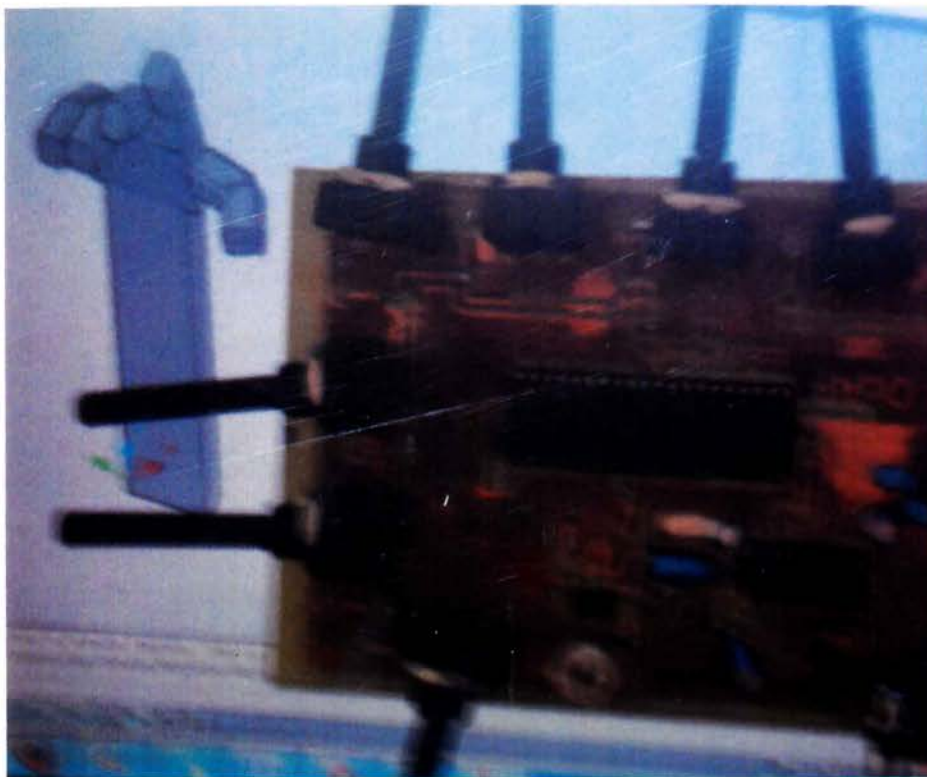


Πτυχιακή εργασία του τμήματος Αυτοματισμού



Τίτλος εργασίας: Αυτοματισμός του AutoCAD και έλεγχος του μέσω της visual basic και του at mega16 και η δημιουργία ενός Συστήματος συλλογής δεδομένων.

Επιβλέποντες καθηγητές: Κος Νικολάου Γρηγόριος, Κος Αλαφοδήμος Κωνσταντίνος

Όνομα σπουδαστή: Καπαρέλης Δημήτριος

Αρ. Μητρώου: 35169

Εξάμηνο Φοίτησης: ΙΑ

Ημερομηνία:

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
Πτυχιακή εργασία του Καπαρέλη Δημήτριου
Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

ΒΙΒΛΙΟΘΗΚΗ
ΤΕΙ ΠΕΙΡΑΙΑ

Περιεχόμενα

Κεφάλαιο 1

1.1)Ανάλυση της πτυχιακής εργασίας	5
1.2)Ανάλυση των ολοκληρωμένων που χρησιμοποιήθηκαν στο κύκλωμα	6
1.2.1 Ο μικροελεγκτής	5
1.2.2 Ανάλυση του MAX232	11
1.3)Ανάλυση των αντικειμένων και των βιβλιοθηκών που χρησιμοποιήθηκαν στη visual basic	13
1.3.1) Σειριακή θύρα	13
1.3.1.1) Τι είναι η σειριακή θύρα	13
1.3.1.2) η σειριακή θύρα στη visual basic	15
1.3.2) Οι χρονιστές	15
1.3.2.1) Η κλάση χρονιστής	15
1.3.3) Άλλα αντικείμενα που χρησιμοποιήθηκαν	16
1.3.4.1) Αυτοματισμός με τη visual basic	16
1.3.4.2) Αυτοματισμός του Autocad με τη visual basic	16
1.3.4.3) Τα αντικείμενα της Autodesk.AutoCAD.Interop.Common	16
1.3.5) Η βάση δεδομένων access και τα αντικείμενά της	18
1.4) Ανάλυση των περιφερικών που χρησιμοποιήθηκαν στον AT mega16	18
1.4.1) η σειριακή θύρα του atmega16	19
1.4.1.1) Γεννήτρια ρυθμού μετάδοσης	19
1.4.1.2) πομπός UART	21
1.4.1.3) ο δέκτης UART	21
1.4.1.4) Ο καταχωρητής δεδομένων της ασύγχρονης σειριακής μονάδας UDR	23
1.4.2) Η αναλογική θύρα	24

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
 Πτυχιακή εργασία του Καπαρέλη Δημήτριου
 Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

1.4.2.1) Η αναλογική μετατροπή	24
1.4.2.2) Η αναλογική θύρα του Atmega16	24
1.5) Λογική του κώδικα που χρησιμοποιήθηκε στον AT mega16 και στη visual basic και πως συνδυάστηκαν	26
1.5.1 Το Μπλοκ διάγραμμα του κώδικα του ATmega16 και η ανάλυσή του	26
1.5.2 Το μπλόκ διάγραμμα του κώδικα της visual basic	27

Κεφάλαιο 2

2.1)Ανάλυση του κυκλώματος της πλακέτας	29
2.1.1) το κύκλωμα του σταθεροποιητή	29
2.1.2)το κύκλωμα του μικροελεγκτή	30
2.1.3)Το κύκλωμα του MAX232	33
2.1.4)Το κύκλωμα όλης της πλακέτας και πως φτιάχτηκε	35

Κεφάλαιο3

3)Ανάλυση του κώδικα του ATMEGA16	37
3.2) Ο κώδικας του ATMEGA 16	37

Κεφάλαιο 4

4.1) Ανάλυση του κώδικα της visual basic για το AutoCAD	41
4.2) Η ανάλυση του γραφικού περιβάλλοντος της visual basic	41
4.3) Η πρώτη ενότητα της φόρμας της visual basic	43
4.4) Η δεύτερη ενότητα της φόρμας της visual basic	60
4.5) Η Τρίτη ενότητα της φόρμας της visual basic	74

Κεφάλαιο 5

5.1) Ανάλυση του κώδικα της visual basic για το Σ.Σ.Δ	85
5.2) Το πρώτο μέρος της φόρμας FrmSensor	89
5.3) Το δεύτερο μέρος της φόρμας	90

<u>5.4) Το τρίτο μέρος της φόρμας</u>	<u>91</u>
<u>5.5) Το τέταρτο κομμάτι της φόρμας</u>	<u>96</u>
<u>Κεφάλαιο 6</u>	
<u>6) Βιβλιογραφία</u>	<u>97</u>
<u>Παράρτημα Α</u>	
<u>Τα κυκλώματα της πτυχιακής</u>	<u>98</u>

Κεφάλαιο 1

1.1 Ανάλυση της πτυχιακής εργασίας

Αυτοματισμός είναι μία επιθυμητή διαδικασία η οποία γίνεται χωρίς τη παρουσία του ανθρώπου, για να επιτύχει ο αυτοματισμός ο σχεδιαστής πρέπει να δημιουργήσει ένα σύστημα το οποίο θα κάνει αυτή τη διεργασία.

Η πτυχιακή αυτή έχει σκοπό να αυτοματοποιήσει το AutoCAD 2009. Το AutoCAD είναι ένα από τα πιο γνωστά προγράμματα σχεδίασης. Το σύστημα σε αυτή τη πτυχιακή δημιουργεί ένα χέρι και ανάλογα με τα κουμπιά που πατάει ο χρήστης στη φόρμα ή περιστρέφοντας τα ποτενσιόμετρα της πλακέτας μπορεί αυτόματα να κάνει περιστροφή στο χέρι ή στα δάκτυλά του.

Επίσης η πτυχιακή αυτή περιέχει ένα σύστημα συλλογής δεδομένων μίας εισόδου και αναπαριστά γραφικά, στο excel και αποθηκεύει τις τιμές της σε ένα αρχείο access.



Σχήμα 1.1 Το βασικό διάγραμμα της πτυχιακής

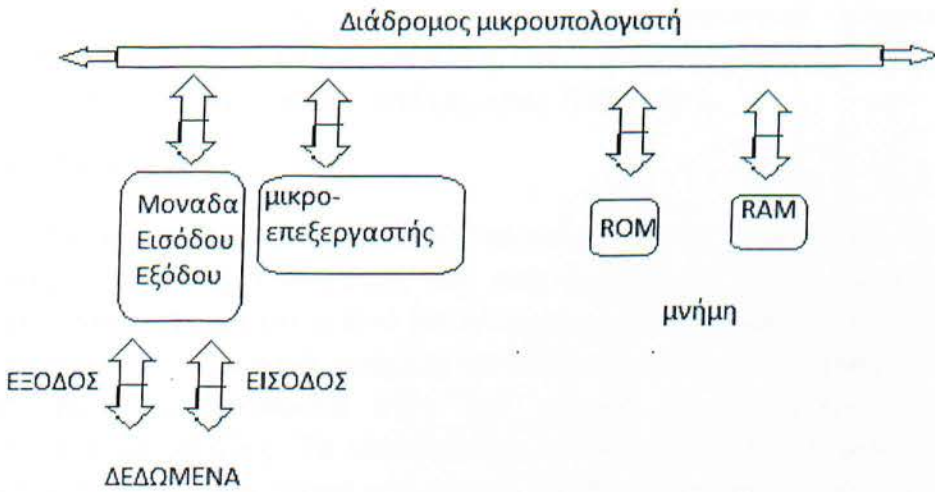
Όπως παρατηρείται και από το παραπάνω block διάγραμμα ο χρήστης περιστρέφοντας τα ποτενσιόμετρα αλλάζει τις αναλογικές τιμές που δέχεται ο μικροελεγκτής, ο μικροελεγκτής μετατρέπει τις τιμές αυτές σε ψηφιακές και τις στέλνει σειριακά στην RS232 του υπολογιστή. Ο υπολογιστής με τη βοήθεια της visual basic επεξεργάζεται τις τιμές αυτές και αφού έχει σχεδιάσει το χέρι στο AutoCAD περιστρέφει ανάλογα με την είσοδο και τη τιμή της κάποιο σημείο του χεριού. Επίσης ο χρήστης μπορεί να περιστρέφει τα σημεία του χεριού με τα κουμπιά που έχει η φόρμα της visual basic . Ο χρήστης επίσης με το πάτημα ενός κουμπιού μεταφέρεται σε μία νέα φόρμα όπου εκεί αναλύεται γραφικά και στο excel μία είσοδος όπου στη προκειμένη περίπτωση είναι το πρώτο από τα οκτώ ποτενσιόμετρα της πλακέτας. Αυτήν την είσοδο ο χρήστης μπορεί να την πολλαπλασιάσει με κάποιες τιμές ούτως ώστε να πάρει τη τιμή που αυτός θέλει να χρησιμοποιήσει και όχι αυτή που του δίνει το αισθητήριο (στη προκειμένη περίπτωση το ποτενσιόμετρο) και να της προσθέσει και φυσικό μέγεθος (βαθμοί Κελσίου, τάση κλπ). Επίσης μπορεί να αποθηκεύσει τις τιμές αυτές σε μία βάση δεδομένων ή να αναζητήσει παλαιότερες με βάση την ημερομηνία και την ώρα που αυτές σημειώθηκαν.

1.2 Ανάλυση των ολοκληρωμένων που χρησιμοποιήθηκαν σε αυτή τη πτυχιακή

Τα ολοκληρωμένα που χρησιμοποιήθηκαν σε αυτή τη πτυχιακή είναι τα at mega16 της Atmel και το MAX232 της MAXIM.

1.2.1 Ο μικροελεγκτής

Οι μικροϋπολογιστές είναι το προϊόν της τεχνολογίας VLSI. Το 1971 η τεχνολογία αυτή που αφορά στην κατασκευή ολοκληρωμένων κυκλωμάτων επέτρεπε τη υλοποίηση μεγάλου αριθμού λογικών πυλών ή τρανζίστορ σε μια επίπεδη ψηφίδα πυριτίου επιφάνειας της τάξεως του mm^2 . Άμεση ήταν η προσπάθεια αξιοποίησης της για την υλοποίηση τμημάτων ενός υπολογιστή όπως μιας απλής κεντρικής μονάδας επεξεργασίας(K.M.E.), της απαραίτητης μνήμης και μονάδων Εισόδου/Εξόδου δεδομένων σε ξεχωριστές ψηφίδες πυριτίου. Η K.M.E. ονομάστηκε Μικροεπεξεργαστής και το σύστημα που συντέθηκε με βάση τις ψηφίδες αυτές Μικροϋπολογιστής λόγω του μεγέθους και της απλότητάς του.



Σχήμα 1.2 Το βασικό διάγραμμα ενός Μικροπολογιστικού Συστήματος

Η απλότητά τους σχετιζόταν με το μικρό ρεπερτόριο εντολών (τάξεως μερικών δεκάδων) και επίσης με το μικρό μήκος λέξης δεδομένων (4-8 δυαδικών ψηφίων) που μπορούσαν να επεξεργαστούν ταυτόχρονα. Από τότε όμως πέρασε πολύς καιρός και το μόνο που έμεινε μικρό ήταν το φυσικό τους μέγεθος. Η βελτίωση της τεχνολογίας επέτρεψε τη γρήγορη αύξηση της εσωτερικής πολυπλοκότητάς τους. Έτσι όχι μόνο αυξήθηκε το ρεπερτόριο των εντολών και το μήκος των δεδομένων και των διευθύνσεων αλλά υλοποιήθηκαν και προχωρημένες αρχιτεκτονικές, που υπήρχαν παλιότερα μόνο σε υπολογιστές υψηλών επιδόσεων. Η πολυπλοκότητά τους τείνει προς το 1 δισεκατομμύριο τρανζίστορ. Ταυτόχρονα έχουμε και πολύ μεγάλη αύξηση στην ταχύτητα λειτουργίας τους. Πολύ μεγάλη αύξηση έχουμε όμως και στη χωρητικότητα των μνημών και στις περιφερειακές μονάδες των Μικροϋπολογιστών ακολούθησαν σε αύξηση πολυπλοκότητας και ταχύτητας.

Η εκρηκτική αυτή ανάπτυξη δημιούργησε μια μεγάλη ποικιλία διαφορετικών τύπων Μικροϋπολογιστών. Το ερώτημα βέβαια που προκύπτει είναι γιατί να κατασκευάσουμε χιλιάδες διαφορετικούς τύπους μικροϋπολογιστών-Μικροεπεξεργαστών αλλά το κυριότερο εκατοντάδες εκατομμύρια αντίτυπα τους. Η απάντηση είναι απλή. Χρειάζονται σχεδόν σε κάθε ηλεκτρονική συσκευή. Μπορούμε να διακρίνουμε τρεις κατηγορίες μικροεπεξεργαστών. Αυτούς που χρησιμοποιούνται σε υπολογιστές γενικής χρήσης, στη βιομηχανία και στην επεξεργασία σήματος.

Οι μικροεπεξεργαστές που χρησιμοποιούνται στους προσωπικούς υπολογιστές για εφαρμογές γραφείου, όπως επεξεργασία κειμένου ή για εφαρμογές που απαιτούν

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

αριθμητικές πράξεις έχουν προχωρημένα αρχιτεκτονικά χαρακτηριστικά. Στους μικροεπεξεργαστές αυτούς η επιδίωξη είναι να έχουμε υψηλές επιδώσεις σε ένα ευρύ φάσμα εφαρμογών(Pentium, επεξεργαστές RISC κλπ).

ΜΙΚΡΟΕΛΕΓΚΤΕΣ

Η δεύτερη κατηγορία μικροεπεξεργαστών βρίσκει εφαρμογή στις ηλεκτρονικές συσκευές, στις τηλεπικοινωνίες και στη βιομηχανία. Κάθε συσκευή, μηχάνημα ή ηλεκτρονικό σύστημα εκτός από τον έλεγχο της σωστής και αποδοτικής λειτουργίας του χρειάζεται επίσης χειρισμό, ρύθμιση και παρακολούθηση. Το ζητούμενο στις εφαρμογές αυτές είναι να βρίσκονται στην ίδια ψηφίδα με τον μικροεπεξεργαστή και οι περιφερειακές μονάδες. Τα ολοκληρωμένα κυκλώματα που περιλαμβάνουν στην ίδια ψηφίδα εκτός από τον μικροεπεξεργαστή και θύρες για την είσοδο- έξοδο δεδομένων(σε παράλληλη και σειριακή μορφή, μετατροπείς αναλογικών σημάτων σε ψηφιακή μορφή και αντίστροφα, χρονιστές και μνήμες(ROM, RAM)είναι υπεύθυνος για την έξοδο, την επεξεργασία, αποθήκευση και μετάδοση των αναλογικών και ψηφιακών σημάτων μιας εφαρμογής.

Είναι φανερό ότι οι μικροελεγκτές και οι μικροεπεξεργαστές αποτελούν απαραίτητο μέρος σύγχρονων τεχνολογικών εφαρμογών. Ωστόσο, οι δύο αυτές συσκευές διαφέρουν μεταξύ τους σε αρκετά σημεία. Μια βασική διαφορά τους και η πιο σημαντική βρίσκεται στη λειτουργικότητά τους. Προκειμένου να λειτουργήσει ένας μικροεπεξεργαστής θα πρέπει να συνδεθεί και με άλλες συσκευές όπως μνήμη ή συσκευή αποστολής και λήψης δεδομένων. Αυτό σημαίνει ότι ένας μικροεπεξεργαστής είναι η καρδιά του συστήματος. Αντίθετα ένας μικροελεγκτής σχεδιάζεται με τέτοιο τρόπο, ώστε να περιέχει όλες τις παραπάνω συσκευές. Συνεπώς δεν χρειαζόμαστε άλλες συσκευές για την λειτουργία του, εφόσον όλα τα απαραίτητα περιφερικά είναι ενσωματωμένα μέσα του. Με αυτό τον τρόπο, εξοικονομούμε χώρο και χρόνο, κατά την κατασκευή ενός συστήματος που βασίζεται σε μικροελεγκτή.

ΕΠΕΞΕΡΓΑΣΤΕΣ ΣΗΜΑΤΟΣ

Πέρα όμως από τις εφαρμογές ελέγχου και αυτοματισμών οι μικροϋπολογιστές χρησιμοποιούνται στο χειρισμό και στην επεξεργασία σημάτων. Στη σημερινή ψηφιακή εποχή, κάθε είδος σήματος όπως ήχος, εικόνα κλπ δειγματοληπτείται και η αντίστοιχη στιγμιαία τιμή του μετατρέπεται σε ένα δυαδικό αριθμό. Στη νέα του αυτή μορφή το λεγόμενο ψηφιακό σήμα προσφέρει πολλά πλεονεκτήματα. Τα πλεονεκτήματα της πληροφορίας είναι στην επεξεργασία η ευελιξία και η υλοποίηση αλγορίθμων με λογισμικό. Στην αποθήκευση η μεγάλη χωρητικότητα, η ακρίβεια, η ελαχιστοποίηση των σφαλμάτων, η ευκολία και η ταχύτητα ανάκτησης των δεδομένων. Στη μετάδοση οι υψηλές ταχύτητες η ευκολία και η ευελιξία η υλοποίηση με λογισμικό πρωτοκόλλων , δρομολόγησης και ελέγχου σφαλμάτων η ακρίβεια και η ελαχιστοποίηση των

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καταρέλη Δημήτριου

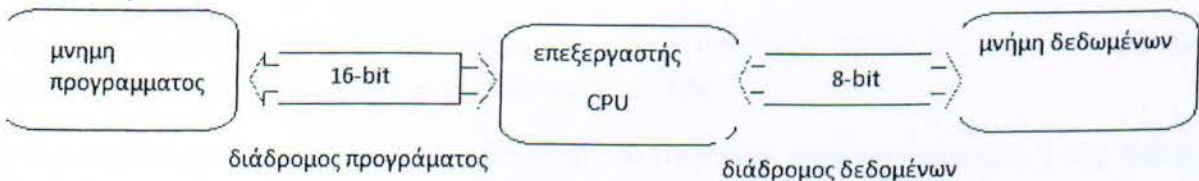
Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

σφαλμάτων. Χωρίς τη χρήση των μικροεπεξεργαστών δε θα ήταν τεχνολογικά και οικονομικά εφικτή η υλοποίηση της ψηφιακής τεχνολογίας. Ακόμη και στις επικοινωνίες όπου απαιτείται ηλεκτρονική επεξεργασία σε πολύ υψηλές ταχύτητες (μικροκύματα, ραδιοσυχνότητες) προβλέπεται σύντομα τα αντίστοιχα συστήματα να υλοποιούνται με τη χρήση νέας γενιάς γρήγορων A/D και D/A μετατροπέων και γρήγορων μικροεπεξεργαστών.

Στις εφαρμογές αυτές η έμφαση είναι στη ταχύτητα και στη βέλτιστη υλοποίηση ειδικών αλγορίθμων όπως των ψηφιακών φίλτρων, του γρήγορου μετασχηματισμού Fourier(FFT) κλπ. Οι μικροεπεξεργαστές που είναι εφοδιασμένοι με αυτές τις ιδιότητες ονομάζονται επεξεργαστές σήματος (Signal processors).

ΜΙΚΡΟΕΛΕΝΚΤΕΣ AVR

Ο μικροελεγκτής AVR περιέχει έναν πύρινα (ΚΜΕ) επεξεργαστή RISC ο οποίος έχει σχεδιαστεί με βάση την αρχιτεκτονική HARVARD. Όπως φαίνεται από το σχήμα στην αρχιτεκτονική αυτή υπάρχει διαφορετικός διάδρομος για τη μεταφορά δεδομένων για τη μεταφορά των εντολών και τη μνήμη προγράμματος (program memory) αυτό καθιστά την αρχιτεκτονική HARVARD αποδοτική αφού μπορεί να εκτελείται ή να διαβάζεται μνήμη. Με τον τρόπο αυτό επιτυγχάνεται η εκτέλεση των περισσότερων εντολών σε ένα μόνο κύκλο μηχανής.



Τα γενικά χαρακτηριστικά των μικροελεγκτών AVR είναι τα παρακάτω:

- A. Οι μικροελεγκτές διαθέτουν ενσωματωμένη μνήμη flash με δυνατότητα προγραμματισμού εντός του συστήματος ISP (In-System-Programmable) σαν μνήμη προγράμματος. Δηλαδή δεν είναι απαραίτητες εξωτερικές μνήμες τύπου ROM-EEPROM που να περιέχουν τον κώδικα του προγράμματος και δεν χρειάζεται για τον προγραμματισμό τους να μεταφερθούν σε κάποια εξωτερική συσκευή.
- B. Οι μικροελεγκτές διαθέτουν 32 καταχωρητές εργασίας των 8 bit. Συνεπώς υπάρχουν αρκετές θέσεις ώστε οι διάφορες μεταβλητές να μπορούν να αποθηκεύονται εντός της

ΚΜΕ αντί των θέσεων μνήμης, όπου η διαδικασία πρόσβασης απαιτεί περισσότερο χρόνο.

Γ. Οι μικροελεγκτές διαθέτουν ενσωματωμένη μνήμη δεδομένων (data memory) τύπου EPROM και SRAM στα περισσότερα μοντέλα σειράς. Οι μνήμες αυτές χρησιμοποιούνται για την αποθήκευση σταθερών τιμών και μεταβλητών.

Δ. Διαθέτουν ενσωματωμένο προγραμματιζόμενο χρονιστή με μονάδα διαίρεσης συχνότητας (prescaler). Η μονάδα αυτή χρησιμοποιείται σε εφαρμογές όπου η ακρίβεια του χρονισμού είναι κρίσιμη.

Ε. προγραμματιζόμενο χρονιστή επιτήρησης(Watch Dog Timer-WDT) ο συνήθης ρόλος του είναι η αποφυγή εσφαλμένων λειτουργιών σε περίπτωση κατάρρευσης του προγράμματος.

ΣΤ. Έως 4 παράλληλες θύρες των (8-bit) και πληθώρα περιφερικών όπως UART,SPI, αναλογικός συγκριτής, μετατροπέας ADC, κλπ. Για την υποστήριξη των θυρών εισόδου εξόδου διαθέτει έως και 64 καταχωρητές .

Ζ. δυνατότητα για εσωτερικές και εξωτερικές διακοπές

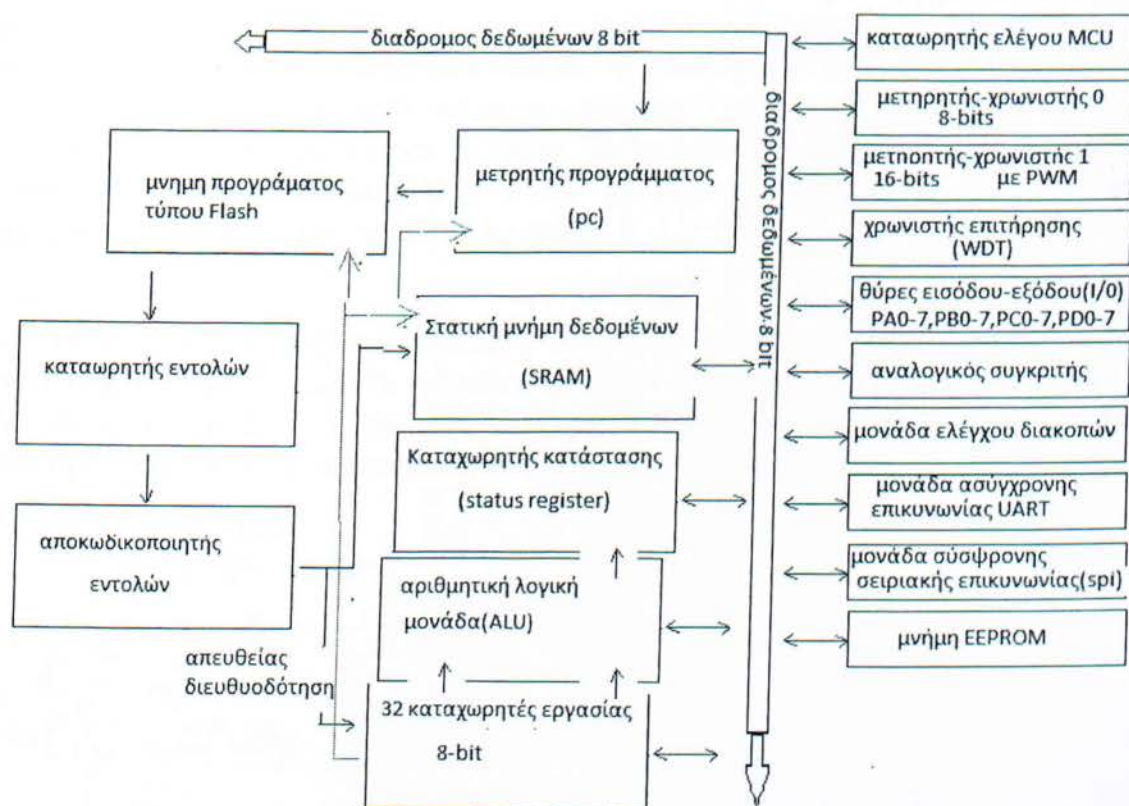
Η. Λειτουργία σε συχνότητες χρονισμού από 0 ως 10 MHz. Καθώς οι περισσότερες εντολές ολοκληρώνονται σε μία περίοδο κεντρικού σήματος χρονισμού(το ρολόι της ΚΜΕ), η επίδοση είναι ανώτερη σε σύγκριση με τους κλασικούς μικροελεγκτές που λειτουργούν στην ίδια συχνότητα χρονισμού αλλά απαιτούν για την εκτέλεση μιας εντολής περισσότερους κύκλους ρολογιού.

Θ. Διάφορα μοντέλα διαθέτουν ενσωματωμένο ταλαντωτή τύπου RC για επιλογή της συχνότητας χωρίς τη χρήση εξωτερικού κρυστάλλου.

Ι. Εσωτερικό κύκλωμα που επανατοποθετεί το σύστημα κατά την εφαρμογή της τάσης τροφοδοσίας.(Power on reset-POR)

Κ. Λειτουργίες αποκοπής (Power down) και ηρεμίας(sleep). Ο επεξεργαστής τίθεται στις καταστάσεις αυτές όταν δεν υπάρχει δραστηριότητα, ούτως ώστε να μειωθεί η κατανάλωση ισχύος.

