.7. HEF40106 Πύλες NOT – Hex Inverting Schmitt trigger

Το ολοκληρωμένο HEF40106 αποτελείται από έξι πύλες Not Schmitt trigger ανεξάρτητες μεταξύ τους. Τα παρακάτω δυο σχήματα κάνουν περιγραφή των ακίδων (pins) καθώς και του εσωτερικού του ολοκληρωμένου.



Σχήμα 2.19: Προσδιορισμός των ακίδων



Σχήμα 2.20: Εσωτερικό ολοκληρωμένου

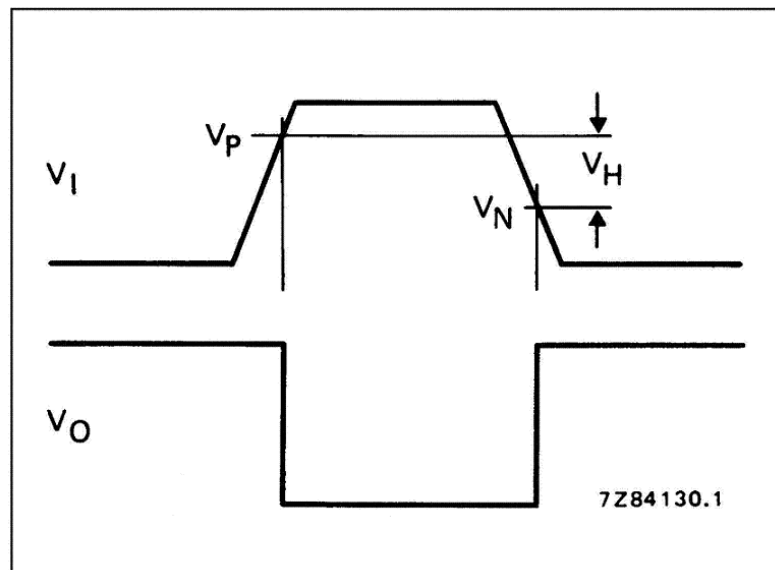
Ο παρακάτω πίνακας κάποια χαρακτηριστικά της κάθε πύλης ανάλογα με τη τάση τροφοδοσίας:

	VDD V	Σύμβολο	MIN.	Τυπικό	MAX.
Hysteresis voltage	5	VH	0,5	0,8	V
	10		0,7	1,3	V
	15		0,9	1,8	V

Switching levels positive-going input voltage	5	VP	2	3,0	3,5	V
	10		3,7	5,8	7	V
	15		4,9	8,3	11	V
negative-going input voltage	5	VN	1,5	2,2	3	V
	10		3	4,5	6,3	V
	15		4	6,5	10,1	V

Πίνακας 2.4: Χαρακτηριστικά των πυλών

Τα σύμβολα του παραπάνω πίνακα φαίνονται στο σχήμα 2.21



Σχήμα 2.21: Τάση εισόδου – Τάση εξόδου

Σύμφωνα με τη μελέτη αυτή αυτό το ολοκληρωμένο φάνηκε χρήσιμο στην εφαρμογή διότι με ελάχιστη μεταβολή στη φωτεινότητα η έξοδος της πύλης μετέβαινε από υψηλό δυναμικό "1" σε χαμηλό δυναμικό "0" και το αντίθετο. Άρα είχαμε φτιάξει το επιθυμητό. Συνεπώς συνδέσαμε την μια άκρη της φωτοαντίστασης (δηλαδή το άκρο OUT του σχήματος 2.18) με την είσοδο μιας πύλης. Όμως τα αποτελέσματα που παίρνουμε είναι ακριβώς αντίθετα γιατί η πύλη είναι ταυτόχρονα και αντιστροφέας. Οπότε την έξοδο της πύλης τη συνδέσαμε σε άλλη μια ίδια πύλη και από την έξοδο της δεύτερης λαμβάνουμε τα αποτελέσματα που θέλουμε. Δηλαδή είτε σε σκοτεινό χώρο είτε σε φωτεινότητα δωματίου και με το led απενεργοποιημένο στην έξοδο της δεύτερης πύλης θα έχουμε χαμηλό δυναμικό "0". Επίσης στους ίδιες περιπτώσεις αλλά με αναμμένο led θα έχουμε υψηλό δυναμικό "1". Για να γίνεται αυτό όμως πετυχημένα πρέπει να πειραματιστούμε μέχρι να ρυθμίσουμε στη σωστή αντίσταση το ποτενσιόμετρο του σχήματος 2.18.

## 2.8. Συγκέντρωση όλων των βαθμίδων σε μια τελική κατασκευή

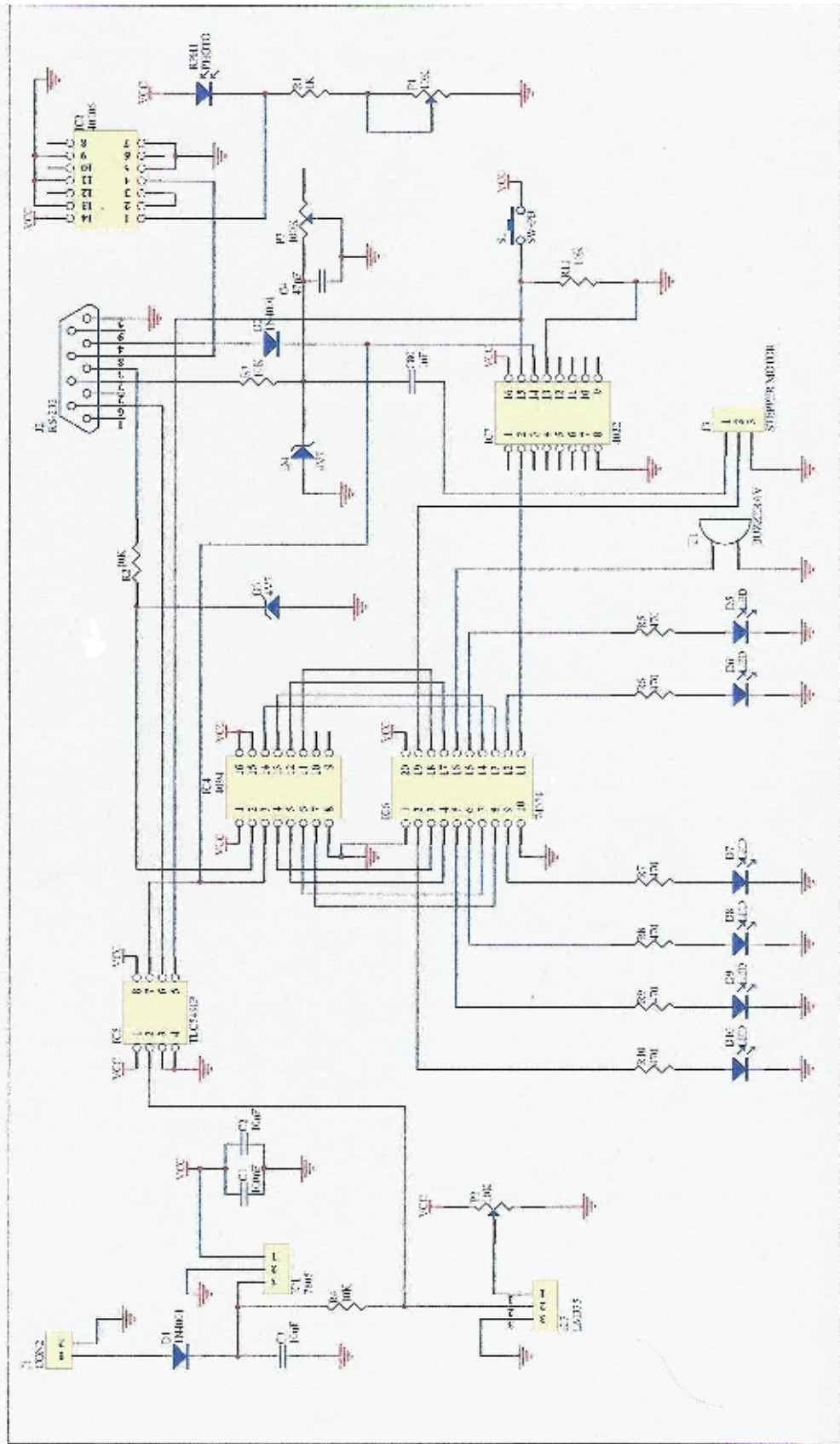
Μέχρι εδώ έχουμε δει όλες τις βαθμίδες της λειτουργικής κάρτας μια μια αναλυτικά. Τώρα το μόνο που μένει είναι να δούμε όλες τις βαθμίδες μαζί συγκροτώντας τη λειτουργική

κάρτα. Θα δούμε πρώτα το ηλεκτρονικό σχέδιο και κατόπιν το τυπωμένο κύκλωμα μαζί με τα εξαρτήματα που ήταν και το τελικό στάδιο της κατασκευής.

Στον επόμενο πίνακα βλέπουμε αναλυτικά όλα τα εξαρτήματα που χρησιμοποιήθηκαν για τη κατασκευή της λειτουργικής κάρτας.

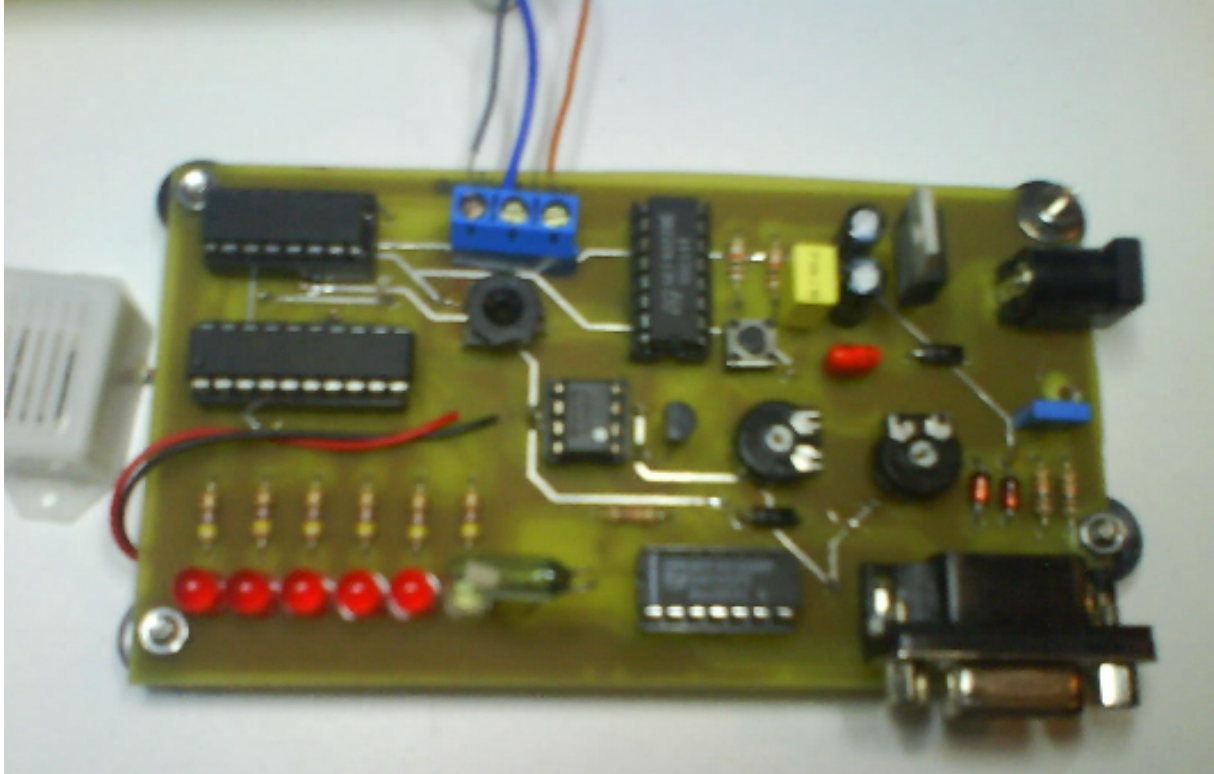
Όνομα	Τύπος	Τιμή	Ποσότητα
4022	Μετρητής – Octal counter	-	1
TC74HC4094	καταχωρητής	-	1
MC74HC374A	καταχωρητής	-	1
LM335	Αισθητήρας θερμοκρασίας	-	1
TLC549IP	Analog to digital converter	-	1
HEF40106	Inverting Schmitt Trigger	-	1
LM7805	Σταθεροποιητής Τάσης	-	1
R1	Αντίσταση	1KΩ	1
R2-R4	Αντίσταση	10KΩ	3
R5-R10	Αντίσταση	470Ω	6
R11	Αντίσταση	10KΩ	1
C1	Πυκνωτής	100nF	1
C2	Πυκνωτής	10μF	1
C3	Πυκνωτής	10μF	1
C4	Πυκνωτής	47μF	1
C5	Πυκνωτής	1μF	1
LED	Led Ισχύος	-	1
LED	κόκκινα	-	5
Φωτοαντίσταση	-	-	1
P1-P2	10KΩ	-	2
P3	100KΩ	-	1
D1-D2	Δίοδοι 1N4001	-	2
D3-D4	Δίοδοι Zener 4,7 Volt	-	2
Serial Connector (Θηλυκός)	-	-	1
Connector τροφοδοσίας	διπολικός	-	1
Connector	Με τρεις υποδοχές	-	1
Button	-	-	1
Buzzer	-	-	1

**Πίνακας 2.5: Κατάλογος εξαρτημάτων**



Σχήμα 2.22: Ηλεκτρονικό Σχέδιο Λειτουργικής Κάρτας

Στην παραπάνω εικόνα ακολουθεί το ηλεκτρονικό σχέδιο της λειτουργικής κάρτας συγκροτημένο με όλες τις βαθμίδες που περιγράψαμε σε όλο το δεύτερο κεφάλαιο. Κατόπιν στην εικόνα 2.23 έχουμε μια φωτογραφία της λειτουργικής κάρτας μαζί με όλα τα εξαρτήματά της.



*Σχήμα 2.23: Λειτουργική Κάρτα μαζί με τα εξαρτήματα*

### 3. Βηματικοί κινητήρες και κυκλώματα οδήγησης

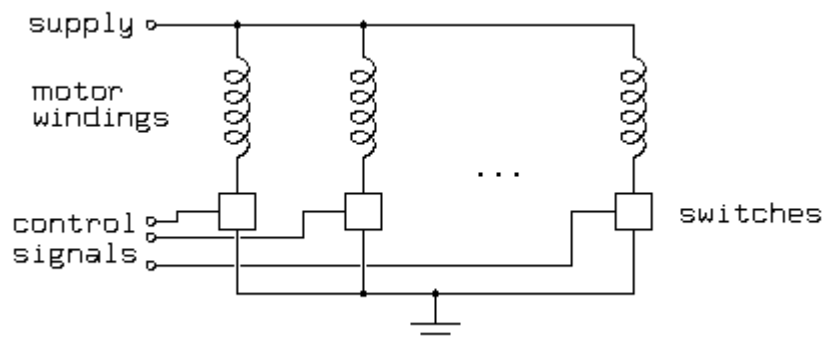
Σε αυτό το κεφάλαιο εξετάζονται τα βασικά στοιχεία κυκλωμάτων για να οδηγηθούν τα διάφορα είδη των βηματικών μοτέρ. Αυτά τα στοιχεία κυκλωμάτων έχουν την ίδια ιδιότητα, δηλαδή να αναστρέφουν τη φορά του ρεύματος σε κάθε περιέλιξη μεταβάλλοντας τη τάση μεταξύ μιας άνω στάθμης και μηδενικής. Επίσης τα κυκλώματα αυτά να μπορούν ελέγχουν την κατεύθυνση περιστροφής του. Οι περιελίξεις των βηματικών κινητήρων ελέγχονται από ένα ψηφιακό κύκλωμα που καθορίζει μέσω διακοπών ροή ρεύματος ή μη.

Το κεφάλαιο αυτό καλύπτει όλους τους τύπους βηματικών κινητήρων, από τα στοιχειώδη στοιχεία κυκλώματος που απαιτούνται για να δημιουργήσουν ένα μεταβλητό μηχανισμό τάσεων, έως το τύπων των γεφυρών - Η που απαιτούνται για να ελέγξουν ένα διπολικό βηματικό κινητήρα μόνιμου μαγνήτη. Κάθε κατηγορία κυκλώματος είναι απεικονισμένη με τα πρακτικά παραδείγματα. Αυτά όμως τα παραδείγματα δεν περιέχουν όλο το κατάλογο των εμπορικά διαθέσιμων κυκλωμάτων ελέγχου βηματικών κινητήρων. Με κάποιες μικρές αλλαγές στο κύκλωμά τους μπορούν να προσαρμοστούν σε οποιοδήποτε τύπο βηματικών κινητήρων που υπάρχουν στην αγορά. Αυτό το κεφάλαιο καλύπτει μόνο τα πιο στοιχειώδη στοιχεία κυκλώματος ελέγχου για κάθε κατηγορία κινητήρων. Όλα αυτά τα κυκλώματα ρυθμίζουν την παρεχόμενη ηλεκτρική ισχύ ούτως ώστε η τάση στις περιελίξεις του κινητήρα να μην υπερβαίνει τη τιμή οπου έχει οριστεί από τον κατασκευαστή. Και από την άλλη φυσικά η τάση στις περιελίξεις να μη πέφτει σε χαμηλά επίπεδα και χάνεται έτσι η απόδοση του κινητήρα.

#### 3.1. Βασικά κυκλώματα ελέγχου βηματικών κινητήρων

##### 3.1.1. Βηματικοί κινητήρες μεταβλητής αντίδρασης

Στο σχήμα 3.1, τα κουτιά χρησιμοποιούνται για να αντιπροσωπεύσουν τις μονάδες ελέγχου όπου είναι αρμόδιες για την παροχή των σημάτων ούτως ώστε να ανοίξουν και να κλείσουν οι διακόπτες στις κατάλληλες χρονικές στιγμές προκειμένου να περιστρέφουν τους κινητήρες. Σε πολλές περιπτώσεις, η μονάδα ελέγχου θα είναι ένας υπολογιστής ή προγραμματισμένος ελεγκτής επαφών, με το λογισμικό του οποίου παράγει άμεσα τα αποτελέσματα που απαιτούνται για να ελέγξουν τους διακόπτες.



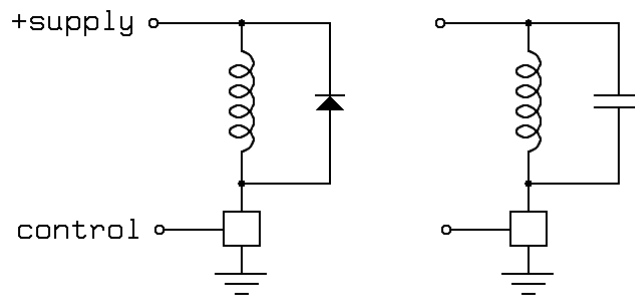
Σχήμα 3.1: Κύκλωμα ελέγχου των σημάτων

Οι περιελίξεις κινητήρων, τα σωληνοειδή και οι παρόμοιες συσκευές είναι όλα επαγωγικά φορτία. Υπό αυτήν τη μορφή, το ρεύμα μέσω της περιελίξεως κινητήρων δεν



μπορεί να μεταβεί από την κατάσταση λειτουργίας (ON) στη κατάσταση αποκοπής (OFF) στιγμιαία χωρίς εφαρμογή άπειρων τάσεων. Όταν ο διακόπτης που ελέγχει τη περιέλιξη κλείσει, επιτρέπει τη ροή ρεύματος, και ως αποτέλεσμα αυτό είναι μια αργή άνοδος στη τιμή του ρεύματος. Όταν ο διακόπτης ανοίξει, το αποτέλεσμα αυτού είναι ένα ηλεκτρικό τόξο τάσης που μπορεί να καταστρέψει το διακόπτη εκτός αν έχουν ληφθεί μέτρα για την αποφυγή αυτού του αποτελέσματος.

Υπάρχουν δύο βασικοί τρόποι για την αποφυγή του ηλεκτρικού τόξου. Ο πρώτος είναι να τοποθετηθεί παράλληλα στη περιέλιξη με μια δίοδος πολωμένη ανάστροφα. Ο δεύτερος τρόπος είναι αντί για μια δίοδο να τοποθετήσουμε ένα πυκνωτή. Το σχήμα 3.2 απεικονίζει και τις δύο περιπτώσεις:



**Σχήμα 3.2: Τρόποι για την αποφυγή του ηλεκτρικού τόξου.**

Η δίοδος που παρουσιάζεται στο σχήμα 3.2 πρέπει να είναι σε θέση να μην άγει όταν το ρεύμα άγει πλήρως διαμέσου των περιελίξεων, αλλά μόνο εν συντομία κάθε φορά που κλείνει ο διακόπτης, καθώς το ρεύμα μέσω της περιελίξεως μειώνεται σταδιακά. Εάν τοποθετηθεί μια σχετικά αργή στη μετάβαση δίοδος όπως η κοινή οικογένεια 1N400X τότε, μπορεί να είναι απαραίτητο να προστεθεί ένας μικρός πυκνωτής παράλληλα με τη δίοδο.

Ο πυκνωτής που παρουσιάζεται στο σχήμα 3.2 έχει την εξής ιδιότητα. Όταν ο διακόπτης κλείσει, ο πυκνωτής θα εκφορτίσει μέσω του διακόπτη προς τη γη, και ο διακόπτης πρέπει να είναι σε θέση να χειριστεί αυτήν την σύντομη και απότομη αύξηση του ρεύματος εκφόρτισης. Μια αντίσταση σε σειρά με τον πυκνωτή ή σε σειρά με την τάση τροφοδοσίας θα περιορίσει αυτό το ρεύμα. Όταν ο διακόπτης ανοίξει, η αποθηκευμένη ενέργεια στις περιελίξεις θα φορτίσουν τον πυκνωτή με μια τάση σημαντικά μεγαλύτερη από την τάση τροφοδοσίας, και ο διακόπτης πρέπει να είναι σε θέση να αντέξει αυτήν την τάση. Για να προσδιοριστεί το μέγεθος του πυκνωτή, εξισώνουμε τους δύο τύπους για την αποθηκευμένη ενέργεια σε ένα συντονισμένο κύκλωμα:

$$P = CV^2/2$$

$$P = LI^2/2$$

P - Αποθηκευμένη ενέργεια

C - Χωρητικότητα

V - Τάση στο πυκνωτή

L - Αυτεπαγωγή των περιελίξεων σε Henrys

I - Ρεύμα που διαρρέει τις περιελίξεις

Η επίλυση για την ελάχιστη χωρητικότητα του πυκνωτή που απαιτείται για να αποτρέψει την υπέρταση στο διακόπτη είναι:

$$C > L I^2 / (V_b - V_s)^2$$

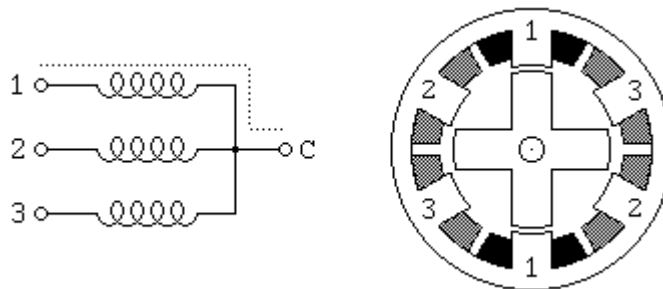
Όπου:

$V_b$  – Η τάση κατωφλίου του διακόπτη  
 $V_s$  – Η τάση τροφοδοσίας

Αυτού του τύπου κινητήρες έχουν μεταβλητή αυτεπαγωγή που εξαρτάται από τη γωνία που σχηματίζει ο άξονάς τους. Επομένως να επιλεγθεί η κατάλληλη χωρητικότητα πυκνωτή. Ο πυκνωτής και οι περιελίξεις, σε συνδυασμό, διαμορφώνουν ένα συντονισμένο κύκλωμα. Εάν το σύστημα ελέγχου οδηγεί το κινητήρα στις συχνότητες κοντά στη συχνότητα συντονισμού του κυκλώματος, το ρεύμα των περιελίξεων του κινητήρα, ώστε η ροπή που ασκείται από το κινητήρα, θα είναι διαφορετική από τη ροπή ονομαστικής τάσης λειτουργίας. Η συχνότητα συντονισμού είναι:

$$f = 1 / ( 2\pi (L C)^{0.5} )$$

Η ηλεκτρική συχνότητα συντονισμού για ένα κινητήρα αυτού του είδους θα εξαρτηθεί από τη γωνία του άξονα του κινητήρα. Όταν οι κινητήρες χρησιμοποιούνται με παλμούς, το μεταβαλλόμενο ρεύμα στη περιέλιξη των κινητήρων θα δημιουργήσει ένα μαγνητικό πεδίο που θα μηδενίζεται δύο φορές κατά την συχνότητα συντονισμού, και αυτό μπορεί να μειώσει σοβαρά τη διαθέσιμη ροπή.



