

**Τ.Ε.Ι. ΠΕΙΡΑΙΑ**  
**ΤΜΗΜΑ**  
**Η/Υ ΣΥΣΤΗΜΑΤΩΝ**

**▼ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ▼**

**ΤΕΡΜΑΤΙΚΟ ΕΛΕΓΧΟΥ**  
**ΠΡΟΣΒΑΣΗΣ ΣΕ ΧΩΡΟ ΚΑΙ**  
**ΕΝΔΕΙΞΗΣ ΜΗΝΥΜΑΤΩΝ ΑΠΟ Η/Υ**



**ΣΠΟΥΔΑΣΤΗΣ:**  
**ΠΑΝΑΓΙΩΤΗΣ ΔΙΟΝ. ΚΕΝΤΕΡΛΗΣ**

**ΕΙΣΗΓΗΤΗΣ:**  
**κ. ΙΩΑΝΝΗΣ ΕΛΛΗΝΑΣ**

---

Με την ευκαιρία της κατάθεσης της πτυχιακής μου εργασίας θα ήθελα να ευχαριστήσω τα εξής άτομα:

- ❖ την ομάδα ανάπτυξης της γλώσσας προγραμματισμού SDCC για μικροελεγκτές της οικογένειας 8051.
- ❖ τον υπεύθυνο επιβλέπων καθηγητή κ. Ιωάννη Έλληνα για την σημαντική του καθοδήγηση στην εργασία αυτή, αλλά και για την ανάπτυξη του ενδιαφέροντος μου για ενασχόληση με τους μικροελεγκτές και τα μικροϋπολογιστικά συστήματα.
- ❖ τους συμφοιτητές και φίλους μου για την συντροφιά και συνεργασία που μοιραστήκαμε στα τέσσερα χρόνια φοίτησης στο Ίδρυμα.
- ❖ το υπόλοιπο εκπαιδευτικό προσωπικό του εργαστηρίου Μικροεπεξεργαστών και Μικροϋπολογιστών, και ιδιαίτερα τον καθηγητή κ. Παναγιώτη Δροσινόπουλο, για την συνεργασία τους και την φιλική αντιμετώπιση τους τόσο κατά την διάρκεια της πρακτικής μου εργασίας, όσο και κατά την διάρκεια της απασχόλησης μου εντός του χώρου του εργαστηρίου.

Ο σπουδαστής

Παναγιώτης Κεντερλής

---

---

# Περιεχόμενα

---

<b>ΚΕΦΑΛΑΙΟ 1<sup>ο</sup></b>	<b>1</b>
ΣΚΟΠΟΣ ΔΗΜΙΟΥΡΓΙΑΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	1
<b>ΚΕΦΑΛΑΙΟ 2<sup>ο</sup></b>	<b>2</b>
<b>Ο ΜΙΚΡΟΕΛΕΓΚΤΗΣ</b>	<b>2</b>
ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΕΛΕΓΧΟΥ ΤΟΥ ΜΙΚΡΟΕΛΕΓΚΤΗ.	4
ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΥΠΟΛΟΓΙΣΤΗ.	4
<b>ΧΑΡΤΗΣ ΜΝΗΜΗΣ ΤΟΥ ΤΕΡΜΑΤΙΚΟΥ</b>	<b>5</b>
<b>Η ΓΡΑΦΙΚΗ ΟΘΟΝΗ LCD</b>	<b>7</b>
ΠΕΡΙΦΕΡΕΙΑΚΟ HARDWARE ΤΗΣ ΟΘΟΝΗΣ LCD	8
Ο ΧΑΡΤΗΣ ΜΝΗΜΗΣ ΤΟΥ ΕΛΕΓΚΤΗ T6963C	9
Η ΓΕΝΝΗΤΡΙΑ ΧΑΡΑΚΤΗΡΩΝ	11
<b>ΤΟ ΡΟΛΟΙ ΠΡΑΓΜΑΤΙΚΟΥ ΧΡΟΝΟΥ (RTC – REAL TIME CLOCK)</b>	<b>16</b>
<b>ΤΟ ΠΛΗΚΤΡΟΛΟΓΙΟ</b>	<b>18</b>
Κωδικοί Σάρωσης Γραμμών Πληκτρολογίου ScanPattern	19
Κωδικοί Επιστροφής Στηλών Πληκτρολογίου ReturnCode	19
Πίνακας Αναζήτησης & Αντιστοίχισης ScanCode	20
<b>ΤΟ ΚΛΕΙΔΙ ΑΣΦΑΛΟΥΣ ΠΡΟΣΒΑΣΗΣ</b>	<b>21</b>
ΑΡΧΙΚΟΠΟΙΗΣΗ ΕΠΙΚΟΙΝΩΝΙΑΣ	22
ΑΝΑΓΝΩΣΗ ΤΟΥ IBUTTON	23
ΛΟΓΟΙ ΕΠΙΛΟΓΗΣ ΤΟΥ IBUTTON	24
<b>ΈΛΕΓΧΟΣ ΤΗΣ ΚΛΕΙΔΑΡΙΑΣ</b>	<b>25</b>
ΥΠΟΛΟΓΙΣΜΟΣ ΕΞΑΡΤΗΜΑΤΩΝ ΚΥΚΛΩΜΑΤΟΣ ΟΔΗΓΗΣΗΣ ΡΕΛΕ	25
ΕΠΕΞΗΓΗΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΚΥΚΛΩΜΑΤΟΣ ΕΛΕΓΧΟΥ ΚΛΕΙΔΑΡΙΑΣ	26
<b>ΣΕΙΡΙΑΚΟ INTERFACE</b>	<b>27</b>
<b>ΛΟΙΠΑ ΚΥΚΛΩΜΑΤΑ</b>	<b>29</b>
ΚΥΚΛΩΜΑ ΤΟΥ ΒΟΜΒΗΤΗ	29
ΑΝΙΧΝΕΥΣΗ ΚΑΤΑΣΤΑΣΗΣ ΠΟΡΤΑΣ	29
<b>ΚΕΦΑΛΑΙΟ 3<sup>ο</sup></b>	<b>30</b>
<b>ΥΛΟΠΟΙΗΣΗ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΤΕΡΜΑΤΙΚΟΥ</b>	<b>30</b>
ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ	30
ΤΥΠΩΜΕΝΟ ΚΥΚΛΩΜΑ	32
Κατάλογος Εξαρτημάτων	32
Συνδέσεις Τυπωμένου Κυκλώματος	33
Τοπογραφικό Εξαρτημάτων	34
Κάτω Όψη	35
Πάνω Όψη	36
ΤΡΟΦΟΔΟΣΙΑ ΚΥΚΛΩΜΑΤΟΣ	37
ΕΞΩΤΕΡΙΚΕΣ ΣΥΝΔΕΣΕΙΣ ΚΥΚΛΩΜΑΤΟΣ ΤΕΡΜΑΤΙΚΟΥ	38

---

---

**ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>** **39**

---

<b>ΠΡΩΤΟΚΟΛΛΟ ΕΠΙΚΟΙΝΩΝΙΑΣ ΤΕΡΜΑΤΙΚΟΥ – Η/Υ</b>	<b>39</b>
Εντολές του Λογισμικού του Η/Υ προς το Τερματικό	39
Εντολές του Τερματικού προς το Λογισμικό του Η/Υ	41
<b>ΔΙΑΧΕΙΡΙΣΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ ΛΟΓΙΣΜΙΚΟ ΤΟΥ Η/Υ</b>	<b>43</b>
ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΣΥΣΤΗΜΑΤΟΣ	43
Δομή Πίνακα Announcements	44
Δομή Πίνακα CourseDBs	44
Δομή Πίνακα Entry Log	45
Δομή Πίνακα Key Owners	45
Δομή Πίνακα KeyMediaPool	46
ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΒΑΘΜΟΛΟΓΙΚΩΝ ΠΙΝΑΚΩΝ	46
Δομή Πίνακα ‘Βαθμολογίες’	46

---

**ΚΕΦΑΛΑΙΟ 5<sup>ο</sup>** **47**

---

<b>ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ ΤΕΡΜΑΤΙΚΟΥ</b>	<b>47</b>
<b>ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ ΥΠΟΛΟΓΙΣΤΗ</b>	<b>48</b>

---

**ΚΕΦΑΛΑΙΟ 6<sup>ο</sup>** **57**

---

<b>ΚΩΔΙΚΑΣ ΛΟΓΙΣΜΙΚΟΥ ΤΟΥ ΤΕΡΜΑΤΙΚΟΥ</b>	<b>57</b>
MAIN.C	57
DS5000.C	61
DS5000.H	62
IBUTTON.C	65
IBUTTON.H	70
KEYB.C	71
KEYB.H	73
T6963C.C	74
T6963C.H	83
SERPORT.C	85
SERPORT.H	90
RTC.C	91
RTC.H	94
HEADERS.H	95
HEADERS2.H	97
FUNC.C	98
FUNC.H	117

---

**ΚΕΦΑΛΑΙΟ 7<sup>ο</sup>** **118**

---

<b>ΚΩΔΙΚΑΣ ΛΟΓΙΣΜΙΚΟΥ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΥΠΟΛΟΓΙΣΤΗ</b>	<b>118</b>
ΦΟΡΜΕΣ	118
MainWindow Form	118
Κώδικας Φόρμας MainWindow	119
AccessLogs Form	129
Κώδικας Φόρμας AccessLogs	129
AdminPwdChange Form	131

---

---

Κώδικας Φόρμας AdminPwdChange	131
AdminPwdCheck Form	132
Κώδικας Φόρμας AdminPwdCheck	132
AllCourses Form	133
Κώδικας Φόρμας AllCourses	133
AllKeyOwners Form	135
Κώδικας Φόρμας AllKeyOwners	135
AnnouncementEditor Form	138
Κώδικας Φόρμας AnnouncementEditor	138
Announcements Form	139
Κώδικας Φόρμας Announcements	139
DesignerInfo Form	141
Κώδικας Φόρμας DesignerInfo	141
DiagCheckResults Form	142
Κώδικας Φόρμας DiagCheckResults	142
KeyOwnerDetails Form	143
Κώδικας Φόρμας KeyOwnerDetails	143
KeyPoolSelect Form	147
Κώδικας Φόρμας KeyPoolSelect	147
LogClear Form	150
Κώδικας Φόρμας LogClear	150
MarqueeSetup Form	152
Κώδικας Φόρμας MarqueeSetup	152
MasterKeySet Form	154
Κώδικας Φόρμας MasterKeySet	154
RestrictDates Form	156
Κώδικας Φόρμας RestrictDates	156
ΑΡΧΕΙΑ ΡΟΥΤΙΝΩΝ (MODULES)	158
Ρουτίνες Αρχείου Routines.bas	158
Ρουτίνες Αρχείου Routines2.bas	168
Ρουτίνες Αρχείου Routines3.bas	174
ΦΟΡΜΕΣ ΑΝΑΦΟΡΑΣ	182
EntryLogsReport Designers Form	182
ΠΕΡΙΒΑΛΛΟΝ ΔΙΑΧΕΙΡΙΣΗΣ ΔΕΔΟΜΕΝΩΝ	183
DataEnvironment1 Designers Form	183

---

---

---

# Κεφάλαιο 1<sup>ο</sup>

---

---

## Σκοπός Δημιουργίας της Πτυχιακής Εργασίας

Το μικροϋπολογιστικό σύστημα της πτυχιακής αυτής εργασίας, δημιουργήθηκε με σκοπό να καλύψει τρεις βασικές και μελλοντικές ανάγκες του Εργαστηρίου Μικροεπεξεργαστών και Μικροϋπολογιστών του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Πειραιά.

Η ανάγκη για προστασία του χώρου από αυθαίρετη και αδικαιολόγητη πρόσβαση καλύπτεται σε σημαντικό βαθμό, καθώς η πρόσβαση στον ελεγχόμενο χώρο είναι δυνατή μόνο από κατόχους ειδικής μορφής ηλεκτρονικών κλειδιών ασφαλείας. Τα κλειδιά αυτά συμπληρώνονται σε βαθμό ασφάλειας από έναν προσωπικό εξαψήφιο αριθμητικό κωδικό, μοναδικό για κάθε κάτοχο κλειδιού. Η επιβεβαίωση της ταυτότητας του κατόχου και ο έλεγχος του δικαιώματος για πρόσβαση στον ελεγχόμενο χώρο γίνεται συνεργαζόμενο το τερματικό (το μικροϋπολογιστικό σύστημα) με έναν Ηλεκτρονικό Υπολογιστή, στον οποίο εκτελείται ειδικό λογισμικό ελέγχου του πρώτου. Οι δυνατότητες που δίνει ο συνδυασμός τερματικού-Η/Υ για ασφάλεια του χώρου είναι αρκετές, καθώς παράλληλα με την παροχή πρόσβασης στον χώρο, γίνεται καταγραφή αυτής, για ενίσχυση της ασφάλειας και επιτήρησης του χώρου. Για την επίτευξη του στόχου αυτού γίνεται χρήση ηλεκτρομηχανικής κλειδαριάς, εγκατεστημένης στην είσοδο του ελεγχόμενου χώρου.

Η ανάγκη για πληροφόρηση των σπουδαστών αλλά και κάθε ενδιαφερομένου για τις δραστηριότητες του εργαστηριακού χώρου, γίνεται πραγματικότητα με την ύπαρξη ειδικής λειτουργίας του τερματικού για δημοσίευση ανακοινώσεων μέσω μιας οθόνης τύπου LCD (Liquid Crystal Display). Η ηλεκτρονική μορφή των ανακοινώσεων είναι άριστα οργανωμένη σε ομάδα οκτώ θέσεων δημοσιεύσιμων ανακοινώσεων. Ο ενδιαφερόμενος με την χρήση ειδικού περιβάλλοντος (Interface) επιλογών, μπορεί να αποφασίσει ποια ανακοίνωση τον ενδιαφέρει επιλέγοντας τον τίτλο, και να διαβάσει το κείμενο της. Οι ανακοινώσεις βρίσκονται αποθηκευμένες σε ηλεκτρονική μορφή στον Η/Υ και διαχειρίζονται από το λογισμικό αυτού. Ανά πάσα στιγμή ο διαχειριστής του συστήματος, μπορεί να δημιουργήσει, επεξεργαστεί, διαγράψει μια ανακοίνωση και οι αλλαγές αυτές να είναι άμεσα διαθέσιμες στο κοινό, χωρίς άλλες ενδιάμεσες διαδικασίες (π.χ. συγγραφή ανακοίνωσης, εκτύπωση, ανάρτηση σε πίνακα ανακοινώσεων).

Η ανάγκη για πληροφόρηση των σπουδαστών για την βαθμολογική τους κατάσταση στα μαθήματα του εργαστηριακού χώρου και όχι μόνο, καλύπτεται επίσης από την εργασία αυτή. Οι σπουδαστές με την χρήση του Interface του τερματικού μπορούν να επιλέξουν, κατά τρόπο αντίστοιχο με το περιβάλλον δημοσίευσης ανακοινώσεων, να δουν τους βαθμολογικούς πίνακες μαθημάτων που δημοσιεύονται. Οι βαθμολογικοί πίνακες αυτοί αποθηκεύονται στον Η/Υ και δημοσιεύονται μόνο κατ' επιλογή του διαχειριστή του συστήματος. Είναι δυνατή η δημοσίευση βαθμολογικών πινάκων ενός συνόλου οκτώ μαθημάτων. Οι βαθμολογικοί πίνακες είναι διασφαλισμένοι από κάθε προσπάθεια αλλοίωσης, καθώς το τερματικό δεν έχει δυνατότητα αλλαγής των στοιχείων, τα οποία παραμένουν προστατευμένα από την ίδια την ασφάλεια που χαρακτηρίζει και τον Η/Υ.

Στα επόμενα κεφάλαια αναπτύσσεται η λογική και το υλικό που έκαναν δυνατή την δημιουργία της πτυχιακής εργασίας.

# Κεφάλαιο 2<sup>ο</sup>

## Ο Μικροελεγκτής

Για την υλοποίηση της πτυχιακής εργασίας χρησιμοποιήθηκε ο μικροελεγκτής DS5000T της Dallas Semiconductors, ο οποίος περιέχει πυρήνα συμβατό με την οικογένεια μικροελεγκτών MCS-51 που ανέπτυξε η εταιρεία Intel.

Διατίθεται σε μορφή Encapsulated DIP 40 ακροδεκτών με ενσωματωμένα τα εξής στοιχεία: έναν πλήρως λειτουργικό 8051, μνήμη τύπου RAM, στοιχείο λιθίου των 3V και ένα RTC (Real Time Clock – Ρόλοι Πραγματικού Χρόνου) με τον αντίστοιχο κρύσταλλο χρονοισμού 32,768KHz.

Το στοιχείο λιθίου χρησιμοποιείται για την τροφοδοσία της μνήμης RAM, αλλά και του RTC, όταν ο μικροελεγκτής είναι εκτός εξωτερικής τροφοδοσίας. Με την τεχνική αυτή η μνήμη RAM μετατρέπεται σε NVRAM (Non Volatile RAM – μη πτητική μνήμη RAM), ενώ το RTC συνεχίζει να μετράει την ώρα ακόμα και χωρίς εξωτερική τροφοδοσία.

Η μνήμη NVRAM χρησιμοποιείται από τον πυρήνα 8051 μέσω ενός ειδικού διαύλου Byte-Wide ξεχωριστό από τις πόρτες I/O, προκειμένου να αποθηκεύει, τόσο το προς εκτέλεση πρόγραμμα, όσο και τα δεδομένα του συστήματος. Το συγκεκριμένο ολοκληρωμένο επιλέχθηκε με μνήμη 32Kbytes η οποία χωρίστηκε κατά την φάση του προγραμματισμού του προγράμματος ελέγχου του τερματικού, σε 24Kbytes για αποθήκευση προγράμματος και 8Kbytes για αποθήκευση δεδομένων.

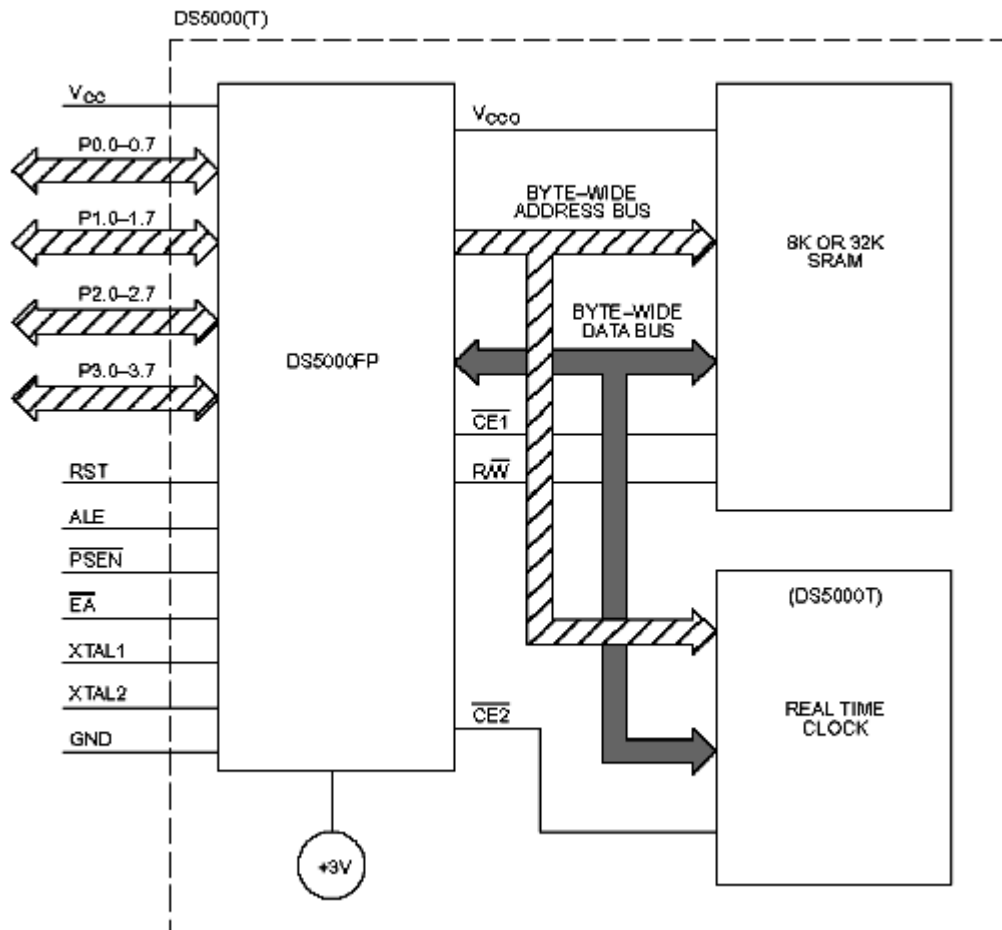
Παρότι στο σύστημα της εργασίας αυτής απαιτείτο η χρήση ενός RTC, δεν χρησιμοποιήθηκε το ήδη ενσωματωμένο στον μικροελεγκτή για έναν σημαντικό λόγο. Το ενσωματωμένο RTC προσπελάζεται σειριακά μέσω του εσωτερικού Byte-Wide δίαυλου και δεν διαθέτει ακροδέκτη πρόκλησης διακοπών. Η ικανότητα πρόκλησης διακοπών κρίνεται ιδιαίτερα χρήσιμη γιατί απαλλάσσει τον μικροελεγκτή από την ανάγκη για συνεχή ανάγνωση του RTC προκειμένου να αναγνωρίσει την αλλαγή του ελάχιστου χρονικού διαστήματος του ενός δευτερολέπτου. Επίσης η δυνατότητα παραγωγής διακοπών σε σταθερά χρονικά διαστήματα διάρκειας ενός δευτερολέπτου είναι χρήσιμη για την ανάπτυξη μερικών ρουτινών του συστήματος. Αντίθετα επιλέχθηκε η χρήση ενός κλασσικού ολοκληρωμένου RTC, το οποίο περιγράφεται παρακάτω.

## PIN ASSIGNMENT

P1.0	1	40	V <sub>CC</sub>
P1.1	2	39	P0.0 AD0
P1.2	3	38	P0.1 AD1
P1.3	4	37	P0.2 AD2
P1.4	5	36	P0.3 AD3
P1.5	6	35	P0.4 AD4
P1.6	7	34	P0.5 AD5
P1.7	8	33	P0.6 AD6
RST	9	32	P0.7 AD7
RXD P3.0	10	31	$\overline{EA}$
TXD P3.1	11	30	ALE
$\overline{INT0}$ P3.2	12	29	$\overline{PSEN}$
$\overline{INT1}$ P3.3	13	28	P2.7 A15
T0 P3.4	14	27	P2.6 A14
T1 P3.5	15	26	P2.5 A13
$\overline{WR}$ P3.6	16	25	P2.4 A12
$\overline{RD}$ P3.7	17	24	P2.3 A11
XTAL2	18	23	P2.2 A10
XTAL1	19	22	P2.1 A9
GND	20	21	P2.0 A8

40-Pin ENCAPSULATED PACKAGE

Ο μικροελεγκτής διαθέτει ενσωματωμένο UART (Universal Asynchronous Receiver Transmitter). Επίσης ενσωματωμένα στον μικροελεγκτή είναι 4 πόρτες I/O μήκους 8bit, 2 προγραμματιζόμενοι Timers των 16bit, 2 ακροδέκτες αίτησης διακοπών /INT0 και /INT1 καθώς και ενσωματωμένο Watchdog Timer. Να αναφέρουμε ότι κάνοντας χρήση δυο εκ των πορτών I/O P0 και P2, μπορούμε να εξομοιώσουμε την λειτουργία ενός 16bit address bus, τα πρώτα 8bit του οποίου είναι χρονικά πολυπλεγμένα με το 8bit data bus, όπως αυτό υλοποιείται σε όλους τους μικροεπεξεργαστές τεχνολογίας 8bit της εταιρίας Intel (π.χ. 8085).



Ο κρύσταλλος χρονισμού του μικροελεγκτή (ρολόι χρονισμού) επιλέχθηκε στην συχνότητα των 11,0592MHz, καθώς η συχνότητα αυτή εξασφαλίζει την μετάδοση και λήψη δεδομένων μέσω της ενσωματωμένης σειριακής θύρας (UART) στα προτυποποιημένα baud rates έως 19200baud με μηδενικό σφάλμα. Ο κρύσταλλος αποτελεί εξωτερικό στοιχείο και συνδέεται στον μικροελεγκτή στους αντίστοιχους ακροδέκτες XTAL1 και XTAL2.

Οι λόγοι που οδήγησαν στην επιλογή του συγκεκριμένου μικροελεγκτή, αφορούν την ευκολία μεταφοράς προγραμμάτων στην μνήμη NVRAM χωρίς την ανάγκη ειδικού προγραμματιστή, το μεγάλο μέγεθος μνήμης που διαθέτει, το γεγονός ότι η μνήμη συνδέεται με το εσωτερικό Byte-Wide δίαυλο και δεν καταλαμβάνει τις πόρτες I/O P0 και P2, το διαθέσιμο software ανάπτυξης εφαρμογών καθώς και η ενσωμάτωση τόσων λειτουργιών (όπως το UART και Timers) εντός ενός ολοκληρωμένου 40 ακροδεκτών. Επίσης σημαντικό για την ασφάλεια του συστήματος είναι η δυνατότητα κρυπτογράφησης και κλειδώματος της μνήμης του μικροελεγκτή.

Αναλυτικά στοιχεία για τον μικροελεγκτή θα βρείτε στα τεχνικά φυλλάδια που συμπεριλαμβάνονται με την έκδοση αυτή.



---

---

## *Εργαλεία Ανάπτυξης Λογισμικού Ελέγχου του Μικροελεγκτή.*

Η ανάπτυξη των ρουτινών του τερματικού, έγιναν με την γλώσσα προγραμματισμού SDCC (Small Device C Compiler), η οποία διατίθεται δωρεάν μέσω του Internet, και υποστηρίζεται από μια ομάδα ανάπτυξης δεκάδων ατόμων. Αποτελεί μια έκδοση της γλώσσας ANSI-C που παράγει κώδικα για μικροελεγκτές της οικογένειας MCS-51, ενώ επίσης υποστηρίζει και άλλους μικροελεγκτές όπως ο AVR της ATMEL, οι 14bit PIC της Microchip, αλλά και τον μικροεπεξεργαστή Z80.

Η γλώσσα προγραμματισμού αυτή προτιμήθηκε για την ευκολία της, αλλά και τις δυνατότητες της στην δημιουργία τελικού κώδικα που εκτελεί πολύπλοκες διαδικασίες, οι οποίες αν αναπτύσσονταν σε συμβολική γλώσσα (Assembly) θα απαιτούσαν τεράστια προγραμματιστική προσπάθεια. Επίσης σημαντικό ρόλο στην επιλογή του software ανάπτυξης έπαιξε και το κόστος απόκτησης κάποιου εμπορικού πακέτου που ήταν ιδιαίτερα απαγορευτικό, καθώς θα ξεπερνούσε 2 έως 4 φορές το συνολικό κόστος της πτυχιακής εργασίας.

Για την εκτέλεση της γλώσσας προγραμματισμού SDCC απαιτήθηκε η χρήση του λογισμικού πακέτου εξομίωσης περιβάλλοντος UNIX, Cygwin. Στο περιβάλλον αυτό έγινε η μεταγλώττιση των επιμέρους αρχείων και η σύνδεση μεταξύ τους.

Η ανάπτυξη του κώδικα προγραμματισμού έγινε με το λογισμικό πακέτο διαχείρισης αρχείων Norton Commander 5.0 σε περιβάλλον DOS (παράθυρο DOS κάτω από Windows).

Όλα τα εργαλεία ανάπτυξης λογισμικού για το τερματικό εκτελούνταν σε περιβάλλον Microsoft Windows Me/2000.

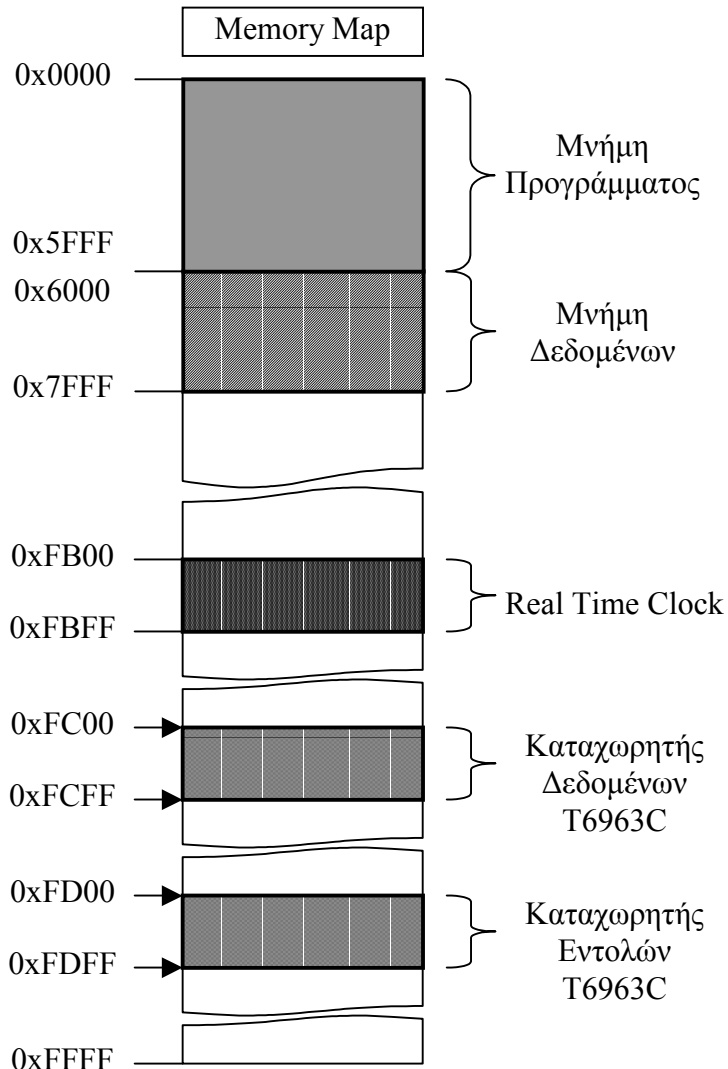
## *Εργαλεία Ανάπτυξης Λογισμικού του Ηλεκτρονικού Υπολογιστή.*

Για την ανάπτυξη της εφαρμογής διαχείρισης του τερματικού, η οποία εκτελείται στον ηλεκτρονικό υπολογιστή, έγινε χρήση της παραθυρικής γλώσσας προγραμματισμού Microsoft Visual Basic 6.0.

Αναλυτικά στοιχεία για την γλώσσα προγραμματισμού Visual Basic θα βρείτε σε βιβλία που διατίθενται στο εμπόριο.

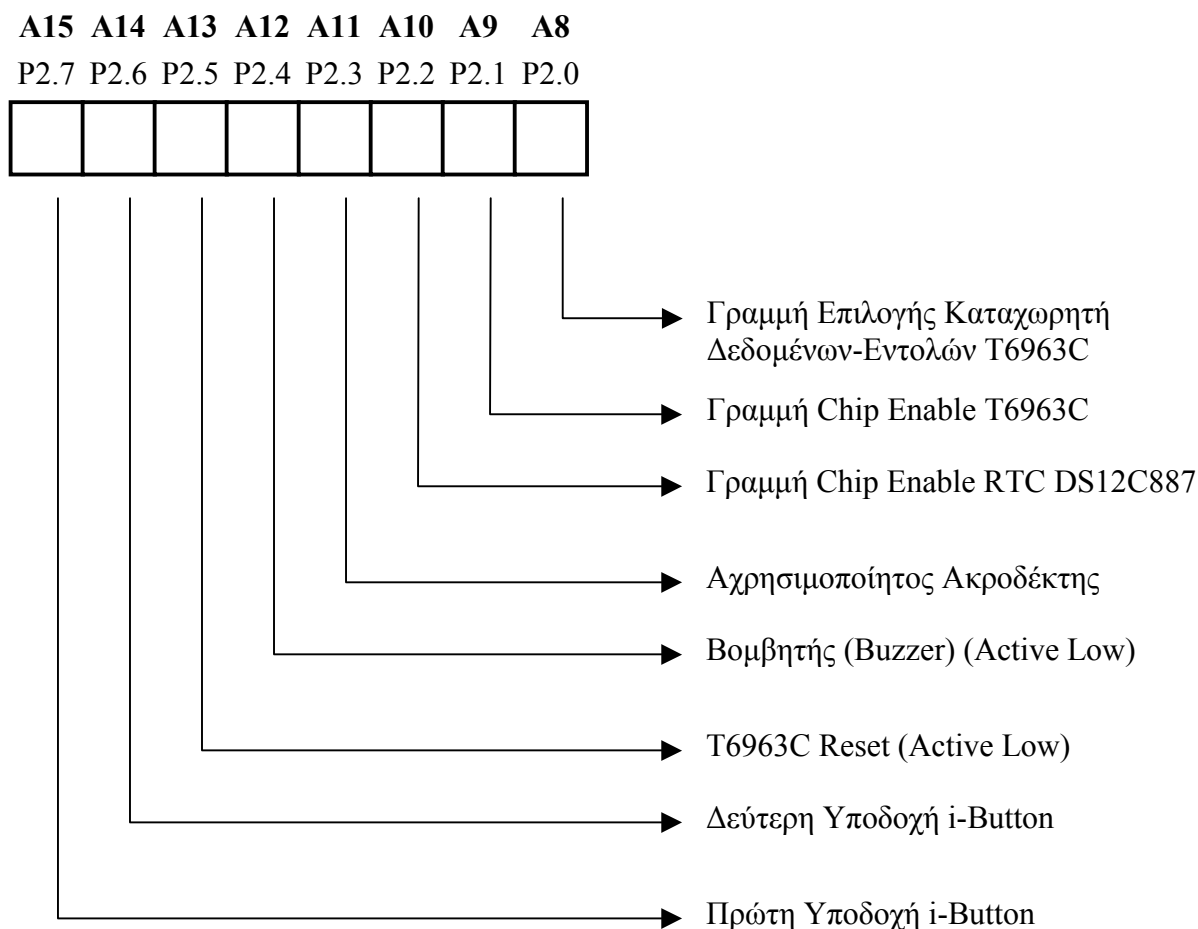
## Χάρτης Μνήμης του Τερματικού

Ο χάρτης μνήμης του μικροϋπολογιστικού συστήματος του τερματικού διαμορφώνεται σύμφωνα με το παρακάτω σχεδιάγραμμα.



Οι θέσεις μνήμης για τις διευθύνσεις 0x0000→0x7FFF χαρτογραφούνται στην εσωτερική μνήμη του μικροελεγκτή DS5000T και προσπελαύνονται μέσω του εσωτερικού Byte-Wide bus. Οι διευθύνσεις 0x8000→0xFFFF προσπελαύνονται μέσω των πόρτων I/O P0 και P2, οι οποίες δουλεύουν ως χρονικά πολυπλεγμένο Address Bus (16bit P0-P2) και Data Bus (8bit P0). Για τις διευθύνσεις αυτές ενεργοποιούνται επίσης οι ακροδέκτες P3.6 και P3.7 με τις ειδικές λειτουργίες διαχείρισης διαύλου /WR και /RD αντίστοιχα. Ο καταχωρητής δεδομένων και ο καταχωρητής εντολών του ελεγκτή T6963C της οθόνης LCD, καταλαμβάνουν ο καθένας μια περιοχή 256 διευθύνσεων. Ο λόγος που συμβαίνει αυτό είναι ότι έτσι απλοποιείται το κύκλωμα αποκωδικοποίησης διευθύνσεων, που στην πράξη είναι ανύπαρκτο καθώς θα προσέθετε κόστος, πολυπλοκότητα στην κατασκευή ενώ δεν θα εξασφάλιζε κάποια προγραμματιστική ή άλλου είδους ευκολία στο κύκλωμα. Το ίδιο ισχύει και για το ολοκληρωμένο RTC που παρότι απαιτεί 128 θέσεις μνήμης, του έχουν αποδοθεί συνολικά 256. Τα σήματα /CE (Active Low Chip Enable) για το RTC αλλά και για τον ελεγκτή της οθόνης LCD παράγονται με την εξής τεχνική, που παρότι δημιουργεί εικόνες στον χάρτη μνήμης (που δεν εμφανίζονται στο παραπάνω διάγραμμα), στην ουσία δεν δημιουργούν πρόβλημα για δύο λόγους. Πρώτον οι εικόνες για τις διευθύνσεις 0x0000→0x7FFF

δεν προσπελαύνονται ποτέ, καθώς για την περιοχή αυτή ανταποκρίνεται η εσωτερική μνήμη NVRAM του μικροελεγκτή. Δεύτερον και κυριότερο το software διαχείρισης του τερματικού δεν κάνει προσπέλαση παρά σε συγκεκριμένες περιοχές, στις οποίες ανταποκρίνονται τα ολοκληρωμένα αυτά, προσπελάζοντας σαφώς καθορισμένες διευθύνσεις.

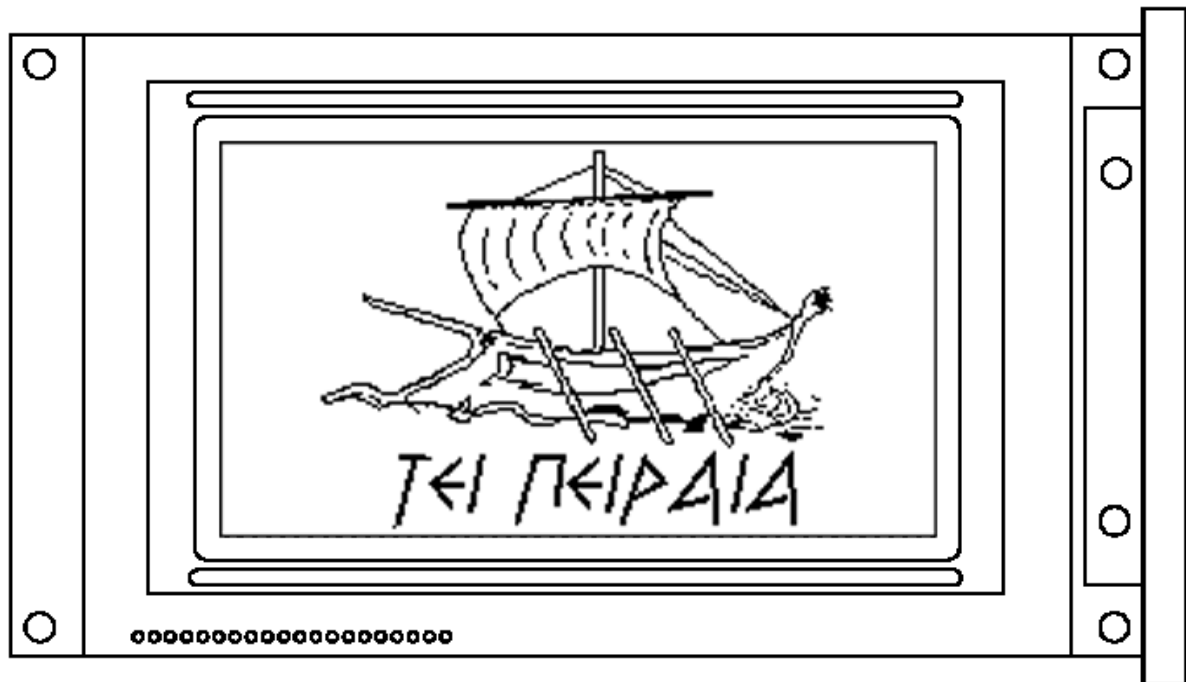


Όπως φαίνεται και από το παραπάνω σχεδιάγραμμα δυο ακροδέκτες της πόρτας P2, η οποία χρησιμοποιείται ως έξοδος του upper byte της διεύθυνσης του Address Bus, αναλαμβάνουν τον ρόλο του /CE των δυο εξαρτημάτων (RTC – T6963C), ενώ ένας ακόμα ακροδέκτης επιλέγει τον καταχωρητή δεδομένων ή τον καταχωρητή εντολών του T6963C.

Το lower byte της σχηματιζόμενης διεύθυνσης δεν χρησιμοποιείται σε κάποιο κύκλωμα αποκωδικοποίησης, δίνοντας σε κάθε εικόνα που σχηματίζεται στην μνήμη, μέγεθος 256bytes. Έτσι όμως καθίσταται και ανούσια η χρήση κάποιου ολοκληρωμένου συγκράτησης και απομόνωσης (Latching Buffer 74HC373 ή 74HC573) για διαχωρισμό μεταξύ Address και Data Bus.

---

## Η Γραφική Οθόνη LCD



Η οθόνη γραφικών τύπου LCD (Liquid Crystal Display) που χρησιμοποιήθηκε για το κύκλωμα είναι η Hantronix 128GS24 οριζόντιας και κάθετης ανάλυσης αντίστοιχα των 240x128 pixel. Για την απόκτηση της έγινε ειδική παραγγελία από το εξωτερικό, καθώς δεν ευρέθηκε άλλη οθόνη με τα ίδια χαρακτηριστικά από κάποια αντιπροσωπεία ξένων εταιριών στην Ελλάδα. Έγινε η επιλογή οθόνης γραφικών λόγω της δυνατότητας εμφάνισης πολλών γραμμών κειμένου σε αντίθεση με τις οθόνες χαρακτήρων, των οποίων ο αριθμός γραμμών συνήθως περιορίζεται στις 4 ή 8 με μικρό μήκος χαρακτήρων ανά γραμμή. Οι ανάλυση της οθόνης επιτρέπει, για χρήση μήτρας χαρακτήρων 8x8 pixel, την εμφάνιση συνολικά 16 γραμμών μήκους 30 χαρακτήρων η κάθε μια. Έτσι είναι δυνατό να αποδοθούν περισσότερες πληροφορίες σε μορφή κειμένου, καθώς επίσης λόγω της δυνατότητας εμφάνισης γραφικών είναι δυνατή η σχεδίαση ενός εύχρηστου και καλαίσθητου Interface επικοινωνίας με τον χρήστη του τερματικού.

Η οθόνη ως αυτόνομο στοιχείο παρέχει αυτοφωτισμό με χρήση LED χρώματος πράσινου τροφοδοτούμενα από εξωτερική τροφοδοσία των +5V. Επίσης διαθέτει ενσωματωμένο κύκλωμα παραγωγής αρνητικών τάσεων, καθώς και κυκλώματα σάρωσης και ανανέωσης των οπτικών πληροφοριών της οθόνης. Η επικοινωνία με τον μικροελεγκτή του τερματικού γίνεται μέσω δίαυλου παράλληλης μεταφοράς δεδομένων μήκους 8bit, συμβατό με διαύλους επεξεργαστών Intel και κατά συνέπεια και με την οικογένεια μικροελεγκτών 8051.

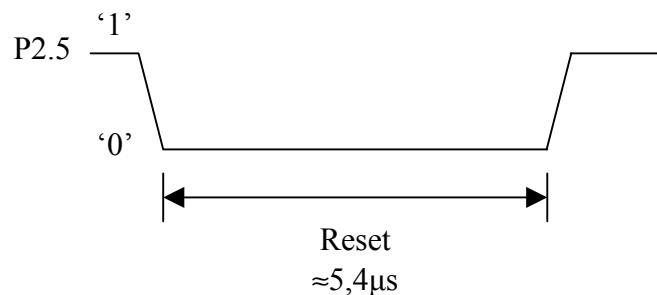
Ο ελεγκτής της οθόνης LCD είναι ο Toshiba T6963C υποστηριζόμενος από 8Kbytes μνήμης RAM, στην οποία αποθηκεύονται ανά περιοχές, το κείμενο, τα γραφικά και ο πίνακας αναφοράς (lookup-table) της γεννήτριας χαρακτήρων που ενσωματώνει. Καταλαμβάνει συνολικά δυο θέσεις μνήμης στον χάρτη μνήμης του συστήματος. Συγκεκριμένα η διεύθυνση 0xFC00 αντιστοιχεί στον καταχωρητή δεδομένων του ελεγκτή (Data Register), ενώ η διεύθυνση 0xFD00 στον καταχωρητή εντολών (Command Register). Περισσότερες πληροφορίες για τον τρόπο προσπέλασης των καταχωρητών αυτών καθώς και για τους λόγους που οδήγησαν στην επιλογή των διευθύνσεων αυτών θα βρείτε στην ανάλυση του χάρτη μνήμης του τερματικού. Μαζί με το βιβλίο αυτό περιλαμβάνεται εκτενές φυλλάδιο οδηγιών για τον τρόπο προγραμματισμού του ελεγκτή T6963C.

---

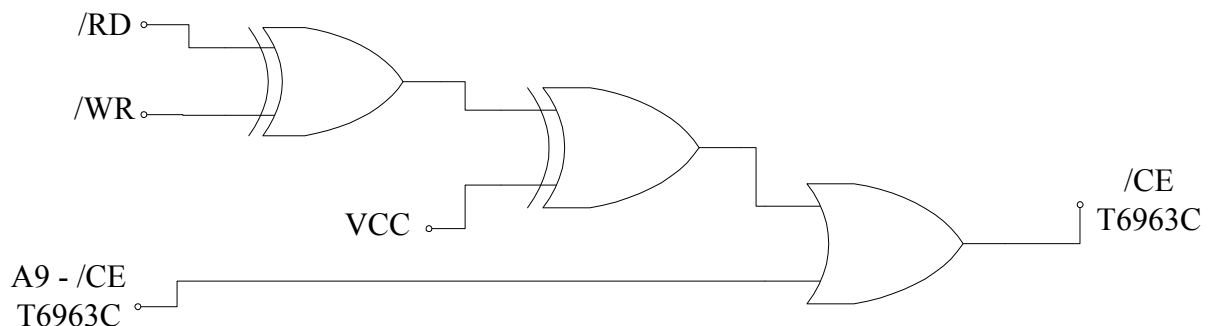
Οι ρουτίνες προγραμματισμού και χειρισμού της οθόνης γραφικών με τον ελεγκτή T6963C περιέχονται στο αρχείο **T6963C.C**. Να σημειωθεί ότι οι περισσότερες ρουτίνες αφορούν τον συγκεκριμένο ελεγκτή ανεξάρτητα σε ποια οθόνη αυτός ενσωματώνεται και μπορούν να τροποποιηθούν στην εκάστοτε σύνθεση με ελάχιστες τροποποιήσεις.

## Περιφερειακό Hardware της οθόνης LCD

Ο ελεγκτής T6963C, κατά την αρχικοποίηση των συστημάτων του τερματικού, εξαναγκάζεται σε κατάσταση Reset με την ρουτίνα ResetLCD( ) η οποία θέτει σε λογικό '0' τον ακροδέκτη P2.5, στον οποίο είναι συνδεδεμένος ο ακροδέκτης /Reset του T6963C. Ο ακροδέκτης P2.5 παραμένει σε λογική κατάσταση '0' για χρονικό διάστημα 5 κύκλων εντολής ενός Byte, δηλαδή για 5,4μsec, χρόνο περισσότερο από αυτόν που ορίζεται από το φυλλάδιο χρήσης του T6963C. Κατόπιν επαναφέρεται στην λογική κατάσταση '1' όπου και παραμένει καθ' όλη την διάρκεια λειτουργίας του τερματικού.

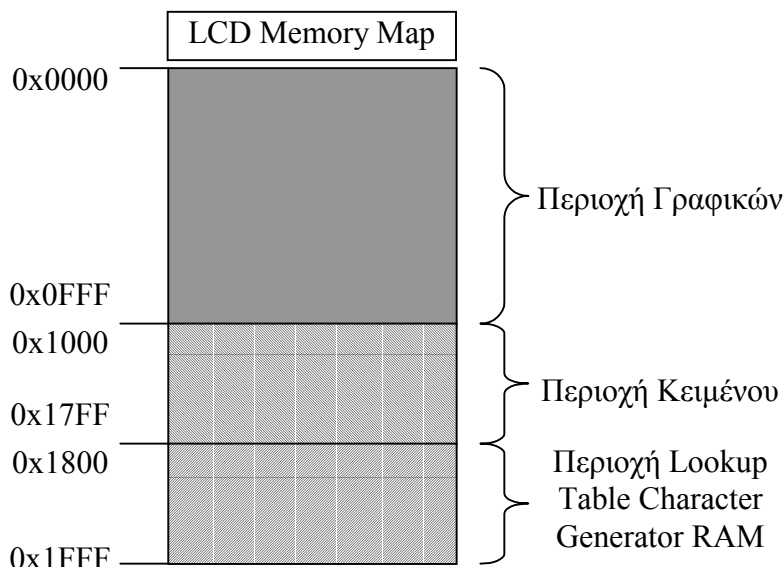


Για την τήρηση των διαγραμμάτων χρονισμού του ελεγκτή T6963C για εγγραφή και ανάγνωση των καταχωρητών του, απαιτείται η χρήση ενός λογικού κυκλώματος σύνδεσης (Glue Logic) αποτελούμενο από δύο ολοκληρωμένα. Στη θέση της δεύτερης λογικής πύλης XOR, θα μπορούσε να συνδεθεί ένας αναστροφέας (πύλη NOT) αλλά για λόγους ελαχιστοποίησης του κόστους και αποφυγής χρήσεως ενός επιπλέον ολοκληρωμένου, προτιμήθηκε η ισοδύναμη συνδεσμολογία μιας εκ των τεσσάρων πυλών XOR που περιέχονται στο ολοκληρωμένο 74HC76. Η λογική πύλη OR περιέχεται στο ολοκληρωμένο 74HC32.



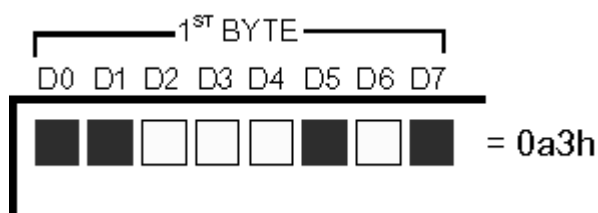
## Ο Χάρτης Μνήμης του Ελεγκτή T6963C

Η ενσωματωμένη μνήμη της οθόνης LCD μεγέθους 8Kbytes, χωρίζεται σε τρεις περιοχές. Η πρώτη περιοχή καταλαμβάνει τα πρώτα 4Kbytes (0x0000→0x0FFF), τα οποία χρησιμοποιούνται για την αποθήκευση των δεδομένων που εμφανίζονται ως γραφικά στην οθόνη. Τα επόμενα 2Kbytes (0x1000→0x17FF) χρησιμοποιούνται για την αποθήκευση των χαρακτήρων που εμφανίζονται ως κείμενο στην οθόνη. Τα τελευταία 2Kbytes (0x1800→0x1FFF) φιλοξενούν τον πίνακα περιγραφής των raster των χαρακτήρων που δημιουργεί η ενσωματωμένη στον ελεγκτή T6963C γεννήτρια χαρακτήρων.



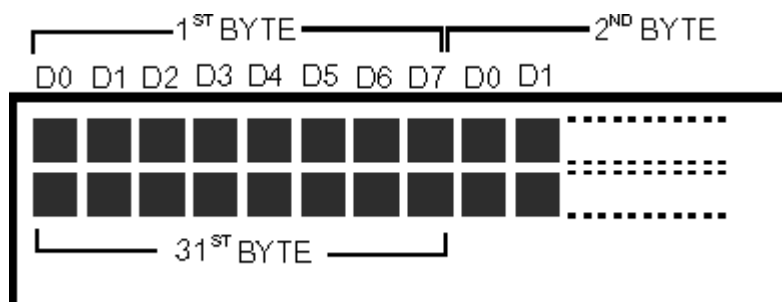
Λόγω των διαστάσεων της οθόνης σε εικονοστοιχεία (pixels), μια πλήρης οθόνη γεμάτη γραφικά απαιτεί συνολικά:

$$\frac{240 \text{ pixels}}{8 \text{ bit}} (\text{Horizontal}) \times 128 (\text{Vertical}) = 30 \text{ bytes} \times 128 \text{ rasters} = 3.840 \text{ bytes}$$



Κάθε Byte εμφανίζει 8pixels.

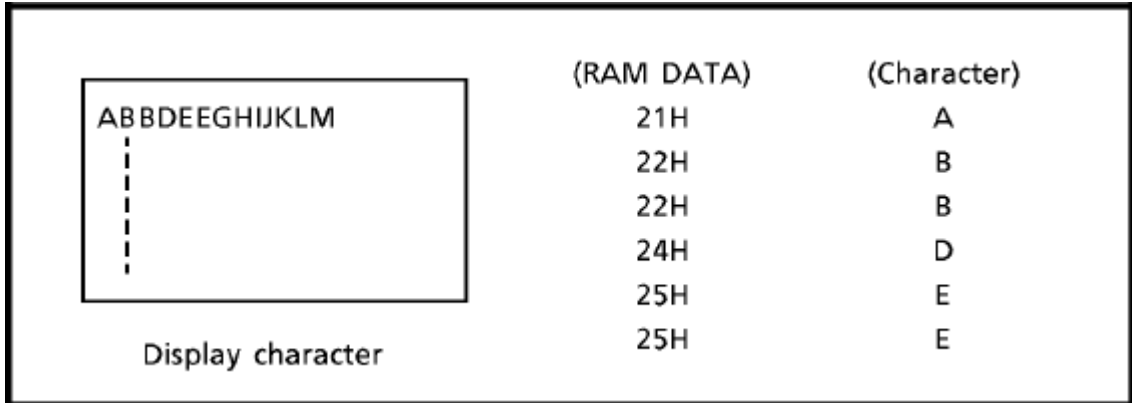
Διαδοχικές διευθύνσεις αντιστοιχούν σε γειτονικές ομάδες των 8pixels.  
Ανά 30 θέσεις μνήμης υπάρχει αλλαγή raster της οθόνης.



Για την εμφάνιση μια οθόνης γεμάτη χαρακτήρες (κείμενο), όπου η μήτρα χαρακτήρα είναι 8x8 pixels, απαιτούνται συνολικά:

$$\frac{240 \text{ pixels}}{8 \text{ bit}} (\text{Horizontal}) \times \frac{128 \text{ pixels}}{8} (\text{Vertical}) = 30 \text{ bytes} \times 16 \text{ γραμμές} = 480 \text{ bytes}$$

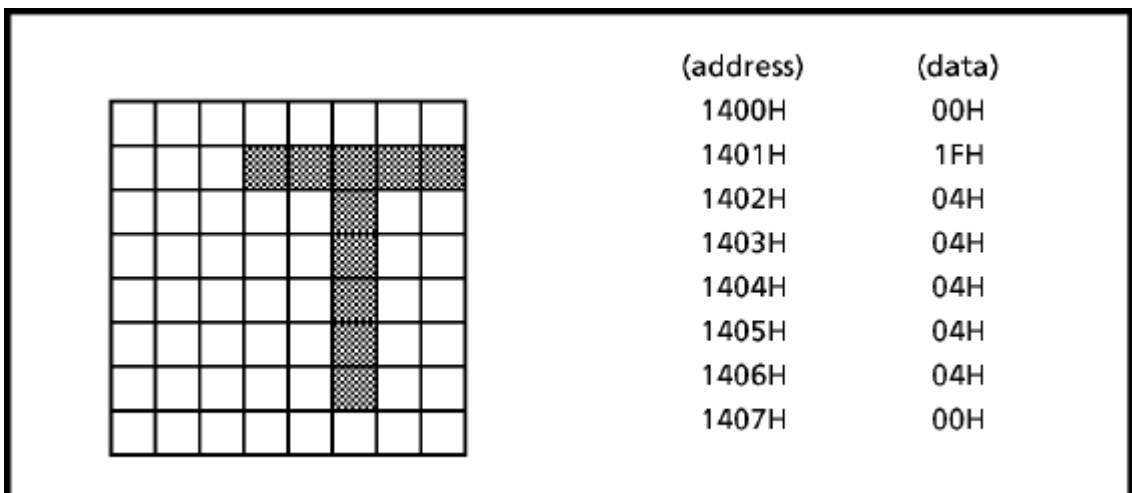
Κάθε Byte εμφανίζει ένα χαρακτήρα.



