



**ΤΕΙ ΚΡΗΤΗΣ – ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ**

# **Σχεδίαση και υλοποίηση ολοκληρωμένης διεπαφής για τον έλεγχο βηματικών κινητήρων μέσω της θύρας USB ενός Η/Υ.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**



**Γιαννακόπουλου Νικήτα**

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

**Χανιά,  
Σεπτέμβρης 2008**

## Περίληψη

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η κατασκευή μιας περιφερειακής συσκευής USB. Η συσκευή αυτή κατά κύριο λόγο θα επιτρέψει τον έλεγχο της θέσης δυο βηματικών κινητήρων. Πρέπει να αναφερθεί ότι η κατασκευή είναι πλήρως συμβατή με τις αρχές λειτουργίας του πρωτόκολλου USB1.1.

Η εργασία μπορεί να διαιρεθεί σε τρία βασικά μέρη. Το πρώτο μέρος έχει να κάνει με τη δημιουργία ενός αυτόνομου προγράμματος σε μια γλώσσα υψηλού επιπέδου προγραμματισμού όπως την Visual Basic 6. Το δεύτερο μέρος περιλαμβάνει την κατανόηση στοιχειωδών εννοιών γύρω από το πρωτόκολλο USB 1.1 και η δημιουργία ενός αρχείου .asm σε γλώσσα χαμηλού επιπέδου κατάλληλο για τον προγραμματισμό του μικροελεγκτή USB cy7c63001 της CYPRESS. Το τρίτο μέρος αποτελείται από την σχεδίαση και την κατασκευή του υλικού που περιλαμβάνει τρεις πλακέτες, κατάλληλα συνδεδεμένες μεταξύ τους, οι οποίες αναλαμβάνουν τον έλεγχο της κίνησης των βηματικών κινητήρων καθώς και την ένδειξη της θερμοκρασίας του περιβάλλοντος χώρου.

## Abstract

The object of the present thesis is the construction of a USB interface device. The device actually will be used to control the position of two stepper motors. To be mentioned that the whole project has been constructed under the features of the Universal Serial Bus protocol 1.1. The project consists of three main units. The first one includes the development of a stepper motor controller application using a high level programming language like Visual Basic 6. The second one involves the understanding of various things about USB protocol and writing firmware for the CYPRESS cy7c63001 USB microcontroller. The third one covers the construction of the hardware, including three PCB boards, to control the movement of the stepper motors, and also to capture the environment temperature.

## 1. Εισαγωγή

Σήμερα γινόμαστε μάρτυρες μιας εκπληκτικής τεχνολογικής επανάστασης που έχει σαν κέντρο της τη μετατόπιση της τεχνολογίας από τα αναλογικά συστήματα στα ψηφιακά. Οι ηλεκτρονικοί υπολογιστές έχουν πλέον την δικιά τους θέση στην

καθημερινότητα του καθενός από εμάς. Πλέον μέσα από εφαρμογές και προγράμματα έχουμε την δυνατότητα να εργαστούμε να ενημερωθούμε να επικοινωνήσουμε, να διασκεδάσουμε και όλα αυτά μέσα από τη χρήση του δικού μας ηλεκτρονικού υπολογιστή. Ο θαυμαστός κόσμος της βιομηχανίας πλέον αποτελείται από έξυπνα συστήματα που ελέγχονται από ένα ή περισσότερους ηλεκτρονικούς υπολογιστές.

Το γεγονός ότι μέσα από ένα ηλεκτρονικό υπολογιστή έχει κάποιος την δυνατότητα να ελέγχει τόσα πολλά πράγματα στο περιβάλλοντα χώρο με ώθησε στο να επιλέξω την παρούσα πτυχιακή εργασία, η οποία αναφέρεται στον έλεγχο μιας κατασκευής μέσω ενός προγράμματος γραμμένου σε γλώσσα υψηλού επιπέδου προγραμματισμού. Η επικοινωνία μεταξύ αυτών επιτυγχάνεται μέσω μιας USB θύρας ενός Η/Υ.

### Τα μέρη της εφαρμογής USB

Τα μέρη που εμπλέκονται σε μια τέτοια κατασκευή είναι:

- Ένας μικροελεγκτής USB και ότι αυτός ελέγχει, ο οποίος θα πρέπει να είναι κατάλληλα προγραμματισμένος (firmware) για τις ανάγκες της εκάστοτε εφαρμογής.
- Το λειτουργικό σύστημα του ηλεκτρονικού υπολογιστή (ξενιστής) που πρέπει να περιέχει τους κατάλληλους οδηγούς για τη συσκευή (device drivers) και τα κατάλληλα αρχεία INF. Τα αρχεία INF συσχετίζουν τη συσκευή με τους οδηγούς της κατά την διαδικασία της απαρίθμησης.
- Η εφαρμογή η οποία ελέγχει τη συσκευή σε επίπεδο προγράμματος (application program). Στα πλαίσια αυτής της πτυχιακής η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Visual Basic 6.

Η παρούσα πτυχιακή εργασία εκτείνεται σε επτά κεφάλαια και δύο παραρτήματα. Εκτός από το παρόν κεφάλαιο που αποτελεί μια σύντομη εισαγωγή, το δεύτερο κεφάλαιο αναφέρεται σε βασικές αρχές της θύρας USB. Γίνεται αναφορά σε βασικά χαρακτηριστικά της θύρας, στις διάφορες κλάσεις των συσκευών USB στους υποστηριζόμενους τρόπους μετάδοσης δεδομένων καθώς επίσης και στα μέρη μιας μετάδοσης μέσω της θύρας USB. Αναφέρονται τα βήματα απαρίθμησης μιας συσκευής USB και γίνεται μια περιγραφή στην αρχιτεκτονική του μικροελεγκτή cy7c63001 ο όποιος και χρησιμοποιείται στην παρούσα πτυχιακή.

Στο κεφάλαιο 3 γίνεται μία πρώτη γνωριμία με τους βηματικούς κινητήρες, τα είδη που υπάρχουν, τις μορφές λειτουργίας τους, την οδήγησή τους και τις χαρακτηριστικές τους λειτουργίες. Επίσης αναφέρονται τα πλεονεκτήματα και τα μειονεκτήματα που προκύπτουν από την χρησιμοποίησή τους.

Το κεφάλαιο 4 περιλαμβάνει μια περιληπτική αναφορά στο περιβάλλον εργασίας της Visual Basic 6. Γίνεται αναφορά στα σημαντικότερα στοιχεία ελέγχου που χρησιμοποιήθηκαν στην παρούσα εφαρμογή και αναφέρονται ορισμένα σημεία που αφορούν στα προγράμματα εφαρμογών με τα οποία μπορεί κανείς να διαχειριστεί την είσοδο και την έξοδο δεδομένων προς τις συσκευές HID, μέσω των συναρτήσεων API.

Το κεφάλαιο 5 αναφέρεται στο υλικό μέρος της κατασκευής. Συγκεκριμένα παρουσιάζονται οι κυκλωματικές διατάξεις οδήγησης των βηματικών κινητήρων οι οποίες αποτελούνται από κυκλώματα ψηφιακής λογικής (translator circuits), και κυκλώματα που παρέχουν ικανοποιητικό ρεύμα στις περιελίξεις των βηματικών

κινητήρων για την κίνηση τους (driver circuits). Επίσης γίνεται αναφορά στην πλακέτα διασύνδεσης με τον Η/Υ μέσω της θύρας USB (USB controller).

Το κεφάλαιο 6 είναι το εγχειρίδιο χρήσης της εφαρμογής. Δίδεται η ευκαιρία στον αναγνώστη να δει την συνολική εφαρμογή στη γλώσσα προγραμματισμού Visual Basic 6. Παρουσιάζονται τα στοιχεία ελέγχου που χρησιμοποιήθηκαν και γίνεται πλήρη αναφορά για κάθε ένα από αυτά.

Στο κεφάλαιο 7 αναφέρονται κάποια συμπεράσματα σχετικά με την παρούσα πτυχιακή εργασία, καθώς επίσης παρουσιάζονται κάποιες σκέψεις για μελλοντική επέκταση των λειτουργιών σε εφαρμογές μετρήσεων και ελέγχου.

Τέλος παρατίθεται βιβλιογραφία που περιέχει τίτλους βιβλίων, περιοδικά και ιστοσελίδες, όπως επίσης και δύο παραρτήματα τα οποία αναφέρονται στις συσκευές διασύνδεσης χειριστή (HIDs) και στους οδηγούς της συσκευής (device drivers) αντίστοιχα.

Η πτυχιακή εργασία συνοδεύεται από CD-ROM, που περιέχει το πρόγραμμα της εφαρμογής, το firmware του μικροελεγκτή και άλλο χρήσιμο υλικό.

Στο σημείο αυτό έχω την ανάγκη να εκφράσω τις ευχαριστίες μου σε κάποια άτομα που με βοήθησαν ο καθένας με το δικό του τρόπο να υλοποιήσω την παρούσα πτυχιακή.

Θα ήθελα να ευχαριστήσω

Τον κύριο Πετράκη Νικόλαο  
Καθηγητή Εφαρμογών του τμήματος Ηλεκτρονικής  
Για την βοήθεια και τις συμβουλές του

Τους φίλους και συναδέλφους  
Μεσαρχάκη Γεώργιο και Κασσιμάτη Κωνσταντίνο  
Για την βοήθεια τους στη Visual Basic 6

Την φίλη  
Καία Βασιλική  
για την βοήθεια της στην υλοποίηση του αρχείου Word

Τον φίλο και συνάδελφο  
Σκορδύλη Νικόλαο  
Που μέσα από τις συζητήσεις έδινε “τροφή για σκέψη”  
πάνω σε θέματα της πτυχιακής

Η παρούσα πτυχιακή εργασία αφιερώνεται στην οικογένεια μου  
Νικήτας



## 2. Η θύρα USB

### 2.1. Ο δρόμος προς τη USB

Όποιος αποκτά μια εξοικείωση με την επικοινωνία μέσω της παράλληλης και της σειριακής θύρας θεωρεί ότι είναι φυσικό κάποια στιγμή να μεταβεί στη χρήση της θύρας USB. Δύο τέτοιες θύρες τουλάχιστον βρίσκονται στο πίσω μέρος κάθε υπολογιστή ενώ όλα τα περιφερειακά κατασκευάζονται πλέον με την δυνατότητα να συνδέονται τουλάχιστον στην USB. Είναι φανερό ότι η αγορά έχει αποδεχτεί πλήρως την νέα αυτή θύρα επικοινωνίας που καταφανώς υπερέχει σε ευελιξία και σε τεχνικά χαρακτηριστικά όπως είναι η ταχύτητα μεταφοράς δεδομένων, ο μέγιστος αριθμός των υποστηριζόμενων περιφερειακών η διαχείριση της ηλεκτρονικής ενέργειας, και αλλά.

Όμως ο μέσος μηχανικός, η ερασιτέχνης που αποφασίζει να εμβαθύνει στις λεπτομέρειες που θα τον οδηγήσουν να στέλνει και να λαμβάνει δεδομένα μέσω της θύρας USB προκειμένου να ελέγξει περιφερειακά δικής του κατασκευής γρήγορα θα βρεθεί μπροστά σε μια αφιλόξενη πραγματικότητα: η διαδικασία επικοινωνίας μέσω USB δεν έχει καμία σχέση με τα απλά πρωτοκόλλα που συναντήσαμε στην επικοινωνία με την σειριακή ή την παράλληλη θύρα. Οι μέρες που διαχειριζόμασταν την επικοινωνία από και προς το εξωτερικό περιβάλλον απλώς γράφοντας και διαβάζοντας γνωστές θέσεις μνήμης και καταχωρητές χρησιμοποιώντας εντολές αρχείων η απλές εντολές INP και OUT έχουν πλέον τελειώσει. Ο υποψήφιος χρήστης της USB που ανυπομονεί να μάθει ένα σύνολο θέσεων μνήμης και IRQs προκειμένου να μεταβάλλει τη λογική κατάσταση κάποιων ακροδεκτών μέσω απλών ψηφιακών κυκλωμάτων σκέφτεται ήδη με λάθος τρόπο.

Ένας από τους λόγους που οδήγησαν στο νέο πρότυπο ήταν η κατάργηση ακριβώς αυτής της φιλοσοφίας η οποία αν και διευκολύνει το μέσο σχεδιαστή συσκοτίζει τον μέσο χρήστη και περιορίζει την εύκολη επέκταση ενός υπολογιστικού συστήματος.

Έτσι λοιπόν ας θέσουμε τους ορούς από την αρχή. Η επικοινωνία με τη USB έχει καταρχήν κάποιες απαιτήσεις από τον υπολογιστή και από το λειτουργικό σύστημα, ο υπολογιστής πρέπει να ενσωματώνει το κατάλληλο ελεγκτή USB ο οποίος καταλήγει σε μια οι περισσότερες θύρες USB. Αν ένας υπολογιστής δεν διαθέτει ελεγκτή USB μπορεί κάποιος να τον προσθέσει μέσω κάρτας επέκτασης που συνδέεται στον δίαυλο PCI. Όσο αναφορά στο λειτουργικό σύστημα θα πρέπει να ξεκαθαρίσουμε ότι το dos και τα windows 3.x. δεν περιλαμβάνουν υποστήριξη για USB, εκτός αν προμηθευτεί κάποιος σχετικό λογισμικό με την μορφή οδηγών (drivers) από τρίτους κατασκευαστές. Τα Windows 95 (OEM SERVICE RELEASE 2.1) περιλαμβάνουν στοιχειώδη μόνον υποστήριξη για τη θύρα USB, ενώ η μετέπειτα εκδόσεις των Windows από τα 98 μέχρι και τα Vista υποστηρίζουν κανονικά τη USB. Πάντως τα Windows 95 δεν χρησιμοποιούν τους ίδιους οδηγούς (drivers) με τα Windows 98 ενώ θα πρέπει να σημειωθεί ότι τα Windows NT μπορούν να υποστηρίξουν τη USB μονό με τη βοήθεια ειδικού λογισμικού που προστίθεται εκ των ύστερων.

Εκτός από τις παραπάνω προϋποθέσεις από πλευράς υλικού και λογισμικού θα πρέπει να καταλάβει κανείς ότι η επικοινωνία με τη USB είναι μια περίπλοκη υπόθεση διότι περιλαμβάνει εξαιρετικά περίπλοκα πρωτόκολλα. Έτσι η περιφερειακή διάταξη που συνδέεται με τη USB οφείλει να καταλαβαίνει και να μπορεί να αποκρίνεται στα σήματα επικοινωνίας που απευθύνει ο ελεγκτής από την πλευρά του υπολογιστή (host controller) συνεπώς δεν μπορεί να αποτελείται από απλά κυκλώματα εισόδου/εξόδου δηλαδή διακόπτες ηλεκτρονόμους και μετατροπείς αναλογικούς σήματος σε ψηφιακό όπως συνέβαινε στις εφαρμογές με την RS232 και την παράλληλη. Αναγκαστικά πλέον το περιφερειακό θα πρέπει να είναι ένα κύκλωμα με υπολογιστική νοημοσύνη δηλαδή ένας

μικροελεγκτής ικανός να συνδεθεί με τον δίαυλο USB. Ο μικροελεγκτής θα μπορεί να προγραμματιστεί και να αποθηκεύσει πληροφορίες ώστε να ανταποκρίνεται στις απαιτήσεις επικοινωνίας του ελεγκτή USB. Ο ίδιος ο μικροελεγκτής θα πρέπει να προγραμματιστεί κατάλληλα ώστε να χειρίζεται αποτελεσματικά το εξωτερικό περιβάλλον ανάλογα με τα δεδομένα που λαμβάνει από τον υπολογιστή μέσω του διαύλου USB.

Ακόμη και όταν ένας μηχανικός λύσει το πρόβλημα του προγραμματισμού του κατάλληλου μικροελεγκτή – κάτι που βεβαίως απαιτεί εργαλεία και γνώση απομένει το πρόβλημα πως θα επικοινωνήσει με το περιφερειακό σε κάποια ανώτερη γλώσσα προγραμματισμού. Η χρήση της απλής γλώσσας Basic πρέπει να αποκλειστεί αφού δεν μας παρέχει κανένα μηχανισμό επικοινωνίας με το λειτουργικό σύστημα Windows που υποστηρίζει τη USB. Ο μόνος τρόπος για να επικοινωνήσει ένα πρόγραμμα με κάποιο περιφερειακό USB είναι μέσω ένα οδηγού (device driver) αυτό είναι λογισμικό που ενεργεί σαν ενδιάμεση βαθμίδα ανάμεσα σε μια εφαρμογή software application και στην κυκλωματική διάταξη την οποία ελέγχει η εφαρμογή αυτή.

Πως μπορεί κανείς να επικοινωνήσει μέσα από μια εφαρμογή γραμμένη σε ανώτερη γλώσσα με τις λειτουργίες των Windows που επιτρέπουν την πρόσβαση σε συσκευές; Τα Windows περιλαμβάνουν λειτουργίες (functions) που επιτρέπουν την επικοινωνία ανάμεσα σε προγραμματιστικές εφαρμογές και οδηγούς συσκευών (device drivers). Η δυνατότητα αυτή βρίσκεται στο μέρος του λειτουργικού συστήματος που αναφέρεται ως διασύνδεση εφαρμογών προγραμματισμού (Applications Programmers Interface ή API). Οι λειτουργίες αυτές δίνουν τη δυνατότητα σε εφαρμογές να διαβάζουν και να γραφούν στη μνήμη σε δίσκους και σε άλλες συσκευές.

Κάποιες από αυτές τις λειτουργίες χρησιμεύουν για την προσπέλαση της USB και μπορούν να κληθούν μέσα από ανώτερες γλώσσες όπως είναι η Visual Basic , η C, η C++ κτλ. Οι γλώσσες αυτές περιλαμβάνουν ειδικό μέρος που χρησιμεύει για την κλήση των λειτουργιών API των Windows.

Με βάση τα παραπάνω για την κατασκευή μιας περιφερειακής συσκευής USB είναι απαραίτητο, εκτός από το προγραμματισμό του κατάλληλου μικροελεγκτή, να διαθέτουμε και κατάλληλο οδηγό (driver) για τη συσκευή καθώς και ένα πρόγραμμα σε ανώτερη γλώσσα το οποίο θα επικοινωνεί με τη συσκευή καλώντας τον συγκεκριμένο οδηγό, μέσω των κατάλληλων συναρτήσεων API, προκειμένου να στείλει και να λάβει τα απαραίτητα δεδομένα.

## **2.2. Βασικά χαρακτηριστικά**

Ο Γενικός Σειριακός Δίαυλος (Universal Serial Bus) γεννήθηκε μέσα από την ανάγκη να συνδέσουμε διάφορων ειδών περιφερειακά με ένα τύπο συνδέσμου κατά το δυνατό μικρό σε μέγεθος, καταργώντας την ποικιλία των καλωδίων και των μετασχηματιστών τροφοδοσίας που συναντάμε σχεδόν σε κάθε υπολογιστικό σύστημα.

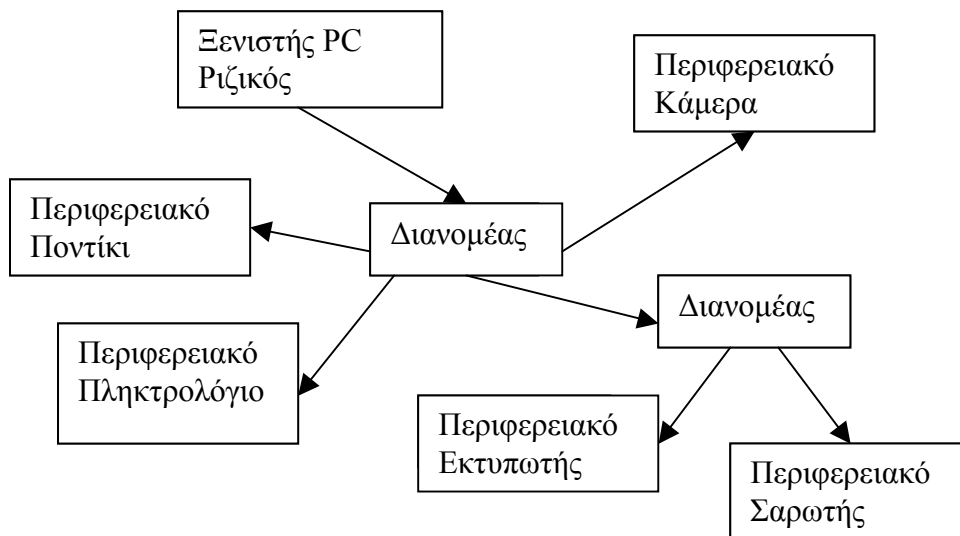
Από φυσική άποψη ο σύνδεσμος USB αποτελείται από τέσσερις επαφές. Το ένα ζεύγος καλωδίων χρησιμοποιείται από το πρωτόκολλο για την σειριακή εκπομπή των σημάτων ανάμεσα στον υπολογιστή (host ή ξενιστή) και την περιφερειακή συσκευή. Το δεύτερο ζεύγος χρησιμεύει για την παροχή τροφοδοσίας στα περιφερειακά

Ο δίαυλος USB υποστηρίζει σηματοδοσία πλήρους ταχύτητας (full speed) στα 12Mb/sec και σηματοδοσία χαμηλής ταχύτητας (low speed) στα 1,5Mb/sec. Για την σύνδεση της πλήρους ταχύτητας προβλέπεται μήκος καλωδίου 5 μέτρα, ενώ το αντίστοιχο μήκος για την σύνδεση χαμηλής ταχύτητας είναι 3 μέτρα. Αυτά ισχύουν για τα τρέχοντα

πρότυπα 1.0 και 1.1 ενώ στις προδιαγραφές του USB2.0 ο μέγιστος ρυθμός επικοινωνίας προβλέπεται στα 480 Mb/sec.

Άλλα βασικά χαρακτηριστικά είναι:

- *Μεγάλος Αριθμός Περιφερειακών Συσκευών* συγκεκριμένα έχουμε την δυνατότητα να συνδέσουμε σε ένα μόνο ξενιστή 127 περιφερειακά. Αυτό επιτυγχάνεται με την διαδοχική σύνδεση διανομέων (hubs). Βλέπε σχήμα 2.1.
- *Αυτόματη Αναγνώριση Κάθε Περιφερειακού* όπου με την σύνδεση του περιφερειακού USB σε υπολογιστή που βρίσκεται ήδη σε λειτουργία τα Windows αναγνωρίζουν αυτόματα την συσκευή και φορτώνουν τον αντίστοιχο οδηγό (driver).
- *Διαχείριση Ενέργειας*, είναι εφικτό να συνδέσουμε ένα περιφερειακό στον διάυλο USB, χωρίς να απαιτείται η διακοπή της τροφοδοσίας του συστήματος. Επιπλέον η συσκευή μπορεί να αντλήσει την τροφοδοσία της εξ ολοκλήρου από τα καλώδια USB με την προϋπόθεση ότι δεν αντλεί ρεύμα περισσότερο από 100 mA.



Σχήμα 2.1. Διασύνδεση διανομέων (Hubs) και περιφερειακών συσκευών στη τοπολογία του διαύλου USB.

### 2.3. Κλάσεις συσκευών USB

Οι προδιαγραφές της θύρας USB χωρίζουν τις διάφορες συσκευές σε κλάσεις ανάλογα με τις λειτουργίες και τις απαιτήσεις. Μια κατηγορία συσκευών αποτελεί συγκεκριμένη κλάση, όταν οι συσκευές αυτές μοιράζονται κοινές απαιτήσεις εισόδου/εξόδου. Με τις τρέχουσες προδιαγραφές διακρίνουμε συνολικά ένδεκα κλάσεις συσκευών: συνήθης συσκευή, συσκευή ήχου συσκευή επικοινωνιών διανομέας (hub), χειριστήριο (HID-Human Interface Device), συσκευή εικόνας, μόνιτορ, συσκευή φυσικής διασύνδεσης (PID – Physical Interface Device), συσκευή ισχύος, εκτυπωτής, και μονάδα αποθήκευσης. Κάθε κλάση συσκευών USB ορίζει τους δικούς της οδηγούς συσκευών (drivers). Ο σχεδιαστής που επιθυμεί να κατασκευάσει το δικό του περιφερειακό USB θα

πρέπει να λάβει υπόψη του ότι η κλάση που αναφέρεται ως χειριστήρια, ή (Συσκευές ελεγχόμενες από χειριστή -HIDs) είναι από πλευρά οδηγών η πλέον υποστηριζόμενη από τα Windows.

## 2.4. Μετάδοση δεδομένων μέσω του διαύλου USB

Παρακάτω δίνονται ορισμοί κάποιων απαραίτητων εννοιών ώστε να εξοικειωθούμε με την ορολογία της θύρας USB. Κατόπιν θα αναφερθούμε με συντομία στους τύπους μετάδοσης δεδομένων που προβλέπονται από τις προδιαγραφές της USB.

- **Τελικά σημεία Συσκευής (Device endpoints)**

Οι επικοινωνίες μέσω της θύρας USB γίνονται ανάμεσα στο ξενιστή (host Pc) και σε τελικά σημεία που βρίσκονται στα περιφερειακά. Σύμφωνα με τις προδιαγραφές της USB ως τελικό ή αλλιώς καταληκτικό σημείο συσκευής αναφέρεται ένα μέρος της μνήμης ή ένα σύνολο καταχωρητών (buffer) μέσα στον μικροελεγκτή του περιφερειακού, το οποίο μπορεί να διευθυνσιοδοτηθεί με μοναδικό τρόπο ώστε να γίνει ο αποδέκτης ή η πηγή δεδομένων σε μια ροή πληροφοριών ανάμεσα στο ξενιστή και στη συσκευή. Τα δεδομένα που βρίσκονται αποθηκευμένα σε ένα τελικό σημείο μπορεί να έχουν ληφθεί ή να περιμένουν την εκπομπή τους.

Η διεύθυνση (address) ενός τελικού σημείου περιλαμβάνει ένα κωδικό με τιμή από H00 έως H0F (δεκαδικού αριθμού από 0 έως 15) καθώς και έναν κωδικό που υποδεικνύει την κατεύθυνση της ροής. Η κατεύθυνση της ροής κρίνεται πάντα από την πλευρά του ξενιστή. Έτσι αν το τελικό σημείο χαρακτηρίζεται ως IN αν υποστηρίζει ροή από την συσκευή προς τον ξενιστή και χαρακτηρίζεται ως OUT εάν η ροή δεδομένων είναι από το περιφερειακό προς το ξενιστή. Υπάρχουν ειδικά τελικά σημεία, που χαρακτηρίζονται ως σημεία ελέγχου τα οποία είναι δικατευθυντικά. Ο τύπος μετάδοσης που υποστηρίζουν τα σημεία αυτά είναι η λεγόμενη **μετάδοση ελέγχου** με την οποία ο ξενιστής και το περιφερειακό ανταλλάσσουν τα απαραίτητα σήματα ελέγχου καθώς και άλλες μορφές πληροφορίας. Σε συσκευές που υποστηρίζουν (low speed) επικοινωνία εκτός από το μηδενικό σημείο ελέγχου έχουμε ακόμη δυο το πολύ σημεία (endpoints) που μπορούν να υποστηρίξουν IN ή OUT μεταδόσεις δεδομένων.

- **Επικοινωνιακός Σωλήνας (Pipe)**

Για να επιτευχθεί η επικοινωνία μεταξύ ενός τελικού σημείου και του ξενιστή είναι απαραίτητο να δημιουργηθεί ένα επικοινωνιακός δίαυλος ή αλλιώς σωλήνας που να συνδέει τον ξενιστή με το συγκεκριμένο τελικό σημείο. Πρόκειται για μια εικονική διασύνδεση και όχι για μια φυσική γραμμή επικοινωνίας. Μπορεί να την φαντάζεται κανείς σαν επιλογή ρυθμίσεων που κάνει ο ελεγκτής του ξενιστή στο λογισμικό προκειμένου να εξυπηρετήσει τις ανάγκες επικοινωνίας του συγκεκριμένου τελικού σημείου. Για ένα άλλο τελικό σημείο του ίδιου περιφερειακού θα γίνει μια άλλη επιλογή ρυθμίσεων, με αποτέλεσμα να δημιουργηθεί διαφορετικός σωλήνας επικοινωνίας.

Ο σωλήνας δημιουργείται από τον ξενιστή με βάση τις πληροφορίες που αυτός αντλεί για τα τελικά σημεία της συσκευής κατά την σύνδεση της συσκευής στο δίαυλο ή κατά την έναρξη της τροφοδοσίας. Στη φάση αυτή ενεργοποιείται μια διαδικασία που ονομάζεται απαρίθμηση (enumeration) των συσκευών με την οποία όπως θα δούμε ο ξενιστής συγκεντρώνει όλες τις απαραίτητες πληροφορίες για τις περιφερειακές συσκευές. Κατά την απαρίθμηση των συσκευών, κάθε καταληκτικό σημείο αποστέλλει

μια δομή δεδομένων (data structure) που ονομάζεται «**περιγραφέας**» (descriptor) του καταληκτικού σημείου. Υπάρχουν πολλών ειδών περιγραφείς που αποστέλλονται προς τον ξενιστή κατά την απαρίθμηση, όμως για την δημιουργία των σωλήνων επικοινωνίας η κρίσιμη πληροφορία περιέχεται στα πεδία του περιγραφέα του καταληκτικού σημείου (endpoint descriptor). Η πληροφορία αυτή περιλαμβάνει την διεύθυνση του καταληκτικού σημείου, τον τύπο της μετάδοσης που θα χρησιμοποιηθεί, τον μέγιστο αριθμό των bytes που θα περιλαμβάνουν τα πακέτα δεδομένων και την συχνότητα με την οποία θα αποστέλλονται τα πακέτα δεδομένων. Έτσι ο ξενιστής δημιουργεί ένα «σωλήνα», μια εικονική διασύνδεση με συγκεκριμένο εύρος ανάλογα με την προβλεπόμενη «ροή».

- **Τύποι Μετάδοσης Δεδομένων (Data Transfers)**

Ανάλογα με το είδος της επικοινωνίας που εξυπηρετεί ο δίαυλος USB χρησιμοποιεί για τις ανάγκες της επικοινωνίας ένα από τους εξής δυνατούς τρόπους μετάδοσης δεδομένων.

- Μετάδοση Ελέγχου (Control Transfer)
- Συμπαγής Μετάδοση (Bulk Transfer)
- Ισόχρονη Μετάδοση (Isochronous transfer)
- Μετάδοση Διακοπών (Interrupt transfer)

Ο καθένας από τους τέσσερις παραπάνω τύπους μετάδοσης έχει τις δικές του προδιαγραφές για ρυθμό μετάδοσης, ποσότητα δεδομένων, ακρίβεια στο χρονοισμό των δεδομένων και διόρθωση σφαλμάτων.

Η **Μετάδοση Ελέγχου** (Control Transfer) αποσκοπεί κυρίως στην αποστολή και λήψη πληροφοριών για την διαμόρφωση (configuration) και τις ρυθμίσεις μιας συσκευής κατά την διαδικασία της απαρίθμησης (enumeration). Αυτός ο τρόπος μετάδοσης διαθέτει προδιαγεγραμμένο σύνολο «αιτήσεων» ή εντολών (requests) τις οποίες ο ελεγκτής USB του ξενιστή χρησιμοποιεί για να στείλει ή να λάβει σήματα ελέγχου. Μια εφαρμογή μπορεί να χρησιμοποιήσει τις εντολές αυτές μέσω του αντίστοιχου οδηγού (driver) για να στείλει ή να λάβει άλλα δεδομένα (οποιαδήποτε bytes). Όλα τα περιφερειακά USB οφείλουν αυτό τον τρόπο μετάδοσης, μέσω του μηδενικού καταληκτικού σημείου. Η ορθότητα των μεταδιδόμενων δεδομένων ελέγχεται με bits που υπολογίζονται μέσω του αλγόριθμου CRC (Κυκλικός Έλεγχος Πλεονασμού – Cyclic Redundancy Check). Όταν ο έλεγχος αποτυγχάνει, ενεργοποιείται η επανεκπομπή των δεδομένων.

Η μετάδοση ελέγχου είναι σαφώς προδιαγεγραμμένη μορφή και αποτελείται από τρία στάδια: το Στάδιο Προετοιμασίας (Setup Stage), το Στάδιο των Δεδομένων (Data Stage) και το Στάδιο Κατάστασης (Status Stage). Με δύο λόγια στο στάδιο προετοιμασίας περιλαμβάνεται ο κωδικός της αίτησης που ενεργοποιεί την συγκεκριμένη λειτουργία ελέγχου. Στο στάδιο δεδομένων περιέχονται οι ρυθμίσεις ή οι πληροφορίες ελέγχου ή τα μεταδιδόμενα δεδομένα, ενώ στο τελικό στάδιο κατάστασης ελέγχεται η επιτυχία της διαδικασίας με αποστολή σημάτων χειραψίας.

Σε συσκευές χαμηλής ταχύτητας ο μέγιστος ρυθμός μετάδοσης σε μια Μετάδοση Ελέγχου είναι 24 bytes/ms ή 24 Kbytes per sec.