**Σχήμα 3.3: Κινητήρας με τρεις περιελίξεις συνδεδεμένες με κοινό κόμβο.**

Εάν ο κινητήρας έχει τρεις περιελίξεις, συνδεδεμένες όπως φαίνεται στο σχηματικό διάγραμμα στο σχήμα 3.3, με έναν τελικό κοινό κόμβο για όλες τις περιελίξεις, τότε σε λειτουργία, το κοινό καλώδιο μεταβαίνει σε θετικό δυναμικό και η κάθε περιέλιξη ενεργοποιείται διαδοχικά. Για να περιστρέψουμε αυτόν τον κινητήρα συνεχώς, εφαρμόζουμε τάση διαδοχικά στις τρεις περιελίξεις. Υποθέτοντας λογικό '1', όπου '1' σημαίνει ότι το ρεύμα ρέει μέσω μιας περιέλιξης του κινητήρα, η ακόλουθη ακολουθία ελέγχου που φαίνεται παρακάτω θα περιστρέψει το κινητήρα που απεικονίζεται στο σχήμα 3.3 δεξιόστροφα κατά 24 βήματα ή 2 περιστροφές:

```

Περιέλιξη 1.1001001001001001001001001
Περιέλιξη 2.0100100100100100100100100
Περιέλιξη 3.0010010010010010010010010
χρόνος >

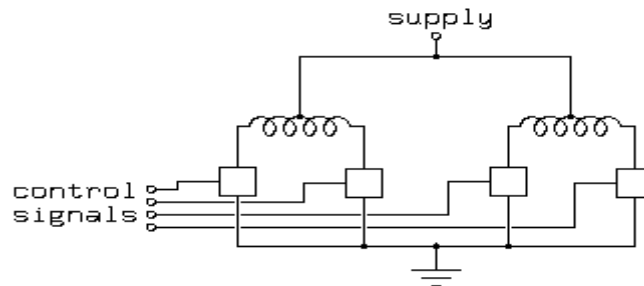
```

Υπάρχουν επίσης κυκλώματα οδήγησης αυτού του τύπου με 4 και 5 περιελίξεις, που απαιτεί 5 ή 6 καλώδια. Η αρχή για αυτές τις μηχανές είναι η ίδια με την προηγούμενη, αλλά είναι σημαντικό να επιλυθεί η σωστή δυαδική ακολουθία για να ενεργοποιηθούν σωστά τις περιελίξεις και να δίνουν το βήμα περιστροφής. Η γεωμετρία κινητήρων του παραπάνω

σχήματος δίνει 30 μοίρες ανά βήμα. Ανάλογα με τον αριθμό των δοντιών του ρότορα και τον αριθμό των πόλων του στάτη τότε έτσι δημιουργούνται τα ανάλογα βήματα σε μοίρες. Περισσότερα δόντια και πόλοι τότε μικρότερο βήμα έως και λίγες μοίρες.

### 3.1.2. Μονοπολικοί μόνιμου μαγνητικού πεδίου και Υβριδικοί βηματικοί κινητήρες

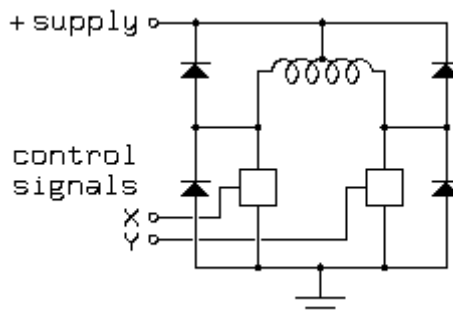
Τα κυκλώματα οδήγησης για μονοπολικούς κινητήρες είναι παρόμοια με αυτό που παρουσιάζεται στο σχήμα 3.4:



Σχήμα 3.4: κύκλωμα οδήγησης μονοπολικών κινητήρων.

Στο σχήμα 3.4, όπως στο σχήμα 3.1, τα κουτιά χρησιμοποιούνται για να αντιπροσωπεύσουν τους διακόπτες. Μια μονάδα ελέγχου, που δεν παρουσιάζεται, είναι αρμόδια για την παροχή των σημάτων ελέγχου για να ανοίγει και να κλείνει τους διακόπτες στις κατάλληλες χρονικές στιγμές προκειμένου να περιστραφούν οι κινητήρες.

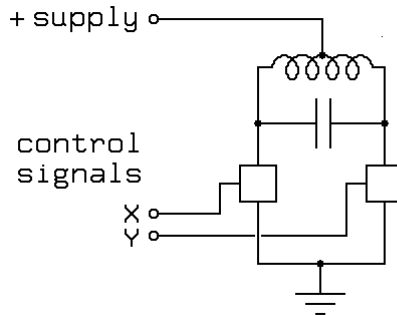
Όπως με τα στοιχεία κυκλώματος στο προηγούμενο είδος κυκλωμάτων οδήγησης, πρέπει να εξετάσουμε την επαγωγική δύναμη που παράγεται όταν κλείνει κάθε ένας από αυτούς τους διακόπτες. Παρομοίως, τοποθετώντας 4 διόδους φαίνεται στο σχήμα 3.5:



Σχήμα 3.5: κύκλωμα ελέγχου μονοπολικών κινητήρων με προσθήκη διόδων

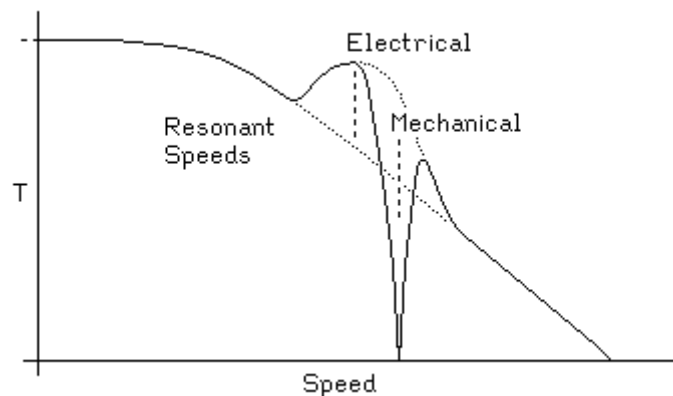
Οι πρόσθετες διόδους απαιτούνται επειδή η περιέλιξη των κινητήρων δεν είναι δύο ανεξάρτητες περιελίξεις, είναι μια ενιαία περιέλιξη με ένα κόμβο στο κέντρο της περιελίξεως με το κεντρικό κόμβο σε μια σταθερή τάση. Αυτό ενεργεί ως αυτό μετασχηματιστής. Όταν το ένα άκρο της περιελίξεως καταβάλλεται, το άλλο άκρο θα βρίσκεται στον αέρα. Όταν ένας διακόπτης ανοίγει, η επαγωγική αντίδραση θα οδηγήσει εκείνο το άκρο της περιελίξεως στη θετική τάσης τροφοδοσίας, όπου συνδέεται με τη diόδο.

Ένας πυκνωτής μπορεί επίσης να χρησιμοποιηθεί για να περιορίσει την τάση αντίδρασης, όπως φαίνεται στο σχήμα 3.6:



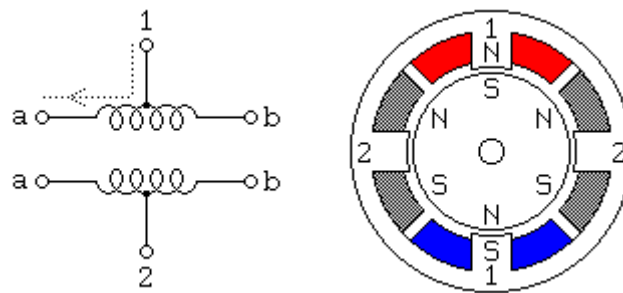
**Σχήμα 3.6: Προσθήκη πυκνωτή για το περιορισμό της τάσης**

Οι ιδιότητες του πυκνωτή που παρουσιάζεται στο σχήμα 3.6 είναι οι ίδιες με τις ιδιότητες για τον πυκνωτή που παρουσιάζεται στο σχήμα 3.2, αλλά η επίδραση του συντονισμού είναι αρκετά διαφορετική. Με ένα κινητήρα μόνιμων μαγνητών, εάν ο πυκνωτής οδηγείται σε ή κοντά στη συχνότητα συντονισμού, η ροπή θα αυξηθεί σε τόσο βαθμό όσο κατά δύο φορές τη ροπή σε λίγες στροφές περιστροφής. Η προκύπτουσα ροπή σε σχέση με τη καμπύλη ταχύτητας μπορεί να είναι αρκετά σύνθετη, όπως απεικονίζεται στο σχήμα 3.7:



**Σχήμα 3.7: Διάγραμμα ταχύτητας – Ροπής**

Το σχήμα 3.7 παρουσιάζει μια αιχμή στη καμπύλη της ροπής στην ηλεκτρική συχνότητα συντονισμού, και ένα αυλάκι στη μηχανική συχνότητα συντονισμού. Εάν η ηλεκτρική συχνότητα συντονισμού τοποθετείται κατάλληλα, η ταχύτητα αποκοπής των διακοπών για το κινητήρα που χρησιμοποιεί οδηγούς διόδων, η επίδραση μπορεί να είναι μια ιδιαίτερη αύξηση στην αποτελεσματική ταχύτητα αποκοπών. Η μηχανική συχνότητα συντονισμού εξαρτάται από τη ροπή, έτσι εάν η μηχανική συχνότητα συντονισμού είναι οπουδήποτε κοντά στην ηλεκτρική, θα μετατοπιστεί από την ηλεκτρικό συντονισμό. Επιπλέον, το πλάτος της μηχανικής αντίδρασης εξαρτάται από την κλίση της ροπής σε σύγκριση με της καμπύλης ταχύτητας εάν η ροπή μειώνεται γρήγορα, τότε η μηχανική αντίδραση θα είναι αιχμηρότερη, ενώ εάν η ροπή αυξάνεται με την ταχύτητα, θα είναι ευρύτερη ή ακόμα και χωρισμένη στις πολλαπλάσιες συχνότητες συντονισμού.



**Σχήμα 3.8: Συνδεσμολογία περιελίξεων ενός μονοπολικού κινητήρα**

Οι μονοπολικοί βηματικοί κινητήρες, όπως και με μόνιμους μαγνήτες συνδέονται με καλώδιο συνήθως όπως φαίνεται στη σχηματική αναπαράσταση στο σχήμα 3.8, με μόνη διαφορά έχουν ένα κοινό κόμβο με ένα ακροδέκτη σε κάθε περιέλιξη του κινητήρα. Σε χρήση, οι ακροδέκτες των κόμβων κάθε περιέλιξης συνδέονται με κοινό καλώδιο στο θετικό της τάσης τροφοδοσίας και οι δύο άκρες κάθε περιέλιξης γειώνονται εναλλάξ για να αντιστρέψουν την κατεύθυνση του μαγνητικού πεδίου το οποίο δημιουργείται από κάθε περιέλιξη.

Όπως φαίνεται στο σχήμα, το ρεύμα ρέει από το κεντρικό κόμβο της περιέλιξεως 1 στον ακροδέκτη α και προκαλεί τον άνω πόλο του στάτη να είναι βόρειος πόλος ενώ ο κάτω πόλος του στάτη να είναι νότιος πόλος. Αυτό έχει σαν αποτέλεσμα ο ρότορας θα γυρίσει 30 μοίρες, ή ένα βήμα.

Για να περιστρέψουμε τη μηχανή συνεχώς, εφαρμόζουμε ακριβώς την ίδια τάση στις περιελίξεις εναλλάξ. Υποθέτοντας ότι έχουμε θετικό δυναμικό ή λογικό '1' σημαίνει ότι το ρεύμα ρέει μέσω μιας περιέλιξεως του κινητήρα, οι παρακάτω δύο ακολουθίες ελέγχου θα περιστρέψουν το κινητήρα που απεικονίζεται στο σχήμα 3.8 δεξιόστροφα 24 βήματα ή 4 περιστροφές:

```

Περιέλιξη 1a 1000100010001000100010001
Περιέλιξη 1b 0010001000100010001000100
Περιέλιξη 2a 0100010001000100010001000
Περιέλιξη 2b 0001000100010001000100010
χρόνος >

```

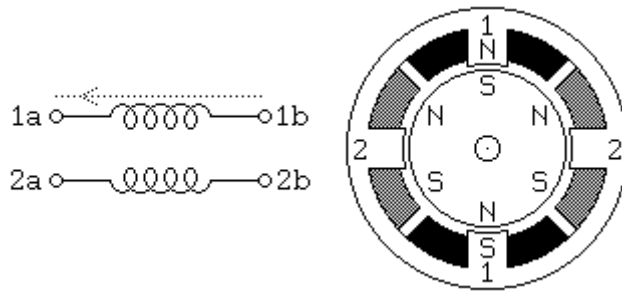
```

Περιέλιξη 1a 1100110011001100110011001
Περιέλιξη 1b 0011001100110011001100110
Περιέλιξη 2a 0110011001100110011001100
Περιέλιξη 2b 1001100110011001100110011
χρόνος >

```

Σημειώνουμε ότι τα δύο μισά κάθε περιέλιξης δεν ενεργοποιούνται ποτέ συγχρόνως. Και οι δύο ακολουθίες που παρουσιάζονται παραπάνω θα περιστραφούν έναν μόνιμο μαγνήτη ένα βήμα με κάθε χρονική στιγμή.

**3.1.3. Διπολικοί βηματικοί κινητήρες και Γέφυρες-H**



*Σχήμα 3.9: Συνδεσμολογία περιελίξεων ενός διπολικού βηματικού κινητήρα*

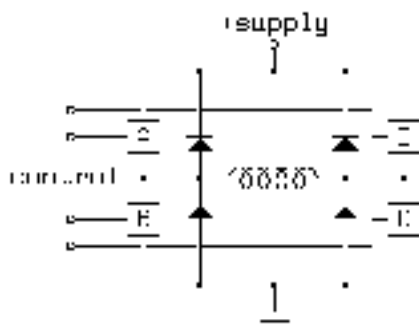
Οι διπολικοί βηματικοί κινητήρες η μόνη διαφορά που έχουν από τους μονοπολικούς είναι ότι δεν έχουν ένα παραπάνω ακροδέκτη στο μέσω ακριβώς κάθε περιελίξεως. Όσον αφορά το κύκλωμα οδήγησης τους είναι πιο σύνθετο από των μονοπολικών. Η σχηματική αναπαράσταση του σχήματος 3.9 επιδεικνύει πώς μια τέτοια μηχανή συνδέεται, ενώ το διαγώνιο τμήμα μηχανών που παρουσιάζεται εδώ δηλαδή η γεωμετρία των πόλων είναι ακριβώς το ίδιο με το διαγώνιο τμήμα που παρουσιάζεται στο σχήμα 3.8.

Τα στοιχεία κυκλώματος οδήγησης για ένα τέτοιο κινητήρα απαιτούν ένα κύκλωμα γέφυρας - H για κάθε περιέλιξη. Εν συντομία, μια γέφυρα - H επιτρέπει την πολικότητα της τάσης που εφαρμόζεται σε κάθε ακροδέκτη κάθε μιας περιελίξεως για να μπορεί να ελεγχθεί ανεξάρτητα. Οι ακολουθίες ελέγχου για ένα απλό κύκλωμα οδήγησης μοτέρ, χρησιμοποιώντας σύμβολα + και - για να δείξουν την πολικότητα της τάσης που εφαρμόζεται σε κάθε ακροδέκτη των περιελίξεων του κινητήρα:

Ακροδέκτης 1a	+---+---+---+---	++---++---++---+
Ακροδέκτης 1b	--+---+---+---+	--++---++---++---
Ακροδέκτης 2a	-+---+---+---+---	-++---++---++---
Ακροδέκτης 2b	---+---+---+---+	+++---++---++---
	χρόνος --->	

Σημειώστε ότι αυτές οι ακολουθίες είναι ίδιες με εκείνους για ένα μονοπολικούς, σε περιληπτικό επίπεδο, άρα τα συστήματα ελέγχου για τους δύο τύπους κινητήρων μπορούν να είναι παρόμοια.

Τα πράγματα είναι πιο σύνθετα για τους διπολικούς κινητήρες μόνιμου μαγνητικού πεδίου επειδή δεν έχουν κάποιο κεντρικό κόμβο στις περιελίξεις τους. Επομένως, για να αντιστρέψουμε την κατεύθυνση του μαγνητικού πεδίου που παράγεται από μια περιέλιξη, πρέπει να αντιστρέψουμε το ρεύμα που ρέει διαμέσου της περιελίξεως.



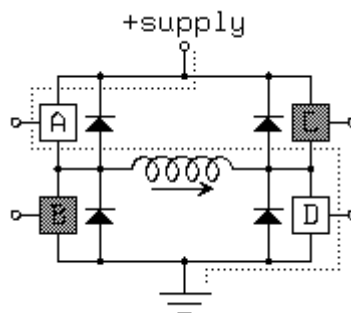
*Σχήμα 3.10: Κυκλωματικό σχήμα γέφυρας - H.*

Θα μπορούσαμε να χρησιμοποιήσουμε ένα διπολικό διακόπτη ο οποίος θα αλλάζει τη πολικότητα ηλεκτρομηχανικά. Το στοιχείο αυτό καλείται γέφυρα - Η και περιγράφεται στο σχήμα 3.10:

Όπως με τα μονοπολικά κυκλώματα οδήγησης (unipolar) που αναφέρθηκαν προηγουμένως, οι διακόπτες Mosfet που χρησιμοποιούνται στη γέφυρα - Η πρέπει να προστατευθούν από τα ηλεκτρικά τόξα που προκαλούνται με τις αυξομειώσεις του ρεύματος του κάθε περιελίξεως. Αυτό γίνεται συνήθως με τις διόδους, όπως φαίνεται στο σχήμα 3.10. Αξίζει ότι οι γέφυρες - Η ισχύουν όχι μόνο στον έλεγχο των διπολικών κινητήρων, αλλά και στον έλεγχο των κινητήρων συνεχούς ρεύματος (DC) και πολλών άλλων εφαρμογών. Με τέσσερις διακόπτες, η βασική γέφυρα-Η προσφέρει 16 πιθανούς τρόπους λειτουργίας, επτά των οποίων η παροχή ηλεκτρικού ρεύματος μηδενίζεται. Οι ακόλουθοι λειτουργικοί τρόποι είναι ενδιαφέροντες:

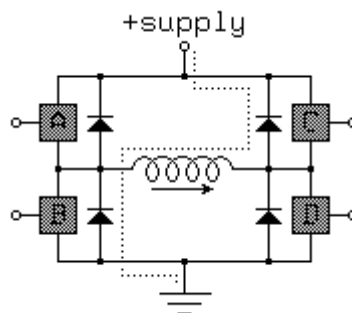
*Forward Mode* οι διακόπτες Α και D κλειστοί.  
*Reverse mode* οι διακόπτες Β και Γ κλειστοί.

Αυτοί είναι οι συνηθισμένοι τρόποι λειτουργίας, που επιτρέπουν στο ρεύμα να ρέει από την τάση τροφοδοσίας, μέσω της περιελίξεως του κινητήρα προς τη γείωση. Το σχήμα 3.11 επεξηγεί τον forward mode



**Σχήμα 3.11:** Λειτουργία του forward mode.

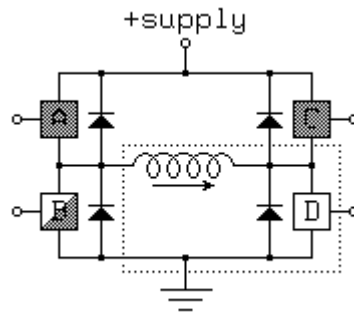
Fast decay mode or coasting mode: όλοι οι διακόπτες είναι ανοιχτοί. Οποιαδήποτε ροή ρεύματος των περιελίξεων θα λειτουργεί ενάντια στην τάση τροφοδοσίας, συν δύο πτώσεις τάσεων των διόδων, έτσι το ρεύμα θα μειωθεί γρήγορα. Αυτός ο τρόπος παρέχει ελάχιστη ή καμία δυναμική επίδραση φρεναρίσματος στο ρότορα των κινητήρων, έτσι ο ρότορας θα στρέφεται ελεύθερα εάν όλες οι περιελίξεις τροφοδοτούνται με αυτόν τον τρόπο. Το σχήμα 3.12 επεξηγεί την ροή ρεύματος αμέσως μετά από τη μετατροπή από τον forward mode στον fast decay mode.



**Σχήμα 3.12:** Μετάβαση από τον forward mode στον fast decay mode.

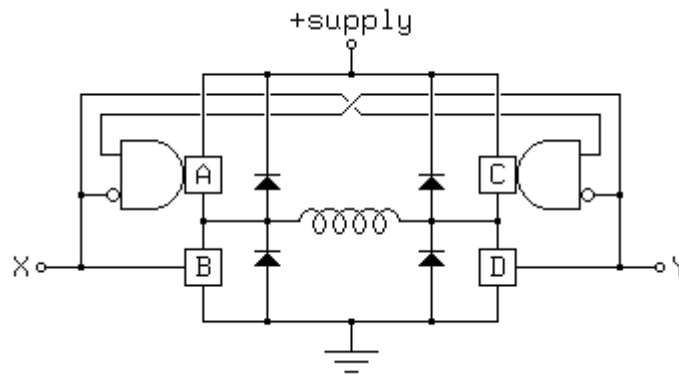
**Slow decay modes or dynamic braking modes**

Σε αυτούς τους τρόπους, το ρεύμα μπορεί να επανακυκλοφορήσει μέσω των περιελίξεων του κινητήρα με την ελάχιστη αντίσταση. Κατά συνέπεια, αν το ρεύμα ρέει σε μια περιέλιξη του κινητήρα όταν δημιουργείται ένας από αυτούς τους τρόπους λειτουργίας, το ρεύμα θα μειωθεί αργά, και εάν ο ρότορας γυρίζει, θα προκαλέσει ένα ρεύμα που θα ενεργήσει ως φρένο στο ρότορα. Το σχήμα 3.13 επεξηγεί ένα από τους πολύ χρήσιμους slow-decay modes, με το διακόπτη D κλειστό. Εάν η περιέλιξη ήταν τελευταία forward running mode, η κατάσταση του διακόπτη B μπορεί να είναι είτε ανοικτή είτε κλειστή:



**Σχήμα 3.13: Ο τρόπος slow-decay mode με κλειστό το διακόπτη D**

Οι περισσότερες γέφυρες - H σχεδιάζονται έτσι ώστε λογικά κυκλώματα είναι απαραίτητα για τη καλή λειτουργία του κυκλώματος οδήγησης έτσι ώστε το κύκλωμα να μεταβαίνει στους τρόπους λειτουργίας (modes) τους οποίους περιγράψαμε προηγουμένως. Το σχήμα 3.14 επεξηγεί ποια είναι πιθανώς η καλύτερη ρύθμιση:



**Σχήμα 3.14: Χρήση λογικών κυκλωμάτων για την οδήγηση της γέφυρας - H**

Εδώ, είναι οι ακόλουθοι διαθέσιμοι τρόποι λειτουργίας:

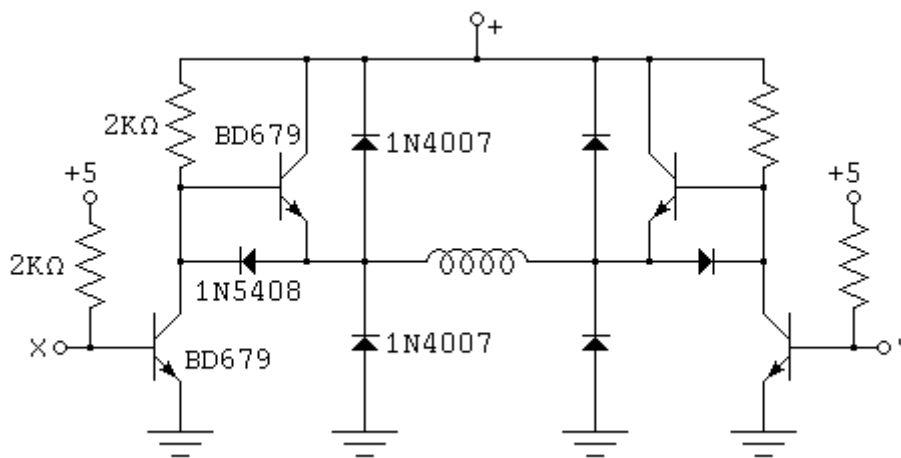
XY	ABCD	Modes
00	0000	fast decay
01	1001	<b>Forward</b>
10	0110	<b>Reverse</b>
11	0101	slow decay



Το πλεονέκτημα αυτής της ρύθμισης είναι ότι όλοι οι χρήσιμοι τρόποι λειτουργίας διατηρούνται, και κωδικοποιούνται με έναν ελάχιστο αριθμό bits και αυτό γιατί κατά την χρησιμοποίηση ενός συστήματος μικροελεγκτών ή ηλεκτρονικών υπολογιστών για να οδηγηθεί μια γέφυρα – Η ο αριθμός των bits είναι περιορισμένος. Δυστυχώς, λίγα από τα ολοκληρωμένα κυκλώματα με γέφυρα - Η που βρίσκουμε στην αγορά έχουν ένα τέτοιο απλό σχηματικό.