Διαδοχικές διευθύνσεις εμφανίζουν γειτονικούς χαρακτήρες κειμένου.  
Ανά 30 θέσεις μνήμης υπάρχει αλλαγή γραμμής κειμένου.

Για την αποθήκευση ενός σετ 256 χαρακτήρων για μήτρα χαρακτήρα 8x8 απαιτείται συνολικά:

$$256 \text{ χαρακτήρες} \times \frac{8 \text{ γραμμές}}{\text{χαρακτήρα}} \times \frac{1 \text{ byte}}{\text{γραμμή}} = 2048 \text{ bytes}$$



Ένα μπλοκ 8Bytes στοιχειοθετεί ένα χαρακτήρα.

---

---

## Η Γεννήτρια Χαρακτήρων

Το ενσωματωμένο σετ χαρακτήρων του ελεγκτή T6963C είναι αρκετά φτωχό καθώς προσφέρει μόνο το βασικό σετ των λατινικών χαρακτήρων ενώ δεν διαθέτει αρκετά χρησιμοποιήσιμα ειδικά σύμβολα. Ο ελεγκτής αυτός όμως δίδει την δυνατότητα χρήσης μιας ενσωματωμένης γεννήτριας χαρακτήρων για την οπτική απόδοση χαρακτήρων δικής μας σχεδίασης, που θα καλύπτουν τις ανάγκες μας.

Η γεννήτρια χαρακτήρων του ελεγκτή T6963C τροφοδοτείται από ένα πίνακα τιμών που περιγράφουν σε επίπεδο pixel το κάθε raster των χαρακτήρων της κωδικοσελίδας 437. Οι λόγοι που οδήγησαν στην επιλογή αυτής της κωδικοσελίδας είναι πολύ απλοί: εντός αυτής περιέχεται σημαντικός αριθμός ειδικών εκτυπώσιμων χαρακτήρων που χρησιμοποιήθηκαν για την δημιουργία παραθύρων και γραμμών στο Interface του τερματικού, ενώ επίσης υπάρχει πλήρες το ελληνικό αλφάβητο.

Ο πίνακας δεδομένων περιγραφής της κωδικοσελίδας 437 αποσπάστηκε από μια μνήμη EPROM κάρτας γραφικών παλαιότερης τεχνολογίας (Monochrome ή Hercules) με την χρήση προγραμματιστή-αναγνώστη ολοκληρωμένων μνήμης. Προτιμήθηκε αυτή η μέθοδος από την περίπτωση επανασχεδίασης του σετ χαρακτήρων καθώς οι θέσεις των χαρακτήρων και η μορφή τους είναι τυποποιημένες εδώ και πολλά χρόνια από την εποχή της κυριαρχίας του DOS, θα απαιτείτο ελάχιστος χρόνος σχεδίασης και θα αποφεύγονταν οποιαδήποτε σφάλματα που θα οδηγούσαν σε κακή εμφάνιση επάνω στην οθόνη LCD, επίσης η εισαγωγή τους κατά την φάση της συγγραφής του software του τερματικού (η οποία έγινε σε DOS) δεν θα απαιτούσε ιδιαίτερες προσπάθειες.

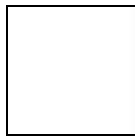


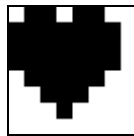
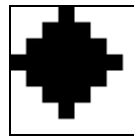
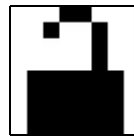
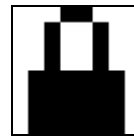
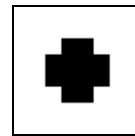

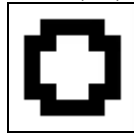

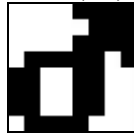
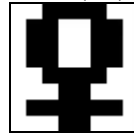


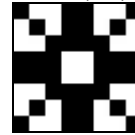
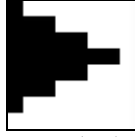
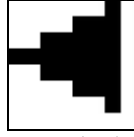
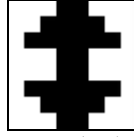
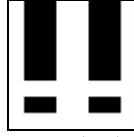
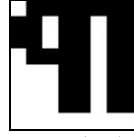
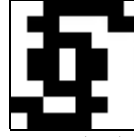
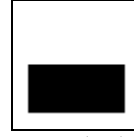
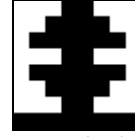
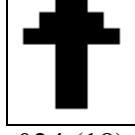
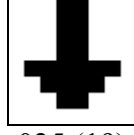
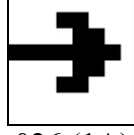
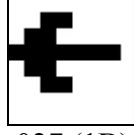
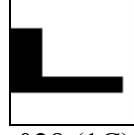
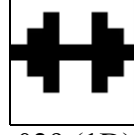
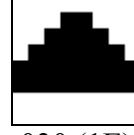
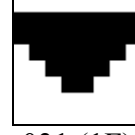
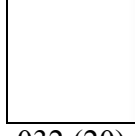
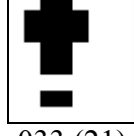
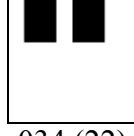
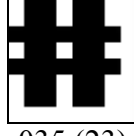
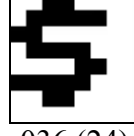
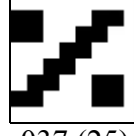
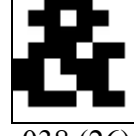
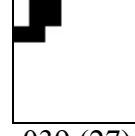
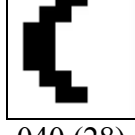
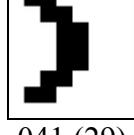
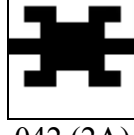
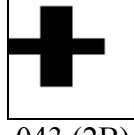
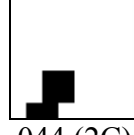


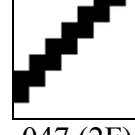
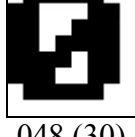
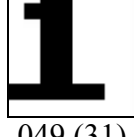
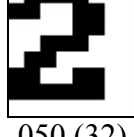
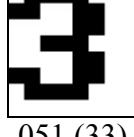
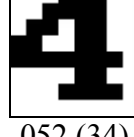
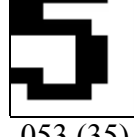
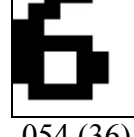
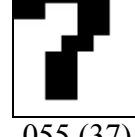
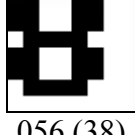
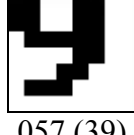
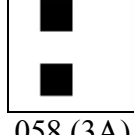
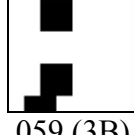
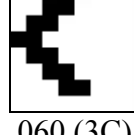
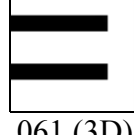
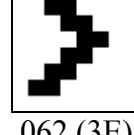
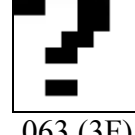
Οι χαρακτήρες 005 και 006 τροποποιήθηκαν από το αρχικό τους περιεχόμενο ώστε να εμφανίζουν το λουκέτο (ανοικτό – κλειστό) που χρησιμοποιείται για την ένδειξη της κατάστασης της κλειδαριάς. Προτιμήθηκε αυτή η ιδέα καθώς σε άλλη περίπτωση θα έπρεπε να γίνει σχεδιασμός των συμβόλων σε μορφή Bitmap και η εμφάνιση τους στην οθόνη θα απαιτούσε από την πλευρά του software του τερματικού επιπρόσθετες ρουτίνες εκτύπωσης γραφικών. Επίσης τροποποιήθηκε ο χαρακτήρας 235 ώστε να δείχνει τον τονισμένο χαρακτήρα Ε (Έ).

Επειδή το λειτουργικό σύστημα Microsoft Windows κάνει χρήση διαφορετικού προτύπου απόδοσης των ελληνικών χαρακτήρων (κωδικοσελίδα 928), προκειμένου να εμφανίζονται χωρίς προβλήματα τα ελληνικά κείμενα στην οθόνη του τερματικού, προτού αυτά αποσταλούν στο τερματικό θα πρέπει να γίνει αντιστοίχιση των ελληνικών χαρακτήρων μεταξύ των δυο κωδικοσελίδων. Τέτοιο πρόβλημα δεν υφίσταται για τα κείμενα που κάνουν χρήση του λατινικού αλφάβητου. Πίνακας αντιστοίχισης των δυο κωδικοσελίδων περιλαμβάνεται στο αρχείο **928-437.XLT** που συνοδεύει την εφαρμογή διαχείρισης του τερματικού.


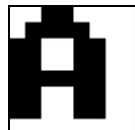

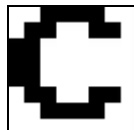



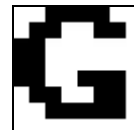
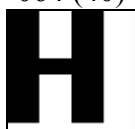
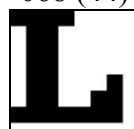
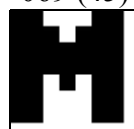
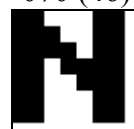
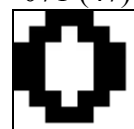
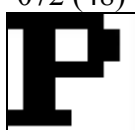
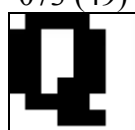
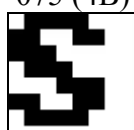
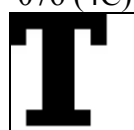
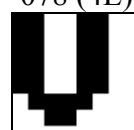
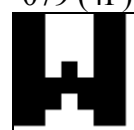
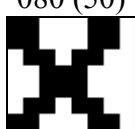
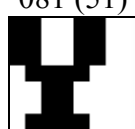
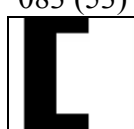
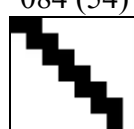
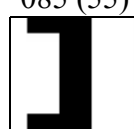
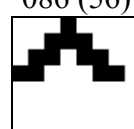
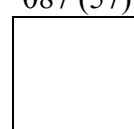
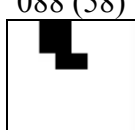
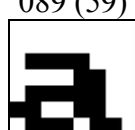
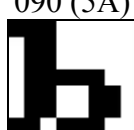
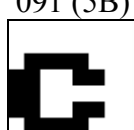
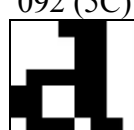
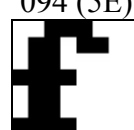
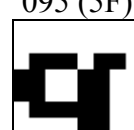
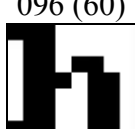
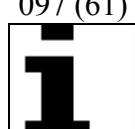
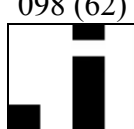
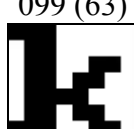
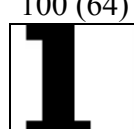
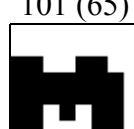
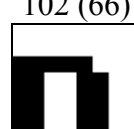
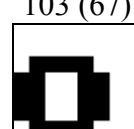
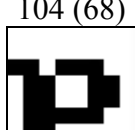
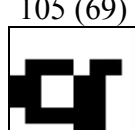
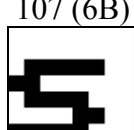
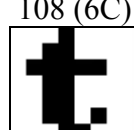
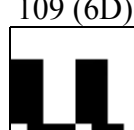
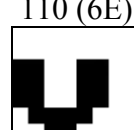
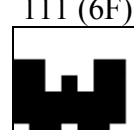
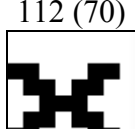
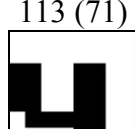
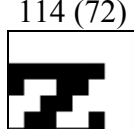
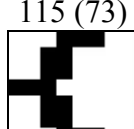
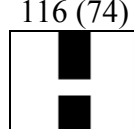
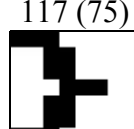
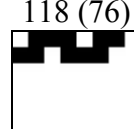
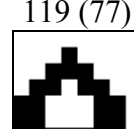
Στις επόμενες σελίδες ακολουθούν πίνακες που εμφανίζουν ολόκληρο το σετ χαρακτήρων της κωδικοσελίδας 437 σε μορφή bitmap για λόγους αναφοράς και επισκόπησης, καθώς και τον αντίστοιχο αριθμό θέσης σε δύο αριθμητικά συστήματα (δεκαδικό και δεκαεξαδικό). Οι πίνακες δεδομένων περιγραφής των raster του κάθε χαρακτήρα της κωδικοσελίδας 437 περιέχονται στο αρχείο **ASCII437.ASM**.



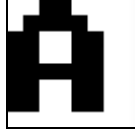


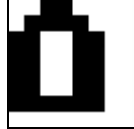

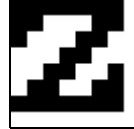

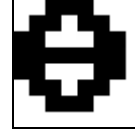

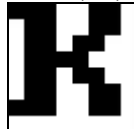
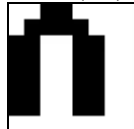



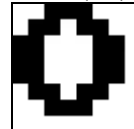


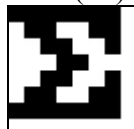
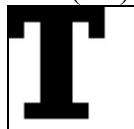
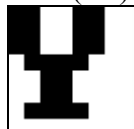
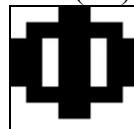
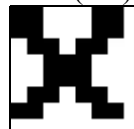
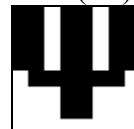
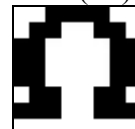
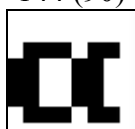
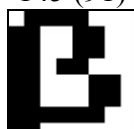
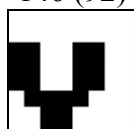
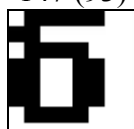
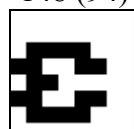
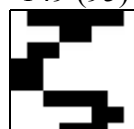
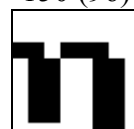
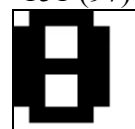
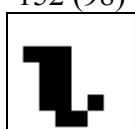
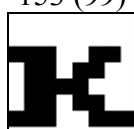
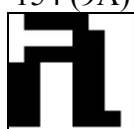
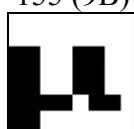
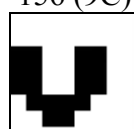
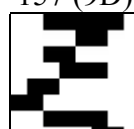
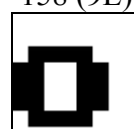
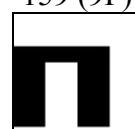
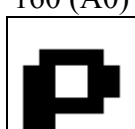
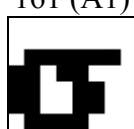
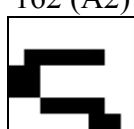
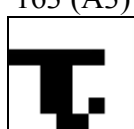
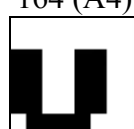
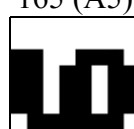
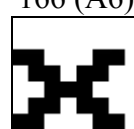
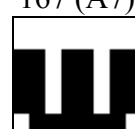
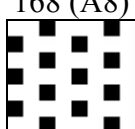
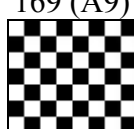
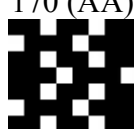

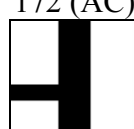
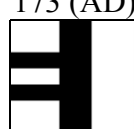

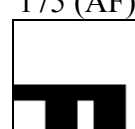
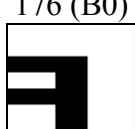
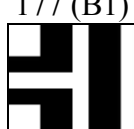
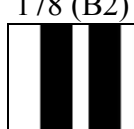
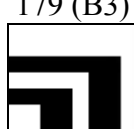
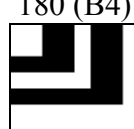
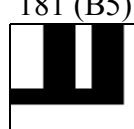
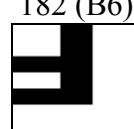
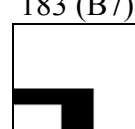
437 ASCII CHARACTER SET 0-63

							
000 (00)	001 (01)	002 (02)	003 (03)	004 (04)	005 (05)	006 (06)	007 (07)
							
008 (08)	009 (09)	010 (0A)	011 (0B)	012 (0C)	013 (0D)	014 (0E)	015 (0F)
							
016 (10)	017 (11)	018 (12)	019 (13)	020 (14)	021 (15)	022 (16)	023 (17)
							
024 (18)	025 (19)	026 (1A)	027 (1B)	028 (1C)	029 (1D)	030 (1E)	031 (1F)
							
032 (20)	033 (21)	034 (22)	035 (23)	036 (24)	037 (25)	038 (26)	039 (27)
							
040 (28)	041 (29)	042 (2A)	043 (2B)	044 (2C)	045 (2D)	046 (2E)	047 (2F)
							
048 (30)	049 (31)	050 (32)	051 (33)	052 (34)	053 (35)	054 (36)	055 (37)
							
056 (38)	057 (39)	058 (3A)	059 (3B)	060 (3C)	061 (3D)	062 (3E)	063 (3F)

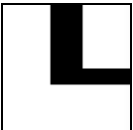
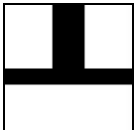
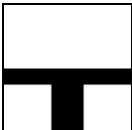
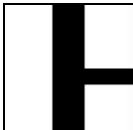

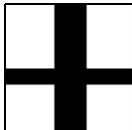


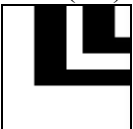

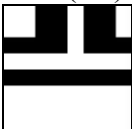
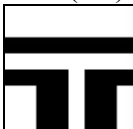

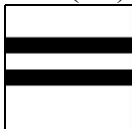
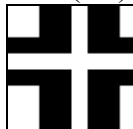
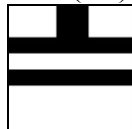
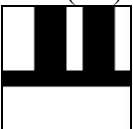
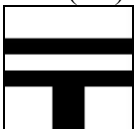
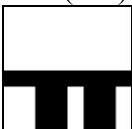




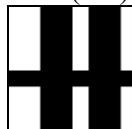
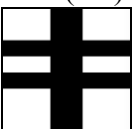
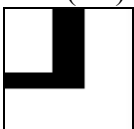
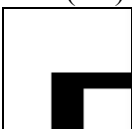
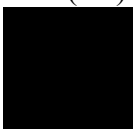

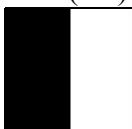
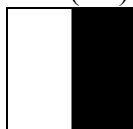

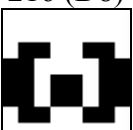
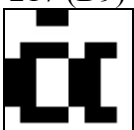
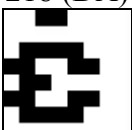
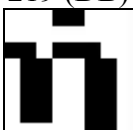
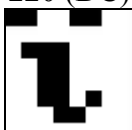
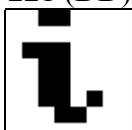
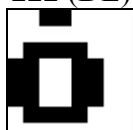
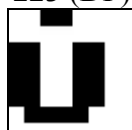
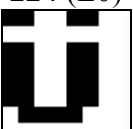
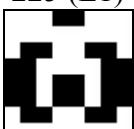
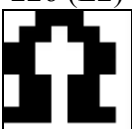
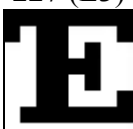
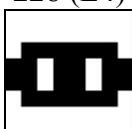
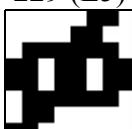
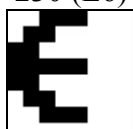
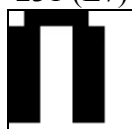
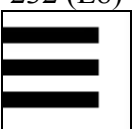
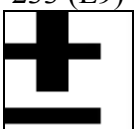
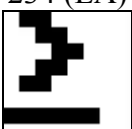
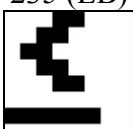
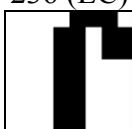
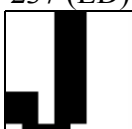
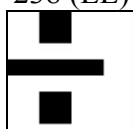
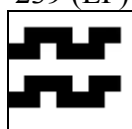
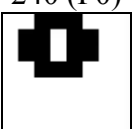
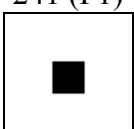
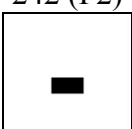
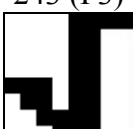
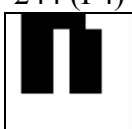
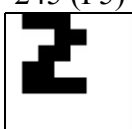
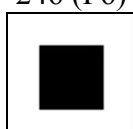
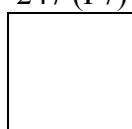
437 ASCII CHARACTER SET 64-127

							
064 (40)	065 (41)	066 (42)	067 (43)	068 (44)	069 (45)	070 (46)	071 (47)
							
072 (48)	073 (49)	074 (4A)	075 (4B)	076 (4C)	077 (4D)	078 (4E)	079 (4F)
							
080 (50)	081 (51)	082 (52)	083 (53)	084 (54)	085 (55)	086 (56)	087 (57)
							
088 (58)	089 (59)	090 (5A)	091 (5B)	092 (5C)	093 (5D)	094 (5E)	095 (5F)
							
096 (60)	097 (61)	098 (62)	099 (63)	100 (64)	101 (65)	102 (66)	103 (67)
							
104 (68)	105 (69)	106 (6A)	107 (6B)	108 (6C)	109 (6D)	110 (6E)	111 (6F)
							
112 (70)	113 (71)	114 (72)	115 (73)	116 (74)	117 (75)	118 (76)	119 (77)
							
120 (78)	121 (79)	122 (7A)	123 (7B)	124 (7C)	125 (7D)	126 (7E)	127 (7F)

437 ASCII CHARACTER SET 128-191

							
128 (80)	129 (81)	130 (82)	131 (83)	132 (84)	133 (85)	134 (86)	135 (87)
							
136 (88)	137 (89)	138 (8A)	139 (8B)	140 (8C)	141 (8D)	142 (8E)	143 (8F)
							
144 (90)	145 (91)	146 (92)	147 (93)	148 (94)	149 (95)	150 (96)	151 (97)
							
152 (98)	153 (99)	154 (9A)	155 (9B)	156 (9C)	157 (9D)	158 (9E)	159 (9F)
							
160 (A0)	161 (A1)	162 (A2)	163 (A3)	164 (A4)	165 (A5)	166 (A6)	167 (A7)
							
168 (A8)	169 (A9)	170 (AA)	171 (AB)	172 (AC)	173 (AD)	174 (AE)	175 (AF)
							
176 (B0)	177 (B1)	178 (B2)	179 (B3)	180 (B4)	181 (B5)	182 (B6)	183 (B7)
							
184 (B8)	185 (B9)	186 (BA)	187 (BB)	188 (BC)	189 (BD)	190 (BE)	191 (BF)

437 ASCII CHARACTER SET 192-255

							
192 (C0)	193 (C1)	194 (C2)	195 (C3)	196 (C4)	197 (C5)	198 (C6)	199 (C7)
							
200 (C8)	201 (C9)	202 (CA)	203 (CB)	204 (CC)	205 (CD)	206 (CE)	207 (CF)
							
208 (D0)	209 (D1)	210 (D2)	211 (D3)	212 (D4)	213 (D5)	214 (D6)	215 (D7)
							
216 (D8)	217 (D9)	218 (DA)	219 (DB)	220 (DC)	221 (DD)	222 (DE)	223 (DF)
							
224 (E0)	225 (E1)	226 (E2)	227 (E3)	228 (E4)	229 (E5)	230 (E6)	231 (E7)
							
232 (E8)	233 (E9)	234 (EA)	235 (EB)	236 (EC)	237 (ED)	238 (EE)	239 (EF)
							
240 (F0)	241 (F1)	242 (F2)	243 (F3)	244 (F4)	245 (F5)	246 (F6)	247 (F7)
							
248 (F8)	249 (F9)	250 (FA)	251 (FB)	252 (FC)	253 (FD)	254 (FE)	255 (FF)

## Το Πολύ Πραγματικού Χρόνου (RTC – Real Time Clock)

Το χρησιμοποιούμενο RTC DS12C887 της Dallas Semiconductors, αποτελεί ένα ολοκληρωμένο σύστημα καταγραφής χρόνου. Διατίθεται σε μορφή Encapsulated DIP 24 ακροδεκτών. Εντός αυτού περιέχονται ο πυρήνας του RTC, 113bytes μνήμης RAM, ένα στοιχείο λιθίου 3V και ο κρύσταλλος χρονοισμού 32,768KHz.

Η μνήμη RAM με την χρήση του στοιχείου λιθίου μετατρέπεται σε NVRAM. Το στοιχείο λιθίου επιτρέπει την καταγραφή του χρόνου ακόμα και όταν δεν εφαρμόζεται τροφοδοσία στο κύκλωμα του τερματικού. Έχει διάρκεια ζωής περίπου 10χρόνια σύμφωνα με την εταιρία κατασκευής, όταν το εξάρτημα βρίσκεται εκτός τροφοδοσίας, όπως ακριβώς και στην περίπτωση του μικροελεγκτή.

Τα 113bytes ενσωματωμένης μνήμης RAM χρησιμοποιούνται από το Software του τερματικού για την αποθήκευση σημαντικών παραμέτρων λειτουργίας αυτού.

Το ολοκληρωμένο RTC περιέχει 15 καταχωρητές απαρίθμησης χρόνου και ελέγχου, απαιτώντας έτσι αθροιστικά με τις θέσεις μνήμης RAM, ένα σύνολο 128 θέσεων-διευθύνσεων. Είναι συμβατό με τα παλαιότερου τύπου ολοκληρωμένα RTC για προσωπικούς υπολογιστές. Ο χάρτης των καταχωρητών παρατίθεται στον επόμενο πίνακα.

### PIN ASSIGNMENT

MOT	1	24	V <sub>cc</sub>
NC	2	23	SQW
NC	3	22	NC
AD0	4	21	NC
AD1	5	20	NC
AD2	6	19	IRQ
AD3	7	18	RESET
AD4	8	17	DS
AD5	9	16	NC
AD6	10	15	R/W
AD7	11	14	AS
GND	12	13	CS

DS12C887 24-Pin  
ENCAPSULATED PACKAGE

### PIN DESCRIPTION

- AD0-AD7 - Multiplexed Address/Data Bus
- NC - No Connect
- MOT - Bus Type Selection
- CS - RTC Chip Select Input
- AS - Address Strobe
- R/W - Read/Write Input
- DS - Data Strobe
- RESET - Reset Input
- IRQ - Interrupt Request Output
- SQW - Square Wave Output
- V<sub>cc</sub> - +5 Volt Main Supply
- GND - Ground

### TIME, CALENDAR AND ALARM DATA MODES

ADDRESS LOCATION	FUNCTION	DECIMAL RANGE	RANGE	
			BINARY DATA MODE	BCD DATA MODE
0	Seconds	0-59	00-3B	00-59
1	Seconds Alarm	0-59	00-3B	00-59
2	Minutes	0-59	00-3B	00-59
3	Minutes Alarm	0-59	00-3B	00-59
4	Hours 12-hr, Mode	1-12	01-0C AM, 01-9C PM	01-12 AM, 01-92 PM
	Hours 24-hr, Mode	0-23	00-17	00-23
5	Hours Alarm 12-hr, Mode	1-12	01-0C AM, 01-9C PM	01-12 AM, 01-92 PM
	Hours Alarm 24-hr, Mode	0-23	00-17	00-23
6	Day of the week Sunday=1	1-7	01-07	01-07
7	Date of Month	1-31	01-1F	01-31
8	Month	1-12	01-0C	01-12
9	Year	0-99	00-63	00-99
50	Century	0-99	NA	19,20

---

---

Το RTC συνδέεται με τον μικροελεγκτή μέσω ενός χρονικά πολυπλεγμένου Address/Data Bus μήκους 8bit τύπου αρχιτεκτονικής διαύλου Intel. Διαθέτει ενσωματωμένο κύκλωμα διαχωρισμού Address/Data Bus ελεγχόμενο από τον ακροδέκτη ALE του μικροελεγκτή. Συνεπώς δεν απαιτεί την ύπαρξη κάποιου εξωτερικού ολοκληρωμένου διαχωρισμού (74HC373 ή 74HC573), μειώνοντας έτσι τον αριθμό των απαιτούμενων ακροδεκτών της συσκευασίας του για σύνδεση σε ένα μικροϋπολογιστικό σύστημα.

Οι καταχωρητές του RTC μπορούν να προσπελαστούν στις διευθύνσεις 0xFB00→0xFB7F.

Κύριο πλεονέκτημα του, απέναντι στο ενσωματωμένο RTC του μικροελεγκτή, και λόγος που οδήγησε στην επιλογή του, είναι η δυνατότητα παραγωγής παλμών πρόκλησης διακοπής κάθε φορά που γίνεται ενημέρωση των καταχωρητών καταμέτρησης χρόνου (δηλαδή ανά δευτερόλεπτο). Επίσης δίνει την δυνατότητα προγραμματισμού συγκεκριμένης ημερομηνίας και ώρας για πρόκληση διακοπής. Η τελευταία δυνατότητα δεν χρησιμοποιήθηκε από το software του τερματικού, καθώς το Interface αυτού, αναλαμβάνει να δείχνει την τρέχουσα ώρα και ημερομηνία σε ανάλυση δευτερολέπτου. Τελευταίο σημαντικό χαρακτηριστικό, το οποίο όμως δεν έτυχε χρήσης, είναι η δυνατότητα παραγωγής τετραγωνικών παλμών προγραμματιζόμενης συχνότητας από τον ακροδέκτη SQW του ολοκληρωμένου.

Περισσότερες πληροφορίες προγραμματισμού και ελέγχου του RTC DS12C887 βρίσκονται στο συνοδευτικό φυλλάδιο. Οι ρουτίνες που διαχειρίζονται το RTC βρίσκονται στο αρχείο **RTC.C**.

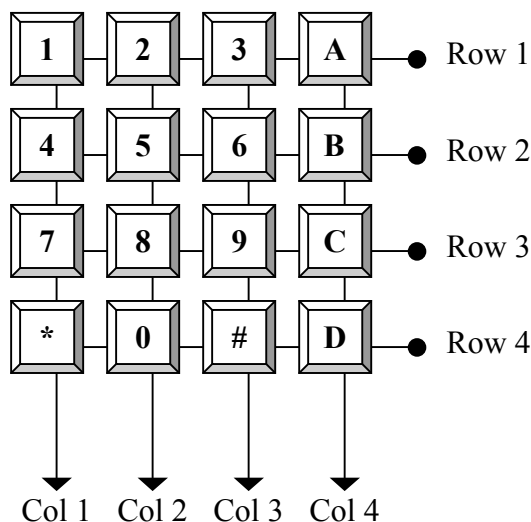
---

## Το πληκτρολόγιο

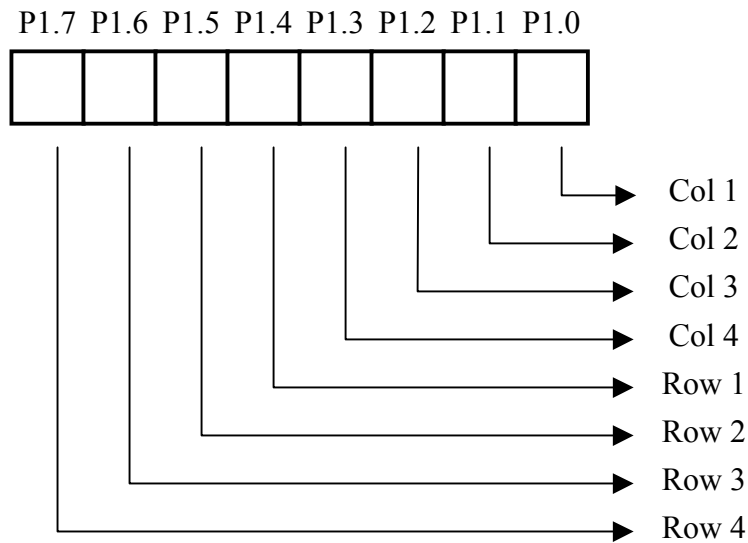
Το χρησιμοποιηθέν πληκτρολόγιο για την εισαγωγή πληροφοριών αλλά και για την πλοήγηση εντός του Interface χρήσης του τερματικού ακολουθεί την μορφή του κλασσικού πληκτρολογίου μήτρας 4x4. Για τον λόγο αυτό η αναγνώριση πατήματος πλήκτρου απαιτεί σαφώς καθορισμένα βήματα και τεχνικές.



Η ανάγνωση της τιμής του πατημένου πλήκτρου γίνεται με την τεχνική σάρωσης μήτρας. Καθώς πρόκειται για πληκτρολόγιο μήτρας 4x4 απαιτούνται συνολικά 8 γραμμές για την σάρωση και ανάγνωση της καταστάσεως των διακοπών-πλήκτρων. Οι γραμμές αυτές στο κύκλωμα του τερματικού δίδονται από την 8bit πόρτα I/O P1 του μικροελεγκτή. Επειδή οι πόρτες του μικροελεγκτή μπορούν να εισάγουν στοιχεία αν εγγράψουμε στον ακροδέκτη που θέλουμε να συμπεριφερθεί ως είσοδος το λογικό '1', κρίνεται απαραίτητη η σάρωση της μήτρας του πληκτρολογίου με κινούμενο λογικό '0'. Αυτό ανιχνεύεται εύκολα όταν πατηθεί πλήκτρο καθώς περνάει από του ακροδέκτες που έχουμε γράψει λογικό '0' για έξοδο, στον ακροδέκτη που έχει ορισθεί ως είσοδος (γράφοντας σε αυτόν το λογικό '1'), απορροφώντας ρεύμα από αυτόν και εμφανίζοντας έτσι χαμηλό δυναμικό όταν διαβασθεί η κατάσταση του.



Η φυσική σύνδεση του πληκτρολογίου μήτρας με τους ακροδέκτες της πόρτας P1 του μικροελεγκτή εμφανίζεται στο επόμενο σχήμα.



Για την σάρωση των γραμμών πρέπει να αποσταλούν οι παρακάτω κωδικοί στην πόρτα P1. Στους ακροδέκτες των στηλών παρατηρούμε ότι γράφεται το λογικό '1' για να συμπεριφερθούν σαν είσοδοι, ενώ στους ακροδέκτες γραμμής ολισθαίνει το λογικό '0', που πρέπει να ανιχνεύσουμε.

Κωδικοί Σάρωσης Γραμμών Πληκτρολογίου ScanPattern									
Γραμμή	R4	R3	R2	R1	C4	C3	C2	C1	Scan Code (Hex)
	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
1 <sup>η</sup>	1	1	1	0	1	1	1	1	0xFF
2 <sup>η</sup>	1	1	0	1	1	1	1	1	0xDF
3 <sup>η</sup>	1	0	1	1	1	1	1	1	0xBF
4 <sup>η</sup>	0	1	1	1	1	1	1	1	0x7F

Διαβάζοντας τα περιεχόμενα της πόρτας P1, χρησιμοποιούμε ως μάσκα το byte 0x0F με λογική πράξη AND για να αποκρύψουμε την πληροφορία των γραμμών σάρωσης. Αν ανακαλύψουμε κάποιο λογικό '0' στα ψηφία των στηλών σημαίνει ότι έχει πατηθεί σε αυτή την στήλη ένα πλήκτρο. Αν δεν υπάρχει κάποιο λογικό '0' σημαίνει ότι δεν έχει πατηθεί κάποιο πλήκτρο.

Κωδικοί Επιστροφής Στηλών Πληκτρολογίου ReturnCode									
Πλήκτρο σε Στήλη	R4	R3	R2	R1	C4	C3	C2	C1	Scan Code (Hex)
	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
Ηρεμία	0	0	0	0	1	1	1	1	0x0F
1 <sup>η</sup> Στήλη	0	0	0	0	1	1	1	0	0x0E
2 <sup>η</sup> Στήλη	0	0	0	0	1	1	0	1	0x0D
3 <sup>η</sup> Στήλη	0	0	0	0	1	0	1	1	0x0B
4 <sup>η</sup> Στήλη	0	0	0	0	0	1	1	1	0x07

Οι πληροφορίες που μας δίνουν οι πίνακες αυτοί, συνδυάζονται με μια τεχνική πίνακα αναζήτησης και αντιστοίχισης, από τον οποίο μπορούμε να αποκωδικοποιήσουμε τις συντεταγμένες του πιεσμένου πλήκτρου σε οποιοδήποτε σύστημα κωδικοποίησης χρειάζεται να χειριστούμε.



---

Για το κύκλωμα του τερματικού, και για λόγους ευκολίας προγραμματισμού, έχει προτιμηθεί η άμεση μετατροπή σε κωδικό ASCII του συμβόλου που αναγράφει το πλήκτρο που πατήθηκε.

<b>Πίνακας Αναζήτησης &amp; Αντιστοίχισης ScanCode</b>				
<b>Στήλες</b>	<b>Γραμμές</b>			
	<b>1<sup>η</sup> Γραμμή</b>	<b>2<sup>η</sup> Γραμμή</b>	<b>3<sup>η</sup> Γραμμή</b>	<b>4<sup>η</sup> Γραμμή</b>
<b>1<sup>η</sup> Στήλη</b>	<b>1</b>	<b>4</b>	<b>7</b>	<b>*</b>
<b>2<sup>η</sup> Στήλη</b>	<b>2</b>	<b>5</b>	<b>8</b>	<b>0</b>
<b>3<sup>η</sup> Στήλη</b>	<b>3</b>	<b>6</b>	<b>9</b>	<b>#</b>
<b>4<sup>η</sup> Στήλη</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>

Τα πλήκτρα \* (αστερίσκος) και # (δίεση), χρησιμοποιούνται για τις λειτουργίες Εξόδου (ESCAPE) και Αποδοχής (ENTER ή SELECT) αντίστοιχα. Για την πλοήγηση μέσα στις λειτουργίες του τερματικού έχουν επιλεγθεί τα πλήκτρα C (ΠΡΟΗΓΟΥΜΕΝΟ) και D (ΕΠΟΜΕΝΟ). Το πλήκτρο A χρησιμοποιείται για την παροχή Βοηθείας (F1, HELP) ενώ το πλήκτρο B ενεργοποιεί την εμφάνιση των στοιχείων της πτυχιακής εργασίας. Τα δυο αυτά τελευταία πλήκτρα (A και B) παρέχουν επιπλέον από μια λειτουργία όταν απαιτείται η εισαγωγή του προσωπικού κωδικού του κατόχου κλειδιού ασφαλείας για την παροχή πρόσβασης, εφόσον όμως πατηθούν αντί του πρώτου αριθμού του προσωπικού κωδικού. Συγκεκριμένα το πλήκτρο A θα προκαλέσει αίτηση κλειδώματος της πόρτας από τον κάτοχο, ενώ το πλήκτρο B θα προσθέσει το αναγνωρισμένο κλειδί ασφαλείας στην βάση διαθέσιμων κλειδιών.

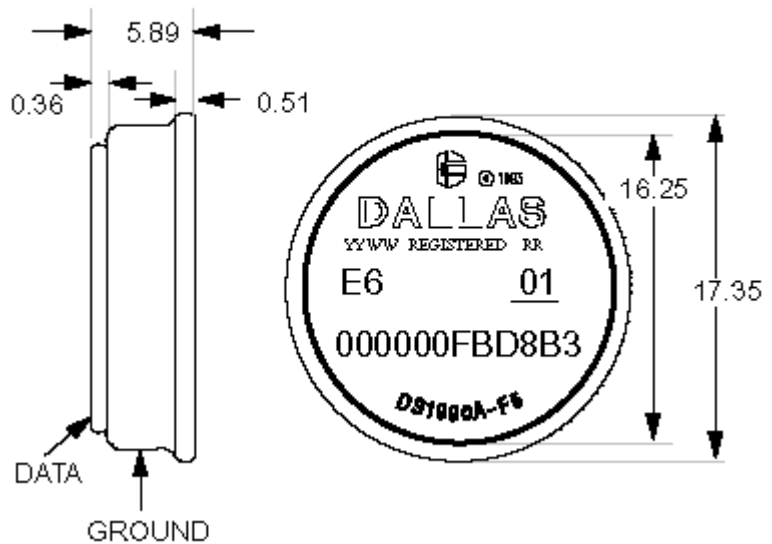
Οι ρουτίνες που σαρώνουν το πληκτρολόγιο, ανιχνεύουν το πλήκτρο και αντιστοιχούν τις συντεταγμένες του στην μήτρα 4x4 σε κώδικα ASCII, συμπεριλαμβάνονται στο αρχείο **KEYB.C**. Επίσης συμπεριλαμβάνονται και οι παραπάνω πίνακες.

## Το Κλειδί Ασφαλούς Πρόσβασης

Για την εξασφάλιση της ασφαλούς παροχής πρόσβασης στον ελεγχόμενο χώρο επιλέχθηκε η χρήση της σειράς iButton της εταιρίας Dallas Semiconductors. Συγκεκριμένα επιλέχθηκε το εξάρτημα με κωδικό DS1990A.

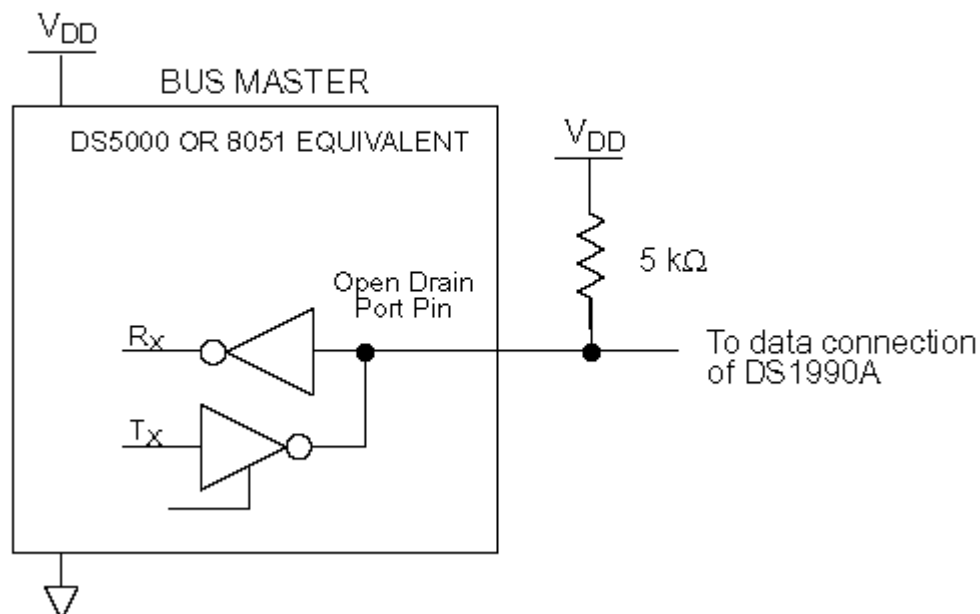
Πρόκειται για ένα ολοκληρωμένο κύκλωμα μνήμης, που περιέχει έναν μοναδικό σειριακό αριθμό μήκους 64bit επιτρεπόμενης μόνο ανάγνωσης. Η μοναδικότητα του αριθμού εξασφαλίζεται από την ίδια την κατασκευάστρια εταιρία με την χρήση εγγραφής Laser του αριθμού επάνω στο κύκλωμα μνήμης.

### F5 MICROCAN™



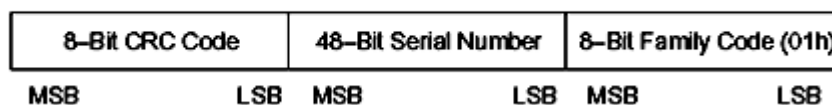
Διατίθεται σε συσκευασία F5 MICROCAN με διαστάσεις που εμφανίζονται στο παραπάνω σχήμα. Διαθέτει δύο σημεία σύνδεσης με το κύκλωμα ανάγνωσης σημειωμένα ως DATA και GROUND. Παρατηρείται ότι απουσιάζει κάποιος ακροδέκτης τροφοδοσίας. Η παροχή τροφοδοσίας καθώς και η μεταφορά δεδομένων διαχειρίζονται από έναν δίαυλο (σχεδιασμένο από την Dallas Semiconductors), ο οποίος ονομάζεται MicroLan™ και διαχειρίζεται από το πρωτόκολλο 1-Wire™. Ο δίαυλος αυτός επιτρέπει την τροφοδότηση με θετική τάση και μεταφορά δεδομένων από έναν αγωγό, ενώ απαιτείται και ένας αγωγός γείωσης. Ο αγωγός DATA μπορεί να συνδεθεί σε έναν ακροδέκτη ανοικτής εκροής ενός μικροελεγκτή σύμφωνα με την παρακάτω συνδεσμολογία. Η αντίσταση pull up είναι απαραίτητη για την τροφοδότηση του iButton με ρεύμα.

### Open Drain



Η ανάγνωση των δεδομένων επιτυγχάνεται μόνο εφόσον τηρηθούν οι χρονισμοί και οι κανόνες που επιβάλλει το πρωτόκολλο 1-Wire™. Τα δεδομένα που θα διαβασθούν από το συγκεκριμένο εξάρτημα (DS1990A) χωρίζονται σε 8bit αφιερωμένα για την αναγνώριση του τύπου (οικογένεια DS19xx) στον οποίο ανήκει το εξάρτημα (τιμή 0x01), 48bit που περιέχουν τον μοναδικό σειριακό αριθμό και ακόμα 8bit που περιέχουν την τιμή CRC (Cyclic Redundancy Check) των προηγούμενων 56bit. Εξασφαλίζεται έτσι η σωστή αναγνώριση του τύπου του εξαρτήματος αλλά και η αναγνώριση πιθανών σφαλμάτων ανάγνωσης. Το CRC Byte βασίζεται σε ένα οκτανήφιο πολυώνυμο  $x^8 + x^5 + x^4 + 1$ . Για την επιβεβαίωση της σωστής επικοινωνίας απαιτείται από την πλευρά του μικροελεγκτή να εκτελέσει το πολυώνυμο αυτό στα δεδομένα που διάβασε από το iButton και να το συγκρίνει με το αποθηκευμένο σε αυτό CRC Byte. Αν οι τιμές αυτές είναι οι ίδιες τότε η επικοινωνία έγινε χωρίς πρόκληση σφαλμάτων.

### DS1990A MEMORY MAP

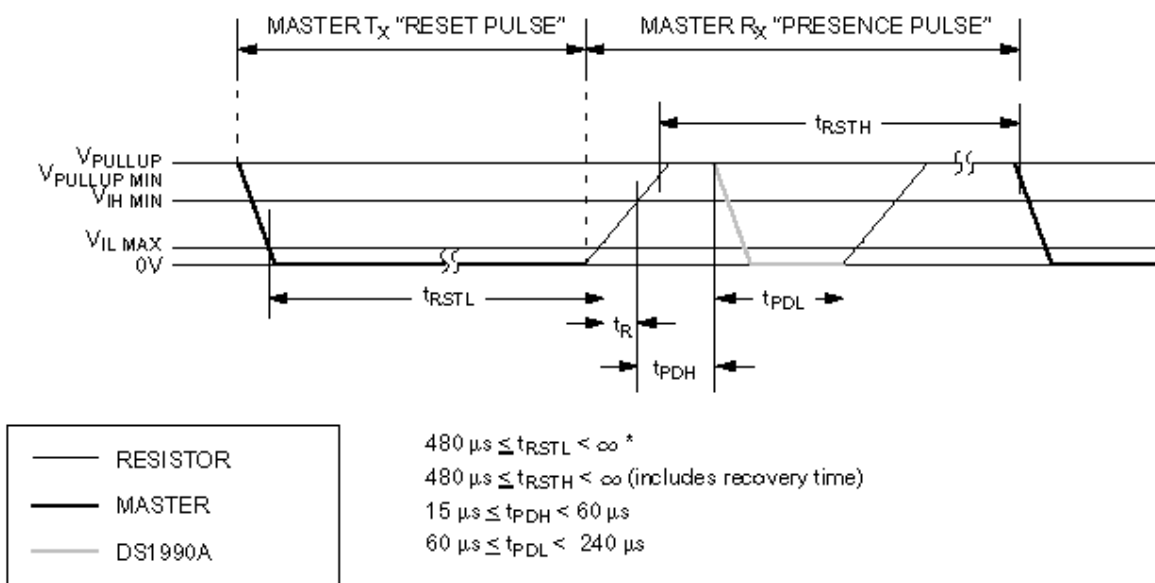


Η μεταλλική συσκευασία του εξαρτήματος προστατεύει το ολοκληρωμένο από σκόνη, μηχανική καταπόνηση, υγρασία. Λειτουργεί σε μεγάλο εύρος θερμοκρασίας περιβάλλοντος (-40°C έως +85°C). Επίσης επιτρέπει την εύκολη ανάγνωση από υποδοχή δύο επαφών, χωρίς την ανάγκη ύπαρξης ειδικού Hardware.

## Αρχικοποίηση Επικοινωνίας

Με την τοποθέτηση του iButton στην υποδοχή, αυτό τροφοδοτείται με ρεύμα από τον αγωγό DATA. Το iButton τότε παράγει έναν παλμό παρουσίας που εκδηλώνεται με πτώση τάσης. Η πρώτη εντολή που αποστέλλεται, είναι μια εντολή που εκδηλώνεται με ένα παλμό χαμηλής στάθμης (Master Reset Pulse) και επιβεβαιώνεται από το iButton με έναν ακόμα παλμό παρουσίας.

### INITIALIZATION PROCEDURE "RESET AND PRESENCE PULSES"



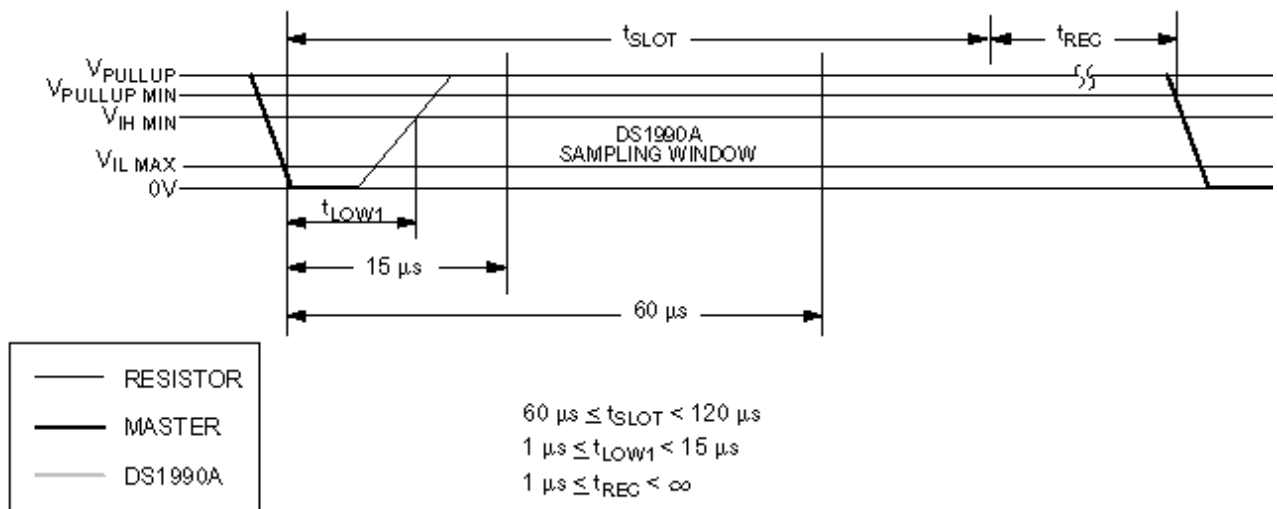
\* In order not to mask interrupt signalling by other devices on the 1-Wire bus,  $t_{RSTL} + t_R$  should always be less than 960  $\mu s$ .

## Ανάγνωση του *iButton*

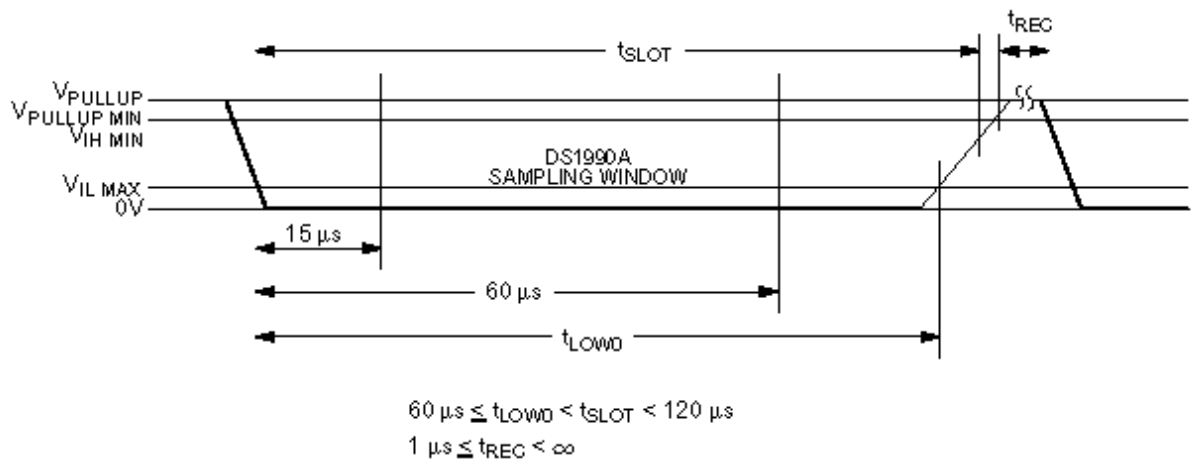
Το πρωτόκολλο 1-Wire™ προβλέπει τρεις διαφορετικούς κύκλους πρόσβασης: τον κύκλο εγγραφής λογικού '1', τον κύκλο εγγραφής λογικού '0', και τον κύκλο ανάγνωσης δυαδικού ψηφίου. Ο κάθε ένας από αυτούς καθορίζει ένα σύνολο χρονικών διαστημάτων μέσα στα οποία η ηλεκτρική στάθμη του καλωδίου σύνδεσης πρέπει να αλλάζει καταστάσεις, προκειμένου να σηματοδοτηθούν σωστά οι παραπάνω ενέργειες. Με τη λήξη του κύκλου εκκίνησης το DS1990A οδηγείται σε κατάσταση αναμονής περιμένοντας νέες εντολές. Από τη στιγμή που ολοκληρωθεί η μετάδοση μιας από αυτές, σπεύδει να την εκτελέσει επιστρέφοντας στον μικροελεγκτή τα αποτελέσματα.