Οι **Συμπαγείς Μεταδόσεις** (Bulk Transfer) χρησιμοποιούνται για μαζικές μεταδόσεις δεδομένων, που δεν απαιτούν πειστικά την έγκαιρη αποστολή τους. Αν και η Συμπαγής μετάδοση μπορεί να είναι πολύ γρήγορη είναι πάντα χαμηλής προτεραιότητας και έτσι

παραχωρεί την σειρά σε άλλους τύπους μετάδοσης. Οι συμπαγείς μεταδόσεις γεμίζουν τα χρονικά κενά, διεκδικώντας το εύρος του διαύλου το οποίο περισσεύει. Τυπικά παραδείγματα συσκευών που χρησιμοποιούν συμπαγείς μεταδόσεις δεδομένων μέσω της θύρας USB είναι οι εκτυπωτές, οι οπτικοί σαρωτές και οι οδηγοί δίσκων (disk drives). Ας σημειωθεί ότι αυτός ο τύπος μετάδοσης δεν υποστηρίζεται από συγκεκριμένες κλάσεις συσκευών, όπως αυτές που προαναφέρθηκαν.

Αν ο δίαυλος USB είναι κατά τα άλλα αδρανής και μεταφέρει μόνο μια συμπαγή μετάδοση, ο ρυθμός μετάδοσης μπορεί να φτάσει το μέγιστο των 1216 bytes ανά ms ή 1,2160 Mbytes per sec. Ας σημειωθεί ότι ο συγκεκριμένος τύπος μετάδοσης υποστηρίζεται μόνο από συσκευές υψηλής ταχύτητας.

Οι **Μεταδόσεις Διακοπών** (Interrupt transfer) προορίζονται για μεσαίες ποσότητες δεδομένων που πρέπει να μεταδοθούν, μέσα σε κάποια χρονικά όρια, τα οποία είναι σχετικά μικρά. Για παράδειγμα η μετάδοση των δεδομένων που παράγει ένα πληκτρολόγιο δεν μπορεί να αργεί τόσο έτσι ώστε να γίνεται ενοχλητικά αντιληπτή. Από την άλλη μεριά μια τέτοια μετάδοση δεν έχει ιδιαίτερα αυστηρές απαιτήσεις χρονισμού. Στην περίπτωση αυτή οι προδιαγραφές της θύρας USB προβλέπουν ένα μέγιστο περιθώριο μέσα στο οποίο ο ελεγκτής της USB πρέπει να εκτελέσει την μετάδοση. Έτσι δεν είναι αυστηρά καθορισμένο το πότε θα γίνει η μετάδοση ενός πακέτου δεδομένων, είναι όμως αυστηρά καθορισμένος ο μέγιστος χρόνος που μπορεί να διέλθει ανάμεσα σε διαδοχικές μεταφορές.

Οι μεταδόσεις διακοπών εξασφαλίζουν ότι ανά τακτά και μικρά χρονικά διαστήματα ο ελεγκτής του ξενιστή θα διερευνήσει την ανάγκη μεταφοράς δεδομένων από ή προς την συσκευή. Δεν πρόκειται λοιπόν για τις διακοπές (hardware interrupts) που προκαλούν την διακοπή του προγράμματος και την εξυπηρέτηση της ρουτίνας διακοπής, κατά τα γνωστά. Πρόκειται για την διαδικασία των διαδοχικών ερωτήσεων προς την συσκευή, η οποία είναι γνωστή ως polling. Κάτι παρόμοιο συμβαίνει και με τους άλλους τύπους μεταδόσεων, στους οποίους η μεταφορά συμβαίνει όταν ο ξενιστής απευθύνεται στη συσκευή. Ο συσχετισμός εδώ με τις διακοπές σχετίζεται με τη άμεση σχεδόν ανταπόκριση στις ανάγκες της συσκευής, μέσα στο όρια ενός καθορισμένου μικρού χρονικού περιθωρίου.

Τυπικές συσκευές που χρησιμοποιούν μεταδόσεις διακοπών είναι τα ποντίκια τα πληκτρολόγια και τα joysticks. Η κλάση συσκευών USB που αναφέρονται ως HIDs (Human Interface Devices- Συσκευές Διασύνδεσης χειριστή ή χειριστήρια) υποστηρίζει γενικά τις μεταδόσεις διακοπών εκτός από τις αναγκαστικές μεταδόσεις ελέγχου. Ο μέγιστος ρυθμός μιας μετάδοσης διακοπών σε ένα αδρανή δίαυλο USB είναι 64 Kbytes per sec για μια συσκευή υψηλής ταχύτητας. Ο δίαυλος μπορεί δηλαδή να εκπέμψει, ένα πακέτο δεδομένων αποτελούμενο κατά μέγιστο από 64 bytes με χρονικό περιθώριο εκπομπής από 1 ms μέχρι 255 ms. Ο μέγιστος ρυθμός προκύπτει με βάση το ελάχιστο περιθώριο. Για μετάδοση χαμηλής ταχύτητας έχουμε ένα πακέτο των 8 bytes με ελάχιστο χρονικό περιθώριο τα 10 ms (μέχρι 255 ms), δηλαδή έχουμε μέγιστο ρυθμό μετάδοσης 0.8 Kbytes per sec.

Οι **Ισόχρονες Μεταδόσεις** (Isochronous transfer) προορίζονται για δεδομένα συνεχούς ροής (streaming data), όπως αυτά που μεταφέρουν σε πραγματικό χρόνο μουσική ή σήμα εικόνας (video). Όπως είναι φυσικό ο χρονισμός τέτοιων δεδομένων είναι πολύ κρίσιμος ώστε να εξασφαλίζεται η συνέχεια. Έτσι ο ρυθμός μετάδοσης εδώ είναι απολύτως σταθερός, με αποτέλεσμα να μεταφέρεται συγκεκριμένος αριθμός πακέτων δεδομένων

ανά ms. Ο μέγιστος ρυθμών είναι 1023 bytes/ms ή 1,0230 Mbytes per sec. Η πρόσβαση στο δίαυλο για τις ισόχρονες μεταδόσεις είναι εξασφαλισμένη αφού αυτές έχουν την υψηλότερη δυνατή προτεραιότητα. Είναι ο μόνος τύπος μετάδοσης που δεν περιλαμβάνει ελέγχους σφαλμάτων. Έτσι εξασφαλίζεται μεν η έγκαιρη μετάδοση αλλά το τίμημα είναι η λήψη περιστασιακών σφαλμάτων. Ας σημειωθεί ότι ισόχρονες μεταδόσεις υποστηρίζουν μόνον οι συσκευές υψηλής ταχύτητας.

## 2.5. Τα μέρη μιας μετάδοσης USB

Όπως εξηγήσαμε παραπάνω, ο ριζικός διανομέας του ξενιστή είναι ο κόμβος όπου συγκεντρώνονται όλες οι περιφερειακές συσκευές USB. Το εύρος του διαύλου πρέπει να κατανέμεται στις συσκευές αυτές ανάλογα με τις απαιτήσεις τους. Άρα ο ελεγκτής USB οφείλει να καταμερίζει τις μεταδόσεις, ώστε να εξυπηρετεί όλη τη ζήτηση.

Η βάση του καταμερισμού που κάνει ο ελεγκτής είναι το χρονικό πλαίσιο (frame) του ενός millisecond. Δηλαδή οι μεταδόσεις γίνονται σε διαδοχικά πλαίσια του 1ms που το καθένα περιλαμβάνει τμήματα διαφορετικών μεταδόσεων προς διαφορετικές συσκευές. Σχηματικά ο τρόπος αυτός καταμερισμού της ροής φαίνεται στο σχήμα 2.2.

Αρχή Πλαισίου	Συσκευή 3 Τελικό σημείο 2	Συσκευή 1 Τελικό σημείο 0	Συσκευή 4 Τελικό σημείο 3	Συσκευή 3 Τελικό σημείο 1	Συσκευή 2 Τελικό σημείο 0	Κενό
---------------	------------------------------	------------------------------	------------------------------	------------------------------	------------------------------	------

Σχήμα 2.2. Προγραμματισμός διεξαγωγών προς τελικά σημεία συσκευών, μέσα σε πλαίσια του 1 ms. Το επόμενο ms θα αποσταλεί ένα άλλο παρόμοιο πλαίσιο.

Όταν ο ξενιστής ζητήσει να στείλει ή να λάβει ένα σύνολο από δεδομένα τότε ξεκινά μια διαδικασία που, όπως είδαμε ονομάζεται μετάδοση (transfer). Η μετάδοση ολοκληρώνεται μόνο όταν έχει μεταδοθεί όλος ο όγκος της ζητούμενης πληροφορίας. Στα επόμενα περιγράφονται τα μέρη μιας μετάδοσης.

- **Διεξαγωγές**

Μια μετάδοση συνήθως χωρίζεται σε μικρότερα τμήματα που ονομάζονται «διεξαγωγές» (transactions), τα οποία ταξινομούνται μέσα στα πλαίσια του 1ms όπως φαίνεται στο σχήμα 2.1. Μια διεξαγωγή ορίζεται ως η αποστολή ή λήψη ενός τεμαχίου πληροφορίας προς ή από καταληκτικό σημείο συσκευής. Για να ολοκληρωθεί η αιτούμενη μετάδοση θα χρειαστεί να προγραμματίσει ο ξενιστής διαδοχικές διεξαγωγές, μέσα στο ίδιο ή σε διαδοχικά πλαίσια. Αν και μια πλήρης μετάδοση τεμαχίζεται σε αυτόνομες διεξαγωγές οι οποίες μπορεί να καλύπτουν παραπάνω από ένα πλαίσια κάθε διεξαγωγή είναι μια απλή επικοινωνία που ολοκληρώνεται χωρίς διακοπή. Οι διεξαγωγές ονομάζονται και στάδια (stages) μιας μετάδοσης.

- **Πακέτα** (Φάσεις μιας Διεξαγωγής)

Από τι αποτελείται μια απλή διεξαγωγή;

Κάθε διεξαγωγή περιέχει ένα αριθμό από φάσεις που περιλαμβάνουν ένα πακέτο. Κάθε πακέτο είναι ένας δομικός λίθος πληροφορίας με καθορισμένη μορφή: περιέχει μια σειρά από πεδία, από τα οποία το πρώτο είναι το πεδίο SYNC για το συγχρονισμό των κυκλωμάτων χρονισμού και το δεύτερο είναι το πεδίο PID (Packet Identifier), το οποίο πληροφορεί τις συσκευές για το είδος του αποστελλόμενου πακέτου. Ένα τρίτο πεδίο

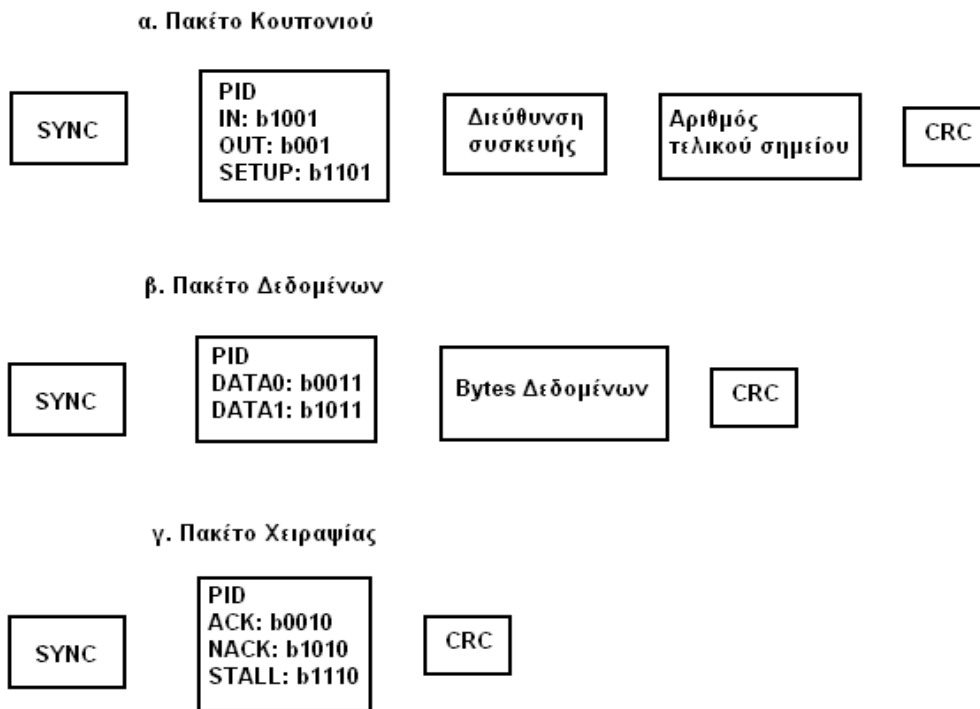
περιλαμβάνει επιπλέον πληροφορία που μπορεί να διαφέρει ανάλογα με το είδος του πακέτου. Τέλος ένα πακέτο περιλαμβάνει συνήθως 16 bits για τον έλεγχο CRC.

Υπάρχουν τριών ειδών βασικά πακέτα και δυο βοηθητικά ή ειδικά πακέτα.

Διακρίνονται μεταξύ τους από το περιεχόμενο της επιπλέον πληροφορίας που περιέχουν. Η ταξινόμηση τους φαίνεται στον πίνακα 2.1. Τα τρία βασικά πακέτα (κουπόνι δεδομένα και χειραψία – token data and handshake) συνιστούν τις τρεις φάσεις μιας διεξαγωγής. Τα ειδικά πακέτα (SOF και PRE) προσκολλώνται συνήθως σε κάποιο από τα τρία βασικά.

Το πακέτο κουπονιού (token packet) είναι η πρώτη φάση σε μια διεξαγωγή.

Περιέχει όλα τα βασικά πεδία ενός πακέτου, δηλαδή SYNC ,PID επιπλέον πληροφορία και CRC bits. Το περιεχόμενο του PID στο πακέτο κουπονιού περιέχει πληροφορία για το τύπο της διεξαγωγής. Έτσι όταν το PID στο κουπόνι περιέχει τον δυαδικό αριθμό 0001, αυτό σημαίνει ότι η διεξαγωγή είναι OUT, δηλαδή ότι η ροή της πληροφορίας κατευθύνεται από τον ξενιστή προς την συσκευή. Όταν το PID στο κουπόνι περιέχει τον δυαδικό αριθμό 1001, αυτό σημαίνει ότι η διεξαγωγή είναι IN δηλαδή ότι η ροή της πληροφορίας κατευθύνεται από τη συσκευή προς τον ξενιστή. Όταν το PID είναι 1101 τότε η διεξαγωγή ονομάζεται διεξαγωγή προετοιμασίας (Setup Stage), οπότε το πακέτο δεδομένων που ακολουθεί αποστέλλει τον κωδικό μιας εντολής ελέγχου (control request) Στο σχήμα 2.3 παρουσιάζεται η συγκρότηση ενός πακέτου κουπονιού.



Σχήμα 2.3. Οι τρεις φάσεις μιας διεξαγωγής (α) Πακέτο Κουπονιού, (β) Πακέτο Δεδομένων, (γ) Πακέτο Χειραψίας.

Το πακέτο κουπονιού αποστέλλεται πάντα από τον ξενιστή, ο οποίος με τον τρόπο αρχίζει μια διεξαγωγή, καθορίζοντας τον τύπο της διεξαγωγής (IN, OUT ή SETUP) τη συσκευή και το τελικό σημείο. Αν πρόκειται για διεξαγωγή χαμηλής ταχύτητας προς την κατωφέρεια, τότε πριν από το PID αποστέλλεται πάντα ένα ειδικό πακέτο το PRE. Αυτό περιέχει ένα επιπλέον PID με τον κωδικό 1100, που ενεργοποιεί την ρύθμιση χαμηλής ταχύτητας στο δίαυλο. Επίσης το πακέτο κουπονιού μπορεί να μεταφέρει το βοηθητικό



πακέτο SOF (Start Of Frame –Αρχή Πλαισίου) το οποίο αποστέλλει ο ξενιστής στην αρχή κάθε πλαισίου ως χρονική αναφορά.

Το **πακέτο δεδομένων** (data packet) ακολουθεί μετά το πακέτο κουπονιού και αποτελεί τη δεύτερη φάση μιας διεξαγωγής. Εκτός από τα συμβατικά πεδία SYNC PID και CRC ένα πακέτο δεδομένων περιέχει ως επιπλέον πληροφορία μια ροή από bytes τα οποία περιέχουν την προς μετάδοση πληροφορία, σε κάθε τύπο μετάδοσης.

Το PID των πακέτων δεδομένων περιέχει ένα πεδίο το λεγόμενο DATA0/1 το οποίο χρησιμεύει για το συγχρονισμό ανάμεσα στον αποστολέα και στο δεκτή των δεδομένων. Το Bit 3, που ονομάζεται και μεταβαλλόμενο bit, εναλλάσσεται από 0 σε 1 σε διαδοχικά πακέτα δεδομένων και έτσι πληροφορεί τον δέκτη ότι λαμβάνει την αναμενόμενη σειρά πακέτων χωρίς παραλείψεις.

Τύπος Πακέτου	Μετάδοση που το χρησιμοποιεί	Επιπλέον περιεχόμενα (εκτός από SYNC, PID)
Κουπόνι	Όλες	Διεύθυνση συσκευής και τελικού σημείου, CRC
Δεδομένα	Όλες	Bytes Δεδομένων (μέχρι 1023), CRC
Χειραψία	Μετάδοση Ελέγχου, Διακοπών, Συμπαγής	Δεν υπάρχουν. Το PID περιέχει τον κωδικό χειραψίας
SOF	Το στέλνει ο ξενιστής	Αριθμός Πλαισίου, CRC
PRE	Όλες οι μεταδόσεις χαμηλής ταχύτητας	Δεν υπάρχουν

Πίνακας 2.1. Τύποι Πακέτων.

Τέλος το **πακέτο χειραψίας** (handshake packet) που ονομάζεται αλλιώς και πακέτο κατάστασης (status packet), περιέχει στο PID ένα κώδικα, με τον οποίο ο παραλήπτης ενημερώνει τον αποστολέα για την κατάσταση των λαμβανόμενων πακέτων δεδομένων. Υπάρχουν τρεις κώδικες χειραψίας για την επιβεβαίωση μιας διεξαγωγής: Ο κώδικας **ACK** σημαίνει ότι ο παραλήπτης έλαβε με ασφάλεια και χωρίς σφάλματα τα δεδομένα. Ο κώδικας **NAC** στο PID του πακέτου χειραψίας σημαίνει ότι προσωρινά ο παραλήπτης δεν είναι σε θέση να λάβει, οπότε ο αποστολέας θα πρέπει να ξαναδοκιμάσει αργότερα. Τέλος ο κώδικας **STALL** ειδοποιεί τον αποστολέα ότι ο παραλήπτης της διεξαγωγής αδυνατεί να τη φέρει σε πέρας, είτε επειδή έχει κάποιο πρόβλημα είτε επειδή δεν σχεδιάστηκε για να ανταποκρίνεται σε συγκεκριμένα αιτήματα.

Τα διαδοχικά πακέτα κουπονιού, δεδομένων και χειραψίας αποτελούν τον κύκλο μιας διεξαγωγής. Μόνον οι Ισόχρονες μεταδόσεις δεν περιλαμβάνουν την φάση της χειραψίας στις διεξαγωγές, οπότε αποστέλλουν μόνο τα πακέτα κουπονιού και δεδομένων. Ο λόγος είναι ότι ο αυστηρός χρονισμός των ισόχρονων μεταδόσεων δεν επιτρέπει ούτως ή άλλως την επανεκπομπή λανθασμένων δεδομένων.

- **Ειδικό Θέμα : Στάδια Των Μεταδόσεων Ελέγχου**

Όπως αναφέραμε παραπάνω, κάθε διεξαγωγή αποτελούμενη από τις φάσεις ή πακέτα που περιγράψαμε είναι ένα στάδιο σε μια μετάδοση. Ανάλογα με το πλήθος των δεδομένων, κάθε μετάδοση μπορεί να περιλαμβάνει πολλές διεξαγωγές προγραμματισμένες σε διάφορα πλαίσια. Ειδικά οι μεταδόσεις ελέγχου περιλαμβάνουν τρία στάδια ,δηλαδή τρεις διαδοχικές διεξαγωγές που η κάθε μια έχει διαφορετικό ρόλο. Θυμίζουμε ότι οι μεταδόσεις ελέγχου προορίζονται κυρίως για την μετάδοση εντολών ελέγχου και ρυθμίσεων.

Έτσι το πρώτο στάδιο, που ονομάζεται στάδιο προετοιμασίας (setup stage) είναι μια διεξαγωγή (κουπόνι-δεδομένα-χειραψία) της οποίας το κουπόνι δηλώνει τον τύπο της διεξαγωγής ως SETUP, το πακέτο δεδομένων περιέχει τον κωδικό της εντολής ή αίτησης που αποστέλλει ο ξενιστής και η χειραψία επιβεβαιώνει την ασφαλή λήψη.

Το δεύτερο στάδιο ονομάζεται στάδιο δεδομένων (data stage) και δεν είναι υποχρεωτικό. Αποστέλλεται μόνο εάν υπάρχουν συγκεκριμένα δεδομένα προς αποστολή. Περιλαμβάνει τις τρεις φάσεις (κουπόνι-δεδομένα-χειραψία) της οποίας το κουπόνι δηλώνει εάν η διεξαγωγή είναι IN ή OUT, το πακέτο δεδομένων περιέχει τις αποστέλλόμενες ρυθμίσεις ή άλλα δεδομένα και η χειραψία επιβεβαιώνει πάλι την ασφαλή λήψη των δεδομένων. Αν δεν αρκεί μόνο μια διεξαγωγή για τα αποστέλλόμενα δεδομένα (μέγιστο μήκος 64 bytes σε υψηλή ταχύτητα και 8 σε χαμηλή), τότε θα προγραμματιστούν περισσότερες.

Το τρίτο στάδιο ονομάζεται στάδιο κατάστασης (status stage) και περιλαμβάνει ένα κουπόνι που εκπέμπει ο ξενιστής και χαρακτηρίζει την διεξαγωγή ως IN και OUT, ένα πακέτο δεδομένων που περιλαμβάνει τον κώδικα ACK, NAK ή STALL και ένα πακέτο χειραψίας που επιβεβαιώνει την ασφαλή λήψη του κωδικού κατάστασης.

Ας σημειωθεί ότι οι άλλοι τύποι μετάδοσης είναι πιο απλοί και δομούνται σε απλά στάδια δεδομένων (IN ή OUT), που το καθένα αντιστοιχεί στις τρεις φάσεις μιας διεξαγωγής (κουπόνι-δεδομένα-χειραψία θυμηθείτε όμως ότι οι ισόχρονες μεταδόσεις δεν περιέχουν τη χειραψία). Ο αριθμός των απαιτούμενων διεξαγωγών για μια μετάδοση εξαρτάται από το πλήθος των δεδομένων.

## **2.6. Απαρίθμηση (Enumeration)**

Προτού μπορέσει μια εφαρμογή να επικοινωνήσει με μια συσκευή USB, θα πρέπει ο ξενιστής να μάθει ότι είναι απαραίτητο για την επικοινωνία του με τις συσκευές. Θα πρέπει να αντιστοιχήσει μια διεύθυνση σε κάθε συσκευή, να διαβάσει πληροφορίες σχετικά με τους δυνατούς τρόπους διαμόρφωσης και ρύθμισης της συσκευής, να βρει την περιγραφή των καταληκτικών σημείων που υποστηρίζει η συσκευή, να βρει και να φορτώσει τον απαραίτητο οδηγό της συσκευής (device-driver) και τέλος να επιλέξει μια από τις δυνατές ρυθμίσεις (configurations) της συσκευής. Η απαρίθμηση συμβαίνει κάθε φορά που ξεκινά η λειτουργία του διαύλου USB ή συνδέεται μια νέα συσκευή στο δίαυλο.

Για την απαρίθμηση των συνδεδεμένων συσκευών ο ξενιστής αποστέλλει μεταδόσεις ελέγχου, που απευθύνουν αιτήσεις για μεταφορά δεδομένων προς το τελικό σημείο 0 των συσκευών. Όλες οι συσκευές οφείλουν να υποστηρίζουν μεταδόσεις ελέγχου και τουλάχιστον το μηδενικό τελικό σημείο. Ο προγραμματισμός (firmware) των συσκευών οφείλει να προβλέπει την καταχώρηση των απαραίτητων «περιγραφών», δηλαδή των δομών δεδομένων που είναι απαραίτητες για την αρχική αναγνώριση και χρήση της συσκευής.

Από τα σπουδαιότερα πεδία που χρησιμεύουν για την εύρεση και φόρτωση του κατάλληλου οδηγού της συσκευής (device driver) είναι η ταυτότητα του κατασκευαστή και του προϊόντος, που αναφέρονται ως Vendor ID (VID) και Product ID (PID). Αυτά περιέχονται στον λεγόμενο περιγραφέα συσκευής (device descriptor), μια από τις πρώτες δομές που ανασύρει ο ξενιστής από τη συσκευή.

### Τα βήματα της διαδικασίας απαρίθμησης είναι τα εξής:

1. Η συσκευή συνδέεται στη θύρα του διανομέα
2. Ο διανομέας ειδοποιεί τον ξενιστή για την αλλαγή κατάστασης, όταν ο ξενιστής τον ρωτήσει (με μια αίτηση Get\_Port\_Status).
3. Ο ξενιστής ζητάει από τον διανομέα να θέσει την συσκευή σε κατάσταση Reset.
4. Η συσκευή βρίσκεται τώρα στην λεγόμενη κατάσταση τροφοδοσίας (powered state) και μπορεί να ανταποκριθεί σε αιτήσεις ελέγχου προς το μηδενικό σημείο. Η προσωρινή διεύθυνση της συσκευής είναι η H00.
5. Ο ξενιστής ανασύρει τον «περιγραφέα συσκευής» και μαθαίνει το μέγιστο μέγεθος πακέτου, για το μηδενικό τελικό σημείο, καθώς και το VID και PID της συσκευής. Έτσι μπορεί να συσχετίσει τη συσκευή με κάποιον οδηγό (driver).
6. Ο ξενιστής αποδίδει μια μοναδική διεύθυνση στη συσκευή.
7. Ο ξενιστής ανασύρει τον «περιγραφέα διαμόρφωσης» (configurations descriptor) καθώς και τους υπόλοιπους περιγραφείς για τις δυνατές χρήσεις και τα τελικά σημεία της συσκευής (interface και endpoint descriptors).
8. Έχοντας όλη την απαραίτητη πληροφορία για την συσκευή, ο ξενιστής βρίσκει και φορτώνει τον απαραίτητο οδηγό της συσκευής. Για το σκοπό αυτό συσχετίζει κυρίως τα πεδία VID και PID του «περιγραφέα συσκευής» με τα περιεχόμενα του αρχείου INF, το οποίο περιέχει το όνομα του οδηγού που είναι κατάλληλος για τη συσκευή.
9. Ο οδηγός της συσκευής επιλέγει μια από τις δυνατές ρυθμίσεις και διαμορφώσεις για τη συσκευή.

Κατά την διάρκεια της παραπάνω διαδικασίας, η συσκευή περνά διαδοχικά από τις εξής φάσεις ή καταστάσεις, που προβλέπονται από τις προδιαγραφές της θύρας USB:

- A. Κατάσταση Τροφοδοσίας (Powered State).
- B. Κατάσταση Επανεκκίνησης (Reset State).
- Γ. Γενική Κατάσταση (Default State).
- Δ. Διαμορφωμένη Κατάσταση (Configured State).

## 2.7. Περιγραφείς (Descriptors)

Για να μπορέσει να επικοινωνήσει ο ξενιστής με μια συσκευή πρέπει απαραίτητα να γνωρίζει ποιες λειτουργίες υποστηρίζει η συσκευή, πόσα και πια τελικά σημεία διαθέτει για κάθε λειτουργική διαμόρφωση, τι ανάγκη σε εύρος ζώνης (bandwidth) έχει κάθε τελικό σημείο κ.α

Οι προδιαγραφές της θύρας USB προβλέπουν ότι κάθε συσκευή πρέπει να είναι σε θέση να στείλει πέντε δομές δεδομένων που περιέχουν τις πληροφορίες αυτές, όταν ερωτηθεί από τον ξενιστή κατά την διάρκεια της απαρίθμησης. Οι δομημένες αυτές πληροφορίες λέγονται «περιγραφείς» (descriptors) και εμπεριέχονται στον κώδικα firmware που καταχωρείτε με προγραμματισμό στη μνήμη EPROM της συσκευής.

Κάθε συσκευή έχει έναν «περιγραφέα συσκευής» (device descriptor), ο οποίος δίνει πληροφορίες για τον κατασκευαστή, τον κωδικό του προϊόντος, τον αριθμό σειράς,

την κλάση της συσκευής, τις ιδιότητες του τελικού μηδενικού σημείου και τον αριθμό των λειτουργικών διαμορφώσεων (configurations) που υποστηρίζει η συσκευή.

Σε κάθε διαμόρφωση της συσκευής αντιστοιχεί και ένας **«περιγραφέας διαμόρφωσης»** (configuration descriptor). Οι διαφορετικές διαμορφώσεις μπορεί να αντιστοιχούν σε διαφορετικές χρήσεις της συσκευής. Για παράδειγμα μια συσκευή υψηλής ισχύος μπορεί να βρίσκει χρήση και ως συσκευή χαμηλής ισχύος. Ο περιγραφέας διαμόρφωσης περιλαμβάνει πεδία που πληροφορούν τον ξενιστή για την χρήση της ισχύος που κάνει η συσκευή σε κάθε λειτουργική της διαμόρφωση. Επίσης πληροφορεί για τον αριθμό των διασυνδέσεων (interfaces) που παρέχει η συσκευή προς το εξωτερικό περιβάλλον, για την συγκεκριμένη διαμόρφωση.

Με τον όρο «διασύνδεση» (interface), οι προδιαγραφές της USB εννοούν τον συνδυασμό και τις ιδιότητες των τελικών σημείων, με τα οποία η συσκευή επικοινωνεί με το περιβάλλον της, προκειμένου να εισάγει και να εξάγει δεδομένα. Για κάθε δυνατή διαμόρφωση της συσκευής υπάρχει ο **«περιγραφέας διασύνδεσης»** (interface descriptor), που περιέχει πεδία για τον αριθμό των τελικών σημείων που υποστηρίζει η συσκευή, καθώς και για την κλάση στην οποία εμπίπτει η συγκεκριμένη χρήση της συσκευής.

Εκτός από το μηδενικό τελικό σημείο, κάθε τελικό σημείο που προβλέπεται από τον περιγραφέα διασύνδεσης έχει έναν **«περιγραφέα τελικού σημείου»**. Κάθε τέτοιος περιγραφέας περιλαμβάνει καταρχήν την διεύθυνση του τελικού σημείου. Το πεδίο αυτό περιέχει τον αριθμό του τελικού σημείου (συνολικά μπορούμε να έχουμε τρία τελικά σημεία με αριθμούς 0-2 σε συσκευές χαμηλής ταχύτητας και 0-15 σε συσκευές υψηλής ταχύτητας). Επίσης περιέχει ένα bit που υποδεικνύει την κατεύθυνση (IN ή OUT) του τελικού σημείου. Άλλα πεδία του περιγραφέα πληροφορούν τον ξενιστή για τον τύπο της μετάδοσης (ελέγχου, διακοπών, συμπαγή ή ισόχρονη) που υποστηρίζει το τελικό σημείο και για το μέγιστο αριθμό bytes που υπάρχουν στο πακέτο δεδομένων που μεταδίδεται από το τελικό σημείο. Τέλος ένα πεδίο του περιγραφέα τελικού σημείου πληροφορεί τον ξενιστή για τη συχνότητα (polling frequency) με την οποία θα λαμβάνει ή θα αποστέλλει δείγματα δεδομένων από ή προς τη συσκευή. Αυτή η πληροφορία έχει νόημα στην περίπτωση των μεταδόσεων διακοπών και των ισόχρονων μεταδόσεων.

Ένας τελευταίος τύπος περιγραφέα είναι ο **«περιγραφέας κειμένου»** (string descriptor). Αυτός περιέχει προαιρετικές πληροφορίες σχετικά με τον κατασκευαστή και το προϊόν, σε μορφή κειμένου (Unicode).

Στον **περιγραφέα διασύνδεσης** ανήκει και ο **περιγραφέας κλάσης** (class descriptor), όπου περιέχονται οι πληροφορίες που ορίζουν την κλάση της συσκευής και τη μορφή των επικοινωνιών που υποστηρίζει. Έτσι εάν το περιφερειακό ανήκει στην κλάση των χειριστηρίων (HIDs), τότε ο περιγραφέας κλάσης είναι ο **«περιγραφέας αναφορών»**, που ορίζει τη μορφή των αναφορών εισόδου/εξόδου που χρησιμοποιεί το χειριστήριο.

Όλοι οι παραπάνω περιγραφείς διαβάζονται από τον ξενιστή κατά την διάρκεια της απαρίθμησης (enumeration) της συσκευής. Ο σχεδιαστής που θέλει να κατασκευάσει ένα περιφερειακό USB θα πρέπει να λάβει μέριμνα να συμπεριλάβει στον προγραμματισμό του μικροελεγκτή από τη μεριά του περιφερειακού ένα αρχείο που περιέχει τους περιγραφείς που αρμόζουν στη συγκεκριμένη συσκευή.