Ο AT mega16 έχει 40 ακροδέκτες 16KB flash memory 512B EEPROM 1024 B RAM και διαθέτει UART ADC



Αρχιτεκτονική των μικροελεγκτών AVR και τα περιφερικά τους.

ΑΡΙΘΜΙΤΙΚΗ ΛΟΓΙΚΗ ΜΟΝΑΔΑ(ALU)

Είναι η μονάδα αριθμητικών λογικών πράξεων του μικροελεγκτή. Αυτή εκτελεί αριθμητικές, λογικές και σε επίπεδο bit εντολές μεταξύ καταχωρητών εργασίας ή και καταχωρητή εργασίας με σταθερά και αποθηκεύει το αποτέλεσμα στον καταχωρητή εργασίας που δείχνει η εντολή στη διάρκεια μιας περιόδου. Κατά την εκτέλεση των πράξεων ενημερώνονται οι σημαίες του καταχωρητή κατάστασης .

1.2.2 Ανάλυση του MAX232

Πολλές φορές χρειαζόμαστε ένα απλό RS232 interface, μεταξύ κάποιου κυκλώματος TTL/Cmos και της RS-232, η οποία σε ορισμένα Laptops ειδικότερα, δεν μπορεί να

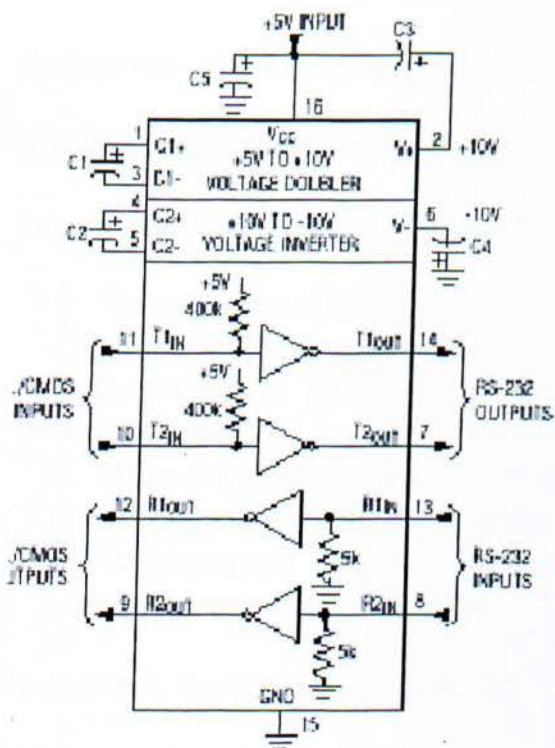
Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

λειτουργήσει αν δεν υπάρχει συμμετρική στάθμη τάσεως. Λέγοντας TTL output Level εννοούμε ότι, το λογικό 1 (Hi) λαμβάνει την τιμή των +5 Volts και το λογικό 0 (Low) ισούται με 0 V. Στην περίπτωση των CMOS, το επίπεδο "Hi" μπορεί να είναι από +5 έως +18V, αλλά και πάλι το "Low" είναι μηδενικού Voltage. Η στάθμες αυτές είναι ασύμβατες για κάποιες COM Ports, οι οποίες για να λειτουργήσουν αποδεκτά, απαιτούν το λογικό "1" να είναι ίσο με + 5-12 V και το "0" ίσο με -5 έως -12 Volts, δηλαδή απαιτούν συμμετρικότητα εν σχέση με το μηδέν.

Το MAX 232 είναι ένα ειδικό IC, σκοπός του οποίου είναι να παρέχει "συμβατότητα" μεταξύ ενός κυκλώματος με TTL ή Cmos στάθμη και μίας άλλης πόρτας, που απαιτεί + / - V, όπως η RS-232. Το κύκλωμα είναι σχεδιασμένο έτσι ώστε, απαιτεί για την λειτουργία του μόνο 5 εξωτερικούς πυκνωτές.



1.3 Ανάλυση των βιβλιοθηκών και των αντικειμένων που χρησιμοποιήθηκαν στον κώδικα της visual basic

1.3.1 Σειριακή θύρα

1.3.1.1 Τι είναι η σειριακή θύρα

Το **RS-232** (*Recommended Standard 232*) είναι ένα πρότυπο για σειριακή μετάδοση δυαδικών σημάτων δεδομένων μεταξύ ενός *DTE* (Data terminal equipment) και ενός *DCE* (Data Circuit-terminating equipment). Χρησιμοποιείται συχνά στις σειριακές θύρες των προσωπικών υπολογιστών. Ένα παρόμοιο πρότυπο της ITU-T είναι το **V.24**.

Χρησιμοποιείται ένας αγωγός για εκπομπή δεδομένων, ένας για λήψη και μία γείωση.

[Επεξεργασία] Εύρος ορισμού του προτύπου

Το πρότυπο RS-232-C της Electronics Industries Alliance (EIA) ^[1] του 1969 ορίζει:

- Χαρακτηριστικά ηλεκτρικών σημάτων όπως επίπεδα τάσης, ρυθμό μετάδοσης, χρονισμό και ρυθμό ανόδου των σημάτων, ανώτατο επίπεδο τάσης, συμπεριφορά σε βραχυκύκλωμα, μέγιστη παρασιτική χωρητικότητα και μήκος καλωδίου.
- Μηχανικά χαρακτηριστικά διεπαφής, συνδέσιμα βύσματα και προσδιορισμό pins.
- Λειτουργίες του κάθε κυκλώματος στο βύσμα διεπαφής.
- Τυποποιημένα υποσύνολα των κυκλωμάτων διεπαφής για επιλεγμένες τηλεπικοινωνιακές εφαρμογές.

Το πρότυπο δεν ορίζει στοιχεία όπως:

- κωδικοποίηση χαρακτήρων (για παράδειγμα, ASCII, Baudot ή EBCDIC)
- τη διαμόρφωση των χαρακτήρων στη ροή δεδομένων (μπιτ ανά χαρακτήρα, μπιτ έναρξης/διακοπής, ισοτιμία)
- πρωτόκολλα για εντοπισμό σφαλμάτων ή αλγόριθμους για συμπίεση δεδομένων.
- ρυθμούς μπιτ για μετάδοση, αν και το πρότυπο αναφέρει ότι προορίζεται για ρυθμούς μπιτ μικρότερους από 20.000 μπιτ ανά δευτερόλεπτο. Πολλές μοντέρνες συσκευές υποστηρίζουν ταχύτητες 115.200 bps και άνω.
- τροφοδοσία ρεύματος σε εξωτερικές συσκευές.

Λεπτομέρειες της διαμόρφωσης χαρακτήρων και του ρυθμού μετάδοσης μπιτ ελέγχονται από το υλικό της σειριακής θύρας, συχνά ένα μοναδικό ολοκληρωμένο κύκλωμα, το UART, το οποίο μετατρέπει δεδομένα από παράλληλη σε σειριακή μορφή. Μια τυπική σειριακή θύρα περιλαμβάνει εξειδικευμένα ολοκληρωμένα κυκλώματα για τον οδηγό και το δέκτη, ώστε να μετατρέπει μεταξύ εσωτερικών λογικών επιπέδων και συμβατών με το RS-232 επιπέδων σήματος.

[Επεξεργασία] Ιστορία

Τα πρώτα DTEs ήταν ηλεκτρομηχανικά τηλέτυπα και τα πρώτα DCEs ήταν (συνήθως) μόντεμ. Όταν άρχισαν να χρησιμοποιούνται τερματικά, ήταν συχνά σχεδιασμένα με τέτοιο τρόπο ώστε να είναι συνδιαλέξιμα με τα τηλέτυπα, και έτσι υποστήριζαν το RS-232. Η αναθεώρηση C του προτύπου εκδόθηκε το 1969 για να προσαρμόσει τα ηλεκτρικά χαρακτηριστικά αυτών των συσκευών στο πρότυπο.

Αφού η εφαρμογή σε συσκευές όπως υπολογιστές, εκτυπωτές, πειραματικά όργανα, και ούτω καθεξής δεν λαμβανόταν υπόψη από το πρότυπο, οι σχεδιαστές που υλοποιούσαν μια συμβατή με το RS-232 διεπαφή του εξοπλισμού τους συχνά ερμήνευαν τις απαιτήσεις ιδιόμορφα. Συχνά προβλήματα ήταν η μη πρότυπη απόδοση ακίδων των κυκλωμάτων στα βύσματα, και λανθασμένα ή απόντα σήματα ελέγχου. Η έλλειψη συνέπειας προς τα πρότυπα παρήγε μια ακμάζουσα βιομηχανία breakout κουτιών, κουτιών patch, δοκιμαστικού εξοπλισμού, βιβλίων, και άλλων βοηθημάτων για την σύνδεση αταίριαστων συσκευών. Μια κοινή απόκλιση από το πρότυπο ήταν η οδήγηση των σημάτων σε μειωμένο δυναμικό: το πρότυπο απαιτεί ο πομπός να χρησιμοποιεί +12 V και -12 V, αλλά απαιτεί από τον δέκτη να ξεχωρίζει δυναμικά μικρά όσο +3 V και -3 V. Μερικοί κατασκευαστές λοιπόν κατασκεύαζαν πομπούς οι οποίοι παρείχαν +5 V και -5 V αποκαλώντας τους «RS-232 συμβατούς».

Μεταγενέστεροι προσωπικοί υπολογιστές (και άλλες συσκευές) ξεκίνησαν να κάνουν χρήση του προτύπου ώστε να μπορούν να συνδέσουν υπάρχοντα εξοπλισμό. Για πολλά χρόνια, μια θύρα συμβατή με το RS-232 ήταν το χαρακτηριστικό γνώρισμα για τις σειριακές επικοινωνίες, όπως οι συνδέσεις μέσω μόντεμ, σε πολλούς υπολογιστές. Παρέμεινε σε ευρεία χρήση μέχρι το τέλος της δεκαετίας του 1990, και παρότι έχει σε μεγάλο βαθμό αντικατασταθεί από άλλα πρότυπα διεπαφής σε προϊόντα υπολογιστών, χρησιμοποιείται ακόμη για να συνδέσει παρωχημένα περιφερειακά, βιομηχανικό εξοπλισμό (όπως βασισμένο σε PLCs), και θύρες κονσόλας.

Το πρότυπο έχει μετονομαστεί αρκετές φορές στη διάρκεια της ιστορίας του καθώς ο εγγυητής οργανισμός άλλαζε το όνομά του, και έχει γίνει γνωστό ως EIA RS 232, EIA

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

232, και πιο πρόσφατα TIA 232. Το πρότυπο συνεχίζει να αναθεωρείται και να αναβαθμίζεται από την EIA και από το 1988 από την TIA^[2]. Η Αναθεώρηση C εκδόθηκε σε ένα έγγραφο τον Αύγουστο του 1969. Η Αναθεώρηση D εκδόθηκε το 1986. Η τρέχουσα αναθεώρηση είναι η *TIA-232-F Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*, η οποία εκδόθηκε το 1997. Αλλαγές από την Αναθεώρηση C, στον χρονισμό και σε λεπτομέρειες, αποσκοπούσαν στο να βελτιώσουν την αρμονική συνεργασία με το CCITT πρότυπο V.24, αλλά ο εξοπλισμός που έχει κατασκευαστεί με το τρέχον πρότυπο συνεργάζεται με παλαιότερες εκδόσεις.

1.3.1.2 η σειριακή θύρα στη visual basic

Στο προγραμματισμό της σειριακής θύρας χρησιμοποιήθηκε η κλάση `System.IO.Ports.serialport`. Έτσι μπορούμε να δημιουργήσουμε όσα αντικείμενα θέλουμε αυτής της κλάσης. Στη πτυχιακή αυτή χρησιμοποιήθηκε μόνο ένα αντικείμενο το `sp1`. Με αυτή τη κλάση υποστηρίζονται οι κωδικοποιήσεις `ASCIIEncoding`, `UTF8Encoding`, `Unicode Encoding`, `UTF32Encoding`, και οποιαδήποτε κωδικοποίηση που καθορίζεται σε `mscorlib.dll`. Στη πτυχιακή αυτή χρησιμοποιήθηκε η κωδικοποίηση `ASCIIEncoding`. Οι μέθοδοι της κλάσης σειριακή επικοινωνία της `visual basic` είναι το `bound rate` (Το `baud rate` είναι ο ρυθμός μετάδοσης των μεταβολών της στάθμης δυναμικού (τάσης) της γραμμής επικοινωνίας), `data bits` (ο αριθμός των `bits` της πληροφορίας), `handshake` (χειραψία), `parity` κλπ. Τα στοιχεία ελέγχου που δημιουργούνται από αυτή τη κλάση είναι αόρατα στη φόρμα.

1.3.2 Οι χρονιστές

1.3.2.1 Η κλάση χρονιστής

Η κλάση χρονιστής της `visual basic` τρέχει ένα σημείο του κώδικα μόλις ο χρονιστής εκπνεύσει. Στο πρόγραμμα χρησιμοποιήθηκαν τρία αντικείμενα της κλάσης χρονιστής το `timer`. Οι μέθοδοι της κλάσης `timer` είναι οι `interval` (ο χρόνος που χρειάζεται για να εκπνεύσει ο χρονιστής) κλπ. Τα στοιχεία ελέγχου που δημιουργούνται από αυτή τη κλάση είναι αόρατα στη φόρμα.

1.3.3 Άλλα αντικείμενα που χρησιμοποιήθηκαν

Τα υπόλοιπα αντικείμενα που χρησιμοποιήθηκαν σε αυτή τη πτυχιακή είναι απλά στοιχεία ελέγχου που χρησιμοποιήθηκαν για τη λειτουργικότητα της φόρμας. Αντικείμενα όπως κουμπιά, ετικέτες, κουτιά κειμένου για την εισαγωγή και εξαγωγή των δεδομένων .

1.3.4.1 Αυτοματισμός με τη visual basic

Για να χρησιμοποιηθούν τα αντικείμενα ενός προγράμματος που υποστηρίζει αυτοματοποίηση, . Πρέπει πρώτα να αναφερθεί η βιβλιοθήκη τύπων(type library) του συγκεκριμένου προγράμματος. Η βιβλιοθήκη τύπων ενός προγράμματος είναι ένα αρχείο που περιέχει μια περιγραφή του μοντέλου-αντικειμένου του προγράμματος. Αφού αναφερθεί, η βιβλιοθήκη τύπων ενός διακομιστή αυτοματοποίησης(που ονομάζεται επίσης συστατικό), μπορούμε να προσπελάσουμε τα αντικείμενα του διακομιστή σαν να ήταν εσωτερικά αντικείμενα της visual basic.

1.3.4.2 Αυτοματισμός του AutoCAD με τη visual basic

Η βιβλιοθήκη που χρησιμοποιήθηκε σε αυτή τη πτυχιακή για τον αυτοματισμό του AutoCAD είναι η "Autodesk.AutoCAD.Interop.Common". Με αυτή τη βιβλιοθήκη έχουμε πλήρη πρόσβαση στο autocad μπορούμε κυριολεκτικά να κάνουμε ότι θέλουμε σε αυτό. Επίσης μπορούμε να χειριστούμε το autocad και από τη visual basic αλλά και από τα εργαλεία του autocad αρκεί αυτό να έχει προβλεφθεί στο σχεδιασμό του προγράμματος γιατί υπάρχει κίνδυνος το πρόγραμμα της visual basic να πέσει σε αναδρομή.

1.3.4.3 Τα αντικείμενα της Autodesk.AutoCAD.Interop.Common

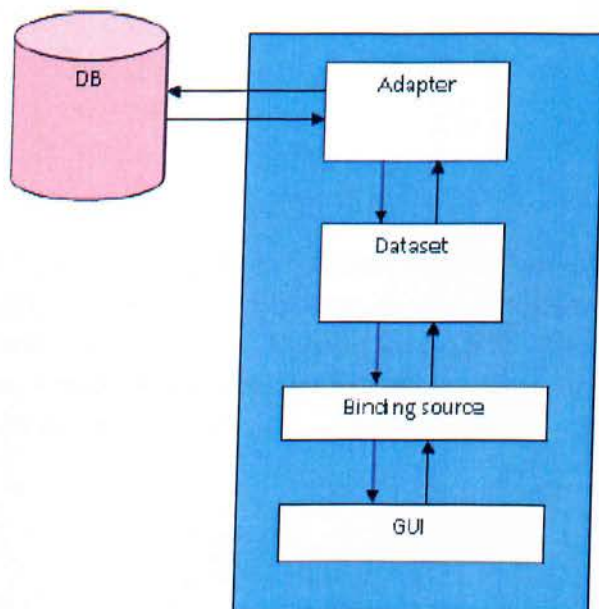
1)Το αντικείμενο Autodesk.AutoCAD.Interop.AcadApplication. Με το αντικείμενο αυτό δημιουργώ μια σύνδεση με το autocad. Στη πτυχιακή αυτή δημιουργήθηκε μόνο ένα αντικείμενο αυτού του τύπου το acadapp. Ουσιαστικά δημιουργώ μια νέα εφαρμογή και πάνω σε αυτή δημιουργώ το χέρι. Με αυτό το αντικείμενο θα μπορούσα, εάν ήθελα να .άνοιξω μία οποιαδήποτε εφαρμογή και να την τροποποιήσω (κάτι που δεν χρησιμοποιείτε σε αυτή τη πτυχιακή) ή να δημιουργήσω και να άνοιξω ταυτόχρονα όσα αρχεία autocad ήθελα. Η μέθοδος του αντικειμένου αυτού που χρησιμοποίησα είναι update για να μπορώ να ενημερώνω το autocad, όποτε θέλω, για τις αλλαγές που θέλω να κάνει.

2) Το αντικείμενο AcadDocument. Το αντικείμενο αυτό είναι το αντικείμενο που μου προσδιορίζει για το πιο αρχείο autocad χειρίζομαι, στη πτυχιακή αυτή το AcadDocument το προσκόμω στη ενεργό αρχείο του αντικειμένου Autodesk.AutoCAD.Interop.AcadApplication(acadDoc = acadapp.ActiveDocument)

3) Το αντικείμενο AcadLine. Με το αντικείμενο αυτό μπορούμε να σχεδιάσουμε γραμμές στο autocad. Για να γίνει αυτό θα πρέπει πρώτα να δημιουργήσουμε 2 πίνακες κάποιου τύπου που υποστηρίζει αριθμούς(εδώ χρησιμοποίησα τον τύπο δεδομένων double) 3 σειρών, αυτό είναι απαραίτητο να γίνει για να δηλώσουμε την αρχή και το τέλος της γραμμής και στους τρεις άξονες του autocad. Μία μέθοδος αυτού του αντικειμένου είναι η AddLine με αυτή τη μέθοδο μπορούμε να προσθέσουμε τη γραμμή .πχ lineYxeri = acadDoc.ModelSpace.AddLine(APlineYxeri, TPlineYxeri)

4) Το αντικείμενο Acad3DSolid. Με το αντικείμενο αυτό μπορούμε να δημιουργήσουμε τρισδιάστατα αντικείμενα στο autocad πολύ εύκολα και απλά όπως κύβους σπείρες, κώνους, κύκλους κλπ. με τη βοήθεια αυτού του αντικειμένου στη πτυχιακή αυτή δημιουργία κυλίνδρους και κύβους στο autocad με μόνο μία εντολή. Πχ palamiObj = acadDoc.ModelSpace.AddBox(centerPalami, XPalami, YPalami, ZPalami). Έτσι έχω δημιουργήσει ένα κύβο τον palamiObj με κέντρο το centerPalami, μήκος XPalami κλπ. εάν άλλαζα το AddBox σε AddCylinder θα δημιουργούσα ένα κύλινδρο. Το αντικείμενο Acad3DSolid έχει πολλές μεθόδους όπως διαγραφή, περιστροφή αλλαγή χρώματος κλπ, σε αυτή τη πτυχιακή χρησιμοποιήθηκαν μόνο οι μέθοδοι πρόσθεσης των αντικειμένων στο autocad η μέθοδος περιστροφής τους και η μέθοδος παρέμβασης ελέγχου αυτών των αντικειμένων. στη πτυχιακή τα δάκτυλα η παλάμη και γενικά όλο το χέρι δεν είναι ούτε κύβοι ούτε κύλινδροι είναι ουσιαστικά κύβοι χωρίς γωνίες για να το επιτύχω αυτό δημιουργήσα ένα κύβο και ένα κύλινδρο με ίδιες διαστάσεις και θέση στο χώρο του autocad και τα πρόσθεσα, πχ thebratsoObj = bratsoObj.CheckInterference(Kbratso, True) με αυτή την γραμμή κώδικα το αντικείμενο thebratsoObj είναι ένας κύβος και ένας κύλινδρος (bratsoObj, Kbratso) ουσιαστικά είναι σαν να έχουμε προσθέσει τα αντικείμενα που είναι κύβος και κύλινδρος

1.3.5 Η βάση δεδομένων access και τα αντικείμενά της



Πρέπει να αναφερθεί πως η Visual Basic μπορεί εύκολα να συνδεθεί με μία βάση δεδομένων με πολύ λίγο κώδικα αλλά με κάποιες επιλογές. Στο παραπάνω σχήμα βλέπουμε πως μία βάση διαχειριζόμαστε μία βάση δεδομένων με κάποια αντικείμενα της visual basic. Στη προκειμένη περίπτωση το GUI είναι το πρόγραμμά μας το Binding source είναι τα combobox και τα textbox το dataset είναι το αντικείμενο που αποθηκεύονται προσωρινά τα δεδομένα ενώ ο adapter είναι το αντικείμενο που είναι υπεύθυνο για τη μεταφορά των δεδομένων μας στη βάση.

1.4 Ανάλυση των περιφερικών που χρησιμοποιήθηκαν στον at mega16

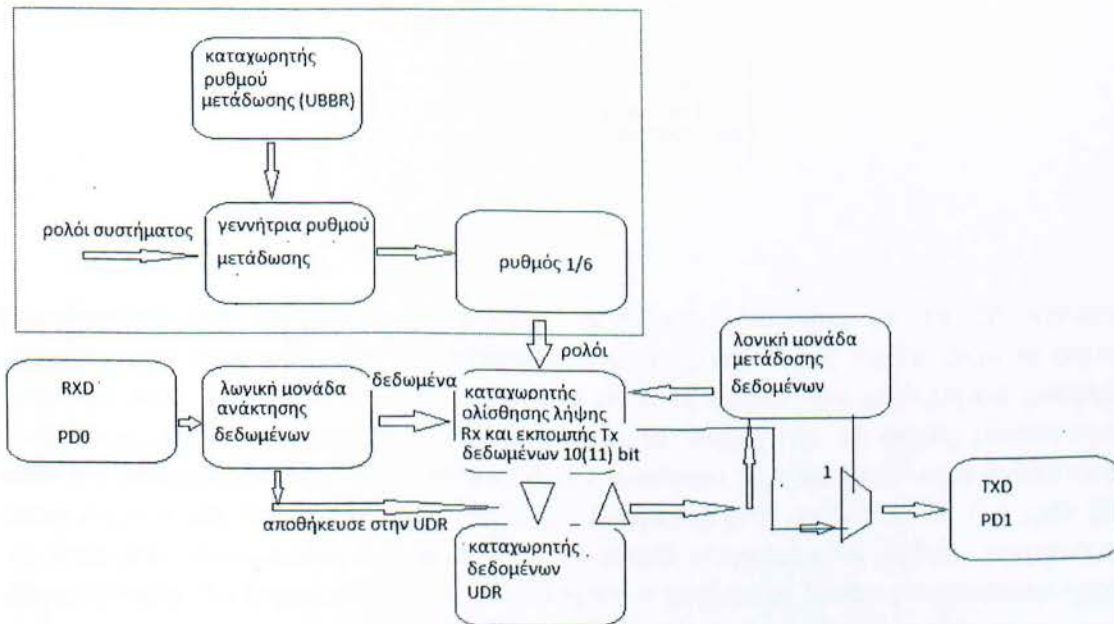
Τα περιφερειακά που χρησιμοποιήθηκαν στον atmega16 για αυτή τη πτυχιακή είναι η σειριακή επικοινωνία του και η αναλογική είσοδος του.

1.4.1) η σειριακή θύρα του atmega16

Η μονάδα UART είναι ένα ισχυρό και χρήσιμο περιφερειακό. Χρησιμοποιείται για αποστολή και λήψη δεδομένων από έναν υπολογιστή, για αποσφαλμάτωση κώδικα, για επικοινωνία με τον χρήστη κλπ. Η μονάδα UART μπορεί να μεταδώσει τους εξής 30 συνδυασμούς πλαισίου:

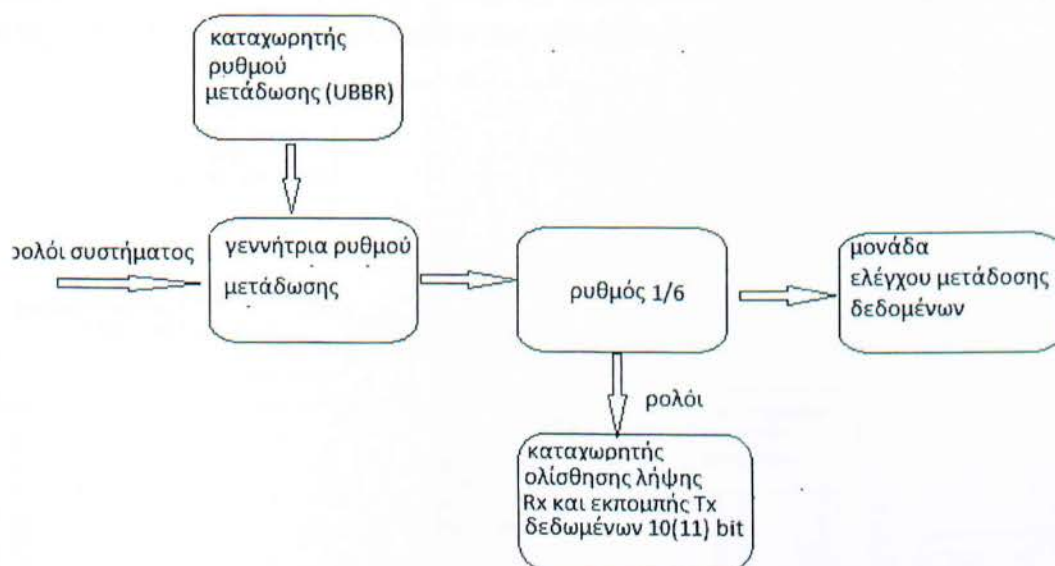
- 1 bit εκκίνησης
- 5,6,7,8 ή 9 bit
- Ένα ή δύο bit λήξης
- Καμία, άρτια ή περιπτή ισοτιμία.

Φιλτράρει τα ληφθέντα δεδομένα και ανιχνεύει σφάλματα πλαισίου και υπέρβασης. Διαθέτει τρία σήματα διακοπών και επιτρέπει μεγάλη ροή δεδομένων με απομονωτές υλοποιημένους από λογισμικό. Διαθέτει πομπό και δέκτη οι οποίοι μοιράζονται τη γεννήτρια του ρυθμού μετάδοσης και τους καταχωρητές ελέγχου. Η βασική δομή τους φαίνεται στο σχήμα



1.4.1.1 Γεννήτρια ρυθμού μετάδοσης

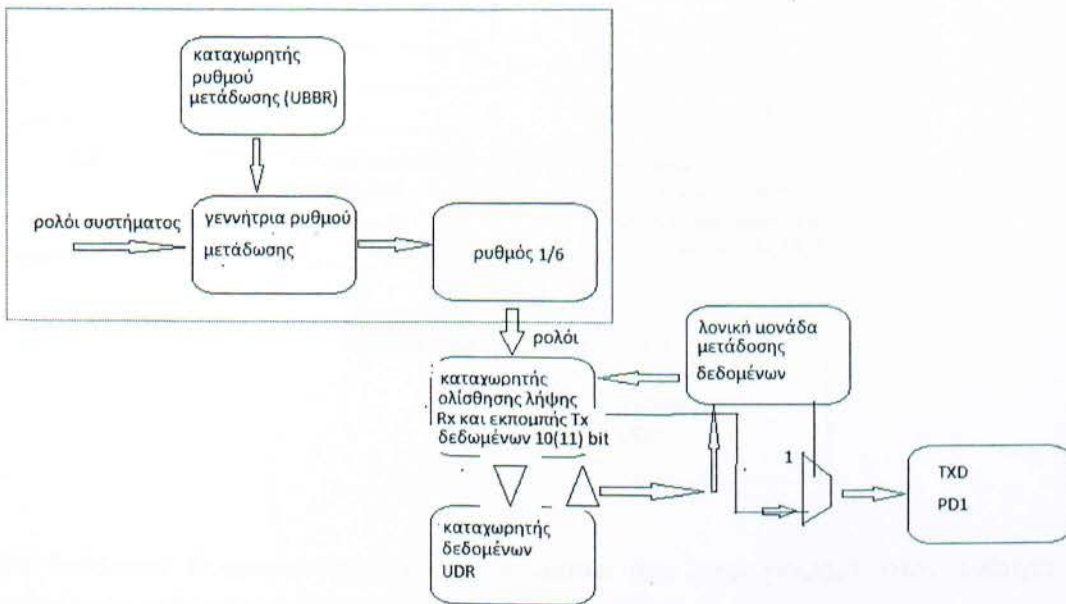
Η γεννήτρια ρυθμού μετάδοσης καθορίζει το σήμα ρολογιού που χρησιμοποιείται για μετάδοση και λήψη δεδομένων μέσω UART. Το σήμα αυτό διαβαθμίζεται με μεγάλη ακρίβεια, με αποτέλεσμα μια σχεδόν αλάνθαστη μεταφορά δεδομένων.



