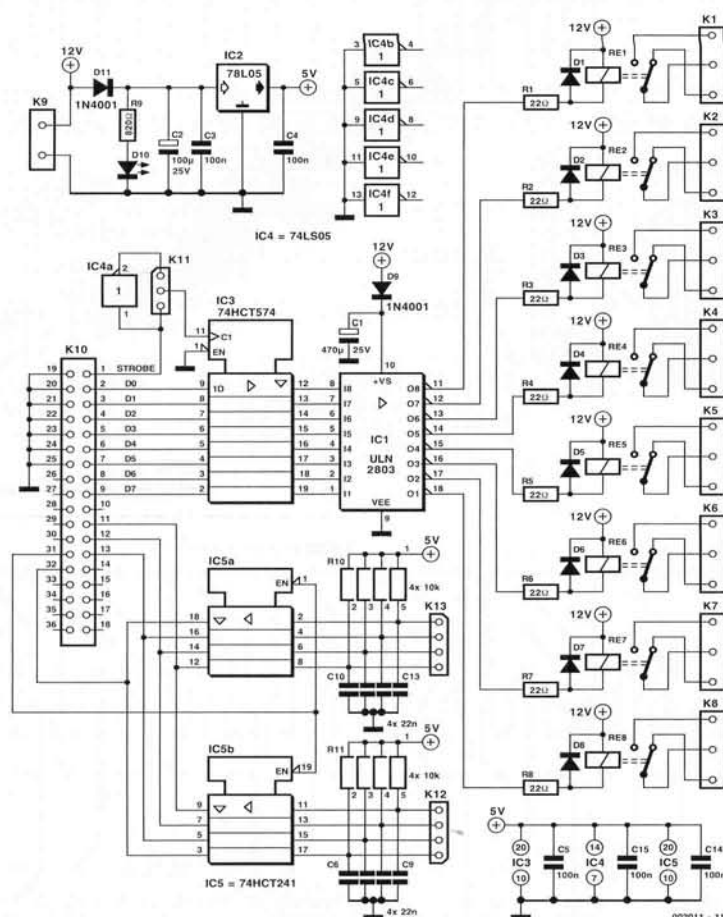


Σύστημα Παραλλήλων Εισόδων / Εξόδων Γενικής Χρήσης για PC

Σχεδίαση εφαρμογής (Hardware) από τον Δημητριάδη Δ. Άγγελο

Μια απλή εφαρμογή που περιλαμβάνει ένα πρόγραμμα για Windows το οποίο επιτρέπει τον έλεγχο μέχρι και οκτώ ηλεκτρονόμων (relays) ή ακόμα και άλλου τύπου μηχανισμών, καθώς επίσης και την ανάγνωση από έναν ίδιο αριθμό εισόδων, εξολοκλήρου μέσα από την παράλληλη θύρα του υπολογιστή.



Σχήμα 1. Το σχηματικό διάγραμμα του UIPO.

Το λογισμικό και το υλικό της εφαρμογής UIPO - **U**niversal **P**arallel **I**nput/**O**utput - (ένα σύστημα που συνδέεται στην παράλληλη θύρα του PC) επιτρέπει τον απευθείας έλεγχο μέχρι οκτώ εξόδων τύπου ηλεκτρονόμου χρησιμοποιώντας ένα παραθυρικό περιβάλλον (Windows-style) το οποίο σχεδιάστηκε με σκοπό την ευκολία στη χρήση. Η ίδια εφαρμογή (UIPO) επιτρέπει την ανάγνωση των λογικών

καταστάσεων που εμφανίζονται στις γραμμές εισόδου.

Περιγραφή του κυκλώματος

Το UIPO είναι αρκετά απλό και αποτελείται από ελάχιστα εξαρτήματα χαμηλού κόστους που μπορείτε εύκολα να προμηθευτείτε. Το σχηματικό διάγραμμα του κυκλώματος φαίνεται στο **Σχήμα 1**. Ένας ολοκληρωμένος απο-

μονωτής τριών καταστάσεων (tri-state buffer) τύπου 74HCT241 (IC5), είναι υπεύθυνος για την ανάγνωση όλων των επαφών εισόδου. Επίσης, ένα ολοκληρωμένο κύκλωμα μανδάλωσης (latch) 8-bit τύπου 74HCT574 (IC3), χρησιμοποιείται για τη συγκράτηση των καταστάσεων εξόδου κατά τη διάρκεια εκτέλεσης των εντολών. Το τρίτο στη σειρά βασικό εξάρτημα της κατασκευής, είναι ένας οδηγός ισχύος τύπου ULN2803 (IC1), με τον οποίον οδηγούνται οι εξοδοί που ενεργοποιούν και απενεργοποιούν τους ηλεκτρονόμους Re1-Re8.