## 2.8. Ο μικροελεγκτής CY7C63001

Όπως έγινε φανερό από τα προηγούμενα, ο χειρισμός των απαιτήσεων της θύρας USB από τη μεριά της συσκευής απαιτεί ενσύρματη και προγραμματιζόμενη λογική με τη μορφή ενός ολοκληρωμένου μικροελεγκτή. Το κύκλωμα αυτό θα πρέπει να έχει

ενσωματωμένη θύρα USB και αντίστοιχους ακροδέκτες ικανούς να συνδεθούν σε θύρα του ξενιστή. Επιπλέον θα πρέπει να διαθέτει κατάλληλες, μνήμες, και ψηφιακές θύρες εισόδου/εξόδου. Ένας από τους πλέον γνωστούς ολοκληρωμένους μικροελεγκτές USB είναι ο CY7C63001 της εταιρίας CYPRESS. Παρακάτω ο αναγνώστης θα βρει περισσότερες πληροφορίες για τον συγκεκριμένο μικροελεγκτή.

Ο CY7C63001 ανήκει στην οικογένεια 8-bit RISK One Time Programmable (OTP) (Προγραμματιζόμενο Μια Φορά) μικροελεγκτών με ενσωματωμένη σειριακή μηχανή επικοινωνίας 1.5 Mbps (Serial Interface Engine - SIE). Ο μικροελεγκτής έχει ξεχωριστή μνήμη προγράμματος της τάξης των 4 Kbytes καθώς και ξεχωριστή μνήμη δεδομένων (128 byte). Το σύνολο των εντολών που υποστηρίζει ο μικροελεγκτής περιέχει 35 εντολές που χρησιμοποιούνται στις εκάστοτε εφαρμογές. Η μνήμη του προγράμματος χωρίζεται σε δυο λειτουργικά μέρη. Το πρώτο είναι το διάνυσμα διακοπών (interrupt vector) και το δεύτερο είναι ο χώρος του προγράμματος (program code). Ο συγκεκριμένος μικροελεγκτής δέχεται ως πηγή ρολογιού κρύσταλλο συχνότητας 6 MHz. Το σήμα του ρολογιού διπλασιάζεται στο εσωτερικό του μικροελεγκτή.

Όνομα	Περιγραφή
VCC	1 ακροδέκτης. Ο ακροδέκτης αυτός συνδέεται με το αντίστοιχο ακροδέκτη από την πλευρά του ξενιστή ο οποίος και παρέχει τάση τροφοδοσίας 5 volt. Στην πραγματικότητα η τάση τροφοδοσίας μπορεί να ποικίλει από 4 μέχρι και 5.25 Volt
Vss	1 ακροδέκτης. Συνδέεται στην γείωση.
Vpp	1 ακροδέκτης. Χρησιμοποιείται για να προγραμματίσουμε την εσωτερική μνήμη EPROM. Αυτός ο ακροδέκτης πρέπει να συνδέεται με την γείωση κάτω από κανονικές συνθήκες λειτουργίας.
XTALIN	1 ακροδέκτης. Είσοδο από τον κρύσταλλο του κυκλώματος.
XTALOUT	1 ακροδέκτης. Κλείνει κύκλωμα με το κρύσταλλο του κυκλώματος. (Πρέπει να μένει ασύνδετο σε περίπτωση που χρησιμοποιούμε εξωτερικό ταλαντωτή).
P0.0-P0.7, P1.0-P1.3	12 ακροδέκτες. Όλοι οι I/O ακροδέκτες αποτελούνται από εσωτερικές αντιστάσεις pull-up. Επίσης έχουμε την δυνατότητα να ρυθμίσουμε το ρεύμα οδήγησης κάθε ακροδέκτη. Το ρεύμα οδήγησης μπορεί να προγραμματιστεί σε μια από 16 διαφορετικές στάθμες.
D+, D-	2 ακροδέκτες. Αμφίδρομες γραμμές μεταφοράς δεδομένων της usb. Μια εξωτερική αντίσταση πρέπει να χρησιμοποιηθεί μεταξύ του D- ακροδέκτη και της VCC έτσι ώστε να επιλέξουμε χαμηλής ταχύτητας επικοινωνία.
CEXT	1 ακροδέκτης. Ανοικτού δρένου έξοδος και είσοδος Schmitt trigger. Μπορεί να συνδεθεί σε ένα εξωτερικό κύκλωμα RC για να δημιουργήσει μια περιοδική υπενθύμιση όταν βρίσκεται σε κατάσταση suspend mode.

Πίνακας 2.2. Λειτουργία ακροδεκτών του μικροελεγκτή cy7c63001.

Ο μικροελεγκτής διαθέτει δυο πόρτες, την P0 και την P1. Κάθε ακροδέκτης από τους συνολικά 12 μπορεί να χρησιμοποιηθεί για να δημιουργήσει interrupt (ρουτίνα εξυπηρέτησης διακοπής) στον μικροελεγκτή. Οι τέσσερις ακροδέκτες της P1 έχουν την ικανότητα να παρέχουν ρεύμα ικανό για την οδήγηση φωτοεκπέμπουσων διόδων (led) ενώ οι οκτώ ακροδέκτες της πόρτας P0 παρέχουν μικρότερο ρεύμα και είναι ικανοί για την οδήγηση φωτοδιόδων.

### Οργάνωση μνήμης προγράμματος

Στην παρακάτω εικόνα απεικονίζεται πως κατανέμεται η μνήμη προγράμματος στο εσωτερικό του μικροελεγκτή.

After Reset  
PC >>>>>>

0x0000	Reset Vector.
0x0002	Interrupt Vector – 128 $\mu$ s.
0x0004	Interrupt Vector – 1024 ms.
0x0006	Interrupt Vector – USB Endpoint 0.
0x0008	Interrupt Vector – USB Endpoint 1.
0x000A	Reserved.
0x000C	Interrupt Vector – GPIO.
0x000E	Interrupt Vector – CEXT.
0x0010	On-chip program Memory.
0x0FFF	4K ROM (CY7C63001A,CY7C63101A).

Σχήμα 2.4. Χώρος μνήμης προγράμματος.

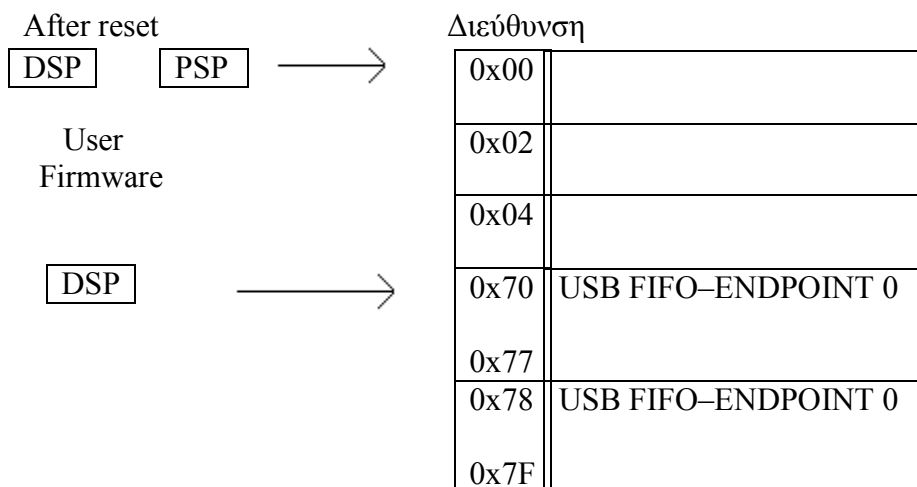
Σημείωση:

Στον συγκεκριμένο μικροελεγκτή έχουμε την δυνατότητα να προστατεύσουμε τον κώδικα που γράφει κάποιος χρήστης προγραμματίζοντας ένα συγκεκριμένο bit (fuse bit).

### Οργάνωση της μνήμης δεδομένων

Ο συγκεκριμένος ελεγκτής περιέχει 128 bytes μνήμης RAM. Τα 16 πρώτα byte της μνήμης των δεδομένων χρησιμοποιούνται σαν USB FIFOs για τα τελικά σημείο 0 και 1.

Κάθε τελικό σημείο συνδέεται με ένα 8-byte FIFO. Ο controller περιέχει δυο δείκτες (pointers) στη μνήμη RAM, τον Program Stack Pointer (PSP) και τον Data Stack Pointer (DSP). Η τιμή του PSP αυξάνεται κατά 2 όποτε εκτελείτε μια call εντολή και μειώνεται κατά δύο κάθε φορά που εκτελείτε μια RET εντολή. Ο DSP μειώνεται (pre-decrement) κατά ένα κάθε φορά που μια εντολή PUSH εκτελείτε και αυξάνει κατά ένα κάθε φορά που εκτελείτε μια εντολή POP. Η αρχική τιμή του DSP μετά από επανεκκίνηση (reset) είναι 0X00, το οποίο θα προκαλέσει την πρώτη PUSH να γράψει μέσα στο διάστημα του USB FIFO για το τελικό σημείο 1. Ωστόσο ο DSP πρέπει να τοποθετηθεί σε μια περιοχή όπως είναι η διεύθυνση 0X70 πριν αρχικοποιήσουμε οποιαδήποτε λειτουργία του σωρού δεδομένων (data stack) . Στο σχήμα 2.5 απεικονίζεται ο χώρος μνήμης Δεδομένων.



Σχήμα 2.5. Χώρος μνήμης Δεδομένων. (Data Memory Space).

**Περίληψη καταχωρητών**

Οι I/O καταχωρητές προσεγγίζονται από τις I/O Read και I/O Write (IOWR, IOWX) εντολές.

A/A	Register Name	I/O Address	Read/Write	Function.....
1	P0 Data	0X00	R/W	General Purpose I/O Port
2	P1 Data	0X01	R/W	General Purpose I/O Port
3	P0 IE	0X04	W	Interrupt enable for Port 0 pins
4	P1 IE	0X05	W	Interrupt enable for Port 1 pins
5	P0 Pull-up	0X08	W	Pull-up resistor control for Port 0 pins
6	P1 Pull-up	0X09	W	Pull-up resistor control for Port 1 pins
7	EPO TX Config	0X10	R/W	USB endpoint 0 transmit configuration
8	EP1 TX Config	0X11	R/W	USB endpoint 1 transmit configuration
9	USB DA	0X12	R/W	USB device address
10	USB SCR	0X13	R/W	USB status and control
11	EPO RX Status	0X14	R/W	USB endpoint 0 receive status
12	GIE	0X20	R/W	Global Interrupt Enable
13	WDT	0X21	W	Watch Dog Timer Clear
14	Cext	0X22	R/W	External R-C Timing circuit control
15	Timer	0X23	R	Free running timer
16	P0 Isink	0X30-0X37	W	Input Sink current control for port 0 pins. There is one Isink register for each pin
17	P1 Isink	0X38-0X3F	W	Input Sink current control for port 1 pins. There is one Isink register for each pin
18	SCR	0XFF	R/W	Processor status and control register

Πίνακας 2.3. Περίληψη Καταχωρητών.

**Επανεκκίνηση (Reset)**

Ο συγκεκριμένος μικροελεγκτής υποστηρίζει 3 είδη επανεκκινήσεων (reset). Όλοι οι καταχωρητές μεταβαίνουν στις αρχικές τους καταστάσεις κατά την διάρκεια μιας επανεκκίνησης. Η διεύθυνση της συσκευής τίθεται στο 0 και όλες οι ρουτίνες εξυπηρέτησης διακοπών (interrupts) απενεργοποιούνται. Επιπλέον ο δείκτης σωρού των δεδομένων (PSP) τίθεται στην διεύθυνση 0X00 καθώς το ίδιο συμβαίνει και για τον δείκτη σωρού των δεδομένων (DSP). Ο χρήστης θα πρέπει να θέσει τον DSP σε μια περιοχή όπως η 0X70 επιφυλάσσοντας έτσι 16 bytes του διαστήματος του USB FIFO. Οι εντολές σε γλώσσα assembly που το κάνουν αυτό είναι:

MOV A,70h ; μετακίνησε τον 70h μέσα στον Accumulator, χρησιμοποίησε 70 αντί για 6F επειδή ο DSP πάντα μειώνεται κατά 1 πριν προκύψει η μεταφορά δεδομένων λόγω της εντολής PUSH

SWAP A, DSP ; μετακίνησε την τιμή του Accumulator μέσα στο DSP.

Οι τρεις τύποι επανεκκινήσεων είναι:

1. Power-On Reset (POR)
2. Watch Dog Reset (WDR)
3. USB Reset

Η εμφάνιση μιας επανεκκίνησης καταγράφεται στον καταχωρητή SCR (Status and Control Register) ο οποίος βρίσκεται στην I/O διεύθυνση 0xFF. Το να γράφεις και να διαβάζεις σε αυτό το καταχωρητή υποστηρίζεται από τις εντολές IORD και IOWR. Τα bit ένα δύο και επτά αντιστρέφονται και πρέπει να γραφτούν ως μηδενικά κατά την διάρκεια μιας εγγραφής. Κατά την διάρκεια μιας εγγραφής τα bit που έχουν αντιστραφεί πρέπει να αγνοηθούν. Τα bit 4,5 και 6 χρησιμοποιούνται για την καταγραφή της εμφάνισης POR, USB και WDR RESET αντίστοιχα. Ο κώδικας (firmware) μπορεί να «ρωτήσει» αυτά τα bit για να καθοριστεί η αιτία μιας επανεκκίνησης. Εάν εμφανιστεί ένα Watch Dog Reset ο κώδικας θα πρέπει να καθαρίσει (clear) το WDR bit (bit 6) στον καταχωρητή SCR για να ενεργοποιηθεί ξανά η USB μετάδοση.

B7	B6	B5	B4	B3	B2	B1	B0
Reserved	WDR	USBR	POR	SUSPEND	RESERVED	RESERVED	RUN
	R/W	R/W	R/W	R/W			R/W
0	0	0	1	0	0	0	1

Πίνακας 2.4. Καταχωρητής SCR (Status and Control Register).

### Power-On Reset (POR)

Power - On Reset (POR) εμφανίζεται κάθε φορά που τροφοδοτούμε τη συσκευή με τάση.

Το bit 4 του καταχωρητή SCR αναλαμβάνει να καταγράψει αυτό το συμβάν. (τα περιεχόμενα του καταχωρητή γίνονται 00011001 από το POR).

### Watch Dog Reset (WDR)

Ο χρονιστής του Watch Dog Reset εμφανίζεται όταν το περισσότερο σημαντικό bit του συγκεκριμένου 4-bit καταχωρητή μεταβαίνει από LOW σε HIGH. Γράφοντας οποιαδήποτε τιμή στον καταχωρητή γίνεται επανεκκίνηση του καταχωρητή στην θέση 0X21.

### USB Bus Reset

Ο USB ελεγκτής αναγνωρίζει μια επανεκκίνηση όταν ένα Single Ended Zero (SEO) παραμένει το λιγότερο για 8-16 μs. SEO, είναι η κατάσταση στην οποία και το D+ και το D- είναι σε χαμηλό δυναμικό (Low). Το πέμπτο bit του Status and Control καταχωρητή αναλαμβάνει να καταγράψει αυτό το συμβάν. Εάν κάποια επανεκκίνηση εμφανιστεί όσο η συσκευή βρίσκεται σε κατάσταση αναμονής, η κατάσταση αναμονής σταματά και ο ταλαντωτής του ρολογιού επανεκκινείται.

### Instant-on Feature (Suspend Mode)

Ο μικροελεγκτής μπορεί να τεθεί σε κατάσταση χαμηλής κατανάλωσης, αλλάζοντας το τρίτο bit του καταχωρητή Status and Control. Η κατάσταση αναμονής τερματίζεται όταν μια από τις επόμενες τρεις καταστάσεις προκύψει.

1. Δραστηριότητα της USB.



2. GPIO interrupt
3. Cext interrupt

### On-chip Timer

Ο USB ελεγκτής είναι εξοπλισμένος με ένα χρονιστή συχνότητας ίσης με το ένα έκτο της συχνότητας του κρυστάλλου του κυκλώματος. Τα bit 0 έως 7 του μετρητή ελέγχονται από τον καταχωρητή του Timer ο οποίος βρίσκεται στην διεύθυνση 0x23. Ο Timer καταχωρητής καθαρίζεται κατά την διάρκεια μιας Power-On επανεκκίνησης και κάθε φορά που η λειτουργία χαμηλής κατανάλωσης ενεργοποιείται. Η παρακάτω απεικόνιση δείχνει την διαμόρφωση του καταχωρητή και η δεύτερη απεικόνιση δείχνει το block διάγραμμα του. Με ένα κρύσταλλο 6 Mhz, η ανάλυση του Timer είναι 1 μs. (1 Mhz). Ο χρονιστής δημιουργεί δυο διακοπές: κάθε 128 μs και κάθε 1024 μs.

B7	B6	B5	B4	B3	B2	B1	B0
T.7	T.6	T.5	T.4	T.3	T.2	T.1	T.0
R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0

Πίνακας 2.5. Καταχωρητής του Χρονιστή-Timer (Διεύθυνση 0x23).

### Γενικής χρήσης I/O θύρες

Η επικοινωνία με περιφερειακά επιτυγχάνετε μέσω 12 ακροδεκτών (General Purpose I/O Ports). Αυτοί οι ακροδέκτες χωρίζονται σε δυο πόρτες. Την πόρτα P0 και την πόρτα P1. Η P0 αποτελείται από 8 ακροδέκτες και η P1 από 4. Οι δυο πόρτες μπορούν να ελεγχθούν από τις εντολές IORD, IOWR και IODX. Ο καταχωρητής δεδομένων της πόρτας P0 στην I/O διεύθυνση 0x00 και η αντίστοιχη διεύθυνση του καταχωρητή της P1 βρίσκεται στη διεύθυνση 0x01. Τα περιεχόμενα και των δυο καταχωρητών μεταβαίνουν σε κατάσταση High κατά την διάρκεια μιας επανεκκίνησης. Στους πίνακες 2.6. και 2.7. απεικονίζονται οι διαμορφώσεις των καταχωρητών δεδομένων των δυο πορτών.

B7	B6	B5	B4	B3	B2	B1	B0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

Πίνακας 2.6. Καταχωρητής Δεδομένων της Πόρτας 0 (Port 0 Data Register).

B3	B2	B1	B0
P1.3	P1.2	P1.1	P1.0
R/W	R/W	R/W	R/W
1	1	1	1

Πίνακας 2.7. Καταχωρητής Δεδομένων της Πόρτας 1 (Port 1 Data Register).

Κάθε ακροδέκτης περιέχει μια εσωτερική αντίσταση  $R_{up}$ . Ο καταχωρητής παρέχει δυο ιδιότητες, την λειτουργία Pull-up και την λειτουργία slew control. Δυο γεγονότα είναι αρμόδια για την κατάσταση κάθε αντίστασης και αυτά είναι τα παρακάτω:

Η κατάσταση του καταχωρητή Pull-Up καθώς και η κατάσταση του καταχωρητή δεδομένων της εκάστοτε πάντα πόρτας.

Σημείωση: Τα bits ελέγχου στο καταχωρητή Pull up της θύρας είναι ενεργά στο Low.

Ένας ακροδέκτης είναι High κάθε φορά που γράφεται ένας "1" στον καταχωρητή δεδομένων και ένα "0" στην αντίστοιχη θύρα του καταχωρητή Pull-Up. Γράφοντας ένα "0" στον καταχωρητή δεδομένων απενεργοποιείται ο καταχωρητής Pull-Up της πόρτας

και η έξοδος μεταβαίνει σε κατάσταση Low, ανεξάρτητα από τη ρύθμιση στον καταχωρητή Pull-Up. Η έξοδος μεταβαίνει σε κατάσταση High Z εάν το bit του καταχωρητή δεδομένων καθώς και το bit του καταχωρητή της πόρτας Pull-Up είναι και τα δυο “1”. Ο καταχωρητής της πόρτας Sink χρησιμοποιείται για να ελέγξει την στάθμη του ρεύματος εξόδου. Η επεξήγηση του γίνεται αργότερα σε αυτή τη παράγραφο.

Σημείωση: Το λογικό διάγραμμα του I sink απενεργοποιείται όταν βρισκόμαστε σε κατάσταση αναμονής.

Επομένως για να αποτραπουν υψηλά ρεύματα κατά την διάρκεια κατάστασης αναμονής, ο κώδικας (firmware) θα πρέπει να περιέχει ρουτίνες που να θέτουν όλα τα bit των καταχωριστών δεδομένων P0 και P1 (τα οποία δεν οδηγούνται εξωτερικά σε μια γνωστή κατάσταση) **σε “1”**. Επίσης θα πρέπει όλα τα Bit των καταχωρητών Pull-Up (και στις δυο πόρτες) να τεθούν **σε “0”** για να ενεργοποιηθούν οι θύρες pull-ups πριν τεθεί το Suspend bit (το τρίτο bit του Status and Control καταχωρητή). Παρακάτω απεικονίζεται ο πίνακας αληθείας για τον έλεγχο της εξόδου.

Data Register	Port Pull-Up Register	Output at I/O Pin	Interrupt Polarity
0	0	Sink Current ('0')	High to Low
0	1	Sink Current ('0')	Low to High
1	0	Pull-Up Resistor ('1')	High to Low
1	1	Hi-Z	Low to High

Πίνακας 2.8. Πίνακας αληθείας για τον έλεγχο της εξόδου κάθε ακροδέκτη.

Για να καθορίσεις ένα ακροδέκτη ως είσοδο, ένας “1” πρέπει να γραφτεί στο bit του καταχωρητή δεδομένων, για να απενεργοποιήσει την λειτουργία pull-down του Isink DAC. Όταν η πόρτα του καταχωρητή δεδομένων διαβάζεται, η τιμή του Bit είναι ένας “1” εάν το δυναμικό πάνω στο ακροδέκτη είναι μεγαλύτερο από την τάση κατωφλίου του Schmitt trigger, ή “0” εάν είναι κάτω από αυτό. Στις εφαρμογές όπου μια εσωτερική pull-up απαιτείται, η αντίσταση pull up μπορεί να δεσμευθεί γράφοντας ένα “0” στο κατάλληλο Bit του καταχωρητή Pull-up της πόρτας. Ο καταχωρητής Pull-up της P0 και της P1 είναι μόνο για «γράψιμο» (write only) δεξ τις παρακάτω φωτογραφίες).

Ο καταχωρητής Pull-Up της πόρτας P0 βρίσκεται στην I/O διεύθυνση 0x08 και ο αντίστοιχος καταχωρητής της πόρτας P1 βρίσκεται στην διεύθυνση 0x09. Τα περιεχόμενα των καταχωρητών Pull-Up της πόρτας καθαρίζονται κατά την διάρκεια μιας επανεκκίνησης, επιτρέποντας έτσι στις εξόδους να ελέγχονται από την κατάσταση των καταχωρητών δεδομένων. Οι καταχωρητές της πόρτας Pull-Up επιπλέον επιλέγουν την πολικότητα της μετάβασης που δημιουργεί μια διακοπή σε ένα ακροδέκτη (GPIO Interrupt). Ένα “0” επιλέγει μια μετάβαση από High σε Low ενώ ένας “1” επιλέγει μια μετάβαση από Low σε High.

B7	B6	B5	B4	B3	B2	B1	B0
Pull0.7	Pull0.6	Pull0.5	Pull0.4	Pull0.3	Pull0.2	Pull0.1	Pull0.0
W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0

Πίνακας 2.9. Pull-Up καταχωρητής της Θύρας 0.

B3	B2	B1	B0
P1.3	P1.2	P1.1	P1.0
W	W	W	W
0	0	0	0

Πίνακας 2.10. Pull-Up καταχωρητής της Θύρας 1.

Γράφοντας ένα “0” στον καταχωρητή δεδομένων οδηγείτε η έξοδος σε κατάσταση Low. Αντ’ αυτού στο να παρέχει μια μεταβαλλόμενη έξοδο, ο USB ελεγκτής επιτρέπει στο χρήστη να επιλέξει ένα μεταβαλλόμενο ρεύμα για κάθε I/O ακροδέκτη. Το ρεύμα οδήγησης κάθε εξόδου ελέγχεται από ένα καταχωρητή, τον Port Isink. Τα τέσσερα λιγότερα σημαντικά bit αυτού του καταχωρητή είναι αρμόδια για την δημιουργία 16 διαφορετικών σταθμών ρεύματος. Τα τέσσερα περισσότερα σημαντικά bit του καταχωρητή αγνοούνται. Η P0 είναι πόρτα χαμηλού ρεύματος κατάλληλη για σύνδεση Φώτο-τρανζίστορ. Η P1 είναι πόρτα υψηλού ρεύματος ικανή να οδηγήσει led. Η μικρότερη τιμή του ρεύματος είναι 0000 και η μεγαλύτερη τιμή είναι 1111.

### Xtalin-Xtalout

Οι ακροδέκτες Xtalin και Xtalout υποστηρίζουν τη σύνδεση με κρύσταλλο συχνότητας 6 MHz. Οι πυκνωτές παράκαμψης και οι αντιστάσεις πόλωσης είναι εσωτερικά στο Ic, όπως φαίνεται και στην εικόνα 5-14. Πρέπει να σημειωθεί πως όταν οδηγούμε το Xtalin από ένα εξωτερικό ταλαντωτή πρέπει να αφήσουμε το Xtalout ασύνδετο.

### Εξυπηρέτηση ρουτίνας διακοπών - Interrupt

Οι διακοπές δημιουργούνται από της γενικής χρήσης I/O γραμμές, τον ακροδέκτη Cext, τον εσωτερικό χρονιστή και την μηχανή USB. Όλες οι διακοπές ελέγχονται από το Global Enable Interrupt καταχωρητή. Πρόσβαση σε αυτόν τον καταχωρητή γίνεται μέσω των εντολών IORD, IOWR και IOWX στην διεύθυνση 0x20. Γράφοντας ένα “1” σε ένα συγκεκριμένο bit ενεργοποιεί την διακοπή που συσχετίζεται με αυτή τη θέση. Κατά την διάρκεια μιας επανεκκίνησης, τα περιεχόμενα του καταχωρητή καθαρίζονται, απενεργοποιώντας έτσι όλες τις διακοπές.

B7	B6	B5	B4	B3	B2	B1	B0
CEXTIE	GPIOIE	Reserved	EP1IE	EP0IE	1024IE	128IE	Reserved
R/W	R/W		R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0

Πίνακας 2.11. Global Interrupt Enable Register (GIER- Address 0x20).

### GPIO Interrupt

Οι γενικοί χρήσεις I/O διακοπές (General Purpose Input Output- GPIO) δημιουργούνται από μεταβάσεις του σήματος στους ακροδέκτες των πορτών P0 και P1. Οι GPIO διακοπές είναι ευαίσθητες στην ακμή, επίσης έχουμε την δυνατότητα να προγραμματίσουμε τις πολικότητες των διακοπών. Θέτοντας ένα bit «High» στον καταχωρητή Pull-Up της πόρτας επιλέγουμε ένα «Low σε High» σκανδαλισμό για τον αντίστοιχο ακροδέκτη. Θέτοντας ένα Bit «Low» ενεργοποιούμε ένα «High σε Low» σκανδαλισμό διακοπής. Κάθε GPIO διακοπή αποκτά πρόσβαση αφού ενεργοποιηθεί το συγκεκριμένο Bit στον καταχωρητή Port Interrupt Enable. Αυτοί οι καταχωρητές καθαρίζονται κατά την διάρκεια μιας επανεκκίνησης, απενεργοποιώντας έτσι όλες τις GPIO διακοπές. Εάν ένας ακροδέκτης μιας πόρτας προκαλέσει ένα interrupt, κανένας άλλος ακροδέκτης δεν μπορεί να προκαλέσει μια GPIO διακοπή μέχρι ο ακροδέκτης που είχε προκαλέσει τη διακοπή επανέλθει στην ανενεργή κατάσταση (non-trigger).

### USB Interrupt

Μια διακοπή του τελικού σημείου 0 δημιουργείτε αφού ο ξενιστής γράψει δεδομένα στο τελικό σημείο 0 ή αφού ο USB ελεγκτής έχει μεταδώσει ένα πακέτο από το

τελικό σημείο 0 και λαμβάνει μια (ACK) από τον ξενιστή)). Μια διακοπή του τελικού σημείου 1 δημιουργείται αφού ο USB ελεγκτής έχει μεταδώσει ένα πακέτο από το τελικό σημείο 1 και έχει λάβει μια (ACK) από το ξενιστή.

### Timer Interrupt

Υπάρχουν δυο είδη διακοπών χρονιστή: Η διακοπή 128μs και η διακοπή 1024ms. Ελέγχονται από τα bit 1 και 2 του Global Interrupt Enable αντίστοιχα.

### Wake-Up Interrupt

Μια Wake-Up διακοπή δημιουργείται όταν ο Cext ακροδέκτης μεταβαίνει σε υψηλή κατάσταση. Μπορεί να ελεγχθεί από το Bit 7 του Global Enable Interrupt καταχωρητή.

Αυτή η διακοπή μπορεί να χρησιμοποιηθεί για να δημιουργήσει περιοδικούς ελέγχους σε περιφερειακά όταν ο USB ελεγκτής βρίσκεται σε λειτουργία χαμηλής κατανάλωσης.

### USB Engine

Η USB “μηχανή” περιέχει μια σειριακή μηχανή επικοινωνίας την (Serial Interface Engine- S.I.E.) και τους χαμηλής ταχύτητας USB I/O πομποδέκτες. Η S.I.E. εκτελεί τις περισσότερες λειτουργίες με μικρή μόνο υποστήριξη από το πυρήνα του μικροελεγκτή. Υποστηρίζονται δυο τελικά σημεία. Το τελικό σημείο 0 χρησιμοποιείται για να λάβει και να μεταδώσει πακέτου ελέγχου (συμπεριλαμβανομένου και πακέτων Setup), ενώ το τελικό σημείο 1 χρησιμοποιείται μόνο για να μεταδώσει πακέτα δεδομένων.

### USB Enumeration Process

Ο USB ελεγκτής παρέχει έναν καταχωρητή, τον USB Device Address Register στη διεύθυνση 0x12. Το να γράφεις και να διαβάζεις σε αυτό το καταχωρητή πετυχαίνετε μέσω των εντολών IORD και IOWR. Τα περιεχόμενα του καταχωρητή καθαρίζονται κατά την διάρκεια μιας επανεκκίνησης, θέτοντας την USB διεύθυνση του ελεγκτή στο 0 (ADR0).

B7	B6	B5	B4	B3	B2	B1	B0
Reserved	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Πίνακας 2.12. USB Device Address Register (USB DA – Address 0x12).

### Endpoint 0

Όλες οι USB συσκευές πρέπει να έχουν ένα τελικό σημείο 0 το οποίο θα χρησιμοποιείτε για αρχικοποίηση και χειρισμό της συσκευής. Το τελικό σημείο 0 παρέχει πρόσβαση στις πληροφορίες ρυθμίσεων (configuration) της συσκευής και επιτρέπει προσβάσεις ελέγχου. Το τελικό σημείο 0 μπορεί να λάβει και να μεταδώσει δεδομένα, και στις δυο περιπτώσεις τα δεδομένα μοιράζονται το ίδιο 8-byte FIFO που βρίσκεται στην θέση μνήμης 0x70 με 0x77. Τα λαμβανόμενα δεδομένα μπορούν να επικαλύψουν τα δεδομένα στο FIFO.

### Endpoint 0 Receive

Αφού ληφθεί ένα πακέτο και τοποθετηθούν τα δεδομένα μέσα στο FIFO του τελικού σημείου, ο ελεγκτής αναπροσαρμόζει τον καταχωρητή RX του τελικού σημείου

0 για να καταγράψει τα λαμβανόμενα στοιχεία και κατόπιν προκαλεί ένα interrupt (του τελικού σημείου 0).

Η διαμόρφωση του καταχωρητή RX του τελικού σημείου 0 απεικονίζεται στον παρακάτω πίνακα.

B7	B6	B5	B4	B3	B2	B1	B0
COUNT3	COUNT2	COUNT1	COUNT0	TOGGLE	IN	OUT	SETUP
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Πίνακας 2.13. Καταχωρητής RX του τελικού σημείου 0.

Αυτός ο καταχωρητής βρίσκεται στην I/O διεύθυνση 0x14. Οτιδήποτε γραφτεί σε αυτόν τον καταχωρητή έχει ως αποτέλεσμα να καθαριστούν όλα τα bit εκτός από το bit 3 το οποίο παραμένει αμετάβλητο. Επίσης όλα τα bit καθαρίζονται κατά την διάρκεια επανεκκίνησης. Το bit 0 τίθεται 1 όταν ένα SETUP token για το τελικό σημείο 0 λαμβάνεται. Όταν τεθεί 1 για μια φορά αυτό το bit παραμένει High μέχρι να καθαριστεί από μια I/O εγγραφή ή μια επανεκκίνηση.

### Endpoint 0 Transmit

Ο καταχωρητής TX του τελικού σημείου 0 βρίσκεται στην I/O διεύθυνση 0x10. Ο καταχωρητής αυτός είναι υπεύθυνος για τον έλεγχο της μετάδοσης δεδομένων από το τελικό σημείο 0.