### READ/WRITE TIMING DIAGRAM

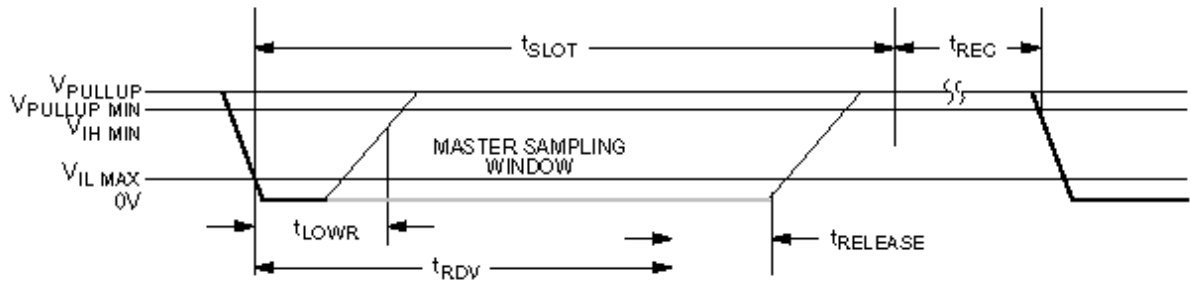
#### Write-One Time Slot



#### Write-Zero Time Slot



## Read-Data Time Slot



$$60 \mu s \leq t_{SLOT} < 120 \mu s$$

$$1 \mu s \leq t_{LOWR} < 15 \mu s$$

$$0 \leq t_{RELEASE} < 45 \mu s$$

$$1 \mu s < t_{REC} < \infty$$

$$t_{RDV} = 15 \mu s$$

Η αναγνώριση εισαγωγής και ανάγνωση του εξαρτήματος γίνεται από ρουτίνες που συμπεριλαμβάνονται στο αρχείο **IBUTTON.C**, και αποτελούν τροποποίηση των ρουτινών σε Assembly που προσφέρει η ίδια η κατασκευάστρια εταιρία. Η ανάγνωση γίνεται με απλή επαφή στην υποδοχή και ολοκληρώνεται σε λιγότερο από 5ms.



## Λόγοι Επιλογής του iButton

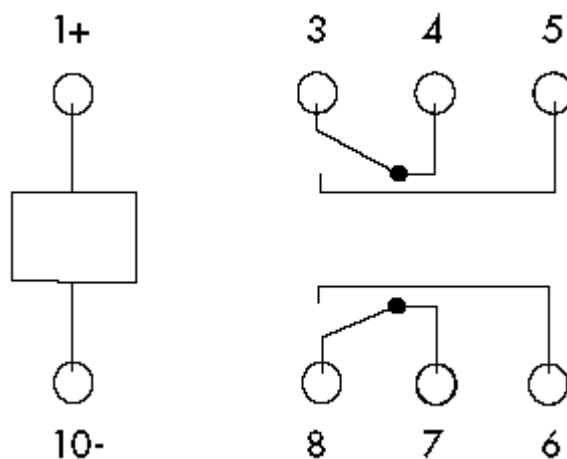
Συγκεντρωτικά τα στοιχεία που οδήγησαν στην επιλογή του συγκεκριμένου εξαρτήματος είναι:

- Μοναδικός σειριακός αριθμός 64bit.
- Δυνατότητα αναγνώρισης σφάλματος επικοινωνίας.
- Χρήση δυο καλωδίων για την επικοινωνία.
- Απλό πρωτόκολλο επικοινωνίας.
- Άμεση χρήση με τον μικροελεγκτή του τερματικού.
- Εύκολη και γρήγορη αναγνώριση και ανάγνωση του εξαρτήματος.
- Υψηλή ανθεκτικότητα σε άσχημες συνθήκες περιβάλλοντος.
- Χαμηλό κόστος απόκτησης.
- Μηδενικό κόστος συντήρησης.

Για το κύκλωμα του τερματικού έχει γίνει χρήση δυο ακροδεκτών του μικροελεγκτή, με σκοπό να ελέγχονται δυο υποδοχές για κλειδιά iButton. Μια υποδοχή βρίσκεται στο εξωτερικό μέρος του ελεγχόμενου χώρου, ενώ η άλλη στο εσωτερικό αυτού. Οι ακροδέκτες που χρησιμοποιούνται είναι οι P2.7 και P2.6 για την πρώτη και δεύτερη υποδοχή αντίστοιχα. Κατά τον τρόπο αυτό είναι δυνατός ο έλεγχος τόσο για είσοδο, όσο και για έξοδο από τον ελεγχόμενο χώρο, εφόσον κάτι τέτοιο είναι επιθυμητό.

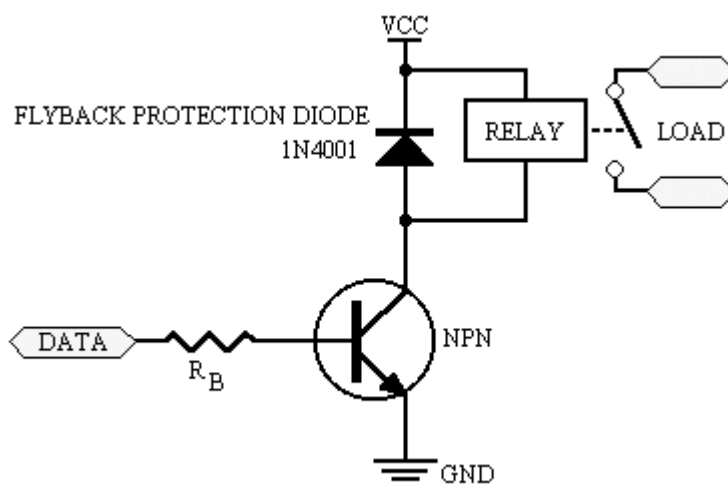
## Έλεγχος της Κλειδαριάς

Ο μηχανισμός ασφαλείας της πόρτας του ελεγχόμενου χώρου, ελέγχεται με την χρήση ενός απλού κυκλώματος με ηλεκτρονόμο (Ρελέ). Το επιλεγμένο Ρελέ είναι τύπου διπλού SPCO (Single Pole Change Over) ή αλλιώς DPDT (Double Pole Double Throw). Διαθέτει δηλαδή δυο μεταγωγικούς διακόπτες. Ο ένας μεταγωγικός διακόπτης χρησιμοποιείται για να ελέγξει την κλειδαριά, ενώ ο άλλος είναι διαθέσιμος προς χρήση για συνεργασία με κάποιο σύστημα συναγερμού.



## Υπολογισμός Εξαρτημάτων Κυκλώματος Οδήγησης Ρελέ

Το πλήρες κύκλωμα ελέγχου του ρελέ εμφανίζεται στο σχηματικό διάγραμμα του κυκλώματος του τερματικού. Παρακάτω ακολουθεί ένα απλοποιημένο κύκλωμα που χρησιμοποιείται στην ανάλυση των παραμέτρων για τον προσδιορισμό των τιμών των εξαρτημάτων αυτού.



Το ρελέ που χρησιμοποιείται στο κύκλωμα του τερματικού είναι το FX2 D3209 της AXICOM. Η αντίσταση του πηνίου του σύμφωνα με το φυλλάδιο του κατασκευαστή είναι  $178\Omega$ . Για να μετάγει τις επαφές του, πρέπει να το διαρρεύσει ρεύμα  $I_C = \frac{V_{CC}}{R_{Relay}} = \frac{5V}{178\Omega} \cong 28mA$ . Έχοντας

επιλέξει ως τρανζίστορ το BC547A με ελάχιστο λόγο ενίσχυσης  $hfe = 110$  και γνωρίζοντας το ρεύμα συλλέκτη υπολογίζουμε το απαιτούμενο ρεύμα βάσης.  $I_B = \frac{I_C}{hfe} = \frac{28mA}{110} \cong 0,255mA$ .

Υποθέτοντας ότι ο ακροδέκτης του μικροελεγκτή αποδίδει κατ' ελάχιστο 3,5V στην έξοδο του για λογική κατάσταση '1' υπολογίζουμε την τιμή της αντίστασης βάσεως αφαιρώντας την πτώση τάσεως  $V_{BE}$  μεταξύ βάσης και εκπομπού,  $R_B = \frac{V_{in} - V_{BE}}{I_B} = \frac{3,5V - 0,65V}{0,255mA} \cong 11K\Omega$ . Επιλέγουμε την αμέσως χαμηλότερη τιμή αντίστασης που μπορεί να βρεθεί στο εμπόριο και έχει τιμή 10KΩ.

## Επεξήγηση Λειτουργίας Κυκλώματος Ελέγχου Κλειδαριάς

Σκοπός της λειτουργίας του κυκλώματος είναι να ξεκλειδώνει την πόρτα και να παραμένει σε αυτή την κατάσταση μέχρι να δοθεί εντολή να κλειδώσει. Αντίθετα δηλαδή με τα συνηθισμένα συστήματα ασφαλείας τα οποία ξεκλειδώνουν την πόρτα και την διατηρούν στην κατάσταση αυτή μέχρι να εισέλθει το άτομο, και κλειδώνει μόλις αυτό κλείσει την πόρτα.

Όταν ο ακροδέκτης P3.3 βρίσκεται σε λογικό επίπεδο '1', τότε το τρανζίστορ οδήγησης του ρελέ άγει και το ρελέ μεταγεται στην θέση NO (Normally Open). Όταν ο ακροδέκτης P3.3 βρίσκεται σε λογικό επίπεδο '0', τότε το τρανζίστορ οδήγησης του ρελέ δεν άγει και το ρελέ παραμένει στην θέση NC (Normally Closed).

Αν η κλειδαριά είναι συνδεδεμένη στον ακροδέκτη NC και εφόσον δέχεται τροφοδοσία ξεκλειδώνει, τότε θα πρέπει ο ακροδέκτης P3.3 να παραμένει στο λογικό επίπεδο '0' για να μένει ανοικτή η πόρτα. Αντίθετα για να μένει κλειδωμένη θα πρέπει να υπάρχει λογικό επίπεδο '1' στον ακροδέκτη P3.3.

Δηλαδή ισχύουν οι παρακάτω καταστάσεις:

		Σύνδεση Κλειδαριάς	
P3.3	Transistor	NO	NC
'0'	Δεν άγει	Κλειστή	Ανοικτή
'1'	Άγει	Ανοικτή	Κλειστή

Η αντίθετη κατάσταση ισχύει αν η κλειδαριά κλειδώνει όταν δέχεται τροφοδοσία κάτι ιδιαίτερα επικίνδυνο καθώς σε περίπτωση απώλειας ισχύος, ο χώρος παραμένει με ξεκλειδωτή πόρτα, καταστρατηγώντας έτσι κάθε επιδίωξη για ασφάλιση του χώρου. Για τον λόγο αυτό δεν έχει ληφθεί υπ' όψιν τέτοιος τύπος κλειδαριάς.

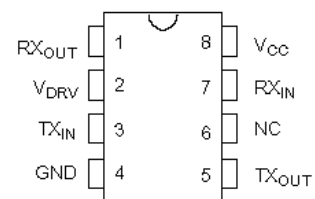
## Σειριακό Interface

Για την επικοινωνία με τον Ηλεκτρονικό Υπολογιστή που φιλοξενεί το λογισμικό διαχείρισης του τερματικού, χρησιμοποιείται η σειριακή θύρα του υπολογιστή σε σύνδεση με το ενσωματωμένο UART του μικροελεγκτή. Επειδή όμως η σειριακή θύρα του υπολογιστή διαχειρίζεται σήματα που υπακούουν στο πρωτόκολλο RS-232C, ενώ το UART του μικροελεγκτή αντιλαμβάνεται μόνο σήματα TTL (0V, +5V), πρέπει να υπάρξει κάποια μετατροπή πρωτοκόλλων. Την εργασία αυτή αναλαμβάνει ένα ολοκληρωμένο κύκλωμα, το DS275 της εταιρίας Dallas Semiconductors.

Αυτό δέχεται τα σήματα μορφής TTL από τον ακροδέκτη TxD (P3.1) και τα μετατρέπει σε σήματα μορφής RS-232C και τα μεταδίδει στον ακροδέκτη Rx (pin 2 του συνδετήρα DB9), ενώ αντίθετα μετατρέπει τα σήματα μορφής RS-232C από τον ακροδέκτη Tx (pin 3 του συνδετήρα DB9) και τα μεταδίδει στον ακροδέκτη RxD (P3.0) του μικροελεγκτή σε μορφή TTL.

Παρακάτω εμφανίζονται σε σειρά το τυπικό διάγραμμα συνδεσμολογίας του ολοκληρωμένου DS275, η αντιστοιχία των ακροδεκτών του συνδετήρα DB9 της σειριακής θύρας του υπολογιστή, ενώ επίσης εμφανίζεται και διάγραμμα των επιπέδων τάσεων που επικρατούν στην σειριακή θύρα ακολουθώντας το πρότυπο RS-232C.

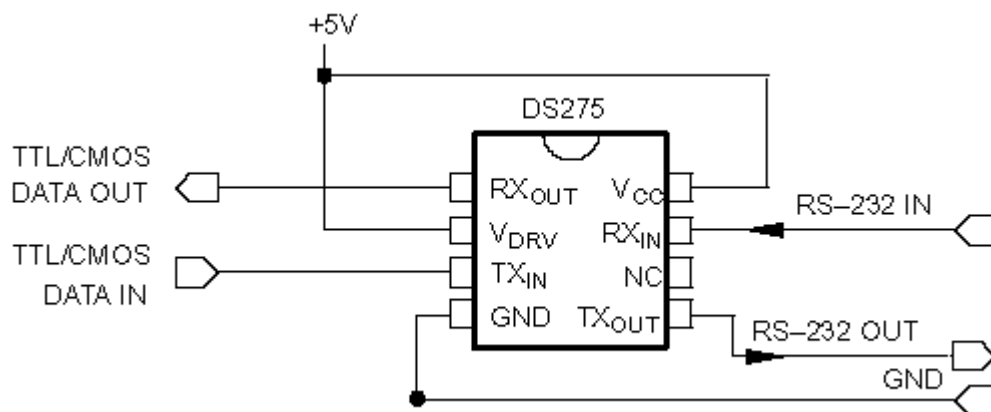
### PIN ASSIGNMENT



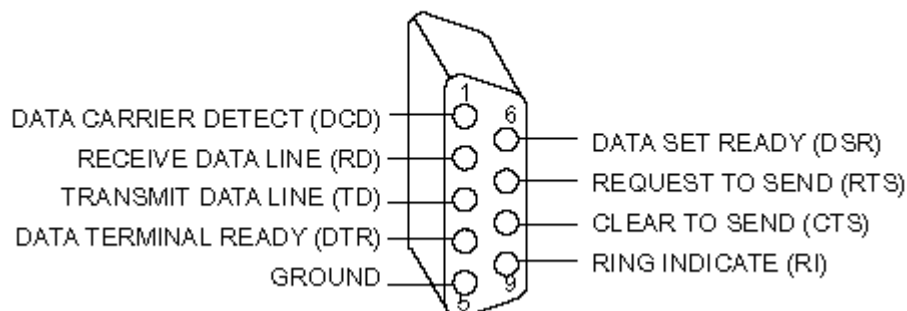
DS275 8-PIN DIP (300 Mil.)

### PIN DESCRIPTION

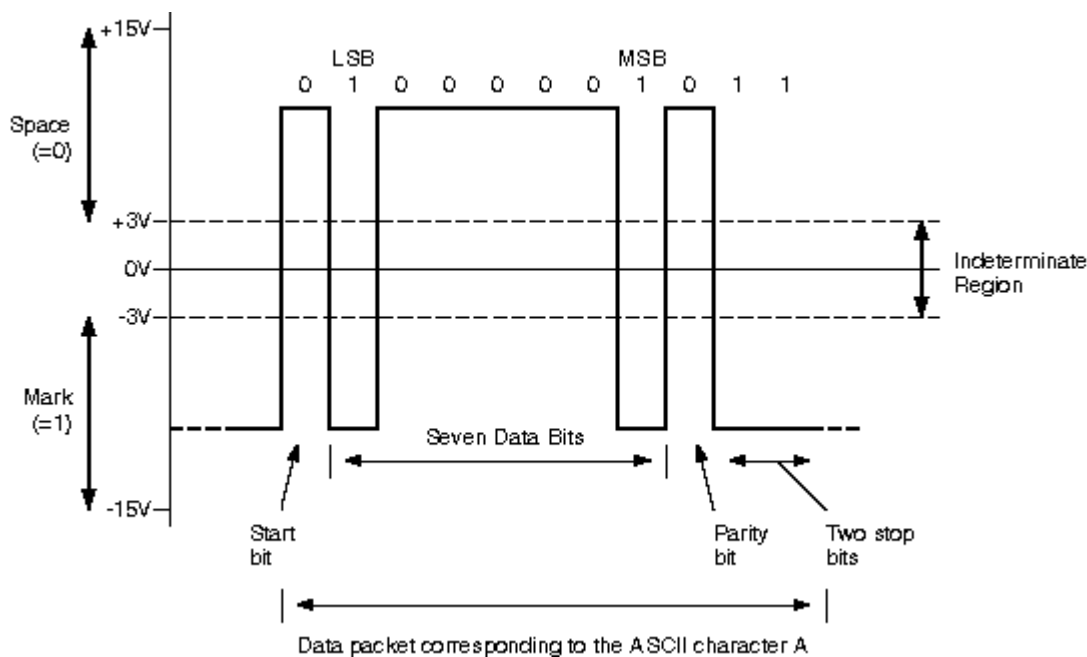
RX <sub>OUT</sub>	- RS-232 Receiver Output
V <sub>DRV</sub>	- Transmit driver +V
TX <sub>IN</sub>	- RS-232 Driver Input
GND	- System Ground (0V)
TX <sub>OUT</sub>	- RS-232 Driver Output
NC	- No Connection
RX <sub>IN</sub>	- RS-232 Receive Input
V <sub>CC</sub>	- System Logic Supply (+5V)



9-PIN CONNECTOR







Στο παραπάνω σχήμα το λογικό '0' σημειώνεται με SPACE ενώ το λογικό '1' με MARK, σύμφωνα με την ορολογία του κώδικα Morse.

Το ολοκληρωμένο DS275 εργάζεται σε Half Duplex, δηλαδή δεν επιτρέπει την ταυτόχρονη μετάδοση και λήψη, αλλά μόνο μια κατάσταση γραμμής για κάθε δεδομένη χρονική στιγμή. Ο λόγος που δεν επιτρέπεται η Full Duplex λειτουργία είναι ότι το ολοκληρωμένο κάνει χρήση των αρνητικών δυναμικών που επικρατούν στην γραμμή μεταφοράς για να παράγει τα αρνητικά επίπεδα τάσεως για την αποστολή από την πλευρά του τερματικού. Δεν απαιτεί έτσι κάποια εξωτερική συμμετρική τροφοδοσία ή πυκνωτές για χρήση εσωτερικού κυκλώματος παραγωγής αντίστροφων τάσεων. Έτσι μειώνεται η ανάγκη χρήσεως πολύπλοκων κυκλωμάτων, που θα είχαν αντίκτυπο στο κόστος αλλά και στην σχεδίαση του κυκλώματος. Εξάλλου η δυνατότητα μεταφοράς σε Full Duplex λειτουργία δεν απαιτείται από ένα απλό διαλογικό σύστημα επικοινωνίας ερωταποκρίσεων που υιοθετεί το λογισμικό του τερματικού.

Η επικοινωνία επιτυγχάνεται με την χρήση μόνο τριών αγωγών από αυτούς που προβλέπει το πρωτόκολλο RS-232C, τους Rx, Tx και Ground. Αυτοί είναι οι ελάχιστοι απαραίτητοι για επικοινωνία χωρίς χειραγία (Handshaking) μεταξύ των συνδεδεμένων συστημάτων.

Για λόγους εγκυρότητας των μεταφερόμενων δεδομένων, η σύνδεση μεταξύ του τερματικού και του Η/Υ γίνεται με καλώδιο συνεστραμμένων ζευγών αγωγών (UTP – Unshielded Twisted Pair). Αν το περιβάλλον χρήσης δημιουργεί θόρυβο στις γραμμές μεταφοράς δεδομένων θα πρέπει να γίνει χρήση θωρακισμένου καλωδίου συνεστραμμένων ζευγών αγωγών (STP – Shielded Twisted Pair). Τα καλώδια αυτά χρησιμοποιούνται σε δίκτυα υπολογιστών συνεπώς δεν είναι δύσκολη η εύρεση και χρήση τους.

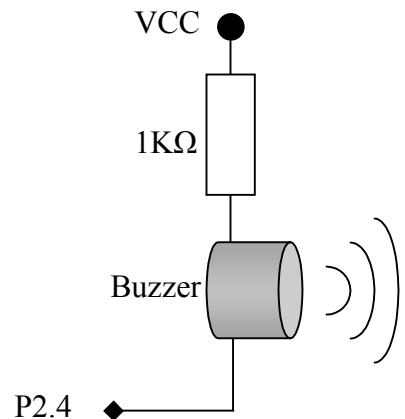
Γίνεται χρήση δύο εκ των τεσσάρων συνολικά ζευγών του καλωδίου. Ένα ζεύγος αποτελείται από την γραμμή RxD και την γείωση, ενώ το άλλο από την γραμμή TxD και την γείωση. Προστατεύεται έτσι σε μεγάλο βαθμό η ακεραιότητα των δεδομένων και αυξάνεται η αναισθησία στον θόρυβο, προερχόμενο τόσο από το περιβάλλον αλλά και από το φαινόμενο παραδιαφωνίας μεταξύ των αγωγών (Cross talk).

# Λοιπά Κυκλώματα

## Κύκλωμα του Βομβητή

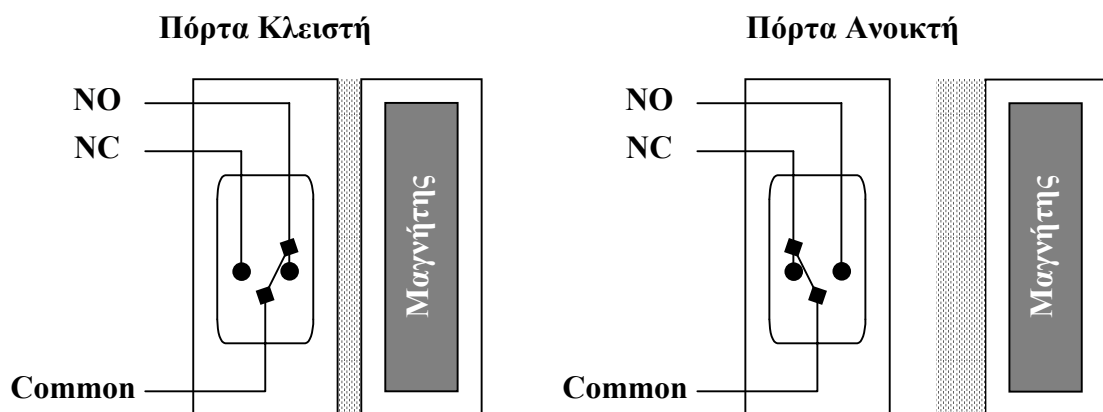
Για την καλύτερη χρήση του τερματικού, σημαντικό στοιχείο θεωρήθηκε η δυνατότητα παραγωγής ηχητικών σημάτων. Σκοπός αυτών είναι η προειδοποίηση για πρόκληση σφάλματος, ολοκλήρωση κάποιας εργασίας αλλά και την αλλαγή της ώρας.

Το κύκλωμα που χρησιμοποιείται είναι απλούστατο αφού το μόνο στοιχείο που περιλαμβάνει πλέον του βομβητή (buzzer) είναι μια αντίσταση περιορισμού ρεύματος, τιμής 1KΩ. Ο έλεγχος του βομβητή γίνεται μέσω λογισμικού από ρουτίνες που απλά θέτουν σε κατάσταση λογικού '0' τον ακροδέκτη P2.4 στον οποίο είναι συνδεδεμένος, όταν πρέπει να ενεργοποιηθεί κάποιο ηχητικό σήμα κλείνοντας κύκλωμα προς τη γη, και θέτουν σε κατάσταση λογικού '1' όταν θέλουν να το απενεργοποιήσουν.



## Ανίχνευση Κατάστασης Πόρτας

Η ανίχνευση της κατάστασης της πόρτας, δηλαδή αν η πόρτα είναι ανοικτή ή κλειστή, γίνεται με την χρήση ενός στοιχείου Reed Relay. Αυτό κάνει χρήση ενός Πελέ μορφής SPDT (Single Pole Double Throw) που ενεργοποιείται μαγνητικά. Τέτοια στοιχεία χρησιμοποιούνται συχνά σε συστήματα συναγερμού συνεπώς δεν αποτελεί κάποιο εξωτικό εξάρτημα. Οι ακροδέκτες του Reed Relay συνδέονται στους ακροδέκτες P3.4 (NC) και P3.5 (NO), ενώ ο κοινός ακροδέκτης συνδέεται στην γη. Το τερματικό διαβάζοντας αυτούς τους ακροδέκτες μπορεί να γνωρίζει αν η πόρτα είναι ανοικτή ή όχι.



NO	NC	Κατάσταση Πόρτας
'0'	'1'	Ανοικτή
'1'	'0'	Κλειστή

Οποιαδήποτε άλλη κατάσταση σημαίνει κακή συνδεσμολογία του εξαρτήματος ή δυσλειτουργία του.

---

---

# Κεφάλαιο 3<sup>ο</sup>

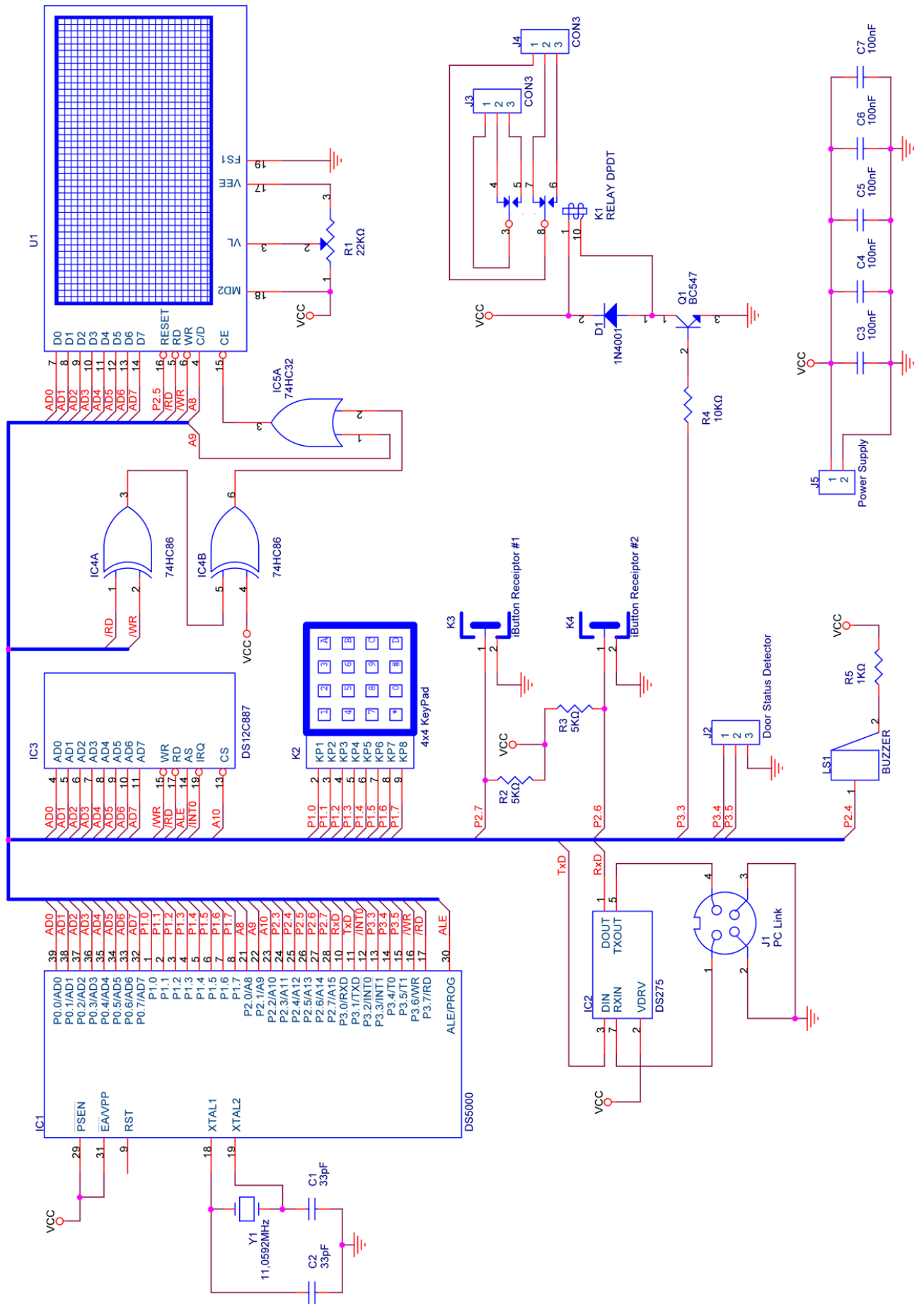
## Υλοποίηση Μικροϋπολογιστικού Συστήματος Τερματικού

### Σχηματικό Διάγραμμα

Στην επόμενη σελίδα περιλαμβάνεται το σχηματικό διάγραμμα συνδέσεων των κυκλωμάτων και εξαρτημάτων που απαρτίζουν το τερματικό της πτυχιακής αυτής εργασίας.

Η σχεδίαση του σχηματικού και υλοποίηση του τυπωμένου κυκλώματος έχει γίνει με την χρήση του λογισμικού πακέτου OrCAD 9.1.

Τα αρχεία αυτά βρίσκονται στον συνοδευτικό οπτικό δίσκο αποθήκευσης δεδομένων (CD-ROM) της πτυχιακής εργασίας.



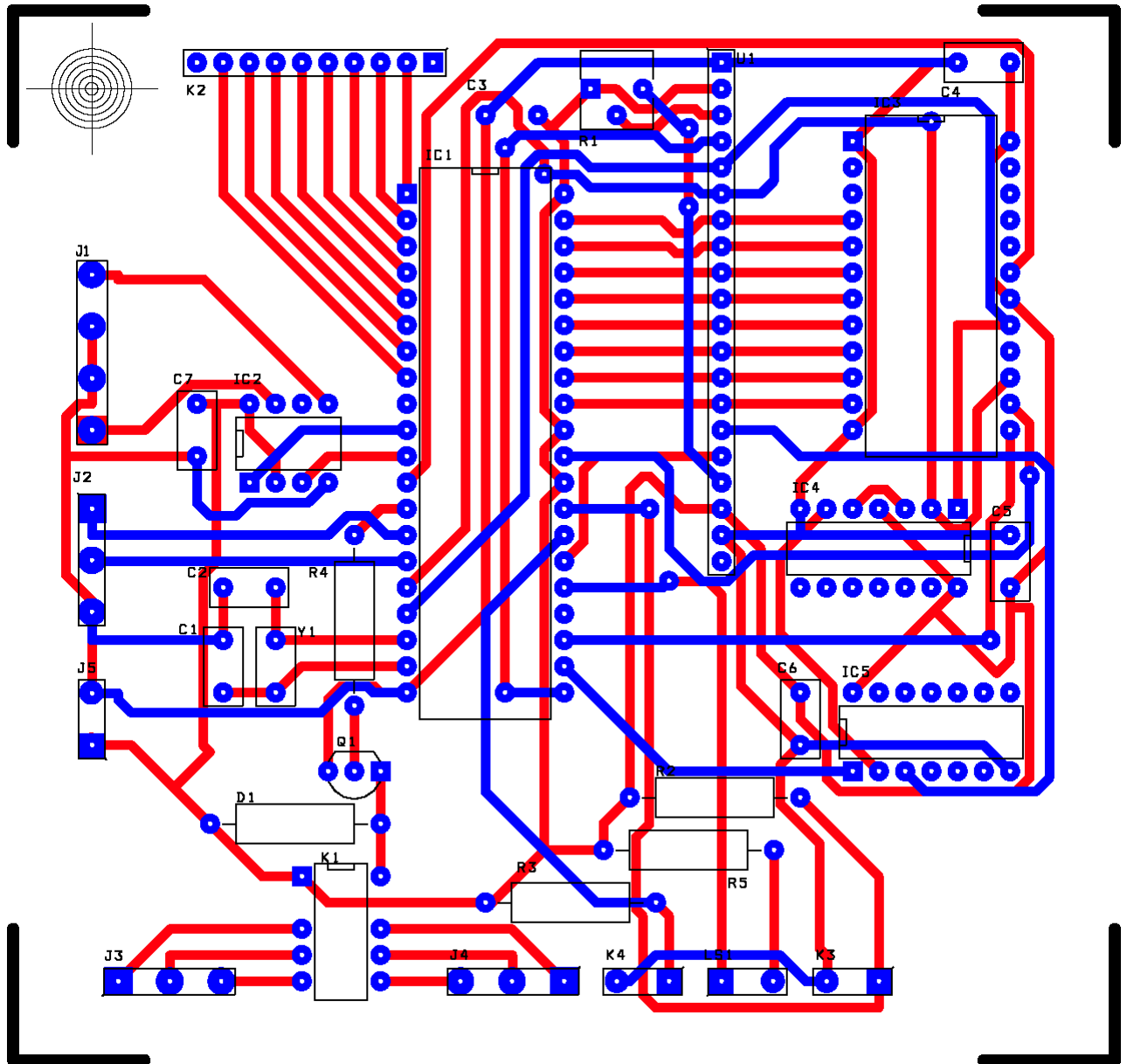
---

## Τυπωμένο Κύκλωμα

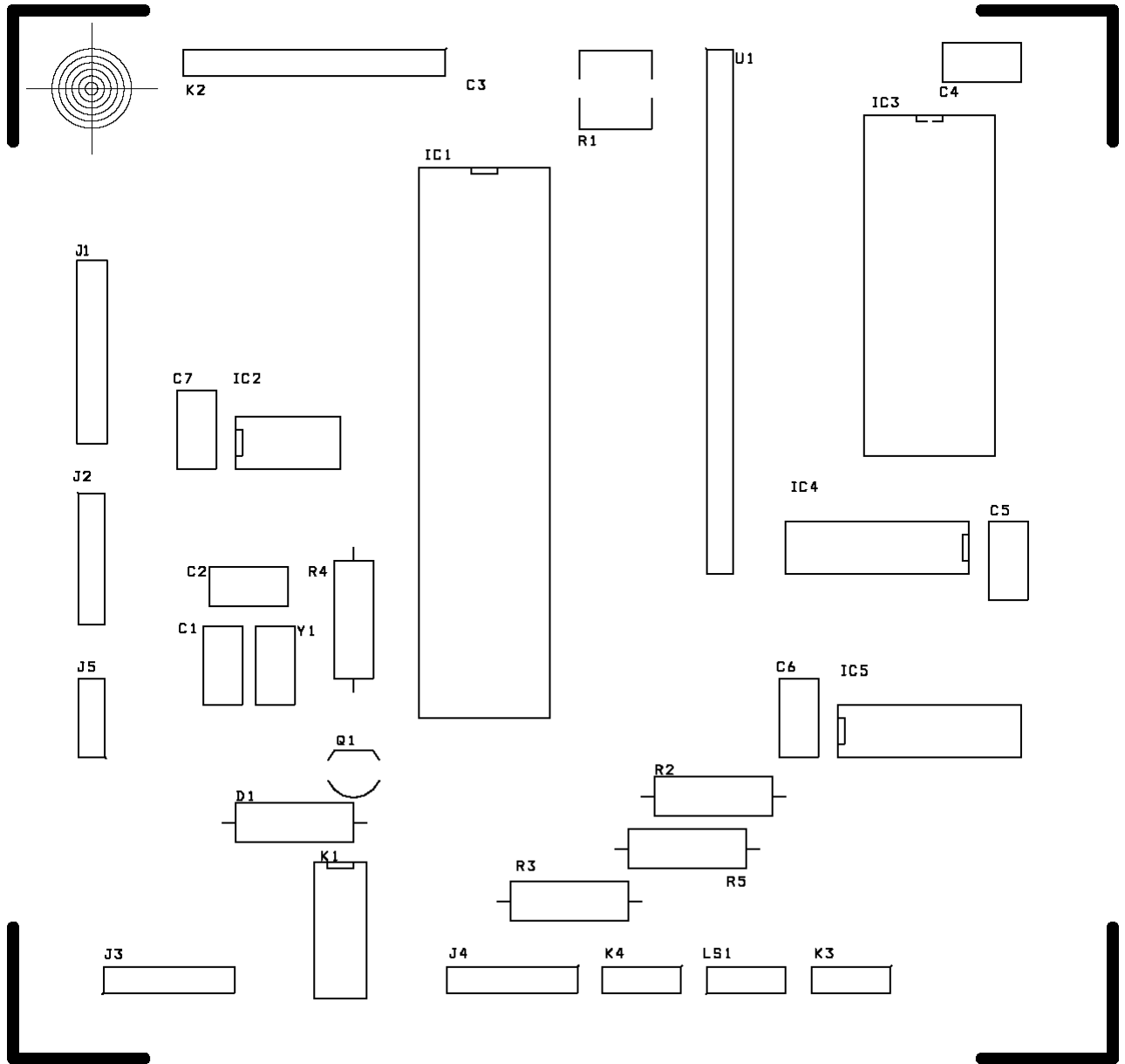
Κατάλογος Εξαρτημάτων	
<b>IC1</b>	DS5000T-32KB-12MHz
<b>IC2</b>	DS275
<b>IC3</b>	DS12C887
<b>IC4</b>	74HC86
<b>IC5</b>	74HC32
<b>C1-C2</b>	33pF
<b>C3-C7</b>	100nF
<b>D1</b>	1N4001
<b>J1</b>	Κλέμμα πλακέτας 4 υποδοχών
<b>J2-J4</b>	Κλέμμα πλακέτας 3 υποδοχών
<b>J5</b>	Κλέμμα πλακέτας 2 υποδοχών
<b>K1</b>	Relay AXICOM FX2 D3209
<b>K2</b>	Συνδετήρας καλωδιοταινίας 10 αγωγών
<b>K3-K4</b>	Κλέμμα πλακέτας 2 υποδοχών
<b>LS1</b>	Κλέμμα πλακέτας 2 υποδοχών
<b>Q1</b>	BC547A
<b>R1</b>	Trimmer πλακέτας 22KΩ
<b>R2-R3</b>	5KΩ
<b>R4</b>	10KΩ
<b>R5</b>	1KΩ
<b>U1</b>	Συνδετήρας καλωδιοταινίας 20 αγωγών
<b>Y1</b>	XTAL 11,0592MHz

Τα διαγράμματα που ακολουθούν βρίσκονται υπό κλίμακα και δεν μπορούν να χρησιμοποιηθούν για την αντιγραφή της πλακέτας. Για τον σκοπό αυτό έχουν προστεθεί τα απαραίτητα αρχεία στο συνοδευτικό οπτικό δίσκο αποθήκευσης δεδομένων (CD-ROM).

## Συνδέσεις Τυπωμένου Κυκλώματος

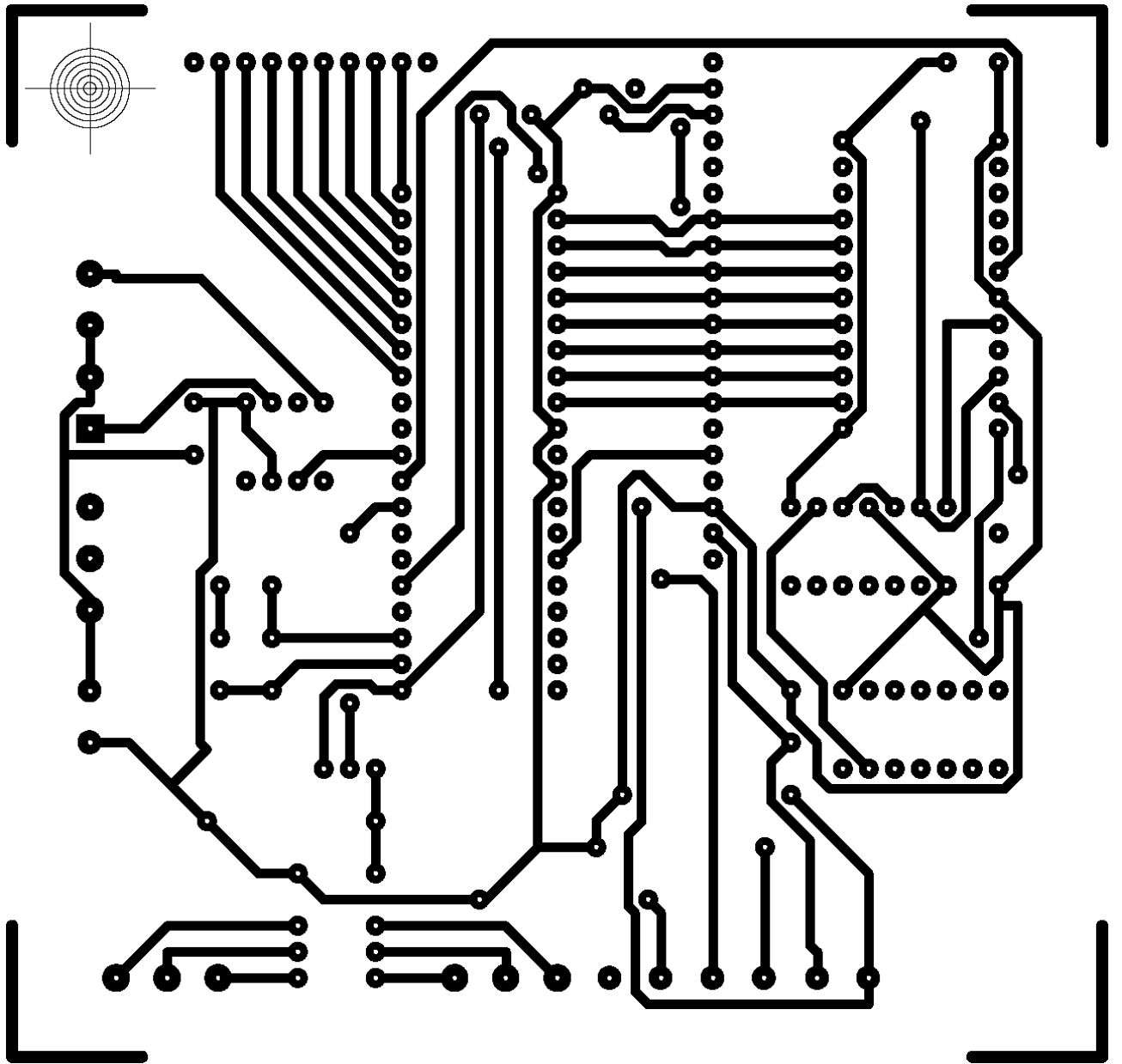


# Τοπογραφικό Εξαρτημάτων



---

Κάτω Όψη

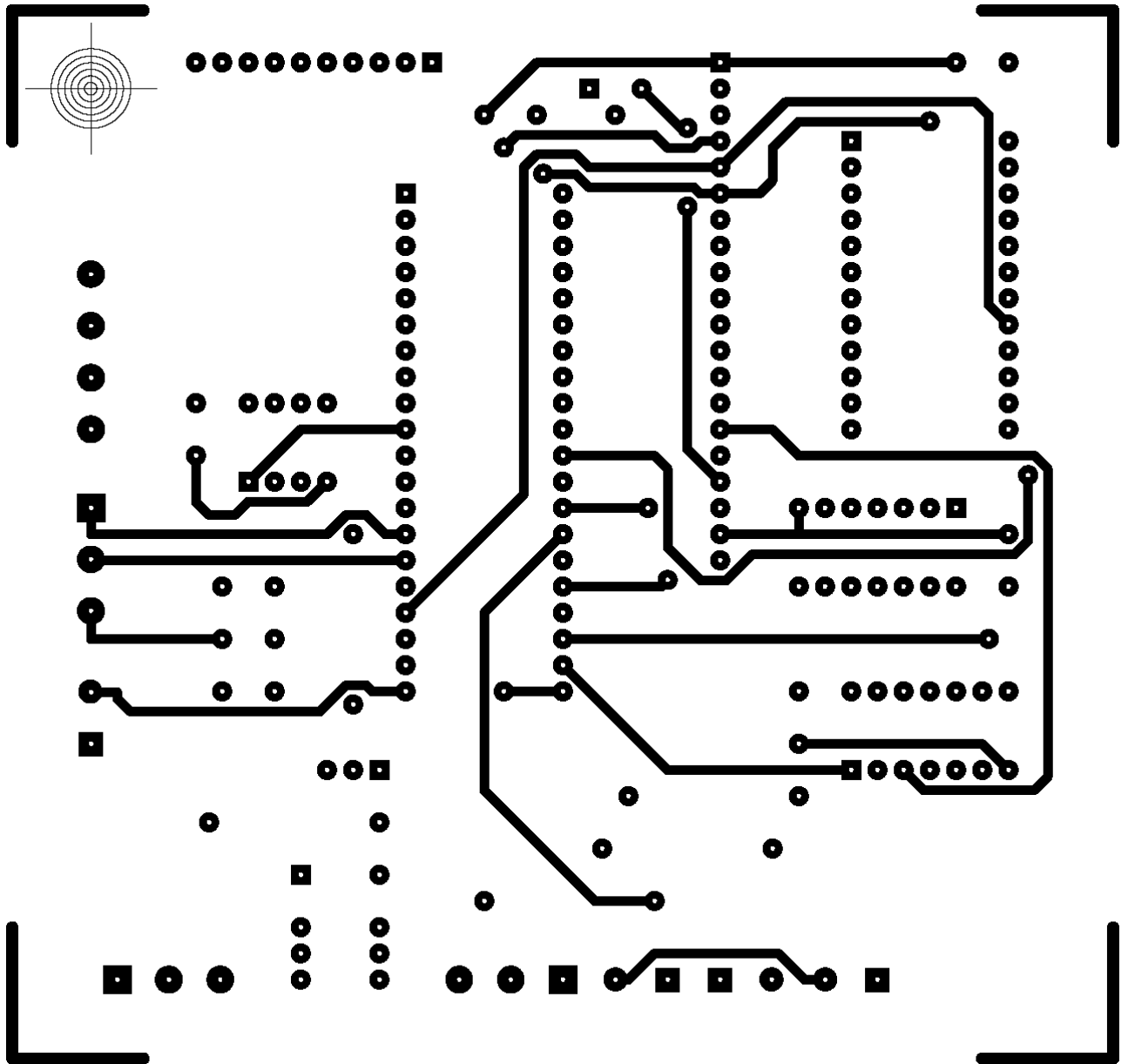




---

---

Πάνω Όψη



---

## Τροφοδοσία Κυκλώματος

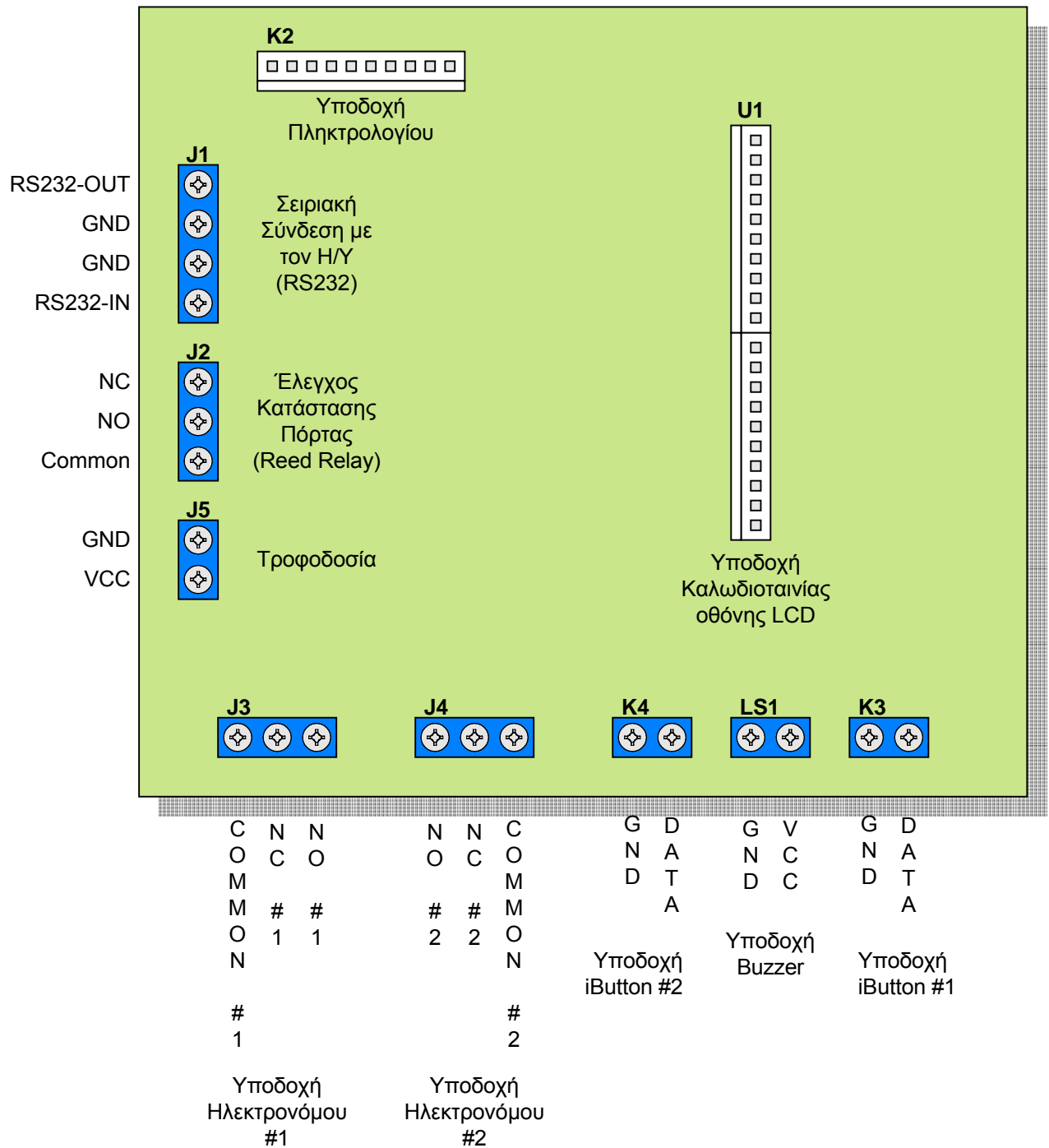
Το κύκλωμα του τερματικού τροφοδοτείται από μια διάταξη που παρέχει σταθεροποιημένη τάση +5V με μέγιστη παροχή ρεύματος 3A.

Τα εξαρτήματα ψηφιακής λογικής του κυκλώματος απορροφούν ελάχιστο ρεύμα καθώς πρόκειται για εξαρτήματα τεχνολογίας CMOS.

Το μεγαλύτερο ποσοστό ρεύματος απορροφά το κύκλωμα αυτοφωτισμού της οθόνης και κυμαίνεται από 600mA έως και 2,5A. Όσο μεγαλύτερο ρεύμα απορροφάει το κύκλωμα αυτό, τόσο πιο έντονος ο φωτισμός της οθόνης. Παράλληλα όμως τόσο μεγαλύτερη είναι και η θερμοκρασία που παράγεται στην επιφάνεια αυτής. Αποτέλεσμα της περίπτωσης αυτής είναι η υπερθέρμανση του υλικού κατασκευής της οθόνης (υγρός κρύσταλλος) και η αλλαγή της χημικής του κατάστασης με την πάροδο του χρόνου, μειώνοντας τον χρόνο συνολικής ζωής της οθόνης. Οπτικά η κατάσταση αυτή εμφανίζεται ως μια έντονη αμαύρωση της οθόνης και θα πρέπει να αποφεύγεται.

Για τον λόγο αυτό γίνεται περιορισμός του ρεύματος που παρέχεται στο κύκλωμα αυτοφωτισμού της οθόνης με την χρήση ενός ροοστάτη.

## Εξωτερικές Συνδέσεις Κυκλώματος Τερματικού



# Κεφάλαιο 4<sup>ο</sup>

## Πρωτόκολλο Επικοινωνίας Τερματικού – Η/Υ

Η επικοινωνία μεταξύ τερματικού και λογισμικού στον Η/Υ γίνεται όπως έχει ήδη αναφερθεί μέσω της σειριακής θύρας. Η σειριακή θύρα τόσο του υπολογιστή όσο και του τερματικού έχει προγραμματισθεί για επικοινωνία με παραμέτρους:

<b>Baud Rate:</b>	19,200
<b>Data Bits:</b>	8
<b>Stop Bits:</b>	1
<b>Parity Bits:</b>	Χωρίς Parity
<b>Handshaking:</b>	Χωρίς Handshaking

Ειδικοί χαρακτήρες ελέγχου που χρησιμοποιούνται από το πρωτόκολλο επικοινωνίας.

<b>ComEOT</b>	(End Of Transmission)	0x4
<b>ComEOS</b>	(End Of String)	0x0
<b>ComVT</b>	(Virtual Terminal)	0xb

Κάθε εντολή του τερματικού προς το λογισμικό του Η/Υ τερματίζεται με τον χαρακτήρα **ComVT**. Οι εντολές που αποστέλλονται από τον Η/Υ είναι καλά καθορισμένες συνεπώς δεν χρειάζεται κάποιος ειδικός χαρακτήρας τερματισμού εντολής, εκτός από περιπτώσεις όπου ειδικός χαρακτήρας απαιτείται για την αναγνώριση ορίων έναρξης-τερματισμού μεγάλου όγκου δεδομένων.

Εντολές του Λογισμικού του Η/Υ προς το Τερματικό	
Εντολή	Περιγραφή
<b>ATA</b>	Απάντηση του λογισμικού του Η/Υ ότι βρίσκεται σε κατάσταση αναμονής. Το τερματικό αποκρίνεται με την εντολή <b>ATOK</b> .
<b>ATB</b>	Ακολουθεί συμβολοσειρά που περιέχει το ονοματεπώνυμο του κατόχου του κλειδιού που εισήχθη. Το τερματικό αποκρίνεται με την εντολή <b>ATOK</b> .
<b>ATC</b>	Κλήση αναγνώρισης παρουσίας τερματικού στην θύρα COM. Το τερματικό αποκρίνεται με την εντολή <b>ATACK</b> ( <b>ACK</b> nowledge)
<b>ATD</b>	Εντολή εκτέλεσης διαγνωστικών ελέγχων τερματικού. Το τερματικό αποκρίνεται με την εντολή <b>ATOK</b> . Ακολουθεί εντολή προς τον υπολογιστή με πληροφορίες για την κατάσταση των υποσυστημάτων του τερματικού.
<b>ATE</b>	Δήλωση πρόκλησης σφάλματος από τον διαχειριστή τερματικού (λογισμικό του Η/Υ).
<b>ATL</b>	Εντολή για κλείδωμα της πόρτας. Επιστρέφει είτε σφάλμα ότι η πόρτα είναι ανοικτή <b>ATEDO</b> ( <b>Err</b> or <b>Do</b> or <b>O</b> pen), είτε επιβεβαίωση ότι η πόρτα κλειδώθηκε <b>ATDL</b> ( <b>Do</b> or <b>L</b> ocked).
<b>ATM</b>	Εντολή αποθήκευσης αριθμού του κυρίου κλειδιού ( <b>Master Key</b> ). Ακολουθεί ο 64bit σειριακός αριθμός του κυρίου κλειδιού σε μορφή 16 χαρακτήρων ASCII. Το τερματικό αποκρίνεται με την εντολή <b>ATOK</b> .