### 3.1.4. Πρακτικά κυκλώματα οδήγησης διπολικών κινητήρων

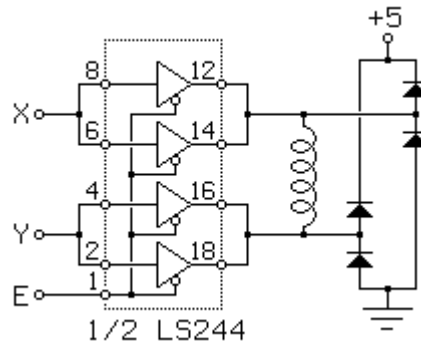
Υπάρχουν στη αγορά διάφορα ολοκληρωμένα κυκλώματα τα οποία περιέχουν κύκλωμα οδήγησης βηματικών κινητήρων με χρήση γεφυρών – Η. Παρόλα αυτά είναι χρήσιμο να μελετήσουμε μια ξεχωριστή υλοποίηση για το πώς λειτουργούν οι γέφυρες - Η. Ο Antonio Raposo (ajr@cybill.inesc.pt) πρότεινε το κύκλωμα με χρήση γέφυρας - Η που παρουσιάζεται στο σχήμα 3.15



Σχήμα 3.15: Κύκλωμα οδήγησης που σχεδίασε ο Antonio Raposo

Οι εισόδους X και Y σε αυτό το κύκλωμα μπορούν να οδηγηθούν από εξόδους ανοιχτού συλλέκτη TTL όπως στο μονοπολικό κύκλωμα οδήγησης (darlington-based unipolar) στο σχήμα 3.8. Η περιέλιξη του κινητήρα θα ενεργοποιηθεί εάν μια από τις εισόδους X και Y είναι σε υψηλό δυναμικό και ακριβώς μια από τις εισόδους είναι σε χαμηλό δυναμικό. Εάν και οι δύο είναι σε χαμηλό δυναμικό, τότε και τα δύο pull-down τρανζίστορ θα βρίσκονται σε κατάσταση αποκοπής. Το ίδιο με τα pull-down τρανζίστορ συμβαίνει όταν και οι δύο εισόδους είναι σε υψηλό δυναμικό. Κατά συνέπεια, αυτό το απλό κύκλωμα βάζει το κινητήρα στο dynamic braking mode στις καταστάσεις των XY 11 και 00.

Το κύκλωμα στο σχήμα 3.16 αποτελείται από δύο ίδια μισά, κάθε ένα από τα οποία μπορεί να περιγραφεί κατάλληλα ως κύκλωμα push-pull. Η μισή γέφυρα – Η εφαρμόζεται μερικές φορές σε αυτά τα κυκλώματα. Αξίζει επίσης να σημειωθεί ότι μια μισή γέφυρα - Η έχει ένα κύκλωμα αρκετά παρόμοιο με το κύκλωμα κύκλωμα εξόδου λογικής TTL. Στην πραγματικότητα, οι γραμμές μεταφοράς TTL tri-state όπως το 74LS12Δ και το 74LS244 μπορούν να χρησιμοποιηθούν ως μισές γέφυρες – Η για μικρά φορτία, όπως απεικονίζεται στο σχήμα 3.16:



**Σχήμα 3.16: κύκλωμα push pull μισών γεφυρών – H**

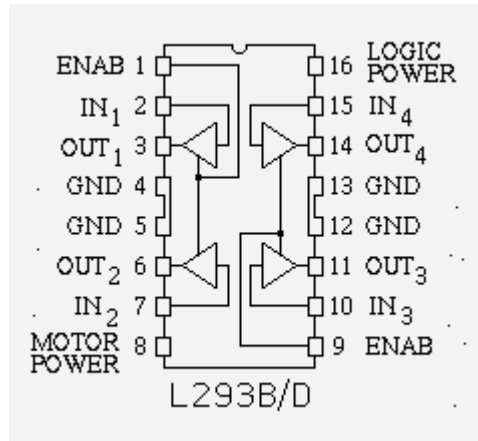
Αυτό το κύκλωμα είναι αποτελεσματικό για κινητήρες με αντίσταση μέχρι περίπου 50Ω σε κάθε περιέλιξη και τάσεις πάνω από 4,5 βολτ όταν χρησιμοποιούν τάση τροφοδοσίας 5 βολτ. Κάθε τριων καταστάσεων (tri-state) απομονωτής στο LS244 μπορεί να εξασθενήσει δύο φορές το ρεύμα, και η εσωτερική αντίσταση των απομονωτών είναι επαρκής, όταν το ρεύμα της πηγής διαιρεθεί ομοιόμορφα ρεύμα μεταξύ των οδηγών που συνδέονται παράλληλα. Αυτό το κύκλωμα οδήγησης επιτρέπει όλες τις δυνατές καταστάσεις που επιτυγχάνονται από τον κύκλωμα οδήγησης στο σχήμα 3.14, αλλά αυτές οι καταστάσεις δεν κωδικοποιούνται όμως αποτελεσματικά:

XYE	Modes
--1	fast decay
000	slower decay
010	<b>forward</b>
100	<b>reverse</b>
110	slow decay

Ο δεύτερος τρόπος λειτουργίας, XYE=110, παρέχει μια ελαφρώς πιο αδύνατη επίδραση φρεναρίσματος από τον πρώτο λόγω του γεγονότος ότι οι οδηγοί LS244 μπορούν να εξασθενήσουν περισσότερο το ρεύμα αντί να το παράγουν.

Ένα από τα προβλήματα με τα εμπορικά διαθέσιμα ολοκληρωμένα κυκλώματα οδήγησης βηματικών κινητήρων ότι πολλά από αυτά έχουν σχετικά σύντομη διάρκεια ζωής στην αγοράς. Παραδείγματος χάριν, οι σειρές Seagate IPxMxx ολοκληρωμένων με διπλή γεφυρών παράγονταν μόνο όταν η Seagate τα χρησιμοποιούσε για οδήγηση βηματικών μοτέρ σε disk drives. Τα Toshiba TA7279 διπλής γέφυρας – H θα ήταν μια άλλη επιλογή για τους κινητήρες κατανάλωσης κάτω από 1 amp, αλλά πάλι, εμφανίζεται το πρόβλημα να γίνεται για εσωτερική χρήση της Toshiba μόνο.

L293 της [SGS-Thompson](#) με διπλή γέφυρα - H είναι βασικός ανταγωνιστής για τα ανωτέρω ολοκληρωμένα κυκλώματα, αλλά αντίθετα από αυτά, δεν περιλαμβάνει τις διόδους προστασίας. Κατόπιν το L293D σε νεότερη παραγωγή περιέχει και τις διόδους προστασίας. Η L293 οικογένεια προσφέρει τις άριστες επιλογές για να οδηγηθούν μικρά διπολικά βηματικά μοτέρ που καταναλώνουν πάνω από ένα amp ανά περιέλιξη σε τάση τροφοδοσίας έως 36 βολτ. Το σχήμα 3.17 παρουσιάζει το εσωτερικό σχηματικό κοινό και για τα δυο ολοκληρωμένα L293B και L293D:



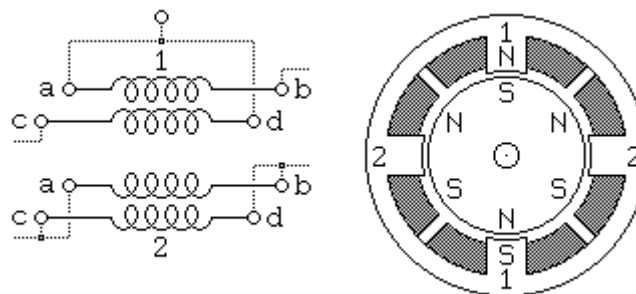
**Σχήμα 3.17:** Ολοκληρωμένο κύκλωμα για την οδήγηση βηματικών κινητήρων

Αυτό το ολοκληρωμένο μπορεί να φανεί ως τέσσερις ανεξάρτητες μισές γέφυρες - H, που ενεργοποιούνται ανά ζευγάρι, ή ως δύο πλήρεις γέφυρες - H.

### 3.1.5. Βηματικοί κινητήρες διπλού σύρματος

Οι περιελίξεις αυτών των κυκλωμάτων οδήγησης έχουν την ίδια γεωμετρία σε ρότορα και στάτη όπως σε ένα διπολικό κινητήρα, αλλά αντί του τυλίγματος κάθε σπείρας στο στάτη με ένα ενιαίο καλώδιο, υπάρχουν δύο καλώδια τυλιγμένα το ένα παράλληλα με το άλλο. Κατά συνέπεια, η μηχανή έχει οκτώ καλώδια, αντί τέσσερα.

Στην πράξη, οι μηχανές με διπλές περιελίξεις τροφοδοτούνται πάντα είτε ως μονοπολικοί είτε ως διπολικοί κινητήρες. Το σχήμα 3.18 παρουσιάζει εναλλακτικές συνδέσεις περιελίξεων ενός τέτοιου κινητήρα.

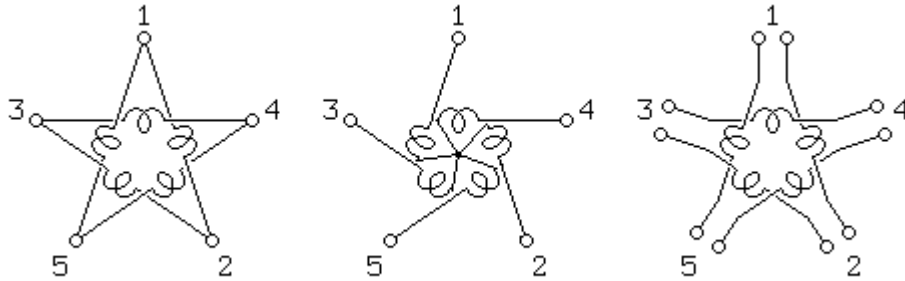


**Σχήμα 3.18:** Συνδεσμολογία περιελίξεων διπλού σύρματος βηματικών κινητήρων

Για να χρησιμοποιήσουμε ένα κινητήρα διπλού σύρματος ως μονοπολικό, τα δύο καλώδια κάθε περιελίξεως συνδέονται σε σειρά και το σημείο της σύνδεσης χρησιμοποιείται ως κεντρικός κόμβος με ακροδέκτη. Η περιέλιξη 1 στο σχήμα 3.18 παρουσιάζεται η συνδεσμολογία με αυτόν τον τρόπο. Για να χρησιμοποιήσουν ένα κινητήρα διπλού σύρματος ως διπολικό κινητήρα, τα δύο καλώδια κάθε περιελίξεως συνδέονται είτε παράλληλα είτε σε σειρά. Η Περιέλιξη 2 στο σχήμα 3.18 παρουσιάζεται με μια παράλληλη σύνδεση. Αυτό επιτρέπει χαμηλή τάση και υψηλό ρεύμα λειτουργίας. Η Περιέλιξη 1 στο σχήμα 3.18 παρουσιάζεται με μια σύνδεση σειράς. Εάν ο κεντρικός κόμβος αγνοείται, αυτό επιτρέπει τη

λειτουργία σε μια υψηλότερη τάση και ένα χαμηλότερο ρεύμα από ότι να συνδέονταν οι περιελίξεις παράλληλα.

**3.1.6. Πολυφασικοί βηματικοί κινητήρες**



*Σχήμα 3.19: Τρόποι συνδεσμολογίας των Πολυφασικών βηματικών κινητήρων*

Μια λιγότερο κοινή κατηγορία κυκλωμάτων οδήγησης κινητήρων με μόνιμο μαγνήτη και ένα καλώδιο για όλες τις περιελίξεις του κινητήρα σε μια κυκλική σειρά, αφήνοντας ένα κοινό ακροδέκτη μεταξύ κάθε ζευγαριού περιελίξεων στον κύκλο, ή με μόνο έναν ακροδέκτη κάθε περιελίξεως ασύνδετο οι άλλοι ακροδέκτες κάθε περιελίξεως συνδέονται σε μια απρόσιτη εσωτερική σύνδεση. Στα πλαίσια των τριφασικών κινητήρων, αυτές οι συνδέσεις θα περιγράφονταν ως συνδέσεις Delta και Y, αλλά χρησιμοποιούνται επίσης με τους κινητήρες 5 φάσεων, όπως απεικονίζεται στο σχήμα 3.19. Μερικοί πολυφασικοί κινητήρες εκθέτουν όλους τους ακροδέκτες όλων των περιελίξεων, αφήνοντας τα στο χρήστη να αποφασίσει μεταξύ των συνδέσεων Delta και Y, ή εναλλακτικά, επιτρέποντας κάθε περιέλιξη να οδηγηθεί ανεξάρτητα.

Ο έλεγχος είτε μια από αυτές τις πολυφασικές μηχανές είτε τη Delta είτε τη Y σύνδεση απαιτεί το 1/2 μιας γέφυρας – H για κάθε ακροδέκτη. Είναι αξιοσημείωτο ότι οι μηχανές 5 φάσεων έχουν τη δυνατότητα της απόδοσης περισσότερης ροπής.. Μερικές μηχανές 5 φάσεων έχουν πολύ μικρό βήμα 0,72 μοιρών ανά βήμα (500 βήματα ανά περιστροφή).

Με ένα κινητήρα 5 φάσεων, υπάρχουν 10 βήματα ανά επανάληψη σε κάθε κύκλο βήματος, όπως παρουσιάζονται παρακάτω:

```

Ακροδέκτης 1 +++-----+++++-----++
Ακροδέκτης 2 --+++++-----+++++----
Ακροδέκτης 3 +-----+++++-----+++++
Ακροδέκτης 4 +++++-----+++++-----
Ακροδέκτης 5 -----+++++-----+++++-
χρόνος ---->
    
```

Με ένα τριφασικό κινητήρα, υπάρχουν 6 βήματα ανά επανάληψη σε κάθε κύκλο βήματος, όπως παρουσιάζονται παρακάτω:

```

Ακροδέκτης 1 +++-----+----
Ακροδέκτης 2 --++++-----+--
Ακροδέκτης 3 +-----+----
χρόνος ---->
    
```

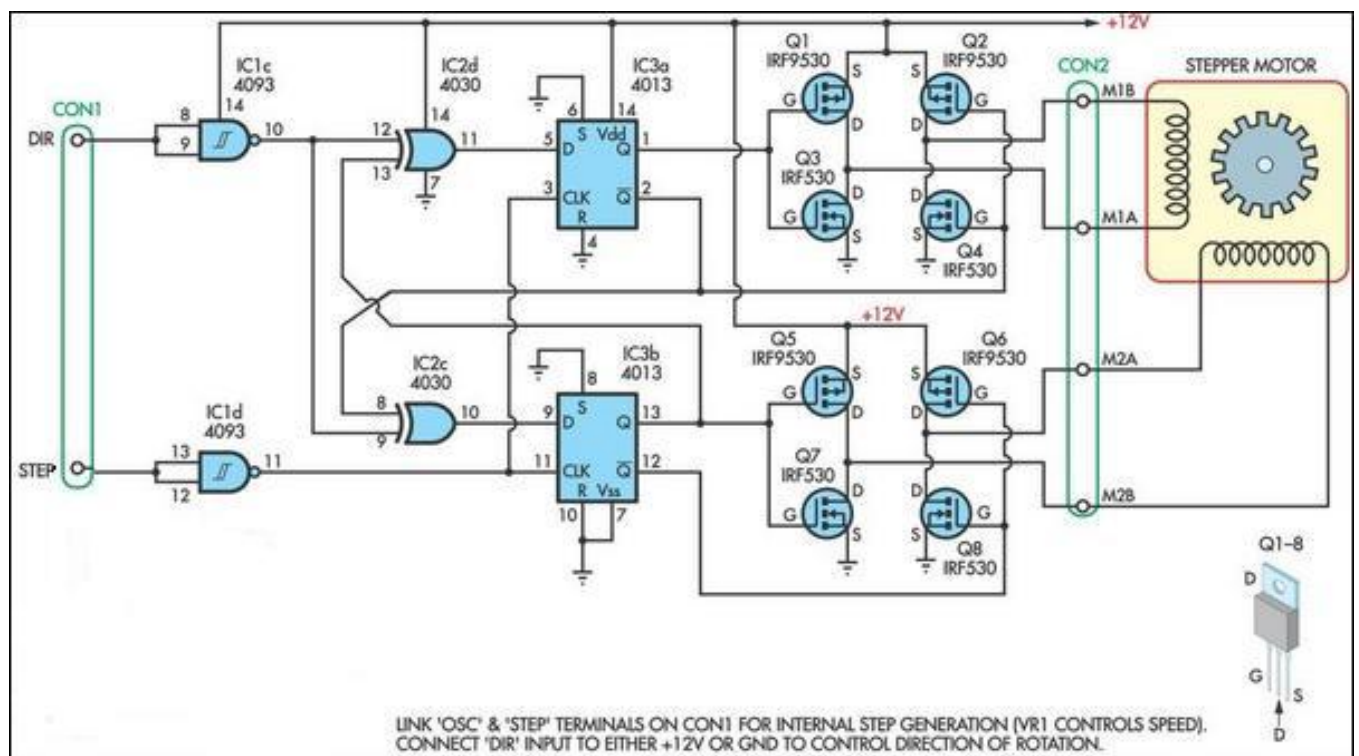
Εδώ, όπως στη περίπτωση διπολικού κινητήρα, κάθε ακροδέκτης φαίνεται να είναι συνδεδεμένος είτε με το θετικό είτε με το αρνητικό δυναμικό. Σημειώστε ότι, σε κάθε βήμα, μόνο ένας ακροδέκτης αλλάζει την πολικότητα. Λαμβάνοντας υπόψη τη γεωμετρία κινητήρων που απεικονίζονται από το σχήμα 3.19, αυτή η ακολουθία ελέγχου θα οδηγήσει το κινητήρα σε δυο περιστροφές.

Σε μερικούς κινητήρες 5 φάσεων έχουν 5 χωριστές περιελίξεις, με συνολικά 10 ακροδέκτες. Αυτοί μπορούν να συνδεθούν σε συνδεσμολογία αστέρα που απεικονίζεται παραπάνω, χρησιμοποιώντας 5 κυκλώματα μισών γεφυρών - H, ή κάθε περιέλιξη μπορεί να οδηγηθεί από μια πλήρης γέφυρα

### 3.2. Εργαστηριακή Εφαρμογή πάνω στους βηματικούς κινητήρες

## Κύκλωμα οδήγησης Διπολικού Βηματικού κινητήρα

#### 3.2.1. Ηλεκτρονικό σχέδιο κυκλώματος οδήγησης

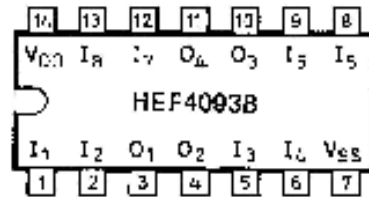
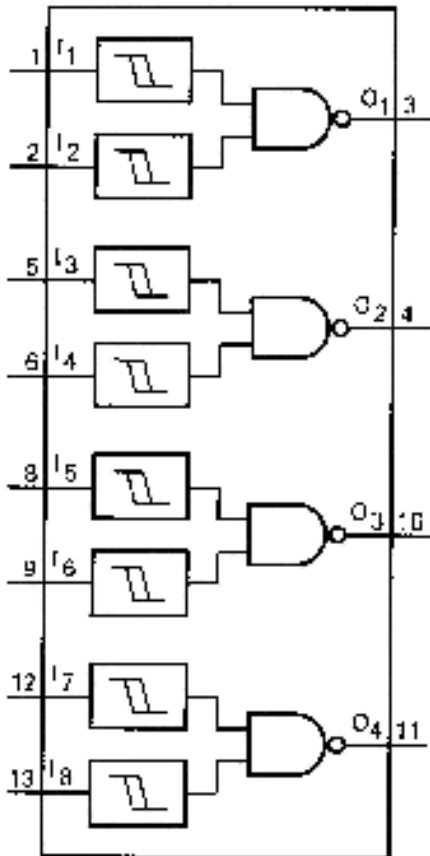


Εικόνα 3.20: ηλεκτρονικό σχέδιο κυκλώματος οδήγησης διπολικού βηματικού κινητήρα

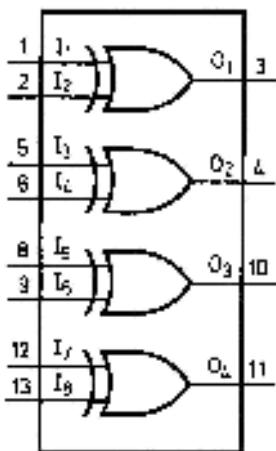
Το παραπάνω κύκλωμα οδήγησης βηματικού μοτέρ αποτελείται από δυο μέρη: α) το ψηφιακό μέρος του κυκλώματος και β) το αναλογικό μέρος του κυκλώματος. Το ψηφιακό αποτελείται από τις πύλες 2-Input Nand Smith Trigger (HEF4093B), τις πύλες exclusive OR (HEF4030B) και τα D-Flip Flop (HEF4013B) Το αναλογικό μέρος του κυκλώματος αποτελείται από τα Mosfets IRF530 και IRF9530. Αναλυτικά το εσωτερικό κύκλωμα του κάθε ολοκληρωμένου επεξηγήστε παρακάτω:

3.2.2. Datasheet ολοκληρωμένων κυκλωμάτων

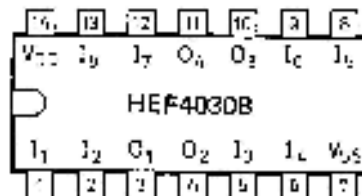
1) 2-Input Nand Smith Trigger (HEF4093B)



- Το εσωτερικό περιέχει 4 ανεξάρτητα κυκλώματα Nand Smith Trigger που το καθένα αποτελείται από μια πύλη Nand 2 εισόδων και κάθε είσοδος της πύλης διέρχεται από ένα Smith Trigger
- Τάση λειτουργίας του ολοκληρωμένου 5-15Volt
- Θερμοκρασία λειτουργία 25°C
- $V_{SS} = 0\text{ V}$



2) Exclusive OR (HEF4030B)



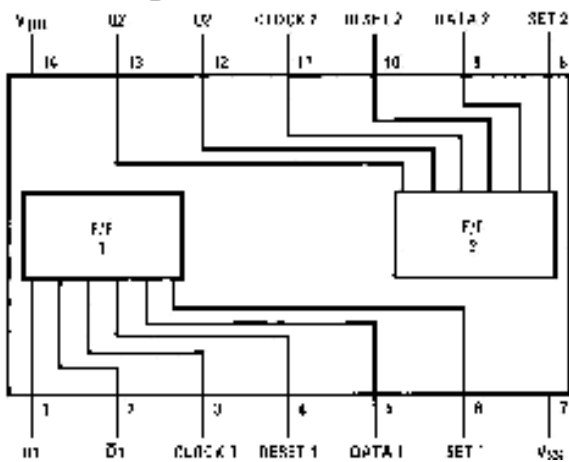
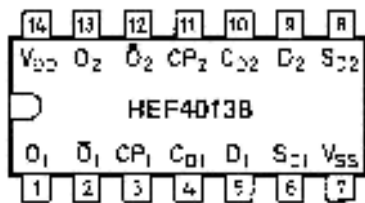
- Τέσσερις πύλες Exclusive-OR ανεξάρτητες μεταξύ τους
- Τάση λειτουργίας του ολοκληρωμένου 5-15Volt

- Οι εξόδοι είναι ηλεκτρικά πλήρως απομονωμένοι από κάθε είδους παράσιτα
- Θερμοκρασία λειτουργία 25°C
- $V_{SS} = 0\text{ V}$

**Πίνακας αληθείας :**

I <sub>1</sub>	I <sub>2</sub>	O <sub>1</sub>
L	L	L
H	L	H
L	H	H
H	H	L

**3) D-Flip Flop (HEF4013B)**

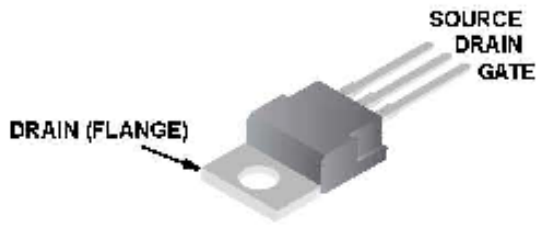


CL (Note 1)	D	R	S	Q	$\bar{Q}$
~	0	0	0	0	·
~	1	1	0	·	0
~	x	1	0	Q	Q
x	x	1	0	0	·
x	x	1	·	·	0
x	x	1	·	·	·

· No Change  
 x Don't Care Case  
 Note 1: level Change

- Αποτελείται από δυο ανεξάρτητα Flip Flop τύπου D
- Εύρος Τάσης τροφοδοσίας 3,0Volt έως 15Volt
- Υψηλή απομόνωση θορύβου
- Κατασκευασμένο από C-MOS κυκλώματα
- Εκτός τις εισόδους δεδομένων και τις εξόδους παρέχει και σήματα set, reset και τους παλμούς ρολογιού

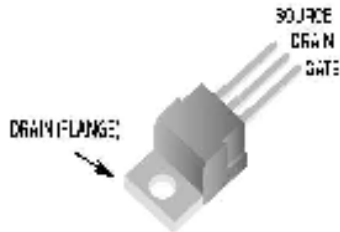
**4) N-Channel Power MOSFET IRF530**



**Χαρακτηριστικά:**

- Ένταση ρεύματος έως 14A και τάση έως 100Volt
- Χρόνος μετάβασης σε nanosecond
- Γραμμικά χαρακτηριστικά μεταφοράς
- Υψηλή απομόνωση εισόδου

**5) P-Channel Power MOSFET IRF9530.**



**Χαρακτηριστικά:**

- Ένταση ρεύματος έως 12A και τάση έως 100Volt
- Γραμμικά χαρακτηριστικά μεταφοράς
- Χρόνος μετάβασης σε nanosecond
- Υψηλή απομόνωση εισόδου

**3.2.3. Αρχές λειτουργίας Διπολικών βηματικών κινητήρων**

- α) Υπάρχουν δυο περιελίξεις στο εσωτερικό του κινητήρα και επομένως για τη λειτουργία του πρέπει να δημιουργηθεί μια διαφορά φάσης στα σήματα των κάθε περιελίξεων.
- β) Πρέπει να χρησιμοποιηθεί μια γέφυρα – Η επειδή τα διπολικά μοτέρ έχουν 4 ακροδέκτες σε σχέση με τους μονοπολικούς που έχουν 5. Η γέφυρα – Η βοηθά στην αντιστροφή των σημάτων ούτως ώστε να μπορεί να αλλάζει η φορά περιστροφής όποτε επιθυμούμε. Τα C-MOS ολοκληρωμένα συνδέονται ανά δυο σε συνδεσμολογία πύλης αντιστροφεία.
- γ) Τα C-MOS ολοκληρωμένα τοποθετούνται και για ένα δεύτερο λόγω, για ενίσχυση ρεύματος.
- δ) οι πύλες του 4030 και τα flip flop του 4013 είναι υπεύθυνα για τη διαφορά φάσης στις περιελίξεις.
- ε) Τα flip flop ανταποκρίνονται κατά την άνοδο του παλμού του ρολογιού.

