



Τ.Ε.Ι Κρήτης



Τμήμα Ηλεκτρονικής



Πτυχιακή Εργασία:

***Ανάπτυξη μονάδας αλληλεπίδρασης
με σύστημα διαχείρισης
ενέργειας σε κτίρια.***

**Των φοιτητών:
Κουτρουλή Χριστόδουλου
Φακουκάκη Μιχαήλ**

Εισηγήτρια & επιβλέπουσα καθηγήτρια:

Κολοκοτσά Διονυσία

ΠΕΡΙΕΧΟΜΕΝΑ

1. Περίληψη	3
2. Αναφορά στη χρηστικότητα των GUI	4
2.1 Εισαγωγή.....	3
2.2 Ιστορικά	
2.2.1 Ιστορική αναφορά για GUI.....	4,5,6
2.2.2 Μετάβαση από γραμμή εντολών σε GUI.....	6,7,8
2.3 Ειδικευμένα GUI.....	9,10
2.4 GUI σε συνεργασία με αυτοματισμούς.....	10,11
2.5 Συμπεράσματα.....	11
3. Τα έξυπνα κτήρια	12
3.1 Ορισμός.....	12,13
3.2 Έξυπνες συσκευές.....	13 - 15
3.3 Διασφάλιση του μέλλοντος των έξυπνων κτηρίων και επένδυση σε αυτά μέσω της εκπαίδευσης.....	15 - 18
3.4 Περιγραφή του πειραματικού θαλάμου.....	19
3.4.1 Γενικά.....	19
3.4.2 Η τεχνολογία που χρησιμοποιείται στο θάλαμο.....	19
3.4.2.1 Η Τεχνική EIB.....	19,20
3.4.2.2 Το EIS & οι EIB Λειτουργίες.....	20,21
3.4.2.3 Το ψηφιακό Τηλεγράφημα.....	21 - 27
3.4.2.4 Η φυσική δομή ενός EIB συστήματος.....	28,29
3.4.2.5 Το EIB Tool Software (ETS).....	29 - 33
3.4.3 Αναλυτική ματιά στον πειραματικό θάλαμο.....	33 - 37
3.4.4 Αναφορά στο λογισμικό παρακολούθησης και ελέγχου του πειραματικού θαλάμου.....	37,38
4. 4 Ανάπτυξη του λογισμικού παρακολούθησης και ελέγχου του πειραματικού θαλάμου	38
4.1 Ανάπτυξη οποιουδήποτε Gui με τη Matlab (Εργαλεία).....	38 – 49
4.2 Τοποθέτηση αντικειμένων στο κύριο Gui του θαλάμου (booth.fig) και συνοπτική λειτουργία τους.....	48-51
4.3 Το gui actuat.fig.....	51,52
4.4 Το gui exit_conf.fig.....	52
4.5 Τα αρχεία Booth.m, actuat.m & exit_conf.m και η λογική ανάθεσης εργασιών για κάθε στοιχείο.....	53,54
4.6 Ροή διεργασιών και ανάλυση κώδικα του Booth.m	54 – 68
4.7 Περιγραφή των Gui actuat.m και exit_conf.m.....	69 – 71
5. Συμπεράσματα	71,72

1. Περίληψη.

Τα G.U.I (Graphical User Interface ή αλλιώς Γραφικό Περιβάλλον Αλληλεπίδρασης) χρησιμοποιούνται κατά κόρον εδώ και μερικά χρόνια σε όλες τις μορφές λογισμικού με σκοπό να καταστήσουν ένα πρόγραμμα ευκολότερα κατανοητό και εύχρηστο για το διαχειριστή του. Το κείμενο που ακολουθεί έχει στόχο να εξηγήσει τη λειτουργία και την κατασκευή ενός γραφικού περιβάλλοντος, το οποίο δημιουργήθηκε ως πτυχιακή εργασία στο τμήμα Ηλεκτρονικής του Τ.Ε.Ι. Χανίων. Το περιβάλλον αυτό έχει το ρόλο του διαμεσολαβητή μεταξύ ενός αυτοματοποιημένου θαλάμου και του ανθρώπου-διαχειριστή. Πιο συγκεκριμένα, υπάρχει στο χώρο του Τ.Ε.Ι. ένας θάλαμος εξοπλισμένος με αισθητήρες (φωτισμού, θερμοκρασίας κτλ.) και συσκευές που ελέγχονται αυτόματα από έναν επεξεργαστή με βάση το Ευρωπαϊκό σύστημα εγκαταστάσεων EIBUS, ο οποίος με απλά λόγια αποτελεί ένα μικρό μοντέλο αυτού που ονομάζεται «έξυπνο σπίτι». Ο κύριος ρόλος του GUI είναι η λήψη τιμών από τους αισθητήρες του θαλάμου ανά τακτά χρονικά διαστήματα και η απεικόνισή τους σε γραφικές παραστάσεις (σε πραγματικό χρόνο) συναρτημένη του χρόνου. Συμπληρωματικός ρόλος του είναι η πληροφόρηση, με γραπτό κείμενο, για την κατάσταση των αυτοματισμών του θαλάμου όπως είναι το μηχανικό παράθυρο ή το κλιματιστικό. Επιπλέον, το GUI είναι υπεύθυνο να εκκινήσει το θάλαμο κάνοντας τις απαραίτητες αρχικοποιήσεις και να τον θέσει σε κατάσταση λειτουργίας που αποτελείται από κύκλους εργασιών (λήψη τιμών-λήψη αποφάσεων) οι οποίοι καθορίζονται από κώδικα ασαφούς λογικής (fuzzy logic) ενώ δίνεται η δυνατότητα να καθοριστούν μελλοντικά από διαφορετικά είδη ελέγχου, τους PID και ON/OFF.

1. Abstract

GUIs (Graphical User Interface) are used predominantly for several years in all forms of software in order to make a program more easily understood and easy to use for the manager. The following text aims to explain the operation and construction of a graphic environment, which was created as a graduation project in the part of TEI Electronics Chania. The environment is the role of mediator between an automated booth and the manager. In particular, there is a booth at TEI equipped with sensors (light, temperature, etc.) and devices that are automatically controlled by a processor based on the European system installations EIBUS, which in simple terms is a small model of what is called «smart building». The main role of the GUI is to receive values from the sensors of the chamber at regular intervals and display them in graphics (real time) versus time. Complementary role is information, in writing, on the status of automation of the chamber such as “mechanical window” or the air conditioner. In addition, the GUI is responsible to “start” the booth by doing all the necessary initialisations and set him in running condition, consisting of turnovers (value-taking decision-making), which are determined by a code of fuzzy logic and given the opportunity to fix prospectively from different types of control, PID and the ON / OFF.

2 Αναφορά στη χρηστικότητα των GUI.



2.1 Εισαγωγή.

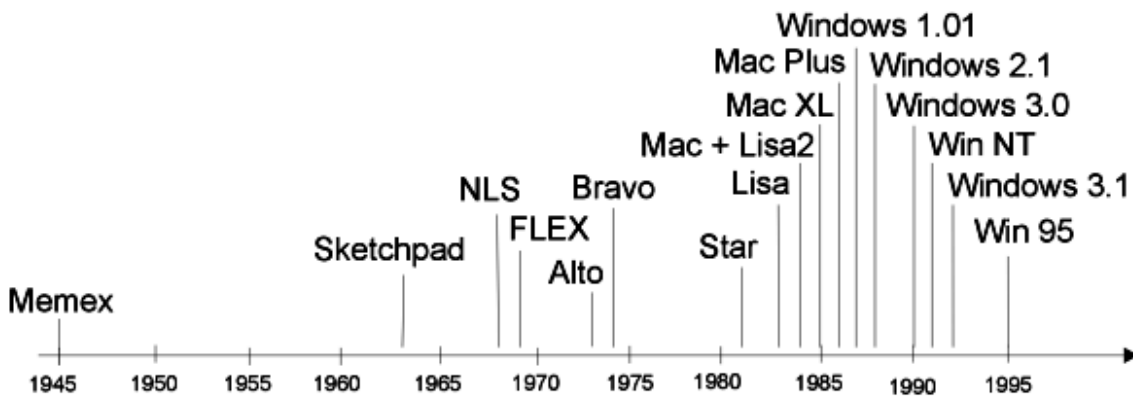
Τα GUIs, αρχικά των αγγλικών λέξεων Graphical User Interface (γραφικό περιβάλλον αλληλεπίδρασης με το χρήστη), δημιουργήθηκαν από την ανάγκη του προγραμματιστή να κάνει ένα πρόγραμμα πιο κατανοητό για τους χρήστες. Καθώς κάποιος διαβάζει αυτές τις γραμμές (εφόσον τις διαβάζει στην ηλεκτρονική τους μορφή και όχι στη έντυπη), κοιτάζει ταυτόχρονα και το GUI ενός συγκεκριμένου προγράμματος. Ο λόγος της ύπαρξης του όρου αυτού είναι ότι οι πρώτες αλληλεπιδρούσες διασυνδέσεις χρήστη με τους υπολογιστές δεν ήταν με γραφικά. Ήταν προσανατολισμένες μόνο γύρω από το κείμενο και το πληκτρολόγιο (text-and-keyboard oriented) και συνήθως αποτελούνταν από εντολές που ο χρήστης έπρεπε να θυμάται και αποκρίσεις του υπολογιστή οι οποίες ήταν «εξευτελιστικά» σύντομες. Η διασύνδεση με εντολές του λειτουργικού συστήματος MS-DOS (την οποία ακόμα μπορεί κάποιος να δει και να χρησιμοποιήσει από το λειτουργικό σύστημα Windows) είναι ένα κλασικό παράδειγμα της τυπικής διασύνδεσης χρήστη - υπολογιστή πριν την άφιξη του GUI. Ένα ενδιάμεσο βήμα στις διασυνδέσεις χρήστη, μεταξύ της διασύνδεσης με εντολές και του GUI, ήταν η διασύνδεση βασισμένη σε μενού αλλά χωρίς γραφικά (non-graphical menu-based interface), η οποία επέτρεπε στο χρήστη να αλληλεπιδρά χρησιμοποιώντας το ποντίκι, αντί να τον υποχρεώνει να πληκτρολογεί εντολές.

Το πιο διαδεδομένο λειτουργικό σύστημα στις μέρες μας παρέχει ένα καλό GUI. Οι διάφορες εφαρμογές τυπικά χρησιμοποιούν τα στοιχεία του GUI που παρέχονται από το λειτουργικό σύστημα και παράλληλα προσθέτουν τα δικά τους GUI στοιχεία και ιδέες. Ένα GUI μερικές φορές χρησιμοποιεί μία ή περισσότερες «μεταφορές» αντικειμένων με τα οποία ο χρήστης είναι εξοικειωμένος στην καθημερινή του ζωή, όπως είναι το desktop (μία επιτραπέζια επιφάνεια) και η θέα από ένα παράθυρο. Στοιχεία ενός GUI περιλαμβάνουν οντότητες όπως : τα παράθυρα, pull-down μενού, τα κουμπιά, η κυλιόμενη μπάρα (scroll bar), τα εικονίδια, οι βοηθοί / εγχειρίδια (wizards), τον κέρσορα του ποντικιού και αναμφίβολα πολλά άλλα που δεν έχουν εφευρεθεί ακόμα.

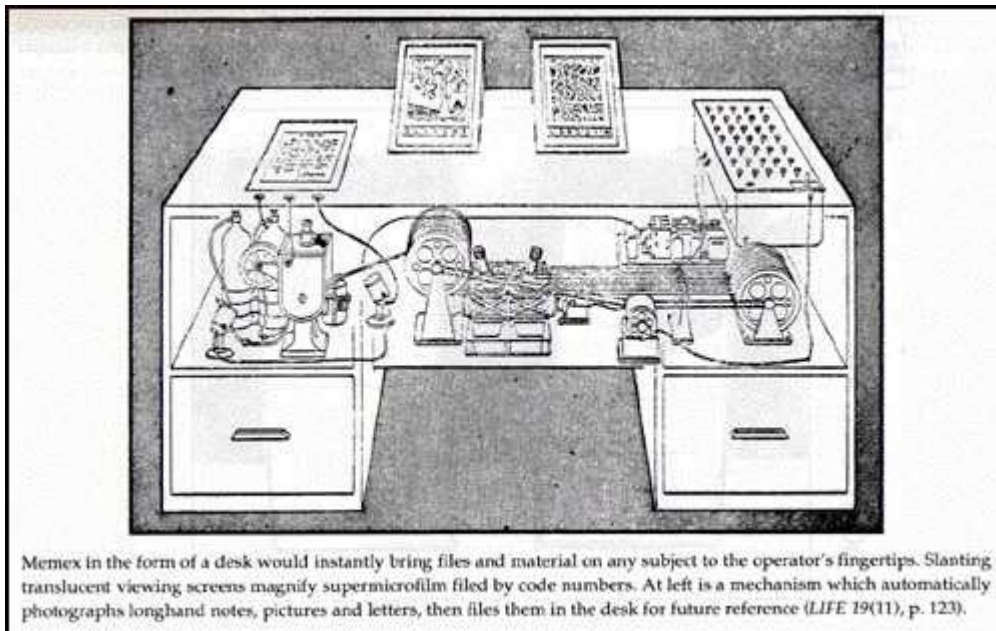
2.2 Ιστορικά.

2.2.1 Ιστορική αναφορά για GUI.

Η εξελικτική πορεία των γραφικών λειτουργικών συστημάτων, σύμφωνα με το διαδικτυακό τόπο <http://www2.iicm.edu/cguetl/education/projects/mischitz/Seminar.htm>, είναι μεγάλης διάρκειας και εντυπωσιακή. Παρακάτω φαίνεται μια χρονική γραμμή της εξέλιξης των GUI με τα ορόσημα του σχεδιασμού των γραφικών λειτουργικών συστημάτων.



Όλα ξεκίνησαν με ένα άρθρο από τον Dr. Vannevar Bush με όνομα “As we may think” (μτφ. Όπως εμείς ίσως σκεφτούμε) το 1945 στην εφημερίδα Atlantic Monthly (μτφ. Μηνιαία Ατλαντικού). Μέσα στο άρθρο υπέδειξε στους στρατιωτικούς επιστήμονες ένα διαφορετικό ρόλο αφού ο δεύτερος παγκόσμιος πόλεμος είχε τελειώσει. Το ρόλο του να κάνουν το απόθεμα των γνώσεων μας πιο προσιτό στον καθένα. Ο Bush συνειδητοποίησε ότι το πλήθος των δημοσιεύσεων είχε εκταθεί πολύ πιο πέρα από την ικανότητα των ανθρώπων να χρησιμοποιούν αποτελεσματικά τα αρχεία. Σύμφωνα με τον Bush το κύριο πρόβλημα ήταν να αποθηκεύσουμε τις πληροφορίες με τέτοιο τρόπο ώστε να έχουμε εύκολη πρόσβαση σε αυτές αργότερα. Σαν λύση σε αυτό το πρόβλημα σχεδίασε μια συσκευή που ονόμασε Διαστολέα Μνήμης (Memory Expander ή Memex) και εξήγησε ότι αυτό ήταν μια συσκευή μέσα στην οποία κάποιος θα αποθήκευε τα βιβλία και αρχεία του και θα μπορούσε να την συμβουλευτεί με υπερβολική ταχύτητα και ευελιξία. Κάτι σαν ένα μεγεθυμένο και οικείο συμπλήρωμα της μνήμης του ανθρώπου. Η πρώτη ιδέα μιας τέτοιας συσκευής του ήρθε στα μέσα του 1930 πολύ πριν την ανάπτυξη του ψηφιακού υπολογισμού. Σχεδίασε το Memex σαν ένα γραφείο με ημιδιαφανείς οθόνες, μοχλούς και κινητήρες για γρήγορη αναζήτηση των μικροφίλμ αρχείων του. Επίσης σκέφτηκε να βάλει στο περιθώριο σημειώσεις και σχόλια κατά την ανάγνωση ενός αρχείου κάτι όμοιο με τα σημερινά σχόλια πάνω στην οθόνη. Το απαραίτητο χαρακτηριστικό του όμως εκτός από τις ικανότητες του για ανάκτηση και σχολιασμό ήταν η ικανότητα του να τοποθετεί συνδετικούς δείκτες, δηλαδή με τη χρήση σημερινής ορολογίας να τοποθετεί συνδέσμους (links).



Σχέδιο του Bush για το Memex

Η πρώτη πρακτική προσπάθεια σχεδιασμού γραφικών σε υπολογιστή αποδίδεται στον Ivan Sutherland. Στις αρχές του 1960, μια εποχή που οι περισσότεροι υπολογιστές ήταν διάτρητων καρτών εκτός από μια εξαίρεση, τον TX-2 υπολογιστή στο εργαστήριο του MIT's Lincoln ο οποίος αποτέλεσε κλειδί για την ανάπτυξη της αλληλεπίδραση στους υπολογιστές με γραφικά. Ο Sutherland κοιτώντας την οθόνη καθοδικού σωλήνα και το στυλό με φως στην κονσόλα του TX-2 σκέφτηκε ότι θα ήταν χρήσιμο κάποιος να μπορούσε να ζωγραφίσει με το στυλό-φακό, που υπήρχε ήδη στο σύστημα, πάνω στην οθόνη και αποφάσισε να το υλοποιήσει σαν θέμα της διδακτορικής διατριβής του. Έτσι γεννήθηκε το Sketchpad μια συσκευή για γραφική αλληλεπίδραση με τον υπολογιστή.

TX-2 System	
TX-2 Hardware	
Assembled	1959 Lincoln Laboratory MIT
Disassembled	1979 Lincoln Laboratory MIT
Footprints	~1000 square feet
Main Memory	68K words, 38-bit words ~=300kbytes Magnetic Core memory about a cubic yard in size
Speed	memory cycle of 6usec = 1/12 MIP
Backup Storage	7 inch, 1024 x 1024, point plotting capability only
I/O	Light Pen input @ 1 point; 16 words of switches = 592; 4 digital knobs microphone; speakers
TX-2 Software	
Operating System	None, users rolled their own
Compiler	None, except a Macro Assembler



Πίνακας χαρακτηριστικών του TX-2

Το Sketchpad στην οθόνη του TX-2

Το επόμενο σημαντικό ορόσημο στην πορεία της εξέλιξης των GUI, αποτέλεσε το NLS (On-Line System) του Douglas Engelbart. Τα περισσότερα χαρακτηριστικά του NLS χρησιμοποιούνται από υπολογιστικά συστήματα των ημερών μας. Ο Engelbart εφηύρε τα πολυμέσα στον υπολογιστή

περιλαμβάνοντας μέσα στο αρχείο τη διευθυνσιοδότηση και τη συνδεσμοδότηση. Παρέχονταν πολλαπλά παράθυρα με ελαστικό έλεγχο της θέασης και με βίντεο τηλεσκόπηση από την οθόνη του συστήματος. Επιπλέον η συσκευή εισόδου δεν ήταν ένα στυλό-φακό, όπως στο Sketchpad, αλλά μια συσκευή που ο Engelbart ονόμασε «ποντίκι».



Το NLS με την οθόνη, το πληκτρολόγιο, τα επιπλέον πλήκτρα του και το ποντίκι.

Η λειτουργικότητα εφεύρεση του Engelbart ενέπνευσε την εταιρία Xerox να δημιουργήσει το δικό της σύστημα [Xerox PARC](#) στις αρχές της δεκαετίας 1970. Η Xerox συστηματοποίησε τη χρήση των παραθύρων, εικονιδίων, μενού και δεικτών διαμορφώνοντας το περιβάλλον της σε μια ακόμη ομοιότερη της σημερινής, μορφή.

Ομοίως, οι επόμενες προσπάθειες κυρίως από τις γνωστές εταιρίες Macintosh και Microsoft, στόχευσαν στη βελτίωση των υπολογιστικών συστημάτων κάνοντας τις δικές τους προσθήκες-βελτιώσεις οι οποίες όμως χάριν συντομίας δεν θα αναφερθούν σε αυτό το έγγραφο.

2.2.2 Η μετάβαση από γραμμή εντολών σε gui.

Σύμφωνα με την ιστοσελίδα <http://www.catb.org/~esr/writings/taouu/html/ch02s02.html>, τα πρώτα λειτουργικά συστήματα ήταν σε περιβάλλον αλληλεπίδρασης με κείμενο. Αυτά λειτουργούσαν με τη διεξαγωγή μιας σειράς αιτημάτων και αποκρίσεων, με τα αιτήματα και τις αποκρίσεις εκφρασμένα σαν εντολές κειμένου από ένα εξειδικευμένο λεξιλόγιο. Η εισαγωγή του κειμένου στο περιβάλλον γινόταν (και εξακολουθεί να γίνεται) από μια συσκευή εισόδου με περίπου 100 πλήκτρα όμοια της γραφομηχανής που ονομάζεται πληκτρολόγιο. Το περιβάλλον αλληλεπίδρασης που βασιζόταν στο γραπτό κείμενο ονομάστηκε Περιβάλλον Αλληλεπίδρασης Με Γραμμές Εντολών (Command Line Interface, CLI).

Το DOS της Microsoft είναι από τα πιο δημοφιλή και επικρατέστερα λειτουργικά συστήματα εντολών με μακρά πορεία συνεχών αναβαθμίσεων και προσθηκών. Ακολουθεί φωτογραφία από αυτό.

```

Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/P] [/W] [/A[:attribs]] [/O[:sortord]]
  [/S] [/B] [/L] [/C[H]]

[drive:][path][filename] Specifies drive, directory, and/or files to list.
/P      Pauses after each screenful of information.
/W      Uses wide list format.
/A      Displays files with specified attributes.
attribs  D Directories      R Read-only files      H Hidden files
         S System files    A Files ready to archive - Prefix meaning "not"
/O      List by files in sorted order.
sortord  N By name (alphabetic)      S By size (smallest first)
         E By extension (alphabetic)  D By date & time (earliest first)
         G Group directories first  - Prefix to reverse order
         C By compression ratio (smallest first)
/S      Displays files in specified directory and all subdirectories.
/B      Uses bare format (no heading information or summary).
/L      Uses lowercase.
/CIH   Displays file compression ratio; /CH uses host allocation unit size.

Switches may be preset in the DIRCMD environment variable. Override
preset switches by prefixing any switch with - (hyphen)--for example, /-W.

C:\>_

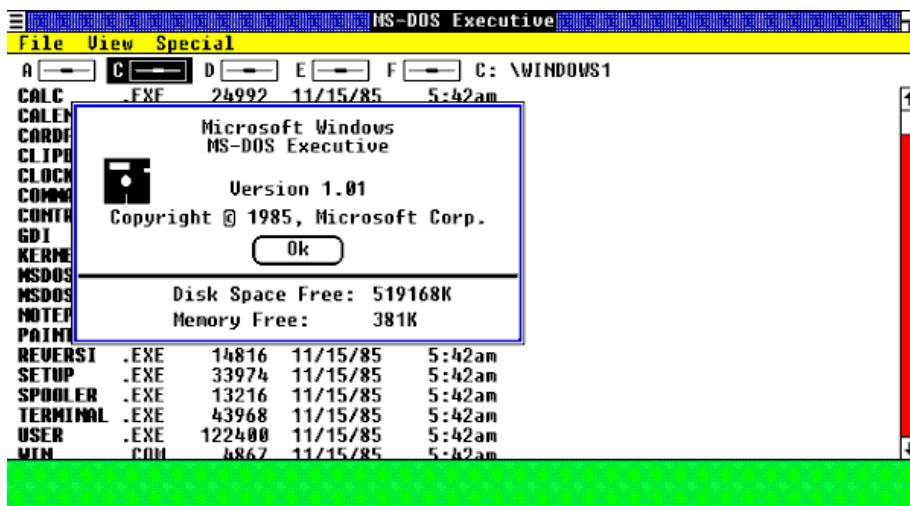
```

MS DOS

Σαν καινοτομία της τότε εποχής είχε θετικά χαρακτηριστικά που έκαναν ως ένα βαθμό τη χρήση του, φιλική προς τους χρήστες. Ο χρόνος απόκρισης ήταν μακράν χαμηλότερος από αυτόν που είχαν οι προγενέστεροι υπολογιστές πέφτοντας από μέρες ή ώρες σε δευτερόλεπτα. Επίσης έδινε στο χρήστη τη δυνατότητα να αλλάξει γνώμη σχετικά με κάποια μετέπειτα στάδια της εκάστοτε διεργασίας, μια δυνατότητα που πρωτοεμφανίστηκε με την άφιξη του. Το λογισμικό πλέον μπορούσε να γίνει εξερευνητικό και πολύ περισσότερο δημιουργικό.

Με τον καιρό όμως, οι ανάγκες για υπολογιστική ισχύ γίνονταν ολοένα και μεγαλύτερες και η πρόοδος στον τομέα των ηλεκτρονικών υπολογιστών κάλυπτε αυτές τις ανάγκες ώσπου δημιουργήθηκε μια νέα. Η ευχρηστία των CLIs χανόταν αφού ενώ η «καρδιά» του συστήματος (επεξεργαστής) ήταν έτοιμη να αντιμετωπίσει μεγαλύτερες και πολυπλοκότερες προκλήσεις, ο μέσος χρήστης δεν μπορούσε να ακολουθήσει. Το πρόβλημα πλέον ήταν η παραγωγικότητα του μέσου χρήστη, ο οποίος περιοριζόταν από τους κανόνες που έθετε το περιβάλλον αλληλεπίδρασης. Πιο συγκεκριμένα, η χρήση των H/Y μέσω γραμμής εντολών ήταν μια χρονοβόρα διαδικασία λόγω του γεγονότος ότι κάποιος έπρεπε να συντάσσει φράσεις και χρειαζόταν αρκετές γνώσεις για να καταφέρει κάποιος να δημιουργήσει ένα πρόγραμμα ή ακόμη να εκτελέσει απλές στη λειτουργία τους εντολές (π.χ. αντιγραφή αρχείων).

Αυτό το πρόβλημα σε συνδυασμό με την διαθεσιμότητα μεγαλύτερης υπολογιστικής ισχύος, ήταν η αφορμή για την εταιρία κατασκευής του DOS για να προχωρήσει στη δημιουργία ενός πολύ πιο εύκολου προς τη χρήση και γρηγορότερου στις εκτελέσεις εντολών λειτουργικού, του παραθυρικού πλέον MS Windows. Το νέο αυτό λειτουργικό βασιζόταν σε ένα γραφικό περιβάλλον στο οποίο ο χρήστης μπορούσε να κάνει τις περισσότερες λειτουργίες του προγενέστερου απλά με τη χρήση ενός ποντικιού.



Microsoft Windows 1.0 (1985)

Σύμφωνα με το διαδικτυακό τόπο http://en.wikipedia.org/wiki/History_of_the_graphical_user_interface, η Microsoft με αυτή την κίνηση, κατάφερε να υπέρ-απλουστεύσει τις λειτουργίες που χρειαζόταν ο χρήστης να εκτελέσει και κατά συνέπεια να «φέρει» ένα H/Y σχεδόν σε κάθε σπίτι. Οι πρώτες εκδόσεις του νέου αυτού λειτουργικού συνυπήρχαν με το παλαιότερο (MS DOS) μέχρι το 1995 όπου και κυκλοφόρησε το πρώτο πλήρως παραθυρικό λειτουργικό, τα MS Windows 95.

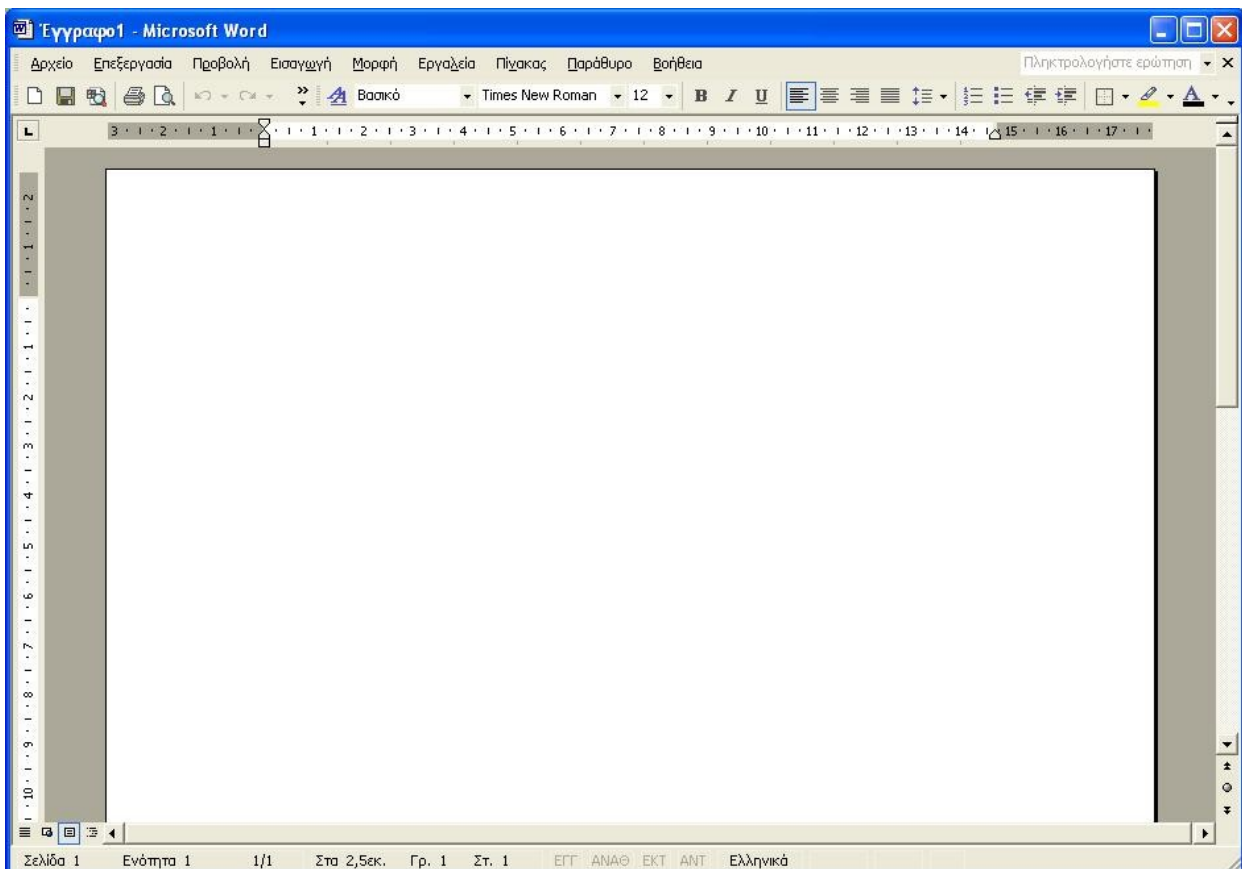
Τα Gui κατόρθωσαν να εκτοπίσουν από το προσκήνιο τα Cli λόγω των σαφών πλεονεκτημάτων που τα συνοδεύουν. Ένα μεγάλο πλεονέκτημα των Gui είναι ότι κάνουν τη λειτουργία του υπολογιστή πιο διαισθητική και έτσι ευκολότερο να μαθευτεί. Για παράδειγμα είναι ευκολότερο για ένα νέο χρήστη να μετακινήσει ένα αρχείο από ένα κατάλογο σε ένα άλλο με το να «τραβήξει» το εικονίδιο του αρχείου με το ποντίκι, από το να πρέπει να θυμηθεί και να πληκτρολογήσει τις εξειδικευμένες εντολές προκειμένου να κάνει την ίδια εργασία. Επιπλέον τα γραφικά περιβάλλοντα παρέχουν τα άμεσα οπτικά αποτελέσματα της κάθε διεργασίας γεγονός που το κάνει πιο πολύ πιο προσιτό και φιλικό από τον προκάτοχο του, το περιβάλλον με γραμμές εντολών. Ένα ακόμα σημαντικό πλεονέκτημα που δίνει στο χρήστη ελαστικότητα και αυξάνει την παραγωγικότητα του ηλεκτρονικού υπολογιστή, είναι η δυνατότητα που παρέχουν τα GUI για πολλαπλές διεργασίες, επιτρέποντας δηλαδή την ταυτόχρονη απεικόνιση διαφόρων διεργασιών / προγραμμάτων. Άμεση συνέπεια αυτού είναι η εξοικονόμηση χρόνου σε αντίθεση με την περιορισμένη σειριακή εκτέλεση που παρείχαν τα CLI.

Αυτοί και άλλοι λόγοι παγίωσαν τη χρήση του οπτικοακουστικού περιβάλλοντος αλληλεπίδρασης. Η παγίωση όμως μιας νέας τεχνολογίας ακολουθείται άρρηκτα από την βελτίωση της, η οποία συμβαίνει από τη γέννηση των GUI μέχρι και σήμερα. Είναι γεγονός ότι ο σχεδιασμός των GUI έχει κάνει φοβερή πρόοδο που οφείλεται σε διάφορους παράγοντες. Αυτοί περιλαμβάνουν την πρόοδο στα φυσικά εξαρτήματα του υπολογιστή, δηλαδή γρηγορότερους επεξεργαστές, περισσότερη μνήμη και οθόνες υψηλότερης ανάλυσης, βελτιώσεις που έγιναν για να ικανοποιήσουν τις συνεχώς ανερχόμενες απαιτήσεις των χρηστών. Η έρευνα για την περαιτέρω εξέλιξη των GUI κινείται στη δημιουργία τρισδιάστατων περιβαλλόντων και ο έλεγχος του συστήματος με τον προφορικό λόγο. Τέτοια επιτεύγματα, όταν λάβουν χώρα, αναμφισβήτητα θα καταστήσουν την αλληλεπίδραση του ανθρώπου με τον ηλεκτρονικό υπολογιστή πολύ πιο

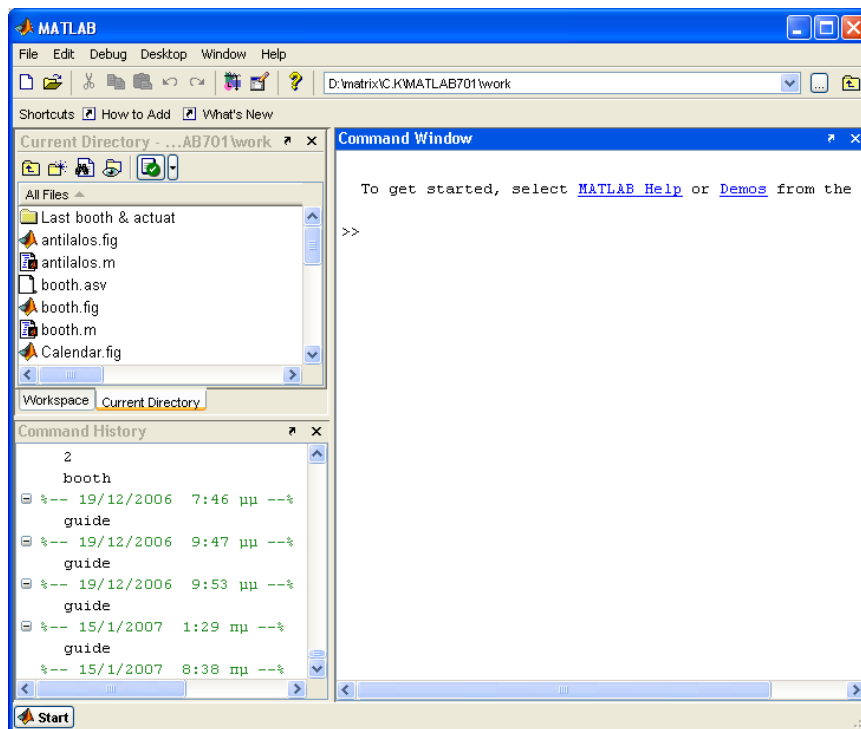
άμεση και προσωπική, αφού αυτός ολοένα και μοιάζει με ένα ζωντανό περιβάλλον ή ζωντανό οργανισμό φτιαγμένο να εξυπηρετεί πρόθυμα και ακούραστα τις καθημερινές μας ανάγκες.

2.3 Ειδικευμένα gui.

Μετά από αυτή την ιστορική μετάβαση από τη γραμμή εντολών σε γραφικό περιβάλλον ακολούθησαν πολλές εταιρίες τα βήματα της Microsoft. Όλες πλέον οι εφαρμογές που εκτελούνταν μέσω του καινούριου λειτουργικού είχαν το δικό τους περιβάλλον. Σύμφωνα με τις ανάγκες του κάθε προγράμματος οι εταιρίες παραγωγής τους κατασκεύαζαν και ένα περιβάλλον με τις ανάλογες επιλογές οι οποίες βοηθούσαν το χρήστη να το χρησιμοποιήσει με άνεση. Για παράδειγμα μια εταιρία κατασκευής ενός κειμενογράφου θα περιείχε ένα περιβάλλον εντελώς διαφορετικό από ένα πρόγραμμα επεξεργασίας εικόνας. Επίσης ο κειμενογράφος μοιραία περιέχει και ένα τμήμα αφιερωμένο στη σύνταξη κειμένου. Το ίδιο ισχύει και για μια σύγχρονη γλώσσα προγραμματισμού όπου είναι απαραίτητη και ύπαρξη εντολών σε κείμενο. Ακολουθούν δύο φωτογραφίες από τέτοια Gui που εμπεριέχουν την αναγκαία χρήση του πληκτρολογίου.



Ms Word



Matlab 7.0.1

Στην πρώτη εικόνα μπορεί κάποιος να δει ένα κειμενογράφο και στη δεύτερη εικόνα μια γλώσσα προγραμματισμού. Πολλές άλλες εφαρμογές δεν περιέχουν χώρο για σύνταξη κειμένου και όλες οι εργασίες διεκπεραιώνονται με το «ποντίκι». Τέτοιες εφαρμογές είναι για παράδειγμα οι επεξεργαστές πολυμέσων (ήχου και εικόνας). Σήμερα μάλιστα υπάρχουν GUI τα οποία εξυπηρετούν τη δημιουργία προσωπικών GUI.

Φαίνεται λοιπόν ότι κάθε gui διαφέρει από τα υπόλοιπα ώστε να ικανοποιεί το χρήστη με τις λειτουργίες του.

2.4 GUI σε συνεργασία με αυτοματισμούς.

Κάποιες χρήσεις των Η/Υ συχνά τους τοποθετούν σε μία ευρύτερη διάταξη, όπου αυτοί καλούνται να συνεργαστούν είτε με άλλους Η/Υ ή με άλλου είδους συσκευές. Αυτό συμβαίνει πολύ συχνά στον τομέα του αυτοματισμού, όταν απαιτείται κάποιο ελεγκτικό όργανο για να επιβλέπει την αυτοματοποιημένη διαδικασία. Τα καθήκοντα που ανατίθενται στους Η/Υ είναι η παρατήρηση ή η παρατήρηση με δυνατότητα επέμβασης.

Όταν συμβαίνει το πρώτο, προκύπτουν αυτοματισμοί των οποίων η λειτουργία παρατηρείται από μια κεντρική μονάδα. Εδώ ο αυτοματισμός είτε λαμβάνει αποφάσεις μόνος του για τις διάφορες διεργασίες του ή τον ενημερώνει κατά καιρούς ο διαχειριστής. Το περιβάλλον της κεντρικής μονάδας παρέχει πληροφορίες στον χρήστη για τις επιλογές του αυτοματισμού και για την κατάσταση λειτουργίας του, μέσω ενός GUI, χωρίς όμως να έχει τη δυνατότητα παραμετροποίησης του συστήματος. Αυτή η κατηγορία κυρίως παρατηρεί το σύστημα για τυχόν δυσλειτουργίες οι οποίες θα μπορούσαν να προκαλέσουν κάποιες βλάβες. Σε περίπτωση προβλήματος ο χρήστης / παρατηρητής θα πρέπει να επέμβει στον αυτοματισμό άμεσα.

Όταν συμβαίνει το δεύτερο, ο αυτοματισμός ελέγχεται από μια κεντρική μονάδα επεξεργασίας. Ο αυτοματισμός παρέχει τις απαραίτητες πληροφορίες στη μονάδα και αυτή με τη σειρά της τις επεξεργάζεται και δίνει τις κατάλληλες εντολές στα διάφορα μέρη του για να επιτευχθεί το επιθυμητό αποτέλεσμα. Έτσι και ο διαχειριστής έχει μια αμφίδρομη επικοινωνία με τον αυτοματισμό, αφού μπορεί να επέμβει στη διαδικασία μέσω του H/Y αντί να την επιβλέπει μόνο. Από τα παραπάνω μπορεί κανείς να συμπεράνει ότι η αντιμετώπιση βλαβών καθώς και ο έλεγχος για τη σωστή λειτουργία γίνονται πολύ εύκολα και τις περισσότερες φορές χωρίς να γίνει επέμβαση στον ίδιο τον αυτοματισμό.

2.5 Συμπεράσματα.

Τα συμπεράσματα από τις παραπάνω σημειώσεις δε μπορούν παρά να κινούνται στην πετυχημένη έκβαση, που είχε η εξέλιξη των περιβαλλόντων αλληλεπίδρασης με τους ηλεκτρονικούς υπολογιστές. Οι λόγοι που επιτρέπουν το χαρακτηρισμό των GUIs ως «πετυχημένη έκβαση της εξέλιξης» έχουν αναλυθεί διάσπαρτα στη μέχρι τώρα αναφορά και επίσης συγκεντρώνονται παρακάτω.

Από τη στιγμή που εισήχθησαν εικόνες στο σύστημα του ηλεκτρονικού υπολογιστή, η χρηστικότητα αυτού βρέθηκε σε πολύ υψηλότερο επίπεδο από αυτό που όριζε το σύστημα σύνταξης κειμένου. Από τη μία, τα CLIs απαιτούσαν εκπαίδευση πάνω στον κώδικα επικοινωνίας με τον υπολογιστή, κάποιο χρόνο για την πληκτρολόγηση των ειδικών φράσεων και τέλος τη διεκπεραίωση μιας μόνο εργασίας κάθε φορά. Από την άλλη, τα GUIs απαιτούν σημαντικά λιγότερη εκπαίδευση για τον τρόπο επικοινωνίας με τον ηλεκτρονικό υπολογιστή, πολύ λιγότερο χρόνο για πληκτρολόγηση αφού έχουν ελαχιστοποιηθεί οι περιπτώσεις όπου αυτή απαιτείται και η ταυτόχρονη εκτέλεση διαφόρων εργασιών από τον ηλεκτρονικό υπολογιστή, με τον περιορισμό φυσικά ότι ο χρήστης ορίζει πλήρως την πρώτη πριν προχωρήσει στην επόμενη (και η πρώτη απαιτεί κάποιο αξιοσημείωτο χρόνο για να εκτελεστεί από τον επεξεργαστή). Στα οπτικοποιημένα περιβάλλοντα λοιπόν φαίνεται πως εξοικονομείται χρόνος και προσπάθεια με άμεση συνέπεια να αυξάνεται αισθητά η χρηστικότητα και η αποτελεσματικότητα του συστήματος. Κάτι ακόμα που συμβάλει στη χρηστικότητα των υπολογιστών είναι ένα χαρακτηριστικό των γραφικών περιβαλλόντων που αφορά τα λάθη που μπορεί να προκαλέσει ο χρήστης κατά τη χρήση του συστήματος. Ένα σωστό γραφικό περιβάλλον παρέχει πάντα πληροφορίες σ' αυτόν που το χειρίζεται για την κατάσταση που βρίσκεται το σύστημα και τον συμβουλεύει για τις επιλογές του. Αντίθετα αν κάποιος εργάζεται σε γραμμή εντολών έχει μεγαλύτερες πιθανότητες να κάνει κάποιο λάθος το οποίο μπορεί να εμφανιστεί αργότερα και να προκαλέσει δυσλειτουργία ή ακόμη και την καταστροφή του συστήματος.

Όπως σε ένα περιοδικό ή ένα βιβλίο, η εισαγωγή εικόνων στον ηλεκτρονικό υπολογιστή, «τράβηξε» την προσοχή του ευρύτερου κοινού όπως είναι τα παιδιά και οι άνθρωποι χωρίς τεχνικές γνώσεις που δεν είχαν πραγματικά ανάγκη την απόκτηση ενός υπολογιστή. Το σύστημα αλληλεπίδρασης έγινε, αυτό που συχνά λέγεται, φιλικό προς το χρήστη κάνοντας όλο και περισσότερους ανθρώπους φιλικούς προς τους υπολογιστές. Αυτό που κατόρθωσε λοιπόν η μετάβαση και εδραίωση των GUIs είναι η μαζική διάδοση των PCs (προσωπικών υπολογιστών) τόσο για επαγγελματικούς αλλά όσο και για ψυχαγωγικούς σκοπούς.

3 Τα έξυπνα κτήρια.



3.1 Ορισμός.

Ένα από τα οφέλη της ταχείας εξέλιξης της τεχνολογίας πληροφορίας, υπήρξε η ανάπτυξη των συστημάτων που είναι ικανά να μετρήσουν, να αξιολογήσουν και να αποκριθούν σε αλλαγές. Τα συστήματα αυτά έφτασαν σε τέτοιο επίπεδο αποτελεσματικότητας που έδωσαν το έναυσμα για την αλλαγή στον τρόπο που σχεδιάζεται το φυσικό μας περιβάλλον, δηλαδή τα κτήρια όπου ζούμε και εργαζόμαστε. Η αλλαγή που επήλθε έχει σαν τελικό προϊόν τη δημιουργία ενός νέου τεχνολογικού τομέα, τα «έξυπνα κτήρια».