<b>ATQ</b>	<p>Εντολή διαχείρισης παραμέτρων εμφάνισης του κυλιόμενου μηνύματος που εμφανίζεται στην οθόνη του τερματικού. Ακολουθεί μια τουλάχιστον παράμετρος.</p> <p><b>Παράμετρος '0':</b> Απενεργοποίηση του κυλιόμενου μηνύματος. Το τερματικό αποκρίνεται με την εντολή <b>ATMOFF (Marquee OFF)</b>.</p> <p><b>Παράμετρος '1':</b> Ενεργοποίηση του κυλιόμενου μηνύματος. Το τερματικό αποκρίνεται με την εντολή <b>ATMON (Marquee ON)</b>.</p> <p><b>Παράμετρος '2':</b> Επαναφορά του προκαθορισμένου κυλιόμενου μηνύματος. Το τερματικό αποκρίνεται με την εντολή <b>ATDMR (Default Marquee Restored)</b>.</p> <p><b>Παράμετρος '3':</b> Αποθήκευση νέου κυλιόμενου μηνύματος. Ακολουθεί συμβολοσειρά 28 χαρακτήρων που περιέχει το νέο μήνυμα, η οποία τερματίζεται με τον χαρακτήρα <b>ComEOS</b>. Το τερματικό αποκρίνεται με την εντολή <b>ATNMS (New Marquee Stored)</b>.</p> <p>Το τερματικό αποκρίνεται με την εντολή <b>ATOK</b>.</p>																																												
<b>ATR</b>	<p>Εντολή αλλαγής ημερομηνίας και ώρας του RTC. Ακολουθούν οι πληροφορίες για την νέα ώρα και ημερομηνία.</p> <p>Τα δεδομένα είναι σε μορφή ASCII και η δομή τους είναι η εξής:</p> <table border="1" data-bbox="480 819 1350 904"> <thead> <tr> <th colspan="6">Τρέχουσα Ώρα</th> <th colspan="8">Τρέχουσα Ημερομηνία</th> </tr> </thead> <tbody> <tr> <td>h</td><td>h</td><td>m</td><td>m</td><td>s</td><td>s</td> <td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>M</td><td>M</td><td>D</td><td>D</td><td>d</td><td>d</td> </tr> </tbody> </table> <p>όπου:</p> <table border="1" data-bbox="611 954 1219 1480"> <tbody> <tr> <td>hh:</td> <td>Ώρα (π.χ. 23)</td> </tr> <tr> <td>mm:</td> <td>Λεπτά της ώρας (π.χ. 59)</td> </tr> <tr> <td>ss:</td> <td>Δευτερόλεπτα (π.χ. 29)</td> </tr> <tr> <td>YYYY:</td> <td>Χρονολογία (π.χ. 2001)</td> </tr> <tr> <td>MM:</td> <td>Μήνας (π.χ. 03 για τον Μάρτιο)</td> </tr> <tr> <td>DD:</td> <td>Μέρα του μήνα (π.χ. 03)</td> </tr> <tr> <td>dd:</td> <td>Μέρα της εβδομάδας (π.χ. 01) 01: Κυριακή 02: Δευτέρα 03: Τρίτη 04: Τετάρτη 05: Πέμπτη 06: Παρασκευή 07: Σάββατο</td> </tr> </tbody> </table> <p>Συνολικά απαιτούνται 16 χαρακτήρες για την ρύθμιση της ώρας και ημερομηνίας.</p>	Τρέχουσα Ώρα						Τρέχουσα Ημερομηνία								h	h	m	m	s	s	Y	Y	Y	Y	M	M	D	D	d	d	hh:	Ώρα (π.χ. 23)	mm:	Λεπτά της ώρας (π.χ. 59)	ss:	Δευτερόλεπτα (π.χ. 29)	YYYY:	Χρονολογία (π.χ. 2001)	MM:	Μήνας (π.χ. 03 για τον Μάρτιο)	DD:	Μέρα του μήνα (π.χ. 03)	dd:	Μέρα της εβδομάδας (π.χ. 01) 01: Κυριακή 02: Δευτέρα 03: Τρίτη 04: Τετάρτη 05: Πέμπτη 06: Παρασκευή 07: Σάββατο
Τρέχουσα Ώρα						Τρέχουσα Ημερομηνία																																							
h	h	m	m	s	s	Y	Y	Y	Y	M	M	D	D	d	d																														
hh:	Ώρα (π.χ. 23)																																												
mm:	Λεπτά της ώρας (π.χ. 59)																																												
ss:	Δευτερόλεπτα (π.χ. 29)																																												
YYYY:	Χρονολογία (π.χ. 2001)																																												
MM:	Μήνας (π.χ. 03 για τον Μάρτιο)																																												
DD:	Μέρα του μήνα (π.χ. 03)																																												
dd:	Μέρα της εβδομάδας (π.χ. 01) 01: Κυριακή 02: Δευτέρα 03: Τρίτη 04: Τετάρτη 05: Πέμπτη 06: Παρασκευή 07: Σάββατο																																												
<b>ATS</b>	<p>Εντολή εναλλαγής κατάστασης ειδικού μηνύματος. Το τερματικό αποκρίνεται με την εντολή <b>ATCIP-OFF (Class In Progress OFF)</b> όταν απενεργοποιείται η ένδειξη του ειδικού μηνύματος, ενώ με την εντολή <b>ATCIP-ON (Class In Progress ON)</b> όταν αυτή ενεργοποιείται.</p>																																												
<b>ATT</b>	<p>Εντολή αποθήκευσης τίτλων ανακοινώσεων. Το τερματικό αποκρίνεται με την εντολή <b>ATOK</b>. Ακολουθούν 8 συμβολοσειρές 25 χαρακτήρων η καθεμία, που αντιστοιχούν στους τίτλους των ανακοινώσεων. Κάθε συμβολοσειρά τερματίζεται με τον επιπρόσθετο χαρακτήρα <b>ComEOS</b>. Συνολικά κάθε τίτλος ανακοίνωσης απαιτεί 26 χαρακτήρες. Αποστέλλεται μπλοκ 8 τίτλων ανεξάρτητα αν περιέχουν πραγματικό κείμενο ή λευκούς χαρακτήρες. Το τέλος των αποστελλόμενων τίτλων ανακοινώσεων σημειώνεται με τον χαρακτήρα <b>ComEOT</b>.</p>																																												
<b>ATU</b>	<p>Εντολή για ξεκλείδωμα της πόρτας. Επιστρέφει είτε σφάλμα ότι η πόρτα είναι ήδη ανοικτή <b>ATEDO (Error Door Open)</b>, είτε επιβεβαίωση ότι η πόρτα ξεκλειδώθηκε <b>ATDU (Door Unlocked)</b>.</p>																																												

<b>ATX</b>	Εντολή αποθήκευσης κειμένου ανακοίνωσης. Το τερματικό αποκρίνεται με την εντολή <b>ATOK</b> . Ακολουθούν συμβολοσειρές 28 χαρακτήρων η καθεμία, που αντιστοιχούν στις γραμμές της ανακοίνωσης. Κάθε συμβολοσειρά τερματίζεται με τον επιπρόσθετο χαρακτήρα <b>ComEOS</b> . Συνολικά κάθε γραμμή απαιτεί 29 χαρακτήρες. Αποστέλλονται μπλοκ 8 γραμμών για κάθε οθόνη ανεξάρτητα αν περιέχουν πραγματικό κείμενο ή λευκούς χαρακτήρες. Το τέλος των αποστελλόμενων συμβολοσειρών της ανακοίνωσης σημειώνεται με τον χαρακτήρα <b>ComEOT</b> .
<b>ATY</b>	Εντολή αποθήκευσης εισερχομένων τίτλων μαθημάτων. Το τερματικό αποκρίνεται με την εντολή <b>ATOK</b> . Ακολουθούν 8 συμβολοσειρές 28 χαρακτήρων η καθεμία, που αντιστοιχούν στους τίτλους μαθημάτων. Κάθε συμβολοσειρά τερματίζεται με τον επιπρόσθετο χαρακτήρα <b>ComEOS</b> . Συνολικά κάθε γραμμή κειμένου απαιτεί 29 χαρακτήρες. Αποστέλλεται μπλοκ 8 τίτλων μαθημάτων. Το τέλος των αποστελλόμενων τίτλων μαθημάτων σημειώνεται με τον χαρακτήρα <b>ComEOT</b> . Το τερματικό αποκρίνεται με την εντολή <b>ATCTOK (Course Titles OK)</b> .
<b>ATZ</b>	Εντολή αποθήκευσης βαθμολογικού πίνακα μαθήματος. Το τερματικό αποκρίνεται με την εντολή <b>ATOK</b> . Ακολουθούν συμβολοσειρές 28 χαρακτήρων η καθεμία, που αντιστοιχούν στο ονοματεπώνυμο σπουδαστή/στριας και τον βαθμό. Κάθε συμβολοσειρά τερματίζεται με τον επιπρόσθετο χαρακτήρα <b>ComEOS</b> . Συνολικά κάθε γραμμή του βαθμολογικού πίνακα απαιτεί 29 χαρακτήρες. Αποστέλλονται μπλοκ 8 γραμμών για κάθε οθόνη ανεξάρτητα αν περιέχουν πραγματικό κείμενο ή λευκούς χαρακτήρες. Το τέλος του αποστελλόμενου βαθμολογικού πίνακα σημειώνεται με τον χαρακτήρα <b>ComEOT</b> . Λόγω περιορισμών της γλώσσας προγραμματισμού του μικροελεγκτή του τερματικού μπορεί να φιλοξενηθεί πίνακας με 248 βαθμολογίες σπουδαστών/σπουδαστριών. Τα υπόλοιπα ονόματα και οι βαθμολογίες απλά δεν αποστέλλονται από το λογισμικό του Η/Υ.

<b>Εντολές του Τερματικού προς το Λογισμικό του Η/Υ</b>									
<b>Εντολή</b>	<b>Περιγραφή</b>								
<b>ATACK</b>	Εντολή αναγνώρισης παρουσίας του τερματικού στην σειριακή θύρα επικοινωνίας ( <b>ACKnowledge</b> ).								
<b>ATAL</b>	Εντολή αίτησης αποστολής της λίστας τίτλων ανακοινώσεων ( <b>Announcements List</b> ). Ο Η/Υ ανταποκρίνεται σύμφωνα με την εντολή <b>ATT</b> .								
<b>ATCIP-OFF</b>	Εντολή ένδειξης κατάστασης του ειδικού μηνύματος ( <b>Class In Progress OFF</b> ). Το ειδικό μήνυμα απενεργοποιήθηκε.								
<b>ATCIP-ON</b>	Εντολή ένδειξης κατάστασης του ειδικού μηνύματος ( <b>Class In Progress ON</b> ). Το ειδικό μήνυμα ενεργοποιήθηκε.								
<b>ATCM</b>	Εντολή αίτησης αποστολής του βαθμολογικού πίνακα για τον αριθμό του μαθήματος που ακολουθεί ως παράμετρος της εντολής ( <b>Course Marks</b> ). Ο υπολογιστής ανταποκρίνεται με την εντολή <b>ATZ</b> όταν υπάρχουν δεδομένα να αποστείλει, σε αντίθετη περίπτωση απαντά με την εντολή <b>ATE</b> .								
<b>ATCTOK</b>	Απάντηση στο λογισμικό του Η/Υ ότι αποθηκεύθηκαν οι τίτλοι των μαθημάτων, που του απεστάλησαν ( <b>Course Titles OK</b> ).								
<b>ATDIAG</b>	Εντολή που περιέχει την κατάσταση των υποσυστημάτων του τερματικού ( <b>DIAGnostics</b> ). Η δομή της εντολής είναι η εξής: <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th><b>Πληκτρολόγιο</b></th> <th><b>Υποδοχές iButton</b></th> <th><b>Οθόνη LCD</b></th> <th><b>Μπαταρία RTC</b></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">S</td> <td style="text-align: center;">S</td> <td style="text-align: center;">S</td> <td style="text-align: center;">S</td> </tr> </tbody> </table> <p>όπου S: ο χαρακτήρας '1' για ένδειξη καλής κατάστασης ή '0' για ένδειξη προβλήματος.</p>	<b>Πληκτρολόγιο</b>	<b>Υποδοχές iButton</b>	<b>Οθόνη LCD</b>	<b>Μπαταρία RTC</b>	S	S	S	S
<b>Πληκτρολόγιο</b>	<b>Υποδοχές iButton</b>	<b>Οθόνη LCD</b>	<b>Μπαταρία RTC</b>						
S	S	S	S						

<b>ATDL</b>	Εντολή ένδειξης κατάστασης της κλειδαριάς ( <b>Door Locked</b> ). Η πόρτα είναι κλειδωμένη.
<b>ATDMR</b>	Εντολή επιβεβαίωσης ότι επαναφέρθηκε το προκαθορισμένο κείμενο του κυλιόμενου μηνύματος ( <b>Default Marquee Restored</b> ).
<b>ATDU</b>	Εντολή ένδειξης κατάστασης της κλειδαριάς ( <b>Door Unlocked</b> ). Η πόρτα είναι ξεκλειδωμένη.
<b>ATEDO</b>	Εντολή ένδειξης κατάστασης της πόρτας ( <b>Entr Door Open</b> ). Η πόρτα είναι ανοικτή.
<b>ATEXRQ-</b>	Εντολή αίτησης εξόδου από κάτοχο κλειδιού ασφαλείας ( <b>EXit ReQuest</b> ). Ακολουθεί ο σειριακός αριθμός του κλειδιού που εισήχθη στην υποδοχή. Ισχύει μόνο για την υποδοχή #2. Το λογισμικό του υπολογιστή ανταποκρίνεται είτε με την εντολή <b>ATU</b> εφόσον αναγνώρισε τον κάτοχο του κλειδιού και διαπίστωσε ότι έχει δικαίωμα πρόσβασης, είτε με την εντολή <b>ATE</b> δηλώνοντας σφάλμα αναγνώρισης.
<b>ATID</b>	Εντολή αίτησης αναγνώρισης κατόχου του κλειδιού ασφαλείας ( <b>IDentify</b> ). Ακολουθεί ο σειριακός αριθμός του κλειδιού που εισήχθη στην υποδοχή. Ισχύει μόνο για την υποδοχή #1. Το λογισμικό του υπολογιστή ανταποκρίνεται είτε με την εντολή <b>ATB</b> εφόσον αναγνώρισε τον κάτοχο του κλειδιού, είτε με την εντολή <b>ATE</b> δηλώνοντας σφάλμα αναγνώρισης.
<b>ATKMP-</b>	Εντολή προσθήκης του σειριακού αριθμού του εισαχθέντος κλειδιού ασφαλείας στην βάση δεδομένων διαθέσιμων κλειδιών ( <b>Key Media Pool</b> ). Ακολουθεί ο σειριακός αριθμός του κλειδιού που εισήχθη στην υποδοχή #1.
<b>ATLRQ-</b>	Εντολή αίτησης κλειδώματος της πόρτας από κάτοχο κλειδιού ασφαλείας ( <b>Lock ReQuest</b> ). Ακολουθεί ο σειριακός αριθμός του κλειδιού που εισήχθη στην υποδοχή. Ισχύει μόνο για την υποδοχή #1. Το λογισμικό του υπολογιστή ανταποκρίνεται είτε με την εντολή <b>ATL</b> εφόσον αναγνώρισε τον κάτοχο του κλειδιού και διαπίστωσε ότι έχει δικαίωμα πρόσβασης, είτε με την εντολή <b>ATE</b> δηλώνοντας σφάλμα αναγνώρισης.
<b>ATMOFF</b>	Εντολή επιβεβαίωσης απενεργοποίησης του κυλιόμενου μηνύματος ( <b>Marquee OFF</b> ).
<b>ATMON</b>	Εντολή επιβεβαίωσης ενεργοποίησης του κυλιόμενου μηνύματος ( <b>Marquee ON</b> ).
<b>ATNMS</b>	Εντολή επιβεβαίωσης αποθήκευσης του νέου κειμένου του κυλιόμενου μηνύματος ( <b>New Marquee Stored</b> ).
<b>ATOK</b>	Αποδοχή εντολής που εστάλη από το λογισμικό του H/Y ( <b>OK</b> ).
<b>ATREQ</b>	Εντολή αίτησης πρόσβασης από κάτοχο κλειδιού ασφαλείας ( <b>REQuest</b> ). Ακολουθεί ο σειριακός αριθμός του κλειδιού που εισήχθη στην υποδοχή καθώς και ο προσωπικός κωδικός του κατόχου. Ισχύει μόνο για την υποδοχή #1. Το λογισμικό του υπολογιστή ανταποκρίνεται είτε με την εντολή <b>ATU</b> εφόσον αναγνώρισε τον κάτοχο του κλειδιού και διαπίστωσε ότι έχει δικαίωμα πρόσβασης, είτε με την εντολή <b>ATE</b> δηλώνοντας σφάλμα αναγνώρισης.
<b>ATSA</b>	Εντολή αίτησης αποστολής κειμένου ανακοίνωσης ( <b>Send Announcement</b> ). Ακολουθεί ο αριθμός της ανακοίνωσης που ζητήθηκε. Το λογισμικό του H/Y αν εντοπίσει την ανακοίνωση, την διαμορφώνει και την αποστέλλει με την εντολή <b>ATX</b> . Αν η ανακοίνωση που ζητήθηκε δεν έχει κείμενο τότε επιστρέφεται η εντολή <b>ATE</b> δηλώνοντας σφάλμα.
<b>ATTA</b>	Εντολή αίτησης αποστολής νέας ημερομηνίας και ώρας. Το λογισμικό του H/Y ανταποκρίνεται με την εντολή <b>ATR</b> .
<b>ComEOT</b>	Αποστέλλοντας τον χαρακτήρα <b>ComEOT</b> το τερματικό ζητάει από τον H/Y να δηλώσει παρουσία. Το λογισμικό του H/Y ανταποκρίνεται με την εντολή <b>ATA</b> .

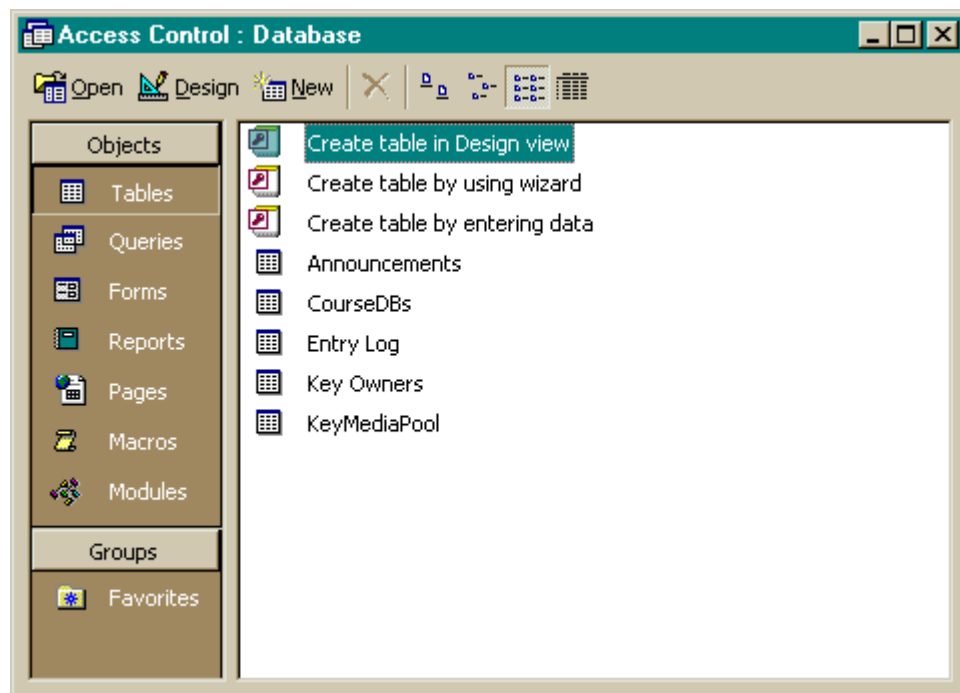
---

## Διαχείριση Βάσεων Δεδομένων από το Λογισμικό του Η/Υ

Για το πρόγραμμα διαχείρισης του τερματικού που εκτελείται από τον Ηλεκτρονικό Υπολογιστή, έγινε χρήση βάσεων δεδομένων στηριγμένες στην μηχανή της Microsoft Access 2000. Η προσπέλαση των πινάκων και εγγραφών μέσα από την γλώσσα προγραμματισμού Visual Basic 6.0 έγινε με την χρήση των αντικειμένων ADO (ActiveX Data Objects) και διασύνδεση Microsoft Jet 4.0 OLE DB.

### Βάση Δεδομένων Συστήματος

Σχεδόν όλες οι απαραίτητες πληροφορίες που εμφανίζονται στο τερματικό είναι αποθηκευμένες σε ένα αρχείο, το οποίο μπορεί να βρεθεί στον σκληρό δίσκο του ηλεκτρονικού υπολογιστή στον οποίο είναι εγκατεστημένο το λογισμικό, στην διαδρομή “C:\ACS\Access Control.mdb”. Εντός αυτού βρίσκονται οι πίνακες που φαίνονται στο παρακάτω σχήμα.



Στις επόμενες σελίδες παρατίθενται αναλυτικές καταστάσεις περιγραφών των πινάκων στην βάση δεδομένων.



Δομή Πίνακα Announcements					
Field Name	Data Type	Field Size	Required	Allow Zero Length	Indexed
<b>ID</b>	Autonumber	Long Integer			Yes (No Duplicates)
<b>AnnouncementTitle</b>	Text	24	No	Yes	Yes (Duplicates OK)
<b>PageNo1</b>	Text	255	No	Yes	No
<b>PageNo2</b>	Text	255	No	Yes	No
<b>PageNo3</b>	Text	255	No	Yes	No
<b>PageNo4</b>	Text	255	No	Yes	No
<b>PageNo5</b>	Text	255	No	Yes	No
<b>PageNo6</b>	Text	255	No	Yes	No
<b>PageNo7</b>	Text	255	No	Yes	No
<b>PageNo8</b>	Text			Yes	No

Στον πίνακα Announcements αποθηκεύονται τα κείμενα των ανακοινώσεων. Επειδή το μέγιστο μέγεθος ενός πεδίου δεδομένων τύπου Text είναι 255 χαρακτήρες, στάθηκε απαραίτητη η διάσπαση του κειμένου κάθε ανακοίνωσης σε περισσότερα πεδία μεγέθους 255. Συγκεκριμένα το κυρίως κείμενο της ανακοίνωσης διασπάται σε 8 πεδία (PageNo1 έως PageNo8). Στο πεδίο AnnouncementTitle αποθηκεύεται ο τίτλος της ανακοίνωσης. Ο μέγιστος αριθμός εγγραφών που συναντώνται στον πίνακα αυτό πρέπει να είναι 8, καθώς αυτός είναι και ο περιορισμός που έχει τεθεί τόσο από το λογισμικό του τερματικού όσο και από το λογισμικό διαχείρισης αυτού που βρίσκεται εγκατεστημένο στον ηλεκτρονικό υπολογιστή.

Δομή Πίνακα CourseDBs					
Field Name	Data Type	Field Size	Required	Allow Zero Length	Indexed
<b>CourseName</b>	Text	28	No	Yes	No
<b>DBFileName</b>	Text	255	No	Yes	
<b>CanBeViewed</b>	Yes/No		Yes		No

Στον πίνακα CourseDBs αποθηκεύονται πληροφορίες σχετικά με τα μαθήματα και τους βαθμολογικούς πίνακες. Συγκεκριμένα το πεδίο CourseName περιέχει τον τίτλο του μαθήματος. Το πεδίο DBFileName περιέχει την πλήρη διαδρομή και το όνομα του αρχείου εντός του οποίου είναι αποθηκευμένος ο βαθμολογικός πίνακας του μαθήματος. Το πεδίο CanBeViewed δέχεται μια Boolean τιμή (Λογικό True ή False), επιτρέποντας ή απαγορεύοντας την δημοσίευση του βαθμολογικού πίνακα από την αντίστοιχη λειτουργία του τερματικού. Ο μέγιστος αριθμός εγγραφών που συναντώνται στον πίνακα αυτό πρέπει να είναι 8, καθώς αυτός είναι και ο περιορισμός που έχει τεθεί τόσο από το λογισμικό του τερματικού όσο και από το λογισμικό διαχείρισης αυτού που βρίσκεται εγκατεστημένο στον ηλεκτρονικό υπολογιστή.

Δομή Πίνακα Entry Log					
Field Name	Data Type	Field Size	Required	Allow Zero Length	Indexed
<b>ID</b>	Autonumber	Long Integer			Yes (No Duplicates)
<b>DateOfEntry</b>	Date/Time		Yes		Yes (Duplicates OK)
<b>TimeOfEntry</b>	Date/Time		Yes		No
<b>NameOfKeyOwner</b>	Text	50	Yes	No	No
<b>IOType</b>	Text	15	Yes	No	No
<b>Lock</b>	Text	15	No	Yes	No

Στον πίνακα Entry Log αποθηκεύονται πληροφορίες σχετικά με την πρόσβαση κατόχων κλειδιών ασφαλείας στον ελεγχόμενο χώρο. Συγκεκριμένα στο πεδίο DateOfEntry αποθηκεύεται η ημερομηνία πρόσβασης στον χώρο. Στο πεδίο TimeOfEntry αποθηκεύεται η ώρα που έγινε αποδεκτή η αίτηση πρόσβασης. Στο πεδίο NameOfKeyOwner αποθηκεύεται το ονοματεπώνυμο του κατόχου του κλειδιού ασφαλείας. Στο πεδίο IOType αποθηκεύεται με κείμενο περιγραφικά ο τύπος πρόσβασης, δηλαδή «Είσοδος» ή «Εξοδος». Στο πεδίο Lock αποθηκεύεται η ένδειξη «Κλειδώμα» σε περίπτωση που ζητηθεί η λειτουργία αυτή από κάτοχο έγκυρου κλειδιού ασφαλείας, σε άλλη περίπτωση δεν αποθηκεύεται κάποια πληροφορία.

Δομή Πίνακα Key Owners					
Field Name	Data Type	Field Size	Required	Allow Zero Length	Indexed
<b>ID</b>	Autonumber	Long Integer			Yes (No Duplicates)
<b>FirstName</b>	Text	20	Yes	No	No
<b>Surname</b>	Text	25	No	No	Yes (Duplicates OK)
<b>AssociationType</b>	Text	50	No	No	No
<b>Address</b>	Text	100	No	Yes	No
<b>TelephoneNumber</b>	Text	15	No	No	No
<b>MobileNumber</b>	Text	10	No	No	No
<b>email_address</b>	Text	50	No	No	No
<b>ValidKeyUntil</b>	Date/Time		Yes		No
<b>ActiveKey</b>	Yes/No		Yes		No
<b>iKeyNumber</b>	Text	16	Yes	No	Yes (No Duplicates)
<b>PassCode</b>	Text	12	Yes	No	Yes (No Duplicates)
<b>Details</b>	Text	255	No	No	No

Στον πίνακα Key Owners αποθηκεύονται όλα τα στοιχεία που αφορούν τους κατόχους κλειδιών ασφαλείας. Συγκεκριμένα στο πεδίο FirstName αποθηκεύεται το μικρό όνομα του κατόχου. Στο πεδίο Surname αποθηκεύεται το επώνυμο του κατόχου. Στο πεδίο AssociationType αποθηκεύεται η ιδιότητα του κατόχου, δηλαδή η σχέση του επαγγελματική ή όχι με τον ελεγχόμενο χώρο. Στο πεδίο Address αποθηκεύεται η διεύθυνση της οικίας ή επαγγελματικής στέγης του κατόχου. Στο πεδίο TelephoneNumber αποθηκεύεται ο αριθμός σταθερού τηλεφώνου επικοινωνίας με τον κάτοχο. Στο πεδίο MobileNumber αντίστοιχα αποθηκεύεται ο αριθμός του κινητού τηλεφώνου. Στο πεδίο email\_address αποθηκεύεται η διεύθυνση της ηλεκτρονικής αλληλογραφίας του κατόχου. Στο πεδίο ValidKeyUntil δίδεται μια ημερομηνία πέρα της οποίας το ο κάτοχος του κλειδιού παύει να έχει πρόσβαση στον ελεγχόμενο χώρο. Το πεδίο ActiveKey δίνει την δυνατότητα προσωρινής ή μόνιμης αφαίρεσης δικαιωμάτων πρόσβασης χωρίς την ανάγκη αφαίρεσης και του ίδιου του κατόχου από την βάση δεδομένων. Στο πεδίο iKeyNumber αποθηκεύεται ο σειριακός αριθμός του κλειδιού που έχει παραχωρηθεί στον κάτοχο. Στο πεδίο PassCode αποθηκεύεται ο προσωπικός

κωδικός του κατόχου σε κρυπτογραφημένη μορφή από το λογισμικό διαχείρισης. Στο πεδίο Details, ο διαχειριστής του συστήματος ασφαλείας μπορεί να εισάγει επιπλέον σχόλια, που πιθανότατα να χρειάζεται για την τήρηση κανόνων ασφαλείας ή στοιχεία για τον ίδιο τον κάτοχο ως υπενθύμιση.

Δομή Πίνακα KeyMediaPool					
Field Name	Data Type	Field Size	Required	Allow Zero Length	Indexed
iKeyCode	Text	16	Yes	No	Yes (No Duplicates)

Στον πίνακα KeyMediaPool αποθηκεύονται τόσο τα διαθέσιμα κλειδιά ασφαλείας, όσο και αυτά που έχουν παραχωρηθεί σε άτομα που χρειάζεται να έχουν πρόσβαση στον ελεγχόμενο χώρο. Δηλαδή δίνεται η δυνατότητα στον διαχειριστή του συστήματος ελέγχου πρόσβασης να καταγράψει όλα τα κλειδιά ασφαλείας που έχει διαθέσιμα για να αποδώσει σε κατόχους που θα έχουν πρόσβαση στον ελεγχόμενο χώρο με την χρήση αυτών.

### *Βάση Δεδομένων Βαθμολογικών Πινάκων*

Για την προβολή των βαθμολογικών πινάκων των μαθημάτων γίνεται χρήση βάσεων δεδομένων σχεδιασμένες σε Microsoft Access. Η δομή που πρέπει να τηρούν τα αρχεία με τις βαθμολογίες είναι απλή.

Κάθε αρχείο πρέπει να περιέχει έναν πίνακα με ονομασία 'Βαθμολογίες'. Αν υπάρχουν άλλοι πίνακες αυτοί αγνοούνται.

Η δομή του πίνακα είναι ελεύθερη να περιέχει οποιονδήποτε αριθμό πεδίων και τύπων πληροφοριών, αρκεί να υπάρχουν τα συγκεκριμένα πεδία, που απαιτούνται για την ανακοίνωση των βαθμολογιών, με τήρηση της συγκεκριμένης ονοματολογίας πεδίων.

Δομή Πίνακα 'Βαθμολογίες'	
Field Name	Data Type
Επώνυμο	Text
Όνομα	Text
Βαθμός	Text

Εφόσον τηρούνται οι απλές αυτές απαιτήσεις δεν υπάρχει πρόβλημα προσπέλασης του βαθμολογικού πίνακα από το πρόγραμμα διαχείρισης του τερματικού, αλλά και εμφάνισης των βαθμολογιών στην οθόνη του τερματικού.

Λόγω περιορισμών της γλώσσας προγραμματισμού του λογισμικού του τερματικού ο βαθμολογικός πίνακας θα πρέπει να έχει μέγιστο αριθμό 248 ονομάτων. Αν ο αριθμός των εγγραφών υπερβαίνει το μέγιστο αυτό όριο, θα πρέπει ο διαχειριστής να διασπάσει τον βαθμολογικό πίνακα σε δύο ξεχωριστά αρχεία και να κάνει ανακοινώσιμες τις υπόλοιπες βαθμολογίες κάνοντας χρήση πιθανής ελεύθερης θέσης στον πίνακα μαθημάτων (μέγιστος αριθμός 8 μαθημάτων). Αν δεν προβεί ο διαχειριστής σε αυτή την ενέργεια, απλά οι εγγραφές πέρα του μεγίστου ορίου δεν θα γίνουν ανακοινώσιμες, χωρίς να δημιουργηθεί άλλο πρόβλημα.

# Κεφάλαιο 5<sup>ο</sup>

## Οδηγίες Χρήσης Τερματικού

Επειδή η χρήση του τερματικού είναι προφανής στα περισσότερα της σημεία, ανάλυση θα γίνει μόνο σε αυτά που δεν είναι εμφανή για τους απλούς χρήστες (σπουδαστές), παρά μόνο για τους κατόχους κλειδιών ασφαλείας και τον διαχειριστή.

Κατά την εισαγωγή του κλειδιού ασφαλείας στην υποδοχή #1, η οποία βρίσκεται στην πρόσοψη του κουτιού που φιλοξενεί την οθόνη και το πληκτρολόγιο, γίνεται προσπάθεια σύνδεσης με τον Η/Υ. Αν η προσπάθεια αυτή είναι επιτυχής, τότε ελέγχεται η ταυτότητα του κατόχου, σύμφωνα με τον σειριακό αριθμό του κλειδιού του. Εφόσον αναγνωρισθεί επιτυχώς ο κάτοχος του κλειδιού, στην οθόνη θα εμφανιστεί το ονοματεπώνυμο του και η προτροπή για εισαγωγή του προσωπικού του κωδικού, διαφορετικά αντί για τα στοιχεία του κατόχου θα εμφανιστεί ο σειριακός αριθμός του κλειδιού. Ο προσωπικός κωδικός του κατόχου πρέπει να εισαχθεί εντός χρονικού διαστήματος 20 δευτερολέπτων, αλλιώς θα ακυρωθεί η αίτηση πρόσβασης. Αν ο κάτοχος διαθέτει δικαιώματα πρόσβασης στον χώρο και ο προσωπικός του κωδικός είναι σωστός, τότε η πόρτα θα ξεκλειδώσει, δίδοντας ταυτόχρονα οπτική και ακουστική ένδειξη. Σε διαφορετική περίπτωση θα εμφανιστεί μήνυμα απόρριψης της αίτησης εισόδου, ενώ επίσης θα υπάρξει ηχητική σήμανση.

Αν αντί του πρώτου ψηφίου του προσωπικού κωδικού, πατηθεί το πλήκτρο «Α», τότε η αίτηση πρόσβασης μετατρέπεται σε εντολή για κλείδωμα της πόρτας. Φυσικά για την εκτέλεση της εντολής αυτής θα πρέπει ο κάτοχος να έχει δικαίωμα πρόσβασης στον χώρο. Αν αντίθετα πατηθεί το πλήκτρο «Β» τότε ο σειριακός αριθμός του κλειδιού που εισήχθη στην υποδοχή θα προστεθεί στην λίστα των διαθέσιμων κλειδιών του λογισμικού του Η/Υ. Η λειτουργία αυτή ισχύει για την αυτόματη εισαγωγή νέων κλειδιών στην λίστα, χωρίς την ανάγκη οπτικής ανάγνωσης του κωδικού του κλειδιού και χειροκίνητη εισαγωγή στην λίστα.

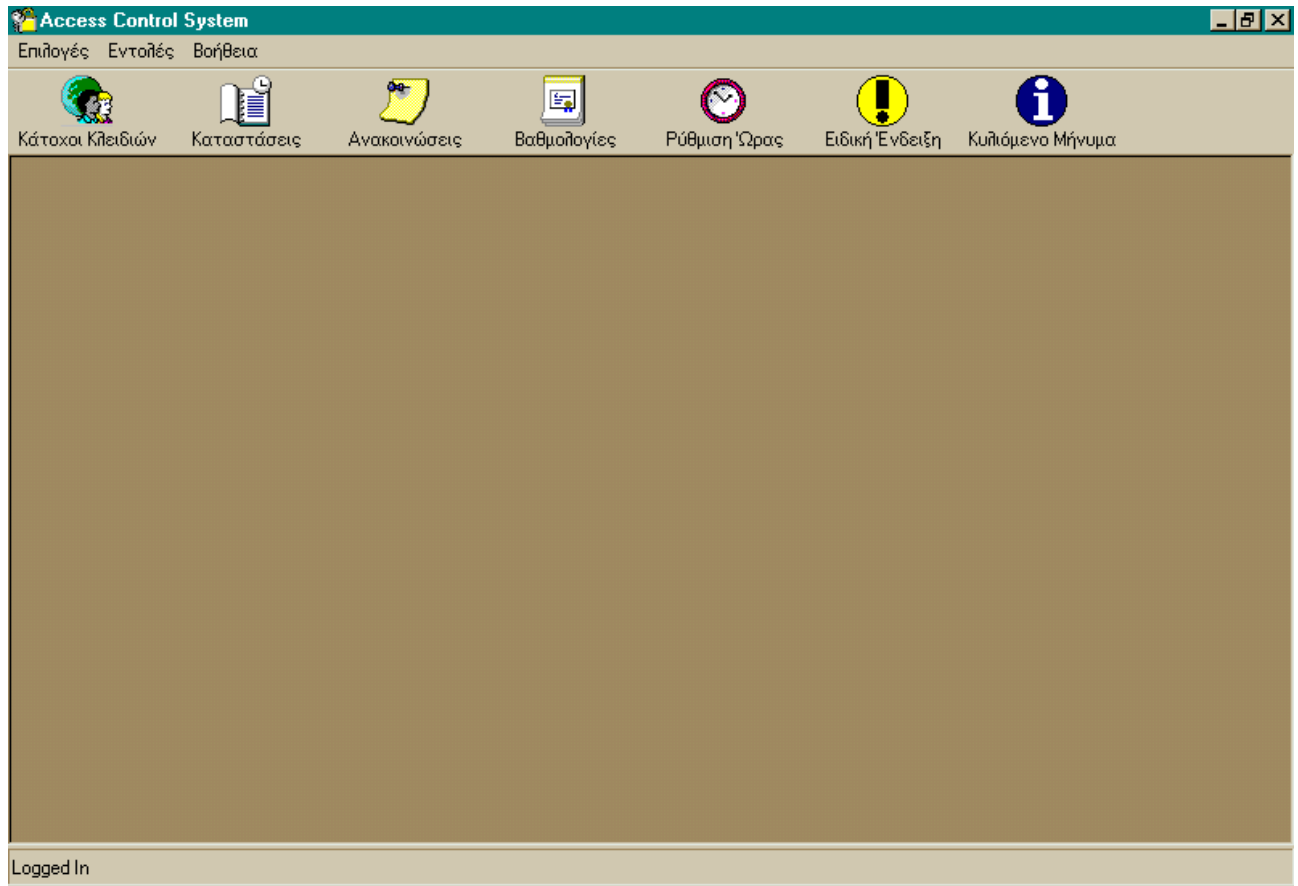
Η κατάσταση της κλειδαριάς εμφανίζεται με το αντίστοιχο σύμβολο, δίπλα ακριβώς από την ένδειξη της ημερομηνίας και ώρας.

Για την υποδοχή #2, όταν εισαχθεί κλειδί ασφαλείας, εκδίδεται αίτηση ξεκλειδώματος της πόρτας, η οποία επεξεργάζεται από τον Η/Υ. Αν ο κάτοχος του κλειδιού έχει δικαίωμα πρόσβασης στον χώρο, τότε η πόρτα ξεκλειδώνει από το εσωτερικό και μπορεί να εξέλθει. Διαφορετικά η αίτηση εξόδου απορρίπτεται. Δεν απαιτείται η εισαγωγή προσωπικού κωδικού, και για τον λόγο αυτό η υποδοχή αυτή, πρέπει να βρίσκεται σε σημείο εντός του χώρου.

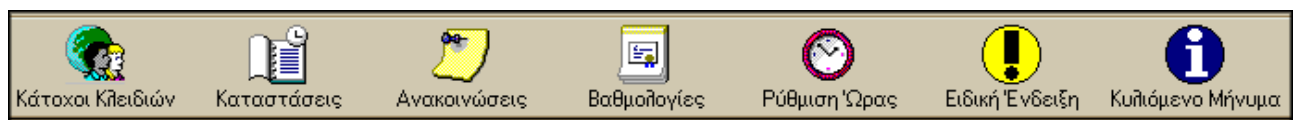
Το κύριο κλειδί, επιτρέπει στον κάτοχο του να αποκτήσει πρόσβαση στον ελεγχόμενο χώρο (είσοδο, έξοδο, κλείδωμα) χωρίς να απαιτείται η επικοινωνία με τον Η/Υ. Φυσικά πρέπει να έχει ήδη καθορισθεί το κύριο κλειδί από το λογισμικό του Η/Υ και να έχει ενημερωθεί το τερματικό για τον σειριακό αριθμό αυτού. Αν το κύριο κλειδί παραμείνει στην υποδοχή (#1 ή #2) για τρεις φορές περισσότερο χρόνο από αυτόν που απαιτείται για την ανάγνωση του κωδικού του, τότε η κατάσταση της κλειδαριάς εναλλάσσεται μεταξύ των δύο καταστάσεων (κλειδωμένη, ξεκλειδωτή). Για λόγους ασφαλείας το κύριο κλειδί αυτό πρέπει να είναι στην κατοχή του διαχειριστή ή να βρίσκεται εντός του ελεγχόμενου χώρου και να χρησιμοποιείται μόνο σε περιπτώσεις ανάγκης. Για λόγους ελέγχους η παρατεταμένη πίεση του πλήκτρου «0» για 8 περίπου δευτερόλεπτα θα προκαλέσει την επανεκκίνηση του τερματικού.

## Οδηγίες Χρήσης Λογισμικού Υπολογιστή

Εκτελούμενο το λογισμικό διαχείρισης του τερματικού εμφανίζει ένα εικονίδιο στην Tray Bar των Windows. Η επιλογή του εικονιδίου αυτού με το ποντίκι εμφανίζει ένα κεντρικό παράθυρο διαχείρισης όλων των λειτουργιών του λογισμικού ελέγχου του τερματικού.



Στο παράθυρο αυτό ξεχωρίζει η παρακάτω μπάρα εικονιδίων, με την οποία μπορεί να γίνει η εκτέλεση των συχνότερα χρησιμοποιούμενων εργασιών. Οι ίδιες εντολές-λειτουργίες μπορούν να εκτελεστούν από τα μενού επιλογών του ίδιου παραθύρου.



Οι επιλογές που δίδει το μενού με τίτλο «Επιλογές» είναι οι εξής:

Επιλογές	Εντολές	Βοήθεια
Κάτοχοι Κλειδιών Πρόσβασης		F5
Καταστάσεις Πρόσβασης		F6
Ανακοινώσεις		F7
Βαθμολογίες		F8
Log in		
Log out		
Έξοδος		Ctrl+X

- ❖ Η ενεργοποίηση της επιλογής «Κάτοχοι Κλειδιών Πρόσβασης» εμφανίζει ένα παράθυρο που περιέχει σε μορφή λίστας, όλους τους κατόχους κλειδιών ασφαλείας και έχει την παρακάτω μορφή.


Επώνυμο	Όνομα	Ιδιότητα	Ενεργό Κλειδί	Αριθμός Κλειδιού	Λήξης Εγκυρότητας
Καπετανάκης	Σπύρος	Συνεργάτης	Ενεργό	A7000005A430CE01	Τετάρτη, 12 Δεκέμβριος 2001
Κεντερλής	Παναγιώτης	Τεχνικό Προσωπικό	Ενεργό	E9000005A4025501	Σάββατο, 17 Νοέμβριος 2001

Προσθήκη   Αφαίρεση   Επεξεργασία   Κλείσιμο

Οι κάτοχοι για τους οποίους δεν υπάρχει δικαίωμα απόκτησης πρόσβασης στον ελεγχόμενο χώρο εμφανίζονται με χρώμα κόκκινο. Με διπλό πάτημα στο όνομα του κατόχου ή ενεργοποίηση του αντίστοιχου πλήκτρου, μπορεί να γίνει επεξεργασία των στοιχείων του κατόχου. Επίσης μπορεί να γίνει προσθήκη νέου κατόχου καθώς και αφαίρεση κάποιου κατόχου. Τα στοιχεία που αφορούν κάθε κάτοχο κλειδιού, εμφανίζονται σε ένα παράθυρο της παρακάτω μορφής.

**Κάτοχος Κλειδιού Πρόσβασης**

Προσωπικά Στοιχεία   Περισσότερα Στοιχεία

 Κεντερλής Παναγιώτης

Όνομα: Παναγιώτης

Επώνυμο: Κεντερλής

Ιδιότητα: Τεχνικό Προσωπικό

Διεύθυνση: Α. Ειρήνης 248  
Πέρασμα  
Πειραιάς  
Τ.Κ. 18863

Τηλέφωνο: 4415552

Κινητό: 0977586571


e-mail: panos@mprolab.teipir.gr

Ενεργό Κλειδί

Αποδοχή   Ακύρωση

**Κάτοχος Κλειδιού Πρόσβασης**

Προσωπικά Στοιχεία   Περισσότερα Στοιχεία



Ημ/νία Λήξης Ισχύος: Σάββατο, 17 Νοέμβριος 2001

Αριθμός Κλειδιού: E9000005A4025501

Κωδικός: \*\*\*\*\*

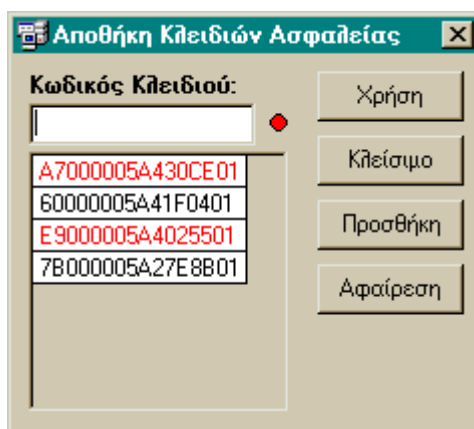
Παρατηρήσεις:  
Δημιουργός της πτυχιακής εργασίας.

Αποδοχή   Ακύρωση

---

Είναι δυνατός ο χρονικός περιορισμός της χρήσης του κλειδιού. Επίσης μπορεί να αλλαχτεί το κλειδί ασφαλείας και ο προσωπικός κωδικός του κατόχου. Ο κάτοχος μπορεί μέσω μιας απλής επιλογής να τύχει αποκλεισμού δικαιωμάτων πρόσβασης στον ελεγχόμενο χώρο.

Το πλήκτρο με το σήμα «i» εμφανίζει το παρακάτω παράθυρο.



Στο παράθυρο αυτό εμφανίζονται όλα τα κλειδιά που βρίσκονται υπό την ιδιοκτησία του ελεγχόμενου χώρου, και μπορεί να γίνει η επιλογή κάποιου για χρήση από κάτοχο. Με κόκκινο χρώμα εμφανίζονται τα κλειδιά που έχουν ήδη αποδοθεί σε κατόχους, ενώ με μαύρο τα διαθέσιμα. Μπορεί να γίνει χειροκίνητη εισαγωγή κάποιου κλειδιού, καθώς και αφαίρεση κλειδιού από την λίστα.

- ❖ Η ενεργοποίηση της επιλογής «Καταστάσεις Πρόσβασης», εμφανίζει ένα παράθυρο της παρακάτω μορφής.

Ημερομηνία	Ώρα	Ονοματεπώνυμο	Τύπος Πρόσβασης	Κατάσταση
Σάββατο, 24 Φεβρουάριος 2001	00:54:08	Καπετανάκης Σπύρος	Είσοδος	
	00:54:53	Καπετανάκης Σπύρος	Έξοδος	Κλειδωμα
	00:55:02	Καπετανάκης Σπύρος	Έξοδος	
	00:55:36	Καπετανάκης Σπύρος	Έξοδος	Κλειδωμα
	02:14:36	Καπετανάκης Σπύρος	Έξοδος	
	02:20:12	Καπετανάκης Σπύρος	Είσοδος	
Δευτέρα, 26 Φεβρουάριος 2001	18:21:29	Καπετανάκης Σπύρος	Είσοδος	
	05:23:24	Καπετανάκης Σπύρος	Είσοδος	
Σάββατο, 03 Μάρτιος 2001	19:23:12	Καπετανάκης Σπύρος	Είσοδος	
	19:24:43	Καπετανάκης Σπύρος	Είσοδος	
	19:26:19	Καπετανάκης Σπύρος	Είσοδος	
	19:26:39	Καπετανάκης Σπύρος	Είσοδος	
	19:28:26	Καπετανάκης Σπύρος	Έξοδος	Κλειδωμα
	19:28:31	Καπετανάκης Σπύρος	Έξοδος	
	19:28:39	Καπετανάκης Σπύρος	Έξοδος	Κλειδωμα
	19:28:42	Καπετανάκης Σπύρος	Έξοδος	

Στο παράθυρο αυτό, εμφανίζονται κατά χρονολογική σειρά οι αιτήσεις πρόσβασης που έγιναν αποδεκτές από το λογισμικό. Για κάθε αίτηση πρόσβασης γίνεται καταγραφή των στοιχείων Ημερομηνία, Ώρα, Ονοματεπώνυμο Κατόχου, Τύπος Πρόσβασης, Κατάσταση Κλειδαριάς.

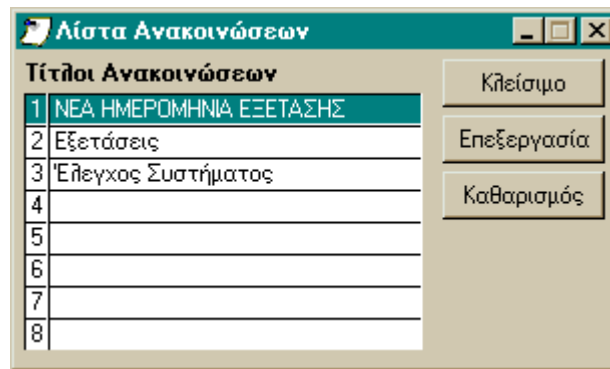
Εκτελώντας την λειτουργία «Αφαίρεση» μπορεί να γίνει επιλεκτική αφαίρεση των εγγραφών αιτήσεων πρόσβασης για συγκεκριμένες ημερομηνίες ή χρονικά διαστήματα.

Εκτελώντας την λειτουργία «Περιορισμοί» εμφανίζεται το παρακάτω παράθυρο.

Από το παράθυρο αυτό μπορεί να επιβληθεί περιορισμός στο χρονικό πλαίσιο ημερομηνιών που εμφανίζονται στην λίστα των Καταστάσεων Πρόσβασης. Επίσης με την επιλογή Αναφορά, μπορεί να δημιουργηθεί μια αναφορά προς εκτύπωση, με περιεχόμενο τις αιτήσεις πρόσβασης για το χρονικό πλαίσιο που έχει επιβληθεί. Η λειτουργία «Αναίρεση» εμφανίζει όλες τις εγγραφές που υπάρχουν στην βάση δεδομένων αιτήσεων πρόσβασης.

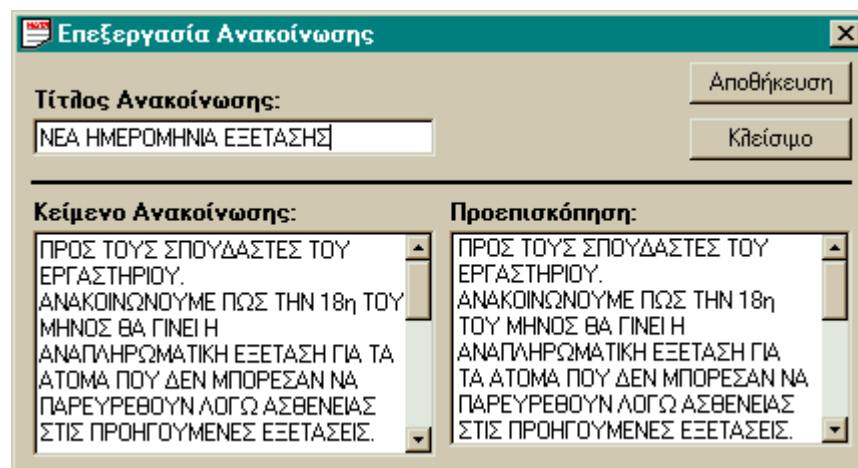


- ❖ Η ενεργοποίηση της επιλογής «Ανακοινώσεις», εμφανίζει ένα παράθυρο της παρακάτω μορφής.



Από το παράθυρο αυτό γίνεται η επιλογή για επεξεργασία των ανακοινώσεων που δημοσιεύονται από το τερματικό. Δυνατές επιλογές είναι η επεξεργασία του κειμένου και τίτλου της ανακοίνωσης, καθώς και ο καθαρισμός (αφαίρεση) της ανακοίνωσης από τον πίνακα ανακοινώσεων.

Εκτελώντας την λειτουργία «Επεξεργασία» θα εμφανιστεί ένα παράθυρο της παρακάτω μορφής.



Από το παράθυρο αυτό μπορεί να γίνει η επεξεργασία του κειμένου της ανακοίνωσης, καθώς και να αλλαχτεί ο τίτλος αυτής. Στο πλαίσιο Προεπισκόπηση εμφανίζεται το κείμενο της ανακοίνωσης διαμορφωμένο κατάλληλα, όπως ακριβώς θα εμφανιστεί στην οθόνη του τερματικού. Προκειμένου να γίνουν αποδεκτές οι αλλαγές, πρέπει να πατηθεί το πλήκτρο «Αποθήκευση».

Η αλλαγές που γίνονται στις ανακοινώσεις, είναι άμεσα εφαρμόσιμες και εμφανίζονται την επόμενη φορά που κάποιος θα χρησιμοποιήσει την λειτουργία «Ανακοινώσεις» του τερματικού.

- ❖ Η ενεργοποίηση της επιλογής «Βαθμολογίες» εμφανίζει ένα παράθυρο της παρακάτω μορφής.

Τίτλος Μαθήματος	Αρχείο Σπουδαστών	Δημοσίευση
Εργαστήριο Μικροεπεξεργαστών	C:\acs\theory1.mdb	Ενεργή
Εργαστήριο Οικ.μΕπεξεργαστών	C:\acs\theory2.mdb	Ανενεργή
Εργαστήριο Οικ. μΥπολογιστών		Ανενεργή
Θεωρία Μικροεπεξεργαστών	C:\acs\db1.mdb	Ενεργή
Θεωρία Οικ. μΕπεξεργαστών		Ανενεργή
Θεωρία Οικ. μΥπολογιστών		Ανενεργή
Θεωρία Βλάβες & Συντήρηση		Ενεργή
		Ανενεργή

**Αλλαγή Στοιχείων Μαθήματος**

1. **Εργαστήριο Μικροεπεξεργαστών** Αλλαγή

C:\acs\theory1.mdb

Δημοσίευση Βαθμολογιών

Από το παράθυρο αυτό είναι δυνατή η εμφάνιση και επεξεργασία των τίτλων, καθώς και πληροφοριών που σχετίζονται με τα μαθήματα και τους βαθμολογικούς πίνακες αυτών, που δημοσιεύονται από την αντίστοιχη λειτουργία του τερματικού. Από το σημείο αυτό γίνεται η σύνδεση του αρχείου βάσης δεδομένων που περιέχει τον πίνακα βαθμολογιών του μαθήματος, καθώς και η επιλογή, αν ο βαθμολογικός πίνακας είναι δημοσιεύσιμος ή όχι.