**3.2.4. Πίνακας αληθείας για τη διέγερση διπολικών κινητήρων**

Πίνακας Αληθείας για δεξιόστροφη κατεύθυνση περιστροφής του κινητήρα δηλαδή DIRECTION συνδεδεμένο σε χαμηλό δυναμικό. Τα X στη πρώτη σειρά υποδηλώνουν τη τυχαία κατάσταση του flip flop στη αρχή της διαδικασίας. Η κατάσταση τώρα στα άκρα των περιελίξεων είναι αντίστοιχα  $\rightarrow M1A = Q1'$   $M1B = Q1$   $M2A = Q2$   $M2B = Q2'$

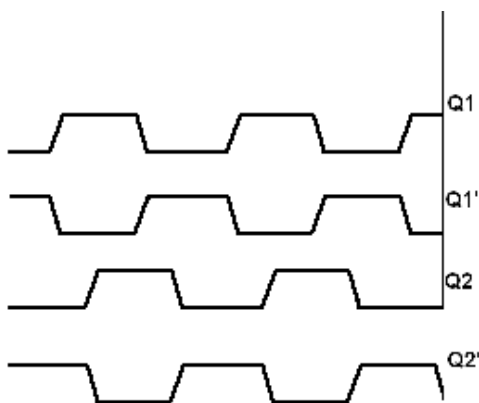
XOR1			XOR2			Flip flop 1			Flip flop 2		
I1	I2	Q1	I1	I2	Q1	D1	Q1	Q1'	D2	Q2	Q2'
1	X	X	X	1	X	X	0	1	X	0	1
1	0	1	1	1	0	1	1	0	0	0	1
1	0	1	0	1	1	1	1	0	1	1	0



1	1	0	0	1	1	0	0	1	1	1	0
1	1	0	1	1	0	0	0	1	0	0	1
1	0	1	1	1	0	1	1	0	0	0	1
1	0	1	0	1	1	1	1	0	1	1	0
1	1	0	0	1	1	0	0	1	1	1	0
1	1	0	1	1	0	0	0	1	0	0	1
1	0	1	1	1	0	1	1	0	0	0	1
1	0	1	0	1	1	1	1	0	1	1	0

Πίνακας Αληθείας για αριστερόστροφη κατεύθυνση περιστροφής του κινητήρα δηλαδή το DIRECTION συνδεδεμένο σε υψηλό δυναμικό. Τα X στη πρώτη σειρά υποδηλώνουν τη τυχαία κατάσταση του flip flop στη αρχή της διαδικασίας. Η κατάσταση τώρα στα άκρα των περιελίξεων είναι αντίστοιχα  $\rightarrow M1A = Q1'$   $M1B = Q1$   $M2A = Q2$   $M2B = Q2'$

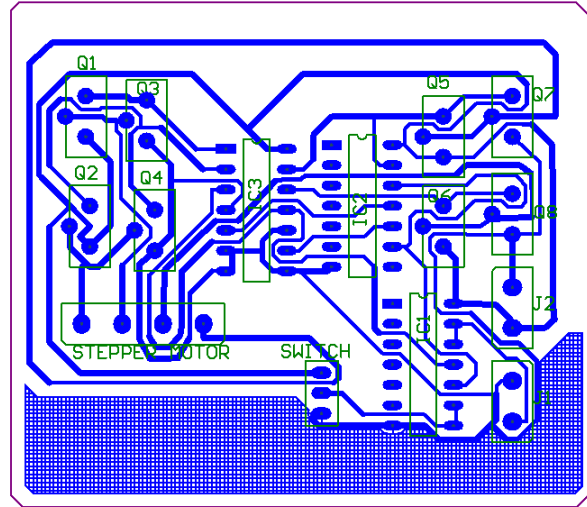
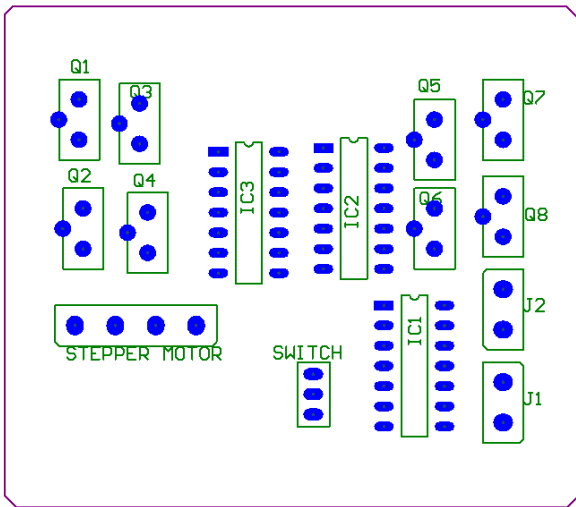
XOR1			XOR2			Flip flop 1			Flip flop 2		
I1	I2	Q1	I1	I2	Q1	D1	Q1	Q1'	D2	Q2	Q2'
0	X	X	X	0	X	X	0	1	X	0	1
0	0	0	1	0	1	0	0	1	1	1	0
0	1	1	1	0	1	1	1	0	1	1	0
0	1	1	0	0	0	1	1	0	0	0	1
0	0	0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1	1	0
0	1	1	1	0	1	1	1	0	1	1	0
0	1	1	0	0	0	1	1	0	0	0	1
0	0	0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1	1	0
0	1	1	1	0	1	1	1	0	1	1	0



Στο παραδίπλα σχέδιο βλέπουμε τη διαφορά φάσης που έχουν οι έξοδοι των flip flop. Κατά συνέπεια αυτή τη διαφορά φάσης θα έχουν και τα σήματα στους ακροδέκτες των περιελίξεων με αναφορά πάντα τη γείωση

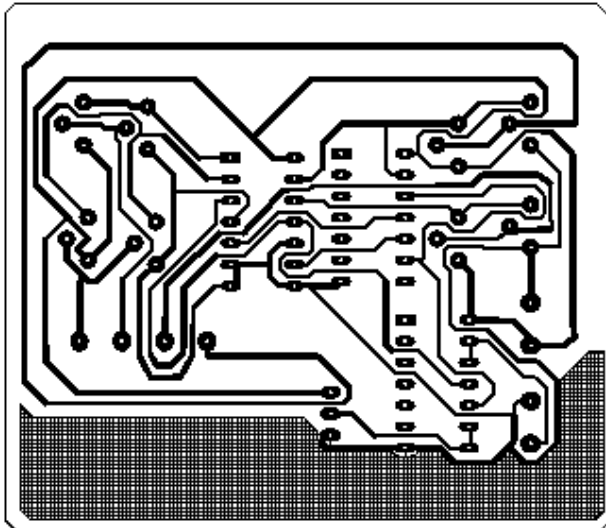
Εικόνα 3.21: Οι φάσεις των τολιγμάτων

3.2.5. Κατασκευή πλακέτας κυκλώματος οδήγησης



Εικόνα 3.22: κάτοψη εξαρτημάτων

Εικόνα 3.23: κάτοψη εξαρτημάτων μαζί με τους αγωγούς



Εικόνα 3.24: Τυπωμένο κύκλωμα

**Κατάλογος εξαρτημάτων**

- IC1 HEF4093B
- IC2 HEF4030B
- IC3 HEF4013B
- Q1, Q2, Q5, Q6 IRF9530
- Q3, Q4, Q7, Q8 IRF530

## 4. Εισαγωγή στη Visual Basic

Η Visual Basic είναι μια γλώσσα προγραμματισμού πολύ εύχρηστη και με πολύ απλοποιημένο περιβάλλον εργασίας. Με τη χρήση της Visual Basic μπορούν να γίνουν πολλές λειτουργίες. Από το σχεδιασμό της διεπαφής (Interface) προγράμματος μέχρι την επικοινωνία με τις πόρτες του Ηλεκτρονικού Υπολογιστή (Σειριακή, Παράλληλη, Universal Serial Bus). Υπάρχουν έτοιμα εργαλεία για την πιο εύκολη σχεδίαση μιας διεπαφής (Interface) όπως μετρητές, timers κτλ. καθώς και έτοιμες κωδικοποιημένες εντολές για πιο εύκολη ρύθμιση των ιδιοτήτων κάθε επιμέρους σχεδιασμένου εργαλείου.

### 4.1. Μελέτη της διεπαφής - Interface της πτυχιακής εργασίας

Η διεπαφή (Interface) της συγκεκριμένης εργασίας αποτελείται από διάφορα τμήματα. Το πρώτο τμήμα είναι το λογισμικό με βάση το οποίο θα δίνεται η εντολή για την περιστροφή ενός βηματικού κινητήρα κατά τόσα βήματα όσα εμείς επιθυμούμε ούτως ώστε μια κάμερα η οποία θα περιστρέφεται με το βηματικό κινητήρα να προσανατολιστεί εκεί που θέλουμε.

Το δεύτερο κομμάτι είναι ο υπολογισμός μιας στάθμης υγρών ενός δοχείου. Ο υπολογισμός της στάθμης φαίνεται εικονικά και με αναλογική μέτρηση και με ψηφιακή. (Το συγκεκριμένο εργαλείο μελετήθηκε και δημιουργήθηκε μόνο από λογισμικής πλευράς αφήνοντας βέβαια τη περίπτωση βελτίωσης στο μέλλον ούτως ώστε να επικοινωνεί με τη πλακέτα και να παίρνει πραγματικές τιμές μέτρησης από κάποιο δοχείο. Προς το παρόν ο χρήστης δίνει τη πληρότητα επί τις εκατό του δοχείου μέσα σε ένα εργαλείο κειμένου (Textbox) και έτσι φαίνεται η στάθμη στο εργαλείο αναλογικής μέτρησης στάθμης υγρών.

Το τρίτο κομμάτι είναι ένα θερμομέτρο επίσης με αναλογική και ψηφιακή μέτρηση. Θα παίρνει ψηφιακά τη θερμοκρασία από τη πλακέτα μέσω της σειριακής θύρας και ανά χρονικές περιόδους θα απεικονίζει τη θερμοκρασία στο κατασκευασμένο πρόγραμμα μέσω της οθόνης του υπολογιστή.

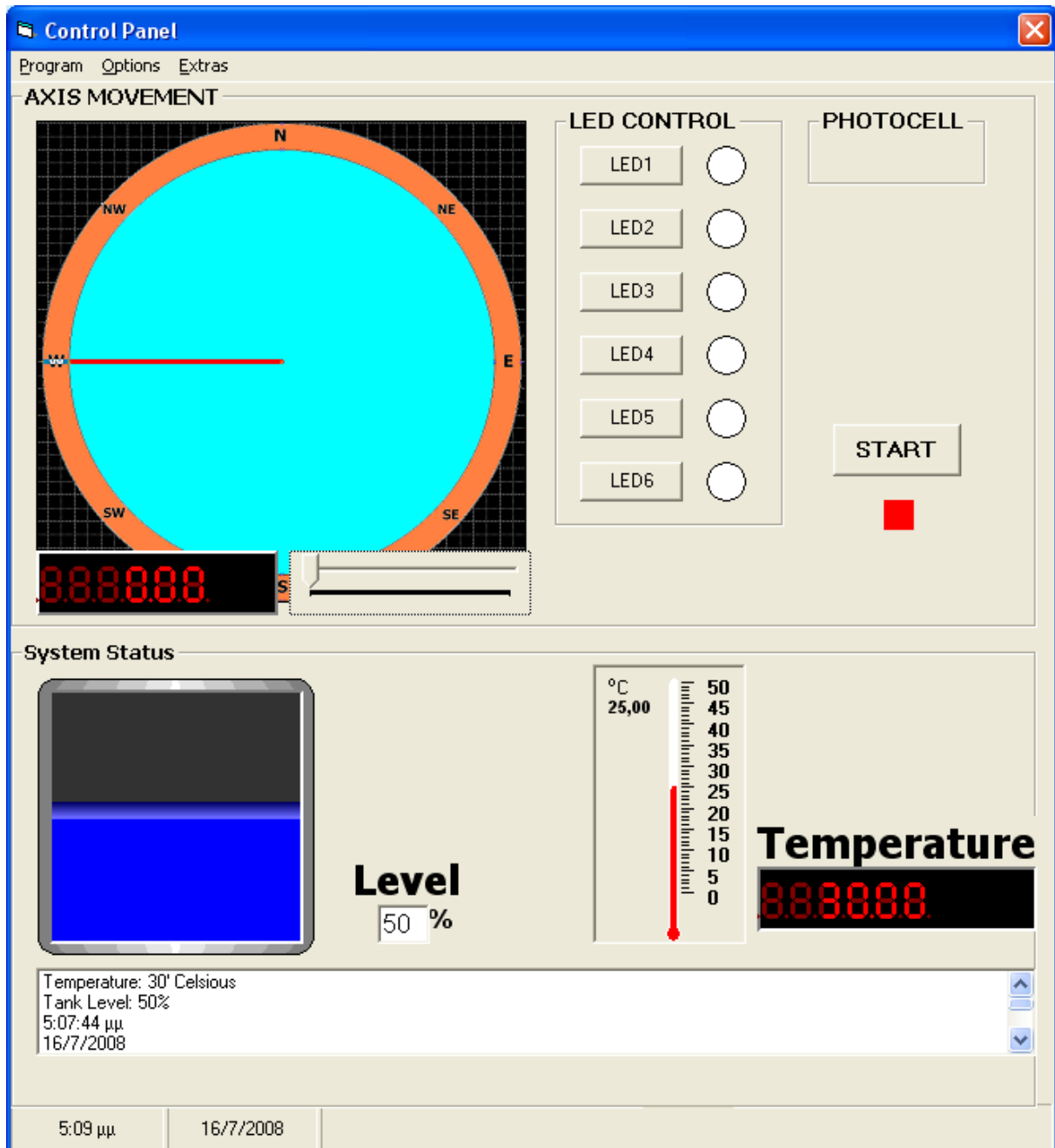
Το τέταρτο κομμάτι είναι κάποια κουμπιά που πατώντας τα θα ενεργοποιούνται κάποιες συσκευές όπως φωτισμός και ότι άλλο χρήσιμο για το χρήστη. Σε αυτό το κομμάτι υπάρχει και επιπρόσθετος χώρος στη διεπαφή (interface) για προσθήκη και άλλων συσκευών στο μέλλον. Στη θέση κάθε συσκευής εμείς έχουμε τοποθετήσει led.

Το πέμπτο και τελευταίο κομμάτι είναι ο χώρος ο οποίος συλλέγονται οι μετρήσεις θερμοκρασίας και στάθμης υγρών ανά τριάντα δευτερόλεπτα και αρχειοθετούνται όσο το πρόγραμμα είναι σε εκτέλεση μαζί με την ημερομηνία και ώρα λήψης τους. Αυτά είναι τα βασικά κομμάτια της βασικής φόρμας σχεδιασμού της διεπαφής.

Η εργασία αποτελείται από άλλες δυο φόρμες οι οποίες ενεργοποιούνται με το μενού που υπάρχει στη γραμμή εργαλείων της βασικής φόρμας. Η δεύτερη φόρμα και η τρίτη φόρμα όπως θα επεξηγηθεί παρακάτω είναι υπεύθυνες για τις ρυθμίσεις αυτοματοποιημένου ελέγχου της στάθμης υγρών και θερμοκρασίας αντίστοιχα.

## 4.2. Πρώτη φόρμα

Η πρώτη φόρμα την οποία θα μελετήσουμε είναι η φόρμα η οποία εμφανίζεται κατά την εκκίνηση του προγράμματος. Μέσω αυτής εμφανίζεται και η δεύτερη και η τρίτη φόρμα. Στην εικόνα 4.1 βλέπουμε ολόκληρη τη πρώτη φόρμα η οποία συγκεντρώνει όλες τις βαθμίδες που θα μελετήσουμε αναλυτικότερα παρακάτω.



Εικόνα 4.1: Η βασική φόρμα της διεπαφής - interface

### 4.2.1. Δήλωση μεταβλητών

Για να τρέξουμε το κώδικα μας πρέπει στην αρχή να ορίσουμε κάποιες μεταβλητές. Στις μεταβλητές αυτές θα αποθηκεύουμε δεδομένα που λαμβάνουμε από τη σειριακή θύρα όπως η θερμοκρασία και θα γίνονται κατόπιν οι κατάλληλοι υπολογισμοί. Παρακάτω ακολουθεί το κομμάτι του κώδικα γραμμένο σε Visual Basic το οποίο είναι υπεύθυνο για τη δήλωση των μεταβλητών (δήλωση ονόματος και τύπου μεταβλητής)

`Dim x As Integer` → μεταβλητή για το μετρητή που χρειάζεται για τον *timer 1*

`Dim Temp As Integer` → ακέραιος μεταβλητή η οποία αποθηκεύει την δυαδική λέξη η οποία προέρχεται από τη μέτρηση της θερμοκρασίας (έχοντας μετατραπεί από δυαδικό σε δεκαδικό)

`Dim Pulses As Integer` → ακέραιος μεταβλητή που αποθηκεύει τον αριθμό των βημάτων που πρέπει να εκτελέσει ο βηματικός κινητήρας

`Dim P As Boolean` → δυαδική μεταβλητή η οποία δίνει το σήμα *step* του βηματικού κινητήρα

`Dim Previous As Integer` → εδώ αποθηκεύεται η προηγούμενη θέση του βηματικού κινητήρα

`Dim Current As Integer` → εδώ αποθηκεύεται η θέση του βηματικού κινητήρα που θα λάβει

`Dim Left1 As Boolean` δυαδική → μεταβλητή που καθορίζει τη φορά περιστροφής του βηματικού κινητήρα

`Dim Buzzer As Boolean` → δυαδική μεταβλητή για τη κατάσταση του *Buzzer*

`Dim Led1 As Boolean` → δυαδική μεταβλητή για τη κατάσταση του πρώτου *led*. Ομοίως και για τα υπόλοιπα *led*

`Dim Led2 As Boolean`

`Dim Led3 As Boolean`

`Dim Led4 As Boolean`

`Dim Led5 As Boolean`

`Dim Led6 As Boolean`

`Dim Buzzer1 As Boolean` → δυαδική μεταβλητή για το θερμοκρασιακό έλεγχο

`Dim Buzzer2 As Boolean` → δυαδική μεταβλητή για τον αυτόματο έλεγχο στάθμης υγρών

`Dim Temp1 As Boolean` → δυαδική μεταβλητή για την αποθήκευση της ψηφιακής λέξης της θερμοκρασίας. Όμοιες μεταβλητές και οι υπόλοιπες 7 με λιγότερο σημαντικό ψηφίο το *Temp8*

`Dim Temp2 As Boolean`

`Dim Temp3 As Boolean`

`Dim Temp4 As Boolean`

`Dim Temp5 As Boolean`

`Dim Temp6 As Boolean`

`Dim Temp7 As Boolean`

`Dim Temp8 As Boolean`

`Dim Opto As Boolean` → δυαδική μεταβλητή για τη κατάσταση του φωτοκύτταρου

Οι επόμενες μεταβλητές έχουν δηλωθεί σε ένα ξεχωριστό αρχείο με όνομα *Module2*

`Public MaxTank As Integer` → ακέραιος μεταβλητή για τη τιμή της στάθμης που δεν πρέπει να υπερβεί

`Public MinTank As Integer` → ακέραιος μεταβλητή για τη τιμή της στάθμης ως η ελάχιστη

`Public MaxTemp As Single` → μεταβλητή για τη μέγιστη τιμή θερμοκρασίας ελέγχου

#### 4.2.2. Αρχικοποίηση των μεταβλητών

Με το που θα φορτώσουμε τη πρώτη φόρμα που είναι και η κεντρική (δηλαδή τη φόρμα που βλέπουμε πρώτα κατά την εκκίνηση του προγράμματος) θα τρέξει κάποιος κώδικας ο οποίος θα δώσει κάποιες αρχικές τιμές στις μεταβλητές οι οποίες έχουν δηλωθεί ως δημόσιες (Public). Αυτές οι τιμές είναι κάποιες τιμές τις οποίες θεωρούμε ως κανονικές και ο κάθε χρήστης εάν το επιθυμεί μπορεί να τις αλλάξει εισερχόμενος στις άλλες δυο φόρμες. Παρακάτω βλέπουμε το κώδικα:

```
Private Sub Form_Load()
MaxTank = 90
MinTank = 10
MaxTemp = 40
  x = 0
Pulses = 0
P = 0
XY_Slider.Value = 0
Current = 0
Previous = 0
End Sub
```

Με το παραπάνω κώδικα έχουμε λοιπόν την αρχικοποίηση των ορίων της στάθμης για τον αυτόματο έλεγχο της στάθμης υγρών καθώς και μηδενισμό των σημαντικών μεταβλητών του προγράμματος.

