



ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΣΧΟΛΗ Σ.Τ.Ε.Φ

ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**«ΣΥΣΤΗΜΑ ΠΑΡΑΛΛΗΛΗΣ ΨΗΦΙΑΚΗΣ ΕΙΣΟΔΟΥ ΚΑΙ
ΕΞΟΔΟΥ ΜΕ ΧΡΗΣΗ ΤΗΣ ΘΥΡΑΣ USB»**



Ονησιφόρου Φίλιππος

Επιβλέπων καθηγητής: Κιζήρογλου Μιχάλης

*Αφιερωμένη στους γονείς μου και στην οικογένεια μου, που με στήριξαν
καθόλη την διάρκεια των σπουδών μου.*

*Επίσης θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Κιζήρογλου
Μιχάλη , που μου εμπιστεύτηκε την παρούσα διπλωματική εργασία και
μου προσέφερε βοήθεια όποτε τη χρειαζόμουν.*

ΕΙΣΑΓΩΓΗ

Για τον έλεγχο ηλεκτρονικών συσκευών και κυκλωμάτων μέσω θύρας έχω επιλέξει τον driver **UM245R USB TO PARALLEL module** της FTDI chips ο οποίος είναι ένας εναλλακτικός τρόπος με τον οποίο μπορώ να ελέγγω διάφορες ηλεκτρονικές συσκευές με παράλληλη και ταυτόχρονη επικοινωνία η και να δέχομαι δεδομένα μέσω του Driver και να καταγράφονται τα αποτελέσματα στον υπολογιστή. Γνώριζα αρκετά λίγα για τον τρόπο λειτουργίας των θύρων USB και επίσης πίστευα ότι με το συγκεκριμένο module θα ήταν πολύ απλό στον προγραμματισμό οπως και με την σειριακή θύρα έχοντας μόνο 8 data pins να ελέγξω .

Δυστυχώς επειδή τα αρχεία που η Εταιρεία FDTI χορηγά για την συγκεκριμένη συσκευή δεν έχουν αρκετές λεπτομέρειες και πληροφορίες , επίσης στο Ίντερνετ δεν μπορούσα να βρώ κάποιες εξηγήσεις και πληροφορίες για το πως δουλεύει και για το πως μπορεί να δουλέψει με τον καλύτερο δυνατο τρόπο .Επρέπε πρώτα να κατανοήσω τον τρόπο λειτουργίας των θυρών USB.

Οι πάρακατω πληροφορίες που παρέχω είναι αποτέλεσμα απο πολλές ώρες διαβάσματος , και πειραμάτικων διαδικασιών και ελπίζω να γίνει ένα βοήθημα για τους συμφοιτητές μου οι οποίοι στο μέλλον θα θέλουν να χρησιμοποιήσουν τον ίδιο driver είτε για τις πτυχιακές τους εργασίες είτε για δικά τους πρότζεκτ . Δεν είμαι έμπειρος στον τομέα αυτό και σιγούρα θα έχω καποιά λάθη που μπορεί καποιός εύκολα στο μέλλον να κάνει τις βελτιώσεις για τον συγκεκριμένο Driver .

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....

ΕΙΣΑΓΩΓΗ.....

ΚΕΦΑΛΑΙΟ 1: Θύρες USB

1.	Εννοιολογική προσέγγιση της θύρας USB.....
2.	Ιστορική αναδρομή..... ..
3.	Χαρακτηριστικά της θύρας USB.....
4.	Ταχύτητες της θύρας USB.....
5.	Λειτουργία USB.....
1.	Host Controllers.....
2.	Hot swappable.....
6.	Πλεονεκτήματα και Μειονεκτήματα των USB.....
1.	Πλεονεκτήματα USB.....
2.	Μειονεκτήματα USB.....

ΚΕΦΑΛΑΙΟ 2: ΕΙΣΑΓΩΓΗ ΣΤΟΝ UM245R

7.	Περιγραφή του UM245R
8.	Χαρακτηριστικά UM245R/FT245R.....
1.	Block διάγραμμα του FT245R.....

2. Περιγραφή εισόδων και εξόδων του UM245R.....
3. Περιγραφή θυρών και Γέφυρων όπως δίνεται απο το φύλλο δεδομένων της εταιρείας.....
.....
4. FT245RL χαρακτηριστικά της συσκευής και Όρια προδιαγραφών και αντοχής.....
 1. Όρια προδιαγραφών και αντοχής του FT245RL.....
 2. DC Χαρακτηριστικά.....
.....
 3. Χαρακτηριστικά της μνήμης EEPROM.....
 4. Χαρακτηριστικά του εσωτερικού Ρολογιού του UM245R.....
 5. Διαστάσεις του UM245R.....
.....
5. Τυπικές Εφαρμογές του UM245R.....

ΚΕΦΑΛΑΙΟ 3: ΠΡΟΓΡΑΜΜΑΤΙΖΟΝΤΑΣ ΤΟΝ UM245R

9. Καταστάσεις Λειτουργίας του UM245R.....
10. Επιλογή των θυρών ως εισόδοι/εξόδοι.....
11. Επιλογή γλώσσας προγραμματισμού.....

- 12. Τρόποι
προγραμματισμού.....
- 13. Υποστήριξη και περιγραφή των
DRIVERS.....
 - 1. Υποστήριξη των
Drivers.....
 - 2. Περιγραφή των Drivers και
διαφορές.....
- 14. Προσθήκη Drivers στην Visual Basic
2010.....

ΚΕΦΑΛΑΙΟ 4: ΕΛΕΓΧΟΣ ΤΩΝ ΘΥΡΩΝ ΤΟΥ UM245R ΩΣ ΕΞΟΔΟΙ

- 15. Περιγραφή του
προγράμματος.....
- 16. Κύκλωμα ελέγχου θυρών του UM245r ως εξόδοι με χρήση
LED.....
 - 1. Περιγραφή του
Κυκλώματος.....
- 17. Interface ελέγχου των θυρών του UM245R με
Led.....
 - 1. Πηγαίος Κώδικας σε Visual Basic 2010 (Source Code) Παράρτημα 2^ο
και περιγραφή του
προγράμματος.....

ΚΕΦΑΛΑΙΟ 5: ΕΛΕΓΧΟΣ ΤΩΝ ΘΥΡΩΝ ΤΟΥ UM245R ΩΣ ΕΙΣΟΔΟΙ

- 18. Περιγραφή του
προγράμματος.....

- 19. Κύκλωμα ελέγχου των θυρών του UM245R ως είσοδοι με διακοπτάκια.....
 - 1. Περιγραφή του Κυκλώματος.....
- 20. Interface ελέγχου των θυρών του UM245R ως είσοδοι του με διακοπτάκια.....
 - 1. Πηγαίος Κώδικας σε Visual Basic 2010 (Source Code) Παράρτημα 3^ο και περιγραφή του προγράμματος.....

ΠΑΡΑΡΤΗΜΑ I: Αρχείο D2XX_MODULE.VB.....

ΠΑΡΑΡΤΗΜΑ II: Πηγαίος Κώδικας ελέγχου θυρών του UM245R ως έξοδοι.....

ΠΑΡΑΡΤΗΜΑ III Πηγαίος Κώδικας ελέγχου θυρών του UM245R ως είσοδοι.....

Κεφάλαιο 1: Θύρες USB

1.1 Εννοιολογική προσέγγιση της θύρας USB

Οι θύρες USB (Universal Serial Bus) ή αλλιώς Ενιαίος Σειριακός Διάυλος, αποτελούν ένα σύστημα διαύλου, το οποίο χρησιμοποιείται για την επικοινωνία ενός υπολογιστή με περιφερειακά συστήματα. Η σύνδεση του USB, απαιτεί λιγότερο χώρο και μπορεί επίσης να παρέχει ενέργεια σε απλές συσκευές όπως το ποντίκι, το πληκτρολόγιο ή η ιστοκάμερα.

Έχουν ως σκοπό να επιτρέπουν στις περιφερειακές μονάδες να συνδέονται με τον υπολογιστή χρησιμοποιώντας μια ενιαία τυποποιημένη υποδοχή διεπαφών βελτιώνοντας έτσι τις έτοιμες προς χρήση ικανότητες των συσκευών για σύνδεση ή αποσύνδεσή τους με το σύστημα χωρίς να χρειάζεται επανεκκίνηση.

1.2 Ιστορική αναδρομή

Η προδιαγραφή USB 1.0 παρουσιάστηκε τον Νοέμβριο του 1995. Ο USB προωθήθηκε από την Intel (UHCI and open software stack), τη Microsoft (Windows software stack), την Philips (Hub, uSB-ήχος) και την US Robotics. Αρχικά ο USB προοριζόταν να αντικαταστήσει το πλήθος θυρών στο πίσω μέρος των Η/Υ, καθώς επίσης και να απλοποιήσει τη διαμόρφωση του λογισμικού των συσκευών επικοινωνίας. Η USB ήταν, επίσης, η κύρια θύρα στον Apple iMac που παρουσιάστηκε στις 6 Μαΐου του 1998, συμπεριλαμβανομένης τη θύρας για το νέο πληκτρολόγιο και το ποντίκι. Το πρότυπο USB 1.1 παρουσιάστηκε τον Σεπτέμβριο του 1998 για να αποκαταστήσει τα προβλήματα που εμφανίστηκαν με τις προηγούμενη έκδοσή του. Από το 2008 η προδιαγραφή USB βρίσκεται στην έκδοση 2.0 (με τις αναθεωρήσεις).

Η Hewlett Packard, η Intel, η Lucent (τώρα Alcatel-Lucent), η Microsoft, η NEC, και η Philips συνεργάστηκαν από κοινού στην πρωτοβουλία να αναπτυχθεί ένα πρότυπο με υψηλότερο ποσοστό μεταφοράς δεδομένων από την προδιαγραφή 1.1. Η προδιαγραφή USB 2.0 παρουσιάστηκε τον Απρίλιο του 2000 και τυποποιήθηκε από την USB-IF στα τέλη του 2001. Ο εξοπλισμός προσαρμόζεται σε οποιαδήποτε έκδοση των προτύπων (1.0, 1.1, 2.0) και λειτουργεί, επίσης, με τις συσκευές που σχεδιάστηκαν σε οποιαδήποτε προηγούμενη προδιαγραφή (1.0, 1.1, 2.0) (γνωστή και ως οπίσθια συμβατότητα).

Τα μικρότερα βύσματα και υποδοχές USB για τη χρήση στις φορητές και κινητές συσκευές, αποκαλούμενες mini-B, προστέθηκαν στην προδιαγραφή USB με τη πρώτη ειδοποίηση για την αλλαγή του σχήματος των αρχικών βυσμάτων και υποδοχών. Μια νέα παραλλαγή των μικρότερων βυσμάτων USB και των υποδοχών, αποκαλούμενη micro-USB, αναγγέλθηκε από το φόρουμ των εταιρειών που υλοποιούν το USB στις 4 Ιανουαρίου του 2007.

1.3 Χαρακτηριστικά της θύρας USB

1. Παροχή ηλεκτρικού ρεύματος στις συσκευές χαμηλής κατανάλωσης χωρίς την ανάγκη εξωτερικής παροχής ηλεκτρικού ρεύματος.
2. Παρέχει την δυνατότητα χρησιμοποίησης πολλών συσκευών USB χωρίς την απαίτηση ρυθμίσεων ή μεμονωμένων προγραμμάτων οδήγησης (drivers) από τους κατασκευαστές για να εγκατασταθούν.
3. Η θύρα USB προορίζεται να βοηθήσει να αποσυνδεθούν όλες οι διαφορετικού τύπου θύρες, σειριακές ή παράλληλες.
4. Συνδέει τις περιφερειακές μονάδες υπολογιστών όπως το ποντίκι, το πληκτρολόγιο, τα PDA, τα χειριστήρια παιχνιδιών, οι σαρωτές, τις ψηφιακές φωτογραφικές μηχανές, τους εκτυπωτές, τις συσκευές MP3 και τους φορητούς δίσκους ή μνήμες. Για πολλές από τις προαναφερόμενες συσκευές η μέθοδος σύνδεσης USB έχει γίνει η τυποποιημένη μέθοδος σύνδεσης.
5. Συνδέει τους μη-δικτυακούς εκτυπωτές με θύρα USB και απλοποιεί τη σύνδεση πολλών εκτυπωτών με έναν υπολογιστή.
6. Παρέχει ασφάλεια των προσωπικών δεδομένων. Μέσω λογισμικού μπορούν να κλειδωθούν οι συσκευές USB (ακόμα και οι φορητές μνήμες flash), ώστε να μην λειτουργούν με τον Η/Υ και να επιτρέπεται σε άλλες περιφερειακές μονάδες USB να λειτουργούν, όπως οι εκτυπωτές.

1.4 Ταχύτητες της θύρας USB

Ο USB λειτουργεί με τις ακόλουθες ταχύτητες:

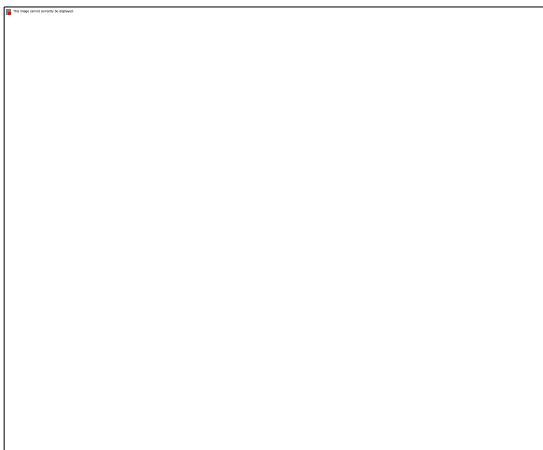
1. **Χαμηλή ταχύτητα - Low Speed (1.5 Mbit/s):** Χρησιμοποιείται περισσότερο σε συσκευές όπως πληκτρολόγιο, ποντίκι και χειριστήρια παιχνιδιών (HID, Human Interface Devices). Ο ρυθμός αυτός συναντάται στις εκδόσεις 1.1 και 2.0.
2. **Πλήρης ταχύτητα - Full Speed (12 Mbit/s):** Οι συσκευές Full Speed διαιρούν το εύρος του USB μεταξύ τους σύμφωνα με την παραδοχή πρώτο-έρχεται πρώτο-εξυπηρετείται (first-come first-served).
3. **Υψηλή ταχύτητα - High Speed (480 Mbit/s)**

4. **Υπερυψηλή ταχύτητα - Super Speed ή SS (5 Gbit/s):** Η πιο πρόσφατη εκδοχή του USB που λειτουργεί με την υπερυψηλή ταχύτητα ονομάζεται USB 3.0.
- 5.

1.5 Λειτουργία USB

Ένα USB σύστημα έχει μια ασύμμετρη διάταξη που αποτελείται από τα εξής τμήματα:

- Τον οικοδεσπότη (host), που είναι το υπολογιστικό σύστημα που παρέχει τις συνδέσεις USB. Είναι ένας γενικός όρος που περιλαμβάνει τη CPU (Central Processing Unit), το λειτουργικό σύστημα, τους οδηγούς USB κτλ.
- Τον κατανεμητή (hub), που είναι μια διάταξη επανάληψης που αναμεταδίδει τις συναλλαγές USB μεταξύ της θύρας αντίθετα στη φορά του ρεύματος και των θυρών στη φορά του ρεύματος που συνδέονται σε τοπολογία αστέρα σε σειρά (4 σε 1 κατανεμητής).
- Τη συσκευή, η οποία είναι το περιφερειακό στην απόληξη που εκτελεί εφαρμογές και συγκεκριμένες λειτουργίες. Ο όρος «αντίθετα στη φορά κατεύθυνσης του ρεύματος» αναφέρεται στη φορά μετακίνησης των δεδομένων από μια συσκευή στον οικοδεσπότη, ενώ ο όρος «στη φορά του ρεύματος» αναφέρεται στην αντίθετη κατεύθυνση. Πρόσθετοι κατανεμητές (hubs) USB μπορούν να περιληφθούν σε σειρές, επιτρέποντας τη διακλάδωση σε μια δομή δένδρων, υπό τον όρο ότι δεν ξεπερνά το όριο των πέντε επιπέδων σειρών. Ο USB οικοδεσπότης μπορεί να έχει μια πληθώρα ελεγκτές οικοδεσποτών και κάθε ένας από αυτούς μπορεί να παρέχει μία ή περισσότερες θύρες USB.



Περισσότερες από 127 συσκευές, συμπεραμβανομένων των συσκευών hub, μπορούν να συνδεθούν σε ένα ενιαίο ελεγκτή οικοδεσποτών.

Οι συσκευές USB συνδέονται μαζικά μέσω των θυρών. Υπάρχει πάντα ένας κατανεμητής γνωστός ως «κόμβος-ρίζα» (root hub), ο οποίος είναι ενσωματωμένος σε ένα ελεγκτή οικοδεσποτών. Επιπλέον υπάρχουν οι επονομαζόμενοι

«διαμοιραστές hubs» (sharing hubs). Αυτοί επιτρέπουν σε πολλούς υπολογιστές να έχουν πρόσβαση στην ίδια (ή ίδιες) περιφερειακές συσκευές, με την ανταλλαγή πρόσβασης μεταξύ των υπολογιστών αυτόματα ή χειροκίνητα.

Μία συσκευή USB μπορεί να αποτελείται από αρκετές λογικές υποσυσκευές οι οποίες αναφέρονται ως «συσκευές λειτουργιών» (device functions), επειδή κάθε ξεχωριστή συσκευή μπορεί να παρέχει αρκετές λειτουργίες, όπως για παράδειγμα μία webcam (συσκευή λειτουργίας βίντεο) με ενσωματωμένο μικρόφωνο (συσκευή λειτουργία ήχου). Η συσκευή επικοινωνίας USB βασίζεται σε «σωλήνες» (λογικά κανάλια, pipes). Οι σωλήνες συνδέουν τον ελεγκτή οικοδεσποτών με μια λογική οντότητα στη συσκευή, που αναφέρεται ως «σημείο κατάληξης» (endpoint). Ο όρος αυτός χρησιμοποιείται ορισμένες φορές για να αναφερθεί σε ένα σωλήνα.

Μια συσκευή USB μπορεί να έχει πάνω από 32 ενεργούς σωλήνες, 16 προς τον ελεγκτή οικοδεσποτών και 16 από τον ελεγκτή. Κάθε σημείο κατάληξης μπορεί να μεταφέρει δεδομένα μόνο προς μια κατεύθυνση, είτε προς είτε από τη συσκευή, έτσι ώστε κάθε δίαυλος να είναι μονοκατευθυντικός. Τα endpoints ομαδοποιούνται σε διεπαφές και κάθε διεπαφή σχετίζεται με μία μοναδική συσκευή. Μια εξαίρεση σε αυτό είναι το endpoint μηδέν, το οποίο χρησιμοποιείται για το σχηματισμό συσκευών και το οποίο δε σχετίζεται με καμία διεπαφή.

1.5.1 Host Controllers

Το υλικό μέρος του υπολογιστή, το οποίο περιλαμβάνει τον ελεγκτή οικοδεσποτών και το γενικό κατανεμητή, παρέχει στον προγραμματιστή μια διεπαφή, η οποία ονομάζεται «Host Controller Device» (HCD) και καθορίζεται από τον εφαρμοστή του υλικού. Στην υλοποίηση 1.x υπήρχαν δύο ανταγωνιστικές υλοποιήσεις HCD, η

Open Host Controller Interface (OHCI) και η Universal Host Controller Interface (UHCI). Η πρώτη αναπτύχθηκε από την Compaq, τη Microsoft και τη National Semiconductor και η δεύτερη από την Intel. Η UHCI είναι περισσότερο καθοδηγούμενη από λογισμικό, κάνοντας την UHCI να επιβαρύνει ελαφρώς πιο πολύ τον επεξεργαστή από ότι η OHCI, αλλά η υλοποίησή της είναι φθηνότερη. Οι αντικρουόμενες υλοποιήσεις ανάγκασαν τα λειτουργικά συστήματα καθώς και το αντίστοιχο hardware να αναπτύξει και να δοκιμάσει και τις δύο υλοποιήσεις, γεγονός που αύξησε το κόστος.

