

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

**Απομακρυσμένη πρόσβαση
μέσω διαδικτύου σε ηλεκτρονικό κύκλωμα
με την χρήση LabVIEW**

Τμήμα

ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ

Σπουδαστής

ΚΟΝΤΥΛΗΣ ΟΝΕΤΟΡ

Εισηγητής

ΜΑΛΑΜΟΣ ΑΘΑΝΑΣΙΟΣ

ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ

ΠΕΡΙΕΧΟΜΕΝΑ

Κεφάλαιο 1^ο

LabVIEW

1.1. Τι είναι;	3
1.2. Περιβάλλον LabVIEW.....	5

Κεφάλαιο 2^ο

Project

2.1 Περιγραφή του Project	7
2.2 Δημιουργία Εφαρμογής	
2.2.1 Interface.....	8
2.2.2 Κώδικας	12

Κεφάλαιο 3^ο

Ρυθμίσεις Web Server

3.1 Ρυθμίσεις	20
3.2 BROWSER –Απομακρυσμένος έλεγχος	27

Κεφάλαιο 4^ο

Hardware

4.1 Hardware	31
--------------------	----

ΚΕΦΑΛΑΙΟ 1^ο

LabVIEW

1.1 Τι είναι;

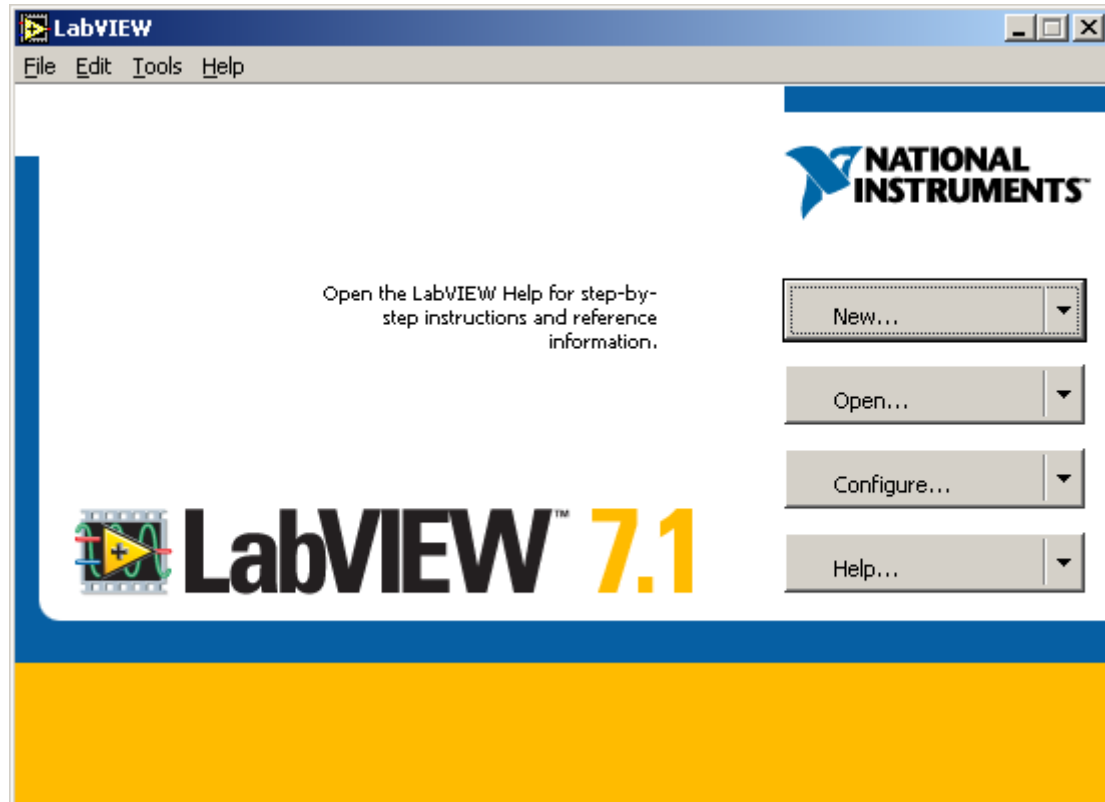
Σκοπός της εργασίας αυτής είναι η δημιουργία μιας εφαρμογής η οποία θα ελέγχει αυτόματα κάποιο Hardware (Υλικό), τοποθετημένο σε ένα συγκεκριμένο χώρο, και ο έλεγχος αυτής μέσω διαδικτύου. Την εφαρμογή αυτή θα την κατασκευάσουμε χρησιμοποιώντας LabVIEW.

Το LabVIEW (Laboratory Virtual Instrument Engineering Workbench) είναι ένα περιβάλλον ανάπτυξης που βασίζεται στον γραφικό προγραμματισμό της γλώσσας G. Το LabVIEW έχει δυνατότητα επικοινωνίας με hardware όπως GPIB, VXI, PXI, RS-232, RS485, καθώς επίσης και με κάρτες συλλογής δεδομένων τεχνολογίας (plug-in). Διαθέτει επίσης και βιβλιοθήκες (Libraries) για χρήση των TCP/IP networking, ActiveX.

Χρησιμοποιώντας το LabVIEW μπορούμε να δημιουργήσουμε 32bit προγράμματα που θα μας προσφέρουν γρήγορες εκτελέσεις οι οποίες είναι απαραίτητες για data acquisition, test, μετρήσεις. Επίσης μπορούμε να δημιουργήσουμε αυτόνομες εκτελέσεις, για μια συγκεκριμένη εφαρμογή του LabVIEW.

Το LabVIEW μπορεί να χρησιμοποιηθεί σε πολλούς τομείς που είναι απαραίτητος ο προγραμματισμός, κατά κύριο λόγο όμως το LabVIEW χρησιμοποιείται προγραμματίζοντας αυτόματα συστήματα ελέγχου. Για την χρήση του LabVIEW χρειάζεται μια μικρή γνώση προγραμματισμού, γιατί χρησιμοποιεί ορολογία, εικονίδια (icons) και ιδέες φιλικές-γνωστές στους τεχνικούς και στους μηχανικούς και βασίζεται πάνω σε γραφικά σύμβολα τα οποία είναι σύμφωνα προς την περιγραφή της ενέργειας που θέλουμε να προγραμματίσουμε. Τα προγράμματα στο LabVIEW μπορεί να είναι από πολύ απλά έως και πάρα πολύ ισχυρά. Το LabVIEW περιέχει βιβλιοθήκες πολύ περιεκτικές για συλλογή δεδομένων, ανάλυση, παρουσίαση και αποθήκευση αυτών.

Περιλαμβάνει επίσης και παραδοσιακά προγράμματα ανάπτυξης. Μπορείτε να τοποθετήσετε σημεία διακοπής (breakpoints), να παρακολουθήσετε με προγράμματα κίνησης την πορεία εκτέλεσης του προγράμματος, βηματική εκτέλεση του προγράμματος για τον εντοπισμό σφαλμάτων για πιο εύκολο προγραμματισμό.