Σύμφωνα με την ιστοσελίδα http://www.nordx.com/public/htmen/3_1.htm#a, πρόκειται για κτήρια (σπίτια ή επαγγελματικοί χώροι) που ενσωματώνουν τεχνολογία της πληροφορίας και συστήματα επικοινωνιών κάνοντας τα πιο άνετα, ασφαλή, παραγωγικά και οικονομικά. Σε λίγα χρόνια η εξάπλωση αυτής της τεχνολογίας είναι δυνατό να κάνει την καθημερινή ζωή όμοια με σκηνές από φουτουριστικές ταινίες. Κάποιος θα επιστρέψει στο σπίτι του από τη δουλειά και καθώς πλησιάζει, ένας σαρωτής αμφιβληστροειδούς τον αναγνωρίζει, του ανοίγει την πόρτα και τα φώτα. Στην κουζίνα προχωρώντας προς το ψυγείο, αυτό τον ενημερώνει ότι κάποια τρόφιμα είναι σε έλλειψη και ο ιδιοκτήτης επιβεβαιώνει την παραγγελία για τα είδη αυτά, τα οποία πλέον θα καταφθάσουν μέσα σε λίγες ώρες. Περνώντας στο σαλόνι, τα τελευταία e-mail φαίνονται στην οθόνη της τηλεόρασης και ενώ τα διαβάζει, έχει άγνοια πως το πλυντήριο πιάτων έχει ανιχνεύσει ένα εσωτερικό σφάλμα και έχει αυτόματα ζητήσει βοήθεια από το τμήμα συντήρησης του κατασκευαστή. Έτσι θα είναι η ζωή σε ένα έξυπνο σπίτι, με στοιχεία αλληλεπίδρασης και αντίδρασης. Τα στοιχεία που αποτελούν το σπίτι (έπιπλα, συσκευές, σύστημα ασφαλείας, κτλ.) θα είναι συνδεδεμένα μεταξύ τους ή και με το διαδίκτυο που θα τα κάνει διαθέσιμα για ρυθμίσεις και παρακολούθηση από οποιοδήποτε μέρος του κόσμου.

Ένας γενικός ορισμός των έξυπνων κτηρίων είναι ο εξής: Ένα έξυπνο κτήριο είναι αυτό που έχει την τηλεπικοινωνιακή υποδομή, η οποία το καθιστά ικανό να ανταποκρίνεται συνεχώς και να προσαρμόζεται σε μεταβαλλόμενες συνθήκες. Έτσι κατορθώνει μια πιο αποδοτική χρήση των πόρων και αυξάνει την άνεση και ασφάλεια των κατοίκων του. Σαν παράδειγμα σχετικά με την αύξηση της ασφάλειας, αναφέρεται μια υποθετική περίπτωση πυρκαγιάς. Σ' αυτή την περίπτωση είναι δυνατό για το συναγερμό της φωτιάς να επικοινωνήσει με το

σύστημα ασφαλείας ώστε αυτό να ξεκλειδώσει τις πόρτες. Το σύστημα ασφαλείας με τη σειρά του να επικοινωνήσει με το σύστημα κλιματισμού ώστε αυτό να μειώσει τη ροή του αέρα για να μην εξαπλωθεί η φωτιά και έπειτα άλλες ενέργειες που απαιτείται να γίνουν θα είναι ήδη προκαθορισμένες και τοποθετημένες στη σωστή σειρά, αφήνοντας ελάχιστα περιθώρια για λανθασμένες αποφάσεις σε μια τέτοια κρίσιμη κατάσταση.

Οι τεχνολογίες για την υλοποίηση τέτοιων χώρων υπάρχουν ήδη και οι εταιρίες που τις αναπτύσσουν εισχωρούν σιγά σιγά στην αγορά και κάνουν τις δικές τους προτάσεις για το «χαρακτήρα» και τις ικανότητες ενός επικοινωνιακού κτηρίου.

3.2 Έξυπνες Συσκευές.

Αναφέρθηκε ήδη ότι το βασικό γνώρισμα ενός έξυπνου κτηρίου είναι ένας χώρος με τηλεπικοινωνιακή υποδομή που εξυπηρετεί την επικοινωνία διαφόρων υποσυστημάτων και συσκευών μέσα στο σπίτι. Ο όρος συσκευές αναφέρεται, όπως είναι αναμενόμενο, στις γνωστές σε όλους οικιακές συσκευές που χρησιμοποιούνται πολλές φορές καθημερινά και δε λείπουν από κανένα σπίτι (π.χ. τηλέφωνο, κουζίνα, πλυντήριο κτλ.). Αυτές όμως, στη μορφή που είναι ευρέως γνωστές, δεν είναι κατασκευασμένες ώστε να επικοινωνούν με άλλες συσκευές, ούτε να διαγνώσκουν την κατάσταση τους ή να λαμβάνουν αποφάσεις. Αυτό που λείπει λοιπόν για να συμπληρωθεί το παζλ ενός έξυπνου χώρου είναι οι έξυπνες συσκευές, που είναι σε θέση να αλληλεπιδρούν και να κάνουν επιπλέον αυτοματοποιημένες διαδικασίες από αυτές που έκαναν συνήθως οι προκάτοχοι τους. Παρακάτω φαίνεται λεπτομερέστερα πως είναι πιθανό να λειτουργήσουν «έξυπνα» κάποιες κύριες οικιακές συσκευές με βάση την περιγραφή του διαδικτυακού τόπου <http://www.jrf.org.uk/housingandcare/smarthomes/devices.asp>.

Κάθε τηλέφωνο μπορεί να χρησιμοποιηθεί για να ελέγχει το σπίτι. Με μια κλήση από εξωτερικό τηλέφωνο και εισάγοντας ένα προσωπικό κωδικό, θα μπορούσε να ενεργοποιείται ένα φωνητικό μενού και με τις κατάλληλες επιλογές να ενεργοποιούνται / απενεργοποιούνται κάποιες συσκευές του σπιτιού ή να ελέγχεται η κατάσταση τους.

Τα ντουλάπια της κουζίνας και ο νεροχύτης μπορούν να τοποθετηθούν σε μηχανισμούς ανύψωσης. Αυτό θα τους επιτρέπει να μετακινούνται στο καταλληλότερο σημείο, προς χρήση από διαφορετικά μέλη της οικογένειας. Για παράδειγμα, αν είναι δύσκολο για κάποιον να φτάσει το νεροχύτη ή το ντουλάπι, αυτά θα μπορούν να χαμηλώνουν στο επιθυμητό ύψος. Οι μηχανισμοί ανύψωσης έστω ότι ελέγχονται από παλινδρομικούς διακόπτες και μπορούν να λειτουργούν όσο οι διακόπτες είναι πιεσμένοι και ελευθερώνοντας τους οι μηχανισμοί θα σταματούν. Θα μπορούν να ελέγχονται επίσης και από μακριά με τηλεχειρισμό, όπου με το πάτημα ενός κουμπιού θα μετακινούνται σε μια προκαθορισμένη θέση. Για να γίνει ακόμα πιο λειτουργικό το σύστημα, ίσως οι μηχανισμοί να μπορούν να ανιχνεύουν τυχόν εμπόδια στην πορεία τους και να σταματούν για να αποφεύγονται ζημιές. Τέλος θα έχουν μια μπαταρία για τις περιπτώσεις διακοπής του ρεύματος.

Λοιπές συσκευές όπως το ψυγείο, το πλυντήριο και ο φούρνος μικροκυμάτων θα διαθέτουν δυνατότητα σύνδεσης με το διαδίκτυο και απεικόνισης δικτυακών τόπων σε δικιά τους οθόνη, από όπου θα ο ιδιοκτήτης θα μπορεί να σερφάρει ενώ

βρίσκεται στην κουζίνα ή στο μπάνιο του. Θα διαθέτουν αυτό-διαγνωστικές ικανότητες και θα μπορούν να συμβουλευόμαστε και να ενημερώνουν την αρμόδια ιστοσελίδα της εταιρίας. Θα προγραμματίζονται να λειτουργούν ή να παύουν από μεγάλες αποστάσεις (π.χ. από άλλη πόλη).

Στην πραγματικότητα, πολλές από αυτές τις λειτουργίες είναι εφικτές ήδη, με κάποια προβλήματα όμως σε μια πολύ βασική, τον χειρισμό των συσκευών από μεγάλη απόσταση ή κατ' επέκταση και τον έλεγχο διαφόρων συνθηκών που επικρατούν σε ένα χώρο. Υπάρχουν περιοριστικοί παράγοντες που απαιτούν χρόνο και προσεκτικό σχεδιασμό για να ξεπεραστούν. Ο σημαντικότερος παράγοντας είναι η διασφάλιση ότι οι χώροι και οι συσκευές τους θα ελέγχονται μόνο από τον εκάστοτε δικαιούχο αυτών. Θα ήταν τραγικό κάποιοι κακόβουλοι να είχαν πρόσβαση στο σύστημα ασφαλείας ενός, ξένου προς αυτούς, σπιτιού. Μια ανεπαρκής προστασία θα άφηνε τον ιδιοκτήτη εκτεθειμένο σε κινδύνους όπως είναι κακόγουστες και ανησυχητικές φάρσες ή παραβίαση του ίδιου του χώρου του. Μιας και τέτοιου είδους προσπάθειες παραβίασης είναι βέβαιο ότι θα υπάρχουν, αν σκεφτεί κανείς κάποιες άλλες μορφές ηλεκτρονικού εγκλήματος που ήδη έχουν σημειωθεί, είναι υψίστης σημασίας να μπορέσει να στηθεί κάποιο αδιάβλητο σύστημα προστασίας. Άλλο ένα ζήτημα που πρέπει να επιλυθεί, αν δεν έχει ακόμα επιλυθεί, είναι αυτό της ταυτοποίησης εκατομμυρίων ακόμα χώρων και συσκευών με τρόπο μοναδικό παγκοσμίως. Η παγκόσμια κλίμακα μπαίνει σαν μέτρο σχεδίασης διότι αυτή είναι και η επιθυμητή εμβέλεια που πρέπει να έχουν τέτοιες λειτουργίες, αφού τα ταξίδια στο εξωτερικό είναι για πολλούς καθημερινότητα. Με την ταυτοποίηση εννοείται η ηλεκτρονική ταυτοποίηση, όπως αυτή που χρησιμοποιείται σήμερα για τους ηλεκτρονικούς υπολογιστές, με ομάδες αριθμών που ανατίθενται σε κάθε σύστημα (IP) και ονομάζεται ειδικότερα «διευθυνσιοδότηση».

Παρά όμως τους περιορισμούς αυτούς, και μη γνωρίζοντας ως πιο βαθμό έχουν αντιμετωπιστεί, γνωστές εταιρίες οικιακών συσκευών τα τελευταία χρόνια έχουν διαθέσει στην αγορά συσκευές που μπορούν να χαρακτηρισθούν έξυπνες αφού έχουν εξελιγμένες λειτουργίες τηλεπικοινωνιακής φύσεως όπως φαίνονται και στην ιστοσελίδα όπως φαίνεται και στην ιστοσελίδα <http://www.lginternetfamily.co.uk>. Για παράδειγμα, η εταιρία LG κατασκεύασε το WM-16100FD , ένα πλυντήριο με δυνατότητα πρόσβασης στο Internet (διαδίκτυο). Σε ένα μικρό άρθρο που παρουσιάζει τις λειτουργίες του αναφέρει χαρακτηριστικά, «Ξεχάσατε να ενεργοποιήσετε το πλυντήριο σας πριν φύγετε για τη δουλειά;» και παρακάτω εξηγεί ότι μπορεί να ενεργοποιηθεί από το περιβάλλον της δουλειάς μέσω του διαδικτύου. Συνεχίζει λέγοντας ότι είναι δυνατό να παρακολουθήσει κανείς τον κύκλο πλύσης, να διαγνώσει σφάλματα και να «κατεβάσει» νέα προγράμματα πλύσεων από τον ειδικά διαμορφωμένο δικτυακό τόπο. Διαθέτει οθόνη υγρών κρυστάλλων και περιβάλλον αλληλεπίδρασης με εικόνες για να είναι φιλικό με το χρήστη. Παρακάτω φαίνεται η πρόσοψη του που προαναφερθέντος πλυντηρίου.



*Διαδικτυακό Πλυντήριο
LG- WM-16100FD*

Η ίδια εταιρία έφερε στην αγορά και ψυγεία ικανά να συνδέονται στο διαδίκτυο και να διαχειρίζονται οπτικοακουστικά ψηφιακά δεδομένα (πολυμέσα). Ένα από αυτά είναι το GRD-267DTU, ψυγείο-καταψύκτης με οθόνη υγρών κρυστάλλων. Σε αυτό μπορεί κανείς να δει τηλεόραση, να ακούσει μουσική ή να «σερφάρει» στο Internet. Πιο συγκεκριμένα, έχει ενσωματωμένη μηχανή αναπαραγωγής ήχου σε μορφή MP3, ενσωματωμένη κάμερα και μικρόφωνο για τη δημιουργία και αποστολή ηλεκτρονικού βίντεο-ταχυδρομείου. Μπορεί να διαγνώσει μικροσφάλματα στην οθόνη του και έχει σελίδα με περιεχόμενα όπου μπορεί ο χρήστης να καταχωρήσει και να παρακολουθήσει τα τρόφιμα που υπάρχουν στο ψυγείο και τις αντίστοιχες ημερομηνίες λήξης τους. Στα χαρακτηριστικά του, προστίθενται και ένα ηλεκτρονικό ημερολόγιο για να φυλάσσονται σημαντικές ημερομηνίες, ηλεκτρονικό αρχείο με πληροφορίες και συμβουλές σχετικά με τη θρεπτική αξία των τροφών που υπάρχουν στο ψυγείο, την ανίχνευση τροφίμων και το χρόνο αποθήκευσης τους μέσα στον ψυγείο-καταψύκτη, ψηφιακά εγχειρίδια λειτουργίας και συντήρησης, συνταγές μαγειρέματος, πληροφορίες για τον καιρό και αναγνώριση γραφικού χαρακτήρα. Η πρόσοψη του φαίνεται παρακάτω



*Διαδικτυακός Ψυγείο-Καταψύκτης
GRD-267DTU*

Υλοποιήσεις όπως οι παραπάνω φέρνουν όλο και πιο κοντά στην πραγματοποίηση τους έξυπνους χώρους, που θα αυτοματοποιήσουν και κατά

συνέπεια θα απλοποιήσουν την καθημερινότητα των ανθρώπων που θα αποφασίσουν να αναβαθμίσουν τη ζωή τους με αυτόν τον τρόπο.

3.3 Διασφάλιση του μέλλοντος των έξυπνων κτηρίων και επένδυση σε αυτά μέσω της εκπαίδευσης.

Η μεγαλύτερη ένδειξη ότι η τεχνολογία των αυτοματοποιημένων κτηρίων έχει εδραιωθεί, παρά την περιορισμένη ακόμα εφαρμογή της, είναι το γεγονός ότι αυτή διδάσκεται σε πάνω από χίλια σχολεία και μάλιστα αξίζει να ειπωθεί πως υπάρχουν μαθήματα διαμορφωμένα για παιδιά του νηπιαγωγείου.

Σύμφωνα με τους Ben Erwin, Martha Cyr & Chris Rogers, το καλοκαίρι του 1993, το αμερικανικό Πανεπιστήμιο Tufts ψάχνοντας για ένα φτηνό σύστημα απόκτησης δεδομένων συνάντησε το Control Lab της εταιρίας Lego. Η ικανότητα χτισίματος με τα γνωστά παιδικά «τουβλάκια» της Lego σε συνδυασμό με το περιβάλλον αλληλεπίδρασης (Control Lab) στον ηλεκτρονικό υπολογιστή είχε σαν αποτέλεσμα ένα ιδανικό συνδυασμό για γρήγορο, δημιουργικό και διασκεδαστικό τρόπο σχεδιασμού και κατασκευής. Το πανεπιστήμιο Tufts, αναβάθμισε αυτό το πακέτο δημιουργώντας το πρόγραμμα οδήγησης LabView. Στη συνέχεια δημιούργησε μια σειρά μαθημάτων και προέβαλε την προσπάθεια του μέσω μιας ιστοσελίδας. Το αποτέλεσμα έπεισε τη NASA να χρηματοδοτήσει ένα ευρείας κλίμακας σχέδιο εκμάθησης στα τοπικά δημοτικά σχολεία.

Σήμερα υπάρχουν πακέτα που περιέχουν την απαραίτητη βιβλιογραφία με θεωρία και πειραματικές ασκήσεις που απευθύνονται σε μαθητές από 5 έως 50 ετών. Η προσαρμοστικότητα των υλικών και του προγράμματος, επιτρέπουν μια μεγάλη ποικιλία κατασκευών τις οποίες οι μαθητές μπορούν να «χτίσουν» και να προγραμματίσουν (ρομπότ, οχήματα και αυτοματοποιημένους χώρους). Καθώς οι μαθητευόμενοι υλοποιούν αυτές τις εργασίες παρακινούνται να μάθουν τα μαθηματικά και τις επιστημονικές αρχές που χρειάζονται για να βελτιστοποιήσουν τις κατασκευές τους. Παρακάτω φαίνονται φωτογραφίες από τη διδασκαλία ενός τέτοιου μαθήματος οι οποίες προέρχονται από την ιστοσελίδα <http://www.marshall.edu/LEGO/dc/24Oct01/DCHouse24Oct01all.html>



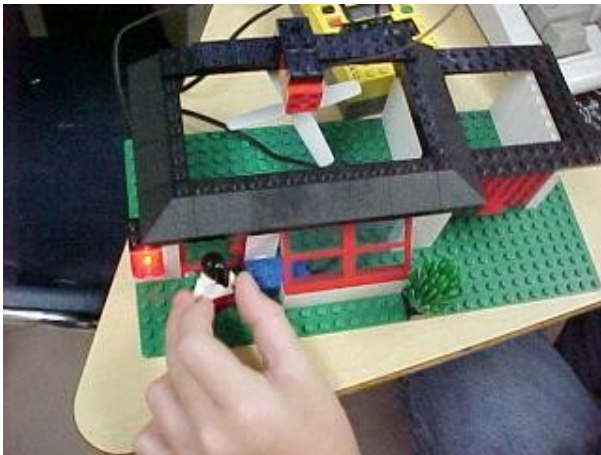
Χτίζοντας το σπίτι με τουβλάκια LEGO



Interface ελέγχου



Το πρόγραμμα Robolab.



Ο ανεμιστήρας είναι προγραμματισμένος να ενεργοποιηθεί όταν το ανθρωπάκι περπατήσει μπροστά από τον αισθητήρα φωτός.



Οι φωτογραφίες αυτές τραβήχτηκαν στις 24 Οκτωβρίου του 2001 στο Davis Creek δημοτικό σχολείο. Στο τελευταίο ζευγάρι φωτογραφιών γίνεται αναφορά σε ανεμιστήρα και σε αισθητήρα φωτός. Όπως είναι φυσικό, για την κατασκευή έστω και ενός «έξυπνου» μοντέλου χρειάζονται αισθητήρες, έλεγχος των λειτουργιών, μηχανικά μέρη (όπως ο ανεμιστήρας) και καλωδίωση για επικοινωνία των τμημάτων. Έτσι το «πακέτο» της Lego δεν μπορεί παρά να τα περιλαμβάνει όλα αυτά και μάλιστα στη χαρακτηριστική μορφή που βγάζει και τα υπόλοιπα της προϊόντα. Παρακάτω φαίνονται κάποιοι από αυτούς τους αισθητήρες που προέρχονται από το διαδικτυακό τόπο <http://www.legoeducation.com/store/default.aspx?pt=23&searchtype=0&sport=3&c=0&t=1&l=0>



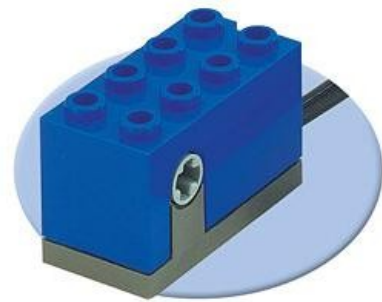
9-Volt Αισθητήρας φωτός. Ευαισθησία: 0,6 LUX έως 760 LUX



9-Volt Αισθητήρας θερμοκρασίας. Ευαισθησία: -20 έως +50 βαθμούς Κελσίου



9-Volt Αισθητήρας αφής



9-Volt Αισθητήρας γωνίας

Το επόμενο τουβλάκι είναι υπεύθυνο για την ανεξαρτητοποίηση του αυτοματισμού από τον ηλεκτρονικό υπολογιστή. Πρόκειται για ένα μικροελεγκτή που δέχεται το πρόγραμμα από τον ηλεκτρονικό υπολογιστή, το αποθηκεύει και έπειτα αυτό ελέγχει και παρακολουθεί τον αυτοματισμό όπως έχει οριστεί. Πιο συγκεκριμένα το εξάρτημα αυτό ονομάζεται RCX και λαμβάνει προγράμματα από το λογισμικό Robolab μέσω ενός υπέρυθρου συνδέσμου με τον ηλεκτρονικό υπολογιστή ενώ αποθηκεύει έως και πέντε προγράμματα κάθε φορά. Συνεργάζεται με μέχρι και τρεις αισθητήρες (είσοδοι) από τους οποίους μπορεί διαδοχικά να καθορίσει τι θα κάνουν οι τρεις έξοδοι (μοτέρ, φώτα, κτλ.) ανάλογα με τα δεδομένα που «αισθάνεται». Εσωτερικά έχει ένα 10-bit μετατροπέα αναλογικού σήματος σε ψηφιακό, 4 χρονομετρητές (timers), 32 εσωτερικές μεταβλητές και οι 3 έξοδοι του είναι διαμορφωμένοι με κώδικες παλμών (PWM). Παρακάτω φαίνεται μια φωτογραφία του



RCX. Προγραμματιζόμενο «τουβλάκι»

Ένα προϊόν εκμάθησης τόσο ελκυστικό για όλες τις ηλικίες, δε μπορεί παρά να εκπληρώσει το στόχο του, εφόσον φυσικά αυτό εξακολουθήσει να υποστηρίζεται και να διαδίδεται από τα αρμόδια όργανα, ενώ το γεγονός ότι απευθύνεται και σε πολύ μικρές ηλικίες σημαίνει ότι αυτά που διδάσκει μπορούν να αποτελέσουν κομμάτι γενικής παιδείας και ιδιαίτερα σημαίνει ότι αυτές οι γνώσεις πρόκειται να αφομοιωθούν σε βάθος αφού όσο νεότερος είναι ο μαθητευόμενος τόσο βαθύτερα χαράζονται οι γνώσεις στη συνείδηση του. Έτσι αυτή η καλά αφομοιωμένη γνώση προς όλους τους μελλοντικούς πολίτες αποτελεί πολύ ισχυρά θεμέλια για την εδραίωση της τεχνολογίας των έξυπνων κτηρίων, αφού θα αποδώσει είτε καλούς τεχνολόγους που θα αναβαθμίσουν το αντικείμενο και θα το φέρουν στην καθημερινότητα ή χρήστες εξοικειωμένους και έτοιμους να αυτοματοποιήσουν τη ζωή τους.

3.4 Περιγραφή του πειραματικού θαλάμου.



3.4.1 Γενικά.

Στις επόμενες παραγράφους, πρόκειται να εξεταστεί ένας πειραματικός θάλαμος που κατασκευάστηκε για την επίδειξη και την παρατήρηση ενός αυτοματοποιημένου χώρου. Υλοποιήθηκε στο Τμήμα Ηλεκτρονικής του Τ.Ε.Ι Χανίων και παρουσιάστηκε σαν Πτυχιακή Εργασία του φοιτητή Σαριδάκη Γιώργου κάτω από την επίβλεψη της καθηγήτριας Κολοκοτσά Διονυσίας. Πρόκειται λοιπόν για ένα μικρό δωμάτιο του οποίου οι διαστάσεις θυμίζουν τηλεφωνικό θάλαμο, και είναι εξοπλισμένο με διάφορες συσκευές όπως είναι το κλιματιστικό, θερμόμετρο, λουξόμετρο, μηχανικά ανακλινόμενο παράθυρο και διάφορους αισθητήρες (φωτισμού, θερμοκρασίας κ.α.). Το ξεχωριστό χαρακτηριστικό του όμως είναι η ειδική διασύνδεση όλων των συσκευών και αισθητήρων του με τρόπο που να επικοινωνούν μεταξύ τους και να ενημερώνουν έναν κεντρικό επεξεργαστή για την εκάστοτε κατάσταση τους. Ο ρόλος του κεντρικού επεξεργαστή, όπως είναι αναμενόμενο, είναι να λαμβάνει αποφάσεις και να δίνει εντολές ανάλογες με τις προκαθορισμένες επιλογές του χρήστη και τις ισχύουσες συνθήκες. Όλο το σύστημα έχει δυνατότητα σύνδεσης με ηλεκτρονικό υπολογιστή για αναλυτικότερη παρακολούθηση και αμεσότερο έλεγχο. Με απλά όμως λόγια, ο θάλαμος είναι ικανός να ρυθμίζει αυτόματα τον φωτισμό, τον κλιματισμό και τον εξαερισμό του ώστε αυτά να παραμένουν σε επίπεδα άνετα για τον άνθρωπο ενώ ταυτόχρονα εξοικονομείται ενέργεια. Η όλη διασύνδεση συστήματος είναι υλοποιημένη εφαρμόζοντας την τεχνική EIB (European Installation BUS).

3.4.2 Η τεχνολογία που χρησιμοποιείται στο θάλαμο.

3.4.2.1 Η Τεχνική EIB.

Το Ευρωπαϊκό σύστημα σύγχρονων εγκαταστάσεων EIBUS είναι ένα κοινό βιομηχανικό πρότυπο εγκαταστάσεων, αποτέλεσμα σύμπραξης δεκάδων ευρωπαϊκών και μη, κατασκευαστικών εταιριών οι οποίες συνέθεσαν το μη κερδοσκοπικό συνεταιρισμό EIBA (European Installation Bus Association) στις 8 Μαΐου του 1990. Το αποτέλεσμα αυτής της σύμπραξης είναι ένα αυτοματοποιημένο σύστημα διαχείρισης και λειτουργίας ενός χώρου. Το σύστημα αυτό έχει σαν βασικό χαρακτηριστικό γνώρισμα την επικοινωνία μεταξύ των συσκευών που το αποτελούν.

Πρακτικά το EIBUS υλοποιείται με τρεις διαφορετικούς, ως προς την επικοινωνία των συσκευών, τρόπους. Έτσι η μετάδοση των δεδομένων μπορεί να γίνεται, με συνεστραμμένο ζεύγος καλωδίων, μέσω της γραμμής ισχύος που είναι διαθέσιμη σε όλα τα σπίτια και με ασύρματο τρόπο. Ο τρόπος που χρησιμοποιείται στο θάλαμο δοκιμών είναι η μετάδοση με το συνεστραμμένο ζεύγος καλωδίων (twisted pair) που χαρακτηρίζεται και ως ο πιο αξιόπιστος. Στο επόμενο σχήμα φαίνονται διάφορα τεχνικά χαρακτηριστικά μιας υλοποίησης EIB με twisted pair.

Γραμμή Bus	Χαρακτηριστικά
Τύπος καλωδίου.	ΥCΥM 2x2x0,08 Ένα ζεύγος (κόκκινο, μαύρο) για μεταφορά σήματος και τροφοδοσία και ένα ζεύγος (κίτρινο-άσπρο) εφεδρικό
Τρόποι διακλάδωσης	Γραμμικά, αστεροειδής, μορφής δένδρου
Μήκος γραμμής	1000m μέγιστο ,συνολικά
Απόσταση δύο συνδρομητών	700m μέγιστη
Απόσταση συνδρομητή και τροφοδοτικού.	350m μέγιστη
Bus συνδρομητές	
Αριθμός bus συνδρομητών ανά γραμμή	μέχρι 64
Αριθμός γραμμών ανά περιοχή	μέχρι 12
Αριθμός περιοχών	μέχρι 15
Τάση τροφοδοσίας	
Σύστημα τάσης	DC 24Volt (SELV)
Τροφοδοσία ανά γραμμή	1 Τροφοδοτικό (320 mA)και ένα πηνίο ή ένα τροφοδοτικό (640 mA)
Τροφοδοσία σε γραμμές με αυξημένες απαιτήσεις	Μέχρι δύο τροφοδοτικά σε απόσταση τουλάχιστον 200m μεταξύ τους.
Μετάδοση πληροφοριών	
Τεχνική μετάδοσης	Αποκεντρωμένη, σειριακή, συμμετρική
Ταχύτητα μετάδοσης	9600 bps
Στοιχεία bus συσκευών (συνδρομητών)	
Βαθμός προστασίας	IP 20
Κλάση προστασίας	II
Αντιπαρασιτική προστασία	Με βάση τις προδιαγραφές IEC,DIN,EIBA
Όρια θερμοκρασιών περιβάλλοντος σε λειτουργία	-5 μέχρι +45 βαθμούς Κελσίου
Όρια θερμοκρασιών αποθήκευσης	-40 μέχρι 55 βαθμούς Κελσίου.
Σήμανση CE	Με βάση τις γραμμές EMV και χαμηλής τάσεως.

Τεχνικά χαρακτηριστικά του EIB με twisted pair

3.4.2.2 Το EIS & οι EIB Λειτουργίες.

Αναφέρθηκε προηγουμένως ότι το σύστημα EIB είναι αποτέλεσμα σύμπραξης δεκάδων κατασκευαστικών εταιριών, έτσι ισχύει στις μέρες μας ότι οι συσκευές που απαρτίζουν ένα σύστημα EIB δεν είναι φτιαγμένες από την ίδια εταιρία αλλά από διαφορετικές. Για είναι αρμονική η συνεργασία (επικοινωνία) των διαφόρων συσκευών πρέπει η κάθε κατασκευαστική εταιρία να δώσει στο προϊόν της μια κοινή προς όλους «γλώσσα» και συμπεριφορά. Έχει φτιαχτεί λοιπόν ένα πρότυπο που πρέπει να ακολουθείται και λέγεται EIS (EIB Interworking Standard) ή Πρότυπο Συνλειτουργίας EIB.

Το EIS προκαθορίζει κανόνες για διάφορα πράγματα όπως είναι ο ρυθμός επανάληψης των μηνυμάτων που αποστέλλονται, οι προτεραιότητες κάποιων συγκεκριμένων μηνυμάτων (συναγερμός, απλό μήνυμα κ.ο.κ.) και η δομή των μηνυμάτων (ή τηλεγραφημάτων). Αναλυτικότερα, έχει δομηθεί ένας πίνακας με τους υπάρχοντες κανόνες EIS, αριθμημένους και αντιστοιχισμένους με κάποιες τυπικές (συνήθειες) λειτουργίες των συσκευών. Αυτές οι τυπικές λειτουργίες λέγονται Λειτουργίες EIB και πρακτικά αυτές είναι μεταβλητές για τη συγκράτηση σημαντικών για τη (συν)λειτουργία τιμών.

Τυποποίηση EIS

EIS #	EIB λειτουργία
EIS 1	Διακοπτικός έλεγχος (Switching)
EIS 2	Ρύθμιση (Dimming)
EIS 3	Χρόνος (Time)
EIS 4	Ημερομηνία (Date)
EIS 5	Τιμή (Value)
EIS 6	Ορισμός Κλίμακας (Scaling)
EIS 7	Έλεγχος οδήγησης (Drive control)
EIS 8	Προτεραιότητα (Priority)
EIS 9	Τιμή τύπου float (Float Value)
EIS 10	16-bit counter Value
EIS 11	32-bit counter Value
EIS 12	Πρόσβαση (Access)
EIS 13	Χαρακτήρες ASCII (ASCII Characters)
EIS 14	8-bit Counter Value
EIS 15	Character String

Τυποποίηση EIS

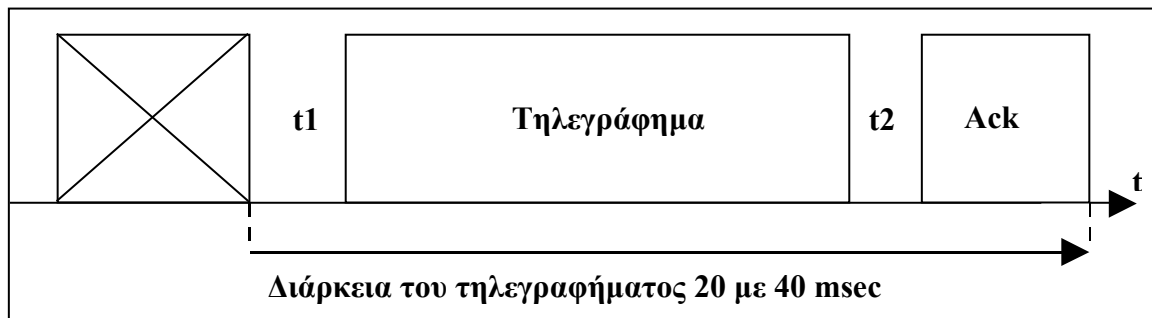
Με μια ματιά στον παραπάνω πίνακα φαίνεται ότι οι EIB Λειτουργίες μπορεί να είναι διάφορες τιμές (ώρα & counter Value), τιμές ελέγχου συσκευών (Switching & Dimming) ή τιμές κατηγοριών (Priority). Η χρησιμότητα των EIB λειτουργιών θα γίνει περισσότερο αντιληπτή στην παράγραφο 3.4.2.5 που αναφέρεται στο πρόγραμμα ETS.

3.4.2.3 Το ψηφιακό τηλεγράφημα.

Η επικοινωνία δεν επιτυγχάνεται μόνο με την επιλογή του μέσου μετάδοσης, εκτός από αυτό αξίζει να αναφερθεί και ο τρόπος μετάδοσης των δεδομένων. Οι πληροφορίες «ταξιδεύουν» με τη μορφή τηλεγραφημάτων που μεταδίδονται συμμετρικά στο Bus (μέσο μετάδοσης πληροφορίας). Τα τηλεγραφήματα αυτά δεν αλλοιώνονται από μια εξωτερική παρεμβολή που (πιθανότατα) θα προσβάλει και τα δύο σύρματα καθώς αυτά έχουν μια μεταξύ τους διαφορά δυναμικού και όχι μια διαφορά δυναμικού με τη γη. Έτσι μια εξωτερική παρεμβολή θα επηρεάσει εξίσου το ηλεκτρικό δυναμικό και των δύο συρμάτων που αποτελούν το ζεύγος, διατηρώντας ταυτόχρονα τη διαφορά δυναμικού τους σταθερή. Ο ρυθμός μετάδοσης της πληροφορίας είναι 9600 bps και ο μέσος χρόνος μετάδοσης είναι 25 msec. Το πρωτόκολλο του EIB ορίζει ότι κάθε στιγμή πρέπει να υπάρχει μονάχα ένα τηλεγράφημα (πακέτο πληροφορίας) στο μέσο μετάδοσης για να αποφευχθεί η απώλεια πληροφοριών. Αυτό επιτάσσει κάποιο είδος ελέγχου της κυκλοφορίας στο Bus, καθώς είναι πιθανό να στείλουν ταυτόχρονα πληροφορίες παραπάνω από μία συσκευές. Ο έλεγχος που χρησιμοποιείται λοιπόν λέγεται Αισθητήρας πολλαπλής πρόσβασης στη γραμμή μεταφοράς με αποφυγή σύγκρουσης “CSMA/CA” (Carrier Sense Multiple Access with Collision Avoidance). Επιπλέον, υπάρχουν και κάποια τηλεγραφήματα που

χαρακτηρίζονται σημαντικά, όπως είναι αυτά που αναφέρουν σφάλματα και έχουν προτεραιότητα στη μετάδοση.

Η γενική συμπεριφορά του συστήματος όσον αφορά την κυκλοφορία των πληροφοριών στο μέσο μετάδοσης φαίνεται συνοπτικά στο ακόλουθο σχήμα, σε συνάρτηση με το χρόνο.



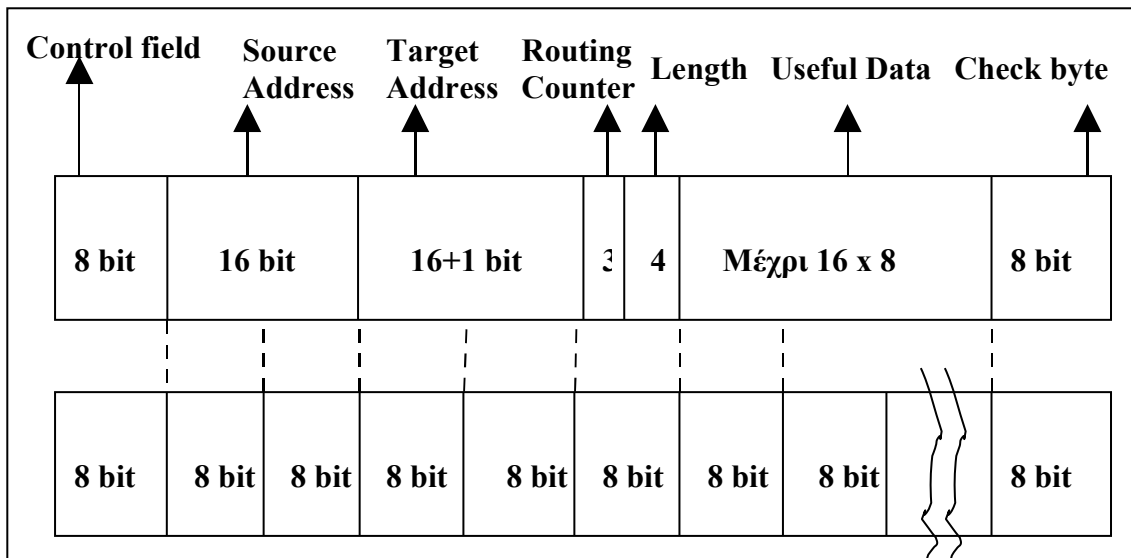
Σχηματική αναπαράσταση της επικοινωνιακής συμπεριφοράς του συστήματος σε συνάρτηση με το χρόνο.

Το πρώτο τετράγωνο κουτάκι με τις δύο διαγώνιους, στην αρχή του σχήματος, αναπαριστά ένα οποιοδήποτε συμβάν στο σύστημα (π.χ. όταν πατηθεί ένας διακόπτης ή όταν αποφασισθεί η ενεργοποίηση μιας συσκευής). Μετά από οποιοδήποτε συμβάν εμφανίζεται η ανάγκη για επικοινωνία μεταξύ δύο ή περισσοτέρων συσκευών. Αυτή δε μπορεί να είναι άμεση, αλλά θα πρέπει να υπακούσει τους κανόνες της κυκλοφορίας που έχουν τεθεί για το Bus. Πιο συγκεκριμένα το Bus, θεωρείται ελεύθερο προς χρήση όταν αυτό παραμείνει αχρησιμοποίητο για τουλάχιστον μια χρονική περίοδο t_1 . Μετά το πέρας μιας τέτοιας περιόδου ξεκινά η επικοινωνία, με τη μετάδοση του τηλεγραφήματος η οποία διαρκεί από 20 έως 40 msec. Στην ολοκλήρωση του τηλεγραφήματος οι συσκευές για ένα διάστημα t_2 χρησιμοποιούν τους διαγνωστικούς αλγορίθμους τους για να αποφασίσουν αν το τηλεγράφημα έχει παραληφθεί σωστά.

Αφού ελεγχθεί το τηλεγράφημα, όλες οι συσκευές που ενεπλάκησαν στέλνουν ένα μήνυμα για την επιβεβαίωση της σωστής επικοινωνίας, ταυτόχρονα. Το μήνυμα της επιβεβαίωσης αναπαριστάται στο σχήμα σαν το τελευταίο τετραγώνάκι.

Μια πιο αναλυτική ματιά στα τηλεγραφήματα του EIB θα βοηθήσει στην κατανόηση της «γλώσσας» που χρησιμοποιούν οι συσκευές για να επικοινωνήσουν και θα αναδείξει την ανάγκη για μετάδοση πρόσθετων πληροφοριών και την ανάγκη για προετοιμασία του συστήματος για την επικοινωνία όπως είναι η ταυτοποίηση των συσκευών μέσα στο σύστημα.

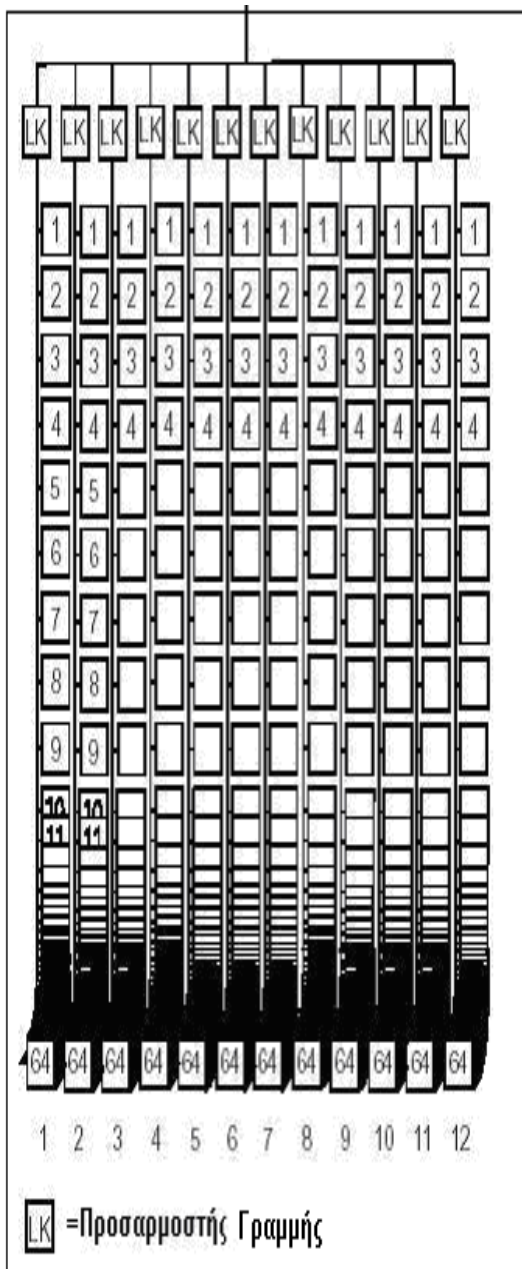
Έχει προσχεδιαστεί λοιπόν μια συγκεκριμένη δομή για τα τηλεγραφήματα η οποία απεικονίζεται παρακάτω.



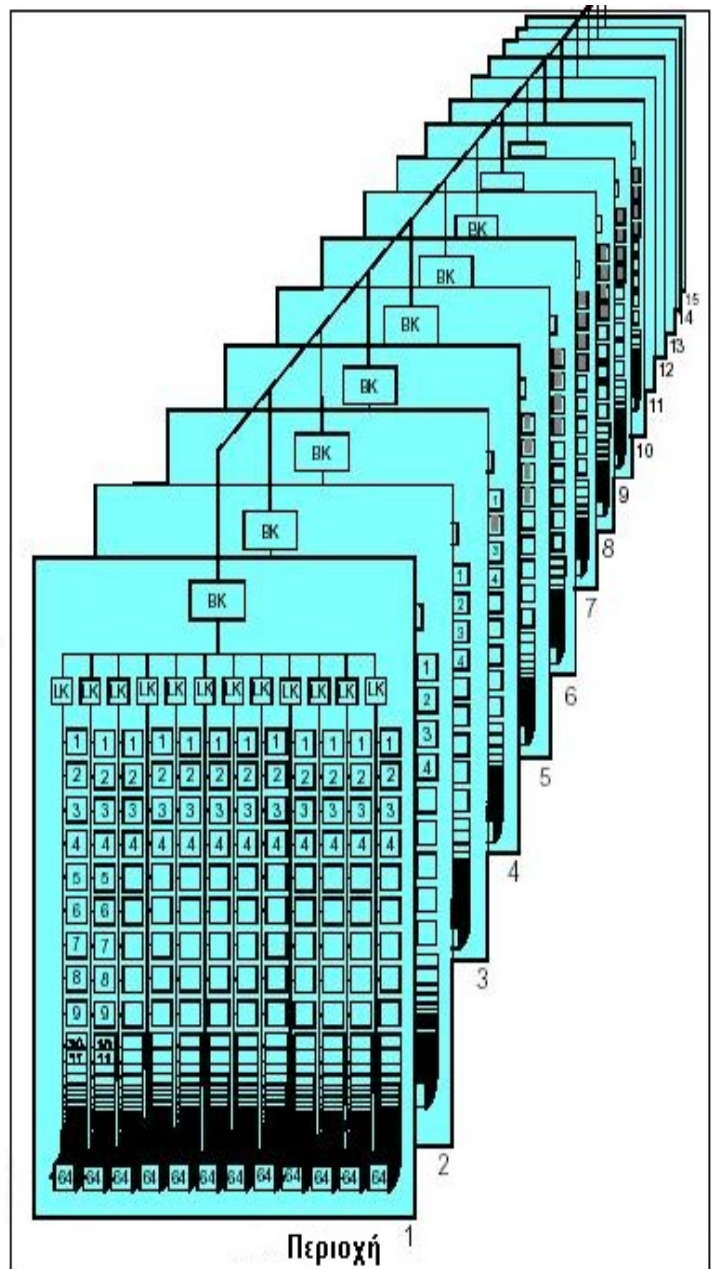
Η δομή των τηλεγραφημάτων στην τεχνική EIB

Το σχήμα αμέσως μαρτυρά ότι ο ελάχιστος όγκος δεδομένων είναι 8 bit. Το πρώτο από αριστερά πεδίο που ορίζεται σαν Πεδίο ελέγχου (Control Field) και το μήνυμα που αυτό μεταφέρει βοηθά στον ορισμό της προτεραιότητας στην περίπτωση που διάφορες συσκευές ζητήσουν να προσπελάσουν το Bus ταυτόχρονα ενώ επίσης εμποδίζει την επανάληψη της ανάγνωσης του τηλεγραφήματος από συσκευές που ήδη το έχουν παραλάβει.