Κατά τη διάρκεια της φάσης σχεδίασης του USB 2.0, το USB-IF επέμενε σε μία μόνο υλοποίηση. Η USB 2.0 HCD υλοποίηση ονομάστηκε Enhanced Host Controller Interface (EHCI). Μόνο το EHCI μπορεί να υποστηρίξει υψηλής ταχύτητας μεταφορές. Οι περισσότερα ελεγκτές EHCI που βασίζονται σε PCI (Peripheral Component Interconnect) διαύλους περιέχουν άλλες υλοποιήσεις HCD που ονομάζονται «companion host controller» για να υποστηρίξουν συσκευές τόσο υψηλής όσο και

χαμηλής ταχύτητας. Οι εικονικοί HCD στην Intel και οι ελεγκτές EHCI της VIA Technologies είναι UHCI, ενώ όλοι οι υπόλοιποι είναι εικονικοί OHCI ελεγκτές.

1.5.2 HOT Swappable

Όπως προαναφέρθηκε, το USB επιτρέπει στο υλικό να συνδεθεί χωρίς απαίτηση των καρτών επέκτασης για να συνδεθεί με το δίαυλο του PC EISA, ISA, ή PCI. Αυτό καθιστά δυνατές τις έτοιμες προς χρήση ικανότητες με την άδεια των συσκευών για να είναι συνεχώς σε επικοινωνία με τη σύνδεση ή την αποσύνδεση χωρίς να κλείσει ο υπολογιστής ή να είναι απαραίτητη η εκ νέου επανεκκίνηση του συστήματος. Όταν μια νέα συσκευή είναι συνδεδεμένη με τον υπολογιστή, ο οικοδεσπότης ανιχνεύει και αναγνωρίζει τη συσκευή, φορτώνοντας αυτόματα τον οδηγό συσκευών που χρειάζεται. Αυτό το χαρακτηριστικό γνώρισμα προσδίδει στο USB ένα σημαντικότερο χαρακτηριστικό, την ευελιξία.

1.6 Πλεονεκτήματα και Μειονεκτήματα των USB

1.6.1 Πλεονεκτήματα USB

Η πληθώρα πλεονεκτημάτων του USB το έχει καταστήσει τη δημοφιλέστερη διεπαφή στο είδος του. Έτσι λοιπόν το USB:

- Είναι εύκολο στη χρήση, εξοικονομεί χρόνο καθώς δεν απαιτεί λεπτομέρειες εγκατάστασης και διαμόρφωσης.
- Είναι γρήγορο, ώστε η διεπαφή να μην υφίσταται τη λεγόμενη «στένωση φιάλης» των αργών επικοινωνιών.
- Είναι αξιόπιστο, ώστε τα σφάλματα να είναι σπάνια, καθώς λαμβάνει χώρα αυτόματη διόρθωση λαθών.
- Είναι ευέλικτο, ώστε πολλά είδη περιφερειακών να μπορούν να χρησιμοποιούν την ίδια διεπαφή.
- Έχει μικρή κατανάλωση ενέργειας, καθώς η παροχή ισχύος σε χαμηλής κατανάλωσης συσκευές γίνεται δίχως τη χρήση εξωτερικής πηγής ισχύος, γεγονός που συμφέρει σε φορητές συσκευές.

- Είναι φθινό, ώστε οι χρήστες (και οι κατασκευαστές που θα κατασκευάσουν τη διεπαφή στα προϊόντα τους) δεν αποθαρρύνονται από την τιμή.
- Υποστηρίζεται από το λειτουργικό σύστημα, επομένως οι προγραμματιστές δε χρειάζεται να προγραμματίζουν τους οδηγούς σε χαμηλού επιπέδου γλώσσα για τα περιφερειακά που χρησιμοποιούν τη διεπαφή.

1.6.2 Μειονεκτήματα USB

Όπως κάθε διεπαφή, έτσι και το USB παρουσιάζει και ορισμένα μειονεκτήματα, τα οποία παρουσιάζονται στη συνέχεια:

- Απουσία υποστήριξης σε παλαιότερα hardware. Οι παλιότεροι υπολογιστές και τα περιφερειακά τους δε διαθέτουν θύρες USB. Για να συνδεθεί ένα περιφερειακό που δεν υποστηρίζει USB σύνδεση με μία USB θύρα, η λύση είναι να χρησιμοποιηθεί ένας μετατροπέας που πραγματοποιεί τη μετατροπή μεταξύ του USB και των παλιότερων διεπαφών. Αρκετές πηγές διαθέτουν μετατροπείς για χρήση με περιφερειακά με RS-232, RS-485 και άλλα. Παρόλα αυτά η χρήση του μετατροπέα είναι χρήσιμη μόνο για περιφερειακά που χρησιμοποιούν συμβατικά πρωτόκολλα υποστηριζόμενα από τον οδηγό συσκευής του μετατροπέα. Σε διαφορετική περίπτωση, η χρήση ενός USB περιφερειακού σε ένα PC που δεν υποστηρίζει USB απαιτεί την πρόσθεση USB δυνατοτήτων στον υπολογιστή με δύο πράγματα: το υλικό του ελεγκτή οικοδεσποτών και ένα λειτουργικό σύστημα που υποστηρίζει USB.
- Περιορισμοί στην ταχύτητα: Το USB είναι ευέλικτο, ωστόσο δεν έχει σχεδιαστεί για όλες τις εφαρμογές. Η μεγάλη του ταχύτητα το καθιστά ανταγωνιστικό με τις IEEE-1394 διεπαφές (FireWire) 400 Mbit/s, αλλά η διεπαφή IEEE-1394b εξακολουθεί να είναι ακόμα πιο γρήγορη στα 3.2 GB/s.
- Περιορισμοί στην απόσταση: Το USB σχεδιάστηκε σαν ένας δίαυλος με την ελπίδα ότι τα περιφερειακά θα βρίσκονταν σχετικά κοντά το ένα στο άλλο. Ένα τμήμα καλωδίου μπορεί να είναι τόσο μακρύ όσο 5 μέτρα. Άλλες διεπαφές, όπως το RS-232, το RS-485 και το Ethernet επιτρέπουν μεγαλύτερα μήκη καλωδίων. Το μήκος ενός USB συνδέσμου μπορεί να αυξηθεί στα 30 μέτρα με τη χρήση καλωδίων που συνδέουν 5 καταναμητές (hubs) και μια συσκευή, χρησιμοποιώντας 6 τμήματα καλωδίων από 5 μέτρα το καθένα.
- Ομότιμες επικοινωνίες: Το γεγονός ότι το USB αποτελεί ένα δίαυλο της επιφάνειας εργασίας σημαίνει επίσης ότι κάθε σύστημα USB διαθέτει ένα οικοδεσπότη-

υπολογιστή που διαχειρίζεται το δίαυλο επικοινωνίας. Τα περιφερειακά δεν μπορούν να επικοινωνούν απευθείας το ένα με το άλλο. Όλες οι επικοινωνίες γίνονται προς ή από τον οικοδεσπότη- υπολογιστή. Άλλες διεπαφές, όπως το IEEE-1394 επιτρέπουν απευθείας επικοινωνία μεταξύ των περιφερειακών. Το USB παρέχει μια μερική λύση σε αυτό με το USB On-The-Go, που όπως προαναφέρθηκε εισάχθηκε το 2001 σαν συμπλήρωμα της προδιαγραφής 2.0. Το USB On-The-Go καθορίζει έναν οικοδεσπότη- υπολογιστή με περιορισμένες δυνατότητες, κατάλληλο για χρήση σε συσκευές που συνδέονται σε ένα μόνο USB περιφερειακό.

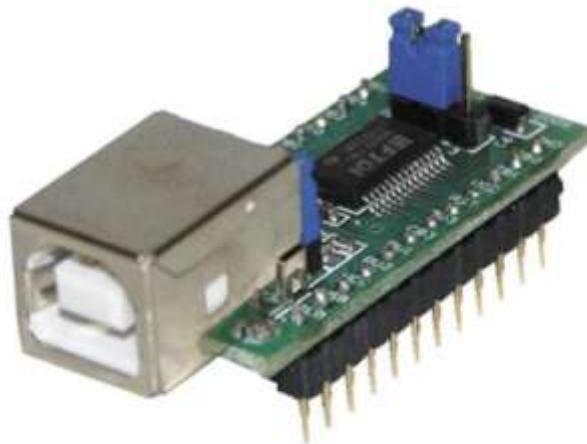
Κεφάλαιο 2: Εισαγωγή στον UM245R

2.1 Περιγραφή του UM245R

Ο UM245r είναι ένας Parallel-FIFO driver της Εταιρίας FTDIChip (Future Technology Devices International Ltd) ο οποίος χρησιμοποιεί το chip FT245R.

Όταν λέμε UM245R εννοούμε ολοκληρη την συσκευή με ενσωματωμένο το chip FT245R

Τον συγκεκριμένο driver μπορεί κάποιος να τον αγοράσει απο την ίδια την Εταιρία η και απο ιστότοπους αγοραπωλησιών όπως το Ebay, Amazon κλπ.



Εικόνα 2: «Συσκευή του driver UM245R»

http://apple.clickandbuild.com/cnb/shop/ftdichip?productID=40&search=FT245r&op=catalogue-product_info-null

Μπορεί φυσικά κάποιος να αγοράσει μόνο το Chip και να φτιάξει απο μόνος του το συγκεκριμένο module .

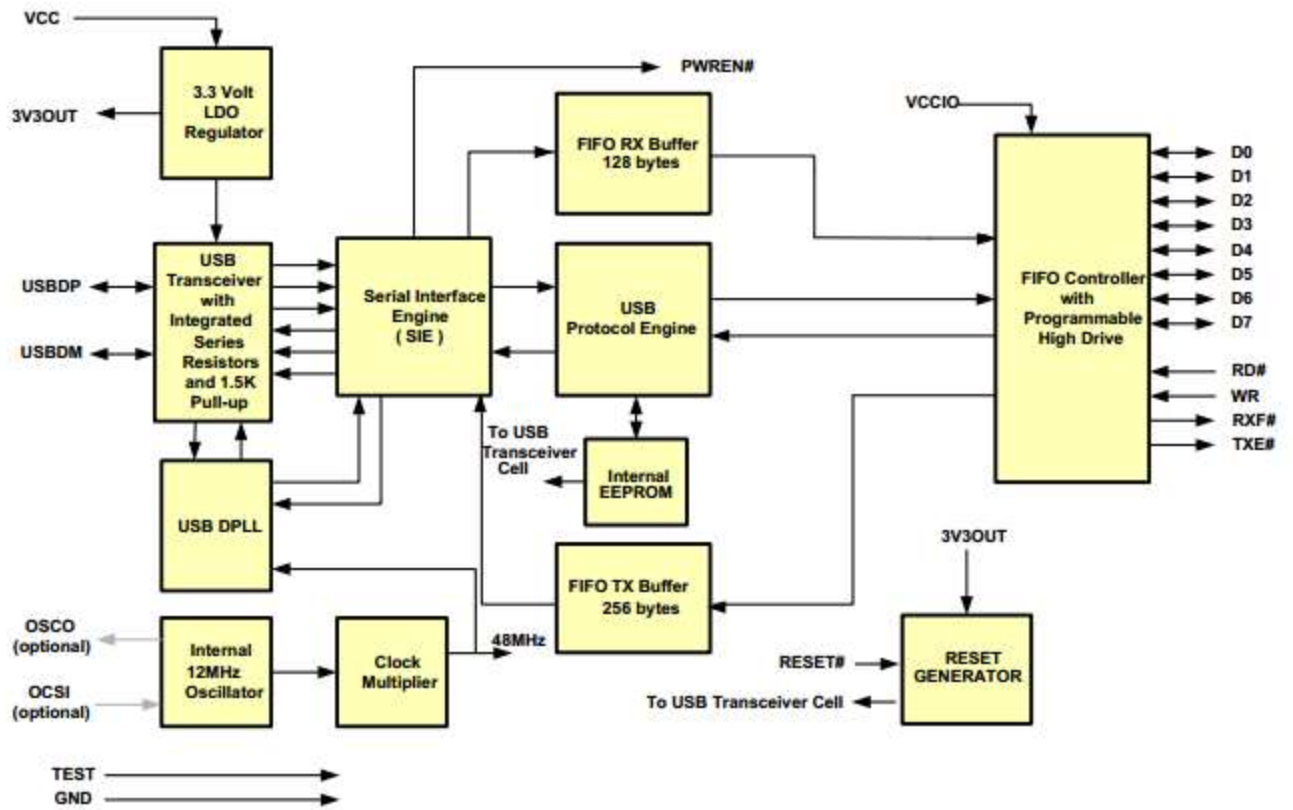


Εικόνα 3: «Chip FT245R»

http://apple.clickandbuild.com/cnb/shop/ftdichip?productID=198&search=FT245r&op=catalogue-product_info-null

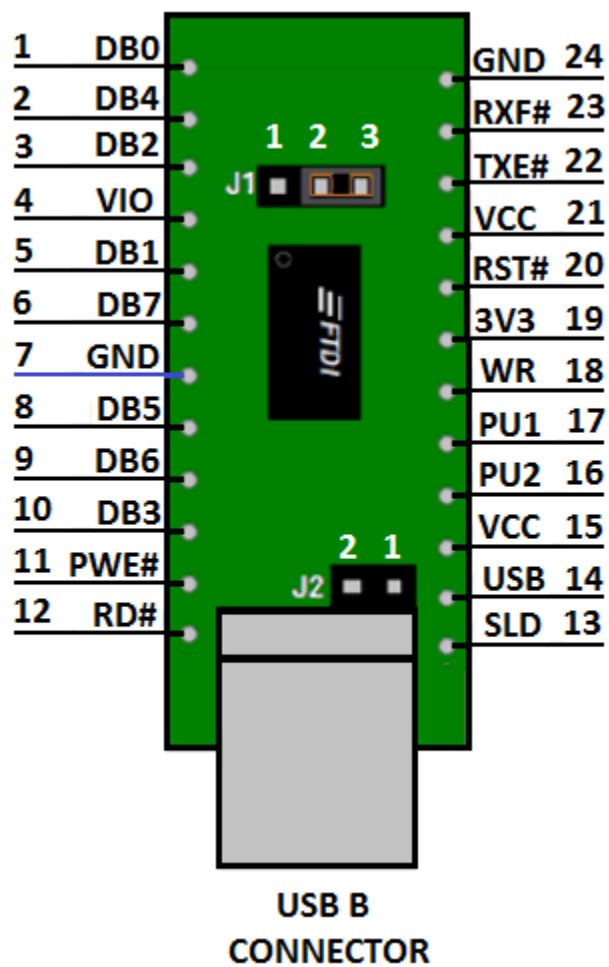
2.2 Χαρακτηριστικά του UM245R/FT245R

2.2.1 Block διάγραμμα του FT245R



Εικόνα 4: «Block διάγραμμα του FT245R»

2.2.2 Περιγραφή Εισόδων και Εξόδων του UM245R



Εικόνα 5: «Εισόδοι και εξόδοι του UM245R»

Η συσκευή αποτελείται από 24 θύρες και 5 γέφυρες

Αριθμός Pin	Ονομασία θυρας	Αριθμός Pin	Ονομασία θυρας	Αριθμός Pin	Ονομασία θυρας
1	DB0	11	PWE#	21	VCC
2	DB4	12	RD#	22	TXE#
3	DB2	13	SLD	23	RXF#
4	VIO	14	USB	24	GND
5	DB1	15	VCC		Γέφυρες
6	DB7	16	PU2	J1.1	3V3
7	GND	17	PU1	J1.2	VIO
8	DB5	18	WR	J1.3	VCC
9	DB6	19	3V3	J2.1	USB
10	DB3	20	RST#	J2.2	VCC

2.2.3 Περιγραφή θυρών και Γεφύρων όπως δίνεται απο το φύλλο δεδομένων της εταιρείας

Αριθμος Pin	Ονομασία	Τύπος	Περιγραφή
1	DB0	I/O	FIFO Data Bus Bit 0*
2	DB4	I/O	FIFO Data Bus Bit 4*
3	DB2	I/O	FIFO Data Bus Bit 2*
4	VIO	PWR	+1.8V to +5.25V supply to the FIFO Interface and Control group pins (1...3, 5, 6, 9...14, 22, 23).

In USB bus powered designs connect to 3V3 OUT to drive out at 3.3V levels (connect jumper J1 pins 1 and 2 together), or connect to VCC to drive at 5V CMOS level (connect jumper J1 pins 2 and 3 together). This pin can

also be supplied with an external 1.8V – 5.0V supply in order to drive at different levels. It should be noted that in this case this supply should originate from the same source as the supply to Vcc. This means that in bus powered designs a regulator which is supplied by the 5V on the USB bus should be used.

5	DB1	I/O	FIFO Data Bus Bit 1*
6	DB7	I/O	FIFO Data Bus Bit 7*
7, 24	GND	PWR	Module ground supply pins
8	DB5	I/O	FIFO Data Bus Bit 5*
9	DB6	I/O	FIFO Data Bus Bit 6*
10	DB3	I/O	FIFO Data Bus Bit 3*
11	PWE#	I/O	Goes low after the device is configured by USB, then high during USB suspend. Can be used to control power to external logic P-Channel logic level MOSFET switch. Enable the interface pull-down option when using the PWREN# pin in this way.
12	RD#	I/O	Enables the current FIFO data byte on D0...D7 when low. Fetches the next FIFO data byte (if available) from the receive FIFO buffer when RD# goes from high to low. See Section 4.4 for timing diagram.*
13	SLD	GND	USB Cable shield.
14	USB	Output	5V Power output USB port. For a low power USB bus powered design, up to 100mA can be sourced from the 5V supply on the USB bus. A maximum of 500mA can be sourced from the USB bus in a high power USB bus powered design.
15, 21	VCC	PWR or Output	These two pins are internally connected on the module PCB. To power the module from the 5V supply on USB bus, connect jumper J2 pins 1 and 2 together (this is the module default configuration). In this case these pins would have the same description as pin 14.

To use the UM245R module in a self powered configuration, ensure that jumper J2 pins 1 and 2 are

not connected together, and apply an external 3.3V to 5.25V supply to one or both of these pins.

17	PU1	Control	Pull up resistor pin connection 2. Connect to pin 17 (RST#) in a self powered configuration.
16	PU2	Control	Pull up resistor pin connection 1. Connect to pin 14 (USB) in a self powered configuration.
19	3V3	Output	3.3V output from integrated LDO regulator. This pin is decoupled to ground on the module PCB with a 100nF capacitor. The prime purpose of this pin is to provide the internal 3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the FT245RL's VCCIO pin by connecting this pin to pin 4 (VIO), or by connecting together pins 1 and 2 on jumper J1.

Pin No.	Όνομασία	Τύπος	Περιγραφή
20	RST#	Input	Can be used by an external device to reset the FT245R. If not required can be left unconnected, or pulled up to VCCIO.
18	WR	I/O	Writes the data byte on the D0...D7 pins into the transmit FIFO buffer when WR goes from high to low. See Section 4.4 for timing diagram.*
1	DB0	I/O	FIFO Data Bus Bit 0*
2	DB4	I/O	FIFO Data Bus Bit 4*
22	TXE#	I/O	When high, do not write data into the FIFO. When low, data can be written into the FIFO by strobing WR high, then low. During reset this signal pin is tri-state, but pulled up to VCCIO via an internal 200kΩ resistor. See Section 4.4 for timing diagram.
23	RXF#	I/O	When high, do not read data from the FIFO. When low, there is data available in the FIFO which can be read by strobing RD# low, then high again. During reset this signal pin is tri-state, but pulled up to VCCIO via an internal 200kΩ resistor. See Section 4.4 for timing

diagram.

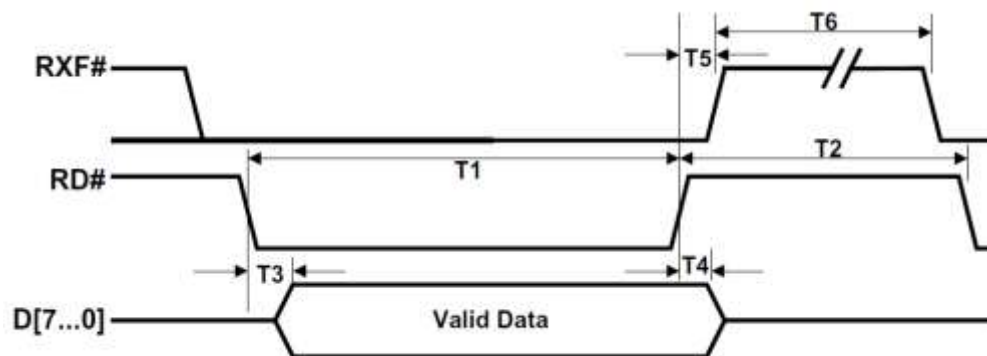
If the Remote Wakeup option is enabled in the internal EEPROM, during USB suspend mode (PWREN# = 1) RXF# becomes an input which can be used to wake up the USB host from suspend mode. Strobing the pin low will cause the device to request a resume on the USB bus.