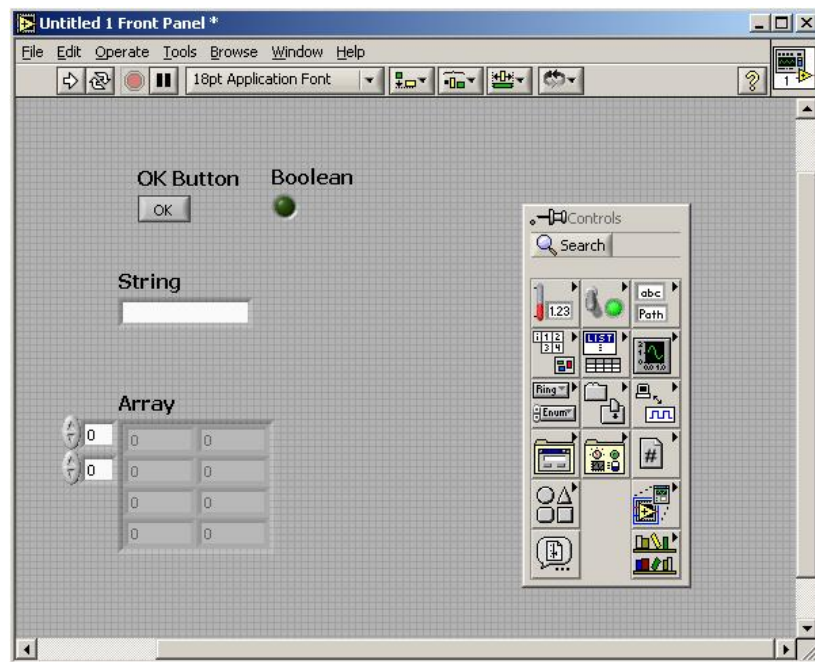


Το παραπάνω παράθυρο είναι το παράθυρο έναρξης του LabVIEW από το οποίο διαλέγουμε με ποιον τρόπο θα χρησιμοποιήσουμε το LabVIEW. Στο συγκεκριμένο Project δούλεψα με το LabVIEW 7

1.2 Περιβάλλον.

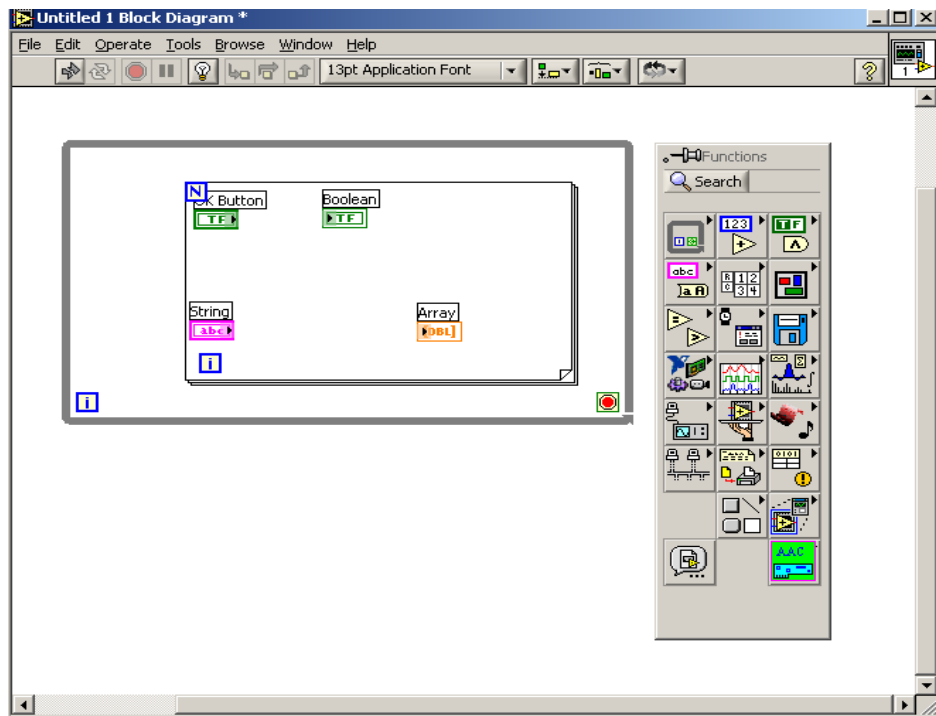
Στο LabVIEW έχουμε 2 χώρους εργασίας

A) To Front Panel



Στο πεδίο αυτό δημιουργούμε το περιβάλλον που θα δουλεύει ο χρήστης με βάση τα εργαλεία που μας δίνει το LabVIEW (Buttons, Textbox, Slides και άλλα Controls).

B) To Block Diagram



Στο πεδίο αυτό δημιουργούμε τον κώδικα ο οποίος θα τρέξει πίσω από το Front Panel.

Το αρχείο που δημιουργούνται από το LabVIEW είναι μορφής .vi. Τα αρχεία αυτά μπορούν να τρέξουν ανεξάρτητα αλλά και να καλούν το ένα το άλλο. Δηλαδή ένα vi μπορεί να είναι μια ξεχωριστή εφαρμογή μπορεί όμως να χρησιμοποιηθεί και ως object σε ένα άλλο vi.

ΚΕΦΑΛΑΙΟ 2^ο

PROJECT

2.1 Περιγραφή του Project

Σκοπός του project είναι η επικοινωνία ενός χρήστη με μια εφαρμογή, και ο έλεγχός της, μέσω WEB. Όπως προείπα θα δουλέψω με το LabVIEW 7.1. Αυτό που θα κάνει το project είναι να λαμβάνει μια θερμοκρασία από ένα ηλεκτρονικό θερμόμετρο και ανάλογα με ένα όριο που εμείς θα έχουμε ορίσει, θα ενεργοποιεί κάποιο ηλεκτρονικό μέσο π.χ. κάποιο κινητήρα ή κάποια σειρήνα κτλ.

Η εφαρμογή αυτή θα περιέχει :

- Ένα γράφημα το οποίο θα δείχνει την θερμοκρασία σε σχέση με το χρόνο
- Μία αναλογική ένδειξη της θερμοκρασίας
- Ένα Textbox το οποίο θα δείχνει την θερμοκρασία με ακρίβεια χιλιοστού
- Ένα ακόμα Textbox το οποίο θα περιέχει το όριο που εμείς θα ορίζουμε
- 2 buttons. Το 1^ο θα εμφανίζει ένα παράθυρο το οποίο θα μας ενημερώνει για της κινήσεις που έχουν γίνει εως εκείνη την στιγμή. Το 2^ο button θα παρουσιάζει ένα παράθυρο με τις ρυθμίσεις της σύνδεσης με την σειριακή θύρα.