Το Πεδίο Διεύθυνσης του Αποστολέα (Source Address), περιέχει τη φυσική διεύθυνση της συσκευής που αποστέλλει το οποιοδήποτε τηλεγράφημα. Η φυσική διεύθυνση είναι μια ψηφιακή λέξη που έχει προκαθοριστεί κατά το σχεδιασμό του αυτοματισμού. Με αυτό τον τόπο όλα τα επιμέρους τμήματα είναι ταυτοποιημένα, γεγονός απαραίτητο για την επικοινωνία και την παρατήρηση. Η ψηφιακή λέξη του Source Address έχει την δική της δομή για τον ταχύτερο και ευκολότερο εντοπισμό, μια δομή που αξιοποιεί την τοπολογία της τεχνολογίας EIB. Έτσι ορίζεται μια «περιοχή» η οποία αποτελείται από 12 γραμμές και η γραμμή αποτελεί τη μικρότερη δυνατή εγκατεστημένη μονάδα κάτω από την οποία όμως μπορούν να λειτουργήσουν μέχρι και 64 Bus συσκευές. Με τη χρήση προσαρμοστών περιοχών μπορούν να συνδεθούν μέχρι και 15 περιοχές.

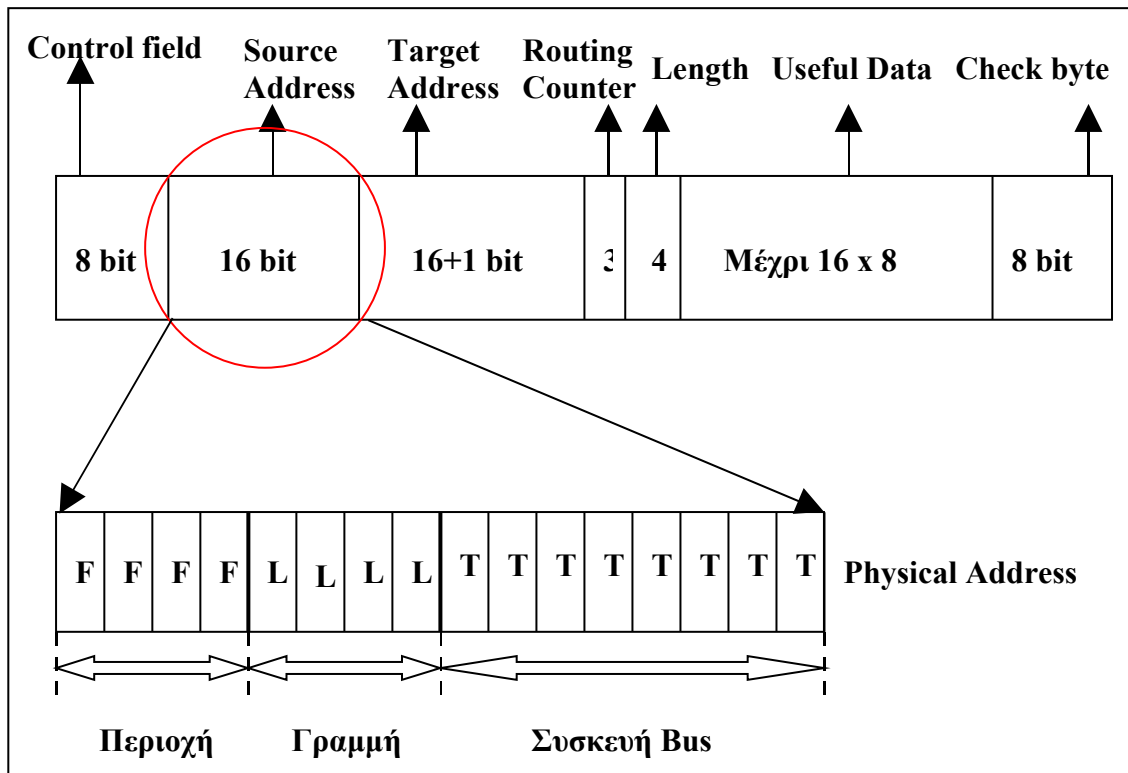


Μια «περιοχή» στην τοπολογία των συστημάτων EIB



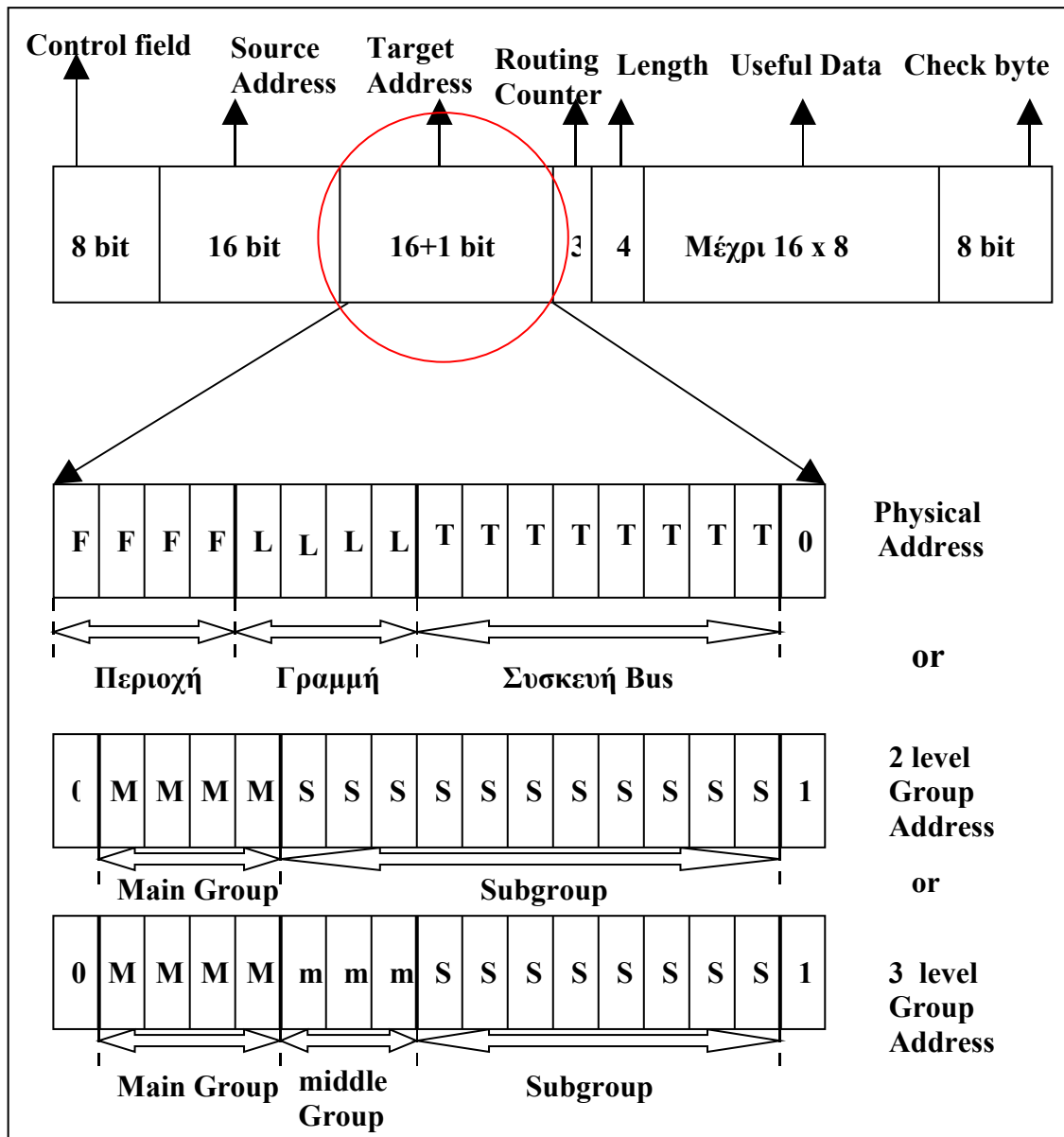
Συνδυασμός 15 περιοχών

Εξετάζοντας τώρα το πεδίο Source Address, από τα 16 bits που καταλαμβάνει συνολικά, τα 4 πρώτα αναφέρονται στην «περιοχή» που βρίσκεται ο αποστολέας και για τη μετάδοση με συνεστραμμένο ζεύγος καλωδίων αυτή παίρνει τιμές από 1(decimal) έως 15(decimal) ή 0001(binary) έως 1111(binary), αφού μέχρι τόσες προβλέπονται από την τοπολογία του συστήματος. Τα επόμενα 4 bits δηλώνουν τη γραμμή στην οποία βρίσκεται ο αποστολέας (από 0001 έως 1100 ή 1 έως 12). Και τα τελευταία δηλώνουν ποια συσκευή του Bus είναι (0001 έως 01000000 ή 1 έως 64). Η δομή του Source Address συνοψίζεται στο παρακάτω σχήμα.



Ανάλυση του πεδίου Source Address σε Bit

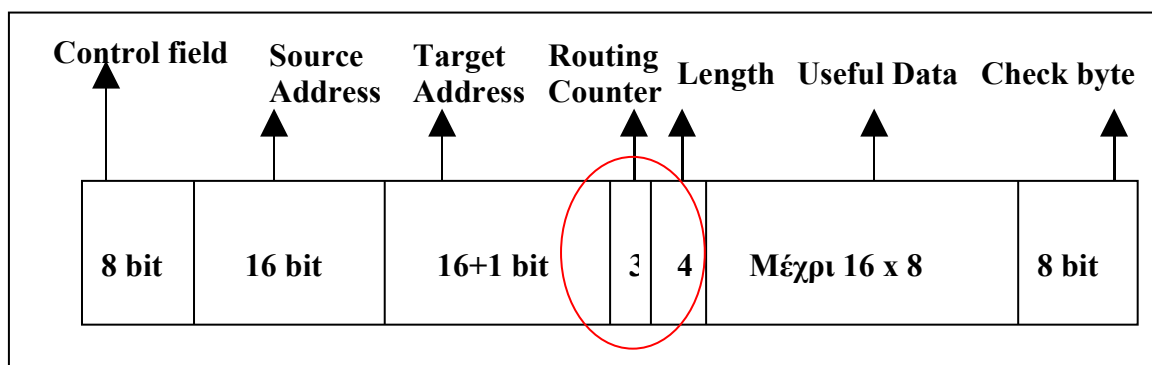
Το επόμενο προς τα δεξιά πεδίο (τετραγωνάκι), είναι η Διεύθυνση του Παραλήπτη (Target Address). Υπάρχει μια διαφοροποίηση από τη διεύθυνση της πηγής, η οποία χρησιμοποιεί τη «φυσική διεύθυνση» των συσκευών που βασίζεται στην τοπολογία του συστήματος. Η διεύθυνση του παραλήπτη, δε βασίζεται απαραίτητα στην τοπολογία του συστήματος και δε λέγεται πάντα «φυσική». Πολλές φορές λέγεται «διεύθυνση ομάδος» (Group Address), και μια τέτοια διεύθυνση μπορεί να αναφέρεται σε μία ή περισσότερες συσκευές ανεξαρτήτου περιοχής ή γραμμής. Αυτός ο τρόπος διευθυνσιοδότησης μπορεί να αποδώσει περισσότερες από μία διευθύνσεις στην ίδια συσκευή και σκοπός της είναι να περιγράψει σχέσεις επικοινωνίας μέσα σε ολόκληρο το σύστημα. Ο πρακτικός ρόλος του Group Addressing είναι η ταυτόχρονη αποστολή μηνύματος σε διαφόρους παραλήπτες. Έτσι, όταν πρόκειται να σταλεί τηλεγράφημα σε μία συσκευή, χρησιμοποιείται η φυσική της διεύθυνση. Όταν όμως ένα τηλεγράφημα απευθύνεται σε περισσότερους από έναν παραλήπτες, τότε χρησιμοποιείται η διεύθυνση ομάδος. Ανάλογα με τις προτιμήσεις που έγιναν κατά τη φάση του προγραμματισμού του συστήματος, υπάρχει και ο ειδικότερος τρόπος απόδοσης μιας διεύθυνσης ομάδος που ονομάζεται διεύθυνση δύο ή τριών επιπέδων. Χρησιμοποιείται για να περιγράψει ομάδες συσκευών και τις κατατάσσει σε «Κύρια Ομάδα» (Main Group), «Μεσαία Ομάδα» (Middle Group) και «Υποομάδα» (Subgroup). Για παράδειγμα θα μπορούσε να έχει οριστεί σαν «Κύρια Ομάδα» μιας διεύθυνσης ομάδος, ο φωτισμός. Σαν «Μεσαία Ομάδα» ο φωτισμός του Χ χώρου και σαν «Υποομάδα» το Ψ φωτιστικό του Χ χώρου. Γνωρίζοντας τα παραπάνω γίνεται εύκολη η κατανόηση της ψηφιακής λέξης που αποδίδει τη Διεύθυνση του Παραλήπτη και φαίνεται γραφικά στο παρακάτω σχήμα.

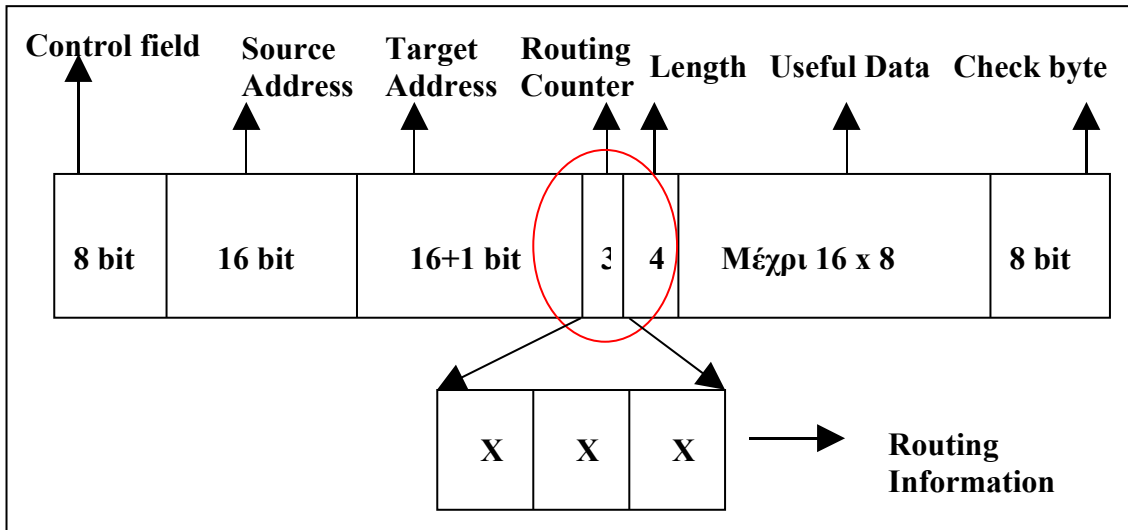


Ανάλυση του πεδίου Target Address σε bit

Το τελευταίο bit είναι "0" όταν η παρεχόμενη διεύθυνση είναι Φυσική Διεύθυνση, αλλιώς το τελευταίο bit είναι "1" και σημαίνει ότι η παρεχόμενη διεύθυνση είναι Διεύθυνση Ομάδος.

Δεξιά του Target Address, υπάρχει το πεδίο Routing Counter (Μετρητής Δρομολόγησης) το οποίο δείχνει εάν επιτρέπεται η μετάδοση ή όχι.

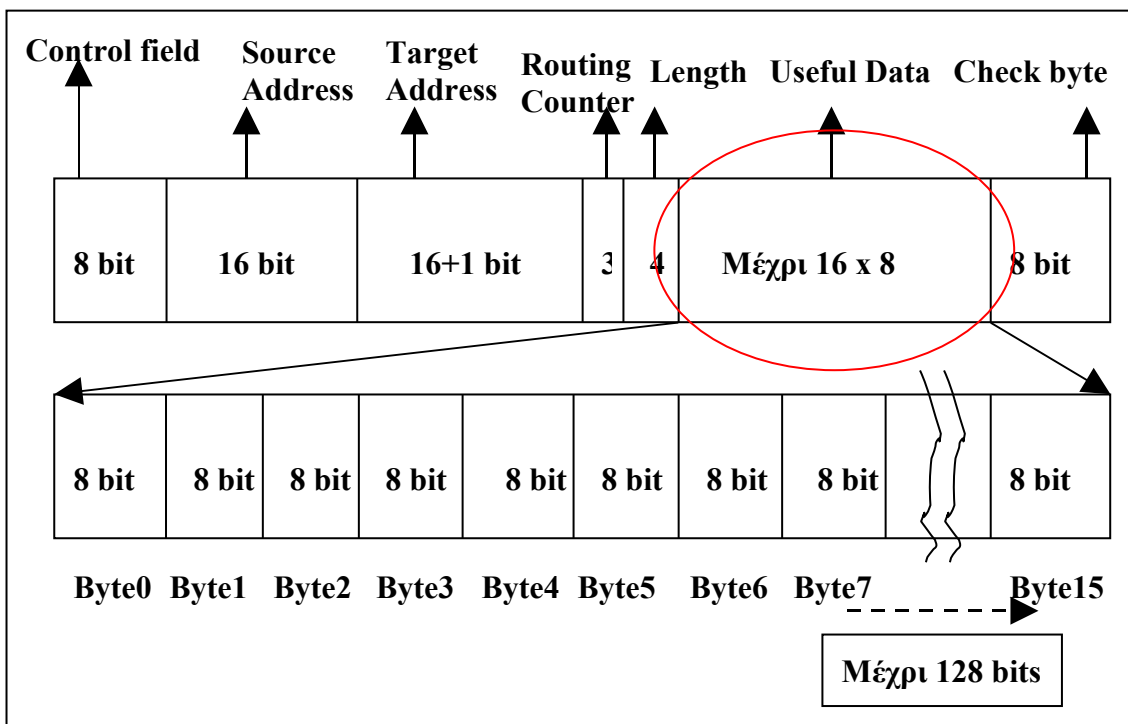




Ανάλυση του πεδίου Routing Counter σε bit

Ακολουθεί το πεδίο μήκους (Length) το οποίο δείχνει τον αριθμό των bytes που υπάρχουν στο πεδίο δεδομένων.

Το πεδίο δεδομένων (Useful Data) είναι μέχρι και 16 bytes και εξαρτάται από τον τύπο των δεδομένων, με τρόπο που έχει οριστεί από το διεθνές πρότυπο της EIS (EIBA Interworking Standard).

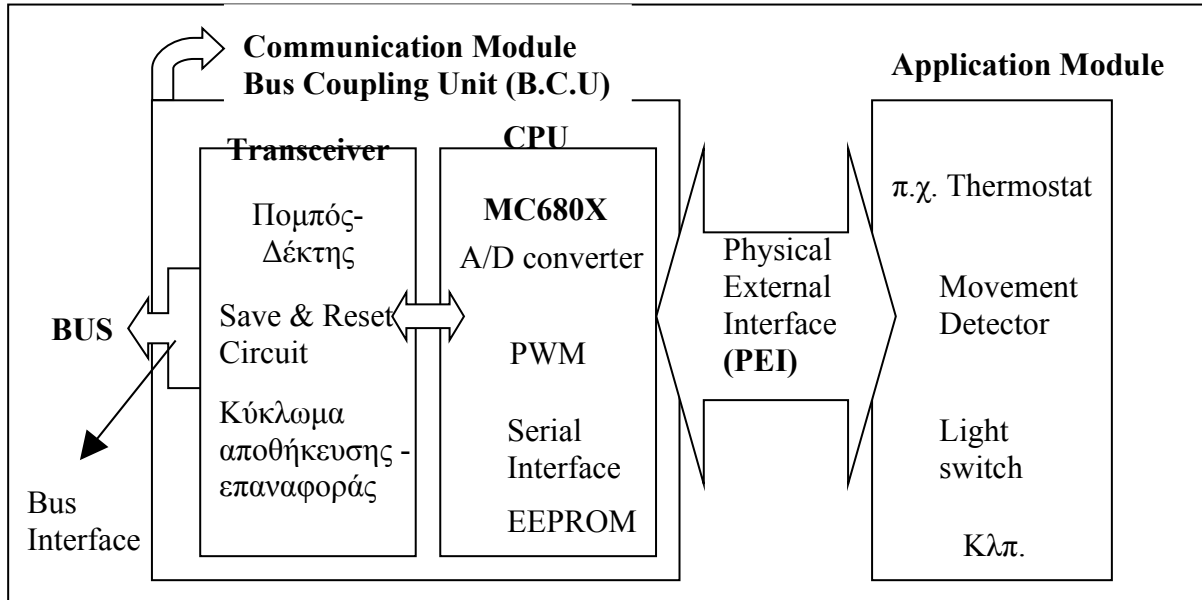


Ανάλυση σε bits & bytes του πεδίου δεδομένων

Στο τέλος του τηλεγραφήματος βρίσκεται το byte ελέγχου (Check byte). Αυτό χρησιμοποιείται για να εντοπίζει τα σφάλματα κατά τη μετάδοση των byte στο πεδίο Useful Data. Χρησιμοποιείται μια τεχνική ισοτιμίας (parity) για τον εντοπισμό σφαλμάτων στη μετάδοση ψηφιακής πληροφορίας, που περιλαμβάνει δύο αθροίσματα και αυτή λέγεται cross-check.

3.4.2.4 Η φυσική δομή ενός EIB συστήματος.

Κάθε EIB συσκευή είναι κατασκευασμένη ώστε να μπορεί να επικοινωνεί με τις υπόλοιπες. Αναλυτικότερα, κάθε συσκευή αποτελείται από δύο υπομονάδες, αυτήν της επικοινωνίας και αυτήν της εφαρμογής.



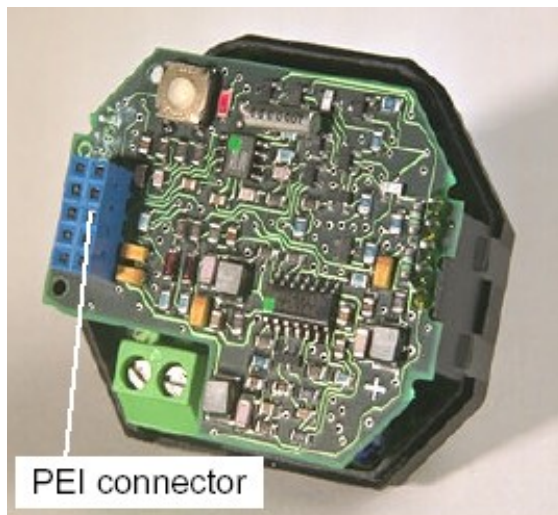
Σχέδιο που αναπαριστά τις υπομονάδες μιας οποιασδήποτε EIB συσκευής, που λειτουργεί σε σύστημα με μέσο μετάδοσης το twisted pair

Η υπομονάδα της επικοινωνίας, αποτελείται με τη σειρά της από έναν πομποδέκτη (Transceiver) και από ένα μικροελεγκτή (microcontroller). Τα μέρη που αποτελούν την υπομονάδα της εφαρμογής, εξαρτώνται από την ίδια την εφαρμογή και την πολυπλοκότητα της. Κάποιες παράμετροι που απαιτούνται για την εκτέλεση της εφαρμογής, αποθηκεύονται στο μικροελεγκτή του τμήματος επικοινωνίας και συγκεκριμένα στην EEPROM (Electrically Erasable Programmable Read-Only Memory). Σε μερικές σύνθετες εφαρμογές το τμήμα εφαρμογής έχει το δικό του μικροελεγκτή και οι απαιτούμενοι παράγοντες πια αποθηκεύονται σε αυτόν και όχι στο τμήμα επικοινωνίας.



*Ο microcontroller στην Υπομονάδα Επικοινωνίας
(στον κόκκινο κύκλο)*

Το μέσον αλληλεπίδρασης (Interface) μεταξύ των δύο προαναφερθέντων υπομονάδων, λέγεται «μέσον αλληλεπίδρασης εξωτερικής διασύνδεσης» (Physical External Interface ή εν συντομία PEI). Στην πράξη, το PEI έχει έναν κονέκτορα με δέκα ακροδέκτες (pins), ο οποίος φαίνεται στην πλακέτα μιας EIB συσκευής στο παρακάτω σχήμα.



Ο 10-pin PEI κονέκτορας

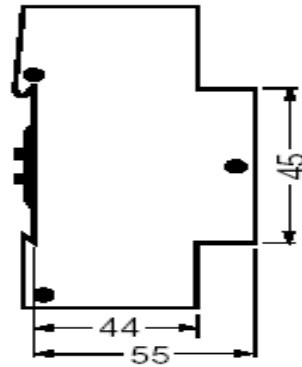
Ο ρόλος του κάθε ακροδέκτη φαίνεται ονομαστικά στο επόμενο σχήμα.

+5Volt 5	I/O 3 4	I/O 1 3	I/O 2 2	GND 1
Typ 6	I/O 4 7	+24 Volts 8	I/O 5 9	GND 10

Τα σήματα που περνούν σε κάθε ένα από τα 10 pins του PEI connector

Τα σήματα αυτά φαίνεται πως μπορεί να είναι είτε δεδομένα είτε τροφοδοσία και αξίζει να ειπωθεί ότι η μεταφορά των σημάτων γίνεται και από και προς την υπομονάδα εφαρμογής.

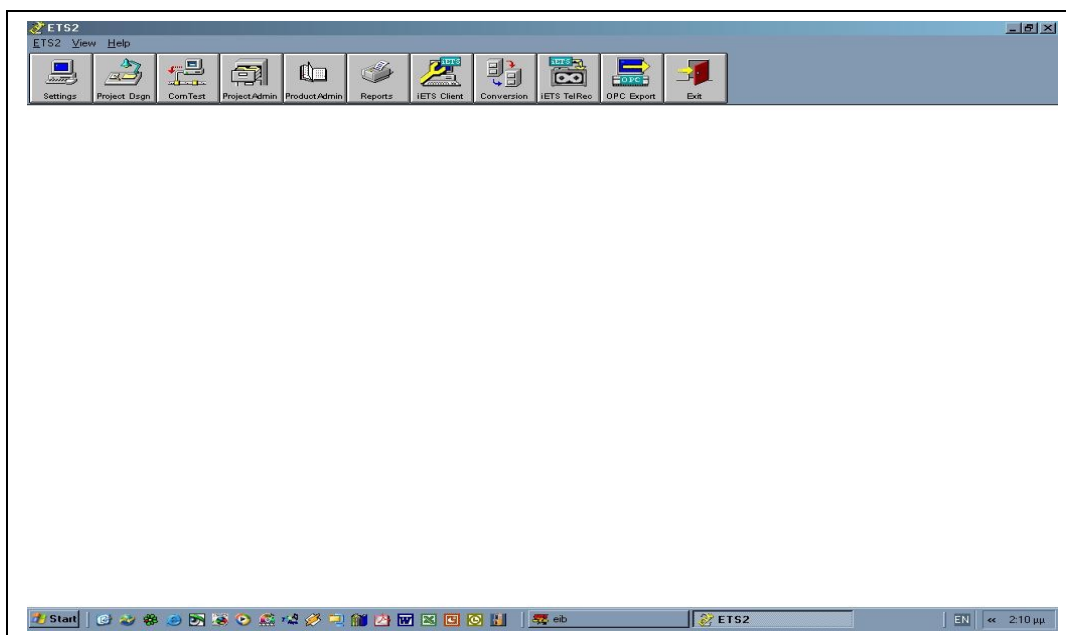
Για να είναι αναγνωρίσιμες οι συσκευές EIB, σημειώνεται ότι διατίθενται σε γνωστά και εύχρηστα περιβλήματα για την καλύτερη τοποθέτησή τους. Αυτά είναι, η θήκη τύπου N και η θήκη τύπου UP (χωνευτού τύπου). Ο πρώτος τύπος μπορεί να τοποθετηθεί σε ένα κοινό ηλεκτρολογικό πίνακα και ο δεύτερος στις κάθετες επιφάνειες ηλεκτρολογικών κουτιών. Στο παρακάτω σχήμα φαίνεται μια EIB συσκευή τύπου N.



Συσκευή EIB τύπου N

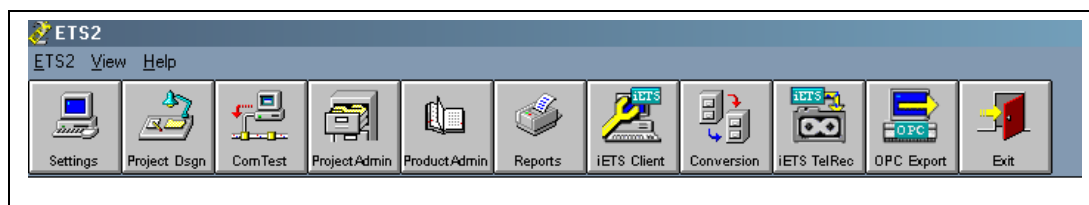
3.4.2.5 Το EIB Tool Software (ETS).

Το ETS είναι ένα λογισμικό (πρόγραμμα) στον ηλεκτρονικό υπολογιστή που εξυπηρετεί τα συστήματα EIB σε σκοπούς σχεδιασμού και διάγνωσης. Λειτουργεί κάτω από το περιβάλλον των Microsoft Windows και μπορεί να δεχτεί βάση δεδομένων οποιασδήποτε κατασκευαστικής εταιρίας.



Το αρχικό περιβάλλον του ETS

Από την εικόνα φαίνεται ότι πρόκειται για ένα GUI, γεγονός που αυτόματα το καθιστά σύγχρονο και εύχρηστο. Το πρόγραμμα επικοινωνεί με την εγκατάσταση μέσω της σειριακής θύρας (RS-232) του υπολογιστή. Στο επάνω μέρος του έχει σχεδιασμένα κάποια κουμπιά (buttons) που εκτελούν τις βασικές λειτουργίες του. Παρακάτω γίνεται αναφορά σε κάποια από αυτά ώστε να δοθεί μια εικόνα των δυνατοτήτων του ETS.



Τα εικονικά κουμπιά για τις βασικές λειτουργίες του ETS

Το πρώτο από αριστερά είναι οι ρυθμίσεις (settings) και επιτρέπει να γίνουν διάφορες επιλογές όπως η επιλογή της σειριακής θύρας με την οποία θα γίνει η επικοινωνία, το πλήθος των επιπέδων στις διευθύνσεις ομάδας (group addresses), η επιλογή του εκτυπωτή, ο ορισμός μυστικού κωδικού κ.α.



Οι επιλογές που περιέχονται στη λειτουργία «Ρυθμίσεις»

Δεξιά του settings βρίσκεται η λειτουργία Project Design (Σχέδιο Έργου). Με αυτό το κουμπί ξεκινά η διαδικασία σχεδίασης ενός συστήματος ή τροποποίησης κάποιου ήδη υπάρχοντος. Σε αυτό το κομμάτι γίνονται οι βασικές αρχικοποιήσεις όπως είναι η διευθυνσιοδότηση, η παραμετροποίηση της επικοινωνίας και ο σχεδιασμός της τοπολογίας. Ένα πλεονέκτημα που δίνει το πρόγραμμα στο σχεδιαστή είναι ότι αυτός μπορεί να σχεδιάσει το σύστημα μακριά από την πραγματική εγκατάσταση.

Το επόμενο κουμπί λέγεται Com Test (Εξέταση Επικοινωνίας) και πρόκειται για ένα διαγνωστικό εργαλείο που χρησιμοποιείται στο πέρας το σχεδιασμού και της εγκατάστασης. Πιο συγκεκριμένα, με αυτό ο σχεδιαστής μπορεί να χειριστεί το σύστημα ενεργοποιώντας και απενεργοποιώντας συσκευές και να επικοινωνήσει γενικότερα με οποιαδήποτε συσκευή για να ελέγξει έτσι τις παραμέτρους τη επικοινωνίας και τη διευθυνσιοδότηση.

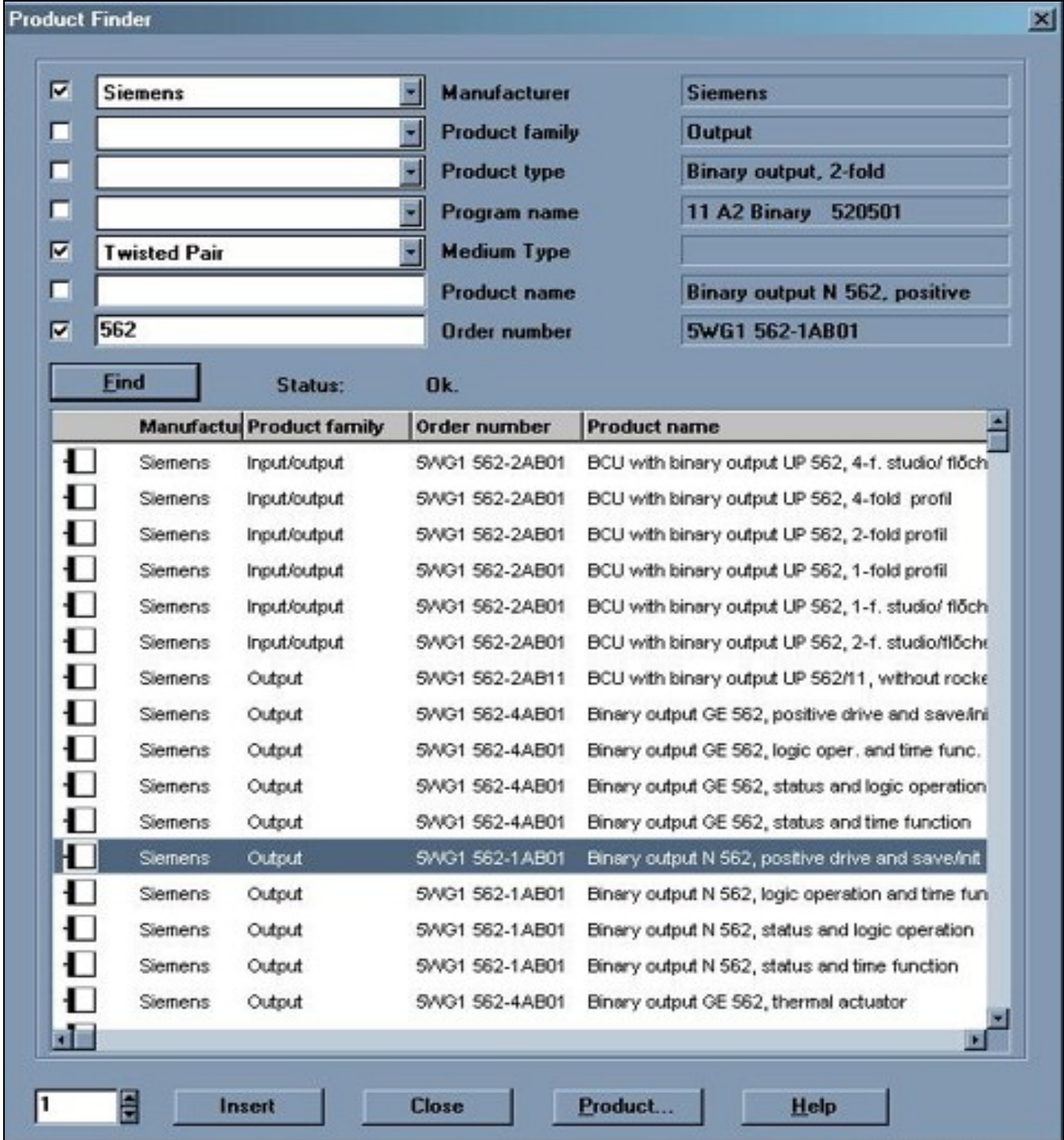
Στα δεξιά του Com Test (Communication Test), υπάρχει το Project Admin (Διαχείριση Έργου). Αυτό είναι ένα εργαλείο που εξάγει το έργο, είτε για αποθήκευση ή για μεταφορά ενώ επίσης δίνει τη δυνατότητα για χωρισμό του project σε τμήματα και το αντίστροφο.

Ακολουθεί το Product Admin (Διαχείριση Προϊόντων) που κάνει την ενημέρωση του ETS σχετικά με τα τεχνικά χαρακτηριστικά διαφόρων EIB συσκευών. Οποιοσδήποτε κατασκευαστής EIB προϊόντων δίνει μια βάση δεδομένων αναγνωρίσιμη από το εν λόγω λογισμικό σαν ένα δέντρο αποτελούμενο από τις συσκευές. Για λόγους καλύτερης κατανόησης και ανάδειξης της χρησιμότητας του λογισμικού, ακολουθεί ένα παράδειγμα σχεδιασμού ενός απλού συστήματος EIB.

Έστω ότι θέλουμε να ελέγξουμε τρία φωτιστικά σώματα (λαμπτήρες πυρακτώσεως), χρησιμοποιώντας 2 φωτιστικά σώματα 230V για απλό έλεγχο on-off τα L1 και L2 και ένα φωτιστικό σώμα 230V για έλεγχο on-off και επιπλέον dimming το L3. Οι απαιτήσεις χειρισμού είναι το κάθε φωτιστικό να ελέγχεται ανεξάρτητα από ένα μπουτόν (3 πλήκτρα συνολικά) και να υπάρχει η δυνατότητα γενικού on-off. Έτσι λοιπόν οι bus συσκευές που θα χρησιμοποιηθούν είναι μία διπλή δυαδική έξοδος 10A με είσοδο για μπουτόν (PEI), ένα dimmer 250W με

είσοδο για μπουτόν, 2 διπλά μπουτόν, ένα bus τροφοδοτικό με πηνίο, ράγα δεδομένων, συνδετήρας bus,σειριακή θύρα για επικοινωνία με τον υπολογιστή. Πρώτο βήμα είναι να ορίσουμε κάποιο χώρο στον οποίο θα γίνεται εισαγωγή των συσκευών. Αμέσως μετά με drag 'n' drop προσθέτουμε τις συσκευές εκείνες που θα χρησιμοποιηθούν στο έργο μας, τις οποίες εντοπίζουμε στις εισαχθείσες βάσεις δεδομένων. Στην συγκεκριμένη εφαρμογή θα χρησιμοποιηθούν συσκευές της Siemens-EIB. Έχουμε λοιπόν την διπλή δυαδική έξοδο με κωδικό 5wg1 562-2ab01, το dimmer με κωδικό 5wg1 525-2ab01 και την RS-232 κωδικό 5wg1 148-1ab-1ab02. Στο παρακάτω σχήμα φαίνεται η λίστα στην οποία βρίσκουμε τα προϊόντα που χρησιμοποιούμε με τα στοιχεία που τα συνοδεύουν όπως είναι ο «κωδικός» τους.

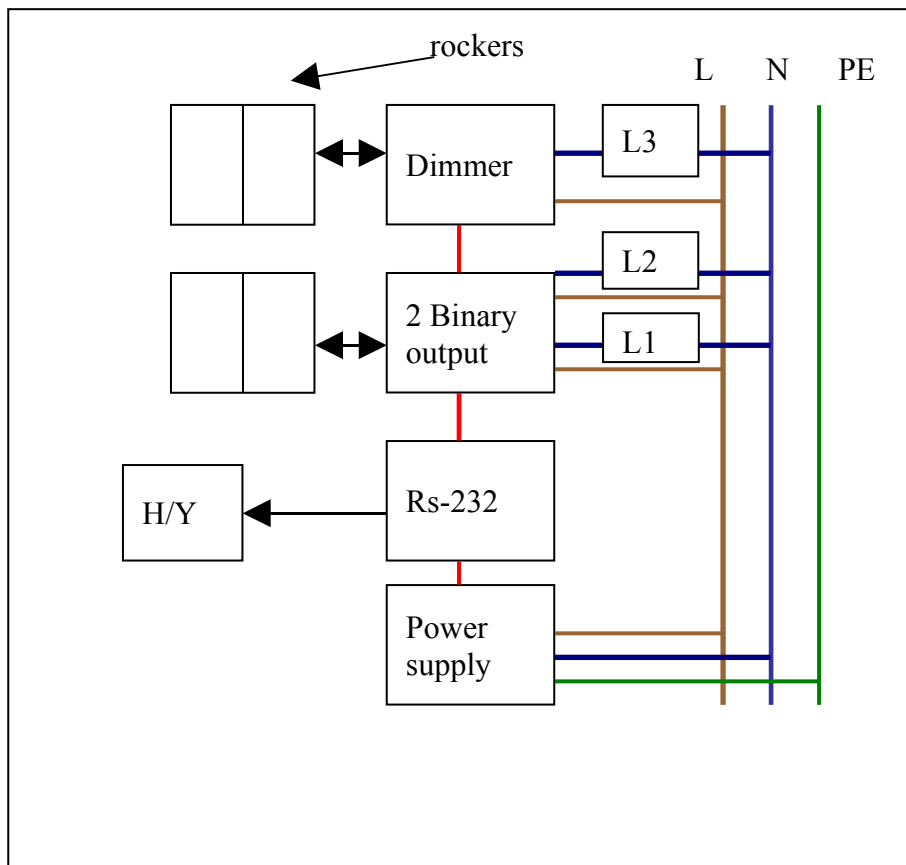
Το πεδίο από όπου εισάγουμε μια EIB συσκευή στο έργο



Κατόπιν δημιουργούμε τις διευθύνσεις ομάδας (group addresses) με βάση τις απαιτήσεις χειρισμού που υπάρχουν. Αυτές εδώ θα είναι, L1 on-off, L2 on-off, L3 on-off, L3 dimming, General on-off. Στη λίστα των επιλεγμένων συσκευών τσεκάρουμε την επιλογή show objects ώστε να φανούν τα στοιχεία επικοινωνίας των συσκευών. Στην συνέχεια θα επιλέξουμε εκείνα τα στοιχεία επικοινωνίας που θέλουμε να συνδέσουμε με την διεύθυνση ομάδος που πρέπει. Για παράδειγμα στην συσκευή dimmer επιλέγουμε το on/off object και θα το συνδέσουμε τόσο στο group address General on-off ,όσο και στο L3 on-off. Ακόμη χρειάζεται να συνδέσουμε μερικά επιπλέον objects με κάποιες διευθύνσεις ομάδος. Αυτά είναι το object dimming που πρέπει να το συνδέσουμε στο L3 dimming και τα objects των button (δεξί, αριστερό) up και down στα ανάλογα στοιχεία επικοινωνίας. Στην συνέχεια πρέπει να γίνει η μεταφορά του προγράμματος στις bus- συσκευές.

Η μεταφορά αυτή μπορεί να γίνει αφού το bus τροφοδοτείται με τάση, έχουν συνδεθεί τα φωτιστικά και υπάρχει σύνδεση με τον H/Y. Μια άποψη για το πώς

είναι όλα αυτά συνδεδεμένα φαίνονται στο παρακάτω σχήμα.

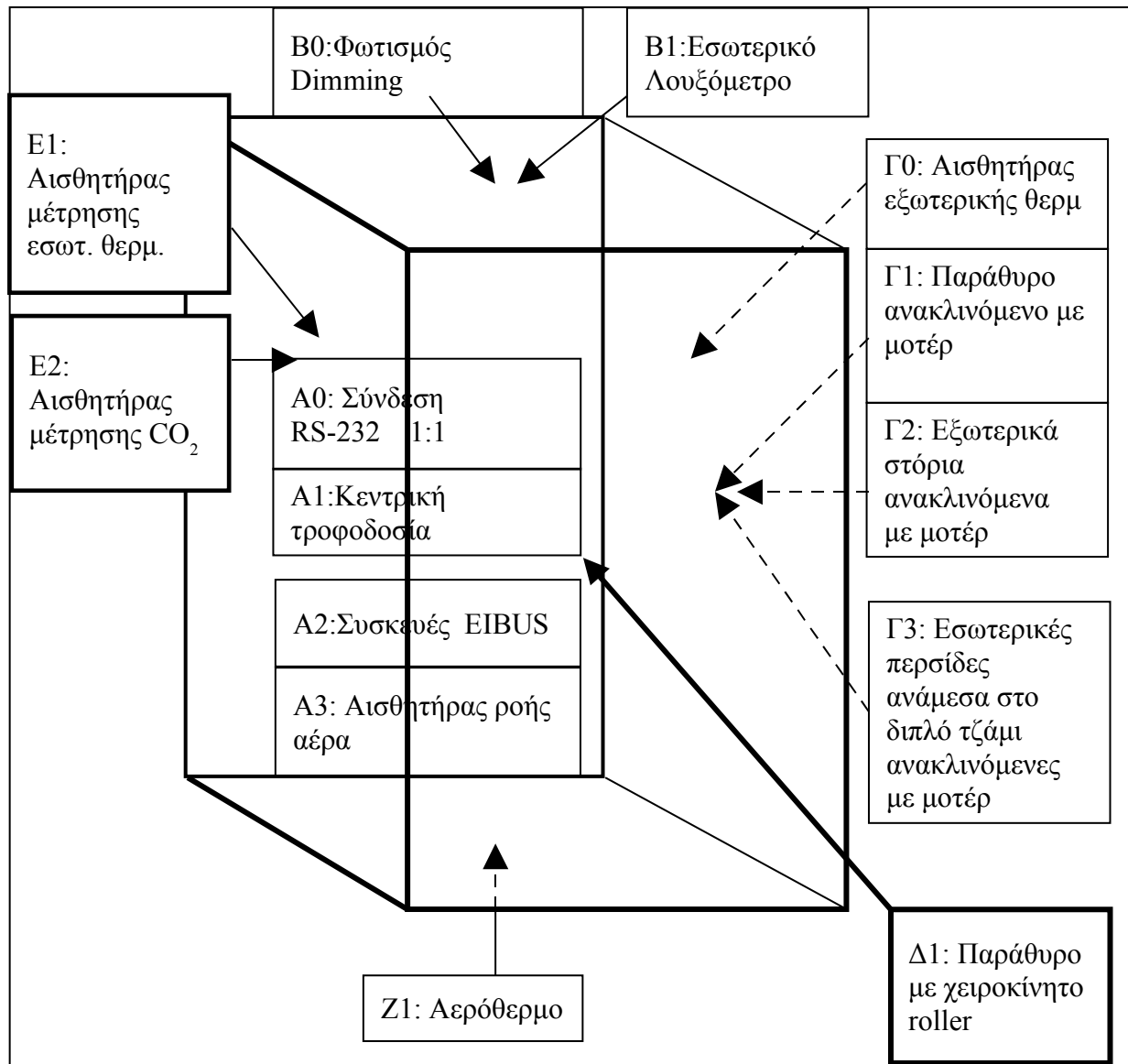


Η τοπολογία στο παράδειγμα που αναλύθηκε

3.4.3 Αναλυτική ματιά στον πειραματικό θάλαμο.