B7	B6	B5	B4	B3	B2	B1	B0
INEN	DATA1/0	STALL	ERR	COUNT3	COUNT2	COUNT1	COUNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Πίνακας 2.14. Καταχωρητής TX Του Τελικού Σημείου 0.

Τα Bit 0 έως 3 δείχνουν τον αριθμό των bytes δεδομένων τα οποία μεταδίδονται κατά την διάρκεια ενός IN πακέτου, έγκυρες τιμές θεωρούνται από 0 έως 8 αποκλειστικά. Το bit 4 δείχνει ότι έγινε σφάλμα κατά τη διάρκεια μετάδοσης ενός πακέτου δεδομένων (CRC, PID, ή bitstuffing λάθος.. Το bit 5 καθαρίζεται οποτεδήποτε ένα πακέτο SETUP έχει ληφθεί από το τελικό σημείο 0. Το Bit 6 (Data 1/0) πρέπει να τεθεί 0 ή 1 για να επιλεγεί η εναλλακτική (toggle) κατάσταση του πακέτου δεδομένων (0 για Data0 και 1 για Data1). Αφού τα μεταδιδόμενα δεδομένα φορτωθούν μέσα στο FIFO, το Bit 6 πρέπει να τεθεί σύμφωνα με την κατάσταση των δεδομένων και το bit 7 πρέπει να τεθεί ίσο με "1". Αυτό επιτρέπει στο ελεγκτή να ανταποκριθεί σε ένα IN πακέτο. Το bit 7 επίσης καθαρίζεται όταν ένα SETUP πακέτο ληφθεί. Η Ρουτίνα διακοπής εξυπηρέτησης μπορεί να ελέγξει το Bit 7 για να επιβεβαιωθεί ότι η μεταφορά δεδομένων ήταν επιτυχής.

### Endpoint 1

Το τελικό σημείο 1 είναι ικανό μόνο για μετάδοση. Τα δεδομένα τα οποία πρόκειται να μεταδοθούν αποθηκεύονται μέσα στο 8-byte FIFO που βρίσκεται στη μνήμη προγράμματος δεδομένων 0x78 έως 0x7F.

### Endpoint 1 Transmit

Η μετάδοση ελέγχεται από τον καταχωρητή TX του τελικού σημείου 1 και βρίσκεται στη I/O διεύθυνση 0x11 (εικόνα 5-23). Τα bit από 0 έως και το 3 δείχνουν τον αριθμό των bytes που μεταδίδονται κατά την διάρκεια ενός IN πακέτου

B7	B6	B5	B4	B3	B2	B1	B0
INEN	DATA1/0	STALL	EP1EN	COUNT3	COUNT2	COUNT1	COUNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Πίνακας 2.15. USB Endpoint 1 TX Configuration Register ( Address 0x11).

. Έγκυρες τιμές είναι από 0 μέχρι 8 αποκλειστικά. Το bit 4 πρέπει να τεθεί “1” πριν χρησιμοποιηθεί το τελικό σημείο 1. Εάν αυτό το bit είναι καθαρισμένο, ο ελεγκτής αγνοεί όλη τη επικοινωνία στο τελικό σημείο 1. Θέτοντας το Stall bit 5 stalls IN και OUT πακέτα μέχρι αυτό το bit να καθαριστεί. Το bit 6 πρέπει να τεθεί 1 ή 0 ανάλογα με την κατάσταση του πακέτου δεδομένων. Μετά την ολοκλήρωση της μεταφοράς δεδομένων μέσα στο FIFO, το bit 6 πρέπει να τεθεί ανάλογα με την κατάσταση των δεδομένων, καθώς και το bit 7 πρέπει να τεθεί σε κατάσταση ‘1’. Αυτό επιτρέπει στον USB ελεγκτή να ανταποκριθεί σε ένα IN πακέτο.

### USB Physical Layer Characteristics

#### Low Speed Driver Characteristics

Ο μικροελεγκτής CY7C63001 χρησιμοποιεί ένα διαφορικό οδηγό εξόδου για να οδηγήσει τα χαμηλής ταχύτητας δεδομένα μέσα από το USB καλώδιο. Η ταλάντωση μεταξύ των διαφορικών High και Low καταστάσεων είναι καλά ισορροπημένη έτσι ώστε να μειώνεται στο ελάχιστο η παραμόρφωση μεταξύ των δυο σημάτων. Ο cy7c63001 έχει την ικανότητα να ανεκτή τάση πάνω στους ακροδέκτες του στα όρια (-0.5 έως 3.8 volt). Ένα καλώδιο USB χαμηλής ταχύτητας είναι Unshielded Μη στρεφόμενο (untwisted) καλώδιο μέγιστου μήκους τριών μέτρων. Οι χρόνοι ανόδου και καθόδου των σημάτων σε αυτό το καλώδιο ελέγχονται με τέτοιο τρόπο έτσι ώστε να μειώνονται οι ηλεκτρομαγνητικές παρεμβολές. Αυτό το καλώδιο προορίζεται μόνο για την επικοινωνία μεταξύ συσκευών χαμηλών ταχυτήτων.

#### Receiver Characteristics.

Ο cy7c63001 έχει στην είσοδο του ένα διαφορικό δέκτη ο οποίος είναι ικανός να δέχεται σήματα δεδομένων της USB. Ο δέκτης έχει ευαισθησία εισόδου το λιγότερο 200 mV όταν και οι δυο διαφορικές εισοδοί βρίσκονται μεταξύ των ορίων (0.8 Volt -2.5 Volt).

#### External USB Pull-up Resistor

Το USB σύστημα διευκρινίζει πως μια pull-up αντίσταση πρέπει να συνδεθεί μεταξύ του ακροδέκτη D- και της VCC έτσι ώστε να πληρούνται οι προδιαγραφές της USB1.1.

Κάτι επίσης που θα πρέπει να έχει στα υπόψη του ο κατασκευαστής της συσκευής για να πληρούνται οι προδιαγραφές της USB 1.1, είναι πως η συνολική χωρητικότητα στις D+ και D- γραμμές επικοινωνίας του περιφερειακού χαμηλής ταχύτητας (Χωρητικότητα του μικροεπεξεργαστή + Χωρητικότητα της πλακέτας που είναι ο μικροελεγκτής + Χωρητικότητα καλωδίων) πρέπει να είναι μικρότερη από 250 pF. Ο πομποδέκτης του μικροελεγκτή έχει μέση τιμή χωρητικότητας στην είσοδο του της τάξης των 20 pF. Έτσι η συνολική επιτρεπτή χωρητικότητα που θα έχουν το καλώδιο και η πλακέτα δεν πρέπει να ξεπερνούν τα 230 pF. Εάν τώρα η συνολική χωρητικότητα ξεπερνάει τα 230 pF θα πρέπει να χρησιμοποιηθεί ένας σταθεροποιητής 3.3 Volt.

### 3. Βηματικοί κινητήρες - τρόποι οδήγησης

Οι βηματικοί κινητήρες (stepper motors) αποτελούν αναπόσπαστο κομμάτι σχεδόν κάθε ρομποτικού μηχανισμού, αφού με αυτούς επιτυγχάνεται ο ακριβής έλεγχος της κίνησης των διαφόρων μηχανικών μερών. Μια αυτόματη εργαστηριακή ή βιομηχανική διάταξη περιλαμβάνει συνήθως και κινητά μέρη, των οποίων η θέση ή η γωνία πρέπει να ρυθμίζεται λεπτομερώς. Τέτοιες ρυθμίσεις μπορούν να γίνουν με τη βοήθεια βηματικών κινητήρων. Σε πολλές περιπτώσεις βέβαια χρησιμοποιούνται και σερβοκινητήρες.

Σαν ένα παράδειγμα αυτοματισμού, που η κίνηση του στηρίζεται σε βηματικούς κινητήρες, μπορεί να αναφέρει κανείς ένα ρομποτικό βραχίονα, στον οποίο η θέση των διάφορων μηχανικών αρθρώσεων ρυθμίζεται με μεγάλη ακρίβεια με τη βοήθεια ενός αριθμού βηματικών κινητήρων. Άλλο σύνηθες παράδειγμα είναι ο έλεγχος της θέσης του βραχίονα ανάγνωσης ενός οδηγού μαλακού ή σκληρού δίσκου. Οι οδηγοί των οπτικών δίσκων (CD-ROM) και οι οπτικοί σαρωτές επίσης περιέχουν βηματικούς κινητήρες. Έτσι μια πηγή βηματικών κινητήρων αποτελούν οι παλαιοί οδηγοί δίσκων, οι οποίοι έχουν περιπέσει σε αχρηστία

Με την κατάλληλη λογική ελέγχου, ένα βηματικός κινητήρας μπορεί να αποδώσει ικανοποιητική ροπή και ταχύτητα περιστροφής σε δύσκολα φορτία, με μεγάλη ακρίβεια, χωρίς απώλεια βημάτων. Αυτό όμως προϋποθέτει μια βασική κατανόηση των διαφόρων τρόπων λειτουργίας και οδήγησης τους. Επειδή ο βηματικός κινητήρας περιστρέφεται με εναλλαγή παλμών στα τυλίγματα του, ο καταλληλότερος τρόπος για τον έλεγχο του είναι η ψηφιακή λογική. Στα επόμενα θα παρουσιάσουμε τις βασικές αρχές λειτουργίας των πιο συνηθισμένων βηματικών κινητήρων.



Σχήμα 3.1. Βηματικοί κινητήρες από έναν εκτυπωτή inject.

#### 3.1. Αρχές λειτουργίας βηματικών κινητήρων

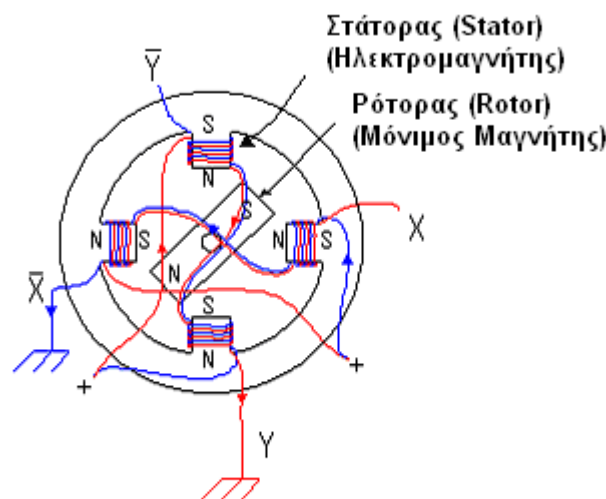
Ο βηματικός κινητήρας είναι κινητήρας συνεχούς ρεύματος διότι λειτουργεί με παλμούς συνεχής τάσης (dc) που εμφανίζονται στα τυλίγματα του με τον κατάλληλο συγχρονισμό και την κατάλληλη διαδοχή. Η διαδοχή των παλμών έχει σαν αποτέλεσμα ένα κατάλληλα «περιστρεφόμενο» μαγνητικό πεδίο στον στάτορα, με τη βοήθεια του οποίου στρέφεται ο άξονας του κινητήρα κατά μια ορισμένη γωνία με πολύ μεγάλη ακρίβεια.

Οι βηματικοί κινητήρες είναι ηλεκτρομηχανικοί κινητήρες οι οποίοι μετατρέπουν ηλεκτρικούς παλμούς σε μηχανική κίνηση, και πιο συγκεκριμένα σε καθορισμένα

διακριτά βήματα. Ως **βήμα** ορίζουμε την γωνιακή μετατόπιση που δημιουργείται στον άξονα του κινητήρα κάθε φορά που δέχεται ένα παλμό.

Οι συνηθέστεροι βηματικοί κινητήρες έχουν κυρίως 4 ή 6 ακροδέκτες. Αλλά σε πολλές περιπτώσεις μπορούμε να βρούμε και να χρησιμοποιήσουμε βηματικούς κινητήρες με 5 ή και με 8 περιελίξεις. Αν και γενικά αναφερόμαστε στις περιελίξεις ενός βηματικού κινητήρα, αυτό συνιστά μια απλοποίηση. Η αλήθεια είναι πως κάθε πηνίο είναι φτιαγμένο από πολλές μικρότερες περιελίξεις, συνδεδεμένες στη σειρά και τοποθετημένες σε κύκλο.

Η βασική αρχή λειτουργίας των βηματικών κινητήρων είναι ότι ο δρομέας, είτε με μόνιμους, είτε με πρόσκαιρους μαγνητικούς πόλους, εξαναγκάζεται να στραφεί ανταποκρινόμενος στο βηματικά στρεφόμενο πεδίο που παρέχεται από τα τυλίγματα του στάτη. Κάθε βηματικός κινητήρας έχει τις δικές του απαιτήσεις σε τάση τροφοδοσίας και σε ρεύμα ανά τύλιγμα. Για την καλύτερη κατανόηση θα αναφερθούμε σε ένα τετραπολικό κινητήρα (τέσσερις πόλους). Σχήμα 3.2. Οι πόλοι του στάτορα γενικά είναι αυτοί που καθορίζουν το βήμα (γωνιακή μετατόπιση) του ρότορα. Στη συνέχεια γίνεται η απαραίτητη διευκρίνιση της λειτουργίας τους.



Σχήμα 3.2. Τετραπολικός βηματικός κινητήρας με βήμα 90 μοιρών.

Στους τετραπολικούς κινητήρες, κάθε δυο απέναντι τυλίγματα λειτουργούν σαν ζευγάρια. Δηλαδή το πηνίο  $X$  και το πηνίο  $\bar{X}$ , όπως και το  $Y$  με το  $\bar{Y}$  λειτουργούν ταυτόχρονα με ανάποδη όμως μαγνητική ροή. Για παράδειγμα τα πηνία  $Y$  και  $\bar{Y}$  βρίσκονται στον κάτω και στον επάνω πόλο, και είναι τοποθετημένα έτσι ώστε όταν περάσει ρεύμα από μέσα τους να παράγουν αντίθετη μαγνητική ροή. Το ίδιο ισχύει και για τα άλλα δυο πηνία. Η περιστροφή του κινητήρα, γίνεται με τη ροή ρεύματος μέσα από τα πηνία  $X$ ,  $\bar{X}$ ,  $Y$  και  $\bar{Y}$ . (Βλέπε σχήμα 3.2). Ο ρότορας του βηματικού κινητήρα μένει σταθερός ανάμεσα σε δυο πόλους του στάτορα. Όταν ένα ζευγάρι πηνίων του στάτορα, πάρει ρεύμα τότε δημιουργείται ηλεκτρομαγνητικό πεδίο που επιδρά στο μόνιμο μαγνήτη του ρότορα, και παράγεται κίνηση του ρότορα κατά ένα βήμα. Ένα χαρακτηριστικό των βηματικών κινητήρων είναι πως τα βήματα της κίνησης τους μπορούν να ελεγχθούν με μεγάλη ακρίβεια. Επίσης η ροπή που παρουσιάζουν είναι αρκετά μεγάλη έτσι ώστε να υπάρχει η δυνατότητα σταματήματος της περιστροφής σε οποιοδήποτε βήμα (γωνία). Αναλυτικότερα, το ρεύμα, εφαρμόζεται επιτυχώς στην κάθε



περιέλιξη του στάτορα, δημιουργεί μια σειρά μαγνητικών πεδίων. Ο ρότορας, αλληλεπιδρώντας με τη μαγνητική αντίδραση του πεδίου, ευθυγραμμίζεται καθώς κάθε περιέλιξη ενεργοποιείται. Κάθε επιτυχημένη επανευθυγράμμιση, παράγει μια σταδιακή κίνηση του ρότορα, σ' αυτή την περίπτωση βήματα  $90^\circ$ , αφού ο πόλοι του στάτορα βρίσκονται σε απόσταση  $90^\circ$  μεταξύ τους. Με μια αλλαγή στην ακολουθία διέγερσης στις περιελίξεις, ο κινητήρας μπορεί να λειτουργήσει δεξιόστροφα ή αριστερόστροφα. Εξαιτίας της αντίδρασης του ρότορα και τη σχετικά μεγάλη γωνία βήματος του PM, ο ρότορας έχει λίγη κλίση για να υπερβεί. "**Υπέρβαση**", είναι η κατάσταση όταν ο ρότορας πάει πίσω από τον πόλο που τον έλκει. Ο πόλος, που ακόμα τον έλκει, τον τραβά πίσω. Στο σχήμα 3.2 φαίνεται ένας τετραπολικός βηματικός κινητήρας του οποίου το βήμα είναι  $90$  μοίρες. Η περιστροφή του κινητήρα, γίνεται με την ροή ρεύματος μέσα από τα πηνία  $X$ ,  $\bar{X}$ ,  $Y$  και  $\bar{Y}$ . Η ταχύτητα περιστροφής όπως και η φορά, ελέγχονται με συνδυασμό παλμικών ρευμάτων που ρέουν μέσα από τα δύο αυτά ζευγάρια πηνίων.

### 3.2. Κατηγορίες βηματικών κινητήρων

Οι βηματικοί κινητήρες χωρίζονται σε τρεις μεγάλες κατηγορίες οι οποίες είναι οι παρακάτω:

- **Βηματικοί κινητήρες μόνιμου μαγνήτη (Permanent Magnet Motors).**
- **Βηματικοί κινητήρες μεταβλητής μαγνητικής αντίστασης (Variable Reluctance Motors).**
- **Υβριδικοί βηματικοί κινητήρες (Hybrid Motors).**

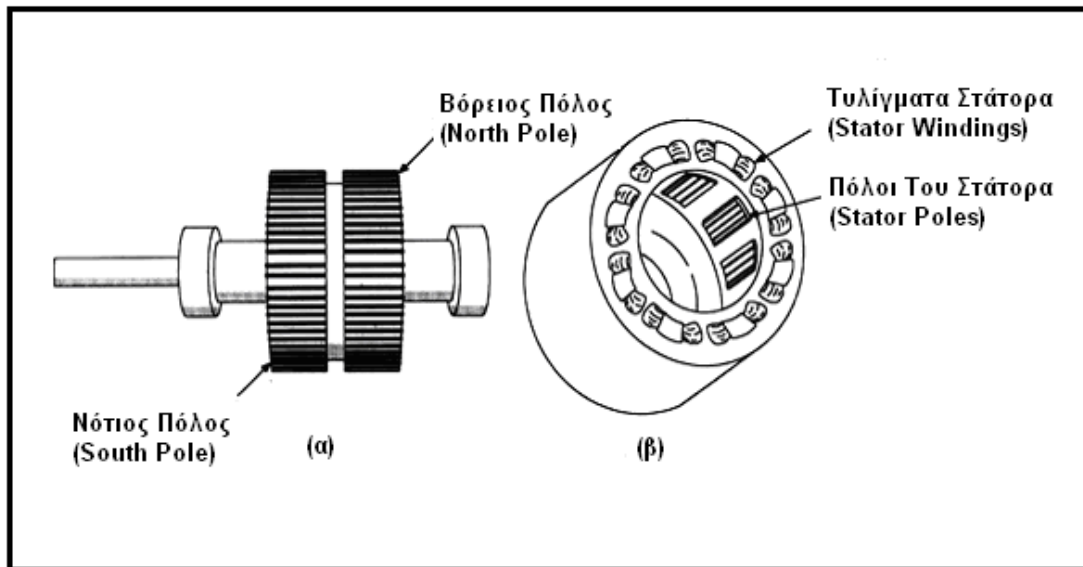
#### 3.3.1. Βηματικοί κινητήρες μόνιμου μαγνήτη (Permanent Magnet Motors-PM).

Οι **βηματικοί κινητήρες μόνιμου μαγνήτη** αποτελούνται από ένα μόνιμο μαγνήτη που λέγεται ρότορας και τυλίγματα (πηνία) στο σώμα του κινητήρα που λέγονται στάτορας. Ο αριθμός των δοντιών του στάτορα και του ρότορα καθορίζουν την γωνία κάθε βήματος που θα δημιουργηθεί κάθε φορά που η πολικότητα των τυλιγμάτων αναστρέφεται. Όσο μεγαλύτερος είναι ο αριθμός των δοντιών, τόσο μικρότερη θα είναι και η γωνία που δημιουργείται από ένα βήμα. Κάθε φορά που ένα DC σήμα παρέχεται σε ένα τύλιγμα του στάτορα, ο ρότορας θα υπερνικήσει την υπόλοιπη ροπή και θα κινηθεί προς μια κατεύθυνση. Η **holding ροπή** καθορίζεται ως το ποσοστό της ροπής που απαιτείται για να κινηθεί ο ρότορας ένα βήμα. Ένα σημαντικό χαρακτηριστικό του PM κινητήρα είναι ότι μπορεί να διατηρήσει την holding ροπή όταν ο ρότορας είναι σταματημένος. Όταν καμία δύναμη δεν εφαρμόζεται στα τυλίγματα, μια μικρή μαγνητική δύναμη αναπτύσσεται μεταξύ του μόνιμου μαγνήτη και του στάτορα. Αυτή η μαγνητική δύναμη καλείται **residual** ή **detent ροπή**. Η detent ροπή μπορεί να παρατηρηθεί με τη στροφή ενός βηματικού κινητήρα με το χέρι και είναι γενικά περίπου ίσο με το ένα δέκατο της holding ροπής. Στο παρακάτω σχήμα 3.2. φαίνονται τα εξαρτήματα που απαρτίζουν ένα PM βηματικό κινητήρα.

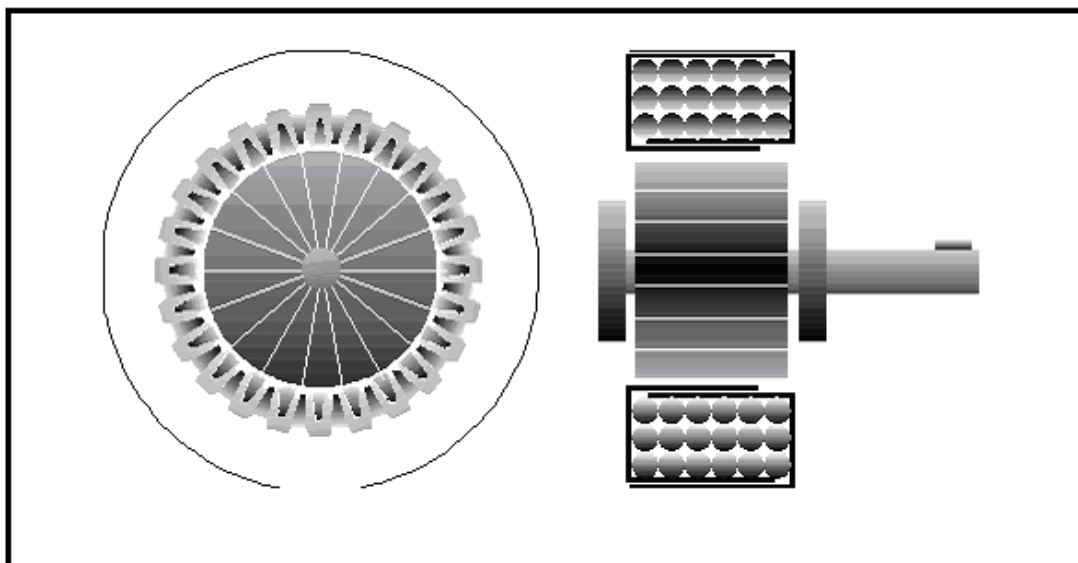
#### Παράμετροι βηματικού κινητήρα PM

Η σωστή ποσότητα τάσεων και ρευμάτων διέγερσης, πρέπει να τοποθετείται στις περιελίξεις του στάτορα. Ο ρότορας μπορεί να απομαγνητιστεί από πλεονάζουσα διέγερση. Οι καλύτερες πηγές για τη σωστή ποσότητα, βρίσκονται στα δεδομένα του

κατασκευαστή. Μερικοί απ' αυτούς, δίνουν την αντίσταση του ρότορα και τη μέγιστη ισχύ που μπορεί αυτός να καταναλώσει αντί για ρεύμα και τάση.



Σχήμα 3.3. Εξαρτήματα ενός PM βηματικού κινητήρα: (α) Ρότορας (β) Στάτορας.



Σχήμα 3.4. Διαγώνια απεικόνιση ενός βηματικού κινητήρα τύπου PM.

Η αδράνεια του ρότορα, είναι γενικά υψηλότερη στους PM από τους VR. κινητήρες. **Αδράνεια**, "είναι η αντίσταση που έχει ένα σώμα για να ξεκινήσει μια κίνηση ή να αλλάξει την κατεύθυνση μιας κίνησης. Είναι ανάλογη της μάζας του αντικειμένου, όσο περισσότερη η μάζα τόσο περισσότερη και η αδράνεια. Μετριέται σε γραμμάρια ανά τετραγωνικό μέτρο ( $\text{g}\cdot\text{m}^2$ ). Επίσης χρησιμοποιείται και η μονάδα κιλά ανά τετραγωνικά εκατοστά ( $\text{Kg}\cdot\text{cm}^2$ ). Οι PM κινητήρες επιτρέπουν υψηλότερες ταχύτητες ρότορα και αντίστροφα χαμηλότερες τιμές βημάτων.

Άλλη μια παράμετρος που φαίνεται συχνά στα δεδομένα του κατασκευαστή, είναι η γωνία βήματος. Αυτή είναι η συγκεκριμένη ποσότητα περιστροφής του κινητήρα σε μοίρες που προκαλείται από μια αλλαγή στην πολικότητα της περιτύλιξης. Γενικώς, είναι μεγαλύτερη στους βηματικούς κινητήρες PM και κυμαίνεται από  $0,72^\circ$  -  $90^\circ$  με πιο

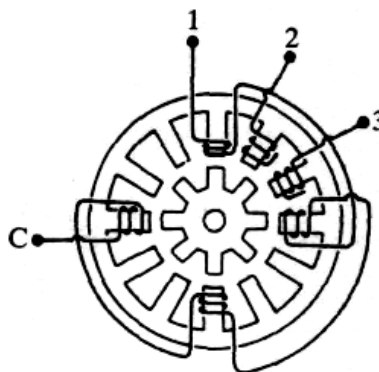
κοινές τις  $7,5^\circ$  - $18^\circ$ . Οι βηματικοί κινητήρες PM, έχουν συνήθως 12 ή 24 πόλους, επιτρέποντας γωνίες  $3,75^\circ$ ,  $7,5^\circ$  ή  $15^\circ$ . Κανονικά, μόνο  $45^\circ$  ή  $90^\circ$  γωνίες είναι δυνατές σε ρότορες δύο πόλων. Αύξηση στον αριθμό πόλων, έχει αποτέλεσμα μικρότερες γωνίες και υψηλότερες μέγιστες ταχύτητες βημάτων.

Στα δεδομένα του κατασκευαστή, δίνεται επίσης ο μέγιστος αριθμός βημάτων ανά δευτερόλεπτο (steps/sec). Αυτή η παράμετρος αντικαθιστά τη μέγιστη τιμή στροφών - ανά - λεπτό των DC και AC κινητήρων. Η μέγιστη ταχύτητα βημάτων είναι αντιστρόφως ανάλογη της μάζας (και αδράνειας) του ρότορα.

Οι βηματικοί κινητήρες PM, έχουν μια ενδιαφέρουσα και χρήσιμη παράμετρο που λέγεται "ροπή κράτησης". Παράγεται όταν οι περιελίξεις του στάτορα ενεργοποιούνται. Όταν ο στάτορας ενεργοποιηθεί και ο κινητήρας έχει προχωρήσει στη νέα του θέση, ο ρότορας θα κρατηθεί στη θέση του από την έλξη των δύο μαγνητικών πεδίων. Η ροπή κράτησης καθορίζεται σαν τη ροπή που χρειάζεται για να κινηθεί ο ρότορας ένα ολόκληρο βήμα, με το στάτορα ενεργοποιημένο αλλά σταθερό. Είναι γενικώς μεγαλύτερη από τη ροπή κίνησης και δρα σαν δυναμικός μηχανισμός φρένου στην κράτηση ενός φορτίου.

### 3.3.2. Βηματικοί κινητήρες μεταβλητής μαγνητικής αντίστασης (Variable Reluctance Motors-VR).

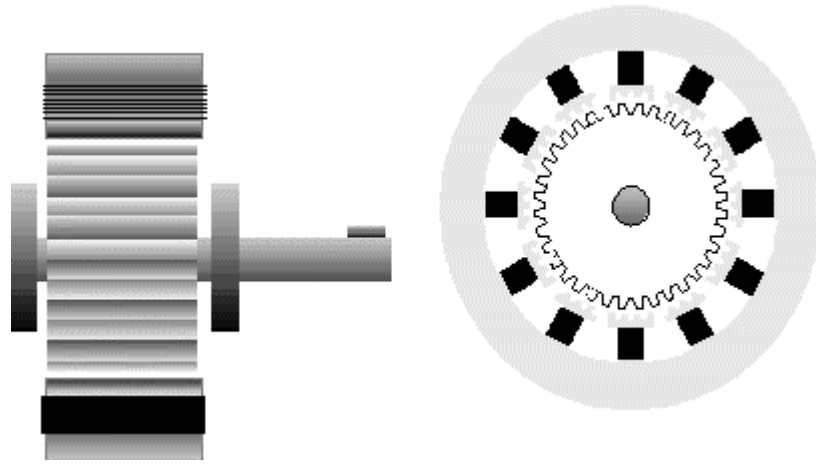
Στους βηματικούς κινητήρες μεταβλητής μαγνητικής αντίστασης ο **στάτορας με τον ρότορα έχουν διαφορετικό αριθμό δοντιών**. Ο ρότορας σε αντίθεση με τους βηματικούς κινητήρες τύπου PM δεν έχει καθόλου μόνιμο-μαγνήτη ρότορα και καμιά υπόλοιπη ροπή για να κρατήσει το ρότορα σε μια θέση όταν διακοπεί η τροφοδοσία τους. Όταν τα τυλίγματα του στάτορα ενεργοποιηθούν, τα δόντια του ρότορα θα ευθυγραμμιστούν με τους ενεργοποιημένους πόλους του στάτη. Με την εναλλαγή των τυλιγμάτων που ενεργοποιούνται στο στάτορα, το πεδίο του στάτορα αλλάζει, και ο στροφέας κινείται προς μια νέα θέση. Οι κινητήρες αυτοί, χρησιμοποιούν έναν, περιοδοντικό ρότορα με έναν ηλεκτρομαγνητικό στάτορα παρόμοιο με αυτόν του βηματικού κινητήρα PM. Στο Σχήμα 4.26. φαίνεται ένας τυπικός τριφασικός βηματικός κινητήρας VR με 12 πόλους στάτορα τοποθετημένους σε  $30^\circ$  μεταξύ τους, ενώ ο ρότορας έχει οκτώ πόλους σε διαστήματα  $45^\circ$ . Οι πόλοι του στάτορα, ενεργοποιούνται σταδιακά από την τριφασική περιέλιξη. Όταν το ρεύμα εφαρμοστεί στη φάση 1, τα δόντια του ρότορα που είναι πιο κοντά στους 4 ενεργοποιημένους πόλους ευθυγραμμίζονται.



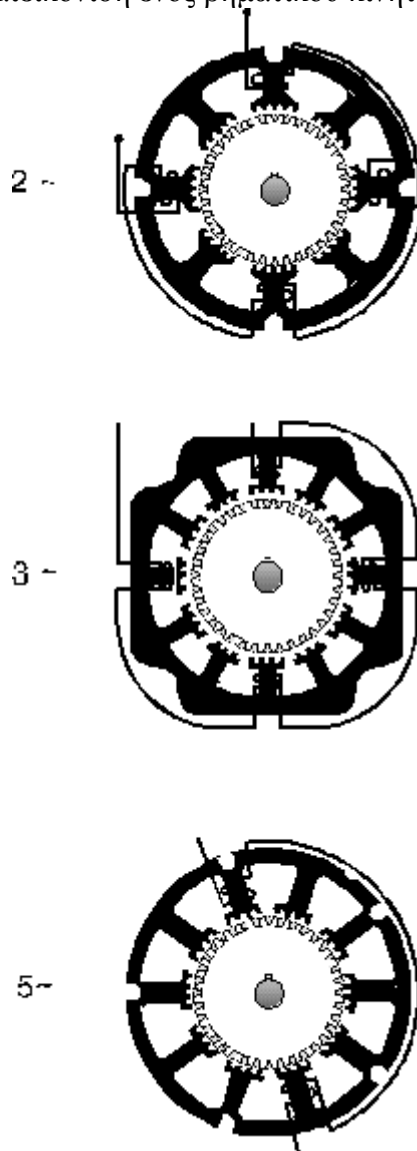
Σχήμα 3.5. Απεικόνιση βηματικού κινητήρα τύπου VR.

Τα τέσσερα εναπομείναντα δόντια του ρότορα, ευθυγραμμίζονται στο μέσο των μη ενεργοποιημένων πόλων του στάτορα. Αυτή η θέση, είναι η θέση της λιγότερης μαγνητικής αντίστασης ανάμεσα στα πεδία του ρότορα και του στάτορα. Η φάση 2

ενεργοποιημένη παράγει μια παρόμοια ανταπόκριση. Η δεύτερη ομάδα των 4 πόλων, έλκει τα 4 πλησιέστερα δόντια του ρότορα, προκαλώντας τον να προχωρήσει από το μονοπάτι της ελάχιστης αντίστασης προς την ευθυγράμμιση.