2.2 Δημιουργία Εφαρμογής

2.2.1 Interface

Ξεκινώντας την εφαρμογή θα εμφανίζεται ένα παράθυρο το οποίο θα ζητάει από τον χρήστη να ρυθμίσει την σύνδεση με την σειριακή θύρα.

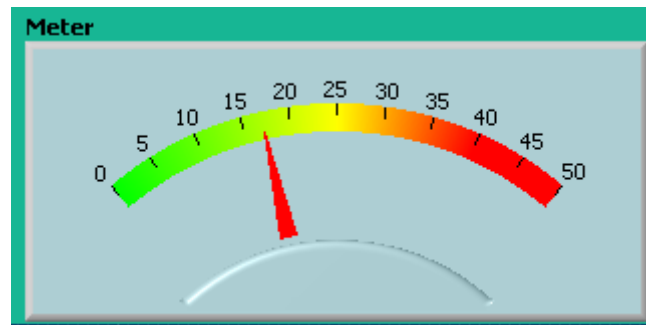
Όπως φαίνεται και από την εικόνα έχει τρία πεδία που πρέπει να συμπληρώσουμε:

1. COM to search: στο πεδίο αυτό επιλέγουμε από μία λίστα τη θύρα του υπολογιστή που θέλουμε να χρησιμοποιήσουμε
2. Time Out Setting: Εδώ ορίζουμε τον χρόνο
3. Baud Rate to Search

Επιλέγοντας τις σωστές ρυθμίσεις στο παράθυρο αυτό, η εφαρμογή φορτώνει το κυρίως παράθυρό της.

Στο παράθυρο λεπτομερειακά έχουμε:

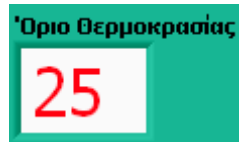
- Αναλογική ένδειξη η οποία μας ενημερώνει για την θερμοκρασία



- Το ίδιο και εδώ με ακρίβεια χιλιοστού



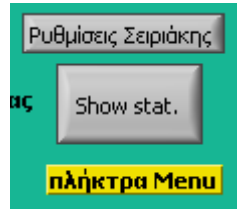
- Εδώ έχουμε το μεταβαλλόμενο, από εμάς, όριο που ορίζει τις περαιτέρω ενέργειες στο project.



- Με το πέρας του παραπάνω ορίου σκοπός του project είναι να σταλεί μία εντολή η οποία θα ενεργοποιήσει το ηλεκτρονικό μέσο. Στο πεδίο αυτό λοιπόν έχουμε μία φωτεινή ένδειξη η οποία μας πληροφορεί εάν η εντολή μας στάλθηκε στο ηλεκτρονικό μέσο. Επίσης έχουμε ένα διακόπτη ο οποίος απενεργοποιεί χειροκίνητα την αποστολή δεδομένων στο μέσο, καθώς επίσης απενεργοποιεί και την σειρήνα.

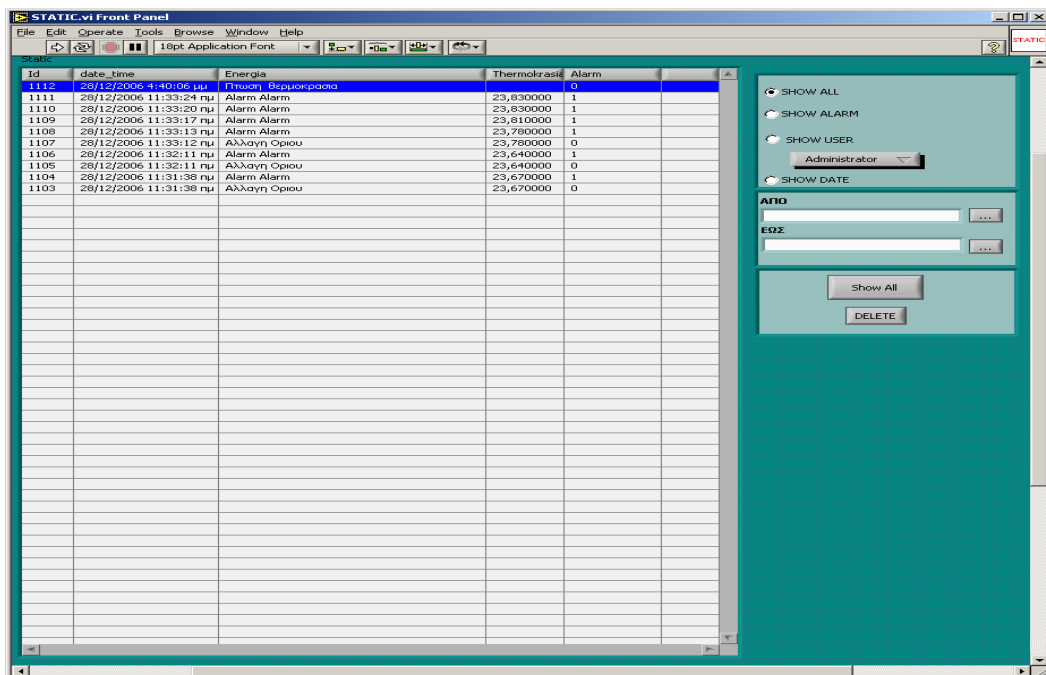


- Τέλος έχουμε 2 κουμπιά. Το πρώτο κουμπί “Ρυθμίσεις Σειριακής” μας επιτρέπει να αλλάξουμε τις ρυθμίσεις στη σύνδεση με τη σειριακή θύρα. Το 2^ο κουμπί “Show stat.” μας εμφανίζει ένα παράθυρο το οποίο περιέχει πληροφορίες για τις όποιες αλλαγές έχουν γίνει.



Αυτό είναι το κύριο .vi της εφαρμογής και το σώζουμε με το όνομα “Main.vi”.

Πατώντας λοιπόν το κουμπί “Show stat.” εμφανίζει το παρακάτω παράθυρο



Σ' αυτό το σημείο η εφαρμογή συνδέεται με μια βάση δεδομένων η οποία συγκερατεί πληροφορίες και τις εμφανίζει στον χρήστη.

Id	date_time	Energia	Thermokrasia	Orio_Thermo	Alarm
1159	8/1/2007 3:39:51 μμ	Πτωση θερμοκρασια	14,600639	20,000000	0
1158	8/1/2007 3:39:51 μμ	Αλλαγη Οριου	14,302449	20,000000	0
1157	8/1/2007 3:39:41 μμ	Πτωση θερμοκρασια	11,661342	15,000000	0
1156	8/1/2007 3:39:23 μμ	Πτωση θερμοκρασια	0,000000	15,000000	0
1155	8/1/2007 3:39:09 μμ	Αλλαγη Οριου		15,000000	0
1154	8/1/2007 3:11:27 μμ	Αυξηση θερμοκρασια	17,742279	25,000000	0

Εδώ αναγράφονται τα στοιχεία: ημερομηνία και ώρα, η ενέργεια που πραγματοποιήθηκε, η θερμοκρασία την χρονική στιγμή που έγινε η αλλαγή, το όριο θερμοκρασίας την στιγμή εκείνη και εάν η συσκευή που ελέγχουμε είναι ενεργή=1 ή όχι=0.