Στην παράγραφο 3.4.1 έγινε μια γενική αναφορά στον πειραματικό θάλαμο που υπάρχει στο Τ.Ε.Ι Ηλεκτρονικής στα Χανιά. Σε αυτή την παράγραφο θα γίνει λεπτομερέστερη αναφορά σε αυτήν την, εκπαιδευτικού χαρακτήρα, υλοποίηση.

Επιφανειακά, πρόκειται για μια ορθογώνια παραλληλεπίπεδη κατασκευή, διαστάσεων 1m μήκος x 1m πλάτος x 2m ύψος, κατασκευασμένη από αλουμίνιο, της οποίας οι έδρες αποτελούν ανεξάρτητα κομμάτια που συναρμολογούνται και αποσπώνται όταν αυτό απαιτείται. Το αλουμίνιο ως ελαφρύ υλικό και το γεγονός ότι ο θάλαμος αποσυναρμολογείται εξυπηρετούν ένα βασικό σκοπό για τον οποίο κατασκευάστηκε, που είναι η εύκολη μεταφορά για την παρουσίαση & επίδειξη του σε διάφορους χώρους. Παρακάτω φαίνεται ένα σχέδιο του θαλάμου με ονοματισμένα τμήματα που βοηθά στην αναφορά των τμημάτων αυτών.



Σχέδιο του θαλάμου

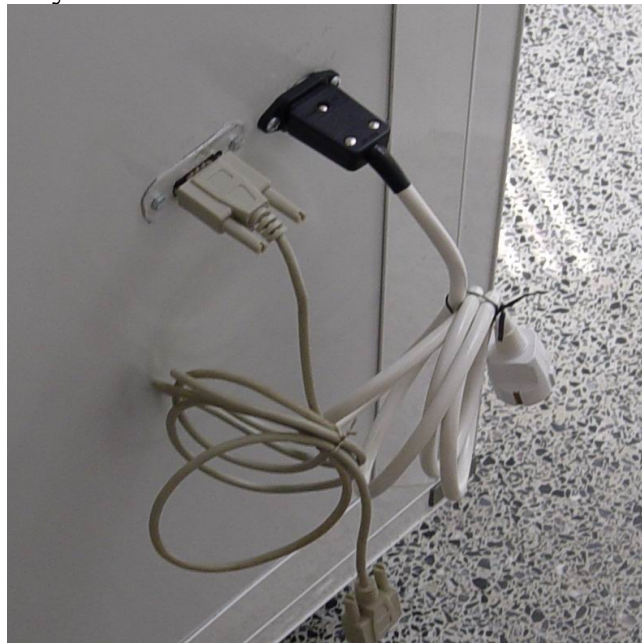
Η βάση του, είναι αρκετά ανθεκτική ώστε να αντέχει έως και το βάρος ενός μεγαλόσωμου ανθρώπου και κάτω από αυτή, στις τέσσερις γωνίες της έδρας, υπάρχουν ρόδες εξίσου ανθεκτικές με τα απαραίτητα φρένα τους. Στην πρόσοψη του υπάρχει μια αλουμινένια πόρτα που δε διαφέρει από τις πόρτες που συναντά κανείς στα μπαλκόνια των σπιτιών. Συγκεκριμένα το επάνω μισό μέρος της πόρτας είναι από τζάμι και στο επάνω άκρο της έχει τοποθετηθεί ένα χειροκίνητο ρόλερ με αλυσίδα για να μεταβάλλεται εύκολα το επίπεδο του φυσικού φωτός που μπαίνει στο θάλαμο. Η εξωτερική πλευρά της έδρας Γ είναι επίσης υπεύθυνη για το επίπεδο του φυσικού φωτισμού μέσα στο θάλαμο. Εκεί έχει εγκατασταθεί ένα ανακλινόμενο παράθυρο με ηλεκτρικό μοτέρ A.C. Στο εξωτερικό μέρος του παραθύρου υπάρχουν μεταλλικά στόρια επίσης ανακλινόμενα με μοτέρ A.C. Το παράθυρο είναι θερμικά μονωμένο με διπλά κρύσταλλα και ανάμεσα σε αυτά είναι τοποθετημένες περσίδες οι οποίες και πάλι ελέγχονται από ένα ηλεκτρικό μοτέρ A.C. Φαίνεται λοιπόν ότι το παράθυρο αυτό ελέγχει το φυσικό φωτισμό με αυτόματο τρόπο και μάλιστα με δύο μηχανισμούς διαφορετικής αποτελεσματικότητας. Τα μεταλλικά στόρια μπορούν να σκιάσουν σε μεγάλο βαθμό το χώρο του θαλάμου ενώ οι περσίδες σε πολύ μικρότερο, έτσι ο συνδυασμός τους πετυχαίνει ακριβέστερες συνθήκες φυσικού φωτισμού.

Ακολουθεί μια εικόνα του θαλάμου στην οποία φαίνονται η πρόσοψη και η πλευρά με το «αυτόματο» παράθυρο.



Εξωτερική όψη του πειραματικού θαλάμου

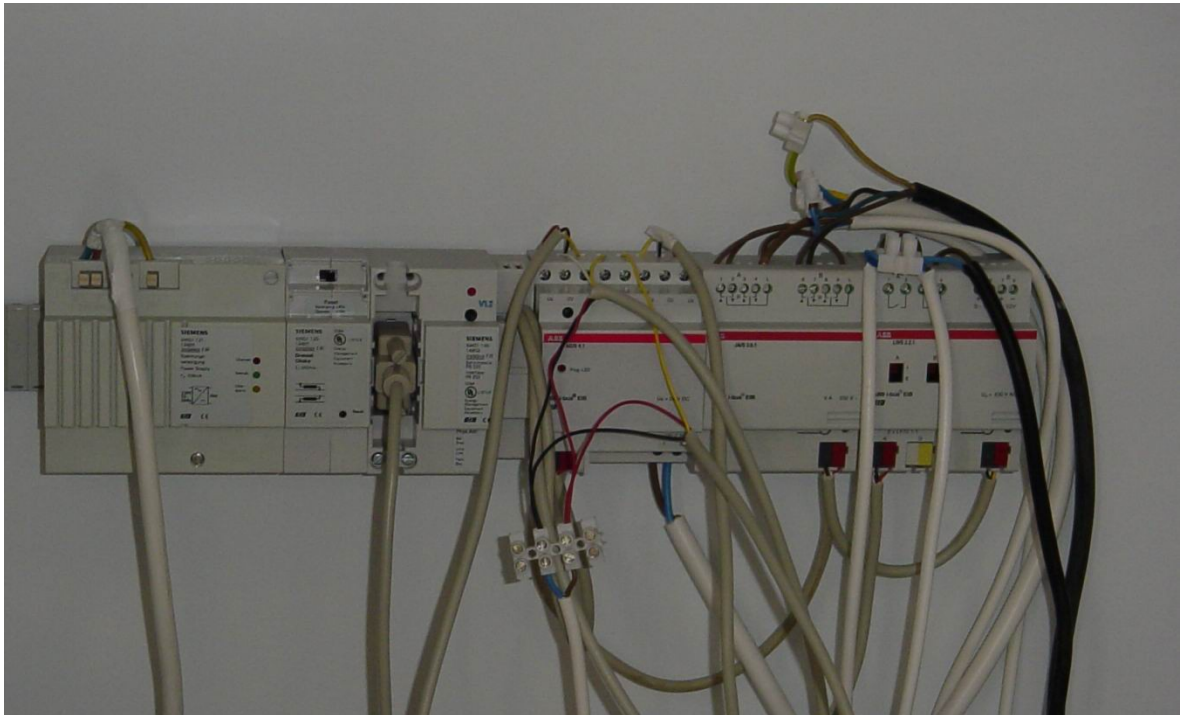
Κοντά στα μεταλλικά στόρια υπάρχει ένας αισθητήρας εξωτερικής θερμοκρασίας. Εξωτερικά, στο πίσω μέρος του θαλάμου υπάρχει υποδοχή για τροφοδοσία από το ηλεκτρικό δίκτυο και μια σειριακή θύρα (RS-232) για τη σύνδεση με ηλεκτρονικό υπολογιστή (για προγραμματισμό, έλεγχο & παρακολούθηση). Ακολουθεί η εικόνα του πίσω μέρους.



Τροφοδοσία και Σειριακή θύρα στο πίσω μέρος του θαλάμου

Εσωτερικά βρίσκονται κάποιες ακόμα συσκευές που συναντώνται σε όλα τα σπίτια καθώς και οι απαραίτητοι για τον αυτοματισμό, αισθητήρες. Επάνω στη βάση υπήρχε αρχικά ένα αερόθερμο, όμως αργότερα αντικαταστάθηκε από ένα κλιματιστικό που εγκαταστάθηκε ψηλά στο πίσω μέρος του θαλάμου. Στην οροφή υπάρχει μια λάμπα που οδηγείται από ένα Dimmer (ρυθμιστής έντασης) και με αυτό τον τρόπο ελέγχεται και ο τεχνητός φωτισμός του χώρου. Ο αισθητήρας του φωτισμού (λουξόμετρο) είναι επίσης τοποθετημένος στην οροφή. Στην αριστερή πλευρά του θαλάμου υπάρχουν ακόμα δύο αισθητήρες, αυτός της θερμοκρασίας και αυτός του διοξειδίου του άνθρακα (CO₂). Στο πίσω μέρος εσωτερικά του θαλάμου, υπάρχει ακόμα ο αισθητήρας μέτρησης της ροής του αέρα (airflow sensor), του οποίου οι ενδείξεις καθορίζουν το άνοιγμα και το κλείσιμο του αυτόματου παραθύρου.

Στο παρακάτω σχήμα φαίνεται τοποθετημένη η «ψυχή» του αυτοματισμού, το σύστημα EIB.



Το σύστημα EIB τοποθετημένο στην εσωτερική πλάτη του θαλάμου

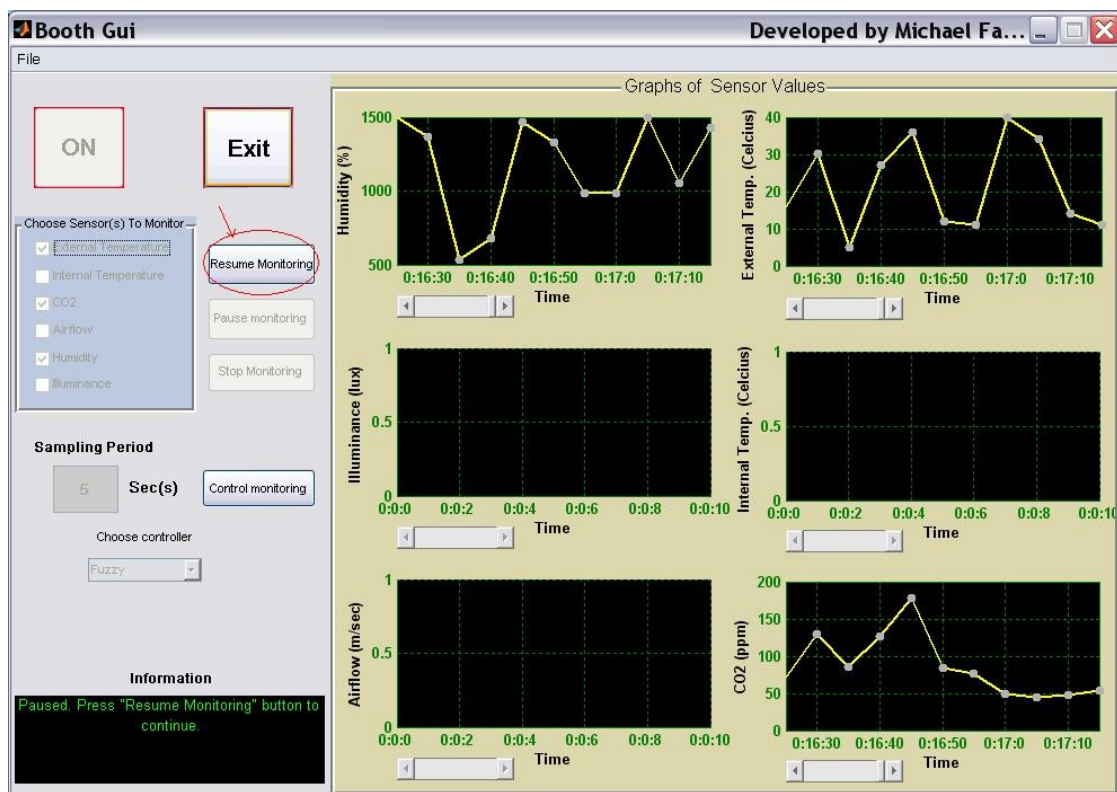
Το σύστημα που φαίνεται στην εικόνα αποτελείται από διάφορα τμήματα υπεύθυνα για διαφορετικές λειτουργίες. Ένα από αυτά τα κουτιά είναι το τροφοδοτικό, που παίρνει από την πρίζα 220V εναλλασσομένου ρεύματος και το μετατρέπει σε +24V και -24V που απαιτεί το υπόλοιπο σύστημα για να λειτουργήσει. Κατόπιν σε σειρά συνδέεται η συσκευή EIBUS της Siemens N-148. Η συσκευή αυτή παίζει τον ρόλο της ένωσης του bus με τον ηλεκτρονικό υπολογιστή τόσο κατά την φάση του προγραμματισμού των συσκευών και των ιδιοτήτων τους από το ETS, όσο και κατά την διάρκεια λειτουργίας του συστήματος με κεντρικό έλεγχο από υπολογιστή. Σε σειρά συνδέεται μία συσκευή της ABB (4 fold Analog Input, MDRC) mod AE/S 4.1. Αυτή η συσκευή έχει 4 εισόδους για συλλογή και επεξεργασία διαφορετικών αναλογικών τιμών ρεύματος 0-20 mA ή τάσης 0-1V, 0-5V, 0-10V. Η συσκευή αυτή δεν τροφοδοτείται μόνο από το ίδιο το Bus αλλά χρειάζεται και εξωτερική τροφοδότηση 220V. Αυτό γιατί η

συσκευή αυτή παράγει 24 Volts DC για την τροφοδότηση εξωτερικών αισθητηρίων που χρειάζονται αυτήν την τιμή τάσης για να λειτουργήσουν. Στην συνέχεια σε σειρά συνδέεται μια συσκευή της ABB (Shutter Actuator 2fold, MDRC) mod JA/S 2.6.1. Σε αυτή την συσκευή υπάρχουν 2 εξόδοι οι οποίες προορίζονται για συσκευές που δέχονται μία τάση σε μία είσοδό τους για να τεθούν σε μία κατάσταση A ενώ αν δεχτούν αυτήν την τάση στην άλλη τους είσοδο τίθενται στην κατάσταση B. Μια τέτοια συσκευή που χρησιμοποιήθηκε στον θάλαμο, είναι το μοτέρ για την ανάκλιση του παραθύρου ενώ μια άλλη είναι το μοτέρ για τα εξωτερικά στόρια. Άλλη μια συσκευή EIBUS είναι ο light controller (switch/dim MDRC) mod LR/S 2.2.1 η οποία προορίζεται για τον φωτισμό του θαλάμου εξασφαλίζοντας dimming σε λαμπτήρες φθορισμού. Επιπρόσθετα στην συσκευή αυτή συνδέεται μια ακόμη συσκευή Bus. Αυτή είναι ο light sensor LF/U 1.1 που διαβάζει την φωτεινότητα μέσα στον εσωτερικό χώρο του θαλάμου. Τέλος άλλη μια συσκευή bus που χρησιμοποιήθηκε είναι ένας sensor θερμοκρασίας της Siemens (5WG1- 256-1AB01). Αυτός ο αισθητήρας θερμοκρασίας δεν χρειάζεται να συνδεθεί με την 4πλή αναλογική είσοδο που προαναφέρθηκε διότι μεταφέρει τα αποτελέσματα της θερμοκρασίας αμέσως σε σήματα κατανοητά για το bus, σε αντίθεση με τους άλλους αισθητήρες άλλων κατασκευαστών που για να γίνουν κατανοητά τα σήματά τους χρειάζονται «μετάφραση» από την 4πλή αναλογική είσοδο.

3.4.4 Αναφορά στο λογισμικό παρακολούθησης και ελέγχου του πειραματικού θαλάμου.

Αναφέρθηκε έως τώρα η δυνατότητα που παρέχει το σύστημα να διασυνδεθεί με H/Y μέσω μια σειριακής θύρας και μια κατάλληλης για την επικοινωνία συσκευής. Στην οθόνη του ηλεκτρονικού υπολογιστή, το εργαλείο του ανθρώπου για την επικοινωνία είναι ένα γραφικό περιβάλλον αλληλεπίδρασης, φτιαγμένο αποκλειστικά για να επικοινωνεί με το συγκεκριμένο θάλαμο.

Το περιβάλλον αυτό είναι προϊόν μιας σύγχρονης γλώσσας προγραμματισμού, την Matlab. Η Matlab ειδικεύεται σε μαθηματικές εφαρμογές αφού είναι δομημένη να χρησιμοποιεί κατά βάση αριθμητικούς πίνακες και μια πληθώρα μαθηματικών συναρτήσεων. Στα πλαίσια των κοινώς γνωστών δυνατοτήτων της είναι η δημιουργία γραφικών παραστάσεων, η εξομοίωση σειρών παραγωγής ή επεξεργασίας και ο σχεδιασμός γραφικών περιβαλλόντων αλληλεπίδρασης. Παρακάτω φαίνεται σε λειτουργία το περιβάλλον διασύνδεσης σε Matlab που παρακολουθεί τους αισθητήρες του πειραματικού θαλάμου σε πραγματικό χρόνο.



To Booth Gui του πειραματικού θαλάμου

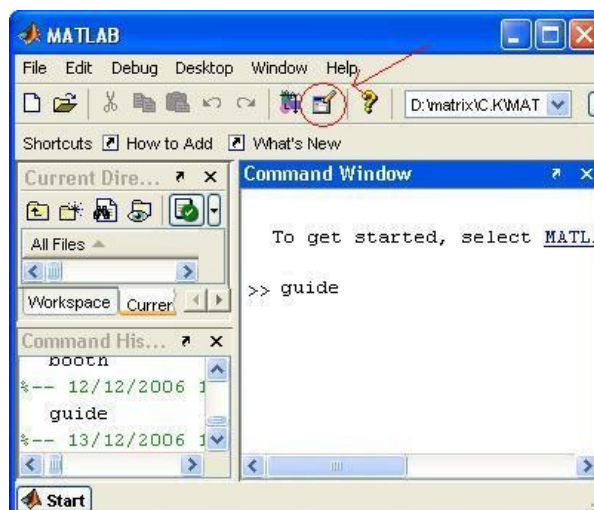
Με την πρώτη ματιά φαίνεται ότι πρόκειται για ένα πολύ φιλικό πρόγραμμα για οποιοδήποτε χρήστη, αφού έχει λίγα και μεγάλα κουμπιά, πλαίσιο με επεξηγήσεις στην κάτω αριστερή γωνία και ευανάγνωστους άξονες που καταλαμβάνουν περίπου τα $\frac{3}{4}$ της συνολικής φόρμας. Ένα από αυτά τα κουμπιά εμφανίζει ένα άλλο παράθυρο που αναφέρει την κατάσταση των υπόλοιπων συσκευών πέρα από τους αισθητήρες. Στις επόμενες παραγράφους θα περιγραφεί και θα εξηγηθεί αναλυτικά το εν λόγω πρόγραμμα που δημιουργήθηκε ως πτυχιακή εργασία και ως σημαντικό κομμάτι για την επίδειξη του πειραματικού θαλάμου, αφού επιτρέπει στο θεατή να δει με μια ματιά τις συνθήκες που επικρατούν μέσα στο θάλαμο.

4 Ανάπτυξη του λογισμικού παρακολούθησης και ελέγχου του πειραματικού θαλάμου.

4.1 Ανάπτυξη οποιουδήποτε Gui με τη Matlab (Εργαλεία).

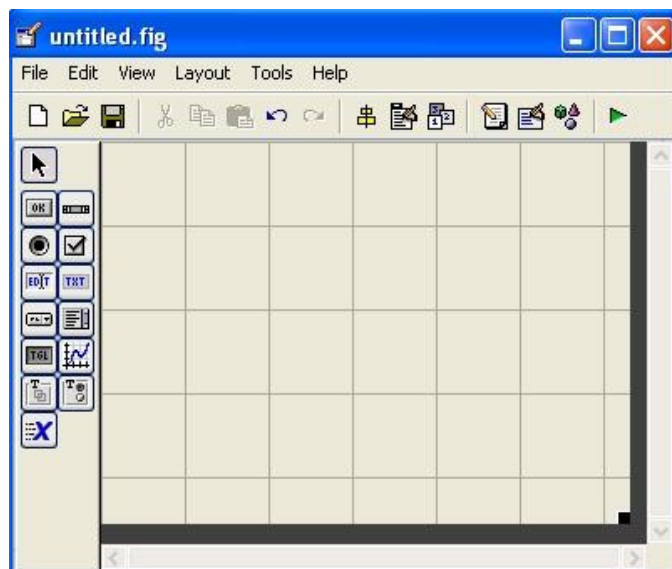
Όπως αναφέρθηκε στην προηγούμενη παράγραφο, για τον έλεγχο και την παρατήρηση του πειραματικού θαλάμου χρησιμοποιείται ένα προϊόν της μαθηματικής γλώσσας προγραμματισμού Matlab. Με τη χρήση διαφόρων εργαλείων, το προϊόν αυτό συνδυάζει τη γραφική αναπαράσταση εξωτερικά, αφού τα αποτελέσματα των εργασιών του είναι γραφικά αντικείμενα καθώς και την χρήση εντολών ή στην περίπτωση μας, κυρίως, συναρτήσεων που αποτελούν τον πυρήνα προγράμματος, αφού αυτές είναι υπεύθυνες τόσο για τους υπολογισμούς και τον έλεγχο της ροής όσο και για τον τελικό σχεδιασμό του οπτικού αποτελέσματος (GUI). Η ανάπτυξη ενός GUI αποτελείται από δύο κύρια στάδια: α) Το σχεδιασμό του «παραθύρου» τοποθετώντας στοιχεία που λειτουργούν ως είσοδοι και άλλα ως έξοδοι και καθορίζοντας τη θέση, το μέγεθος το χρώμα και πολλά άλλα χαρακτηριστικά τους. β) Την ανάθεση εργασιών σε κάθε ένα στοιχείο του «παραθύρου» χρησιμοποιώντας τη βασική λειτουργία της γλώσσας που είναι η σύνταξη του προγράμματος σε περιβάλλον γραμμής εντολών.

Πιο συγκεκριμένα, η Matlab έχει ενσωματώσει στις ποικίλες λειτουργίες της, ένα σύνολο εργαλείων με το οποίο είναι δυνατός ο σχεδιασμός και προγραμματισμός ενός Graphical User Interface. Η λειτουργία αυτή είναι διαθέσιμη από το περιβάλλον της γλώσσας ως Guide. Έτσι για τη δημιουργία ενός γραφικού περιβάλλοντος αλληλεπίδρασης, με τη βοήθεια της Matlab, η διαδικασία ξεκινά με την επιλογή του εργαλείου Guide. Αυτό γίνεται εύκολα είτε πληκτρολογώντας τη λέξη `guide` στο Command Window (παραθύρο εντολών) ή επιλέγοντας με το ποντίκι το εικονίδιο που βρίσκεται μέσα σε κόκκινο κύκλο στο παρακάτω σχήμα.



Τρόποι πρόσβασης στη λειτουργία Guide της Matlab

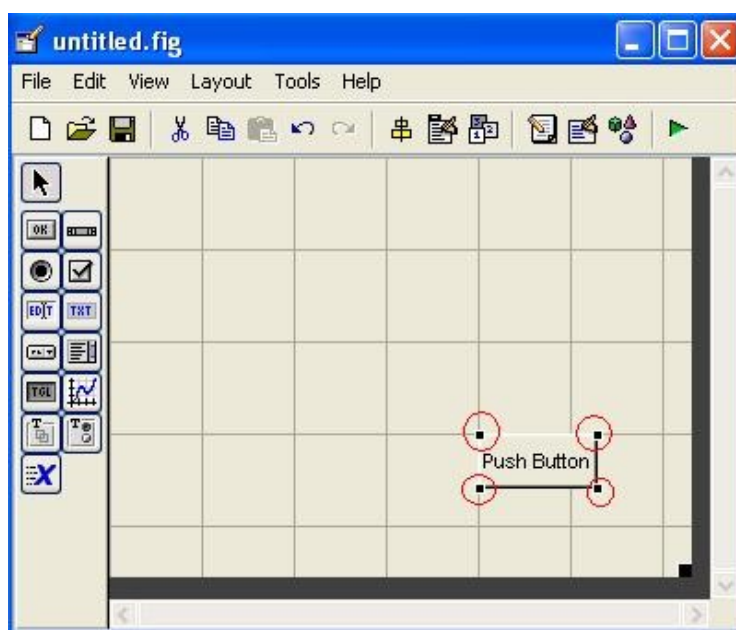
Η ενέργεια αυτή μεταφέρει το χρήστη σε μια επιφάνεια που αναπαριστά το «παραθύρο» που πρόκειται να δημιουργηθεί. Αρχικά αυτό είναι κενό και περιτριγυρίζεται από διάφορες επιλογές και επιπλέον αντικείμενα τα οποία ανάλογα με τις επιλογές του προγραμματιστή θα συνθέσουν το τελικό αποτέλεσμα. Παρακάτω φαίνεται αυτό το αρχικό στάδιο.



Το κενό GUI που φαίνεται μετά από την επιλογή του Guide

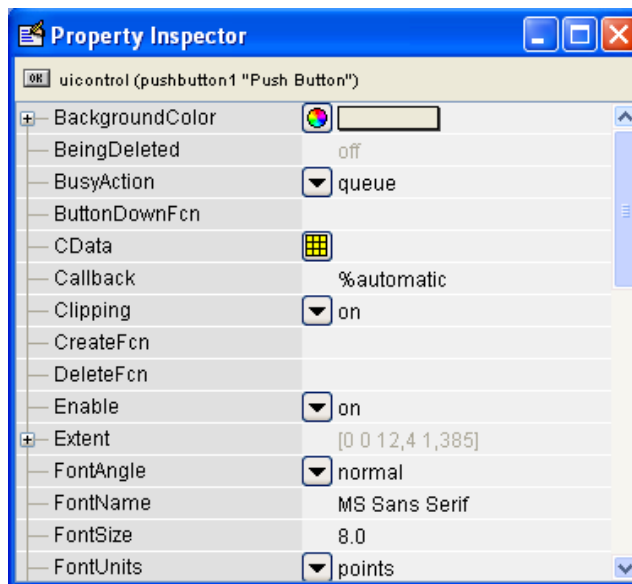
Στα αριστερά βλέπουμε τα εν λόγω αντικείμενα στα οποία θα γίνει μια, ένα προς ένα, αναφορά.

Το πρώτο από πάνω αντικείμενο, με σύμβολο τον κέρσορα ενός ποντικιού, λέγεται “Select” (επιλογή) και μας επιστρέφει στη δυνατότητα επιλογής ενός αντικειμένου που βρίσκεται πάνω στη φόρμα όταν κάνουμε κλικ σε αυτό. Αυτό χρησιμεύει για την επιλογή και περαιτέρω επεξεργασία ενός ή περισσότερων αντικειμένων, τα οποία έχουν μια πληθώρα από ιδιότητες, που θα συζητηθούν παρακάτω. Ακριβώς κάτω από το “Select” υπάρχει ίσως το βασικότερο αντικείμενο για τη λειτουργία ενός GUI. Αυτό είναι το “Push Button” (πλήκτρο) και πρόκειται για ένα εικονικό κουμπί το οποίο κατά τη λειτουργία του προγράμματος το «πατάει» ο χρήστης για να εκτελέσει την οποιαδήποτε εργασία που έχει ανατεθεί σε αυτό. Οι τυχόν εργασίες του είναι φυσικά απεριόριστες αλλά αναφορικά αυτές μπορεί να είναι από μία απλή επικύρωση για δεδομένα που εισήχθησαν έως και τον υπολογισμό πολυσύνθετων μαθηματικών συναρτήσεων και ταυτόχρονα την ανάδειξη των γραφικών τους παραστάσεων. Παρακάτω φαίνεται τοποθετημένο ένα τέτοιο πλήκτρο στη φόρμα:



Φόρμα GUI με ένα απλό Push Button

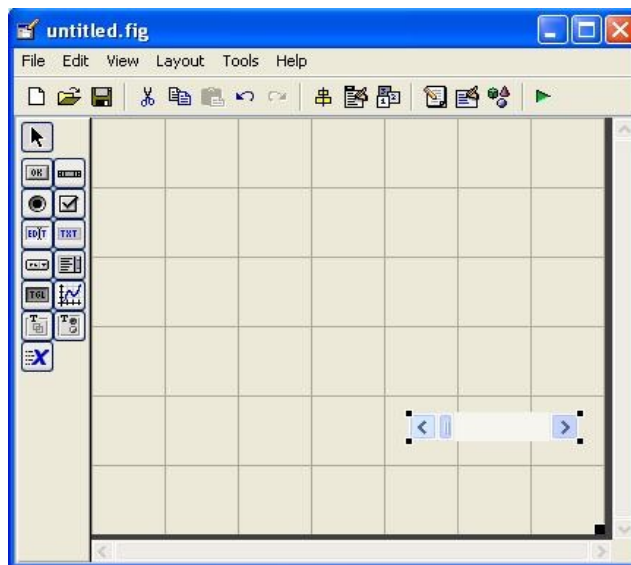
Με κόκκινους κύκλους είναι σημαδεμένη η ένδειξη του προγράμματος ότι το συγκεκριμένο αντικείμενο είναι επιλεγμένο, αποτέλεσμα της χρήσης του προαναφερθέντος “Select”. Αφού το “Push Button” έχει επιλεχθεί, έχουμε τη δυνατότητα να ορίσουμε τη θέση του στο «παράθυρο» σέρνοντας το με το ποντίκι, να του αλλάξουμε το μέγεθος και κάνοντας διπλό κλικ σε αυτό να εμφανιστούν οι ιδιότητες του, όπου υπάρχουν περισσότερα στοιχεία του αντικειμένου στα οποία μπορούμε να επέμβουμε. Οι ιδιότητες του εμφανίζονται σε ένα παράθυρο με τον τίτλο “Property Inspector” (επιθεωρητής ιδιοτήτων).



To “Property Inspector” του “Push Button”

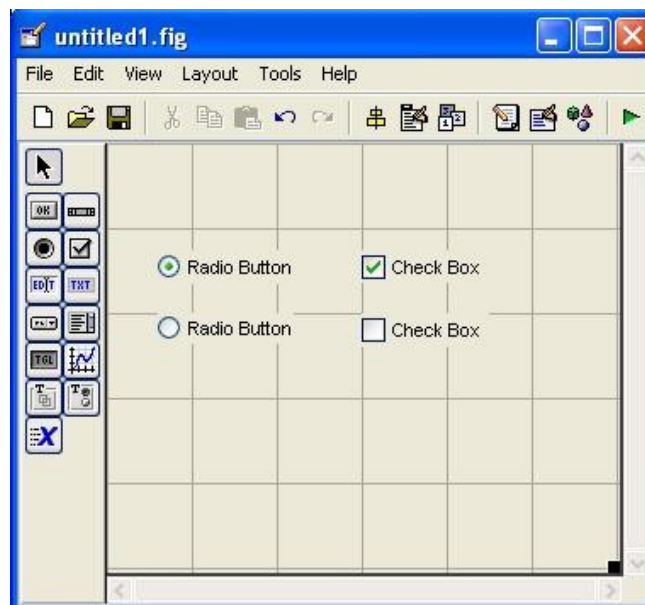
Σε αυτό το πλαίσιο υπάρχει μια πληθώρα επιλογών που είναι δυνατό να γίνουν και επηρεάζουν την εμφάνιση, διαθεσιμότητα και ενέργειες που εκτελούνται υπό κάποιες συνθήκες αναφορικά με το αντικείμενο. Έτσι, μπορεί κάποιος να αλλάξει το χρώμα του φόντου και των γραμμάτων στο αντικείμενο. Επιλογές έχουμε επίσης στο όνομα του αντικειμένου, με το οποίο θα αναφέρεται μέσα στο πρόγραμμα, ή στο τι θα αναγράφει επάνω του αυτό διαλέγοντας χρώμα και γραμματοσειρά. Ο χρήστης μπορεί να θέσει το αντικείμενο ανενεργό εξ αρχής ή υπό συνθήκη, να το κάνει αόρατο, να ορίσει τις τιμές που θα παίρνει και να το παραπέμπει σε αυτόματη εκτέλεση εργασιών όταν αυτό σβήνεται, ή επιλέγεται. Στην πλειοψηφία τους αυτές οι ιδιότητες είναι κοινές σε όλα τα αντικείμενα εκτός από κάποιες που οφείλονται στο διαφορετικό χαρακτήρα των αντικειμένων.

Στα δεξιά του “Push Button” υπάρχει το “Slider” , ένα αντικείμενο πολύ γνωστό σε όσους είναι εξοικειωμένοι με τα Windows της Microsoft, το οποίο κινείται από άκρη σε άκρη μέσα σε μια προκαθορισμένη ή μεταβλητή κλίμακα και η κάθε του διακριτή θέση, στην πραγματικότητα, αντιστοιχεί σε κάποια αριθμητική τιμή. Οι διάφορες τιμές του συνήθως αποτελούν συνθήκες για την εκτέλεση κάποιων εργασιών. Η πιο συχνή εργασία που ελέγχεται απ’ αυτό είναι η μετατόπιση μιας εικόνας μέσα σε παράθυρο συγκεκριμένου μεγέθους, κάνοντας έτσι ορατό διαφορετικό τμήμα της εικόνας, ανάλογα με τη θέση του “Slider”.



“Slider” τοποθετημένος στο κάτω δεξιά μέρος της φόρμας.

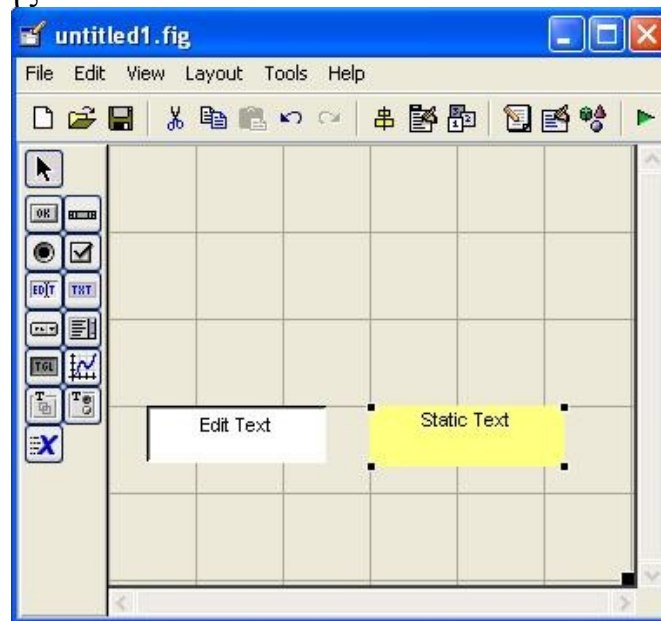
Τα επόμενα δύο αντικείμενα είναι όμοια στη λειτουργία και διαχείριση και το μόνο που τα διαχωρίζει είναι η εμφάνιση τους. Πρόκειται για το “Radio Button” και το διπλανό του “Check Box”. Παρακάτω φαίνονται αυτά σε διάταξη δύο οριζοντίων ζευγών, εκ των οποίων το επάνω ζεύγος είναι στην κατάσταση που λέγεται «επιλεγμένα» ενώ το κάτω στη «μη επιλεγμένα».



«Επιλεγμένα» και «μη επιλεγμένα» “Radio Button” & “Check Box”

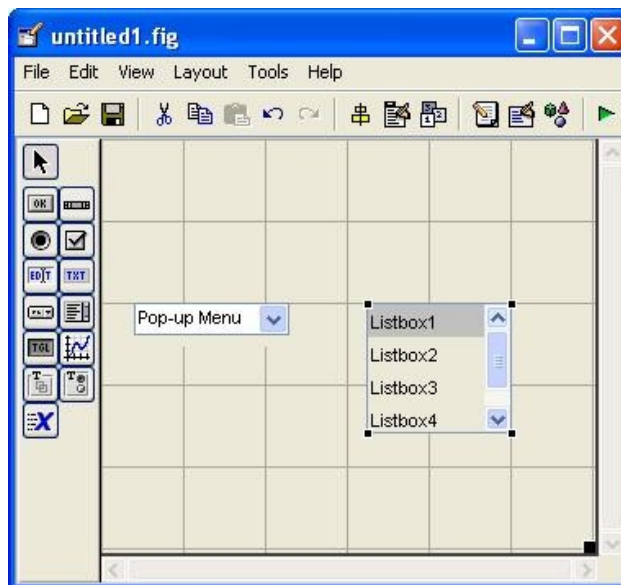
Η λειτουργία τους είναι πολύ απλή και συνηθισμένη. Πρόκειται για διάφορες επιλογές (ονόματα, ερωτήματα κτλ.) οι οποίες συνοδεύονται από ένα κυκλικό ή τετραγωνικό «κουτάκι». Αυτές τις επιλογές μπορεί ο χρήστης να ενεργοποιήσει επιλέγοντας το διπλανό «κουτάκι» ή να μην ενεργοποιήσει κάνοντας το αντίθετο. Σε συνδυασμό μεταξύ τους και σε μεγαλύτερο πλήθος αυτά τα αντικείμενα μπορούν να συνθέσουν ολόκληρες λίστες αιτημάτων που επηρεάζουν το πρόγραμμα με δυαδικό τρόπο. Η δυαδική λειτουργία τους φαίνεται από το γεγονός ότι η επιλογή τους έχει σαν προγραμματιστικό αποτέλεσμα την εισαγωγή στην εκάστοτε καθορισμένη μεταβλητή της τιμής 1 ενώ η μη επιλογή τους σημαίνει ότι η μεταβλητή θα περιέχει την αριθμητική τιμή 0.

Πηγαίνοντας προς τα κάτω στην εργαλειοθήκη, συναντάμε άλλα δύο συγγενικά μεταξύ τους αντικείμενα των οποίων η λειτουργία βασίζεται στο γραπτό κείμενο. Το πρώτο από αριστερά είναι το “Edit Text” και το διπλανό του είναι το “Static Text”. Το πρώτο είναι ένα πλαίσιο στο οποίο κυρίως ο χρήστης εισάγει κείμενο από το πληκτρολόγιο το οποίο αποθηκεύεται σε μια μεταβλητή και χρησιμοποιείται για τις όποιες μετέπειτα εργασίες. Το δεύτερο είναι «στατικό κείμενο», δηλαδή μια ετικέτα ή μια δήλωση που προσφέρεται στο χρήστη από το πρόγραμμα μόνο για ανάγνωση και μπορεί αυτό να είναι είτε προκαθορισμένο και αμετάβλητο ή να μεταβάλλεται αυτόματα μέσα στην πορεία του προγράμματος παίζοντας το ρόλο ενός πίνακα ανακοινώσεων. Το “Edit text” έχει και αυτό τη δυνατότητα να μεταβληθεί αυτόματα κατά τη διάρκεια του προγράμματος και συνήθως αυτό συμβαίνει με τη μορφή του αυτόματου σβησίματος του κειμένου που εισήγαγε ο χρήστης.



“Edit Text” & “Static Text”

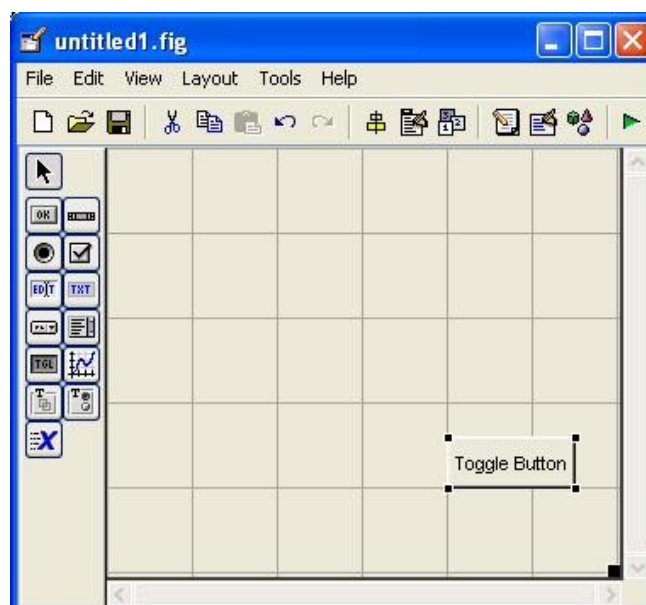
Ένα ακόμα συγγενικό ζεύγος αντικειμένων ακολουθεί τα δύο προηγούμενα. Η δυνατότητα να γίνουν επιλογές από μια προκαθορισμένη λίστα δεν δίνεται μόνο από τα προαναφερθέντα “Radio Button” και “Check Box” αλλά και από αυτόνομες λίστες με επικεφαλίδες για την κάθε επιλογή. Η χρήση λιστών σε αντιπαράθεση με ένα πλήθος από «κουτάκια» έχει το πλεονέκτημα ότι ένα μόνο αντικείμενο με μία αντίστοιχη συνάρτηση στο πρόγραμμα αντικαθιστά τα διάφορα αντικείμενα με τις αντίστοιχες διάφορες συναρτήσεις. Οι λίστες λοιπόν είναι πιο συγκεντρωτικές όσον αφορά τον προγραμματιστικό τους χαρακτήρα αλλά και τον εμφανισιακό τους.



“Pop-up Menu” & “List Box”

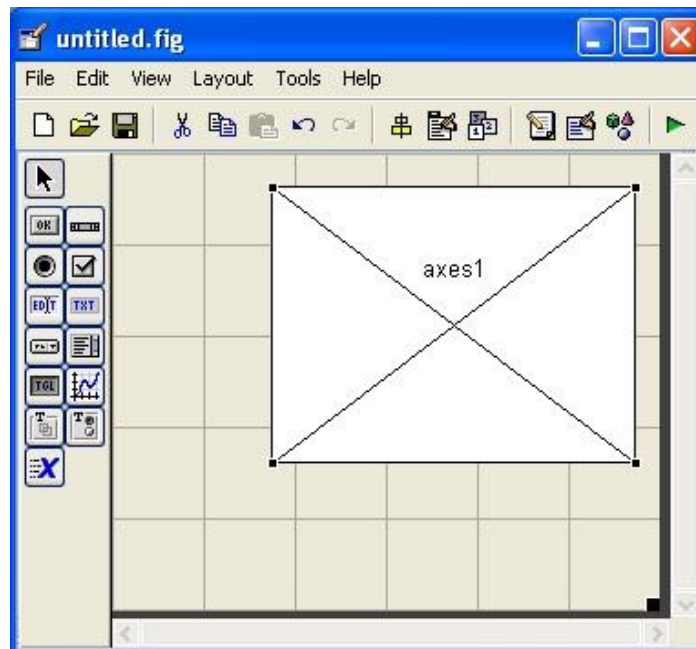
Οι λίστες είναι λοιπόν το “Pop-up Menu” και το “List Box”. Το “Pop-up Menu” αρχικά δείχνει μόνο μία επιλογή αφού καταλαμβάνει μία μόνο σειρά από το χώρο της φόρμας. Όταν όμως επιλεγεί, απλώνεται προς τα κάτω μία στήλη που φανερώνει όλα τα περιεχόμενα της λίστας και στη συνέχεια γίνεται η επιλογή ενός εξ’ αυτών. Το “List Box” από την άλλη αρχικά μπορεί να παρουσιάζει από μία έως και όλες τις διαθέσιμες επιλογές του ανάλογα με τον ορισμό που γίνεται κατά τη δημιουργία του. Ο τρόπος λειτουργίας του όμως διαφέρει από το προηγούμενο διότι επιτρέπει την ταυτόχρονη επιλογή στοιχείων του, πράγμα που αιτιολογεί πρακτικά την ξεχωριστή παρουσία του στην εργαλειοθήκη της Matlab.