Παρατηρείται στο σχήμα, ότι ο ρυθμός μετάδοσης διαιρείται με το 16 προτού τροφοδοτήσει τους καταχωρητές ολίσθησης εκπομπής και λήψης Rx/Tx. Διότι το σήμα ρολογιού που παράγεται από τη γεννήτρια είναι 16πλάσιο του επιθυμητού ρυθμού μετάδοσης. Έτσι, επιτυγχάνουμε δειγματοληψία με συχνότητα 16 φορές μεγαλύτερη από τον ρυθμό μετάδοσης. Επιπλέον το σήμα τροφοδοτεί το λογικό κύκλωμα ανάκτησης δεδομένων (Data Recovery Logic), το οποίο δειγματοληπτεί τα δεδομένα. Για κάθε bit λαμβάνονται 16 συνολικά δείγματα, από τα οποία ελέγχονται το όγδοο, έννατο και δέκατο δείγμα. Το λογικό επίπεδο δύο τουλάχιστον δειγμάτων (από τα παραπάνω τρία) θεωρείτε το πραγματικό λογικό επίπεδο του bit. Νετλα τη δειγματοληψία όλα τα bits του λαμβανόμενου δεδομένου ολισθαίνουν στον καταχωρητή ολίσθησης λήψης. Από την άλλη πλευρά ο καταχωρητής ολίσθησης εκπομπής δεν απαιτεί δειγματοληψία και συνδέεται απευθείας με το σήμα χρονισμού μετάδοσης.

1.4.1.2 πομπός UART

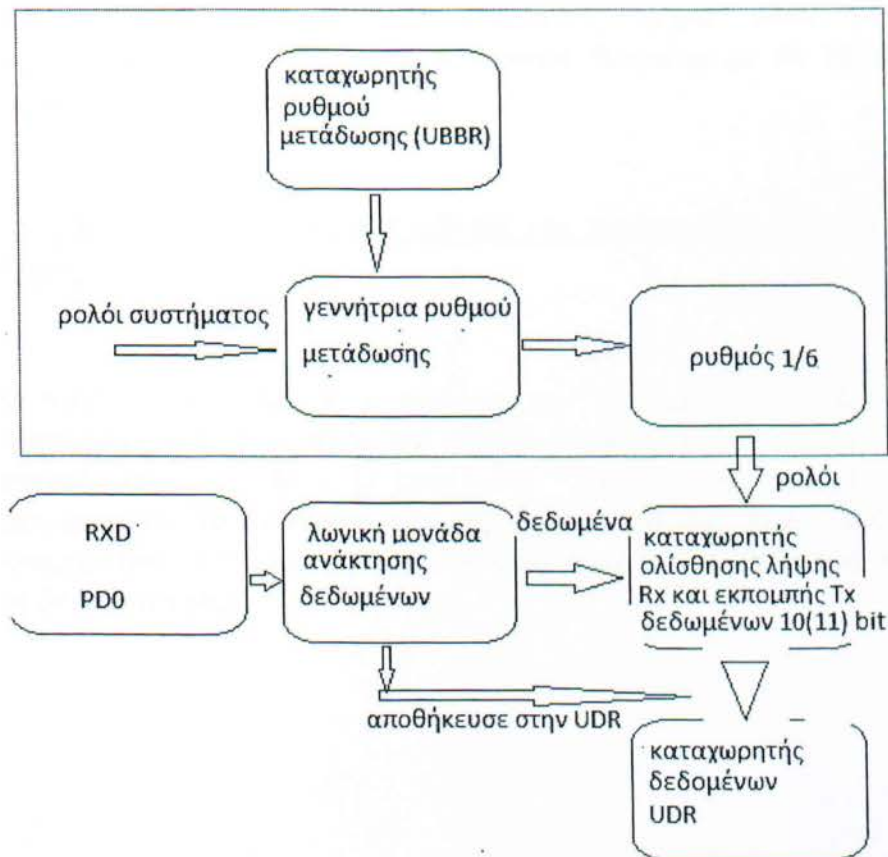
Ο πομπός UART στέλνει δεδομένα από τον μικροελεγκτή σε διάφορες συσκευές (data logger, pc) στον επιθυμητό ρυθμό μετάδοσης. Η εκπομπή αρχικοποιείται εγγράφοντας δεδομένα στον καταχωρητή UDR. Έπειτα, τα δεδομένα μεταφέρονται στον καταχωρητή ολίσθησης εκπομπής TX εφόσον το προηγούμενο byte έχει ολισθήσει ολόκληρα. Όταν ένα byte μεταφέρεται στον καταχωρητή ολίσθησης TX, η σημαία UDRE τίθεται. Η ρουτίνα εξυπηρέτησης διακοπής UDRE ISR μπορεί να γράψει το επόμενο byte στον καταχωρητή UDR χωρίς να αλλοιώσει την παρούσα διαδικασία εκπομπής.



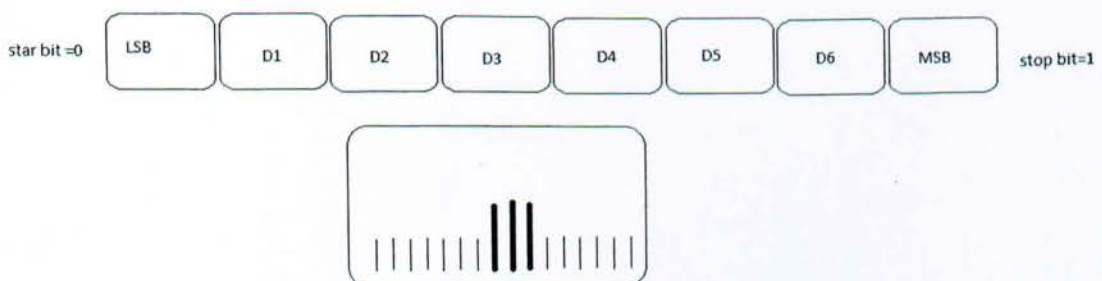
Όταν ένα byte έχει ολισθήσει πλήρως και δεν έχουν γραφτεί στον καταχωρητή UDR από την ρουτίνα εξυπηρέτησης διακοπής UDRE ISR, τότε η σημαία TXC τίθεται.

1.4.1.3 ο δέκτης UART

Ο δέκτης UART έχει παρόμοια δομή με τον πομπό, αλλά επιπλέον διαθέτει τα απαραίτητα κυκλώματα για λήψη δεδομένων: το λογικό κύκλωμα ανάκτησης δεδομένων και μια διακοπή για την ολοκλήρωση της λήψης δεδομένου. Χρησιμοποιεί τις ίδιες ρυθμίσεις για το baudrate όπως ο πομπός.



Το δεδομένο δειγματοληπτείται με τον τρόπο που περιγράψαμε στην ενότητα της γεννήτριας ρυθμού μετάδοσης και φαίνεται στο σχήμα.



από τα 16 το 8,9,19
δείγμα

Διακρίνουμε τις μικρές κάθετες γραμμές που αντιστοιχούν στους παλμούς ρολογιού που παράγει η γεννήτρια ρυθμού μετάδοσης. Οι τρεις ψηλότερες αντιστοιχούν στο 8,9,10

δείγμα. Συνεπώς, ο ρυθμός μετάδοσης αρχικά είναι 16πλάσιος ώστε να δειγματοληπτήσει τα δεδομένα, και έπειτα διαιρείται με το 16 για να ολισθήσει τα δεδομένα.

1.4.1.4 Ο καταχωρητής δεδομένων της ασύγχρονης σειριακής μονάδας UART (UDR)

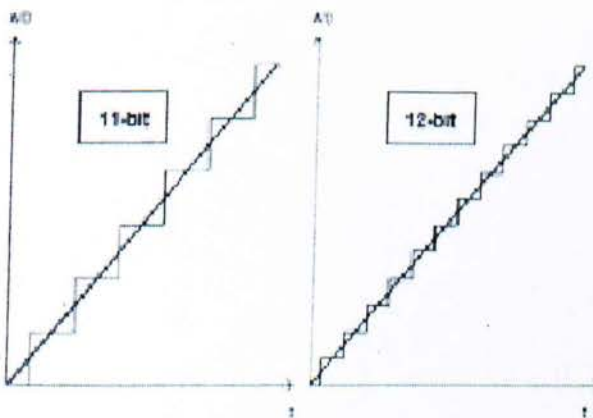
Ο πομπός και ο δέκτης μοιράζονται τον ίδιο καταχωρητή δεδομένων UDR. Στην πραγματικότητα όμως πρόκειται για δύο διαφορετικούς καταχωρητές που έχουν κοινή φυσική διεύθυνση \$0C. Ο ένας είναι καταχωρητής δεδομένων εκπομπής όπου μεταφέρονται τα δεδομένα που γράφουμε στη διεύθυνση \$0C. Ο άλλος είναι ο καταχωρητής δεδομένων λήψης όπου μεταφέρονται τα δεδομένα που διαβάζουμε από τη διεύθυνση \$0C.

1.4.2 Η αναλογική θύρα

1.4.2.1 Η αναλογική μετατροπή

Βασικό περιφερειακό των μικροελεγκτών είναι ο μετατροπέας από αναλογική θύρα σε ψηφιακή (A/D CONVERTER) μορφή για την εισαγωγή, έλεγχο επεξεργασία αναλογικών ποσοτήτων από τον εξωτερικό φυσικό κόσμο όπως θερμοκρασία, πίεση βάρους κλπ. Γενικά επιτρέπει τη μετατροπή σε ψηφιακή μορφή διαφορών δυναμικού που παράγονται από τους αισθητήρες και χρησιμεύει για την μετατροπή του ήχου σε ψηφιακή μορφή, και την ένδειξη επιπέδου τάσης μπαταρίας κλπ. Η λειτουργία ενός μετατροπέα από αναλογική σε ψηφιακή μορφή είναι η εξής:

- Θεωρούμε αναλογική τάση εισόδου V_{in} και τάση αναφοράς V_{ref} βάση της οποίας γίνεται η μετατροπή.
- Επίσης θεωρούμε ότι για να γίνει σωστά η μετατροπή πρέπει $V_{in} < V_{ref}$
- Για ADC ακριβείας χωρίζεται η δυναμική περιοχή (0- V_{ref}) σε 2^n ίσα διαστήματα και αντιστοιχείται το καθένα σε ένα δυαδικό αριθμό των n-bit ανάλογα με την τάξη του. Το κάθε διάστημα Q που αντιπροσωπεύει την ανάλυση του ADC είναι $Q = V_{REF} / 2^n$. αν υποθέσουμε ότι $V_{ref} = 2.56V$ και $n=10$ τότε $Q=2,5\text{ mV}$
- Γενικά η ψηφιακή έξοδος X του ADC είναι $X = \lfloor V_{in} \cdot 2^n / V_{ref} \rfloor = V_{in} / Q$ όπου με το ' δηλώνεται το στρογγυλοποιημένο ακέραιο μέρος



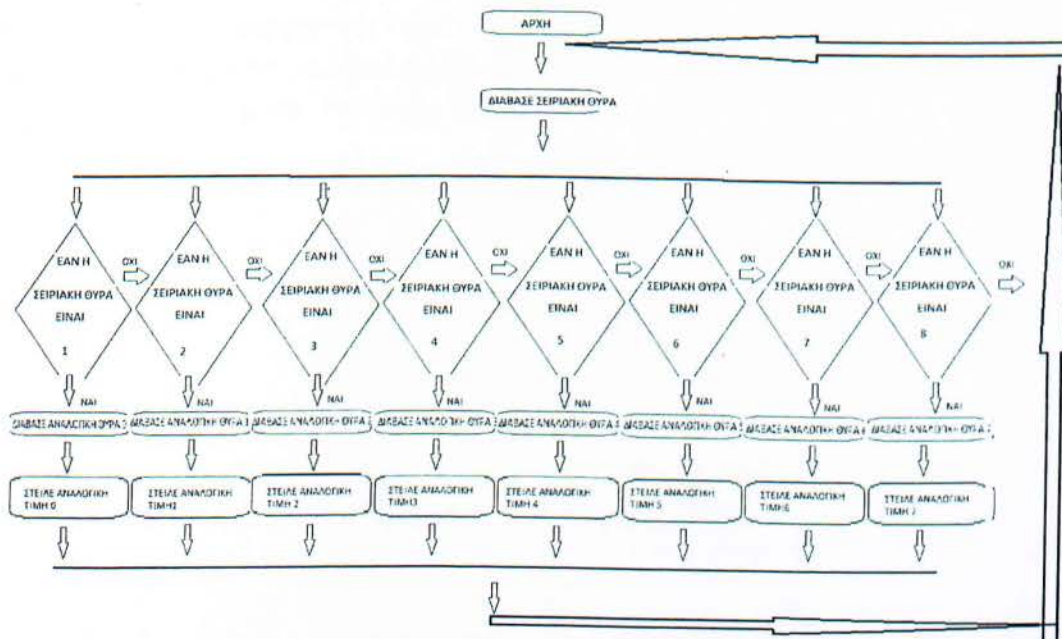
1.4.2.2 Η αναλογική θύρα του Atmega16

Ο μικροελεγκτής AVR MEGA16 διαθέτει ένα μετατροπέα από αναλογική σε ψηφιακή μορφή ακριβείας 10 bit που βασίζεται στη μέθοδο των διαδοχικών προσεγγίσεων. Ο ADC είναι συνδεδεμένος με ένα πολυπλέκτη 8 εισόδων που επιτρέπει την απλή μετατροπή των τάσεων από τα pins της θύρας PORT A.

Επιτρέπει επίσης και μετατροπή από διαφορετικές εισόδους (16 συνδυασμοί των 8 εισόδων) που συνδυάζονται και με ένα στάδιο Προγραμματιζόμενης (χ1,χ10,χ200) ενίσχυσης.

1.5 Λογική ανάλυση του κώδικα του ATmega16 και της visual basic και πως συνδυάστηκαν

1.5.1 Το Μπλοκ διάγραμμα του κώδικα του ATmega16 και η ανάλυσή του

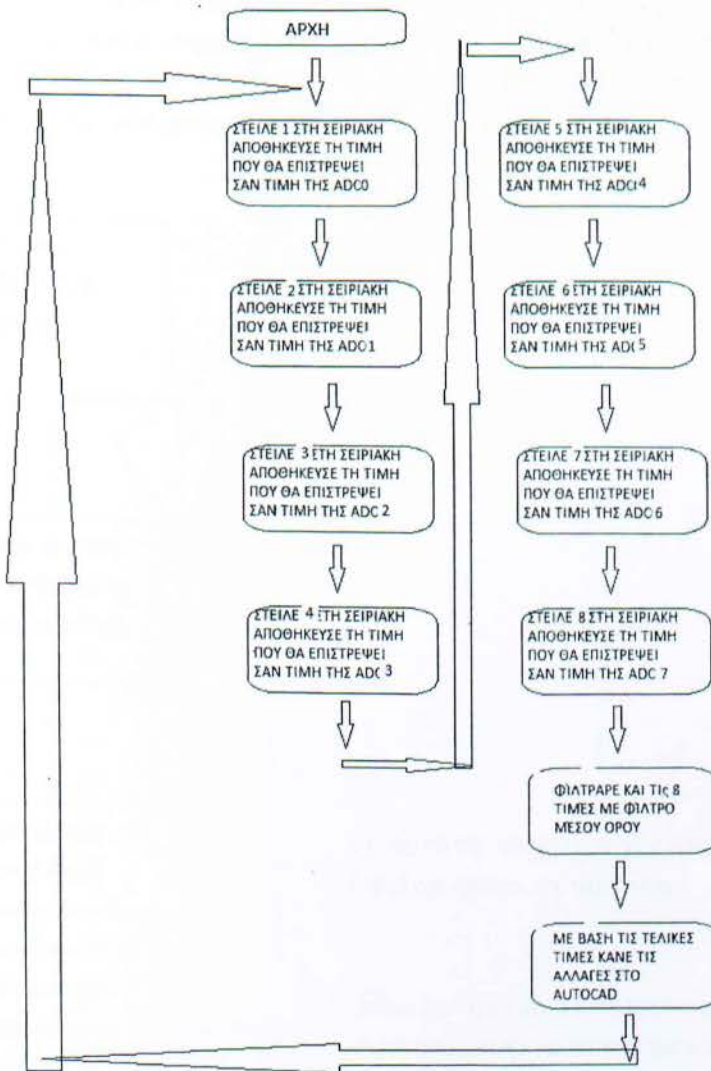


Στο κώδικα του ATmega16 στέλνετε από τη σειριακή θύρα μία τιμή σε μορφή χαρακτήρα. Εάν αυτή η τιμή είναι 1(σε χαρακτήρα) τότε διαβάσετε η αναλογική θύρα της πόρτας A του ακροδέκτη 0 και επιστρέψετε πάλι από τη σειριακή έπειτα ξαναδιαβάζεται η τιμή της σειριακής θύρας (αφού πρώτα έχει ξανάρθει η τιμή σε αυτήν) και αν δεν είναι 1 ελέγχετε εάν είναι 2. Εάν είναι 2 τότε διαβάζεται η αναλογική είσοδος της πόρτας A του ακροδέκτη 1 και η τιμή επιστρέφεται στη σειριακή θύρα. Αυτό γίνεται συνέχεια για τις σειριακές εισόδους από 1 έως 8. Εάν η τιμή της σειριακής τιμής που εισέλθει στη σειριακή θύρα δεν είναι από 1 έως 8 τότε δε διαβάσετε καμία αναλογική είσοδος της θύρας A και το πρόγραμμα περιμένει να ξαναεισέλθει νέα τιμή στη σειριακή θύρα.

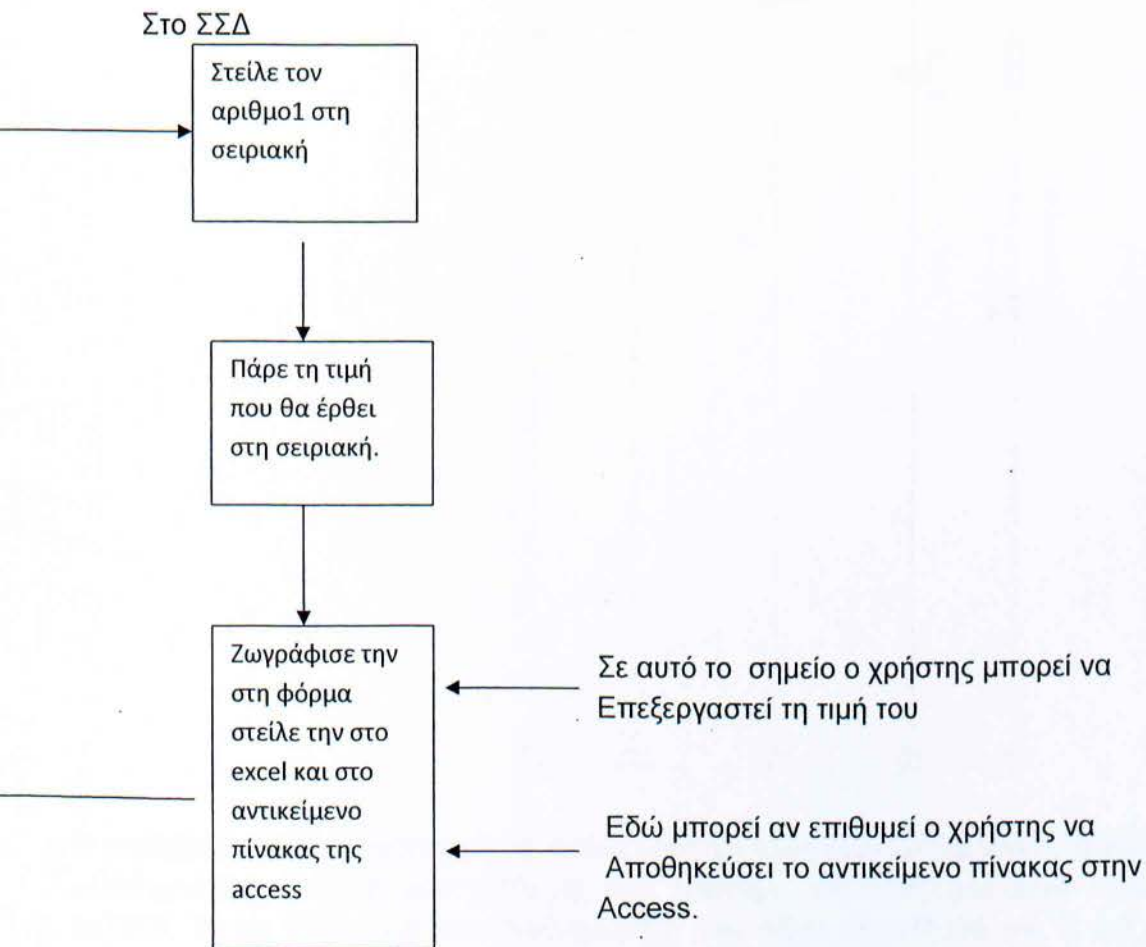
1.5.2 Το μπλοκ διάγραμμα του κώδικα της visual basic

Με βάση τον κώδικα που βλέπουμε στον ATmega16 περιμένουμε πως η visual basic στέλνει σειριακά τις τιμές 1 έως 8 σε χαρακτήρες, επιστρέφονται σε αυτήν οι αναλογικές τιμές του ATmega16 και έπειτα αναλύονται. Έτσι γίνεται!

Επειδή η visual basic έχει πολλά σημεία κώδικα εδώ θα αναλυθεί μόνο το μέρος του κώδικα που επικοινωνεί με τον ATmega16 σειριακά.

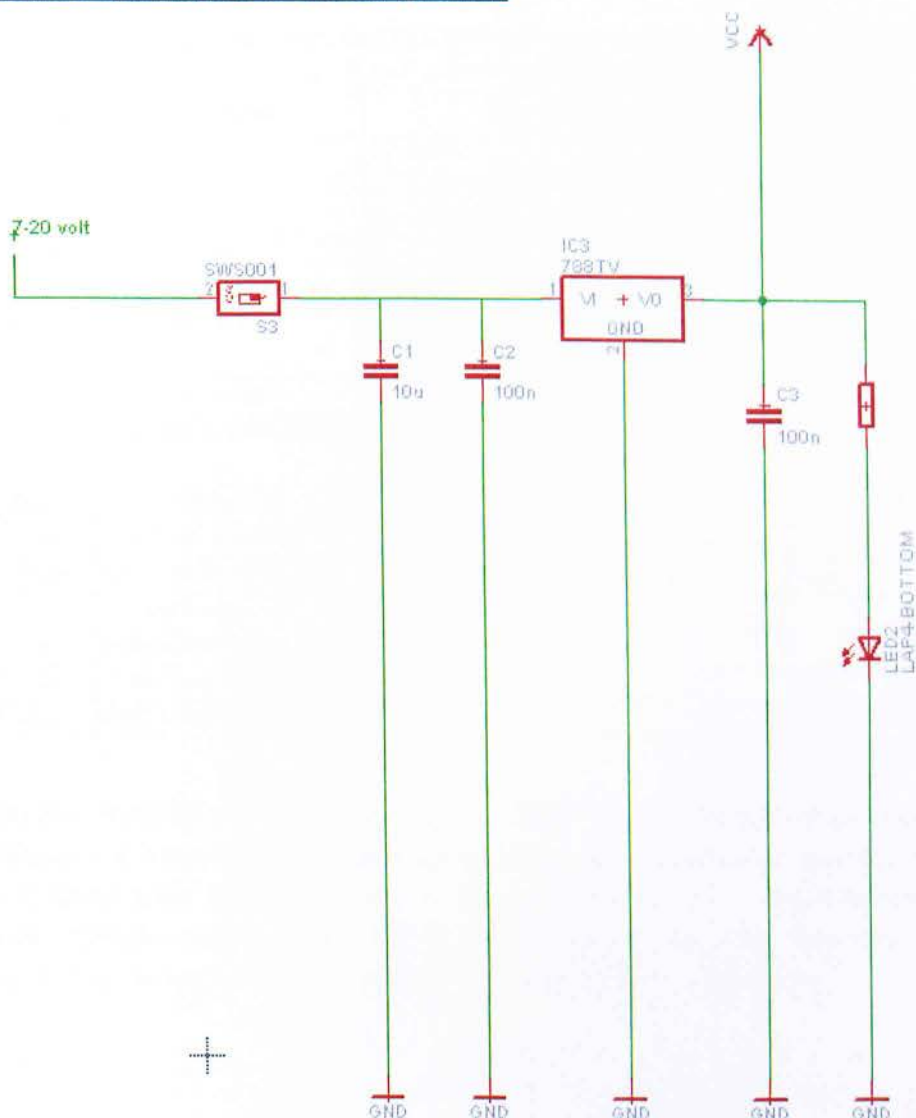


Όπως βλέπουμε και στο μπλοκ διάγραμμα του κώδικα της visual basic ο υπολογιστής στέλνει σειριακά τις τιμές από 1 έως 8 και παράλληλα παίρνει τις 8 αναλογικές τιμές από τα ποτενσιόμετρα που είναι συνδεδεμένα στους ακροδέκτες της θύρας A του μικροελεγκτή. Τις αποθηκεύει και αφού τις μετατρέψει σε αριθμητικές από χαρακτήρες μπορεί να τις επεξεργαστεί. Το πρόγραμμα αυτές τις τιμές τις εμφανίζει σε text boxes τις φιλτράρει με φίλτρο μέσου όρου και εμφανίζει τις φιλτραρισμένες τιμές σε text boxes και κάνει rotate στις κλειδώσεις του χεριού του autocad. Ο νέος κύκλος ενεργοποιείται μόλις εκπνεύσει ο χρονιστής της visual basic και ξεκινά πάλι η ανάγνωση του κώδικα από τον υπολογιστή. Αυτή είναι η "συνομιλία" του μικροελεγκτή με τον υπολογιστή. Ο υπολογιστής "ρωτάει" και ο μικροελεγκτής "απαντάει". Ουσιαστικά ο υπολογιστής ζητάει από τον μικροελεγκτή να του "πει" τη τιμή των αναλογικών το θυρών και ο μικροελεγκτής είναι προγραμματισμένος να απαντήσει.



2.1 Ανάλυση του κυκλώματος της πλακέτας

2.1.1 το κύκλωμα του σταθεροποιητή



Ο σταθεροποιητής χρειάζεται για να στείνει σταθερή τάση στον μικροελεγκτή 5 volt. Το ολοκληρωμένο που χρησιμοποιήθηκε στο κύκλωμα του σταθεροποιητή είναι το lm7805. Το lm 7805 είναι ένα ολοκληρωμένο που δίνει στην έξοδό του 5 volt ενώ δίνει μέχρι και 1.5 a οπότε είναι ότι πρέπει για την τροφοδοσία του μικροελεγκτή αφού θέλει 5 volt τάση και 1.1ma. το κύκλωμα τροφοδοσίας του μικροελεγκτή τροφοδοτεί και τα περιφερειακά κυκλώματά του, τα led τις αντιστάσεις και το max232 με το κύκλωμά

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

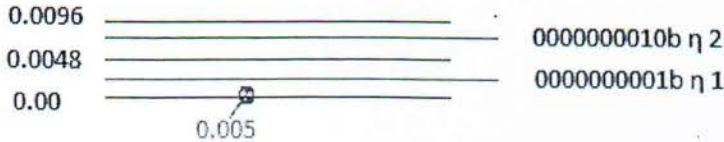
του χωρίς κανένα πρόβλημα. Το κύκλωμα του lm7805 αντέχει ως είσοδο μέχρι και 30 volt είσοδο για να δουλέψει σωστά στη πτυχιακή αυτή χρησιμοποίησα τροφοδοτικό στα 22 volt και με μέγιστο ρεύμα 1,25a.