Σ' αυτό το παράθυρο έχουμε και ένα πεδίο με Radio Buttons στο οποίο ανάλογα με την επιλογή μας έχουμε και τις ανάλογες ενδείξεις.

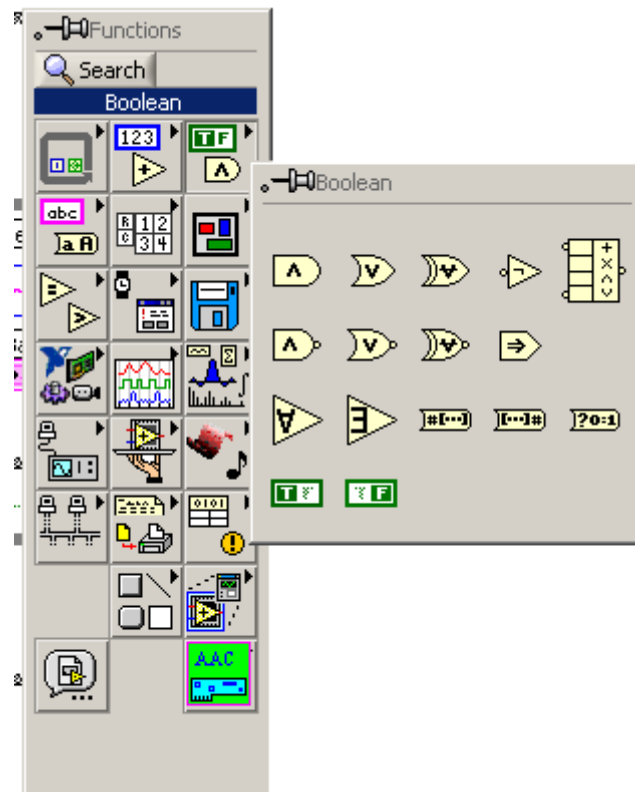
The screenshot shows a control panel with the following elements:

- Three radio buttons for filtering: SHOW ALL, SHOW ALARM, and SHOW DATE.
- Two input fields for date and time selection, labeled "ΑΠΟ" (From) and "ΕΩΣ" (To), each with a dropdown arrow button.
- A "Show All" button.
- A "DELETE" button.

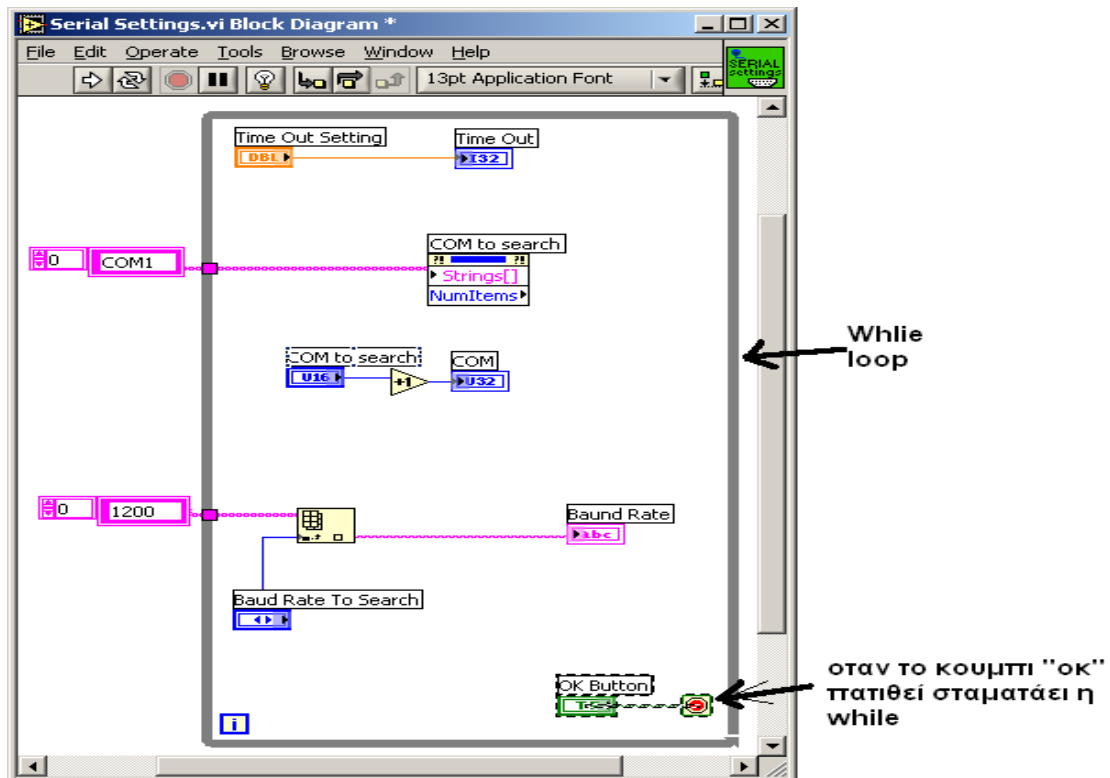
2.2.2 Κώδικας

Όπως προαναφέραμε το LabVIEW είναι ένα περιβάλλον ανάπτυξης που βασίζεται στον γραφικό προγραμματισμό, συνεπώς όταν μιλάμε για κώδικα όπως θα δούμε και πιο κάτω, μιλάμε για ένα γραφικό περιβάλλον, που η κάθε εντολή έχει αντικατασταθεί με ένα συγκεκριμένο σχέδιο.

Ας το δούμε όμως στην πράξη. Στο παράθυρο που ακολουθεί υπάρχει μια παλέτα εργαλείων από την οποία παίρνουμε την κάθε εντολή που χρειαζόμαστε. Στη συνέχεια οι εντολές αυτές συνδέονται μεταξύ τους με γραμμές οι οποίες μεταφέρουν την πληροφορία.