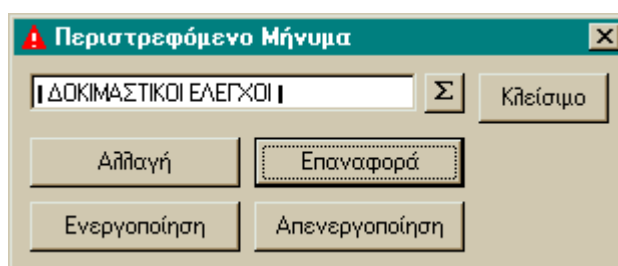
Η λειτουργία «Ενημέρωση» μεταφέρει τους τίτλους των μαθημάτων στο τερματικό. Πρέπει να εκτελείται κάθε φορά που γίνεται αλλαγή στους τίτλους μαθημάτων, προκειμένου να εμφανίζονται οι σωστές πληροφορίες στην οθόνη του τερματικού, όταν από αυτό ενεργοποιηθεί η λειτουργία «Βαθμολογίες». Μετά την επιτυχή ενημέρωση του τερματικού, εμφανίζεται μήνυμα επιβεβαίωσης. Αν αυτό δεν εμφανιστεί η ενημέρωση δεν εκτελέστηκε επιτυχώς.

- ❖ Η ενεργοποίηση των επιλογών «Log in/Log out» επιτρέπει την αύξηση της ασφάλειας χρήσης του λογισμικού. Στην κατάσταση «Not Logged In» όλες οι λειτουργίες του λογισμικού δεν είναι εκτελέσιμες από τον χρήστη του Η/Υ. Για να μπορέσει να γίνει χρήση αυτών, πρέπει ο χρήστης να πιστοποιήσει την ταυτότητα του ως διαχειριστής με την εισαγωγή ειδικού κωδικού και να βρεθεί το λογισμικό σε κατάσταση «Logged In».
- ❖ Η ενεργοποίηση της επιλογής «Εξοδος» τερματίζει την εκτέλεση του λογισμικού. Η επιλογή αυτή θα πρέπει να αποφεύγεται, γιατί αφήνει το τερματικό χωρίς έλεγχο από τον Η/Υ, απενεργοποιώντας όλες τις λειτουργίες του. Θα πρέπει να χρησιμοποιείται μόνο σε ειδικές περιπτώσεις κατά την κρίση του διαχειριστή του συστήματος.

Οι επιλογές που δίδει το μενού επιλογών με τίτλο «Εντολές» εμφανίζονται στην εικόνα.

Επιλογές	Εντολές	Βοήθεια
	Ρύθμιση Ώρας	Ctrl+T
	Ειδική Ένδειξη	Ctrl+S
	Κυλιόμενο Μήνυμα	Ctrl+M
	Διαγνωστικοί Έλεγχοι	Ctrl+D
	Αλλαγή Κωδικού Διαχειριστή	
	Καθορισμός Κύριου Κλειδιού	
	Αλλαγή Θύρας Επικοινωνίας	
	Κλείδωμα Πόρτας	Ctrl+L
	Ξεκλείδωμα Πόρτας	Ctrl+U

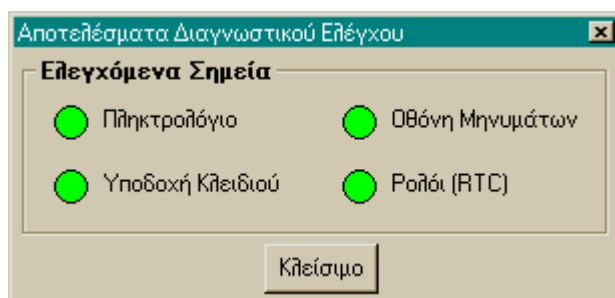
- ❖ Η ενεργοποίηση της εντολής «Ρύθμιση Ώρας», αποστέλλει την τρέχουσα ώρα και ημερομηνία στο τερματικό προκειμένου να επαναρυθμιστεί το RTC που διαθέτει. Η λειτουργία αυτή δεν είναι ιδιαίτερα απαραίτητη να εκτελεστεί από τον διαχειριστή, καθώς το ίδιο το λογισμικό του τερματικού ζητάει την εκτέλεση της λειτουργίας αυτής ανά χρονικό διάστημα μισής ώρας. Συγκεκριμένα η αυτόματη ρύθμιση της ώρας γίνεται ένα τέταρτο πριν και ένα τέταρτο μετά την αλλαγή της ώρας (π.χ. 12:15 και 12:45).
- ❖ Η ενεργοποίηση της εντολής «Ειδική Ένδειξη» εναλλάσσει την κατάσταση εμφάνισης του ειδικού μηνύματος («ΜΗΝ ΕΝΟΧΛΕΙΤΑΙ») στην οθόνη του τερματικού. Μετά από επιτυχή εκτέλεση της εντολής αυτής από το τερματικό εμφανίζεται μήνυμα επιβεβαίωσης και πληροφόρησης για την τρέχουσα κατάσταση εμφάνισης του ειδικού μηνύματος. Αν η εντολή δεν εκτελέστηκε εντός συγκεκριμένου χρονικού διαστήματος από το τερματικό θα εμφανιστεί μήνυμα πρόκλησης σφάλματος.
- ❖ Η ενεργοποίηση της εντολής «Κυλιόμενο Μήνυμα» θα εμφανίσει το παρακάτω παράθυρο.



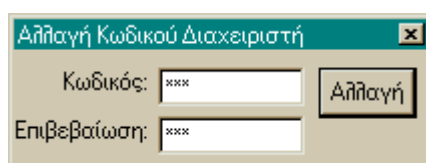
Από το παράθυρο αυτό μπορεί να γίνει η διαχείριση του εμφανιζόμενου κυλιόμενου μηνύματος στην οθόνη του τερματικού. Συγκεκριμένα μπορεί να γίνει η αλλαγή του μηνύματος με νέο κείμενο, επαναφορά του προκαθορισμένου μηνύματος («<http://mprolab.teipir.gr>»), ενεργοποίηση και απενεργοποίηση της εμφάνισης του μηνύματος. Με το πλήκτρο «Σ» γίνεται η εισαγωγή ειδικού χαρακτήρα στο κείμενο δίδοντας τον ASCII κωδικό, σύμφωνα με την κωδικοσελίδα 437.

- ❖ Η ενεργοποίηση της εντολής «Διαγνωστικοί Έλεγχοι», αποστέλλει στο τερματικό την εντολή εκτέλεσης διαγνωστικών ελέγχων. Όταν το τερματικό εκτελέσει τους διαγνωστικούς του ελέγχους, θα εμφανιστεί ένα παράθυρο στο οποίο σημειώνεται η κατάσταση του τερματικού. Τα ελεγχόμενα σημεία είναι το πληκτρολόγιο, οι υποδοχές των κλειδιών ασφαλείας, η οθόνη του τερματικού και το RTC (κατάσταση μπαταρίας αυτού). Τα σημεία που εκτέλεσαν επιτυχώς τον διαγνωστικό έλεγχο εμφανίζονται με πράσινο χρώμα, ενώ με κόκκινο χρώμα αυτά στα οποία εμφανίστηκε πρόβλημα. Αν κατά την εκτέλεση των διαγνωστικών ελέγχων γίνεται χρήση

του πληκτρολογίου, είναι πολύ πιθανό να εμφανιστεί εσφαλμένα ότι αυτό παρουσιάζει πρόβλημα λειτουργίας.



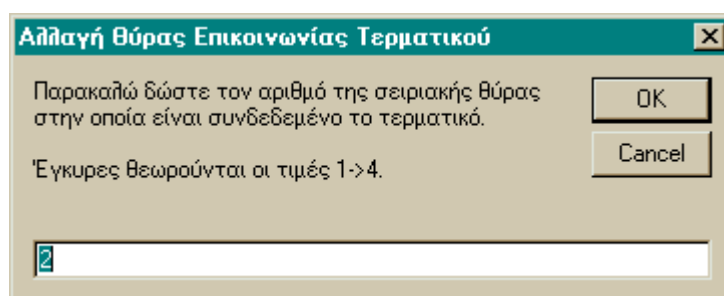
- ❖ Η ενεργοποίηση της εντολής «Αλλαγή Κωδικού Διαχειριστή» θα εμφανίσει ένα παράθυρο, στο οποίο πρέπει να δοθεί δυο φορές ο νέος κωδικός του διαχειριστή. Ο νέος αυτός κωδικός γίνεται άμεσα ενεργός και προστατεύεται από κρυπτογράφιση.



- ❖ Η ενεργοποίηση της εντολής «Καθορισμός Κύριου Κλειδιού», θα εμφανίσει ένα παράθυρο στο οποίο γίνεται ο χειροκίνητος καθορισμός του κυρίου κλειδιού, το οποίο δίνει δυνατότητα πρόσβασης στον ελεγχόμενο χώρο, χωρίς την ανάγκη ύπαρξης σύνδεσης με το λογισμικό διαχείρισης που εκτελείται στον Ηλεκτρονικό Υπολογιστή. Η επιλογή και ενεργοποίηση ενός κλειδιού ως κύριο κλειδί, αποθηκεύει τον σειριακό του αριθμό στην κρυπτογραφημένη και προστατευμένη μνήμη του μικροελεγκτή του τερματικού, αυξάνοντας έτσι την ασφάλεια του συστήματος. Ο αριθμός του κλειδιού αυτού δεν εμφανίζεται πουθενά αλλού στον υπολογιστή και για λόγους ασφαλείας καλό θα είναι να μην βρίσκεται στην λίστα των διαθέσιμων κλειδιών ασφαλείας.



- ❖ Η ενεργοποίηση της εντολής «Αλλαγή Θύρας Επικοινωνίας» δίνει την δυνατότητα στον διαχειριστή του συστήματος, να τοποθετήσει την σύνδεση του τερματικού σε θύρα επικοινωνίας διαφορετική της προεπιλεγμένης COM2.



- ❖ Η ενεργοποίηση των εντολών «Κλείδωμα Πόρτας»/«Ξεκλείδωμα Πόρτας», αποστέλλει αντίστοιχα στο τερματικό την εντολή κλειδώματος/ξεκλειδώματος της κλειδαριάς, του

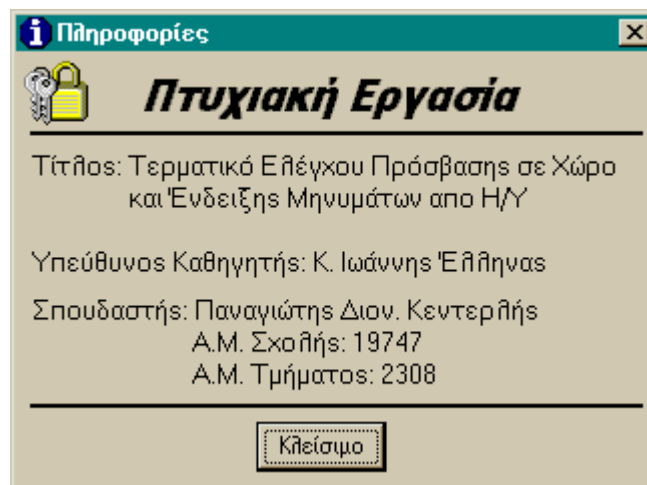
---

ελεγχόμενου χώρου. Αν η εντολή είναι επιτυχής, τότε θα εμφανιστεί μήνυμα επιβεβαίωσης. Σε άλλη περίπτωση θα εμφανιστεί κάποιο μήνυμα ένδειξης πρόκλησης σφάλματος. Οι συγκεκριμένες εντολές δεν καταγράφονται στην λίστα Καταστάσεων Πρόσβασης και για τον λόγο αυτό θα πρέπει να γίνεται χρήση τους σε ελάχιστες περιπτώσεις.

Το μενού επιλογών με τίτλο «Βοήθεια» περιέχει μόνο μια επιλογή.



- ❖ Η εκτέλεση της επιλογής αυτής, εμφανίζει το παρακάτω παράθυρο που περιέχει πληροφορίες για την πτυχιακή εργασία.



# Κεφάλαιο 6<sup>ο</sup>

## Κώδικας Λογισμικού του Τερματικού

### MAIN.C

```
#include <ds5000.h>
#include <t6963c.h>
#include <ibutton.h>
#include <serport.h>
#include <headers.h>
#include <func.h>
#include <keyb.h>
#include <rtc.h>

// -----
// Ρουτίνα εξυπηρέτησης διακοπής του Timer0.
// Κατά την εκτέλεση της γίνονται οι έλεγχοι για εισαγωγή
// i-Button στις υποδοχές #1 και #2 καθώς και χειρισμός
// της λειτουργίας αίτησης πρόσβασης στον χώρο.
// Επίσης καλείται η ρουτίνα διαχείρισης του κεντρικού
// καταλόγου επιλογών του τερματικού. Γίνεται έλεγχος
// για την κατάσταση της μπαταρίας του RTC, την ένδειξη
// του ειδικού μηνύματος προειδοποίησης, και εμφάνιση
// του περιστρεφόμενου μηνύματος.
// -----
void Timer0Int(void) interrupt 1
{
    unsigned char i;

    HandleMainMenu(KeyDetect());
    if ((MarqueeMode & 0x0f)==ShowMarquee)
    {
        if (MarqueeDelay==0)
        {
            Marquee();
            MarqueeDelay=7;
        }
        else MarqueeDelay--;
    }
    if (Check_RTC_Battery()==false)
    {
        DisableRTCInt();
        TextGoto(8,3);
        PrintString("Replace RTC. Battery Low.");
        EnableRTCInt();
    }
    iButton2();
    if (iButtonFound==true)
    {
        SoundBuzzer();
        if (IsMasterKey())
        {
            HandleMasterKeyInsertion();
        }
    }
    else

```

```

        {
            if (DoorIsLocked==true)
            {
                SendString("ATEXRQ-");
                Send_iButtonData();
                SendChar(ComVT);
            }
        }
    }
    iButton();
    if (iButtonFound==true)
    {
        SoundBuzzer();
        if (IsMasterKey())
        {
            HandleMasterKeyInsertion();
        }
        else
        {
            ButtonSelect(SelButton,0xff);
            ComProgressBar();
            ErrorReturned=false;
            KeyOwnerIDStored=false;
            if (ServerOnLine)
            {
                SendString("ATID");
                Send_iButtonData();
                SendChar(ComVT);
                while
                ((KeyOwnerIDStored==false)&&(ErrorReturned==false))
                {
                    }
            }
            else goto Skip_All;
            ClearMainScreen();
            CreateWindow(6,5,11,26);
            CreateDivider(8,5,21);
            DisableRTCInt();
            TextGoto(7,6);
            PrintString("Εισάγετε τον Κωδικό.");
            TextGoto(9,6);
            if (ErrorReturned==true)
            {
                PrintString("ID:");
                for(i=8;i>0;i--)
                {
                    PrintHex(iButtonData[i-1]);
                }
            }
            if (KeyOwnerIDStored)
            {
                SoundBuzzer();
                PrintString(KeyOwnerID);
            }
            TextGoto(10,6);
            PrintString("Κωδικός:");
            Secs2Delay=20;
            TextGoto(10,15);
            PrintChar('_');
            EnableRTCInt();
            ReEnter1:PassCode[0]=WaitForKeyPress();
            if (TimedOut==true) goto Skip_All2;
            if (((PassCode[0]>='0')&&(PassCode[0]<='9'))||
                ((PassCode[0]=='A')||((PassCode[0]=='B')))==false)
                goto ReEnter1;
        }
    }
}

```

```

ShowClock=false;
TextGoto(10,15);
PrintChar('*');
if ((PassCode[0]!='A')&&(PassCode[0]!='B'))
{
    for (i=1;i<6;i++)
    {
        TextGoto(10,15+i);
        SetCursorPosition(10,15+i);
        PrintChar('_');
        ShowClock=true;
        ReEnter:PassCode[i]=WaitForKeyPress();

        ShowClock=false;
        if (TimedOut==true) goto Skip_All2;

if((PassCode[i]<'0')||(PassCode[i]>'9')) goto ReEnter;
        TextGoto(10,15+i);
        PrintChar('*');

    }
    ClearMainScreen();
    WaitForRequestReply=true;
    ShowClock=true;
    SendString("ATREQ-");
    Send_iButtonData();
    for(i=0;i<6;i++) SendChar(PassCode[i]);
    SendChar(ComVT);
    while (WaitForRequestReply)
    {
        }
    if (ErrorReturned==false)
    {
        CreateWindow(7,3,10,28);
        TextGoto(8,4);
        PrintString("Καλωσήρθατε στο
μProLab.");
        TextGoto(9,5);
        PrintString("Η πόρτα είναι ανοικτή.");
        DelaySecs(2);

    }
    else
    {
        CreateWindow(7,6,9,25);
        TextGoto(8,7);
        PrintString("Δεν έχετε πρόσβαση");

    }
}
else
{
    if (PassCode[0]=='A')
    {
        SendString("ATLRQ-");
        Send_iButtonData();
        SendChar(ComVT);

    }
    else
    {
        SendString("ATKMP-");
        Send_iButtonData();
        SendChar(ComVT);

    }
    goto Skip_All2;
}
}

```



```

        Skip_All:
            BuzzerOn();
            DelaySecs(2);
            BuzzerOff();
        Skip_All2:
            ClearMainScreen();
            MainMenu();
            ShowClock=true;
    }
}
if (CIPtemp!=ClassInProgress) CIPMessage();
TH0=0x9f;
TL0=0xff;
}

// -----
// Ρουτίνα αρχικοποίησης συσκευών και λειτουργιών του
// τερματικού. Εμφάνιση του γραφικού λογότυπου του Ιδρύματος.
// Αρχικοποίηση της οθόνης LCD. Τοποθέτηση του τερματικού σε
// ατέρμονα βρόχο ώστε όλες οι λειτουργίες του Interface να
// εκτελούνται από ρουτίνες εξυπηρέτησης διακοπών.
// -----
void main ()
{
    ResetWDTimer();
    DisableRTCInt();
    InitRTC();
    iDataReset();
    IP=0x11;
    SP=0x40;
    IE=0x80;
    TH0=0x7f;
    TL0=0xff;
    TF0=false;
    ResetLCD();
    InitLCD();
    SerInit();
    ClearTextScreen();
    LoadFullScreen();
    LoadCGRAM();
    ShowClock=false;
    EnableRTCInt();
    DelaySecs(4);
    DisableRTCInt();
    ClearGraphicsScreen();
    DrawInitScreen();
    EnableRTCInt();
    SendTerminalCommand("ATTA");
    EnableSerialInt();
    iDetectionDelay=0;
    MarqueeDelay=4;
    ChangeTimeAt=0x15;
    if ((MarqueeMode & 0xf0)!=ShowMarqueeInRAM || (MarqueeMsg[28]!=ComEOS))
InitMarquee();
    Count2Reset=0x3f;
    KeybDelay=0;
    EnableTimer0Int();
    ShowClock=true;
    while (true)
    {
    }
}

```

## DS5000.C

```
#include <ds5000.h>

// -----
// Απενεργοποίηση του WatchDog Timer του μικροελεγκτή.
// -----
void ResetWDTimer(void)
{
    _asm
        push acc
        mov acc,#0x55
        mov TA,#0xaa
        mov TA,acc
        mov acc,pcon
        orl acc,#0x2
        mov pcon,acc
        pop acc
    _endasm;
}

// -----
// Ρουτίνα εκτέλεσης Software Reset του μικροελεγκτή.
// -----
void SoftResetMicro (void)
{
    _asm
        mov IE,#0x0
        pop ACC
        pop ACC
        mov ACC,#0x00
        push ACC
        push ACC
        reti
    _endasm;
}
```

## DS5000.H

```
/*-----  
Register Declarations for 8051 Processor  
  
    Written By - Sandeep Dutta . sandeep.dutta@usa.net (1998)  
  
This program is free software; you can redistribute it and/or modify it  
under the terms of the GNU General Public License as published by the  
Free Software Foundation; either version 2, or (at your option) any  
later version.  
  
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.  
  
You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.  
  
In other words, you are welcome to use, share and improve this program.  
You are forbidden to forbid anyone else to use, share and improve  
what you give them. Help stamp out software-hoarding!  
-----*/  
  
#ifndef REG51_H  
#define REG51_H  
  
#define TA 0xc7  
#define rwt 0xbf  
  
void SoftResetMicro (void);  
void ResetWDTimer(void);  
  
/* BYTE Register */  
sfr at 0x80 P0 ;  
sfr at 0x90 P1 ;  
sfr at 0xA0 P2 ;  
sfr at 0xB0 P3 ;  
sfr at 0xD0 PSW ;  
sfr at 0xE0 ACC ;  
sfr at 0xF0 B ;  
sfr at 0x81 SP ;  
sfr at 0x82 DPL ;  
sfr at 0x83 DPH ;  
sfr at 0x87 PCON ;  
sfr at 0x88 TCON ;  
sfr at 0x89 TMOD ;  
sfr at 0x8A TL0 ;  
sfr at 0x8B TL1 ;  
sfr at 0x8C TH0 ;  
sfr at 0x8D TH1 ;  
sfr at 0x8E AUXR ;  
sfr at 0xA8 IE ;  
sfr at 0xB8 IP ;  
sfr at 0x98 SCON ;  
sfr at 0x99 SBUF ;  
  
/* BIT Register */  
/* PSW */  
sbit at 0xD7 CY ;  
sbit at 0xD6 AC ;  
sbit at 0xD5 F0 ;  
sbit at 0xD4 RS1 ;  
sbit at 0xD3 RS0 ;
```

```

sbit at 0xD2 OV ;
sbit at 0xD0 P ;

/* TCON */
sbit at 0x8F TF1 ;
sbit at 0x8E TR1 ;
sbit at 0x8D TF0 ;
sbit at 0x8C TR0 ;
sbit at 0x8B IE1 ;
sbit at 0x8A IT1 ;
sbit at 0x89 IE0 ;
sbit at 0x88 IT0 ;

/* IE */
sbit at 0xAF EA ;
sbit at 0xAC ES ;
sbit at 0xAB ET1 ;
sbit at 0xAA EX1 ;
sbit at 0xA9 ET0 ;
sbit at 0xA8 EX0 ;

/* IP */
sbit at 0xBC PS ;
sbit at 0xBB PT1 ;
sbit at 0xBA PX1 ;
sbit at 0xB9 PT0 ;
sbit at 0xB8 PX0 ;
sbit at 0xBF RWT ;

/* P3 */
sbit at 0xB7 RD ;
sbit at 0xB6 WR ;
sbit at 0xB5 T1 ;
sbit at 0xB4 T0 ;
sbit at 0xB3 INT1 ;
sbit at 0xB2 INT0 ;
sbit at 0xB1 TXD ;
sbit at 0xB0 RXD ;
sbit at 0xB7 P3_7 ;
sbit at 0xB6 P3_6 ;
sbit at 0xB5 P3_5 ;
sbit at 0xB4 P3_4 ;
sbit at 0xB3 P3_3 ;
sbit at 0xB2 P3_2 ;
sbit at 0xB1 P3_1 ;
sbit at 0xB0 P3_0 ;

/* P1 */
sbit at 0x90 P1_0 ;
sbit at 0x91 P1_1 ;
sbit at 0x92 P1_2 ;
sbit at 0x93 P1_3 ;
sbit at 0x94 P1_4 ;
sbit at 0x95 P1_5 ;
sbit at 0x96 P1_6 ;
sbit at 0x97 P1_7 ;

/* P2 */
sbit at 0xA0 P2_0 ;
sbit at 0xA1 P2_1 ;
sbit at 0xA2 P2_2 ;
sbit at 0xA3 P2_3 ;
sbit at 0xA4 P2_4 ;
sbit at 0xA5 P2_5 ;
sbit at 0xA6 P2_6 ;
sbit at 0xA7 P2_7 ;

```

---

---

```
/* SCON */
sbit at 0x9F SM0 ;
sbit at 0x9E SM1 ;
sbit at 0x9D SM2 ;
sbit at 0x9C REN ;
sbit at 0x9B TB8 ;
sbit at 0x9A RB8 ;
sbit at 0x99 TI ;
sbit at 0x98 RI ;

/* TMOD bits */
#define GATE1 (1<<7)
#define C_T1 (1<<6)
#define M1_1 (1<<5)
#define M0_1 (1<<4)
#define GATE0 (1<<3)
#define C_T0 (1<<2)
#define M1_0 (1<<1)
#define M0_0 (1<<0)

#endif
```

## IBUTTON.C

```
#include <ds5000.h>
#include <serport.h>
#include <headers.h>
#include <ibutton.h>
#include <func.h>

// -----
// Ρουτίνα καθαρισμού δεδομένων από προηγούμενη εισαγωγή
// i-Button στο κύκλωμα.
// -----
void iDataReset(void)
{
    unsigned char i;
    Wait=0;
    Times=0;
    for (i=0;i<8;i++)
    {
        iButtonData[i]=0;
    }
}

// -----
// Ρουτίνα πρόκλησης κατάστασης Reset στον δίαυλο 1-Wire
// που συνδέονται τα i-Button για την υποδοχή #1.
// -----
void TouchReset (void)
{
    _asm
        push b
        push acc
        mov a,#0x4
        clr iPin
        mov b,#221
0000$:    djnz b,0000$
        setb iPin
        mov b,#0x6
        clr c
0001$:    jb iPin,0002$
        djnz b,0001$
        djnz acc,00001$
        sjmp 0004$
0002$:    mov b,#111
0003$:    orl c,/iPin
        djnz b,0003$
0004$:    pop acc
        pop b
    _endasm;
}

// -----
// Ρουτίνα ανάγνωσης πληροφοριών από τον δίαυλο 1-Wire
// για την υποδοχή #1.
// -----
unsigned char iDataIn (unsigned char status)
{
    if (status==1) ACC=0xff;
    if (status==0) ACC=0x0f;
}
```

```

_asm
    push b
    mov b,#0x8
0001$:    rrc a

0002$:    clr iPin
    nop
    nop
    nop
    nop
    mov iPin,C
    nop
    nop
    nop
    nop
    nop
    nop
    mov c,iPin
    push b
    mov b,#0x12
0003$:    djnz b,0003$
    pop b
    setb iPin
    djnz b,0001$
    rrc a
    pop b
_endasm;
return ACC;
}

// -----
// Ρουτίνα υπολογισμού του Cyclic Redundancy Check Byte
// των περιεχομένων του εισαχθέντος i-Button.
// -----
void CRC_Check (unsigned char CRCtemp)
{
    ACC=CRCtemp;
    _asm
        push aTemp
        push acc
        mov dptr,#aCRC
        movx a,@dptr
        mov aTemp,a
        pop acc

        push acc
        push b
        push acc
        mov b,#0x8
0000$:    xrl a,aTemp
        rrc a
        mov a,aTemp
        jnc 0001$
        xrl a,#0x18
0001$:    rrc a
        mov aTemp,a
        movx @dptr,a
        pop acc
        rr a
        push acc
        djnz b,0000$
        pop acc
        pop b

```

```

        pop acc
        pop aTemp
    _endasm;
}

// -----
// Ρουτίνα χειρισμού του δίαυλου 1-Wire των i-Button
// για αναγνώριση και ανάγνωση του μοναδικού σειριακού κωδικού
// που περιέχεται στον τύπο DS1990. Χρήση της υποδοχής #2.
// -----
unsigned char iButton (void) critical
{
    unsigned char i;

    EnableOnlySerialInt();
    iButtonFound=false;
    if ((Times==0) || (Wait==false))
    {
        TR0=0;
        Wait=false;
        TouchReset();
        cCRC=0x0;
        iDataIn(0);
        iButtonData[0]=iDataIn(1);
        CRC_Check(iButtonData[0]);
        if (cCRC==0x5e)
        {
            for (i=1;i<7;i++)
            {
                iButtonData[i]=iDataIn(1);
                CRC_Check(iButtonData[i]);
            }
            iButtonData[7]=iDataIn(1);
            if (cCRC == iButtonData[7])
            {
                Wait=true;
                iButtonFound=true;
            }
        }
        skipall:Times=iDelay;
        TF0=0;
        TR0=1;
    }

    if (Wait==true) Times--;
    RestoreInterrupts();
}

// -----
// Ρουτίνα πρόκλησης κατάστασης Reset στον δίαυλο 1-Wire
// που συνδέονται τα i-Button για την υποδοχή #2.
// -----
void TouchReset2 (void)
{
    _asm
        push b
        push acc
        mov a,#0x4
        clr iPin2
        mov b,#221
0000$:    djnz b,0000$
        setb iPin2

```



```

        mov b,#0x6
        clr c
0001$:    jb iPin2,0002$
          djnz b,0001$
          djnz acc,00001$
          sjmp 0004$
0002$:    mov b,#111
0003$:    orl c,/iPin2
          djnz b,0003$
0004$:    pop acc
          pop b
        _endasm;
}

// -----
// Ρουτίνα ανάγνωσης πληροφοριών από τον δίαυλο 1-Wire
// για την υποδοχή #2.
// -----
unsigned char iDataIn2 (unsigned char status)
{
    if (status==1) ACC=0xff;
    if (status==0) ACC=0x0f;

    _asm
        push b
        mov b,#0x8
0001$:    rrc a

0002$:    clr iPin2
          nop
          nop
          nop
          nop
          mov iPin2,C
          nop
          nop
          nop
          nop
          nop
          nop
          mov c,iPin2
          push b
          mov b,#0x12
0003$:    djnz b,0003$
          pop b
          setb iPin2
          djnz b,0001$
          rrc a
          pop b
    _endasm;
    return ACC;
}

// -----
// Ρουτίνα χειρισμού του δίαυλου 1-Wire των i-Button
// για αναγνώριση και ανάγνωση του μοναδικού σειριακού κωδικού
// που περιέχεται στον τύπο DS1990. Χρήση της υποδοχής #2.
// -----
unsigned char iButton2 (void) critical
{
    unsigned char i;

```

```

EnableOnlySerialInt();
iButtonFound=false;
if ((Times==0) || (Wait==false))
{
    TR0=0;
    Wait=false;
    TouchReset2();
    cCRC=0x0;
    iDataIn2(0);
    iButtonData[0]=iDataIn2(1);
    CRC_Check(iButtonData[0]);
    if (cCRC==0x5e)
    {
        for (i=1;i<7;i++)
        {
            iButtonData[i]=iDataIn2(1);
            CRC_Check(iButtonData[i]);
        }
        iButtonData[7]=iDataIn2(1);
        if (cCRC == iButtonData[7])
        {
            Wait=true;
            iButtonFound=true;
        }
    }
    skipall:Times=iDelay;
    TF0=0;
    TR0=1;
}

if (Wait==true) Times--;
RestoreInterrupts();
}

// -----
// Ρουτίνα σειριακής αποστολής των δεδομένων που ανεγνώσθησαν
// από το εισαχθέν i-Button προς την εφαρμογή διαχείρισης του τερματικού.
// -----
void Send_iButtonData (void)
{
    unsigned int i;
    for(i=8;i>0;i--) SendHex(iButtonData[i-1]);
}

```

---

---

## IBUTTON.H

```
#define iDelay 0x2f
#define iPin   P2.7
#define iPin2  P2.6
#define aCRC   0xfb10
#define aTemp  0x7f

// -----
// Δηλώσεις ρουτινών-συναρτήσεων.
// -----

void Send_iButtonData (void);
void iDataReset(void);
void TouchReset (void);
unsigned char iDataIn (unsigned char);
void CRC_Check (unsigned char);
unsigned char iButton (void) critical;
void TouchReset2 (void);
unsigned char iDataIn2 (unsigned char);
unsigned char iButton2 (void) critical;
```

## KEYB.C

```
#include <ds5000.h>
#include <headers.h>
#include <keyb.h>
#include <func.h>
#include <serport.h>

// -----
// Δηλώσεις σταθερών τιμών για σάρωση και αναγνώριση
// πατημένου πλήκτρου στο πληκτρολόγιο μήτρας που συνδέεται
// στην αμφίδρομη πόρτα P1.
// -----

code unsigned char ReturnCode[4]={0x0e,0x0d,0x0b,0x07};
code unsigned char ScanPattern[4]={0xef,0xdf,0xbf,0x7f};
code unsigned char ScanCode[4][4]={
    {'1','4','7','*'},
    {'2','5','8','0'},
    {'3','6','9','#'},
    {'A','B','C','D'};

// -----
// Χρονική Καθυστέρηση για το φιλτράρισμα του πληκτρολογίου.
// Η καθυστέρηση ορίζεται από τη τιμή HighTimer-LowTimer
// η οποία δίνει καθυστέρηση περίπου 13msecs.
// -----
void FilterDelay (void)
{
    unsigned char temp1, temp2;
    ET0=0;
    temp1=TH0;
    temp2=TL0;
    TH0=HighTimer;
    TL0=LowTimer;
    TF0=0;
loop: if (TF0==false) goto loop;
    TF0=false;
    ET0=1;
    TH0=temp1;
    TL0=temp2;
}

// -----
// Ρουτίνα αναγνώρισης πληκτρολόγησης. Αν δεν βρεθεί
// πατημένο πλήκτρο γίνεται έξοδος από την ρουτίνα.
// -----
unsigned char KeyDetect (void)
{
    unsigned char i, x, y;
    unsigned char KeyPressed;

    i=0;

    P1=KeybCalm;
    if (KeybDelay==0)
    {
        KeyPressed=false;
        if (P1!=KeybCalm)
```

```

        {
            y=0;
            FilterDelay();
            for (x=0;x<4;x++)
            {
                P1=ScanPattern[x];
                i=P1 & KeybCalm;
                for (y=0;y<4;y++)
                {
                    if (i==ReturnCode[y])
                    {
                        KeyPressed=true;
                        break;
                    }
                }
                if (KeyPressed==true) break;
            }
        }
        i=0;
        P1=KeybCalm;
        if (KeyPressed==true)
        {
            i=ScanCode[y][x];
            KeybDelay=KeyRepDelay;
        }
    }
    else
    {
        KeybDelay--;
    }
    return i;
}

// -----
// Ρουτίνα αναγνώρισης πληκτρολόγησης. Αναμονή μέχρι να
// πατηθεί κάποιο πλήκτρο και έξοδος από την ρουτίνα.
// -----
unsigned char WaitForKeyPress (void)
{
    unsigned char i, x, y;
    unsigned char KeyPressed;

    P1=KeybCalm;
    KeyPressed=false;
    TimedOut=false;
    while (KeyPressed==false)
    {
        y=0;
        for (x=0;x<4;x++)
        {
            P1=ScanPattern[x];
            i=P1 & KeybCalm;
            for (y=0;y<4;y++)
            {
                if (TimedOut==true) goto Escape;
                if (i==ReturnCode[y])
                {
                    KeyPressed=true;
                    break;
                }
            }
        }
    }
}

```

```

        if (KeyPressed==true) break;
    }
}
P1=KeybCalm;
i=ScanCode[y][x];
if (KeyPressed==true) SoundBuzzer();
while (P1!=KeybCalm)
{
    FilterDelay();
}
Escape:
return i;
}

```

## KEYB.H

```

// -----
// Δηλώσεις σταθερών τιμών
// -----

#define LowTimer  0x00  // low byte της χρονικής καθυστέρησης φιλτραρίσματος
#define HighTimer 0xe0  // high byte της χρονικής καθυστέρησης φιλτραρίσματος
#define KeybCalm  0x0f  // αναγνώσιμη τιμή όταν δεν υπάρχει πατημένο πλήκτρο
#define KeyRepDelay 2  // αριθμός επαναλήψεων μετά απο αναγνώριση πλήκτρου

// -----
// Δηλώσεις ρουτινών-συναρτήσεων.
// -----

void FilterDelay (void);
unsigned char KeyDetect (void);
unsigned char WaitForKeyPress (void);

```

## T6963C.C

```
#include <t6963c.h>
#include <headers.h>
#include <ds5000.h>
#include <func.h>
#include <rtc.h>

// -----
// Δηλώσεις σύνδεσης με πίνακες εξωτερικών δεδομένων στη
// μνήμη προγράμματος.
// -----
extern code unsigned char cgram[];
extern code unsigned char teilogo[];
extern code unsigned char title[];
extern code unsigned char lock[];

// -----
// Ρουτίνα αρχικοποίησης της περιοχής Character Generator RAM
// του ελεγκτή της οθόνης LCD.
// -----
void LoadCGRAM(void)
{
    unsigned int cgcount;
    DataTransfer(0,CGRAMBEGIN);
    SendCommand(ADPSET);
    SendCommand(AWRON);
    for (cgcount=0;cgcount<2048;cgcount++)
    {
        AutoDataWrite(cgram[cgcount]);
    }
    SendCommand(AWROFF);
}

// -----
// Ρουτίνα εμφάνισης του γραφικού λογότυπου του Ιδρύματος
// στην οθόνη LCD.
// -----
void LoadFullScreen(void)
{
    unsigned int count;
    SetPosition(GRAFHOME_ADDR);
    SendCommand(AWRON);
    for (count=0;count<3840;count++)
    {
        AutoDataWrite(teilogo[count]);
    }
    SendCommand(AWROFF);
}

// -----
// Ρουτίνα Hardware Reset του ελεγκτή της οθόνης LCD.
// Ο ακροδέκτης /Reset του ελεγκτή συνδέεται με τον ακροδέκτη
// P2.5 του μικροελεγκτή.
// -----
void ResetLCD()
{
    _asm
        clr p2.5
```

```

        nop
        nop
        nop
        nop
        nop
        setb p2.5
    _endasm;
}

// -----
// Ρουτίνα αρχικοποίησης κατάστασης λειτουργίας του ελεγκτή T6963C
// της οθόνης LCD.
// -----
void InitLCD(void)
{
    DataTransfer(0,0);
    SendCommand(AWROFF);
    DataTransfer(0,TEXTBEGIN); // set text home
    SendCommand(TXHOME);
    DataTransfer(TEXTCOLUMNS,0); // set number of columns
    SendCommand(TXAREA);
    DataTransfer(0,GRAFHOME_HI); //
    SendCommand(GRHOME);
    DataTransfer(GRAFCOLUMNS,0);
    SendCommand(GRAREA);
    DataTransfer(CGRAMOFFSET,0); // char gen memory offset
    SendCommand(OFFSET);
    DataTransfer(0,0);
    SendCommand(MODESET + CGRAM + XR_GRAF);
    DataTransfer(0,0);
    SendCommand(DISPLAY + TEXT_ON + GRAF_ON + CURS_DISPLAY);
    DataTransfer(0,TEXTBEGIN);
    SendCommand(CURSET);
    SendCommand(CURBLK+7);
    DataTransfer(0,TEXTBEGIN);
    SendCommand(ADPSET);
}

// -----
// Ρουτίνα καθαρισμού περιοχής χαρακτήρων-κειμένου της οθόνης LCD.
// -----
void ClearTextScreen(void)
{
    unsigned int cells;
    DataTransfer(0,TEXTBEGIN);
    SendCommand(ADPSET);
    SendCommand(AWRON);
    cells=0x800;
    while(cells--)
    { AutoDataWrite(0); }
    SendCommand(AWROFF);
}

// -----
// Ρουτίνα καθαρισμού της περιοχής γραφικών της οθόνης LCD.
// -----
void ClearGraphicsScreen(void)
{
    unsigned int cells;
    SetPosition(GRAFHOME_ADDR);
    SendCommand(AWRON);
}

```



```

        cells=GRAF_COLUMNS * GRAF_ROWS;
        while (cells--)
            {AutoDataWrite(0);}
        SendCommand(AWROFF);
    }

// -----
// Ρουτίνα καθαρισμού γραφικών της κύριας περιοχής της οθόνης LCD.
// -----
void ClearGrafMainScreen(void)
{
    unsigned int cells;
    SetPosition(GRAFHOME_ADDR+3*8*GRAF_COLUMNS);
    SendCommand(AWRON);
    cells=GRAF_COLUMNS * GRAF_ROWS-2*3*8*GRAF_COLUMNS;
    while (cells--)
        {AutoDataWrite(0);}
    SendCommand(AWROFF);
}

// -----
// Ρουτίνα εκτύπωσης συμβολοσειράς στην οθόνη LCD από το σημείο
// που ήδη δείχνει ο δείκτης στην περιοχή χαρακτήρων.
// -----
void PrintString(char * String)
{
    SendCommand(AWRON);
    while (*String)
        {AutoDataWrite(*String++);}
    SendCommand(AWROFF);
}

// -----
// Ρουτίνα αποστολής εντολής στον ελεγκτή T6963C
// -----
void SendCommand(unsigned char datbyte)
{
    while ((CommandRegister & 0xf3) != 0x23)
        { /* spin */ };
    CommandRegister = datbyte;
}

// -----
// Ρουτίνα αποστολής δυο Byte στον ελεγκτή T6963C ως δεδομένα.
// -----
void DataTransfer(unsigned char x, unsigned char y)
{
    while ((CommandRegister & 0xf3) != 0x23)
        { /* spin */ };
    DataRegister = x;
    while ((CommandRegister & 0xf3) != 0x23)
        { /* spin */ };
    DataRegister = y;
}

// -----
// Ρουτίνα εγγραφής δεδομένων στην κατάσταση εγγραφής
// και αυτόματης μετακίνησης του δείκτη διευθύνσεων στον

```

```

// χάρτη μνήμης του ελεγκτή T6963C.
// -----
void AutoDataWrite (char datbyte)
{
    while ((CommandRegister & 0x28) != 0x28)
        { /* wait for ready */ };
    DataRegister = datbyte;
}

// -----
// Ρουτίνα εγγραφής ενός Byte δεδομένων, στην θέση μνήμης
// του ελεγκτή T6963C, που δείχνει ο δείκτης διεύθυνσης.
// -----
void BinaryDataWrite(unsigned char datbyte)
{
    while ((CommandRegister & 0xf3) != 0x23 );
    DataRegister = datbyte;
}

// -----
// Ρουτίνα ανάγνωσης δεδομένων από την θέση μνήμης που
// δείχνει ο δείκτης διευθύνσεων του ελεγκτή T6963C.
// -----
unsigned char BinaryDataRead (void)
{
    DataTransfer(0,0);
    SendCommand(DRNADP);
    while ((CommandRegister & 0xf3) != 0x23 );
    return DataRegister;
}

// -----
// Ρουτίνα εγγραφής της νέας διεύθυνσης δεδομένων στον δείκτη
// διευθύνσεων του ελεγκτή T6963C.
// -----
void SetPosition(int Position)
{
    unsigned char addrhigh, addrlow;

    if (Position >= 0xff)
    {
        addrhigh=(unsigned char)((Position >> 8) & 0xff);
        addrlow=(unsigned char)(Position & 0xff);
    }
    else
    {
        addrhigh=0;
        addrlow=(unsigned char)Position;
    }

    DataTransfer(addrlow, addrhigh);
    SendCommand(ADPSET);
}

// -----
// Ρουτίνα τοποθέτησης του δρομέα στην θέση που ορίζουν οι
// συντεταγμένες x, y.
// -----
void SetCursorPosition(unsigned char y, unsigned char x)
{

```

```

unsigned char addrhigh, addrlow;
unsigned int Position;

y-=1;
x-=1;
if (x>TEXTCOLUMNS)
{
    y+=(x/TEXTCOLUMNS);
    x=x%TEXTCOLUMNS;
}

Position = TEXTSTART + (TEXTCOLUMNS * y) + x;

if (Position >= 0xff)
{
    addrhigh=(unsigned char)((Position >> 8) & 0xff);
    addrlow=(unsigned char)(Position & 0xff);
}
else
{
    addrhigh=0;
    addrlow=(unsigned char)Position;
}
DataTransfer(addrlow, addrhigh);
SendCommand(CURSET);
}

// -----
// Ρουτίνα καθορισμού της θέσης του δείκτη διευθύνσεων
// του ελεγκτή T6963C ώστε να δείχνει στην αντίστοιχη θέση
// μνήμης για τις συντεταγμένες που δέχεται ως ορίσματα, στην
// περιοχή χαρακτήρων.
// -----
void TextGoto(unsigned char y, unsigned char x)
{
    unsigned int Position;
    unsigned char temp;

    temp=ShowClock;
    ShowClock=false;
    y-=1;
    x-=1;
    if (x>TEXTCOLUMNS)
    {
        y+=(x/TEXTCOLUMNS);
        x=x%TEXTCOLUMNS;
    }

    Position = TEXTSTART + (TEXTCOLUMNS * y) + x;
    SetPosition(Position);
    ShowClock=temp;
}

// -----
// Ρουτίνα καθαρισμού των περιεχομένων της οριζόμενης γραμμής
// χαρακτήρων.
// -----
void ClearTextLine(unsigned char line)
{
    unsigned char cells;
    unsigned int Position;

```

```

    line-=1;
    Position=TEXTSTART + TEXTCOLUMNS * line;
    SetPosition(Position);
    SendCommand(AWRON);
    cells=TEXTCOLUMNS;
    while (cells--)
    {
        BinaryDataWrite(0);
    }
    SendCommand(AWROFF);
}

// -----
// Ρουτίνα εκτύπωσης ενός χαρακτήρα πολλαπλές φορές στην οθόνη.
// -----
void PrintMultiChar(unsigned char charcode, unsigned char CharTimes)
{
    DisableRTCInt();
    SendCommand(AWRON);
    while (CharTimes--)
    {
        AutoDataWrite(charcode);
    }
    SendCommand(AWROFF);
    EnableRTCInt();
}

// -----
// Ρουτίνα εκτύπωσης ενός χαρακτήρα στην οθόνη LCD.
// -----
void PrintChar(unsigned char charcode)
{
    BinaryDataWrite(charcode);
    SendCommand(DWIADP);
}

// -----
// Ρουτίνα εκτύπωσης της Hex τιμής ενός Byte στην οθόνη
// LCD ως κείμενο.
// -----
void PrintHex(unsigned char charcode)
{
    PrintChar(HEX2ah(charcode));
    PrintChar(HEX2al(charcode));
}

// -----
// Ρουτίνα εκτύπωσης της BCD τιμής ενός Byte στην οθόνη
// LCD ως κείμενο.
// -----
void PrintBCD(unsigned char charcode)
{
    PrintChar(HEX2al(charcode));
}

// -----
// Ρουτίνα δημιουργίας παραθύρων στην οθόνη με ειδικούς χαρακτήρες
// σε σημεία και με διαστάσεις που ορίζουν οι συντεταγμένες

```

```

// που δίδονται ως ορίσματα της.
// -----
void CreateWindow(unsigned char y1, unsigned char x1, unsigned char y2, unsigned
char x2)
{
    unsigned char i;

    DisableRTCInt();
    TextGoto(y1,x1);
    PrintChar('ϣ');
    TextGoto(y2,x2);
    PrintChar('⏏');
    TextGoto(y2,x1);
    PrintChar('⏏');
    TextGoto(y1,x2);
    PrintChar('⏏');
    //Σχεδίαση Οριζόντιων Διαχωριστικών Γραμμών
    TextGoto(y1,x1+1);
    PrintMultiChar('=',x2-x1-1);
    TextGoto(y2,x1+1);
    PrintMultiChar('=',x2-x1-1);
    //Σχεδίαση Κάθετων Διαχωριστικών Γραμμών
    for (i=1;i<y2-y1;i++)
    {
        TextGoto(y1+i,x1);
        PrintChar('⏏');
        TextGoto(y1+i,x2);
        PrintChar('⏏');
    }
    EnableRTCInt();
}

// -----
// Ρουτίνα εκτύπωσης διαχωριστικής γραμμής στα σημεία που
// ορίζουν οι διαστάσεις και συντεταγμένες που δέχεται ως
// ορίσματα.
// -----
void CreateDivider (unsigned char y1, unsigned char x1, unsigned char cols)
{
    DisableRTCInt();
    TextGoto(y1,x1);
    PrintChar('⏏');
    TextGoto(y1,x1+cols);
    PrintChar('⏏');
    TextGoto(y1,x1+1);
    PrintMultiChar('=',cols-1);
    EnableRTCInt();
}

// -----
// Κύλιση κειμένου προς τα πάνω μια γραμμή.
// -----
void ScrollTextUp(unsigned char linea, unsigned char lineb)
{
    unsigned char i;
    unsigned char j;
    unsigned char temp;

    for (i=linea;i<lineb;i++)
    {
        j=2;

```

```

        TextGoto(i+1,j);
        temp=BinaryDataRead();
        TextGoto(i,j);
        PrintChar(temp);
        for (j=2;j<30;j++)
        {
            TextGoto(i+1,j);
            temp=BinaryDataRead();
            TextGoto(i,j);
            PrintChar(temp);
        }
    }
}

// -----
// Κύλιση Κειμένου προς τα κάτω μια γραμμή.
// -----
void ScrollTextDown(unsigned char linea, unsigned char lineb)
{
    unsigned char i;
    unsigned char j;
    unsigned char temp;

    for (i=lineb;i>linea;i--)
    {
        j=2;
        TextGoto(i-1,j);
        temp=BinaryDataRead();
        TextGoto(i,j);
        PrintChar(temp);

        for (j=2;j<30;j++)
        {
            TextGoto(i-1,j);
            temp=BinaryDataRead();
            TextGoto(i,j);
            PrintChar(temp);
        }
    }
}

// -----
// Δημιουργία ανεστραμμένων περιοχών κάνοντας χρήση της κατάστασης
// XOR του ελεγκτή T6963C, σε σημεία που ορίζουν οι συντεταγμένες και
// διαστάσεις που δέχεται ως ορίσματα.
// -----
void InvertArea (unsigned char pattern, unsigned char x, unsigned char y,
unsigned char xpixbytes, unsigned char rasters) reentrant critical
{
    unsigned char rows;
    unsigned int Position;

    Position=GRAFHOME_ADDR+(unsigned int)(y*GRAFCOLUMNS)+(unsigned int)x;
    SetPosition(Position);

    for (rows=0;rows<rasters;rows++)
    {
        PrintMultiChar(pattern,xpixbytes);
        if (pattern==0xff)
            PrintChar(0x80);
        else PrintChar(0x0);
        Position=Position+GRAFCOLS;
    }
}

```

```

        SetPosition(Position);
    }
}

// -----
// Ρουτίνα εμφάνισης του γραφικού λογότυπου του Ιδρύματος
// στην οθόνη LCD.
// -----
void LoadTitlePic (void) reentrant
{
    unsigned char rasterbytes;
    unsigned int rasters;
    unsigned int count;

    count=0;
    for (rasters=0;rasters<18;rasters++)
    {
        SetPosition(0x2d9+rasters*GRAFCOLUMNS);
        SendCommand(AWRON);
        for (rasterbytes=0;rasterbytes<20;rasterbytes++)
        {
            AutoDataWrite(title[count]);
            count++;
        }
        SendCommand(AWROFF);
    }
}

void LoadLockPic (void) reentrant
{
    unsigned char rasterbytes;
    unsigned int rasters;
    unsigned int count;

    count=0;
    for (rasters=0;rasters<64;rasters++)
    {
        SetPosition(0x2d1+rasters*GRAFCOLUMNS);
        SendCommand(AWRON);
        for (rasterbytes=0;rasterbytes<8;rasterbytes++)
        {
            AutoDataWrite(lock[count]);
            count++;
        }
        SendCommand(AWROFF);
    }
}

```

## T6963C.H

```
// -----  
// Δηλώσεις εντολών του ελεγκτή T6963C.  
// -----  
#define TXHOME    0x40    // set text area home command  
#define TXAREA    0x41    // set text area command  
#define GRHOME    0x42    // set graphics area home command  
#define GRAREA    0x43    // set graphics area command  
#define CURSET    0x21    // set cursor command  
#define OFFSET    0x22    // set offset command  
#define ADPSET    0x24    // set address pointer command  
#define AWRON     0xB0     // auto-write on  
#define AWROFF    0xB2     // auto-write off  
#define BITSET    0xF8     // bit set/reset command  
#define BITUST    0xF0     // bit set/reset command  
#define DWIADP    0xc0     // Data Write and Increment ADP  
#define DRIADP    0xc3     // Data Read and Increment ADP  
#define DWNADP    0xc4     // Data Write and Nonvariable ADP  
#define DRNADP    0xc5     // Data Read and Nonvariable ADP  
#define CURBLK    0xa0     // Cursor Pattern Block  
  
// -----  
// Παράμετροι καθορισμού του τρόπου εμφάνισης των περιοχών  
// γραφικών και κειμένου όταν έχουν ενεργοποιηθεί και οι δύο.  
// Παράμετροι καθορισμού του τρόπου εμφάνισης της περιοχής  
// κειμένου και του χρησιμοποιούμενου σετ χαρακτήρων.  
// -----  
#define MODESET      0x80  
#define CGRAM        0x08  
#define TXTONLY     0x04  
#define OR_GRAF     0x00  
#define AN_GRAF     0x03  
#define XR_GRAF     0x01  
  
// -----  
// Παράμετροι λειτουργίας οθόνης - εμφάνισης περιοχών.  
// -----  
#define DISPLAY      0x90  
#define NO_CURS_BLNK 0x01  
#define CURS_DISPLAY 0x02  
  
#define TEXT_ON      0x04  
#define GRAF_ON      0x08  
  
// -----  
// Δηλώσεις καθορισμού περιοχών στην μνήμη του ελεγκτή T6963C  
// και διαχωρισμός τους σε περιοχές Γραφικών, Κειμένου, CGRAM.  
// Δηλώσεις παραμέτρων των περιοχών αυτών.  
// -----  
#define TEXTBEGIN    0x10  
#define TEXTSTART    0x1000  
#define CGRAMOFFSET  0x3  
#define CGRAMBEGIN   0x18  
#define CGRAMAREA    0x1800 // αποκωδικοποιημένη διεύθυνση από το CGRAMOFFSET  
#define BYTESPERLINE 0x1e // το δεκαδικό 30  
#define TEXTCOLUMNS 0x1e // το δεκαδικό 30  
#define TEXTROWS     16  
#define GRAFCOLUMNS 0x1e // 30 - κελιά χαρακτήρων πλάτους 8bit  
#define GRAFCOLS     0x001e // 30 - κελιά χαρακτήρων πλάτους 8bit  
#define GRAFROWS     128 // 128 dots  
#define GRAFHOME_LO  0x00 // Καθορισμός του low byte της διεύθυνσης  
#define GRAFHOME_HI  0x00 // καθορισμός του high byte της διεύθυνσης
```



```

#define GRAFHOME_ADDR    0x0000 // καθορισμός πλήρους διεύθυνσης

// -----
// Δηλώσεις ονομάτων των διευθύνσεων αρχής κάθε γραμμής
// κειμένου στην οθόνη LCD.
// -----
#define LINE1    TEXTSTART
#define LINE2    TEXTSTART + TEXTCOLUMNS * 1
#define LINE3    TEXTSTART + TEXTCOLUMNS * 2
#define LINE4    TEXTSTART + TEXTCOLUMNS * 3
#define LINE5    TEXTSTART + TEXTCOLUMNS * 4
#define LINE6    TEXTSTART + TEXTCOLUMNS * 5

#define LINE7    TEXTSTART + TEXTCOLUMNS * 6
#define LINE8    TEXTSTART + TEXTCOLUMNS * 7
#define LINE9    TEXTSTART + TEXTCOLUMNS * 8
#define LINE10   TEXTSTART + TEXTCOLUMNS * 9
#define LINE11   TEXTSTART + TEXTCOLUMNS * 10
#define LINE12   TEXTSTART + TEXTCOLUMNS * 11
#define LINE13   TEXTSTART + TEXTCOLUMNS * 12
#define LINE14   TEXTSTART + TEXTCOLUMNS * 13
#define LINE15   TEXTSTART + TEXTCOLUMNS * 14
#define LINE16   TEXTSTART + TEXTCOLUMNS * 15

// -----
// Δηλώσει απόδοσης ονομάτων στις διευθύνσεις που βρίσκονται
// οι καταχωρητές του ελεγκτή T6963C.
// -----
xdata at 0xfc00 unsigned char DataRegister;
xdata at 0xfd00 unsigned char CommandRegister;

// -----
// Δηλώσεις ρουτινών-συναρτήσεων.
// -----
void InitLCD (void);
void LoadCGRAM (void);
void ResetLCD (void);
void ClearTextScreen (void);
void ClearGraphicsScreen (void);
void ClearGrafxMainScreen (void);
void PrintHex (unsigned char);
void PrintBCD (unsigned char);
void PrintChar (unsigned char);
void PrintString (char *);
void PrintMultiChar (unsigned char, unsigned char);
void DataTransfer(unsigned char, unsigned char);
void SendCommand (unsigned char);
void AutoDataWrite (char);
void BinaryDataWrite(unsigned char);
unsigned char BinaryDataRead (void);
void SetPosition (int);
void SetCursorPosition (unsigned char, unsigned char);
void TextGoto (unsigned char, unsigned char);
void ClearTextLine (unsigned char);
void LoadFullScreen (void);
void CreateWindow (unsigned char, unsigned char, unsigned char, unsigned char);
void CreateDivider (unsigned char, unsigned char, unsigned char);
void ScrollTextUp (unsigned char, unsigned char);
void ScrollTextDown (unsigned char, unsigned char);
void InvertArea (unsigned char, unsigned char, unsigned char, unsigned char,
unsigned char) reentrant critical;
void LoadLockPic (void) reentrant;
void LoadTitlePic (void) reentrant;