Σχήμα 3.6. Απεικόνιση ενός βηματικού κινητήρα τύπου VR.



Σχήμα 3.7. Απεικόνιση υβριδικών κινητήρων 2, 3 και 5 φάσεων.

Αυτή η δράση επαναλαμβάνεται όσο το ηλεκτρομαγνητικό πεδίο του στάτορα γυρίζει γύρω από το ρότορα. Η ενεργοποίηση των πόλων σε συγκεκριμένη ακολουθία, παράγει κίνηση αριστερόστροφη ή δεξιόστροφη. Η ακριβής αύξηση της κίνησης (γωνία βήματος), είναι η διαφορά στη γωνία των δοντιών στάτορα και ρότορα, σε αυτή την περίπτωση 30° και 45° αντίστοιχα για μια διαφορά δικτύου των 15°. Οι γωνίες βημάτων του VR είναι μικρές, κάνοντας δυνατή μια λεπτότερη διάκριση από αυτή που μπορεί να επιτευχθεί με τον PM. Οι ανώτατες τιμές βημάτων είναι γενικά υψηλότερες από ότι στον PM.

### 3.3.3. Υβριδικοί βηματικοί κινητήρες (Hybrid Motors)

Οι υβριδικοί βηματικοί κινητήρες είναι πιθανώς οι πιο χρησιμοποιούμενοι όλων των βηματικών κινητήρων, και αυτό διότι συνδυάζουν τα καλύτερα χαρακτηριστικά των κινητήρων τύπου PM και VR. Οι αυξανόμενες απαιτήσεις για βηματικούς κινητήρες που θα πληρούν καλύτερα χαρακτηριστικά, όπως ο μειωμένος ακουστικός θόρυβος, η καλύτερη οδήγηση καθώς και το χαμηλό κόστος ικανοποιούνται με τη χρήση των υβριδικών βηματικών κινητήρων. Ο ρότορας των υβριδικών κινητήρων αποτελείται από δυο τμήματα «ενωμένα» από μαγνήτη.

## 3.4. Τρόποι λειτουργίας των βηματικών κινητήρων

Διακρίνουμε δυο βασικούς τρόπους λειτουργίας των βηματικών κινητήρων την **μονοπολική** και την **διπολική** λειτουργία.

Στη μονοπολική λειτουργία η οδήγηση των τυλιγμάτων γίνεται με πηγή σταθερής πολικότητας. Κατά συνέπεια η φορά του ρεύματος στις περιελίξεις είναι σταθερή και δεν αντιστρέφεται ποτέ. Το κύκλωμα οδήγησης οδηγεί απλά την κάθε περιέλιξη σε κατάσταση ON ή OFF, με τη βοήθεια ηλεκτρονικών διακοπών.

Στη διπολική λειτουργία η περιστροφή του κινητήρα επιτυγχάνεται με διαδοχική αντίστροφη της φοράς του ρεύματος στα τυλίγματα. Κατά την διπολική λειτουργία η μεσαία λήψη των τυλιγμάτων μένει ασύνδετη (εφόσον πρόκειται για unipolar κινητήρα). Κάθε στιγμή είναι ενεργοποιημένες και οι τέσσερις περιελίξεις του κινητήρα, με αποτέλεσμα να έχουμε την μεγαλύτερη δυνατή ροπή. Το κύκλωμα οδήγησης αναλαμβάνει να αντιστρέφει σε κάθε βήμα την φορά του ρεύματος σε ένα από τα τυλίγματα.

Όπως γίνεται εύκολα κατανοητό, η μονοπολική λειτουργία είναι ευκολότερη από την άποψη της κυκλωματικής υλοποίησης, αφού στηρίζεται σε διακοπτική λειτουργία. Η διπολική λειτουργία είναι αποτελεσματικότερη αλλά χρειάζεται πιο προσεκτική σχεδίαση, ώστε να επιτυγχάνεται η αντίστροφη της φοράς του ρεύματος. Στην **μονοπολική λειτουργία**, διακρίνουμε τις παρακάτω επιμέρους μεθόδους οδήγησης:

#### α. Οδήγηση μόνης φάσης (single phase)

Στη λειτουργία αυτή σε κάθε βήμα ενεργοποιείται διαδοχικά μια από τις τέσσερις περιελίξεις, που εδώ τις αναφέρουμε ως φάσεις. Είναι προφανές πως μόνο κινητήρες με έξι και πέντε ακροδέκτες μπορούν να οδηγηθούν με αυτό το τρόπο. Αν και η μέθοδος αυτή αντιπροσωπεύει την απλούστερη λογική οδήγησης, δημιουργεί τη μικρότερη ροπή στρέψης και γι αυτό δεν είναι κατάλληλη να οδηγήσει δύσκολα φορτία.

#### β. Οδήγηση διπλής φάσης (dual phase)

Με την λογική αυτή σε κάθε βήμα ενεργοποιούνται δυο από τις περιελίξεις. Αυτό συνεπάγεται μεγαλύτερη αποδοτικότητα, όπως συμβαίνει και σε ένα δίχρονο κινητήρα

αυτοκινήτου. Είναι προφανές ότι με τον τρόπο αυτό έχουμε περισσότερες κινητήριες ώσεις σε κάθε περιστροφή και καλύτερο έλεγχο της κίνησης. Οι δυο ενεργοποιούμενες φάσεις πρέπει να ανήκουν σε διαφορετικό τύλιγμα.

Ακολουθία (Sequence)	Πολικότητα (Polarity)	Όνομα (Name)	Περιγραφή (Description)
0001 0010 0100 1000	- - - + - - + - - + - - + - - -	Μονή Φάση (Single Phase ή Wave Drive)	Στη λειτουργία αυτή σε κάθε βήμα ενεργοποιείται διαδοχικά μόνο μια από τις τέσσερις περιελίξεις που εδώ τις αναφέρουμε ως φάσεις
0011 0110 1100 1001	- - + + - + + - + + - - + - - +	Διπλή Φάση (Dual Phase)	Με τη λογική αυτή σε κάθε βήμα ενεργοποιούνται δυο από τις περιελίξεις, που αυτό συνεπάγεται μεγαλύτερη αποδοτικότητα. Έχουμε την μεγαλύτερη δυνατή ροπή
0001 0011 0010 0110 0100 1100 1000 1001	- - - + - - + + - - + - - + + - - + - - + + - - + - - - + - - +	Μισού Βήματος (Half Step)	Πρόκειται για ένα συνδυασμό των παραπάνω δυο μεθόδων οδήγησης. Αυτό οφείλεται διότι ανάμεσα στα βήματα διπλής ενεργοποίησης παρεμβάλλεται ένα βήμα μόνης ενεργοποίησης.

Πίνακας 3.1. Μέθοδοι οδήγησης της μονοπολικής λειτουργίας.

### γ. Οδήγηση μισού βήματος (half step)

Πρόκειται για μια παραλλαγή της παραπάνω μονοπολικής λειτουργίας διπλής φάσης, στην οποία ανάμεσα στα βήματα διπλής ενεργοποίησης παρεμβάλλεται ένα βήμα μόνης ενεργοποίησης. Το αποτέλεσμα αυτής της οδήγησης είναι περιστροφή με μισό βήμα σε σχέση με την οδήγηση διπλής φάσης. Ο τρόπος αυτός λειτουργίας εξασφαλίζει μια ομαλότερη κίνηση του άξονα περιστροφής, αλλά τα μισά βήματα δημιουργούν τη μίση ροπή στρέψης. Η λειτουργία του μισού βήματος εξομαλώνει μεν την περιστροφή αφού

διπλασιάζει τα βήματα αλλά δεν αυξάνει πραγματικά την ακρίβεια στο έλεγχο της τελικής θέσης. Στον πίνακα 3.1. απεικονίζεται η ακολουθία των βημάτων για κάθε μια από τις τρεις επιμέρους μεθόδους οδήγησης.

### 3.5. Διπολική οδήγηση βηματικού κινητήρα

Όπως ήδη αναφέρθηκε η διπολική λειτουργία εξασφαλίζει μεγάλη ροπή στρέψης, τετραπλάσια από αυτή της μονοπολικής λειτουργίας μονής φάσης, και επιπλέον μπορεί να εφαρμοστεί σε οποιοδήποτε βηματικό κινητήρα, ανεξάρτητα από τον αριθμό των ακροδεκτών του. Η διπολική λειτουργία απαιτεί τα τυλίγματα να διαρρέονται συνεχώς από ρεύμα, ενώ σε κάθε βήμα αλλάζει η φορά του ρεύματος σε ένα από τα δυο τυλίγματα. Άρα χρειαζόμαστε ένα δυαδικό κώδικα τεσσάρων bit στον οποίο κατά την απαρίθμηση θα αλλάζουν δυο bit κάθε φορά. Ένας τέτοιος κώδικας είναι ο κώδικας GRAY. Ο πίνακας 3.2 δείχνει ένα κατάλληλο τρόπο οδήγησης.

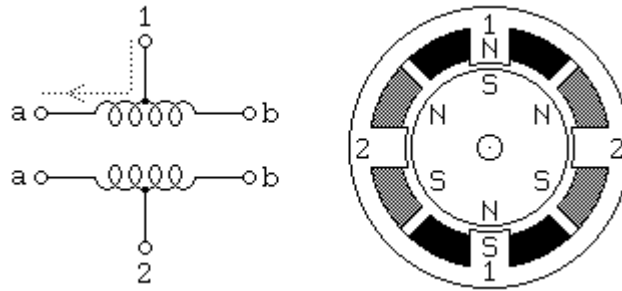
Δυαδικό σύστημα	Κώδικας Gray
0000	0101
0001	0110
0010	1010
0011	1001
0100	0101
0101	0110
0110	1010
0111	1001

Πίνακας 3.2. Κώδικας GRAY.

Πρακτικά τα δυο περισσότερο σημαντικά bit (bit 0 και bit 1) από τα τέσσερα του κώδικα GRAY θα οδηγούν το ένα τυλίγμα του κινητήρα και τα δυο περισσότερα σημαντικά bit (bit 2 και bit 3) θα οδηγούν το άλλο τυλίγμα του κινητήρα. Όπως προαναφέραμε και παρατηρούμαι και από τον παραπάνω πίνακα κάθε φορά αλλάζει η φορά του ρεύματος σε ένα μόνο από τα δυο τυλίγματα.

### 3.6. Μονοπολικοί βηματικοί κινητήρες

Γενικά οι μονοπολικοί (Unipolar) βηματικοί κινητήρες είναι εύκολο να οδηγηθούν. Ο αριθμός των φάσεων ενός κινητήρα είναι διπλάσιος από τον αριθμό των τυλιγμάτων, δεδομένου ότι κάθε τυλίγμα χωρίζεται στη μέση άρα ένα Unipolar κινητήρας 6 ακροδεκτών είναι 4 φάσεων. Οι Unipolar βηματικοί κινητήρες χαρακτηρίζονται από τη μεσαία λήψη που έχουν σε κάθε ένα από τα δυο τυλίγματα τους. Έχουν συνήθως 5,6 ή 8 καλώδια από τα οποία δέχονται τα ρεύματα από τον έξω κόσμο όπως φαίνεται και στο παρακάτω σχήμα. Τα τυλίγματα είναι όμοια και δεν είναι ηλεκτρικά συνδεδεμένα. Συνήθως η αντίσταση του τυλίγματος μεταξύ της μεσαίας λήψης και του ενός τερματικού σημείου αναγράφεται πάνω στον κινητήρα. Έτσι για παράδειγμα εάν αναγράφεται «5 Ohms/φάση (phase)» δηλώνει την αντίσταση μεταξύ της μεσαίας λήψης του τυλίγματος και του ενός τερματικού σημείου. Η αντίσταση μεταξύ των δύο τερματικών σημείων θα πρέπει να είναι 10 Ohms.

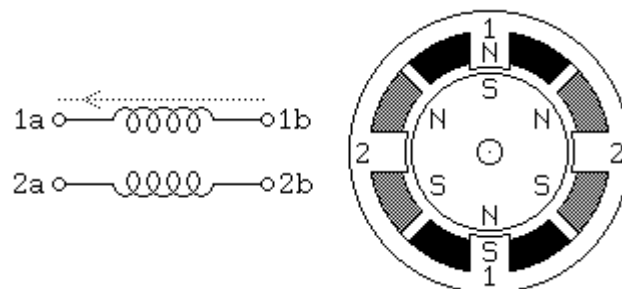


Σχήμα 3.8. Αναπαράσταση τυλιγμάτων και πόλων ενός μονοπολικού βηματικού κινητήρα.

Στην πράξη οι κεντρικές λήψεις των τυλιγμάτων συνδέονται συνήθως με την τάση τροφοδοσίας, και εναλλακτικά συνδέονται με την γείωση για αλλαγή κατεύθυνσης. Ο ρότορας είναι μόνιμου μαγνήτη με 6 πόλους, 3 βόρειους και 3 νότιους. Για να έχουμε μικρότερη γωνιακή μετατόπιση κάθε φορά που ο ρότορας κινείται κατά ένα βήμα θα πρέπει να υπάρχουν περισσότεροι πόλοι. Στην αγορά είναι ευρέως διαθέσιμοι βηματικοί κινητήρες με γωνιακή μετατόπιση ενός βήματος 7.5 μοιρών και 15 μοιρών. Όπως φαίνεται και στο σχήμα 3.1. το ρεύμα ρέει με κατεύθυνση από τη μεσαία λήψη προς το ένα τερματικό σημείο του τυλιγματος 1 (a), το οποίο προκαλεί τον πάνω πόλο του στάτορα να γίνει βόρειος πόλος ενώ αντίθετα ο κάτω πόλος του στάτορα γίνεται νότιος πόλος.

### 3.7. Διπολικό βηματικό κινητήρες

Οι βηματικοί κινητήρες μόνιμου μαγνήτη (PM) διπλής φάσης (bipolar) κατασκευάζονται με τον ίδιο μηχανισμό που κατασκευάζονται οι Unipolar βηματικοί κινητήρες. Η διαφορά τους έγκειται στο ότι δεν έχουν μεσαία λήψη στα τυλίγματα τους. Κατά συνέπεια ο κινητήρας από μόνος του είναι πιο απλός αλλά τα κυκλώματα οδήγησης του γίνονται πολύπλοκα. Το παρακάτω σχήμα αναπαριστά ένα Βηματικό Κινητήρα Με Διπολικά Τυλίγματα



Σχήμα 3.9. Ένας διπολικός Βηματικός Κινητήρας.

Τα κυκλώματα οδήγησης απαιτούν την χρήση ενός κυκλώματος ελέγχου με τη ονομασία Γέφυρα H (H-Bridge). Ένα τέτοιο κύκλωμα απαιτείτε για κάθε τύλιγμα του κινητήρα. Θα μιλήσουμε εκτενέστερα για το συγκεκριμένο κύκλωμα ελέγχου σε επόμενο κεφάλαιο. Ένας bipolar κινητήρας μπορεί να οδηγηθεί τόσο διπολικά όσο και μονοπολικά. Η διπολική οδήγηση δίνει μεγαλύτερη ροπή στρέψης.



### **3.8. Πλεονεκτήματα και μειονεκτήματα των βηματικών κινητήρων**

#### Πλεονεκτήματα

- Δε ζητείται ανάδραση όταν ο κινητήρας τροφοδοτείται κατάλληλα. Πάντως είναι τέλεια συμβατός και με την αναλογική και με την ψηφιακή ανάδραση, είτε για ταχύτητα είτε για θέση είτε και για τα δυο.
- Οι βηματικοί κινητήρες δεν έχουν ουδέτερη θέση. Η δικατευθυντική περιστροφή είναι συνεχής.
- Η μέγιστη ροπή συμβαίνει σε χαμηλές τιμές παλμού. Ο βηματικός κινητήρας μπορεί να επιταχύνει το φορτίο του. Όταν φτάσει η επιθυμητή θέση και οι παλμοί εντολών παύσουν, ο άξονας σταματά. Δεν υπάρχει ανάγκη για συνδέσμους ή φρένα. Όταν σταματήσει ο βηματικός κινητήρας, υπάρχει μια μικρή κλίση για μετατόπιση. Πράγματι, οι βηματικοί κινητήρες PM είναι μαγνητικά κρατημένοι στην τελευταία θέση. Για να συνοψίσουμε ένα φορτίο μπορεί να αρχίσει να κινηθεί προς κάθε κατεύθυνση σε μια θέση και θα μείνει εκεί μέχρι να του δοθεί εντολή ξανά.
- Μια μεγάλη περιοχή γωνιών βημάτων, είναι διαθέσιμη στη γραμμή του κινητήρα  $-1.8^\circ$ ,  $7.5^\circ$ ,  $15^\circ$ ,  $45^\circ$ ,  $90^\circ$  - χωρίς ειδικούς οδηγούς και λογικό χειρισμό.
- Η εσωτερική χαμηλή ταχύτητα, είναι διαθέσιμη χωρίς γρανάζια μείωσης. Πολλοί AC και περιτυλιγμένου πεδίου DC κινητήρες, έχουν δυσκολία στην περιστροφή τους σε τόσο χαμηλή ταχύτητα.
- Οι βηματικοί κινητήρες είναι αληθινές ψηφιακές συσκευές. Δε χρειάζονται μετατροπή D/A στην είσοδο όπως οι συμβατικοί. Μπορούν λοιπόν να χρησιμοποιηθούν με κομπιούτερ χωρίς αυτούς.
- Προσφέρουν κλειστό έλεγχο ταχύτητας και δυνατότητα αναστροφής πάνω σε μεγάλη περιοχή.
- Το ρεύμα έναρξης είναι χαμηλό.
- Η αδράνεια του ρότορα είναι συνήθως χαμηλή.
- Πολλοί βηματικοί κινητήρες, οδηγούνται από την ίδια πηγή παραμένοντας σε τέλειο συγχρονισμό, που είναι πρόβλημα σε άλλους τύπους DC.

#### Μειονεκτήματα

- Η απόδοση είναι χαμηλή. Πολύ από την ενέργεια εισόδου πρέπει να καταναλωθεί σε θερμότητα.
- Τα φορτία πρέπει ν' αναλυθούν προσεκτικά για την άριστη λειτουργία του βηματικού κινητήρα. Οι εισοδοί (πηγές παλμών και ελεγκτές), πρέπει επίσης να ταιριάζουν με τον κινητήρα και το φορτίο.
- Όταν η αδράνεια φορτίου είναι εξαιρετικά υψηλή, ίσως χρειάζεται απόσβεση. Ειδικοί υβριδικοί βηματικοί κινητήρες, έχουν παραχθεί έχοντας βελτιωμένη λειτουργία σε μερικές περιοχές λειτουργίας.

## 4. Visual Basic 6

### 4.1. Περιβάλλον της Visual Basic

Η Visual Basic (VB), μέλος της ομάδας προγραμμάτων του Microsoft® Visual Studio, είναι η πιο πρόσφατη και καλύτερη έκδοση της παλιάς γλώσσας Basic. Με την εκκίνηση της Visual Basic θα εμφανιστεί αμέσως ένας οδηγός επιλογής του τύπου προγράμματος ή στοιχείου που επιθυμούμε να δημιουργήσουμε.

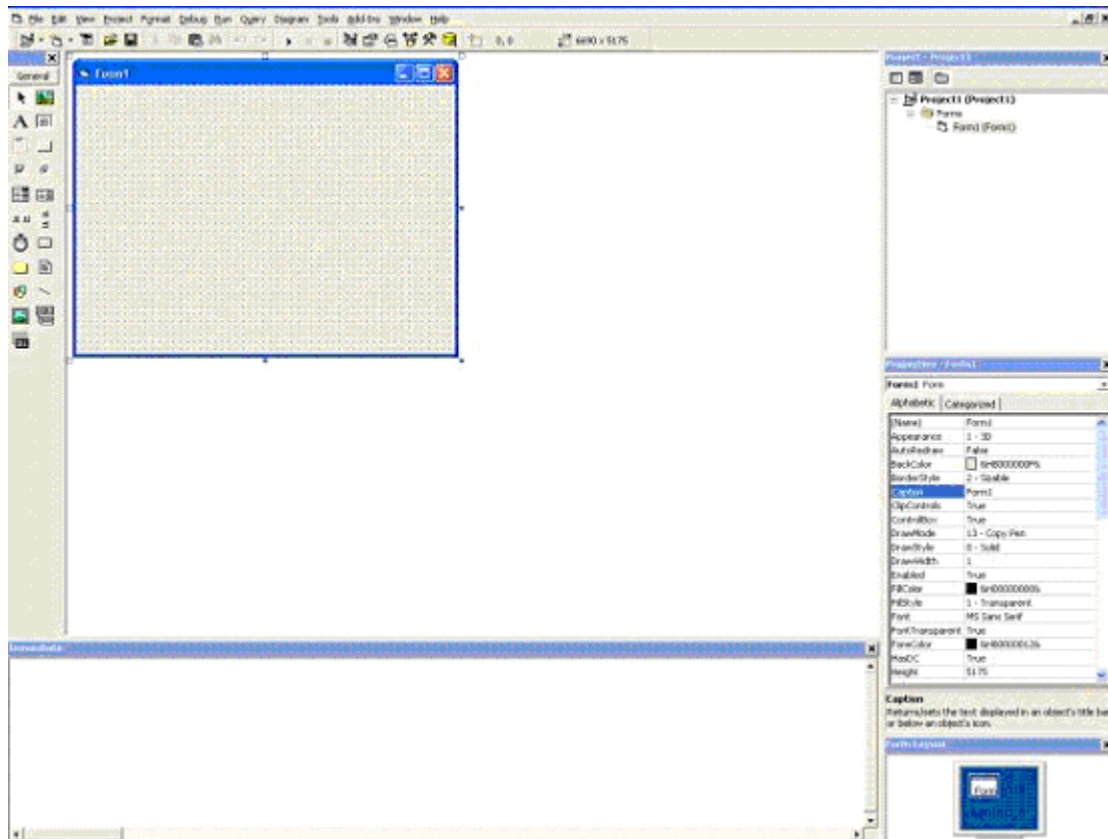


Σχήμα 4.1. Οδηγός επιλογής του τύπου προγράμματος ή στοιχείου.

Στην παραπάνω εικόνα φαίνονται οι επιλογές μας. Θα ασχοληθούμε μόνο με το Standard Exe, δηλαδή με την δημιουργία ενός πλήρους περιγράμματος. Παρατηρώντας την εικόνα βλέπουμε τρεις βασικές επιλογές.

- ✓ New. Επιλέγουμε τον τύπο προγράμματος που θα δημιουργήσουμε από το «μηδέν»
- ✓ Existing. Ψάχνουμε τον δίσκο για να βρούμε και να επιλέξουμε ένα ήδη δημιουργημένο αποθηκευμένο πρόγραμμα
- ✓ Recent. Μας περιέχει μια λίστα με τα προσφάτως ανοιγμένα προγράμματα.

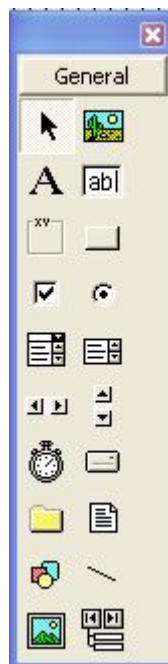
Μόλις επιλέξουμε το Standard Exe εισερχόμαστε στο κύριο περιβάλλον της Visual Basic 6.



Σχήμα 4.2. Περιβάλλον Visual Basic 6.

Από την παρατήρηση του σχήματος 4.2. βλέπουμε πως το περιβάλλον είναι διαιρεμένο σε υποπεριοχές. Κάθε μια από αυτές, έχει ένα συγκεκριμένο ρόλο στη υποβοήθηση του προγραμματιστή για τη σύνταξη ενός ολοκληρωμένου project.

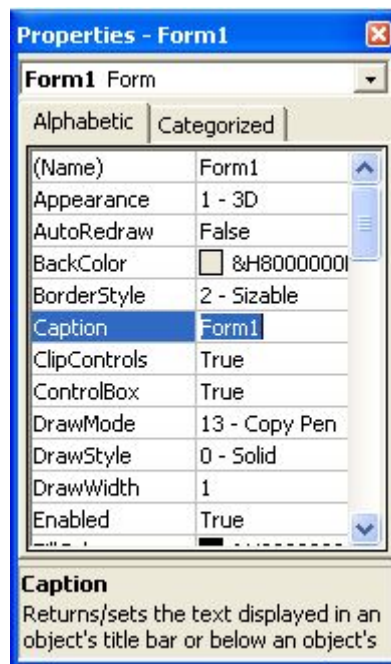
### Γραμμή Εργαλείων



Σχήμα 4.3. Γραμμές Εργαλείων.

Η γραμμή εργαλείων βρίσκεται στα αριστερά της κεντρικής οθόνης. Περιλαμβάνει τα «στοιχεία ελέγχου» πάνω στα οποία βασίζεται η δημιουργία ενός project. Τα αντικείμενα αυτά καθοδηγούν τον χρήστη στη σύνταξη ενός λογισμικού. Σε αυτά οφείλεται ο όρος «αντικειμενοστραφής προγραμματισμός». Η γραμμή εργαλείου περιλαμβάνει τα βασικά στοιχεία ελέγχου. Ωστόσο ο χρήστης μπορεί εύκολα να προσθέσει επιπλέον όσα προαιρετικά εργαλεία επιθυμεί.

### Properties Window.



Σχήμα 4.4. Ιδιότητες στοιχείων ελέγχου.

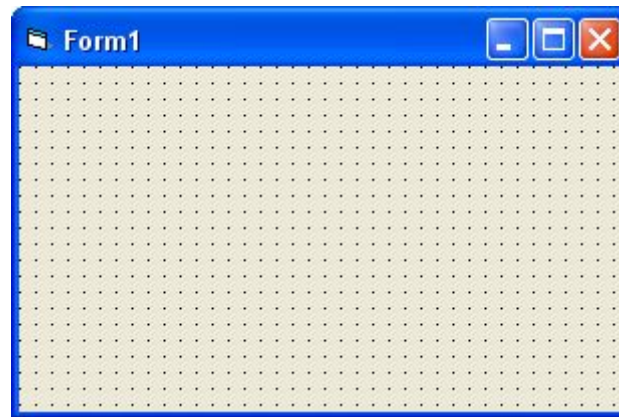
Όλα τα στοιχεία ελέγχου έχουν ιδιότητες. Αυτές είναι παράμετροι που σχετίζονται με την εμφάνιση ενός αντικειμένου (χρώμα, γραμματοσειρά, τίτλος κτλ) και άλλα. Τα στοιχεία ελέγχου εκτός από τις ιδιότητες έχουν συμβάντα (events) και μεθόδους (methods).

Συμβάντα (events) ονομάζονται όλες εκείνες οι ενέργειες οι οποίες γίνονται από το χρήστη κατά τη χρήση του προγράμματος. Για παράδειγμα συμβάν για ένα κουμπί (button) το οποίο προσθέσαμε σε μια φόρμα, είναι το «κλικ» του ποντικιού, το «δεξί κλικ», το πέρασμα του ποντικιού πάνω από το button κτλ. Είναι λοιπόν ενέργειες του χρήστη στις οποίες εμείς προσθέτουμε κάποιες λειτουργίες.

Μέθοδοι (methods) ονομάζονται κάποιες λειτουργίες οι οποίες είναι κατά το πλείστον προαποφασισμένου αποτελέσματος, ενώ εμείς προσθέτουμε απλά μια παράμετρο. Για παράδειγμα μια ενδεικτική μέθοδος για το button είναι η «move». Το τι θα κάνει είναι προαποφασισμένο, δηλαδή η μετακίνηση του κουμπιού. Το που όμως θα μετακινηθεί το δίνει ο χρήστης του προγράμματος.

### Φόρμα

Φόρμα είναι εκείνο το αντικείμενο πάνω στο οποίο δημιουργούμε το project μας, δηλαδή το αντικείμενο πάνω στο οποίο θα τοποθετήσουμε τα στοιχεία ελέγχου που θα απαρτίζουν το πρόγραμμά μας.



Σχήμα 4.5. Φόρμα Εργασίας.

### Εξερευνητής (Project Explorer)

Ο Project Explorer μας δείχνει από ποια συστατικά αποτελείται το project που σχεδιάζουμε και μας επιτρέπει τη μετάβαση από το ένα στο άλλο. Περιεχόμενα του εξερευνητή είναι οι φόρμες, τα modules, τα class modules κτλ.

Module ονομάζεται ένα αυτόνομο τμήμα κώδικα το οποίο δεν περιέχει γραφικό περιβάλλον. Δεν ανήκει σε μια φόρμα, έχει δικό του όνομα, αποθηκεύεται ξεχωριστά και είναι κατάλληλο για χρήση σε πολλά προγράμματα. Ένα module περιέχει συνήθως δηλώσεις τύπων και μεταβλητών, συναρτήσεις και υπορουτίνες. Class Module είναι ένα προωθημένο χαρακτηριστικό της VB. Όταν ένα Class Module μεταγλωττίζετε (compilation & build), δημιουργείτε μια βιβλιοθήκη DLL. Αποτελεί μια αυτόνομη ομάδα διαδικασιών και συναρτήσεων η οποία μπορεί να δουλέψει ξεχωριστά από το υπόλοιπο project .

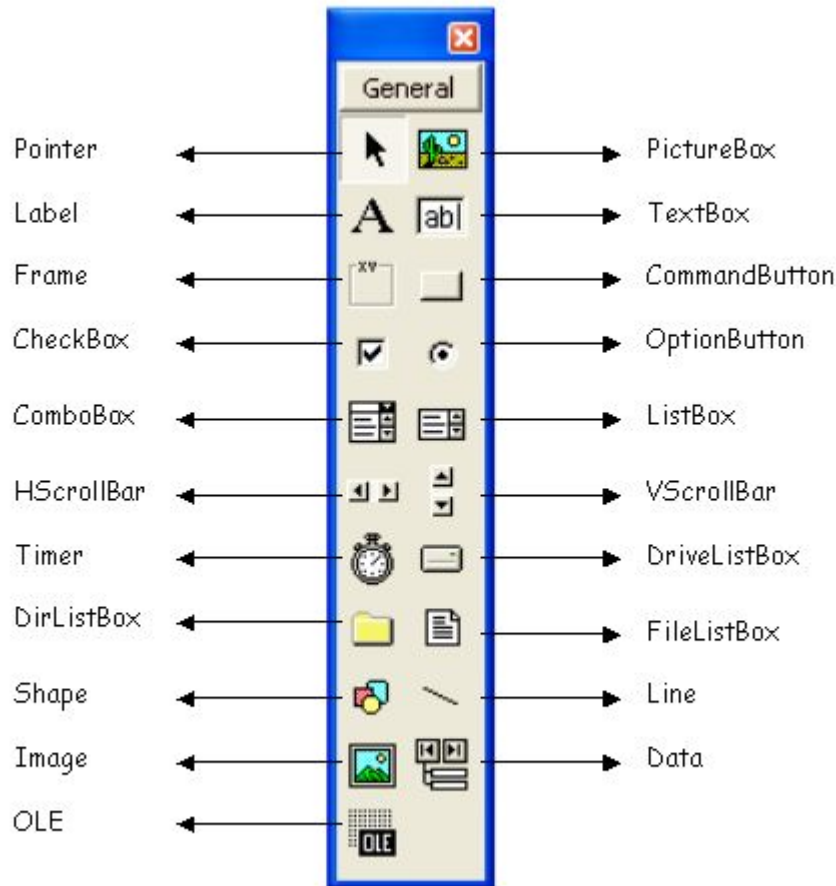
### Μενού επιλογών (Menu)

Το μενού επιλογών βρίσκεται στην κορυφή του περιβάλλοντος εργασίας και περιλαμβάνει ταξινομημένες όλες τις λειτουργίες της Visual Basic, που είναι το File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help.

## 4.2. Στοιχεία ελέγχου, χρήση, ιδιότητες, συμβάντα, μέθοδοι

Όπως έχει αναφερθεί στοιχεία ελέγχου είναι τα αντικείμενα εκείνα με τα οποία ο χρήστης εκτελεί κάποιες διαδικασίες και λειτουργεί το πρόγραμμα. Ο χρήστης γνωρίζει πως όταν βλέπει ένα στοιχείου ελέγχου, για παράδειγμα ένα κουμπί ότι πρέπει να το αιτήσει για να κάνει κάτι. Τι γίνεται όμως όταν υπάρχουν πολλά κουμπιά;

Το πρόβλημα λύνεται δίνοντας ένα περιγραφικό τίτλο στο κουμπί που λέει στον χρήστη τι κάνει. Για παράδειγμα εάν γράφει exit ο χρήστης καταλαβαίνει πως αν το πατήσει, θα τερματιστεί το πρόγραμμα. Αυτός ο τίτλος είναι μια Ιδιότητα.



Σχήμα 4.6. Βασικά Στοιχεία Ελέγχου στη Visual Basic 6.

Στο Σχ. 4.6. βλέπουμε κάποια στοιχεία ελέγχου καθώς και τις ονομασίες τους αλλά θα επικεντρωθούμε σε αυτά που απαρτίζουν το περιβάλλον της παρούσας πτυχιακής.