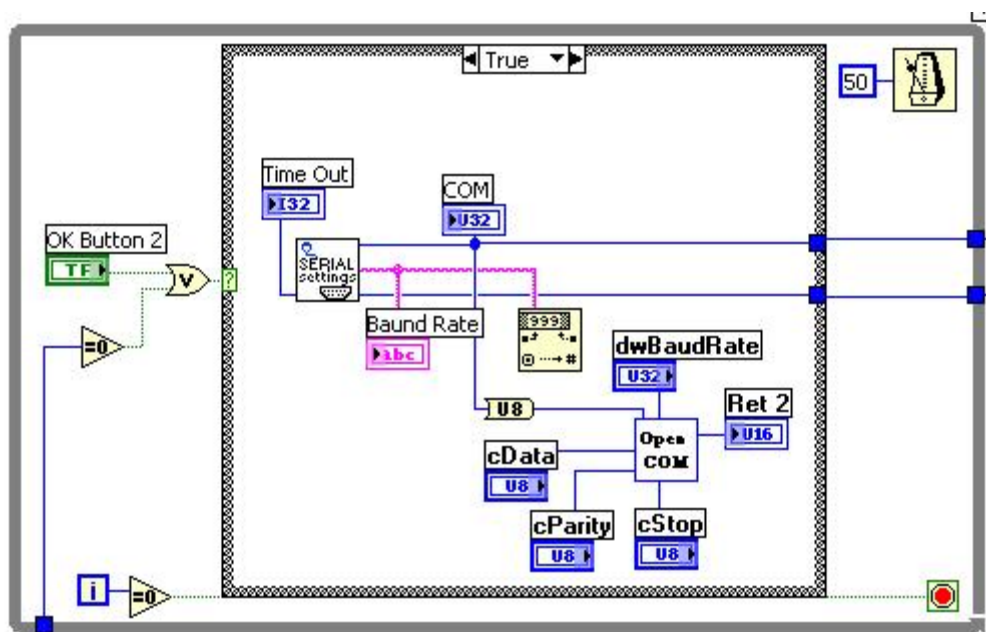
Πρώτα ξεκινάμε με την ρύθμιση της σειριακής πόρτας και δημιουργούμε ένα καινούριο .vi το “Serial Settings.vi”.



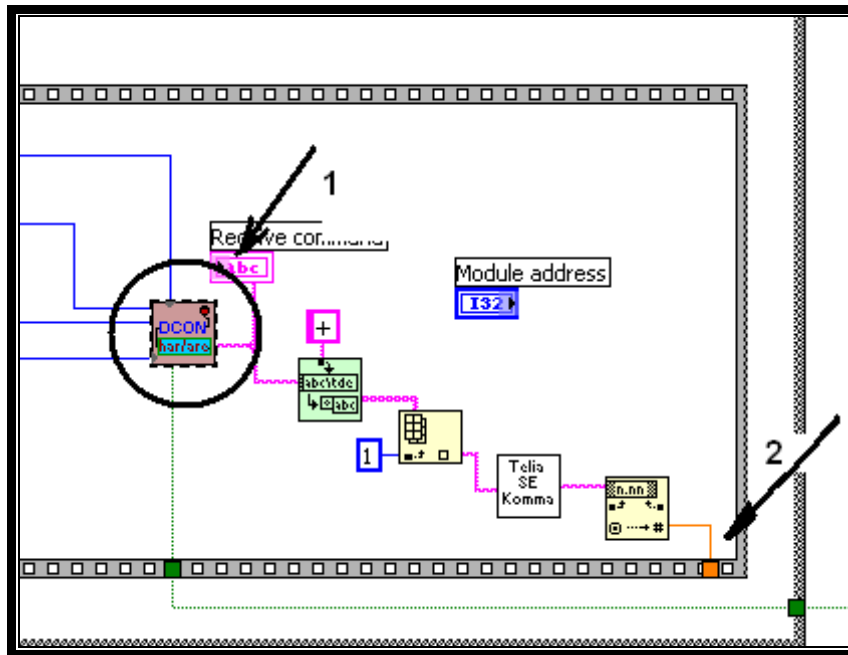
Βάζουμε ένα drop down menu με 15 επιλογές, αριθμός που αλλάζει ανάλογα με το project. Από αυτό το menu επιλέγουμε τη θύρα που θέλουμε. Ορίζουμε ένα Time Out για την σειριακή σύνδεση.

Στο “Main.vi”, στο οποίο καλούμε το παραπάνω “Serial Settings.vi”, ο κώδικας είναι πιο σύνθετος.

Αρχικά φτιάχνουμε μια while loop στην οποία θα περιέχονται όλα τα object της εφαρμογής. Η while αυτή θα σταματάει όταν το πρόγραμμα θα τερματίζεται. Παρακάτω δίνεται μία εξήγηση για μερικά μέρη από τον κώδικα. Θα αρχίσουμε με τον τρόπο που επικοινωνούμε με το hardware και πώς παίρνεται η πληροφορία.



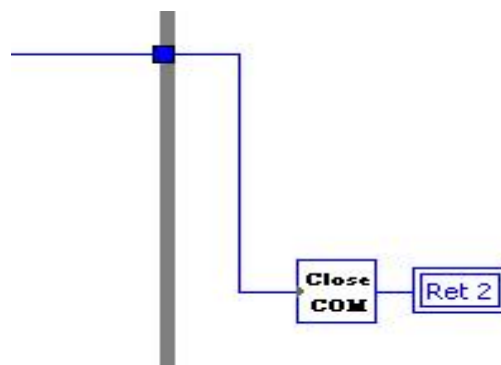
Με τις ρυθμίσεις που πήραμε από το SerialSettings.vi ανοίγουμε μια σύνδεση με την σειριακή θύρα. Αυτό γίνεται μια φορά στο πρώτο κύκλο της while. Στην συνέχεια η σύνδεση παραμένει ανοιχτή για όση ώρα η εφαρμογή μας τρέχει με αυτές τις ρυθμίσεις.



1. HardwareConnection.vi

Εδώ έχουμε δημιουργήσει ένα .vi το οποίο μέσα του περιέχει κάποια νί's που έχουμε βρει έτοιμα από την dcon (εταιρία του hardware, το οποίο δουλεύουμε) και που μέσω της σειριακής θύρας επικοινωνούν με το αντίστοιχο Hardware

2. Στο σημείο αυτό φέρνουμε την πληροφορία, η οποία στην αρχική μορφή της, είναι String, όμως με τα διαφορά objects της γλωσσας τη μετατρέπουμε σε εκμεταλλεύσιμη μορφή float.



Στην έξοδο από την While έχουμε το Close_Conn.vi με το οποίο τερματίζεται η σύνδεση με την σειριακή θύρα.

Η εφαρμογή μας όπως είπαμε παραπάνω κρατάει στοιχεία σε μία βάση

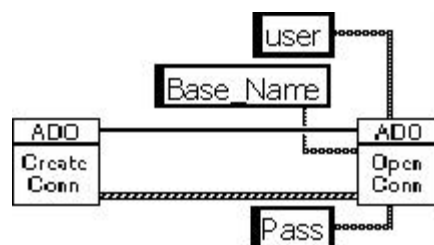
Τα στάδια που ακολουθούνται είναι τα εξής (σχημα):

- **“ADO Create Conn”.**

Όπως λέει και το όνομα του δημιουργεί μια σύνδεση με την βάση και φτιάχνει ένα RefNum, έναν αριθμό δηλαδή ξεχωριστό, ο οποίος αντιπροσωπεύει αυτή την σύνδεση. Δημιουργείται λοιπόν μια σύνδεση την οποία όμως πρέπει να ανοίξουμε δίνοντας της, τις απαραίτητες πληροφορίες.