```

## SERPORT.C

```
#include <ds5000.h>
#include <headers.h>
#include <t6963c.h>
#include <serport.h>
#include <func.h>
#include <rtc.h>

// -----
// Ρουτίνα αρχικοποίησης της σειριακής θύρας του μικροελεγκτή.
// -----
void SerInit (void)
{
    TH1=0xfd;    // Baudrate=9600
    PCON=0x80;  // Baudrate=19200
    TL1=TH1;
    SCON=0x52;
    TMOD=0x21;
    TCON=0x40;
    RI=0;
    TI=0;
}

// -----
// Ρουτίνα σειριακής αποστολής χαρακτήρα.
// -----
void SendChar (unsigned char s)
{
    DisableSerialInt();
    SBUF=s;
    while (TI==false);
    TI=false;
    EnableSerialInt();
}

// -----
// Ρουτίνα σειριακής αποστολής ASCII χαρακτήρων Hex τιμής.
// -----
void SendHex (unsigned char s)
{
    SendChar(HEX2ah(s));
    SendChar(HEX2al(s));
}

// -----
// Ρουτίνα σειριακής αποστολής συμβολοσειράς.
// -----
void SendString (unsigned char *String)
{
    DisableSerialInt();
    while (*String)
    {
        SendChar(*String);
        String++;
    }
    EnableSerialInt();
}
```

```

// -----
// Ρουτίνα σειριακής αποστολής εντολής προς τον H/Y.
// -----
void SendTerminalCommand (unsigned char *String)
{
    DisableSerialInt();
    while (*String)
    {
        SendChar(*String);
        String++;
    }
    SendChar(ComVT);
    EnableSerialInt();
}

// -----
// Ρουτίνα σειριακής λήψης χαρακτήρα.
// -----
unsigned char ReceiveChar (void)
{
    DisableSerialInt();
    while (RI==false);
    RI=false;
    EnableSerialInt();
    return SBUF;
}

// -----
// Ρουτίνα σειριακής αποστολής χαρακτήρα χωρίς έλεγχο Interrupts.
// -----
void SIntSendChar (unsigned char s)
{
    SBUF=s;
    while (TI==false);
    TI=false;
}

// -----
// Ρουτίνα σειριακής αποστολής συμβολοσειράς χωρίς έλεγχο
// Interrupts.
// -----
void SIntSendString (unsigned char *String)
{
    while (*String)
    {
        SendChar(*String);
        String++;
    }
}

// -----
// Ρουτίνα σειριακής αποστολής εντολής προς τον H/Y χωρίς έλεγχο
// Interrupts.
// -----
void SIntSendTerminalCommand (unsigned char *String)
{
    while (*String)
    {
        SendChar(*String);
        String++;
    }
}

```

```

    }
    SendChar(ComVT);
}

// -----
// Ρουτίνα σειριακής λήψης χαρακτήρα χωρίς έλεγχο Interrupts.
// -----
unsigned char SIntReceiveChar (void)
{
    while (RI==false);
    RI=false;
    return SBUF;
}

// -----
// Ρουτίνα σειριακής λήψης χαρακτήρα χωρίς έλεγχο Interrupts.
// Επίσης διαθέτει χρονική καθυστέρηση προτού κάνει Timeout
// και έξοδο από την ρουτίνα, για να αποφευχθεί η παγίδευση
// σε συνεχή βρόχο.
// -----
unsigned char SIntReceiveChar2 (void)
{
    unsigned char i;

    TH0=0x0;
    TL0=0x0;
    TR0=true;
    i=0;
    while (i<4)
    {
        TF0=false;
        if (RI==true) break;
        while ((RI==false)&&(TF0==false));
        i++;
        if (RI==true) break;
    }
    RI=false;
    return SBUF;
}

// -----
// Ενεργοποίηση αποδοχής αιτήσεων εξυπηρέτησης διακοπών
// από την σειριακή θύρα του μικροελεγκτή.
// -----
void EnableSerialInt (void)
{
    ES=true;
}

// -----
// Απενεργοποίηση αποδοχής αιτήσεων εξυπηρέτησης διακοπών
// από την σειριακή θύρα του μικροελεγκτή.
// -----
void DisableSerialInt (void)
{
    ES=false;
}

// -----

```

```

void EnableOnlySerialInt (void)
{
    IE_Backup=IE;
    IE=0x90;
}

// -----
void RestoreInterrupts (void)
{
    IE=IE_Backup;
}

// -----
// Επίτρεψη εισερχόμενων χαρακτήρων από την σειριακή θύρα.
// -----
void EnableReceive(void)
{
    RI=false;
    REN=true;
}

// -----
// Απαγόρευση εισερχόμενων χαρακτήρων από την σειριακή θύρα.
// -----
void DisableReceive(void)
{
    RI=false;
    REN=false;
}

// -----
// Μετατροπή ASCII χαρακτήρα στην Hex τιμή του.
// -----
unsigned char ASC2HEX ()
{
    unsigned char temp;

    temp=SIntReceiveChar()-0x30;
    if (temp>0x9) temp-=0x7;
    return temp;
}

// -----
// Μετατροπή διπλού ASCII χαρακτήρα στην διπλή Hex τιμή του.
// -----
unsigned char dASC2HEX ()
{
    unsigned char temp;
    temp=ASC2HEX();
    if (temp!=0x0) {temp=(temp<<4);}
    temp+=ASC2HEX();
    return temp;
}

// -----
// Αναζήτηση της συμβολοσειράς "AT" για αναγνώριση εισερχόμενης
// εντολής προς το τερματικό.
// -----

```

```

unsigned char ComPortSearchAT()
{
    unsigned char temp;
    temp=false;
    if ((SIntReceiveChar())=='A')
    {
        if ((SIntReceiveChar2())=='T') temp=true;
    }
    return temp;
}

// -----
// Ρουτίνα εξυπηρέτησης διακοπών από την σειριακή θύρα.
// Αναγνώριση των εισερχομένων εντολών και εκτέλεση των
// αντίστοιχων λειτουργιών.
// -----
void SerINTHandle (void) interrupt 4 critical
{
    DisableRTCInt();
    DisableSerialInt();
    if (ComPortSearchAT())
    {
        switch (SIntReceiveChar())
        {
            case 'A'://Απάντηση του διαχειριστή τερματικού ότι είναι Online.
                ServerOnLine=true;
                SIntSendTerminalCommand("ATOK");
                break;
            case 'B'://Αποθήκευση στοιχείων κατόχου κλειδιού ασφαλείας.
                StoreKeyHolderID();
                break;
            case 'C'://Κλήση αναγνώρισης παρουσίας τερματικού στην θύρα COM.
                SIntSendTerminalCommand("ATACK");
                break;
            case 'D'://Εντολή εκτέλεσης διαγνωστικών ελέγχων τερματικού.
                SIntSendTerminalCommand("ATOK");
                DiagCheck();
                break;
            case 'E'://Δήλωση πρόκλησης σφάλματος από τον διαχειριστή
τερματικού.
                WaitForRequestReply=false;
                ErrorReturned=true;
                break;
            case 'L'://Εντολή για κλείδωμα της πόρτας.
                LockDoor();
                break;
            case 'M'://Εντολή αποθήκευσης αριθμού του κυρίου κλειδιού.
                StoreMasterKey();
                SIntSendTerminalCommand("ATOK");
                break;
            case 'Q'://Εντολή διαχείρισης παραμέτρων εμφάνισης κυλιόμενου
μηνύματος.
                MarqueeSetup();
                break;
            case 'R'://Εντολή αλλαγής ημερομηνίας και ώρας του RTC.
                Set_RTC_Time();
                break;
            case 'S'://Εντολή εναλλαγής κατάστασης ειδικού μηνύματος.
                Toggle_CIP_Status();
                break;
            case 'T'://Εντολή αποθήκευσης τίτλων ανακοινώσεων.
                StoreAnnouncementTitles();
                break;
        }
    }
}

```

```

        case 'U'://Εντολή για ξεκλείδωμα της πόρτας.
            UnlockDoor();
            break;
        case 'X'://Εντολή αποθήκευσης κειμένου ανακοίνωσης.
            StoreAnnouncement();
            DeployAnnouncement=true;
            break;
        case 'Y'://Εντολή αποθήκευσης εισερχομένων τίτλων μαθημάτων.
            StoreCourseTitles();
            break;
        case 'Z'://Εντολή αποθήκευσης βαθμολογικού πίνακα μαθήματος.
            StoreCourseMarks();
            DeployCourseMarks=true;
            break;
    }
    SoundBuzzer();
}
EnableSerialInt();
EnableRTCInt();
}

```

## SERPORT.H

```

// -----
// Δηλώσεις ρουτινών-συναρτήσεων.
// -----

void SerInit (void);
void SendChar (unsigned char);
void SendHex (unsigned char);
void SendString (unsigned char *);
void SendTerminalCommand (unsigned char *);

void SIntSendChar (unsigned char);
void SIntSendString (unsigned char *);
void SIntSendTerminalCommand (unsigned char *);
unsigned char SIntReceiveChar (void);
unsigned char SIntReceiveChar2 (void);
unsigned char ReceiveChar (void);
unsigned char ComPortSearchAT ();
void EnableSerialInt (void);
void DisableSerialInt (void);
void EnableOnlySerialInt (void);
void RestoreInterrupts (void);
unsigned char ASC2HEX ();
unsigned char dASC2HEX ();
void SerINTHandle (void) interrupt 4 critical;

```

## RTC.C

```
#include <ds5000.h>
#include <serport.h>
#include <headers.h>
#include <t6963c.h>
#include <func.h>
#include <rtc.h>

// -----
// Δήλωση των συμβολοσειρών των ημερών της εβδομάδας.
// -----
code unsigned char WeekDay[7][11]={ " Κυριακή ",
                                     " Δευτέρα ",
                                     " Τρίτη ",
                                     " Τετάρτη ",
                                     " Πέμπτη ",
                                     " Παρασκευή ",
                                     " Σάββατο "};

// -----
// Ρουτίνα αρχικοποίησης των λειτουργιών του RTC.
// -----
void InitRTC (void)
{
    RTC_RegisterA=0x20;
    RTC_RegisterB=0x12;
}

// -----
// Ρουτίνα διαγνωστικού ελέγχου της μπαταρίας του RTC.
// Επιστρέφεται η τιμή true αν είναι η μπαταρία σε καλή
// κατάσταση, ενώ η τιμή false αν η μπαταρία είναι εξασθενημένη.
// -----
unsigned char Check_RTC_Battery (void)
{
    if (RTC_RegisterD==RTC_BatteryOK) return true;
    else return false;
}

// -----
// Ρουτίνα εξυπηρέτησης διακοπών απο τον ακροδέκτη /INT0
// στον οποίο είναι συνδεδεμένος ο αντίστοιχος ακροδέκτης
// του RTC. Κάθε δευτερόλεπτο εκτυπώνεται στην οθόνη
// η ημερομηνία και η ώρα. Επίσης ενημερώνεται η μεταβλητή
// καθυστέρησης που χρησιμοποιείται απο την ρουτίνα πρόκλησης
// χρονικής καθυστέρησης.
// -----
void RTC_Int (void) interrupt 0 critical
{
    unsigned char temp;

    if (ShowClock)
    {
        TextGoto(15,3);
        PrintString(WeekDay[RTC_DayOfWeek-1]);
        temp=RTC_Hours;
        PrintChar(HEX2ah(temp));
        PrintChar(HEX2al(temp));
    }
}
```



```

        PrintChar(':');
        temp=RTC_Minutes;
        PrintChar(HEX2ah(temp));
        PrintChar(HEX2al(temp));
        PrintChar(':');
        temp=RTC_Seconds;
        PrintChar(HEX2ah(temp));
        PrintChar(HEX2al(temp));
        PrintChar(' ');
        temp=RTC_DateOfMonth;
        PrintChar(HEX2ah(temp));
        PrintChar(HEX2al(temp));
        PrintChar('/');
        temp=RTC_Month;
        PrintChar(HEX2ah(temp));
        PrintChar(HEX2al(temp));
        PrintChar('/');
        temp=RTC_Year;
        PrintChar(HEX2ah(temp));
        PrintChar(HEX2al(temp));

    }

    if (RTC_Seconds==00)
    {
        if (((RTC_Minutes==0x45)&&(ChangeTimeAt==0x45)) ||
            ((RTC_Minutes==0x15)&&(ChangeTimeAt==0x15)))
        {
            if (RTC_Minutes==0x45) ChangeTimeAt=0x15;
            if (RTC_Minutes==0x15) ChangeTimeAt=0x45;
            SendTerminalCommand("ATTA");
        }
    }

    if ((RTC_Minutes==0)&&(RTC_Seconds<6))
    {
        for (temp=0;temp<32;temp++) SoundBuzzer();
    }

    if (Secs2Delay==0) TimedOut=true;
    if (Secs2Delay!=0) Secs2Delay--;
    NullByte=RTC_RegisterC;
}

// -----
// Ρουτίνα ενεργοποίησης αλλαγής των περιεχομένων
// των καταχωρητών του RTC.
// -----
void RTC_ChangeEnable(void)
{
    RTC_RegisterB=RTC_EnableTimeChange;
}

// -----
// Ρουτίνα απενεργοποίησης αλλαγής των περιεχομένων
// των καταχωρητών του RTC.
// -----
void RTC_ChangeDisable(void)
{
    RTC_RegisterB=RTC_NormalMode;
}

```

```

// -----
// Ρουτίνα ενεργοποίησης αποδοχής αίτησης διακοπής απο το RTC.
// -----
unsigned char EnableRTCInt (void)
{
    EX0=true;
}

// -----
// Ρουτίνα απενεργοποίησης αποδοχής αίτησης διακοπής απο το RTC.
// -----
unsigned char DisableRTCInt (void)
{
    EX0=false;
}

// Αλλαγή Ώρας και Ημερομηνίας του RTC
void Set_RTC_Time (void)
{
    RTC_ChangeEnable();
    RTC_Hours=dASC2HEX();
    RTC_Minutes=dASC2HEX();
    RTC_Seconds=dASC2HEX();
    RTC_Century=dASC2HEX();
    RTC_Year=dASC2HEX();
    RTC_Month=dASC2HEX();
    RTC_DateOfMonth=dASC2HEX();
    RTC_DayOfWeek=dASC2HEX();
    RTC_ChangeDisable();
    SIntSendTerminalCommand("ATOK");
}

```

## RTC.H

```
// -----  
// Δηλώσεις απόδοσης ονομάτων σε σταθερές τιμές του RTC.  
// -----  
#define RTC_NormalMode 0x12  
#define RTC_EnableTimeChange 0x82  
#define RTC_BatteryOK 0x80  
  
// -----  
// Δηλώσεις απόδοσης ονομάτων στις διευθύνσεις των καταχωρητών  
// του RTC.  
// -----  
xdata at 0xFB00 unsigned char RTC_Seconds;  
xdata at 0xFB01 unsigned char RTC_SecondsAlarm;  
xdata at 0xFB02 unsigned char RTC_Minutes;  
xdata at 0xFB03 unsigned char RTC_MinutesAlarm;  
xdata at 0xFB04 unsigned char RTC_Hours;  
xdata at 0xFB05 unsigned char RTC_HoursAlarm;  
xdata at 0xFB06 unsigned char RTC_DayOfWeek;  
xdata at 0xFB07 unsigned char RTC_DateOfMonth;  
xdata at 0xFB08 unsigned char RTC_Month;  
xdata at 0xFB09 unsigned char RTC_Year;  
xdata at 0xFB0A unsigned char RTC_RegisterA;  
xdata at 0xFB0B unsigned char RTC_RegisterB;  
xdata at 0xFB0C unsigned char RTC_RegisterC;  
xdata at 0xFB0C unsigned char RTC_Status;  
xdata at 0xFB0D unsigned char RTC_RegisterD;  
xdata at 0xFB32 unsigned char RTC_Century;  
  
void InitRTC (void);  
unsigned char Check_RTC_Battery (void);  
void RTC_Int (void) interrupt 0;  
void RTC_ChangeEnable (void);  
void RTC_ChangeDisable (void);  
unsigned char EnableRTCInt (void);  
unsigned char DisableRTCInt (void);  
void Set_RTC_Time (void);
```

## HEADERS.H

```
// -----  
// Δηλώσεις σταθερών τιμών με χαρακτηριστική ονομασία.  
// -----  
  
#define true 1  
#define false 0  
  
#define ComEOS 0x0 // Χαρακτήρας δήλωσης τέλους συμβολοσειράς.  
#define ComEOT 0x4 // Χαρακτήρας δήλωσης τέλους μετάδοσης.  
#define ComVT 0xb // Χαρακτήρας δήλωσης τέλους αποστελλόμενης εντολής.  
  
#define BuzzerDelay 0xf800 // Ηχητική προειδοποίηση διάρκειας 2ms  
#define iDetDelay 3 // κύκλοι αναμονής μετά από αναγνώριση i-Button  
  
#define ShowMarquee 0x01  
#define ShowMarqueeInRAM 0x10  
  
#define ATStorageStart 0x7700  
#define AnnouncementStorage 0x7800  
#define CourseTitlesStorage 0x6000  
#define CourseTitlesStrings 0x6001  
#define CourseMarksStorage 0x6100  
#define KeyHolderIDStorage 0xfb60  
  
// -----  
// Δηλώσεις ονομάτων μεταβλητών της λειτουργίας Βαθμολογιών  
// Μαθημάτων.  
// -----  
xdata at 0x6000 unsigned char NoOfCourses;  
xdata at 0x6001 unsigned char CourseTitles[8][29];  
  
// -----  
// Δηλώσεις ονομάτων μεταβλητών της λειτουργίας Ελέγχου  
// Πρόσβασης.  
// -----  
xdata at 0x60ea unsigned char PassCode[6];  
xdata at 0x60f0 unsigned char MasterKey[8];  
xdata at 0x60f8 unsigned char iButtonData[8];  
  
// -----  
// Δηλώσεις απόδοσης ονομάτων σε θέσεις-διευθύνσεις μνήμης  
// που χαρτογραφούνται στην εσωτερική μνήμη του RTC.  
// -----  
xdata at 0xfb0e unsigned char MarqueeMode;  
xdata at 0xfb0f unsigned char MarqueeDelay;  
xdata at 0xfb10 unsigned char cCRC;  
xdata at 0xfb11 unsigned char Wait;  
xdata at 0xfb12 unsigned char Times;  
xdata at 0xfb13 unsigned char iButtonFound;  
xdata at 0xfb14 unsigned char Secs2Delay;  
xdata at 0xfb15 unsigned char ShowClock;  
xdata at 0xfb16 unsigned char ChangeTimeAt;  
xdata at 0xfb17 unsigned char ServerOnLine;  
xdata at 0xfb18 unsigned char TimedOut;  
xdata at 0xfb19 unsigned char DoorIsLocked;  
xdata at 0xfb1a unsigned char WaitForRequestReply;  
xdata at 0xfb1b unsigned char IE_Backup;  
xdata at 0xfb1c unsigned char NullByte;
```

```
xdata at 0xfb1d unsigned char SelButton;
xdata at 0xfb1e unsigned char SelMenu;
xdata at 0xfb1f unsigned char DeployAnnouncement;
xdata at 0xfb20 unsigned char ErrorReturned;
xdata at 0xfb21 unsigned char ATStored;
xdata at 0xfb22 unsigned char * xdata StorageSpace;
xdata at 0xfb24 unsigned char * xdata ATStorageSpace;
xdata at 0xfb26 unsigned char StorageLines;
xdata at 0xfb27 unsigned char iDetectionDelay;
xdata at 0xfb28 unsigned char ClassInProgress;
xdata at 0xfb29 unsigned char CIPTemp;
xdata at 0xfb2a unsigned char CMStorageLines;
xdata at 0xfb2b unsigned char DeployCourseMarks;
xdata at 0xfb2c unsigned char LineNumber;
xdata at 0xfb2d unsigned char Index;
xdata at 0xfb2e unsigned char KeyOwnerIDStored;
xdata at 0xfb2f unsigned char KeybDelay;
xdata at 0xfb30 unsigned char Count2Reset;

// -----
// Δήλωση του κυκλικού Buffer για το κυλιόμενο μήνυμα
// -----
xdata at 0xfb40 unsigned char MarqueeMsg[29];

// -----
// Δήλωση του Buffer αποθήκευσης του ονοματεπώνυμου του
// κατόχου του κλειδιού ασφαλείας.
// -----
xdata at 0xfb60 unsigned char KeyOwnerID[21];
```

## HEADERS2.H

```
// -----  
// Δηλώσεις σταθερών τιμών αποθηκευμένων σε μορφή πινάκων  
// στην εσωτερική μνήμη προγράμματος του μικροελεγκτή.  
// -----  
  
// -----  
// Συντεταγμένες και διαστάσεις πλαισίων κουμπιών επιλογών  
// του κυρίου καταλόγου επιλογών.  
// -----  
code unsigned char BCoord[4][4]={{2,32,14,23},  
                                   {2,64,14,23},  
                                   {17,32,11,23},  
                                   {17,64,11,23}};  
  
// -----  
// Συντεταγμένες και διαστάσεις πλαισίων μπάρας επιλογών  
// των καταλόγων επιλογών των υπολειτουργιών Ανακοινώσεις  
// και Βαθμολογίες.  
// -----  
code unsigned char MCoord[8][4]={{1,40,28,8},  
                                   {1,48,28,8},  
                                   {1,56,28,8},  
                                   {1,64,28,8},  
                                   {1,72,28,8},  
                                   {1,80,28,8},  
                                   {1,88,28,8},  
                                   {1,96,28,8}};  
  
code unsigned char HelpMessages[3][4][25]={  
    {  
        "C'=Προηγούμενη Επιλογή.",  
        "D'=Επόμενη Επιλογή.   ",  
        "#'=Αποδοχή Επιλογής.   ",  
        "*'=Ακύρωση Λειτουργίας."  
    },  
    {  
        "Για να διαβάσετε κάποια ",  
        "ανακοίνωση, επιλέξτε τον",  
        "τίτλο αυτής και πατήστε ",  
        "το πλήκτρο Αποδοχής '#'. "  
    },  
    {  
        "Για να δείτε τον βαθμό  ",  
        "σας σε κάποιο μάθημα,   ",  
        "επιλέξτε το και πατήστε ",  
        "το πλήκτρο Αποδοχής '#'. "  
    }  
};
```

## FUNC.C

```
#include <ds5000.h>
#include <t6963c.h>
#include <ibutton.h>
#include <headers.h>
#include <serport.h>
#include <keyb.h>
#include <func.h>
#include <rtc.h>
#include <headers2.h>

code unsigned char MarqueeProto[29]={" http://mprolab.teipir.gr",0x0};

// -----
// Έλεγχος αν το εισαχθέν i-Button είναι το κύριο κλειδί.
// -----
unsigned char IsMasterKey (void)
{
    unsigned char i;
    unsigned char j;

    j=0;
    for (i=0;i<8;i++)
    {
        if (MasterKey[i]==iButtonData[i]) j++;
    }

    if (j==8) return true;
    else return false;
}

// -----
// Αποθήκευση του καινούριου κύριου κλειδιού.
// -----
void StoreMasterKey (void)
{
    unsigned char i;
    for (i=8;i>0;i--)
    {
        MasterKey[i-1]=dASC2HEX();
    }
}

// -----
// Έλεγχος και χειρισμός του κυρίου κλειδιού.
// Καθυστέρηση αναγνώσεων για κλείδωμα-ξεκλείδωμα με το κύριο κλειδί.
// -----
void HandleMasterKeyInsertion (void)
{
    if (iDetectionDelay==0)
    {
        if (DoorIsLocked==true) UnlockDoor();
        else LockDoor();
        iDetectionDelay=iDetDelay;
    }
    if (iDetectionDelay>0) iDetectionDelay--;
}
```

```

// -----
// Ρουτίνα αποθήκευσης του ονοματεπώνυμου του κατόχου του
// εισαχθέντος i-Button.
// -----
void StoreKeyHolderID (void)
{
    unsigned char temp;
    StorageSpace=KeyHolderIDStorage;
    temp=0x0;
    while (temp!=ComEOT)
    {
        temp=SIntReceiveChar();
        *StorageSpace=temp;
        StorageSpace++;
    }
    KeyOwnerIDStored=true;
}

// -----
// Ρουτίνα μετατροπής του packed BCD στον αντίστοιχο Binary.
// -----
unsigned char dBCD2Binary (unsigned char temp)
{
    unsigned char low;
    unsigned char high;

    low=temp & 0xf0;
    high=temp & 0x0f;
    temp=high*10+low;

    return temp;
}

// -----
// Η ρουτίνα αυτή επιστρέφει τον Hex αριθμό του upper byte
// ως ASCII χαρακτήρα.
// -----
unsigned char HEX2ah(unsigned char HEXByte)
{
    unsigned char temp;
    temp=(HEXByte & 0xf0)>>4;
    if (temp<10)
    {temp+=0x30;}
    else
    {temp+=0x37;}
    return temp;
}

// -----
// Η ρουτίνα αυτή επιστρέφει τον Hex αριθμό του lower byte
// ως ASCII χαρακτήρα.
// -----
unsigned char HEX2al(unsigned char HEXByte)
{
    unsigned char temp;

    temp=(HEXByte & 0xf);
    if (temp<10)
    {temp+=0x30;}
    else

```



```

        {temp+=0x37;}
        return temp;
    }

// -----
// Αναμονή για x αριθμό δευτερολέπτων.
// Χρήση του RTC interrupt.
// -----
void DelaySecs (unsigned char secs)
{
    Secs2Delay=secs;
    while (Secs2Delay!=0)
        { /*spin*/ }
}

// -----
// Ενεργοποίηση των Interrupts.
// -----
void EnableInts (void)
{
    EA=true;
}

// -----
// Απενεργοποίηση των Interrupts.
// -----
void DisableInts (void)
{
    EA=false;
}

// -----
// Ενεργοποίηση των διακοπών από τον Timer0.
// -----
void EnableTimer0Int (void)
{
    ET0=true;
    TR0=true;
}

// -----
// Απενεργοποίηση των διακοπών από τον Timer0.
// -----
void DisableTimer0Int (void)
{
    ET0=false;
    TR0=false;
}

// -----
// Εκτύπωση του κυρίου καταλόγου επιλογών.
// -----
void MainMenu (void)
{
    DisableRTCInt();
    CreateWindow(5,3,7,16);
    TextGoto(6,4);
    PrintString("Ανακοινώσεις");
}

```

```

        CreateWindow(9,3,11,16);
        TextGoto(10,4);
        PrintString("Βαθμολογίες");
        CreateWindow(5,18,7,28);
        TextGoto(6,20);
        PrintString("Βοήθεια");
        CreateWindow(9,18,11,28);
        TextGoto(10,20);
        PrintString("Σχετικά");
        SelButton=0;
        ButtonSelect(SelButton,SelButton);
        CIPMessage();
        EnableRTCIInt();
    }

// -----
// Εκτύπωση του τίτλου του Interface.
// -----
void DrawInitScreen (void)
{
    ShowClock=false;
    CreateWindow(1,1,16,30);
    CreateDivider(3,1,29);
    CreateDivider(14,1,29);
    TextGoto(2,2);
    PrintString("ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΩΝ");
    InvertArea(0xff,1,6,28,11);
    MainMenu();
    ShowClock=true;
}

// -----
// Γέμισμα της οθόνης επιλογών με έναν χαρακτήρα ASCII.
// -----
void FillMainScreen (unsigned char fillchar)
{
    unsigned char y, x;

    ShowClock=false;
    for (y=4;y<14;y++)
    {
        for (x=2;x<30;x++)
        {
            TextGoto(y,x);
            PrintChar(fillchar);
        }
    }
    ShowClock=true;
}

// -----
// Κλήση της ρουτίνας γεμίματος της οθόνης με τον κενό
// χαρακτήρα.
// -----
void ClearMainScreen (void)
{
    FillMainScreen(' ');
}

// -----

```

```

// Ρουτίνα ξεκλειδώματος της πόρτας.
// -----
void UnlockDoor (void)
{
    if (DoorStatus()==false)
    {
        //Η πόρτα είναι ήδη ανοικτή.
        SoundBuzzer();
        SendTerminalCommand("ATEDO");
    }
    else
    {
        //Η πόρτα ξεκλειδώνεται.
        P3_3=1;
        TextGoto(15,2);
        PrintChar(0x5);
        SendTerminalCommand("ATDU");
        DoorIsLocked=false;
    }
    WaitForRequestReply=false;
}

// -----
// Ρουτίνα κλειδώματος της πόρτας.
// -----
void LockDoor (void)
{
    if (DoorStatus()==true)
    {
        //Η πόρτα κλειδώνεται.
        P3_3=0;
        TextGoto(15,2);
        PrintChar(0x6);
        SendTerminalCommand("ATDL");
        DoorIsLocked=true;
    }
    else
    {
        //Η πόρτα είναι ανοικτή, δεν μπορεί να κλειδωθεί.
        SoundBuzzer();
        SendTerminalCommand("ATEDO");
    }
}

// -----
// Ρουτίνα εξακρίβωσης της κατάστασης επικοινωνίας με τον
// Εξυπηρετητή της εφαρμογής τερματικού.
// -----
void ComProgressBar (void)
{
    unsigned char i;

    ServerOnLine=false;
    SendChar(ComEOT);
    ClearMainScreen();
    CreateWindow(7,10,9,21);
    if (ServerOnLine==false)
    {
        for (i=0;i<10;i++)
        {
            if (ServerOnLine==true) break;
            DelaySecs(1);
        }
    }
}

```

```

        SendChar(ComEOT);
        TextGoto(8,11+i);
        PrintChar('█');
    }
}

TextGoto(8,11);
if (ServerOnLine==false)
{
    BuzzerOn();
    PrintString("Link Error");
}
else
{
    PrintString("PC Link OK");
}
DelaySecs(1);
BuzzerOff();
}

// -----
// Ρουτίνα αντιστροφής πλαισίου κουμπιών επιλογών.
// -----
void ButtonSelect (unsigned char prbno, unsigned char bno)
{
    InvertArea(0x00,BCoord[prbno][0],BCoord[prbno][1],BCoord[prbno][2],
BCoord[prbno][3]);
    if (bno!=0xff)
        InvertArea(0xff,BCoord[bno][0],BCoord[bno][1],BCoord[bno][2],
BCoord[bno][3]);
}

// -----
// Ρουτίνα αντιστροφής μπάρας καταλόγου επιλογών.
// -----
void MenuSelect (unsigned char prmno, unsigned char mno)
{
    InvertArea(0x00,MCoord[prmno][0],MCoord[prmno][1],MCoord[prmno][2],
MCoord[prmno][3]);
    if (mno!=0xff)
        InvertArea(0xff,MCoord[mno][0],MCoord[mno][1],MCoord[mno][2],
MCoord[mno][3]);
}

// -----
// Εκτύπωση παραθύρου πληροφοριών της πτυχιακής.
// -----
void AboutScreen (void)
{
    EnableOnlySerialInt();
    ShowClock=false;
    ClearMainScreen();
    LoadLockPic();
    LoadTitlePic();
    TextGoto(6,10);
    PrintMultiChar('-',20);
    TextGoto(7,10);
    PrintString("Παναγιώτης Κεντερολής");
    TextGoto(8,10);
    PrintChar(6);
    PrintString("Τερματικό ελέγχου");
    TextGoto(9,10);
}

```

```

    PrintString("πρόσβασης σε χώρο");
    TextGoto(10,10);
    PrintString("& ένδειξης μηνυμάτων");
    TextGoto(11,10);
    PrintString("από H/Y.");
    TextGoto(12,2);
    PrintMultiChar('-',28);
    TextGoto(13,2);
    PrintString("Εισηγητής κος. Ιωαν. Έλληνας");
    ShowClock=true;
    RestoreInterrupts();
    DelaySecs(8);
}

// -----
// Εκτύπωση παραθύρου πληροφοριών χρήσης.
// -----
void HelpScreen (unsigned char HelpContext)
{
    EnableOnlySerialInt();
    ShowClock=false;
    ClearMainScreen();
    CreateWindow(5,3,12,28);
    TextGoto(6,10);
    PrintString("==Βοήθεια==");
    TextGoto(7,4);
    PrintMultiChar('-',24);
    TextGoto(8,4);
    PrintString(HelpMessages[HelpContext][0]);
    TextGoto(9,4);
    PrintString(HelpMessages[HelpContext][1]);
    TextGoto(10,4);
    PrintString(HelpMessages[HelpContext][2]);
    TextGoto(11,4);
    PrintString(HelpMessages[HelpContext][3]);
    ShowClock=true;
    RestoreInterrupts();
    DelaySecs(6);
}

// -----
// Εκτύπωση του πλαισίου τίτλου υπό-επιλογών.
// -----
void DrawTitleScreen ()
{
    ClearMainScreen();
    TextGoto(5,2);
    PrintMultiChar('-',28);
}

// -----
// Εκτύπωση πλαισίου και τίτλου εφαρμογής Ανακοινώσεων.
// -----
void DrawBBScreen ()
{
    DrawTitleScreen();
    TextGoto(4,10);
    PrintString("ΑΝΑΚΟΙΝΩΣΕΙΣ");
}

```

```

// -----
// Ρουτίνα διαχείρισης του κυρίου καταλόγου επιλογών.
// -----
void HandleMainMenu (unsigned char MenuKeyPressed)
{
    unsigned char previousButton;

    switch (MenuKeyPressed)
    {
        case '0':
            if (Count2Reset==0) SoftResetMicro();
            else Count2Reset--;
            break;
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
        case '*':
            Count2Reset=0x3f;
            break;
        case 'A':
            Count2Reset=0x3f;
            SoundBuzzer();
            ButtonSelect(SelButton,0xff);
            HelpScreen(0);
            ClearMainScreen();
            MainMenu();
            break;
        case 'B':
            Count2Reset=0x3f;
            SoundBuzzer();
            ButtonSelect(SelButton,0xff);
            AboutScreen();
            ClearMainScreen();
            ClearGrafxMainScreen();
            MainMenu();
            break;
        case 'C':
            Count2Reset=0x3f;
            previousButton=SelButton;
            if (SelButton>0)
            {
                SelButton--;
                ButtonSelect(previousButton,SelButton);
                SoundBuzzer();
            }
            break;
        case 'D':
            Count2Reset=0x3f;
            previousButton=SelButton;
            if (SelButton<3)
            {
                SelButton++;
                ButtonSelect(previousButton,SelButton);
                SoundBuzzer();
            }
            break;
        case '#':
            Count2Reset=0x3f;
    }
}

```

```

        SoundBuzzer();
        switch (SelButton)
        {
        case 0: ButtonSelect(SelButton,0xff);
                ComProgressBar();
                if (ServerOnLine)
                {
                    SendTerminalCommand("ATAL");
                    ATStored=false;
                    while (ATStored==false)
                    {
                    }
                    while (BulletinBoard())
                    {
                    }
                }
                break;
        case 1:
                ButtonSelect(SelButton,0xff);
                ComProgressBar();
                if (ServerOnLine)
                {
                    while (CourseMarks())
                    {
                    }
                }
                break;
        case 2: ButtonSelect(SelButton,0xff);
                HelpScreen(0);
                break;
        case 3: ButtonSelect(SelButton,0xff);
                AboutScreen();
                ClearGrafxMainScreen();
                break;
        }
        ClearMainScreen();
        MainMenu();
        break;
    }
}

// -----
// Ρουτίνα εκτέλεσης διαγνωστικών ελέγχων.
// -----
void DiagCheck (void)
{
    unsigned char DiagCode;
    unsigned char temp;

    temp>ShowClock;
    ShowClock=false;

    SendString("ATDIAG");

    //Έλεγχος Keyboard
    P1=0x55;
    if (P1==0x55)
    {
        P1=0xc3;
        if (P1==0xc3) DiagCode='1';
    }
    else DiagCode='0';
    SendChar(DiagCode);

    //Έλεγχος iButton Socket #1 & #2

```

```

P2_7=true;
if (P2_7==true)
{
    P2_6=true;
    if (P2_6==true) DiagCode='1';
}
else DiagCode='0';
SendChar(DiagCode);

//Έλεγχος οθόνης LCD
SetPosition(0x0fff);
BinaryDataWrite(0x55);
SendCommand(DWNADP);
if (BinaryDataRead()==0x55)
{
    BinaryDataWrite(0xaa);
    SendCommand(DWNADP);
    if (BinaryDataRead()==0xaa) DiagCode='1';
}
else DiagCode='0';
SendChar(DiagCode);

//Έλεγχος RTC
if (Check_RTC_Battery) DiagCode='1';
else DiagCode='0';
SendChar(DiagCode);

SendChar(ComVT);
ShowClock=temp;
}

// -----
// Ρουτίνα αποθήκευσης αποσταλμένης ανακοίνωσης.
// -----
void StoreAnnouncement (void)
{
    unsigned char temp;
    StorageSpace=AnnouncementStorage;
    StorageLines=0;
    temp=0x0;
    TextGoto(4,2);
    PrintChar(16);
    PrintString("Μεταφορά Ανακοίνωσης... ");
    SendTerminalCommand("ATOK");
    while (temp!=ComEOT)
    {
        temp=SIntReceiveChar();
        if (temp==ComEOS) StorageLines++;
        *StorageSpace=temp;
        StorageSpace++;
    }
}

// -----
// Ρουτίνα αποθήκευσης αποσταλμένων τίτλων ανακοινώσεων.
// -----
void StoreAnnouncementTitles (void)
{
    unsigned char temp;

```



```

ATStorageSpace=ATStorageStart;
temp=0x0;
SendTerminalCommand("ATOK");
while (temp!=ComEOT)
{
    temp=SIntReceiveChar();
    *ATStorageSpace=temp;
    ATStorageSpace++;
}
ATStored=true;
}

// -----
// Ρουτίνα εκτύπωσης τίτλων ανακοινώσεων σε μορφή καταλόγου επιλογών.
// -----
void PrintAnnouncementTitles (void)
{
    unsigned char i;

    ShowClock=false;
    ATStorageSpace=ATStorageStart;
    for (i=0;i<8;i++)
    {
        TextGoto(6+i,2);
        PrintBCD(i+1);
        PrintString(". ");
        ATStorageSpace=ATStorageStart+i*26;
        PrintString(ATStorageSpace);
    }
    ShowClock=true;
}

// -----
// Ρουτίνα εκτύπωσης ανακοίνωσης και πλοήγησης εντός αυτής.
// -----
void PrintAnnouncement (void)
{
    unsigned char temp;
    unsigned char NavKey;
    unsigned char InvalidKey;
    unsigned char NoEscape;

    ShowClock=false;
    DrawTitleScreen();
    MenuSelect(SelMenu,0xff);
    TextGoto(4,2);
    ATStorageSpace=ATStorageStart+SelMenu*26;
    PrintString(ATStorageSpace);
    ShowClock=true;
    LineNumber=6;
    Index=0;
    NoEscape=true;
    while (NoEscape)
    {
        temp=true;
        if (Index<StorageLines)
        {
            DisableRTCInt();
            TextGoto(LineNumber,2);
            StorageSpace=AnnouncementStorage+(unsigned int)(Index*29);
            PrintString(StorageSpace);

```

```

        if (*StorageSpace!=ComEOT)
        {
            Index++;
            LineNumber++;
            temp=false;
        }
        EnableRTCInt();
    }

    if (temp) LineNumber=14;

    if (LineNumber==14)
    {
        LineNumber=6;
        TextGoto(4,27);
        PrintBCD(Index/8);
        PrintChar('/');
        PrintBCD(StorageLines/8);
        InvalidKey=true;
        while (InvalidKey)
        {
            Secs2Delay=60;
            NavKey=WaitForKeyPress();
            if (NavKey=='*')
            {
                InvalidKey=false;
                NoEscape=false;
            }
            if (TimedOut==true)
            {
                InvalidKey=false;
                NoEscape=false;
            }

            if (NavKey=='D') InvalidKey=false;
            if (NavKey=='C')
            {
                if (Index>=16)
                {
                    Index-=16;
                    InvalidKey=false;
                }
            }
        }
    }
}

// -----
// Ρουτίνα χειρισμού Interface ανακοινώσεων.
// -----

unsigned char BulletinBoard (void)
{
    unsigned char temp;
    unsigned char NavKey;
    unsigned char Select;

    ShowClock=false;
    DrawBBScreen();
    PrintAnnouncementTitles();
    SelMenu=0;
    Select=true;
}

```

```

MenuSelect(SelMenu,SelMenu);
Secs2Delay=30;
temp=SelMenu;
ShowClock=true;
while (Select)
{
    if (temp!=SelMenu) MenuSelect(temp,SelMenu);
    temp=SelMenu;
    NavKey=WaitForKeyPress();

    if (NavKey=='C')
    {
        if (SelMenu>0) SelMenu--;
        Secs2Delay=30;
    }

    if (NavKey=='#')
    {
        ShowClock=false;
        DeployAnnouncement=false;
        ErrorReturned=false;
        SendString("ATSA");
        SendHex(SelMenu+1);
        SendChar(ComVT);
        while (DeployAnnouncement==false)
        {
            if (ErrorReturned==true) break;
        }
        ShowClock=true;
        Secs2Delay=30;
        if (ErrorReturned==false)
        {
            PrintAnnouncement();
            Select=false;
        }
    }
}
if (NavKey=='D')
{
    if (SelMenu<7) SelMenu++;
    Secs2Delay=30;
}
if (NavKey=='A')
{
    MenuSelect(SelMenu,0xff);
    HelpScreen(1);
    Select=false;
}
if (TimedOut) NavKey='*';
if (NavKey=='*') Select=false;
}
MenuSelect(SelMenu,0xff);
if (NavKey=='*') temp=false;
else temp=true;
return temp;
}

// -----
// Ρουτίνα διάγνωσης της κατάστασης της πόρτας.
// -----
unsigned char DoorStatus(void)
{
    unsigned char temp;
    temp=P3;

```

```

    temp=temp & 0x30;
    if (temp==0x20) return true;
    if (temp==0x10) return false;
}

// -----
// Ρουτίνα εκτέλεσης ηχητικού σήματος προειδοποίησης.
// -----
void SoundBuzzer (void)
{
    TL0=(unsigned char)(BuzzerDelay & 0xff);
    TH0=(unsigned char)((BuzzerDelay >> 8) & 0xff);
    TF0=false;
    TR0=true;
    P2_4=0;
    while (TF0==false)
    {
        }
    P2_4=1;
}

// -----
// Έλεγχος για εμφάνιση ειδικού μηνύματος στην οθόνη.
// -----
void CIPMessage (void) critical
{
    if (ClassInProgress==true)
    {
        TextGoto(12,9);
        PrintString("MHN ENOXΛEITE");
        TextGoto(13,2);
        PrintString("ΩΡΑ ΕΡΓΑΣΤΗΡΙΑΚΟΥ ΜΑΘΗΜΑΤΟΣ.");
    }
    else
    {
        TextGoto(12,2);
        PrintMultiChar(' ',28);
        TextGoto(13,2);
        PrintMultiChar(' ',28);
    }
    SoundBuzzer();
    CIPtemp=ClassInProgress;
}

// -----
// Εναλλαγή κατάστασης εμφάνισης ειδικού προειδοποιητικού
// μηνύματος.
// -----
void Toggle_CIP_Status (void)
{
    if (ClassInProgress==true)
    {
        ClassInProgress=false;
        SIntSendTerminalCommand("ATCIP-OFF");
    }
    else
    {
        ClassInProgress=true;
        SIntSendTerminalCommand("ATCIP-ON");
    }
}

```

```

// -----
// Ενεργοποίηση του βομβητή στον ακροδέκτη P2.4
// -----
void BuzzerOn (void)
{
    P2_4=0;
}

// -----
// Απενεργοποίηση του βομβητή στον ακροδέκτη P2.4
// -----
void BuzzerOff (void)
{
    P2_4=1;
}

// -----
// Ρουτίνα αποθήκευσης αποσταλμένων τίτλων μαθημάτων.
// -----
void StoreCourseTitles (void)
{
    unsigned char temp;

    StorageSpace=CourseTitlesStorage;
    temp=0x0;
    while (temp!=ComEOT)
    {
        temp=SIntReceiveChar();
        *StorageSpace=temp;
        StorageSpace++;
    }
    SendTerminalCommand("ATCTOK");
}

// -----
// Ρουτίνα εκτύπωσης τίτλων μαθημάτων σε μορφή καταλόγου επιλογών.
// -----
void PrintCourseTitles (void)
{
    unsigned char i;

    if (NoOfCourses==8)
    {
        DisableRTCInt();
        for (i=0;i<NoOfCourses;i++)
        {
            TextGoto(6+i,2);
            PrintString(CourseTitles[i]);
        }
        EnableRTCInt();
    }
}

// -----
// Εκτύπωση του καταλόγου Μαθημάτων δημοσιεύσιμων βαθμολογιών.
// Πλοηγήσιμο Interface επισκόπησης βαθμολογιών.
// -----
unsigned char CourseMarks (void)
{

```

```

unsigned char temp;
unsigned char Select;
unsigned char NavKey;

ShowClock=false;
DrawTitleScreen();
TextGoto(4,5);
PrintString("ΒΑΘΜΟΛΟΓΙΕΣ ΜΑΘΗΜΑΤΩΝ");
PrintCourseTitles();
SelMenu=0;
Select=true;
Secs2Delay=30;
temp=SelMenu;
MenuSelect(temp,SelMenu);
ShowClock=true;
while (Select)
{
    if (temp!=SelMenu) MenuSelect(temp,SelMenu);
    temp=SelMenu;
    NavKey=WaitForKeyPress();
    if (NavKey=='C')
    {
        if (SelMenu>0) SelMenu--;
        Secs2Delay=30;
    }
    if (NavKey=='#')
    {
        ShowClock=false;
        DeployCourseMarks=false;
        ErrorReturned=false;
        SendString("ATCM");
        SendHex(SelMenu+1);
        SendChar(ComVT);
        while (DeployCourseMarks==false)
        {
            if (ErrorReturned==true) break;
        }
        ShowClock=true;
        Secs2Delay=30;
        if (ErrorReturned==false)
        {
            PrintCourseMarks();
            Select=false;
        }
    }
    if (NavKey=='D')
    {
        if (SelMenu<7) SelMenu++;
        Secs2Delay=30;
    }
    if (NavKey=='A')
    {
        MenuSelect(SelMenu,0xff);
        HelpScreen(2);
        Select=false;
    }
    if (TimedOut) NavKey='*';
    if (NavKey=='*') Select=false;
}
MenuSelect(SelMenu,0xff);
if (NavKey=='*') temp=false;
else temp=true;
return temp;
}

```

```

// -----
// Ρουτίνα αποθήκευσης βαθμολογικού πίνακα μαθήματος όπως
// αυτός αποστέλλεται από την εφαρμογή διαχείρισης τερματικού.
// -----
void StoreCourseMarks (void)
{
    unsigned char temp;

    TextGoto(4,2);
    PrintChar(16);
    PrintString("Μεταφορά Βαθμολογιών... ");
    SendTerminalCommand("ATOK");
    CMStorageLines=0;
    StorageSpace=CourseMarksStorage;
    temp=0x0;
    while (temp!=ComEOT)
    {
        temp=SIntReceiveChar();
        if (temp==ComEOS) CMStorageLines++;
        *StorageSpace=temp;
        StorageSpace++;
    }
}

// -----
// Ρουτίνα εκτύπωσης βαθμολογικού πίνακα μαθήματος και
// πλοήγηση σε αυτόν.
// -----
void PrintCourseMarks (void)
{
    unsigned char NoEscape;
    unsigned char temp;
    unsigned char NavKey;
    unsigned char InvalidKey;

    MenuSelect(SelMenu,0xff);
    ShowClock=false;
    DrawTitleScreen();
    if (NoOfCourses==8)
    {
        TextGoto(4,2);
        PrintString(CourseTitles[SelMenu]);
    }
    ShowClock=true;
    LineNumber=6;
    Index=0;
    NoEscape=true;
    while (NoEscape)
    {
        temp=true;
        if (Index<CMStorageLines)
        {
            DisableRTCInt();
            TextGoto(LineNumber,2);
            StorageSpace=CourseMarksStorage+(unsigned int)(Index*29);
            PrintString(StorageSpace);
            if (*StorageSpace!=ComEOT)
            {
                Index++;
                LineNumber++;
                temp=false;
            }
        }
    }
}

```

```

    }
    EnableRTCInt();
}
if (temp) LineNumber=14;
if (LineNumber==14)
{
    LineNumber=6;
    InvalidKey=true;
    while (InvalidKey)
    {
        Secs2Delay=60;
        NavKey=WaitForKeyPress();
        if (NavKey=='*')
        {
            InvalidKey=false;
            NoEscape=false;
        }

        if (TimedOut==true)
        {
            InvalidKey=false;
            NoEscape=false;
        }
        if (NavKey=='D') InvalidKey=false;
        if (NavKey=='C')
        {
            if (Index>=16)
            {
                Index-=16;
                InvalidKey=false;
            }
        }
    }
}
}
}

// -----
// Ρουτίνα αντιγραφής του προκαθορισμένου μηνύματος από την
// μνήμη προγράμματος στον κυκλικό Buffer στην μνήμη δεδομένων.
// -----
void InitMarquee (void)
{
    unsigned char i;

    for (i=0;i<29;i++)
    {
        MarqueeMsg[i]=MarqueeProto[i];
    }
    MarqueeMode=MarqueeMode & 0x0f;
}

// -----
// Ρουτίνα περιστροφής του μηνύματος στον κυκλικό Buffer και
// εκτύπωση αυτού στην οθόνη LCD.
// -----
void Marquee (void) critical
{
    unsigned char temp;
    unsigned char i;

    EnableOnlySerialInt();

```



```

temp=MarqueeMsg[0];
for (i=0;i<27;i++)
{
    MarqueeMsg[i]=MarqueeMsg[i+1];
}
MarqueeMsg[27]=temp;
TextGoto(4,2);
PrintString(MarqueeMsg);
RestoreInterrupts();
}

// -----
// Ρουτίνα αποθήκευσης του νέου μηνύματος, που αποστέλλεται
// από τον διαχειριστή του τερματικού, στον κυκλικό Buffer.
// -----
void StoreMarqueeText (void)
{
    unsigned char i;

    for (i=0;i<29;i++)
    {
        MarqueeMsg[i]=SIntReceiveChar();
    }
    MarqueeMode=ShowMarqueeInRAM+ShowMarquee;
}

// -----
// Ρουτίνα ελέγχου των παραμέτρων εμφάνισης του κυλιόμενου
// μηνύματος από τον διαχειριστή του τερματικού.
// -----
void MarqueeSetup (void)
{
    unsigned char temp;

    temp=SIntReceiveChar();
    switch (temp)
    {
    case '0'://Απενεργοποίηση του κυλιόμενου μηνύματος.
        MarqueeMode=MarqueeMode & 0xf0;
        TextGoto(4,2);
        PrintMultiChar(' ',28);
        SIntSendTerminalCommand("ATMOFF");
        break;
    case '1'://Ενεργοποίηση του κυλιόμενου μηνύματος.
        MarqueeMode=(MarqueeMode & 0xf0)+ ShowMarquee;
        MarqueeDelay=4;
        SIntSendTerminalCommand("ATMON");
        break;
    case '2'://Επαναφορά του προκαθορισμένου κυλιόμενου μηνύματος.
        InitMarquee();
        MarqueeMode=(MarqueeMode & 0x0f);
        SIntSendTerminalCommand("ATDMR");
        break;
    case '3'://Αποθήκευση νέου κυλιόμενου μηνύματος.
        StoreMarqueeText();
        MarqueeMode=(MarqueeMode & 0x0f) + ShowMarqueeInRAM;
        SIntSendTerminalCommand("ATNMS");
        break;
    }
    SIntSendTerminalCommand("ATOK");
}

```

---

---

## FUNC.H

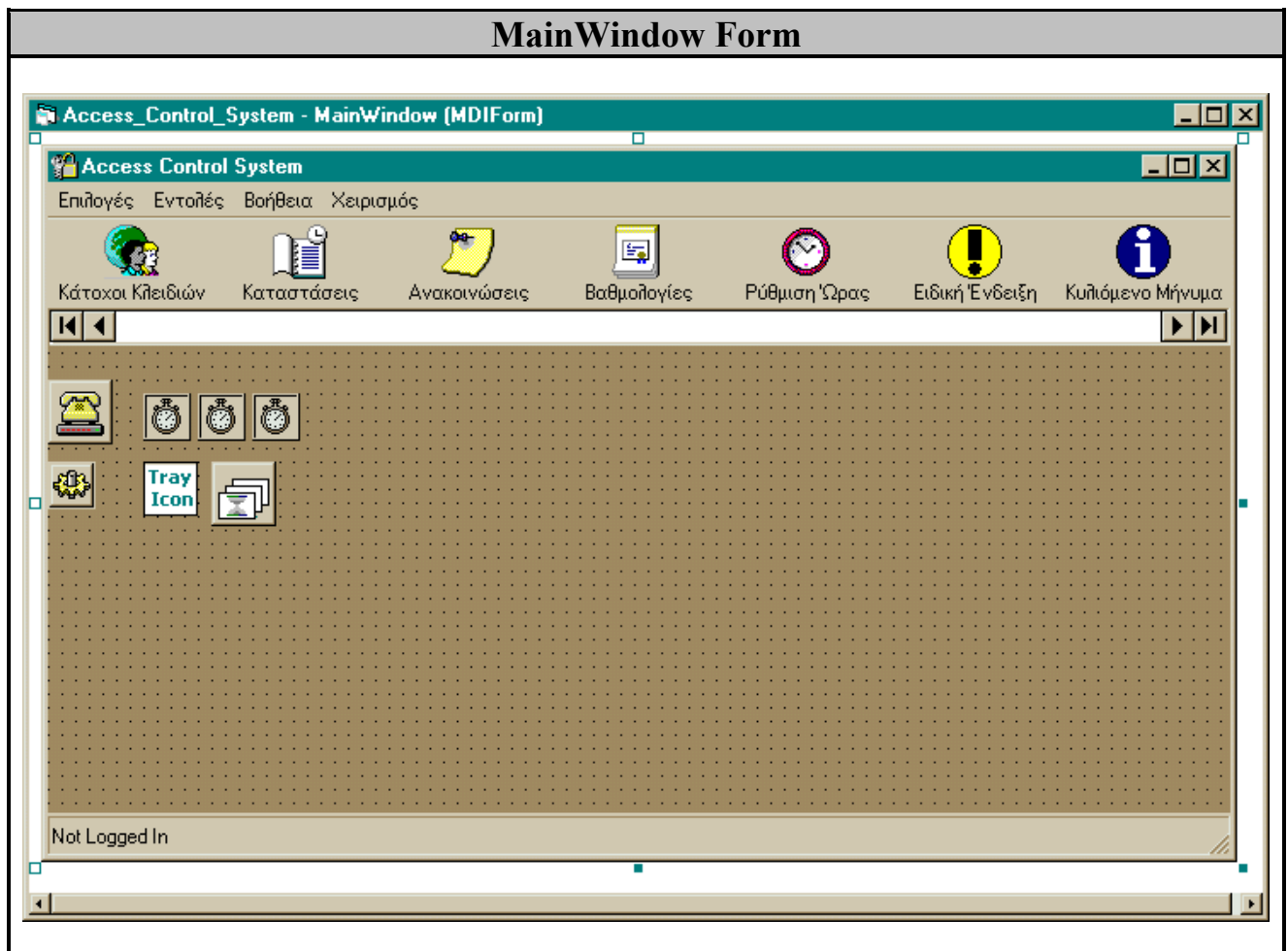
```
// -----  
// Δηλώσεις ρουτινών-συναρτήσεων.  
// -----  
  
unsigned char HEX2ah (unsigned char);  
unsigned char HEX2al (unsigned char);  
unsigned char dBCD2Binary (unsigned char);  
void DelaySecs (unsigned char);  
void EnableInts (void);  
void DisableInts (void);  
void EnableTimer0Int (void);  
void DisableTimer0Int (void);  
void MainMenu (void);  
void FillMainScreen (unsigned char);  
void ClearMainScreen (void);  
void DrawInitScreen (void);  
void LockDoor (void);  
void UnlockDoor (void);  
void ComProgressBar (void);  
void ButtonSelect (unsigned char, unsigned char);  
void MenuSelect (unsigned char, unsigned char);  
void HelpScreen (unsigned char);  
void AboutScreen (void);  
void DrawTitleScreen ();  
void DrawBBScreen ();  
unsigned char BulletinBoard (void);  
void HandleMainMenu (unsigned char);  
void DiagCheck (void);  
void PrintAnnouncement (void);  
void PrintAnnouncementTitles (void);  
void StoreAnnouncementTitles (void);  
void StoreAnnouncement (void);  
  
unsigned char IsMasterKey (void);  
void StoreMasterKey (void);  
void HandleMasterKeyInsertion (void);  
void StoreKeyHolderID (void);  
  
unsigned char DoorStatus (void);  
void SoundBuzzer (void);  
void BuzzerOn (void);  
void BuzzerOff (void);  
  
unsigned char CourseMarks (void);  
void StoreCourseTitles (void);  
void PrintCourseTitles (void);  
void PrintCourseMarks (void);  
void StoreCourseMarks (void);  
  
void Toggle_CIP_Status (void);  
void CIPMessage (void) critical;  
  
void InitMarquee (void);  
void Marquee (void) critical;  
void StoreMarqueeText (void);  
void MarqueeSetup (void);
```

# Κεφάλαιο 7<sup>ο</sup>

## Κώδικας Λογισμικού του Ηλεκτρονικού Υπολογιστή

Το λογισμικό που διαχειρίζεται το τερματικό και εκτελείται από τον Ηλεκτρονικό Υπολογιστή έχει δημιουργηθεί στην γλώσσα προγραμματισμού Microsoft Visual Basic 6.0 και περιλαμβάνει τον παρακάτω κώδικα και φόρμες παραθύρων. Το λογισμικό διαχείρισης του τερματικού, με την χρήση ειδικού control μπορεί να εκτελείται σε έναν υπολογιστή με λειτουργικό σύστημα Windows NT/2000 ως υπηρεσία συστήματος (System Service). Δεν απαιτεί έτσι την ανάγκη επιτήρησης από κάποιο χρήστη, ενώ δίδει πολλές δυνατότητες στον διαχειριστή για έλεγχο εξ αποστάσεως.