Το δέκατο αντικείμενο είναι το “Toggle Button” δηλαδή ένα «πλήκτρο» που έχει την τάση να παραμένει πατημένο αν πιεστεί και το αντίστροφο. Η ιδιότητα του αυτή να διατηρεί μία από τις δύο καταστάσεις του ωσότου του ζητηθεί το αντίθετο, βρίσκει πρακτική εφαρμογή σε εργασίες που έχουν χρονική διάρκεια και ελεγχόμενο πέρας ή σε ανάθεση δύο διαφορετικών εργασιών στις δύο διαφορετικές καταστάσεις του, ενώ ταυτόχρονα απαιτείται η παύση της μίας για την εκκίνηση της άλλης.



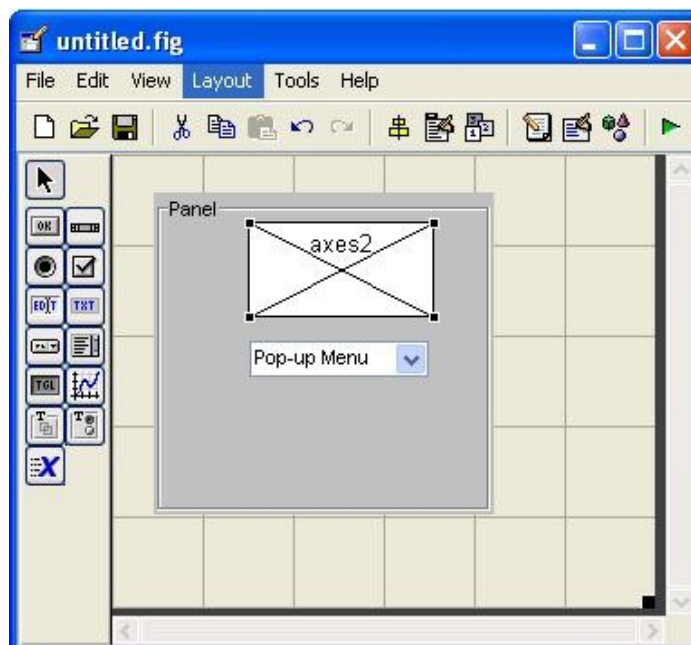
“Toggle Button” στο κάτω δεξιό τμήμα της φόρμας.

Δίπλα στο “Toggle Button” υπάρχει ένα διαφορετικής φύσης εργαλείο από τα προηγούμενα με καθαρά μαθηματικό χαρακτήρα. Το όνομα του είναι “Axes” (άξονες) και όπως αυτό μαρτυρά είναι ένα εργαλείο που βοηθά στην απεικόνιση γραφικών παραστάσεων. Συνεργάζεται με την εντολή της Matlab “plot(...)”, η οποία ευθύνεται για τη δημιουργία της γραφικής παράστασης και το τελικό της αποτέλεσμα εμφανίζεται στο χώρο που οριοθετεί το “Axes” επάνω στη φόρμα. Σε συνδυασμό με άλλα αντικείμενα, όπως είναι ένα “Push Button” ή ένα “List box” είναι δυνατό να κατασκευαστεί ένας αυτόματος τρόπος σχεδιασμού γραφημάτων, ο οποίος να απαιτεί από το χρήστη κάποιες στοιχειώδεις επιλογές που αφορούν στα μεγέθη που πρόκειται να χρησιμοποιηθούν και την τελική εντολή με το «πάτημα» ενός εικονικού πλήκτρου. Με τη χρήση περισσότερων από ένα “Axes” μπορεί κανείς να ταξινομήσει τα διάφορα γραφήματα του, με σειρά προτεραιότητας για παράδειγμα, πάνω σε μια φόρμα αποφεύγοντας τη χρήση πολλών «παραθύρων» που καθιστούν δύσκολη την ταυτόχρονη παρακολούθησή τους. Παρακάτω φαίνεται ένα πλαίσιο “Axes”.



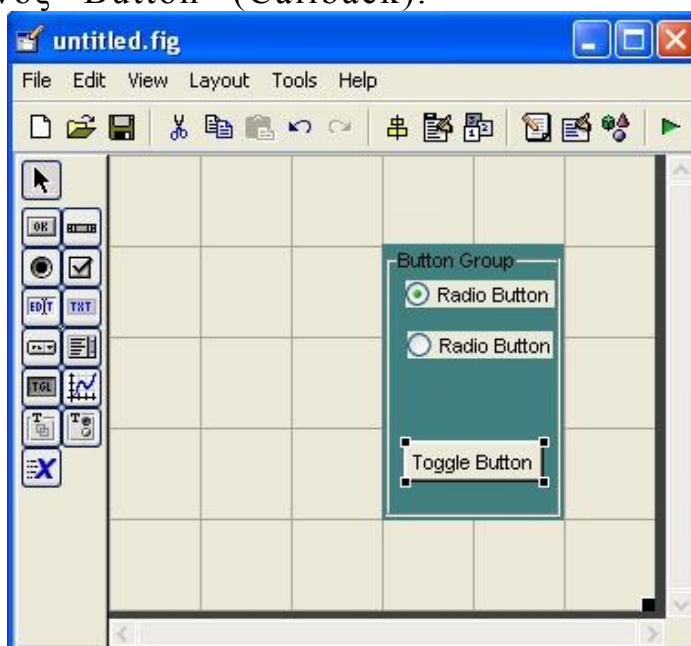
Το “Axes” στο σχεδιασμό GUI από τη Matlab.

Στη λίστα με τα εργαλεία που βοηθούν στην ανάπτυξη του γραφικού περιβάλλοντος, ακολουθεί το “Panel” (πλαίσιο). Το “Panel” ομαδοποιεί τα στοιχεία του GUI και κάνει τη φόρμα πιο εύκολη στην κατανόηση αφού η τοποθέτηση κάποιων αντικειμένων μέσα στο πλαίσιο τα διαχωρίζει (οπτικά) από τα υπόλοιπα και κάνει ξεκάθαρο το μεταξύ τους συσχετισμό. Μια επιπλέον δυνατότητα, που βοηθά τον προγραμματιστή, είναι ότι τα αντικείμενα που πλαισιώνονται με ένα “Panel” αυτομάτως η θέση τους πάνω στη φόρμα μεταφράζεται σε θέση σχετικά με το “Panel”. Έτσι όταν αλλάζει το μέγεθος του πλαισίου, είναι πιθανό να χρειάζονται επανατοποθέτηση τα αντικείμενα, πράγμα που μπορεί να γίνει αυτόματα αν ορισθεί κατάλληλα στην ειδική συνάρτηση “ResizeFcn” που συνοδεύει το “Panel” και εκτελείται όταν μεταβάλλεται το μέγεθος του. Τα αντικείμενα που περιέχονται σε ένα πλαίσιο μπορεί να είναι άλλα “Panel”, “Axes” ή “Button Groups” για τα οποία θα γίνει αναφορά παρακάτω.



Ένα “Panel” που περιέχει ένα “Axes” & ένα “Pop-up Menu”

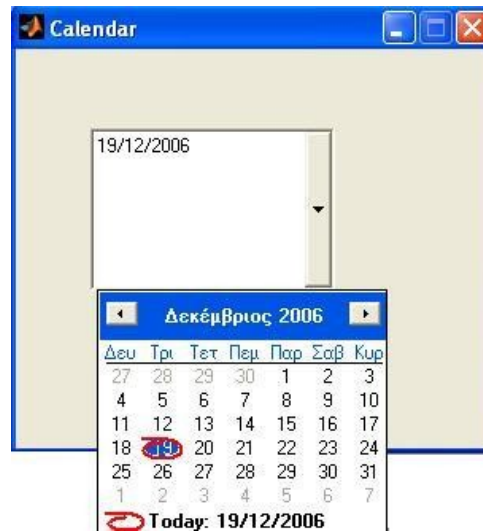
Το επόμενο και συγγενικό του προηγούμενου αντικείμενο, είναι το “Button Group”. Πρόκειται για ένα πλαίσιο που μπορεί να περικλείει “Toggle Buttons” και “Radio Buttons”. Όπως το προαναφερθέν πλαίσιο “Panel”, έτσι και αυτό χρησιμοποιείται για να ομαδοποιεί τα περιεχόμενα αντικείμενα του, για να αλλάζει τη θέση τους ανάλογα με τη μεταβολή του μεγέθους του, μια εργασία που ομοίως αναλαμβάνει η συνάρτηση “ResizeFcn”. Αυτό όμως που εξειδικεύει το “Button Group” στην ομαδοποίηση των “Buttons” είναι η συνάρτηση “SelectionChangeFcn” που το συνοδεύει και δίνει τη δυνατότητα ενημέρωσης του προγράμματος για οποιαδήποτε μεταβολή στην κατάσταση των “Buttons” του. Έτσι ο προγραμματιστής μπορεί να ενσωματώσει όλες τις ενέργειες των «κουμπιών» σε αυτή τη συνάρτηση αντί να χρησιμοποιήσει την ξεχωριστή συνάρτηση του καθενός “Button” (Callback).



Ένα “Button Group” που περιέχει ένα “Toggle Button” & δύο “Radio Buttons”

Το τελευταίο εργαλείο στη λίστα που μας παρέχει η Matlab είναι το “ActiveX Control”. Αυτό χρησιμοποιείται όταν το παραγόμενο GUI πρόκειται να λειτουργήσει σε λειτουργικό σύστημα Microsoft Windows. Είναι ένα

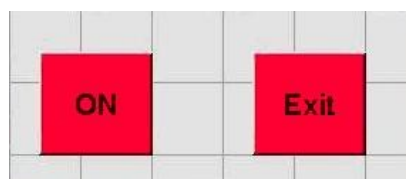
εξειδικευμένο εργαλείο που αναφέρεται σε μια μεγάλη λίστα πιο πολύπλοκων υπό-εργαλείων που μπορεί κανείς να τοποθετήσει στη φόρμα του, όπως είναι ένα ημερολόγιο, μία φόρμα με κελιά όμοια αυτής του Microsoft Excel, ένας “browser”, η δημιουργία διαφορετικών φύλλων στην ίδια φόρμα και άλλα πολύ ενδιαφέροντα και χρήσιμα αντικείμενα. Παρακάτω φαίνεται ένα ημερολόγιο που δημιουργήθηκε μόνο με την επιλογή του από τη λίστα των “ActiveX Controls”.



4.2 Τοποθέτηση αντικειμένων στο κύριο Gui του θαλάμου (booth.fig) και συνοπτική λειτουργία τους.

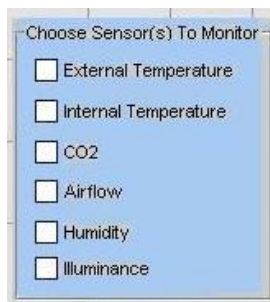
Κάποια από τα παραπάνω εργαλεία χρησιμοποιήθηκαν για τη δημιουργία του γραφικού περιβάλλοντος. Τοποθετήθηκαν το καθένα σε μια συγκεκριμένη θέση πάνω στη φόρμα και ομαδοποιήθηκαν κάποια από αυτά για τη διευκόλυνση της χρήσης και για ένα οπτικά όμορφο αποτέλεσμα. Στην αριστερή πλευρά βρίσκονται όλες οι επιλογές που μπορεί ή πρέπει να πραγματοποιήσει ο χρήστης και δεξιά οι άξονες παρακολούθησης των αισθητήρων.

Ξεκινώντας από την αριστερά πάνω θέση, μπορεί κάποιος να διακρίνει τα κουμπιά ON και EXIT. Χρησιμοποιήθηκε κόκκινο χρώμα σ' αυτά επειδή ο ρόλος τους είναι σημαντικός για την εκκίνηση και τον τερματισμό λειτουργίας του gui.



Τα κουμπιά ON και EXIT.

Ακριβώς κάτω από το κουμπί ON τοποθετήθηκαν έξι “checkboxes” με τις ονομασίες των αισθητήρων που υπάρχουν στο θάλαμο και όλα μαζί ομαδοποιήθηκαν μέσα σε ένα “button group” με τίτλο “Choose Sensor(s) To Monitor” με διαφορετική απόχρωση για να τονίζει την ομαδοποίησή τους. Αυτά χρησιμοποιούνται για την επιλογή αισθητήρων προς παρακολούθηση.



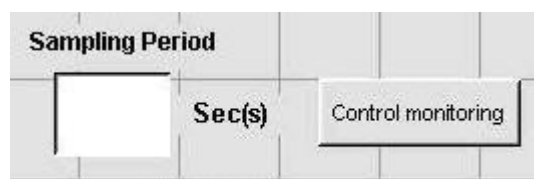
Το ‘button group’ που περιέχει τα ‘checkboxes’ με τις ονομασίες των αισθητήρων.

Δεξιά από το παραπάνω ‘button group’ βρίσκονται τρία κουμπιά με τίτλους ‘Start monitoring’, ‘Pause monitoring’ και ‘Stop monitoring’. Όπως εύκολα μπορεί να διαπιστώσει ο χρήστης, οι λειτουργίες τους είναι «έναρξη παρακολούθησης», «παύση παρακολούθησης» και «τερματισμός παρακολούθησης» αντίστοιχα για τους προεπιλεγμένους αισθητήρες.



Το κουμπιά για διάφορες επιλογές παρακολούθησης.

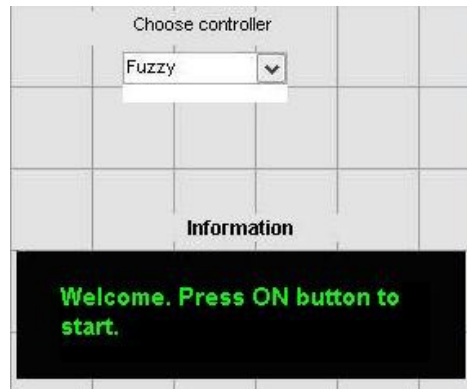
Κάτω από το ‘button group’ είναι τοποθετημένα δύο ‘static text’ με την ονομασία Sampling period και Sec(s), ένα ‘edit text’ και ένα κουμπί με την ονομασία control monitoring. Η μοναδική λειτουργία των ‘static text’ είναι να παρέχουν πληροφορίες στο χρήστη. Το ‘edit text’ χρησιμοποιείται για την εισαγωγή χρόνου σε δευτερόλεπτα, για το ρυθμό δειγματοληψίας κατά την παρακολούθηση. Το κουμπί στα δεξιά τους, ενεργοποιεί ένα παράθυρο στο οποίο παρατηρούνται οι τιμές που στέλνει ο κώδικας ασαφούς λογικής στους διάφορους μηχανισμούς ελέγχου και κίνησης.



Το ‘edit text’ για την περίοδο δειγματοληψίας και το κουμπί για την παρακολούθηση των τιμών που ‘στέλνει’ η matlab στο θάλαμο.

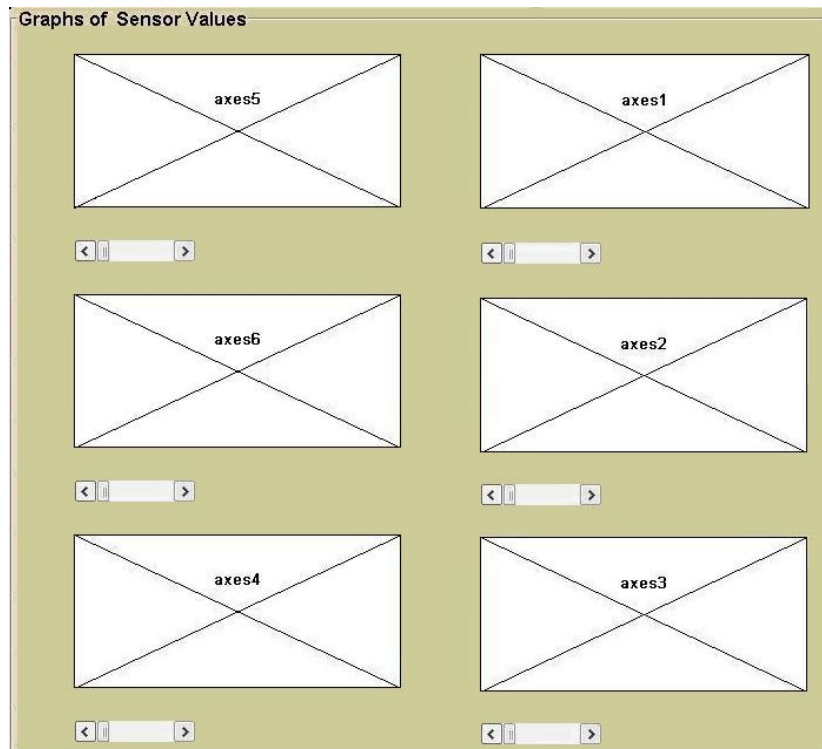
Τα τελευταία αντικείμενα που τοποθετήθηκαν στην αριστερή πλευρά της φόρμας είναι τρία ‘static text’ και ένα ‘pop-up menu’. Τα δύο ‘static text’ επεξηγούν στο χρήστη για το αντικείμενο που βρίσκεται κάτω απ’ αυτά. Το ένα βρίσκεται πάνω από το ‘pop-up menu’ και το άλλο πάνω από το τρίτο ‘static text’. Το τρίτο παρέχει πληροφορίες στο χρήστη για το χειρισμό και τις λειτουργίες του gui αναλόγως σε ποια κατάσταση βρίσκεται αυτό. Το χρώμα που επιλέχθηκε για το υπόβαθρο του ‘text’ πληροφοριών είναι μαύρο και το χρώμα των γραμμάτων είναι έντονο πράσινο για να τραβάνε την προσοχή του χρήστη κατά τη διάρκεια

χρήσης του gui. Το “pop-up menu” χρησιμοποιείται από το χρήστη για να επιλεγεί το είδος του ελέγχου με τον οποίο πρόκειται να λειτουργήσει ο θάλαμος. Fuzzy, PID και On-Off έλεγχος είναι οι τρεις διαθέσιμες επιλογές.



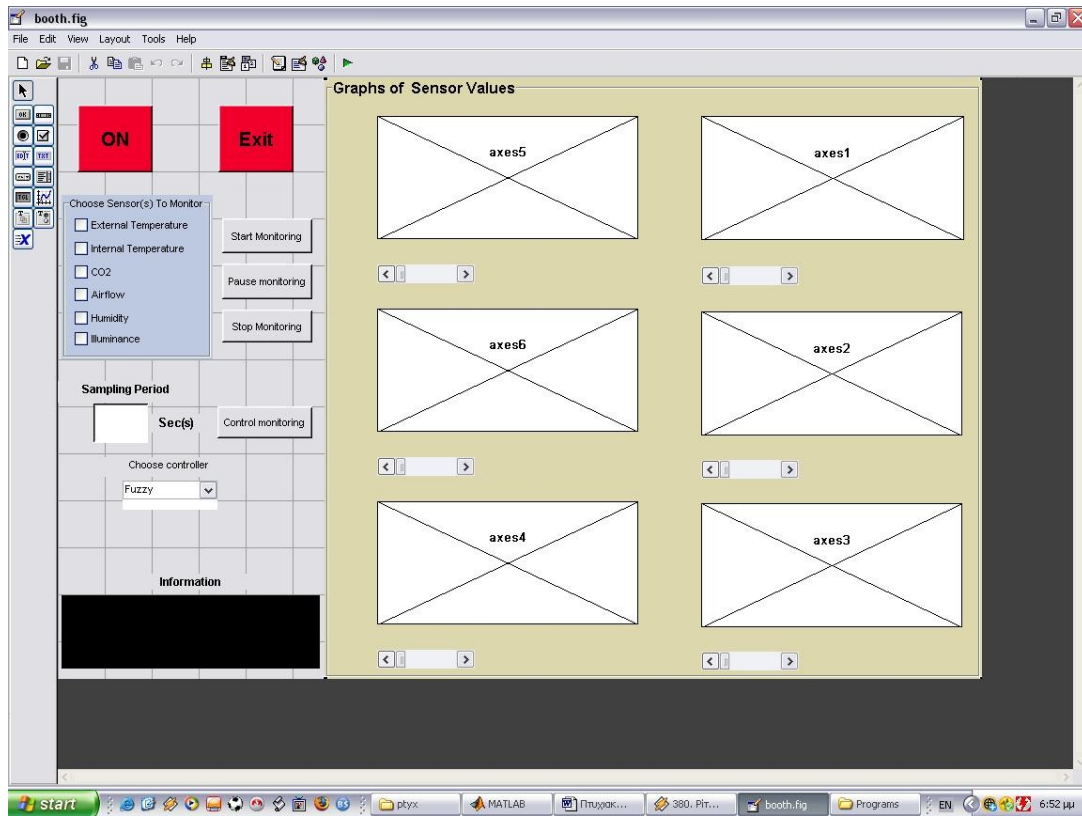
Το “pop-up menu” με τους διαθέσιμους controllers και το “static text” με τις πληροφορίες για το χρήστη.

Στη δεξιά πλευρά της φόρμας τοποθετήθηκαν σαν ομάδα έξι άξονες μαζί με έξι ολισθητές “sliders” κάτω από τον καθένα κι όλα μαζί πάνω σε ένα πάνελ με ονομασία “Graphs of Sensor Values” χρωματισμένο ελαφρύ κίτρινο για να διαφέρει απ’ το υπόλοιπο gui και να μην κουράζει το χρήστη. Οι άξονες χρησιμοποιούνται για τη γραφική παρακολούθηση των τιμών που λαμβάνονται απ’ τους αισθητήρες του θαλάμου ως προς το χρόνο δειγματοληψίας. Οι “sliders” χρησιμοποιούνται για την ολίσθηση του γραφήματος κατά μήκος του άξονα του χρόνου για την παρατήρηση προηγούμενων τιμών.



Το “panel” με τους άξονες και τους “sliders”.

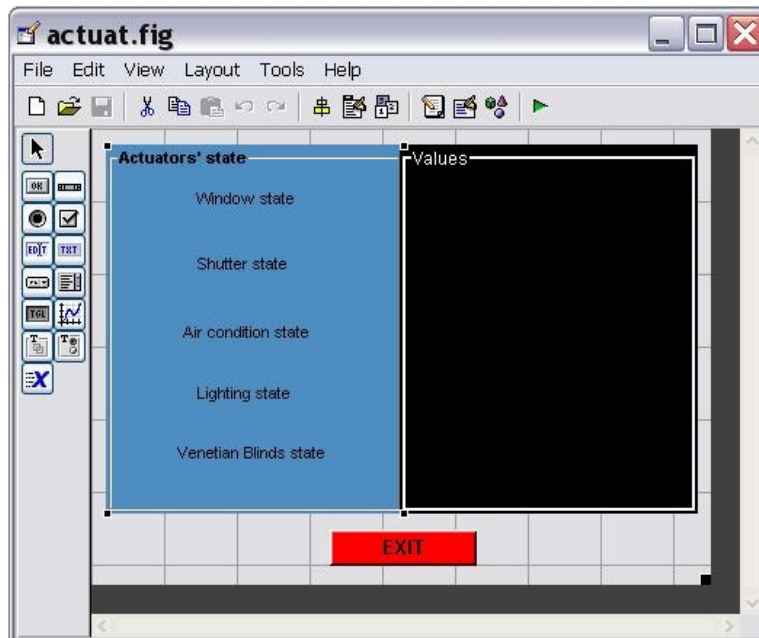
Η τελική μορφή του GUI είναι η παρακάτω



Το booth στην τελική του μορφή με όλα τα εργαλεία που χρησιμοποιήθηκαν.

4.3 Το gui actuat.fig

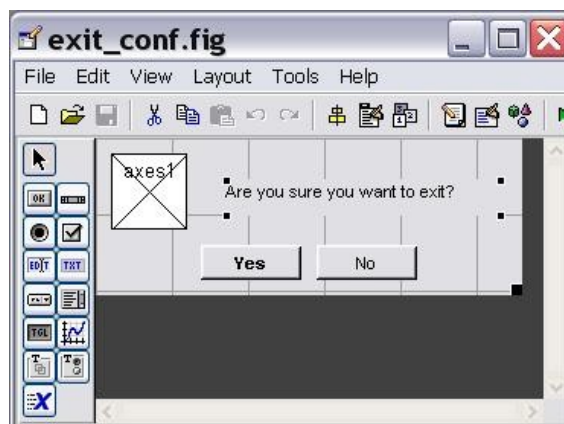
Ένα δεύτερο περιβάλλον χρησιμοποιείται για την παρακολούθηση των τιμών με τις οποίες ελέγχονται οι μηχανισμοί που βρίσκονται στο θάλαμο. Η ονομασία του gui αυτού είναι control monitoring και εκτελείται κάνοντας click στο ομώνυμο κουμπί πάνω στο περιβάλλον του booth. Τα εργαλεία που έχουν χρησιμοποιηθεί είναι δυο *panel* με ονομασία “actuators’ state” και “values”. Το *panel* “actuators’ state” έχει γαλάζιο χρώμα και περιέχει τα πέντε *static texts* με τις ονομασίες από τους αντίστοιχους μηχανισμούς του θαλάμου. Το *panel* “values” είναι τοποθετημένο δεξιά και περιέχει άλλα πέντε *static texts* για την εμφάνιση των αντίστοιχων τιμών για κάθε μηχανισμό. Το περιβάλλον έχει ονομαστεί *actuat* από την αγγλική ονομασία των μηχανισμών, actuators. Στην εικόνα που ακολουθεί παρουσιάζεται το gui στην τελική του μορφή.



To actuat.fig στην τελική του μορφή με όλα τα εργαλεία που χρησιμοποιήθηκαν.

4.4 Το gui exit_conf.fig

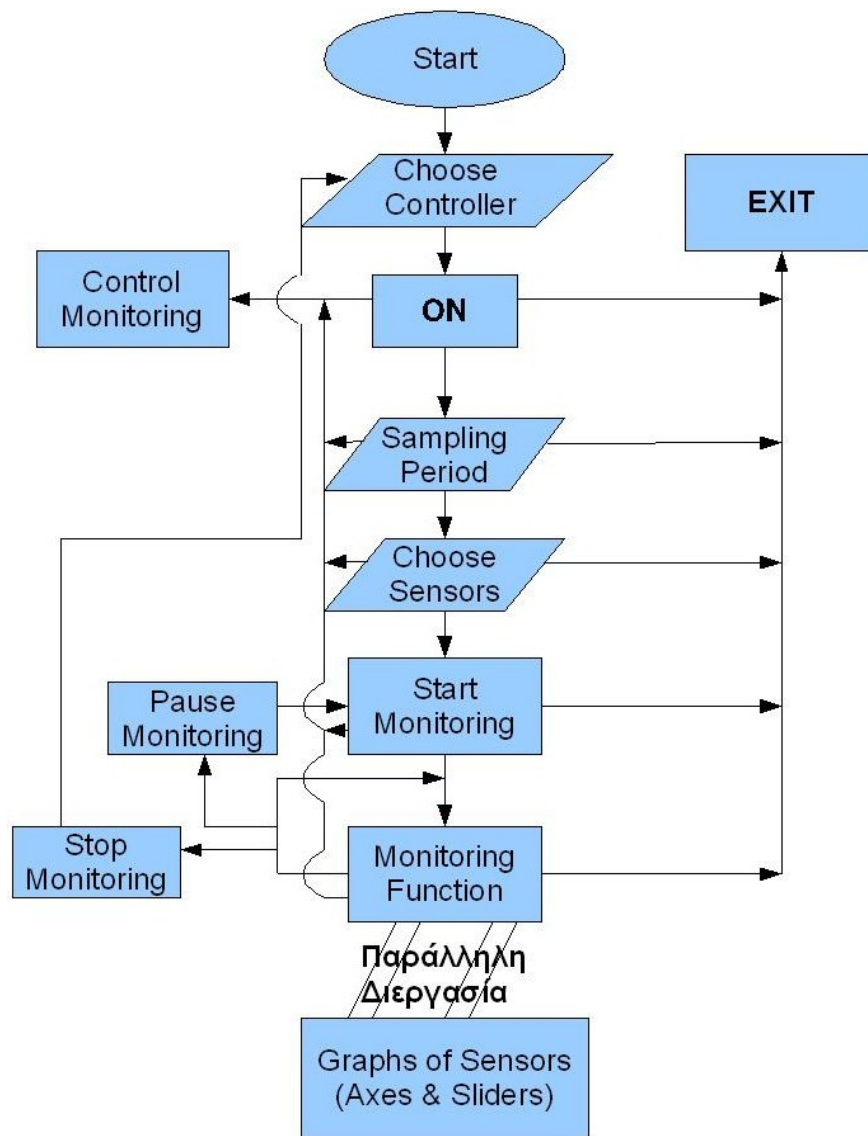
Το τρίτο και τελευταίο περιβάλλον που έχει δημιουργηθεί ονομάζεται “exit_conf” συντομογραφία του exit confirmation. Αυτό χρησιμοποιείται κατά την επιλογή του χρήστη να εγκαταλείψει το περιβάλλον και η μόνη λειτουργία του είναι η επιβεβαίωση για αυτή την επιλογή. Τα εργαλεία που χρησιμοποιήθηκαν είναι ένα *axes* το οποίο παρουσιάζει το αγγλικό σύμβολο του ερωτηματικού “?” (χρησιμοποιήθηκε για καλλωπιστικούς λόγους), ένα *static text* με την ερώτηση προς το χρήστη “ Are you sure you want to exit ? ” και δυο κουμπιά για την επιβεβαίωση της επιλογής του. Η εικόνα που ακολουθεί παρουσιάζει το περιβάλλον του exit_conf.fig .



To gui exit_conf.fig

4.5 Τα αρχεία Booth.m, actuat.m & exit_conf.m και η λογική ανάθεσης εργασιών για κάθε στοιχείο.

Με την ολοκλήρωση της δημιουργίας του περιβάλλοντος έχουν δημιουργηθεί έξι αρχεία, τρία με κατάληξη .fig και τρία με αντίστοιχη ονομασία με τα αρχεία περιβάλλοντος (.fig), με κατάληξη .m . Τα πρώτα δύο ονομάζονται booth.fig , με τον κώδικα για τη γραφική απεικόνιση του περιβάλλοντος, και booth.m στο οποίο γίνεται η ανάθεση εργασιών για κάθε στοιχείο του περιβάλλοντος. Τα επόμενα δύο είναι τα actuat.fig και actuat.m και τα τελευταία δύο είναι τα exit_conf.fig και exit_conf.m (Τα gui actuat.fig και exit_conf.fig καλούνται από το χρήστη μέσω του κεντρικού περιβάλλοντος booth.fig.). Τα αρχεία με κατάληξη .m περιέχουν τουλάχιστον τόσες συναρτήσεις όσα και τα στοιχεία του GUI με αντίστοιχη ονομασία. Αρχικά αυτές οι συναρτήσεις είναι κενές και ο προγραμματιστής πρέπει να ορίσει τις διεργασίες για κάθε στοιχείο σε περιβάλλον γραμμής εντολών. Επίσης στην αρχή του κώδικα υπάρχει μία πρόσθετη συνάρτηση (για το booth.m η ονομασία της είναι “ *function booth_OpeningFcn* ”) η οποία εκτελείται κατά την εκκίνηση του γραφικού περιβάλλοντος και αφορά όλες τις αρχικοποιήσεις που απαιτείται να γίνουν για τη σωστή λειτουργία του GUI. Η δομή του κώδικα είναι τέτοια ώστε κάθε φορά που ο χρήστης επιλέγει κάποιο στοιχείο του περιβάλλοντος (π.χ. κάνει “click” σε ένα κουμπί) εκτελείται αυτόματα η αντίστοιχη συνάρτηση και κατά συνέπεια ο κώδικας που έχει προστεθεί σε κάθε μία, δηλαδή αν ο χρήστης επιλέξει το κουμπί “Start Monitoring” εκτελείται η συνάρτηση “*function Start_monitor_butt_Callback*”. Η λογική προγραμματισμού στη Matlab επιτρέπει την ταυτόχρονη εκτέλεση περισσότερων από μια συναρτήσεων οι οποίες μπορεί για παράδειγμα να κάνουν τα εξής, μία να συλλέγει πληροφορίες από τους αισθητήρες, μία δεύτερη να προβάλλει τις τιμές στα γραφήματα και μια τρίτη να διακόπτει τη λειτουργία των δύο προηγούμενων. Για να γίνει εφικτή η συνεργασία μεταξύ των συναρτήσεων απαιτείται η ανταλλαγή δεδομένων μεταξύ τους και γι’ αυτό έχουν χρησιμοποιηθεί μεταβλητές τύπου “global” (μτφ. παγκόσμιες). Ακολουθεί ένα γενικό διάγραμμα ροής του αρχείου booth.m.

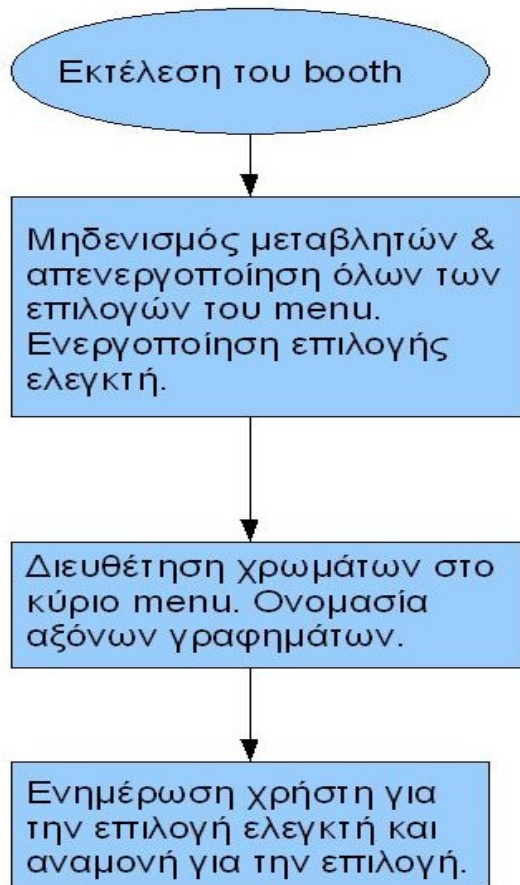


Διάγραμμα ροής.

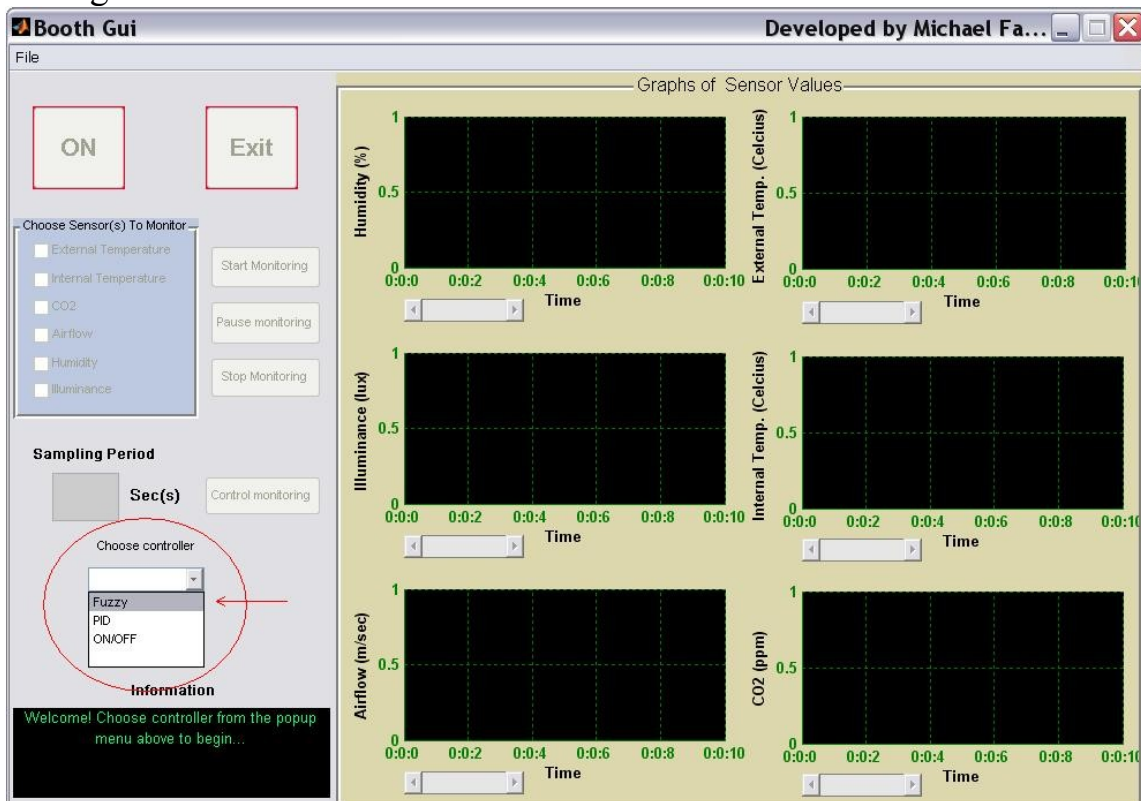
Η συνάρτηση “control monitoring” καλεί το δεύτερο GUI, `actuat.fig` του οποίου η λειτουργία, όπως έχει αναφερθεί, είναι να παρέχει στο χρήστη πληροφορίες για τις εντολές από τον αλγόριθμο ασαφούς λογικής προς τους μηχανισμούς ελέγχου και κίνησης, οι οποίοι είναι τοποθετημένοι στο θάλαμο.

4.6 Ροή διεργασιών και ανάλυση κώδικα του `Booth.m`

Κατά την εκτέλεση του αρχείου `booth.m` (εκκίνηση του `gui`) μέσα από το περιβάλλον λειτουργίας της `matlab`, οι πρώτες τρεις συναρτήσεις που εκτελούνται αυτόματα είναι οι : “`function varargout = booth(varargin)`” , “`function booth_OpeningFcn(hObject, eventdata, handles, varargin)`” , “`function varargout = booth_OutputFcn(hObject, eventdata, handles)`” και χρησιμοποιούνται για την αρχικοποίηση του περιβάλλοντος. Η πρώτη και η τρίτη (`function varargout = booth(varargin)` , `function varargout = booth_OutputFcn`) περιέχουν κώδικα μη παραμετροποιήσιμο από τον προγραμματιστή και δε θα γίνει αναφορά σ’ αυτές. Στη δεύτερη (`function booth_OpeningFcn`) έχουν προστεθεί στοιχεία για τις λειτουργίες των παραθύρων κατά την εκκίνηση του `booth`. Οι λειτουργίες της συνάρτησης περιγράφονται παρακάτω σε μορφή διαγράμματος ροής

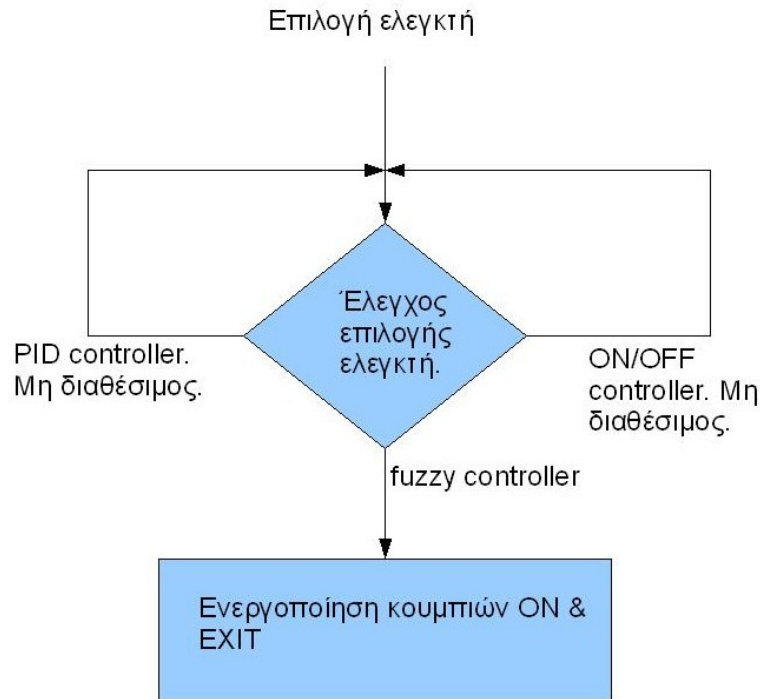


Με την ολοκλήρωση των αρχικών ρυθμίσεων το περιβάλλον βρίσκεται σε μια μορφή αδράνειας. Όλες οι επιλογές είναι απενεργοποιημένες εκτός από το pop-up menu “choose controller”. Η συνάρτηση αυτού του pop-up menu παρέχει στο χρήστη τη δυνατότητα επιλογής του controller με τον οποίο θα λειτουργήσει ο θάλαμος. Στην παρούσα φάση είναι διαθέσιμος μόνο ο ελεγκτής ασαφούς λογικής (fuzzy). Οι υπόλοιπες επιλογές είναι διαθέσιμες για μελλοντική επέκταση των δυνατοτήτων του gui. Στην εικόνα που ακολουθεί παρατηρείται το στάδιο που βρίσκεται το gui.



Αρχικό στάδιο περιβάλλοντος.

Το διάγραμμα ροής της συνάρτησης του pop-up menu φαίνεται παρακάτω :

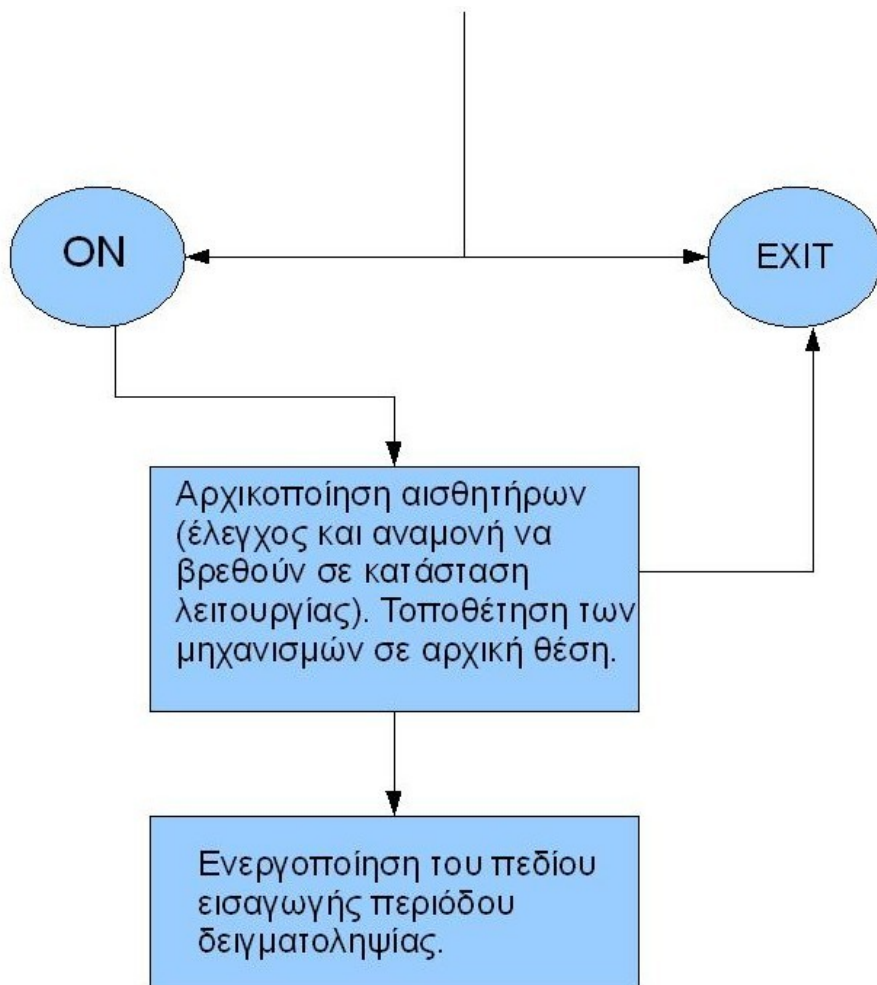


Η συνάρτηση που ακολουθεί είναι αυτή του κουμπιού ON καθώς είναι το επόμενο και τελευταίο βήμα για την αρχικοποίηση του περιβάλλοντος. Αυτή περιέχει όλες τις αναγκαίες λειτουργίες για την αρχικοποίηση των μηχανισμών (hardware) του θαλάμου. Αυτό επιτυγχάνεται με την επικοινωνία του κώδικα της matlab με τον θάλαμο μέσω του OPC client – server (ενδιάμεσο πρόγραμμα επικοινωνίας). Μετά τον πρώτο κύκλο αρχικών ρυθμίσεων του θαλάμου ο κώδικας λειτουργεί με τη λογική του loop λαμβάνοντας μετρήσεις από τους αισθητήρες και δίνοντας διάφορες τιμές στους μηχανισμούς για την επίτευξη των όσο το δυνατόν κοντινότερων τιμών, σε αυτές που έχουν προκαθοριστεί ως τιμές συνθηκών άνετης διαβίωσης για τον άνθρωπο. Στην παρακάτω εικόνα παρατηρείται το σημείο που βρίσκεται το περιβάλλον.



Επιλεγμένος ελεγκτής και αναμονή για την αρχικοποίηση (ON).