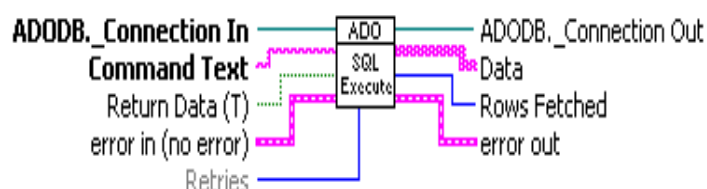
- **“ADO Open Conn”**

“Ανοίγει”, δηλαδή την σύνδεση παίρνοντας για ορίσματα το Όνομα (όνομα χρήστη το οποίο έχουμε δημιουργήσει στον SQL Server), τον Κωδικό, και το Όνομα Βάσης που θέλω να χρησιμοποιήσω.



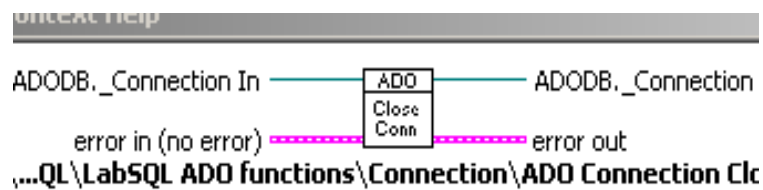
- **“ADO SQL Execute”**

Το object αυτό με τις απαραίτητες εντολές μεταφέρει δεδομένα από και προς τη βάση δεδομένων. Εδώ εισχωρούμε όποια εντολή θέλουμε να “ακούσει” ο SQL Server.



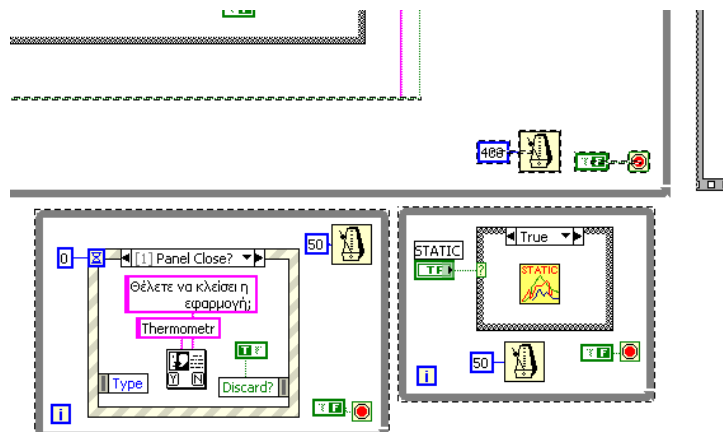
- **“ADO Close Conn”**

Κάθε σύνδεση όμως, πρέπει εφόσον δεν την χρειαζόμαστε πια, να τερματίζεται. Το παρακάτω Object κάνει ακριβώς αυτή την δουλειά.

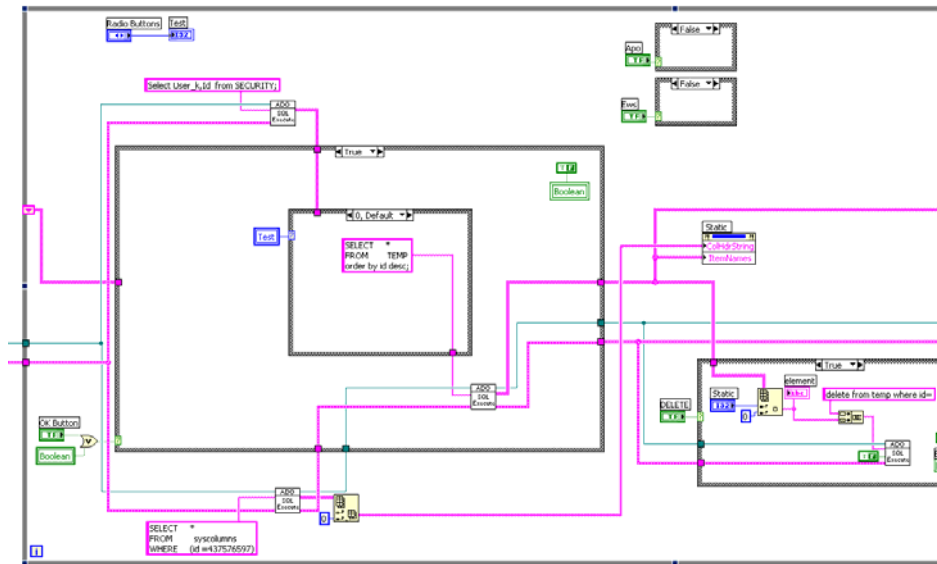


Αυτό που μας ενδιαφέρει σε εφαρμογές αυτού του τύπου είναι να γίνονται πολλές λειτουργίες παράλληλα. Αυτό επιτυγχάνεται όταν οι λειτουργίες αυτές “τρέχουν” ανεξάρτητα. Για να γίνει αυτό πρέπει κάθε λειτουργία να εμπεριέχεται σε μία ξεχωριστή “While” και όλες οι “While loops” να τρέχουν παράλληλα.

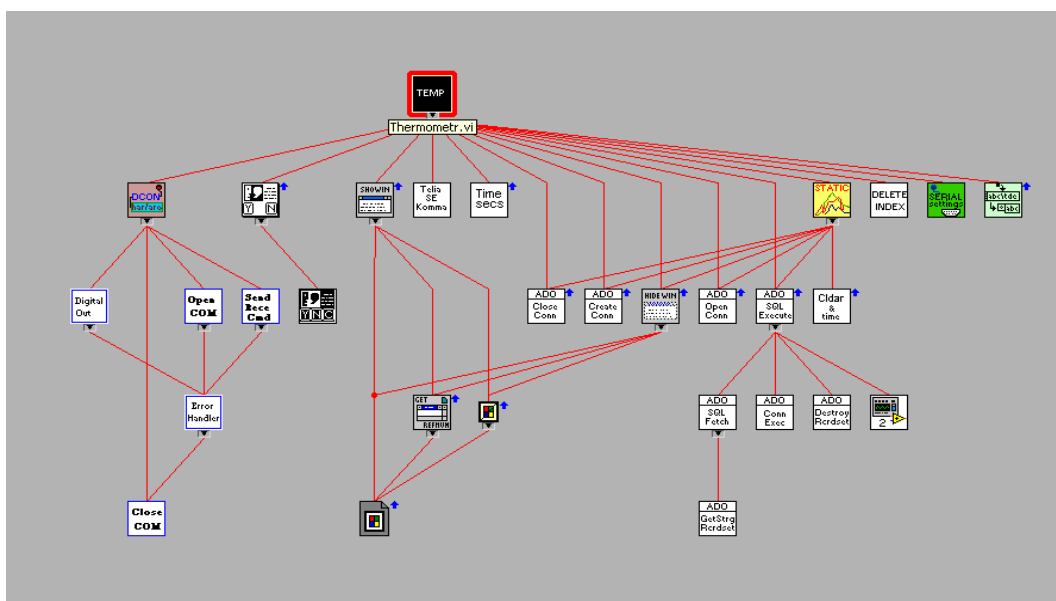
Όπως βλέπουμε και στον κώδικα μας, χρησιμοποιούμε πολλές “While” ανεξάρτητες μεταξύ τους, έτσι ώστε εάν κάποιες από αυτές έχουν λειτουργίες οι οποίες σε άλλες συνθήκες θα διέκοπταν την ροή του προγράμματος, με αυτόν τον τρόπο το αποφεύγουμε. (π.χ. στην περίπτωση που ο χρήστης θα θέλει να ενημερωθεί για τα στατιστικά του προγράμματος, δηλαδή αλλαγές θερμοκρασίας, ώρας κ.τ.λ., να μην σταματάει το πρόγραμμα να καταγράφει την θερμοκρασία.)