### Φόρμες



## Κώδικας Φόρμας MainWindow

Option Explicit

Dim mlWindowState As Long 'Remember the previous window state when form is minimized

Private Sub aboutme\_Click()

    Call ShowForm(DesignerInfo)

End Sub

Private Sub admin\_pwd\_change\_Click()

    AdminPwdChange.Show

End Sub

Private Sub AL\_List\_Click()

    Call ShowForm(AccessLogs)

End Sub

Private Sub AnnouncementsWindow\_Click()

    Call ShowForm(Announcements)

End Sub

Private Sub app\_end\_Click()

    MDIForm\_Terminate

    End

End Sub

Private Sub ChangeComPort\_Click()

Dim X As Byte

Dim msg As String

    X = 0

    While X > 4 Or X < 1

        On Error Resume Next

        X = InputBox("Παρακαλώ δώστε τον αριθμό της σειριακής θύρας στην οποία είναι συνδεδεμένο το τερματικό." + CRLF + CRLF + "Έγκυρες θεωρούνται οι τιμές 1->4.", "Αλλαγή Θύρας Επικοινωνίας Τερματικού", MSComm1.CommPort)

        If X = 0 Then Exit Sub

    Wend

    MSComm1.PortOpen = False

    MSComm1.CommPort = X

    MSComm1.PortOpen = True

```
Call NTService1.SaveSetting("Parameters", "ComPort", X)
msg = "Ορίσθηκε ως νέα θύρα επικοινωνίας η θύρα COM" + Trim$(Str$(X)) + "."
MsgBox msg, vbInformation
Call ServiceLogMessage(msg)
```

End Sub

```
Private Sub CIPMessageShow_Click()
```

```
    WaitForTerminalAcknowledge = True
    CIPMessageShow.Enabled = False
    Toolbar1.Enabled = False
    CIPMessageIsDisplayed = False
    Call TimeOutPeriod(4)
    SendCommandToTerminal ("ATS")
    While WaitForTerminalAcknowledge = True And TimedOut = False
        DoEvents
    Wend
    If TimedOut = True Then
        MsgBox "Πρόβλημα επικοινωνίας με το τερματικό." + CRLF + "Δεν ανταποκρίθηκε εντός χρονικού διαστήματος 2 δευτερολέπτων." + CRLF + CRLF + "Ελέγξτε τις συνδέσεις και την τροφοδοσία του τερματικού.", vbCritical
        SendCommandToTerminal ("@" )
    End If
    CIPMessageShow.Enabled = True
    Toolbar1.Enabled = True
```

End Sub

```
Private Sub DelayTimer_Timer()
```

```
    DelayTimer.Enabled = False
```

End Sub

```
Private Sub diag_Click()
```

```
    Call SendCommandToTerminal("ATD")
    Call TimeOutPeriod(10)
    WaitForTerminalAcknowledge = True
    Me.Enabled = False
    While WaitForTerminalAcknowledge = True And TimedOut = False
        DoEvents
    Wend
    Me.Enabled = True
    If TimedOut = True Then MsgBox "Πρόβλημα επικοινωνίας με το τερματικό." + CRLF + "Δεν ανταποκρίθηκε εντός χρονικού διαστήματος 5 δευτερολέπτων." + CRLF + CRLF + "Ελέγξτε τις συνδέσεις και την τροφοδοσία του τερματικού.", vbCritical
    WaitForTerminalAcknowledge = True
```

End Sub

---

---

```
Private Sub displayform_Click()
```

```
    TrayControl1_DblClick
```

```
End Sub
```

```
Private Sub KO_List_Click()
```

```
    Call ShowForm(AllKeyOwners)
```

```
End Sub
```

```
Private Sub lockdoor_Click()
```

```
    If TxInProgress = True Then Exit Sub
```

```
    WaitForTerminalAcknowledge = True
```

```
    TxInProgress = True
```

```
    DoorLocked = False
```

```
    DoorIsOpen = False
```

```
    Me.Enabled = False
```

```
    Call TimeOutPeriod(10)
```

```
    Call SendCommandToTerminal("ATL")
```

```
    While DoorLocked = False And TimedOut = False And DoorIsOpen = False
```

```
        DoEvents
```

```
    Wend
```

```
    Me.Enabled = True
```

```
    TxInProgress = False
```

```
    If TimedOut = True Then
```

```
        Call SendCommandToTerminal("ATA")
```

```
        MsgBox "Πρόβλημα επικοινωνίας με το τερματικό." + CRLF + "Δεν ανταποκρίθηκε εντός  
χρονικού διαστήματος 10 δευτερολέπτων." + CRLF + CRLF + "Ξαναδοκιμάστε και αν το  
πρόβλημα επιμένει ελέγξτε τις συνδέσεις και την τροφοδοσία του τερματικού.", vbCritical
```

```
    End If
```

```
    If DoorIsOpen = True Then MsgBox "Η πόρτα είναι ανοικτή." + CRLF + "Πρέπει πρώτα να την  
κλείσετε.", vbInformation
```

```
    If DoorLocked = True Then MsgBox "Η πόρτα κλειδώθηκε.", vbInformation
```

```
End Sub
```

```
Private Sub login_Click()
```

```
    AdminPwdCheck.Show vbModal
```

```
End Sub
```

```
Private Sub logout_Click()
```

```
    With Me
```

```
        Unload AccessLogs
```

```

Unload AllKeyOwners
Unload DiagCheckResults
Unload Announcements
Unload AdminPwdChange
Unload MarqueeSetup
.KO_List.Enabled = False
.AL_List.Enabled = False
.AnnouncementsWindow.Enabled = False
.Publicise_Marks.Enabled = False
.Toolbar1.Enabled = False
.StatusBar.SimpleText = "Not Logged In"
.login.Enabled = True
.logout.Enabled = False
.commands.Enabled = False
If FirstTimeCalled = False Then Call ServiceLogMessage("Ο διαχειριστής αποσυνδέθηκε.")
FirstTimeCalled = False
End With

```

End Sub

```
Private Sub MarqueeChange_Click()
```

```
    MarqueeSetup.Show
```

End Sub

```
Private Sub MDIForm_Load()
```

```
    Dim strDisplayName As String
```

```
    Dim cp As Byte
```

```
    Dim X
```

```
    If App.PrevInstance Or NTService1.Running Then
```

```
        TrayControl1.Enabled = False 'minimize to tray
```

```
        Me.Hide
```

```
    End
```

```
    End If
```

```
    Call SetConstants
```

```
    Call LoadConversionTable
```

```
    On Error GoTo Err_Load
```

```
    strDisplayName = NTService1.DisplayName
```

```
    If Command = "-install" Then
```

```
        Call FindTerminalComPort
```

```
        If TerminalFound = False Then
```

```
            MsgBox "Το Τερματικό δεν βρέθηκε συνδεδεμένο. Η εγκατάσταση της υπηρεσίας  
τερματίστηκε." + CRLF + "Ξαναπροσπαθήστε.", vbCritical + vbApplicationModal, "Σφάλμα  
αναγνώρισης θύρας σύνδεσης τερματικού"
```

```
            app_end_Click
```

```
        End If
```

```

' Enable interaction with desktop.
NTService1.Interactive = True
If NTService1.Install Then
    Call NTService1.SaveSetting("Parameters", "AdminPwd", Crypt("dooradmin",
EncryptionPassword))
    Call NTService1.SaveSetting("Parameters", "ComPort", MSComm1.CommPort)
    MsgBox "Η υπηρεσία " + strDisplayName + " εγκαταστάθηκε κανονικά και το τερματικό
βρέθηκε στην σειριακή θύρα COM" + MSComm1.CommPort, vbInformation
Else
    MsgBox "Η υπηρεσία " + strDisplayName + " δεν εγκατεστάθη.", vbCritical
End If
app_end_Click
ElseIf Command = "-uninstall" Then
If NTService1.Uninstall Then
    MsgBox "Η υπηρεσία " + strDisplayName + " αφαιρέθηκε επιτυχώς", vbInformation
Else
    MsgBox "Η υπηρεσία " + strDisplayName + " δεν αφαιρέθηκε επιτυχώς.", vbCritical
End If
app_end_Click
ElseIf Command = "-debug" Then
NTService1.Debug = True
ElseIf Command <> "" Then
    MsgBox "Άγνωστη εντολή", vbExclamation
app_end_Click
End If

' Connect service to Windows NT services controller.
On Error Resume Next
NTService1.StartService
MSComm1.CommPort = NTService1.GetSetting("Parameters", "ComPort", 2)
AdministratorPassword = DeCrypt(NTService1.GetSetting("Parameters", "AdminPwd",
Crypt("dooradmin", EncryptionPassword)), EncryptionPassword)
On Error Resume Next
MSComm1.PortOpen = True

TrayControl1.Enabled = True
TrayControl1.ToolTipText = "Access Control System"
TrayControl1_DblClick
adminmenu.Visible = False
FirstTimeCalled = True
logout_Click
Exit Sub
Err_Load:
' Error starting service
app_end_Click
End Sub

Private Sub MDIForm_Resize()

If Me.WindowState = vbMaximized Then
displayform.Caption = "Απόκρυψη"

```



```

Else
    displayform.Caption = "Εμφάνιση"
    Me.Hide
End If

End Sub

Private Sub MDIForm_Terminate()

    On Error Resume Next
    MSComm1.PortOpen = False
    TrayControl1.Enabled = False

End Sub

Private Sub MDIForm_Unload(Cancel As Integer)

    Cancel = True
    Me.WindowState = vbMinimized
    Me.Hide
    TrayControl1.Enabled = True 'minimize to tray

End Sub

Private Sub MSComm1_OnComm()

Static RxTemp As String

If MSComm1.CommEvent = comEvReceive Then
    RxTemp = MSComm1.Input

    If RxTemp = ComEOT Then Call SendCommandToTerminal("ATA")

    If RxTemp = ComVT Then

        Beep
        InSerDataBuffer2 = InSerDataBuffer
        InSerDataBuffer = ""
        ReceivedCommand = Right$(InSerDataBuffer2, Len(InSerDataBuffer2) + 1 - InStr(1,
InSerDataBuffer2, "AT", vbTextCompare))

        If ReceivedCommand = "ATCIP-ON" Then
            WaitForTerminalAcknowledge = False
            MsgBox "Ενεργοποιήθηκε η ειδική ένδειξη.", vbInformation
        End If
        If ReceivedCommand = "ATCIP-OFF" Then
            WaitForTerminalAcknowledge = False
            MsgBox "Απενεργοποιήθηκε η ειδική ένδειξη.", vbInformation
        End If

        If ReceivedCommand = "ATCTOK" Then
            MsgBox "Ενημερώθηκαν οι τίτλοι μαθημάτων.", vbInformation, "Βαθμολογίες"
        End If
    End If
End Sub

```

```

End If
If ReceivedCommand = "ATNMS" Then
    MsgBox "Ενημερώθηκε το κείμενο του κυλιόμενου μηνύματος.", vbInformation,
"Κυλιόμενο μήνυμα τερματικού"
End If
If ReceivedCommand = "ATDMR" Then
    MsgBox "Επαναφέρθηκε το προκαθορισμένο κείμενο του κυλιόμενου μηνύματος.",
vbInformation, "Κυλιόμενο μήνυμα τερματικού"
End If
If ReceivedCommand = "ATMON" Then
    MsgBox "Ενεργοποιήθηκε το κυλιόμενο μήνυμα.", vbInformation, "Κυλιόμενο μήνυμα
τερματικού"
End If
If ReceivedCommand = "ATMOFF" Then
    MsgBox "Απενεργοποιήθηκε το κυλιόμενο μήνυμα.", vbInformation, "Κυλιόμενο μήνυμα
τερματικού"
End If
If ReceivedCommand = "ATOK" Then WaitForTerminalAcknowledge = False
If ReceivedCommand = "ATEDO" Then DoorIsOpen = True
If ReceivedCommand = "ATDL" Then DoorLocked = True
If ReceivedCommand = "ATDU" Then DoorLocked = False
If ReceivedCommand = "ATAACK" Then TerminalFound = True
If ReceivedCommand = "ATTA" Then
    Call Delay_msecs(10)
    TimeSet_Click
End If

If Left$(ReceivedCommand, 4) = "ATID" Then Call ID_Request
If Left$(ReceivedCommand, 4) = "ATAL" Then Call SendAnnouncementsList
If Left$(ReceivedCommand, 4) = "ATSA" Then Call SendAnnouncement
If Left$(ReceivedCommand, 4) = "ATCM" Then Call SendCourseMarks
If Left$(ReceivedCommand, 5) = "ATREQ" Then Call EntryRequest
If Left$(ReceivedCommand, 7) = "ATEXRQ-" Then Call ExitRequest
If Left$(ReceivedCommand, 6) = "ATLRQ-" Then Call LockRequest
If Left$(ReceivedCommand, 6) = "ATKMP-" Then Call AddKeyToKeyMediaPool
If Left$(ReceivedCommand, 6) = "ATDIAG" Then Call DiagnosticsDecode

Else
    InSerDataBuffer = InSerDataBuffer + RxTemp
End If

End If

End Sub

Private Sub NTService1_Start(Success As Boolean)

On Error GoTo Err_Start

' TODO: Begin processing

```

```

    Success = True      ' Report success
    Exit Sub

Err_Start:
    Call ServiceLogError "[" & Err.Number & "]" & Err.Description)
End Sub

Private Sub NTService1_Stop()
On Error GoTo Err_Stop

    ' TODO: Suspend processing and release resources
    app_end_Click      ' End process
    Exit Sub

Err_Stop:
    Call ServiceLogError "[" & Err.Number & "]" & Err.Description)
End Sub

Private Sub Publicise_Marks_Click()

    Call ShowForm(AllCourses)

End Sub

Private Sub RefreshTrayIcon_Timer()

    TrayControl1.Enabled = True

End Sub

Private Sub Set_MasterKey_Click()

    Call ShowForm(MasterKeySet)

End Sub

Private Sub TimeOutTimer_Timer()

    TimedOut = True
    TimeOutTimer.Enabled = False

End Sub

Private Sub TimeSet_Click()

    Toolbar1.Enabled = False
    Call SetRTCTime
    Toolbar1.Enabled = True

End Sub

```

---

---

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
Select Case Button.Index
```

```
Case 1
```

```
    KO_List_Click
```

```
Case 2
```

```
    AL_List_Click
```

```
Case 3
```

```
    AnnouncementsWindow_Click
```

```
Case 4
```

```
    Publicise_Marks_Click
```

```
Case 5
```

```
    TimeSet_Click
```

```
Case 6
```

```
    CIPMessageShow_Click
```

```
Case 7
```

```
    MarqueeChange_Click
```

```
End Select
```

```
End Sub
```

```
Private Sub unlockdoor_Click()
```

```
    If TxInProgress = True Then Exit Sub
```

```
    WaitForTerminalAcknowledge = True
```

```
    TxInProgress = True
```

```
    DoorIsOpen = False
```

```
    DoorLocked = True
```

```
    Me.Enabled = False
```

```
    Call TimeOutPeriod(10)
```

```
    Call SendCommandToTerminal("ATU")
```

```
    While DoorIsOpen = False And TimedOut = False And DoorLocked = True
```

```
        DoEvents
```

```
    Wend
```

```
    Me.Enabled = True
```

```
    TxInProgress = False
```

```
    If TimedOut = True Then
```

```
        Call SendCommandToTerminal("ATA")
```

```
        MsgBox "Πρόβλημα επικοινωνίας με το τερματικό." + CRLF + "Δεν ανταποκρίθηκε εντός  
χρονικού διαστήματος 10 δευτερολέπτων." + CRLF + CRLF + "Ξαναδοκιμάστε και αν το  
πρόβλημα επιμένει ελέγξτε τις συνδέσεις και την τροφοδοσία του τερματικού.", vbCritical
```

```
    End If
```

```
    If DoorIsOpen = True Then MsgBox "Η πόρτα είναι ήδη ανοικτή.", vbInformation
```

```
    If DoorLocked = False Then MsgBox "Η πόρτα ξεκλειδώθηκε.", vbInformation
```

```
End Sub
```

```
Private Sub TrayControl1_Db1Click()
```

```
    'TrayControl1.Enabled = False 'remove tray icon
```

```
    If Me.WindowState = vbMinimized Or (isHidden And Me.WindowState = vbNormal) Then  
        isHidden = False
```

---

---

```
displayform.Caption = "Απόκρυψη"  
Me.WindowState = vbMaximized  
Call ShowForm(Me)  
Else  
Me.WindowState = vbMinimized  
isHidden = True  
displayform.Caption = "Εμφάνιση"  
Me.Hide  
End If  
  
End Sub  
  
Private Sub TrayControl1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
If Button = vbRightButton Then  
PopupMenu adminmenu  
End If  
End Sub
```

## AccessLogs Form

DateOfEntry	TimeOfEntry	NameOfKeyOwner	IOType	Lock
-------------	-------------	----------------	--------	------

## Κώδικας Φόρμας AccessLogs

Option Explicit

Private Const MARGIN\_SIZE = 60 ' in Twips

Private Sub cmdClearAccessLogs\_Click()

    Call ShowForm(LogClear)

End Sub

Private Sub cmdRestrict\_Click()

    Call ShowForm(RestrictDates)

End Sub

Private Sub Form\_Load()

    Call AccessLogBanding

    datPrimaryRS.RecordSource = "select  
DateOfEntry,TimeOfEntry,NameOfKeyOwner,IOType,Lock from [Entry Log] Order by

---

---

```
DateOfEntry,TimeOfEntry"  
    datPrimaryRS.Refresh  
    MSHFlexGrid1.Refresh  
    Call Refresh_AL_Grid
```

```
End Sub
```

```
Private Sub cmdClose_Click()
```

```
    Unload Me
```

```
End Sub
```

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

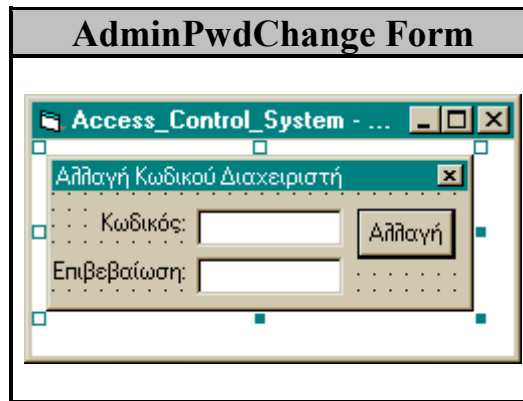
```
    Unload RestrictDates
```

```
    Unload LogClear
```

```
End Sub
```

```
Private Sub MSHFlexGrid1_Click()
```

```
End Sub
```



Με την φόρμα αυτή γίνεται δυνατή η αλλαγή του κωδικού του διαχειριστή του λογισμικού ελέγχου του τερματικού. Για να εκτελεστεί αυτή η λειτουργία θα πρέπει ήδη να έχει συνδεθεί ο διαχειριστής με τον παλιό του κωδικό.

### Κώδικας Φόρμας AdminPwdChange

```
Option Explicit
```

```
Private Sub ChangePwd_Click()
```

```
    If pwd(0).Text = pwd(1).Text Then
```

```
        If pwd(0).Text = "" Or Trim$(pwd(0).Text) = "" Then
```

```
            MsgBox "Ο κωδικός του διαχειριστή δεν μπορεί να είναι μηδενικός.", vbExclamation, "Αλλαγή κωδικού διαχειριστή"
```

```
        Else
```

```
            Call MainWindow.NTService1.SaveSetting("Parameters", "AdminPwd", Crypt(pwd(0).Text, EncryptionPassword))
```

```
            MsgBox "Ο κωδικός του διαχειριστή άλλαξε.", vbInformation, "Αλλαγή κωδικού διαχειριστή"
```

```
            Unload Me
```

```
        End If
```

```
    Else
```

```
        MsgBox "Οι κωδικοί που δώσατε δεν συμφωνούν. Παρακαλούμε ξαναπροσπαθήστε.", vbExclamation, "Λάθος κατα την εισαγωγή"
```

```
    End If
```

```
End Sub
```

```
Private Sub Form_Deactivate()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Call CXForm(Me)
```

```
End Sub
```

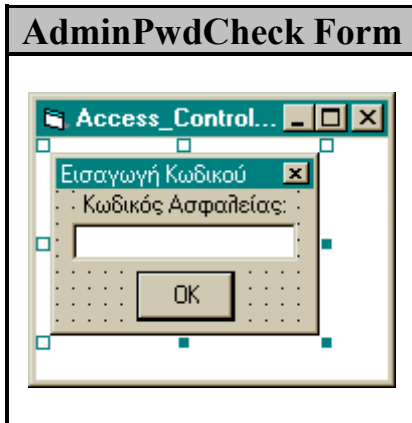
```
Private Sub pwd_KeyPress(Index As Integer, KeyAscii As Integer)
```

```
    KeyAscii = Asc(Chr$(KeyAscii))
```

```
    If KeyAscii = Asc(" ") Then KeyAscii = 0
```

```
End Sub
```





Με την φόρμα αυτή γίνεται ο έλεγχος του κωδικού του διαχειριστή του λογισμικού για την παροχή πρόσβασης σε ειδικές λειτουργίες.

### Κώδικας Φόρμας AdminPwdCheck

Option Explicit

Private Sub Command1\_Click()

```
If AdminPassword = DeCrypt(MainWindow.NTService1.GetSetting("Parameters",
"AdminPwd", Crypt("dooradmin", EncryptionPassword)), EncryptionPassword) Then
```

```
With MainWindow
```

```
.KO_List.Enabled = True
```

```
.AL_List.Enabled = True
```

```
.AnnouncementsWindow.Enabled = True
```

```
.Publicise_Marks.Enabled = True
```

```
.StatusBar.SimpleText = "Logged In"
```

```
.login.Enabled = False
```

```
.logout.Enabled = True
```

```
.commands.Enabled = True
```

```
.Toolbar1.Enabled = True
```

```
Call ServiceLogMessage("Ο διαχειριστής συνδέθηκε.")
```

```
Beep
```

```
Unload Me
```

```
End With
```

```
Else
```

```
AdminPassword.SelStart = 0
```

```
AdminPassword.SelLength = Len(AdminPassword.Text)
```

```
Beep
```

```
End If
```

End Sub

Private Sub Form\_Load()

```
Call CXForm(Me)
```

End Sub

**AllCourses Form**

### Κώδικας Φόρμας AllCourses

Option Explicit

Private Sub AllCoursesGrid\_Db1Click()

    With AllCoursesGrid

        .Col = 0

        CourseNo.Caption = Str\$(.Row) + ". "

        CourseTitle.Text = .Text

        .Col = 1

        DBFileName.Text = .Text

        .Col = 2

        If .Text = "Ενεργή" Then

            PublishMarks.Value = vbChecked

        Else

            PublishMarks.Value = vbUnchecked

        End If

        .Col = 0

        .ColSel = 2

    End With

End Sub

Private Sub cmdChange\_Click()

    Dim CourseNumber As Byte

        CourseNumber = Val(CourseNo.Caption)

```

If CourseNumber = 0 Then Exit Sub
datPrimaryRS.Refresh
On Error Resume Next
datPrimaryRS.Recordset.AbsolutePosition = CourseNumber
datPrimaryRS.Recordset.Fields("CourseName").Value = CourseTitle.Text
datPrimaryRS.Recordset.Fields("DBFileName").Value = DBFileName.Text
If PublishMarks.Value = vbChecked Then
    datPrimaryRS.Recordset.Fields("CanBeViewed").Value = True
Else
    datPrimaryRS.Recordset.Fields("CanBeViewed").Value = False
End If
On Error Resume Next
datPrimaryRS.Recordset.Update
AllCoursesGrid.Redraw = False
AllCoursesGrid.Refresh
datPrimaryRS.Refresh
datPrimaryRS.Refresh
AllCoursesGrid.Redraw = True
Call AllCoursesGrid_Refresh
End Sub

Private Sub cmdClose_Click()

    Unload Me

End Sub

Private Sub Form_Load()

    Call AllCoursesGrid_Refresh

End Sub

Private Sub SelectDBFile_Click()
    On Error GoTo errhandler
    CommonDialog1.Flags = cdIOFNFileMustExist + cdIOFNHideReadOnly + cdIOFNLongNames
+ cdIOFNPathMustExist
    CommonDialog1.ShowOpen
    DBFileName.Text = CommonDialog1.FileName
errhandler:
End Sub

Private Sub SendChangedCourseTitles_Click()

    Call SendCourseTitles

End Sub

```

## AllKeyOwners Form

The screenshot shows a Windows application window titled "Access\_Control\_System - AllKeyOwners (Form)". Inside the window, there is a form titled "Κάτοχοι Κλειδιών Ασφαλούς Πρόσβασης". The form contains a data grid with the following columns: "FirstNam", "Associatic", "ActiveKey", "iKeyNumb", and "ValidKeyUnrl". Below the grid, there are four buttons: "Προσθήκη", "Αφαίρεση", "Επεξεργασία", and "Κλείσιμο".

## Κώδικας Φόρμας AllKeyOwners

Option Explicit

```
Private Sub cmdAdd_Click()  
    With NewKeyOwner  
        .Caption = "Προσθήκη Κατόχου Κλειδιού Πρόσβασης"  
        .KO_SurNameLabel.Caption = ""  
        .KO_NameLabel.Caption = ""  
        On Error Resume Next  
        .datPrimaryRS.Recordset.MoveLast  
        .datPrimaryRS.Recordset.AddNew  
        .KO_PassCode.Text = ""  
        .Show  
    End With  
End Sub
```

```
Private Sub cmdEdit_Click()  
    MSHFlexGrid1_Db1Click  
End Sub
```

```
Private Sub cmdRemove_Click()  
    Dim surname As String  
    Dim firstname As String  
    Dim X As Integer  
  
    With datPrimaryRS
```

```

MSHFlexGrid1.Col = 0
surname = MSHFlexGrid1.Text
MSHFlexGrid1.Col = 1
firstname = MSHFlexGrid1.Text
X = MsgBox("Η διαδικασία αυτή θα αφαιρέσει τον κάτοχο κλειδιού ασφαλείας με τα
ακόλουθα στοιχεία;" + CRLF + CRLF + HorizontalTab + surname + " " + firstname + CRLF +
CRLF + "Είσαστε σίγουροι ότι θέλετε να συνεχίσετε με την αφαίρεση;", vbYesNo + vbQuestion,
"Αφαίρεση κατόχου κλειδιού ασφαλείας")
If X = vbYes Then
    .RecordSource = "select
Surname,FirstName,AssociationType,ActiveKey,iKeyNumber,ValidKeyUntil from [Key Owners]
where Surname = '" + surname + "' and FirstName ='" + firstname + "'"
    .Refresh
    If .Recordset.RecordCount = 1 Then .Recordset.Delete
    .Recordset.Update
    .RecordSource = "select
Surname,FirstName,AssociationType,ActiveKey,iKeyNumber,ValidKeyUntil from [Key Owners]
Order by Surname"
    .Recordset.Requery
    .Refresh
    Refresh_KO_Grid
End If
End With

End Sub

Private Sub Form_Load()

    Call Refresh_KO_Grid

End Sub

Private Sub cmdClose_Click()

    Unload Me

End Sub

Private Sub Form_Unload(Cancel As Integer)

    Unload KeyOwnerDetails

End Sub

Private Sub MSHFlexGrid1_DblClick()

With MSHFlexGrid1
    If .Row = 0 Then Exit Sub
    .Col = 0
    KeyOwnerDetails.KO_SurNameLabel.Caption = .Text

```

---

---

```
.Col = 1
KeyOwnerDetails.KO_NameLabel.Caption = .Text

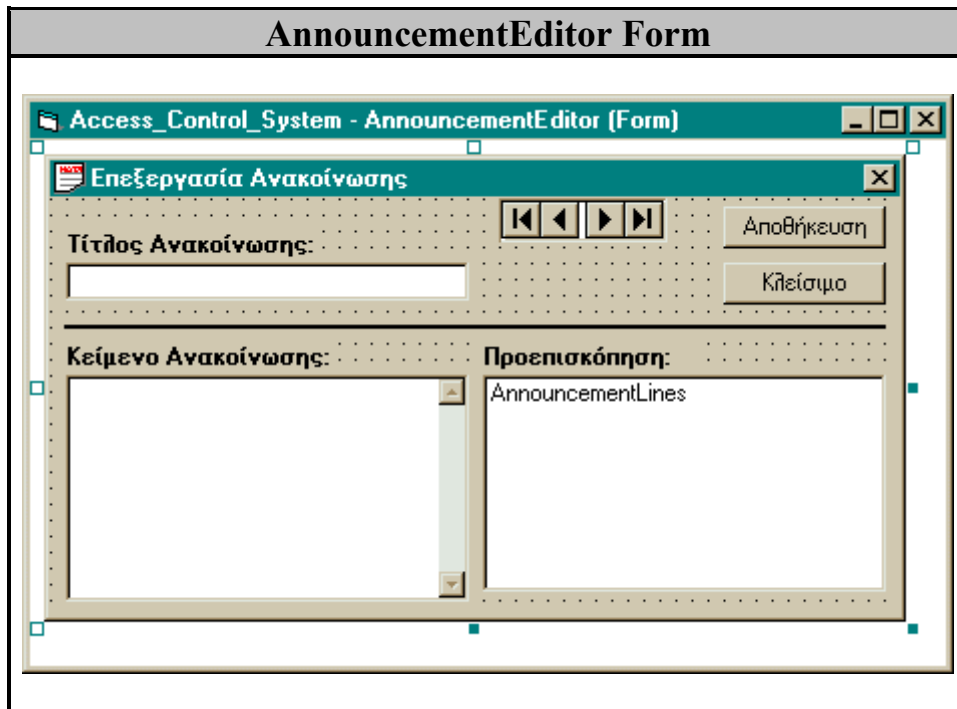
.Col = 0
.ColSel = .Cols - 1

With KeyOwnerDetails
    .datPrimaryRS.RecordSource = "select * from [Key Owners] where Surname = '" +
.KO_SurNameLabel.Caption + "' and FirstName='" + .KO_NameLabel.Caption + "'"
    .datPrimaryRS.Refresh
    .Visible = False

    Screen.MousePointer = vbHourglass
    KeyOwnerDetails.Show
    Screen.MousePointer = vbDefault

    .Visible = True
End With
End With

End Sub
```



### Κώδικας Φόρμας AnnouncementEditor

Option Explicit

Private Sub cmdClose\_Click()

    Me.Hide

End Sub

Private Sub cmdSaveAnnouncement\_Click()

    Call SaveAnnouncementChanges

End Sub

Private Sub Form\_Load()

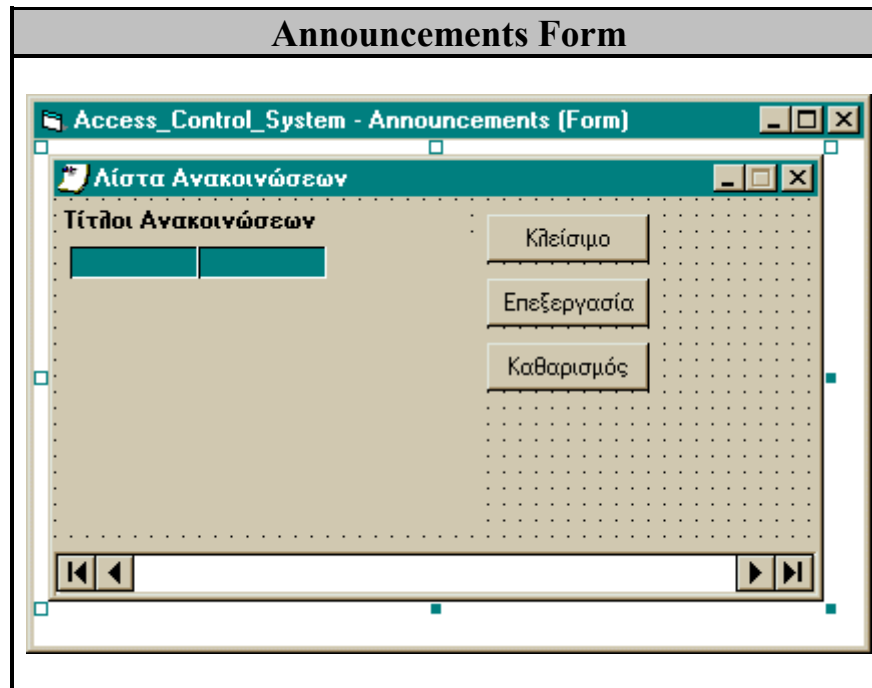
    CreateFormattedMessagePreview

End Sub

Private Sub MessageEditor\_Change()

    CreateFormattedMessagePreview

End Sub



### Κώδικας Φόρμας Announcements

Option Explicit

```
Private Sub AnnouncementsList_DblClick()
```

```
    Call EditAnnouncement
```

```
    Call ShowForm(AnnouncementEditor)
```

```
End Sub
```

```
Private Sub ClearSelectedAnnouncement_Click()
```

```
Dim X As Integer
```

```
    AnnouncementsList.Col = 1
```

```
    X = MsgBox("Η διαδικασία αυτή θα αφαιρέσει την ανακοίνωση με τίτλο: " + CRLF + CRLF + HorizontalTab + Str$(AnnouncementsList.Row) + ". " + AnnouncementsList.Text + CRLF + CRLF + "Είσαστε σίγουροι ότι θέλετε να συνεχίσετε με την αφαίρεση;", vbYesNo + vbCritical + vbQuestion, "Αφαίρεση ανακοινώσεων")
```

```
    If X = vbYes Then
```

```
        Call EditAnnouncement
```

```
        AnnouncementEditor.Announcement_Title.Text = ""
```

```
        AnnouncementEditor.MessageEditor.Text = ""
```

```
        Call SaveAnnouncementChanges
```

```
    End If
```

```
End Sub
```

```
Private Sub cmdClose_Click()
```

```
    Unload Me
```



---

---

End Sub

Private Sub EditSelectedAnnouncement\_Click()

    AnnouncementsList\_DblClick

End Sub

Private Sub Form\_Load()

    AnnouncementsList.ColWidth(0) = 180  
    AnnouncementsList.ColWidth(1) = 2800

End Sub

Private Sub Form\_Resize()

    With Me

        .Label1.Top = MARGIN\_SIZE  
        .Label1.Left = MARGIN\_SIZE  
        .AnnouncementsList.Top = MARGIN\_SIZE + Label1.Height  
        .AnnouncementsList.Left = MARGIN\_SIZE  
        .cmdClose.Top = MARGIN\_SIZE  
        .cmdClose.Left = .AnnouncementsList.Width + 3 \* MARGIN\_SIZE  
        .EditSelectedAnnouncement.Top = .cmdClose.Top + .cmdClose.Height + MARGIN\_SIZE  
        .ClearSelectedAnnouncement.Top = .EditSelectedAnnouncement.Top +  
        .EditSelectedAnnouncement.Height + MARGIN\_SIZE  
  
        .EditSelectedAnnouncement.Left = .cmdClose.Left  
        .ClearSelectedAnnouncement.Left = .EditSelectedAnnouncement.Left

        .Height = .AnnouncementsList.Height + 2 \* MARGIN\_SIZE + Label1.Height + 3 \*  
MARGIN\_CORRECTION

        .Width = .EditSelectedAnnouncement.Left + .EditSelectedAnnouncement.Width + 3 \*  
MARGIN\_SIZE

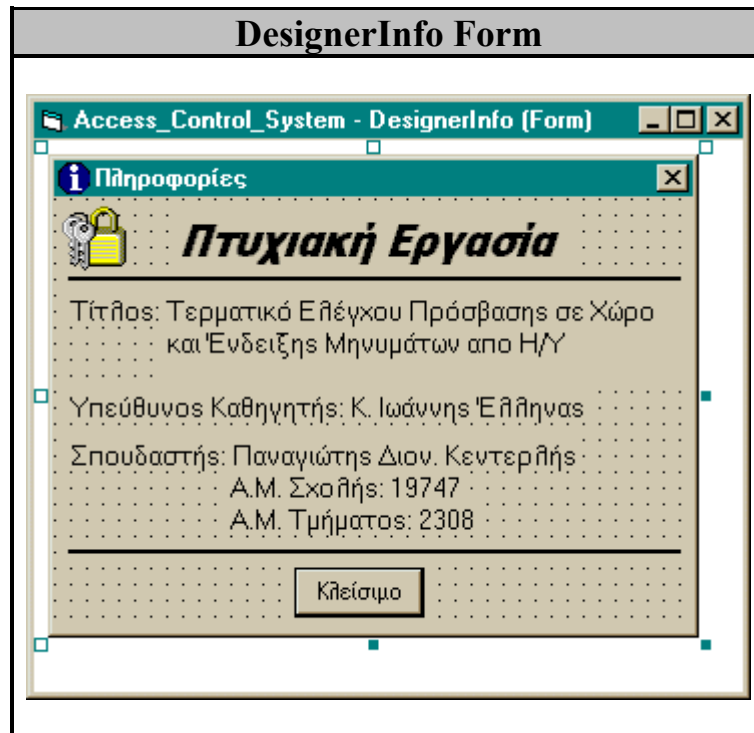
    End With

End Sub

Private Sub Form\_Unload(Cancel As Integer)

    Unload AnnouncementEditor

End Sub



### Κώδικας Φόρμας DesignerInfo

```
Option Explicit
```

```
Private Sub cmdClose_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Deactivate()
```

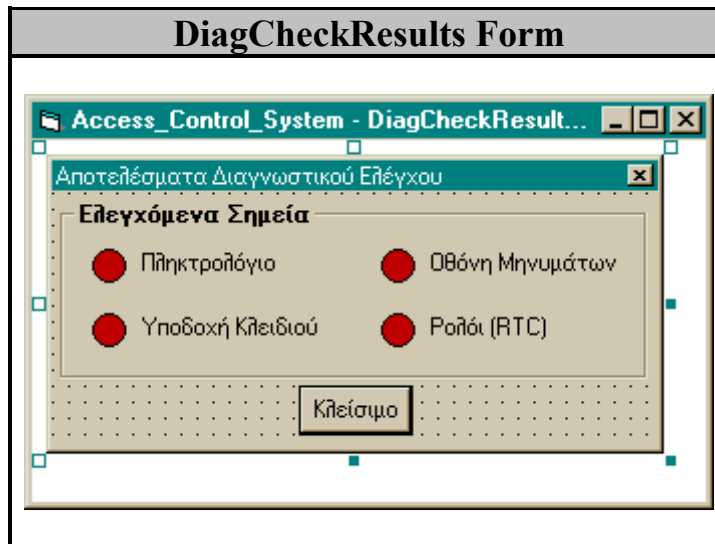
```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Call CXForm(Me)
```

```
End Sub
```



Με την φόρμα αυτή εμφανίζονται τα αποτελέσματα των διαγνωστικών ελέγχων που εκτελέστηκαν στο τερματικό, μετά από αίτηση του διαχειριστή.

### Κώδικας Φόρμας DiagCheckResults

Option Explicit

Private Sub Form\_Deactivate()

    Unload Me

End Sub

Private Sub Form\_Load()

    Frame1.Left = MARGIN\_SIZE

    Frame1.Top = MARGIN\_SIZE

    Me.Width = Frame1.Width + 2 \* MARGIN\_SIZE + MARGIN\_CORRECTION

    ok\_button.Left = (Me.Width - ok\_button.Width) / 2

    ok\_button.Top = Frame1.Height + 2 \* MARGIN\_SIZE

    Me.Height = Frame1.Height + (3 \* MARGIN\_SIZE) + ok\_button.Height + 5 \* MARGIN\_SIZE

    Call CXForm(Me)

End Sub

Private Sub ok\_button\_Click()

    Unload Me

End Sub

## KeyOwnerDetails Form

The image displays two side-by-side screenshots of the 'KeyOwnerDetails (Form)' window from the 'Access\_Control\_System'. Both windows have a title bar with the text 'Access\_Control\_System - KeyOwnerDetails (Form)'. The left window is titled 'Κάτοχος Κλειδιού Πρόσβασης' and has two tabs: 'Προσωπικά Στοιχεία' (selected) and 'Περισσότερα Στοιχεία'. It contains a profile picture icon, a name field ('Όνομα:'), a surname field ('Επώνυμο:'), an ID field ('Ιδιότητα:'), an address field ('Διεύθυνση:'), a phone field ('Τηλέφωνο:'), a mobile field ('Κινητό:'), and an email field ('e-mail:'). There is also a checkbox for 'Ενεργό Κλειδί'. The right window is titled 'Κάτοχος Κλειδιού Πρόσβασης' and has the same tabs. It features an information icon, an expiry date dropdown ('Ημερία Λήξης Ισχύος: Παρασκευή, 17 Νοέμβριος 2000'), a key number field ('Αριθμός Κλειδιού:'), and a code field ('Κωδικός:'). Below these is a large text area for 'Παρατηρήσεις:'. Both windows have navigation buttons at the bottom: back, forward, and buttons for 'Αποδοχή' (Accept) and 'Ακύρωση' (Cancel).

## Κώδικας Φόρμας KeyOwnerDetails

Option Explicit

Private Sub AcceptChanges\_Click()

On Error GoTo errhandle

If ValidPassCodeLength.FillColor = vbRed Then GoTo errhandle

If ValidKeyCodeLength.FillColor = vbRed Then GoTo errhandle

With datPrimaryRS

KO\_SurNameLabel.Caption = KO\_SurName.Text

KO\_NameLabel.Caption = KO\_FirstName.Text

On Error Resume Next

datPrimaryRS.Recordset.Fields(11).Value = Crypt(KO\_PassCode.Text, EncryptionPassword)

On Error GoTo errhandle

.Recordset.Update

.Refresh

AllKeyOwners.datPrimaryRS.Refresh

On Error Resume Next

MainWindow.datSecondaryRS.Recordset.Requery

```

MainWindow.datSecondaryRS.Refresh
Call Refresh_KO_Grid
End With
Unload Me
Exit Sub

errhandle:
If Err.Number = -2147467259 Then
    MsgBox "Κάποια στοιχεία έχουν ήδη αποδοθεί σε άλλο χρήστη.", vbCritical, "Πολλαπλή
χρήση στοιχείων"
Else
    MsgBox "Κάποια στοιχεία δεν έχουν συμπληρωθεί." + CRLF + "Παρακαλώ συμπληρώστε τα
αναγκαία πεδία και ξαναδοκιμάστε.", vbCritical, "Ελλιπή στοιχεία"
End If
End Sub

Private Sub CancelChanges_Click()

    Unload Me

End Sub

Private Sub EncryptedPassword_Change()

'On Error Resume Next
If IsNull(datPrimaryRS.Recordset.Fields(11).Value) Then
datPrimaryRS.Recordset.Fields(11).Value = ""
KO_PassCode.Text = DeCrypt(datPrimaryRS.Recordset.Fields(11).Value, EncryptionPassword)

End Sub

Private Sub Form_Activate()

If Me.Caption = "Προσθήκη Κατόχου Κλειδιού Πρόσβασης" Then
KO_ValidKey.Value = vbChecked
KO_ValidUntilDate.Value = Format$(Format$(Date$, "mm/dd/yyyy"), "dd/mm/yyyy")
End If

End Sub

Private Sub Form_Load()

SSTab1.Tab = 0

End Sub

Private Sub Form_Unload(Cancel As Integer)

Unload KeyPoolSelect

End Sub

```

```

Private Sub KO_FirstName_Change()

    KO_SurNameLabel.Caption = KO_SurName.Text
    KO_NameLabel.Caption = KO_FirstName.Text

End Sub

Private Sub KO_iButtonCode_Change()

    If Len(KO_iButtonCode.Text) < 16 Then ValidKeyCodeLength.FillColor = vbRed Else
ValidKeyCodeLength.FillColor = vbGreen

End Sub

Private Sub KO_iButtonCode_KeyPress(KeyAscii As Integer)

    KeyAscii = Asc(UCase$(Chr$(KeyAscii)))
    If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9")) And Not (KeyAscii >= Asc("A") And
KeyAscii <= Asc("F")) And Not (KeyAscii = 8) Then KeyAscii = 0

End Sub

Private Sub KO_PassCode_Change()

    If Len(KO_PassCode.Text) < 6 Then ValidPassCodeLength.FillColor = vbRed Else
ValidPassCodeLength.FillColor = vbGreen

End Sub

Private Sub KO_PassCode_KeyPress(KeyAscii As Integer)

    If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9")) And Not (KeyAscii = 8) Then
KeyAscii = 0

End Sub

Private Sub KO_SurName_Change()

    KO_SurNameLabel.Caption = KO_SurName.Text
    KO_NameLabel.Caption = KO_FirstName.Text

End Sub

Private Sub KO_SurNameLabel_Change()

    KO_NameLabel.Left = KO_SurNameLabel.Left + KO_SurNameLabel.Width + 60

End Sub

Private Sub mailto_Click()

Dim ret As Long

```

---

---

```
ret = Shell("Start.exe " & "mailto:" & """" & KO_mailto.Text & """" , 0)
```

```
End Sub
```

```
Private Sub SelectFromKeyPool_Click()
```

```
    If Me.Caption = "Κάτοχος Κλειδιού Πρόσβασης" Then
```

```
        SendKeyCode2KO = True
```

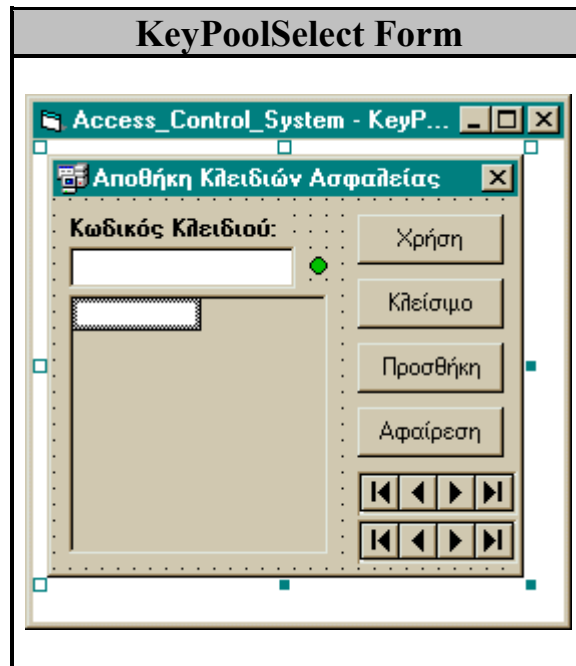
```
    Else
```

```
        SendKeyCode2KO = False
```

```
    End If
```

```
    Call ShowForm(KeyPoolSelect)
```

```
End Sub
```



### Κώδικας Φόρμας KeyPoolSelect

Option Explicit

```
Private Sub cmdAdd_Click()
```

```
    If ValidKeyCodeLength.FillColor = vbGreen Then
        With datPrimaryRS
            .Recordset.MoveLast
            .Recordset.AddNew
            .Recordset.Fields(0).Value = KO_iButtonCode.Text
            On Error Resume Next
            .Recordset.Update
            .Refresh
            Call KMP_Refresh
        End With
    End If
```

```
End Sub
```

```
Private Sub cmdClose_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub cmdRemove_Click()
```

```
    Dim color As Integer
    Dim keycode As String * 16
```

```
    With datPrimaryRS
        KeyPool.Col = 0
```



```

If KeyPool.Row = 0 Then Exit Sub
color = KeyPool.CellForeColor
keycode = KeyPool.Text

KeyPool.Redraw = False
.RecordSource = "select iKeyCode from [KeyMediaPool] where iKeyCode=" + KeyPool.Text
+ """"
.Refresh

If color = vbRed Then
    datSecondaryRS.RecordSource = "Select Surname,FirstName from [Key Owners] where
iKeyNumber=" + keycode + """"
    datSecondaryRS.Refresh
    Call SetConstants
    MsgBox "Το κλειδί αυτό χρησιμοποιείται απο τον κάτοχο: " + CRLF + CRLF +
HorizontalTab + datSecondaryRS.Recordset.Fields(0).Value + " " +
datSecondaryRS.Recordset.Fields(1).Value + CRLF + CRLF + "Αφαιρέστε πρώτα τον κάτοχο και
μετά το κλειδί.", vbCritical, "Προσπάθεια αφαίρεσης χρησιμοποιούμενου κλειδιού"
    Else
        On Error Resume Next
        .Recordset.Delete
        On Error Resume Next
        .Recordset.Update
        .Refresh
    End If
    KeyPool.Redraw = True
    Call KMP_Refresh

End With

End Sub

Private Sub cmdUse_Click()

If SendKeyCode2KO = True Then
    KeyOwnerDetails.KO_iButtonCode.Text = KO_iButtonCode.Text
Else
    NewKeyOwner.KO_iButtonCode.Text = KO_iButtonCode.Text
End If
Unload Me

End Sub

Private Sub Form_Deactivate()