#### 4.2.3. Αποστολή και λήψη δεδομένων μέσω σειριακής θύρας

Εκτελώντας το πρόγραμμα της διεπαφής το επόμενο βήμα είναι να πατήσουμε το κουμπί με ετικέτα “start” (εικόνα 4.1). Πατώντας το εκτελείται αμέσως ένα κομμάτι κώδικα το οποίο είναι υπεύθυνο για το άνοιγμα της σειριακής θύρας. Επίσης διαβάζει συνεχώς (με το ρυθμό ενός timer) τα δεδομένα εισόδου. Στη προκειμένη περίπτωση τη θερμοκρασία και τη κατάσταση του φωτοκύτταρου. Όλα αυτά εκτελούνται ασχέτως αν κάνουμε οτιδήποτε άλλο στη διεπαφή (interface). Παρακάτω βλέπουμε το κώδικα που εκτελείται με το πάτημα του κουμπιού “start”:

```
Private Sub Command7_Click()
If Command7.Caption = "START" Then
  Command7.Caption = "STOP"
  StartTimer.Enabled = True
  Picture1.Visible = True
Else
  Command7.Caption = "START"
End If
End Sub
```

Πατώντας λοιπόν το “start” έχουμε τις εξής εργασίες μέσω ενός ελέγχου με την εντολή if. Η πρώτη εντολή αλλάζει την ετικέτα από “start” σε “stop” ούτως ώστε ο χρήστης να χρησιμοποιεί το κουμπί για να σταματήσει τη λειτουργία του προγράμματος. Κατόπιν στη δεύτερη εντολή ενεργοποιείται η λειτουργία του μετρητή (timer) με όνομα “StartTimer”. Κατόπιν εκτελούνται οι εντολές που ελέγχει ο που θα μελετήσουμε αμέσως μετά. Η τρίτη εντολή εμφανίζει ένα εικονικό εφέ σε σχήμα τριγώνου δίνοντας την αίσθηση ότι το πρόγραμμα τρέχει. Σε αντίθετη περίπτωση που δε πατήσουμε το start (else) η ετικέτα του κουμπιού παραμένει με όνομα “start”

Όταν λοιπόν ενεργοποιηθεί η λειτουργία του μετρητή με όνομα StartTimer έχουμε την εκτέλεση του αντίστοιχου κώδικα:

```
Private Sub StartTimer_Timer()
If Command7.Caption = "STOP" Then
    Pulses = Abs(Current - Previous) * 2
    If Current < Previous Then Left1 = True Else Left1 = False
    Previous = Current
    OPENCOM ("COM1,1200,N,8,1")
    RTS 0
    x = 24
    Timer1.Enabled = True
Else
    Picture1.Visible = False
End If
StartTimer.Enabled = False
End Sub
```

Μόλις λοιπόν ενεργοποιηθεί ο StartTimer η χρονική διάρκεια όλων των εντολών είναι 1 δευτερόλεπτο (μέσα στις ιδιότητες του μετρητή (timer) ορίζουμε το χρόνο επανάληψης (interval) σε 1000ms. Στη διάρκεια του ενός δευτερολέπτου λοιπόν γίνεται επικοινωνία με τη σειριακή θύρα (αποστολή και λήψη δεδομένων) και η διαδικασία επαναλαμβάνεται διαρκώς και στα επόμενα δευτερόλεπτα εφόσον η ετικέτα του κουμπιού είναι “stop” δηλαδή προηγουμένως πατήθηκε το “start” (έλεγχος if στη δεύτερη εντολή). Στη τρίτη γραμμή κώδικα γίνεται ο υπολογισμός των βημάτων που θα εκτελέσει ο βηματικός κινητήρας για να μεταβεί στη νέα του θέση. Κάνουμε αφαίρεση της θέσης που βρίσκεται από τη θέση που πρέπει να πάει ο κινητήρας. Επίσης πολλαπλασιάζουμε με το δυο γιατί ο βηματικός κινητήρας κάνει βήμα μόνο στο θετικό παλμό του σήματος γι αυτό θέλουμε διπλάσιες μεταβάσεις. Η επόμενη εντολή κάνει έλεγχο ούτως ώστε αν κινούμε το slider προς το δεξιά αντίστοιχα και ο βηματικός κινητήρας να στραφεί δεξιά. Αλλιώς σε αντίθετη περίπτωση να κινηθεί αριστερά. Στη τέταρτη εντολή η νέα θέση που έχει λάβει ο κινητήρας θεωρείται προηγούμενη για ξαναγίνει πάλι έλεγχος στον επόμενο κύκλο του μετρητή starttimer. Η επόμενη εντολή είναι για το άνοιγμα της σειριακής θύρας και συγκεκριμένα της COM1 καθώς και κάποια ορίσματα για το ρυθμό μετάδοσης κ.τ.λ. Στην επόμενη εντολή έχουμε μηδενισμό του σήματος RTS (υπεύθυνο για το σήμα του βήματος - STEP του κινητήρα) γιατί σε αντίθετη περίπτωση ο κινητήρας αρχικά θα έκανε ένα παραπάνω βήμα και θα χάναμε την αρχική θέση του βηματικού κινητήρα. Κατόπιν ορίζουμε μια μεταβλητή X ως μετρητής δεδομένων και δίνουμε τιμή το 24. Ο αριθμός αυτός έχει να κάνει με 24 bit. 8 bit είναι τα δεδομένα, 8 bit είναι οι θετικοί παλμοί ρολογιού, και 8 bit οι αρνητικοί παλμοί ρολογιού (αφού μια περίοδος ρολογιού έχει και θετική ακμή και αρνητική). Στην επόμενη εντολή ενεργοποιείται ο επόμενος μετρητής (timer) με ετικέτα timer1 ο οποίος είναι υπεύθυνος για την αποστολή δεδομένων όπως θα αναλύσουμε παρακάτω. Αν το κουμπί γράφει ετικέτα “start” τότε στις δυο τελευταίες εντολές έχουμε την εμφάνιση της εικόνας με το τριγωνικό σύμβολο του τριγώνου (play button) καθώς και απενεργοποίηση του μετρητή startTimer ούτως ώστε να μη γίνεται αποστολή και λήψη δεδομένων.

Όταν ο μετρητής Timer 1 ενεργοποιηθεί εκτελεί τι κώδικά του με περίοδο 30 ms. Για κάθε τιμή του x (του οποίου δώσαμε τιμή 24) γίνεται μια διαδικασία ανάγνωση και αποστολής δεδομένων. Ας δούμε αναλυτικά παρακάτω:

```
Private Sub Timer1_Timer()
If x = 0 Then
    Opto = CTS()
    Delay.Enabled = True
    Timer1.Enabled = False
```

```
End If

If x = 1 Then
    Temp8 = DSR()
DTR 0
    x = x - 1
End If

    If x = 2 Then
DTR 1
        x = x - 1
End If

        If x = 3 Then
Temp7 = DSR()
            If Led6 = True Then TXD 1 Else TXD 0
            x = x - 1
End If

                If x = 4 Then
DTR 0
                    x = x - 1
End If

                    If x = 5 Then
DTR 1
                        x = x - 1
End If

                        If x = 6 Then
Temp6 = DSR()
                            If Led5 = True Then TXD 1 Else TXD 0
                            x = x - 1
End If

                                If x = 7 Then
DTR 0
                                    x = x - 1
End If

                                    If x = 8 Then
DTR 1
                                        x = x - 1
End If
                                        If x = 9 Then
Temp5 = DSR()
                                            If Led4 = True Then TXD 1 Else TXD 0
                                            x = x - 1
End If

                                                If x = 10 Then
DTR 0
                                                    x = x - 1
End If

                                                        If x = 11 Then
DTR 1
                                                            x = x - 1
End If

                                                                If x = 12 Then
```



```
Temp4 = DSR()
If Led3 = True Then TXD 1 Else TXD 0
x = x - 1
End If

If x = 13 Then
DTR 0
x = x - 1
End If

If x = 14 Then
DTR 1
x = x - 1
End If

If x = 15 Then
Temp3 = DSR()
If Led2 = True Then TXD 1 Else TXD 0
x = x - 1
End If

If x = 16 Then
DTR 0
x = x - 1
End If

If x = 17 Then
DTR 1
x = x - 1
End If

If x = 18 Then
Temp2 = DSR()
If Led1 = True Then TXD 1 Else TXD 0
x = x - 1
End If

If x = 19 Then
DTR 0
x = x - 1
End If

If x = 20 Then
DTR 1
x = x - 1
End If

If x = 21 Then
Temp1 = DSR()
If Buzzer = True Then TXD 1 Else TXD 0
x = x - 1
End If

If x = 22 Then
DTR 0
x = x - 1
End If

If x = 23 Then
DTR 1
x = x - 1
```

```

End If

If x = 24 Then
If Left1 = True Then TXD 1 Else TXD 0
x = x - 1
End If

```

Για  $x=0$  έχουμε τις εξής εκτελέσεις: ανάγνωση και αποθήκευση της κατάστασης του φωτοκύτταρου το οποίο είναι συνδεδεμένο στο ακροδέκτη CTS της σειριακής θύρας. Έχουμε ενεργοποίηση του μετρητή Delay Timer και απενεργοποίηση του μετρητή timer1. Εκεί έχει σταματήσει πια η μετάδοση των δεδομένων. Σε προηγούμενο κομμάτι κώδικα το  $x$  αρχικοποιείται στη τιμή 24. άρα ο κώδικας ξεκινά για συνθήκη που το  $x$  είναι 24. Κάθε επανάληψη (loop) η οποία εκτελείται έχουμε και μείωση της τιμής του  $x$ . Αυτό το κάνουμε για να μην εκτελούνται όλες οι συνθήκες if ταυτόχρονα αλλά με φθίνουσα σειρά της τιμής του  $x$ . Ξεκινώντας λοιπόν για  $x=24$  έχουμε τα εξής. Ελέγχουμε τη κατάσταση του βηματικού μοτέρ από τη διεπαφή (Interface). Δηλαδή αν τον στρέφουμε δεξιά ή αριστερά. Ανάλογα αν είναι λογικό 1 ή λογικό 0 το στέλνουμε στη λειτουργική κάρτα στο καταχωρητή σειριακής φόρτωσης μέσω του ακροδέκτη TxD της σειριακής θύρας. Μειώνουμε τη τιμή του  $x$ . Για  $x=23$  και για  $x=22$  στέλνουμε το θετικό και τον αρνητικό παλμό αντίστοιχα μέσω του ακροδέκτη DTR της σειριακής θύρας. Ο ακροδέκτης DTR είναι για τη δημιουργία του ρολογιού των ψηφιακών κυκλωμάτων.

Στη συνέχεια για  $x=21$  ελέγχουμε μέσω προγράμματος τη κατάσταση που έχει στη διεπαφή (interface) το βομβητής (buzzer). Αποστέλλουμε τη κατάσταση του στη κάρτα και κατόπιν για  $x=20$  και για  $x=19$  στέλνουμε το παλμό του ρολογιού. Σε αυτή την επανάληψη (loop) στέλνουμε στον υπολογιστή από τη λειτουργική κάρτα το πρώτο bit της θερμοκρασίας (η θερμοκρασία απεικονίζεται από οκταδική ψηφιακή λέξη) το οποίο το λαμβάνεται και αποθηκεύεται στη μεταβλητή της διεπαφής (interface) Temp1. Η θερμοκρασία μεταφέρεται μέσω του ακροδέκτη DSR της σειριακής θύρας. Κατόπιν το  $x$  θα έχει τιμή 18. Ωρα είναι να ελέγξουμε από τη διεπαφή τη κατάσταση του πρώτου led και να το στείλουμε στη λειτουργική κάρτα. Επίσης να λάβουμε στη διεπαφή το δεύτερο bit της θερμοκρασίας και να καταχωρηθεί στη μεταβλητή Temp2. Για  $x=17$  και για  $x=16$  στέλνουμε το παλμό του ρολογιού.

Έτσι λοιπόν για τις υπόλοιπες τιμές του  $x$  έχουμε την αποστολή στη λειτουργική κάρτα και τις καταστάσεις των υπόλοιπων led (led2-led6) και τα υπόλοιπα έξι bit της ψηφιακής λέξης για τη θερμοκρασία. Μόλις ολοκληρωθούν θα έχουμε στο  $x$  τη τιμή 0. Τότε απενεργοποιείται ο μετρητής timer1 και ενεργοποιείται ο μετρητής delay timer. Ο timer1 θα ενεργοποιηθεί ξανά όταν πάρει την εντολή από το μετρητή StartTimer όπως είπαμε προηγουμένως. Όλη αυτή η διαδικασία διαρκεί 30 ms.

Παρακάτω θα δούμε το κώδικα και την επεξήγηση για το μετρητή Delay Timer:

```

Private Sub Delay_Timer()
If Pulses > 0 Then
Pulses = Pulses - 1
If P = True Then P = False Else P = True
If P = True Then RTS 1 Else RTS 0
End If
If Pulses < 1 Then
Delay.Enabled = False
CLOSECOM
Temp = 0
If Temp1 = True Then Temp = Temp + 128
If Temp2 = True Then Temp = Temp + 64
If Temp3 = True Then Temp = Temp + 32
If Temp4 = True Then Temp = Temp + 16

```

```

If Temp5 = True Then Temp = Temp + 8
If Temp6 = True Then Temp = Temp + 4
If Temp7 = True Then Temp = Temp + 2
If Temp8 = True Then Temp = Temp + 1
Thermo.Value = Int(((500 / 255) * Temp) - 273)
PanelTemp.Text = Thermo.Value
StartTimer.Enabled = True
End If
End Sub

```

Ο μετρητής Delay Timer ξεκινάει να εκτελεί το κώδικα του. Οι εντολές του είναι για την αποστολή των παλμών για τη λειτουργία του βηματικού κινητήρα και επίσης ο υπολογισμός της θερμοκρασίας σε °C μέσω της ψηφιακής λέξης που έχει αποσταλθεί από τον αισθητήρα θερμότητας.

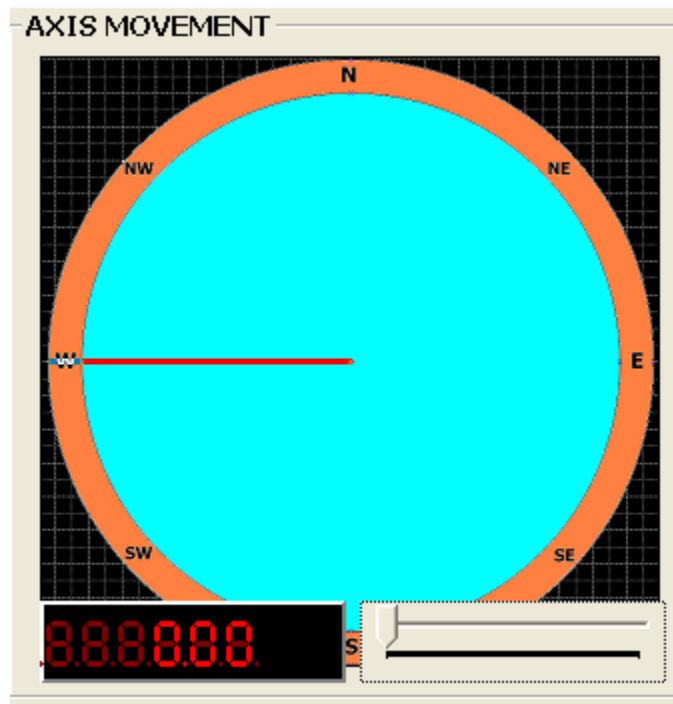
Η πρώτη επανάληψη (loop) είναι υπεύθυνη για τη δημιουργία των παλμών για τη λειτουργία του βηματικού κινητήρα. Ο ακροδέκτης RTS της σειριακής θύρας στέλνει τους παλμούς στην είσοδο παλμών του κυκλώματος οδήγησης βηματικού κινητήρα που περιγράψαμε στο τρίτο κεφάλαιο. Μόλις λοιπόν εμείς μετακινήσουμε τη ράβδο (slider) τότε δίνουμε στη μεταβλητή Pulses κάποιο αριθμό παλμών ανάλογα με το πόσο πολύ μεταβάλαμε τη ράβδο (slider). Για κάθε παλμό εκτελείται μια επανάληψη (loop). Πρώτον στέλνεται ένας παλμός μέσω του ακροδέκτη RTS και κατόπιν μειώνεται κατά 1 η μεταβλητή pulses. Μόλις η μεταβλητή pulses γίνει μηδέν τότε σημαίνει ότι όλοι οι απαιτούμενοι παλμοί έχουν κινηθεί το βηματικό κινητήρα στο επιθυμητό σημείο. Αν η μεταβλητή pulses γίνει μικρότερη του μηδενός τότε σημαίνει ότι όλοι οι παλμοί εκτελέστηκαν και η επόμενη εντολή είναι να απενεργοποιηθεί ο ο μετρητής Delay Timer. Κατόπιν κλείνει και η επικοινωνία με τη σειριακή θύρα με την εντολή “CLOSECOM”. Μετά μηδενίζεται η τιμή του θερμομέτρου στη διεπαφή για να λάβει τη νέα τιμή της θερμοκρασίας από τον υπολογισμό που γίνεται στις επόμενες εντολές του μετρητή delay Timer.

Οι επόμενες εντολές είναι για μετατροπή της ψηφιακής λέξης της θερμοκρασίας σε °C. Οι επόμενες λοιπόν 8 εντολές λαμβάνουν το δυαδικό bit από τις μεταβλητές temp1 έως temp8 που είδαμε σε προηγούμενο κώδικα κατά τη μεταφορά δεδομένων από τη κάρτα προς τον υπολογιστή. Οι εντολές αυτές το μόνο που κάνουν είναι να μετατρέπουν το δυαδικό αριθμό σε δεκαδικό. Στην επόμενη εντολή η τιμή του θερμομέτρου (εργαλείο thermo) υπολογίζεται από την ακόλουθη σχέση. Ο αισθητήρας του LM335 έχει την εξής ιδιότητα. Κάθε 10 m Volt αντιστοιχούν σε 1 βαθμό Kelvin. Οπότε στη σχέση αυτή γίνεται μέθοδος των τριών και βρίσκουμε τη θερμοκρασία για την αντίστοιχη τάση σε βαθμούς Kelvin. Κατόπιν στην ίδια σχέση αφαιρούμε τον αριθμό 273 για να έχουμε τη τιμή της θερμοκρασίας σε βαθμούς Κελσίου. Η επόμενη εντολή αντιγράφει τη τιμή από το αναλογικό θερμομέτρο στη διεπαφή (interface) και στη ψηφιακή οθόνη. Η τελευταία εντολή κάνει ξανά ενεργό το μετρητή StartTimer για να αρχίσει ξανά η διαδικασία της αποστολής και λήψης δεδομένων μέσω σειριακής θύρας.

#### 4.2.4. Προσανατολισμός ενός αντικειμένου

Στη παρακάτω εικόνα φαίνεται το κομμάτι του προγράμματος που με αυτό μπορούμε να ρυθμίσουμε τη θέση μιας κάμερας σε μοίρες. Το πρόγραμμα θα στέλνει σήματα προς τη σειριακή θύρα με αποτέλεσμα ο βηματικός κινητήρας να περιστρέφεται όσα βήματα θέλουμε.

Ο σχεδιασμένος κύκλος είναι για τη περιστροφή της κάμερας γύρω από τον άξονά της. Επίσης μας δείχνει σε σχέση με το βορά προς τα πού κατευθύνεται η κάμερα ούτως ώστε να μπορούμε μακριά από τη κάμερα να τη κατευθύνουμε εκεί που θέλουμε.



**Εικόνα 4.2:** Τμήμα της διεπαφής για τη ρύθμιση της κάμερας.

Όπως βλέπουμε και στην εικόνα 4.2 με τη χρήση της μεταβαλλόμενης ράβδου περιστρέφεται και ο δείκτης στο αντίστοιχο σημείο. Επίσης αυτόματα η νέα κατεύθυνση απεικονίζεται και στη ψηφιακή ένδειξη σε μοίρες. Στο κύκλο η τιμή σε μοίρες μεταβάλλεται από 0 σε 360°. Αυτό το εύρος καθορίζεται από τις ιδιότητες του εργαλείου slider (μεταβαλλόμενη ράβδος). Αναλυτικότερα θα το καταλάβουμε και στην επεξήγηση του κώδικα και ιδιότητες για το αντίστοιχο εργαλείο. Ο επόμενος κώδικας είναι για τη μεταβλητή ράβδο (slider):

### **Ο κώδικας:**

```
Private Sub XY_Slider_KeyUp(KeyCode As Integer, Shift As Integer)
Current = XY_Slider.Value
End Sub
```

Με τις παραπάνω εντολές γίνονται οι εξής διεργασίες:

Το συγκεκριμένο εργαλείο (slider) μετακινείται αριστερά και δεξιά με τη χρήση των βελών του πληκτρολογίου. Με τις εντολές λοιπόν αυτές ανάλογα με τη διάρκεια που πατάμε το βέλος τότε και ο slider κινείται προς την αντίστοιχη κατεύθυνση του βέλους. Στην επόμενη εντολή η μεταβλητή Current λαμβάνει τη τιμή του slider. Η μεταβλητή current όπως είπαμε στέλνει το βηματικό κινητήρα στη νέα του θέση. Στον επόμενο κώδικα βλέπουμε κάποιες εντολές που κάνουν τις ίδιες ενέργειες με τις προηγούμενες αλλά με τη χρήση του ποντικιού.

```
Private Sub XY_Slider_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
Current = XY_Slider.Value
End Sub
```

Στο επόμενο κομμάτι κώδικα βλέπουμε τις υπόλοιπες ενέργειες που γίνονται κατά το σύρσιμο της ράβδου (slider) είτε δεξιά είτε αριστερά.

```

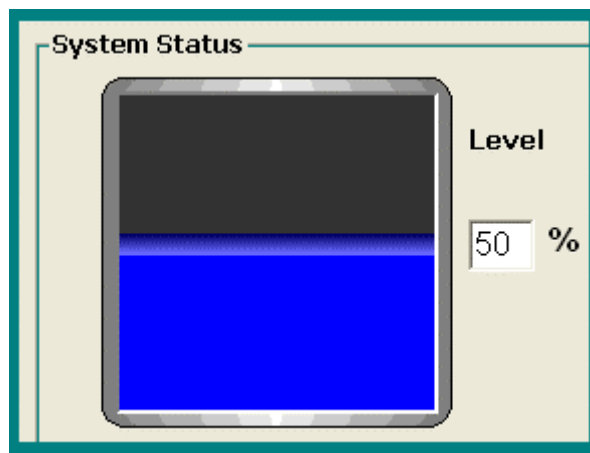
Private Sub XY_Slider_Scroll()
Azimuth.Value = XY_Slider.Value * 1.8
PanelAzim.Text = Int(XY_Slider.Value * 1.8)
End Sub

```

Καθώς λοιπόν μεταβάλλεται η οριζόντια μπάρα (slider), ο αναλογικός δείκτης κατεύθυνσης παίρνει τη τιμή της ράβδου. Πολλαπλασιάζουμε όμως με 1.8 ούτως ώστε να ο δείκτης να ακολουθεί ακριβώς τη ράβδο. Δηλαδή όταν η ράβδος φτάνει στο τέλος τότε και ο δείκτης να έχει γυρίσει 360°. Η δεύτερη εντολή αντιγράφει τη τιμή των μοιρών και στη ψηφιακή οθόνη (panel). (αυτή όμως σε ακέραιη μορφή)

#### 4.2.5. Μέτρηση στάθμης υγρών

Το τμήμα αυτό της διεπαφής (Interface) σχεδιάστηκε για την απεικόνιση της στάθμης υγρών σε έναν ηλεκτρονικό υπολογιστή. Το τμήμα αυτό βέβαια είναι όπως προαναφέρθηκε κατασκευασμένο μόνο από λογισμικής πλευράς. Για την ώρα δεν θα λειτουργήσει με εξωτερικό αισθητήρα για την απόδοση της στάθμης ενός πραγματικού δοχείου υγρών. Βέβαια έχει τη δυνατότητα να γίνει αυτή η επέκταση και σε επίπεδο υλικού (Hardware) μελλοντικά. Επίσης έχει μέσω προγράμματος και λειτουργίες ούτως ώστε να γίνεται αυτοματοποιημένα παρακολούθηση της στάθμης. Πιο συγκεκριμένα η επόμενη εικόνα δείχνει λεπτομερειακά το τμήμα αυτό.



**Εικόνα 4.3:** Τμήμα της διεπαφής για την απεικόνιση της στάθμης των υγρών

Όπως φαίνεται και στην εικόνα η στάθμη απεικονίζεται με το μπλε χρώμα και επίσης σε ένα διπλανό κουτί υπολογίζεται η πληρότητα του δοχείου επί της εκατό. Το τμήμα αυτό μέσω λογισμικού έχει επίσης τη δυνατότητα να οριστεί ένα άνω και ένα κάτω όριο ελέγχου επί τις εκατό με αποτέλεσμα όταν η στάθμη βρεθεί εκτός των ορίων αυτών τότε κάποιος συναγερμός θα ενεργοποιείται για να μας ειδοποιήσει. Οπότε εμείς ανάλογα με το πόσο υγρό θέλουμε να έχει η δεξαμενή τότε ανάλογα ρυθμίζουμε αυτά τα όρια. Τα όρια αυτά ρυθμίζονται μέσω της δεύτερης φόρμας που θα δούμε στο υποκεφάλαιο της δεύτερης φόρμας.