### Ετικέτα (Label)

Πρόκειται για ένα απλό εργαλείο το οποίο χρησιμοποιούμε για να γράψουμε κάτι (μια λεζάντα/ caption). Κατά την εκτέλεση του προγράμματος δεν μπορεί ο χρήστης να επέμβει στο κείμενο της.

Εδώ εμφανίζεται η ιδιότητα Caption

Σχήμα 4.7. Ετικέτα.

### Πλαίσιο (Frame)

Ο τίτλος του Frame δίνεται στην ιδιότητα Caption

Σχήμα 4.8. Πλαίσιο.

Πρόκειται για ένα εργαλείο ομαδοποίησης στοιχείων ελέγχου πάνω στη φόρμα. Είναι εξαιρετικά χρήσιμο γιατί παρέχει ευελιξία και διευκολύνει την οργάνωση των εργαλείων πάνω στη φόρμα.

### Πλαίσιο αναδιπλούμενης λίστας (ComboBox)

Το ComboBox είναι ένα πλαίσιο το οποίο χρησιμοποιούμε για οικονομία χώρου, όταν θέλουμε να διαθέσουμε ένα πλήθος επιλογών στο χρήστη. Μας δίνει την δυνατότητα της απλής ή πολλαπλής επιλογής ανάλογα με την τιμή μιας συγκεκριμένης ιδιότητας.



Σχήμα 4.9. Πλαίσιο Αναδιπλούμενης Λίστας.

### Οριζόντια μπάρα κύλισης (HscrollBar), Κατακόρυφη μπάρα κύλισης (VscrollBar)



Σχήμα 4.10. Οριζόντια μπάρα κύλισης και κατακόρυφη μπάρα κύλισης.

Τόσο η κατακόρυφη όσο και η οριζόντια μπάρα κύλισης είναι στοιχεία ελέγχου «πλοηγητικού χαρακτήρα». Μπορούμε να μετακινηθούμε μεταξύ μιας ελάχιστης και μιας μέγιστης τιμής εύκολα και γρήγορα. Οι ιδιότητες, τα συμβάντα και οι μέθοδοι είναι κοινές και για τα δυο.

### Χρονομετρητής (Timer)



Σχήμα 4.11. Χρονομετρητής.

Είναι ένα εργαλείο που μας επιτρέπει να θέσουμε μια έγκυρη τιμή χρονικής διάρκειας, που με τη λήξη αυτού «πυροδοτεί» κάποιο συμβάν. Είναι αόρατο στο τελικό χρήστη.

### Γραμμή (Line)



Σχήμα 4.12. Γραμμή στη Visual Basic 6.

Το εργαλείο Line ανήκει ουσιαστικά στην κατηγορία Shape. Δεν έχει συμβάντα.



**Εικόνα (Image)**

Το εργαλείο αυτό μας δίνει την δυνατότητα να προσθέτουμε εικόνες σε μια φόρμα.



Σχήμα 4.13. Εικόνα.

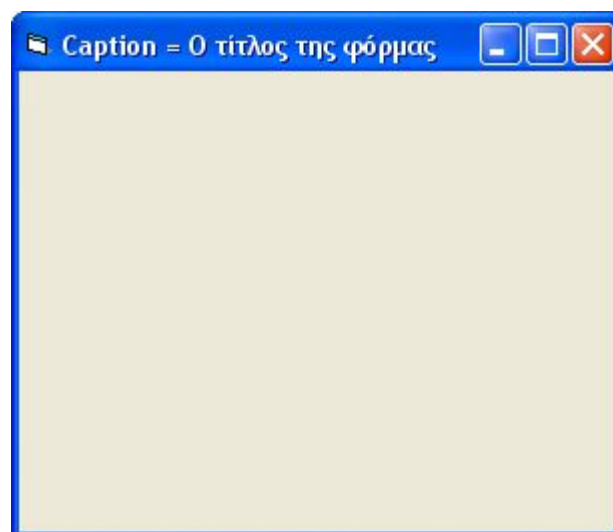
**Πλαίσιο κειμένου (TextBox)**

Το πλαίσιο κειμένου είναι ένα χρήσιμο εργαλείο το οποίο μας επιτρέπει να εισάγουμε δεδομένα πριν και κατά τη φάση μιας εκτέλεσης. Το text είναι η βασικότερη ιδιότητα του πλαισίου κειμένου. Με αυτή μπορούμε να γράψουμε ένα κείμενο ή να λάβουμε αυτό που έγραψε ο χρήστης.

**Κουμπί επιλογής ή πλήκτρο επιλογής (OptionButton)**

Το OptionButton είναι ένα στοιχείο ελέγχου το οποίο μας επιτρέπει να πραγματοποιήσουμε μια και μόνο μια επιλογή από αυτές που μας δίνονται

Μοιάζει με το εργαλείο CheckBox αλλά η μεγάλη διαφορά έγκειται στο γεγονός ότι στο checkbox μπορούμε να επιλέξουμε ταυτόχρονα παραπάνω από μια επιλογές. Στο optionbutton η επιλογή είναι μια και αποκλειστική. Για παράδειγμα ο άξονας του βηματικού κινητήρα θέλουμε κάθε φορά να κινείται προς μια κατεύθυνση (δεξιά ή αριστερά). Προφανώς δεν είναι εφικτό να χρησιμοποιούμε ταυτόχρονα και τις δυο επιλογές.

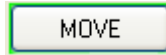
**Φόρμα (Form)**

Σχήμα 4.14. Φόρμα.



Η φόρμα αποτελεί το πιο σημαντικό στοιχείο ελέγχου. Πάνω της τοποθετούνται όλα τα στοιχεία ελέγχου τα οποία συνθέτουν το τελικό αποτέλεσμα. Έχει πολλές ιδιότητες συμβάντα και μεθόδους.

### Κουμπί ή πλήκτρο (CommandButton)



Σχήμα 4.15. Κουμπί ή Πλήκτρο.

Το «κουμπί» είναι το πιο άμεσο εργαλείο που χρησιμοποιείτε για να «πυροδοτήσει» την έναρξη κάποιας διαδικασίας. Χρησιμοποιείται σε πολλές εφαρμογές λόγω της απλότητας χειρισμού του.

### Περισσότερα στοιχεία ελέγχου

Πέρα από τα στοιχεία ελέγχου που συναντήσαμε στη μπάρα Toolbox υπάρχουν εκατοντάδες ακόμα που δημιουργήθηκαν είτε από τη Microsoft είτε από άλλους κατασκευαστές. Κάποια από αυτά τα στοιχεία ελέγχου μπορεί να τα βρει κανείς στη διαδρομή Project-Components. Ένα στοιχείο ελέγχου το οποίο χρησιμοποιείτε στην παρούσα πτυχιακή είναι το MicroHelp Gauge Control, το οποίο και αναπαριστά σχηματικά την χρονοκαθυστέρηση και την ακριβή θέση των βηματικών κινητήρων.

## 4.3. Ιδιότητες

Κάθε φόρμα και χειριστήριο έχει διάφορες ιδιότητες που αφορούν την εμφάνιση τους (π.χ. μέγεθος, χρώμα, κείμενο), τη συμπεριφορά τους απέναντι στις ενέργειες του χρήστη κτλ. Όλες αυτές οι ιδιότητες εμφανίζονται στο παράθυρο ιδιοτήτων συγκεντρωμένες είτε αλφαβητικά (alphabetic), είτε ανά κατηγορία (categorized). Το παράθυρο ιδιοτήτων μας επιτρέπει να τις τροποποιήσουμε. Πατώντας το μικρό βελάκι στο πάνω δεξιά μέρος του παραθύρου εμφανίζεται μια λίστα με τα ονόματα της φόρμας και των χειριστηρίων του προγράμματός μας, όπου μπορούμε να ορίσουμε τιμές στις ιδιότητες τους. Οι ιδιότητες έχουν μια προκαθορισμένη τιμή από την Visual Basic. Πολλά χειριστήρια έχουν κοινές ιδιότητες. Στη συνέχεια αναφέρονται ορισμένες βασικές ιδιότητες χειριστηρίων.

**Name** (όνομα): χρησιμοποιείται όταν γίνεται αναφορά στο χειριστήριο σε μια διαδικασία. Η Visual Basic για κάθε νέο χειριστήριο προσφέρει ένα προκαθορισμένο όνομα, το οποίο αποτελείται από τον τύπο του χειριστηρίου και έναν αριθμό π.χ. Label1, Label2, Text1, Command1, List1 κλπ.

**Caption** (λεζάντα ή τίτλος): περιέχει το κείμενο το οποίο εμφανίζεται στην περιοχή (επιφάνεια) του χειριστηρίου.

**Font** (γραμματοσειρά): μπορεί να αλλάξει τη γραμματοσειρά του κειμένου το οποίο εμφανίζεται μέσα στο χειριστήριο. Εμφανίζει ένα παράθυρο διαλόγου Font στο οποίο μπορείτε να επιλέξετε το όνομα, το στυλ και μέγεθος της γραμματοσειράς που επιθυμείτε.

**ForeColor**: μπορεί να αλλάξει το χρώμα του κειμένου ή των γραφικών που εμφανίζονται σε ένα χειριστήριο. Υποστηρίζει δύο επιλογές για τον τρόπο αλλαγής του χρώματος μέσω των καρτελών System και Palette που περιέχει. Η καρτέλα System εμφανίζει τα τρέχοντα χρώματα που χρησιμοποιεί το σύστημά σας για τα στοιχεία της

διασύνδεσης χρήστη. Η καρτέλα Palette περιέχει όλα τα διαθέσιμα χρώματα του συστήματός σας.

**BackColor:** μπορεί να αλλάξει το χρώμα του φόντου του χειριστηρίου.

**Visible** (εμφανές): καθορίζει εάν ένα χειριστήριο θα είναι εμφανές ή κρυμμένο από τον χρήστη.

**Enabled** (ενεργοποίηση): καθορίζει εάν ένα αντικείμενο μπορεί να χρησιμοποιηθεί. Η ιδιότητα Enabled μπορεί να λάβει είτε την τιμή True (αληθής) η οποία δηλώνει ενεργοποίηση είτε την τιμή False (ψευδής) που δηλώνει απενεργοποίηση.

**Width** (πλάτος): αναφέρεται στο πλάτος του χειριστηρίου σε twips. Twip είναι μια μονάδα μέτρησης η οποία ισούται με το ένα εικοστό της στιγμής (point). Η στιγμή είναι ίση με 1/72 της ίντσας οπότε το twip ισούται με 1/1440 της ίντσας, περίπου 17,6 χιλιοστά του χιλιοστού.

**Height** (ύψος): αναφέρεται στο ύψος του περιγράμματος του χειριστηρίου σε twips.

**Left:** περιέχει τον αριθμό σε twips από την αριστερή πλευρά του χειριστηρίου μέχρι την αριστερή πλευρά της φόρμας.

**Top:** περιέχει τον αριθμό σε twips από την κορυφή του χειριστηρίου μέχρι την κορυφή του παραθύρου φόρμας.

#### 4.4. Κώδικας

Όταν δημιουργήσει κάποιος τη διασύνδεση χρήστη, το επόμενο βήμα είναι να γράψει προγραμματιστικές εντολές BASIC (γνωστές με την ονομασία κώδικας) έτσι ώστε να τεθεί το πρόγραμμά σε λειτουργία. Προγραμματιστική εντολή (ή απλούστερα εντολή) είναι οποιοσδήποτε συνδυασμός λέξεων-κλειδιών, ιδιοτήτων, συναρτήσεων, τελεστών και συμβόλων που όλα μαζί αποτελούν μια έγκυρη οδηγία αναγνωρίσιμη από τον μεταγλωττιστή της Visual Basic. Η Visual Basic διατηρεί πολλούς από τους συντακτικούς κανόνες των προγενέστερων εκδόσεων της γλώσσας προγραμματισμού BASIC καθώς και των μεταγλωττιστών άλλων γλωσσών. Για παράδειγμα, η σύνταξη της εντολής για την απόδοση τιμής σε μια ιδιότητα ενός αντικειμένου είναι η εξής: *Αντικείμενο . Ιδιότητα = Τιμή*

Παράδειγμα: Text1.Text= "Η πρώτη μου εντολή σε Visual Basic"

Ο σκοπός του κώδικα της Visual Basic είναι να λέει στα αντικείμενα μιας φόρμας τι να κάνουν όταν ο χρήστης κάνει κάποια ενέργεια (π.χ. πάτημα ενός πλήκτρου). Για παράδειγμα, εάν κάνετε κλικ με το ποντίκι σας σε ένα αντικείμενο μιας φόρμας, δεν πρόκειται να συμβεί τίποτε εάν δεν έχετε γράψει εντολές που θα λένε στο υπολογιστή σας τι ακριβώς να κάνει. Όταν ο χρήστης πατήσει ένα πλήκτρο ή κάνει κλικ με το ποντίκι του τότε έχουμε ένα *συμβάν* (event). Κάθε φορά που προκαλείται ένα συμβάν, οι εντολές της Visual Basic λένε στον υπολογιστή τι να κάνει.

Παράδειγμα: Έστω ότι έχετε δημιουργήσει ένα πρόγραμμα στη Visual Basic το οποίο έχει μια φόρμα στη οποία έχετε τοποθετήσει ένα πλήκτρο εντολής (CommandButton). Όμως το πρόγραμμά αυτό δεν μπορεί να κάνει τίποτε εφόσον δεν περιέχει κώδικα. Για να το επιβεβαιώσετε αυτό, από τη γραμμή μενού της Visual Basic πατήστε *Run* ⇒ *Start* (ή εναλλακτικά πατήστε το εικονίδιο με το μικρό μπλε τρίγωνο της γραμμής εργαλείων). Τότε θα εμφανιστεί ένα παράθυρο με το πλήκτρο εντολής. Προχωρήστε και κάντε κλικ με το ποντίκι σας στο πλήκτρο εντολής (Command1) του προγράμματός σας, τίποτε όμως δεν συμβαίνει. Στη συνέχεια, από τη γραμμή μενού της Visual Basic πατήστε *Run* ⇒ *End* (ή εναλλακτικά πατήστε το εικονίδιο με το μικρό μπλε τετράγωνο της γραμμής

εργαλείων). Τότε οδηγείστε στο κύριο παράθυρο της Visual Basic. Κάντε διπλό κλικ στο πλήκτρο Command1 της φόρμα σας. Τότε θα εμφανιστεί το παράθυρο κώδικα (Code Window) στο οποίο μπορείτε να πληκτρολογήσετε τον κώδικα του προγράμματός σας. Η Visual Basic όχι μόνο θα εμφανίσει το παράθυρο κώδικα αλλά θα δημιουργήσει και κάποιον πηγαίο κώδικα ο οποίος ονομάζεται διαδικασία (procedure). Δηλαδή θα δημιουργήσει αυτόματα τον παρακάτω κώδικα:

```
Private Sub Command1_Click()
```

```
End Sub
```

Στη συνέχεια, προσθέτουμε στον παραπάνω κώδικα την εντολή: MsgBox "ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ" ώστε το πρόγραμμα σας κατά την εκτέλεσή του να εμφανίζει ένα πλαίσιο μηνύματος με το μήνυμα: ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ.

```
Private Sub Command1_Click()
```

```
    MsgBox "Γεωπονικό ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ "
```

```
End Sub
```

Αυτή τη φορά, εάν επιλεγεί από τη γραμμή μενού *Run*  $\Rightarrow$  *Start* και κάνετε κλικ στο πλήκτρο εντολής Command1 θα εμφανιστεί ένα πλαίσιο μηνύματος με το μήνυμα ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ. Όταν ένας χρήστης κάνει κλικ στο πλήκτρο Command1 τότε η Visual Basic πηγαίνει στη διαδικασία Command1\_Click() και εκτελεί με τη σειρά όλες τις εντολές που περιέχονται στη διαδικασία.

## 4.5. Μεταβλητές (τύποι, χρήση, δήλωση, σύνταξη)

### Μεταβλητές

Μεταβλητές ονομάζονται εκείνες οι παράμετροι οι οποίες χρησιμοποιούμε για να αποθηκεύσουμε κάποια τιμή. Υπάρχουν εξ ορισμού διάφοροι τύποι μεταβλητών οι οποίοι χρησιμοποιούνται ανάλογα με τον τύπο των δεδομένων των οποίων επιθυμούμε να «αποθηκεύσουμε» την τιμή. Οι μεταβλητές μπορούν να καταχωρήσουν οποιαδήποτε τιμή (όχι μόνο αριθμητική). Στον πίνακα 4.1. διακρίνονται οι διάφοροι τύποι μεταβλητών καθώς και τα ορίσματα που δέχονται ανάλογα με το τύπο.

Η δήλωση κάθε μεταβλητής γίνεται με μια συγκεκριμένη (δεσμευμένη από τη Visual Basic) έκφραση. Ο τρόπος δήλωσης διαφέρει ανάλογα με το αν επιθυμούμε τη χρήση της τιμής κάποιας μεταβλητής «δημόσια» στο πρόγραμμα μας ή μόνο «τοπικά». Στην πραγματικότητα υπάρχουν περισσότεροι τρόποι δήλωσης μεταβλητών από τις δυο βασικές που αναλύονται παρακάτω.

### Τοπική Δήλωση Dim

Σύνταξη:

**Dim** My variable **as** Τύπος

Η Dim χρησιμοποιείτε μέσα σε οποιαδήποτε κλήση συμβάντος, αλλά ποτέ έξω από αυτές. Η τιμή της μεταβλητής που δηλώνεται κατά αυτόν τον τρόπο κρατάει την τιμή της μόνο μέσα στην υπορουτίνα στην οποία δηλώνεται.

Τύπος	Μέγεθος	Επεξήγηση	Όρισμα που δέχεται
Byte	1 byte	Μικρός ακέραιος	ακέραιο αριθμό στο [0...255]
Integer	2 bytes	Ακέραιος	ακέραιο αριθμό στο [-32768...32767]
Long	4 bytes	Μεγάλος ακέραιος	ακέραιο αριθμό στο [-2147483648...2147483647]
Single	4 bytes	Πραγματικός αριθμός απλής ακρίβειας	Πραγματικό αριθμό στο [-3.402823*10 <sup>38</sup> ... - 1.401298*10 <sup>-45</sup> ] για αρνητικούς και [1.401298*10 <sup>-45</sup> ... 3.402823*10 <sup>38</sup> ] για θετικούς.
Double	8 bytes	Πραγματικός αριθμός διπλής ακρίβειας	Πραγματικό αριθμό στο [-1.79769313486232*10 <sup>308</sup> ... - 4.94065645841247*10 <sup>-324</sup> ] για αρνητικούς και [4.94065645841247*10 <sup>-324</sup> ...1.79769313486232*10 <sup>308</sup> ] για θετικούς.
String	1 byte / χαρακτήρα	Κείμενο	Οτιδήποτε, αλλά το αναγνωρίζει ως κείμενο
Variant	8 bytes	Διάφοροι	Όλους τους τύπους δεδομένων.
Boolean	1 byte	Λογικός	True ή False

Πίνακας 4.1. Μεταβλητές στην Visual Basic 6.

**Δήλωση Δημόσια Public**

Σύνταξη:

**Public** My variable **as** Τύπος

Η Public χρησιμοποιείτε μόνο στο τμήμα Declarations του κώδικα (Δηλώσεις). Δεν πρέπει δηλαδή να βρίσκεται μέσα σε κάποια υπορουτίνα .

**4.6. Συναρτήσεις API**

Θα συζητηθούν ορισμένα σημεία που αφορούν στα προγράμματα εφαρμογών με τα οποία μπορεί κανείς να διαχειριστεί την είσοδο και την έξοδο δεδομένων προς τις συσκευές HID. Είναι προφανές ότι θα πρέπει να χρησιμοποιήσουμε μια γλώσσα προγραμματισμού ικανή να διαχειριστεί τους οδηγούς συσκευών των Windows. Αυτός είναι και ο λόγος που χρησιμοποιήσαμε ως γλώσσα προγραμματισμού την Visual Basic 6, η οποία χειρίζεται το υλικό του υπολογιστή με τη βοήθεια των συναρτήσεων API (Application Programmers's Interface) των Windows. Πρόκειται για συναρτήσεις που μπορούν να χρησιμοποιήσουν οι εφαρμογές για να επικοινωνήσουν με το λειτουργικό σύστημα. Στη VB οι συναρτήσεις αυτές είναι ορισμένες με τον ίδιο τρόπο που ορίζονται όλες οι συναρτήσεις, αλλά ο κώδικας που εκτελείτε κατά την κλήση της συνάρτησης δεν βρίσκεται μέσα στη ίδια τη εφαρμογή, αλλά σε αρχεία .DLL (Dynamic Linked Library), που περιέχονται στα Windows. Έτσι, κατά τις δηλώσεις των συναρτήσεων API, ορίζονται τα κατάλληλα αρχεία βιβλιοθήκης .DLL. Για επικοινωνία με χειριστήρια HIDs τα Windows περιέχουν τα παρακάτω αρχεία βιβλιοθήκης.

hid.dll, setupi.dll, kernell32.dll

Τα αρχεία αυτά κανονικά βρίσκονται στον καθιερωμένο φάκελο του λειτουργικού συστήματος \windows\system. Καλό είναι να υπάρχουν και στο φάκελο εργασίας της

Visual Basic, δηλαδή στο φάκελο που είναι εγκαταστημένη η VB6. Πολλές συναρτήσεις API λαμβάνουν σαν είσοδο και επιστρέφουν σαν έξοδο δομές δεδομένων που αποτελούνται από διάφορους τύπους μεταβλητών, που πρέπει να οριστούν κατάλληλα. Έτσι η συνάρτηση API **HidD\_GetAttributes**, που χρησιμοποιείται για να επιστρέφει το VID (Vendor ID) και το PID (Product ID) της συσκευής δηλώνεται ως εξής:

```
Public Declare Function HidD_GetAttributes _
    Lib "hid.dll" _
    (ByVal HidDeviceObject As Long, _
    By Ref Attributes As HIDD_ATTRIBUTES) _
    As Long
```

Ο τύπος της δομής **HIDD\_ATTRIBUTES** οφείλει να έχει δηλωθεί κατάλληλα:

```
Public Type HIDD_ATTRIBUTES
    Size As Long
    VendorID As Integer
    ProductID As Integer
    VersionNumber As Integer
End Type
```

Όταν καλείτε η συνάρτηση **HidD\_GetAttributes** τα πεδία που ορίζονται σε αυτήν αποκτούν νέες τιμές και περνούν στην εφαρμογή σε κατάλληλες μεταβλητές. Ένα παράδειγμα κλήσης της συνάρτησης είναι το εξής:

```
Dim Result As Long
Result = HidD_GetAttributes(HidDevice, _
DeviceAttributes)
```

Όταν γίνει επιτυχής κλήση της συνάρτησης **HidD\_GetAttributes**, η δομή **DeviceAttributes** περιέχει το **VendorID**, το **ProductID** και την παραλλαγή του μοντέλου της συσκευής, όπως αυτά έχουν ανακτηθεί από τους περιγραφείς της συσκευής, κατά την διαδικασία της απαρίθμησης. Η μεταβλητή **HidDevice** περιέχει το κανάλι ή τη διαδρομή (pathname) της συσκευής, η οποία μπορεί να χρησιμοποιηθεί από τις συναρτήσεις API για ανταλλαγή δεδομένων μ' αυτήν. Ας θυμηθούμε ότι μπορεί να υπάρχουν πολλές συσκευές USB συνδεδεμένες στον ελεγκτή USB του ξενιστή και η κάθε μια πρέπει να έχει την δικιά της διαδρομή. Για να βρεθεί η τιμή της **HidDevice** για τη συσκευή μας πρέπει να προηγηθεί η κλήση άλλων συναρτήσεων API, με τη σειρά που γίνεται στο προτεινόμενο παράδειγμα. Μετά από επιτυχής κλήση της **HidD\_GetAttributes** με την παραπάνω μορφή, η μεταβλητή **Result** είναι μη μηδενική. Το παράδειγμα της εντολής API που αναφέραμε δίνει μια εικόνα για το πώς ανακτούμε πληροφορίες για τις συσκευές HID από το λειτουργικό σύστημα. Όλο το πρώτο μέρος του προγράμματος της εφαρμογής αποτελείται από διαπλεκόμενες κλήσεις συναρτήσεων API, που περνούν τιμές η μια στη άλλη. Ο σκοπός είναι να δημιουργηθεί ένα κανάλι επικοινωνίας με τη συγκεκριμένη συσκευή, για την οποία είναι φτιαγμένη η εφαρμογή. Όταν ταυτοποιηθεί η συσκευή με βάση το VID και το PID, τότε η συνάρτηση API **CreateFile** ανοίγει την απαραίτητη διαδρομή (**HidDevice**) προς τη συσκευή προκειμένου να μεταφέρουμε δεδομένα από και προς τη συσκευή, όπως κάνουμε σε οποιοδήποτε αρχείο δεδομένων. Οι συναρτήσεις API για την εγγραφή και την ανάγνωση δεδομένων είναι οι **WriteFile** και η **ReadFile**.

## 5. Η Κατασκευή του υλικού μέρους

Σε αυτό το κεφάλαιο θα γίνει ανάλυση γύρω από το υλικό (Hardware) της όλης κατασκευής. Συγκεκριμένα θα μελετηθούν οι ελεγκτές (translators) που παρέχουν την κατάλληλη ψηφιακή λογική πχ (κώδικας GRAY) για την οδήγηση του διπολικού (bipolar) βηματικού κινητήρα. Θα μελετηθούν επίσης τα κυκλώματα οδήγησης (drivers) τα οποία παρέχουν το απαραίτητο ρεύμα για την οδήγηση των βηματικών κινητήρων. Θα γίνει αναφορά στο κύκλωμα του USB ελεγκτή που ενσωματώνει τον μικροελεγκτή cy7c63001. Τέλος θα γίνουν κάποιες σχετικές αναφορές γύρω από τη σχεδίαση της πλακέτας.

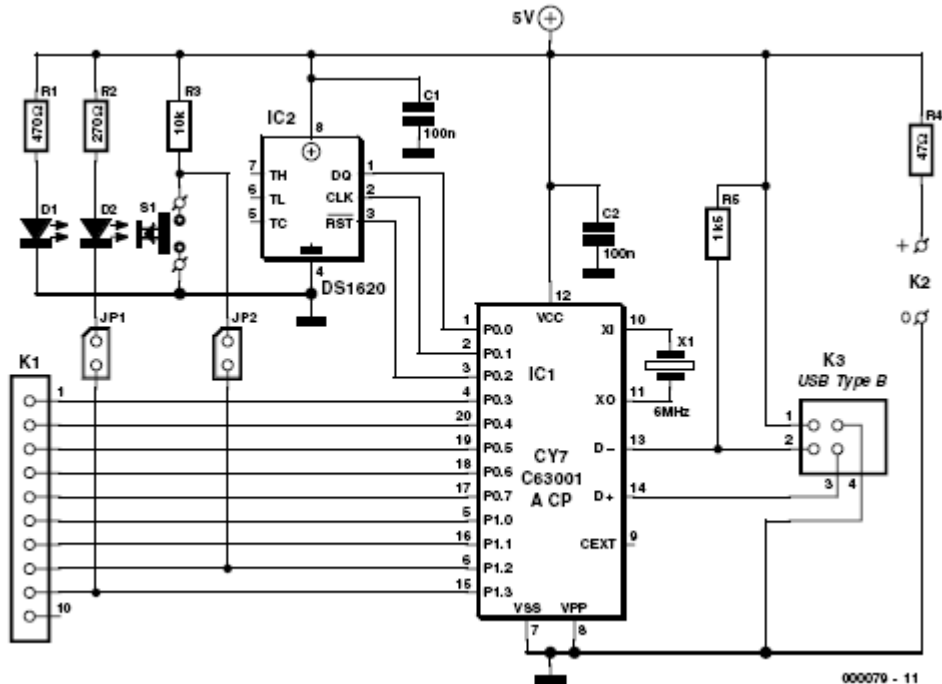
### 5.1. Ελεγκτής USB

Το πρώτο τυπωμένο κύκλωμα είναι ο USB ελεγκτής (controller). Το κύκλωμα βασίζεται στο μικροελεγκτή cy7c63001 της CYPRESS για τον οποίο έχει γίνει σχετική αναφορά στην ενότητα 2.8. Δύο γραμμές δεδομένων (data lines) το D+ και το D- χρησιμοποιούνται για την μεταφορά των δεδομένων από τον ξενιστή προς το μικροελεγκτή και αντίστροφα. Οι δύο αυτές γραμμές δεδομένων όπως παρουσιάζεται στο σχήμα 5.1. συνδέονται απ' ευθείας σε δυο ακροδέκτες του μικροελεγκτή. Ένας αντιστάτης pull up τιμής 1K5 Ohm σηκώνει τη γραμμή D- στην τάση των 5 Volt, ώστε να πληροφορήσει τον ξενιστή ότι πρόκειται για συσκευή χαμηλής ταχύτητας, όπως προβλέπεται από τις προδιαγραφές της USB1.1. Ο πυκνωτής C1 χρησιμοποιείται για απόζευξη υψηλών συχνοτήτων και χρησιμοποιείται για τη σταθερότητα της τροφοδοσίας VUSB.

Ο USB ελεγκτής περιλαμβάνει τις ακόλουθες λειτουργίες.

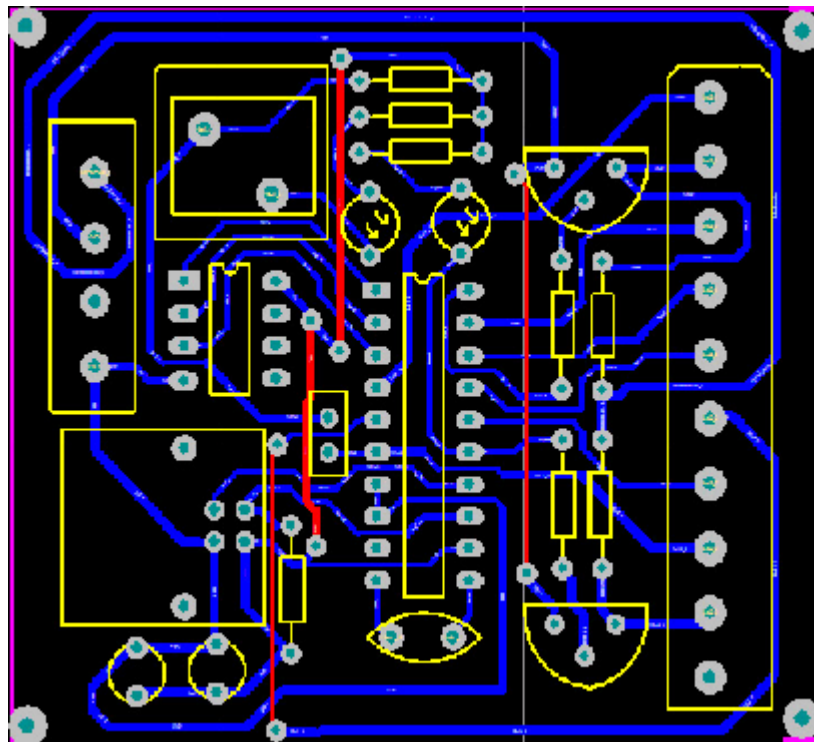
- Μέτρηση θερμοκρασίας. Τα τρία πρώτα bit της πόρτας P0 είναι αφιερωμένα για την μέτρηση της θερμοκρασίας του περιβάλλοντος χώρου. Το αισθητήριο που χρησιμοποιείται είναι το DS1620 της Maxim Dallas. Μια 9-bit τιμή διαβάζεται από το θερμόμετρο κάθε 10 ms, αντιστοιχίζεται σε μια θερμοκρασία στην κλίμακα των βαθμών Κελσίου και επιστρέφεται σε μια ετικέτα (label) στο περιβάλλον της Visual Basic 6.
- Πλήκτρο. Ένα πλήκτρο (push button) χρησιμοποιείται για να αλλάξει το περιεχόμενο μιας ετικέτας στο περιβάλλον της Visual Basic. Η ετικέτα επιστρέφει αλλαγή από "0" σε "1" και από "1" σε "0"
- Ρύθμιση φωτεινότητας μιας Φώτο εκπέμπουσας διόδου (led). Κάθε ένας από τους 12 ακροδέκτες του μικροελεγκτή περιλαμβάνει ένα μετατροπέα από ψηφιακό σε αναλογικό ακρίβειας 4 bit (DAC). Έτσι έχουμε την δυνατότητα μέσω κατάλληλου προγραμματισμού του μικροελεγκτή να μεταβάλλουμε το ρεύμα στην έξοδο κάθε ακροδέκτη σε μια από 16 διαφορετικές στάθμες. Η πόρτα P1 παρέχει ικανά ρεύματα για να οδηγήσει απ' ευθείας led. Στην εφαρμογή ο ακροδέκτης P1.3 συνδέεται με το πράσινο led. Μέσα από το περιβάλλον εργασίας της Visual Basic επιτυγχάνουμε να μεταβάλλουμε τη φωτεινότητα του από την θέση 0000 (ελάχιστη φωτεινότητα) στη θέση 1111 (μέγιστη φωτεινότητα).
- Παροχή σήματος ρολογιού (clock) για το βήμα των κινητήρων. Δύο ακροδέκτες του μικροελεγκτή χρησιμοποιούνται για να παρέχουν το επιθυμητό βήμα (step) για την κίνηση των βηματικών κινητήρων. Δυο hjt τρανζίστορ σε συνδεσμολογία διακόπτη χρησιμοποιούνται για να μετατρέψουν τις στάθμες 0-5 Volt που αντιστοιχούν στο βήμα (step) του καθενός σε 0-12 Volt.