Επειδή η είσοδος της παράλληλης θύρας (LPT) του υπολογιστή αποτελείται μόνο από τέσσερα bits (ένα nibble), απαιτούνται δύο διαδοχικές διαδικασίες ανάγνωσης από το HCT241 έτσι ώστε να ληφθεί ολόκληρη η λέξη (μήκους 8-bit) που αντιστοιχεί στις λογικές καταστάσεις των εισόδων. Η τροφοδοσία λαμβάνεται από εξωτερική πηγή (adaptor) η οποία θα πρέπει να παρέχει περίπου 12 Volts DC. Ένας ολοκληρωμένος σταθεροποιητής τύπου 78L05 (IC2), παρέχει μια σταθεροποιημένη τάση 5-Volts για ολοκληρωμένα τύπου HCT και LS. Παρατηρήστε ότι το ολοκληρωμένο ULN2803 καθώς και οι ηλεκτρονόμοι, τροφοδοτούνται απευθείας από την μη σταθεροποιημένη γραμμή των 12-Volts. Προφανώς, η εξωτερική πηγή τροφοδοσίας (adaptor) πρέπει να είναι σε θέση να παρέχει ικανοποιητικό ρεύμα (υπάρχει περίπτωση να ενεργοποιούνται ταυτόχρονα όλοι οι ηλεκτρονόμοι!).

Η πλακέτα

Το τυπωμένο κύκλωμα της κατασκευής, σχεδιάστηκε από την αρχή, με βάση το πρωτότυπο σχέδιο της εργασίας του σχεδιαστή της εφαρμογής, έτσι ώστε να συμφωνεί με τις προδιαγραφές του περιοδικού Ελέκτορ. Το

τελικό αποτέλεσμα φαίνεται στο **Σχήμα 2**. Η έτοιμη πλακέτα διατίθεται από το τμήμα εξυπηρέτησης αναγνωστών του περιοδικού (κωδικός **002011-1**).

Στην πλακέτα περιλαμβάνεται και ένας συνδετήρας 36 επαφών (Centronics) πλάγιας στήριξης, με σκοπό να διευκολύνει τη σύνδεση της κατασκευής με την παράλληλη θύρα του υπολογιστή. Στην πλευρά των 'εξόδων' της πλακέτας τοποθετούνται τριπολικές κλέμες (που συνδέονται απευθείας με τις επαφές των ηλεκτρονόμενων), οι οποίες επιτρέπουν στον χρήστη να επιλέξει οποιοδήποτε συνδυασμό επαφών κανονικά κλειστών (N.C. ή *Normally Closed*) ή κανονικά ανοικτών (N.O. ή *Normally Open*). Οι λογικές (ψηφιακές) γραμμές εισόδου χωρίζονται σε δύο απλούς συνδετήρες 4 επαφών, τους K12 και K13.

Κατάλογος υλικών

Αντιστάσεις

R1-R8 = 22Ω

R9 = 820Ω R10,

R11 = Δικτύωμα αντιστάσεων SIL

4x10KΩ

Πυκνωτές

C1 = 470μF/25V κάθετος

C2 = 100μF/ 25V κάθετος

C3, C4, C5, C14, C15 = 100nF

C6-C13 = 22nF

Ημιαγωγοί

D1-D8 = 1N4148

D9, D11 = 1N4001

D10 = LED

IC1 = ULN2803

IC2 = 78L05

IC3 = 74HCT574

IC4 = 74LS05

IC5 = 74HCT241

Διάφορα.