Table 4: Electrical Characteristics Of L7805 (refer to the test circuits, $T_J = -55$ to 150°C , $V_I = 10\text{V}$, $I_O = 500$ mA, $C_I = 0.33$ μF , $C_O = 0.1$ μF unless otherwise specified).

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_O	Output Voltage	$T_J = 25^\circ\text{C}$	4.8	5	5.2	V
V_O	Output Voltage	$I_O = 5$ mA to 1 A $P_O \leq 15\text{W}$ $V_I = 8$ to 20 V	4.65	5	5.35	V
$\Delta V_O(^*)$	Line Regulation	$V_I = 7$ to 25 V $T_J = 25^\circ\text{C}$		3	50	mV
		$V_I = 8$ to 12 V $T_J = 25^\circ\text{C}$		1	25	
$\Delta V_O(^*)$	Load Regulation	$I_O = 5$ mA to 1.5 A $T_J = 25^\circ\text{C}$			100	mV
		$I_O = 250$ to 750 mA $T_J = 25^\circ\text{C}$			25	
I_d	Quiescent Current	$T_J = 25^\circ\text{C}$			6	mA
ΔI_d	Quiescent Current Change	$I_O = 5$ mA to 1 A			0.5	mA
		$V_I = 8$ to 25 V			0.8	
$\Delta V_O/\Delta T$	Output Voltage Drift	$I_O = 5$ mA		0.6		mV/ $^\circ\text{C}$
eN	Output Noise Voltage	B = 10Hz to 100KHz $T_J = 25^\circ\text{C}$			40	$\mu\text{V}/V_O$
SVR	Supply Voltage Rejection	$V_I = 8$ to 18 V $f = 120\text{Hz}$	68			dB
V_d	Dropout Voltage	$I_O = 1$ A $T_J = 25^\circ\text{C}$		2	2.5	V
R_O	Output Resistance	$f = 1$ KHz		17		m Ω
I_{sc}	Short Circuit Current	$V_I = 35$ V $T_J = 25^\circ\text{C}$		0.75	1.2	A
I_{scp}	Short Circuit Peak Current	$T_J = 25^\circ\text{C}$	1.3	2.2	3.3	A

Με βάση τον παραπάνω πίνακα από το datasheet του lm7805 βλέπουμε πως η έξοδος του κυκλώματος τροφοδοσίας του κυκλώματος του μικροελεγκτή και της υπόλοιπης πλακέτας δίνει τάση από 4.65 έως 5.35 volt. Οι πυκνωτές του κυκλώματος είναι πυκνωτές εξομάλυνσης γι αυτό στο κύκλωμα πρέπει να είναι όσο πιο κοντά στο ολοκληρωμένο για να υπάρχει όσο το δυνατόν μικρότερος θόρυβος.

μικροελεγκτή. Οι ακροδέκτες 14 και 15(TX,RX) είναι συνδεδεμένοι με το max232 για την σειριακή επικοινωνία του μικροελεγκτή με τον υπολογιστή. Οι 8 αναλογικές εισόδους του μικροελεγκτή έχουν συνδεθεί με 8 κυκλώματα διέρευσης τάσης. Ο μικροελεγκτής μετράει τις αναλογικές αυτές εισόδους, οι αντιστάσεις που έχουν χροισιμοποιηθεί για τους διαιρέτες τάσης είναι του 10KΩ ενώ τα ποτενσιόμετρα είναι γραμμικά 0-1KΩ.. Ο τύπος του διαιρέτη τάσης είναι $V_{out}=V_{in}(\frac{R2}{R1+R2})$ για τάση εισόδου χροισιμοποιήθηκαν 5 volt και για τη τιμή της R1 10KΩ οπότε έχουμε $V_{out}=5(\frac{R2}{10000+R2})$ για R2=0 , η τάση στην έξοδο του κυκλώματος είναι $V_{out}=\frac{5}{10000}=0.0005v$. Τα 0.0005 volt ο μικροελεγκτής τα βλέπει εώς εξής, $Q=\frac{V_{ref}}{2^n}$ και για $V_{ref}=5v$ και για $n=10$ bits έχουμε $Q=\frac{5}{1024}=0.0048$ αυτό είναι το βήμα της ADC. Η τιμή που θα διαβάσει ο μικροελεγκτής για 0,0005 volt είναι $\chi=\frac{V_{in}}{Q}$ και εδώ για τάση εισόδου έχουμε $V_{in}= 0.0005$ volt και για βήμα μέτρησης της αναλογικής εισόδου $Q=0.0048$ volt. Άρα η τιμή που διαβάζει ο μικροελεγκτής όταν το ποτενσιόμετρο είναι στη μικρότερη δυνατή τιμή (δηλαδή στα 0 Ω) ο μικροελεγκτής διαβάζει τη τιμή $\chi=\frac{0.0005}{0.0048}= 0.1$. Τη τιμή 0.1 ο μικροελεγκτής τη διαβάζει σαν 1 επειδή η τιμή της αναλογικής εισόδου είναι στο πρώτο βήμα της αναλογικής εισόδου.

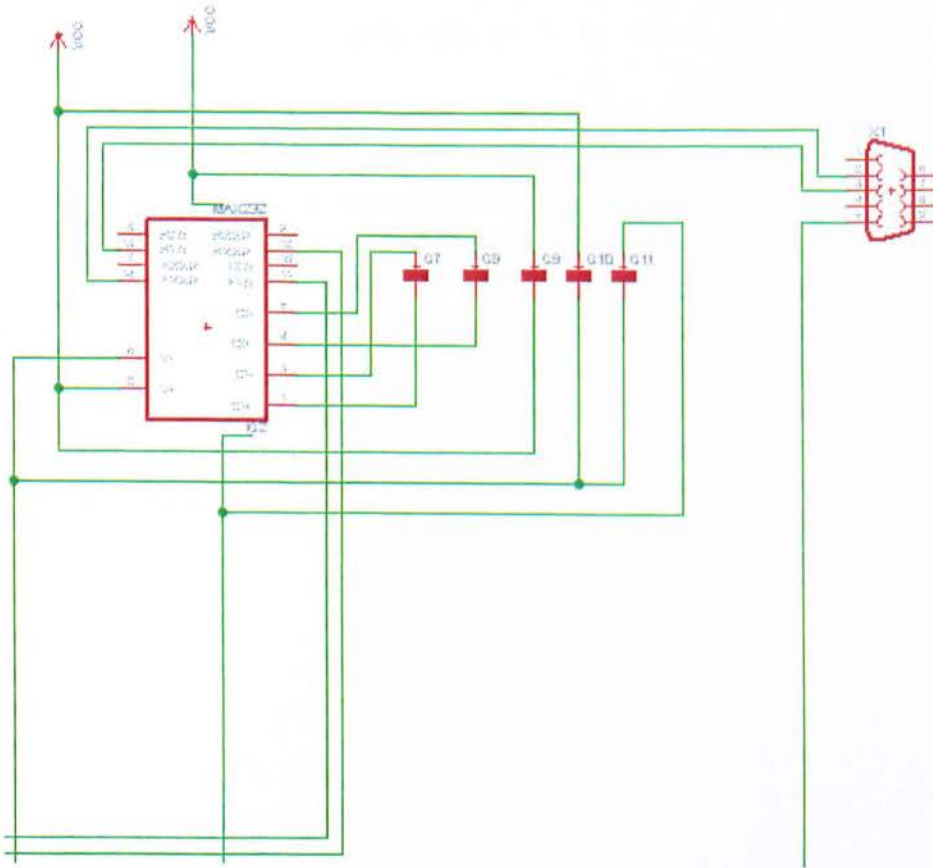


Όταν τώρα το ποτένσιόμετρο έχει τη τιμή 1KΩ η έξοδος στον διαιρέτη τάσης είναι

είναι $V_{out}=V_{in}(\frac{R2}{R1+R2})= 5(\frac{1000}{10000+1000})=0.45v$ αυτό στον μικροελεγκτή μεταφράζεται ως εξής,

$\chi=\frac{V_{in}}{Q}=\frac{0.45}{0.0048}=94.7$ όπου αυτό στην αναλογική τιμή της αναλογικής θύρας μεταφράζεται σε 95. Αυτό σημαίνει πως οι τιμές των ποτενσιομέτρων διαβάζονται από την αναλογική θύρα του atmega16 από 0 εως 95. Αυτό βολεύει στη συγκεκριμένη πτυχιακή επειδή η κίνηση των δακτύλων ή του καρπού ενός πραγματικού χεριού είναι σχεδόν κινήσεις μέχρι 90 μοίρες. Όπως θα δούμε στη συνέχεια αυτές οι τιμές αλλάζουν κατά πολύ στο πρόγραμμα της visual basic αλλά το εύρος τους είναι από 0 εώς 90 μοίρες.

2.1.3 Το κύκλωμα του MAX232



Όπως αναφέρθηκε και στην ενότητα ανάλυσης των ολοκληρωμένων που χρησιμοποιήθηκαν για την υλοποίηση της πτυχιακής το max232 είναι ένας TTL μετατροπέας ο οποίος μετατρέπει τη παλμοσειρά της εξόδου του υπολογιστή που είναι +15volt για λογικό 1 και -15volt για λογικό 0 σε +5volt για λογικό 1 και 0volt για λογικό 0 και αντίθετα τη παλμοσειρά του atmega16 από 0,+5 volt σε +15,-15 volt για να μπορούν να επικοινωνούν ο atmega16 και ο υπολογιστής χωρίς κανένα πρόβλημα. Αυτό όμως δημιουργεί ένα μικρό πρόβλημα στην επικοινωνία επειδή όταν στέλνει σήμα ο υπολογιστής στον atmega16 μικραίνει πολύ το αναλογικό εύρος των τιμών της παλμοσειράς. Δηλαδή όταν ο υπολογιστής στέλνει σειριακά τα δεδομένα η μέγιστη απόσταση του λογικού 0 από το λογικό 1 στο εύρος της παλμοσειράς είναι από -15 εως 15 volt δηλαδή 30 volt και πρέπει αυτή η απόσταση να γίνει 5 volt(από 30 που ήτανε). Αυτό το πρόβλημα βέβαια το λύνει το max 232 και οι λογικές πύλες που χρησιμοποιεί αλλά και οι 5 πυκνωτές εξομάλυνσης του κυκλώματος που προτείνει ο κατασκευαστής(MAXIM). Για περισσότερη ασφάλεια βέβαια τοποθετούντε οι πυκνωτές όσο πιο κοντά

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

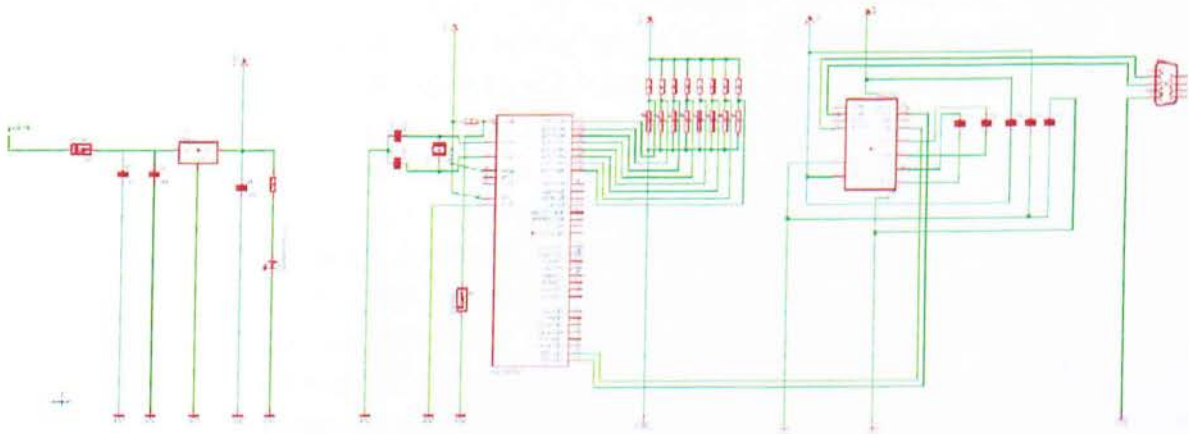
Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

γίνεται στο ολοκληρωμένο και βέβαια το ολοκληρωμένο όσο πιο κοντά γίνεται στον μικροελεγκτή(στην ίδια πλακέτα). Ενώ η απόσταση του max232 από τον υπολογιστή παίζει δευτερεύοντα ρόλο αφού το εύρος είναι των τιμών της παλμοσειράς του υπολογιστή είναι 6 φορές μεγαλύτερο από το εύρος των τιμών της σειριακής παλμοσειράς του atmega16.

Το max232 υποστηρίζει δύο σειριακές παλμοσειρές αλλά σε αυτή αυτή την εργασία χρησιμοποιήθηκε μόνο η μία επειδή υπήρχε μόνο ένας μικροελεγκτής και ένας υπολογιστής. Όπως φαίνετε και στο σχήμα οι ακροδέκτες 11 και 12 του max232 είναι οι ακροδέκτες που συνδέονται στον υπολογιστή. Ο ακροδέκτης 12 του max232 στέλνει σειριακά δεδομένα στον ακροδέκτη RX του atmega16 αφού πρώτα τα έχει λάβει από τον υπολογιστή με τον ακροδέκτη 13 που συνδέεται στον ακροδέκτη TX της σειριακής θύρας του υπολογιστή. Αντίθετα από τον ακροδέκτη 11 που είναι συνδεδεμένος στον ακροδέκτη TX του atmega16 λαμβάνει τα δεδομένα από τον atmega και από τον ακροδέκτη 12 τα στέλνει στον ακροδέκτη 14 που είναι συνδεδεμένος με τον ακροδέκτη RX της σειριακής θύρας του υπολογιστή.

2.1.4 Το κύκλωμα όλης της πλακέτας και πως φτιάχτηκε



Εδώ φαίνεται το πλήρες κύκλωμα της πλακέτας που δημιουργήθηκε για αυτή την εργασία. Δε θα αναφερθώ στο τι κάνει το κύκλωμα επειδή αναφέρθηκα στις παραπάνω ενότητες. Σε αυτή την ενότητα θα παρουσιάσω το τρόπο κατασκευής της πλακέτας.

Η πλακέτα κατασκευάστηκε με τη μέθοδο PCB. Η πλακέτα είναι φωτοευαίσθητη δύο όψεων. Οι φωτοευαίσθητες πλακέτες αποτελούνται από πολλές στρώσεις ή επίπεδα. Τα επίπεδα αυτά είναι το μονωτικό υλικό, ο χαλκός ή η φωτοευαίσθητη επιφάνεια και η προστατευτική μεμβράνη.



Για να σχηματιστεί το κύκλωμα στον χαλκό πρέπει αρχικά να σχεδιαστεί το κύκλωμα σε μία διαφάνεια με τη βοήθεια ενός σχεδιαστικού προγράμματος. Για αυτή την πτυχιακή χρησιμοποιήθηκε το σχεδιαστικό πρόγραμμα eagle με τη βοήθεια του eagle τυπώθηκαν δύο διαφάνειες που περιείχαν και οι δύο διαφάνειες ολόκληρο το κύκλωμα

της πλακέτας. Δηλαδή στη μία διαφάνεια ήταν το μισό κύκλωμα και στην άλλη διαφάνεια το άλλο μισό κύκλωμα.

Αφαιρέθηκαν οι προστατευτικές μεμβράνες και οι διαφάνειες τοποθετήθηκαν πάνω στα φωτοευαίσθητα υλικά της πλακέτας με τέτοιο τρόπο ώστε να εφάπτονται καλά και ακριβώς πάνω στην επιφάνεια που θα γίνει το κύκλωμα στη πλακέτα. Οι πλακέτες και οι διαφάνειες τοποθετήθηκαν σε ένα θάλαμο με μια λάμπα υπεριώδους ακτινοβολίας για 20 λεπτά η κάθε πλευρά. Έτσι το φωτοευαίσθητο υλικό κάηκε στα σημεία που δεν θα υπάρχουν διάδρομοι ή οι ακροδέκτες των ηλεκτρονικών εξαρτημάτων. Μετά η πλακέτα μπήκε σε ένα δοχείο που περιείχε καυστική σόδα η οποία άλλαξε τη σύσταση του φωτοευαίσθητου υλικού που έχει μείνει στη πλακέτα. Η πλακέτα έμεινε στο δοχείο αυτό για περίπου 5 λεπτά. Μετά τοποθετήθηκε σε ένα άλλο δοχείο που περιείχε το αποχαλκοτικό υγρό. Το αποχαλκοτικό υγρό αφαίρεσε από τη πλακέτα τον χαλκό εκτός από τα σημεία που είχαν σχεδιαστεί στις διαφάνειες. Η πλακέτα παρέμεινε στο δοχείο με το αποχαλκοτικό περίπου 15-25 λεπτά μέχρι να φύγει τελείως ο χαλκός σε ζεστό αλλά ΟΧΙ καυτό μείγμα. Τέλος η πλακέτα τρυπήθηκε με τρυπάνια του 1 χιλιοστού και 0.7 χιλιοστά και τοποθετήθηκαν τα εξαρτήματα τα οποία κόλλησαν στη πλακέτα με κολλητήρι και καλάι. Η όλη διαδικασία βοήθησε πολύ πρακτικά την εργασία αυτή αφού η πλακέτα δουλεύει άψογα σε βάθος χρόνου. Επίσης η πλακέτα επειδή είναι διπλής όψης δεν υπήρχε πρόβλημα σύγκρουσης των διαδρόμων της αφού διασταυρώνονται σε διαφορετικές όψεις.

ΚΕΦΑΛΑΙΟ 3ο

3.1 Η ανάλυση του κώδικα του AT mega 16

Σε αυτό το κεφάλαιο θα αναλυθεί ο κώδικας του AT mega 16 για να γίνει αυτό όσο πιο κατανοητά γίνεται μαζί με τον κώδικα που θα είναι με μαύρο χρώμα θα χρησιμοποιηθούν και παρατηρήσεις με κόκκινο χρώμα για την εξήγηση του.

3.2 Ο κώδικας του AT mega 16

```
#include <avr/io.h>
#include <avr/interrupt.h>
#define USART_BAUDRATE 9600
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)
```

Εδώ βλέπουμε τις βιβλιοθήκες που χρησιμοποιήθηκαν στον κώδικα. Οι βιβλιοθήκες είναι βιβλιοθήκες για τις διακοπές για τις εισόδους/ εξόδους του AT mega 16 καθώς και βιβλιοθήκες που δηλώνεται ο ρυθμός των ακμών που θα στέλνει η σειριακή θύρα στην έξοδό της. Με τις δύο τελευταίες εντολές γλιτώνουμε αρκετές σειρές κώδικα

```
//adc~~~~~
~~~~~
void InitADC()
{

ADMUX=(1<<REFS0);
ADCSRA=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1); //Rrescalar div factor =64
}
```

Η συνάρτηση αυτή αρχικοποιεί την αναλογική συμπεριφορά του μικροελεγκτή.

Με τη πρώτη εντολή ο AT mega16 παίρνει τη τάση αναφοράς από τον ακροδέκτη Vref που στο κύκλωμα αυτό είναι 5volt. Όπως βλέπουμε στη δεύτερη εντολή χρησιμοποιείται Prescaler 64, δηλαδή ο κάθε κύκλος μηχανής που χρησιμοποιεί ο AT mega16 για την ανάγνωση των αναλογικών τιμών είναι το εξωτερικό ρολόι 8MHZ /64. Έτσι ο μικροελεγκτής μπορεί και κάνει αναλογική ανάγνωση στους ακροδέκτες P0-7 σύμφωνα με τις οδηγίες του κατασκευαστή(ATMEL)

```
uint16_t ReadADC(uint8_t ch)
```

```
{
  Η εντολή αυτή ασφαρίζει πως η τιμή ch δηλαδή η τιμή που στέλνουμε για να
  καλέσουμε αυτή τη συνάρτηση θα είναι πάντα από 0 εως 7
  //Select ADC Channel ch must be 0-7
  ch=ch&0b00000111;
  Εδώ καθαρίζονται τα 3 τελευταία bits του καταχωρητή ADMUX για να εισέλθει
  μετά το καινούριο.
  // clear the old channel
  ADMUX &= 0b11111000;
  Εδώ ορίζεται το νέο κανάλι. Η παραπάνω εντολή είναι απαραίτητη επειδή το νέο
  κανάλι ορίζεται με λογικό OR και εάν δεν υπήρχε ο AT mega 16 δε θα διάβαζε το
  κανάλι που θα του στέλναμε εμείς (ch)
  ADMUX|=ch;
  Με αυτή την εντολή ξεκινά η αναλογική μετατροπή
  //Start Single conversion
  ADCSRA|=(1<<ADSC);

  //Wait for conversion to complete
  while(!(ADCSRA & (1<<ADIF)));

  ADCSRA|=(1<<ADIF);

  return(ADC);
  οι τρεις παραπάνω εντολές ολοκληρώνουν την αναλογική ανάγνωση και
  επιστρέφουν τη τιμή της
}
  Η παραπάνω συνάρτηση μας επιστρέφει την τιμή της αναλογικής θύρας. Πιο
  συγκεκριμένα όταν καλούμε την συνάρτηση αυτή της δίνουμε και μία τιμή από το
  0 εως το 7 έτσι επιλέγουμε πιο κανάλι θα διαβάσει( δηλαδή ποιον ακροδέκτη P0
  έως P7)
//-----MAIN
int main (void)
{
  //serialINSTAL
  UCSRB |= (1 << RXEN) | (1 << TXEN);
  UCSRC |= (1 << URSEL) | (1 << UCSZ0) | (1 << UCSZ1);

  UBRRL = BAUD_PRESCALE;
```

```
UBRRH = (BAUD_PRESCALE >> 8);
//ENDserialINSTAL
```

Το παραπάνω σεί εντολών αρχικοποιεί την σειριακή θύρα του AT mega16
 uint16_t adc_result;

```
char a;
```

Στις 2 παραπάνω εντολές δημιουργούμε δύο μεταβλητές η πρώτη είναι 16bit ενώ η δεύτερη που είναι χαρακτήρας είναι 8bits

```
InitADC();
```

Και αφού αρχικοποιήσουμε την αναλογική συμπεριφορά του AT mega 16

ξεκινάμε

```
begin:
```

εδώ περιμένουμε να έρθει τιμή στη σειριακή θύρα

```
while ((UCSRA & (1 << RXC)) == 0) {}; // Do nothing until read serial
```

όταν έρθει την περνάμε στην μεταβλητή

```
a=UDR;
```

```
if(a==((char) '1')){
```

εδώ ελέγχουμε εάν η σειριακή τιμή που πήραμε είναι 1 και αν είναι διαβάζουμε το πρώτο κανάλι P0 του μικροελεγκτή

```
adc_result=ReadADC(0);
```

```
;
```

Το στέλνουμε στη σειριακή θύρα

```
while ((UCSRA & (1 << UDRE)) == 0) {};
```

```
UDR=adc_result;
```

Και πάμε πάλι στην αρχή.

```
goto begin;
```

```
}
```

```
if(a==((char) '2')){
```

```
adc_result=ReadADC(1);
```

```
;
```

```
while ((UCSRA & (1 << UDRE)) == 0) {};
```

```
UDR=adc_result;
```

```
goto begin;
}
```

```
if(a==(char)'3'){
adc_result=ReadADC(2);
while ((UCSRA & (1 << UDRE)) == 0) {};
UDR=adc_result;
goto begin;
}
```

```
if(a==(char)'4'){
adc_result=ReadADC(3);
while ((UCSRA & (1 << UDRE)) == 0) {};
UDR=adc_result;
goto begin;
}
```

```
if(a==(char)'5'){
adc_result=ReadADC(4);
;
while ((UCSRA & (1 << UDRE)) == 0) {};
UDR=adc_result;
goto begin;
}
```

```
if(a==(char)'6'){
adc_result=ReadADC(5);
while ((UCSRA & (1 << UDRE)) == 0) {};
UDR=adc_result;
goto begin;
}
```

```
if(a==(char)'7'){
adc_result=ReadADC(6);
while ((UCSRA & (1 << UDRE)) == 0) {};
UDR=adc_result;
goto begin;
}
```

```
if(a==(char)'8'){
```



```

adc_result=ReadADC(7);
while ((UCSRA & (1 << UDRE)) == 0) {};
UDR=adc_result;
goto begin;
}

goto begin;
}

```

Αυτό που συνέβη στο πρώτο σετ εντολών τις κύριες συνάρτησης συμβαίνει για όλα τα υπόλοιπα σετ εντολών της. Δηλαδή εάν έρθει κάποια τιμή στη σειριακή θύρα από 0 ως 7 τρέχει το κατάλληλο σετ εντολών με τη βοήθεια της if . Τέλος σε περίπτωση που η τιμή της σειριακής τιμής που θα εισέλθει στον AT mega 16 δεν είναι από 1 έως 8 τότε το πρόγραμμα δεν κάνει απολύτως τίποτα και επιστρέφει πάλι στην αρχή.

4.1) Η ανάλυση του κώδικα της visual basic για το AutoCad

Η ανάλυση του κώδικα της visual basic θα γίνει με τον ίδιο τρόπο που έγινε η ανάλυση στον AT mega 16 δηλαδή θα γίνουν σχόλια με κόκκινα γράμματα ανάμεσα στον κώδικα αφού πρώτα χωρίσουμε τον κώδικα σε κάποιες ενότητες.

4.2 Η ανάλυση του γραφικού περιβάλλοντος της visual basic

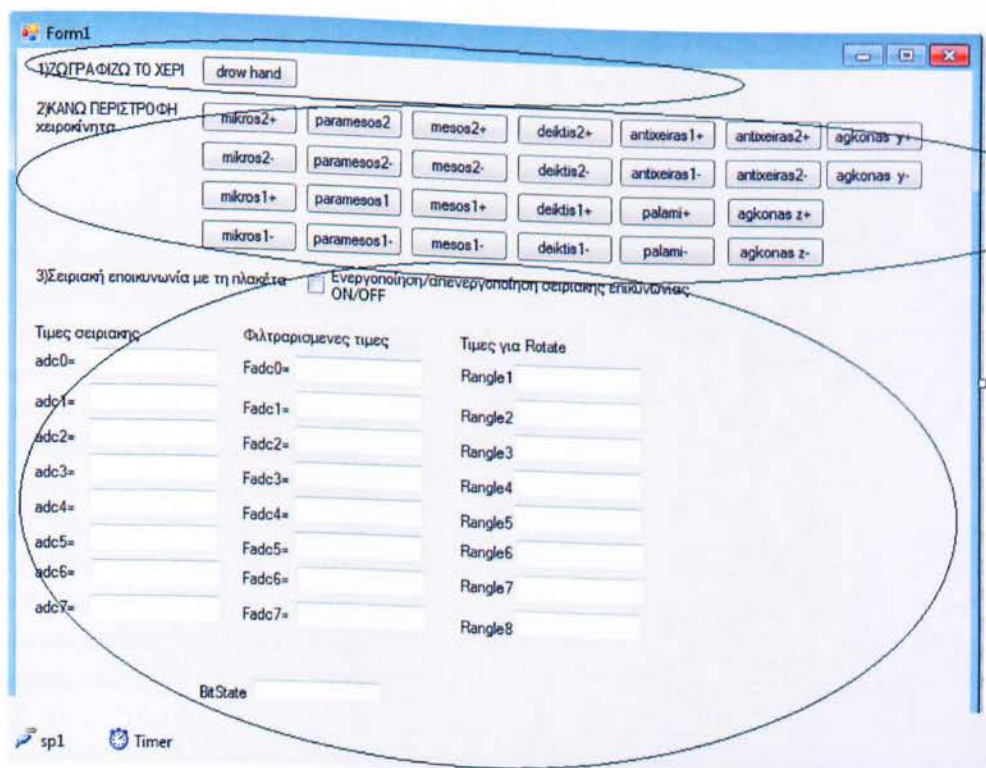
The screenshot shows a Windows-style window titled "Form1" with a blue title bar. The form contains the following elements:

- 1) ΖΩΓΡΑΦΙΣΤΩ ΤΟ ΧΕΡΙ**: A button labeled "drow hand".
- 2) ΚΑΝΩ ΠΕΡΙΣΤΡΟΦΗ χειροκίνητα**: A grid of 24 buttons arranged in 4 rows and 6 columns. The buttons are labeled as follows:

mikros2+	paramesos2	mesos2+	deiktis2+	antixeiras1+	antixeiras2+	agkonas y+
mikros2-	paramesos2-	mesos2-	deiktis2-	antixeiras1-	antixeiras2-	agkonas y-
mikros1+	paramesos1	mesos1+	deiktis1+	palami+		agkonas z+
mikros1-	paramesos1-	mesos1-	deiktis1-	palami-		agkonas z-
- 3) Σειριακή επικοινωνία με τη πλακέτα**: A checkbox labeled "Ενεργοποίηση/απενεργοποίηση σειριακής επικοινωνίας ON/OFF".
- Τιμές σειριακής**: A vertical list of 8 textboxes labeled "adc0=" through "adc7=".
- Φιλτραρισμένες τιμές**: A vertical list of 8 textboxes labeled "Fadc0=" through "Fadc7=".
- Τιμές για Rotate**: A vertical list of 8 textboxes labeled "Rangle1" through "Rangle8".
- Bit State**: A text box at the bottom center.
- Taskbar**: At the bottom left, there are icons for "sp1" and "Timer".

Η παραπάνω φόρμα όπως βλέπουμε αποτελείται από κουμπιά πολλές ετικέτες που εξηγούν τι μετράει το κάθε textbox επίσης βλέπουμε και ένα χρονικό και ένα αντικείμενο σειριακής επικοινωνίας. Επίσης μπορούμε να παρατηρήσουμε πως το περιβάλλον της φόρμας χωρίζεται σε τρία μέρη. Το πρώτο μέρος της φόρμας αποτελείται από μόλις ένα κουμπί το οποίο όμως σχεδιάζει το χέρι στο autocad. Στο δεύτερο μέρος της φόρμας υπάρχουν πολλά κουμπιά με τα οποία μπορούμε να κουνίσουμε το χέρι από τα κουμπιά αυτά. Όπως μπορούμε να καταλάβουμε το κάθε κουμπί επηρεάζει μία κλειδωση του χεριού προς μία κατεύθυνση. Το τρίτο μέρος της φόρμας είναι αυτό που ενεργοποιεί τη σειριακή επικοινωνία με την πλακέτα. Ενώ τα 24 textboxes απεικονίζουν τις τιμές που διαβάζει η σειριακή θύρα του υπολογιστή, τις τιμές αυτές μετά από φιλτράρισμα και τέλος τη μετατροπή των φιλτραρισμένων τιμών για να μπορούν να κάνουν περιστροφή σε λογικά πλαίσια της κάθε κλειδωσης του χεριού.

Οι τρεις ενότητες



4.3 Η ανάλυση της πρώτης ενότητας της φόρμας της visual basic

Η πρώτη ενότητα της φόρμας είναι ένα κουμπί το οποίο ζωγραφίζει το χέρι στο autocad.

Ο ΚΩΔΙΚΑΣ ΤΗΣ ΠΡΩΤΗΣ ΕΝΟΤΗΤΑΣ

Οι βιβλιοθήκες τύπων για τη σύνδεση στο autocad

`Imports Autodesk.AutoCAD.Interop`

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
 Πτυχιακή εργασία του Καπαρέλη Δημήτριου
 Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

Imports Autodesk.AutoCAD.Interop.Common

Επειδή το πρόγραμμα είναι πάρα πολύ μεγάλο οι σημειώσεις που κράτησα με βοήθησαν πάρα πολύ αφού πρώτα μελετούσα με πιο τρόπο θα κάνω αυτό που ήθελα να κάνω και έπειτα σκεφτόμουν ποιες εντολές θα χρησιμοποιήσω.

Public Class autocad