- Παροχή σήματος για τη κατεύθυνση των κινητήρων (direction). Η αλλαγή του δυναμικού ενός ακροδέκτη του μικροελεγκτή από σε +5 Volt σε 0 Volt και αντίστροφα απαιτείται για την αλλαγή της κατεύθυνσης του βηματικού κινητήρα.



Σχήμα 5.1. Σχηματικό διάγραμμα του ελεγκτή (USB controller).

Η φώτο εκπέμπουσα διάδος D1 (κόκκινο χρώμα) του σχήματος 5.1. μας δείχνει ότι η κατασκευή συνδέεται κανονικά με της τάση τροφοδοσίας της USB, ενώ D2 μας ενημερώνει ότι η διαδικασία της απαρίθμησης ολοκληρώθηκε με επιτυχία. Τέλος ο χρονισμός του μικροελεγκτή γίνεται μέσω ενός κρυστάλλου ιδιοσυχνότητας έξι MHz.



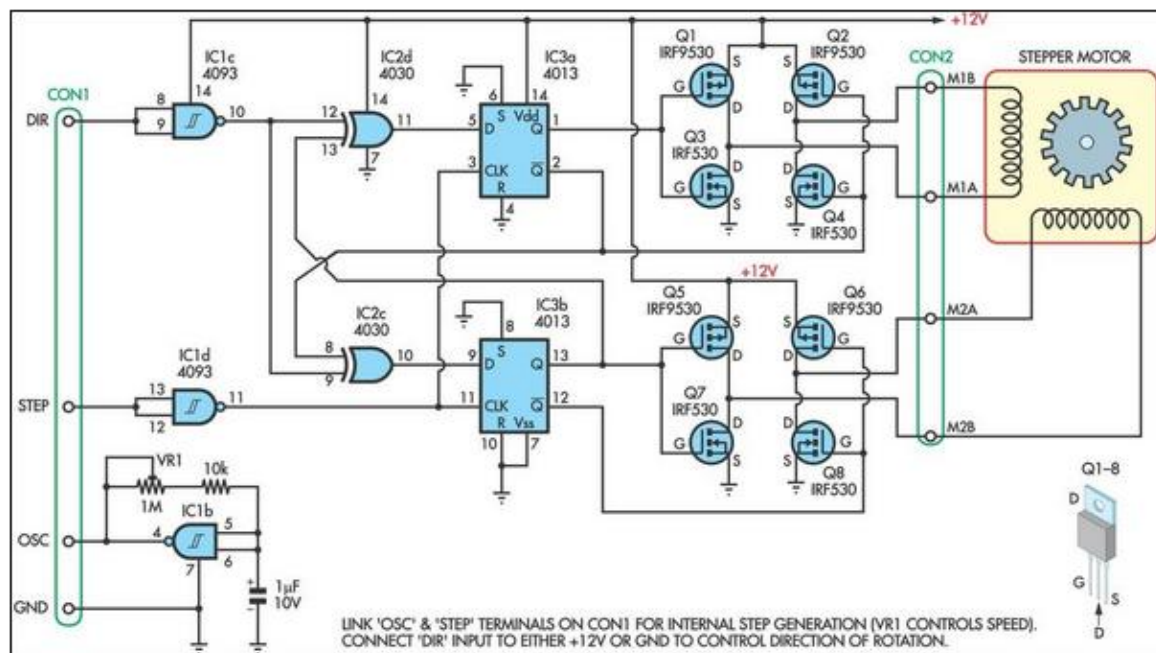
Σχήμα 5.2. Το τυπωμένο κύκλωμα του USB Controller.



## 5.2. Κύκλωμα οδήγησης του βηματικού κινητήρα 4 ακροδεκτών.

Όπως έχει αναφερθεί ένας βηματικός κινητήρας δεν μπορεί να περιστρέφεται συνεχώς (= χωρίς ασυνέχειες), αλλά μόνο με βήματα. Έτσι βάζοντας +12 ή 0 Volt στην είσοδο "κατεύθυνση" του κυκλώματος δεν πρόκειται να συμβεί τίποτα, εφόσον δεν στείλουμε έναν παλμό στην δεύτερη είσοδο του κυκλώματος (που ονομάζεται Βήμα (Step)). Κάθε φορά που στέλνουμε έναν παλμό (το κύκλωμα ανταποκρίνεται στο θετικό μέτωπο του παλμού) στην είσοδο Βήμα (Step), ο βηματικός κινητήρας κάνει ένα βήμα (μιας καθορισμένης γωνίας, που εξαρτάται από τον τύπο του κινητήρα). Αν ξαναστείλετε έναν παλμό (σαν τον προηγούμενο), ο κινητήρας θα ξανακάνει ένα βήμα. Αν στείλετε πέντε συνεχόμενους παλμούς, ο βηματικός κινητήρας θα κάνει πέντε βήματα κ.ο.κ. Η είσοδος "κατεύθυνση" καθορίζει την φορά περιστροφής του βηματικού κινητήρα. Αν αυτή είναι 12 Volt τότε ο κινητήρας περιστρέφεται προς τη μια κατεύθυνση. Όταν η τάση είναι μηδέν volt, τότε ο βηματικός κινητήρας περιστρέφεται προς την αντίθετη κατεύθυνση.

Το κύκλωμα οδήγησης του βηματικού κινητήρα 4 ακροδεκτών αποτελείται από δυο επιμέρους κυκλώματα. Το πρώτο κύκλωμα είναι ο ελεγκτής (translator) ο οποίος δημιουργεί μια συγκεκριμένη ψηφιακή λογική (διπολική οδήγηση). Θυμίζουμε ότι η διπολική λειτουργία απαιτεί τα τυλίγματα να διαρρέονται συνεχώς από ρεύμα, ενώ σε κάθε βήμα αλλάζει η φορά του ρεύματος σε ένα από τα δυο τυλίγματα.. Το δεύτερο κύκλωμα είναι το κύκλωμα οδήγησης το οποίο παρέχει ικανό ρεύμα στα τυλίγματα του βηματικού κινητήρα για να τον θέσει σε κίνηση.



Σχήμα 5.3. Κύκλωμα οδήγησης του διπολικού κινητήρα.

Το κύκλωμα του ελεγκτή (translator) αποτελείται από 2 πύλες XOR δύο εισόδων ενσωματωμένες στο ολοκληρωμένο κύκλωμα CD4030, δυο flip-flop τύπου D ενσωματωμένα στο ολοκληρωμένο CD4013 και δυο smith trigger που ενσωματώνονται στο ολοκληρωμένο κύκλωμα CD4093. Οι δυο τελευταίες πύλες δέχονται στην είσοδο τους την κατεύθυνση (direction) και το βήμα (step) από δύο ακροδέκτες του μικροελεγκτή. Η ακολουθία των bits που δημιουργεί το κύκλωμα του translator στην έξοδο του κυκλώματος είναι τέτοια ώστε τα τυλίγματα να διαρρέονται συνεχώς από ρεύμα, ενώ σε κάθε βήμα αλλάζει η φορά του ρεύματος σε ένα από τα δυο τυλίγματα. η



ψηφιακή λογική αυτή είναι όμοια με αυτή του κώδικα GRAY. Στον πίνακα 5.1. παρουσιάζεται η συγκεκριμένη ακολουθία. Όσο αφορά το κύκλωμα οδήγησης, αποτελείται από 8 HEXFET. Πρόκειται ουσιαστικά για δυο γέφυρες H (H-Bridge). Κάθε μια από αυτές οδηγεί ένα τύλιγμα του βηματικού κινητήρα. Τα P-MOS ονομάζονται IRF 540 και τα N-MOS IRF 9540 Το κάθε ένα από αυτά είναι ικανό να παρέχει ρεύμα κοντά στα 20 Amper. Ρεύμα παραπάνω από αρκετό για τις απαιτήσεις του βηματικού κινητήρα.

Δυαδικό σύστημα	Κώδικας Gray
0000	0101
0001	0110
0010	1010
0011	1001
0100	0101
0101	0110
0110	1010
0111	1001

Πίνακας 5.1. Κώδικας GRAY.

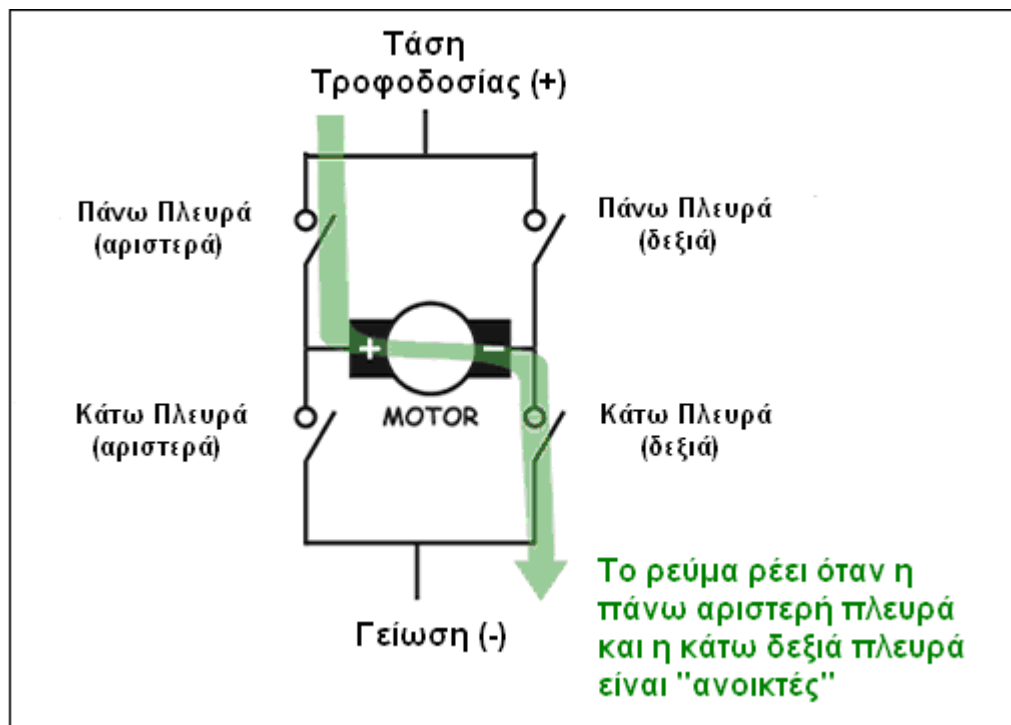
### Γέφυρα H

Ας αρχίσουμε πρώτα με την ονομασία Γέφυρα H (H-Bridge). Καλείτε έτσι διότι έχει τέσσερα διακοπτικά στοιχεία στην γωνία του γράμματος H και ο κινητήρας αναπαριστά την μεσαία νοητή «γραμμή» (cross bar). Η βασική γέφυρα H απεικονίζεται στο σχήμα 5.2. Τα τέσσερα αυτά στοιχεία συχνά καλούνται ως εξής:

«Πλευρά πάνω αριστερά» (high side left) «Πλευρά πάνω δεξιά» (high side right),

«Πλευρά κάτω αριστερά» (low side left), «Πλευρά κάτω δεξιά» (low side right).

Οι διακόπτες «ανοίγουν» πάντα σε ζευγάρια, είτε (high side left) με (low side right) είτε (high side right) με (low side left). Σε καμία περίπτωση όμως δεν θα πρέπει να ανοίγουν ταυτόχρονα οι διακόπτες της ίδιας πλευράς της γέφυρας. Σε περίπτωση που συμβεί το παραπάνω τότε δημιουργείται βραχυκύκλωμα μεταξύ των πόλων της μπαταρίας.



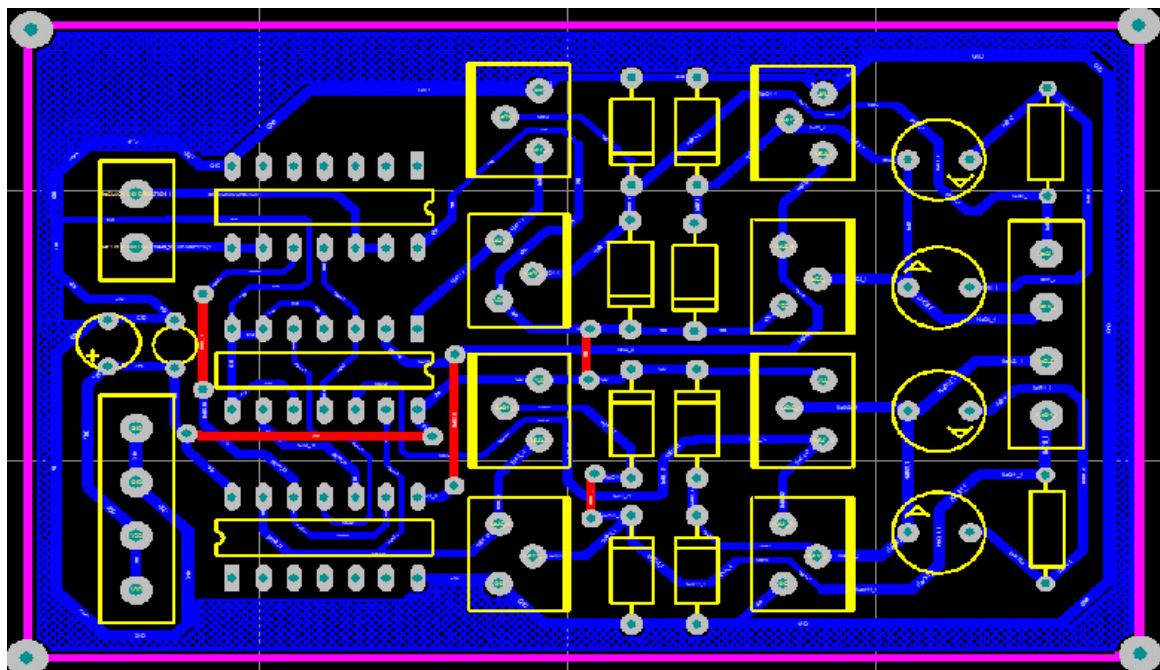
Σχήμα 5.4. Κατεύθυνση ρεύματος σε μια Γέφυρα H.

Στο πίνακα 5.2. απεικονίζεται ένας πίνακας αληθείας για τις πιθανές καταστάσεις που μπορούν να υπάρξουν σε μια γέφυρα Η.

High Side Left	High Side Right	Low Side Left	Low Side Right	Λειτουργία
On	Off	Off	On	Ο κινητήρας κατευθύνεται Δεξιόστροφα (Clockwise)
Off	On	On	Off	Ο κινητήρας κατευθύνεται Αριστερόστροφα (Counter-Clockwise)
On	On	Off	Off	Φρενάρει Και Επιβραδύνεται (Brakes and Decelerates)
Off	Off	On	On	Φρενάρει Και Επιβραδύνεται (Brakes and Decelerates)

Πίνακας 5.2. Απεικόνιση ενός πίνακα αληθείας για τις πιθανές καταστάσεις που μπορούν να υπάρξουν σε μια γέφυρα Η.

Στις τελευταίες δυο γραμμές του πίνακα φαίνεται η απόκριση του κινητήρα στην περίπτωση που δεν «ανοιχτούν - κατάσταση ON» τα κατάλληλα στοιχεία. Στις περιπτώσεις αυτές ο κινούμενος κινητήρας δημιουργεί μια τάση, η οποία προσπαθεί να εξαναγκάσει τον κινητήρα να κινηθεί προς την αντίθετη κατεύθυνση. Αυτό έχει ως αποτέλεσμα, ο κινητήρας να σταματήσει γρήγορα την περιστροφή και συχνά καλείτε φρενάρισμα (braking). Εννοείτε πως υπάρχει και μια κατάσταση στην οποία όλα τα στοιχεία είναι κλειστά (turned off). Στην περίπτωση αυτή ο κινητήρας σταματάει εάν βρισκόταν σε κίνηση, και δεν κάνει τίποτα εάν ήταν ακίνητος. Σε σχέση με το αρχικό σχηματικό διάγραμμα έχουν προστεθεί οκτώ διόδοι που προστατεύουν τα HEXFET από ανάστροφα ρεύματα που προέρχονται από τα τυλίγματα του βηματικού κινητήρα καθώς και τέσσερις φώτο εκπέμπουσες διόδοι που αναπαριστούν την ροή του ρεύματος στα τυλίγματα του βηματικού κινητήρα σε κάθε βήμα.

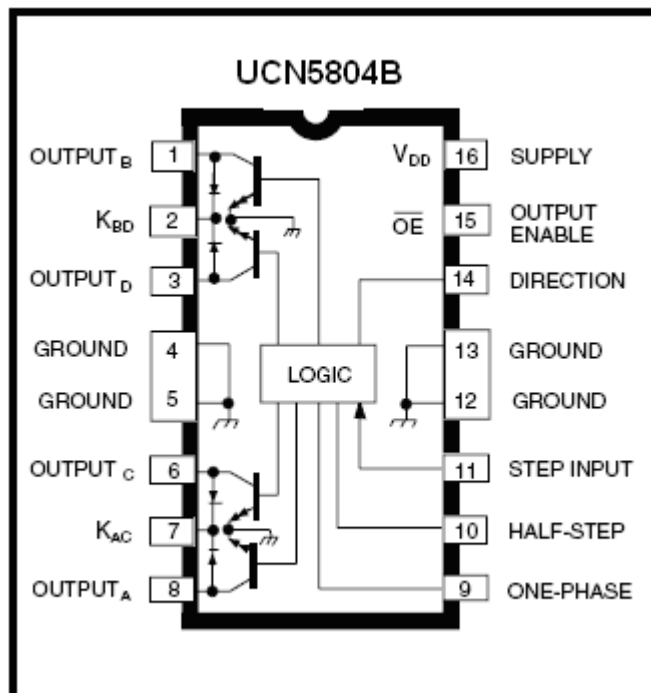


Σχήμα 5.5. Το τυπωμένο κύκλωμα του διπολικού βηματικού κινητήρα.

Στο σχήμα 5.5 απεικονίζεται η ολοκληρωμένη σχεδίαση του τυπωμένου κυκλώματος μέσα από το PROTEL 99 SE . Το κύκλωμα ελέγχου (translator circuit) όπως και το κύκλωμα οδήγησης (driver circuit) περιλαμβάνονται στο ίδιο P.C.B. Η σχεδίαση είναι μονής όψης (single layer).

### 5.3. Κύκλωμα οδήγησης του βηματικού κινητήρα 5 ακροδεκτών.

Η καρδιά του συγκεκριμένου Unipolar ελεγκτή είναι το ολοκληρωμένο κύκλωμα UCN5804B. Ο συγκεκριμένος ελεγκτής εκτός του ότι παρέχει την κατάλληλη ψηφιακή λογική, είναι ικανός να οδηγήσει τα τυλίγματα του κινητήρα με ρεύμα που φτάνει τα 1.25 Amper για κάθε φάση κάτι που μας απλοποιεί σε μεγάλο βαθμό την σχεδίαση του όλου κυκλώματος. Συγκεκριμένα έχουμε την δυνατότητα να επιλέξουμε μια από τις τρεις δυνατές ακολουθίες οδήγησης που συναντάμε στη μονοπολική λειτουργία. Δηλαδή την οδήγηση μόνης φάσης (Single phase), την οδήγηση διπλής φάσης (Dual phase), καθώς και την οδήγηση μισού βήματος (Half step). Στο σχήμα 5.4. απεικονίζεται το εσωτερικό του ελεγκτή, καθώς και η ονομασία των ακροδεκτών του.



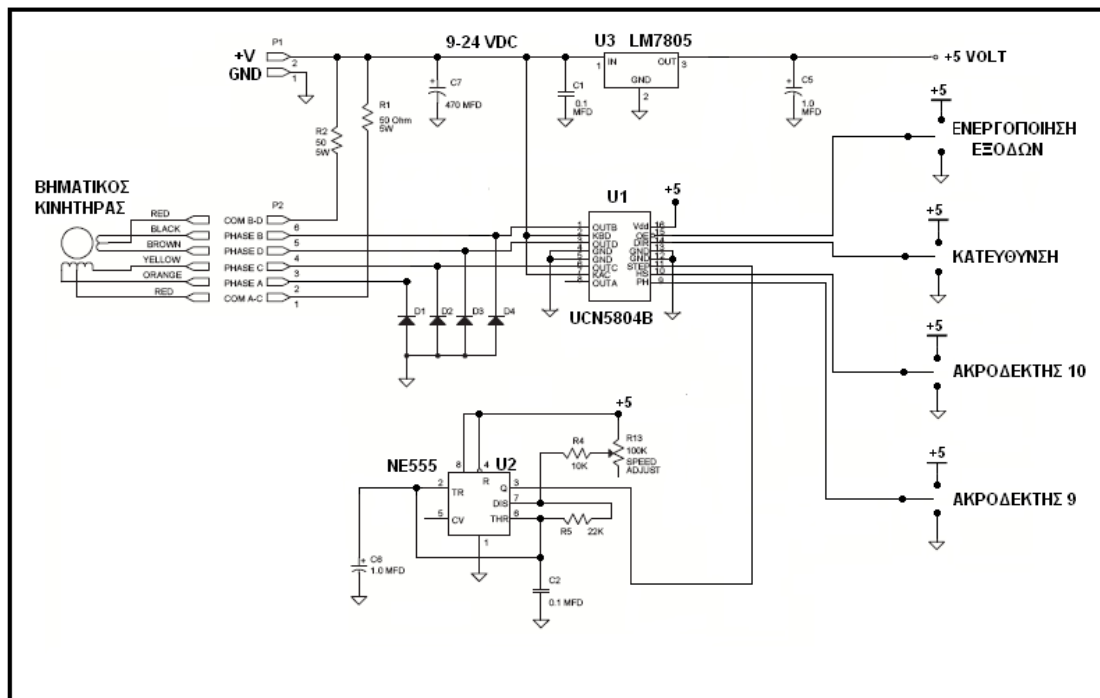
Σχήμα 5.6. Εσωτερικό του ελεγκτή UCN5804B.

#### Ακροδέκτες

Όπως παρατηρούμε από το σχήμα οι ακροδέκτες 9 και 10 είναι εισοδοι, που μας επιτρέπουν να επιλέξουμε ένα από τους συνολικά τρεις τρόπους οδήγησης. Ο ακροδέκτης 11 είναι η είσοδος των παλμών ρολογιού (clock pulses). Επίσης έχουμε την δυνατότητα να απενεργοποιήσουμε όλες τις εξόδους του ελεγκτή μέσω του ακροδέκτη 15 (Output Enable), αλλά θέτοντας τον συγκεκριμένο ακροδέκτη σε κατάσταση "1" (High). Από την δεξιά μεριά του σχήματος παρατηρούμε τις τέσσερις εξόδους του κυκλώματος (τέσσερις ακροδέκτες) που παρέχουν το απαραίτητο ρεύμα για την κίνηση του κινητήρα. Οι ακροδέκτες  $K_{AC}$  και  $K_{BD}$  πρέπει να συνδεθούν με την τάση εισόδου του κυκλώματος που στη συγκεκριμένη περίπτωση είναι 12 Volt. Τέλος υπάρχουν και τέσσερις ακροδέκτες με την ονομασία Γείωση (Ground).

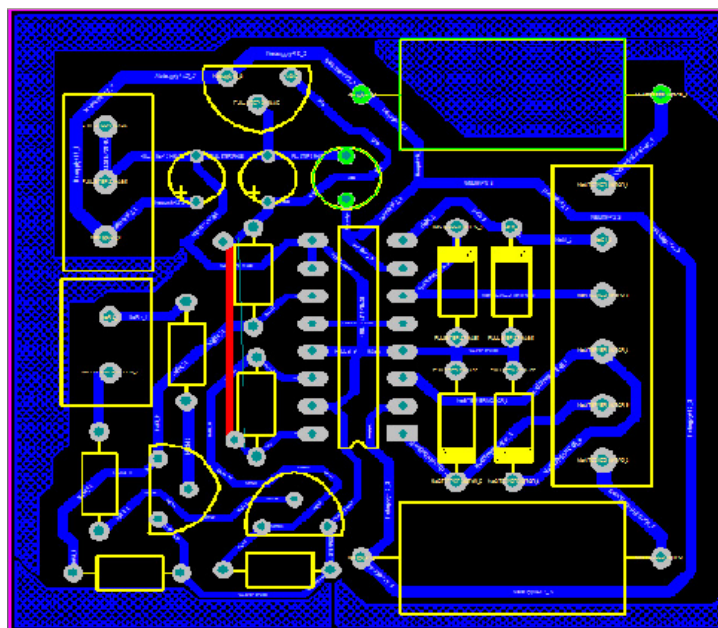
### Το σχηματικό διάγραμμα

Στο σχήμα 5.7. απεικονίζεται το κύκλωμα οδήγησης του βηματικού κινητήρα με 5 ακροδέκτες (Unipolar Stepper Motor Controller) πάνω στον οποίο βασίστηκε η τελική σχεδίαση του συγκεκριμένου ελεγκτή. Στο σχήμα 5.7. απεικονίζεται εκτός του UCN 5804 και ένα ολοκληρωμένο με την ονομασία NE555. Ο NE555 έχει συνδεσμοποιηθεί ως ασταθής πολυδονητής (astable multivibrator) το οποίο μας παρέχει τους παλμούς ρολογιού (clock pulses) στην είσοδο βήμα (step) του UCN5804. Πρέπει να σημειωθεί ότι οι παλμοί του ρολογιού στην τελική κατασκευή δίνονται μέσω του μικροελεγκτή CY7C63001. Στο πάνω μέρος του σχήματος παρατηρούμε έναν σταθεροποιητή τάσης (LM7805). Σκοπός αυτού του εξαρτήματος είναι να παρέχει σταθεροποιημένη τάση των 5 volt στα διάφορα μέρη του κυκλώματος που χρειάζεται. Παρατηρούμε επίσης τέσσερις διόδους ανάστροφα πολωμένες οι οποίες συνδέονται στους τέσσερις ακροδέκτες εξόδου του UCN5804B. Η χρήση αυτών των διόδων αποσκοπεί στο να αποφευχθεί η καταστροφή του UCN5804B σε περίπτωση που δημιουργηθούν ανάστροφα ρεύματα από τα τυλίγματα του κινητήρα. Τέλος κρίθηκε απαραίτητη η χρήση δυο αντιστάσεων ισχύος στο κύκλωμα, με σκοπό τον περιορισμό του ρεύματος που εισάγεται στις μεσαίες λήψεις του κινητήρα.



Σχήμα 5.7. Σχηματικό διάγραμμα του με τον UCN 5804.

Στη συγκεκριμένη κατασκευή μπορούμε να χρησιμοποιήσουμε τόσο μονοπολικούς βηματικούς κινητήρες 5 και 6 ακροδεκτών. Έχει επιλεγεί μονοπολική λειτουργία οδήγησης - μονής φάσης .



Σχήμα 5.8. Το τυπωμένο κύκλωμα του μονοπολικού βηματικού κινητήρα.

Στο σχήμα 5.8 απεικονίζεται η ολοκληρωμένη σχεδίαση του τυπωμένου κυκλώματος μέσα από το PROTEL 99 SE . Η σχεδίαση είναι μονής όψης (single layer).

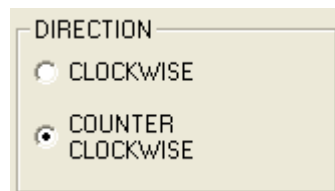
## 6. Οδηγίες χρήσης

Στην ενότητα που ακολουθεί περιγράφονται τα στοιχεία ελέγχου που χρησιμοποιούνται στην εφαρμογή καθώς και οι λειτουργίες που επιτελούν.

### Κουμπιά επιλογής (Option Button)

Τα κουμπιά επιλογής εμφανίζονται σε ομάδες από τις οποίες ο χρήστης μπορεί να επιλέξει μόνο ένα. Κάθε φορά που επιλέγεται κάποιο, επιλέγονται και τα υπόλοιπα της ίδιας ομάδος. Λειτουργεί και αυτό σαν διακόπτης και η ιδιότητα του Checked γίνεται True όταν είναι επιλεγμένο και False όταν δεν είναι επιλεγμένο.

Ο έλεγχος της κατεύθυνσης επιτυγχάνεται μέσω της χρήσης δυο κουμπιών επιλογής. Για να κινηθεί ο βηματικός κινητήρας δεξιόστροφα επιλέγουμε Clockwise, και να κινηθεί αριστερόστροφα επιλέγουμε Counter Clockwise.



Σχήμα 6.1. Επιλογή Κατεύθυνσης.

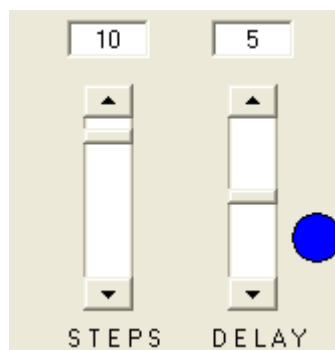
### Πλαίσιο (Frame)

Το εργαλείο αυτό χρησιμοποιείται για τη σχεδίαση πλαισίων που σκοπό έχουν την ομαδοποίηση άλλων στοιχείων. Παρατηρώντας το σχήμα 6.1. βλέπουμε ότι τα δυο κουμπιά επιλογής ομαδοποιούνται μέσα σε ένα frame.

### Κατακόρυφη γραμμή κύλισης (Vertical Scroll Bar)

Η επιλογή των επιθυμητών βημάτων (steps) καθορίζεται από ένα Vertical Scroll Bar με την ονομασία STEPS. Μετακινώντας τον κέρσορα προς τα κάτω αυξάνουμε τον αριθμό των βημάτων.

Η επιλογή της ταχύτητας του κινητήρα καθορίζεται από ένα Vertical Scroll Bar με την ονομασία DELAY. Έχουμε την δυνατότητα να μετακινήσουμε τον κέρσορα σε μια από δέκα διαφορετικές τιμές. Όσο μεγαλύτερη είναι η τιμή του text box που βρίσκεται ακριβώς πάνω από το VscrollBar τόσο μεγαλύτερη θα είναι και η συχνότητα της εναλλαγής των βημάτων από “1” σε “0” και από “0” σε “1”, και κατ’ επέκταση θα είναι και πιο γρήγορη η ταχύτητα κίνησης του δρομέα του βηματικού κινητήρα.



Σχήμα 6.2. Αριθμός Βημάτων Και Ταχύτητας Κινητήρα.



Έτσι για παράδειγμα σύμφωνα με το σχήμα 6.1. και 6.2. , μόλις πατηθεί το κουμπί Start ο κινητήρας θα εκτελέσει 10 βήματα αριστερόστροφα με την ταχύτητα που αντιστοιχεί για  $\text{delay} = 5$ .

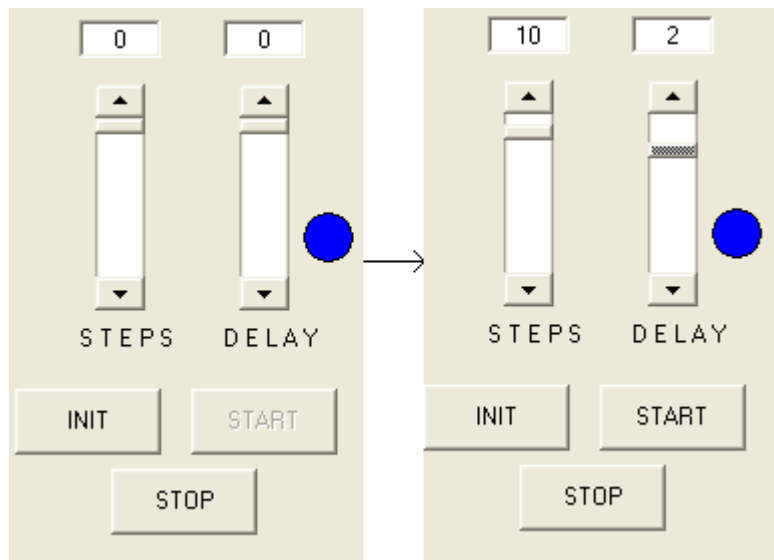
### Κουμπί (Command Button)

Είναι το συνηθέστερο εργαλείο που χρησιμοποιείται στη φόρμα. Στο εργαλείο αυτό πρέπει να γράψουμε κώδικα στο Click Event ώστε να εκτελεστεί η ενέργεια που θέλουμε όταν ο χρήστης κάνει κλικ στο κουμπί

Στο σχήμα 6.3. απεικονίζονται τα κουμπιά με ονομασία Start , Init και Stop Πατώντας το πλήκτρο **START** ο δρομέας του βηματικού κινητήρα θα εκτελέσει 10 βήματα με  $\text{delay}=2$ .

#### Σημείωση

Όπως παρατηρούμε από το σχήμα 6.3. το κουμπί Start είναι ενεργό μόνο εάν έχουμε επιλέξει βήματα και χρονοκαθυστέρηση.