K1.....K9 = κλέμες

RE1..RE8 = ρελέδες 12V

K9 = 2-πολικός τερματικός ακροδέκτης για πλακέτα με απόσταση ακροδεκτών 5mm

K10 = Παράλληλος συνδετήρας-κτυπωτή 36 επαφών (Centronics) υπό γωνία

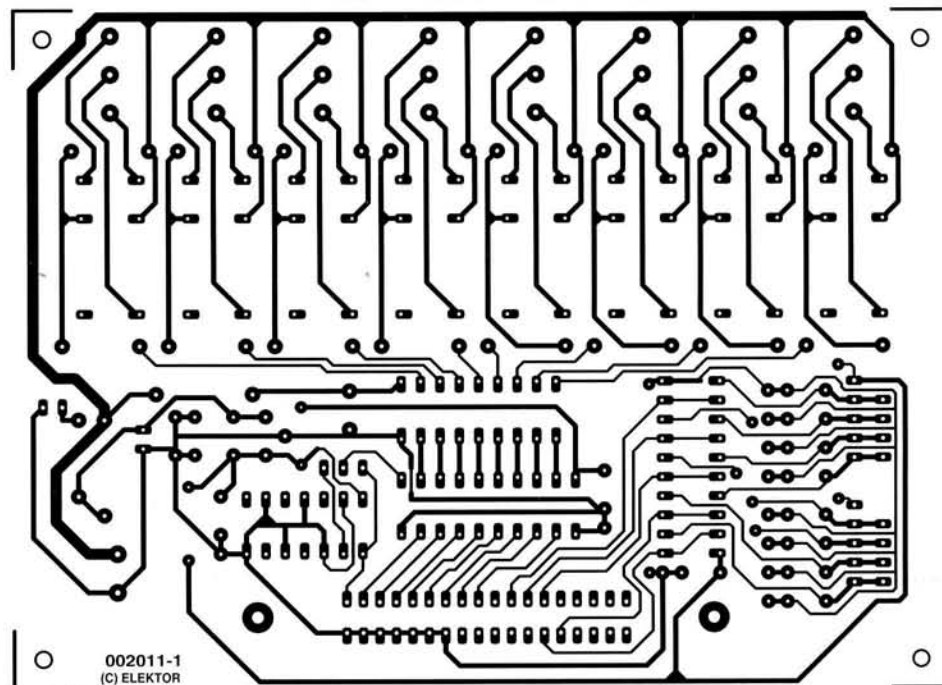
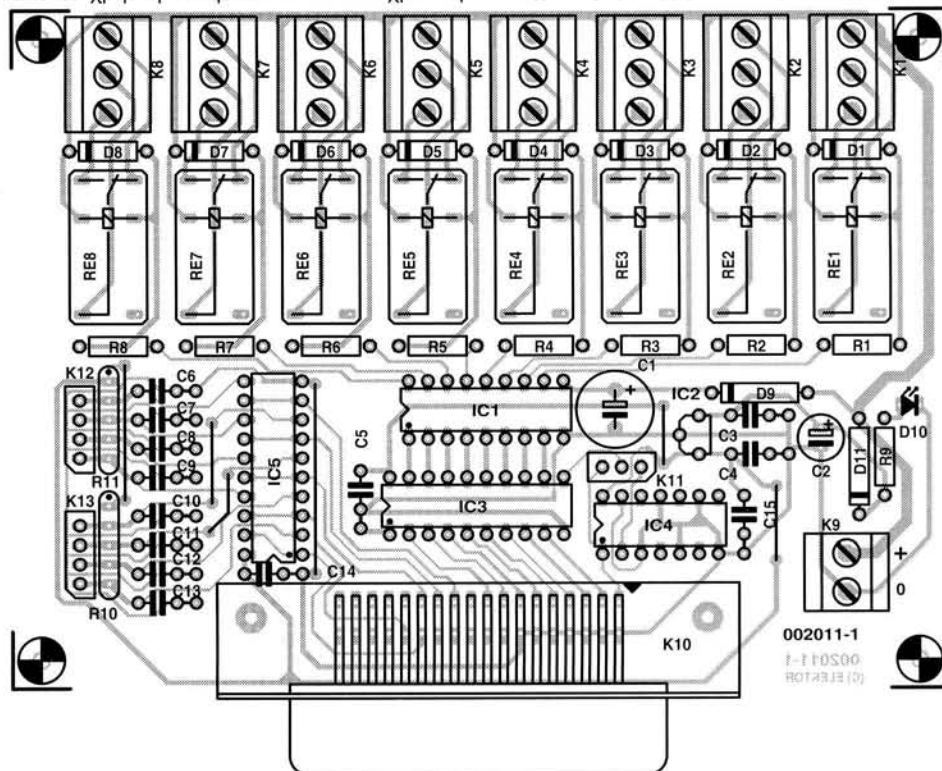
K11= 3-πολικός βραχυκυκλωτήρας
K12,K13 = 4-πολικός βραχυκυκλωτήρας