    Unload Me

End Sub

Private Sub Form_Load()

```

---

---

```
Call CXForm(Me)
ValidKeyCodeLength.FillColor = vbRed
KMP_Refresh
```

```
End Sub
```

```
Private Sub KeyPool_Click()
```

```
    KO_iButtonCode.Text = KeyPool.Text
```

```
End Sub
```

```
Private Sub KO_iButtonCode_Change()
```

```
    If Len(KO_iButtonCode.Text) < 16 Then ValidKeyCodeLength.FillColor = vbRed Else
ValidKeyCodeLength.FillColor = vbGreen
```

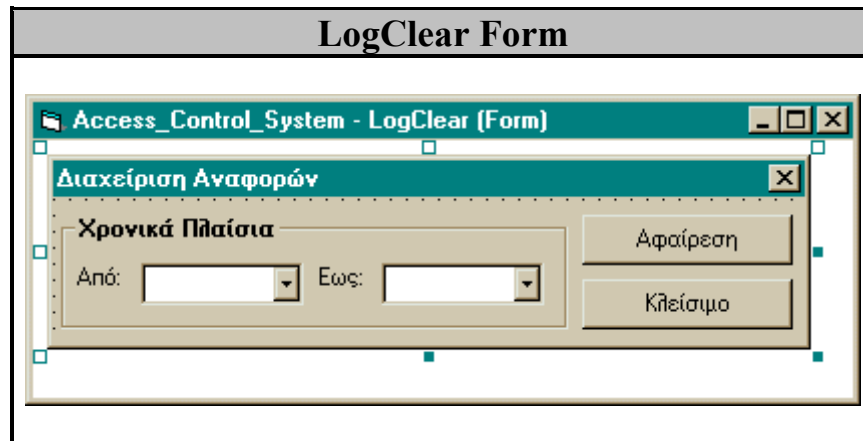
```
End Sub
```

```
Private Sub KO_iButtonCode_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase$(Chr$(KeyAscii)))
```

```
    If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9")) And Not (KeyAscii >= Asc("A") And
KeyAscii <= Asc("F")) And Not (KeyAscii = 8) Then KeyAscii = 0
```

```
End Sub
```



### Κώδικας Φόρμας LogClear

```
Private Sub CloseFrm_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub DeleteLogsForDates_Click()
```

```
    Dim i As Integer
```

```
    Dim j As Integer
```

```
    Screen.MousePointer = vbHourglass
```

```
    If From_Date.Text = "" Or Till_Date.Text = "" Then Exit Sub
```

```
    i = MsgBox("Η διαδικασία αυτή θα αφαιρέσει τις καταστάσεις για τις ημερομηνίες απο: " +  
CRLF + CRLF + HorizontalTab + HorizontalTab + From_Date.Text + " έως " + Till_Date.Text +  
CRLF + CRLF + "Είσαστε σίγουροι ότι θέλετε να συνεχίσετε με την αφαίρεση;", vbYesNo +  
vbCritical + vbQuestion, "Αφαίρεση καταστάσεων")
```

```
    If i = vbYes Then
```

```
        With datPrimaryRS
```

```
            .RecordSource = "select * from [Entry Log] where DateOfEntry between datevalue(" +  
From_Date.Text + ") and datevalue(" + Till_Date.Text + ") Order by DateOfEntry"
```

```
            .Refresh
```

```
            j = .Recordset.RecordCount
```

```
            For i = 0 To j - 1
```

```
                .Recordset.Delete
```

```
                .Refresh
```

```
            Next i
```

```
            Call Analyze_Log
```

```
            AccessLogs.datPrimaryRS.Refresh
```

```
            Call Refresh_AL_Grid
```

```
        End With
```

```
    End If
```

```
    Screen.MousePointer = vbDefault
```

```
End Sub
```

---

---

```
Private Sub Form_Deactivate()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Screen.MousePointer = vbHourglass
```

```
    On Error Resume Next
```

```
    datPrimaryRS.Refresh
```

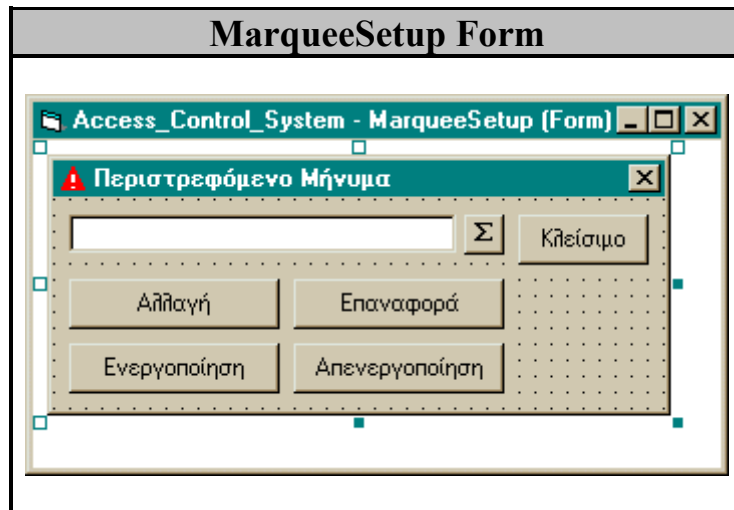
```
    Call Analyze_Log
```

```
    LastDateLogged = Format(Date, "ddmmyy")
```

```
    Call CXForm(Me)
```

```
    Screen.MousePointer = vbDefault
```

```
End Sub
```



### Κώδικας Φόρμας MarqueeSetup

Option Explicit

```
Private Sub ActivateMarquee_Click()
```

```
    ActivateMarquee.Enabled = False
    SendCommandToTerminal ("ATQ1")
    ActivateMarquee.Enabled = True
```

```
End Sub
```

```
Private Sub AsciiCharInsertion_Click()
```

```
Dim ascii_char As Byte
```

```
    On Error GoTo errhandler
    ascii_char = InputBox("Παρακαλώ δώστε τον αριθμό του ASCII χαρακτήρα που θέλετε να εισάγετε:", "Εισαγωγή ειδικού ASCII χαρακτήρα")
    NewMarquee.Text = NewMarquee.Text + Chr$(ascii_char)
```

```
errhandler:
```

```
End Sub
```

```
Private Sub cmdClose_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub DeactivateMarquee_Click()
```

```
    DeactivateMarquee.Enabled = False
    SendCommandToTerminal ("ATQ0")
```

```
DeactivateMarquee.Enabled = True

End Sub

Private Sub Form_Deactivate()

    Unload Me

End Sub

Private Sub Form_Load()

    Call CXForm(Me)

End Sub

Private Sub ResetMarquee_Click()

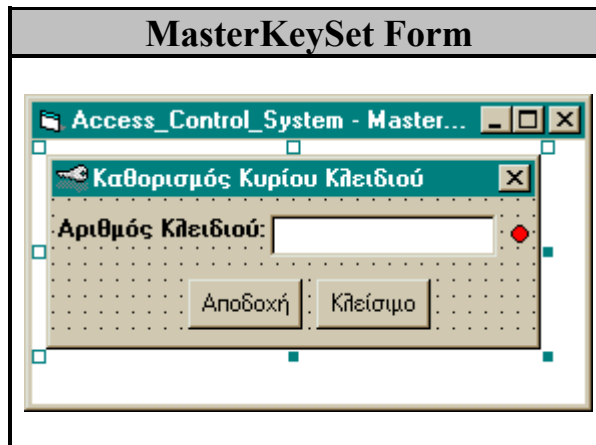
    ResetMarquee.Enabled = False
    SendCommandToTerminal ("ATQ2")
    ResetMarquee.Enabled = True

End Sub

Private Sub SetMarquee_Click()
Dim i As Byte
Dim MarqueeText As String

    SetMarquee.Enabled = False
    MarqueeText = NewMarquee.Text
    For i = Len(MarqueeText) To 27
        MarqueeText = " " + MarqueeText
    Next i
    SendCommandToTerminal ("ATQ3")
    SendToTerminal (ConvertTo437(MarqueeText))
    SendToTerminal (ComEOS)
    SetMarquee.Enabled = True

End Sub
```



### Κώδικας Φόρμας MasterKeySet

Option Explicit

Private Sub CmdAccept\_Click()

    If ValidKeyCodeLength.FillColor = vbGreen Then

        WaitForTerminalAcknowledge = True

        Call TimeOutPeriod(5)

        SendCommandToTerminal ("ATM" + KO\_iButtonCode.Text)

        While WaitForTerminalAcknowledge = True And TimedOut = False

            DoEvents

        Wend

        If TimedOut = True Then

            MsgBox "Το τερματικό δεν αποκρίθηκε εντός του χρονικού διαστήματος 5 δευτερολέπτων.  
Ελέγξτε τις συνδέσεις και ξαναπροσπαθήστε.", vbInformation

        Else

            MsgBox "Το νέο κύριο κλειδί αποθηκεύτηκε στο τερματικό.", vbInformation

            Call ServiceLogMessage("Εγινε ορισμός νέου κύριου κλειδιού.")

        End If

    Else

        MsgBox "Δεν έχει ορισθεί έγκυρο μήκος κλειδιού. Ελέγξτε την εισαγωγή σας στο αντίστοιχο πεδίο.", vbExclamation, "Σφάλμα εισαγωγής κωδικού κλειδιού"

    End If

End Sub

Private Sub cmdClose\_Click()

    Unload Me

End Sub

Private Sub Form\_Deactivate()

    Unload Me

End Sub

---

---

```
Private Sub Form_Load()
```

```
    ValidKeyCodeLength.FillColor = vbRed  
    Call CXForm(Me)
```

```
End Sub
```

```
Private Sub KO_iButtonCode_Change()
```

```
    If Len(KO_iButtonCode.Text) < 16 Then ValidKeyCodeLength.FillColor = vbRed Else  
    ValidKeyCodeLength.FillColor = vbGreen
```

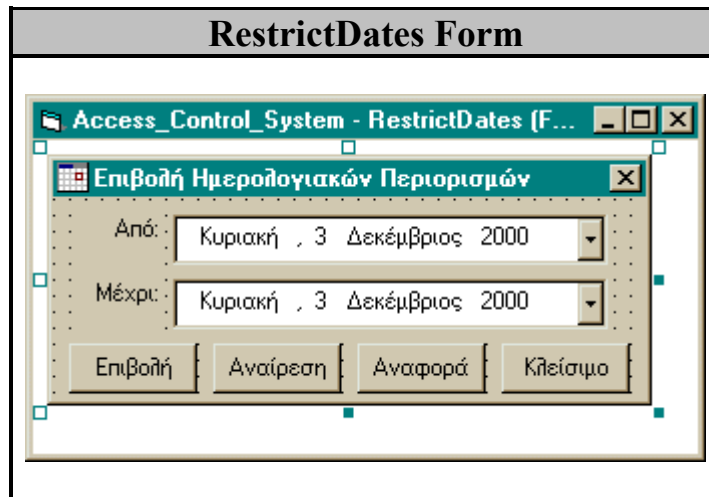
```
End Sub
```

```
Private Sub KO_iButtonCode_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase$(Chr$(KeyAscii)))  
    If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9")) And Not (KeyAscii >= Asc("A") And  
    KeyAscii <= Asc("F")) And Not (KeyAscii = 8) Then KeyAscii = 0
```

```
End Sub
```





### Κώδικας Φόρμας RestrictDates

Option Explicit

Private Sub ApplyRestrictions\_Click()

```

    With AccessLogs.datPrimaryRS
        .RecordSource = "select DateOfEntry,TimeOfEntry,NameOfKeyOwner,IOType,Lock from
[Entry Log] where DateOfEntry between datevalue("'" + Str(From_Date.Value) + "'") and
datevalue("'" + Str(Till_Date.Value) + "'") Order by DateOfEntry,TimeOfEntry"
        .Refresh
        If .Recordset.RecordCount = 0 Then
            Call CancelRestrictions_Click
        Else
            Call Refresh_AL_Grid
        End If
    End With

```

End Sub

Private Sub CancelRestrictions\_Click()

```

    With AccessLogs.datPrimaryRS
        .RecordSource = "select DateOfEntry,TimeOfEntry,NameOfKeyOwner,IOType,Lock from
[Entry Log] Order by DateOfEntry,TimeOfEntry"
        .Refresh
    End With
    Call Refresh_AL_Grid

```

End Sub

Private Sub cmdClose\_Click()

Unload Me

---

---

End Sub

Private Sub CreateReport\_Click()

    DataEnvironment1.commands("EntryLogsByDate").Parameters("StartingDate").Value =  
From\_Date.Value

    DataEnvironment1.commands("EntryLogsByDate").Parameters("EndingDate").Value =  
Till\_Date.Value

    On Error Resume Next

    DataEnvironment1.rsEntryLogsByDate.Close

    On Error Resume Next

    DataEnvironment1.rsEntryLogsByDate.Open

    On Error Resume Next

    DataEnvironment1.rsEntryLogsByDate.Requery

    If DataEnvironment1.rsEntryLogsByDate.RecordCount > 0 Then

        EntryLogsReport.Show

    Else

        MsgBox "Δεν υπάρχουν εγγραφές για το συγκεκριμένο χρονικό διάστημα.", vbExclamation,  
"Ακύρωση διαδικασίας"

    End If

End Sub

Private Sub Form\_Deactivate()

    Unload Me

End Sub

Private Sub Form\_Load()

    From\_Date.Value = Format\$(Format\$(Date\$, "mm/dd/yyyy"), "dd/mm/yyyy")

    Till\_Date.Value = Format\$(Format\$(Date\$, "mm/dd/yyyy"), "dd/mm/yyyy")

    Call CXForm(Me)

End Sub

---

---

## Αρχεία Ρουτινών (Modules)

Στα αρχεία που ακολουθούν περιέχονται ρουτίνες που χρησιμοποιούνται από το λογισμικό διαχείρισης του τερματικού.

### Ρουτίνες Αρχείου Routines.bas

Option Explicit  
'Δηλώσεις χρησιμοποιούμενων μεταβλητών και σταθερών

Public ComEOT As String \* 1  
Public ComVT As String \* 1  
Public ComEOS As String \* 1  
Public CRLF As String \* 2  
Public HorizontalTab As String \* 1

Public isHidden As Boolean

Public InSerDataBuffer As String  
Public InSerDataBuffer2 As String  
Public ReceivedCommand As String  
Public EncryptionPassword As String  
Public AdministratorPassword As String

Public DoorLocked As Boolean  
Public DoorIsOpen As Boolean  
Public TimedOut As Boolean  
Public FirstTimeCalled As Boolean  
Public TxInProgress As Boolean  
Public TerminalFound As Boolean  
Public SendKeyCode2KO As Boolean  
Public RunningAsService As Boolean  
Public CIPMessageIsDisplayed As Boolean  
Public AllowMessageEditor\_Change As Boolean  
Public WaitForTerminalAcknowledge As Boolean  
Public NewKeyOwner As New KeyOwnerDetails

Public Sub SetConstants()  
'Δήλωση σταθερών  
ComEOT = Chr(&H4)  
ComVT = Chr(&HB)  
ComEOS = Chr(&H0)

AdministratorPassword = "dooradmin"  
EncryptionPassword = "mprodoorlock"  
HorizontalTab = Chr(&H9)  
CRLF = Chr(&HD) + Chr(&HA)

End Sub

```
Public Sub SetRTCTime()
```

```
'Ρουτίνα ρύθμισης των καταχωρητών ώρας και ημερομηνίας του Real Time Clock του τερματικού.  
Static i As Byte
```

```
SendCommandToTerminal ("ATR")
```

```
SendToTerminal (Format$(Time$, "hhmmss"))
```

```
SendToTerminal (Format$(Date$, "yyyy"))
```

```
SendToTerminal (Format$(Format$(Date$, "mm-dd-yyyy"), "mm"))
```

```
SendToTerminal (Format$(Format$(Date$, "mm-dd-yyyy"), "dd"))
```

```
For i = 1 To 7
```

```
    If Format$(Format$(Date$, "mm-dd-yyyy"), "dddd") = Format$(i, "dddd") Then  
SendToTerminal (Format$(i, "00"))
```

```
Next i
```

```
End Sub
```

```
Public Sub EntryRequest()
```

```
' Ρουτίνα επεξεργασίας αίτησης πρόσβασης στον ελεγχόμενο χώρο.
```

```
Dim X As Integer
```

```
Dim Y As Integer
```

```
Screen.MousePointer = vbHourglass
```

```
If InStr(1, ReceivedCommand, "REQ", vbTextCompare) <> 0 Then
```

```
    ReceivedCommand = Right$(ReceivedCommand, 22)
```

```
    With MainWindow.datSecondaryRS
```

```
        .RecordSource = "select  
Surname,FirstName,ActiveKey,iKeyNumber,PassCode,ValidKeyUntil from [Key Owners] where  
iKeyNumber=" + Left$(ReceivedCommand, 16) + " and PassCode=" +  
Crypt(Right$(ReceivedCommand, 6), EncryptionPassword) + " and ActiveKey=True and  
datevalue(ValidKeyUntil) >= datevalue(" + Format$(Format$(Date$, "mm/dd/yyyy"),  
"dd/mm/yyyy") + ")"
```

```
        .Refresh
```

```
    End With
```

```
    If MainWindow.datSecondaryRS.Recordset.RecordCount = 1 Then
```

```
        'Ο κάτοχος του κλειδιού διαθέτει δικαίωμα εισόδου
```

```
        Beep
```

```
        Call SendCommandToTerminal("ATU") 'Αποστολή εντολής ξεκλειδώματος κλειδαριάς
```

```
        With AccessLogs
```

```
            With .datPrimaryRS
```

```
                With .Recordset
```

```
                    'Προσθήκη εγγραφής αίτησης εισόδου στην βάση δεδομένων
```

```
                    If .RecordCount > 0 Then .MoveLast
```

```
                    .AddNew
```

```
                    With .Fields
```

```
                        .Item(0).Value = Format$(Format$(Date$, "mm/dd/yyyy"), "dd/mm/yyyy")
```

```
                        .Item(1).Value = Format$(Time$, "hh:mm:ss")
```

```
                        .Item(2).Value = MainWindow.datSecondaryRS.Recordset.Fields.Item(0).Value +
```

```
                        " " + MainWindow.datSecondaryRS.Recordset.Fields.Item(1).Value
```

```

        .Item(3).Value = "Είσοδος"
        .Item(4).Value = ""
    End With
    .Update
    End With
    .Refresh
    End With
    If isHidden = True Then
        Unload AccessLogs
    Else
        Y = AccessLogs.Top
        X = AccessLogs.Left
        If .datPrimaryRS.Recordset.RecordCount > 1 Then
            .MSHFlexGrid1.Clear
            .datPrimaryRS.Refresh
            .MSHFlexGrid1.Refresh
            Call Refresh_AL_Grid
        Else
            Unload AccessLogs
            AccessLogs.Top = Y
            AccessLogs.Left = X
            Call ShowForm(AccessLogs)
        End If
    End If
    End With

Else
    'Ο κάτοχος του κλειδιού δεν έχει δικαίωμα εισόδου.
    Call SendCommandToTerminal("ATE")
End If

End If
Screen.MousePointer = vbDefault

End Sub

Public Sub ExitRequest()
    'Ρουτίνα επεξεργασίας αίτησης εξόδου απο τον ελεγχόμενο χώρο.

    Screen.MousePointer = vbHourglass
    If InStr(1, ReceivedCommand, "EXRQ-", vbTextCompare) <> 0 Then
        ReceivedCommand = Right$(ReceivedCommand, 16)

        With MainWindow.datSecondaryRS
            .RecordSource = "select
Surname,FirstName,ActiveKey,iKeyNumber,PassCode,ValidKeyUntil from [Key Owners] where
iKeyNumber=" + Left$(ReceivedCommand, 16) + " and ActiveKey=True and
datevalue(ValidKeyUntil) >= datevalue(" + Format$(Format$(Date$, "mm/dd/yyyy"),
"dd/mm/yyyy") + ")"
            .Refresh
        End With
    End With

```

```

If MainWindow.datSecondaryRS.Recordset.RecordCount = 1 Then
    Beep
    'Ο κάτοχος κλειδιού έχει δικαίωμα εξόδου.
    Call SendCommandToTerminal("ATU") 'Εντολή ξεκλειδώματος κλειδαριάς
    With AccessLogs
        With .datPrimaryRS
            With .Recordset
                On Error Resume Next
                .MoveLast
                .AddNew
                'Προσθήκη αίτησης εξόδου στην βάση δεδομένων
                With .Fields
                    .Item(0).Value = Format$(Format$(Date$, "mm/dd/yyyy"), "dd/mm/yyyy")
                    .Item(1).Value = Format(Time$, "hh:mm:ss")
                    .Item(2).Value = MainWindow.datSecondaryRS.Recordset.Fields.Item(0).Value +
" " + MainWindow.datSecondaryRS.Recordset.Fields.Item(1).Value
                    .Item(3).Value = "Εξοδος"
                    .Item(4).Value = ""
                End With
                .Update
            End With
            .Refresh
        End With
        .MSHFlexGrid1.Clear
        .datPrimaryRS.Refresh
        .MSHFlexGrid1.Refresh
    End With
    Call Refresh_AL_Grid
    If isHidden = True Then Unload AccessLogs
End If

End If
Screen.MousePointer = vbDefault

End Sub

Public Sub LockRequest()
'Ρουτίνα επεξεργασίας αίτησης ενεργοποίησης κλειδαριάς

Screen.MousePointer = vbHourglass
If InStr(1, ReceivedCommand, "LRQ-", vbTextCompare) <> 0 Then
    ReceivedCommand = Right$(ReceivedCommand, 16)

    With MainWindow.datSecondaryRS
        .RecordSource = "select
Surname,FirstName,ActiveKey,iKeyNumber,PassCode,ValidKeyUntil from [Key Owners] where
iKeyNumber=" + Left$(ReceivedCommand, 16) + " and ActiveKey=True and
datevalue(ValidKeyUntil) >= datevalue(" + Format$(Format$(Date$, "mm/dd/yyyy"),
"dd/mm/yyyy") + ")"
        .Refresh
    End With

```

```

If MainWindow.datSecondaryRS.Recordset.RecordCount = 1 Then
    'Ο κάτοχος έχει δικαίωμα πρόσβασης
    Beep
    Call SendCommandToTerminal("ATL") 'Εντολή ενεργοποίησης κλειδαριάς
    With AccessLogs
        With .datPrimaryRS
            With .Recordset
                .MoveLast
                'Προσθήκη αίτησης κλειδώματος στην βάση δεδομένων
                .AddNew
                With .Fields
                    .Item(0).Value = Format$(Format$(Date$, "mm/dd/yyyy"), "dd/mm/yyyy")
                    .Item(1).Value = Format(Time$, "hh:mm:ss")
                    .Item(2).Value = MainWindow.datSecondaryRS.Recordset.Fields.Item(0).Value +
" " + MainWindow.datSecondaryRS.Recordset.Fields.Item(1).Value
                    .Item(3).Value = "Εξοδος"
                    .Item(4).Value = "Κλείδωμα"
                End With
                .Update
            End With
            .Refresh
        End With
        .MSHFlexGrid1.Clear
        .datPrimaryRS.Refresh
        .MSHFlexGrid1.Refresh
    End With
    Call Refresh_AL_Grid
End If
If isHidden = True Then Unload AccessLogs
End If

Screen.MousePointer = vbDefault

End Sub

Public Sub DiagnosticsDecode()
'Ρουτίνα αποκωδικοποίησης στοιχείων διαγνωστικού ελέγχου.

Dim i As Integer

    WaitForTerminalAcknowledge = False
    Screen.MousePointer = vbHourglass
    If InStr(1, ReceivedCommand, "DIAG", vbTextCompare) <> 0 Then
        ReceivedCommand = Right$(ReceivedCommand, 4)
        With DiagCheckResults
            For i = 1 To 4
                If Mid(ReceivedCommand, i, 1) = "1" Then .Shape(i - 1).FillColor = vbGreen Else
                .Shape(i - 1).FillColor = vbRed
            Next i
        End With
        WaitForTerminalAcknowledge = False
        Call ShowForm(DiagCheckResults)
    End If

```

```
End With
End If
Screen.MousePointer = vbDefault
```

```
End Sub
```

```
Public Sub CXForm(X As Form)
```

'Ρουτίνα τοποθέτησης της φόρμας στο κέντρο της οθόνης.

```
On Error Resume Next
X.Top = (Screen.Height - X.Height) / 3
On Error Resume Next
X.Left = (Screen.Width - X.Width) / 2
```

```
End Sub
```

```
Function Crypt(iString As String, Pass$) As String
```

'Ρουτίνα κρυπτογράφησης.

```
Dim Strg As String
Dim H As String
Dim j As String
Dim i As Integer
```

```
If Trim$(iString) = "" Then Exit Function
Strg$ = iString: Call Crypt_(Strg$, Pass$)
H$ = ""
```

```
For i = 1 To Len(Strg$)
    j$ = Hex$(Asc(Mid$(Strg$, i, 1)))
    If Len(j$) = 1 Then j$ = "0" + j$
    H$ = H$ + j$
```

```
Next i
Crypt = H$
```

```
End Function
```

```
Function DeCrypt(iString As String, Pass As String) As String
```

'Ρουτίνα αποκρυπτογράφησης.

```
Dim Strg As String
Dim H As String
Dim j As String
Dim i As Integer
```

```
If Trim$(iString) = "" Then Exit Function
H$ = iString
Strg$ = ""
```

```
For i = 1 To Len(H$) Step 2
    j$ = Mid$(H$, i, 2)
    Strg$ = Strg$ + Chr$(Val("&h" + j$))
```

```
Next i
Call Crypt_(Strg$, Pass)
DeCrypt = Strg$
```



---

---

End Function

Public Sub Crypt\_(Strg\$, Pass\$)

Dim a As Integer

Dim b As Integer

Dim i As Integer

    a = 1

    For i = 1 To Len(Strg\$)

        b = Asc(Mid\$(Pass\$, a, 1))

        a = a + 1

        If a > Len(Pass\$) Then a = 1

        Mid\$(Strg\$, i, 1) = Chr\$(Asc(Mid\$(Strg\$, i, 1)) Xor b)

    Next i

End Sub

Public Sub SendCommandToTerminal(DataPacket As String)

'Ρουτίνα αποστολής εντολής στο τερματικό.

Dim i As Integer

    For i = 1 To Len(DataPacket)

        Call Delay\_msecs(1)

        MainWindow.MSComm1.Output = Mid(DataPacket, i, 1)

    Next i

End Sub

Public Sub SendToTerminal(DataPacket As String)

'Ρουτίνα αποστολής δεδομένων στο τερματικό.

Dim i As Integer

    For i = 1 To Len(DataPacket)

        MainWindow.MSComm1.Output = Mid(DataPacket, i, 1)

    Next i

End Sub

Public Sub FindTerminalComPort()

'Ρουτίνα εύρεσης θύρας σύνδεσης του τερματικού.

Dim i As Integer

    With MainWindow

        For i = 1 To 4

            Screen.MousePointer = vbHourglass

            If .MSComm1.PortOpen = True Then

                On Error Resume Next

                .MSComm1.PortOpen = False

            End If

```

.MSComm1.CommPort = Val(i)
If .MSComm1.PortOpen = False Then
    On Error Resume Next
    .MSComm1.PortOpen = True
End If
Call TimeOutPeriod(4)
TerminalFound = False
Call SendCommandToTerminal("ATC")
While TerminalFound = False And TimedOut = False
    DoEvents
Wend
Screen.MousePointer = vbDefault
If TerminalFound = True Then Exit Sub
Next i
End With

```

End Sub

```
Public Sub ShowForm(X As Form)
```

'Ρουτίνα εμφάνισης της φόρμας.

```

Screen.MousePointer = vbHourglass
X.Show
On Error Resume Next
X.SetFocus
Screen.MousePointer = vbDefault

```

End Sub

```
Public Sub AddKeyToKeyMediaPool()
```

'Ρουτίνα προσθήκης του εισαχθέντος κλειδιού στην βάση δεδομένων διαθέσιμων κλειδιών.

```
Dim NewKeyCode As String * 16
```

```

Load KeyPoolSelect
With KeyPoolSelect
    .KeyPool.Redraw = False
    NewKeyCode = Right$(ReceivedCommand, 16)
    .datPrimaryRS.RecordSource = "select iKeyCode from [KeyMediaPool] where iKeyCode=" +
NewKeyCode + ""
    .datPrimaryRS.Refresh
    If .datPrimaryRS.Recordset.RecordCount = 0 Then
        .datPrimaryRS.RecordSource = "select iKeyCode from [KeyMediaPool]"
        .datPrimaryRS.Refresh
        .datPrimaryRS.Recordset.MoveLast
        .datPrimaryRS.Recordset.AddNew
        .datPrimaryRS.Recordset.Fields("iKeyCode").Value = NewKeyCode
        On Error Resume Next
        .datPrimaryRS.Recordset.Update
    End If
    .datPrimaryRS.RecordSource = "select iKeyCode from [KeyMediaPool]"

```

```
.datPrimaryRS.Refresh
.KeyPool.Redraw = True
Call KMP_Refresh
If isHidden = True Then Unload KeyPoolSelect
End With
```

End Sub

```
Public Sub ServiceLogError(strMessage As String)
```

'Ρουτίνα καταγραφής σφάλματος στην υπηρεσία Γεγονότων (Event Viewer).

```
Call MainWindow.NTService1.LogEvent(svcEventError, svcMessageError, strMessage)
```

End Sub

```
Public Sub ServiceLogMessage(strMessage As String)
```

'Ρουτίνα καταγραφής μηνύματος πληροφόρησης στην υπηρεσία Γεγονότων (Event Viewer).

```
Call MainWindow.NTService1.LogEvent(svcEventInformation, svcMessageInfo, strMessage)
```

End Sub

```
Public Sub Delay_msecs(X As Integer)
```

'Ρουτίνα δημιουργίας χρονικής καθυστέρησης x miliseconds.

```
MainWindow.DelayTimer.Interval = X
```

```
MainWindow.DelayTimer.Enabled = True
```

```
Do While MainWindow.DelayTimer.Enabled = True
```

```
DoEvents
```

```
Loop
```

End Sub

```
Public Sub TimeOutPeriod(X As Integer)
```

'Ρουτίνα δημιουργίας χρονικού παραθύρου εκτέλεσης γεγονότων.

```
With MainWindow.TimeOutTimer
```

```
TimedOut = False
```

```
.Enabled = False
```

```
.Interval = 1000 * X
```

```
.Enabled = True
```

```
End With
```

End Sub

```
Public Sub ID_Request()
```

'Ρουτίνα επεξεργασίας αίτησης αποστολής στοιχείων του κατόχου κλειδιού ασφαλείας απο το τερματικό.

```
Screen.MousePointer = vbHourglass
```

```
With MainWindow.datSecondaryRS
```

```

.RecordSource = "select Surname,FirstName,ActiveKey,iKeyNumber from [Key Owners]
where iKeyNumber=" + Right$(ReceivedCommand, 16) + ""
.Refresh

If .Recordset.RecordCount = 1 Then
    'Αποστολή στοιχείων κατόχου κλειδιού ασφαλείας.
    SendCommandToTerminal ("ATB")
    SendToTerminal (ConvertTo437(Left$(.Recordset.Fields.Item(0).Value + " " +
.Recordset.Fields.Item(1).Value), 20)))
    SendToTerminal (ComEOS)
    SendToTerminal (ComEOT)
Else
    'Εμφανίστηκε σφάλμα κατα την αναζήτηση του κατόχου.
    SendCommandToTerminal ("ATE")
End If

End With

Screen.MousePointer = vbDefault
End Sub

Public Function ConvertTo437(text928 As String) As String
'Ρουτίνα μετατροπής απο την κωδικοσελίδα 928 των Windows στην κωδικοσελίδα 437 που
χειρίζεται το τερματικό.

Dim temp As String
Dim i As Byte

For i = 1 To Len(text928)
    temp = temp + Chr$(Ansi_to_437(Asc(Mid(text928, i, 1))))
Next i
ConvertTo437 = temp
End Function

```

## Ρουτίνες Αρχείου Routines2.bas

Option Explicit

'Δήλωση σταθερών τιμών.

Public Const MARGIN\_SIZE = 60 ' in Twips

Public Const MARGIN\_CORRECTION = 100 ' in Twips

Public Sub Refresh\_KO\_Grid()

'Ρουτίνα επανασχεδιασμού αντικειμένου MSHFlexGrid στην φόρμα AllKeyOwners.

Dim i As Integer

Dim j As Integer

Dim color As Integer

Dim RowTemp As Integer

Screen.MousePointer = vbHourglass

With AllKeyOwners.MSHFlexGrid1

.Refresh

RowTemp = .Row

.Redraw = False

.Row = 0

' set grid's column widths and captions

.Col = 0

.ColWidth(0) = 1800

.Text = "Επώνυμο"

.Col = 1

.Text = "Όνομα"

.ColWidth(1) = 1500

.Col = 2

.Text = "Ιδιότητα"

.ColWidth(2) = 2000

.Col = 3

.Text = "Ενεργό Κλειδί"

.ColWidth(3) = 1500

.Col = 4

.Text = "Αριθμός Κλειδιού"

.ColWidth(4) = 2000

.Col = 5

.Text = "Λήξης Εγκυρότητας"

.ColWidth(5) = 2500

' set grid's style

.AllowBigSelection = True

.FillStyle = flexFillRepeat

' make header bold

.Row = 0

.Col = 0

.RowSel = 0

.ColSel = .Cols - 1

.CellFontBold = True

```

' mark invalid key owners with red color
For i = 1 To .Rows - 1
    .Row = i
    .Col = 3
    If .Text = "False" Then color = vbRed Else color = vbBlack
    For j = 0 To .Cols - 1
        .Col = j
        .CellForeColor = color
    Next j
Next i

' user detailed text instead of boolean expression
For i = 1 To .Rows - 1
    .Col = 3
    .Row = i
    If .Text = "True" Then .Text = "Ενεργό"
    If .Text = "False" Then .Text = "Μη ενεργό"
Next i

' convert date field to long date text
For i = 1 To .Rows - 1
    .Col = 5
    .Row = i
    .Text = Format$(Format$(.Text, "mm/dd/yyyy"), "dddd, dd mmmm yyyy")
    .CellAlignment = flexAlignRightCenter
Next i

' align columns 3 to 5 to center
.Col = 3
.Row = 0
.RowSel = .Rows - 1
.ColSel = .Cols - 2
.CellAlignment = flexAlignCenterCenter
.Col = 5
.Row = 0
.CellAlignment = flexAlignCenterCenter

.Row = RowTemp
Call KO_SelectFullLine

.FillStyle = flexFillSingle
.AllowBigSelection = True
.Redraw = True
End With
Screen.MousePointer = vbDefault

End Sub

Public Sub Refresh_AL_Grid()
'Ρουτίνα επανασχεδιασμού αντικειμένου MSHFlexGrid στην φόρμα AccessLogs.

```

---

---

Dim i As Integer

Screen.MousePointer = vbHourglass

With AccessLogs.MSHFlexGrid1

.Redraw = False

.Row = 0

' set grid's column widths

.Col = 0

.ColWidth(0) = 2700

.Text = "Ημερομηνία"

.Col = 1

.Text = "Ωρα"

.ColWidth(1) = 1000

.Col = 2

.Text = "Όνοματεπώνυμο"

.ColWidth(2) = 3000

.Col = 3

.Text = "Τύπος Πρόσβασης"

.ColWidth(3) = 1700

.Col = 4

.Text = "Κατάσταση"

.ColWidth(4) = 1200

' set grid's style

.AllowBigSelection = True

.FillStyle = flexFillRepeat

' make header bold

.Row = 0

.Col = 0

.RowSel = 0

.ColSel = .Cols - 1

.CellFontBold = True

For i = 1 To .Rows - 1

.Col = 0

.Row = i

.Text = Format\$(Format\$(.Text, "mm/dd/yyyy"), "dddd, dd MMMM yyyy")

Next i

For i = 1 To .Rows - 1

.Col = 1

.Row = i

.Text = Format\$(.Text, "hh:mm:ss")

Next i

' align column 1 to center

.Col = 1

.Row = 0

.RowSel = .Rows - 1

.ColSel = 1

.CellAlignment = flexAlignCenterCenter

```

' align columns 3 to 4 to center
.Col = 3
.Row = 0
.RowSel = .Rows - 1
.ColSel = .Cols - 1
.CellAlignment = flexAlignCenterCenter

.AllowBigSelection = True
.FillStyle = flexFillSingle
.Redraw = True
On Error GoTo errhandler
.Row = 1
.Col = 0
On Error Resume Next
.TopRow = .Rows - 1
End With

errhandler:
    Screen.MousePointer = vbDefault
    Exit Sub

End Sub

Public Sub KO_SelectFullLine()
'Ρουτίνα επιλογής ολόκληρης γραμμής στο αντικείμενο MSFlexGrid της φόρμας AllKeyOwners.

With AllKeyOwners
    If .MSFlexGrid1.Row = 0 Then .MSFlexGrid1.Row = 1
        .MSFlexGrid1.Col = 0
        .MSFlexGrid1.ColSel = .MSFlexGrid1.Cols - 1
End With

End Sub

Public Sub Analyze_Log()
'Ρουτίνα ανάλυσης των εγγραφών της βάσης δεδομένων Entry Log για την εύρεση των
ημερομηνιών που υπήρξαν καταγραφές αιτήσεων πρόσβασης.

Dim i As Integer
Dim LastDay As String

Screen.MousePointer = vbHourglass
With LogClear.datPrimaryRS
    .RecordSource = "select * from [Entry Log] order by DateOfEntry,TimeOfEntry"
    .Refresh
    If .Recordset.RecordCount > 0 Then .Recordset.MoveFirst

    LastDay = ""
    LogClear.From_Date.Clear

```



```

LogClear.Till_Date.Clear
If .Recordset.RecordCount = 0 Then Exit Sub

For i = 1 To .Recordset.RecordCount

    If i = 1 Then LogClear.From_Date.Text = .Recordset.Fields.Item(1).Value

    If .Recordset.Fields.Item(1).Value <> LastDay Then
        LastDay = .Recordset.Fields.Item(1).Value
        LogClear.From_Date.AddItem LastDay
        LogClear.Till_Date.AddItem LastDay

    End If

    .Recordset.MoveNext
Next i
LogClear.Till_Date.Text = LogClear.From_Date.Text
On Error Resume Next
.Recordset.MoveFirst
End With
Screen.MousePointer = vbDefault

End Sub

Public Sub AccessLogBanding()
'Ρουτίνα δημιουργίας ζωνών ομαδοποίησης στοιχείων του αντικειμένου MSHFlexGrid της φόρμας
AccessLogs.

Dim i As Integer

With AccessLogs.MSHFlexGrid1
.Redraw = False
' set grid's column merging and sorting
For i = 0 To .Cols - 1
.MergeCol(i) = True
Next i

.Sort = flexSortGenericAscending

.AllowBigSelection = False
.FillStyle = flexFillSingle
.Redraw = True
End With

End Sub

Public Sub AllCoursesGrid_Refresh()
'Ρουτίνα επανασχεδιασμού αντικειμένου MSHFlexGrid στην φόρμα AllCourses.

Dim i As Integer

With AllCourses.AllCoursesGrid

```

---

---

```
.Row = 0
.ColWidth(0) = 2700
.ColWidth(1) = 1700
.ColWidth(2) = 1000

.Col = 2
For i = 1 To .Rows - 1
    .Row = i
    .CellAlignment = flexAlignCenterCenter
    If .Text = "True" Then
        .Text = "Ενεργή"
    Else
        .Text = "Ανενεργή"
    End If
Next i

End With
End Sub
```

## Ρουτίνες Αρχείου Routines3.bas

Option Explicit

'Δήλωση Πίνακα αντιστοίχισης & μετατροπής κωδικοσελίδων 928 σε 437.

Public Ansi\_to\_437(0 To 255) As Byte

Public Sub LoadConversionTable()

'Ρουτίνα εισαγωγής δεδομένων στον πίνακα αντιστοίχισης & μετατροπής κωδικοσελίδων απο αρχείο στον σκληρό δίσκο.

Dim i As Integer

Dim temp As String

Screen.MousePointer = vbHourglass

On Error GoTo errhandle

Open "c:\acs\928-437.xlt" For Input As #1

For i = 0 To 255

Line Input #1, temp

Ansi\_to\_437(i) = Val(Right\$(temp, Len(temp) - InStr(1, temp, "=")))

Next i

Close #1

Screen.MousePointer = vbDefault

Exit Sub

errhandle:

Screen.MousePointer = vbDefault

MsgBox "Υπήρξε πρόβλημα πρόσβασης στο αρχείο: c:\acs\928-437.xlt" + CRLF + "Παρακαλώ επαναφέρατε το αρχικό αρχείο.", vbCritical, "Πρόβλημα πρόσβασης σε αρχείο του προγράμματος"

Exit Sub

End Sub

Public Sub EditAnnouncement()

'Ρουτίνα σύνδεσης κειμένου ανακοίνωσης απο την βάση δεδομένων σε αντικείμενο TextBox.

Dim i As Integer

Screen.MousePointer = vbHourglass

With AnnouncementEditor.datPrimaryRS

Announcements. AnnouncementsList.Col = 1

.Recordset.AbsolutePosition = Announcements. AnnouncementsList.Row + 1

AnnouncementEditor.MessageEditor.Text = ""

For i = 1 To 8

On Error Resume Next

AnnouncementEditor.MessageEditor.Text = AnnouncementEditor.MessageEditor.Text +

.Recordset.Fields.Item(i + 1).Value

Next i

End With

Screen.MousePointer = vbDefault

End Sub

Public Sub SaveAnnouncementChanges()

---

---

Ρουτίνα διάσπασης κειμένου ανακοίνωσης σε πεδία της βάσης δεδομένων.

Dim i As Integer

Dim j As Integer

Screen.MousePointer = vbHourglass

With AnnouncementEditor

.datPrimaryRS.RecordSource = "select \* from [Announcements] where AnnouncementTitle=""  
+ AnnouncementEditor.Announcement\_Title.Text + """

j = 1

For i = 1 To Len(.MessageEditor.Text) Step 255

On Error Resume Next

.datPrimaryRS.Recordset.Fields.Item(j + 1).Value = Mid(.MessageEditor.Text, i, 255)

j = j + 1

Next i

For i = j - 1 To 8

.datPrimaryRS.Recordset.Fields.Item(j + 1).Value = ""

Next i

.datPrimaryRS.Recordset.Update

.datPrimaryRS.Refresh

End With

Announcements.datPrimaryRS.Refresh

Unload AnnouncementEditor

Screen.MousePointer = vbDefault

End Sub

Public Sub CreateFormattedMessagePreview()

*'Δημιουργία προεπισκόπησης μορφής της ανακοίνωσης όπως θα εμφανίζεται στο τερματικό.*

Dim i As Integer

Dim X As Integer

Dim Y As Integer

Dim inc As Integer

Dim dec As Integer

Dim temp As String

Dim Columns As Integer

With AnnouncementEditor

.AnnouncementLines.Clear

With .MessageEditor

For i = 1 To Len(.Text)

inc = 0

dec = 0

If Len(.Text) >= 28 Then

If Len(.Text) - i >= 28 Then

Columns = 28

Else

Columns = Len(.Text) - i

```

    If Columns = 0 Then Columns = 1
End If

If Mid(.Text, i + Columns, 1) <> " " Or Mid(.Text, i + Columns, 2) <> CRLF Then

    dec = 0
    inc = 0
    Y = i + Columns
    For X = i To i + Columns
        If Mid(.Text, X, 1) = " " Then
            inc = 0
            dec = 0
            Y = X
        ElseIf Mid(.Text, X, 1) = Chr$(13) Then
            inc = 2
            dec = -1
            Y = X
        Exit For
    End If
Next X

    If X = i + Columns + 1 And X = Y + 1 Then
        X = Y
        If Columns < 28 Then X = X + 1
        inc = 0
        If Mid(.Text, X, 1) = " " Then dec = 0 Else dec = -1
    Else
        X = Y
    End If

Else
    X = i + Columns + 1
End If

Else
    X = Len(.Text) + 1
End If

temp = Mid(.Text, i, X - i)
AnnouncementEditor.AnnouncementLines.AddItem temp

i = X + inc + dec
If i >= Len(.Text) Then Exit For

Next i
End With
End With
End Sub

Public Sub SendAnnouncement()
'Ρουτίνα αποστολής του κειμένου της ανακοίνωσης που ζητήθηκε για επισκόπηση στο τερματικό.

```

```

Dim i As Integer
Dim j As Integer
Dim temp As String
Dim AnnouncementLine As String

Screen.MousePointer = vbHourglass
Announcements.AnnouncementsList.Col = 1
Announcements.AnnouncementsList.Row = Val(Right$(ReceivedCommand, 2)) - 1
Call EditAnnouncement

With AnnouncementEditor.AnnouncementLines

    If .ListCount < 1 Or Trim$(AnnouncementEditor.MessageEditor.Text) = "" Then
        Call SendCommandToTerminal("ATE")
        Exit Sub
    End If

    WaitForTerminalAcknowledge = True
    Call TimeOutPeriod(4)
    Call SendCommandToTerminal("ATX")
    While WaitForTerminalAcknowledge = True And TimedOut = False
        DoEvents
    Wend
    If TimedOut = True Then Exit Sub
    temp = MainWindow.StatusBar.SimpleText
    MainWindow.StatusBar.SimpleText = "Αποστολή Ανακοίνωσης..."

    For i = 0 To .ListCount - 1

        .ListIndex = i
        AnnouncementLine = .Text

        Call SendToTerminal(ConvertTo437(AnnouncementLine))

        If Len(AnnouncementLine) < 28 Then
            For j = 1 To 28 - Len(AnnouncementLine)
                Call SendToTerminal(" ")
            Next j
        End If

        Call SendToTerminal(ComeEOS)
        '8 Σελίδες των 8 γραμμών για κάθε ανακοίνωση
        If .ListIndex = 64 Then Exit For
    Next i

    If (.ListIndex Mod 8) <> 0 Then
        For i = 1 To 8 - (.ListIndex Mod 8) - 1
            For j = 1 To 28
                Call SendToTerminal(" ")
            Next j
        End For
    End If

```

```

        Call SendToTerminal(ComeEOS)
    Next i
End If
Call SendToTerminal(ComeEOT)
If isHidden = True Then Unload AnnouncementEditor
MainWindow.StatusBar.SimpleText = temp
End With
Screen.MousePointer = vbDefault
End Sub

Public Sub SendAnnouncementsList()
'Αποστολή των τίτλων των ανακοινώσεων στο τερματικό.

Dim j As Integer
Dim i As Integer
Dim temp As String
Dim TitleLine As String

temp = MainWindow.StatusBar.SimpleText
MainWindow.StatusBar.SimpleText = "Αποστολή Λίστας Ανακοινώσεων..."
With Announcements.AnnouncementsList
    WaitForTerminalAcknowledge = True
    Call TimeOutPeriod(20)
    Call SendCommandToTerminal("ATT")
    While WaitForTerminalAcknowledge = True And TimedOut = False
        DoEvents
    Wend
    If TimedOut = True Then Exit Sub
    For i = 0 To 7
        .Col = 1
        .Row = i
        TitleLine = ""
        TitleLine = ConvertTo437(.Text)
        If Len(TitleLine) < 25 Then
            For j = 1 To 25 - Len(TitleLine)
                TitleLine = TitleLine + " "
            Next j
        End If
        Call SendToTerminal(TitleLine)
        Call SendToTerminal(ComeEOS)
    Next i
    Call SendToTerminal(ComeEOT)
End With
If isHidden = True Then Unload Announcements
MainWindow.StatusBar.SimpleText = temp

End Sub

Public Sub KMP_Refresh()
'Ρουτίνα επανασχεδιασμού του αντικειμένου MSHFlexGrid της φόρμας KeyPoolSelect και
μορφοποίηση των περιεχομένων του.

```

```

Dim color As Integer
Dim i As Integer

With KeyPoolSelect.KeyPool
    KeyPoolSelect.datPrimaryRS.RecordSource = "select iKeyCode from [KeyMediaPool]"
    KeyPoolSelect.datPrimaryRS.Refresh

    For i = 1 To KeyPoolSelect.datPrimaryRS.Recordset.RecordCount
        .Row = i - 1
        KeyPoolSelect.datSecondaryRS.RecordSource = "Select * from [Key Owners] where
iKeyNumber=" + .Text + ""
        KeyPoolSelect.datSecondaryRS.Refresh
        If KeyPoolSelect.datSecondaryRS.Recordset.RecordCount = 1 Then
            color = vbRed
        Else
            color = vbBlack
        End If
        .CellForeColor = color
    Next i

    .Col = 0
    .Row = 1
    .ColWidth(0) = 1600
End With
End Sub

Public Sub SendCourseMarks()
'Ρουτίνα αποστολής βαθμολογικού πίνακα για το μάθημα που ζητήθηκε στο τερματικό.

Dim i As Integer
Dim j As Integer
Dim r As Integer
Dim NoOfRecords As Integer
Dim DBCourseMarks As Object
Dim DBFileName As String
Dim CourseTitle As String
Dim temp As String
Dim surname As String
Dim firstname As String
Dim Mark As String
Dim CourseMarkLine As String

Screen.MousePointer = vbHourglass
AllCourses.AllCoursesGrid.Row = Val(Right$(ReceivedCommand, 2))
AllCourses.AllCoursesGrid.Col = 0
CourseTitle = AllCourses.AllCoursesGrid.Text
AllCourses.AllCoursesGrid.Col = 1
DBFileName = AllCourses.AllCoursesGrid.Text
If DBFileName = "" Then GoTo ReportError
AllCourses.AllCoursesGrid.Col = 2
If AllCourses.AllCoursesGrid.Text = "Ανενεργή" Then GoTo ReportError

```



```

Set DBCourseMarks = AllCourses.CourseMarks
On Error Resume Next
DBCourseMarks.RecordSource = "Select Επώνυμο,Όνομα,Βαθμός from [Βαθμολογίες] order by
Επώνυμο,Όνομα"
On Error Resume Next
DBCourseMarks.Refresh
On Error Resume Next
DBCourseMarks.Recordset.Close
On Error GoTo ReportError
DBCourseMarks.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
DBFileName + ";Mode=ReadWrite;Persist Security Info=False"
DBCourseMarks.RecordSource = "Select Επώνυμο,Όνομα,Βαθμός from [Βαθμολογίες] order by
Επώνυμο,Όνομα"
DBCourseMarks.Refresh
On Error Resume Next
DBCourseMarks.Recordset.Open
DBCourseMarks.Recordset.MoveFirst

WaitForTerminalAcknowledge = True
Call TimeOutPeriod(10)
SendCommandToTerminal ("ATZ")
While WaitForTerminalAcknowledge = True And TimedOut = False
    DoEvents
Wend
If TimedOut = True Then Exit Sub
temp = MainWindow.StatusBar.SimpleText
MainWindow.StatusBar.SimpleText = "Αποστολή Βαθμολογιών [" + CourseTitle + "]"

NoOfRecords = DBCourseMarks.Recordset.RecordCount
If NoOfRecords > 248 Then NoOfRecords = 248

For r = 1 To NoOfRecords

    surname = Trim$(DBCourseMarks.Recordset.Fields(0).Value)

    If surname <> "" Then

        firstname = Trim$(DBCourseMarks.Recordset.Fields(1).Value)
        If DBCourseMarks.Recordset.Fields(2).Value = 0 Then
            Mark = "0.0"
        ElseIf DBCourseMarks.Recordset.Fields(2).Value = 10 Then
            Mark = "10."
        Else
            Mark = Str$(Format(DBCourseMarks.Recordset.Fields(2).Value, "#0.00"))
        End If
        If InStr(1, Trim$(Mark), ".") = 0 Then Mark = Trim$(Mark) + ".0"
        If InStr(1, Trim$(Mark), ".") = 1 Then Mark = "0" + Trim$(Mark)

        CourseMarkLine = Left$(surname + " " + firstname, 24)
        If Len(CourseMarkLine) <= 24 Then
            For i = 1 To 28 - Len(Mark) - Len(CourseMarkLine)
                CourseMarkLine = CourseMarkLine + " "
            Next i
        End If
    End If
End For

```

```

    Next i
End If
CourseMarkLine = CourseMarkLine + Mark

Call SendToTerminal(ConvertTo437(CourseMarkLine))

SendToTerminal (ComeEOS)
MainWindow.StatusBar.SimpleText = "Αποστολή Βαθμολογιών [" + CourseTitle + "]" + ... +
Str$(DBCourseMarks.Recordset.AbsolutePosition * 100) \
DBCourseMarks.Recordset.RecordCount) + "% "
End If

DBCourseMarks.Recordset.MoveNext
Next r

If (NoOfRecords Mod 8 <> 0) Then
    For i = 1 To (8 - NoOfRecords Mod 8)
        For j = 1 To 28
            SendToTerminal (" ")
        Next j
        SendToTerminal (ComeEOS)
    Next i
End If

SendCommandToTerminal (ComeEOT)
MainWindow.StatusBar.SimpleText = temp
Screen.MousePointer = vbDefault

Exit Sub

ReportError:
SendCommandToTerminal ("ATE")
Screen.MousePointer = vbDefault
End Sub

Public Sub SendCourseTitles()
'Ρουτίνα αποστολής των τίτλων των μαθημάτων που δημοσιεύονται απο το τερματικό.

Dim NoOfCourses As Integer
Dim CourseTitle As String
Dim i As Byte
Dim j As Byte
Dim k As Byte

Screen.MousePointer = vbHourglass
NoOfCourses = AllCourses.datPrimaryRS.Recordset.RecordCount
If NoOfCourses > 8 Then NoOfCourses = 8
SendCommandToTerminal ("ATY")
SendToTerminal (Chr(NoOfCourses))

For i = 1 To NoOfCourses

```

```

AllCourses.AllCoursesGrid.Row = i
AllCourses.AllCoursesGrid.Col = 0
CourseTitle = AllCourses.AllCoursesGrid.Text
If Len(Trim$(CourseTitle)) = 0 Then CourseTitle = "--Αχρησιμοποίητος χώρος--"
k = Len(CourseTitle)
For j = k To 27
    CourseTitle = CourseTitle + " "
Next j
SendToTerminal (ConvertTo437(CourseTitle))
SendToTerminal (ComEOS)
Next i
SendToTerminal (ComEOT)
Screen.MousePointer = vbDefault

End Sub

```

### Φόρμες Αναφοράς

Με την φόρμα αναφοράς γίνεται δυνατή η εκτύπωση των εγγραφών αιτήσεων πρόσβασης που έχουν γίνει αποδεκτές από το λογισμικό διαχείρισης του τερματικού, σύμφωνα με την ημερομηνία και ώρα εκτέλεσης τους.

**EntryLogsReport Designers Form**

The screenshot shows a Microsoft Access report designer interface. The report is titled "Ημερολογιακές Καταστάσεις" (Daily Statuses). It features a page header with the same title, a group header for "DateOfEntry", a detail section with columns for "TimeOfEntry", "NameOfKeyOwner [EntryLogs]", and "IOType Lock", and a page footer showing "Σελίδα: %p/%P". The report is displayed in a window titled "Access\_Control\_System - EntryLogsReport (DataReport)".

---

## Περιβάλλον Διαχείρισης Δεδομένων

Με το περιβάλλον Διαχείρισης Δεδομένων DataEnvironment1 γίνεται η σύνδεση της βάσης δεδομένων Entry Logs με την φόρμα δημιουργίας Αναφοράς. Χρησιμοποιείται η γλώσσα SQL με δήλωση παραμέτρων για τα χρονικά πλαίσια εμφάνισης των εγγραφών της βάσης δεδομένων.