Στο εργαλείο κειμένου (Textbox) που βρίσκεται δίπλα από την αναλογική απεικόνιση της στάθμης αν βάλουμε το δείκτη του ποντικιού και πατήσουμε αριστερό κλικ τότε ο

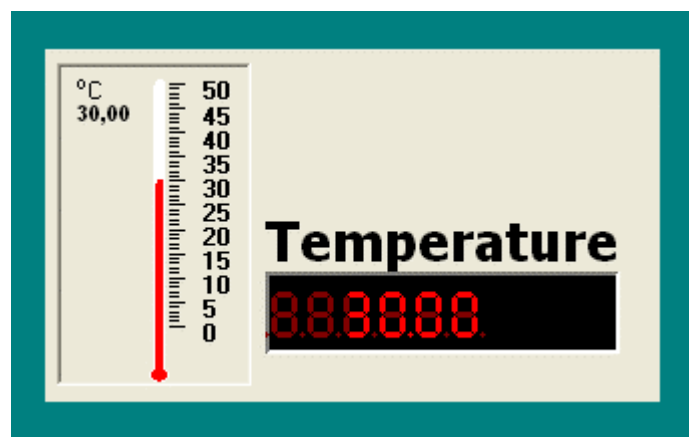
κέρσορας θα αναβοσβήνει. Κατόπιν με τη χρήση του πληκτρολογίου δίνουμε με αριθμό επί της εκατό τη πληρότητα του δοχείου που επιθυμούμε και πατώντας enter αυτόματα η στάθμη του υγρού ανεβαίνει ανάλογα με τη τιμή που δίνουμε. Η υπορουτίνα Select Case κάνει αυτά που αναφέραμε. Δηλώνουμε και το enter ως κουμπί με το οποίο πατώντας το αυτόματα η τιμή που βρίσκεται στο κουτί κειμένου (textbox) να αντιγράφεται στη τιμή (value) του αναλογικού δείκτη στάθμης.

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
Select Case KeyCode
Case vbEnter
Tank.Level = Val(Text1.Text)
End Select
End Sub
```

Με το παραπάνω κώδικα λοιπόν όταν πληκτρολογώ κάτι στο πλαίσιο text1 (πρώτη εντολή) και το πρόγραμμα ελέγχει ποιο πλήκτρο του πληκτρολογίου πάτησα (δεύτερη εντολή) και αν πάτησα το enter (τρίτη εντολή), τότε η στάθμη της δεξαμενής μεταβαίνει στη ποσότητα του αριθμού που έχουμε βάλει στο πλαίσιο (τέταρτη εντολή)

#### 4.2.6. Μέτρηση της θερμοκρασίας

Το τμήμα αυτό της διεπαφής (Interface) σχεδιάστηκε για τη μέτρηση της θερμοκρασίας. Από πλευράς υλικού (hardware) θα υπάρχει αισθητήρας πάνω στη λειτουργική κάρτα και μαζί με ένα αναλογικό σε ψηφιακό μετατροπέα (A/D Converter) μετατρέπεται ανά τακτά χρονικά διαστήματα η θερμοκρασία σε μια ψηφιακή λέξη η οποία και στέλνεται μέσω σειριακής θύρας στον υπολογιστή. Κατόπιν από λογισμικής πλευράς ο υπολογιστής με βάση τις πολλαπλές στάθμες που θα εισέρχονται από τη σειριακή θύρα ανάλογα τη θερμοκρασία (άρα διαφορετική ψηφιακή λέξη σε κάθε μεταβολή της θερμοκρασίας και συνεπώς της τάσης) θα μεταβάλλεται και ο αναλογικός δείκτης της θερμοκρασίας. Μια περαιτέρω εξέλιξη είναι και μια ψηφιακή οθόνη κατασκευασμένη στο περιβάλλον της Visual Basic η οποία θα απεικονίζει και ψηφιακά τη θερμοκρασία σε βαθμούς Κελσίου. Η παρακάτω εικόνα μας επεξηγεί περισσότερο.



Εικόνα 4.4: Τμήμα της διεπαφής για την απεικόνιση της θερμοκρασίας

Το εργαλείο αυτό που απεικονίζει τη θερμοκρασία έχει κάποιες ιδιότητες τις οποίες εμείς μπορούμε να μεταβάλλουμε. Για παράδειγμα το θερμόμετρο έχει ρυθμιστεί στη συγκεκριμένη περίπτωση να μετράει από 0 βαθμούς έως 50 βαθμούς Κελσίου. Εμείς μπορούμε να δώσουμε ένα εύρος που εμείς επιθυμούμε. Τέλος η τιμή του συγκεκριμένου θερμόμετρου μεταφέρεται με τη βοήθεια κώδικα και στη ψηφιακή οθόνη που βρίσκεται ακριβώς δίπλα του. Είδαμε σε προηγούμενες εντολές πώς μεταφέρεται η θερμοκρασία από τη κάρτα προς τη σειριακή θύρα και τέλος στο θερμόμετρο της διεπαφής.

Όπως θα εξηγήσουμε και παρακάτω στη τρίτη φόρμα έχουμε τη δυνατότητα μέσω προγράμματος να φτιάξουμε θερμοστατικό έλεγχο όπου ορίζουμε μια maximum θερμοκρασία λειτουργίας. Όταν υπερβεί αυτή η θερμοκρασία τότε ενεργοποιείται πάλι ένας συναγερμός για να μας προειδοποιήσει.

Χρησιμοποιώντας ένα μετρητή κάθε φορά που ενεργοποιείται η κεντρική φόρμα και με τη βοήθειά του εκτελούνται όλες οι παραπάνω εντολές. Στη πρώτη εντολή γίνεται σύγκριση της παρούσας θερμοκρασίας με το ανώτατο όριο θερμοκρασίας που έχουμε ορίσει και έχει αποθηκευτεί ως δημόσια (Public) μεταβλητή. Εάν η θερμοκρασία αυτή είναι ανώτερη από το όριο τότε ένας δεύτερος μετρητής (timer) ενεργοποιείται του οποίου τη λειτουργία θα εξηγήσουμε παρακάτω. Επίσης ένα εικονικό μήνυμα εμφανίζεται στην οθόνη για να μας ειδοποιήσει καθώς και ένα ηχητικό beep. Εάν η θερμοκρασία είναι κάτω από το όριο τότε ο δεύτερος μετρητής δεν ενεργοποιείται και η διαδικασία if τερματίζεται και η σύγκριση της θερμοκρασίας ξεκινά πάλι απ' την αρχή.

#### 4.2.7. Ενεργοποίηση συσκευών

Ένα βασικό κομμάτι της διεπαφής είναι το κομμάτι το οποίο θα ενεργοποιούμε τις συσκευές. Όπως έχουμε προαναφέρει στη θέση των συσκευών έχουμε τοποθετήσει led αλλά ομοίως θα μπορούσε να είναι και ένας φωτισμός στη θέση των led με τη βοήθεια ενός ρελέ. Κάθε led ενεργοποιείται με ένα κουμπί φτιαγμένο στη visual basic και απενεργοποιείται με το ίδιο. Πατώντας αυτά τα κουμπιά τρέχει ο κώδικας που θα δούμε:

```
Private Sub Command1_Click()
If Led1 = True Then
    Led1 = False
    Shape1.BackColor = vbWhite
Else
    Led1 = True
    Shape1.BackColor = vbRed
End If
End Sub
```

Πατώντας το κουμπί του πρώτου led αμέσως η μεταβλητή led1 αλλάζει κατάσταση (αν είναι ενεργοποιημένο απενεργοποιείται και το αντίστροφο – δεύτερη εντολή), και επίσης το σχήμα για το led γίνεται άσπρο όταν απενεργοποιείται το led (τρίτη εντολή) και κόκκινο όταν ενεργοποιείται το led (τελευταία εντολή)

Ομοίως γίνεται η εκτέλεση των εντολών και για τα υπόλοιπα led:

```
Private Sub Command2_Click()
If Led2 = True Then
    Led2 = False
    Shape2.BackColor = vbWhite
Else
    Led2 = True
    Shape2.BackColor = vbRed
End If
```

```

End Sub

Private Sub Command3_Click()
If Led3 = True Then
    Led3 = False
    Shape3.BackColor = vbWhite
Else
    Led3 = True
    Shape3.BackColor = vbRed
End If
End Sub

Private Sub Command4_Click()
If Led4 = True Then
    Led4 = False
    Shape4.BackColor = vbWhite
Else
    Led4 = True
    Shape4.BackColor = vbRed
End If
End Sub

Private Sub Command5_Click()
If Led5 = True Then
    Led5 = False
    Shape5.BackColor = vbWhite
Else
    Led5 = True
    Shape5.BackColor = vbRed
End If
End Sub

Private Sub Command6_Click()
If Led6 = True Then
    Led6 = False
    Shape6.BackColor = vbWhite
Else
    Led6 = True
    Shape6.BackColor = vbRed
End If
End Sub

```

#### 4.2.8. Αποθήκευση των μετρούμενων θερμοκρασιών και της στάθμης υγρών

Έχει σχεδιαστεί επίσης μια λίστα μετρήσεων με τη βοήθεια του εργαλείου Log list μέσα στο οποία καταγράφονται οι θερμοκρασίες και η τιμή της στάθμης υγρών μαζί με την ώρα καταγραφής τους. Ο παρακάτω κώδικας εκτελεί αυτές τις ενέργειες:

```

Private Sub LogTimer5_Timer()
LogList.AddItem "Temperature: " + PanelTemp.Text + " Celsius"
LogList.AddItem "Tank Level: " + Text1.Text + "%"
LogList.AddItem Time
LogList.AddItem Date
LogList.AddItem ""
End Sub

```

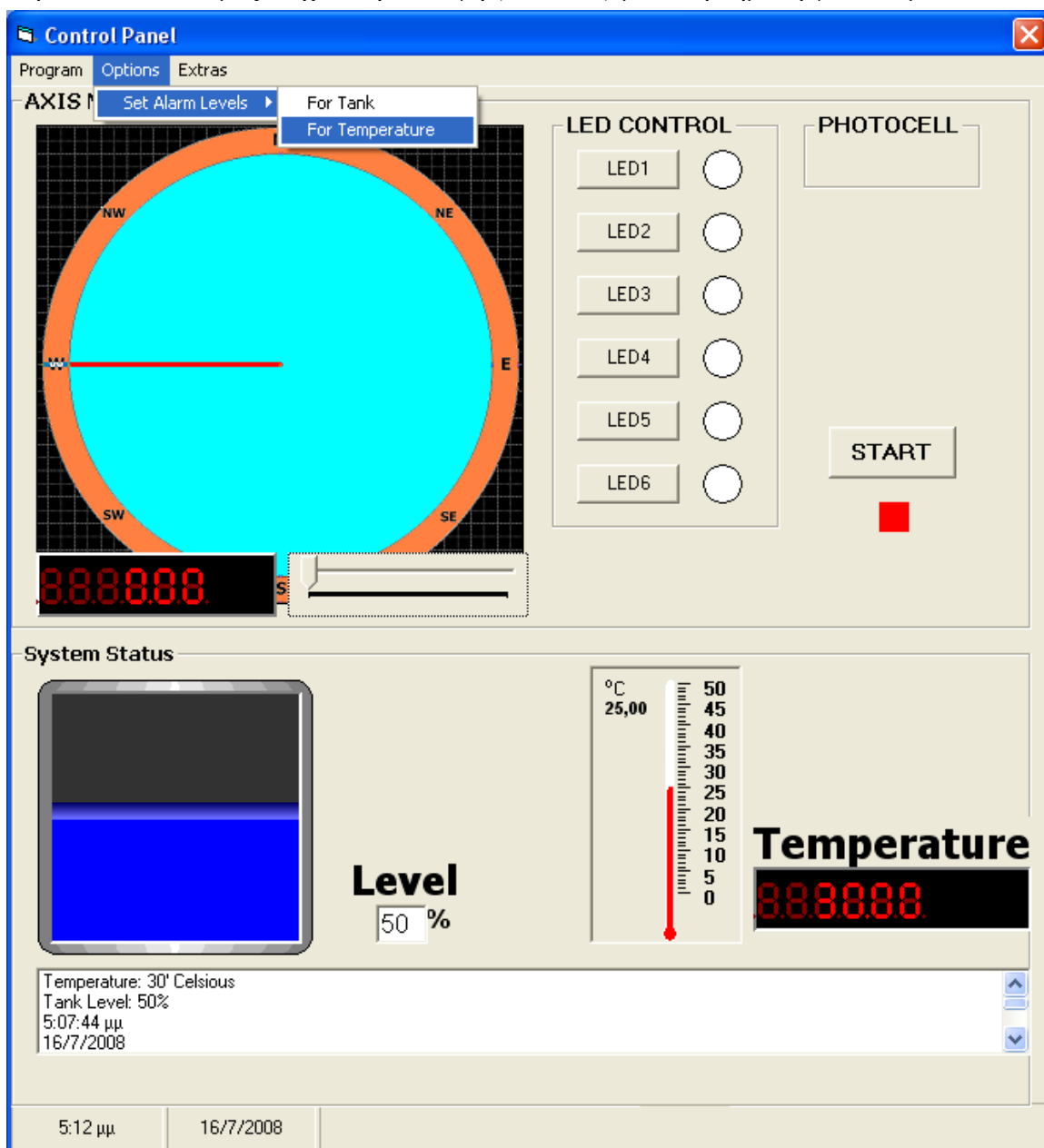
Μόλις ενεργοποιηθεί η βασική φόρμα ο μετρητής LogTimer5 λειτουργεί και κάθε 30 δευτερόλεπτα (interval του timer 30000) μεταφέρει τις μετρήσεις που είπαμε στο LogList το οποίο βρίσκεται στο κάτω μέρος της βασικής φόρμας (βλ. εικόνα 4.1). Αναλυτικά έχουμε:



προσθέτει στη λίστα τη λέξη “temperature” μετά τον αριθμό της θερμοκρασίας που τον αντιγράφει από τη τιμή της ψηφιακής οθόνης θερμοκρασίας και επίσης τη λέξη “Celsius” (πρώτη εντολή). Στη δεύτερη εντολή προσθέτει στη λίστα τη λέξη “Tank Level”, τη τιμή της δεξαμενής που την αντιγράφει απ’ το κουτί κειμένου (Text1 δεξιά του δοχείου) και επίσης προσθέτει και το σύμβολο επί της εκατό. Κατόπιν προσθέτει ώρα λήψης της μέτρησης ( τρίτη εντολή), προσθέτει ημερομηνία (τέταρτη εντολή) και τέλος αφήνει μια κενή γραμμή για να ξεχωρίζουν οι μετρήσεις από τις επόμενες στα επόμενα 30 δευτερόλεπτα (πέμπτη εντολή).

#### 4.2.9. Δημιουργία μενού

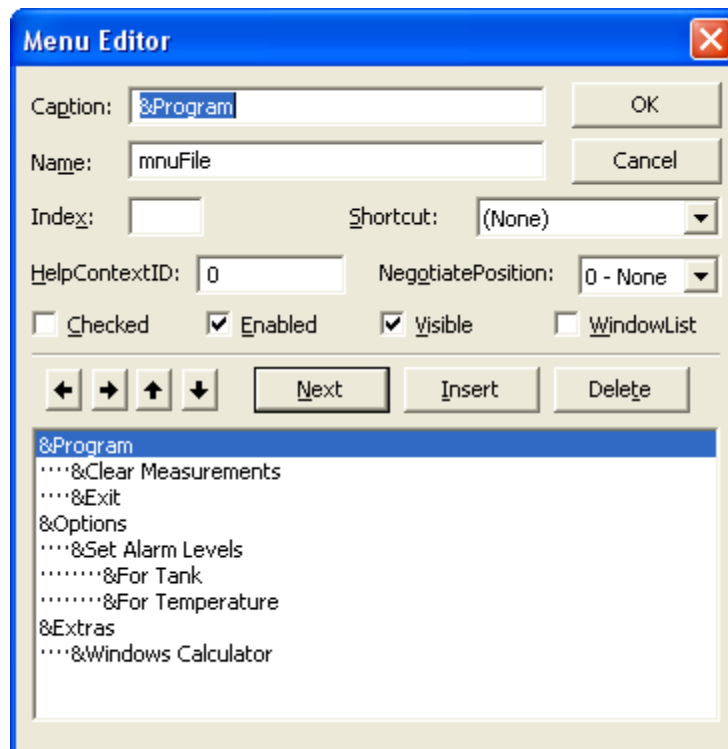
Για να ολοκληρώσουμε τη διεπαφή έπρεπε φυσικά να φτιάξουμε ένα menu στο πάνω μέρος του παραθύρου όπου με αυτό θα κάνουμε κάποιες περαιτέρω ενέργειες. Ας πούμε αποθήκευση μετρήσεων, ρυθμίσεις των αυτοματοποιημένων συναγερμών, έξοδος κ.τ.λ. Η παρακάτω εικόνα μας δείχνει τη διεπαφή (interface) μετά τη δημιουργία των μενού.



Εικόνα 4.5: Η διεπαφή με τη προσθήκη των μενού

Για τη δημιουργία των μενού στο περιβάλλον της Visual Basic επιλέγουμε από το μενού, τη καρτέλα tools και απ' τις επιλογές επιλέγουμε το **Menu Editor** και με αυτό το υποπρόγραμμα μπορούμε στη συγκεκριμένη φόρμα να δημιουργήσουμε οτιδήποτε menu επιθυμούμε. Το συγκεκριμένο μενού της διεπαφής δημιουργήθηκε όπως φαίνεται στην εικόνα 4.6.

Όσες λέξεις γράφονται ακριβώς στην αρχή τότε εμφανίζονται προτεύοντα στη γραμμή εργαλείων. Όσα έχουν τελίτσες μπροστά τότε εμφανίζονται ως υπό μενού των προτευόντων. Όσα τώρα έχουν περισσότερες τελείες εμφανίζονται ως υπό μενού των δευτερευόντων. Δηλαδή σαν τρίτο μενού



**Εικόνα 4.6:** Το εργαλείο της Visual Basic για τη προσθήκη μενού στη φόρμα

Κάθε λέξη του κάθε μενού έχει το δικό του κώδικα που εκτελείται όταν πατηθεί η λέξη με το αριστερό πλήκτρο του ποντικιού. Ας πούμε στη λέξη exit έχουμε το κατωτέρω κώδικα με τον οποίο το πρόγραμμα απενεργοποιείται αμέσως και επίσης ξεφορτώνει απ' τη μνήμη του υπολογιστή και τις τρεις φόρμες:

```
Private Sub mnuExitItem_Click()
End
Unload Form1
Unload Form2
Unload Form3
End Sub
```

Καλό θα είναι μόλις τερματίζουμε τη βασική φόρμα (προηγούμενη εντολή) να δίνεται και άλλη μια εντολή καθώς κλείνει η βασική φόρμα να κλείνει και η επικοινωνία με τη σειριακή θύρα.

```
Private Sub Form_Unload(Cancel As Integer)
CLOSECOM
End Sub
```

Επίσης μπορούμε καθώς τρέχει το πρόγραμμα να σβήσουμε τις καταγεγραμμένες μετρήσεις θερμοκρασίας και στάθμης υγρών του LogList με την ακόλουθη εντολή πατώντας από τη καρτέλα “Program” το κουμπί “Clear Measurements”

```
Private Sub Clear_Click()
LogList.Clear
End Sub
```

Ακόμη εντολές όπως η παρακάτω από τη καρτέλα “Extras” ανοίγει η αριθμομηχανή κάνοντας με ευκολία αριθμητικές πράξεις όποτε εμείς το θελήσουμε. Με αυτό το τρόπο μπορούμε να καλέσουμε μέσω της διεπαφής (interface) οποιοδήποτε πρόγραμμα βρίσκεται σε περιβάλλον windows έτσι απλά γράφοντας το όνομά του.

```
Private Sub Calculator_Click()
Shell "calc", 1 'Run Calculator.
End Sub
```

### 4.3. Δεύτερη Φόρμα

Για να φορτώσουμε τη δεύτερη φόρμα ακολουθούμε το μενού: Options→Set Alarm Levels→For Tank και με το πάτημα του αριστερού πλήκτρου του ποντικιού πάνω στη λέξη θα έχουμε την εμφάνιση της δεύτερης φόρμας με την οποία γίνεται η ρύθμιση των ορίων αυτοματοποιημένου ελέγχου της θερμοκρασίας όπως περιγράψαμε παραπάνω. Ο κώδικας για την απόκρυψη της πρώτης φόρμας και εμφάνιση της δεύτερης είναι ο παρακάτω με το πάτημα αυτού του κουμπιού είναι ο παρακάτω:

```
Private Sub Tanklevel_Click()
Form1.Hide
Form2.Show
End Sub
```

Η δεύτερη φόρμα αμέσως εμφανίζεται. Υπάρχουν δυο οριζόντιες μπάρες (sliders) με τις οποίες δίνουμε το όριο min ή max και η τιμή τους απεικονίζει αντίστοιχα και σε ένα label. Όταν το πρόγραμμα ξεκινά τότε οι μεταβλητές min και max παίρνουν αρχικές τιμές 10% και 90% αντίστοιχα. (το είπαμε στην αρχικοποίηση μεταβλητών).

Για τη πρώτη ράβδο (slider) έχουμε τον ακόλουθο κώδικα για να έχουμε τις επιθυμητές ενέργειες:

```
Private Sub Max_Scroll()
If Max.Value <= Min.Value Then
Max.Value = Min.Value
End If
Tank_lvl.Band1Breakpoint = Max.Value
Max_Label.Caption = Max.Value & "%"
MaxTank = Max.Value
End Sub
```

```
Private Sub Slider1_Scroll()
SFCutaway1.Band2Breakpoint = Slider1.Value → (εντολή για καθορισμό των
χρωμάτων του δοχείου για τη δεύτερη φόρμα)
```

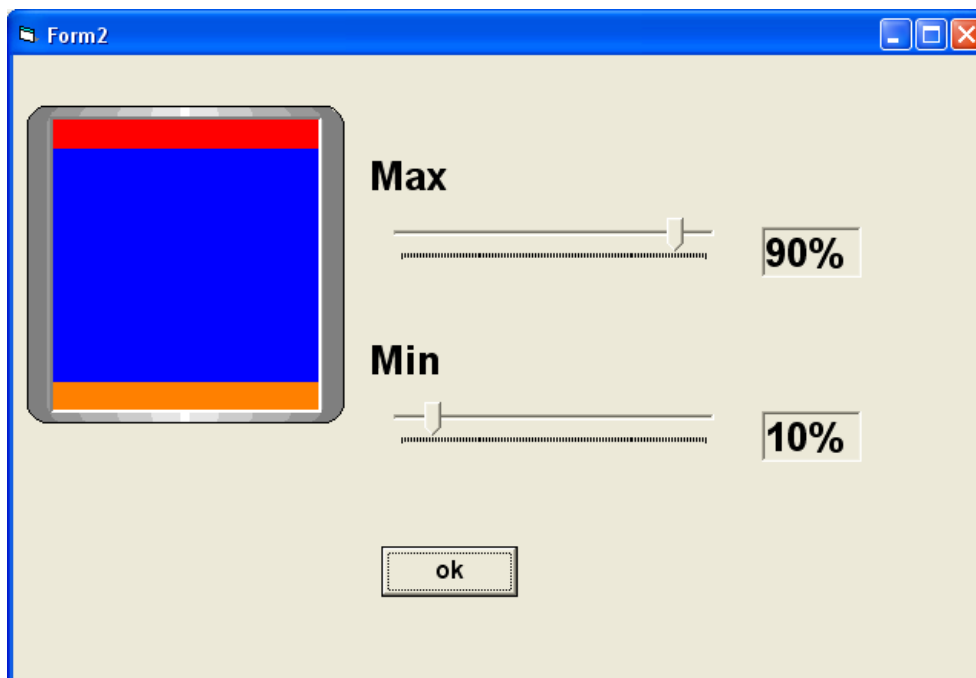
End Sub

Ο παραπάνω κώδικας είναι για την οριζόντια μπάρα (slider) που με αυτήν θα ρυθμίζουμε το άνω όριο της στάθμης υγρών. Η πρώτη διαδικασία If χρησιμοποιείται για τον έλεγχο σε περίπτωση που ο χρήστης δώσει μεγαλύτερο ποσοστό για το όριο min και μικρότερο για το όριο max. Στη χειρότερη των περιπτώσεων επιτρέπεται το min και το max να γίνουν ίσα. Η τέταρτη σειρά κώδικα είναι να αλλάζουν τα χρωματικά επίπεδα του δοχείου (διαφορετικά χρώματα εντός των ορίων του δοχείου και διαφορετικά εκτός των ορίων) ανάλογα με τη μεταβολή της μπάρας τόσο του min όσο και του max. Η πέμπτη σειρά κώδικα είναι για την εμφάνιση της ποσοστιαία κλίμακας του κάθε ορίου σε μια διπλανή ετικέτα (label). Στην έκτη σειρά γίνεται η αποθήκευση στη δημόσια (Public) μεταβλητή τη τιμή του ορίου που καθορίζουμε ως max.

Ομοίως ενεργούμε και για το ελάχιστο όριο της στάθμης υγρών το οποίο ρυθμίζεται με τη δεύτερη ράβδο (slider) και ο κώδικας του είναι ακόλουθος και επεξηγεται όπως ο προηγούμενος.