Πλακέτα με κωδικό 002011-1

Το πρωτότυπο πρόγραμμα της εφαρμογής

Αρχικά το πρόγραμμα ελέγχου που παρουσιάστηκε από το σχεδιαστή της εφαρμογής αυτής, ήταν σχετικά απλό. Με τη βοήθεια ενός αρχείου κειμένου (script), το οποίο παίζει το ρόλο ενός τύπου 'προγράμματος' και που δημιουργείται από το περιβάλλον της ίδιας της εφαρμογής, το σύστημα μπορεί να λειτουργήσει με τη λογική μιας ροής διακριτών λογικών καταστάσεων εξόδου (Output Flow), ενδιάμεσα στις οποίες υπάρχει η δυνατότητα από το χρήστη να προσθέσει κάποια χρονική

καυστήρηση, δημιουργώντας με το τρόπο αυτό ένα συγκεκριμένο χρονοδιάγραμμα για την εκάστοτε εφαρμογή. Επιπλέον υπάρχει και η δυνατότητα συνδυασμού των καταστάσεων εξόδου με αντίστοιχες καταστάσεις που λαμβάνονται από τις οκτώ εισόδους τύπου-επαφής, που διαθέτει συνολικά το σύστημα. Ο συνδυασμός αυτών των δυνατοτήτων, επιτρέπει στο συγκεκριμένο σύστημα να λειτουργεί όχι μόνο σε μια 'μηχανή εναλλαγής διακριτών καταστάσεων', αλλά ταυτόχρονα να συνδυάζει και λειτουργία συγκράτησης (Hold) ή στιγμιαίας δράσης (One-Shot), λειτουργίες που



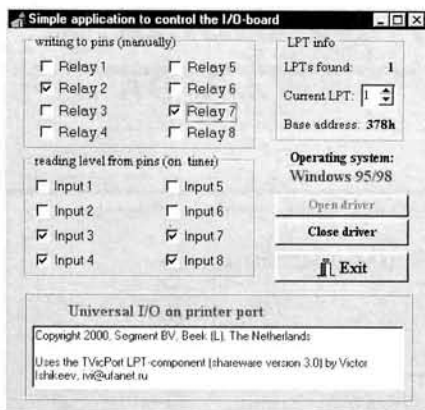
Σχήμα 2. Το τυπωμένο κύκλωμα της κατασκευής του UPIO (Η έτοιμη πλακέτα διατίθεται από τα γραφεία του περιοδικού).

είναι σε μεγάλο βαθμό καθιερωμένες στα συστήματα ελέγχου.

Επικοινωνία με τις θύρες του υπολογιστή κάτω από το περιβάλλον Windows 95/98/NT

Αφού ασχοληθήκαμε για ένα χρονικό διάστημα με το πρωτότυπο πρόγραμμα της εφαρμογής, αντιληφθήκαμε ότι αυτή η αρχική του μορφή δεν είχε επεξεργαστεί όσο έπρεπε και αποδείχθηκε κάπως ασταθής. Έτσι ο συνεργάτης του περιοδικού Luc Lemmens ασχολήθηκε με το να δημιουργήσει μια βελτιωμένη έκδοση του προγράμματος ελέγχου για το κύκλωμα UPIO. Ας ρίξουμε μια ματιά στα κύρια σημεία.

Κάτω από το περιβάλλον του DOS και των Windows 3.1, ήταν σχετικά εύκολος ο έλεγχος του υλικού μέρους (hardware) του υπολογιστή, με τη βοήθεια κάποιων γλωσσών προγραμματισμού υψηλού επιπέδου, όπως η BASIC ή η Pascal. Τότε, εκείνο που χρειαζόταν μόνο ήταν η συνάρτηση IN/OUT της BASIC ή η αντίστοιχη συνάρτηση PORT της Pascal.