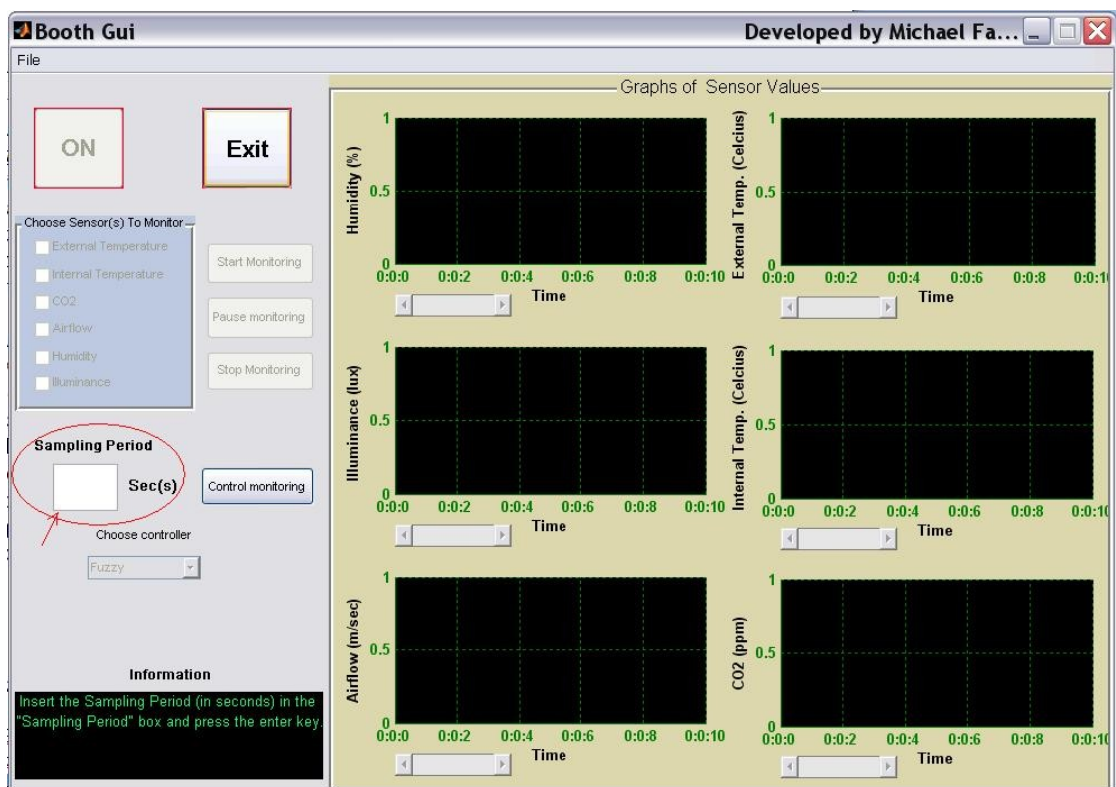
Πρέπει να σημειωθεί ότι η συνάρτηση αυτή έχει δημιουργηθεί από τον καθηγητή κ. Πουλιέζο Α. , καθηγητή στο Α.Τ.Ε.Ι. Κρήτης παρ/μα Χανίων.



Πιο Αναλυτικά, τα βήματα που ακολουθούνται στην συνάρτηση του κουμπιού “ON” είναι τα εξής : Αρχικά εκτελείται ο OPC Client για επίτευξη επικοινωνίας

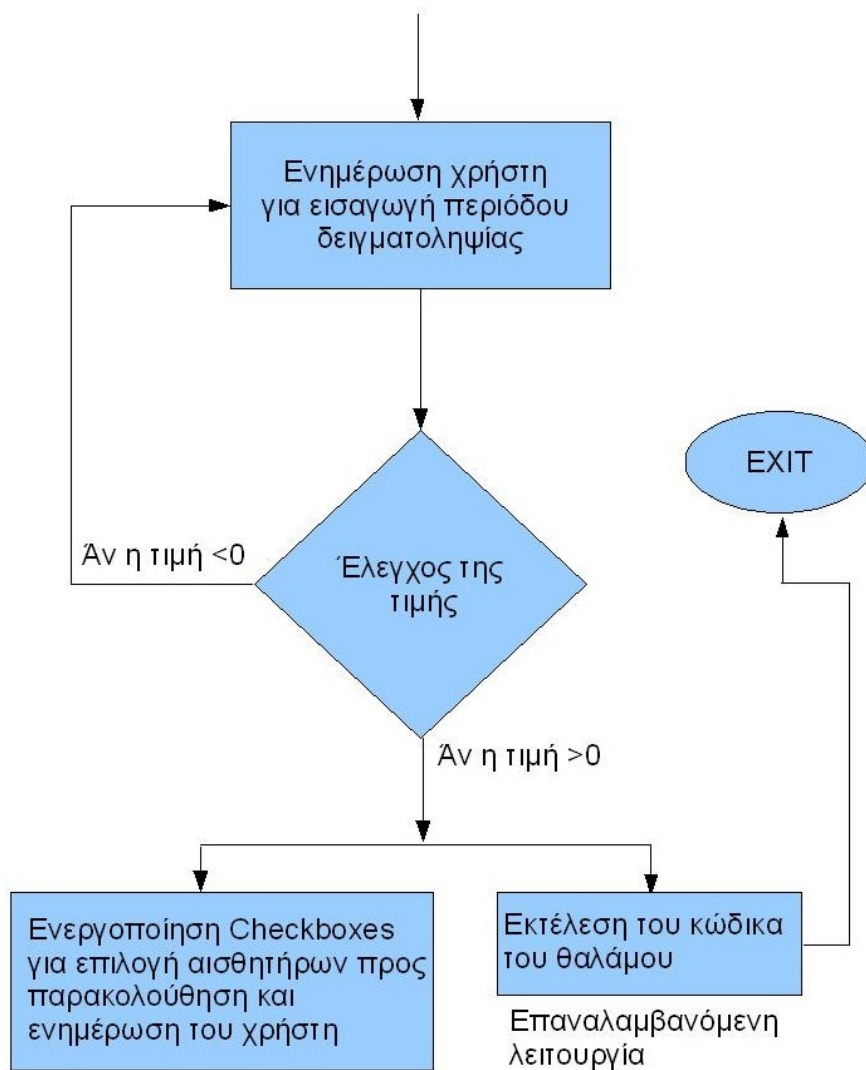
Matlab - θαλάμου και ορίζονται μηδενικές τιμές για όλες τις συσκευές ελέγχου και κίνησης (αρχικοποίηση: Παράθυρο κλειστό, AC απενεργοποιημένο κτλ). Επίσης ορίζεται χρόνος αναμονής για να βρεθούν σε κατάσταση ομαλής λειτουργίας κάποιοι από τους αισθητήρες, όπως ο αισθητήρας του διοξειδίου του άνθρακα (CO₂ warm-up). Μετά τις αρχικοποιήσεις εκτελείται ο επαναλαμβανόμενος κώδικας ελέγχου (loop) ο οποίος λαμβάνει τιμές από τους αισθητήρες του θαλάμου και βάση του αλγόριθμου ασαφούς λογικής ελέγχει τους μηχανισμούς του θαλάμου για την επίτευξη των επιθυμητών τιμών. Κάθε κύκλος ελέγχου διαρκεί μερικά λεπτά και σ' αυτό το διάστημα ελέγχονται όλοι οι μηχανισμοί του θαλάμου. Στο τέλος κάθε κύκλου αποθηκεύονται σε ένα πίνακα μεταβλητών οι τιμές που έχουν ληφθεί από τους αισθητήρες καθώς και τα αποτελέσματα των αποφάσεων που έχουν προκύψει από τον αλγόριθμο της ασαφούς λογικής.

Με την έναρξη του πρώτου κύκλου ελέγχου ενεργοποιείται το “edit box” Period_edit στο οποίο αντιστοιχεί η συνάρτηση “*function* Period_edit_Callback”. Εδώ εισάγεται η τιμή για την περίοδο δειγματοληψίας (σε δευτερόλεπτα) που χρησιμοποιείται για να οριστεί η συχνότητα λήψης τιμών από τους αισθητήρες και κατ' επέκταση ο ρυθμός ανανέωσης των τιμών των γραφημάτων. Ακολουθεί εικόνα για την παρούσα φάση.



Καθορισμός περιόδου δειγματοληψίας.

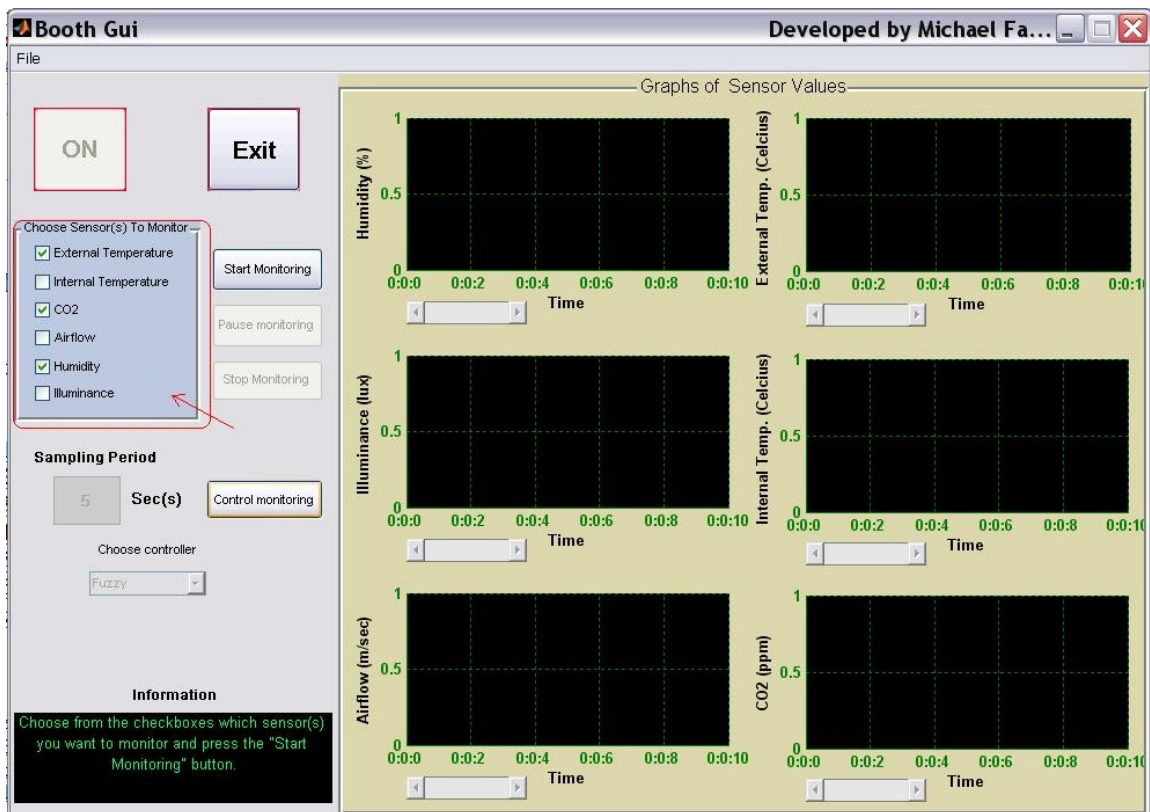
Το διάγραμμα του period_edit φαίνεται παρακάτω.



Οι επόμενες συναρτήσεις που ενεργοποιούνται είναι αυτές των checkboxes για την επιλογή αισθητήρα προς παρακολούθηση. Επιλέγοντας έναν ή περισσότερους από αυτούς ενεργοποιείται μια σημαία (flag, περιοχή τιμών : 0 ή 1) με ονομασία από a1 - a6 για τον κάθε αισθητήρα ξεχωριστά. Ο κώδικας των συναρτήσεων είναι όμοιος, γι' αυτό θα εξηγηθεί μόνο μια συνάρτηση checkbox

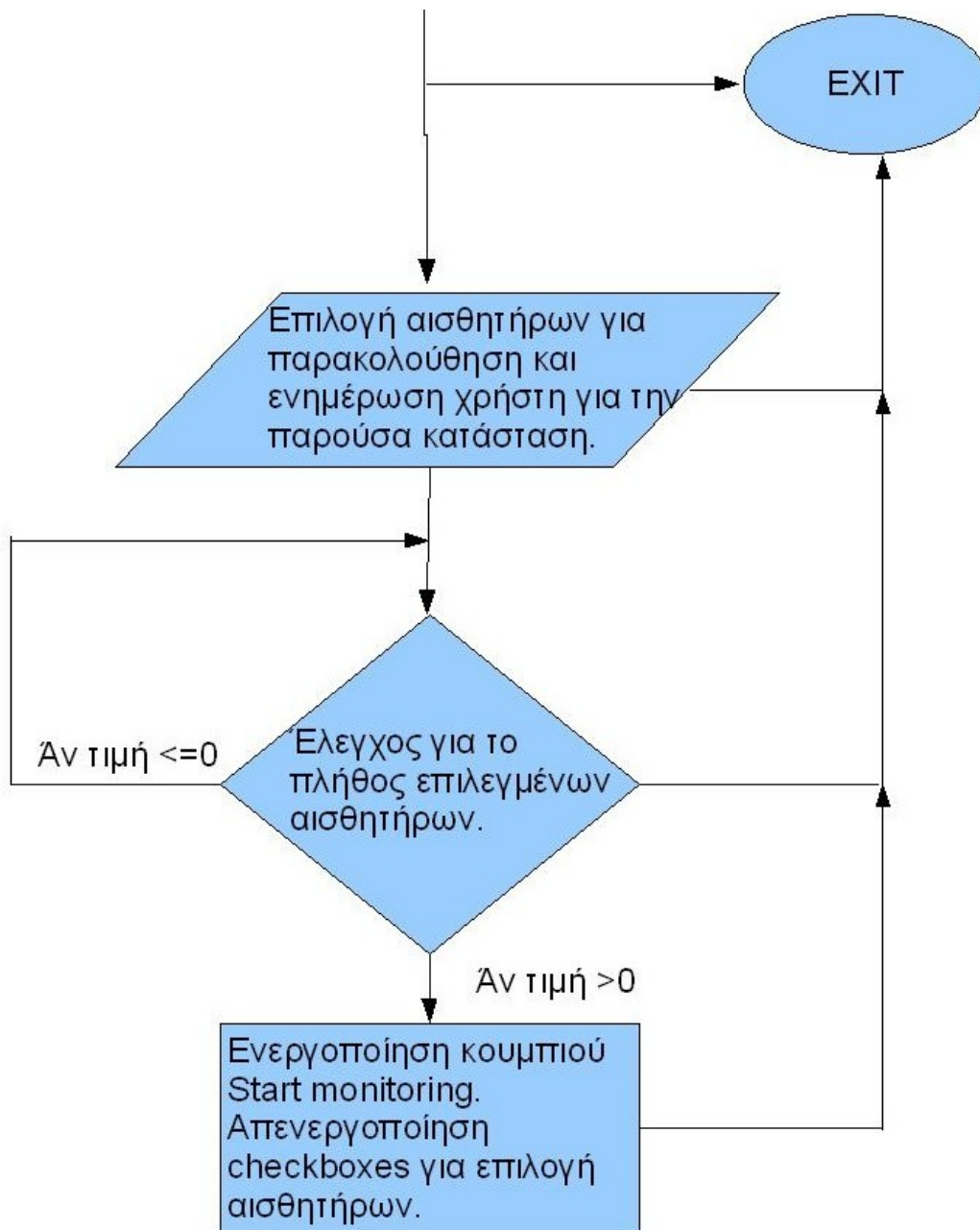
- a1 : Αισθητήρας εξωτερικής θερμοκρασίας
- a2 : Αισθητήρας εσωτερικής θερμοκρασίας
- a3 : Αισθητήρας CO₂
- a4 : Αισθητήρας ροής αέρα
- a5 : Αισθητήρας υγρασίας
- a6 : Αισθητήρας φωτεινότητας

Στην εικόνα παρακάτω φαίνεται το panel με τα check boxes.

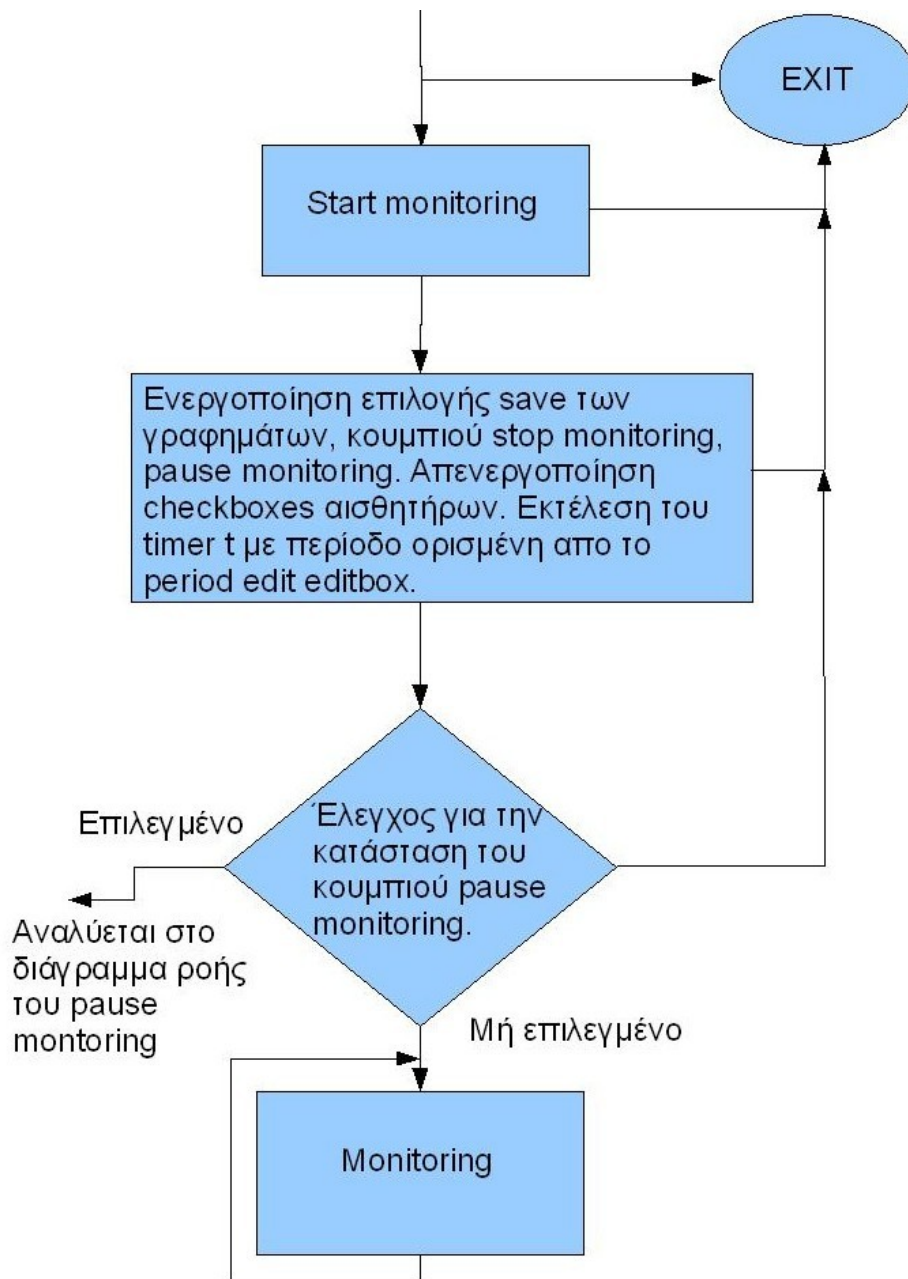


Επιλογή αισθητήρων για παρακολούθηση.

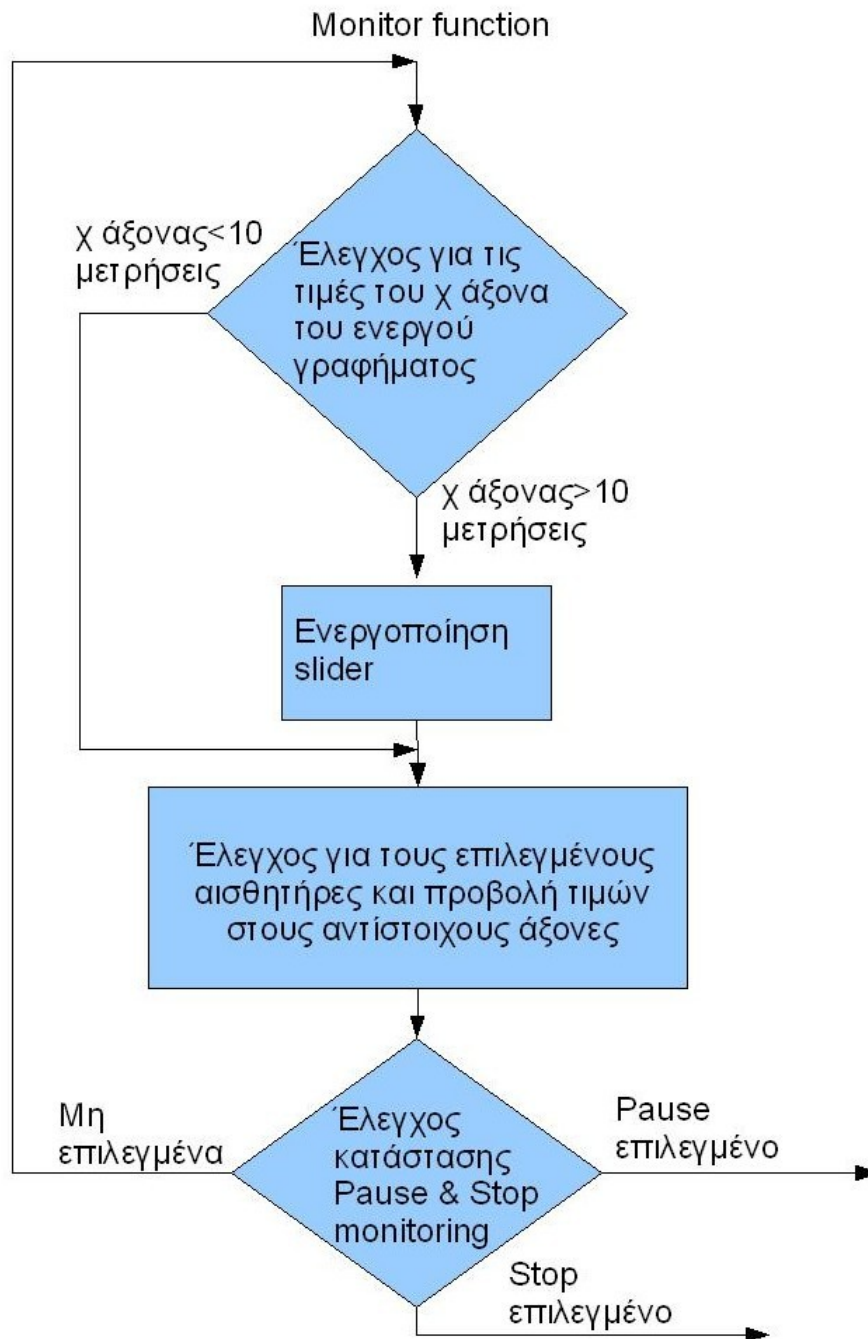
Το διάγραμμα της συνάρτησης :



Έτσι ενεργοποιείται το κουμπί Start monitoring. Αυτό είναι υπεύθυνο για την εκκίνηση δειγματοληψίας των τιμών των αισθητήρων. Οι διεργασίες της συνάρτησης φαίνονται αναλυτικά παρακάτω :



Με τη σειρά της ενεργοποιείται η συνάρτηση monitor. Στην παρούσα συνάρτηση γίνονται όλες οι αναγκαίες ενέργειες για τη δειγματοληψία τιμών από τους αισθητήρες και την παρουσίασή τους σε σχέση με το χρόνο περιόδου που έχει ορίσει ο χρήστης. Κάθε γράφημα στον Ψ άξονα έχει την ονομασία του αισθητήρα του οποίου προβάλλονται τα δεδομένα και στο Χ άξονα τον χρόνο σε Ώρες : Λεπτά : Δευτερόλεπτα για κάθε τιμή. Επίσης κάτω από κάθε γράφημα βρίσκεται ένας ολισθητής (slider) ο οποίος μετά την πάροδο των πρώτων δέκα δειγμάτων ενεργοποιείται και δίνει τη δυνατότητα στο χρήστη ολισθαίνοντάς τον προς τα αριστερά να παρακολουθήσει προηγούμενες τιμές για κάθε αισθητήρα. Όσο ο χρήστης παρατηρεί προηγούμενες τιμές με τη χρήση του slider το gui συνεχίζει να δειγματοληπτεί αλλά το γράφημα μένει στάσιμο στις τιμές που το ολίσθησε ο χρήστης. Όταν τοποθετήσει το slider στην άκρα δεξιά θέση αυτό κλειδώνει και προβάλλει τις νέες τιμές που καταφθάνουν από το θάλαμο.

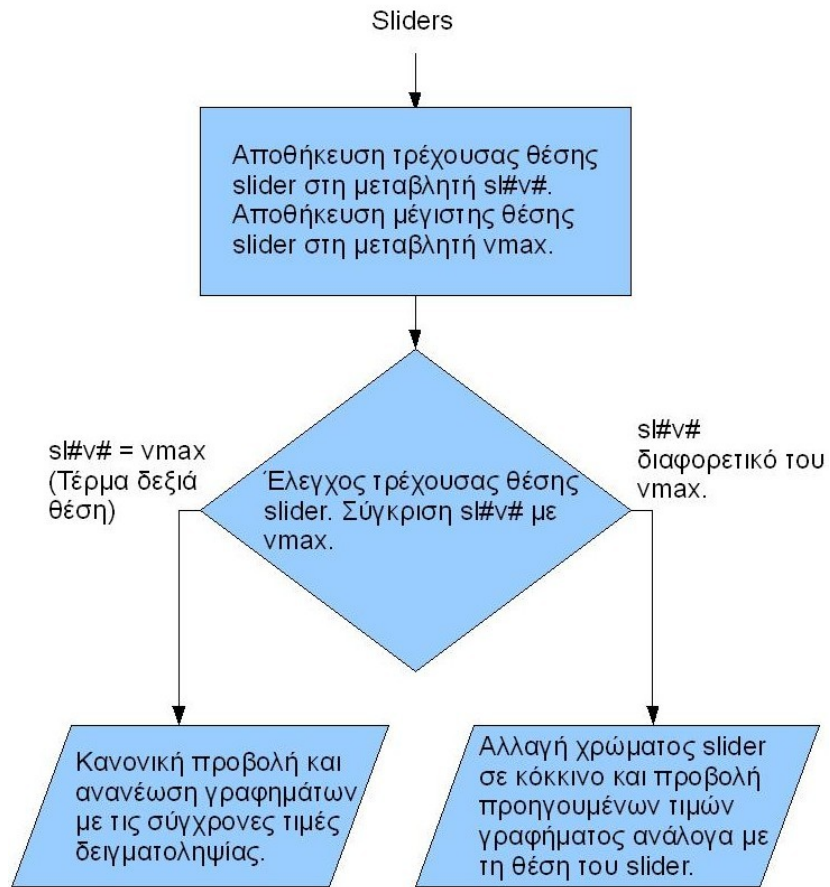


Κατά τη διαδικασία της παρακολούθησης των γραφημάτων, όπως ειπώθηκε παραπάνω, ο χρήστης έχει την επιλογή να επιστρέψει σε προηγούμενες τιμές μετακινώντας τον ολισθητή (slider), που βρίσκεται κάτω από κάθε γράφημα, προς την αριστερή του θέση. Με την ενέργεια αυτή η μπάρα ολίσθησης χρωματίζεται κόκκινη. Για το διάστημα που παρακολουθούνται οι προηγούμενες τιμές στο γράφημα, συνεχίζουν να προστίθενται νέες μετρήσεις, όμως αυτές δεν φαίνονται και το γράφημα παραμένει σταθερό στην περιοχή των τιμών που το άφησε ο χρήστης. Όταν αυτός αποφασίσει να παρακολουθήσει τις νέες τιμές μετακινεί πάλι τον slider στην τέρμα δεξιά θέση, με αποτέλεσμα να «κλειδώσει» και να εμφανίζει αυτόματα τις νέες τιμές καθώς αυτές λαμβάνονται. Στην θέση αυτή η μπάρα του slider έχει χρώμα γκρι.



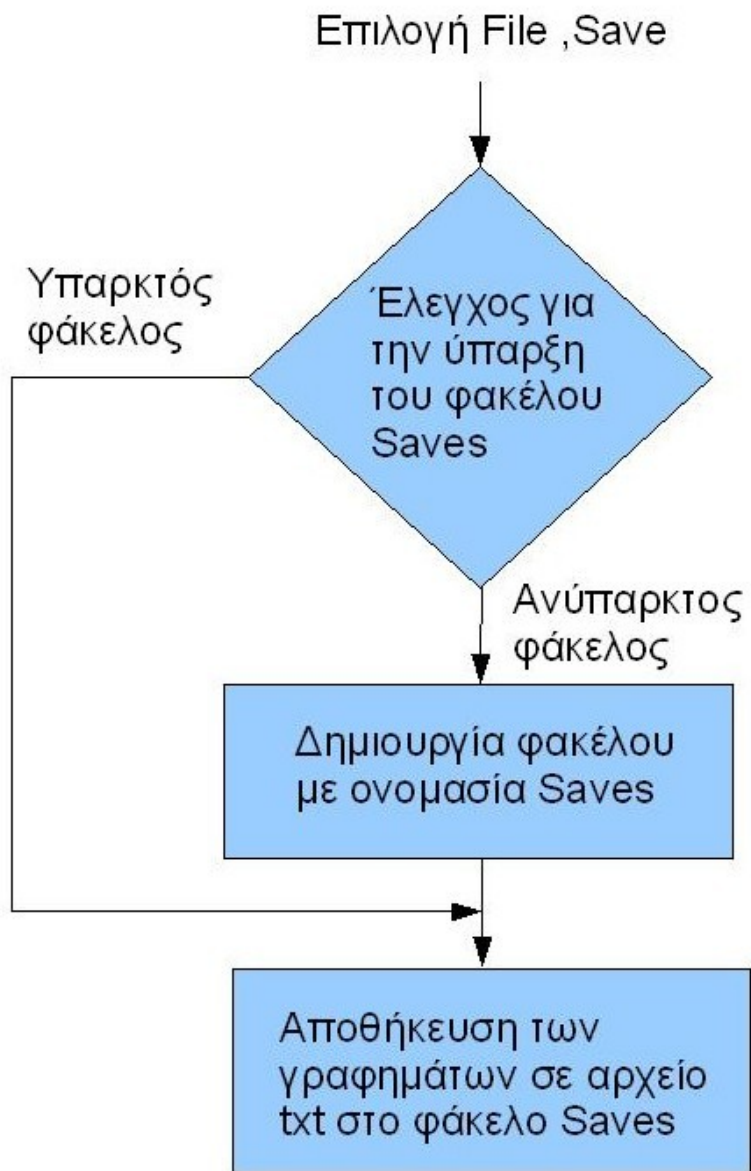
Παρακολούθηση τιμών αισθητήρων με ενεργοποιημένους τους sliders.

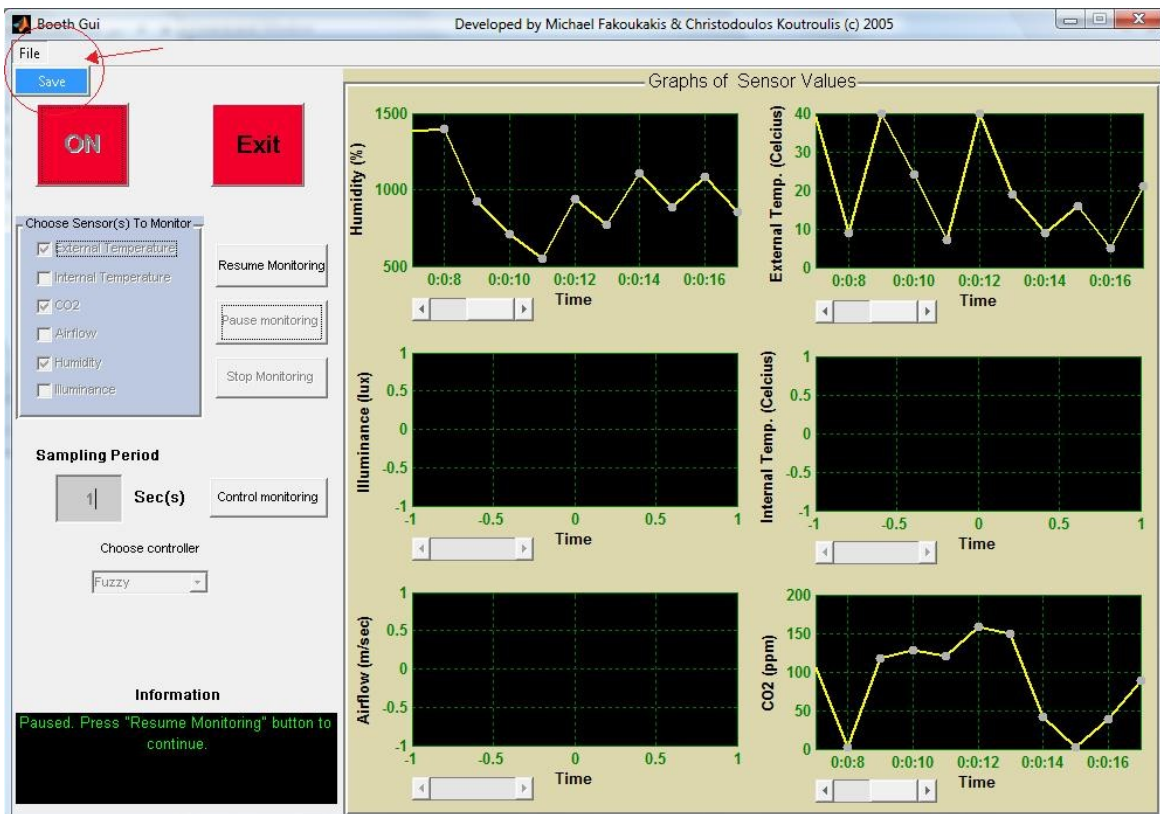
Ακολουθεί το διάγραμμα των slider :



* Όπου # ο αριθμός του επιλεγμένου slider. Περιοχή τιμών 1-6.

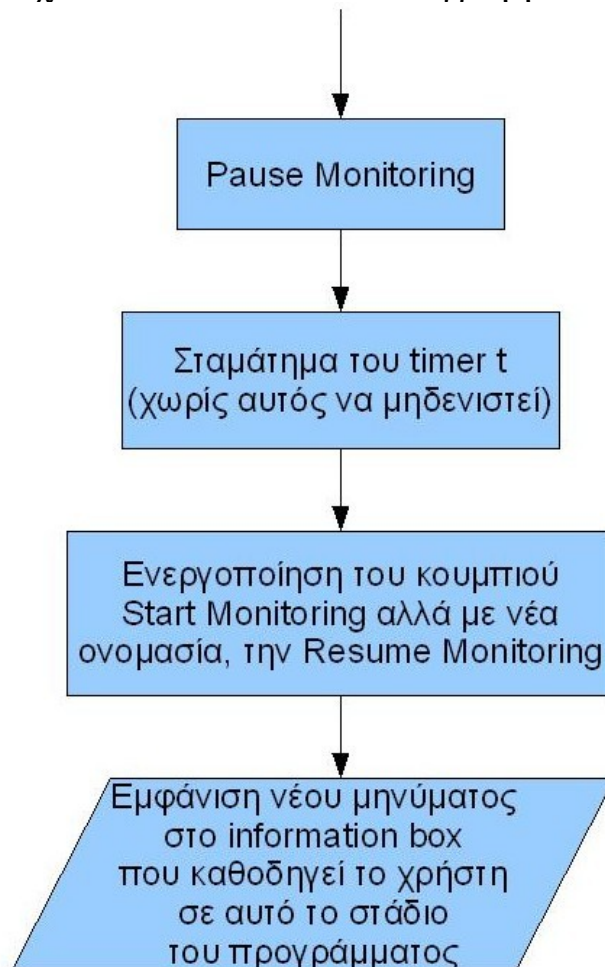
Στην επάνω αριστερά γωνία του παραθύρου υπάρχει ένα μενού “File” που όταν επιλεγθεί εμφανίζει την υπό-επιλογή “Save”. Αυτή αποθηκεύει σε ένα αρχείο κειμένου .txt όλες τις τιμές των αισθητήρων και των μηχανισμών ελέγχου που λήφθηκαν από την αρχή της λειτουργίας του θαλάμου. Μάλιστα δημιουργείται αυτόματα ένας φάκελος “Saves” (εφ’ όσον δεν υπάρχει ήδη) στη διεύθυνση που βρίσκεται το αρχείο booth.m και το αρχείο κειμένου αποθηκεύεται εκεί με όνομα «ημερομηνία_ώρα.txt». Η συνάρτηση που ενεργοποιείται με την επιλογή του menu “Save” είναι η Save_Callback η οποία φαίνεται παρακάτω.

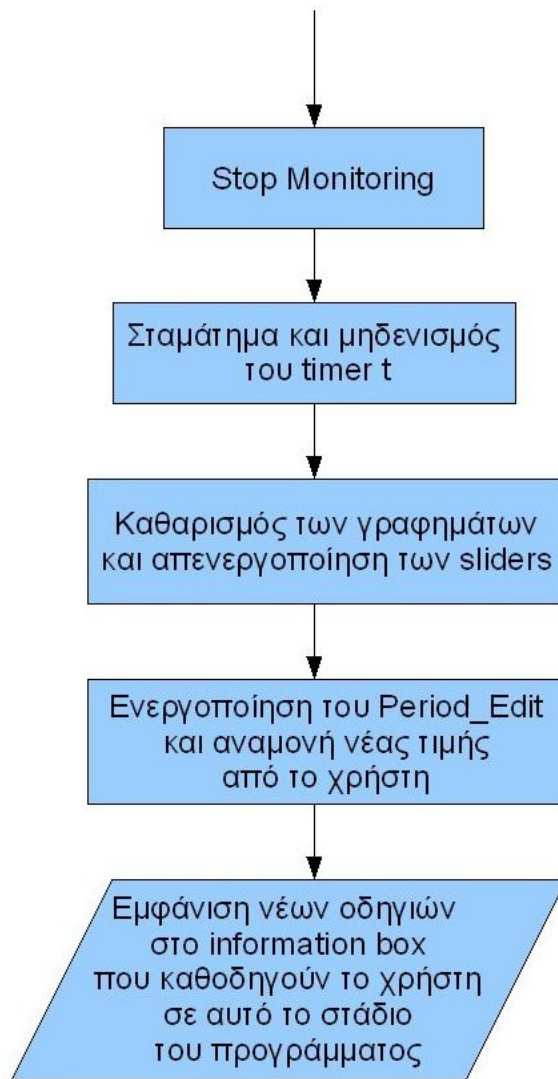




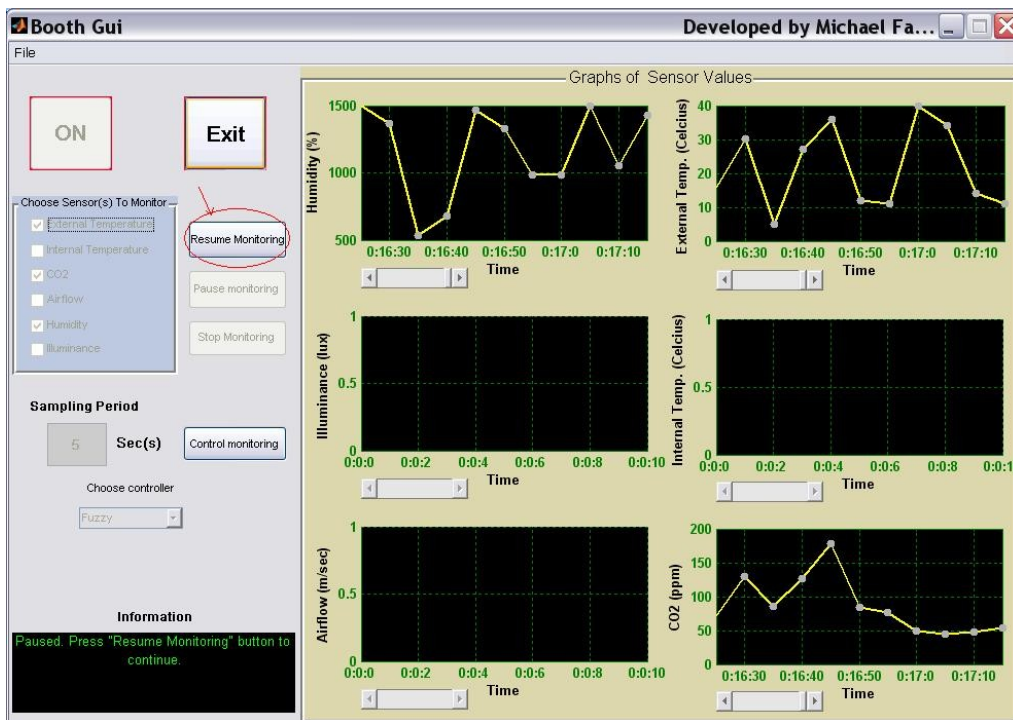
Εικόνα του παραθύρου κατά την επιλογή του “Save”.

Κατά την εκκίνηση δειγματοληψίας από τη συνάρτηση monitor ενεργοποιούνται οι συναρτήσεις “pause monitoring” και “stop monitoring”. Όπως είναι κατανοητό εκτελούνται κάνοντας κλικ στα αντίστοιχα κουμπιά του περιβάλλοντος. Οι λειτουργίες τους είναι να κάνουν παύση στην παρακολούθηση, και να την σταματήσουν αντίστοιχα. Ακολουθούν τα διαγράμματα του κάθε κουμπιού :

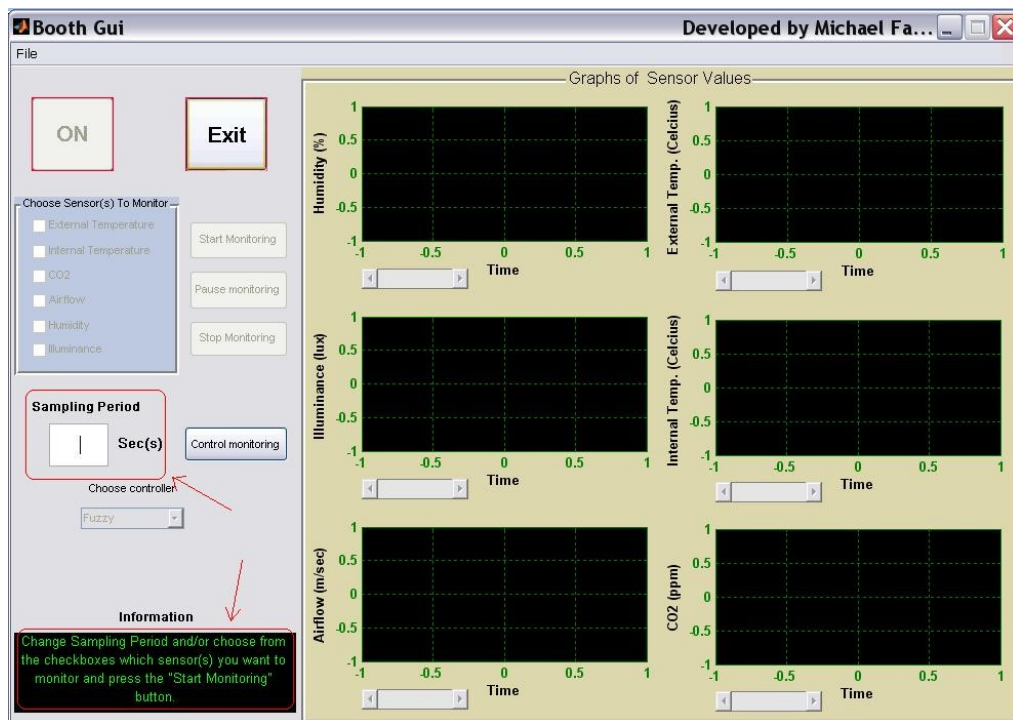




Αν επιλεγεί το κουμπί “Pause monitoring” τότε σταματάει η προβολή τιμών στα γραφήματα και ενεργοποιείται το κουμπί “start monitoring” με ονομασία “resume monitoring” πλέον, το οποίο συνεχίζει την προβολή των τιμών στα γραφήματα από το σημείο που είχαν σταματήσει. Αν επιλεγεί το κουμπί “Stop monitoring” τότε σταματάει η δειγματοληψία, τα γραφήματα διαγράφονται και ο κώδικας επιστρέφει στο σημείο όπου ο χρήστης εισάγει νέο χρόνο δειγματοληψίας με επόμενο βήμα την επιλογή αισθητήρα κ.λπ. Ακολουθεί εικόνα με το περιβάλλον σε κατάσταση pause monitoring και σε κατάσταση stop monitoring.