Pin No.	Ονομασία	Τύπος	Περιγραφή
1	3V3	Output	3.3V output from integrated LDO regulator. This pin is decoupled to ground on the module PCB with a 100nF capacitor. The prime purpose of this pin is to provide the internal 3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the FT245RL's VCCIO pin by connecting this pin to pin 4 (VIO), or by connecting together pins 1 and 2 on jumper J1.
2	VIO	PWR	+1.8V to +5.25V supply to the FIFO Interface and control pins (1...3, 5, 6, 9...14, 22, 23). In USB bus powered designs connect to 3V3 to drive out at 3.3V levels (connect jumper J1 pins 1 and 2 together), or connect to VCC to drive out at 5V CMOS level (connect jumper J1 pins 2 and 3 together). This pin can also be supplied with an external 1.8V - 2.8V supply in order to drive out at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to Vcc. This means that in bus powered designs a regulator which is supplied by the 5V on the USB bus should be used.
3	VCC	PWR	VCC Output. This will be 5V from the USB bus if pins 1 and 2 on jumper J2 are connected. Alternatively, if the module is in a self powered configuration, the supply to the VCC module pins (15 and 21) will be brought out to this jumper pin.

Connect this jumper J1 pin 2 in order to supply the device IO pins

from the supply to VCCIO

Pin No.	Ονομασία	Τύπος	Περιγραφή
1	USB	PWR	5V Power output USB port. For a low power USB bus powered design, up to 100mA can be sourced from the 5V supply on the USB bus. A maximum of 500mA can be sourced from the USB bus in a high power USB bus powered design with the use of an external power switch (See Section 7.3).
2	VCC	PWR or Output	Board supply input. Connect to jumper J2 pin 1 in order to supply the board from the USB bus. This pin is internally connected to the VCC DIP pins. Remove the jumper connector in a self powered design.



Εικόνα 6: « Διαγράμμα Διεπαφής ελέγχου χρονικών κύκλων »

Χρόνος	Περιγραφή	Min	Max	Unit
T7	WR active pulse Width	50		Ns
T8	WR to RD Pre-Charge Time	50		Ns
T9	Data Setup Time before WR Inactive	20		Ns
T10	Data Hold Time from RD Inactive	0		Ns
T11	WR Inactive to TXE#	0	25	Ns
T12	TXE# Inactive After WR Cycle	80		Ns

Πίνακας 1 FIFO Διαγράμματα Διεπαφής ελέγχου χρονικών κύκλων

2.2.4 FT245RL: Χαρακτηριστικά της συσκευής - Όρια προδιαγραφών και αντοχής

2.2.4.1 Όρια προδιαγραφών και αντοχής του FT245RL

Αυτά είναι τα όρια προδιαγραφών του FT245R. Αυτά είναι τα όρια προδιαγραφών σύμφωνα με τα μέγιστα όρια προδιαγραφών του συστήματος (IEC 60134). Εάν ξεπεραστούν τα όρια αυτά, πιθανόν να προκληθεί ζημιά στην συσκευή.

Παραμέτροι	Τιμή	Unit
Storage Temperature	-65 to +150	°C
Floor Life (Out of Bag) At Factory Ambient	168 Hours (IPC/JEDEC J-STD-033A MSL Level 3)	Hours

(30°C / 60% Relative Humidity)	Compliant)*	
Ambient Temperature (Power Applied)	-40 to +85	°C
VCC Supply Voltage	-0.5 to +6.00	V
D.C. Input Voltage - USBDP and USBDM	-0.5 to +3.8	V
D.C. Input Voltage - High Impedance Bidirectional signals	-0.5 to +(Vcc +0.5)	V
D.C. Input Voltage - All other Inputs	-0.5 to +(Vcc +0.5)	V
D.C. Output Current - Outputs	24	mA
DC Output Current - Low Impedance Bidirectionals	24	mA
Power Dissipation (Vcc = 5.25V)	500	mW

Πίνακας 1: Όρια αντοχής και προδιαγραφών

2.2.4.2 DC Χαρακτηριστικά

Παραμέτροι	Περιγραφή	Ελάχιστη	Τυπική	Μεγιστή	Unit	Καταστάσεις
VCC1	VCC Operating Supply Voltage	4.0	---	5.25	V	The UM245R only uses the FT245R internal oscillator circuit
VCC2	VCCIO Operating Supply Voltage	1.8	---	5.25	V	
Icc1	Operating Supply Current	---	15	---	mA	Normal Operation
Icc2	Operating Supply Current	50	70	100	μA	USB Suspend

Πίνακας 2: Τάσεις λειτουργίας και Ρεύματα.

Παραμέτροι	Περιγραφή	Ελάχιστη	Τυπική	Μεγιστή	Unit	Καταστάσεις
------------	-----------	----------	--------	---------	------	-------------

Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 2mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 2mA
Vin	Input Switching Threshold	1.3	1.6	1.9	V	**
VHys	Input Switching Hysteresis	50	55	60	mV	**

Πίνακας 3: FIFO Interface and Control Bus Pin Characteristics (VCCIO = 5.0V, Standard Drive Level).

Παραμέτροι	Περιγραφή	Μικρότερη	Τυπική	Μεγιστή	Unit	Καταστάσεις
Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 6mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 6mA
Vin	Input Switching Threshold	1.3	1.6	1.9	V	**
VHys	Input Switching Hysteresis	50	55	60	mV	**

Πίνακας 4 FIFO Interface and Control Bus Pin Characteristics (VCCIO = 5.0V, High Drive Level)

Παραμέτροι	Περιγραφή	Ελάχιστη	Τυπική	Μεγιστή	Unit	Καταστάσεις
Voh	Output Voltage High	2.2	2.8	3.2	V	I source = 3mA

V _{ol}	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
V _{in}	Input Switching Threshold	1.0	1.2	1.5	V	**
V _{Hys}	Input Switching Hysteresis	20	25	30	mV	**

Πίνακας 5: FIFO Interface and Control Bus Pin Characteristics (VCCIO = 3.3V, High Drive Level).

Παραμέτροι	Περιγραφή	Ελάχιστη	Τυπική	Μεγιστή	Unit	Καταστάσεις
V _{oh}	Output Voltage High	2.1	2.8	3.2	V	I source = 3mA
V _{ol}	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
V _{in}	Input Switching Threshold	1.0	1.2	1.5	V	**
V _{Hys}	Input Switching Hysteresis	20	25	30	mV	**

Πίνακας 6: FIFO Interface and Control Bus Pin Characteristics (VCCIO = 2.8V, High Drive Level).

Παραμέτροι	Περιγραφή	Μικρότερη	Τυπική	Μεγιστή	Unit	Καταστάσεις
V _{in}	Input Switching Threshold	1.3	1.6	1.9	V	
V _{Hys}	Input Switching	50	55	60	mV	

Hysteresis

Πίνακας 7: RESET# and TEST Pin Characteristics

Παραμέτροι	Περιγραφή	Μικρότερη	Τυπική	Μεγιστή	Unit	Καταστάσεις
UVoh	I/O Pins Static Output (High)	2.8	---	3.6	V	RI = 1.5kΩ to 3V3Out (D+) RI = 15kΩ to GND (D-)
UVol	I/O Pins Static Output (Low)	0	---	0.3	V	RI = 1.5kΩ to 3V3Out (D+) RI = 15kΩ to GND (D-)
UVse	Single Ended Rx Threshold	0.8	---	2.0	V	
UCom	Differential Common Mode	0.8	---	2.5	V	
UVDif	Differential Input Sensitivity	0.2	---	---	V	
UDrvZ	Driver Output Impedance	26	29	44	Ohms	***

Πίνακας 8: USB I/O Pin (USB DP, USB DM) Characteristics

2.2.4.3 Χαρακτηριστικά της μνήμης EEPROM

Η εσωτερική μνήμη EEPROM των 1024 bit έχει τα εξής χαρακτηριστικά:

Παραμέτροι	Τιμή		Μον.μετρησής
	Ελάχιστη	Μέγιστη	
Data Retention	15		Years
Read / Write Cycle	100,000		Cycles

Πίνακας 9: Χαρακτηριστικά της μνήμης EEPROM

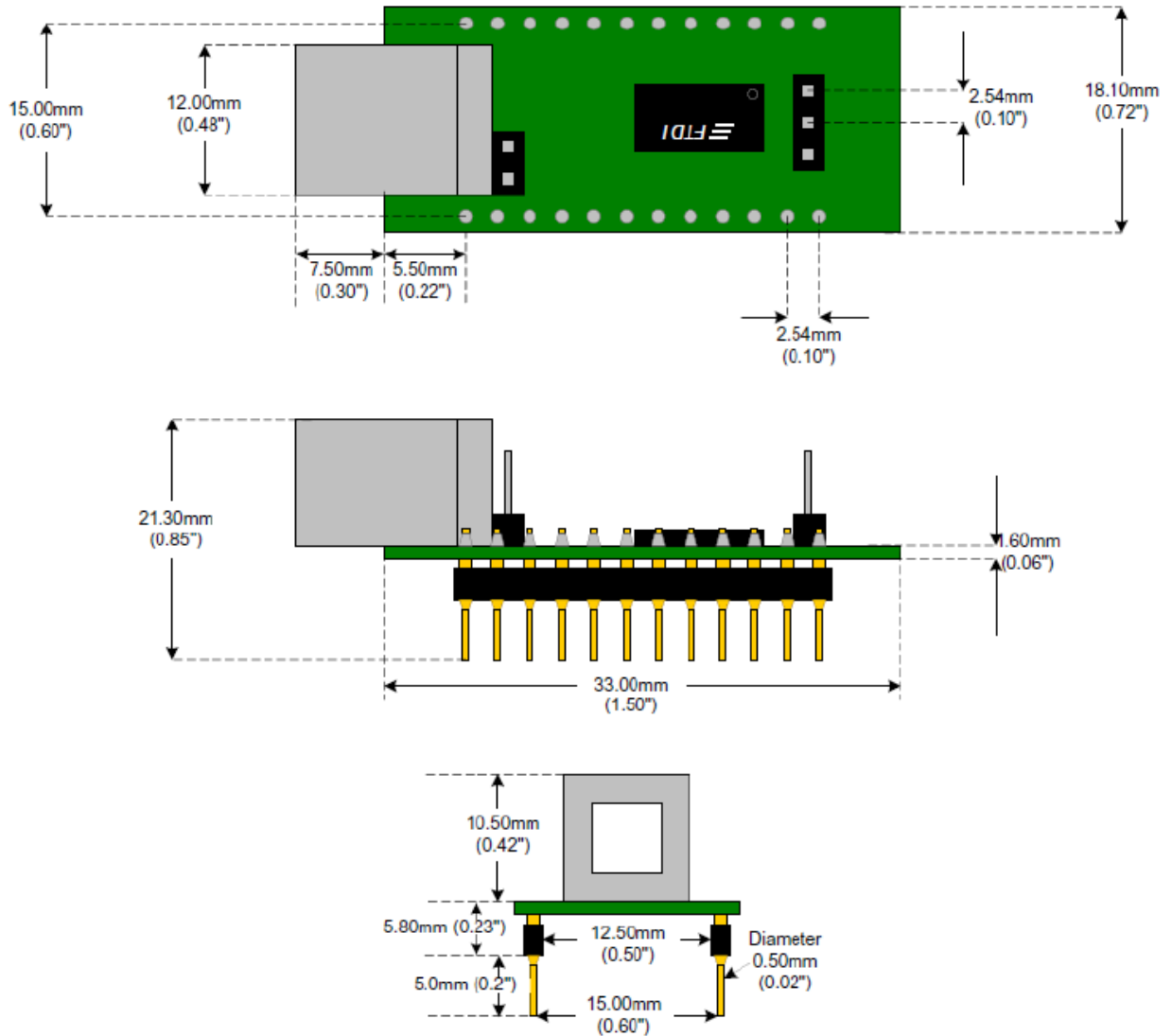
2.2.4.4 Χαρακτηριστικά του εσωτερικού Ρολογιού του UM245R

Η χρήση του εσωτερικού ρολογιού απαιτεί VCC να είναι σε όρια απο 4.0V μέχρι 5.25V.
Το εσωτερικό Ρολόι έχει τα εξής χαρακτηριστικά :

Παραμέτροι	Τιμή		Μον.μετρησής
	Ελάχιστη	Μέγιστη	
Frequency of Operation	11.98	12.00	Frequency of Operation
Clock Period	83.19	83.33	Clock Period

Πίνακας 10: Χαρακτηριστικά εσωτερικού Ρολογιού

2.2.4.5 Διαστάσεις του UM245R :



UM245R Module Dimensions

Όλες οι διαστάσεις του driver είναι σε μιλίμετρα και οι ίντςες στην παρένθεση. Το chip FT245RL που χρησιμοποιείται απο τον UM245R είναι εφαρμόσιμο σε μια μικρή πλακέτα απο 28 ποδαράκια.

2.2.6 Τυπικές Εφαρμογές του UM245R:

1. Αναβάθμιση Περιφερειακών Legacy σε USB
2. Ασύρματα τηλέφωνα USB καλώδια μεταφοράς δεδομένων και διεπαφές
3. Διασύνδεση MCU / PLD / FPGA με βάση τα σχέδια σε USB
4. USB Audio και βίντεο χαμηλής μεταφοράς δεδομένων Bandwidth
5. PDA με τη μεταφορά δεδομένων USB
6. Αναγνώστες USB με έξυπνη κάρτα
7. USB συσκευές
8. USB Βιομηχανικών Ελέγχων
9. USB MP3 Player Interface
10. USB flash card reader / Συγγραφέων
11. Set Top Box PC - USB interface
12. Ψηφιακή φωτογραφική μηχανή USB Interface

13. USB μόντεμ
14. USB ασυρματού μόντεμ
15. USB αναγνώστες γραμμωτού κώδικα
16. USB Software / Hardware Dongles κρυπτογράφης

Κεφάλαιο 3: Προγραμματίζοντας UM245R

3.1 Καταστάσεις Λειτουργίας του UM245R

Ο **UM245R** έχει 3 καταστάσεις λειτουργίας

1. Normal
2. Asynchronous bit-bang
3. Synchronous bit-bang

Το αρχείο που χορηγεί η Εταιρεία FTDI chips δίνει περισσότερες πληροφορίες σχετικά για asynchronous bit-bang mode στην συγκεκριμένη λειτουργία το Chip θα διαβάζει δεδομένα από τα pins του κάθε φορά που θα γράφεται ένα Byte στην είσοδο ή θα ενεργοποιεί ένα από τα pins του κάθε φορά που θα γράφεται ένα Byte στην έξοδο.

3.2 Επιλογή των θυρών ως εισόδοι/εξόδοι

Μπορούμε να επιλέξουμε ποια pins θα είναι για εισόδους και ποια pins θα είναι για εξόδους. Αυτό που πρέπει όμως πρώτα να προσέξουμε είναι ότι εάν ο Υπολογιστής διαβάζει μερικά δεδομένα από το chip, τα οποία Bytes περιλαμβάνουν τις τιμές των pins και παράλληλα είχαν οριστικοποιηθεί ως εισόδοι τότε δεν θα μπορούμε να χρησιμοποιήσουμε σωστά τις εξόδους που ορίσαμε. Πρέπει να χωρίσουμε δηλαδή τα pins σε Low Bytes και σε High Bytes (από 0-127bytes είναι τα pins D0 - D3 που είναι τα LowBytes και από 128-255 είναι τα pins D4-D7 όπου αυτά αποτελούν τα High Bytes). Έτσι ορίζουμε ως εισόδους τα pins D0-D3 και ως εξόδους τα pins D4-D7.

Επίσης μέσα στον κώδικα του προγράμματος πρέπει να δημιουργηθεί μια μνήμη έτσι ώστε να κρατά τα δεδομένα για τις εισόδους. Αν η τιμή των bytes των εισόδων είναι μεγαλύτερη από την τιμή των bytes των εξόδων δε θα μπορούμε να χρησιμοποιήσουμε σωστά την τιμή της εξόδου για αυτό καλύτερα να αποθηκεύονται σε μια μνήμη οι τιμές της εισόδου.

3.3 Επιλογή γλώσσας προγραμματισμού

Έχω επιλέξει σαν γλώσσα προγραμματισμού την VISUAL BASIC 2010. Ο λόγος που επέλεξα αυτή τη γλώσσα είναι επειδή μου ήταν πιο ευκολό να προγραμματίσω το συγκεκριμένο driver λόγω του ότι είναι σχεδόν η ίδια με την C# που κάναμε στον Προγραμματισμό III. Ένας άλλος λόγος είναι ότι ήθελα να έχω ένα interface έτσι ώστε κάποιος άλλος ο οποίος δεν γνωρίζει από προγραμματισμό να μπορεί εύκολα να το χρησιμοποιήσει για μια εφαρμογή που ο ίδιος θέλει.

3.4 Τρόποι προγραμματισμού

Ο προγραμματισμός για το συγκεκριμένο module μπορεί να γίνει με 2 τρόπους:

1. με VCP (Virtual Com Port)
2. με Direct Drivers(D2XX)

3.5 Υποστήριξη και περιγραφή των DRIVERS

3.5.1 Υποστήριξη των Drivers

Οι VIRTUAL COM PORT (VCP) Drivers μπορούν να υποστηριχθούν από:

1. Windows 7 32,64-bit
2. Windows Vista
3. Windows XP 64-bit
4. Windows XP Embedded
5. Windows 98, 98SE, ME, 2000, Server 2003, XP and Server 2008
6. Windows CE.NET 4.2 , 5.0 and 6.0
7. MAC OS 8 / 9, OS-X
8. Linux 2.4 and greater

Οι D2XX Direct Drivers μπορούν να υποστηριχθούν από:

9. Windows 7 32,64-bit
10. Windows Vista
11. Windows XP 64-bit
12. Windows XP Embedded
13. Windows 98, 98SE, ME, 2000, Server 2003, XP and Server 2008
14. Windows CE.NET 4.2, 5.0 and 6.0
15. Linux 2.4 and greater

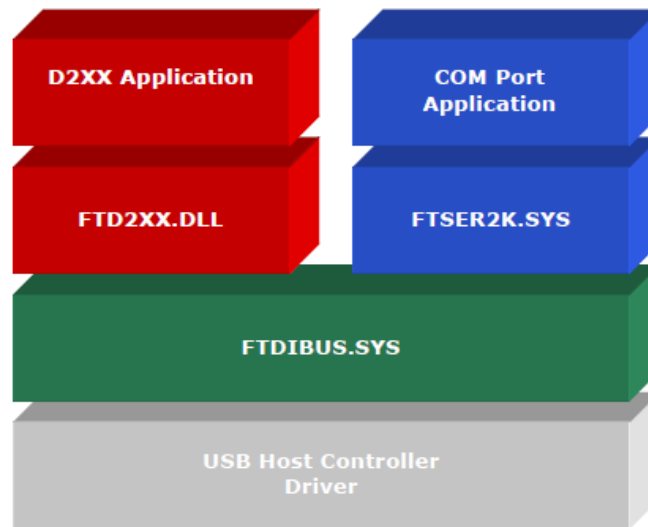
3.5.2 Περιγραφή των Drivers και οι διαφορές τους

Η FTDI παρέχει δύο εναλλακτικές διεπαφές (interface) λογισμικού για το φάσμα των USB-UART και USB-FIFO ICs. Η πρώτη διεπαφή (interface) παρέχει μια εικονική θύρα COM (VCP), η οποία εμφανίζεται στο σύστημα ως θύρα COM κληρονομιά.

Η δεύτερη διεπαφή (interface), D2XX, παρέχεται μέσω ιδιόκτητου DLL (FTD2XX.DLL). Η διεπαφή D2XX παρέχει ειδικές λειτουργίες που δεν είναι διαθέσιμες στο πρότυπο του λειτουργικού συστήματος COM PORT APIs, όπως τη ρύθμιση της συσκευής με ένα διαφορετικό τρόπο ή εγγραφή δεδομένων στη συσκευή EEPROM.

Στην περίπτωση των FTDI Drivers για τα Windows, οι Drivers D2XX και VCP διανέμονται στην ίδια συσκευασία οδηγού η οποία ονομάζεται μοντέλο συνδυασμένων Driver Combined Driver Model (CDM) πακέτο.