Σχήμα 3. Η οθόνη που εμφανίζεται κατά τη λειτουργία του προγράμματος ελέγχου του UPIO.

Σε νεότερες εκδόσεις του γραφικού περιβάλλοντος (GUI) των Windows (από την έκδοση των Windows 95 και μετά), ο χρήστης απομονώνεται ολοένα και περισσότερο από το υλικό (hardware), ενώ δεν είναι πλέον εφικτή (και μάλιστα απαγορευμένη στα Windows 95) η άμεση επικοινωνία με τους διάφορους καταχωρητές και τις θέσεις μνήμης. Το είδος αυτό του 'μικρού διαλόγου' εξαρτάται αποκλειστικά από το περιβάλλον του λειτουργικού συστήματος σε συνεργασία με διάφορα προγράμματα οδηγών (device drivers). Η συγγραφή δικών σας οδηγών θα λέγαμε τουλάχιστον ότι είναι αρκετά επίπονη, έτσι είναι σχεδόν πάντα προτιμότερο να επισκεφθείτε τις σελίδες του Internet για να δείτε μήπως κάποιος άλλος έχει ανακαλύψει το τροχό για σας! Έτσι και εμείς στη περίπτωση αυτή, βρε-

θήκαμε στη διεύθυνση www.entechtaiwan.com όπου βρήκαμε ένα στοιχείο βιβλιοθήκης (shareware) για οδηγό της παράλληλης θύρας (LPT Component) με την ονομασία TVicPort, το οποίο γράφτηκε από κάποιον victor Ishikeev. Αυτό είναι διαθέσιμο σε διάφορες εκδόσεις για τη γλώσσα Delphi (εκδόσεις 2 ως 5), την Java Builder (εκδόσεις 1, 3 και 4), την Visual BASIC (έκδοση 6), την C++ της Borland και την MS Visual C++, για Windows 95/98 και Windows NT.

Όπως ήταν αναμενόμενο, στις μέρες μας πλέον και γενικότερα στην εποχή του αντικειμενοστραφούς προγραμματισμού (Object-Oriented Programming), η παράλληλη θύρα του εκτυπωτή δεν είναι τίποτα περισσότερο από ένα αντικείμενο (Object) με συγκεκριμένες ιδιότητες (Properties) και το οποίο ανταποκρίνεται σε όλα τα είδη των συμβάντων. Κάνοντας χρήση της αρχής αυτής, ο προγραμματιστής έχει εύκολη πρόσβαση στο αντικείμενο (στη παρούσα περίπτωση, τη παράλληλη θύρα), όπου ο σχετικός έλεγχος επιτυγχάνεται μέσα από τις μεθόδους (methods) -που σχετίζονται με το αντίστοιχο στοιχείο (component). Στη συνέχεια δίνεται μια σύντομη περιγραφή των μεθόδων και των ιδιοτήτων που χρησιμοποιούνται από το πρόγραμμα του παραδείγματος (μιας εφαρμογής της έκδοσης 2 της γλώσσας Delphi).

Ουσιαστικά, οι βασικές ρουτίνες (procedures) είναι η *OpenDriver* και η *CloseDriver* οι οποίες είναι υπεύθυνες, όπως θα μαντέψατε, για το άνοιγμα και το κλείσιμο αντίστοιχα, του οδηγού *TVicport*. Η λογική μεταβλητή ή σημαία (ιδιότητα) *ActiveHW* δείχνει κατά πόσο η θύρα είναι ανοικτή ή κλειστή, ή πιο ουσιαστικά, φανερώνει αν ο οδηγός είναι διαθέσιμος ή όχι.

Η ιδιότητα *LPTNumPorts* αντιστοιχεί στον αριθμό των παράλληλων θυρών του συστήματος, ενώ η *LPTNumber* καθορίζει ποια θύρα ελέγχεται από τον οδηγό. Τέλος, η *LPTBaseAddress* αντιστοιχεί με τη σειρά της, στη βασική διεύθυνση της χρησιμοποιούμενης θύρας. Το στοιχείο που αντιστοιχεί στον οδηγό αυτό, διαθέτει επίσης και μια ιδιότητα η οποία καλείται *Pin* και που είναι ένας πίνακας με 25 λογικές μεταβλητές (bits) που περιγράφουν τις λογικές καταστάσεις οι οποίες διαβάζονται από τους ακροδέκτες (pins) της θύρας. Με τη βοήθεια του πίνακα αυτού, ο προγραμματιστής έχει εύκολη πρόσβαση στους ακροδέκτες της παράλληλης θύρας. Πρέπει ωστόσο να είστε προσεκτικοί, ελέγχοντας ποιό από τους ακροδέκτες είναι μόνο εισόδοι ή μόνο έξοδοι ή διπλής κατεύθυνσης.