```
Dim thebratsoObj As Acad3DSolid
Dim thepalamiObj As Acad3DSolid
Dim theantixeiras1Obj As Acad3DSolid
Dim theantixeiras2Obj As Acad3DSolid
Dim thedeiktis1Obj As Acad3DSolid
Dim thedeiktis2Obj As Acad3DSolid
Dim themesos1Obj As Acad3DSolid
Dim themesos2Obj As Acad3DSolid
Dim theparamesos1Obj As Acad3DSolid
Dim theparamesos2Obj As Acad3DSolid
Dim themikros1Obj As Acad3DSolid
Dim themikros2Obj As Acad3DSolid
```

Εδώ έχω φτιάξει κάποιες βοηθητικές γραμμές οι οποίες με βοήθησαν στην περιστροφή (3D Rotate) των αντικειμένων

```
Dim lineYxeri As AcadLine
Dim lineZxeri As AcadLine
Dim lineYpalami As AcadLine
```

```
Dim lineYmikros1 As AcadLine
Dim lineYparamesos1 As AcadLine
Dim lineYmesos1 As AcadLine
Dim lineYdeiktis1 As AcadLine
```

```
Dim lineYmikros2 As AcadLine
Dim lineYparamesos2 As AcadLine
Dim lineYmesos2 As AcadLine
Dim lineYdeiktis2 As AcadLine
```

```
Dim lineZantixeiras1 As AcadLine
```

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
Πτυχιακή εργασία του Καταρέλη Δημήτριου
Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

```
Dim lineZantixeiras2 As AcadLine
```

```
Dim acadapp As Autodesk.AutoCAD.Interop.AcadApplication
```

8 τιμές στις οποίες θα αποθηκεύονται οι τιμές της σειριακής καθώς και ένα bit το οποίο με βοήθησε στο προγραμματισμό του φίλτρου καθώς άλλες 16 μεταβλητές τύπου double

```
Dim MOfilter As Boolean
Dim readSerial1 As String
Dim readSerial2 As String
Dim readSerial3 As String
Dim readSerial4 As String
Dim readSerial5 As String
Dim readSerial6 As String
Dim readSerial7 As String
Dim readSerial8 As String
Dim Ttimi1 As Double = 0
Dim Ttimi2 As Double = 0
Dim Ttimi3 As Double = 0
Dim Ttimi4 As Double = 0
Dim Ttimi5 As Double = 0
Dim Ttimi6 As Double = 0
Dim Ttimi7 As Double = 0
Dim Ttimi8 As Double = 0
Dim Ftimi1 As Double = 0
Dim Ftimi2 As Double = 0
Dim Ftimi3 As Double = 0
Dim Ftimi4 As Double = 0
Dim Ftimi5 As Double = 0
Dim Ftimi6 As Double = 0
Dim Ftimi7 As Double = 0
Dim Ftimi8 As Double = 0
Dim FreadSerial1 As Double
Dim FreadSerial2 As Double
Dim FreadSerial3 As Double
Dim FreadSerial4 As Double
Dim FreadSerial5 As Double
```

```
Dim FreadSerial6 As Double
Dim FreadSerial7 As Double
Dim FreadSerial8 As Double
```

Εδώ δημιουργώ και άλλες μεταβλητές που με βοηθάνε για το σχεδιασμό του φίλτρου. Περισσότερα στη τρίτη ενότητα που αναφέρεται στην σειριακή επικοινωνία

```
Dim Rangle1 As Double
Dim Rangle2 As Double
Dim Rangle3 As Double
Dim Rangle4 As Double
Dim Rangle5 As Double
Dim Rangle6 As Double
Dim Rangle7 As Double
Dim Rangle8 As Double
Dim RangleBit As Boolean
Dim TRangle1 As Double
Dim TRangle2 As Double
Dim TRangle3 As Double
Dim TRangle4 As Double
Dim TRangle5 As Double
Dim TRangle6 As Double
Dim TRangle7 As Double
Dim TRangle8 As Double
Dim FRangle1 As Double
Dim FRangle2 As Double
Dim FRangle3 As Double
Dim FRangle4 As Double
Dim FRangle5 As Double
Dim FRangle6 As Double
Dim FRangle7 As Double
Dim FRangle8 As Double
```

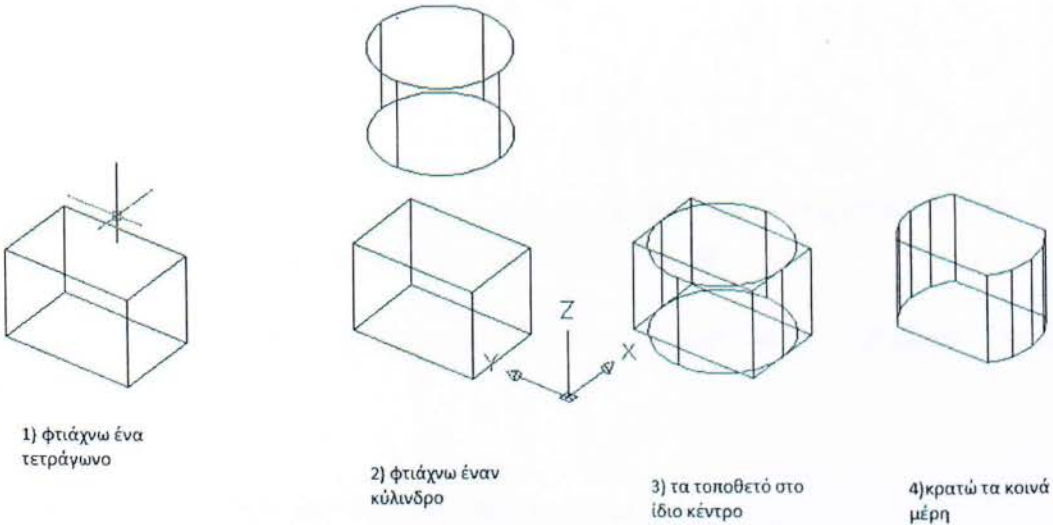
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDrowhand.Click
```

```
    acadapp = New AcadApplication
    acadapp.Visible = True
    Dim acadDoc As AcadDocument
    acadDoc = acadapp.ActiveDocument
```

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
Πτυχιακή εργασία του Καπαρέλη Δημήτριου
Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

Εδώ θα δημιουργήσω κάποιους κύβους οι οποίοι θα με βοηθήσουν στην όμορφη σχεδίαση του χεριού. Η λογική της σχεδίασης του χεριού για να μην βγουν τα δάκτυλα ο πήχης κλπ ούτε τετράγωνα αλλά ούτε κύλινδροι είναι η εξής:

- 1) Φτιάχνω κύβους
 - 2) Στο ίδιο σημείο που έφτιαξα τους κύβους φτιάχνω κυλινδρούς
 - 3) Και στο τέλος προσθέτω τα παραπάνω σχήματα
- Πιο αναλυτικά



Η λογική του παραπάνω σχήματος χρησιμοποιήθηκε για κάθε μέρος του χεριού, δηλαδή για τον πήχη, τη παλάμη και για τα δύο μέρη των δακτύλων.

```
Dim bratsoObj As Acad3DSolid
Dim palamiObj As Acad3DSolid
Dim antixeiras1Obj As Acad3DSolid
Dim antixeiras2Obj As Acad3DSolid
Dim deiktis1Obj As Acad3DSolid
Dim deiktis2Obj As Acad3DSolid
Dim mesos1Obj As Acad3DSolid
Dim mesos2Obj As Acad3DSolid
Dim paramesos1Obj As Acad3DSolid
Dim paramesos2Obj As Acad3DSolid
Dim mikros1Obj As Acad3DSolid
Dim mikros2Obj As Acad3DSolid
Dim centerBratso(0 To 2) As Double 'x,Y,z
Dim XBratso As Double
Dim YBratso As Double
```

```
Dim ZBratso As Double
```

```
'ta dilono
```

```
centerBratso(0) = 15.0# : centerBratso(1) = -35.0# : centerBratso(2) = 150
```

```
XBratso = 30.0# : YBratso = 70.0# : ZBratso = 300.0#
```

```
'ta zografizo
```

```
bratsoObj = acadDoc.ModelSpace.AddBox(centerBratso, XBratso, YBratso, ZBratso)
```

```
Dim centerPalami(0 To 2) As Double 'x,Y,z
```

```
Dim XPalami As Double
```

```
Dim YPalami As Double
```

```
Dim ZPalami As Double
```

```
'ta dilono
```

```
centerPalami(0) = 15.0# : centerPalami(1) = -35.0# : centerPalami(2) = 335
```

```
XPalami = 30.0# : YPalami = 90.0# : ZPalami = 70.0#
```

```
'ta zografizo
```

```
palamiObj = acadDoc.ModelSpace.AddBox(centerPalami, XPalami, YPalami, ZPalami)
```

```
Dim centerAntixeiras1(0 To 2) As Double 'x,Y,z
```

```
Dim XAntixeiras1 As Double
```

```
Dim YAntixeiras1 As Double
```

```
Dim ZAntixeiras1 As Double
```

```
'ta dilono
```

```
centerAntixeiras1(0) = 15.0# : centerAntixeiras1(1) = -100.0# : centerAntixeiras1(2) = 335.0#
```

```
XAntixeiras1 = 30.0# : YAntixeiras1 = 40.0# : ZAntixeiras1 = 30.0#
```

```
'ta zografizo
```

```
antixeiras1Obj = acadDoc.ModelSpace.AddBox(centerAntixeiras1, XAntixeiras1, YAntixeiras1, ZAntixeiras1)
```

```
'ftiaxno megethei Antixeiras2
```

```
Dim centerAntixeiras2(0 To 2) As Double 'x,Y,z
```

```
Dim XAntixeiras2 As Double
```

```
Dim YAntixeiras2 As Double
```

```
Dim ZAntixeiras2 As Double
```

```
'ta dilono
```

```
centerAntixeiras2(0) = 15.0# : centerAntixeiras2(1) = -140.0# : centerAntixeiras2(2) = 335.0#
```

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

XAntixeiras2 = 30.0# : YAntixeiras2 = 40.0# : ZAntixeiras2 = 30.0#

'ta zofrafizo

antixeiras2Obj = acadDoc.ModelSpace.AddBox(centerAntixeiras2, XAntixeiras2, YAntixeiras2, ZAntixeiras2)

'DEIKTIS

'ftiaxno megethei Deiktis1

Dim centerDeiktis1(0 To 2) As Double 'x,Y,z

Dim XDeiktis1 As Double

Dim YDeiktis1 As Double

Dim ZDeiktis1 As Double

'ta dilono

centerDeiktis1(0) = 15.0# : centerDeiktis1(1) = -68.75# : centerDeiktis1(2) = 390

XDeiktis1 = 30.0# : YDeiktis1 = 22.5# : ZDeiktis1 = 40.0#

'ta zofrafizo

deiktis1Obj = acadDoc.ModelSpace.AddBox(centerDeiktis1, XDeiktis1, YDeiktis1, ZDeiktis1)

'ftiaxno megethei Deiktis2

Dim centerDeiktis2(0 To 2) As Double 'x,Y,z

Dim XDeiktis2 As Double

Dim YDeiktis2 As Double

Dim ZDeiktis2 As Double

'ta dilono

centerDeiktis2(0) = 15.0# : centerDeiktis2(1) = -68.75# : centerDeiktis2(2) = 430

XDeiktis2 = 30.0# : YDeiktis2 = 22.5# : ZDeiktis2 = 40.0#

'ta zofrafizo

deiktis2Obj = acadDoc.ModelSpace.AddBox(centerDeiktis2, XDeiktis2, YDeiktis2, ZDeiktis2)

'MESOS

'ftiaxno megethei Mesos1

Dim centerMesos1(0 To 2) As Double 'x,Y,z

Dim XMesos1 As Double

Dim YMesos1 As Double

Dim ZMesos1 As Double

'ta dilono

centerMesos1(0) = 15.0# : centerMesos1(1) = -46.25# : centerMesos1(2) = 390

XMesos1 = 30.0# : YMesos1 = 22.5# : ZMesos1 = 40.0#

```

centerParamesos2(0) = 15.0# : centerParamesos2(1) = -23.75# :
centerParamesos2(2) = 430
XParamesos2 = 30.0# : YParamesos2 = 22.5# : ZParamesos2 = 40.0#
'ta zofrafizo
paramesos2Obj = acadDoc.ModelSpace.AddBox(centerParamesos2,
XParamesos2, YParamesos2, ZParamesos2)

```

```
'MIKROS
```

```
'ftiaxno megethei Mikros1
```

```
Dim centerMikros1(0 To 2) As Double 'x,Y,z
```

```
Dim XMikros1 As Double
```

```
Dim YMikros1 As Double
```

```
Dim ZMikros1 As Double
```

```
'ta dilono
```

```
centerMikros1(0) = 15.0# : centerMikros1(1) = -1.25# : centerMikros1(2) = 390
```

```
XMikros1 = 30.0# : YMikros1 = 22.5# : ZMikros1 = 40.0#
```

```
'ta zofrafizo
```

```
mikros1Obj = acadDoc.ModelSpace.AddBox(centerMikros1, XMikros1, YMikros1,
ZMikros1)
```

```
'ftiaxno megethei Mikros2
```

```
Dim centerMikros2(0 To 2) As Double 'x,Y,z
```

```
Dim XMikros2 As Double
```

```
Dim YMikros2 As Double
```

```
Dim ZMikros2 As Double
```

```
'ta dilono
```

```
centerMikros2(0) = 15.0# : centerMikros2(1) = -1.25# : centerMikros2(2) = 430
```

```
XMikros2 = 30.0# : YMikros2 = 22.5# : ZMikros2 = 40.0#
```

```
'ta zofrafizo
```

```
mikros2Obj = acadDoc.ModelSpace.AddBox(centerMikros2, XMikros2, YMikros2,
ZMikros2)
```

Για να γίνει η τρισδιάστατη περιστροφή του χεριού πρέπει να δηλωθεί στον κώδικα μία γραμμή που γύρω από αυτήν θα γίνεται η περιστροφή. Αυτές οι γραμμές θα είναι κινούμενες επειδή κάθε φορά το σημείο που αφήνεται η περιστροφή μπορεί να αλλάξει από τη περιστροφή ενός αντικείμενου που βρίσκεται πιο κάτω στον άξονα του ύψους(Z). γι' αυτό το λόγο μαζί με τη περιστροφή των μέρων του χεριού γίνεται και περιστροφή αυτών των γραμμών.

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

```

Dim APLineYxeri(2) As Double
Dim TPLineYxeri(2) As Double

APLineYxeri(0) = 15 : APLineYxeri(1) = 0 : APLineYxeri(2) = 0
TPLineYxeri(0) = 15 : TPLineYxeri(1) = -70 : TPLineYxeri(2) = 0
'zografizo th grammh
lineYxeri = acadDoc.ModelSpace.AddLine(APLineYxeri, TPLineYxeri)
Dim APLineZxeri(2) As Double
Dim TPLineZxeri(2) As Double
'ta orizo
APLineZxeri(0) = 15 : APLineZxeri(1) = -35 : APLineZxeri(2) = 0
TPLineZxeri(0) = 15 : TPLineZxeri(1) = -35 : TPLineZxeri(2) = 300
lineZxeri = acadDoc.ModelSpace.AddLine(APLineZxeri, TPLineZxeri)
Dim APLineYpalami(2) As Double
Dim TPLineYpalami(2) As Double
'ta orizo
APLineYpalami(0) = 15 : APLineYpalami(1) = 10 : APLineYpalami(2) = 300
TPLineYpalami(0) = 15 : TPLineYpalami(1) = -80 : TPLineYpalami(2) = 300
'zografizo th grammh
lineYpalami = acadDoc.ModelSpace.AddLine(APLineYpalami, TPLineYpalami)
Dim APLineZantixeiras1(2) As Double
Dim TPLineZantixeiras1(2) As Double
'ta orizo
APLineZantixeiras1(0) = 15 : APLineZantixeiras1(1) = -80 : APLineZantixeiras1(2) =
320
TPLineZantixeiras1(0) = 15 : TPLineZantixeiras1(1) = -80 : TPLineZantixeiras1(2) =
350
'zografizo th grammh
lineZantixeiras1 = acadDoc.ModelSpace.AddLine(APLineZantixeiras1,
TPLineZantixeiras1)

' 5)lineZantixeiras2

'dhmiourgo ta shmeia ths
Dim APLineZantixeiras2(2) As Double
Dim TPLineZantixeiras2(2) As Double
'ta orizo
APLineZantixeiras2(0) = 15 : APLineZantixeiras2(1) = -120 : APLineZantixeiras2(2) =
320

```

TPLineZantixeiras2(0) = 15 : TPLineZantixeiras2(1) = -120 : TPLineZantixeiras2(2) = 350

```
'zografizo th grammh
lineZantixeiras2      =      acadDoc.ModelSpace.AddLine(APLineZantixeiras2,
TPLineZantixeiras2)
```

' 6)lineYmikros2

'dhmiourgo ta shmeia ths

Dim APLineYmikros2(2) As Double

Dim TPLineYmikros2(2) As Double

'ta orizo

APLineYmikros2(0) = 15 : APLineYmikros2(1) = 10 : APLineYmikros2(2) = 410

TPLineYmikros2(0) = 15 : TPLineYmikros2(1) = -12.5 : TPLineYmikros2(2) = 410

'zografizo th grammh

```
lineYmikros2 = acadDoc.ModelSpace.AddLine(APLineYmikros2, TPLineYmikros2)
```

' 7)lineYmikros1

'dhmiourgo ta shmeia ths

Dim APLineYmikros1(2) As Double

Dim TPLineYmikros1(2) As Double

'ta orizo

APLineYmikros1(0) = 15 : APLineYmikros1(1) = 10 : APLineYmikros1(2) = 370

TPLineYmikros1(0) = 15 : TPLineYmikros1(1) = -12.5 : TPLineYmikros1(2) = 370

'zografizo th grammh

```
lineYmikros1 = acadDoc.ModelSpace.AddLine(APLineYmikros1, TPLineYmikros1)
```

' 8)lineYparamesos1

'dhmiourgo ta shmeia ths

Dim APLineYparamesos1(2) As Double

Dim TPLineYparamesos1(2) As Double

'ta orizo

APLineYparamesos1(0) = 15 : APLineYparamesos1(1) = -12.5 :

APLineYparamesos1(2) = 370

TPLineYparamesos1(0) = 15 : TPLineYparamesos1(1) = -35 :

TPLineYparamesos1(2) = 370

```
'zografizo th grammh
lineYparamesos1 = acadDoc.ModelSpace.AddLine(APlineYparamesos1,
TPlineYparamesos1)
```

```
' 9)lineYparamesos2
```

```
'dhmiourgo ta shmeia ths
```

```
Dim APlineYparamesos2(2) As Double
```

```
Dim TPlineYparamesos2(2) As Double
```

```
'ta orizo
```

```
APlineYparamesos2(0) = 15 : APlineYparamesos2(1) = -12.5 :
```

```
APlineYparamesos2(2) = 410
```

```
TPlineYparamesos2(0) = 15 : TPlineYparamesos2(1) = -35 :
```

```
TPlineYparamesos2(2) = 410
```

```
'zografizo th grammh
```

```
lineYparamesos2 = acadDoc.ModelSpace.AddLine(APlineYparamesos2,
TPlineYparamesos2)
```

```
' 10)lineYmesos1
```

```
'dhmiourgo ta shmeia ths
```

```
Dim APlineYmesos1(2) As Double
```

```
Dim TPlineYmesos1(2) As Double
```

```
'ta orizo
```

```
APlineYmesos1(0) = 15 : APlineYmesos1(1) = -35 : APlineYmesos1(2) = 370
```

```
TPlineYmesos1(0) = 15 : TPlineYmesos1(1) = -57.5 : TPlineYmesos1(2) = 370
```

```
'zografizo th grammh
```

```
lineYmesos1 = acadDoc.ModelSpace.AddLine(APlineYmesos1, TPlineYmesos1)
```

```
' 11)lineYmesos2
```

```
'dhmiourgo ta shmeia ths
```

```
Dim APlineYmesos2(2) As Double
```

```
Dim TPlineYmesos2(2) As Double
```

```
'ta orizo
```

```
APlineYmesos2(0) = 15 : APlineYmesos2(1) = -35 : APlineYmesos2(2) = 410
```

```
TPlineYmesos2(0) = 15 : TPlineYmesos2(1) = -57.5 : TPlineYmesos2(2) = 410
```

'zografizo th grammh

lineYmesos2 = acadDoc.ModelSpace.AddLine(APlineYmesos2, TPlineYmesos2)

' 12)lineYdeiktis1

'dhmiourgo ta shmeia ths

Dim APlineYdeiktis1(2) As Double

Dim TPlineYdeiktis1(2) As Double

'ta orizo

APlineYdeiktis1(0) = 15 : APlineYdeiktis1(1) = -57.5 : APlineYdeiktis1(2) = 370

TPlineYdeiktis1(0) = 15 : TPlineYdeiktis1(1) = -80 : TPlineYdeiktis1(2) = 370

'zografizo th grammh

lineYdeiktis1 = acadDoc.ModelSpace.AddLine(APlineYdeiktis1, TPlineYdeiktis1)

' 13)lineYdeiktis2

'dhmiourgo ta shmeia ths

Dim APlineYdeiktis2(2) As Double

Dim TPlineYdeiktis2(2) As Double

'ta orizo

APlineYdeiktis2(0) = 15 : APlineYdeiktis2(1) = -57.5 : APlineYdeiktis2(2) = 410

TPlineYdeiktis2(0) = 15 : TPlineYdeiktis2(1) = -80 : TPlineYdeiktis2(2) = 410

'zografizo th grammh

lineYdeiktis2 = acadDoc.ModelSpace.AddLine(APlineYdeiktis2, TPlineYdeiktis2)

Εδώ φτιάχνονται οι κύλινδροι

Dim lineKantixeiras1 As AcadLine

Dim lineKantixeiras2 As AcadLine

Dim APlineKantixeiras1(2) As Double

Dim TPlineKantixeiras1(2) As Double

Dim APlineKantixeiras2(2) As Double

Dim TPlineKantixeiras2(2) As Double

APlineKantixeiras1(0) = 0 : APlineKantixeiras1(1) = -100 : APlineKantixeiras1(2) =

335

```

TPlaneKantixeiras1(0) = 30 : TPlaneKantixeiras1(1) = -100 : TPlaneKantixeiras1(2) =
335
APLineKantixeiras2(0) = 0 : APLineKantixeiras2(1) = -140 : APLineKantixeiras2(2) =
335
TPlaneKantixeiras2(0) = 30 : TPlaneKantixeiras2(1) = -140 : TPlaneKantixeiras2(2) =
335
lineKantixeiras1      =      acadDoc.ModelSpace.AddLine(APLineKantixeiras1,
TPlaneKantixeiras1)
lineKantixeiras2      =      acadDoc.ModelSpace.AddLine(APLineKantixeiras2,
TPlaneKantixeiras2)
'h paren8esh kleinei
')
```

```

Dim Kbratso As Acad3DSolid
Dim RKbratso As Double
RKbratso = YBratso / 2
Kbratso = acadDoc.ModelSpace.AddCylinder(centerBratso, RKbratso, ZBratso)
```

```

Dim Kpalami As Acad3DSolid
Dim RKpalami As Double
RKpalami = YPalami / 2
Kpalami = acadDoc.ModelSpace.AddCylinder(centerPalami, RKpalami, ZPalami)
```

```

Dim Kantixeiras1 As Acad3DSolid
Dim RKantixeiras1 As Double
RKantixeiras1 = 18.0#
Kantixeiras1      =      acadDoc.ModelSpace.AddCylinder(centerAntixeiras1,
RKantixeiras1, YAntixeiras1)
Kantixeiras1.Rotate3D(APLineKantixeiras1, TPlaneKantixeiras1, 1.575)
```

```

Dim Kantixeiras2 As Acad3DSolid
Dim RKantixeiras2 As Double
```

```

RKantixeiras2 = 18.0#
Kantixeiras2 = acadDoc.ModelSpace.AddCylinder(centerAntixeiras2,
RKantixeiras1, YAntixeiras2)
Kantixeiras2.Rotate3D(APlineKantixeiras2, TPLineKantixeiras2, 1.575)

```

```

lineKantixeiras1.Delete()
lineKantixeiras2.Delete()

```

```

Dim Kdeiktis1 As Acad3DSolid
Dim RKdeiktis1 As Double
RKdeiktis1 = XDeiktis1 / 2
Kdeiktis1 = acadDoc.ModelSpace.AddCylinder(centerDeiktis1, RKdeiktis1,
ZDeiktis1)

```

```

Dim Kdeiktis2 As Acad3DSolid
Dim RKdeiktis2 As Double
RKdeiktis2 = XDeiktis2 / 2
Kdeiktis2 = acadDoc.ModelSpace.AddCylinder(centerDeiktis2, RKdeiktis2,
ZDeiktis2)

```

```

Dim Kmesos1 As Acad3DSolid
Dim RKmesos1 As Double
RKmesos1 = XMesos1 / 2
Kmesos1 = acadDoc.ModelSpace.AddCylinder(centerMesos1, RKmesos1,
ZMesos1)

```

```

Dim Kmesos2 As Acad3DSolid
Dim RKmesos2 As Double
RKmesos2 = XMesos2 / 2
Kmesos2 = acadDoc.ModelSpace.AddCylinder(centerMesos2, RKmesos2,
ZMesos2)

```



```

Dim Kparamesos1 As Acad3DSolid
Dim RKparamesos1 As Double
RKparamesos1 = XParamesos1 / 2
Kparamesos1 = acadDoc.ModelSpace.AddCylinder(centerParamesos1,
RKparamesos1, ZParamesos1)

```

```

Dim Kparamesos2 As Acad3DSolid
Dim RKparamesos2 As Double
RKparamesos2 = XParamesos2 / 2
Kparamesos2 = acadDoc.ModelSpace.AddCylinder(centerParamesos2,
RKparamesos2, ZParamesos2)

```

```

Dim Kmikros1 As Acad3DSolid
Dim RKmikros1 As Double
RKmikros1 = XMikros1 / 2
Kmikros1 = acadDoc.ModelSpace.AddCylinder(centerMikros1, RKmikros1,
ZMikros1)

```

```

Dim Kmikros2 As Acad3DSolid
Dim RKmikros2 As Double
RKmikros2 = XMikros2 / 2
Kmikros2 = acadDoc.ModelSpace.AddCylinder(centerMikros2, RKmikros2,
ZMikros2)

```

Τέλος ζωγραφίζω τα τελικά αντικείμενα και σβήνω τα αντικείμενα κυλίνδρους και κύβους.

```

'ORIZO KAI ZOGRAFIZO TA TELIKA ANTIKEIMENA thexxx
'ta zografizo
'svino kiklous kai kubous

```

```

thebratsoObj = bratsoObj.CheckInterference(Kbratso, True)
bratsoObj.Delete()
Kbratso.Delete()

```

```
thepalamiObj = palamiObj.CheckInterference(Kpalami, True)  
palamiObj.Delete()  
Kpalami.Delete()
```

```
theantixeiras1Obj = antixeiras1Obj.CheckInterference(Kantixeiras1, True)  
antixeiras1Obj.Delete()  
Kantixeiras1.Delete()
```

```
theantixeiras2Obj = antixeiras2Obj.CheckInterference(Kantixeiras2, True)  
antixeiras2Obj.Delete()  
Kantixeiras2.Delete()
```

```
thedeiktis1Obj = deiktis1Obj.CheckInterference(Kdeiktis1, True)  
deiktis1Obj.Delete()  
Kdeiktis1.Delete()
```

```
thedeiktis2Obj = deiktis2Obj.CheckInterference(Kdeiktis2, True)  
deiktis2Obj.Delete()  
Kdeiktis2.Delete()
```

```
themesos1Obj = mesos1Obj.CheckInterference(Kmesos1, True)  
mesos1Obj.Delete()  
Kmesos1.Delete()
```

```
themesos2Obj = mesos2Obj.CheckInterference(Kmesos2, True)  
mesos2Obj.Delete()  
Kmesos2.Delete()
```

```
theparamesos1Obj = paramesos1Obj.CheckInterference(Kparamesos1, True)  
paramesos1Obj.Delete()  
Kparamesos1.Delete()
```

```
theparamesos2Obj = paramesos2Obj.CheckInterference(Kparamesos2, True)  
paramesos2Obj.Delete()  
Kparamesos2.Delete()
```

```
themikros1Obj = mikros1Obj.CheckInterference(Kmikros1, True)
mikros1Obj.Delete()
Kmikros1.Delete()
```

```
themikros2Obj = mikros2Obj.CheckInterference(Kmikros2, True)
mikros2Obj.Delete()
Kmikros2.Delete()
```

Κάνω ενημέρωση στο αντικείμενο της εφαρμογής και το χέρι σχηματίζεται στο autocad
 'kano update sto autocad
 acadapp.Update()

4.4) Η δεύτερη ενότητα της φόρμας της visual basic.