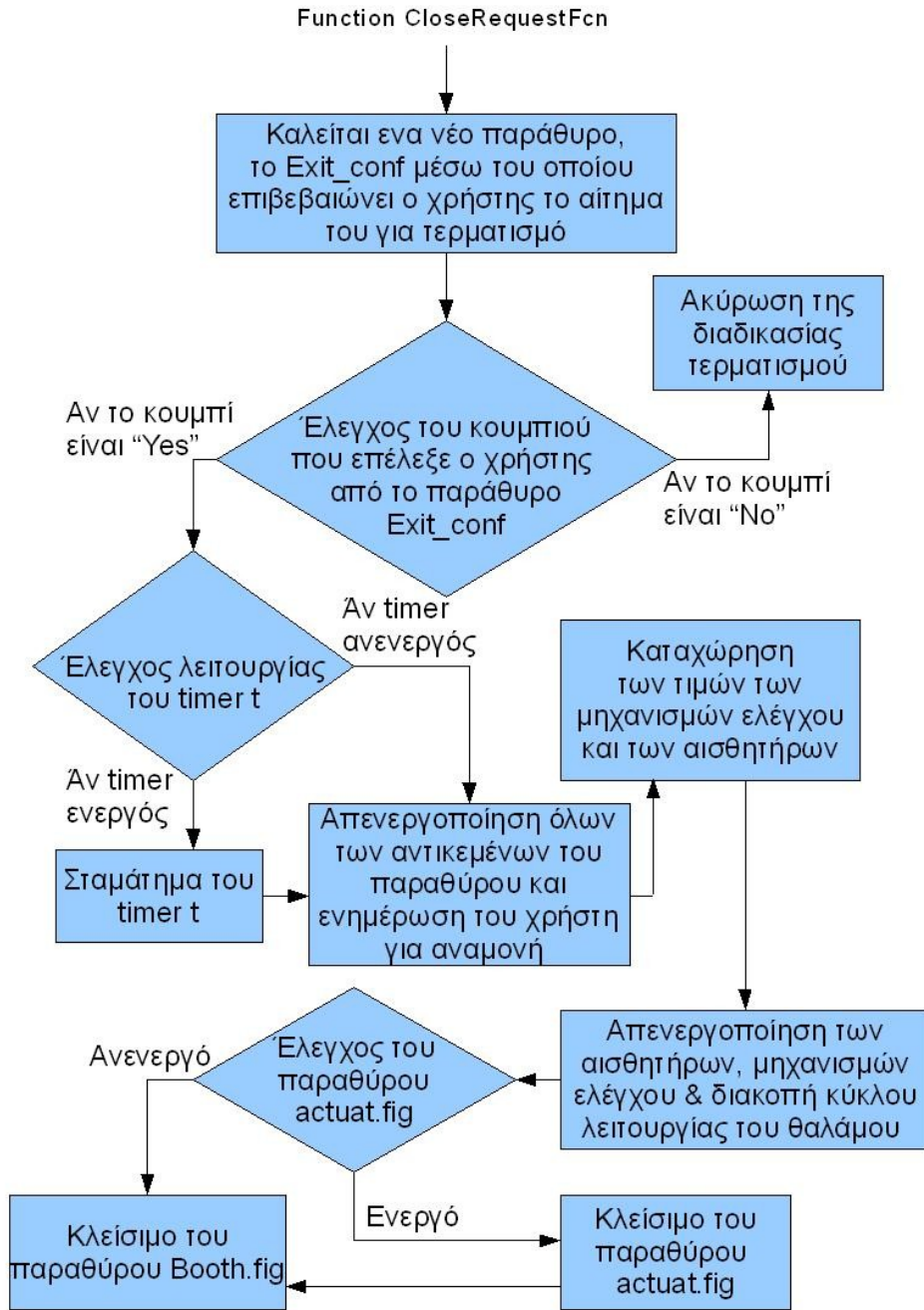
Το περιβάλλον σε κατάσταση Pause monitoring.



Το περιβάλλον σε κατάσταση Stop Monitoring.

Από τη στιγμή που επιλέγεται ποιος controller θα χρησιμοποιηθεί ενεργοποιείται και το κουμπί Exit το οποίο απλά τερματίζει την εφαρμογή. Διάγραμμα για τη συνάρτηση του Exit δεν παρατίθεται αφού αυτή αποτελείται από μία μόνο εντολή η οποία κλείνει το παράθυρο booth. Υπάρχει και ο εναλλακτικός τρόπος να τερματιστεί η εφαρμογή, με το να πατηθεί το κόκκινο “x” που βρίσκεται επάνω δεξιά στο παράθυρο. Ανεξάρτητα από το πως θα κλείσει ο χρήστης το παράθυρο, είναι αναγκαίο να τεθεί και ο θάλαμος εκτός λειτουργίας και μάλιστα με ομαλό τρόπο. Συγκεκριμένα πρέπει να απενεργοποιηθούν οι αισθητήρες, να κλειστούν τα παράθυρα, τα στόρια, το φως, ο κλιματισμός κτλ. Έτσι χρησιμοποιήθηκε μια συνάρτηση που εκτελείται

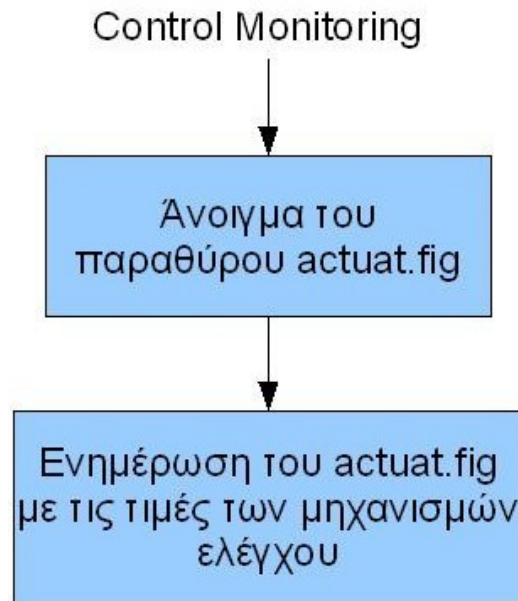
αυτόματα όταν κλείνει το παράθυρο είτε με τον ένα είτε με τον άλλο τρόπο και είναι υπεύθυνη για το σωστό τερματισμό του θαλάμου. Αυτή λέγεται figure1_CloseRequestFcn και η ροή της φαίνεται παρακάτω.



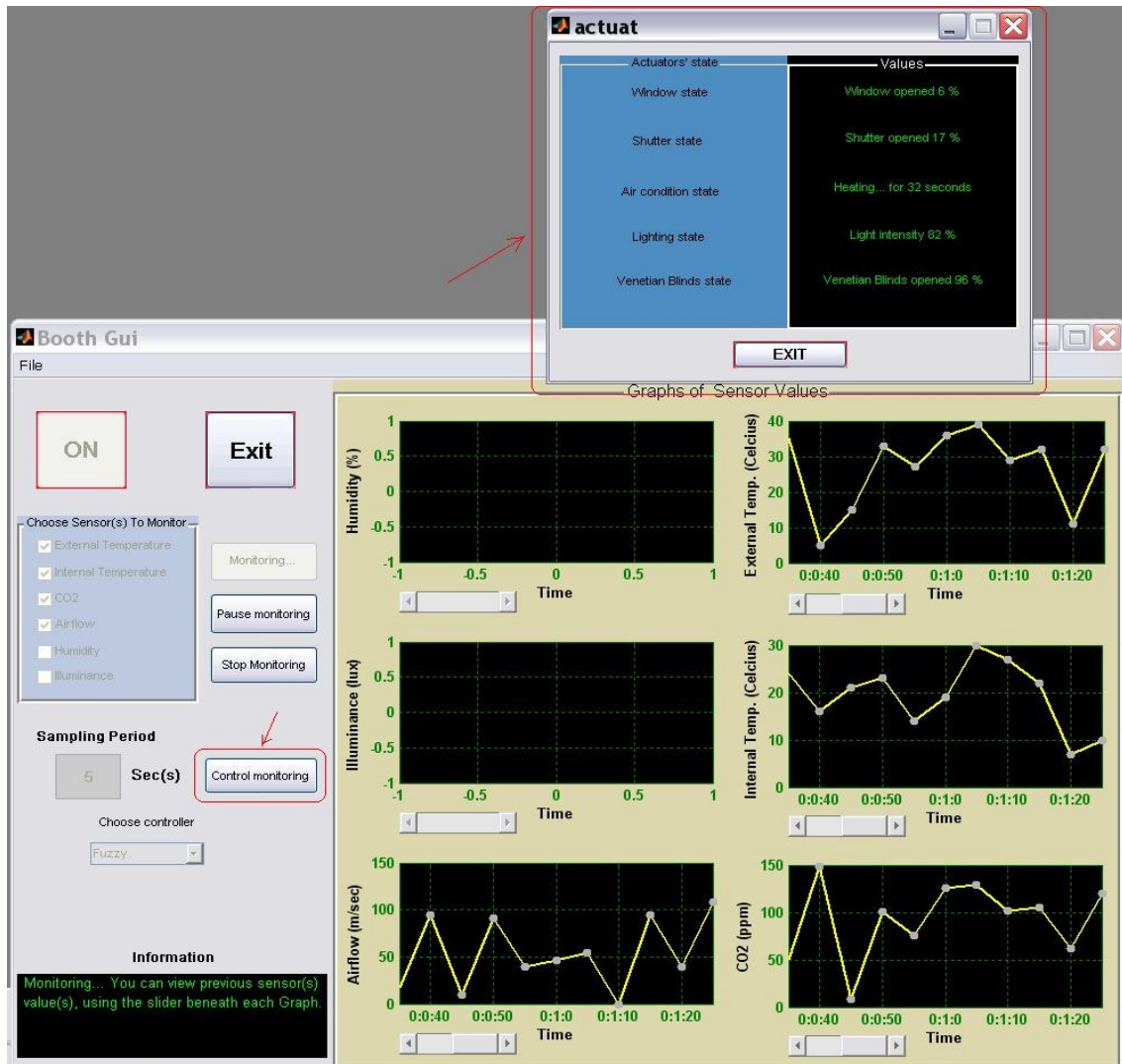
Στο σημείο αυτό ο χρήστης έχει χρησιμοποιήσει όλες τις διαθέσιμες για δειγματοληψία επιλογές. Στο επόμενο κεφάλαιο αναλύονται τα gui actuat.fig και exit_conf.fig .

4.7 Περιγραφή των Gui `actuat.m` και `exit_conf.m`

Μια ακόμη επιλογή η οποία ενεργοποιείται αμέσως μετά τη διαδικασία της αρχικοποίησης είναι το κουμπί “control monitoring”, το οποίο ξεκινάει το `actuat.fig` περιβάλλον που ασχολείται με τις τιμές που στέλνει ο αλγόριθμος ασαφούς λογικής στους μηχανισμούς ελέγχου του θαλάμου. Η συνάρτηση του κουμπιού παριστάνεται σχηματικά παρακάτω :



Το καινούριο gui με ονομασία “`actuat.m`” και “`actuat.fig`” που εκτελείται ανοίγει σε ένα δεύτερο παράθυρο για την ταυτόχρονη παρακολούθηση τιμών αισθητήρων και μηχανισμών ελέγχου από το χρήστη. Τα περιεχόμενα του `actuat.m` δεν αναπαριστώνται γραφικά για το λόγο ότι είναι πολύ απλά δομημένο. Αυτό αποτελείται από μερικά Static Texts που αναγράφουν τις τιμές των μηχανισμών ελέγχου τη στιγμή που αυτές ανανεώνονται, ενώ οι τιμές αυτές που προέρχονται από το `booth.m` μεταφέρονται στο `actuat` μέσω κάποιων global μεταβλητών. Τέλος το παράθυρο έχει στο κάτω μέρος του ένα κουμπί με την ονομασία “Exit”, το οποίο όπως είναι προφανές απλά κλείνει το `actuat.fig`. Ακολουθεί εικόνα του gui `actuat.fig` μαζί με το `booth.fig`.



Το τρίτο gui που χρησιμοποιήθηκε είναι το exit_conf.fig . Χρησιμοποιείται για να προειδοποιήσει τον χρήστη για την επιλογή του να τερματίσει το περιβάλλον δειγματοληψίας booth.fig. Πρόκειται για ένα μικρό παράθυρο με ένα Static Text και δύο κουμπιά, τα Yes & No. Ακολουθεί εικόνα του exit_conf.



To gui exit_conf για την επιβεβαίωση τερματισμού του περιβάλλοντος.

Το exit_conf.fig συνοδεύεται από το exit_conf.m το οποίο αποτελείται από δύο πολύ απλές συναρτήσεις. Μία για το κουμπί Yes και μία για το κουμπί No, αν πατηθεί το πρώτο κουμπί τότε μια μεταβλητή σε ρόλο flag παίρνει την τιμή 1, αν πατηθεί το δεύτερο κουμπί τότε το flag παίρνει την τιμή 0.

5 Συμπεράσματα.

Το παραθυρικό περιβάλλον παρακολούθησης και ελέγχου, εξυπηρετεί τον πειραματικό θάλαμο έως τις μέχρι τώρα δυνατότητες του. Στην περίπτωση που αυτός επεκτείνει τις λειτουργίες του ή ακόμα αλλάξει το βασικό τρόπο εργασίας του, το παρόν GUI μπορεί να τροποποιηθεί ανάλογα ώστε να συνεχίσει να τον εξυπηρετεί. Συγκεκριμένα, υπάρχει η σκέψη αλλαγής του τρόπου ελέγχου των συσκευών του θαλάμου ώστε από έλεγχος ασαφούς λογικής που χρησιμοποιεί τώρα, να αλλάξει σε PID και/ή On-Off. Ήδη υπάρχουν αυτές οι δύο επιλογές στο Pop-Up Menu του παραθύρου, όμως δε λειτουργούν στην πραγματικότητα. Με τις κατάλληλες τροποποιήσεις και προσθήκες σε διάφορα σημεία του κώδικα, ο χρήστης θα μπορεί να εκκινεί και να λειτουργεί το θάλαμο με τον ελεγκτή της επιλογής του με σκοπό τον πειραματισμό και δίνοντας τη δυνατότητα συγκριτικών παρουσιάσεων του θαλάμου. Υποθετικά θα μπορούσαν να αλλάξουν ή να προστεθούν κάποιοι αισθητήρες ή μηχανισμοί, σε αυτήν την περίπτωση αντίστοιχα με λίγες γραμμές εντολών στις κατάλληλες θέσεις και αλλαγές στο γραφικό μέρος του παραθύρου (αύξηση μεγέθους, προσθήκη επιπλέον αξόνων κ.α.) το περιβάλλον θα είναι πάλι σε θέση να συνεργαστεί πλήρως με τον ανανεωμένο θάλαμο. Αλλαγές μπορούν να γίνουν και χωρίς αυτές να προϋποθέτουν τροποποιήσεις στον ίδιο το θάλαμο αλλά μόνο στο ίδιο το παραθυρικό περιβάλλον. Έτσι αναλόγως με τις ανάγκες που μπορεί να παρουσιαστούν στην πορεία, εύκολα μπορούν να προστεθούν κουμπιά (και διαφορετικού είδους στοιχεία) που να κάνουν τις νέες εργασίες που απαιτούνται,

όπως για παράδειγμα να καλείται ένα νέο παράθυρο που να αναπαριστά με γραφικό τρόπο τα μηνύματα που διατρέχουν το δίαυλο επικοινωνίας (bus) την κάθε δεδομένη στιγμή, ώστε να δίνεται στο χρήστη ή θεατή η δυνατότητα μιας εκ βαθέων παρακολούθησης του τρόπου λειτουργίας του θαλάμου.

Αναλύθηκε το πως το καθ' όλα απλό και φιλικό γραφικό περιβάλλον με τα ευανάγνωστα και ευπαρουσίαστα στοιχεία του, συγκροτήθηκε από εκατοντάδες γραμμές κειμένου που αναπαράστησαν δεκάδες αριθμητικές πράξεις και λογικούς ελέγχους. Αναφέρθηκε το πως οι πρόθυμοι, προστατευτικοί και εξυηρητικοί αυτοματισμοί σε ένα χώρο εργασίας ή διαβίωσης ανθρώπων, πραγματοποιούνται με τον αρμονικό συνδυασμό πολύπλοκων τεχνολογιών μέτρησης, ελέγχου και επικοινωνίας και κατορθώνουν να δώσουν προσωπικότητα στο χώρο αυτό. Ο συνδυασμός των έξυπνων αυτοματισμών χώρου και της απλουστευμένης γραφικής διαχείρισης τους, είναι ένας πολλά υποσχόμενος συνδυασμός που είναι μοιραίο να επικρατήσει και να μπει στις ζωές όλων. Λέγεται πως η τεχνολογία δημιουργήθηκε από την ανάγκη του ανθρώπου να κάνει τη ζωή του ευκολότερη, μια ολιγόωρη διαμονή σε έναν έξυπνο χώρο ή ακόμα και μια απλή επίδειξη του πειραματικού θαλάμου, με το φιλικό κεντρικό έλεγχο που το συνοδεύει, συνηγορεί έντονα σε αυτή την άποψη.

Παράρτημα

Μέρος 1^ο (Booth.m)

```
function booth_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
%σχόλια με επεξηγήσεις που αφορούν τη συνάρτηση.
% handles    structure with handles and user data (see GUIDATA)

% varargin   command line arguments to booth (see VARARGIN)

%set(gcf,'CloseRequestFcn','Exit_butt_Callback(hObject, eventdata, handles)');

%Οι Global μεταβλητές που χρησιμοποιούνται.
global pause_b a1 a2 a3 a4 a5 a6 fl1 fl2 fl3 fl4 fl5 fl6 sampling_per ill_elec handl t quit1;

t=0;           % Το t χρησιμοποιείται για τον timer (στη συνάρτηση monitor).
a1=0;         % Αρχικά πρέπει να έχει μηδενική τιμή.
a2=0;
a3=0;         % Αρχικοποίηση μεταβλητών για τις λειτουργίες των axes και sliders.
a4=0;
a5=0;         % Τα a1-6 είναι “σημαίες” (flags) για την ενεργοποίηση των axes.
a6=0;         % Περιοχή τιμών των a1-6 είναι 0 , 1.  0 = ανενεργό, 1=ενεργό.
pause_b=0;    % Flag για το κουμπί Pause. 0 = ανενεργό, 1=ενεργο.
fl1=1;
fl2=1;
fl3=1;         % Τα fl1-6 είναι τα flags για την ενεργοποίηση των sliders.
fl4=1;         % 1=ανενεργό, οποιαδήποτε άλλη τιμή ενεργό.
```

```
fl5=1;  
fl6=1;
```

```
% Απενεργοποίηση όλων των επιλογών του περιβάλλοντος,  
% εκτός από το “popup menu” για την επιλογή ελεγκτή
```

```
set(handles.Exit_butt,'enable','off');  
set(handles.On_butt,'enable','off');  
set(handles.Send_mon,'enable','off');  
set(handles.Save,'enable','off');  
set(handles.Start_monitor_butt,'enable','off');  
set(handles.Stop_monitor_butt,'enable','off');  
set(handles.Pause_monitor_butt,'enable','off');  
set(handles.Ext_temp_checkbox,'enable','off');  
set(handles.Int_temp_checkbox,'enable','off');  
set(handles.humidity_checkbox,'enable','off');  
set(handles.CO2_checkbox,'enable','off');  
set(handles.Airflow_checkbox,'enable','off');  
set(handles.Ill_checkbox,'enable','off');  
set(handles.slider1,'enable','off');  
set(handles.slider2,'enable','off');  
set(handles.slider3,'enable','off');  
set(handles.slider4,'enable','off');  
set(handles.slider5,'enable','off');  
set(handles.slider6,'enable','off');  
set(handles.Period_edit,'enable','off');
```

```
% διευθέτηση χρώματος για διάφορα στοιχεία το παραθύρου
```

```
set(handles.Period_edit,'BackgroundColor',[0.8 0.8 0.8]);  
set(handles.axes1,'color',[0 0 0],'fontweight','bold');  
set(handles.axes2,'color',[0 0 0],'fontweight','bold');  
set(handles.axes3,'color',[0 0 0],'fontweight','bold');  
set(handles.axes4,'color',[0 0 0],'fontweight','bold');  
set(handles.axes5,'color',[0 0 0],'fontweight','bold');  
set(handles.axes6,'color',[0 0 0],'fontweight','bold');
```

```
set(handles.axes1,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);  
set(handles.axes2,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);  
set(handles.axes3,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);  
set(handles.axes4,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);  
set(handles.axes5,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);  
set(handles.axes6,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
```

```
% Ονομασία αξόνων των γραφημάτων.
```

```
ylabel(handles.axes1,'External Temp. (Celcius)','color',[0 0 0],'fontweight','bold');  
ylabel(handles.axes2,'Internal Temp. (Celcius)','color','k','fontweight','bold');  
ylabel(handles.axes3,'CO2 (ppm)','color','k','fontweight','bold');  
ylabel(handles.axes4,'Airflow (m/sec)','color','k','fontweight','bold');  
ylabel(handles.axes5,'Humidity (%)','color','k','fontweight','bold');  
ylabel(handles.axes6,'Illuminance (lux)','color','k','fontweight','bold');  
xlabel(handles.axes1,'Seconds','color','k','fontweight','bold');  
xlabel(handles.axes2,'Seconds','color','k','fontweight','bold');  
xlabel(handles.axes3,'Seconds','color','k','fontweight','bold');  
xlabel(handles.axes4,'Seconds','color','k','fontweight','bold');  
xlabel(handles.axes5,'Seconds','color','k','fontweight','bold');  
xlabel(handles.axes6,'Seconds','color','k','fontweight','bold');  
ylabel(handles.axes1,'External Temp. (Celcius)','color',[0 0 0],'fontweight','bold');
```

```

ylabel(handles.axes2,'Internal Temp. (Celcius)','color','k','fontweight','bold');
ylabel(handles.axes3,'CO2 (ppm)','color','k','fontweight','bold');
ylabel(handles.axes4,'Airflow (m/sec)','color','k','fontweight','bold');
ylabel(handles.axes5,'Humidity (%)','color','k','fontweight','bold');
ylabel(handles.axes6,'Illuminance (lux)','color','k','fontweight','bold');

set (handles.popupmenu1,'Value',4);
% Μήνυμα πληροφόρησης του χρήστη για την επιλογή ελεγκτή.
% Αυτόματα (από τη συνάρτηση function popupmenu1) ενεργοποιείται το κουμπί ON.
set (handles.State_text,'string','Welcome! Choose controller from the popup menu above to
begin... ');

% Εκκίνηση του OPC server.
hr = mxOPC('Open', 'ICON.EIBOPCServer.1');
% Επιλογές λειτουργίας του server.
mxOPC('ReadMode', 'Device');

% Choose default command line output for booth %Κώδικας προερχόμενος από τη
matlab.
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes booth wait for user response (see UIRESUME)
% uiwait(handles.figure1);
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell array
% contents{get(hObject,'Value')} returns selected item from popupmenu1

% Ορισμός global μεταβλητής για την επιλογή του ελεγκτή.
global cont;

% Εισαγωγή σε μεταβλητή της επιλογής ελεγκτή (1 για fuzzy, 2 για PID, 3 για ON/OFF)
cont=get(hObject,'Value');

% Έλεγχος τιμής της μεταβλητής cont. Στην παρούσα φάση το πρόγραμμα συνεχίζει να
εκτελείται μόνο αν επιλεγεί ο fuzzy controller.
if cont==1
    set (handles.State_text,'string','You chose Fuzzy controller.Press On button to begin! ');
    set (handles.On_but,'enable','on');
    set (handles.Exit_but,'enable','on');
elseif cont==2
    set (handles.State_text,'string','You chose PID controller.This controller is not available
yet. Please choose another one to continue. ');
    set (handles.On_but,'enable','off');
    set (handles.Exit_but,'enable','on');
elseif cont==3
    set (handles.State_text,'string','You chose ON/OFF controller.This controller is not
available yet. Please choose another one to continue. ');
    set (handles.On_but,'enable','off');
    set (handles.Exit_but,'enable','on');
end

```



```

% --- Executes on button press in On_butt.
function On_butt_Callback(hObject, eventdata, handles)
% hObject handle to On_butt (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% A. Pouliezos
% v1.0, 22/12/03

% co2: ppm (0-2000)
% ill: Φωτεινότητα (lux 0-2000)
% Ta: Εσωτερική θερμοκρασία (C° -10 - +40)
% Tout: Εξωτερική θερμοκρασία (C° -10 - +40)
% rh: Σχετική υγρασία (% 0-100)
% v: Εσωτερική ταχύτητα ανέμου (m/s 0-16)
% Τα παραπάνω μετριοούνται στην κλίμακα των 0-10V σε millivolts

global sampling_per ill_elec turnoff A_X A_T strt_f sft sft1 pw ps light_int_out coh handl
enabl cool_dur heat_dur shadp;

enabl=0; % Flag που με την τιμή «0» δείχνει ότι το παράθυρο με την
κατάσταση των % μηχανισμών είναι απενεργοποιημένο.

% Ενημέρωση χρήστη για την κατάσταση αρχικοποίησης.
set(handles.State_text,'string','Please wait while initializing...');
set(handles.On_butt,'enable','off');
set(handles.popupmenu1,'enable','off');
set(handles.Period_edit,'enable','off');

% actuators % «Διευθύνσεις ομάδος» λειτουργιών στους μηχανισμούς ελέγχου του
θαλάμου.
open_window = 'main I/O.window-shutter.0/0/1';
close_window = 'main I/O.window-shutter.0/0/2';
open_shutter = 'main I/O.window-shutter.0/0/3';
close_shutter = 'main I/O.window-shutter.0/0/4';
use_AC = 'main I/O.ventilation-heating.0/3/1';
use_valve = 'main I/O.ventilation-heating.0/3/0';
dimmer = 'main I/O.dimmer.0/4/4';
move_ind_shutter_cw = 'main I/O.window-shutter.0/0/5';
move_ind_shutter_acw = 'main I/O.window-shutter.0/0/6';

% sensors % «Διευθύνσεις ομάδος» λειτουργιών αισθητήρων του θαλάμου.
temp_out = 'main I/O.sensors.0/1/0';
temp_in = 'main I/O.sensors.0/1/2';
CO_2 = 'main I/O.sensors.0/1/3';
airflow = 'main I/O.sensors.0/1/4';
light_intensity = 'main I/O.sensors.0/1/5';
smoke_intensity = 'main I/O.sensors.0/1/6';
smoke_alarm = 'main I/O.sensors.0/1/7';
smoke_detector_status = 'main I/O.sensors.0/1/8';
humidity = 'main I/O.sensors.0/1/9';

% initialize sensor values % Αρχικοποίηση τιμών αισθητήρων.
temp_int_prev = 0;
temp_ext_prev = 0;

```

```

CO2_prev = 0;
airflow_prev = 0;
light_int_prev = 0;
smoke_int_prev = 0;
hum_prev = 0;

on = 1;
off = 0;
fault_perc = 10;
fault_id = 0;
fault_time = 20;
    AC_del = 16;
sft = num2str(fault_perc);
sft1 = num2str(fault_id);
pw = 0;
tot_dur_w = 10;
tot_dur_s = 6;
tot_dur_ind_shut = 3;
CO2_warmup = 90;
sampling_per = 120;
% Κατώφλι έντασης καπνού (3% x 15)
smoke_int_thr = 45;
% Φωτεινότητα λόγω τεχνητού φωτισμού (μετρούμενη μια φορά)
ill_elec = 111;

% επανατοποθέτηση όλων των μηχανισμών στο μηδέν.

% Σβήσιμο της λάμπας (φως)
hr2 = mxOPC('WriteDouble', dimmer, off);
illp = 0;

% Απενεργοποίηση του συναγερμού καπνού
hr2 = mxOPC('WriteBool', smoke_alarm, off);
alarm_flag = off;

% Απενεργοποίηση ελεγκτών για παράθυρα και στόρια
hr2 = mxOPC('WriteBool', close_window, off);
hr2 = mxOPC('WriteBool', open_window, off);
hr2 = mxOPC('WriteBool', open_shutter, off);
hr2 = mxOPC('WriteBool', close_shutter, off);
hr2 = mxOPC('WriteBool', move_ind_shutter_cw, off);
hr2 = mxOPC('WriteBool', move_ind_shutter_acw, off);
% Απενεργοποίηση AC
hr2 = mxOPC('WriteBool', use_AC, off);
pause (AC_del);
hr2 = mxOPC('WriteBool', use_valve, off);
    valve_pos = 0;

% Κλείσιμο παραθύρου
hr2 = mxOPC('WriteBool', close_window, on);
pause (tot_dur_w + 1)
hr2 = mxOPC('WriteBool', close_window, off);
pw = 0;

% Κλείσιμο στοριών

```

```

hr2 = mxOPC('WriteBool', close_shutter, on);
pause (tot_dur_s + 1)
hr2 = mxOPC('WriteBool', close_shutter, off);
ps = 1;
shadp = 1;
if enabl==1
    set(handles.blinds,'string',['Venetian Blinds opened ' num2str(100-(shadp*100)) '
%]);
end

% Ανοιγμα στοριών
hr2 = mxOPC('WriteBool', open_shutter, on);
pause (tot_dur_shut + 1)
hr2 = mxOPC('WriteBool', open_shutter, off);
ps = 100;

% Τοποθέτηση εσωτερικών στοριών σε μεσαίο επίπεδο
% Αρχικά μετακίνηση σε μηδενική θέση δεξιόστροφα
hr2 = mxOPC('WriteBool', move_ind_shutter_cw, on);
pause (tot_dur_ind_shut + 1)
hr2 = mxOPC('WriteBool', move_ind_shutter_cw, off);
% και μετά στη μέση αριστερόστροφα
hr2 = mxOPC('WriteBool', move_ind_shutter_acw, on);
pause (1)
hr2 = mxOPC('WriteBool', move_ind_shutter_acw, off);

% Αναμονή <CO2_warmup> δευτερόλεπτα. Για το ζέσταμα του αισθητήρα CO2
pause (CO2_warmup - (tot_dur_s - tot_dur_w));
% Ενεργοποίηση AC
hr2 = mxOPC('WriteBool', use_AC, on);

t0 = clock;
strt = [num2str(t0(3)) '/' num2str(t0(2)) '/' num2str(t0(1)) ' ' num2str(t0(4)) ' '
num2str(t0(5)) "' ' num2str(round(t0(6))) "' '];
strt_f = [num2str(t0(3)) '_' num2str(t0(2)) '_' num2str(t0(1)) '_' num2str(t0(4)) '_'
num2str(t0(5)) '_' num2str(round(t0(6)))];

set (handles.Period_edit,'enable','on');
set (handles.Period_edit,'BackgroundColor',[1 1 1]);
% Πληροφορία για το χρήστη. Εισαγωγή περιόδου δειγματοληψίας.
set (handles.State_text,'string','Insert the Sampling Period (in seconds) in the "Sampling
Period" box and press the enter key.');
```

turnoff=0; **%Σημαία που με την τιμή «0» δείχνει ότι δεν έχει πατηθεί το κουμπί EXIT**
for i = 1: 360 **% Ορισμός 360 κύκλων για τον κώδικα που ακολουθεί**

```

if turnoff==1 %Εάν πατηθεί το κουμπί EXIT
    break % να διακοπεί η επανάληψη
end

% Τοποθέτηση δεδομένων στον buffer
% Εσωτερική θερμοκρασία
[temp_int, hrts, tts, qqs] = mxOPC('ReadDouble', temp_in);
% Εξωτερική θερμοκρασία
```

```

[temp_ext, hrts, tts, qqs] = mxOPC('ReadDouble', temp_out);
% CO2
[Co2, hrts, tts, qqs] = mxOPC('ReadDouble', CO_2);
% Ροή αέρα
[air_fl, hrts, tts, qqs] = mxOPC('ReadDouble', airflow);
% Ένταση φωτός
[light_int, hrts, tts, qqs] = mxOPC('ReadDouble', light_intensity);
% Ένταση καπνού
[smoke_int, hrts, tts, qqs] = mxOPC('ReadDouble', smoke_intensity);
% Υγρασία
[hum, hrts, tts, qqs] = mxOPC('ReadDouble', humidity);

% Εσωτερική θερμοκρασία
[temp_int, h1, tt1, q1] = mxOPC('ReadDouble', temp_in);
pause (1)
[temp_int, h1, tt1, q1] = mxOPC('ReadDouble', temp_in);
if h1 == 0
% success
temp_int_prev = temp_int;
else
% failure
temp_int = temp_int_prev;
end
Ta(i) = temp_int;

% Εξωτερική θερμοκρασία
[temp_ext, h2, tt2, q2] = mxOPC('ReadDouble', temp_out);
pause (1)
[temp_ext, h2, tt2, q2] = mxOPC('ReadDouble', temp_out);
if h2 == 0
% success
temp_ext_prev = temp_ext;
else
% failure
temp_ext = temp_ext_prev;
end
% Μετατροπή σε C°
Tout(i) = -10 + (temp_ext/1000)*5;

% CO2
[Co2, h3, tt3, q3] = mxOPC('ReadDouble', CO_2);
pause (1)
[Co2, h3, tt3, q3] = mxOPC('ReadDouble', CO_2);
if h3 == 0
% success
CO2_prev = Co2;
else
% failure
Co2 = CO2_prev;
end
CO2(i) = (Co2/1000) * 200;

% Ροή αέρα
[air_fl, h4, tt4, q4] = mxOPC('ReadDouble', airflow);
pause (1)

```

```

[air_fl, h4, tt4, q4] = mxOPC('ReadDouble', airflow);
    if h4 == 0
%       success
        airflow_prev = air_fl;
    else
%       failure
        air_fl = airflow_prev;
    end
    v(i) = (air_fl/1000)*1.6;
    v(i) = 0.05;

% Φωτεινότητα
[light_int, h5, tt5, q5] = mxOPC('ReadDouble', light_intensity);
    pause (1)
[light_int, h5, tt5, q5] = mxOPC('ReadDouble', light_intensity);
    if h5 == 0
%       success
        light_int_prev = light_int;
    else
%       failure
        light_int = light_int_prev;
    end
    ill(i) = (light_int/1000)*200;

% Ένταση καπνού
[smoke_int, h6, tt6, q6] = mxOPC('ReadDouble', smoke_intensity);
    if h6 == 0
%       success
        smoke_int_prev = smoke_int;
    else
%       failure
        smoke_int = smoke_int_prev;
    end
    if smoke_int > smoke_int_thr
%       Περίπτωση υψηλού επιπέδου καπνού, ενεργοποίηση συναγερμού
        if alarm_flag == on
            else
                hr2 = mxOPC('WriteBool', smoke_alarm, on);
                alarm_flag = on;
            end
        else
%       Περίπτωση κανονικού επιπέδου καπνού
            if alarm_flag == off
                else
                    hr2 = mxOPC('WriteBool', smoke_alarm, off);
                    alarm_flag = off;
                end
            end
        end

% Υγρασία
[hum, h7, tt7, qq7] = mxOPC('ReadDouble', humidity);
    pause (1)
[hum, h7, tt7, qq7] = mxOPC('ReadDouble', humidity);
    if h7 == 0
%       success

```



```

        hum_prev = hum;
    else
        failure
        hum = hum_prev;
    end
    humid(i) = hum/100;

    [num2str(tt1(3)) '/' num2str(tt1(2)) '/' num2str(tt1(1)) ' ' num2str(tt1(4)) ' '
num2str(tt1(5)) "' ' num2str(round(tt1(6))) ''": ' num2str(Ta(i)) ' C°']
    [num2str(tt2(3)) '/' num2str(tt2(2)) '/' num2str(tt2(1)) ' ' num2str(tt2(4)) ' '
num2str(tt2(5)) "' ' num2str(round(tt2(6))) ''": ' num2str(Tout(i)) ' C°']
    [num2str(tt3(3)) '/' num2str(tt3(2)) '/' num2str(tt3(1)) ' ' num2str(tt3(4)) ' '
num2str(tt3(5)) "' ' num2str(round(tt3(6))) ''": ' num2str(CO2(i)) 'ppm']
    [num2str(tt4(3)) '/' num2str(tt4(2)) '/' num2str(tt4(1)) ' ' num2str(tt4(4)) ' '
num2str(tt4(5)) "' ' num2str(round(tt4(6))) ''": ' num2str(v(i)) 'm/s']
    [num2str(tt5(3)) '/' num2str(tt5(2)) '/' num2str(tt5(1)) ' ' num2str(tt5(4)) ' '
num2str(tt5(5)) "' ' num2str(round(tt5(6))) ''": ' num2str(ill(i)) ' lux']
    [num2str(tt7(3)) '/' num2str(tt7(2)) '/' num2str(tt7(1)) ' ' num2str(tt7(4)) ' '
num2str(tt7(5)) "' ' num2str(round(tt7(6))) ''": ' num2str(humid(i)) '%']

% Υπολογισμός τιμών μηχανισμών ελέγχου και κίνησης
if i > 1
if i >= fault_time
% εξομοίωση λάθους αισθητήρων
switch fault_id
case 0
% χωρίς λάθος
case 1
% εσωτερική θερμοκρασία
Ta(i) = Ta(i)*(1 + fault_perc)/100;
case 2
% φωτεινότητα
ill(i) = ill(i)*(1 + fault_perc)/100;
case 3
% CO2
CO2(i) = CO2(i)*(1 + fault_perc)/100;
case 4
% Υγρασία
humid(i) = humid(i)*(1 + fault_perc)/100;
end
end
act_val = fuz_con(CO2(i), CO2_prev1, ill(i), Ta(i), Tout(i), humid(i), v(i),
shadp, illp);
[ill(i) shadp illp act_val(4) act_val(5)]
% Θέση στοριών
shadp = act_val(4)
pmv(i) = act_val(6);

% Φωτισμός (0-1) ως 0-100
if act_val(5) > 1
light_int_c = 1;
else
light_int_c = act_val(5);
end
light_int_out = light_int_c * 100
hr2 = mxOPC('WriteDouble', dimmer, light_int_out);

```

```

illp = light_int_c;
l_plot(i) = light_int_c;

if enabl==1
    set(handl.dimmer,'string',['Light intensity ' num2str(light_int_out) ' %']);
end
%
% window control: act_val(3) 0-100%
% pw: window position (0: fully closed, 100: fully open)
% cl_dur, op_dur: close/open duration
% window controller is activated if action>1 sec.
if act_val(3) < pw
%
    close window
    cl_dur_w = floor(tot_dur_w * (pw - act_val(3))/100);
    [i cl_dur_w ]
    if cl_dur_w > 1
        hr2 = mxOPC('WriteBool', close_window, on);
        pause (cl_dur_w)
        hr2 = mxOPC('WriteBool', close_window, off);
        w_dur = cl_dur_w;
        pw = pw - (cl_dur_w * tot_dur_w)
    else
        w_dur = 0;
    end
end
else
%
% Άνοιγμα παραθύρου
op_dur_w = floor(tot_dur_w * (act_val(3) - pw)/100);
%
% [i op_dur_w]
if op_dur_w > 1
    hr2 = mxOPC('WriteBool', open_window, on);
    pause (op_dur_w)
    hr2 = mxOPC('WriteBool', open_window, off);
    w_dur = op_dur_w;
    pw = pw + (op_dur_w * tot_dur_w)
else
    w_dur = 0;
end
end
if enabl==1
    set(handl.win,'string',['Window opened ' num2str(pw) ' %']);
end
w_plot(i) = pw;

%
% Έλεγχος για τα στόρια : act_val(4) 0-1
% ps: Θέση στοριών (1: fully closed, 0: fully open)
if act_val(4) > ps
%
% Κλείσιμο στοριών
cl_dur_s = floor(tot_dur_s * (act_val(4) - ps));
if cl_dur_s > 1
    hr2 = mxOPC('WriteBool', close_shutter, on);
    pause (cl_dur_s)
    hr2 = mxOPC('WriteBool', close_shutter, off);
    s_dur = cl_dur_s;
    ps = ps + cl_dur_s/tot_dur_s
else
    s_dur = 0;
end
end

```

```

else
%
    Άνοιγμα στοριών
    op_dur_s = floor(tot_dur_s * (ps - act_val(4)))
    if op_dur_s > 1
        hr2 = mxOPC('WriteBool', open_shutter, on);
        pause (op_dur_s)
        hr2 = mxOPC('WriteBool', open_shutter, off);
        s_dur = op_dur_s;
        ps = ps - op_dur_s/tot_dur_s
    else
        s_dur = 0;
    end
end

if enabl==1
    set(handl.shut,'string',['Shutter opened ' num2str(100-(ps*100)) ' %']);
end
s_plot(i) = ps;

%
Κρύο ή ζέστη? (0-100)
if act_val(1) > act_val(2)
%
    Ζέση
    heat_dur = (act_val(1)/100) * (sampling_per - (w_dur + s_dur));
    vent_dur = heat_dur;
if valve_pos == 0
%
Αλλαγή θέσης βαλβίδας AC, Κλείσιμο AC αρχικά, αναμονή
    hr2 = mxOPC('WriteBool', use_AC, off);
    pause (AC_del/2);
    hr2 = mxOPC('WriteBool', use_valve, on);
        pause (AC_del/2);
        hr2 = mxOPC('WriteBool', use_AC, on);
    vent_dur = vent_dur + AC_del;
end
if heat_dur>5
    hr2 = mxOPC('WriteBool', use_AC, on);
    pause (heat_dur)
end
hr2 = mxOPC('WriteBool', use_AC, off);
    v_plot(i) = act_val(1);
valve_pos = 1;
    coh=0;
    else
%
    Κρύο
    cool_dur = (act_val(2)/100) * (sampling_per - (w_dur + s_dur));
    vent_dur = cool_dur;
if valve_pos == 1
%
Αλλαγή θέσης βαλβίδας AC, Κλείσιμο AC αρχικά, αναμονή
    hr2 = mxOPC('WriteBool', use_AC, off);
    pause (AC_del/2);
    hr2 = mxOPC('WriteBool', use_valve, off);
    pause (AC_del/2);
        hr2 = mxOPC('WriteBool', use_AC, on);
    vent_dur = vent_dur + AC_del;
end
if cool_dur>5
    hr2 = mxOPC('WriteBool', use_AC, on);

```

```

        pause (cool_dur)
    end
    hr2 = mxOPC('WriteBool', use_AC, off);
        v_plot(i) = -act_val(2);
    valve_pos = 0;
        coh=1;
    end
    if enabl==1
        if coh==0
            set(handl.coolheat,'string',['Heating...           for
num2str(round(heat_dur)) ' seconds']);
        end
        if coh==1
            set(handl.coolheat,'string',['Cooling...           for
num2str(round(cool_dur)) ' seconds']);
        end
    end
end

%           Δείγμα σε <sampling_per> δευτερόλεπτα
pause (sampling_per - (w_dur + s_dur + vent_dur))

%           Αποθήκευση τιμών αισθητήρων και μηχανισμών κίνησης
A_X(i, :) = [CO2(i) ill(i) Ta(i) Tout(i) humid(i) v(i) act_val(1) act_val(2)
act_val(3) act_val(4) act_val(5) act_val(6)];
A_T(i, :) = [CO2(i) ill(i) Ta(i) Tout(i) humid(i) v(i) s_plot(i) w_plot(i) l_plot(i)
v_plot(i)];

    if enabl==0
        set (handles.Send_mon,'enable','on');
    end
end
CO2_prev1 = CO2(i);
end

```

function Period_edit_Callback(hObject, eventdata, handles)

% hObject handle to Period_edit (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Period_edit as text

% str2double(get(hObject,'String')) returns contents of Period_edit as a double

global secs i

secs=0;

% **Εισαγωγή περιόδου δειγματοληψίας**

secs=str2double(get(hObject,'String'));

% **Για τιμή περιόδου (secs>0)**

if secs>0

% **Ενεργοποίηση των checkboxes για επιλογή αισθητήρα προς παρακολούθηση**

set (handles.Ext_temp_checkbox,'enable','on');

set (handles.Int_temp_checkbox,'enable','on');

set (handles.humidity_checkbox,'enable','on');

set (handles.CO2_checkbox,'enable','on');

set (handles.Airflow_checkbox,'enable','on');

set (handles.Ill_checkbox,'enable','on');

set (handles.Period_edit,'enable','off');

% Καθορισμός χρώματος για το editbox επιλογής ρυθμού δειγματοληψίας σε γκρι (ανενεργό)

```
set(handles.Period_edit,'BackgroundColor',[0.8 0.8 0.8]);
```

% Εμφάνιση μηνύματος στην «ετικέτα» πληροφοριών: Επέλεξε από τα checkboxes ποιους-ον αισθητήρες-α θέλεις να παρακολουθήσεις και πάτα το κουμπί «Εκκίνηση παρακολούθησης».

```
set(handles.State_text,'string','Choose from the checkboxes which sensor(s) you want to monitor and press the "Start Monitoring" button.');
```

end

% --- Executes on button press in Ext_temp_checkbox.

```
function Ext_temp_checkbox_Callback(hObject, eventdata, handles)
```

```
% hObject handle to Ext_temp_checkbox (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of Ext_temp_checkbox
```

```
global a1 a2 a3 a4 a5 a6;
```

% Εισαγωγή τιμής. Ενεργοποιημένο checkbox (a1=1). Ανενεργό (a1=0)

```
a1=get(hObject,'Value');
```

% Αν όλα τα checkboxes είναι ανενεργά τότε απενεργοποιούνται τα κουμπιά Start, Stop, Pause monitoring

```
if (a1+a2+a3+a4+a5+a6)==0
```

```
set(handles.Start_monitor_butt,'enable','off');
```

```
set(handles.Stop_monitor_butt,'enable','off');
```

```
set(handles.Pause_monitor_butt,'enable','off');
```

else

% Αν το checkbox είναι ενεργό τότε ενεργοποιείται το κουμπί Start monitoring.

```
set(handles.Start_monitor_butt,'enable','on');
```

```
set(handles.Stop_monitor_butt,'enable','off');
```

```
set(handles.Pause_monitor_butt,'enable','off');
```

end

% --- Executes on button press in Int_temp_checkbox.

```
function Int_temp_checkbox_Callback(hObject, eventdata, handles)
```

```
% hObject handle to Int_temp_checkbox (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of Int_temp_checkbox
```

```
global a1 a2 a3 a4 a5 a6;
```

```
a2=get(hObject,'Value');
```

```
if a2==1
```

```
set(handles.Start_monitor_butt,'enable','on');
```

```
set(handles.Stop_monitor_butt,'enable','on');
```

```
set(handles.Pause_monitor_butt,'enable','on');
```

end

```
if (a1+a2+a3+a4+a5+a6)==0
```

```
set(handles.Start_monitor_butt,'enable','off');
```

```
set(handles.Stop_monitor_butt,'enable','off');
```

```
set(handles.Pause_monitor_butt,'enable','off');
```

else

```
set(handles.Start_monitor_butt,'enable','on');
```

```
set(handles.Stop_monitor_butt,'enable','off');
```

```
set(handles.Pause_monitor_butt,'enable','off');
```

end


```
% --- Executes on button press in CO2_checkbox.
function CO2_checkbox_Callback(hObject, eventdata, handles)
% hObject    handle to CO2_checkbox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of CO2_checkbox
```

```
global a1 a2 a3 a4 a5 a6;
a3=get(hObject,'Value');
if a3==1
    set(handles.Start_monitor_but,'enable','on');
    set(handles.Stop_monitor_but,'enable','on');
    set(handles.Pause_monitor_but,'enable','on');
```

```
end
if (a1+a2+a3+a4+a5+a6)==0
    set(handles.Start_monitor_but,'enable','off');
    set(handles.Stop_monitor_but,'enable','off');
    set(handles.Pause_monitor_but,'enable','off');
```

```
else
    set(handles.Start_monitor_but,'enable','on');
    set(handles.Stop_monitor_but,'enable','off');
    set(handles.Pause_monitor_but,'enable','off');
```

```
end
```

```
% --- Executes on button press in Airflow_checkbox.
function Airflow_checkbox_Callback(hObject, eventdata, handles)
% hObject    handle to Airflow_checkbox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of Airflow_checkbox
```

```
global a1 a2 a3 a4 a5 a6;
a4=get(hObject,'Value');
if a4==1
    set(handles.Start_monitor_but,'enable','on');
    set(handles.Stop_monitor_but,'enable','on');
    set(handles.Pause_monitor_but,'enable','on');
```

```
end
if (a1+a2+a3+a4+a5+a6)==0
    set(handles.Start_monitor_but,'enable','off');
    set(handles.Stop_monitor_but,'enable','off');
    set(handles.Pause_monitor_but,'enable','off');
```

```
else
    set(handles.Start_monitor_but,'enable','on');
    set(handles.Stop_monitor_but,'enable','off');
    set(handles.Pause_monitor_but,'enable','off');
```

```
end
```

```
% --- Executes on button press in humidity_checkbox.
function humidity_checkbox_Callback(hObject, eventdata, handles)
% hObject    handle to humidity_checkbox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of humidity_checkbox
```

```

global a1 a2 a3 a4 a5 a6;
a5=get(hObject,'Value');
if a5==1
    set(handles.Start_monitor_but,'enable','on');
    set(handles.Stop_monitor_but,'enable','on');
    set(handles.Pause_monitor_but,'enable','on');
end
if (a1+a2+a3+a4+a5+a6)==0
    set(handles.Start_monitor_but,'enable','off');
    set(handles.Stop_monitor_but,'enable','off');
    set(handles.Pause_monitor_but,'enable','off');
else
    set(handles.Start_monitor_but,'enable','on');
    set(handles.Stop_monitor_but,'enable','off');
    set(handles.Pause_monitor_but,'enable','off');
end

```

```

Function Start_monitor_but_Callback(hObject, eventdata, handles)
% hObject    handle to Start_monitor_but (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

% Οι global μεταβλητές που χρησιμοποιούνται

```

global t pause_b x y1 y2 y3 y4 y5 y6 i secs handles fl1 fl2 fl3 fl4 fl5 fl6

```

% Ενεργοποίηση και απενεργοποίηση διαφόρων επιλογών (κουμπιά, checkboxes και καθορισμός χρωμάτων

```

set(handles.Save,'enable','on');
set(handles.popupmenu1,'enable','off');
set(handles.Start_monitor_but,'enable','off');
set(handles.Stop_monitor_but,'enable','on');
set(handles.Pause_monitor_but,'enable','on');
set(handles.Start_monitor_but,'string','Monitoring...');
set(handles.Pause_monitor_but,'enable','on');
set(handles.Ext_temp_checkbox,'enable','off');
set(handles.Int_temp_checkbox,'enable','off');
set(handles.humidity_checkbox,'enable','off');
set(handles.CO2_checkbox,'enable','off');
set(handles.Airflow_checkbox,'enable','off');
set(handles.Ill_checkbox,'enable','off');
set(handles.Period_edit,'enable','off');
set(handles.Period_edit,'BackgroundColor',[0.8 0.8 0.8]);
set(handles.State_text,'foregroundcolor','g','fontweight','normal');
set(handles.State_text,'string','Monitoring...');

```

% Έλεγχος για την κατάσταση του κουμπιού Pause monitoring. Αν η μεταβλητή pause_b έχει μηδενική τιμή (pause button ενεργοποιημένο) τότε διαγράφονται όλα τα δεδομένα των γραφημάτων.