Το HWTest και το παράδειγμα του προγράμματος

Το πρόγραμμα HWTest που περιέχεται στη

συνοδευτική δισκέτα της κατασκευής, περιγράφει διάφορους πρακτικούς τρόπους χρήσης του στοιχείου αυτού. Θεωρούμε ότι αποτελεί μια εξαιρετική αφετηρία για τις δικές σας εφαρμογές.

Το πρόγραμμα του παραδείγματος (που περιέχεται επίσης στην δισκέτα) προέρχεται ουσιαστικά από το πρόγραμμα HWTest. Οι οκτώ ηλεκτρονόμοι της πλακέτας του UPIO μπορούν να ενεργοποιηθούν και να απενεργοποιηθούν επιλέγοντας (tick) τα αντίστοιχα τετραγωνίδια. Τα αντίστοιχα σημεία (ticks) που φαίνονται στο πλαίσιο της οθόνης κάτω από τα τετραγωνίδια που αντιστοιχούν στις εξόδους των ηλεκτρονόμων, επισημαίνουν τη λογική κατάσταση των εισόδων, δηλαδή αν κάποια είσοδος είναι σε επίπεδο high ή low. Αν ο υπολογιστής σας διαθέτει περισσότερες από μια παράλληλες θύρες, ο έλεγχος του περιστρεφόμενου διακόπτη επιτρέπει την επιλογή μιας άλλης θύρας. Φυσικά, όπως είναι αναμενόμενο ο έλεγχος της παράλληλης θύρας θα λειτουργήσει μόνο αν ο οδηγός φορτωθεί σωστά, πράγμα που επιτυγχάνεται πιέζοντας το κουμπί 'Open Driver'. Στο Σχήμα 3 φαίνεται μια αντιπροσωπευτική οθόνη του προγράμματος ελέγχου του UPIO.

Εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της κατασκευής

Για το σχεδιασμό του πρωτότυπου της εφαρμογής, σε εκτελέσιμη μορφή, ο σχεδιαστής χρησιμοποίησε το αναπτυξιακό περιβάλλον Delphi της Borland (Έκδοση 1 - 16 bits), με το οποίο δημιουργήθηκαν τα αρχεία που σχετίζονται με το πηγαίο κώδικα της εφαρμογής, με τις επεκτάσεις '.dpr', '.pas', '.opt' και '.res'. Όλα τα σχετικά αρχεία του πρωτότυπου της εφαρμογής UPIO, μαζί με την ανανεωμένη έκδοση του προγράμματος ελέγχου, θα τα βρείτε στην δισκέτα με κωδικό **002011-11**. Πρέπει να σημειωθεί ότι οι πρωτότυποι πηγαίοι κώδικες όπως μας παραδόθηκαν από το σχεδιαστή της εφαρμογής, διατίθενται μόνο για πληροφοριακούς λόγους - δε μπορούν να τροποποιηθούν διότι λείπει το στοιχείο του οδηγού της παράλληλης θύρας (LPT Component). Επίσης, εξαιτίας κάποιων προβλημάτων που παρατηρήθηκαν, δε σας συνιστούμε να χρησιμοποιήσετε τον πρωτότυπο εκτελέσιμο κώδικα.

Για λόγους απλά ενημερωτικούς, αναφέρεται ότι το σχηματικό διάγραμμα της κατασκευής σχεδιάστηκε αρχικά με τη βοήθεια του πακέτου 'Protel Schematic Capture', ενώ το λογισμικό 'Advanced Schematics' χρησιμοποιήθηκε για τη σχεδίαση του τυπωμένου κυκλώματος. Τόσο το σχηματικό διάγραμμα, όσο και το τυπωμένο κύκλωμα, σχεδιάστηκαν ξανά από την αρχή για τις ανάγκες του παρόντος άρθρου.