Στη δεύτερη ενότητα γίνονται οι περιστροφές των μέρων του χεριού από τα κουμπιά. Η visual basic δε χρησιμοποιεί μοίρες στην περιστροφή, αλλά ακτίνια. Με το πάτημα του κουμπιού γίνεται η περιστροφή του επιθυμητού μέρους του χεριού κατά 5 μοίρες
 Ο ΚΩΔΙΚΑΣ ΤΗΣ ΔΕΥΤΕΡΗΣ ΕΝΟΤΗΤΑΣ

Θα δούμε ένα κουμπί τι κάνει και είναι εύκολο να καταλάβουμε τι κάνουν και τα υπόλοιπα.

Όταν κάνει ο χρήστης κλικ στο κουμπί btnagkopasz εύκολα μπορούμε να καταλάβουμε πως κάνει περιστροφή στον αγκώνα(άρα και στο υπόλοιπο χέρι) στον άξονα Z

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
 Πτυχιακή εργασία του Καπαρέλη Δημήτριου
 Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

Private Sub btnagkonasz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnagkonasz.Click

Στα σχόλια αναφέρω ποιες γραμμές και πια αντικείμενα θα περιστρέψω.

```
lineYpalami.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineYdeiktis2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineYdeiktis1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineYmesos2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineYmesos1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineYparamesos2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineYparamesos1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineYmikros1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineYmikros2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineZantixeiras1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
lineZantixeiras2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

Ας πάρουμε παράδειγμα την παρακάτω εντολή. Για να κάνουμε περιστροφή στο αντικείμενο thebratsoObj πρέπει αρχικά να πούμε πια ιδιότητα του αντικειμένου αυτού θα χρησιμοποιήσουμε. Εμείς χρησιμοποιούμε εδώ την ιδιότητα Rotate3d, έπειτα πρέπει να δηλώσουμε τη γραμμή πάνω στην οποία θα κάνουμε τη περιστροφή. Η βιβλιοθήκη που χρησιμοποιούμε στην εργασία αυτή για να γίνει η περιστροφή δε δηλώνεται ευθεία αλλά δύο σημεία τα οποία προφανώς απεικονίζουν μια ευθεία. Εδώ χρησιμοποιήθηκε η αρχή και το τέλος της ευθείας lineZxeri που ορίσαμε στο declaration. Τέλος δηλώνουμε πόσα ακτίνια θα γίνει η περιστροφή.

```
thebratsoObj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
thepalamiObj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
theantixeiras1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
theantixeiras2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
thedeiktis1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
thedeiktis2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
themesos1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
themesos2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
theparamesos1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
theparamesos2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
themikros1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
themikros2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, 0.0875#)
```

```
'kano update
```

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

```
acadapp.Update()
```

```
End Sub
```

Σε αυτό το κουμπί γίνεται περιστροφή πάλι του αγκώνα αλλά στον άξονα Y
 Private Sub btnAgkonasy_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAgkonasy.Click

```
lineYxeri.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineZxeri.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineYpalami.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
```

```
lineYdeiktis2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineYdeiktis1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineYmesos2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineYmesos1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineYparamesos2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineYparamesos1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineYmikros1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineYmikros2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
```

```
lineZantixeiras1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
lineZantixeiras2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
```

```
thebratsoObj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
thepalamiObj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
theantixeiras1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
theantixeiras2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
thedeiktis1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
thedeiktis2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
themesos1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
themesos2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
theparamesos1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
theparamesos2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
themikros1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
themikros2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, 0.0875#)
```

```
acadapp.Update()
```

End Sub

```
Private Sub btnPalami_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPalami.Click
```

```
    lineYpalami.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    lineYdeiktis2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    lineYdeiktis1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    lineYmesos2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    lineYmesos1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    lineYparamesos2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
```

```
0.0875#)
```

```
    lineYparamesos1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
```

```
0.0875#)
```

```
    lineYmikros1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    lineYmikros2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    lineZantixeiras1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    lineZantixeiras2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    thepalamiObj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    theantixeiras1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
```

```
0.0875#)
```

```
    theantixeiras2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
```

```
0.0875#)
```

```
    thedeiktis1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    thedeiktis2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    themesos1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    themesos2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    theparamesos1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
```

```
0.0875#)
```

```
    theparamesos2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
```

```
0.0875#)
```

```
    themikros1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    themikros2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, 0.0875#)
```

```
    acadapp.Update()
```

End Sub

```
Private Sub btnMikros1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnMikros1.Click
```

```
    lineYmikros1.Rotate3D(lineYmikros1.StartPoint, lineYmikros1.EndPoint, 0.0875)
    lineYmikros2.Rotate3D(lineYmikros1.StartPoint, lineYmikros1.EndPoint, 0.0875)
    themikros1Obj.Rotate3D(lineYmikros1.StartPoint, lineYmikros1.EndPoint, 0.0875)
    themikros2Obj.Rotate3D(lineYmikros1.StartPoint, lineYmikros1.EndPoint, 0.0875)
```

```
    acadapp.Update()
```

```
End Sub
```

```
Private Sub btnMikros2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnMikros2.Click
```

```
    lineYmikros2.Rotate3D(lineYmikros2.StartPoint, lineYmikros2.EndPoint, 0.0875#)
```

```
    themikros2Obj.Rotate3D(lineYmikros2.StartPoint, lineYmikros2.EndPoint, 0.0875#)
```

```
    acadapp.Update()
```

```
End Sub
```

```
Private Sub btnParamesos1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnParamesos1.Click
```

```
    lineYparamesos1.Rotate3D(lineYparamesos1.StartPoint, lineYparamesos1.EndPoint, 0.0875#)
```

```
    lineYparamesos2.Rotate3D(lineYparamesos1.StartPoint, lineYparamesos1.EndPoint, 0.0875#)
```

```
    theparamesos1Obj.Rotate3D(lineYparamesos1.StartPoint, lineYparamesos1.EndPoint, 0.0875#)
```

```
    theparamesos2Obj.Rotate3D(lineYparamesos1.StartPoint, lineYparamesos1.EndPoint, 0.0875#)
```

```
    acadapp.Update()
```

```
End Sub
```

```
Private Sub btnMesos1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnMesos1.Click
```

```
lineYmesos1.Rotate3D(lineYmesos1.StartPoint, lineYmesos1.EndPoint, 0.0875#)
lineYmesos2.Rotate3D(lineYmesos1.StartPoint, lineYmesos1.EndPoint, 0.0875#)
```

```
themesos1Obj.Rotate3D(lineYmesos1.StartPoint, lineYmesos1.EndPoint,
0.0875#)
themesos2Obj.Rotate3D(lineYmesos1.StartPoint, lineYmesos1.EndPoint,
0.0875#)
```

```
acadapp.Update()
End Sub
```

```
Private Sub btnMesos2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMesos2.Click
```

```
lineYmesos2.Rotate3D(lineYmesos2.StartPoint, lineYmesos2.EndPoint, 0.0875#)
themesos2Obj.Rotate3D(lineYmesos2.StartPoint, lineYmesos2.EndPoint,
0.0875#)
```

```
acadapp.Update()
End Sub
```

```
Private Sub btnDeiktis1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeiktis1.Click
```

```
lineYdeiktis1.Rotate3D(lineYdeiktis1.StartPoint, lineYdeiktis1.EndPoint, 0.0875#)
lineYdeiktis2.Rotate3D(lineYdeiktis1.StartPoint, lineYdeiktis1.EndPoint, 0.0875#)
```

```
thedeiktis1Obj.Rotate3D(lineYdeiktis1.StartPoint, lineYdeiktis1.EndPoint, 0.0875#)
thedeiktis2Obj.Rotate3D(lineYdeiktis1.StartPoint, lineYdeiktis1.EndPoint, 0.0875#)
```

```
acadapp.Update()
End Sub
```

```
Private Sub btnDeiktis2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeiktis2.Click
```

```
lineYdeiktis2.Rotate3D(lineYdeiktis2.StartPoint, lineYdeiktis2.EndPoint, 0.0875#)
thedeiktis2Obj.Rotate3D(lineYdeiktis2.StartPoint, lineYdeiktis2.EndPoint, 0.0875#)
```

```
acadapp.Update()
End Sub
```



```
Private Sub btnAntixeiras1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAntixeiras1plin.Click
```

```
    lineZantixeiras1.Rotate3D(lineZantixeiras1.StartPoint, lineZantixeiras1.EndPoint, 0.0875#)
    lineZantixeiras2.Rotate3D(lineZantixeiras1.StartPoint, lineZantixeiras1.EndPoint, 0.0875#)
```

```
    theantixeiras1Obj.Rotate3D(lineZantixeiras1.StartPoint, lineZantixeiras1.EndPoint, 0.0875#)
    theantixeiras2Obj.Rotate3D(lineZantixeiras1.StartPoint, lineZantixeiras1.EndPoint, 0.0875#)
```

```
    acadapp.Update()
End Sub
```

```
Private Sub btnAntixeiras2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAntixeiras2plin.Click
```

```
    lineZantixeiras2.Rotate3D(lineZantixeiras2.StartPoint, lineZantixeiras2.EndPoint, 0.0875#)
```

```
    theantixeiras2Obj.Rotate3D(lineZantixeiras2.StartPoint, lineZantixeiras2.EndPoint, 0.0875#)
```

```
    acadapp.Update()
End Sub
```

```
Private Sub btnParamesos2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnParamesos2.Click
```

```
    lineYparamesos2.Rotate3D(lineYparamesos2.StartPoint, lineYparamesos2.EndPoint, 0.0875)
    theparamesos2Obj.Rotate3D(lineYparamesos2.StartPoint, lineYparamesos2.EndPoint, 0.0875)
```

```
    acadapp.Update()
End Sub
```

Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAgkonasZplin.Click

```
lineZxeri.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
lineYpalami.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
```

```
lineYdeiktis2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
lineYdeiktis1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
lineYmesos2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
lineYmesos1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
lineYparamesos2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
lineYparamesos1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
lineYmikros1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
lineYmikros2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
```

```
lineZantixeiras1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
lineZantixeiras2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
```

```
thebratsoObj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
thepalamiObj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
theantixeiras1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
theantixeiras2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
thedeiktis1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
thedeiktis2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
themesos1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
themesos2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
theparamesos1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
theparamesos2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
themikros1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
themikros2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, -0.0875#)
acadapp.Update()
```

End Sub

Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAntixeiras1.Click

```
lineZantixeiras1.Rotate3D(lineZantixeiras1.StartPoint, lineZantixeiras1.EndPoint, -0.0875#)
```

```
lineZantixeiras2.Rotate3D(lineZantixeiras1.StartPoint, lineZantixeiras1.EndPoint, -0.0875#)
```

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

```
theantixeiras1Obj.Rotate3D(lineZantixeiras1.StartPoint, lineZantixeiras1.EndPoint,
-0.0875#)
```

```
theantixeiras2Obj.Rotate3D(lineZantixeiras1.StartPoint, lineZantixeiras1.EndPoint,
-0.0875#)
```

```
acadapp.Update()
End Sub
```

```
Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAntixeiras2.Click
```

```
lineZantixeiras2.Rotate3D(lineZantixeiras2.StartPoint, lineZantixeiras2.EndPoint, -
0.0875#)
```

```
theantixeiras2Obj.Rotate3D(lineZantixeiras2.StartPoint, lineZantixeiras2.EndPoint,
-0.0875#)
```

```
acadapp.Update()
End Sub
```

```
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMikros2plin.Click
```

```
lineYmikros2.Rotate3D(lineYmikros2.StartPoint, lineYmikros2.EndPoint, -0.0875#)
themikros2Obj.Rotate3D(lineYmikros2.StartPoint, lineYmikros2.EndPoint, -
0.0875#)
```

```
acadapp.Update()
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnParamesos2plin.Click
```

```
lineYparamesos2.Rotate3D(lineYparamesos2.StartPoint,
lineYparamesos2.EndPoint, -0.0875)
theparamesos2Obj.Rotate3D(lineYparamesos2.StartPoint,
lineYparamesos2.EndPoint, -0.0875)
```

```
acadapp.Update()
End Sub
```

```

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMesos2plin.Click
    lineYmesos2.Rotate3D(lineYmesos2.StartPoint, lineYmesos2.EndPoint, -0.0875#)
    themesos2Obj.Rotate3D(lineYmesos2.StartPoint, lineYmesos2.EndPoint, -
0.0875#)
    acadapp.Update()
End Sub

```

```

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeiktis2plin.Click
    lineYdeiktis2.Rotate3D(lineYdeiktis2.StartPoint, lineYdeiktis2.EndPoint, -0.0875#)
    thedeiktis2Obj.Rotate3D(lineYdeiktis2.StartPoint, lineYdeiktis2.EndPoint, -
0.0875#)

    acadapp.Update()
End Sub

```

```

Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMikros1plin.Click

    lineYmikros1.Rotate3D(lineYmikros1.StartPoint, lineYmikros1.EndPoint, -0.0875)
    lineYmikros2.Rotate3D(lineYmikros1.StartPoint, lineYmikros1.EndPoint, -0.0875)

    themikros1Obj.Rotate3D(lineYmikros1.StartPoint, lineYmikros1.EndPoint, -0.0875)
    themikros2Obj.Rotate3D(lineYmikros1.StartPoint, lineYmikros1.EndPoint, -0.0875)

    acadapp.Update()

End Sub

```

```

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnParamesos1plin.Click
    lineYparamesos1.Rotate3D(lineYparamesos1.StartPoint,
lineYparamesos1.EndPoint, -0.0875#)
    lineYparamesos2.Rotate3D(lineYparamesos1.StartPoint,
lineYparamesos1.EndPoint, -0.0875#)

    theparamesos1Obj.Rotate3D(lineYparamesos1.StartPoint,
lineYparamesos1.EndPoint, -0.0875#)

```

```

theparamesos2Obj.Rotate3D(lineYparamesos1.StartPoint,
lineYparamesos1.EndPoint, -0.0875#)

```

```

acadapp.Update()
End Sub

```

```

Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMesos1plin.Click

```

```

lineYmesos1.Rotate3D(lineYmesos1.StartPoint, lineYmesos1.EndPoint, -0.0875#)
lineYmesos2.Rotate3D(lineYmesos1.StartPoint, lineYmesos1.EndPoint, -0.0875#)

```

```

themesos1Obj.Rotate3D(lineYmesos1.StartPoint, lineYmesos1.EndPoint, -
0.0875#)
themesos2Obj.Rotate3D(lineYmesos1.StartPoint, lineYmesos1.EndPoint, -
0.0875#)

```

```

acadapp.Update()
End Sub

```

```

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeiktis1plin.Click

```

```

lineYdeiktis1.Rotate3D(lineYdeiktis1.StartPoint, lineYdeiktis1.EndPoint, -0.0875#)
lineYdeiktis2.Rotate3D(lineYdeiktis1.StartPoint, lineYdeiktis1.EndPoint, -0.0875#)

```

```

thedeiktis1Obj.Rotate3D(lineYdeiktis1.StartPoint, lineYdeiktis1.EndPoint, -
0.0875#)
thedeiktis2Obj.Rotate3D(lineYdeiktis1.StartPoint, lineYdeiktis1.EndPoint, -
0.0875#)

```

```

acadapp.Update()
End Sub

```

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles palamiplin.Click

```

```

lineYpalami.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)

```

```

lineYdeiktis2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)

```

```

lineYdeiktis1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
lineYmesos2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
lineYmesos1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
lineYparamesos2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -
0.0875#)
lineYparamesos1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -
0.0875#)
lineYmikros1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
lineYmikros2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)

lineZantixeiras1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
lineZantixeiras2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)

thepalamiObj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
theantixeiras1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -
0.0875#)
theantixeiras2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -
0.0875#)
thedeiktis1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
thedeiktis2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
themesos1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
themesos2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
theparamesos1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -
0.0875#)
theparamesos2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -
0.0875#)
themikros1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)
themikros2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, -0.0875#)

acadapp.Update()
End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAgkonasYplin.Click

```

```

lineYxeri.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)
lineZxeri.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)
lineYpalami.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)

```

```

lineYdeiktis2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)

```

```
lineYdeiktis1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
lineYmesos2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
lineYmesos1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
lineYparamesos2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
lineYparamesos1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
lineYmikros1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
lineYmikros2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)
```

```
lineZantixeiras1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
lineZantixeiras2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)
```

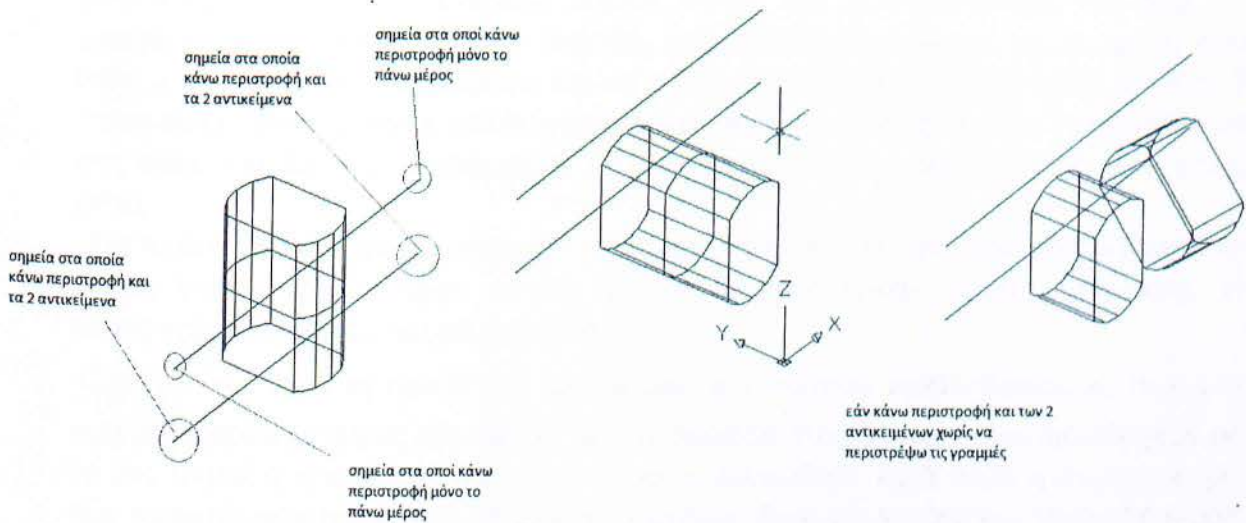
```
thebratsoObj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
thepalamiObj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
theantixeiras1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
theantixeiras2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
thedeiktis1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
thedeiktis2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
themesos1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
themesos2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
theparamesos1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
theparamesos2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
themikros1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)  
themikros2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, -0.0875#)
```

```
acadapp.Update()  
End Sub
```

Μία απορία που ίσως δημιουργηθεί είναι γιατί πρέπει να κάνουμε περιστροφή και στις γραμμές ή ακόμη γιατί να δημιουργήσουμε τις γραμμές και να μη δηλώσουμε απευθείας 3 σημεία (χ,ψ,ζ δηλαδή). Εάν γινόταν αυτό, τότε τα πάνω μέρη του χεριού θα έκαναν περιστροφή σε σημεία στα οποία δε θα έπρεπε.

Για παράδειγμα θα κάνουμε μία περιστροφή δύο αντικειμένων το ένα πάνω στο άλλο με σταθερά σημεία και με κινούμενα να δούμε τις διαφορές.

A) περιστροφή με σταθερά σημεία ή γραμμές



Όπως βλέπουμε και στο σχήμα, εάν γίνει περιστροφή και των δύο αντικειμένων στα μπλε σημεία και δεν κινηθούν τα πράσινα, μετά στη περιστροφή μόνο του πάνω αντικειμένου στη πάνω γραμμή έχουμε αυτά τα μη επιθυμητά αποτελέσματα

B) περιστροφή με κινούμενα σημεία ή γραμμές

Εδώ εάν περιστρέψουμε και τα δύο αντικείμενα, θα περιστρέψουμε και τη πάνω γραμμή ή τα πράσινα σημεία και στη συνέχεια κάνοντας περιστροφή του πάνω αντικειμένου στα πράσινα σημεία η περιστροφή θα είναι επιθυμητή. Γι' αυτό το λόγο η περιστροφή των γραμμών μαζί με τα αντικείμενα κρίνεται απαραίτητη.

4.5 Η Τρίτη ενότητα της φόρμας της visual basic

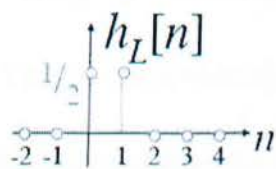
Η Τρίτη ενότητα της φόρμας σχετίζεται με την σειριακή επικοινωνία της visual basic με τον AT mega16. Εδώ παρουσιάζονται, πολλά text boxes τα οποία ο χρήστης μπορεί να διακρίνει τις τιμές της σειριακής και τις αλλαγές τους. Επίσης θα δημιουργήσουμε και έναν χρονιστή. Ο χρονιστής ενεργοποιείται, μόλις επιλέξουμε την ενεργοποίηση της σειριακής επικοινωνίας και όποτε εκπνέει τότε γίνεται η επικοινωνία, η επεξεργασία των ψηφιακών τιμών των αναλογικών θυρών καθώς και οι περιστροφές στο χέρι. Ο χρονιστής εκπνέει κάθε 50 msec- δηλαδή, κάθε 0.05 δευτερόλεπτα και το χρόνο που θέλει ο μικροελεγκτής να διαβάσει και να στείλει τις τιμές στον υπολογιστή γίνονται 8 περιστροφές στις δεκατρείς κλειδώσεις του χεριού. περιστροφές από τα ποτενσιόμετρα στις πάνω κλειδώσεις των δακτύλων δε γίνονται επειδή τα ποτενσιόμετρα είναι μόλις οκτώ.

Για το φιλτράρισμα των αναλογικών τιμών των ποτενσιομέτρων χρησιμοποιήθηκε ένα φίλτρο μέσου όρου (αφού πρώτα φυσικά οι τιμές έγιναν πρώτα ψηφιακές). Η συνάρτηση μεταφοράς του φίλτρου είναι

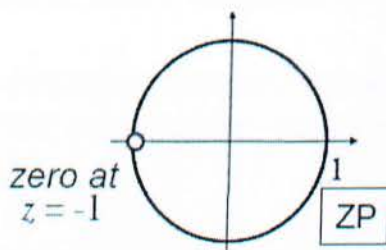
$G(z) = \frac{z}{2} + \frac{z^{-1}}{2}$.Από τη συνάρτηση μεταφοράς του φίλτρου καταλαβαίνουμε, πως μία τιμή της σειριακής θύρας αθροίστηκε με την ακριβώς προηγούμενη και διαιρέθηκαν με το δύο (δηλαδή μέσω όρο). Με αυτό το τρόπο βελτιώθηκε κατά πολύ η συμπεριφορά των περιστροφών του χεριού ειδικά όταν ο χρόνος δειγματοληψίας των τιμών (δηλαδή ο χρονιστής) είναι μικρός.

Simple FIR Lowpass

- $h_L[n] = \left\{ \underset{\uparrow}{1/2} \ 1/2 \right\}$
 (2 pt moving avg.)

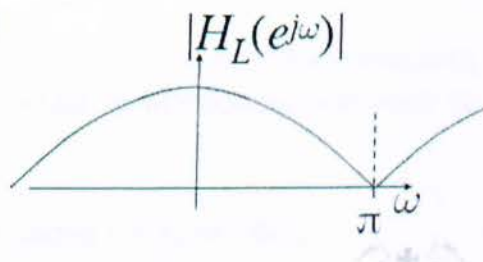


$$H_L(z) = \frac{1}{2}(1 + z^{-1}) = \frac{z+1}{2z}$$



$$\Rightarrow H_L(e^{j\omega}) = e^{-j\omega/2} \overset{e^{j\omega/2} + e^{-j\omega/2}}{\cos(\omega/2)}$$

\uparrow
1/2 sample delay



Στο παραπάνω σχήμα παρατηρείται το φάσμα πλάτους του φίλτρου, καθώς και τον πόλο του στο πεδίο της συχνότητας και την χρονική του απόκριση.

2.3.5.2 Ο κώδικας της τρίτης ενότητας

Η λογική του κομματιού του κώδικα που διαβάζει τις σειριακές τιμές και κάνει τις περιστροφές στο χέρι είναι η εξής.