Στο πάρακατω σχήμα βλέπουμε την Αρχιτεκτονική των Windows Driver CDM η οποία απεικονίζει την αρχιτεκτονική του προγράμματος οδήγησης των Windows CDM.



Στους Drivers VCP η μεταφορά δεδομένων είναι περιορισμένη με μόλις 300kbyte ανα δευτερόλεπτο.

Επίσης οι Direct(D2XX) είναι σαφέστατα πιο γρήγοροι drivers και μας επιτρέπουν τον έλεγχο με πολύ μεγαλύτερες ταχυτητές. Μας δίνει την δυνατότητα να επιλέξουμε εμείς το Baud Rate , όπου μέγιστο Baud rate το οποίο δίνετε απο το Programmer's Guide είναι 115200 όπου μας επιτρέπει μεταφορά δεδομένων μέχρι και 1Mbyte ανα δευτερόλεπτο .

Ο Driver D2XX μπορεί να καλύψει ένα πολύ μεγάλο εύρος στον τομέα του αυτόματου ελέγχου με πολύ μικρό κόστος το οποίο σήμερα λαμβάνεται πάρα πολύ υπόψιν.

Επίσης οι drivers του συγκεκριμένου module γίνονται αυτόματα εγκατάσταση όταν συνδεθεί με το καλώδιο USB στον Υπολογιστή. Το καλώδιο USB που χρειάζεται είναι τύπου A/B αλλά μπορούν να γίνουν και εγκατάσταση κατεβάζοντας απο την ιστοσελίδα της ίδιας της εταιρείας.



Απο τα παραπάνω μπορούμε να διακρίνουμε ότι οι Drivers D2XX είναι η καλύτερη επιλογή για τον προγραμματισμό του UM245R επειδή μας δίνει μεγαλύτερη ευελιξία.

3.6 Προσθήκη Drivers στην Visual Basic 2010

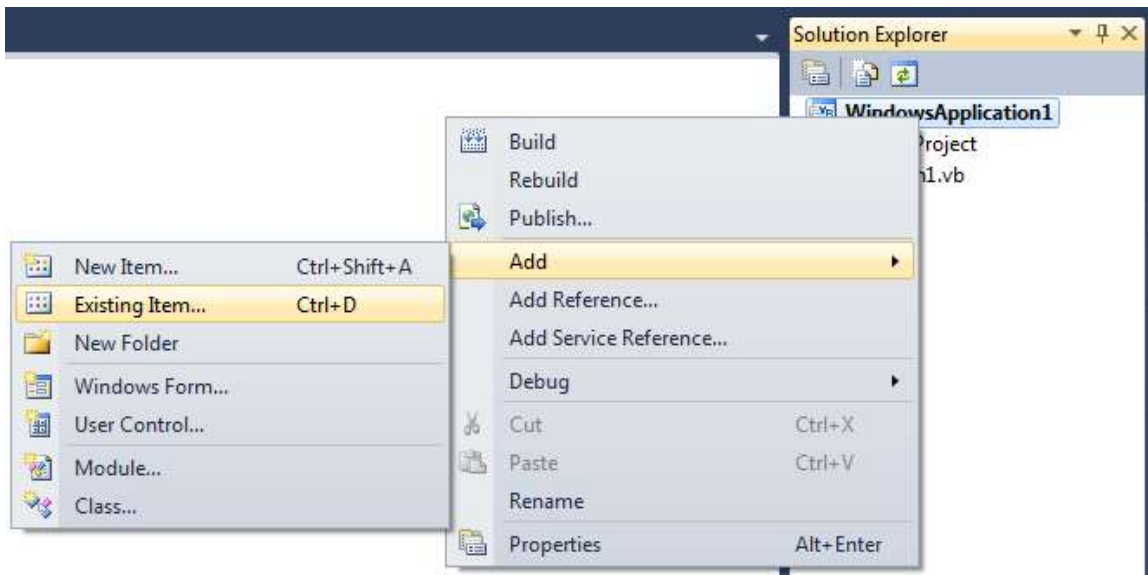
Για να μπορούμε να χρησιμοποιήσουμε τις διάφορες εντολές που χρειάζεται για τον προγραμματισμό του συγκεκριμένου driver πρέπει να κάνουμε προσθήκη τους drivers τους οποίους τους παρέχει η FDTichips και είναι έτοιμοι .Για να τους κάνουμε προσθήκη πρέπει να ακολουθήσουμε την παρακάτω διαδικασία.

Βήμα 1°

Πάμε στην ιστοσελίδα της FDTI chips στο πιο κάτω σύνδεσμο και κατεβάζουμε το αρχείο DXX_module.bas το οποίο το μετονομάζουμε σε DXX_module.**vb** για να μπορεί να γίνει η χρήση του απο την Visual Basic 2010.

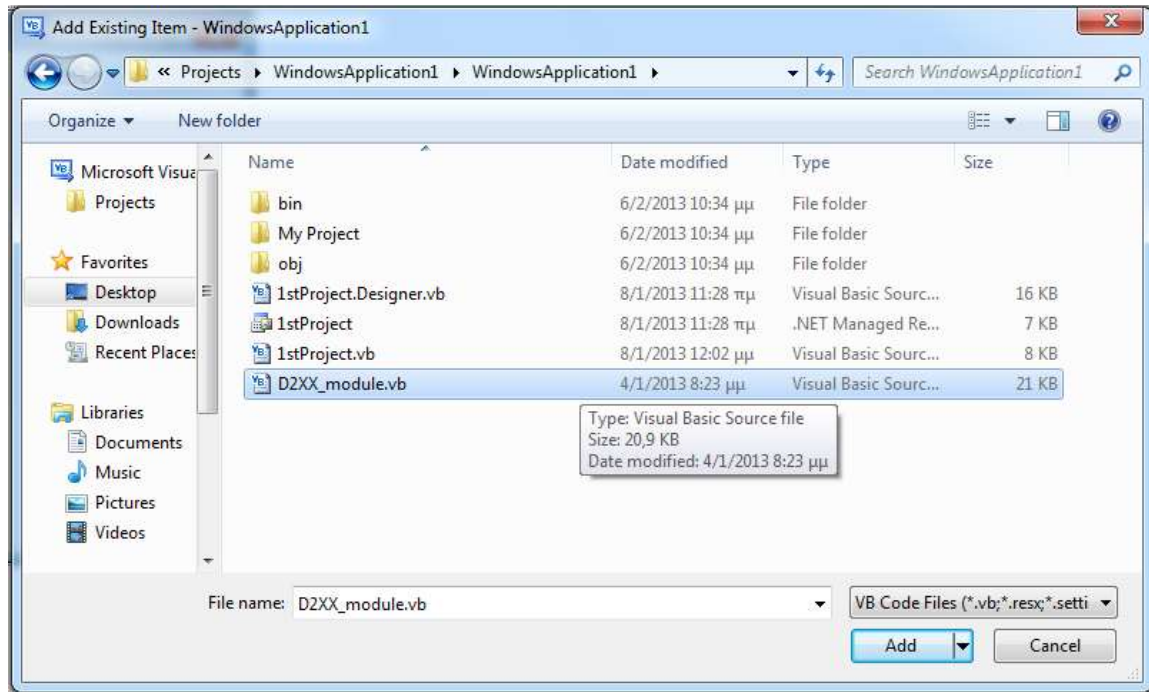
Βήμα 2°

Πάμε στον Solution Explorer και πατάμε δεξί κλικ και επιλεγουμε Add->Existing Item και μας εμφανίζεται το παρακάτω παράθυρο.

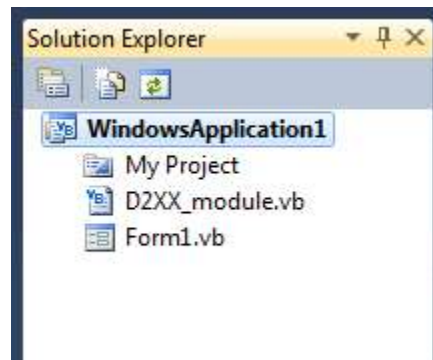


Βήμα 3°

Στην συνέχεια βρίσκουμε που έχουμε αποθηκεύσει το αρχείο **DXX_module.vb**, το επιλέγουμε και στην συνέχεια πατάμε Add.



Αν τα παραπάνω βήματα γίνουν σωστά πρέπει τώρα μέσα στον Solution Explorer να έχουμε και το αρχείο **D2XX_module.vb** όπως φαίνεται στην παρακάτω εικόνα .



Εάν δεν γίνει η παραπάνω διαδικασία τότε οι εντολές οι οποίες βρίσκονται στο αρχείο **D2XX_Programmer's_Guide(FT_000071)** δεν θα μπορούν να χρησιμοποιηθούν.

Κεφάλαιο 4: Έλεγχος των θυρών του UM245R ως εξόδοι

4.1 Περιγραφή του προγράμματος

Σε αυτό το κεφάλαιο θα δούμε πως μπορούμε να ελέξουμε τις θύρες του UM245R μέσω της Visual Basic με την χρήση Led. Με το παράδειγμα αυτό μπορούμε να κατανοήσουμε πως λειτουργεί ο συγκεκριμένος driver και οι θύρες του απλά και εύκολα. Ο κώδικας είναι απλός και εύκολος και μπορεί να τροποποιηθεί σύμφωνα με τις ανάγκες τις οποίες θέλουμε να λειτουργεί και να συμπεριφέρεται.

Με αυτό το παράδειγμα θα παρατηρήσουμε ότι ο συγκεκριμένος driver μπορεί να εφαρμοστεί για :

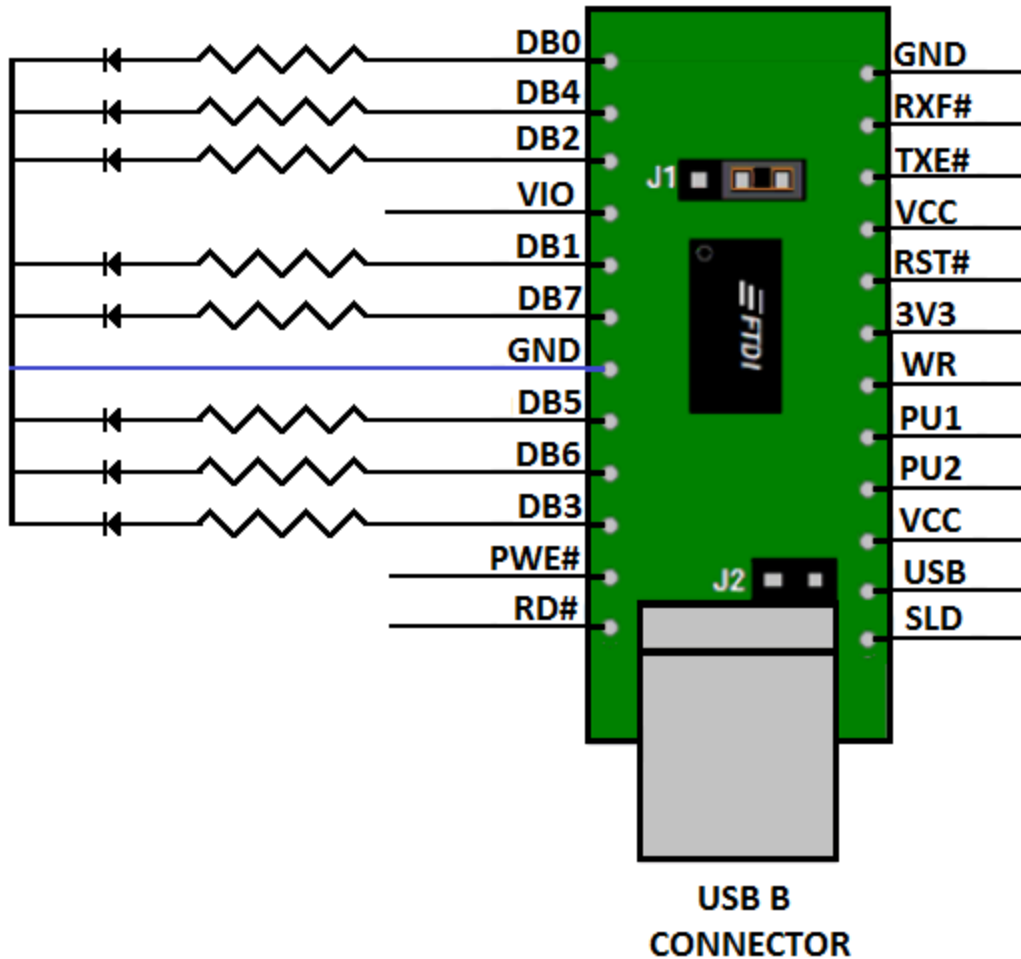
1. τον έλεγχο ενός Βηματικού Κινητήρα

2. τον έλεγχο ενός Ρομποτικού Βραχίονα

3. τον έλεγχο εργαλειομηχανής CNC (Φραιζας / Τόρνου)

4. τον έλεγχο βιομηχανικών αυτοματισμών

4.2 Κύκλωμα ελέγχου θυρών του UM245r ως εξόδοι με χρήση LED



4.2.1 Περιγραφή του Κυκλώματος

Για τον έλεγχο θυρών του Driver χρησιμοποίησα Leds , επειδή είναι ο πιο εύκολος τρόπος για να δούμε πως γίνεται ο έλεγχος των θυρών ως εξόδοι μέσω του προγράμματος.

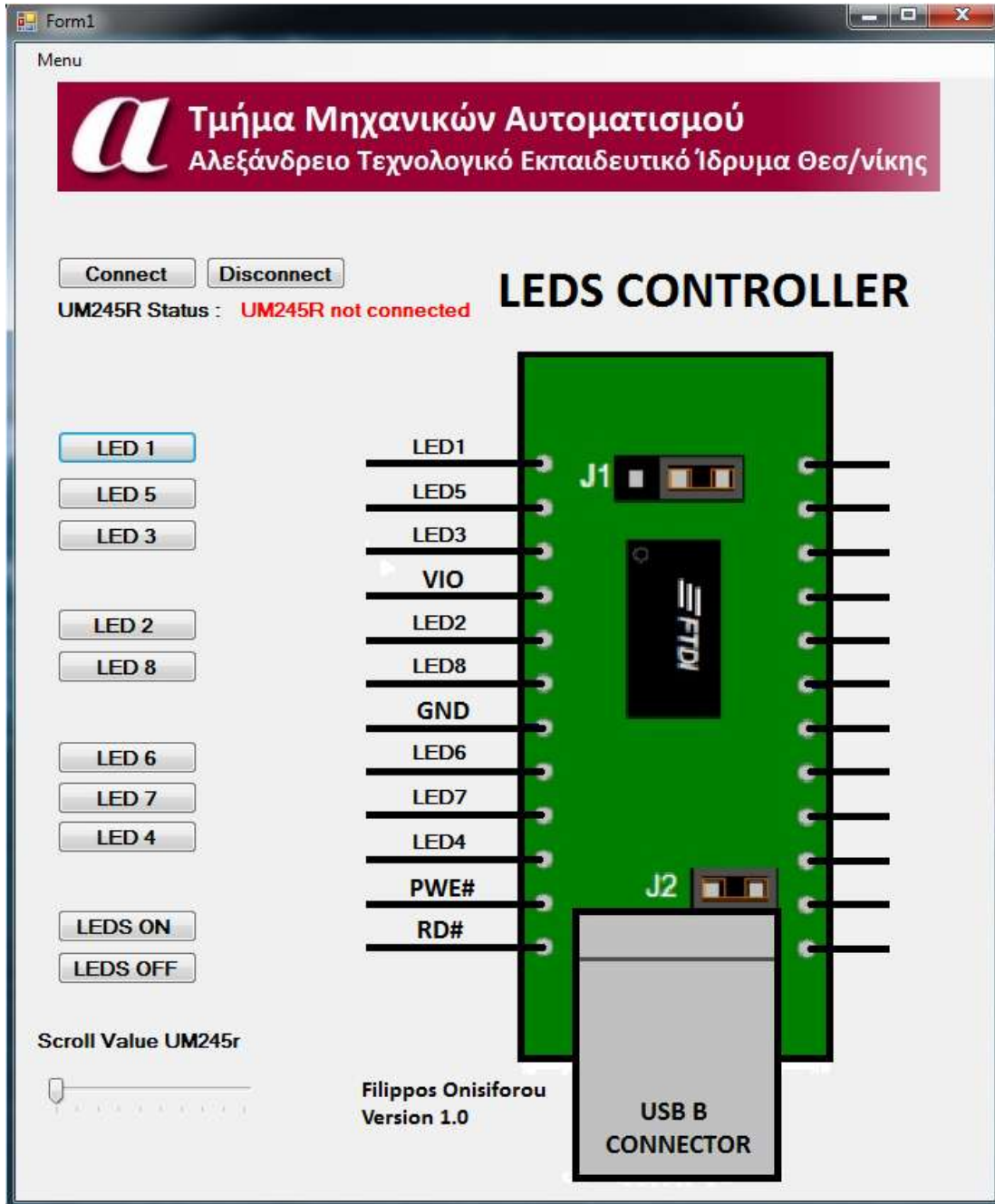
Για το κύκλωμα αυτό χρησιμοποίησα 8 Leds και 8 αντιστάσεις των 300Ω.

Οι τιμές των αντιστάσεων υπολογίζονται από την τιμή της τάσης εισόδου και το ρεύμα που απαιτεί το LED.

Τα άκρα των leds είναι γειωμένα με το pin GND του Driver όπως φαίνεται στην παράπανω εικόνα .

Στο πρόγραμμα τα Leds ενεργοποιούνται με λογικό 1 και απενεργοποιούνται με λογικό 0.

4.3 Interface ελέγχου των θυρών του UM245R με Led



4.3.1 Πηγάιος Κώδικας σε Visual Basic 2010 (Source Code) Παράρτημα 2^ο και περιγραφή του προγράμματος

→ Βήμα 1^ο

```
Public Class Form1
```

```
    Const DEVICE_NO As Short = 0 θέτουμε τον δείκτη της συσκευής σε 0  
    Dim m_DeviceHandle As Integer  
    Dim m_Count As Short  
    Dim ret As Integer
```

→ Βήμα 2^ο : Επικοινωνία μεταξύ VB και UM245R

```
Private Sub ConnectButton_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles ConnectButton.Click  
  
    If FT_Open(DEVICE_NO, m_DeviceHandle) <> FT_OK Then  
        Label9.Text = "UM245R not connected"  
        Exit Sub  
    Else  
        Label9.Text = "UM245R connected"  
    End If  
  
    If Label9.Text = "UM245R connected" Then  
        Label9.ForeColor = Color.Green  
    End If  
    If Label9.Text = "UM245R not connected" Then  
        Label9.ForeColor = Color.Red  
    End If
```

Ο πίο πάνω κώδικας συνδέεται με την συσκευή και επιστρέφει ενα χειριστή ο οποίος θα χρησιμοποιηθεί για τις επόμενες προσπελάσεις.

Device_NO ορίζετε ο δείκτης της συσκευής για να μπορέσει να ενεργοποιηθεί και παίρνει τιμή 0.

m_DeviceHandle είναι ο δείκτης σε μια μεταβλητή του τύπου FT_HANDLE όπου ο χειριστής θα πρέπει να αποθηκεύεται. Αυτός ο χειριστής είναι απαραίτητος για την πρόσβαση της συσκευής.

Η εντολή FT_OK επιστρέφει μια τιμή η οποία επιβεβαιώνει ότι η εντολή FT_OPEN έχει γίνει με επιτυχία.

Στον πίο πάνω κώδικα γίνεται έλεγχος στον οποίο όταν η συσκευή δεν ανιχνεύεται τότε η εντολή FT_OK δεν ισχύει οπότε θα εκτελεστεί η επόμενη εντολή:

```
Label9.Text = "UM245R not connected"
```

Εφόσον ισχυεί θα εκτελεστεί η εντολή : Label9.Text = "UM245R connected"

```

If FT_SetBitMode(m_DeviceHandle, &HFF, &H1) <> FT_OK Then
    Label19.Text = "ERROR"
    Label19.ForeColor = Color.Red
    Exit Sub
End If

```

Ο πιο πάνω κώδικας FT_SetBitMode επιτρέπει διαφορετικούς τρόπους λειτουργίας.

→ Βημα 3^ο : Ρυθμισή ταχύτητας επικοινωνίας

```

FT_SetBaudRate(m_DeviceHandle, 57600)
End Sub

```

Αυτή η εντολή λειτουργίας ρυθμίζει την ταχύτητα επικοινωνίας (baud-rate) μεταξύ της θύρας και της συσκευής η συγκεκριμένη ταχύτητα είναι η ταχύτητα όπως δίνεται απο το αρχείο D2XX_Programmer's_Guide(FT_000071).