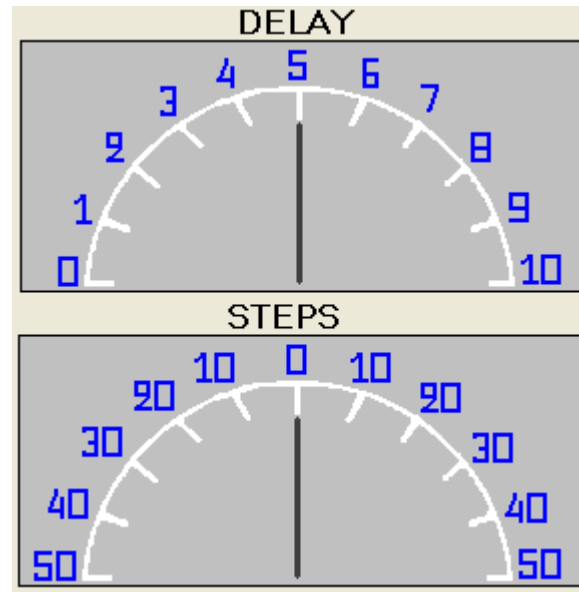


Σχήμα 6.3. Κουμπιά (Command Buttons) INIT, START, STOP.

Με το πάτημα του πλήκτρου **STOP** διακόπτουμε την κίνηση του δρομέα του βηματικού κινητήρα σε οποιοδήποτε σημείο και αν βρίσκεται. Με την χρήση του πλήκτρου **INIT** αρχικοποιούμε την θέση του βηματικού κινητήρα. Έτσι για παράδειγμα εάν ο δρομέας του κινητήρα βρίσκεται 10 βήματα αριστερά από την αρχική του θέση τότε πατώντας το πλήκτρο **INIT** ο δρομέας του βηματικού κινητήρα θα εκτελέσει 10 βήματα δεξιά επιστρέφοντας στην αρχική του θέση Το ενδεικτικό μπλε led που παρατηρούμε μας αναπαριστά την εναλλαγή των παλμών “1” και “0” που δέχονται τα κυκλώματα των ελεγκτών (translators) στην είσοδο τους (step).

### GAUGE

Για την αναπαράσταση της χρονοκαθυστέρησης καθώς και της κίνησης του δρομέα του κινητήρα χρησιμοποιούμε ένα στοιχείο ελέγχου που ονομάζεται Microhelp Gauge Control



Σχήμα 3.4 Gauge Control.

### Ετικέτα (Label)

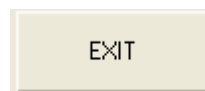
Για την απεικόνιση οδηγιών χρήσεως προς τον χρήστη καθώς και διάφορων πληροφοριών σχετικά με την κίνηση των βηματικών κινητήρων χρησιμοποιούμε το στοιχείο ελέγχου Ετικέτα (Label). Στο σχήμα 3.5. απεικονίζονται κάποιες από αυτές τις πληροφορίες.

Παρακαλώ Επιλέξτε Βήματα-(STEPS), Χρονοκαθυστέρηση-(DELAY), και κατόπιν πατήστε Εκκίνηση-(START)	Ο Δρομέας του Βηματικού Κινητήρα εκτελεί 7 βήματα με συχνότητα οδήγησης 5 Hertz αριστερόστροφα.
Τα Βήματα Ολοκληρώθηκαν	Παρακαλώ Επιλέξτε Δεξιόστροφη φορά

Σχήμα 3.5. Ετικέτα (Label).

### Έξοδος (Exit)

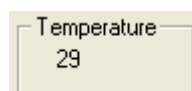
1 κουμπί εντολής (Command Button) με την ονομασία “EXIT” αποσκοπεί στο να τερματίζουμε το πρόγραμμα μας και να βγαίνουμε έξω από το περιβάλλον εργασίας της Visual Basic 6.



Σχήμα 3.6 κουμπί εντολής για την έξοδο από το πρόγραμμα.

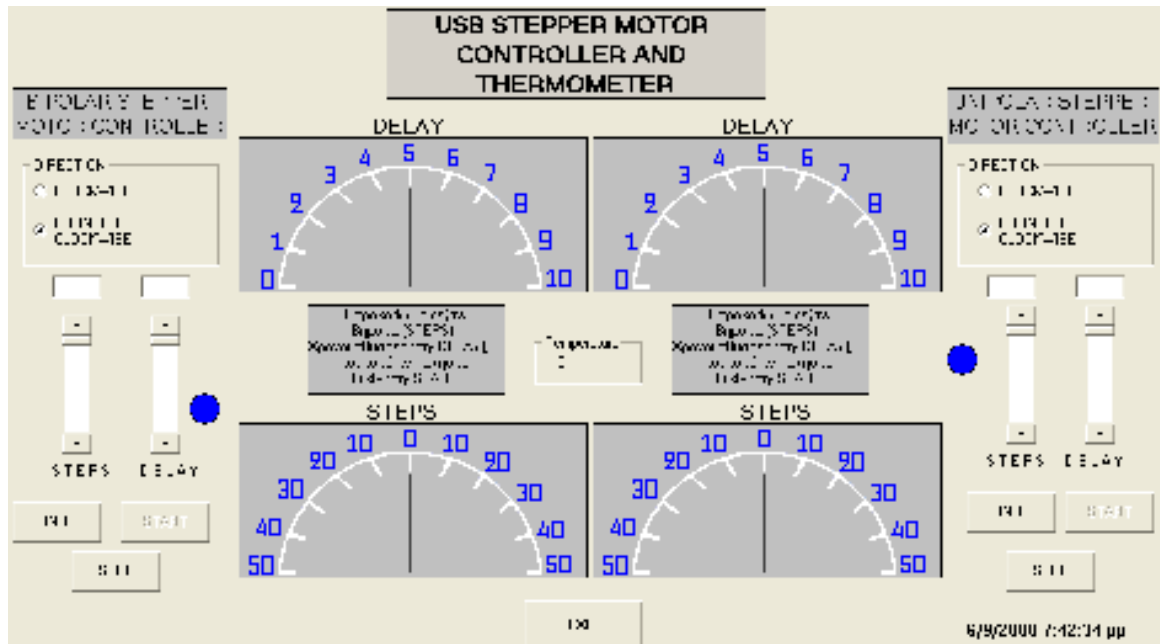
### Θερμοκρασία

Ένα label απεικονίζει την θερμοκρασία του περιβάλλοντος χώρου.





Σχήμα 3.7 Label για την απεικόνιση της θερμοκρασίας.



Σχήμα 3.8. USB Stepper Motor Controller and Thermometer.

Η συνολική φόρμα εργασίας εμφανίζεται στο σχήμα 3.8., όπου και διακρίνουμε την συνολική σχηματική αναπαράσταση της εφαρμογής για τον ακριβή έλεγχο των δυο βηματικών κινητήρων, καθώς και την ένδειξη της θερμοκρασίας του περιβάλλοντος χώρου.

## 7. Συμπεράσματα

Όπως παρουσιάζεται και από το τίτλο της πτυχιακής εργασίας “Σχεδίαση και υλοποίηση ολοκληρωμένης διεπαφής για τον έλεγχο βηματικών κινητήρων μέσω της θύρας USB ενός H/Y”, η πτυχιακή εργασία προϋποθέτει την ενασχόληση με διάφορα αντικείμενα.. Από τα πρώτα αντικείμενα που ασχολήθηκα ήταν η κατανόηση των αρχών λειτουργίας των βηματικών κινητήρων και η δημιουργία κατάλληλων κυκλωματικών διατάξεων για τον έλεγχο αυτών. Τα κυκλώματα οδήγησης θα πρέπει να πληρούν συγκεκριμένες προϋποθέσεις για τον σωστό και ακριβή έλεγχο των βηματικών κινητήρων. Έτσι από τη μια θα πρέπει να παρέχουν στα τυλίγματα των κινητήρων ικανοποιητικό ρεύμα και από τη άλλη θα πρέπει να δημιουργούν την κατάλληλη ψηφιακή λογική μέσω των κυκλωμάτων ελεγκτών (translator circuits) έτσι ώστε σε κάθε παλμό να ενεργοποιούνται τα κατάλληλα τυλίγματα.

Όσο αφορά την οδήγηση των βηματικών κινητήρων χρησιμοποίησα τόσο τον διπολικό όσο και τον μονοπολικό τρόπο οδήγησης και αυτό για να υπάρξει μια πιο ολοκληρωμένη εικόνα για την οδήγησή τους. Συγκεκριμένα στον διπολικό (bipolar) κινητήρα χρησιμοποίησα μια κλασική σχεδίαση ψηφιακής λογικής όπου λογικές πύλες αναλαμβάνουν να δημιουργήσουν τον επιθυμητό κώδικα GRAY και δυο γέφυρες H (Bridge H) να οδηγούν τα τυλίγματα του κινητήρα. Σε γενικές γραμμές δεν παρουσιάστηκαν ιδιαίτερα προβλήματα στην υλοποίηση της συγκεκριμένης κατασκευής. Τόσο τα ολοκληρωμένα όσο και τα διακριτά εξαρτήματα χρησιμοποιούνται ευρέως όποτε μπορεί κάποιος να τα προμηθευτεί εύκολα μέσα από μια αναζήτηση στο Διαδίκτυο. Κάποιες παρατηρήσεις από την πειραματική διαδικασία παρουσιάζονται παρακάτω.

Το συνολικό ρεύμα που παρείχε η μπαταρία στο κύκλωμα ήταν περίπου 2,3 Amper. Τα MOSFET του κυκλώματος έχουν την δυνατότητα να παρέχουν πολύ μεγαλύτερο ρεύμα από αυτό που απαιτούσε ο βηματικός κινητήρας οπότε δεν υπήρχε λόγος να χρησιμοποιηθούν ψήκτρες για την προστασία τους από υπερθέρμανση. Ο βηματικός κινητήρας από την άλλη, έπειτα από μερικά λεπτά λειτουργίας ζεσταίνεται σε τέτοιο βαθμό που δεν είναι εφικτό να τον ακουμπήσει κάποιος για πολλά δευτερόλεπτα. Σε σχηματικά όπως αυτό, όπου σε μια πλακέτα συνυπάρχουν εξαρτήματα που μεταφέρουν απλά σήματα αλλά και εξαρτήματα που παρέχουν μεγάλα ρεύματα θα πρέπει να γίνεται προσεκτική σχεδίαση στο P.C.B του κυκλώματος για να αποφεύγονται ανεπιθύμητες καταστάσεις λειτουργίας. Στην συγκεκριμένη πλακέτα προτίμησα να συνδέσω ξεχωριστά τις δυο γειώσεις του κυκλώματος (translator circuit – driver circuit) και στο τέλος μέσω δυο διαφορετικών δρόμων (καλωδίων) τα σύνδεσα στον αρνητικό πόλο της μπαταρίας.

Αναφορικά με το κύκλωμα της μονοπολικής οδήγησης, ήταν ευκολότερο στη σχεδίαση. Ο UCN 5804 περιλαμβάνει στο κέλυφος του τόσο κύκλωμα ελέγχου (translator circuits) όσο και κύκλωμα οδήγησης (driver circuit), έτσι δεν υπήρχε απαίτηση για την προσθήκη πολλών διακριτών εξαρτημάτων. Βέβαια η ροπή του κινητήρα ήταν αρκετά μικρότερη από αυτή του διπολικού κάτι που δυσκολεύει την οδήγηση μεγάλων φορτίων. Αυτό βασικά οφείλεται στο ότι επιλέχτηκε οδήγηση μονής φάσης (single phase). Θυμίζουμε ότι σε αυτή τη λειτουργία σε κάθε βήμα ενεργοποιείται διαδοχικά μόνο μια από τις 4 περιελίξεις. Το γεγονός ότι ο τρόπος αυτός οδήγησης δημιουργεί τη μικρότερη ροπή στρέψης μπορεί να το διαπιστώσει κάποιος απλά προσπαθώντας να σταματήσει στιγμιαία το δρομέα του βηματικού κινητήρα κατά την κίνηση του. Στην περίπτωση της μονοπολικής λειτουργίας – μονής φάσης ο δρομέας μπορούσε να σταματήσει πολύ εύκολα. Αντίθετα στην διπολική οδήγηση του βηματικού κινητήρα

(bipolar) δεν μπορούσε να γίνει το ίδιο. Γενικά η υλοποίηση των κυκλωμάτων οδήγησης των βηματικών κινητήρων ήταν από τα πλέον εύκολα μέρη της εφαρμογής.

Μετά τον έλεγχο των βηματικών κινητήρων συνέχεια είχε η μελέτη των βασικών αρχών λειτουργίας του πρωτόκολλου USB 1.1. Η εύρεση δυο συγγραμμάτων, αυτό του ξενόγλωσσου “USB Complete” και του Ελληνικού “Έλεγχος Κυκλωμάτων και μετρήσεων με H/Y” βοήθησαν στο να κατανοήσω κάποιες στοιχειώδεις έννοιες του πρωτόκολλου USB. Επίσης βοήθησαν στο να κατανοήσω τα ακριβή βήματα που θα πρέπει να ακολουθήσω για την υλοποίηση της επικοινωνίας με τον H/Y.

Έπειτα από την κατανόηση κάποιων εννοιών του πρωτοκόλλου USB, έπρεπε να γίνει ένα μεγάλο αλλά ταυτόχρονα και καθοριστικό βήμα για την πορεία της πτυχιακής, και δεν ήταν άλλο από την επιλογή του μικροελεγκτή και το προγραμματισμό αυτού. Σε αυτό το σημείο ξεκίνησα μια μεγάλη και συνεχή αναζήτηση πληροφοριών σχετικά με τον μικροελεγκτή cy7c63001 της CYPRESS. Δυστυχώς τα πράγματα ήταν πολύ πιο δύσκολα από ότι περίμενα μιας και η CYPRESS παρόλα τα εγχειρίδια που διαθέτει, κατά την εκτίμηση μου υστερεί σε ολοκληρωμένες εφαρμογές, στις οποίες βήμα - βήμα γίνεται σταδιακά η υλοποίηση της εφαρμογής. Έτσι ο μέσος μηχανικός ο οποίος ασχολείται για πρώτη φορά με την υλοποίηση ενός USB περιφερειακού πρέπει να καταβάλει μεγάλη προσπάθεια για να μεταβεί ένα επίπεδο πιο ψηλά.. Κάποια από τα σχετικά εγχειρίδια που βρήκα στο διαδίκτυο ήταν, το Cypress CY3640 USB Starter Kit, το Designing a Universal Serial Bus Device (A USB Thermometer), Designing an uninterruptible power supply κ.α.

Το επόμενο βήμα ήταν να αγοράσω το programmer που διαθέτει η εταιρία CY3649 και μέσω των δωρεάν ολοκληρωμένων (free samples) που παρέχει σε μηχανικούς να αρχίσω τους πειραματισμούς. – έτσι και έγινε.

Ο CY7C63001 όπως αναφέρεται στη ενότητα 2.8. ανήκει στην οικογένεια 8-bit RISK One Time Programmable (OTP) (Προγραμματιζόμενο Μια Φορά) μικροελεγκτών. Αυτό σημαίνει πως δεν έχουμε την δυνατότητα να προγραμματίσουμε δεύτερη φορά τον ίδιο μικροελεγκτή και ότι αυτό συνεπάγεται. Το τοπίο άρχισε να καθαρίζει ακόμη περισσότερο μόλις έπειτα από αναζήτηση στο διαδίκτυο βρήκα δυο πολύ ενδιαφέροντα άρθρα με εφαρμογές του συγκεκριμένου μικροελεγκτή στην ιστοσελίδα του ELEKTOR. Εκτιμώ πως για κάποιον αρχάριο στην υλοποίηση τέτοιων εφαρμογών ο καλύτερος τρόπος για να μπει στο κλίμα είναι η μελέτη και υλοποίηση ήδη υπαρχόντων εφαρμογών. Επίσης πολύ θετικό ήταν το γεγονός πως υπήρχε ήδη διαθέσιμος ένας οδηγός της συσκευής (device driver) και το κατάλληλο αρχείο INF, οπότε και δεν χρειάστηκε να φτιαχτεί από την αρχή. Μάλιστα στο Διαδίκτυο μπορεί κάποιος να βρει τον οδηγό της συσκευής (device driver) σε αρχείο .pdf που θα βοηθήσει στην καλύτερη κατανόηση της εφαρμογής. Βέβαια εδώ παρουσιάστηκε το πρόβλημα πως ο driver που είχα βρει δεν είχε την δυνατότητα να προσπελάσει κάθε ακροδέκτη του μικροελεγκτή ξεχωριστά αλλά μόνον όλη την πόρτα. Με κάποιες αλλαγές στο firmware του μικροελεγκτή καθώς και στο software της εφαρμογής (visual basic) το πρόβλημα λύθηκε και πλέον ήταν δυνατή η προσπέλαση κάθε ακίδας του μικροελεγκτή ξεχωριστά.

Όσο αφορά την υλοποίηση του αυτόνομου προγράμματος στη Visual Basic βοήθησαν αρκετά, σημειώσεις που υπάρχουν παντού στο Διαδίκτυο, βιβλία στην βιβλιοθήκη του Τμήματος Ηλεκτρονικής όπως και κάποιοι φίλοι. Γενικά η Visual Basic θεωρείται μια εύχρηστη γλώσσα προγραμματισμού και θα την συνιστούσα ανεπιφύλακτα για την υλοποίηση παρόμοιας εφαρμογής.

Τελειώνοντας, για την σχεδίαση του P.C.B. των κυκλωματικών διατάξεων χρησιμοποίησα το PROTEL 99 SE. Τα τρία τυπωμένα κυκλώματα παρουσιάζονται στην ενότητα 5. Η διαδικασία εμφάνισης αποχάλκωσης, τρυπήματος πλακέτας, κόλλησης εξαρτημάτων κ.τ.λ έγινε στο εργαστήριο του σπιτιού μου.

Όπως γίνεται κατανοητό ο έλεγχος μιας συσκευής μέσω της θύρας USB ενός ηλεκτρονικού υπολογιστή, εκτιμώ ότι είναι ένα πολύ ενδιαφέρον θέμα, το οποίο αν και προϋποθέτει την ανάλωση αρκετού χρόνου για την κατανόηση των βασικών εννοιών, όπως: πρωτόκολλο USB, οδηγοί συσκευών (device drivers), προγραμματισμός USB μικροελεγκτών (firmware), παρόλα αυτά η τελική υλοποίηση και ο έλεγχος σωστής λειτουργίας είναι κάτι που σε αποζημιώνει στα σίγουρα.

Κάποιες σκέψεις για μελλοντική επέκταση της παρούσας εφαρμογής αναφέρονται παρακάτω:

**Δημιουργία CNC.** Με την κατασκευή ενός CNC θα δίνεται η δυνατότητα να γίνεται τρύπημα σε διαφορές ηλεκτρονικές πλακέτες, ζωγραφική, φινίρισμα κτλ κάτι που βεβαίως επιτυγχάνεται από την υψηλή ακρίβεια που σου παρέχουν οι βηματικοί κινητήρες. Το CNC θα ελέγχεται από 3 βηματικούς κινητήρες οι όποιοι θα ελέγχουν την κίνηση και στους 3 άξονες X, Y, Z.

**Αισθητήρια.** Ενδιαφέρον αλλά και πρακτικό σε κάποιες περιπτώσεις θα μπορούσε να είναι μια εφαρμογή με χρήση αισθητήριων όπως: Αισθητήρας υγρασίας, φωτεινότητας, αισθητήρας για την κωδικοποίηση της θέσης (encoder position) κτλ.

**Έξυπνο σπίτι.** Μια χρήσιμη εφαρμογή θα μπορούσε να επιτρέπει τον έλεγχο διαφόρων ηλεκτρικών και ηλεκτρονικών συσκευών, Για παράδειγμα διακοπτική λειτουργία φώτων, άνοιγμα και κλείσιμο γκαραζόπορτας, θερμοσίφωνα κτλ.

**Ασύρματη Επικοινωνία.** Σε κάποιες περιπτώσεις η συσκευή που ελέγχουμε ή κάποιο αισθητήριο θα πρέπει να βρίσκεται σε κάποια απόσταση από τον ηλεκτρονικό υπολογιστή οπότε καθίσταται αναγκαία η ασύρματη επικοινωνία μεταξύ των συσκευών. Η ασύρματη επικοινωνία θα μπορούσε να γίνεται μέσω RF, Υπερύθρων, Bluetooth κτλ.

Αυτές είναι απλά κάποιες πρώτες σκέψεις για την επέκταση της χρήσης της εφαρμογής. Η διάθεση για μελέτη και πειραματισμό, καθώς και η καλή συνεργασία μεταξύ των φίλων και συναδέλφων είναι το κλειδί για την επίτευξη αυτών αλλά και μεγαλύτερων στόχων.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

### Βιβλία

- ✓ Έλεγχος Κυκλωμάτων Και Μετρήσεων Με Η/Υ  
Ι. Καλόμοιρος – Σ. Μπουλτουδάκης – Ι. Πεταλάς
- ✓ USB Complete by Jan Axelson
- ✓ Βασική Ηλεκτρονική Malvino

### Περιοδικά

- ✓ Ελέκτορ - τεύχη του 2001

### Ιστοσελίδες

- ✓ [www.cypress.com](http://www.cypress.com)
- ✓ [www.elektor.com](http://www.elektor.com)
- ✓ [www.armony.com/~rstevew/Public/Motors/H-Bridges/Blanchard/h-bridge.htm](http://www.armony.com/~rstevew/Public/Motors/H-Bridges/Blanchard/h-bridge.htm)
- ✓ [www.controlanything.com/Merchant2/merchant.mvc?Screen=CTGY&Store\\_Code=NCD&category\\_Code=VBASIC](http://www.controlanything.com/Merchant2/merchant.mvc?Screen=CTGY&Store_Code=NCD&category_Code=VBASIC)
- ✓ [www.eio.com/jasstep.htm](http://www.eio.com/jasstep.htm)
- ✓ [www.mcmanis.com/chuck/robotics/tutorial/h-bridge/index.html](http://www.mcmanis.com/chuck/robotics/tutorial/h-bridge/index.html)
- ✓ [www.Lvr.com](http://www.Lvr.com)

### Διάφορα

- ✓ Build this stepper motor controller by David Williams –UCN 5804

# ΠΑΡΑΡΤΗΜΑΤΑ

## A. Συσκευές διασύνδεσης χειριστή (HIDs)

### Χρήσιμες πληροφορίες για τα χειριστήρια

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, τα χειριστήρια είναι μια κλάση συσκευών USB, που τυγχάνουν πλήρους υποστήριξης από το λειτουργικό σύστημα Windows 98 SE και ανώτερα. Αν και γενικά ως χειριστήριο νοείτε η συσκευή που αλληλεπιδρά με ανθρώπινο χειριστή, δεν οφείλει κατά ανάγκη να κατευθύνεται από άνθρωπο. Το μόνο που απαιτείται είναι να εμπίπτει στα όρια που προβλέπουν οι προδιαγραφές για τα χειριστήρια. Κατ' αρχήν, οφείλει να ανταλλάσσει δεδομένα με τον ξενιστή με τη μορφή “αναφορών” (reports). Ο ελεγκτής του ξενιστή πρέπει να γνωρίζει τη μορφή των αναφορών που θα χρησιμοποιεί το χειριστήριο για κάθε ανταλλαγή δεδομένων.

Κάθε αναφορά εισόδου ή εξόδου για ένα χειριστήριο μπορεί να αποτελείτε από πολλές διεξαγωγές (transactions). Κάθε διεξαγωγή για συσκευές χαμηλής ταχύτητας, όπως και ο CY7C63001 μπορεί να περιέχει μέχρι 8 bytes, ενώ για συσκευές υψηλής ταχύτητας μπορεί να περιέχει μέχρι 64 Bytes. Κάθε αναφορά μπορεί να περιέχει μέχρι 255 bytes. Δεν προβλέπεται εξασφαλισμένος ρυθμός μεταφοράς, αφού οι μεταδόσεις που υποστηρίζουν τα χειριστήρια είναι ελέγχου ή διακοπών. Το μέγιστο χρονικό περιθώριο των μεταδόσεων διακοπών εξασφαλίζει ένα μέγιστο ρυθμό μετάδοσης δεδομένων 800 bytes ανά δευτερόλεπτο για συσκευές χαμηλής ταχύτητας και 64000 bytes το δευτερόλεπτο για συσκευές υψηλής ταχύτητας.

Κάθε συσκευή HID έχει ένα τελικό σημείο ελέγχου (μηδενικό σημείο), που υποστηρίζει μεταδόσεις ελέγχου εισόδου/εξόδου (IN-OUT). Κάθε HID που εμπίπτει στις προδιαγραφές 1.1 της USB έχει τουλάχιστον ένα τελικό σημείο που υποστηρίζει μεταδόσεις διακοπών για είσοδο και για έξοδο.

Υπενθύμιση: Ο μικροελεγκτής που χρησιμοποιούμε στην παρούσα πτυχιακή υποστηρίζει δυο τελικά σημεία (endpoints). Το τελικό σημείο 1 υποστηρίζει μόνο μετάδοση δεδομένων από τον ελεγκτή προς τον ξενιστή. Έτσι, κάθε HID που υπακούει στο πρωτόκολλο της USB1.1 διαθέτει ένα σωλήνα (pipe) ελέγχου, έναν τουλάχιστο “σωλήνα” διακοπών για είσοδο και ένα τουλάχιστο “σωλήνα” διακοπών στην έξοδο.

Πως δημιουργούνται οι κατάλληλοι “σωλήνες” για την ανταλλαγή δεδομένων και πως προσδιορίζεται η μορφή (format) των αναφορών εισόδου/εξόδου; Οι πληροφορίες αυτές βρίσκονται μέσα στους περιγραφείς της συσκευής. Οι περιγραφείς είναι δομημένοι σύμφωνα με την ιεραρχία του «δέντρου» βλέπε το παρακάτω σχήμα. Η ρίζα είναι περιγραφέας συσκευής (device descriptor) και ακολουθούν ο περιγραφέας διαμόρφωσης (configuration descriptor), ο περιγραφέας διασύνδεσης (interface descriptor), απ όπου εξαρτώνται και οι περιγραφείς των τελικών σημείων (endpoints descriptors) που υποστηρίζονται από κάθε διασύνδεση.

Ο ορισμός του περιφερειακού ως χειριστήριο HID γίνεται στα πεδία του περιγραφέα κλάσης (class descriptor) που ανήκει στον περιγραφέα διασύνδεσης. Υποκείμενος στον περιγραφέα κλάσης είναι ο περιγραφέας αναφορών. Εκεί ορίζεται η μορφή (format) που παίρνει η ανταλλαγή δεδομένων με τον ξενιστή.

Για κάθε τελικό σημείο επιπλέον του μηδενικού πρέπει να δημιουργηθεί ένας περιγραφέας τελικού σημείου. Το μηδενικό τελικό σημείο υποστηρίζει μόνο μεταδόσεις ελέγχου και οι ιδιότητες του είναι απόλυτα καθορισμένες από τις προδιαγραφές. Ο περιγραφέας του τελικού σημείου ορίζει την κατεύθυνση (IN,OUT) των δεδομένων, τον

τύπο της μετάδοσης (πχ μετάδοση διακοπών), τον μέγιστο αριθμό των bytes σε κάθε πακέτο δεδομένων και τον ρυθμό δειγματοληψίας του τελικού σημείου (polling interval), πχ 10 ms.



Σχήμα Α.1. Ιεραρχία περιγραφών (descriptors) για χειριστήρια.

Μόλις τελειώσει η απαρίθμηση (enumeration) της συσκευής, ο ξενιστής έχει πλέον αναγνωρίσει τη συσκευή σαν χειριστήριο και έχει δημιουργήσει τους κατάλληλους σωλήνες επικοινωνίας με βάση την πληροφορία που έλαβε από τους περιγραφείς των τελικών σημείων. Επίσης, έχει λάβει τους λεγόμενους «περιγραφείς αναφορών», δηλαδή αρχεία που περιέχουν την δομή των αναφορών εισόδου και εξόδου, για μεταδόσεις διακοπών.

## **B. Οδηγοί της Συσκευής (Device Drivers)**

Με βάση τις πληροφορίες που αντλεί ο ξενιστής κατά την διαδικασία της απαρίθμησης από τον περιγραφέα της συσκευής (device descriptor), βρίσκει και φορτώνει στη μνήμη το κατάλληλο αρχείο .INF, που με τη σειρά του περιέχει πληροφορίες για τους οδηγούς της συσκευής. Τα Windows 98 έχουν τα βασικά αρχεία .INF στη διαδρομή C:\Windows\INF. Όταν εγκαθιστά κανείς μια καινούρια συσκευή, τότε η δισκέτα εγκατάστασης περιέχει τα κατάλληλα .INF αρχεία, καθώς και τους σχετικούς οδηγούς drivers της συσκευής. Σε κάθε περίπτωση, ο φάκελος INF περιέχει τα γενικά αρχεία που αντιστοιχούν σε καθιερωμένες ομάδες συσκευών και τα οποία παραπέμπουν σε γενικούς καθιερωμένους οδηγούς (generic drivers). Έτσι, ένα ποντίκι μπορεί να έχει από το κατασκευαστή τους δικούς του οδηγούς για την βέλτιστη εγκατάσταση του, όμως πάντα μπορεί να διαλέξει κάποιον γενικό οδηγό για ποντίκι (generic driver) των Windows.

Κατά την εγκατάσταση ενός χειριστηρίου (συσκευής USB που ανήκει στη κλάση HID) ο χρήστης ειδοποιείται από το λειτουργικό σύστημα να εισάγει δισκέτα με τα πλέον κατάλληλα αρχεία .INF και με τους οδηγούς της συγκεκριμένης συσκευής.

Αν δεν βρει κάτι πιο κατάλληλο, εγκαθιστά απλώς τους γενικούς οδηγούς που έχει για τις συσκευές που ανήκουν στη κλάση των χειριστηρίων. Το γενικό αρχείο .INF για τα χειριστήρια, το οποίο θα καλέσει ο ξενιστής όταν δεν βρίσκει πιο κατάλληλο είναι το *hiddev.inf*. Ο χρήστης μπορεί να το βρει στο φάκελο C:\Windows\INF. Μπορούμε μάλιστα να το τροποποιήσουμε κατάλληλα, ώστε να το φέρουμε στις ανάγκες της εφαρμογής μας. Το αρχείο αυτό περιέχει την ταυτότητα πολλών κατασκευαστών, για πολλές συσκευές ήχου ή ελέγχου οθόνης, που μπορούν να εγκατασταθούν χωρίς δικούς τους οδηγούς. Στο τμήμα του αρχείου .INF με τίτλο [SampleHID.CopyList] μπορεί να βρει κανείς τον κατάλογο των οδηγών, που χρησιμοποιούνται ως χειριστήρια:

hidusb.sys hidclass.sys hidparse.sys

Αυτοί είναι οι γενικοί οδηγοί (generic drivers) για τα χειριστήρια HIDs.



## Πίνακας Περιεχομένων

### Περίληψη -Abstract

1. Εισαγωγή .....	1
2. Η θύρα USB.....	5
2.1. Ο δρόμος προς τη USB.....	5
2.2. Βασικά χαρακτηριστικά .....	6
2.3. Κλάσεις συσκευών USB.....	7
2.4. Μετάδοση δεδομένων μέσω του διαύλου USB.....	8
2.5. Τα μέρη μιας μετάδοσης USB .....	11
2.6. Απαρίθμηση (Enumeration) .....	14
2.7. Περιγραφείς (Descriptors) .....	15
2.8. Ο μικροελεγκτής CY7C63001.....	16
3. Βηματικοί κινητήρες - τρόποι οδήγησης.....	27
3.1. Αρχές λειτουργίας βηματικών κινητήρων .....	27
3.2. Κατηγορίες βηματικών κινητήρων .....	29
3.4. Τρόποι λειτουργίας των βηματικών κινητήρων .....	33
3.5. Διπολική οδήγηση βηματικού κινητήρα.....	35
3.6. Μονοπολικό βηματικό κινητήρες.....	35
3.7. Διπολικό βηματικό κινητήρες .....	36
3.8. Πλεονεκτήματα και μειονεκτήματα των βηματικών κινητήρων.....	37
4. Visual Basic 6 .....	38
4.1. Περιβάλλον της Visual Basic .....	38
4.2. Στοιχεία ελέγχου, χρήση, ιδιότητες, συμβάντα, μέθοδοι .....	41
4.3. Ιδιότητες .....	45
4.4. Κώδικας .....	46
4.5. Μεταβλητές (τύποι, χρήση, δήλωση, σύνταξη).....	47
4.6. Συναρτήσεις API .....	48
5. Η Κατασκευή του υλικού μέρους.....	50
5.1. Ελεγκτής USB .....	50
5.2. Κύκλωμα οδήγησης του βηματικού κινητήρα 4 ακροδεκτών.....	52
5.3. Κύκλωμα οδήγησης του βηματικού κινητήρα 5 ακροδεκτών.....	55
6. Οδηγίες χρήσης.....	58
7. Συμπεράσματα.....	62
 BIBΛΙΟΓΡΑΦΙΑ .....	 65
 ΠΑΡΑΡΤΗΜΑΤΑ .....	 66
Α. Συσκευές διασύνδεσης χειριστή (HIDs).....	66
Β. Οδηγοί της Συσκευής (Device Drivers) .....	68