ΑΡΧΗ;

- ΑΝΑΠΟΔΟΓΥΡΙΣΕ ΤΑ BIT ΜΟΦΙΛΤΕΡ ΚΑΙ RANGLEBIT
- ΦΙΛΤΡΑΡΙΣΜΕΝΗ

$$\text{ΤΙΜΗ} = \frac{\text{ΤΙΜΗ ΣΕΙΡΙΑΚΗΣ ΓΙΑ ΜΟΦΙΛΤΕΡ}=1 + \text{ΤΙΜΗ ΣΕΙΡΙΑΚΗΣ ΓΙΑ ΜΟΦΙΛΤΕΡ}=0}{2}$$

- ΠΑΡΟΥΣΙΑΣΗ ΦΙΛΤΡΑΡΙΣΜΕΝΩΝ ΤΙΜΩΝ
- ΔΙΑΒΑΣΜΑ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΤΙΜΩΝ ΣΕΙΡΙΑΚΗΣ

- ΕΑΝ ΤΟ MOFILTER=0 ΑΠΟΘΗΚΕΥΣΕ ΤΙΣ ΤΙΜΕΣ ΣΕΙΡΗΑΚΗΣ ΣΤΟ FTIMH
- ΕΑΝ ΤΟ MOFILTER=1 ΑΠΟΘΗΚΕΥΣΕ ΤΙΣ ΤΙΜΕΣ ΣΕΙΡΙΑΚΗΣ ΣΤΟ TTIMH
- ΕΑΝ ΤΟ RANGLEBIT=1 ΤΟ TRANGLE=FREADSERIAL(το trangle είναι η γωνία όταν το ranglebit είναι 1 και η freadserial είναι η φιλτραρισμένη τιμή της σειριακής θύρας)
- ΕΑΝ ΤΟ RANGLEBIT=0 ΤΟ FRANGLE =FREADSERIAL ΚΑΙ ΚΑΝΕ ΤΙΣ ΠΕΡΙΣΤΡΟΦΕΣ ΜΕ ΒΑΣΗ ΤΙΣ ΜΕΤΑΒΛΗΤΕΣ RANGLEX(X=1,2,3,4 κλπ)
- $RANGLE=(FRANGLE-TRANGLE)*0.01745$
- ΠΗΓΑΙΝΕ ΣΤΗΝ ΑΡΧΗ

Σε αυτή την ενότητα θα αναλυθεί ο κώδικας, ακριβώς όπως και στις προηγούμενες δηλαδή με κόκκινα γράμματα. Επιπλέον, μέσα στο κώδικα θα περιγραφεί τι κάνουν τα κομμάτια.

Ο κώδικας

Το παρακάτω κομμάτι του κώδικα αναφέρεται στην επιλογή του CheckBox.

```
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles checkSeiriaki.CheckedChanged
```

```
    If checkSeiriaki.Checked Then
```

```
        Timer.Enabled = True
```

```
        sp1.Open()
```

```
    Else
```

```
        Timer.Enabled = False
```

```
        sp1.Close()
```

```
    End If
```

```
End Sub
```

Στο παραπάνω κομμάτι του κώδικα εάν “κλικάρουμε” το check box τότε ανοίγουμε τη σειριακή επικοινωνία και ενεργοποιούμε το χρονικό, ενώ αντίθετα εάν “ξεκλικάρουμε” το check box κλείνουμε τη σειριακή επικοινωνία και απενεργοποιούμε τον χρονιστή.

Το παρακάτω κομμάτι του κώδικα ενεργοποιείται, όταν εκπνεύσει ο χρονιστής. Μετά από δοκιμές ο χρόνος εκπνοής του χρονικού επιλέχθηκε στα 50 msec, αλλά, εάν αλλάξουμε το χρόνο σε 1 ή 2 δευτερόλεπτα θα κατανοήσουμε καλύτερα τη λειτουργία του φίλτρου και γενικότερα όλου του κώδικα.

```
Private Sub Timer_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer.Tick
```

Σε αυτό το σημείο αναποδογυρίζω δύο bit τα bit αυτά είναι τα MOfilter και RangleBit . Όπως θα παρατηρηθεί και παρακάτω αυτά κάνουν τα εξής: Το MOfilter, όταν είναι 0 αποθηκεύει τις τιμές της σειριακής θύρας και όταν είναι 1 το ίδιο. Τα προσθέτει και το αποτέλεσμα το διαιρεί με το δύο και παίρνουμε έτσι το μέσο όρο. Στο Ranglebit τώρα όταν είναι 0 αποθηκεύει τη φιλτραρισμένη τιμή και όταν είναι 1 αποθηκεύει πάλι τη φιλτραρισμένη τιμή βρίσκει τη διαφορά τους και έτσι υπολογίζουμε τη γωνία περιστροφής των κλειδώσεων του χειριού. Τέλος κάνουμε τις περιστροφές στο χέρι.

'tooglaro ta bit

MOfilter = Not MOfilter

RangleBit = Not RangleBit

'vlepo th timh tou

txtbit.Text = MOfilter

Εδώ είναι οι φιλτραρισμένες τιμές που παρουσιάζονται στα text boxes.

'oi telikes fitrarismenes times

FreadSerial1 = (Ttimi1 + Ftimi1) / 2

FreadSerial2 = (Ttimi2 + Ftimi2) / 2

FreadSerial3 = (Ttimi3 + Ftimi3) / 2

FreadSerial4 = (Ttimi4 + Ftimi4) / 2

FreadSerial5 = (Ttimi5 + Ftimi5) / 2

FreadSerial6 = (Ttimi6 + Ftimi6) / 2

FreadSerial7 = (Ttimi7 + Ftimi7) / 2

FreadSerial8 = (Ttimi8 + Ftimi8) / 2

'oi telikes fitrarismenes times parousiazontai

txtFadc0.Text = FreadSerial1

txtFadc1.Text = FreadSerial2

txtFadc2.Text = FreadSerial3

txtFadc3.Text = FreadSerial4

txtFadc4.Text = FreadSerial5

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

```
txtFadc5.Text = FreadSerial6
txtFadc6.Text = FreadSerial7
txtFadc7.Text = FreadSerial8
```

Το παρακάτω σημείο του κώδικα είναι το κομμάτι που επικοινωνεί με τη πλακέτα Όπως μπορούμε να δούμε αρχικά γράφουμε τον αριθμό της αναλογικής θύρας που θέλουμε να διαβάσουμε (στέλνουμε στη πλακέτα 1,2,3 κλπ πάντα σε χαρακτήρες). Έπειτα η πλακέτα θα μας επιστρέψει μία τιμή αυτή αποθηκεύεται στην μεταβλητή readseiralx(x=1,2,3,4 κλπ) και έτσι γίνεται η επικοινωνία του υπολογιστή με τον atmega16

```
'gia thn diavaso thn adc0 apo th seireiaki
sp1.Write("1")
```

```
readSerial1 = sp1.ReadChar
```

```
'gia thn adc1
sp1.Write("2")
readSerial2 = sp1.ReadChar
```

```
'gia thn adc2
sp1.Write("3")
readSerial3 = sp1.ReadChar
```

```
'gia thn adc3
sp1.Write("4")
readSerial4 = sp1.ReadChar
```

```
'gia thn adc4
sp1.Write("5")
readSerial5 = sp1.ReadChar
```

```
'gia thn adc5
sp1.Write("6")
```

```
readSerial6 = sp1.ReadChar
```

```
'gia thn adc6
```

```
sp1.Write("7")
```

```
readSerial7 = sp1.ReadChar
```

```
'gia thn adc7
```

```
sp1.Write("8")
```

```
readSerial8 = sp1.ReadChar
```

Εδώ παρουσιάζονται οι τιμές της σειριακής αφιλτράριστες

```
txtB0.Text = readSerial1
```

```
txtB1.Text = readSerial2
```

```
txtB2.Text = readSerial3
```

```
txtB3.Text = readSerial4
```

```
txtB4.Text = readSerial5
```

```
txtB5.Text = readSerial6
```

```
txtB6.Text = readSerial7
```

```
txtB7.Text = readSerial8
```

Εδώ γίνεται η διαδικασία που αναφέρεται παραπάνω, δηλαδή η καταγραφή των τιμών της σειριακής για τη χρονική στιγμή k^{-1} και τη χρονική στιγμή k ή με πιο απλά λόγια για MOfilter=0 και MOfilter=1

```
If MOfilter = True Then
```

```
    Ttimi1 = readSerial1
```

```
    Ttimi2 = readSerial2
```

```
    Ttimi3 = readSerial3
```

```
    Ttimi4 = readSerial4
```

```
    Ttimi5 = readSerial5
```

```
    Ttimi6 = readSerial6
```

```
    Ttimi7 = readSerial7
```

```
    Ttimi8 = readSerial8
```

```
End If
```

```
If MOfilter = False Then
```

```
    Ftimi1 = readSerial1
```

```
    Ftimi2 = readSerial2
```

```
    Ftimi3 = readSerial3
```

```
    Ftimi4 = readSerial4
```

```
    Ftimi5 = readSerial5
```

```
Ftimi6 = readSerial6
Ftimi7 = readSerial7
Ftimi8 = readSerial8
```

```
End If
```

Εδώ υπολογίζεται η γωνία περιστροφής των κλειδώσεων

```
If RangleBit = True Then
```

```
TRangle1 = FreadSerial1
TRangle2 = FreadSerial2
TRangle3 = FreadSerial3
TRangle4 = FreadSerial4
TRangle5 = FreadSerial5
TRangle6 = FreadSerial6
TRangle7 = FreadSerial7
TRangle8 = FreadSerial8
```

```
End If
```

```
If RangleBit = False Then
```

```
FRangle1 = FreadSerial1
FRangle2 = FreadSerial2
FRangle3 = FreadSerial3
FRangle4 = FreadSerial4
FRangle5 = FreadSerial5
FRangle6 = FreadSerial6
FRangle7 = FreadSerial7
FRangle8 = FreadSerial8
```

Εδώ παρουσιάζονται οι περιστροφές αλλά οι τιμές Rangle είναι από τη προηγούμενη προσπέλαση του κώδικα στη Τρίτη ενότητα

```
lineYxeri.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineZxeri.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineYpalami.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)

lineYdeiktis2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineYdeiktis1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineYmesos2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineYmesos1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineYparamesos2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineYparamesos1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineYmikros1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineYmikros2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
```

```
lineZantixeiras1.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineZantixeiras2.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
```

```
thebratsoObj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
thepalamiObj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
theantixeiras1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
theantixeiras2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
thedeiktis1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
thedeiktis2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
themesos1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
themesos2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
theparamesos1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
theparamesos2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
themikros1Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
themikros2Obj.Rotate3D(lineYxeri.StartPoint, lineYxeri.EndPoint, Rangle1)
lineYpalami.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
```

```
lineYdeiktis2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
lineYdeiktis1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
lineYmesos2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
lineYmesos1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
lineYparamesos2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
lineYparamesos1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
lineYmikros1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
lineYmikros2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
lineZantixeiras1.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
lineZantixeiras2.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
```

```
thebratsoObj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
thepalamiObj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
theantixeiras1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
theantixeiras2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
thedeiktis1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
thedeiktis2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
themesos1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
themesos2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
theparamesos1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
theparamesos2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
themikros1Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
themikros2Obj.Rotate3D(lineZxeri.StartPoint, lineZxeri.EndPoint, Rangle6)
```



```

lineYpalami.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, Rangle3)

lineYdeiktis2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, Rangle3)
lineYdeiktis1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, Rangle3)
lineYmesos2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, Rangle3)
lineYmesos1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, Rangle3)
lineYparamesos2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
lineYparamesos1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
lineYmikros1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, Rangle3)
lineYmikros2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, Rangle3)
lineZantixeiras1.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
lineZantixeiras2.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
thepalamiObj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint, Rangle3)
theantixeiras1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
theantixeiras2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
thedeiktis1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
thedeiktis2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
themesos1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
themesos2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
theparamesos1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
theparamesos2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
themikros1Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)
themikros2Obj.Rotate3D(lineYpalami.StartPoint, lineYpalami.EndPoint,
Rangle3)

```

```
lineZantixeiras1.Rotate3D(lineZantixeiras1.StartPoint,
lineZantixeiras1.EndPoint, -Rangle4)
```

```
lineZantixeiras2.Rotate3D(lineZantixeiras1.StartPoint,
lineZantixeiras1.EndPoint, -Rangle4)
```

```
theantixeiras1Obj.Rotate3D(lineZantixeiras1.StartPoint,
lineZantixeiras1.EndPoint, -Rangle4)
```

```
theantixeiras2Obj.Rotate3D(lineZantixeiras1.StartPoint,
lineZantixeiras1.EndPoint, -Rangle4)
```

```
lineYdeiktis1.Rotate3D(lineYdeiktis1.StartPoint,           lineYdeiktis1.EndPoint,
Rangle5)
```

```
lineYdeiktis2.Rotate3D(lineYdeiktis1.StartPoint,           lineYdeiktis1.EndPoint,
Rangle5)
```

```
thedeiktis1Obj.Rotate3D(lineYdeiktis1.StartPoint,           lineYdeiktis1.EndPoint,
Rangle5)
```

```
thedeiktis2Obj.Rotate3D(lineYdeiktis1.StartPoint,           lineYdeiktis1.EndPoint,
Rangle5)
```

```
lineYmesos1.Rotate3D(lineYmesos1.StartPoint,               lineYmesos1.EndPoint,
Rangle2)
```

```
lineYmesos2.Rotate3D(lineYmesos1.StartPoint,               lineYmesos1.EndPoint,
Rangle2)
```

```
themesos1Obj.Rotate3D(lineYmesos1.StartPoint,               lineYmesos1.EndPoint,
Rangle2)
```

```
themesos2Obj.Rotate3D(lineYmesos1.StartPoint,               lineYmesos1.EndPoint,
Rangle2)
```

```
lineYparamesos1.Rotate3D(lineYparamesos1.StartPoint,
lineYparamesos1.EndPoint, Rangle7)
```

```
lineYparamesos2.Rotate3D(lineYparamesos1.StartPoint,
lineYparamesos1.EndPoint, Rangle7)
```

```
theparamesos1Obj.Rotate3D(lineYparamesos1.StartPoint,
lineYparamesos1.EndPoint, Rangle7)
```

```
theparamesos2Obj.Rotate3D(lineYparamesos1.StartPoint,
lineYparamesos1.EndPoint, Rangle7)
```

```
lineYmikros1.Rotate3D(lineYmikros1.StartPoint,             lineYmikros1.EndPoint,
Rangle8)
```

```

        lineYmikros2.Rotate3D(lineYmikros1.StartPoint,
Rangle8)
        themikros1Obj.Rotate3D(lineYmikros1.StartPoint,
Rangle8)
        themikros2Obj.Rotate3D(lineYmikros1.StartPoint,
Rangle8)

        'kano update
        acadapp.Update()

```

End If

Εδώ υπολογίζεται η τελική τιμή της γωνίας περιστροφής όπως βλέπουμε στο κώδικα βρίσκετε η διαφορά της γωνίας και μετατρέπετε σε ακτίνια.

```

'oi telikes times gia to rotate
Rangle1 = (TRangle1 - FRangle1) * 0.01745
Rangle2 = (TRangle2 - FRangle2) * 0.01745
Rangle3 = (TRangle3 - FRangle3) * 0.01745
Rangle4 = (TRangle4 - FRangle4) * 0.01745
Rangle5 = (TRangle5 - FRangle5) * 0.01745
Rangle6 = (TRangle6 - FRangle6) * 0.01745
Rangle7 = (TRangle7 - FRangle7) * 0.01745
Rangle8 = (TRangle8 - FRangle8) * 0.01745
'oi telikes times gia to rotate parousiazontai
txtbRangle1.Text = Rangle1
txtbRangle2.Text = Rangle2
txtbRangle3.Text = Rangle3
txtbRangle4.Text = Rangle4
txtbRangle5.Text = Rangle5
txtbRangle6.Text = Rangle6
txtbRangle7.Text = Rangle7
txtbRangle8.Text = Rangle8

```

End Sub

Κεφάλαιο 5

5.1) Ανάλυση του κώδικα της visual basic για το Σ.Σ.Δ

Πατώντας το κουμπί έναρξη το πρόγραμμα μας μεταφέρει σε μία νέα φόρμα. Ο κώδικας που ενεργοποιεί το κουμπί είναι

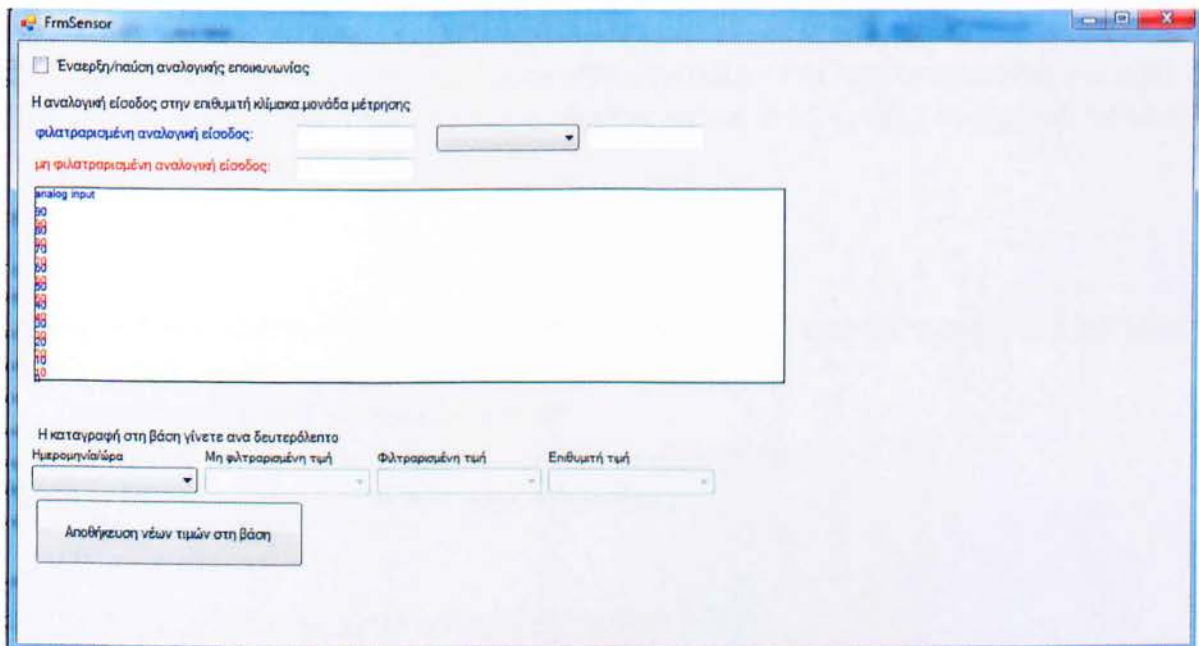
```
Private Sub btn_frm2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_frm2.Click
    FrmSensor.Show()
End Sub

End Class
```

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
Πτυχιακή εργασία του Καπαρέλη Δημήτριου
Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

Εδώ εύκολα καταλαβαίνουμε πως το κομμάτι κώδικα εμφανίζει την φόρμα του Συστήματος συλλογής δεδομένων.

Η νέα φόρμα που θα εμφανιστεί είναι



Ο κώδικας που θα τρέξει με το που ανοίξει η νέα φόρμα είναι
'TODO: This line of code loads data into the 'Analog_inputDataSet.analog_input' table.
You can move, or remove it, as needed.

```
Me.Analog_inputTableAdapter.Fill(Me.Analog_inputDataSet.analog_input)
```

Εδώ το αντικείμενο `Analog_inputTableAdapter` γεμίζει με τα δεδομένα από το αντικείμενο `Analog_inputDataSet` που έχει οριστεί από την αρχή να είναι τα αντικείμενα `comboboxes` που είναι στο τελευταίο μέρος της φόρμας.

```
excel.Visible = True
excel.Workbooks.Add()
excel.Columns.ColumnWidth = 30
```

```

excel.Range("A1").Select()
excel.ActiveCell.FormulaR1C1 = "Αναλογική είσοδος"
excel.Range("B1").Select()
excel.ActiveCell.FormulaR1C1 = "Φιλτραρισμένη είσοδος"
excel.Range("C1").Select()
excel.ActiveCell.FormulaR1C1 = "Επιθυμητή είσοδος"
excel.Range("D1").Select()
excel.ActiveCell.FormulaR1C1 = "Ημερωμηνία/ώρα"

```

Εδώ φτιάχνουμε ένα αντικείμενο excel το οποίο έχουμε λάβει από τη βιβλιοθήκη αυτοματοποίησης του excel. Το εμφανίζω, του προσθέτω ένα φύλλο εργασίας και κάνω κάθε κελί να χωρά μέχρι 30 χαρακτήρες. Έπειτα στο κελί A1 γράφω αναλογική είσοδος στο κελί B1 φιλτραρισμένη είσοδος κλπ.

Το παρακάτω κομμάτι κώδικα ζωγραφίζει τους άξονες χ,ψ που θα εμφανίζεται γραφικά η είσοδος από το αισθητήριο.

```

'zografizo th grafiki parastasi sto pctbx
pctBx.Image = New Bitmap(pctBx.Width, pctBx.Height)

```

Δηλώνω το αντικείμενο pctBx σαν αρχείο Bitmap
'gia na zografizo

```

bit = New Bitmap(pctBx.Width, pctBx.Height)
g = Graphics.FromImage(bit)

```

και με τη συνάρτηση graphics ζωγραφίζω.

```
pctBx.Image = bit
```

```
Pen = New Pen(Color.Black, 1)
```

Εδώ ορίζεται το αντικείμενο pen σαν ένα "μολύβι" το οποίο έχει χρώμα μαύρο και πάχος 1 πιχελ.

```
afont = New System.Drawing.Font("Arial", 7)
```

το αντικείμενο afont είναι ένα αντικείμενο το οποίο χρησιμοποιεί η συνάρτηση graphics για να γράφει γράμματα. Οπότε εδώ ορίζεται να έχουν γραμματοσειρά arial και μέγεθος 7.

```

'ena tetragono sto oporto 8a emfanizonai oi times
g.DrawRectangle(Pen, 2, 0, bit.Width - 3, bit.Height - 2)
'to asprizo

```

```
g.FillRectangle(Brushes.White, 3, 1, bit.Width - 4, bit.Height - 3)
```

Εδώ δημιουργείται, ένα τετράγωνο με το “μολύβι” pen αρχή τα pixels 2,0 και τέλος $\chi, \psi=3$ pixels πριν το τέλος του αντικειμένου bit που είναι η εικόνα που εμφανίζεται στο pctx.

Στην επόμενη εντολή φτιάχνω ένα αντικείμενο άσπρο αυτή τη φορά που είναι λίγο μικρότερο από το προηγούμενο. Έτσι δημιουργήθηκε ένα τετράγωνο μαύρο με άσπρο φόντο.

```
'ta logia ston axona Y
```

```
'gia filtrarismenes times
```

```
g.DrawString("0", afont, Brushes.Blue, 0, bit.Height - 8)
```

```
g.DrawString("10", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.9)
```

```
g.DrawString("20", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.8)
```

```
g.DrawString("30", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.7)
```

```
g.DrawString("40", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.6)
```

```
g.DrawString("50", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.5)
```

```
g.DrawString("60", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.4)
```

```
g.DrawString("70", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.3)
```

```
g.DrawString("80", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.2)
```

```
g.DrawString("90", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.1)
```

```
g.DrawString("analog input", afont, Brushes.Blue, 0, 0)
```

```
'gia mh filtrarismenes times
```

```
g.DrawString("0", afont, Brushes.Red, 0, 10 + (bit.Height - 8))
```

```
g.DrawString("10", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.9))
```

```
g.DrawString("20", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.8))
```

```
g.DrawString("30", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.7))
```

```
g.DrawString("40", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.6))
```

```
g.DrawString("50", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.5))
```

```
g.DrawString("60", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.4))
```

```
g.DrawString("70", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.3))
```

```
g.DrawString("80", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.2))
```

```
g.DrawString("90", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.1))
```

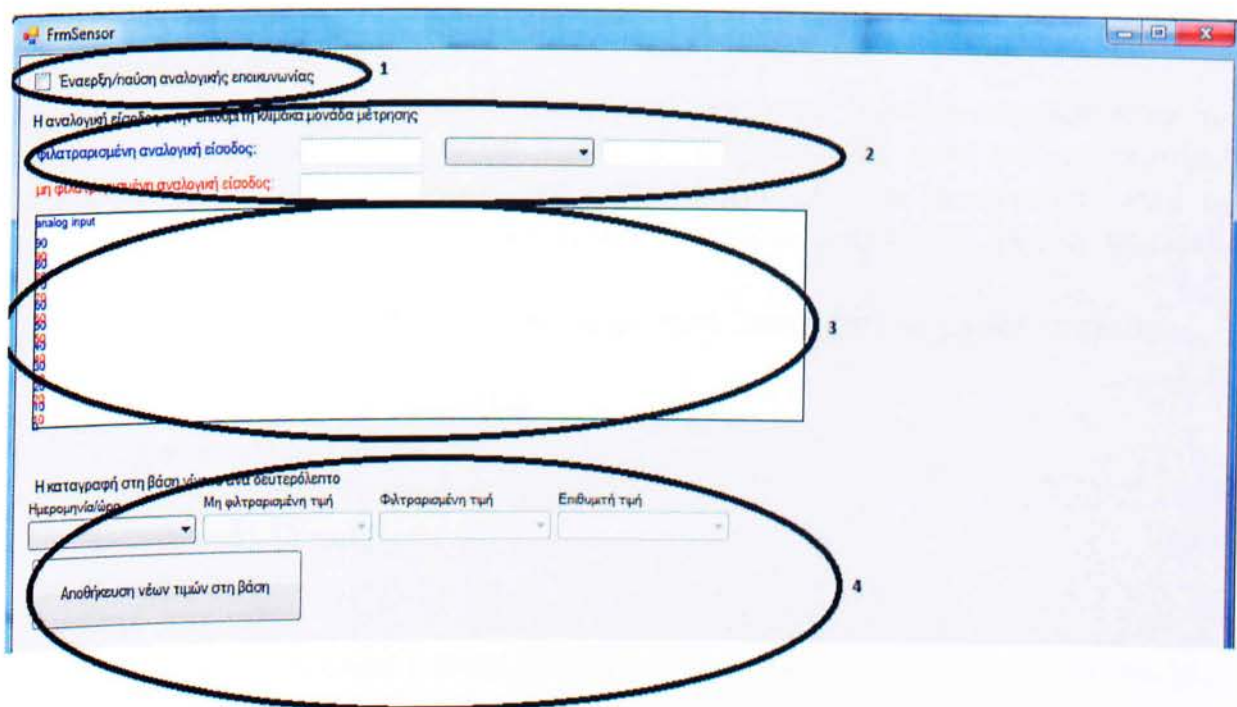
Στο παραπάνω κώδικα ζωγραφίζω στον άξονα Ψ τη κλίμακα. Το κάνω δύο φορές τη πρώτη με μπλέ γράμματα και τη δεύτερη με κόκκινα.

```
End Sub
```

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16

Πτυχιακή εργασία του Καπαρέλη Δημήτριου

Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά



Η νέα φόρμα έχει χωριστεί σε 4 μέρη.

5.2 Το πρώτο μέρος της φόρμας FrmSensor

Σε αυτό το μέρος έχουμε ένα αντικείμενο checkbox όπου όταν τον ενεργοποιούμε(το κλικάρουμε) ενεργοποιείται και ο κώδικας που περιέχει και όταν το απενεργοποιούμε σταματάει να τρέχει το κώδικα.

Ο κώδικας είναι:

```
Private Sub Ckx_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Ckx.CheckedChanged
```

```
If Ckx.Checked = True Then  
    autocad.sp1.Open()
```

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
Πτυχιακή εργασία του Καταρέλη Δημήτριου
Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά


```
Timer_frmSensor.Enabled = True
tmrSec.Enabled = True
```

Εδώ, εάν ενεργοποιηθεί το ckbx(το αντικείμενο checkbox δηλαδή) ενεργοποιείται το αντικείμενο sp1 της φόρμας autocad. Το αντικείμενο SP1 είναι αυτό που ενεργοποιεί την σειριακή θύρα του υπολογιστή και αφού οι ιδιότητές του αντικειμένου είναι οι επιθυμητές, δεν χρειάζεται να δημιουργηθεί καινούριο αντικείμενο για τη σειριακή επικοινωνία.

Επίσης ενεργοποιούνται και δύο χρονικά, το χρονικό Timer_frmSensor και το tmrSec

```
Elseif Ckbx.Checked = False Then
    tmrSec.Enabled = False
    Timer_frmSensor.Enabled = False
    autocad.sp1.Close()
```

```
End If
```

Σε αυτό το σημείο απενεργοποιούνται τα παραπάνω, όταν ο χρήστης απενεργοποιήσει το αντικείμενο ckbx.

```
End Sub
```

5.3) Το δεύτερο μέρος της φόρμας.