→ Βημα 4^ο : Εισαγωγή TIMER

```

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

```

Εισάγουμε ένα Timer με τον οποίο σε κάθε κύκλο θα μπορούμε να ελέγχουμε με τις διάφορες εντολές τον Driver σε πραγματικό χρόνο.

→ Βημα 5^ο : Ενεργοποίηση λειτουργίας του Driver

```

Dim data As Byte

If FT_SetBitMode(m_DeviceHandle, &H0, &H1) <> FT_OK Then
    Label19.Text = "UM245R not connected"
    Label19.ForeColor = Color.Red
    Exit Sub

```

Η εντολή FT_SetBitMode ενεργοποιά διαφορετικούς τρόπους χρήσης του driver. Το m_DeviceHandle είναι ο χειριστής.

Με την τιμή &H0 ρυθμίζουμε την UcMask η οποία ρυθμίζει ποιά bits είναι εισόδοι και ποια εξόδοι. Η τιμή 0 καθορίζει το αντίστοιχο pin σε είσοδο ενώ η τιμή 1 καθορίζει το αντίστοιχο pin σε έξοδο.

Με την τιμή &H1 ρυθμίζουμε το ucMode δηλαδή το είδος λειτουργίας του driver όπου 1 είναι η λειτουργία σε Asynchronous Bit Bang mode.

Οι παρακάτω τιμές δηλώνουν και την χρήση λειτουργίας του Driver. Επίσης ο παρακάτω κατάλογος παρουσιάζει τις λειτουργίες χρήσης καθώς και σε ποιούς driver υποστηρίζονται.

0x0 = Reset

0x1 = Asynchronous Bit Bang

0x2 = MPSSE (FT2232, FT2232H, FT4232H and FT232H devices only)

0x4 = Synchronous Bit Bang (FT232R, FT245R, FT2232, FT2232H, FT4232H and FT232H devices only)

0x8 = MCU Host Bus Emulation Mode (FT2232, FT2232H, FT4232H and FT232H devices only)

0x10 = Fast Opto-Isolated Serial Mode (FT2232, FT2232H, FT4232H and FT232H devices only)

0x20 = CBUS Bit Bang Mode (FT232R and FT232H devices only)

0x40 = Single Channel Synchronous 245 FIFO Mode (FT2232H and FT232H devices only)

→ Βήμα 6^ο : Ρυθμισή παραμέτρων για τον διάυλο δεδομένων

End If

```
ret = FT_GetBitMode(m_DeviceHandle, data)
ret = FT_SetBitMode(m_DeviceHandle, &HFF, &H1)
```

End Sub

Η εντολή Ft_GetBitMode παίρνει τη στιγμιαία τιμή του διαύλου δεδομένων.

→ Βήμα 7^ο : Αποστολή Bytes στην συσκευή

```
Private Sub button1_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button1.MouseDown
```

```
ret = FT_WriteByte(m_DeviceHandle, 1, 1, 0)
```

Με την εντολή FT_WriteByte στέλλουμε bytes στην συσκευή.

Οι παραμέτροι είναι :

m_DeviceHandle: Είναι ο χειριστής της συσκευής.

lpBuffer: Είναι ο δείκτης του buffer που περιέχει τα δεδομένα τα οποία θα σταλούν στην συσκευή.

dwBytesToWrite: Είναι ο αριθμός των Bytes που θα σταλούν στην συσκευή.

lpdwBytesWritten: Είναι ο δείκτης σε μια μεταβλητή τύπου DWORD που δέχεται τον αριθμό των bytes που γράφονται στη συσκευή.

```
Timer1.Enabled = False
Label11.BackColor = Color.Red

Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

End Sub

```
Private Sub button1_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button1.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label11.BackColor = Color.Empty

    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
```

End Sub

```
Private Sub button2_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button2.MouseDown
    ret = FT_WriteByte(m_DeviceHandle, 2, 1, 0)
    Timer1.Enabled = False
    Label12.BackColor = Color.Red

    Label11.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
```



```

Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

Private Sub button2_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button2.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label12.BackColor = Color.Empty

    Label11.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
End Sub

Private Sub button3_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button3.MouseDown
    ret = FT_WriteByte(m_DeviceHandle, 4, 1, 0)
    Timer1.Enabled = False
    Label13.BackColor = Color.Red

    Label12.BackColor = Color.Empty
    Label11.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
End Sub

Private Sub button3_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button3.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label13.BackColor = Color.Empty

    Label12.BackColor = Color.Empty
    Label11.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty

```

```

Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

Private Sub button4_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button4.MouseDown
ret = FT_WriteByte(m_DeviceHandle, 8, 1, 0)
Timer1.Enabled = False
Label14.BackColor = Color.Red

Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label11.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

Private Sub Label14_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button4.MouseUp
ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
Timer1.Enabled = True
Label14.BackColor = Color.Empty

Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label11.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

Private Sub button5_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button5.MouseDown
ret = FT_WriteByte(m_DeviceHandle, 16, 1, 0)
Timer1.Enabled = False
Label15.BackColor = Color.Red

Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label11.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty

```

End Sub

```
Private Sub button5_MouseUp(ByVal sender As Object, ByVal e As  
System.Windows.Forms.MouseEventArgs) Handles Button5.MouseUp
```

```
ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
```

```
Timer1.Enabled = True
```

```
Label15.BackColor = Color.Empty
```

```
Label12.BackColor = Color.Empty
```

```
Label13.BackColor = Color.Empty
```

```
Label14.BackColor = Color.Empty
```

```
Label11.BackColor = Color.Empty
```

```
Label16.BackColor = Color.Empty
```

```
Label17.BackColor = Color.Empty
```

```
Label18.BackColor = Color.Empty
```

```
Button11.BackColor = Color.Empty
```

```
Button12.BackColor = Color.Empty
```

End Sub

```
Private Sub button6_MouseDown(ByVal sender As Object, ByVal e As  
System.Windows.Forms.MouseEventArgs) Handles Button6.MouseDown
```

```
ret = FT_WriteByte(m_DeviceHandle, 32, 1, 0)
```

```
Timer1.Enabled = False
```

```
Label16.BackColor = Color.Red
```

```
Label12.BackColor = Color.Empty
```

```
Label13.BackColor = Color.Empty
```

```
Label14.BackColor = Color.Empty
```

```
Label15.BackColor = Color.Empty
```

```
Label11.BackColor = Color.Empty
```

```
Label17.BackColor = Color.Empty
```

```
Label18.BackColor = Color.Empty
```

```
Button11.BackColor = Color.Empty
```

```
Button12.BackColor = Color.Empty
```

End Sub

```
Private Sub button6_MouseUp(ByVal sender As Object, ByVal e As  
System.Windows.Forms.MouseEventArgs) Handles Button6.MouseUp
```

```
ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
```

```
Timer1.Enabled = True
```

```
Label16.BackColor = Color.Empty
```

```
Label12.BackColor = Color.Empty
```

```
Label13.BackColor = Color.Empty
```

```
Label14.BackColor = Color.Empty
```

```
Label15.BackColor = Color.Empty
```

```
Label11.BackColor = Color.Empty
```

```
Label17.BackColor = Color.Empty
```

```
Label18.BackColor = Color.Empty
```

```
Button11.BackColor = Color.Empty
```

```
Button12.BackColor = Color.Empty
```

End Sub

```
Private Sub button7_MouseDown(ByVal sender As Object, ByVal e As  
System.Windows.Forms.MouseEventArgs) Handles Button7.MouseDown
```

```

ret = FT_WriteByte(m_DeviceHandle, 64, 1, 0)
Timer1.Enabled = False
Label17.BackColor = Color.Red

Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label11.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

Private Sub button7_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button7.MouseUp
ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
Timer1.Enabled = True
Label17.BackColor = Color.Empty

Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label11.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

Private Sub button8_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button8.MouseDown
ret = FT_WriteByte(m_DeviceHandle, 128, 1, 0)
Timer1.Enabled = False
Label18.BackColor = Color.Red

Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label11.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

Private Sub button8_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button8.MouseUp
ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
Timer1.Enabled = True
Label18.BackColor = Color.Empty

```

```

Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label11.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

```

→ Βήμα 8^ο : Φιλοσοφία του Interface και περιγραφή των εντολών

Για να μπορώ να ελέγχω τις εξόδους των 8 pins έχω φτιάξει 8 buttons, απο button1 εως button8.

Έχω δηλώσει τις εντολές ButtonX_MouseDown και ButtonX_MouseUp για τον έλεγχο των pin.

ButtonX_MouseDown: Σε αυτήν την εντολή όταν κρατάμε το δεξί κλικ στο ποντίκι πατημένο τότε θα στέλλει τα bytes και θα ενεργοποιούνται τα κατάλληλα pins.

ButtonX_MouseUp: Σε αυτή την εντολή όταν σταματάμε να κρατάμε το δεξί κλικ στο ποντίκι πατημένο τα bytes που στέλλονται σταματάνε και τα ανάλογα pins που κρατάνε τα bytes απενεργοποιούνται.

→ Βήμα 9^ο : Απενεργοποίηση της συσκευής

Ο πιο κάτω κώδικας είναι για να απενεργοποιηθεί η συσκευή μας.

Η εντολή αυτή που είναι υπεύθυνη για το κλείσιμο της συσκευής μας είναι η FT_Close. Με αυτή όμως την εντολή αντιμετωπίζα ένα προβληματάκι στο ότι εάν καποιά pins εμενάν ενεργοποιημένα και εκτελούσα την εντολή FT_Close αυτά τα pins παρέμεναν ενεργοποιημένα.

Οπότε η λογική για το κλείσιμο είναι ότι πρέπει να βεβαιωθούμε ότι όλα τα pins της συσκευής θα είναι απενεργοποιημένα έτσι θα προστατεύονται , γράφοντας τις εντολές με την ακόλουθη προτεραιότητα .

```

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button10.Click

```

```

    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True

```

```
If FT_WriteByte(m_DeviceHandle, 0, 1, 0) Then
    TrackBar1.Value = 0
End If
```

```
Label11.BackColor = Color.Empty
Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty
```

```
Button12.BackColor = Color.Empty
Button11.BackColor = Color.Empty
```

```
ret = FT_Close(m_DeviceHandle)
Label19.Text = "UM245R Disconnected ! "
Label19.ForeColor = Color.Red
```

```
End Sub
```

Η εντολή αυτή είναι για να απενεργοποιήσουμε τυχόν ενεργά pins.

```
ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
Timer1.Enabled = True
```

Η εντολή αυτή είναι για να φέρουμε το TrackBar1 στην αρχική του τιμή συγκρίνοντας το με την τιμή των pins.

```
If FT_WriteByte(m_DeviceHandle, 0, 1, 0) Then
    TrackBar1.Value = 0
End If
```

→ **Βήμα 9^ο : Κλείσιμο του προγράμματος**

```
Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ExitToolStripMenuItem.Click
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
```

```
Timer1.Enabled = True
```

```
Label11.BackColor = Color.Empty  
Label12.BackColor = Color.Empty  
Label13.BackColor = Color.Empty  
Label14.BackColor = Color.Empty  
Label15.BackColor = Color.Empty  
Label16.BackColor = Color.Empty  
Label17.BackColor = Color.Empty  
Label18.BackColor = Color.Empty
```

```
Close()
```

```
End Sub
```

Στον παραπάνω κώδικα ο οποίος είναι το Exit απο το Menu Tool Strip με την μόνη διαφορά αυτή η εντολή θα κλείσει το interface μας και για να μην μείνουν ενεργοποιημένα κάποια pins , η λογική είναι η ίδια με τον προηγούμενο κώδικα για την απενεργοποίηση των pins.

→ Βήμα 10^ο : Έλεγχος των pins

```
Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button12.Click
```

```
ret = FT_WriteByte(m_DeviceHandle, 255, 1, 0)  
Button12.BackColor = Color.Green  
Button11.BackColor = Color.Empty
```

```
Timer1.Enabled = False
```

```
Label11.BackColor = Color.Red  
Label12.BackColor = Color.Red  
Label13.BackColor = Color.Red  
Label14.BackColor = Color.Red  
Label15.BackColor = Color.Red  
Label16.BackColor = Color.Red  
Label17.BackColor = Color.Red  
Label18.BackColor = Color.Red
```

```
End Sub
```

Ο πιο πάνω κώδικας ενεργοποιεί όλα τα pins όπου 255 είναι η μέγιστη τιμή που μπορούμε να δώσουμε για να ενεργοποιήσουμε όλα τα pins.

```
Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button11.Click
```

```
Button12.BackColor = Color.Empty  
Button11.BackColor = Color.Red  
ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
```

```
Timer1.Enabled = True

Label1.BackColor = Color.Empty
Label2.BackColor = Color.Empty
Label3.BackColor = Color.Empty
Label4.BackColor = Color.Empty
Label5.BackColor = Color.Empty
Label6.BackColor = Color.Empty
Label7.BackColor = Color.Empty
Label8.BackColor = Color.Empty
```

End Sub

Ο παραπάνω κώδικας απενεργοποιεί όλα τα pins της συσκευής.

→ Βήμα 11^ο : Trackbar

```
Private Sub TrackBar1_Scroll(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TrackBar1.Scroll
```

```
    If TrackBar1.Value = 0 Then

        ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)

        Label1.BackColor = Color.Empty
        Label2.BackColor = Color.Empty
        Label3.BackColor = Color.Empty
        Label4.BackColor = Color.Empty
        Label5.BackColor = Color.Empty
        Label6.BackColor = Color.Empty
        Label7.BackColor = Color.Empty
        Label8.BackColor = Color.Empty

        Button11.BackColor = Color.Empty
        Button12.BackColor = Color.Empty

    ElseIf TrackBar1.Value = 1 Then

        ret = FT_WriteByte(m_DeviceHandle, 1, 1, 0)

        Label1.BackColor = Color.Red

        Label2.BackColor = Color.Empty
        Label3.BackColor = Color.Empty
        Label4.BackColor = Color.Empty
        Label5.BackColor = Color.Empty
        Label6.BackColor = Color.Empty
        Label7.BackColor = Color.Empty
        Label8.BackColor = Color.Empty

        Button11.BackColor = Color.Empty
        Button12.BackColor = Color.Empty
```



```
ElseIf TrackBar1.Value = 2 Then

    ret = FT_WriteByte(m_DeviceHandle, 2, 1, 0)

    Label1.BackColor = Color.Empty
    Label2.BackColor = Color.Red

    Label3.BackColor = Color.Empty
    Label4.BackColor = Color.Empty
    Label5.BackColor = Color.Empty
    Label6.BackColor = Color.Empty
    Label7.BackColor = Color.Empty
    Label8.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty

ElseIf TrackBar1.Value = 3 Then

    ret = FT_WriteByte(m_DeviceHandle, 4, 1, 0)

    Label1.BackColor = Color.Empty
    Label2.BackColor = Color.Empty

    Label3.BackColor = Color.Red

    Label4.BackColor = Color.Empty
    Label5.BackColor = Color.Empty
    Label6.BackColor = Color.Empty
    Label7.BackColor = Color.Empty
    Label8.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty

ElseIf TrackBar1.Value = 4 Then

    ret = FT_WriteByte(m_DeviceHandle, 8, 1, 0)

    Label1.BackColor = Color.Empty
    Label2.BackColor = Color.Empty
    Label3.BackColor = Color.Empty

    Label4.BackColor = Color.Red

    Label5.BackColor = Color.Empty
    Label6.BackColor = Color.Empty
    Label7.BackColor = Color.Empty
    Label8.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty

ElseIf TrackBar1.Value = 5 Then

    ret = FT_WriteByte(m_DeviceHandle, 16, 1, 0)
```

```
Label11.BackColor = Color.Empty
Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty

Label15.BackColor = Color.Red

Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 6 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 32, 1, 0)

Label11.BackColor = Color.Empty
Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty

Label16.BackColor = Color.Red

Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 7 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 64, 1, 0)

Label11.BackColor = Color.Empty
Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty

Label17.BackColor = Color.Red

Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 8 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 128, 1, 0)

Label11.BackColor = Color.Empty
Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
```

```
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty

Label18.BackColor = Color.Red

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 9 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 255, 1, 0)
```

```
Label11.BackColor = Color.Red
Label12.BackColor = Color.Red
Label13.BackColor = Color.Red
Label14.BackColor = Color.Red
Label15.BackColor = Color.Red
Label16.BackColor = Color.Red
Label17.BackColor = Color.Red
Label18.BackColor = Color.Red
```

```
Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

```
End If
```

```
End Sub
End Class
```

Το TrackBar το είχα προσθέσει τελευταίο. Με το TrackBar μπορούμε να ενεργοποιούμε τα pins ανάλογα με την τιμή του Trackbar. Η Φιλοσοφία είναι ότι με αυτό τον τρόπο θα μπορούσαμε για παράδειγμα να ελέγχαμε την ταχύτητα ενός βηματικού κινητήρα ακόμα και την φορά του.

ΚΕΦΑΛΑΙΟ 5^ο : ΕΛΕΓΧΟΣ ΤΩΝ ΘΥΡΩΝ ΤΟΥ UM245R ΩΣ ΕΙΣΟΔΟΙ

5.1 Περιγραφή του προγράμματος

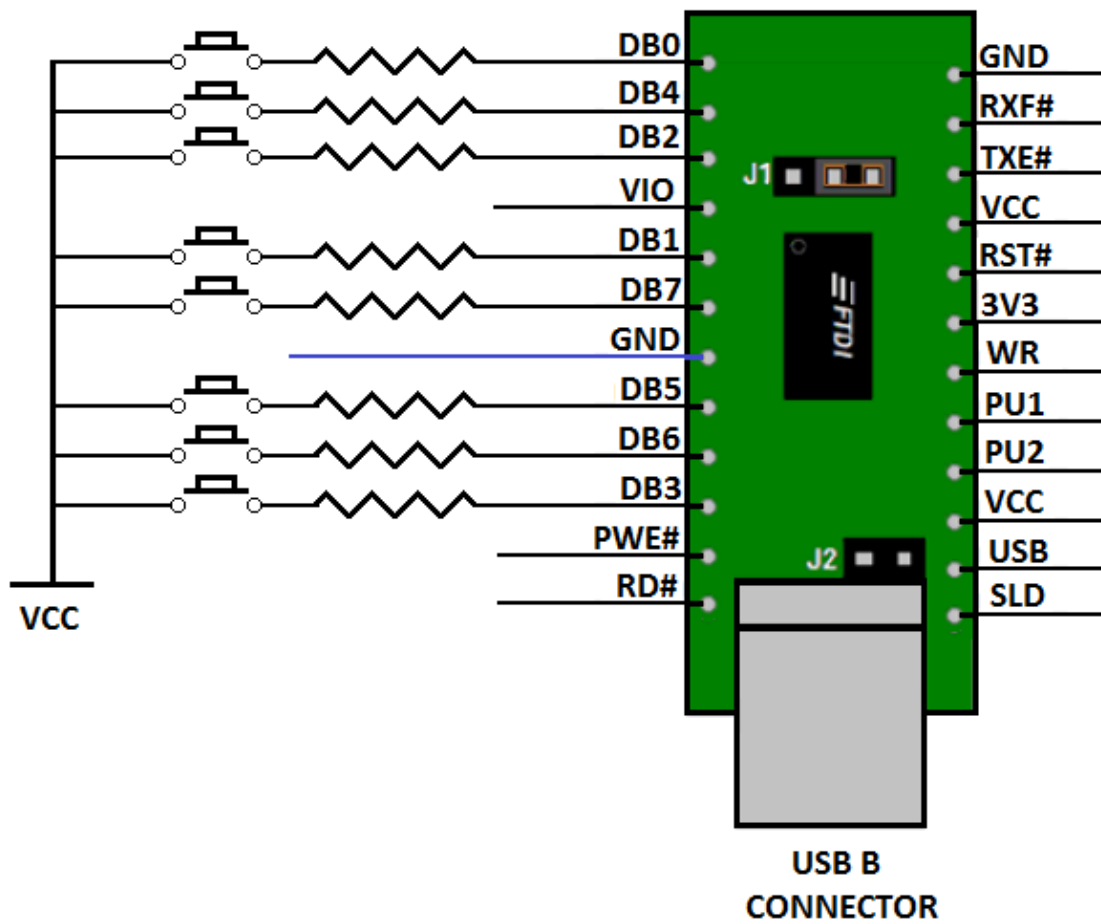
Σε αυτό το κεφάλαιο θα δούμε πως μπορούμε να ελέγχουμε τις θύρες του UM245R ως εισόδους μέσω της Visual Basic με την χρήση διακοπών. Με το παράδειγμα αυτό μπορούμε να κατανοήσουμε πως λειτουργεί ο συγκεκριμένος driver και οι θύρες του ως είσοδοι απλά και εύκολα .Ο κώδικας είναι απλός και εύκολος και μπορεί να τροποποιηθεί σύμφωνα με τις ανάγκες τις οποίες θέλουμε να λειτουργεί και να συμπεριφέρεται .