```

if pause_b==0
    x=[];
    y1=[];
    y2=[];
    y3=[];
    y4=[];
    y5=[];
    y6=[];

```

```

pause_b=0;
i=0;
fl1=1;
fl2=1;
fl3=1;
fl4=1;
fl5=1;
fl6=1;
end

```

```

% Αποθήκευση του πίνακα με τις διάφορες τιμές handles στο handleless για ανταλλαγή
δεδομένων μεταξύ των συναρτήσεων
handleless=handles;

```

***Επεξήγηση:** Ο πίνακας μεταβλητών handles είναι ένας πίνακας που δημιουργείται αυτόματα από τη Matlab όταν αυτή μεταφράζει το γραφικό παράθυρο με τα στοιχεία του σε γραπτό κώδικα. Ο πίνακας αυτός περιέχει πληροφορίες για όλα τα στοιχεία και τις ιδιότητες τους π.χ. τιμές, θέση, χρώμα κτλ. Επίσης αυτός είναι Global, δηλαδή γνωστός και προσβάσιμος από όλες τις συναρτήσεις που δημιουργούνται αυτόματα για να εξυπηρετήσουν τα στοιχεία του παραθύρου. Στο συγκεκριμένο πρόγραμμα, χρειάστηκε εκτός από τις συναρτήσεις των στοιχείων να δημιουργηθεί μια συνάρτηση που δεν προέρχεται από την αυτόματη μετάφραση των γραφικών σε κώδικα, αλλά από τον προγραμματιστή και κάνει ένα σύνολο εργασιών που επηρεάζει τελικώς τα στοιχεία του παραθύρου. Αυτή είναι η συνάρτηση “monitor” και ο ρόλος της θα αναλυθεί παρακάτω. Αυτό που αξίζει να σημειωθεί όμως είναι ότι αυτή η ανεξάρτητη συνάρτηση, δεν μπορεί να προσπελάσει τα περιεχόμενα του πίνακα handles, πράγμα που οφείλεται στο σχεδιασμό της γλώσσας προγραμματισμού, ενώ ταυτόχρονα είναι απαραίτητο να το κάνει για να ολοκληρωθούν οι εργασίες της. Για την αντιμετώπιση αυτής της δυσκολίας χρησιμοποιήθηκε μια νέα μεταβλητή ορισμένη ως Global με όνομα “handleless” και χρησιμοποιείται ως ενδιάμεσος (buffer) μεταξύ των αυτόματων και της ανεξάρτητης συνάρτησης. Πιο συγκεκριμένα, όλες οι τιμές και η δομή του πίνακα handles καταχωρείται στον πίνακα handleless πριν η ροή του προγράμματος μεταφερθεί στη συνάρτηση “monitor” όπου παύουν να «φαίνονται» οι τιμές του πίνακα handles. Το τελικό αποτέλεσμα είναι ότι κατά την εκτέλεση της, όλα τα στοιχεία του παραθύρου είναι προσβάσιμα αφού έχουν μεταφερθεί όλες οι απαραίτητες τιμές μέσω μιας κοινώς αποδεκτής Global μεταβλητής.

```

% Ορισμός ενός timer ο οποίος εκτελεί τη συνάρτηση monitor με περίοδο ορισμένη από
το χρήστη στο editbox sampling period (secs) και εκτελείται μέχρι να διακοπεί από το
χρήστη.

```

```

t=timer('TimerFcn',@monitor,'ExecutionMode','fixedSpacing','Period',secs,'TasksToExecute',
inf,'ErrorFcn',@timererror);
% Εκτέλεση του timer t.
start(t);

```

```

% Ανανέωση των handles.
guidata(hObject, handles);

```

Function monitor (hObject, eventdata, handles)

%Οι global μεταβλητές της συνάρτησης. Επειδή η συνάρτηση monitor επικοινωνεί με σχεδόν όλες %τις υπόλοιπες έχει μεγάλο αριθμό μεταβλητών.

```

global t y1 y2 y3 y4 y5 y6 i secs x handleless a1 a2 a3 a4 a5 a6 fl1 fl2 fl3 fl4 fl5 fl6 sl1v1
sl2v1 sl3v1 sl4v1 sl5v1 sl6v1

```

```
handles=handles;
```

```
%Έλεγχος για την ενεργοποίηση των slider.Αν η τιμή στον χ άξονα υπερβεί τις δέκα μετρήσεις τότε % αρχικοποιείται και ενεργοποιείται ο κάθε slider.
```

```
if i>(10*secs)
```

```
    set(handles.slider1,'max',i-(10*secs));  
    set(handles.slider2,'max',i-(10*secs));  
    set(handles.slider3,'max',i-(10*secs));  
    set(handles.slider4,'max',i-(10*secs));  
    set(handles.slider5,'max',i-(10*secs));  
    set(handles.slider6,'max',i-(10*secs));  
    arrow=secs/(i-(10*secs));  
    trough=10*(secs/(i-(10*secs)));  
    set (handles.slider1,'sliderstep',[arrow trough]);  
    set (handles.slider2,'sliderstep',[arrow trough]);  
    set (handles.slider3,'sliderstep',[arrow trough]);  
    set (handles.slider4,'sliderstep',[arrow trough]);  
    set (handles.slider5,'sliderstep',[arrow trough]);  
    set (handles.slider6,'sliderstep',[arrow trough]);
```

```
if a1==1
```

```
    set(handles.slider1,'enable','on');
```

```
end
```

```
if a2==1
```

```
    set(handles.slider2,'enable','on');
```

```
end
```

```
if a3==1
```

```
    set(handles.slider3,'enable','on');
```

```
end
```

```
if a4==1
```

```
    set(handles.slider4,'enable','on');
```

```
end
```

```
if a5==1
```

```
    set(handles.slider5,'enable','on');
```

```
end
```

```
if a6==1
```

```
    set(handles.slider6,'enable','on');
```

```
end
```

```
%Ενημέρωση του χρήστη μέσω του information text για την ενεργοποίηση των slider.
```

```
    set (handles.State_text,'string','Monitoring... You can view previous sensor(s) value(s),  
using the slider beneath each Graph.');
```

```
end
```

```
%Ορισμός ελάχιστης και μέγιστης τιμής slider.
```

```
x=cat(2,x,i);
```

```
Ma=max(x);
```

```
mi=Ma-(secs*10);
```

```
if mi<0
```

```
    mi=0;
```

```
    Ma=10*secs;
```

```
end
```

```
%Αν επιλεγεί ο αισθητήρας εξωτερικής θερμοκρασίας προβάλλονται οι τιμές στο αντίστοιχο γράφημα, κάνοντας τις απαραίτητες τροποποιήσεις στους άξονες εάν χρησιμοποιείται ο slider.
```

```
if a1==1
```

```
    temp_out = 'main I/O.sensors.0/1/0';
```

```
    [temp_ext, h2, tt2, q2] = mxOPC('ReadDouble', temp_out);
```

```

Tout = -10 + (temp_ext/1000)*5;
y1=cat(2,y1,Tout);

    plot(handles.axes1,x,y1,'-yo','LineWidth',2,'MarkerEdgeColor',[0.7 0.7 0.7],
0.7 0.7,'MarkerFaceColor',[0.7 0.7 0.7],'MarkerSize',5);
    set (handles.axes1,'color',[0 0 0],'fontweight','bold');
    if fl1==1
    set (handles.axes1,'xlim',[mi Ma]);
        sl1_max=get (handles.slider1,'max');
        set (handles.slider1,'value',sl1_max);
    else
        v2=sl1v1+(10*secs);
        set (handles.axes1,'xlim',[sl1v1 v2]);
    end
    set (handles.axes1,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
    xlabel(handles.axes1,'Seconds','color','k','fontweight','bold');
    ylabel(handles.axes1,'External Temp. (Celcius)','color','k','fontweight','bold');

end
%Αν επιλεγεί ο αισθητήρας εσωτερικής θερμοκρασίας προβάλλονται οι τιμές στο αντίστοιχο γράφημα, κάνοντας τις απαραίτητες τροποποιήσεις στους άξονες εάν χρησιμοποιείται ο slider.
if a2==1
    temp_in = 'main I/O.sensors.0/1/2';
    [temp_int, h1, tt1, q1] = mxOPC('ReadDouble', temp_in);
    Ta = temp_int;
    y2=cat(2,y2,Ta);

    plot(handles.axes2,x,y2,'-yo','LineWidth',2,'MarkerEdgeColor',[0.7 0.7 0.7],
0.7 0.7,'MarkerFaceColor',[0.7 0.7 0.7],'MarkerSize',5);
    set (handles.axes2,'color',[0 0 0],'fontweight','bold');
    if fl2==1
    set (handles.axes2,'xlim',[mi Ma]);
        sl2_max=get (handles.slider2,'max');
        set (handles.slider2,'value',sl2_max);
    else
        v2=sl2v1+(10*secs);
        set (handles.axes2,'xlim',[sl2v1 v2]);
    end
    set (handles.axes2,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
    xlabel(handles.axes2,'Seconds','color','k','fontweight','bold');
    ylabel(handles.axes2,'Internal Temp. (Celcius)','color','k','fontweight','bold');
end
%Αν επιλεγεί ο αισθητήρας CO2 προβάλλονται οι τιμές στο αντίστοιχο γράφημα, κάνοντας τις απαραίτητες τροποποιήσεις στους άξονες εάν χρησιμοποιείται ο slider.
if a3==1
    CO_2 = 'main I/O.sensors.0/1/3';
    [Co2, hrts, tts, qqs] = mxOPC('ReadDouble', CO_2);
    CO2 = (Co2/1000) * 200;
    y3=cat(2,y3,CO2);

    plot(handles.axes3,x,y3,'-yo','LineWidth',2,'MarkerEdgeColor',[0.7 0.7 0.7],
0.7 0.7,'MarkerFaceColor',[0.7 0.7 0.7],'MarkerSize',5);
    set (handles.axes3,'color',[0 0 0],'fontweight','bold');
    if fl3==1

```



```

set (handles.axes3,'xlim',[mi Ma]);
    sl3_max=get (handles.slider3,'max');
    set (handles.slider3,'value',sl3_max);
else
    v2=sl3v1+(10*secs);
    set (handles.axes3,'xlim',[sl3v1 v2]);
end
set (handles.axes3,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
xlabel(handles.axes3,'Seconds','color','k','fontweight','bold');
ylabel(handles.axes3,'CO2 (ppm)','color','k','fontweight','bold');
end
%Αν επιλεγεί ο αισθητήρας ροής αέρα προβάλλονται οι τιμές στο αντίστοιχο γράφημα,
κάνοντας τις απαραίτητες τροποποιήσεις στους άξονες εάν χρησιμοποιείται ο slider.
if a4==1
    airflow = 'main I/O.sensors.0/1/4';
    [air_fl, h4, tt4, q4] = mxOPC('ReadDouble', airflow);
    v = (air_fl/1000)*1.6;
    y4=cat(2,y4,v);

    plot(handles.axes4,x,y4,'-yo','LineWidth',2,'MarkerEdgeColor',[0.7 0.7 0.7],'MarkerFaceColor',[0.7 0.7 0.7],'MarkerSize',5);
    set (handles.axes4,'color',[0 0 0],'fontweight','bold');
    if fl4==1
        set (handles.axes4,'xlim',[mi Ma]);
            sl4_max=get (handles.slider4,'max');
            set (handles.slider4,'value',sl4_max);
        else
            v2=sl4v1+(10*secs);
            set (handles.axes4,'xlim',[sl4v1 v2]);
        end
        set (handles.axes4,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
        xlabel(handles.axes4,'Seconds','color','k','fontweight','bold');
        ylabel(handles.axes4,'Airflow (m/sec)','color','k','fontweight','bold');
    end
%Αν επιλεγεί ο αισθητήρας υγρασίας προβάλλονται οι τιμές στο αντίστοιχο γράφημα,
κάνοντας τις απαραίτητες τροποποιήσεις στους άξονες εάν χρησιμοποιείται ο slider.
if a5==1
    humidity = 'main I/O.sensors.0/1/9';
    [hum, h7, tt7, qq7] = mxOPC('ReadDouble', humidity);
    humid = hum/100;
    y5=cat(2,y5,humid);

    plot(handles.axes5,x,y5,'-yo','LineWidth',2,'MarkerEdgeColor',[0.7 0.7 0.7],'MarkerFaceColor',[0.7 0.7 0.7],'MarkerSize',5);
    set (handles.axes5,'color',[0 0 0],'fontweight','bold');
    if fl5==1
        set (handles.axes5,'xlim',[mi Ma]);
            sl5_max=get (handles.slider5,'max');
            set (handles.slider5,'value',sl5_max);
        else
            v2=sl5v1+(10*secs);
            set (handles.axes5,'xlim',[sl5v1 v2]);
        end
        set (handles.axes5,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
        xlabel(handles.axes5,'Seconds','color','k','fontweight','bold');
        ylabel(handles.axes5,'Humidity (%)','color','k','fontweight','bold');
    end

```

```

end
%Αν επιλεγεί ο αισθητήρας έντασης φωτισμού προβάλλονται οι τιμές στο αντίστοιχο
γράφημα, κάνοντας τις απαραίτητες τροποποιήσεις στους άξονες εάν χρησιμοποιείται ο
slider.
if a6==1
    light_intensity = 'main I/O.sensors.0/1/5';
    [light_int, hrs, tts, qqs] = mxOPC('ReadDouble', light_intensity);
    ill = (light_int/1000)*200;
    y6=cat(2,y6,ill);

    plot(handles.axes6,x,y6,'-yo','LineWidth',2,'MarkerEdgeColor',[0.7
0.7],'MarkerFaceColor',[0.7 0.7 0.7],'MarkerSize',5);
    set (handles.axes6,'color',[0 0 0],'fontweight','bold');
    if fl6==1
        set (handles.axes6,'xlim',[mi Ma]);
        sl6_max=get (handles.slider6,'max');
        set (handles.slider6,'value',sl6_max);
    else
        v2=sl6v1+(10*secs);
        set (handles.axes6,'xlim',[sl6v1 v2]);
    end
    set (handles.axes6,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
    xlabel(handles.axes6,'Seconds','color','k','fontweight','bold');
    ylabel(handles.axes6,'Illuminance (lux)','color','k','fontweight','bold');
end

```

i=i+secs;

```

function Save_Callback(hObject, eventdata, handles)
% hObject    handle to Save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Οι μεταβλητές που χρησιμοποιούνται για την ανταλλαγή δεδομένων μεταξύ των
συναρτήσεων.
global A_X A_T str_f sft sft1;
% Εάν δεν υπάρχει ο φάκελος "Saves", τον δημιουργεί.
paron=exist('Saves','dir');
if paron==0
    mkdir('\Saves');
end
% Αποθήκευση των δεδομένων σε αρχείο.
fff = [pwd '\Saves\' str_f '_' sft '_' sft1 '.txt'];
eval(['save(fff,"A_X","-ascii")']);
eval(['save(fff,"A_T","-ascii")']);

```

% --- Executes on slider movement.

```

function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Οι global μεταβλητές που χρησιμοποιούνται για την επικοινωνία με τη συνάρτηση
monitor
global secs fl1 sl1v1

```

```

%Αποθήκευση τρέχουσας τιμής slider (θέσης) στη μεταβλητή sl1v1
sl1v1=get(hObject,'value');

```

```

%Αποθήκευση μέγιστης τιμής slider στη μεταβλητή vmax
vmax=get(hObject,'max');
%Έλεγχος για τη θέση που βρίσκεται ο slider
%Αν το sl1v1 είναι ίσο με το vmax τότε ο slider βρίσκεται τέρμα δεξιά
if sl1v1==vmax
    %Ο slider βρίσκεται τέρμα δεξιά. Η σημαία(flag) fl1=1 και το χρώμα του γίνεται γκρι
    fl1=1;
    set(handles.slider1,'backgroundcolor',[0.898 0.898 0.898]);
else
    %Στην περίπτωση που δεν βρίσκεται τέρμα δεξιά (sl1v1 != vmax) το flag fl1=0 και το
    χρώμα %γίνεται κόκκινο
    fl1=0;
    set(handles.slider1,'backgroundcolor','r');
end
v2=sl1v1+(10*secs);
set (handles.axes1,'xlim',[sl1v1 v2]);

% --- Executes on slider movement.
function slider2_Callback(hObject, eventdata, handles)
% hObject handle to slider2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

global secs fl2 sl2v1

sl2v1=get(hObject,'value');
vmax=get(hObject,'max');
if sl2v1==vmax
    fl2=1;
    set(handles.slider2,'backgroundcolor',[0.898 0.898 0.898]);
else
    fl2=0;
    set(handles.slider2,'backgroundcolor','r');
end

v2=sl2v1+(10*secs);

set (handles.axes2,'xlim',[sl2v1 v2]);

% Hints: get(hObject,'Value') returns position of slider
% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider2_CreateFcn(hObject, eventdata, handles)
% hObject handle to slider2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

```

% --- Executes on slider movement.
function slider3_Callback(hObject, eventdata, handles)
% hObject    handle to slider3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global secs fl3 sl3v1

sl3v1=get(hObject,'value');
vmax=get(hObject,'max');
if sl3v1==vmax
    fl3=1;
    set(handles.slider3,'backgroundcolor',[0.898 0.898 0.898]);
else
    fl3=0;
    set(handles.slider3,'backgroundcolor','r');
end

v2=sl3v1+(10*secs);

set (handles.axes3,'xlim',[sl3v1 v2]);

% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider5_Callback(hObject, eventdata, handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global secs fl5 sl5v1

sl5v1=get(hObject,'value');
vmax=get(hObject,'max');
if sl5v1==vmax
    fl5=1;
    set(handles.slider5,'backgroundcolor',[0.898 0.898 0.898]);
else
    fl5=0;

```

```

    set(handles.slider5,'backgroundcolor','r');
end
v2=s15v1+(10*secs);

set (handles.axes5,'xlim',[s15v1 v2]);

% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider6_Callback(hObject, eventdata, handles)
% hObject    handle to slider6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global secs fl6 sl6v1

sl6v1=get(hObject,'value');
vmax=get(hObject,'max');
if sl6v1==vmax
    fl6=1;
    set(handles.slider6,'backgroundcolor',[0.898 0.898 0.898]);
else
    fl6=0;
    set(handles.slider6,'backgroundcolor','r');
end
v2=sl6v1+(10*secs);

set (handles.axes6,'xlim',[sl6v1 v2]);

% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.

```



```

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

% --- Executes on slider movement.

```

function slider4_Callback(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

global secs fl4 sl4v1

```

```

sl4v1=get(hObject,'value');
vmax=get(hObject,'max');
if sl4v1==vmax
    fl4=1;
    set(handles.slider4,'backgroundcolor',[0.898 0.898 0.898]);
else
    fl4=0;
    set(handles.slider4,'backgroundcolor','r');
end
v2=sl4v1+(10*secs);

set (handles.axes4,'xlim',[sl4v1 v2]);

```

```

% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine range of slider

```

% --- Executes during object creation, after setting all properties.

```

function slider4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

% Hint: slider controls usually have a light gray background.

```

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

% Η συνάρτηση του κουμπιού pause monitoring.

% --- Executes on button press in Pause_monitor_butt.

```

function Pause_monitor_butt_Callback(hObject, eventdata, handles)
% hObject    handle to Pause_monitor_butt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

% Οι global μεταβλητές της συνάρτησης.

```

global pause_b t x y1 y2 y3 y4 y5 y6 i secs

```

% Το flag pause_b ενεργοποιείται.

```

pause_b=1;

```

% Σταματάει ο timer (δεν μηδενίζεται).

```

stop(t);

```

% Το κουπί start monitoring μετονομάζεται σε Resume Monitoring και ενεργοποιείται.

```

set (handles.Start_monitor_butt,'string','Resume Monitoring');

```

```

set(handles.Start_monitor_butt,'enable','on');
set(handles.Stop_monitor_butt,'enable','off');
set(handles.Pause_monitor_butt,'enable','off');
%Αλλαγή μηνύματος στο information box.
set(handles.State_text,'string','Paused. Press "Resume Monitoring" button to continue.');
```

%Συνάρτηση του stop monitoring κουμπιού.

```

function Stop_monitor_butt_Callback(hObject, eventdata, handles)
% hObject    handle to Stop_monitor_butt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Οι μεταβλητές που χρησιμοποιούνται για την ανταλλαγή δεδομένων μεταξύ των συναρτήσεων.
global t pause_b fl1 fl2 fl3 fl4 fl5 fl6
%Η παρακάτω εντολή σταματά τον timer και αμέσως μετά τον μηδενίζουμε.
stop(t);
t=0;
%Μηδενίζονται όλες οι τιμές των γραφημάτων, των slider και απενεργοποιούνται οι επιλογές τους.
pause_b=0;
a=[0];
b=[0];
fl1=0;
fl2=0;
fl3=0;
fl4=0;
fl5=0;
fl6=0;
set(handles.Start_monitor_butt,'string','Start Monitoring');
set(handles.Start_monitor_butt,'enable','on');
set(handles.Pause_monitor_butt,'enable','off');
set(handles.Stop_monitor_butt,'enable','off');
set(handles.slider1,'enable','off');
set(handles.slider2,'enable','off');
set(handles.slider3,'enable','off');
set(handles.slider4,'enable','off');
set(handles.slider5,'enable','off');
set(handles.slider6,'enable','off');
set(handles.slider1,'value',0);
set(handles.slider2,'value',0);
set(handles.slider3,'value',0);
set(handles.slider4,'value',0);
set(handles.slider5,'value',0);
set(handles.slider6,'value',0);
%Το χρώμα των slider γίνεται γκρι (απενεργοποιημένοι).
set(handles.slider1,'backgroundcolor',[0.898 0.898 0.898]);
set(handles.slider2,'backgroundcolor',[0.898 0.898 0.898]);
set(handles.slider3,'backgroundcolor',[0.898 0.898 0.898]);
set(handles.slider4,'backgroundcolor',[0.898 0.898 0.898]);
set(handles.slider5,'backgroundcolor',[0.898 0.898 0.898]);
set(handles.slider6,'backgroundcolor',[0.898 0.898 0.898]);
%Ενεργοποιείται το Period_Edit για να εισαχθεί ο νέος χρόνος δειγματοληψίας.
set(handles.Period_edit,'enable','on');
set(handles.Period_edit,'Backgroundcolor',[1 1 1]);
%Προβάλλονται στα γραφήματα κενοί πίνακες για διαγραφή των τιμών στους άξονες.
plot(handles.axes1,a,b);

```

```

plot(handles.axes2,a,b);
plot(handles.axes3,a,b);
plot(handles.axes4,a,b);
plot(handles.axes5,a,b);
plot(handles.axes6,a,b);
%Ορίζεται μαύρο χρώμα στα γραφήματα.
set (handles.axes1,'color',[0 0 0],'fontweight','bold');
set (handles.axes2,'color',[0 0 0],'fontweight','bold');
set (handles.axes3,'color',[0 0 0],'fontweight','bold');
set (handles.axes4,'color',[0 0 0],'fontweight','bold');
set (handles.axes5,'color',[0 0 0],'fontweight','bold');
set (handles.axes6,'color',[0 0 0],'fontweight','bold');
%Ορίζεται πράσινο το grid των axes.
set (handles.axes1,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
set (handles.axes2,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
set (handles.axes3,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
set (handles.axes4,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
set (handles.axes5,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
set (handles.axes6,'XGrid','on','Xcolor',[0 0.5 0],'YGrid','on','Ycolor',[0 0.5 0]);
%Ορισμός χρώματος για τους τίτλους του γραφήματος.
xlabel(handles.axes1,'Time','color','k','fontweight','bold');
xlabel(handles.axes2,'Time','color','k','fontweight','bold');
xlabel(handles.axes3,'Time','color','k','fontweight','bold');
xlabel(handles.axes4,'Time','color','k','fontweight','bold');
xlabel(handles.axes5,'Time','color','k','fontweight','bold');
xlabel(handles.axes6,'Time','color','k','fontweight','bold');
ylabel(handles.axes1,'External Temp. (Celcius)','color','k','fontweight','bold');
ylabel(handles.axes2,'Internal Temp. (Celcius)','color','k','fontweight','bold');
ylabel(handles.axes3,'CO2 (ppm)','color','k','fontweight','bold');
ylabel(handles.axes4,'Airflow (m/sec)','color','k','fontweight','bold');
ylabel(handles.axes5,'Humidity (%)','color','k','fontweight','bold');
ylabel(handles.axes6,'Illuminance (lux)','color','k','fontweight','bold');
%Αλλαγή μηνύματος στο information box.
set (handles.State_text,'string','Change Sampling Period, then choose from the checkboxes
which sensor(s) you want to monitor and press the "Start Monitoring" button.');
```

```

function Exit_butt_Callback(hObject, eventdata, handles)
% hObject handle to Exit_butt (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Κλείνει το παράθυρο Booth.
close;

function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject handle to figure1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%Οι global μεταβλητές της συνάρτησης.
global turnoff quit1 t enabl actfig;

quit1=0;
%Καλείται ένα υπό-παράθυρο όπου επιβεβαιώνεται το αίτημα για τερματισμό.
exit_conf;
%Εάν ο χρήστης επιβεβαιώσει ότι θέλει να τερματίσει συνεχίζεται η διαδικασία
απενεργοποίησης.
if quit1==1;

```

%Εάν ο timer που κάνει την παρακολούθηση είναι ενεργός απενεργοποιηθεί.

if t~=0

stop(t);

end

% Απενεργοποίηση όλων των στοιχείων του παραθύρου.

set(handles.Send_mon,'enable','off');

set(handles.Start_monitor_butt,'enable','off');

set(handles.Stop_monitor_butt,'enable','off');

set(handles.Pause_monitor_butt,'enable','off');

set(handles.Ext_temp_checkbox,'enable','off');

set(handles.Int_temp_checkbox,'enable','off');

set(handles.humidity_checkbox,'enable','off');

set(handles.CO2_checkbox,'enable','off');

set(handles.Airflow_checkbox,'enable','off');

set(handles.Ill_checkbox,'enable','off');

set(handles.Period_edit,'enable','off');

set(handles.Period_edit,'BackgroundColor',[0.8 0.8 0.8]);

% Ενημέρωση του χρήστη για αναμονή ωσότου ολοκληρωθεί η προετοιμασία του θαλάμου.

set(handles.State_text,'string','Reseting booth and exiting. Please wait...');

% actuators

% Διάβασμα των τιμών των μηχανισμών ελέγχου.

open_window = 'main I/O.window-shutter.0/0/1';

close_window = 'main I/O.window-shutter.0/0/2';

open_shutter = 'main I/O.window-shutter.0/0/3';

close_shutter = 'main I/O.window-shutter.0/0/4';

use_AC = 'main I/O.ventilation-heating.0/3/1';

use_valve = 'main I/O.ventilation-heating.0/3/0';

dimmer = 'main I/O.dimmer.0/4/4';

move_ind_shutter_cw = 'main I/O.window-shutter.0/0/5';

move_ind_shutter_acw = 'main I/O.window-shutter.0/0/6';

% sensors

% Διάβασμα των τιμών των αισθητήρων.

temp_out = 'main I/O.sensors.0/1/0';

temp_in = 'main I/O.sensors.0/1/2';

CO_2 = 'main I/O.sensors.0/1/3';

airflow = 'main I/O.sensors.0/1/4';

light_intensity = 'main I/O.sensors.0/1/5';

smoke_intensity = 'main I/O.sensors.0/1/6';

smoke_alarm = 'main I/O.sensors.0/1/7';

smoke_detector_status = 'main I/O.sensors.0/1/8';

humidity = 'main I/O.sensors.0/1/9';

% initialize sensor values

% Μηδενισμός των τιμών των αισθητήρων.

temp_int_prev = 0;

temp_ext_prev = 0;

CO2_prev = 0;

airflow_prev = 0;

light_int_prev = 0;

smoke_int_prev = 0;

hum_prev = 0;

```

on = 1;
off = 0;
fault_perc = 10;
fault_id = 0;
fault_time = 20;
AC_del = 16;
sft = num2str(fault_perc);
sft1 = num2str(fault_id);
pw = 0;
tot_dur_w = 10;
tot_dur_s = 6;
tot_dur_ind_shut = 3;
CO2_warmup = 90;
sampling_per = 120;
% smoke intensity threshold (3% x 15)
smoke_int_thr = 45;
% illumination due to artificial lighting (measured once)
ill_elec = 111;

% reset all actuators to zero

% Μηδενισμός των τιμών των μηχανισμών ελέγχου.
% close window
% Κλείσιμο του παραθύρου.
hr10 = mxOPC('WriteBool', close_window, on);
pause (tot_dur_w + 1);
hr11 = mxOPC('WriteBool', close_window, off);

% close shutter
% Κλείσιμο των στοριών.
hr12 = mxOPC('WriteBool', close_shutter, on);
pause (tot_dur_s + 1);
hr13 = mxOPC('WriteBool', close_shutter, off);

% close indoor shutter to clockwise position
% Κλείσιμο των περσίδων κατά τη φορά των δεικτών του ρολογιού.
hr2 = mxOPC('WriteBool', move_ind_shutter_cw, on);
pause (tot_dur_ind_shut + 1);
hr2 = mxOPC('WriteBool', move_ind_shutter_cw, off);

% turn off light
% Σβήσιμο της λάμπας.
hr14 = mxOPC('WriteDouble', dimmer, off);

% turn off AC
% Απενεργοποίηση του κλιματιστικού.
hr2 = mxOPC('WriteBool', use_AC, off);

% turn off smoke alarm
% Απενεργοποίηση του συναγερμού καπνού.
hr2 = mxOPC('WriteBool', smoke_alarm, off);
alarm_flag = off;

% Αλλαγή της σημαίας που θα διακόψει τον κύκλο λειτουργίας του θαλάμου (ο κύκλος
τρέχει συνεχώς στη συνάρτηση του κουμπιού On).

```

```

    turnoff=1;
% Εάν είναι ενεργό το παράθυρο actuat.fig
    if enabl==1
% Μηδενισμός της σημαίας του Actuat
    enabl=0;
% Κλείσιμο του παραθύρου Act.fig.
    delete(actfig);
    end
% Κλείσιμο του παραθύρου Booth.fig.
    delete(hObject);
end

function Send_mon_Callback(hObject, eventdata, handles)
% hObject    handle to Send_mon (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Οι global μεταβλητές της συνάρτησης.
global pw ps light_int_out coh handl enabl heat_dur cool_dur shadp;
actuat;
% Η μεταβλητή enabl χρησιμοποιείται σαν flag και δηλώνει ότι το περιβάλλον είναι
% ενεργοποιημένο.
enabl=1;

% Ενημέρωση των static texts του gui για τις τιμές ελέγχου.
set(handl.shut,'string',['Shutter opened ' num2str(100-(ps*100)) ' %']);
set(handl.shut,'foregroundcolor',[0 0.8 0]);

set(handl.dimmer,'string',['Light intensity ' num2str(light_int_out) ' %']);
set(handl.dimmer,'foregroundcolor',[0 0.8 0]);

set(handl.win,'string',['Window opened ' num2str(pw) ' %']);
set(handl.win,'foregroundcolor',[0 0.8 0]);

set(handl.blinds,'string',['Venetian Blinds opened ' num2str(100-(shadp*100)) ' %']);
set(handl.blinds,'foregroundcolor',[0 0.8 0]);

if coh==0
    set(handl.coolheat,'string',['Heating... for ' num2str(round(heat_dur)) '
seconds']);
    end
if coh==1
    set(handl.coolheat,'string',['Cooling... for ' num2str(round(cool_dur)) '
seconds']);
    end
set(handl.coolheat,'foregroundcolor',[0 0.8 0]);

```

Μέρος 2^ο (Actuat.m)

```

% Ο κώδικας παρακάτω προέρχεται από τη matlab και χρησιμοποιείται για την
αρχικοποίηση του περιβάλλοντος.
function varargout = actuat(varargin)
% ACTUAT M-file for actuat.fig
% ACTUAT, by itself, creates a new ACTUAT or raises the existing
% singleton*.
% H = ACTUAT returns the handle to a new ACTUAT or the handle to
% the existing singleton*.

```



```

%
% ACTUAT('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in ACTUAT.M with the given input arguments.
%
% ACTUAT('Property','Value',...) creates a new ACTUAT or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before actuat_OpeningFunction gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to actuat_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help actuat

% Last Modified by GUIDE v2.5 13-Dec-2006 11:31:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',      mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @actuat_OpeningFcn, ...
                  'gui_OutputFcn',  @actuat_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before actuat is made visible.
function actuat_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to actuat (see VARARGIN)
% Τέλος κώδικα αρχικοποίησης.

% Choose default command line output for actuat
handles.output = hObject;
% Global μεταβλητές για το περιβάλλον.
global handl enabl actfig;
actfig=gcf;
% Ορισμός του περιβάλλοντος ως ενεργοποιημένο.

```

```

enabl=1;
handl=handles;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes actual wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = actual_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in exit_but.
function exit_but_Callback(hObject, eventdata, handles)
% hObject handle to exit_but (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

close;

```

Μέρος 3^ο (Exit_conf.m)

```

% Αρχή κώδικα αρχικοποίησης περιβάλλοντος .
function varargout = exit_conf(varargin)
% EXIT_CONF M-file for exit_conf.fig
% EXIT_CONF by itself, creates a new EXIT_CONF or raises the
% existing singleton*.
%
% H = EXIT_CONF returns the handle to a new EXIT_CONF or the handle to
% the existing singleton*.
%
% EXIT_CONF('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in EXIT_CONF.M with the given input arguments.
%
% EXIT_CONF('Property','Value',...) creates a new EXIT_CONF or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before exit_conf_OpeningFunction gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to exit_conf_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

```

```

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help exit_conf

% Last Modified by GUIDE v2.5 25-Mar-2002 12:19:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',      mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @exit_conf_OpeningFcn, ...
                  'gui_OutputFcn',  @exit_conf_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before exit_conf is made visible.
function exit_conf_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to exit_conf (see VARARGIN)

% Choose default command line output for exit_conf
handles.output = 'Yes';

% Update handles structure
guidata(hObject, handles);

% Insert custom Title and Text if specified by the user
% Hint: when choosing keywords, be sure they are not easily confused
% with existing figure properties. See the output of set(figure) for
% a list of figure properties.
if(nargin > 3)
    for index = 1:2:(nargin-3),
        if nargin-3==index, break, end
        switch lower(varargin{index})
            case 'title'
                set(hObject, 'Name', varargin{index+1});
            case 'string'
                set(handles.text1, 'String', varargin{index+1});
        end
    end
end

% Determine the position of the dialog - centered on the callback figure

```

```

% if available, else, centered on the screen
FigPos=get(0,'DefaultFigurePosition');
OldUnits = get(hObject, 'Units');
set(hObject, 'Units', 'pixels');
OldPos = get(hObject,'Position');
FigWidth = OldPos(3);
FigHeight = OldPos(4);
if isempty(gcbf)
    ScreenUnits=get(0,'Units');
    set(0,'Units','pixels');
    ScreenSize=get(0,'ScreenSize');
    set(0,'Units',ScreenUnits);

    FigPos(1)=1/2*(ScreenSize(3)-FigWidth);
    FigPos(2)=2/3*(ScreenSize(4)-FigHeight);
else
    GCBFOldUnits = get(gcbf,'Units');
    set(gcbf,'Units','pixels');
    GCBFPos = get(gcbf,'Position');
    set(gcbf,'Units',GCBFOldUnits);
    FigPos(1:2) = [(GCBFPos(1) + GCBFPos(3) / 2) - FigWidth / 2, ...
        (GCBFPos(2) + GCBFPos(4) / 2) - FigHeight / 2];
end
FigPos(3:4)=[FigWidth FigHeight];
set(hObject, 'Position', FigPos);
set(hObject, 'Units', OldUnits);

% Show a question icon from dialogicons.mat - variables questIconData
% and questIconMap
load dialogicons.mat

IconData=questIconData;
questIconMap(256,:) = get(handles.figure1, 'Color');
IconCMap=questIconMap;

Img=image(IconData, 'Parent', handles.axes1);
set(handles.figure1, 'Colormap', IconCMap);

set(handles.axes1, ...
    'Visible', 'off', ...
    'YDir' , 'reverse' , ...
    'XLim' , get(Img,'XData'), ...
    'YLim' , get(Img,'YData') ...
);

% Make the GUI modal
set(handles.figure1,'WindowStyle','modal')

% UIWAIT makes exit_conf wait for user response (see UIRESUME)
uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = exit_conf_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% The figure can be deleted now
delete(handles.figure1);
% Τέλος κώδικα αρχικοποίησης.

% συνάρτηση κουμπιού “yes”
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Μεταβλητή quit1. Χρησιμοποιείται σαν flag για την επιλογή τερματισμού (1:ναι,
0:όχι.)
global quit1;

handles.output = get(hObject,'String');
% Για επιλογή του κουμπιού “yes” το quit1 ενεργοποιείται (τιμή 1) και το περιβάλλον
τερματίζεται.
quit1=1;

% Update handles structure
guidata(hObject, handles);

% Use UIRESUME instead of delete because the OutputFcn needs
% to get the updated handles structure.
uiresume(handles.figure1);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global quit1;
% Για επιλογή του κουμπιού “no” το quit1 γίνεται 0 και το περιβάλλον δεν τερματίζεται.
handles.output = get(hObject,'String');
quit1=0;

% Update handles structure
guidata(hObject, handles);

% Use UIRESUME instead of delete because the OutputFcn needs
% to get the updated handles structure.
uiresume(handles.figure1);

% Κώδικας προερχόμενος από τη matlab για τη λειτουργία των κουμπιών του
περιβάλλοντος.
% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

if isequal(get(handles.figure1, 'waitstatus'), 'waiting')
    % The GUI is still in UIWAIT, us UIRESUME
    uiresume(handles.figure1);
else
    % The GUI is no longer waiting, just close it
    delete(handles.figure1);
end

% --- Executes on key press over figure1 with no controls selected.
function figure1_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Check for "enter" or "escape"
if isequal(get(hObject,'CurrentKey'),'escape')
    % User said no by hitting escape
    handles.output = 'No';

    % Update handles structure
    guidata(hObject, handles);

    uiresume(handles.figure1);
end

if isequal(get(hObject,'CurrentKey'),'return')
    uiresume(handles.figure1);
end

```

Βιβλιογραφία

Άμεση

1. Gerhard Erich Mischitz, 'The History of Human Computer Interaction', <http://www2.icm.edu/cguetl/education/projects/mischitz/Seminar.htm>
2. Eric Steven Raymond & Rob W. Landley, April 18 2004, 'The Art of Unix Usability', <http://www.catb.org/~esr/writings/taouu/html/ch02s02.html>
3. http://en.wikipedia.org/wiki/History_of_the_graphical_user_interface
4. http://www.nordx.com/public/htmen/3_1.htm#a
5. Joseph Rowntree Foundation, 2007, <http://www.jrf.org.uk/housingandcare/smarthomes/devices.asp>
6. 2002, <http://www.lginternetfamily.co.uk>
7. 2001, <http://www.marshall.edu/LEGO/dc/24Oct01/DCHouse24Oct01all.html>
8. <http://www.ceeo.tufts.edu/>

9. Ben Erwin, Martha Cyr & Chris Roger, LEGO ENGINEER AND ROBOLAB: *Teaching Engineering with LabVIEW from Kindergarten to Graduate School*, Medford,USA.

10. <http://www.legoeducation.com/store/default.aspx?pt=23&searchtype=0&sport=3&c=0&t=1&l=0>

11. Σαριδάκης Γεώργιος, Σχεδιασμός, ανάπτυξη και εγκατάσταση επικοινωνιακού κόμβου EIBUS για σύστημα διαχείρισης ενέργειας σε κτίρια, Πτυχιακή Εργασία, Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης Τμήμα Ηλεκτρονικής, Επιβλέπουσα Καθηγήτρια: Κολοκοτσά Διονυσία

Έμμεση

1. Bush Vannevar, 1945, *As We May Think*, Atlantic Monthly 176, 101-8

2. David T. Craig, 1993, *THE APPLIE LISA COMPUTER: A RETROSPECTIVE*, 736 Edgewater Wichita, KS 67230 USA, viewed 2000-12-03, <http://www.dimensional.com/~djoew/lisa/lisa-retro.html>

3. William K. English, Douglas C. Engelbart & Melvyn L. Berman, 1967, *Display-Selection Techniques for Text Manipulation*, IEEE Transactions on Human Factors in Electronics, , Vol. HFE-8, No. 1, pp. 5-15

4. Douglas Endelbart, 1962, *AUGMENTING HUMAN INTELLECT*, AFOSR-3233, Summary Report, viewed 2000-12-18, http://sloan.stanford.edu/mousesite/EngelbartPapers/B5_F18_ConceptFrameworkInd.html

5. Douglas Engelbart, 1994, *Biographical Sketch*, viewed 2000-11-28, <http://sloan.stanford.edu/MouseSite/dce-bio.htm>