```
Private Sub Min_Scroll()  
If Min.Value >= Max.Value Then  
Min.Value = Max.Value  
End If  
Tank_lvl.Band2Breakpoint = Min.Value  
Min_Label.Caption = Min.Value & "%"  
MinTank = Min.Value  
End Sub
```

Η επόμενη εικόνα μας δείχνει τη δεύτερη φόρμα μαζί με όλα εργαλεία της:



**Εικόνα 4.7:** Η δεύτερη φόρμα για ρύθμιση των ορίων για των αυτοματοποιημένο έλεγχο της στάθμης υγρών

Μόλις έχουμε κάνει τις ρυθμίσεις μας για τα όρια ελέγχου της στάθμης τότε πατάμε με το αριστερό πλήκτρο του ποντικιού το κουμπί **OK**. Ο κώδικας που εκτελείται με το πάτημα του κουμπιού είναι ο ακόλουθος:

```
Private Sub Command1_Click()  
Form2.Hide  
Form1.Show  
End Sub
```

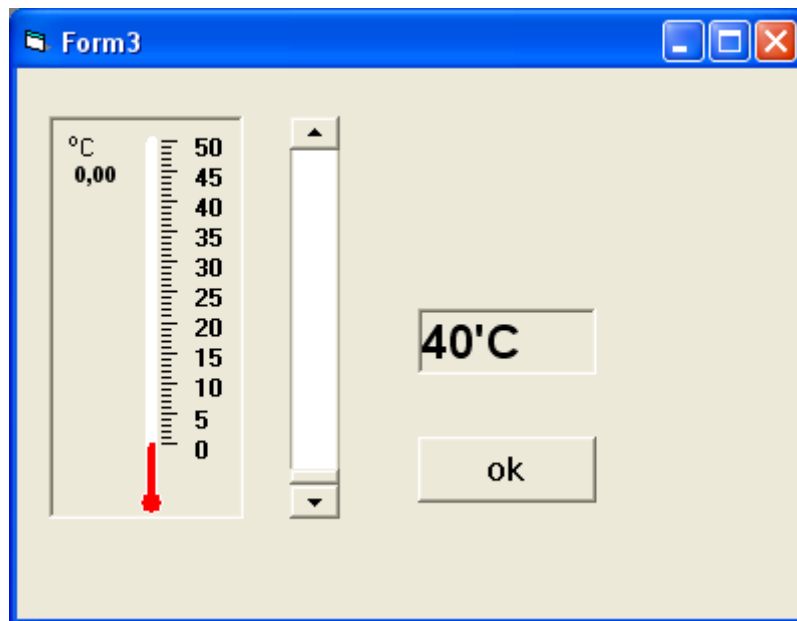
Δηλαδή αυτό που γίνεται είναι απόκρυψη της δεύτερης φόρμας και εμφάνιση πάλι της πρώτης φόρμας.

#### 4.4. Τρίτη Φόρμα

Για να φορτώσουμε τη τρίτη φόρμα ακολουθούμε το μενού: Options→Set Alarm Levels→For Temperature και με το πάτημα του αριστερού πλήκτρου του ποντικιού θα έχουμε την εμφάνιση της τρίτης φόρμας με την οποία γίνεται η ρύθμιση του θερμοκρασίας με βάση την οποία θέλουμε να έχουμε αυτοματοποιημένο θερμοστατικό έλεγχο. Ο κώδικας για την απόκρυψη της πρώτης φόρμας και εμφάνιση της τρίτης είναι ο παρακάτω:

```
Private Sub Temperature_Click()  
Form1.Hide  
Form3.Show  
End Sub
```

Η παρακάτω εικόνα δείχνει τη τρίτη φόρμα:



*Εικόνα 4.8: Η τρίτη φόρμα για τη ρύθμιση του ορίου θερμοκρασίας για το θερμοστατικό έλεγχο*

Όπως βλέπει κανείς υπάρχει μια κάθετη μπάρα με την οποία θα δίνουμε την επιθυμητή θερμοκρασία. Πάνω από αυτή τη θερμοκρασία όπως αναφέραμε και προηγουμένως θα ενεργοποιείται ένας συναγερμός. Αναλυτικά ο κώδικας μας τα εξηγεί με μεγαλύτερη λεπτομέρεια:

```
Private Sub Form_Load()
    alTem1.Value = 50 - (TempScr.Value / 10)
End Sub
```

Με το φόρτωμα της τρίτης φόρμας εκτελείται αμέσως ο παραπάνω κώδικας που βάζει μια τιμή στο θερμομέτρο. Η διαίρεση δια του δέκα είναι για έχουμε δεκαδική ακρίβεια ενός ψηφίου στην ένδειξη της θερμοκρασίας. Η αφαίρεση από το πενήντα είναι για αντιστροφή του τρόπου της μεταβολής. Δηλαδή όσο ανεβαίνει η μπάρα να αυξάνει και η ένδειξη της θερμοκρασίας. Επίσης κατά τη φόρτωση της φόρμας γίνεται και η αντιγραφή της τιμής (value) της μπάρας στο αναλογικής ένδειξης θερμομέτρο της τρίτης φόρμας.

```
Private Sub TempScr_Change()
    Label1.Caption = 50 - (TempScr.Value / 10) & "'C"
    alTem1.Value = 50 - (TempScr.Value / 10)
End Sub
```

Καθώς μεταβάλλουμε την οριζόντια μπάρα εκτελείται ο παραπάνω κώδικας. Γίνεται πάλι η αντιγραφή της τιμής της μπάρας στο αναλογικό θερμομέτρο της τρίτης φόρμας και επίσης γίνεται και μεταφορά της θερμοκρασίας σε ψηφιακή μορφή σε μια ετικέτα (label) μαζί με το χαρακτήρα C που υποδηλώνει τους βαθμούς Κελσίου. Μόλις τελειώσουμε με τη ρύθμιση της θερμοκρασίας μένει να πατήσουμε το πλήκτρο με ετικέτα **OK** και να εκτελεστεί ο παρακάτω κώδικας:

```
Private Sub Command1_Click()
    MaxTemp = 50 - (TempScr.Value / 10)
    Form3.Hide
    Form1.Show
End Sub
```

Η πρώτη εντολή είναι για την απόδοση στη μεταβλητή MaxTemp τη στιγμιαία τιμή της θερμοκρασίας που ρυθμίσαμε με το slider. Αυτή η θερμοκρασία αποθηκεύεται στη συγκεκριμένη μεταβλητή. Κατόπιν όπως περιγράψαμε και πιο πριν θα γίνει η σύγκριση της θερμοκρασίας που έχει καταχωρηθεί στη μεταβλητή MaxTemp με τη θερμοκρασία που μετράει ανά χρονικά διαστήματα η λειτουργική κάρτα μας και αναλόγως θα ενεργοποιείται κάποιος συναγερμός.

#### 4.5. Κώδικας για τον αυτοματοποιημένο έλεγχο

Οι παρακάτω γραμμές κώδικα είναι για τους timer οι οποίοι εκτελούν το κώδικά τους διαρκώς και η ευθύνη τους είναι στο να ελέγχουν διαρκώς τη θερμοκρασία και τη στάθμη της δεξαμενής. Δηλαδή μόλις η θερμοκρασία υπερβεί το όριο το οποίο έχουμε ορίσει ως θερμοκρασία ελέγχου ενεργοποιείται ο συναγερμός. Αντίστοιχα ενεργοποιείται συναγερμός όταν η στάθμη βρεθεί εκτός των ορίων που ορίζουμε (maximum και minimum) βγάζοντας διαφορετικό μήνυμα αν υπερβεί το όριο ή αν πέσει χαμηλότερα από το ελάχιστο.

Μόλις τρέξει η βασική φόρμα έχουμε και την ενεργοποίηση του μετρητή Timer2 ο οποίος έχει interval 20 millisecond. Κάθε 20 m sec εκτελεί το κώδικα του διαρκώς. Η δουλειά του είναι να ελέγχει αν η θερμοκρασία και η στάθμη υγρών βρίσκεται εντός των καθορισμένων ορίων. Επίσης ελέγχει και τη κατάσταση του φωτοκύτταρου. Παρακάτω βλέπουμε αναλυτικά το κώδικά του:

```

Private Sub Timer2_Timer()
If Opto = True Then ledaki.Visible = True Else ledaki.Visible = False →
η εντολή αυτή μας λέει αν η μεταβλητή opto λάβει λογικό 1 τότε δίπλα στο
πρώτο led εμφανίζεται μια λάμπα ως ένδειξη της λειτουργίας του φωτοκύτταρου

If Thermo.Value > MaxTemp Then → αν η θερμοκρασία υπερβεί το όριο ελέγχου
ForbidTimer.Enabled = True → ενεργοποιείται ο forbid timer
Buzzer1 = True → Η μεταβλητή Buzzer1 λαμβάνει λογικό "1"
Else
Buzzer1 = False → Η μεταβλητή Buzzer1 λαμβάνει λογικό "0"
ForbidTimer.Enabled = False → απενεργοποιείται ο forbid timer
Image1.Visible = False → γίνεται απόκρυψη του εικονιδίου της
προειδοποίησης
Label2.Visible = False → γίνεται απόκρυψη της λέξης "warning overheat"
End If

If Tank.Level > MaxTank Then → αν η στάθμη υπερβεί το άνω όριο ελέγχου

Timer4.Enabled = True → ενεργοποιείται ο timer4
Timer4.Tag = "High" → Η ιδιότητα Tag του timer4 λαμβάνει τιμή High
να ξέρουμε ότι χτυπά λόγω υψηλής στάθμης
Buzzer2 = True → Η μεταβλητή Buzzer2 λαμβάνει λογικό "1"
End If
If Tank.Level < MinTank Then → αν η στάθμη πέσει από το κάτω όριο ελέγχου
Buzzer2 = True → Η μεταβλητή Buzzer2 λαμβάνει λογικό "1"
Timer4.Enabled = True → ενεργοποιείται ο timer4
Timer4.Tag = "Low" → Η ιδιότητα Tag του timer4 λαμβάνει τιμή Low να
ξέρουμε ότι χτυπά λόγω χαμηλής στάθμης

End If
If Tank.Level < MaxTank And Tank.Level > MinTank Then → αν η στάθμη είναι
εντός των καθορισμένων ορίων
Buzzer2 = False → Η μεταβλητή Buzzer2 λαμβάνει λογικό "0"
Timer4.Enabled = False → απενεργοποιείται ο timer4
Image2.Visible = False → γίνεται απόκρυψη του εικονιδίου της
προειδοποίησης
lowlabel.Visible = False → γίνεται απόκρυψη της λέξης "low label"
highlabel.Visible = False → γίνεται απόκρυψη της λέξης "high label"
End If

If Buzzer1 = True Or Buzzer2 = True Then Buzzer = True Else Buzzer = False
End Sub → με την εντολή αυτή αν οποιοσδήποτε από τους δυο συναγερμούς
(buzzer1,buzzer2) είναι ενεργοποιημένος τότε η μεταβλητή Buzzer λαμβάνει
λογικό "1" και τότε κτυπά το buzzer που βρίσκεται πάνω στη λειτουργική
κάρτα.

```

Μιλήσαμε στο προηγούμενο κώδικα για ενεργοποίηση του Timer4 και του μετρητή ForbidTimer. Σκοπός του μετρητή Forbidtimer είναι να αναβοσβήνει το εικονίδιο προειδοποίησης για τη θερμοκρασία με περίοδο 500 m sec και να κάνει ένα beep για να προειδοποιεί το χρήστη. Ο μετρητής Timer4 έχει σα σκοπό κάθε 500 m sec να αναβοσβήνει το απαγορευτικό σήμα της δεξαμενής και το προειδοποιητικό μήνυμα. Όμως ανάλογα με τον αν έχει υπερβεί το όριο ή έχει πέσει κάτω από το κάτω όριο θα εμφανίζει και διαφορετικό μήνυμα. Παρακάτω θα δούμε το κώδικα των δυο αυτών μετρητών :

```

Private Sub ForbidTimer_Timer()
Beep → δημιουργία ηχητικού σήματος στον ηλεκτρονικό υπολογιστή για
προειδοποίηση
If Image1.Visible = True Then → αν η προειδοποιητική εικόνα είναι ενεργή

```

```

    Image1.Visible = False → τότε γίνεται απόκρυψη της προειδοποιητικής
    εικόνας
    Label2.Visible = False → γίνεται απόκρυψη του προειδοποιητικού
    μηνύματος
Else
    Image1.Visible = True → τότε γίνεται εμφάνιση της προειδοποιητικής
    εικόνας
    Label2.Visible = True → γίνεται εμφάνιση του προειδοποιητικού
    μηνύματος
End If
End Sub

```

```
Private Sub Timer4_Timer
```

Beep → δημιουργία ηχητικού σήματος στον ηλεκτρονικό υπολογιστή για προειδοποίηση

```
If Timer4.Tag = "High" Then → αν η ιδιότητα Tag του timer4 λαμβάνει τιμή
High τότε
```

```
    If Image2.Visible = True Then → αν το εικονίδιο προειδοποίησης είναι
ενεργό τότε
```

```
        Image2.Visible = False → το εικονίδιο προειδοποίησης απενεργοποιείται
        highlabel.Visible = False → εξαφανίζεται το προειδοποιητικό
        μήνυμα με ετικέτα "High Level"
```

```
        lowlabel.Visible = False → εξαφανίζεται το προειδοποιητικό μήνυμα με
        ετικέτα "Low Level"
```

```
    Else
```

```
        Image2.Visible = True → εμφανίζεται το εικονίδιο προειδοποίησης
        highlabel.Visible = True → εμφανίζεται το προειδοποιητικό μήνυμα με
        ετικέτα "High Level"
```

```
        lowlabel.Visible = False → εξαφανίζεται το προειδοποιητικό μήνυμα με
        ετικέτα "Low Level"
```

```
    End If
```

```
End If
```

```
If Timer4.Tag = "Low" Then → αν η ιδιότητα Tag του timer4 λαμβάνει τιμή Low
    If Image2.Visible = True Then → αν το εικονίδιο προειδοποίησης είναι
    ενεργό τότε
```

```
        Image2.Visible = False → το εικονίδιο προειδοποίησης απενεργοποιείται
        Lowlabel.Visible = False → εξαφανίζεται το προειδοποιητικό
        μήνυμα με ετικέτα "Low Level"
```

```
        highlabel.Visible = False → εξαφανίζεται το προειδοποιητικό μήνυμα
        με ετικέτα "High Level"
```

```
    Else
```

```
        Image2.Visible = True → εμφανίζεται το εικονίδιο προειδοποίησης
        lowlabel.Visible = True → εμφανίζεται το προειδοποιητικό μήνυμα με
        ετικέτα "Low Level"
```

```
        highlabel.Visible = False → εξαφανίζεται το προειδοποιητικό
        μήνυμα με ετικέτα "High Level"
```

```
    End If
```

```
End If
```

```
End Sub
```

## 4.6. Δήλωση κεντρικών Μεταβλητών

Το τελευταίο αυτό κομμάτι κώδικα είναι για τη δήλωση κάποιων κεντρικών μεταβλητών το οποίο είναι αποθηκευμένο σε ένα επιπρόσθετο αρχείο της Visual Basic το Module1 (Ports.bas)