Με αυτό το παράδειγμα θα παρατηρήσουμε ότι ο συγκεκριμένος driver μπορεί να εφαρμοστεί για:

- 1 τον έλεγχο ενός Συστήματος Συναγερμού.
- 2 τον έλεγχο αισθητήρων Θερμοκρασίας και Υγρασίας προσθέτοντας έναν ADC.
- 3 τον έλεγχο αισθητήρων κίνησης ή φωτός.
- 4 τον έλεγχο βιομηχανικών αυτοματισμών.

5.2 Κύκλωμα ελέγχου των θυρών του UM245R ως είσοδοι με διακοπτάκια

Κύκλωμα

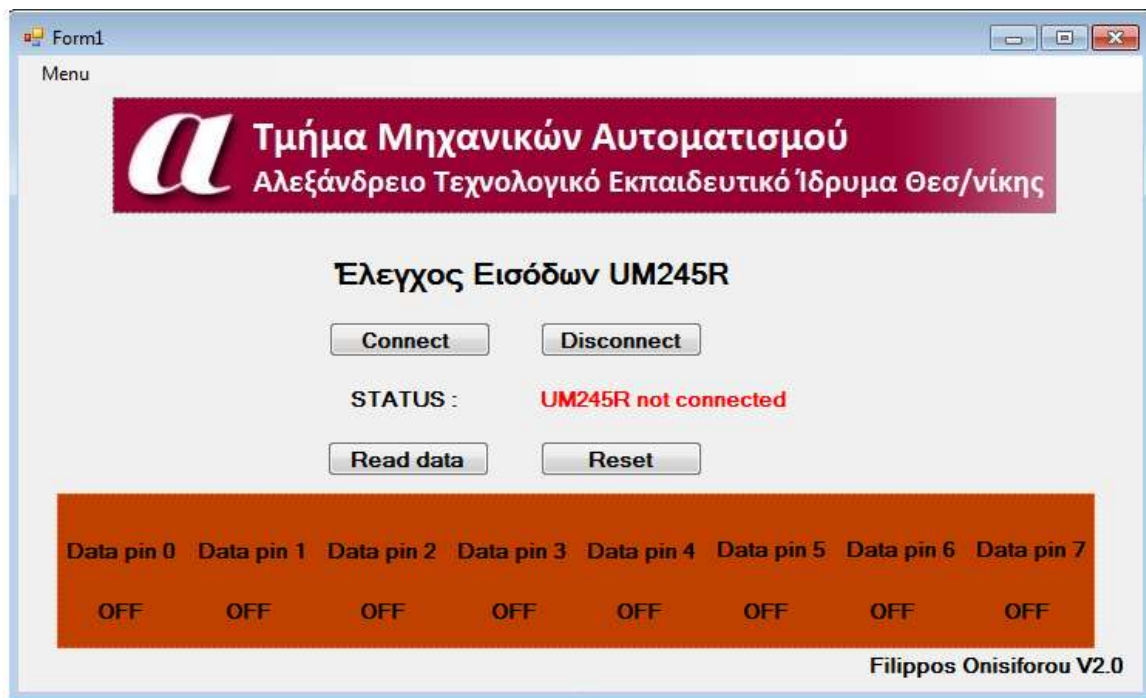


5.2.1 Περιγραφή του Κυκλώματος

Για να ενεργοποιούμε με λογικό 0 τις εισόδους θα πρέπει να χρησιμοποιήσουμε αντιστάσεις στις θύρες απο DB0-DB7 και οι εισόδοι να πηγαίνουν στην γείωση.

Για να ενεργοποιούμε με λογικό 1 τις εισόδους θα πρέπει να χρησιμοποιήσουμε τα 5V που μας δίνει η τροφοδοσία του UM245R και 10KΩ αντιστάσεις για προστασία.

5.3 Interface ελέγχου των θυρών του UM245R ως είσοδοι του με διακοπτάκια



5.3.1 Πηγάιος Κώδικας σε Visual Basic 2010 (Source Code) Παράρτημα 3^ο και περιγραφή του προγράμματος

→ Βήμα 1^ο

```
Public Class Form1
```

```
Const DEVICE_NO As Short = 0
```

```

Dim m_DeviceHandle As Integer
Dim m_Count As Short
Dim ret As Integer

Private Declare Function GetAsyncKeyState Lib "user32" (ByVal vkey As Long) As Short

```

→ Βήμα 2^ο

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

```

```

    If FT_Open(DEVICE_NO, m_DeviceHandle) <> FT_OK Then
        Label11.Text = "UM245R not connected"
        Exit Sub
    Else
        Label11.Text = "UM245R connected"
    End If

    If Label11.Text = "UM245R connected" Then
        Label11.ForeColor = Color.Green
    End If

    If Label11.Text = "UM245R not connected" Then
        Label11.ForeColor = Color.Red
    End If

```

Ο πίο πάνω κώδικας συνδέεται με την συσκευή και επιστρέφει ένα χειριστή οποία θα χρησιμοποιηθεί για τις επόμενες προσπέλασεις ,

Device_NO ορίζετε ο δείκτης της συσκευής για να μπορεί να ανοιξει και παίρνει τιμή 0
m_DeviceHandle Είναι ο δείκτης σε μια μεταβλητή της τύπου FT_HANDLE όπου η λαβή θα πρέπει να αποθηκεύεται. Αυτή ή λαβή είναι απαραίτητη για την πρόσβαση της συσκευής.

FT_OK επιστρέφει μια τιμή η οποία επιβεβαιώνει ότι η εντολή FT_OPEN έχει γίνει με επιτυχία

Στον πίο πάνω κώδικα γίνεται έλεγχος στον οποίο όταν η συσκευή δεν ανιχνευτεί τότε η εντολή FT_OK δεν μπορεί να ισχύσει οπότε θα εκτελεστεί η επόμενη εντολή :

```
Label9.Text = "UM245R not connected"
```

Εφόσον ισχυεί θα εκτελεστεί η εντολή : Label9.Text = "UM245R connected"

```

If FT_SetBitMode(m_DeviceHandle, &HFF, &H1) <> FT_OK Then
    Label11.Text = "ERROR"

```

```
Label11.ForeColor = Color.Red
Exit Sub
End If
```

→ Βημα 3^ο : Ρυθμισή ταχύτητας επικοινωνίας

```
FT_SetBaudRate(m_DeviceHandle, 57600)
End Sub
```

Αυτή η εντολή λειτουργίας ρυθμίζει την ταχύτητα επικοινωνίας (baud-rate) μεταξύ της θύρας και της συσκευής η συγκεκριμένη ταχύτητα είναι η ταχύτητα όπως δίνεται απο το αρχείο D2XX_Programmer's_Guide(FT_000071).

→ Βημα 4^ο : Εισαγωγή TIMER

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
```

Εισάγουμε ένα Timer με τον οποίο σε κάθε κύκλο θα μπορούμε να ελέγχουμε με τις διάφορες εντολές τον Driver σε πραγματικό χρόνο.

→ Βημα 5^ο : Ενεργοποίηση λειτουργίας του Driver

```
Dim data As Byte

If FT_SetBitMode(m_DeviceHandle, &H0, &H1) <> FT_OK Then
Label19.Text = "UM245R not connected"
Label19.ForeColor = Color.Red
Exit Sub
```

Η εντολή FT_SetBitMode ενεργοποιά διαφορετικούς τρόπους χρήσης του driver

Το m_DeviceHandle είναι ο χειριστής.

Με την τιμή &H0 ρυθμίζουμε την UcMask η οποία ρυθμίζει ποιά bits είναι εισόδους και εξόδους. Η τιμή 0 καθορίζει το αντίστοιχο pin σε είσοδο ενώ η τιμή 1 καθορίζει το αντίστοιχο pin σε έξοδο.

Με την τιμή &H1 ρυθμίζουμε το ucMode δηλαδή το είδος λειτουργίας του driver όπου 1 είναι η λειτουργία σε Asynchronous Bit Bang mode.

Οι παρακάτω τιμές δηλώνουν και την χρήση λειτουργίας του Driver, επίσης ο παρακάτω κατάλογος παρουσιάζει τις λειτουργίες χρήσης καθώς και σε ποιούς driver υποστηρίζονται.

0x0 = Reset

0x1 = Asynchronous Bit Bang

0x2 = MPSSE (FT2232, FT2232H, FT4232H and FT232H devices only)

0x4 = Synchronous Bit Bang (FT232R, FT245R, FT2232, FT2232H, FT4232H and FT232H devices only)

0x8 = MCU Host Bus Emulation Mode (FT2232, FT2232H, FT4232H and FT232H devices only)

0x10 = Fast Opto-Isolated Serial Mode (FT2232, FT2232H, FT4232H and FT232H devices only)

0x20 = CBUS Bit Bang Mode (FT232R and FT232H devices only)

0x40 = Single Channel Synchronous 245 FIFO Mode (FT2232H and FT232H devices only)

→ **Βήμα 6^ο : Ρυθμισή παραμέτρων για τον διάυλο δεδομένων**

```
If FT_SetBitMode(m_DeviceHandle, &H0, &H1) <> FT_OK Then
```

```
    Label1.BackColor = Color.Empty  
    Label2.BackColor = Color.Empty  
    Label3.BackColor = Color.Empty  
    Label4.BackColor = Color.Empty  
    Label5.BackColor = Color.Empty  
    Label6.BackColor = Color.Empty  
    Label7.BackColor = Color.Empty  
    Label8.BackColor = Color.Empty
```

```
Exit Sub
```

```
End If
```

```
ret = FT_SetBitMode(m_DeviceHandle, &HFF, &H1)
```

```
End Sub
```

Με την πιο πάνω εντολή ρυθμίζουμε τα pins του Driver σε εισόδους.

→ **Βήμα 7^ο : Κλείσιμο του προγράμματος**

```
Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles ExitToolStripMenuItem.Click  
    Close()  
End Sub
```

→ Βήμα 8^ο : Απενεργοποίηση του Driver

Ο πιο κάτω κώδικας είναι για να απενεργοποιηθεί η συσκευή μας. Η εντολή αυτή που είναι υπεύθυνη για την απενεργοποίηση της συσκευής μας είναι η FT_Close.

Εδώ η λογική για το κλείσιμο είναι ότι πρέπει να βεβαιωθούμε ότι όλα τα Labels του Interface θα είναι απενεργοποιημένα, γράφοντας τις εντολές με την ακόλουθη προτεραιότητα .

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Timer1.Enabled = False

    Label11.Text = "OFF "
    Label11.ForeColor = Color.Black
    Label12.Text = "OFF "
    Label12.ForeColor = Color.Black
    Label13.Text = "OFF "
    Label13.ForeColor = Color.Black
    Label14.Text = "OFF "
    Label14.ForeColor = Color.Black
    Label15.Text = "OFF "
    Label15.ForeColor = Color.Black
    Label16.Text = "OFF "
    Label16.ForeColor = Color.Black
    Label17.Text = "OFF "
    Label17.ForeColor = Color.Black
    Label18.Text = "OFF "
    Label18.ForeColor = Color.Black

    Dim ret As Integer
    ret = FT_Close(m_DeviceHandle)
    Label111.Text = " UM245r disconnected "
    Label111.ForeColor = Color.Red

End Sub
```

→ Βήμα 9^ο : Μπουτόν Read data

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Timer2.Enabled = True

End Sub
```

Με την πιο πάνω εντολή ενεργοποιούμε τον Timer2 ο οποίος είναι υπεύθυνος για το διάβασμα των εισόδων του Driver.

→ Βήμα 10^ο : Μπουτόν Reset

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Timer2.Enabled = False
    Label11.Text = "OFF "
    Label11.ForeColor = Color.Black
    Label12.Text = "OFF "
    Label12.ForeColor = Color.Black
    Label13.Text = "OFF "
    Label13.ForeColor = Color.Black
    Label14.Text = "OFF "
    Label14.ForeColor = Color.Black
    Label15.Text = "OFF "
    Label15.ForeColor = Color.Black
    Label16.Text = "OFF "
    Label16.ForeColor = Color.Black
    Label17.Text = "OFF "
    Label17.ForeColor = Color.Black
    Label18.Text = "OFF "
    Label18.ForeColor = Color.Black

End Sub

```

Ο πιο πάνω κώδικας είναι υπεύθυνος για το Reset σε περίπτωση που θέλουμε να απενεργοποιήσουμε το διάβασμα των pins.

→ Βήμα 11^ο : Ενεργοποίηση διάβασματος των εισόδων του Driver

```

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick

    Dim ret As Integer
    Dim data As Byte

    ret = FT_GetBitMode(m_DeviceHandle, data)

    ret = FT_SetBitMode(m_DeviceHandle, &HFF, &H1)

```

Η πιο πάνω εντολή παίρνει την στιγμιαία τιμή του διαύλου δεδομένων.

→ Βήμα 12^ο : Παραμέτροι για το διάβασμα των εισόδων του Driver

```

If CStr(data) = 1

    Label11.Text = " ON "
    Label11.ForeColor = Color.Red

Else

    Label11.Text = "OFF "
    Label11.ForeColor = Color.Black

End If

```

```
If CStr(data) = 2
    Label2.Text = " ON "
    Label2.ForeColor = Color.Red
Else
    Label2.Text = "OFF "
    Label2.ForeColor = Color.Black
End If

If CStr(data) = 4 Then
    Label3.Text = " ON "
    Label3.ForeColor = Color.Red
Else
    Label3.Text = "OFF "
    Label3.ForeColor = Color.Black
End If

If CStr(data) = 8 Then
    Label4.Text = " ON "
    Label4.ForeColor = Color.Red
Else
    Label4.Text = "OFF "
    Label4.ForeColor = Color.Black
End If

If CStr(data) = 16 Then
    Label5.Text = " ON "
    Label5.ForeColor = Color.Red
Else
    Label5.Text = "OFF "
    Label5.ForeColor = Color.Black
End If

If CStr(data) = 32 Then
    Label6.Text = " ON "
    Label6.ForeColor = Color.Red
Else
    Label6.Text = "OFF "
    Label6.ForeColor = Color.Black
End If

If CStr(data) = 64 Then
    Label7.Text = " ON "
    Label7.ForeColor = Color.Red
```

```

Else
    Label17.Text = "OFF "
    Label17.ForeColor = Color.Black

End If

If CStr(data) = 128 Then
    Label18.Text = " ON "
    Label18.ForeColor = Color.Red
Else
    Label18.Text = "OFF "
    Label18.ForeColor = Color.Black

End If

End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub
End Class

```

Η εντολή `CStr(data)` επιστρέφει True or False για την τιμή που δέχετε στις εισόδους ο Driver και εφόσον η τιμή που κρατάει ο Driver στα pins είναι ίση με την κάθε περίπτωση, τότε το αναλογό Label στο interface ενεργοποιείται.

Επίσης στον κώδικα πρέπει να προσέξουμε τις εισόδους. Δηλαδή όταν είναι ενεργοποιημένα τα pins DB0 και DB1 θα έχουμε 3 , επομένως θα πρέπει να γράψουμε για όλους τους συνδιασμούς τιμών που θα προκύψουν όταν κάποια pins θα είναι ταυτόχρονα ενεργοποιημένα. Σύνολο 255 συνδιασμούς.

ΠΑΡΑΡΤΗΜΑ Ι: Αρχείο D2XX_MODULE.VB

Το αρχείο D2XX_module.vb η οποία βιβλιοθήκη περιέχει τις εντολές για την χρήση του UM245R το οποίο παραχωρείται απο εταιρεία απο στην ιστοσελίδα της .

```
Option Strict Off
Option Explicit On
Module D2XX_module

    'Public MainForm As DEMO_EEPROM
    '=====
    'CLASSIC INTERFACE
    '=====
    Public Declare Function FT_ListDevices Lib "FTD2XX.DLL" (ByVal arg1 As Integer, ByVal arg2 As String, ByVal dwFlags As Integer) As Integer
    Public Declare Function FT_GetNumDevices Lib "FTD2XX.DLL" Alias "FT_ListDevices" (ByRef arg1 As Integer, ByVal arg2 As String, ByVal dwFlags As Integer) As Integer

    Public Declare Function FT_Open Lib "FTD2XX.DLL" (ByVal intDeviceNumber As Short, ByRef lngHandle As Integer) As Integer
    Public Declare Function FT_OpenEx Lib "FTD2XX.DLL" (ByVal arg1 As String, ByVal arg2 As Integer, ByRef lngHandle As Integer) As Integer
    Public Declare Function FT_Close Lib "FTD2XX.DLL" (ByVal lngHandle As Integer) As Integer
    Public Declare Function FT_Read Lib "FTD2XX.DLL" (ByVal lngHandle As Integer, ByVal lpszBuffer As String, ByVal lngBufferSize As Integer, ByRef lngBytesReturned As Integer) As Integer
    Public Declare Function FT_Write Lib "FTD2XX.DLL" (ByVal lngHandle As Integer, ByVal lpszBuffer As String, ByVal lngBufferSize As Integer, ByRef lngBytesWritten As Integer) As Integer
    'UPGRADE_ISSUE: Declaring a parameter 'As Any' is not supported. Click for more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="FAE78A8D-8978-4FD4-8208-5B7324A8F795"'
```

```

    Public Declare Function FT_WriteByte Lib "FTD2XX.DLL" Alias "FT_Write" (ByVal
lngHandle As Integer, ByRef lpszBuffer As Byte, ByVal lngBufferSize As Integer,
ByRef lngBytesWritten As Integer) As Integer
    Public Declare Function FT_SetBaudRate Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByVal lngBaudRate As Integer) As Integer
    Public Declare Function FT_SetDataCharacteristics Lib "FTD2XX.DLL" (ByVal
lngHandle As Integer, ByVal byWordLength As Byte, ByVal byStopBits As Byte, ByVal
byParity As Byte) As Integer
    Public Declare Function FT_SetFlowControl Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByVal intFlowControl As Short, ByVal byXonChar As Byte, ByVal byXoffChar
As Byte) As Integer
    Public Declare Function FT_SetDtr Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer) As Integer
    Public Declare Function FT_ClrDtr Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer) As Integer
    Public Declare Function FT_SetRts Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer) As Integer
    Public Declare Function FT_ClrRts Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer) As Integer
    Public Declare Function FT_GetModemStatus Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByRef lngModemStatus As Integer) As Integer
    Public Declare Function FT_SetChars Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByVal byEventChar As Byte, ByVal byEventCharEnabled As Byte, ByVal
byErrorChar As Byte, ByVal byErrorCharEnabled As Byte) As Integer
    Public Declare Function FT_Purge Lib "FTD2XX.DLL" (ByVal lngHandle As Integer,
ByVal lngMask As Integer) As Integer
    Public Declare Function FT_SetTimeouts Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByVal lngReadTimeout As Integer, ByVal lngWriteTimeout As Integer) As
Integer
    Public Declare Function FT_GetQueueStatus Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByRef lngRxBytes As Integer) As Integer
    Public Declare Function FT_SetBreakOn Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer) As Integer
    Public Declare Function FT_SetBreakOff Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer) As Integer
    Public Declare Function FT_GetStatus Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByRef lngRxBytes As Integer, ByRef lngTxBytes As Integer, ByRef
lngEventsDWord As Integer) As Integer
    Public Declare Function FT_SetEventNotification Lib "FTD2XX.DLL" (ByVal
lngHandle As Integer, ByVal dwEventMask As Integer, ByVal pVoid As Integer) As
Integer
    Public Declare Function FT_ResetDevice Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer) As Integer
    'Public Declare Function FT_SetDivisor Lib "FTD2XX.DLL" (ByVal lngHandle As
Long, ByVal shDivisor) As Short

    'Public Declare Function FT_GetEventStatus Lib "FTD2XX.DLL" (ByVal lngHandle
As Long, ByRef lngEventsDWord As Long) As Long

```

'UPGRADE_ISSUE: Declaring a parameter 'As Any' is not supported. Click for more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="FAE78A8D-8978-4FD4-8208-5B7324A8F795"'