Το δεύτερο μέρος της φόρμας είναι καθαρά παρουσιαστικό και γι' αυτό το λόγο δεν έχει κώδικα. Όπως μπορούμε να διακρίνουμε, αποτελείται από κάποια αντικείμενα textbox και labels και έχει ένα combobox. Το combobox περιέχει τα στοιχεία που ο χρήστης επιλέγει να διαμορφώσει την αναλογική είσοδο. Τα στοιχεία αυτά είναι: 0-1023 analog input 0-5 volt 0-100 volt 10-50 °C

5.4 Το τρίτο μέρος της φόρμας

Στο τρίτο μέρος απεικονίζεται η αναλογική είσοδος σε συνάρτηση του χρόνου. Όπως αναφέρθηκε και στο κεφάλαιο 2.3.6 χρησιμοποιούμε ένα αντικείμενο picturebox το οποίο σχηματίζεται την ώρα που η φόρμα ανοίγει. Πως όμως σχηματίζεται η συνάρτηση καθώς διαβάζεται η αναλογική είσοδος και αναπαρησάται στο τρίτο μέρος; Με τη βοήθεια ενός ακόμη χρονικού το οποίο όταν τερματιστεί διαβάζει την σειριακή είσοδο τη σχηματίζει στο γράφιμα, την εμφανίζει στα κελιά του excel και στον πίνακα της access.

Ο κώδικας του χρονικού αυτού είναι ο παρακάτω:

```
autocad.sp1.Write("1")
str_serial_port_read = autocad.sp1.ReadChar
Εδώ στέλνουμε στη σειριακή είσοδο το χαρακτήρα1

'thn parousiazo sto txtbAnalog_input filtrarismenh
double_serial_port_read = str_serial_port_read
txtbAnalog_input.Text = (double_serial_port_read +
Previous_double_serial_potrt_read) / 2
txtbNfltrDbl.Text = Previous_double_serial_potrt_read
```

Previous_double_serial_potrt_read = double_serial_port_read
Εδώ είναι το κομμάτι του κώδικα όπου η σειριακή τιμή φιλτράρεται και εμφανίζεται στα textboxes της φόρμας στο κομμάτι2. Να αναφερθεί, πως το φίλτρο είναι ένα φίλτρο δευτέρου βαθμού μέσου όρου. Και η λογική του είναι η παρακάτω.

```
ΕΠΑΝΑΛΗΨΗ ΜΕ ΤΟ ΤΕΡΜΑΤΙΣΜΑ ΤΟΥ ΧΡΟΙΚΟΥ
ΣΕΙΡΙΑΚΗ ΤΙΜΗ=ΔΙΑΒΑΣΕ ΤΗ ΣΕΙΡΙΑΚΗ ΤΙΜΗ
ΦΙΛΤΡΑΡΙΣΜΕΝΗ ΤΙΜΗ =(ΣΕΙΡΙΑΚΗ ΤΙΜΗ-ΠΡΟΗΓΟΥΜΕΝΗ ΣΕΙΡΙΑΚΗ ΤΙΜΗ)/2
ΠΡΟΗΓΟΥΜΕΝΗ ΣΕΙΡΙΑΚΗ ΤΙΜΗ = ΣΕΙΡΙΑΚΗ ΤΙΜΗ
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
```

```
'edo parousiazo sto txtbChangeAnalog tis tropopoihmenew times
If cmbboc_change_analog_input.SelectedIndex = 0 Then
    txtbChangeAnalog.Text = txtbAnalog_input.Text * 11.76 & " analog input"
```

```
Elsif cmbboc_change_analog_input.SelectedIndex = 1 Then
```

```
txtbChangeAnalog.Text = txtbAnalog_input.Text * 0.057 & " volt"
```

```
Elseif cmbboc_change_analog_input.SelectedIndex = 2 Then
  txtbChangeAnalog.Text = txtbAnalog_input.Text * 1.15 & " volt"
```

```
Elseif cmbboc_change_analog_input.SelectedIndex = 3 Then
  txtbChangeAnalog.Text = 10 + txtbAnalog_input.Text * 0.45 & " °C"
```

```
Else
  txtbChangeAnalog.Text = "Επιλέξτε επιθυμητή κλίμακα"
```

```
End If
```

Το παραπάνω κομμάτι του κώδικα εμφανίζει την σειριακή είσοδο στα "μέτρα" του χρήστη. Αφού έχει επιλεγθεί από το combobox του δεύτερου κομματιού της φόρμας σε τι θέλει να απεικονίζει την είσοδο. Αυτή πολλαπλασιάζεται/προστίθεται με το κατάλληλο αριθμό, ώστε να έχει το κατάλληλο εύρος. Δίπλα παρουσιάζεται το φυσικό μέγεθος και εμφανίζεται στο textbox.

```
pctBx.Image = bit
```

```
Pen.Color = Color.Red
If c <= bit.Height And c > 0 Then
  g.DrawLine(Pen, x, c, x + 1, c + 1)
End If
```

```
Pen.Color = Color.Blue
```

```
If y <= bit.Height And y > 0 Then
  g.DrawLine(Pen, x, y, x + 1, y + 1)
End If
```

```
x = x + 1
y = bit.Height - (((double_serial_port_read + Previous_double_serial_potrt_read) /
2) / 100) * bit.Height)
c = bit.Height - (double_serial_port_read / 100 * bit.Height) + 10
```

Σε αυτό το σημείο, παρουσιάζεται το κομμάτι, που σχηματίζεται η γραφική παράσταση. Κάθε φορά σχηματίζεται ένα σημείο όπου το Ψ του είναι εν συνάρτηση της εισόδου ενώ το X του είναι μία αύξουσα μεταβλητή. Να αναφερθεί επίσης, πως σχηματίζονται δύο γραφήματα για τη φιλτραρισμένη και για τη μη φιλτραρισμένη τιμή.

If $x = \text{bit.Width}$ Then

```
g.FillRectangle(Brushes.White, 3, 1, bit.Width - 4, bit.Height - 3)
```

```
g.DrawString("0", afont, Brushes.Blue, 0, bit.Height - 8)
```

```
g.DrawString("10", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.9)
```

```
g.DrawString("20", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.8)
```

```
g.DrawString("30", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.7)
```

```
g.DrawString("40", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.6)
```

```
g.DrawString("50", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.5)
```

```
g.DrawString("60", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.4)
```

```
g.DrawString("70", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.3)
```

```
g.DrawString("80", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.2)
```

```
g.DrawString("90", afont, Brushes.Blue, 0, (bit.Height - 8) * 0.1)
```

```
g.DrawString("analog input", afont, Brushes.Blue, 0, 0)
```

```
'gia mh filtrarismenes times
```

```
g.DrawString("0", afont, Brushes.Red, 0, 10 + (bit.Height - 8))
```

```
g.DrawString("10", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.9))
```

```
g.DrawString("20", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.8))
```

```
g.DrawString("30", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.7))
```

```
g.DrawString("40", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.6))
```

```
g.DrawString("50", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.5))
```

```
g.DrawString("60", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.4))
```

```
g.DrawString("70", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.3))
```

```
g.DrawString("80", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.2))
```

```
g.DrawString("90", afont, Brushes.Red, 0, 10 + ((bit.Height - 8) * 0.1))
```

$x = 0$

Το παραπάνω κομμάτι κώδικα σχηματίζει ξανά όλο το γράφημα από την αρχή σε περίπτωση που οι γραφικές φτάσουν στο τέλος του picturebox. Είναι ακριβώς το ίδιο κομμάτι κώδικα που χρησιμοποιήθηκε με το που ανοίγει η φόρμα. Επίσης η μεταβλητή που χρησιμοποιείται για να δηλωθεί στο σημείο x του σημείου μηδενίζεται. Έτσι το γράφημα ξεκινά πάλι από την αρχή.

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
Πτυχιακή εργασία του Καπαρέλη Δημήτριου
Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

```
End If
'excel
If excel.Visible = True Then
```

```
    excel.Range("A" & Yrange).Select()
    excel.ActiveCell.FormulaR1C1 = txtbNfltrDbl.Text
```

```
    excel.Range("B" & Yrange).Select()
    excel.ActiveCell.FormulaR1C1 = txtbAnalog_input.Text
```

```
    excel.Range("C" & Yrange).Select()
    excel.ActiveCell.FormulaR1C1 = txtbChangeAnalog.Text
```

```
    excel.Range("D" & Yrange).Select()
    excel.ActiveCell.FormulaR1C1 = now_date
```

```
    Yrange = Yrange + 1
    'acces
```

```
End If.
```

Έδω οι τιμές στέλνονται οι τιμές στο excel. Χρησιμοποιούμε μία μεταβλητή την Yrange που κάθε φορά αυξάνεται για κάθε επανάληψη του κώδικα να γράφεται η τιμή ακριβώς στο παρακάτω κελί η τιμή. Όπως μπορούμε να παρατηρήσουμε στη στήλη Ai[i=1..] του excel αποθηκεύονται οι σειριακές τιμές, στη στήλη Bi[i=1..] οι φιλτραρισμένες, στη στήλη Ci[i=1..] οι τιμές που έχει επιλέξει ο χρήστης και τέλος στη στήλη Di[i=1..] η ημερομηνία που έχουμε κάθε στιγμή.

Μία ακόμη απορία που μπορεί να έχει κάποιος μέχρι εδώ είναι, πως εμφανίζεται ο χρόνος στον άξονα χ(1sec,2sec,3sec)κ.λ.π..Το πως θα γίνει αυτό με προβλημάτισε

Αυτοματισμός του AutoCAD με τη visual basic και τον at mega16
 Πτυχιακή εργασία του Καπαρέλη Δημήτριου
 Τμήμα Αυτοματισμού Τ.Ε.Ι. Πειραιά

πάρα πολύ αρχικά, μία ιδέα ήταν να μετρήσω τους κύκλους μηχανής που εκτελεί ο μικροελεγκτής και ανάλογα τη ταχύτητά του να υπολόγισω το χρόνο αλλά αν το έκανα αυτό θα έπρεπε να υπολόγισω και τους κύκλους μηχανής του προγράμματος της visual basic κάτι που είναι πάρα πολύ δύσκολο. Έτσι έκανα κάτι άλλο πιο εύκολο. Δημιούργησα ένα ακόμη χρονικό το οποίο τερματίζεται κάθε ένα δευτερόλεπτο και όταν τερματίζεται θα έγραφε το πρόγραμμα στο γράφημα 1sec,2sec κ.λ.π. Έτσι τα πράγματα έγιναν πολύ πιο απλά αλλά επίσης αυτό το χρονικό το χρησιμοποίησα και για να αποθηκεύσω τις τιμές στη βάση δεδομένων μου. Άρα οι τιμές στη βάση αποθηκεύονται ανά δευτερόλεπτο δεν αποθηκεύονται όλες όπως στο excel. Το χρονικό αυτό ονομάστηκε timersec και ο κώδικάς του είναι ο παρακάτω:

```
Private Sub tmrSec_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tmrSec.Tick
```

```
    pctBx.Image = bit
    g.DrawString(CStr(a) & " sec", afont, Brushes.Black, x, (bit.Height - 10))
    a = a + 1
```

Το παραπάνω κομμάτι κώδικα είναι αυτό που γράφει 1sec,2sec κλπ. Έχει χρησιμοποιηθεί μία αύξουσα μεταβλητή που καταγράφει τα δευτερόλεπτα(η μεταβλητή a) και αποικονίζεται στο γράφημα με το sec δίπλα της.[a sec ,a=1,2...]

```
Dim new_row As DataRow = Me.Analog_inputDataSet.analog_input.NewRow
new_row("Αναγνωριστικό") = Analog_inputDataSet.analog_input.Count + 1
new_row("date") = now_date
```

```
new_row("analog inputs") = txtbNftrDbl.Text
new_row("filter analog inputs") = txtbAnalog_input.Text
new_row("values") = txtbChangeAnalog.Text
```

```
Analog_inputDataSet.analog_input.Rows.Add(new_row)
```

```
End Sub
```

Το παραπάνω κομμάτι του κώδικα είναι αυτό που κάθε φορά δημιουργεί μία νέα σειρά στο αντικείμενο Analog_inputDataSet και μεταφέρει τις τιμές ημερομηνία σειριακή τιμή, φιλτραρισμένη τιμή κλπ.

5.5 Το τέταρτο κομμάτι της φόρμας

Το τέταρτο κομμάτι του κώδικα είναι ένα μόνο κουμπί που θα περάσει τα δεδομένα του αντικειμένου Analog_inputDataSet στη βάση, θα τα αποθηκεύσει δηλαδή.

Ο κώδικας είναι:

```
Private Sub Btnsavedatas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSavedatas.Click
```

```
    Analog_inputTableAdapter.Update(Analog_inputDataSet.analog_input)
```

```
End Sub
```

Αυτό που κάνει είναι να ανανεώνει τα δεδομένα του πίνακα Analog_inputTableAdapter με τα δεδομένα του Analog_inputDataSet.

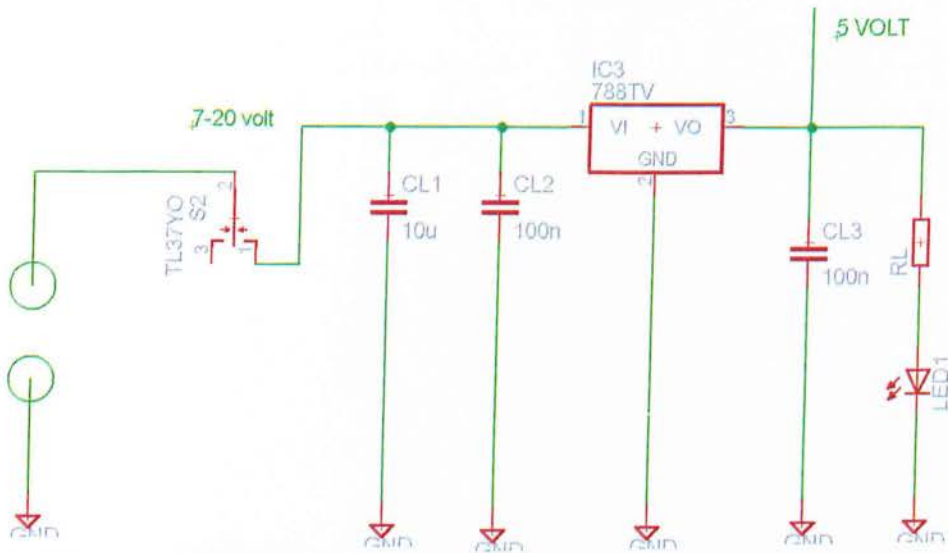
ΚΕΦΑΛΑΙΟ 6

Βιβλιογραφία

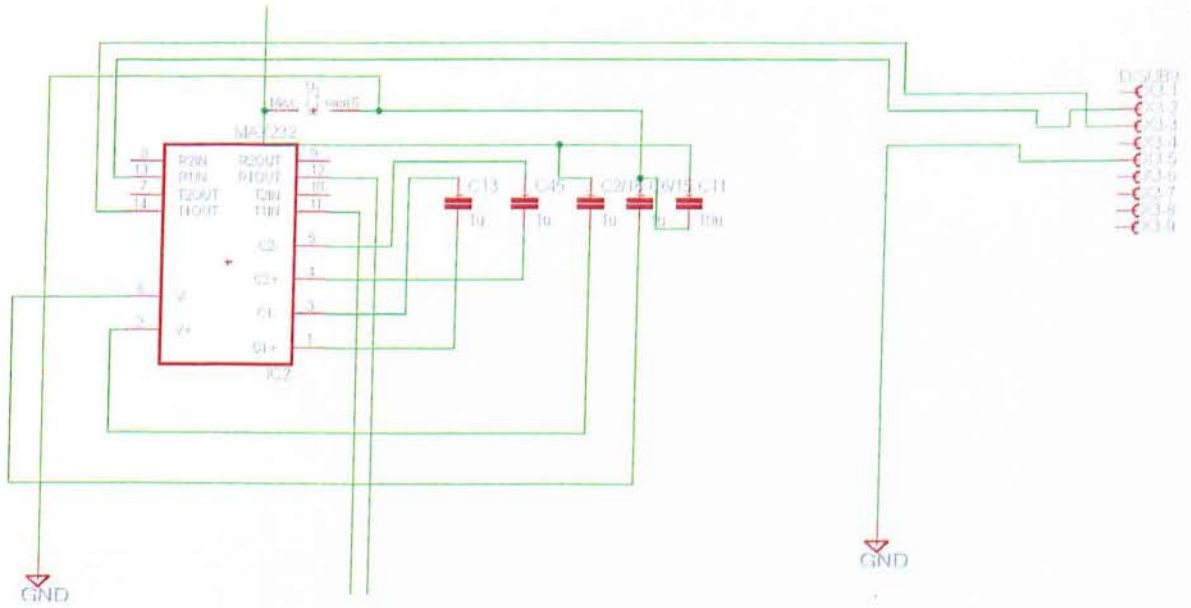
- Η θεωρία για τη λειτουργία του AT mega16 είναι από το βιβλίο “Μικροελεγκτές AVR και PIC” του Κ.Πεκμετζή
- Η θεωρία για το ολοκληρωμένο MAX232 είναι από το datasheet του κατασκευαστή
- Η θεωρία για το ολοκληρωμένο LM7805 είναι από τον κατασκευαστή
- Η θεωρία για τα αντικείμενα της σειρικής επικοινωνίας, του χρονικού και της βιβλιοθήκης τύπων της visual basic είναι από το βιβλίο “Μάθετε την visual basic 2005 σε 24 ώρες” του James Folax καθώς και από τις ιστοσελίδες της wikipedia και της msdn.microsoft
- Ο τρόπος κατασκευής πλακετών PCB είναι από την ιστοσείδα <http://www.hlektronika.gr/forum/downloads.php?do=file&id=25>
- Το κύκλωμα του μικροελεγκτή είναι από το βιβλίο των Steven F.Barrett και Daniel J. Pack “ATMEL AVR MICROCONTROLLER:PROGRAMMING AND INTERFACING”
- Το κύκλωμα του MAX232 είναι από το datasheet του κατασκευαστή
- Το κύκλωμα του LM 7805 είναι από το datasheet του κατασκευαστή
- Οι γνώσεις που πήρα για το προγραμματισμό του AT mega16 είναι από τα βιβλία “ATMEL AVR MICROCONTROLLER:PROGRAMMING AND INTERFACING” και “Μικροελεγκτές AVR και PIC” και από τις ιστοσελίδες avrfreaks.net, hlektronika.gr και <http://extremeelectronics.co.in/>
- Για τον προγραμματισμό των απλών αντικειμένων της visual basic χρησιμοποίησα βιβλίο “Μάθετε την visual basic 2005 σε 24 ώρες” καθώς και από το μάθημα Προγραμματισμός 3 τμήματος Αυτοματισμού Τ.Ε.Ι. Πειραιά του Κ. Ψαρομήλικου.
- Οι πληροφορίες και οι εντολές για το αντικείμενο της σειριακής επικοινωνίας της visual basic είναι από την ιστοσελίδα http://www.serialporttool.com/NCD/ProXR_VB2008.htm
- Οι πληροφορίες για τη βιβλιοθήκη τύπων του Autocad είναι από τις ιστοσελίδες: <http://docs.autodesk.com/ACD/2010/ENU/AutoCAD%20.NET%20Developer%27s%20Guide/index.html>
http://www.kxcad.net/autodesk/autocad/Autodesk_AutoCAD_ActiveX_and_VBA_Developer_Guide/index.htm

ΠΑΡΑΡΤΗΜΑ Α

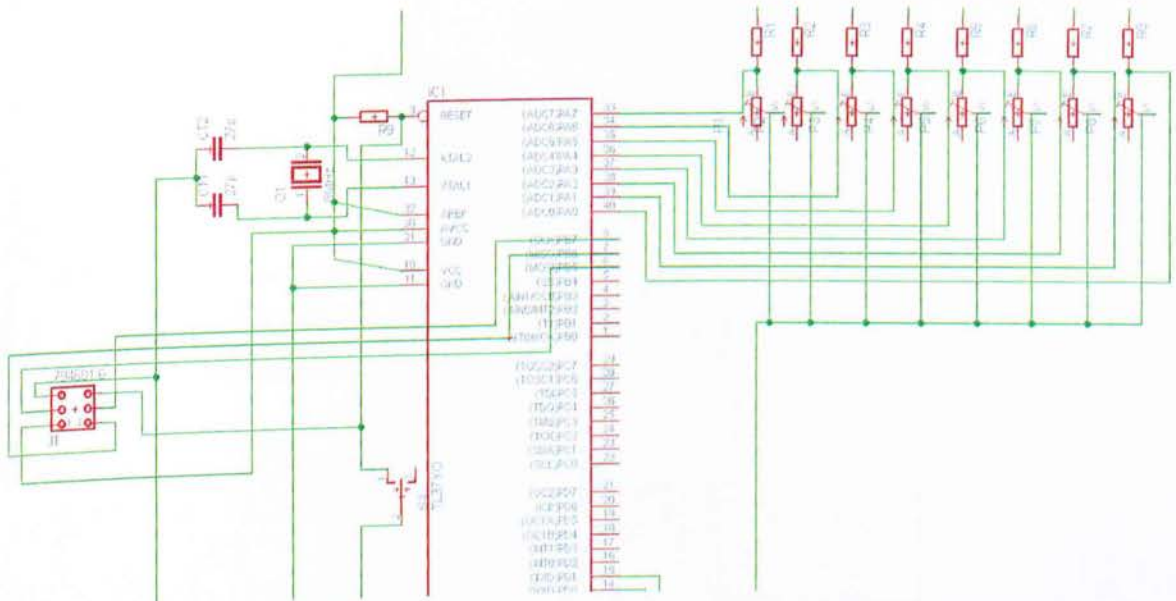
Οι κυκλώματα της πτυχιακής εργασίας



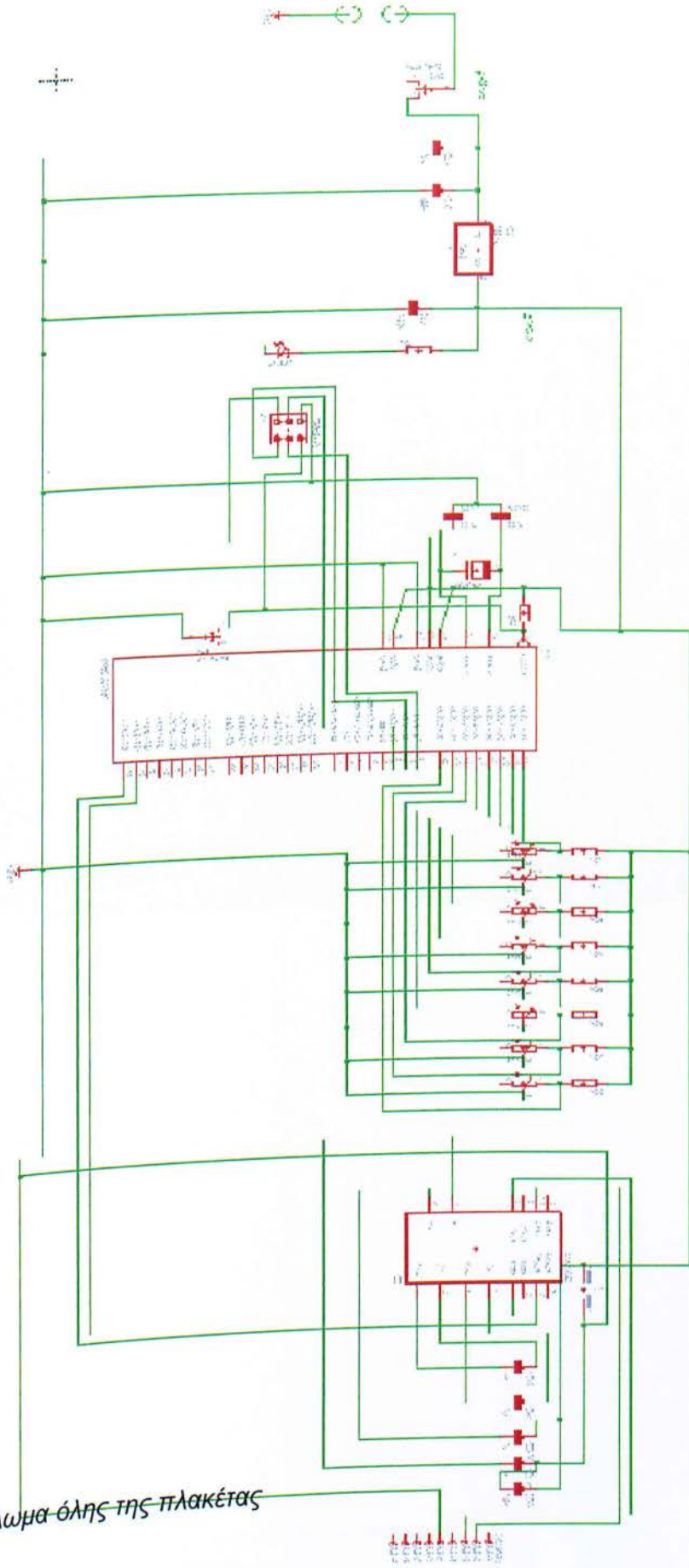
Το κύκλωμα του σταθεροποιητή LM7805



Το κύκλωμα του MAX 232

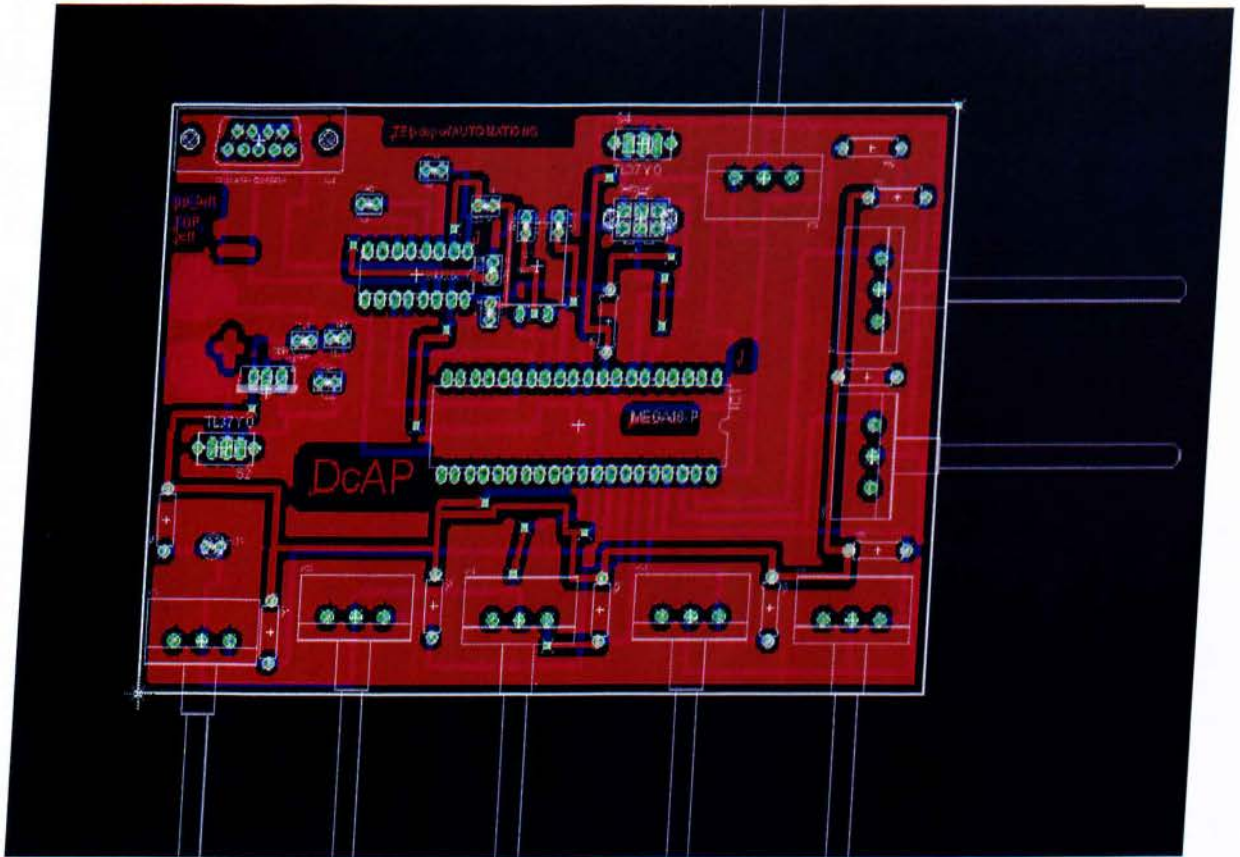


Το κύκλωμα του Atmega16



Το κύκλωμα όλης της πλακέτας

1
1
1



Το τυπωμένο κύκλωμα