```
Declare Function OPENCOM Lib "Port" (ByVal A$) As Integer
Declare Sub CLOSECOM Lib "Port" ()
Declare Sub SENDBYTE Lib "Port" (ByVal b%)
Declare Function READBYTE Lib "Port" () As Integer
Declare Sub DTR Lib "Port" (ByVal b%)
Declare Sub RTS Lib "Port" (ByVal b%)
Declare Sub TXD Lib "Port" (ByVal b%)
Declare Function CTS Lib "Port" () As Integer
Declare Function DSR Lib "Port" () As Integer
Declare Function RI Lib "Port" () As Integer
Declare Function DCD Lib "Port" () As Integer
Declare Sub DELAY Lib "Port" (ByVal b%)
Declare Sub TIMEINIT Lib "Port" ()
Declare Sub TIMEINITUS Lib "Port" ()
Declare Function TIMEREAD Lib "Port" () As Long
Declare Function TIMEREADUS Lib "Port" () As Long
Declare Sub DELAYUS Lib "Port" (ByVal l As Long)
Declare Sub REALTIME Lib "Port" (ByVal i As Boolean)
```

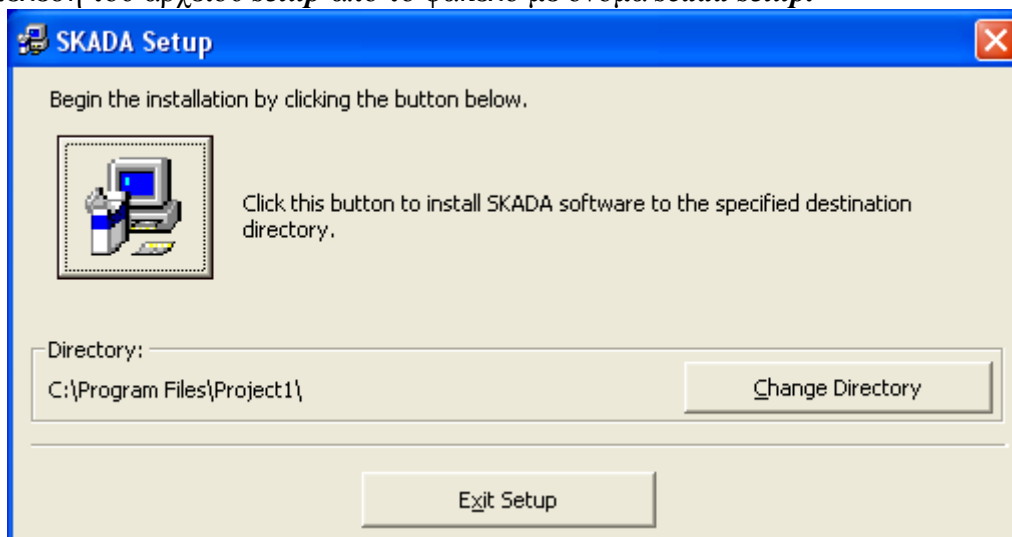
Όλες οι παραπάνω εντολές παραπέμπονται για το αρχείο Port.dll το οποίο είναι υπεύθυνο για την επικοινωνία της σειριακής θύρας. Έτσι λοιπόν μέσω αυτών των εντολών γίνεται διασύνδεση της διεπαφής με το αρχείο Port.dll το οποίο το αντιγράφουμε στο φάκελο system32 των Windows.

## 5. Χρήση λειτουργικής κάρτας

Σε αυτό το κεφάλαιο θα περιγράψουμε τη λειτουργία της πτυχιακής εργασίας και από πλευράς υλικού και από πλευράς λογισμικού ούτως ώστε να μπορεί ο οποιοσδήποτε να θέσει σε λειτουργία τη κατασκευή. Η διεπαφή (Interface) που σχεδιάσαμε με τη βοήθεια της Visual Basic το κάνουμε αρχείο εκτελέσιμο (.exe) για να τρέχει σε οποιοδήποτε υπολογιστή και κατόπιν κάνουμε κάποιες αντιγραφές αρχείων ώστε να μπορεί το εκτελέσιμο αρχείο να λειτουργήσει μαζί με όλα τα εργαλεία του (components). Παρακάτω θα δούμε αναλυτικά τη διαδικασία εγκατάστασης των αρχείων αυτών στον ηλεκτρονικό υπολογιστή σε περιβάλλον Windows.

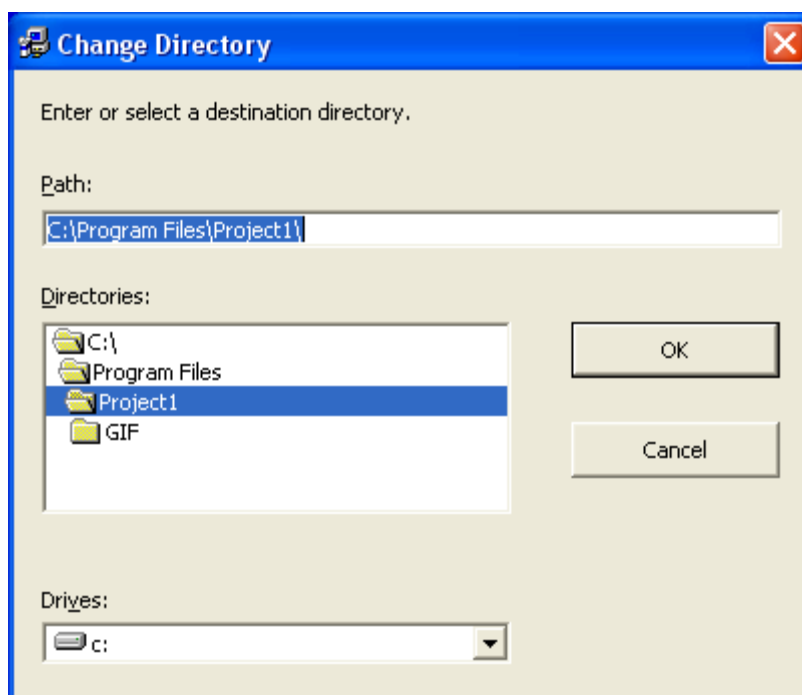
### 5.1. Εγκατάσταση προγράμματος με όνομα “scada”

Το πρόγραμμα αυτό το εγκαθιστούμε στο υπολογιστή εκτελώντας το αρχείο setup του όπως στα πιο συνηθισμένα προγράμματα. Εκτελώντας την εγκατάσταση γίνεται αυτόματη αντιγραφή αρχείων στο φάκελο των windows. Χωρίς αυτά δε θα ήταν δυνατή η λειτουργία του αρχείου .exe στο περιβάλλον των windows Xp. Όμως μερικά αρχεία θέλουν χειροκίνητη αντιγραφή στο φάκελο των Windows όπως θα δούμε παρακάτω. Η παρακάτω εικόνα δείχνει την εκτέλεση του αρχείου *setup* από το φάκελο με όνομα *scada setup*.



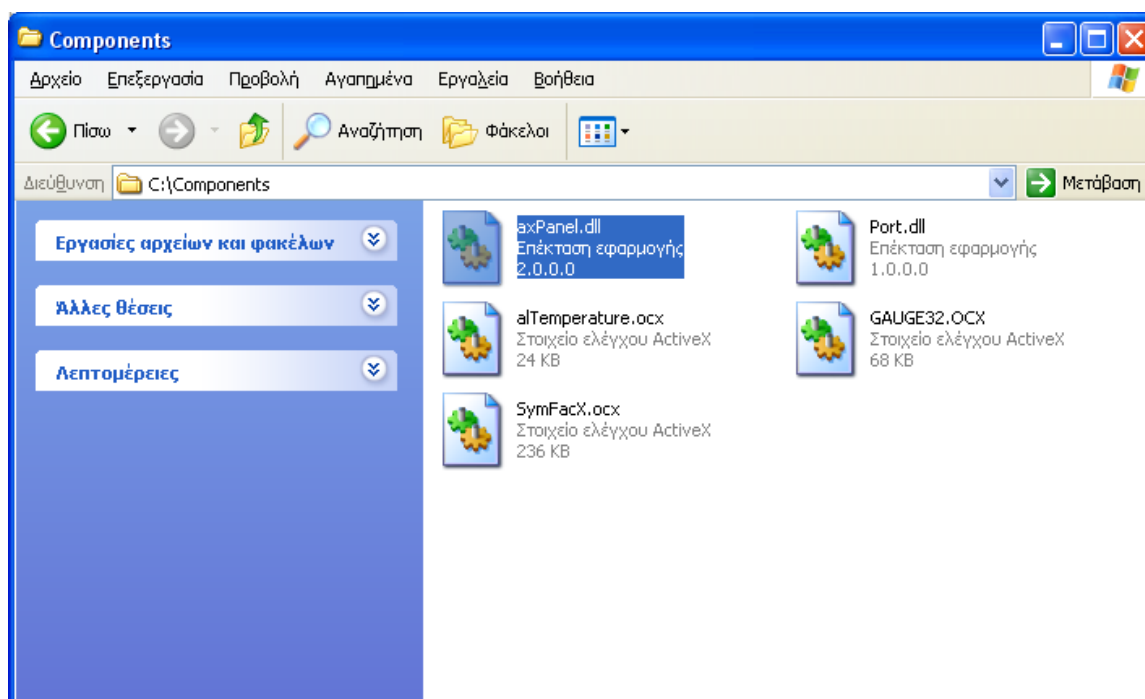
Εικόνα 5.1: εκτέλεση του αρχείου εγκατάστασης

Όπως βλέπουμε στην εικόνα 5.1 υπάρχει και κουμπί για επιλογή της διεύθυνσης εγκατάστασης. Μόλις επιλέξουμε τη διεύθυνση εγκατάστασης πατάμε το κουμπί με το εικονίδιο του αρχείου setup. Κατόπιν το πρόγραμμα θα μας ζητήσει το όνομα για το πρόγραμμα το οποίο θα εγκατασταθεί (το όνομα είναι τυπικό άλλωστε το πρόγραμμα δεν θα το χρησιμοποιούμε είναι μόνο για την αντιγραφή των αρχείων), Μόλις δώσουμε και όνομα και πατήσουμε *ok* τότε η εγκατάσταση των αρχίζει και διαρκεί ελάχιστα δευτερόλεπτα.



*Εικόνα 5.2: Επιλογή διεύθυνσης για την εγκατάσταση του προγράμματος*

Μόλις λοιπόν εγκαταστήσουμε το απαραίτητο αυτό πρόγραμμα μπορούμε να ανοίξουμε τη διεπαφή που κατασκευάσαμε (final\_scada.exe). Εμφανίζεται η διεπαφή (interface) μαζί με όλα τα εργαλεία (components) τα οποία σχεδιάσαμε στο περιβάλλον της Visual Basic. Η διεπαφή σχεδιάστηκε με components (εργαλεία) τα οποία δεν υπήρχαν σαν προκαθορισμένα στο πρόγραμμα της Visual Basic αλλά τα κάναμε προσθήκη (components μετά από εύρεση στο διαδίκτυο). Για να γίνονται αναγνωρίσιμα στο περιβάλλον των Windows πρέπει να κάνουμε αντιγραφή στο φάκελο των Windows κάποια αρχεία dll.



*Εικόνα 5.3: αρχεία dll προς αντιγραφή στο φάκελο των Windows*

Τα αρχεία τα οποία φαίνονται στην εικόνα 5.3 είναι όπως είπαμε υπεύθυνα για την εμφάνιση του αναλογικού δείκτη (gauge32.ocx), του θερμομέτρου (altemperature.ocx) και της ψηφιακής οθόνης (axPanel.dll) για την ένδειξη της θερμοκρασίας. Το αρχείο port.dll είναι υπεύθυνο για το άνοιγμα της σειριακής θύρας για να γίνει εφικτή η επικοινωνία. Χωρίς αυτό το αρχείο ο ηλεκτρονικός υπολογιστής δεν θα μπορούσε να στέλνει δεδομένα ούτε να λαμβάνει. Και τα πέντε λοιπόν αρχεία τα αντιγράψουμε στο φάκελο με διεύθυνση → C:\WINDOWS\system32.

Τώρα είμαστε έτοιμοι από λογισμικής πλευράς να λειτουργήσουμε τη διεπαφή που έχουμε κατασκευάσει. Παρακάτω θα δούμε τις απαραίτητες διεργασίες για τη συνδεσμολογία της λειτουργικής κάρτας με τον ηλεκτρονικό υπολογιστή.

## 5.2. Συνδεσμολογία λειτουργικής κάρτας με τον ηλεκτρονικό υπολογιστή

Μόλις λοιπόν γίνει η απαραίτητη εγκατάσταση λογισμικού μπορούμε να συνδέσουμε τη λειτουργική μας κάρτα με τον ηλεκτρονικό υπολογιστή.

Το πρώτο βήμα είναι να συνδέσουμε το καλώδιο της σειριακής θύρας. Το ένα άκρο του καλωδίου το συνδέουμε στη σειριακή θύρα της μητρικής κάρτας του ηλεκτρονικού υπολογιστή (αυστηρά στη COM1). Το άλλο άκρο του καλωδίου το συνδέουμε στο συνδετήρα (connector) σειριακής θύρας της λειτουργικής μας κάρτας.

Το δεύτερο βήμα είναι να ανοίξουμε τη διεπαφή (interface). Θέτουμε σε εκτέλεση το κατασκευασμένο πρόγραμμα final\_scada.exe.

Επόμενο βήμα είναι η σύνδεση των τάσεων τροφοδοσίας. Συνδέουμε τη λειτουργική μας κάρτα σε τάση τροφοδοσίας από το τροφοδοτικό A. Η τιμή του τροφοδοτικού A είναι στα 12 Volt. Μετά όπως έχουμε πει η τάση σταθεροποιείται στα 5 Volt. Παρόλα αυτά για τάση τροφοδοσίας 12 Volt έχει αποδειχτεί πειραματικά ότι τα ψηφιακά κυκλώματα της κάρτας λειτουργούν καλύτερα χωρίς σφάλματα. Κατόπιν συνδέουμε τη τάση τροφοδοσίας στο κύκλωμα οδήγησης βηματικού μοτέρ από το τροφοδοτικό B. Η τιμή αυτού είναι στα 6 Volt. Μόλις συνδεθεί η τάση στη λειτουργική κάρτα τότε όλοι οι έξοδοι αρχικοποιούνται σε υψηλό δυναμικό. Τότε όλα τα led ανάβουν και χτυπά και ο βομβητής (buzzer). Αυτό που έχουμε να κάνουμε είναι να πατήσουμε το κουμπί (button) του reset που βρίσκεται πάνω στη πλακέτα. Κατόπιν πατάμε το κουμπί start της διεπαφής. Τότε λαμβάνονται τα νέα δεδομένα για τα led από τη λειτουργική κάρτα και αμέσως σβήνουν. Αν εφόσον το κουμπί start είναι πατημένο και η θερμοκρασία που λαμβάνουμε είναι λάθος τότε πατάμε εκ νέου το κουμπί (button) του reset.

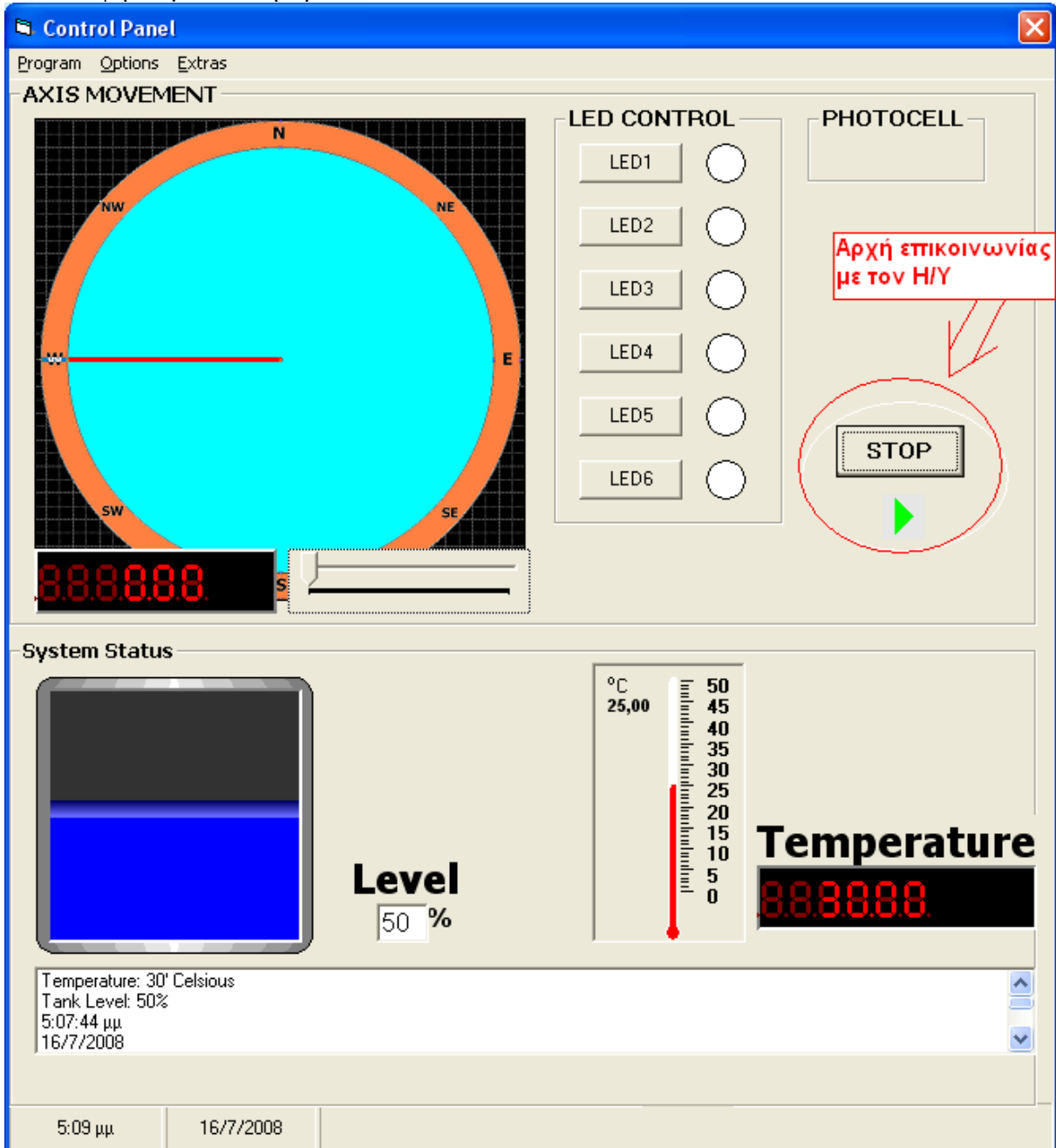
## 5.3. Χρήση της διεπαφής (interface)

Μόλις γίνουν οι ανωτέρω εργασίες η διεπαφή είναι έτοιμη για χρήση. Η διαδικασία του reset για αρχικοποίηση των τιμών γίνεται μόνο όταν η κάρτα βγει από τη τάση τροφοδοσίας και επιθυμούμε να την επανασυνδέσουμε. (ή και σε περίπτωση σφάλματος στα δεδομένα όπως προείπαμε). Εφόσον λοιπόν έχουμε πατημένο το κουμπί με ετικέτα "start" (εικόνα 5.4) ο Η/Υ διαβάζει διαρκώς τη θερμοκρασία και τη κατάσταση του φωτοκύτταρου. Δηλαδή αυτομάτως εισέρχεται η θερμοκρασία στη διεπαφή και το εικονίδιο του φωτοκύτταρου.

Η παρέμβαση του χρήστη είναι σε όλα τα υπόλοιπα. Από το τμήμα *led control* μπορούμε να ενεργοποιήσουμε ή αντίστοιχα να απενεργοποιήσουμε τα led. Μόλις

ενεργοποιήσουμε ένα led αυτομάτως φαίνεται εικονικά στη διεπαφή και κατόπιν ανάβει και το αντίστοιχο στη πλακέτα.

Από το τμήμα *Axis Movement* κατευθύνουμε το βηματικό κινητήρα. Όλα γίνονται με τη βοήθεια της μπάρας (slider). μόλις σύρουμε το slider και αφήσουμε το πλήκτρο του ποντικιού ο βηματικός κινητήρας στρέφεται τόσες μοίρες όσες και στη διεπαφή (interface). Όταν κρατάμε πατημένο το ποντίκι και σέρνουμε τη ράβδο ο βηματικός κινητήρας δεν ξεκινά αν δεν αφήσουμε το πλήκτρο του ποντικιού.



Εικόνα 5.4: *final\_scada.exe*

Το τμήμα *System Status* είναι για μέτρηση της στάθμης υγρών. Δεν υπάρχει ανταπόκριση με τη πλακέτα (μελλοντικά μπορεί να γίνει). Μπορούμε όμως από λογισμικής πλευράς να πειραματιστούμε με αυτό το τμήμα.

Στο τμήμα της θερμοκρασίας έχουμε το θερμόμετρο και τη ψηφιακή ένδειξη η οποία αντιγράφει τη τιμή της από το θερμόμετρο. Η μέτρηση όπως είπαμε γίνεται αυτόματα.

Το τελευταίο τμήμα στο κάτω μέρος είναι η καταγραφή των μετρήσεων της θερμοκρασίας και της στάθμης υγρών. Η καταγραφή γίνεται αυτόματα ανά 30 δευτερόλεπτα με το που ανοίξουμε τη διεπαφή (χωρίς να έχει πατηθεί το κουμπί start).

Τελευταίες λειτουργίες γίνονται από τα μενού της διεπαφής. Όπως καθαρισμός της καταγραφής των μετρήσεων, και ρυθμίσεις των ορίων στάθμης υγρών και θερμοκρασίας (από το μενού options→set alarm levels). Ο καθορισμός των ορίων γίνεται όπως προείπαμε στο κεφάλαιο 4 από τις φόρμες 2 και 3 αντίστοιχα.

Για να τελειώσουμε την εφαρμογή πατάμε το exit του προγράμματος. Πριν αποσυνδέσουμε το καλώδιο της σειριακής αφαιρούμε τις τάσεις τροφοδοσίας από τα τροφοδοτικά A και B.

## 6. Συμπεράσματα

Με το πέρας της πτυχιακής εργασίας έγινε καλή πρακτική πάνω σε πολλούς τομείς. Ένας από αυτούς ήταν το τι μπορεί να φτιάξει κανείς με τη χρήση της σειριακής θύρας και τη μετάδοση δεδομένων μέσα από αυτήν. Έγινε λοιπόν αρχικά μια σκέψη του τι είναι χρήσιμο να κατασκευαστεί και αν είναι εφικτό να πραγματοποιηθεί στη κατασκευή. Μπορεί ο ρυθμός μετάδοσης της σειριακής θύρας να εκτελέσει ικανοποιητικά τη μετάδοση των δεδομένων αυτών; Όλες αυτές οι σκέψεις βοήθησαν πολύ στην βελτίωση των δυνατοτήτων μου πάνω σε σχεδίαση και υλοποίηση αυτομάτων συστημάτων όπως είναι το αντικείμενο με το οποίο και που ασχολήθηκα. Κάθε αυτόματος έλεγχος ας πούμε έχει το δικό του ιδιαίτερο τρόπο υλοποίησης.

Ένας άλλος τομέας ήταν η κατασκευή πλακετών. Οι δυο κατασκευές που φτιάχτηκαν η μια ήταν η λειτουργική κάρτα και η άλλη το κύκλωμα οδήγησης βηματικού κινητήρα, έδωσαν καλή εξάσκηση τόσο στη σχεδίαση του κυκλώματος στον υπολογιστή όσο και στη τύπωση του κυκλώματος πάνω στο χαλκό. Πριν τη τύπωση της πλακέτας πάνω στο χαλκό γινόταν πρώτα πειραματική λειτουργία του κυκλώματος πάνω σε ειδικό εργαλείο (breadboard). Φυσικά η κάθε πλακέτα δεν είχε κατευθείαν τις επιθυμητές ιδιότητες. Έτσι λοιπόν πειραματιζόμουν με διαφορετικά εξαρτήματα όπως π.χ. με διαφορετικές τιμές πυκνωτών κ.τ.λ. Κάποια στιγμή πετυχαίναμε τη σωστή λειτουργία του κυκλώματος. Αυτό φυσικά βοήθησε στη καλή κατανόηση των ιδιοτήτων των ηλεκτρονικών εξαρτημάτων. Στα ψηφιακά κυκλώματα με τη χρήση του βολτόμετρου - αμπερόμετρου και βοήθεια πινάκων αληθείας (μέσα από τα αντίστοιχα datasheet) ελέγχαμε πιθανά προβλήματα στη διεξαγωγή της κατασκευής. Μόλις το κύκλωμα λειτουργούσε σωστά σε πειραματικό επίπεδο τότε ξεκίνησα τη σχεδίαση πάνω στο χαλκό. Η κατασκευή στο χαλκό ήταν πολύ καλή σε εξάσκηση αφήνοντας προοπτικές για νέα μελλοντική ενασχόληση με άλλες κατασκευές.

Η πτυχιακή εργασία δεν προσέφερε γνώσεις μόνο σε επίπεδο υλικού (hardware), αλλά και σε επίπεδο προγραμματισμού (software). Και η γλώσσα προγραμματισμού που επιλέχτηκε για τη δημιουργία της επικοινωνίας και από ήταν μια καλή εξάσκηση. Κάθε γλώσσα προγραμματισμού είναι ιδανική για διαφορετικές εφαρμογές. Μετά από κάποια επαφή με διάφορες γλώσσες προγραμματισμού η ιδανικότερη για τη δικιά μου κατασκευή θεώρησα ότι ήταν η Visual Basic. Είναι πολύ εύκολη για δημιουργία φορμών αφού περιέχει πολλά δικά της έτοιμα εργαλεία. Μετά από λίγη εξάσκηση πάνω σε μικρές και απλές εφαρμογές υπήρχε η δυνατότητα για εγγραφή κώδικα πιο σύνθετου ο οποίος έκανε λειτουργίες τις οποίες ήθελα για τη κατασκευή μου. Ξεκίνησα με τη πιο απλή επικοινωνία όπως αποστολή ενός bit μέσω της σειριακής θύρας στη πλακέτα. Αυτό το bit όταν το έστειλα σε λογικό "1" μέσω υπολογιστή, τότε άναβε led πάνω στη πλακέτα. Σε αντίθετη περίπτωση το led έσβηνε. Σιγά σιγά η επικοινωνία γινόταν όλο και πιο σύνθετη με αποστολή περισσότερων δεδομένων και σε λιγότερο χρόνο. Όλα αυτά για να φτάσουμε στο τελικό στάδιο που ήταν αποστολή και λήψη δεδομένων ταυτόχρονα. Αρκετά γρήγορα η γλώσσα προγραμματισμού έγινε αρκετά εύχρηστη ανακαλύπτοντας ολοένα και περισσότερα κόλπα για την υλοποίηση του επιθυμητού αποτελέσματος.

Τελειώνοντας και με αυτό το κομμάτι του προγραμματισμού είχε φτάσει το κομμάτι της σύνδεσης του υλικού (hardware) με το λογισμικό (software). Το καλό στην σχεδίαση ήταν ο τρόπος σκέψης για μερικά εξαρτήματα. Όπως π.χ. μια αντίσταση δεν ήταν σίγουρο ότι η τιμή της θα μας καλύπτει απόλυτα. Έτσι λοιπόν στη θέση της βάζαμε ένα ποτενσιόμετρο. Άρα είχαμε τη δυνατότητα μεταβολής της τιμής της αντίστασης. Άρα λοιπόν κατά τη λειτουργία της κατασκευής είχαμε τη δυνατότητα εφαρμογής ρυθμίσεων για την επίτευξη της σωστής λειτουργίας του κυκλώματος. Έβλεπα λοιπόν ότι σε κάποιο σημείο υπήρχε κάποιο πρόβλημα. Μεταβάλλοντας τις τιμές των στοιχείων αυτών άλλαζε και η συμπεριφορά του κυκλώματος. Σημαντικό ρόλο στη σχεδίαση έπαιξε και η απλοποίηση του κυκλώματος με

όσο το δυνατόν λιγότερα εξαρτήματα. Στην αρχή σχεδιάστηκε με κάποιο αριθμό εξαρτημάτων και στη διάρκεια διαπιστώθηκε ότι μπορεί μερικά εξαρτήματα να αντικατασταθούν με άλλα πιο απλά και κατά συνέπεια λιγότερους αγωγούς. Πολλές φορές κάποιο τμήμα της κατασκευής δεν λειτουργούσε. Αμέσως η σκέψη μου έτρεχε σε αντικατάσταση εξαρτημάτων με άλλα με διαφορετικές ιδιότητες. Ας πούμε στην αρχή υπήρχε σκέψη για κατασκευή πάνω στη πλακέτα εξωτερικού κυκλώματος για τη δημιουργία του σήματος του ρολογιού. Αυτό όμως αργότερα διαπίστωσα ότι δεν ήταν απαραίτητο και όλο το κύκλωμα αντικαταστάθηκε από έναν ακροδέκτη της σειριακής θύρας όπου το σήμα του ρολογιού δημιουργούταν μέσω λογισμικού με τη βοήθεια ενός timer (εργαλείο της Visual Basic).

Τελειώνοντας λοιπόν τη πτυχιακή εργασία έπρεπε όλες οι ενέργειες και όλα τα στοιχεία να καταγραφούν στο έγγραφο της πτυχιακής εργασίας. Άρα έγινε σωστή συλλογή των απαραίτητων στοιχείων για κάθε εξάρτημα του υλικού (hardware) καθώς και όλα τα βήματα ένα προς ένα τόσο για τη κατασκευή των πλακετών όσο για τη δημιουργία του απαραίτητου λογισμικού.



## Βιβλιογραφία

- Έλεγχος κυκλωμάτων και μετρήσεων με Η/Υ, Εκδόσεις Τζιόλα
- Ψηφιακή σχεδίαση, Morris Mano δεύτερη έκδοση
- Microsoft Visual Basic 5, από Microsoft Press
- Ηλεκτρονικά με τη Visual Basic, εκδόσεις ΕΛΕΚΤΟΡ Ε.Π.Ε.

Sites που χρησιμοποιήθηκαν:

- [www.alldatasheet.com](http://www.alldatasheet.com) για εύρεση οποιουδήποτε datasheet από οτιδήποτε υλικό χρησιμοποιήθηκε
- [www.elektor.com](http://www.elektor.com) πληροφορίες σχετικά με επικοινωνία υπολογιστή με κατασκευασμένες εφαρμογές
- [www.google.gr](http://www.google.gr) για εύρεση πληροφοριών σχετικά με τη θεωρία βηματικών κινητήρων

## Πίνακας Περιεχομένων

1. Εισαγωγή.....	1
2. Σχεδίαση Κάρτας – Πλήρης ανάλυση κάθε βαθμίδας .....	5
2.1. Γενικά περί της κάρτας .....	5
2.2. Σταθεροποιητής τάσεως με χρήση του ολοκληρωμένου LM7805 .....	6
2.3. Καταχωρητές για την αποθήκευση των δεδομένων.....	7
2.3.1. Ολοκληρωμένο κύκλωμα TC74HC4094 καταχωρητής ολίσθησης (Shift register).....	7
2.3.2. Ολοκληρωμένο κύκλωμα MC74HC374A.....	10
2.4. Ολοκληρωμένο κύκλωμα 4022 για χρονισμό του καταχωρητή MC74HC374A.....	13
2.5. Ψηφιακό θερμομέτρο με τη χρήση ολοκληρωμένου LM335 .....	15
2.5.1. Αισθητήρας θερμοκρασίας LM335 .....	15
2.5.2. Αναλογικός σε ψηφιακός μετατροπέας TLC549IP .....	17
2.6. Ανίχνευση φωτεινότητας .....	18
2.7. HEF40106 Πύλες NOT – Hex Inverting Schmitt trigger.....	19
2.8. Συγκέντρωση όλων των βαθμίδων σε μια τελική κατασκευή .....	20
3. Βηματικοί κινητήρες και κυκλώματα οδήγησης.....	24
3.1. Βασικά κυκλώματα ελέγχου βηματικών κινητήρων.....	24
3.1.1. Βηματικοί κινητήρες μεταβλητής αντίδρασης .....	24
3.1.2. Μονοπολικό μόνιμο μαγνητικού πεδίου και Υβριδικοί βηματικοί κινητήρες .....	27
3.1.3. Διπολικό βηματικό κινητήρες και Γέφυρες-H.....	30
3.1.4. Πρακτικά κυκλώματα οδήγησης διπολικών κινητήρων.....	33
3.1.5. Βηματικοί κινητήρες διπλού σύρματος .....	35
3.1.6. Πολυφασικοί βηματικοί κινητήρες.....	36
3.2. Εργαστηριακή Εφαρμογή πάνω στους βηματικούς κινητήρες .....	37
3.2.1. Ηλεκτρονικό σχέδιο κυκλώματος οδήγησης .....	37
3.2.2. Datasheet ολοκληρωμένων κυκλωμάτων .....	38
3.2.3. Αρχές λειτουργίας Διπολικών βηματικών κινητήρων .....	40
3.2.4. Πίνακας αληθείας για τη διέγερση διπολικών κινητήρων.....	40
3.2.5. Κατασκευή πλακέτας κυκλώματος οδήγησης .....	42
4. Εισαγωγή στη Visual Basic.....	43
4.1. Μελέτη της διεπαφής - Interface της πτυχιακής εργασίας.....	43
4.2. Πρώτη φόρμα.....	44
4.2.1. Δήλωση μεταβλητών .....	45
4.2.2. Αρχικοποίηση των μεταβλητών .....	46
4.2.3. Αποστολή και λήψη δεδομένων μέσω σειριακής θύρας.....	46
4.2.4. Προσανατολισμός ενός αντικειμένου.....	51
4.2.5. Μέτρηση στάθμης υγρών .....	53
4.2.6. Μέτρηση της θερμοκρασίας.....	54
4.2.7. Ενεργοποίηση συσκευών.....	55
4.2.8. Αποθήκευση των μετρούμενων θερμοκρασιών και της στάθμης υγρών .....	56
4.2.9. Δημιουργία μενού.....	57
4.3. Δεύτερη Φόρμα.....	59
4.4. Τρίτη Φόρμα .....	61
4.5. Κώδικας για τον αυτοματοποιημένο έλεγχο .....	62
4.6. Δήλωση κεντρικών Μεταβλητών .....	64
5. Χρήση λειτουργικής κάρτας.....	66
5.1. Εγκατάσταση προγράμματος με όνομα “scada”.....	66
5.2. Συνδεσμολογία λειτουργικής κάρτας με τον ηλεκτρονικό υπολογιστή.....	68
5.3. Χρήση της διεπαφής (interface).....	68
6. Συμπεράσματα.....	71
Βιβλιογραφία.....	73