```

    Public Declare Function FT_GetBitMode Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByRef intData As Byte) As Integer
    Public Declare Function FT_SetBitMode Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByVal intMask As Byte, ByVal intMode As Byte) As Integer

```

```
Public Declare Function FT_SetLatencyTimer Lib "FTD2XX.DLL" (ByVal Handle As Integer, ByVal pucTimer As Byte) As Integer
Public Declare Function FT_GetLatencyTimer Lib "FTD2XX.DLL" (ByVal Handle As Integer, ByRef ucTimer As Integer) As Integer
```

```
'=====
'FT_W32 API
'=====
```

```
'UPGRADE_WARNING: Structure LPSECURITY_ATTRIBUTES may require marshalling attributes to be passed as an argument in this Declare statement. Click for more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="C429C3A5-5D47-4CD9-8F51-74A1616405DC"'
```

```
Public Declare Function FT_W32_CreateFile Lib "FTD2XX.DLL" (ByVal lpzName As String, ByVal dwAccess As Integer, ByVal dwShareMode As Integer, ByRef lpSecurityAttributes As LPSECURITY_ATTRIBUTES, ByVal dwCreate As Integer, ByVal dwAttrsAndFlags As Integer, ByVal hTemplate As Integer) As Integer
```

```
Public Declare Function FT_W32_CloseHandle Lib "FTD2XX.DLL" (ByVal ftHandle As Integer) As Integer
```

```
'UPGRADE_WARNING: Structure lpOverlapped may require marshalling attributes to be passed as an argument in this Declare statement. Click for more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="C429C3A5-5D47-4CD9-8F51-74A1616405DC"'
```

```
Public Declare Function FT_W32_ReadFile Lib "FTD2XX.DLL" (ByVal lngHandle As Integer, ByVal lpzBuffer As String, ByVal lngBufferSize As Integer, ByRef lngBytesReturned As Integer, ByRef lpftOverlapped As lpOverlapped) As Integer
```

```
'UPGRADE_WARNING: Structure lpOverlapped may require marshalling attributes to be passed as an argument in this Declare statement. Click for more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="C429C3A5-5D47-4CD9-8F51-74A1616405DC"'
```

```
Public Declare Function FT_W32_WriteFile Lib "FTD2XX.DLL" (ByVal lngHandle As Integer, ByVal lpzBuffer As String, ByVal lngBufferSize As Integer, ByRef lngBytesWritten As Integer, ByRef lpftOverlapped As lpOverlapped) As Integer
```

```
'UPGRADE_WARNING: Structure lpOverlapped may require marshalling attributes to be passed as an argument in this Declare statement. Click for more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="C429C3A5-5D47-4CD9-8F51-74A1616405DC"'
```

```
Public Declare Function FT_W32_GetOverlappedResult Lib "FTD2XX.DLL" (ByVal lngHandle As Integer, ByRef lpftOverlapped As lpOverlapped, ByRef lpdwBytesTransferred As Integer, ByVal bWait As Boolean) As Integer
```

```
'UPGRADE_WARNING: Structure FTDCB may require marshalling attributes to be passed as an argument in this Declare statement. Click for more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="C429C3A5-5D47-4CD9-8F51-74A1616405DC"'
```

```
Public Declare Function FT_W32_GetCommState Lib "FTD2XX.DLL" (ByVal lngHandle As Object, ByRef lpftDCB As FTDCB) As Integer
```

```
'UPGRADE_WARNING: Structure FTDCB may require marshalling attributes to be passed as an argument in this Declare statement. Click for more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="C429C3A5-5D47-4CD9-8F51-74A1616405DC"'
```

```
Public Declare Function FT_W32_SetCommState Lib "FTD2XX.DLL" (ByVal lngHandle As Object, ByRef lpftDCB As FTDCB) As Integer
```

```
Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Integer)
```



```

Public Declare Function CreateEvent Lib "kernel32" Alias "CreateEventA" (ByVal
lpEventAttributes As Integer, ByVal bManualReset As Integer, ByVal bInitialState
As Integer, ByVal lpName As String) As Integer
Public Declare Function WaitForSingleObject Lib "kernel32" (ByVal hHandle As
Integer, ByVal dwMilliseconds As Integer) As Integer
Public Declare Function SetEvent Lib "kernel32" (ByVal hHandle As Integer) As
Integer
'UPGRADE_ISSUE: Declaring a parameter 'As Any' is not supported. Click for
more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="FAE78A8D-
8978-4FD4-8208-5B7324A8F795"'
'UPGRADE_ISSUE: Declaring a parameter 'As Any' is not supported. Click for
more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="FAE78A8D-
8978-4FD4-8208-5B7324A8F795"'
Public Declare Function CreateThread Lib "kernel32" (ByRef lpThreadAttributes
As Byte, ByVal dwStackSize As Integer, ByVal lpStartAddress As Integer, ByRef
lpParameter As Byte, ByVal dwCreationFlags As Integer, ByRef lpThreadId As
Integer) As Integer
Public Declare Function TerminateThread Lib "kernel32" (ByVal hThread As
Integer, ByVal dwExitCode As Integer) As Integer
Public Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Integer)
As Integer

'=====
'APIGID32.DLL by DESAWARE Inc. (www.desaware.com), see Dan Appleman's
'"Visual Basic Programmer's Guide to the WIN32-API"; here used to get
'the addresses of the VB-bytearrays:
'=====
'UPGRADE_NOTE: object was upgraded to object_Renamed. Click for more: 'ms-
help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="A9E4979A-37FA-4718-
9994-97DD76ED70A7"'
'UPGRADE_ISSUE: Declaring a parameter 'As Any' is not supported. Click for
more: 'ms-help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="FAE78A8D-
8978-4FD4-8208-5B7324A8F795"'
Public Declare Function agGetAddressForObject Lib "apigid32.dll" (ByRef
object_Renamed As Byte) As Integer

'=====
'Declarations for the EEPROM-accessing functions in FTD2XX.dll:
'=====
'UPGRADE_WARNING: Structure FT_PROGRAM_DATA may require marshalling attributes
to be passed as an argument in this Declare statement. Click for more: 'ms-
help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="C429C3A5-5D47-4CD9-
8F51-74A1616405DC"'
Public Declare Function FT_EE_Program Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByRef lpData As FT_PROGRAM_DATA) As Integer
'UPGRADE_WARNING: Structure FT_PROGRAM_DATA may require marshalling attributes
to be passed as an argument in this Declare statement. Click for more: 'ms-
help://MS.VSCC.v90/dv_commoner/local/redirect.htm?keyword="C429C3A5-5D47-4CD9-
8F51-74A1616405DC"'
Public Declare Function FT_EE_Read Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByRef lpData As FT_PROGRAM_DATA) As Integer
Public Declare Function FT_EE_UASize Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByRef lpdwSize As Integer) As Integer
Public Declare Function FT_EE_UAWrite Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByVal pucData As String, ByVal dwDataLen As Integer) As Integer
Public Declare Function FT_EE_UARead Lib "FTD2XX.DLL" (ByVal lngHandle As
Integer, ByVal pucData As String, ByVal dwDataLen As Integer, ByRef lpdwBytesRead
As Integer) As Integer

```

```

Public Structure LPSECURITY_ATTRIBUTES
    Dim nLength As Integer
    Dim lpSecurityDescriptor As Integer
    Dim bInheritHandle As Integer
End Structure

Public Structure lpOverlapped
    Dim Internal As Integer
    Dim InternalHigh As Integer
    Dim Offset As Integer
    Dim OffsetHigh As Integer
    Dim hEvent As Integer
End Structure

Public Structure FTDCB
    Dim DCBlength As Integer 'sizeof (FTDCB)
    Dim BaudRate As Integer '9600
    ' fBinary As Long '= 1 Binary mode (skip EOF check)
    ' fParity As Long '= 1 Enable parity checking
    ' fOutxCtsFlow As Long '= 1 CTS handshaking on output
    ' fOutxDsrFlow As Long '= 1 DSR handshaking on output
    ' fDtrControl As Long '= 2 DTR flow control
    ' fDsrSensitivity As Long '= 1 DSR Sensitivity
    ' fTXContinueOnXoff As Long '= 1 Continue TX when Xoff sent
    ' fOutX As Long '= 1 Enable output X-on/X-off
    ' fInX As Long '= 1 Enable input X-on/X-off
    ' fErrorChar As Long '= 1 Enable error replacement
    ' fNull As Long '= 1 Enable null stripping
    ' fRtsControl As Long '= 2 RTS flow control
    ' fAbortOnError As Long '= 1 Abort all reads and writes
on error
    ' fDummy2 As Long '= 17 Reserved
    ' wReserved As Integer 'Not currently used
    ' XonLim As Integer 'Transmit X-on threshold
    ' XoffLim As Integer 'Transmit X-off threshold
    ' ByteSize As Integer 'Number of bits/ byte, 7-8
    ' Parity As Integer '0-4= None, Odd, Even, Mark,
Space
    ' StopBits As Integer '0, 2 = 1, 2
    ' XonChar As Byte 'TX and RX X-ON character
    ' XoffChar As Byte 'TX and RX X-OFF character
    ' ErrorChar As Byte 'Error replacement char
    ' EofChar As Byte 'End of input Character
    ' EvtChar As Byte 'Received event character
    ' wReserved1 As Integer 'BCD (0x0200 => USB2)
End Structure

```

```

'=====
'Type definition as equivalent for C-structure "ft_program_data" used
'in FT_EE_READ and FT_EE_WRITE;
'ATTENTION! The variables "Manufacturer", "ManufacturerID",
'"Description" and "SerialNumber" are passed as POINTERS to
'locations of Bytearrays. Each Byte in these arrays will be
'filled with one character of the whole string.

```

'(See below, calls to "agGetAddressForObject")

'=====

```
Public Structure FT_PROGRAM_DATA
    Dim signature1 As Integer '0x00000000
    Dim signature2 As Integer '0xFFFFFFFF
    Dim version As Integer '0
    Dim VendorId As Short '0x0403
    Dim ProductId As Short '0x6001
    Dim Manufacturer As Integer '32 "FTDI"
    Dim ManufacturerId As Integer '16 "FT"
    Dim Description As Integer '64 "USB HS Serial Converter"
    Dim SerialNumber As Integer '16 "FT000001" if fixed, or NULL
    Dim MaxPower As Short ' // 0 < MaxPower <= 500
    Dim PnP As Short ' // 0 = disabled, 1 = enabled
    Dim SelfPowered As Short ' // 0 = bus powered, 1 = self powered
    Dim RemoteWakeup As Short ' // 0 = not capable, 1 = capable
    'Rev4 extensions:
    Dim Rev4 As Byte ' // true if Rev4 chip, false otherwise
    Dim IsoIn As Byte ' // true if in endpoint is isochronous
    Dim IsoOut As Byte ' // true if out endpoint is isochronous
    Dim PullDownEnable As Byte ' // true if pull down enabled
    Dim SerNumEnable As Byte ' // true if serial number to be used
    Dim USBVersionEnable As Byte ' // true if chip uses USBVersion
    Dim USBVersion As Short ' // BCD (0x0200 => USB2)
    'FT2232C extensions:
    Dim Rev5 As Byte 'non-zero if Rev5 chip, zero otherwise
    Dim IsoInA As Byte 'non-zero if in endpoint is isochronous
    Dim IsoInB As Byte 'non-zero if in endpoint is isochronous
    Dim IsoOutA As Byte 'non-zero if out endpoint is isochronous
    Dim IsoOutB As Byte 'non-zero if out endpoint is isochronous
    Dim PullDownEnable5 As Byte 'non-zero if pull down enabled
    Dim SerNumEnable5 As Byte 'non-zero if serial number to be used
    Dim USBVersionEnable5 As Byte 'non-zero if chip uses USBVersion
    Dim USBVersion5 As Short 'BCD 0x110 = USB 1.1, BCD 0x200 = USB 2.0
    Dim BlsHighCurrent As Byte 'non-zero if interface is high current
    Dim BlsHighCurrent As Byte 'non-zero if interface is high current
    Dim IFAlsFifo As Byte 'non-zero if interface is 245 FIFO
    Dim IFAlsFifoTar As Byte 'non-zero if interface is 245 FIFO CPU target
    Dim IFAlsFastSer As Byte 'non-zero if interface is Fast Serial
    Dim AlsVCP As Byte 'non-zero if interface is to use VCP drivers
    Dim IFBlsFifo As Byte 'non-zero if interface is 245 FIFO
    Dim IFBlsFifoTar As Byte 'non-zero if interface is 245 FIFO CPU target
    Dim IFBlsFastSer As Byte 'non-zero if interface is Fast Serial
    Dim BlsVCP As Byte 'non-zero if interface is to use VCP drivers
    'FT232R extensions
    Dim UseExtOSC As Byte 'non-zero use ext osc
    Dim HighDriveIOs As Byte 'non-zero to use High Drive IO's
    Dim EndPointSize As Byte '64 Do not change
    Dim PullDownEnableR As Byte 'non-zero if pull down enabled
    Dim SerNumEnableR As Byte 'non-zero if pull serial number enabled
    Dim InvertTXD As Byte 'non-zero if invert TXD
    Dim InvertRXD As Byte 'non-zero if invert RXD
    Dim InvertRTS As Byte 'non-zero if invert RTS
    Dim InvertCTS As Byte 'non-zero if invert CTS
    Dim InvertDTR As Byte 'non-zero if invert DTR
    Dim InvertDSR As Byte 'non-zero if invert DSR
```

```
Dim InvertDCD As Byte 'non-zero if invert DCD
Dim InvertRI As Byte 'non-zero if invert RI
Dim Cbus0 As Byte 'Cbus Mux control
Dim Cbus1 As Byte 'Cbus Mux control
Dim Cbus2 As Byte 'Cbus Mux control
Dim Cbus3 As Byte 'Cbus Mux control
Dim Cbus4 As Byte 'Cbus Mux control
Dim RIsVCP As Byte 'non-zero if using VCP driver
End Structure
```

' Return codes

```
Public Const FT_OK As Short = 0
Public Const FT_INVALID_HANDLE As Short = 1
Public Const FT_DEVICE_NOT_FOUND As Short = 2
Public Const FT_DEVICE_NOT_OPENED As Short = 3
Public Const FT_IO_ERROR As Short = 4
Public Const FT_INSUFFICIENT_RESOURCES As Short = 5
Public Const FT_INVALID_PARAMETER As Short = 6
Public Const FT_INVALID_BAUD_RATE As Short = 7
Public Const FT_DEVICE_NOT_OPENED_FOR_ERASE As Short = 8
Public Const FT_DEVICE_NOT_OPENED_FOR_WRITE As Short = 9
Public Const FT_FAILED_TO_WRITE_DEVICE As Short = 10
Public Const FT_EEPROM_READ_FAILED As Short = 11
Public Const FT_EEPROM_WRITE_FAILED As Short = 12
Public Const FT_EEPROM_ERASE_FAILED As Short = 13
Public Const FT_EEPROM_NOT_PRESENT As Short = 14
Public Const FT_EEPROM_NOT_PROGRAMMED As Short = 15
Public Const FT_INVALID_ARGS As Short = 16
Public Const FT_NOT_SUPPORTED As Short = 17
Public Const FT_OTHER_ERROR As Short = 18
```

' Word Lengths

```
Public Const FT_BITS_8 As Short = 8
Public Const FT_BITS_7 As Short = 7
```

' Stop Bits

```
Public Const FT_STOP_BITS_1 As Short = 0
Public Const FT_STOP_BITS_1_5 As Short = 1
Public Const FT_STOP_BITS_2 As Short = 2
```

' Parity

```
Public Const FT_PARITY_NONE As Short = 0
Public Const FT_PARITY_ODD As Short = 1
Public Const FT_PARITY_EVEN As Short = 2
Public Const FT_PARITY_MARK As Short = 3
Public Const FT_PARITY_SPACE As Short = 4
```

' Flow Control

```
Public Const FT_FLOW_NONE As Integer = &H0
Public Const FT_FLOW_RTS_CTS As Integer = &H100
Public Const FT_FLOW_DTR_DSR As Integer = &H200
Public Const FT_FLOW_XON_XOFF As Integer = &H400
```

' Purge rx and tx buffers

```
Public Const FT_PURGE_RX As Short = 1
Public Const FT_PURGE_TX As Short = 2
```

```

' Modem Status
Public Const FT_MODEM_STATUS_CTS As Integer = &H10
Public Const FT_MODEM_STATUS_DSR As Integer = &H20
Public Const FT_MODEM_STATUS_RI As Integer = &H40
Public Const FT_MODEM_STATUS_DCD As Integer = &H80

Public Const FT_EVENT_RXCHAR As Integer = 1
Public Const FT_EVENT_MODEM_STATUS As Short = 2

Const WAIT_ABANDONED As Integer = &H80
Const WAIT_FAILED As Integer = &HFFFFFFF
Const WAIT_OBJECT_0 As Integer = &H0
Const WAIT_TIMEOUT As Integer = &H102

' Flags for FT_ListDevices
Public Const FT_LIST_BY_NUMBER_ONLY As Integer = &H80000000
Public Const FT_LIST_BY_INDEX As Integer = &H40000000
Public Const FT_LIST_ALL As Integer = &H20000000

' Flags for FT_OpenEx
Public Const FT_OPEN_BY_SERIAL_NUMBER As Short = 1
Public Const FT_OPEN_BY_DESCRIPTION As Short = 2

Private Const INFINITE As Integer = 1000 '&HFFFFFFF

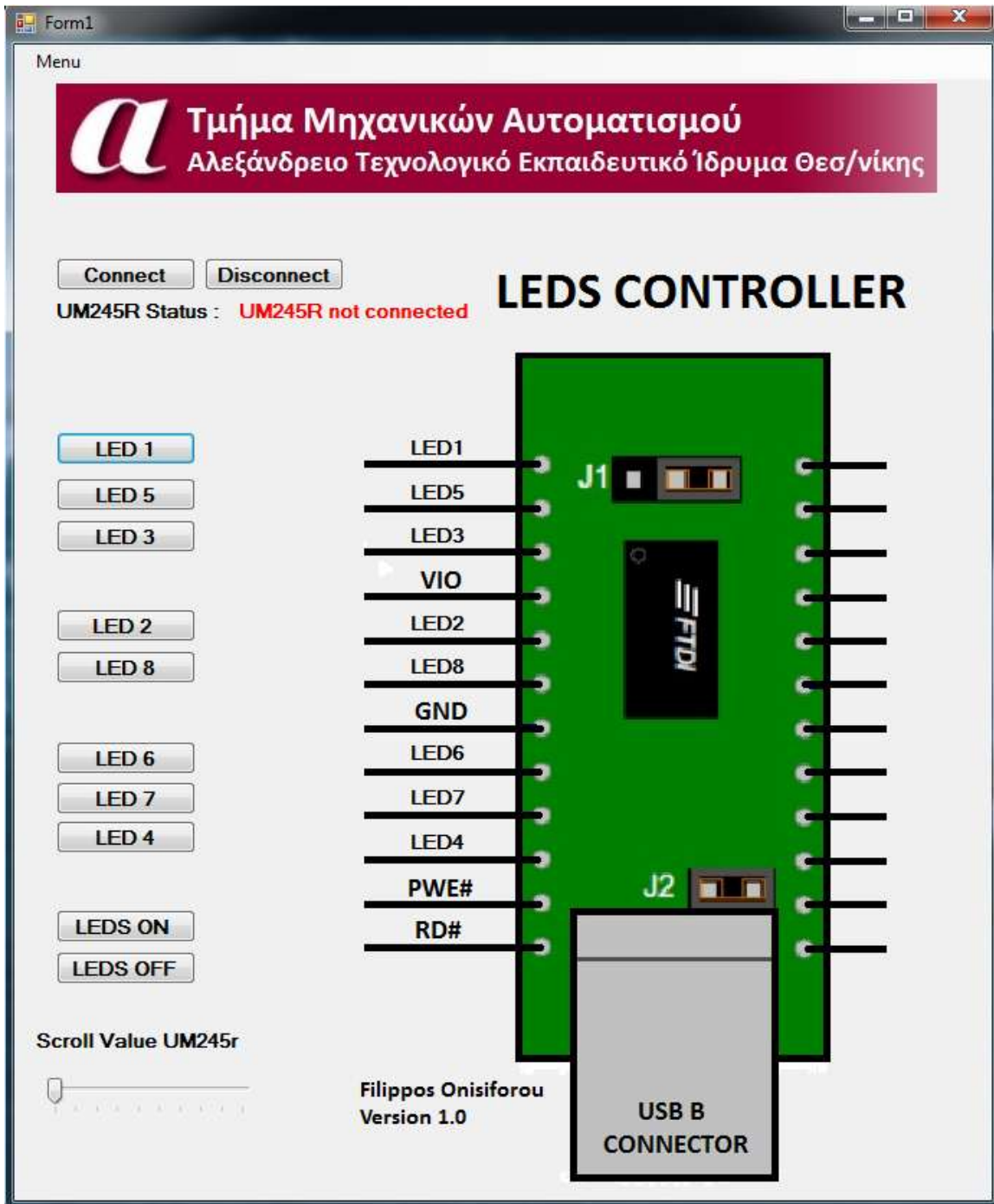
Public hThread As Integer
Public hThreadID As Integer
Public hEvent As Integer
Public EventMask As Integer

Public lngHandle As Integer

'Sub Main()
'   Set fMainForm = New DEMO_EEPROM
'   fMainForm.Show
'End Sub
End Module

```

**ΠΑΡΑΡΤΗΜΑ ΙΙ: Πηγαίος Κώδικας ελέγχου θυρών
του UM245R ως έξοδοι**



Πηγαίος Κώδικας:

```

Public Class Form1

    Const DEVICE_NO As Short = 0
    Dim m_DeviceHandle As Integer
    Dim m_Count As Short
    Dim ret As Integer

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button9.Click

        If FT_Open(DEVICE_NO, m_DeviceHandle) <> FT_OK Then
            Label9.Text = "UM245R not connected"
            Exit Sub
        Else
            Label9.Text = "UM245R connected"
        End If

        If Label9.Text = "UM245R connected" Then
            Label9.ForeColor = Color.Green
        End If

        If Label9.Text = "UM245R not connected" Then
            Label9.ForeColor = Color.Red
        End If

        If FT_SetBitMode(m_DeviceHandle, &HFF, &H1) <> FT_OK Then
            Label9.Text = "ERROR"
            Label9.ForeColor = Color.Red
            Exit Sub
        End If

        FT_SetBaudRate(m_DeviceHandle, 57600)

    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Dim data As Byte

        If FT_SetBitMode(m_DeviceHandle, &H0, &H1) <> FT_OK Then
            Label9.Text = "UM245R not connected"
            Label9.ForeColor = Color.Red
            Exit Sub
        End If

        ret = FT_GetBitMode(m_DeviceHandle, data)
        ret = FT_SetBitMode(m_DeviceHandle, &HFF, &H1)

    End Sub

    Private Sub button1_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button1.MouseDown
        ret = FT_WriteByte(m_DeviceHandle, 1, 1, 0)
    End Sub

```



```
Timer1.Enabled = False
Label11.BackColor = Color.Red

Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

End Sub

```
Private Sub button1_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button1.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label11.BackColor = Color.Empty

    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
```

End Sub

```
Private Sub button2_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button2.MouseDown
    ret = FT_WriteByte(m_DeviceHandle, 2, 1, 0)
    Timer1.Enabled = False
    Label12.BackColor = Color.Red

    Label11.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
```

End Sub

```
Private Sub button2_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button2.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label12.BackColor = Color.Empty
```

```

Label11.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

Private Sub button3_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button3.MouseDown
    ret = FT_WriteByte(m_DeviceHandle, 4, 1, 0)
    Timer1.Enabled = False
    Label13.BackColor = Color.Red

    Label12.BackColor = Color.Empty
    Label11.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
End Sub

Private Sub button3_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button3.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label13.BackColor = Color.Empty

    Label12.BackColor = Color.Empty
    Label11.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
End Sub

Private Sub button4_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button4.MouseDown
    ret = FT_WriteByte(m_DeviceHandle, 8, 1, 0)
    Timer1.Enabled = False
    Label14.BackColor = Color.Red

    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label11.BackColor = Color.Empty
    Label15.BackColor = Color.Empty

```

```

Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
End Sub

Private Sub Label14_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button4.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label14.BackColor = Color.Empty

    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label11.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
End Sub

Private Sub button5_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button5.MouseDown
    ret = FT_WriteByte(m_DeviceHandle, 16, 1, 0)
    Timer1.Enabled = False
    Label15.BackColor = Color.Red

    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label11.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
End Sub

Private Sub button5_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button5.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label15.BackColor = Color.Empty

    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label11.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

```

```
        Button11.BackColor = Color.Empty
        Button12.BackColor = Color.Empty
    End Sub
```

```
    Private Sub button6_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button6.MouseDown
        ret = FT_WriteByte(m_DeviceHandle, 32, 1, 0)
        Timer1.Enabled = False
        Label6.BackColor = Color.Red

        Label12.BackColor = Color.Empty
        Label13.BackColor = Color.Empty
        Label14.BackColor = Color.Empty
        Label15.BackColor = Color.Empty
        Label11.BackColor = Color.Empty
        Label17.BackColor = Color.Empty
        Label18.BackColor = Color.Empty

        Button11.BackColor = Color.Empty
        Button12.BackColor = Color.Empty
    End Sub
```

```
    Private Sub button6_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button6.MouseUp
        ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
        Timer1.Enabled = True
        Label6.BackColor = Color.Empty

        Label12.BackColor = Color.Empty
        Label13.BackColor = Color.Empty
        Label14.BackColor = Color.Empty
        Label15.BackColor = Color.Empty
        Label11.BackColor = Color.Empty
        Label17.BackColor = Color.Empty
        Label18.BackColor = Color.Empty

        Button11.BackColor = Color.Empty
        Button12.BackColor = Color.Empty
    End Sub
```

```
    Private Sub button7_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button7.MouseDown
        ret = FT_WriteByte(m_DeviceHandle, 64, 1, 0)
        Timer1.Enabled = False
        Label17.BackColor = Color.Red

        Label12.BackColor = Color.Empty
        Label13.BackColor = Color.Empty
        Label14.BackColor = Color.Empty
        Label15.BackColor = Color.Empty
        Label16.BackColor = Color.Empty
        Label11.BackColor = Color.Empty
        Label18.BackColor = Color.Empty

        Button11.BackColor = Color.Empty
        Button12.BackColor = Color.Empty
    End Sub
```

```
Private Sub button7_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button7.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label17.BackColor = Color.Empty

    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label11.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
End Sub
```

```
Private Sub button8_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button8.MouseDown
    ret = FT_WriteByte(m_DeviceHandle, 128, 1, 0)
    Timer1.Enabled = False
    Label18.BackColor = Color.Red

    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label11.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
End Sub
```

```
Private Sub button8_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Button8.MouseUp
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
    Timer1.Enabled = True
    Label18.BackColor = Color.Empty

    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label11.BackColor = Color.Empty

    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
End Sub
```

```
Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click

    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
```

```

Timer1.Enabled = True

If FT_WriteByte(m_DeviceHandle, 0, 1, 0) Then
    TrackBar1.Value = 0
End If

Label11.BackColor = Color.Empty
Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button12.BackColor = Color.Empty
Button11.BackColor = Color.Empty

ret = FT_Close(m_DeviceHandle)
Label19.Text = "UM245R Disconnected ! "
Label19.ForeColor = Color.Red

End Sub

Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ExitToolStripMenuItem.Click
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)

    Timer1.Enabled = True

    Label11.BackColor = Color.Empty
    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty

    Close()

End Sub

Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button12.Click

    ret = FT_WriteByte(m_DeviceHandle, 255, 1, 0)
    Button12.BackColor = Color.Green
    Button11.BackColor = Color.Empty

    Timer1.Enabled = False

    Label11.BackColor = Color.Red
    Label12.BackColor = Color.Red

```

```
Label13.BackColor = Color.Red
Label14.BackColor = Color.Red
Label15.BackColor = Color.Red
Label16.BackColor = Color.Red
Label17.BackColor = Color.Red
Label18.BackColor = Color.Red
```

End Sub

```
Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button11.Click
    Button12.BackColor = Color.Empty
    Button11.BackColor = Color.Red
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
```

```
Timer1.Enabled = True
```

```
Label11.BackColor = Color.Empty
Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty
```

End Sub

```
Private Sub TrackBar1_Scroll(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TrackBar1.Scroll
```

```
If TrackBar1.Value = 0 Then
```

```
    ret = FT_WriteByte(m_DeviceHandle, 0, 1, 0)
```

```
    Label11.BackColor = Color.Empty
    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
    Label17.BackColor = Color.Empty
    Label18.BackColor = Color.Empty
```

```
    Button11.BackColor = Color.Empty
    Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 1 Then
```

```
    ret = FT_WriteByte(m_DeviceHandle, 1, 1, 0)
```

```
    Label11.BackColor = Color.Red
```

```
    Label12.BackColor = Color.Empty
    Label13.BackColor = Color.Empty
    Label14.BackColor = Color.Empty
    Label15.BackColor = Color.Empty
    Label16.BackColor = Color.Empty
```

```
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 2 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 2, 1, 0)

Label11.BackColor = Color.Empty

Label12.BackColor = Color.Red

Label13.BackColor = Color.Empty
Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 3 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 4, 1, 0)

Label11.BackColor = Color.Empty
Label12.BackColor = Color.Empty

Label13.BackColor = Color.Red

Label14.BackColor = Color.Empty
Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 4 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 8, 1, 0)

Label11.BackColor = Color.Empty
Label12.BackColor = Color.Empty
Label13.BackColor = Color.Empty

Label14.BackColor = Color.Red

Label15.BackColor = Color.Empty
Label16.BackColor = Color.Empty
Label17.BackColor = Color.Empty
Label18.BackColor = Color.Empty

Button11.BackColor = Color.Empty
```



```
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 5 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 16, 1, 0)
```

```
Label11.BackColor = Color.Empty
```

```
Label12.BackColor = Color.Empty
```

```
Label13.BackColor = Color.Empty
```

```
Label14.BackColor = Color.Empty
```

```
Label15.BackColor = Color.Red
```

```
Label16.BackColor = Color.Empty
```

```
Label17.BackColor = Color.Empty
```

```
Label18.BackColor = Color.Empty
```

```
Button11.BackColor = Color.Empty
```

```
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 6 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 32, 1, 0)
```

```
Label11.BackColor = Color.Empty
```

```
Label12.BackColor = Color.Empty
```

```
Label13.BackColor = Color.Empty
```

```
Label14.BackColor = Color.Empty
```

```
Label15.BackColor = Color.Empty
```

```
Label16.BackColor = Color.Red
```

```
Label17.BackColor = Color.Empty
```

```
Label18.BackColor = Color.Empty
```

```
Button11.BackColor = Color.Empty
```

```
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 7 Then
```

```
ret = FT_WriteByte(m_DeviceHandle, 64, 1, 0)
```

```
Label11.BackColor = Color.Empty
```

```
Label12.BackColor = Color.Empty
```

```
Label13.BackColor = Color.Empty
```

```
Label14.BackColor = Color.Empty
```

```
Label15.BackColor = Color.Empty
```

```
Label16.BackColor = Color.Empty
```

```
Label17.BackColor = Color.Red
```

```
Label18.BackColor = Color.Empty
```

```
Button11.BackColor = Color.Empty
```

```
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 8 Then
```

```
net = FT_WriteByte(m_DeviceHandle, 128, 1, 0)
```

```
Label11.BackColor = Color.Empty  
Label12.BackColor = Color.Empty  
Label13.BackColor = Color.Empty  
Label14.BackColor = Color.Empty  
Label15.BackColor = Color.Empty  
Label16.BackColor = Color.Empty  
Label17.BackColor = Color.Empty
```

```
Label18.BackColor = Color.Red
```

```
Button11.BackColor = Color.Empty  
Button12.BackColor = Color.Empty
```

```
ElseIf TrackBar1.Value = 9 Then
```

```
net = FT_WriteByte(m_DeviceHandle, 255, 1, 0)
```

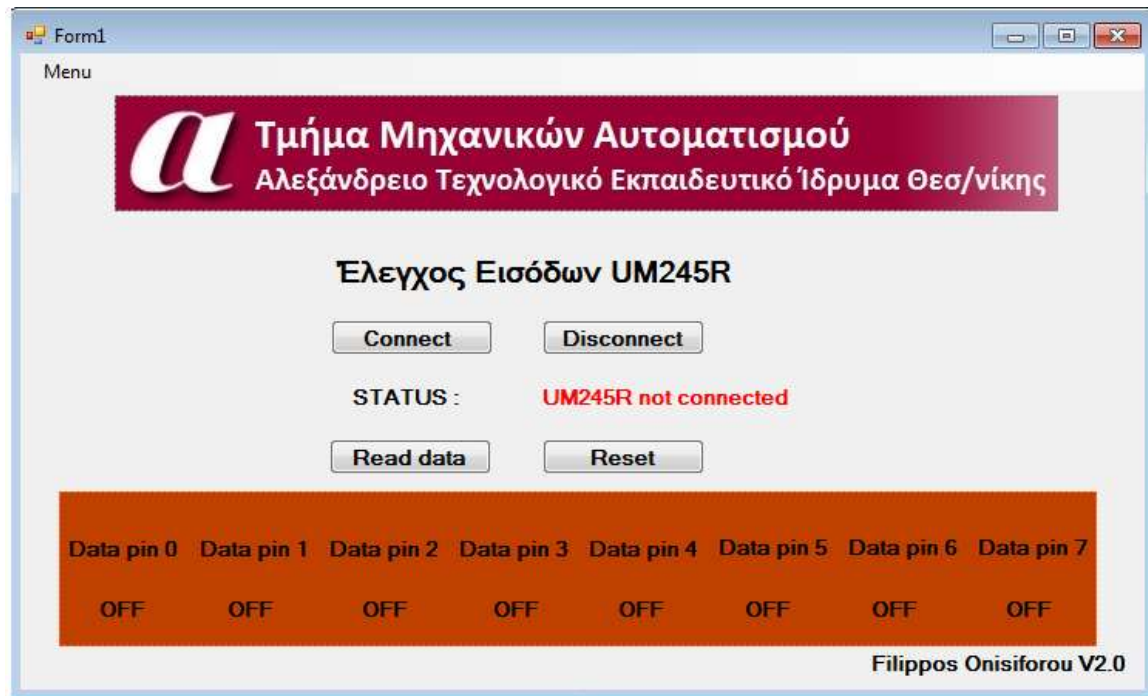
```
Label11.BackColor = Color.Red  
Label12.BackColor = Color.Red  
Label13.BackColor = Color.Red  
Label14.BackColor = Color.Red  
Label15.BackColor = Color.Red  
Label16.BackColor = Color.Red  
Label17.BackColor = Color.Red  
Label18.BackColor = Color.Red
```

```
Button11.BackColor = Color.Empty  
Button12.BackColor = Color.Empty
```

```
End If
```

```
End Sub  
End Class
```

ΠΑΡΑΡΤΗΜΑ ΙΙΙ: Πηγαίος Κώδικας ελέγχου θυρών του UM245R ως είσοδοι



Πηγαίος Κώδικας:

```
Public Class Form1
```

```
    Const DEVICE_NO As Short = 0  
    Dim m_DeviceHandle As Integer  
    Dim m_Count As Short  
    Dim ret As Integer
```

```
    Private Declare Function GetAsyncKeyState Lib "user32" (ByVal vkey As Long) As Short
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
        If FT_Open(DEVICE_NO, m_DeviceHandle) <> FT_OK Then  
            Label11.Text = "UM245R not connected"  
            Exit Sub  
        Else
```

```

        Label11.Text = "UM245R connected"
    End If

    If Label11.Text = "UM245R connected" Then
        Label11.ForeColor = Color.Green
    End If

    If Label11.Text = "UM245R not connected" Then
        Label11.ForeColor = Color.Red
    End If

    If FT_SetBitMode(m_DeviceHandle, &HFF, &H1) <> FT_OK Then
        Label11.Text = "ERROR"
        Label11.ForeColor = Color.Red
        Exit Sub
    End If

    FT_SetBaudRate(m_DeviceHandle, 57600)

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    'Data Read Function

    If FT_SetBitMode(m_DeviceHandle, &H0, &H1) <> FT_OK Then
        Label11.Text = "UM245R not connected"
        Label11.ForeColor = Color.Red

    End If

    If FT_SetBitMode(m_DeviceHandle, &H0, &H1) <> FT_OK Then

        Label11.BackColor = Color.Empty
        Label12.BackColor = Color.Empty
        Label13.BackColor = Color.Empty
        Label14.BackColor = Color.Empty
        Label15.BackColor = Color.Empty
        Label16.BackColor = Color.Empty
        Label17.BackColor = Color.Empty
        Label18.BackColor = Color.Empty

        Exit Sub

    End If

    'Label14.Text = CStr(data) ' pokud bude třeba vytvoř label.4
    ret = FT_SetBitMode(m_DeviceHandle, &HFF, &H1)

```

```

End Sub

Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ExitToolStripMenuItem.Click
    Close()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Timer1.Enabled = False

    Label11.Text = "OFF "
    Label11.ForeColor = Color.Black
    Label12.Text = "OFF "
    Label12.ForeColor = Color.Black
    Label13.Text = "OFF "
    Label13.ForeColor = Color.Black
    Label14.Text = "OFF "
    Label14.ForeColor = Color.Black
    Label15.Text = "OFF "
    Label15.ForeColor = Color.Black
    Label16.Text = "OFF "
    Label16.ForeColor = Color.Black
    Label17.Text = "OFF "
    Label17.ForeColor = Color.Black
    Label18.Text = "OFF "
    Label18.ForeColor = Color.Black

    Dim ret As Integer
    ret = FT_Close(m_DeviceHandle)
    Label111.Text = " UM245r disconnected "
    Label111.ForeColor = Color.Red

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Timer2.Enabled = True

End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Timer2.Enabled = False
    Label11.Text = "OFF "
    Label11.ForeColor = Color.Black
    Label12.Text = "OFF "
    Label12.ForeColor = Color.Black
    Label13.Text = "OFF "
    Label13.ForeColor = Color.Black
    Label14.Text = "OFF "
    Label14.ForeColor = Color.Black
    Label15.Text = "OFF "
    Label15.ForeColor = Color.Black
    Label16.Text = "OFF "
    Label16.ForeColor = Color.Black
    Label17.Text = "OFF "
    Label17.ForeColor = Color.Black

```

```
Label8.Text = "OFF "  
Label8.ForeColor = Color.Black
```

```
End Sub
```

```
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Timer2.Tick
```

```
Dim ret As Integer
```

```
Dim data As Byte
```

```
Dim data1 As Byte
```

```
Dim data2 As Byte
```

```
Dim data3 As Byte
```

```
Dim data4 As Byte
```

```
Dim data5 As Byte
```

```
Dim data6 As Byte
```

```
Dim data7 As Byte
```

```
Dim data8 As Byte
```

```
ret = FT_GetBitMode(m_DeviceHandle, data)
```

```
ret = FT_SetBitMode(m_DeviceHandle, &HFF, &H1)
```

```
If CStr(data) = 1 Then
```

```
Label11.Text = " ON "
```

```
Label11.ForeColor = Color.Red
```

```
Else
```

```
Label11.Text = "OFF "
```

```
Label11.ForeColor = Color.Black
```

```
End If
```

```
If CStr(data) = 2 Then
```

```
Label12.Text = " ON "
```

```
Label12.ForeColor = Color.Red
```

```
Else
```

```
Label12.Text = "OFF "
```

```
Label12.ForeColor = Color.Black
```

```
End If
```

```
If CStr(data) = 4 Then
```

```
Label13.Text = " ON "
```

```
Label13.ForeColor = Color.Red
```

```
Else
```

```
Label13.Text = "OFF "
```

```
Label13.ForeColor = Color.Black
```

```
End If
```

```
If CStr(data) = 8 Then

    Label14.Text = " ON "
    Label14.ForeColor = Color.Red
Else

    Label14.Text = "OFF "
    Label14.ForeColor = Color.Black

End If
If CStr(data) =  Then
    Label15.Text = " ON "
    Label15.ForeColor = Color.Red

Else

    Label15.Text = "OFF "
    Label15.ForeColor = Color.Black

End If

If CStr(data) = 32 Then

    Label16.Text = " ON "
    Label16.ForeColor = Color.Red
Else

    Label16.Text = "OFF "
    Label16.ForeColor = Color.Black

End If
If CStr(data) = 64 Then
    Label17.Text = " ON "
    Label17.ForeColor = Color.Red

Else

    Label17.Text = "OFF "
    Label17.ForeColor = Color.Black

End If

If CStr(data) = 128 Then

    Label18.Text = " ON "
    Label18.ForeColor = Color.Red
Else

    Label18.Text = "OFF "
    Label18.ForeColor = Color.Black

End If

End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
End Sub  
End Class
```