Το Static.vi έχει πιο απλό κώδικα. Δημιουργεί μια σύνδεση με τη βάση δεδομένων, την ανοίγει και αυτό που κάνει είναι να αποθηκεύει, να σβήνει ή να ανανεώνει την βάση στην οποία έχει συνδεθεί.



Στην παρακάτω εικόνα μπορούμε να δούμε και τον τρόπο με τον οποίον το κάθε vi συνδέεται με τα υπόλοιπα.



Αυτή λοιπόν ήταν μια μικρή περιγραφή του κώδικα.

ΚΕΦΑΛΑΙΟ 3^ο

WEB SERVER

3.1 Ρυθμίσεις

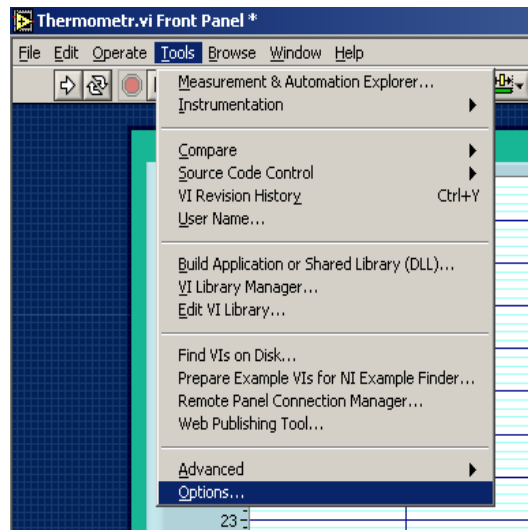
Εφ' όσον δημιουργήσαμε την εφαρμογή με την οποία θα μετράμε την θερμοκρασία μένει να κάνουμε τις σωστές ρυθμίσεις έτσι ώστε στην εφαρμογή αυτή να μπορούμε να έχουμε πρόσβαση μέσω διαδικτύου.

Το LabVIEW είναι ένα πρόγραμμα το οποίο σου επιτρέπει να φτιάξεις τέτοιου είδους εφαρμογές με τις κατάλληλες ρυθμίσεις.

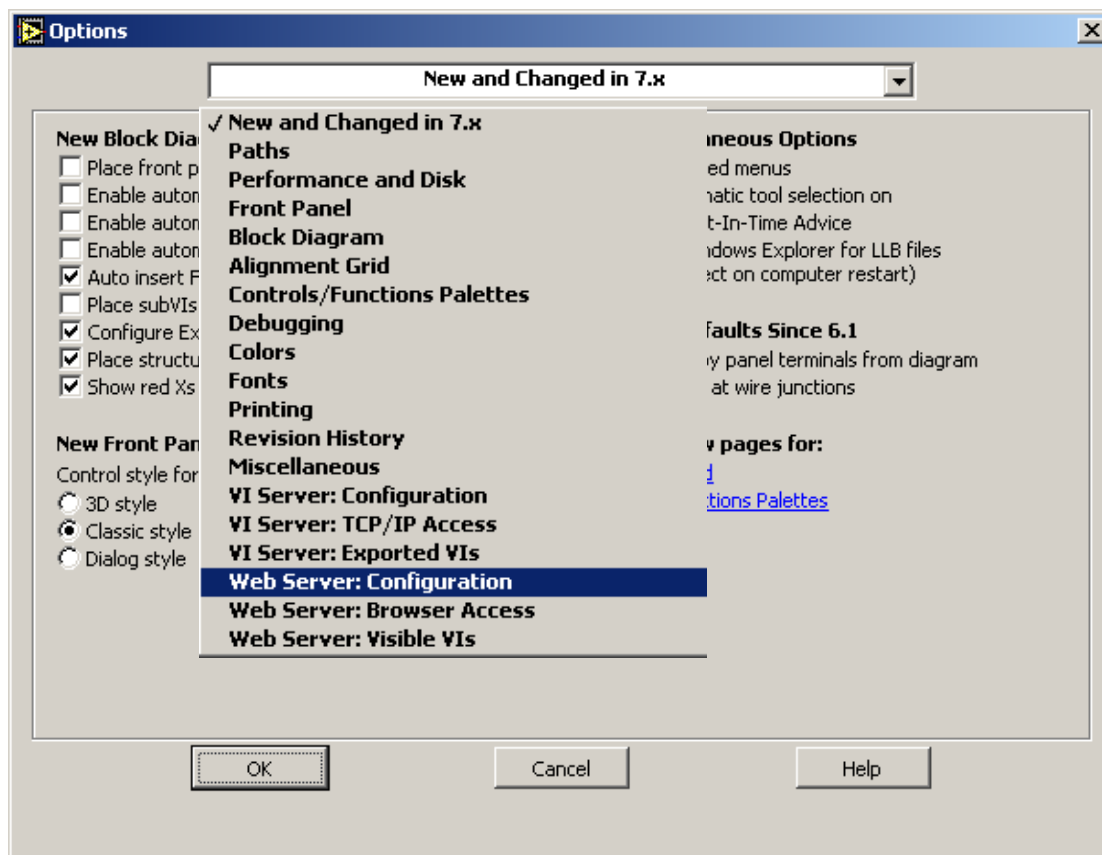
Για να επιτύχουμε τον απομακρυσμένο έλεγχο πρέπει να ρυθμίσουμε τον Web Server του LabVIEW

Ακολουθούμε την εξής διαδικασία:

1. Tools → Options



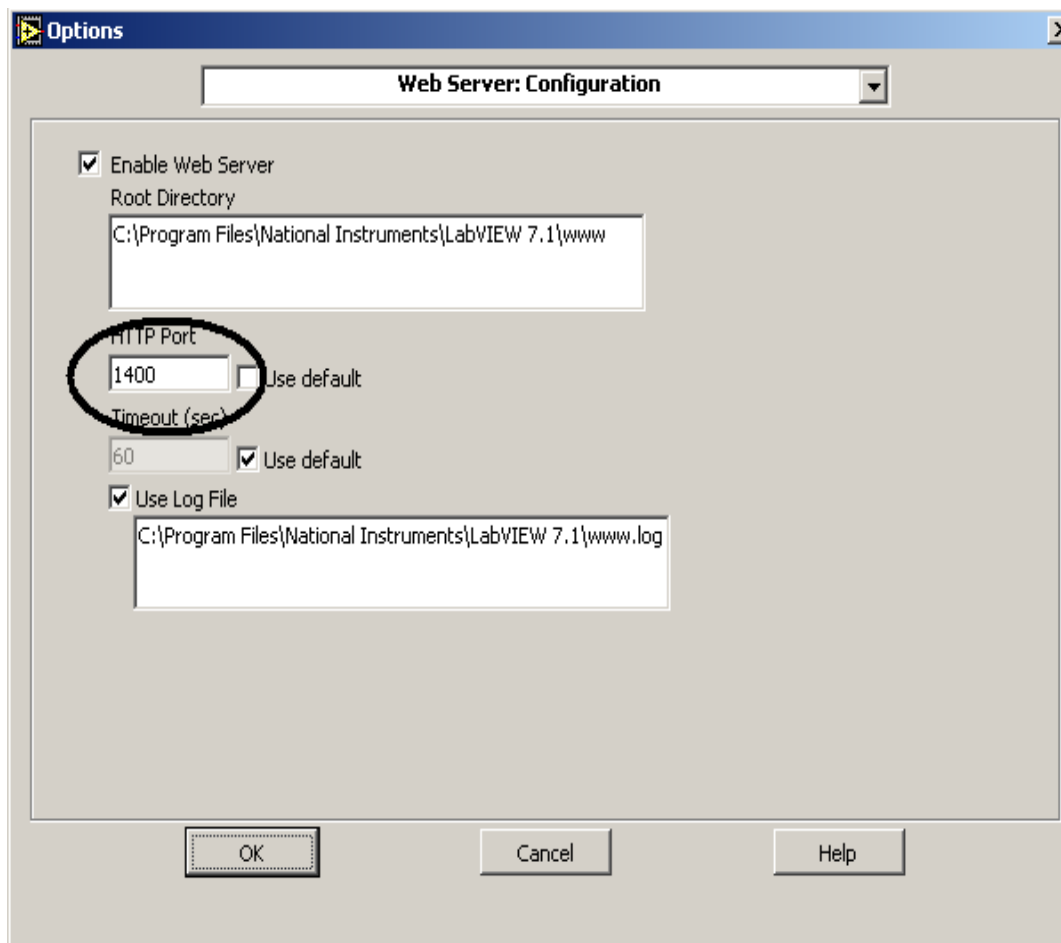
Σ' αυτή την επιλογή εμφανίζεται το παρακάτω παράθυρο



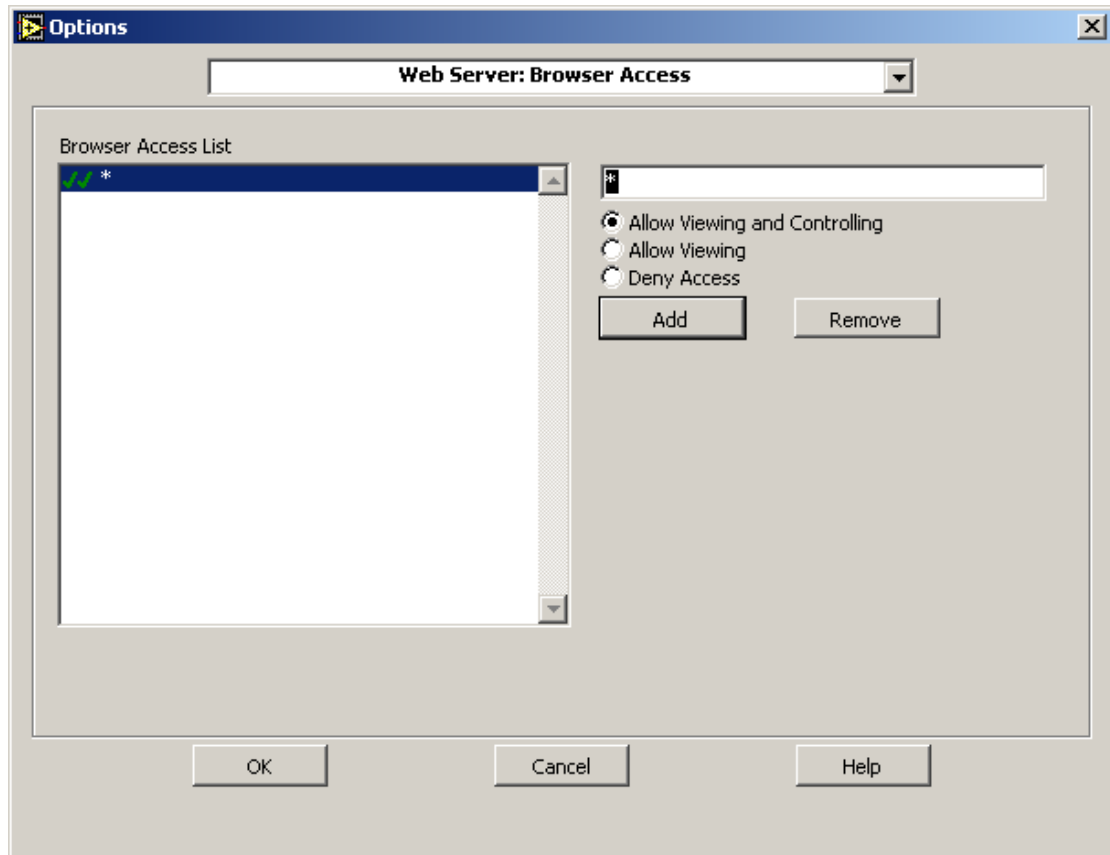
Στην λίστα που εμφανίζεται πατώντας το Drag&Drop, στο πάνω μέρος του παραθύρου, επιλέγουμε **“Web Server:Configuration”**

Στη καρτέλα που εμφανίζεται τα πεδία **“Root Directory”** και **“Use Log File”** τα αφήνουμε ως έχουν .

Τσεκάρουμε το **“Enable Web Server”**, ρυθμίζουμε την θύρα (HTTP Port) από την οποία θέλουμε να γίνει η σύνδεση και το πεδίο **“Timeout”** και πατάμε το **“OK”**.

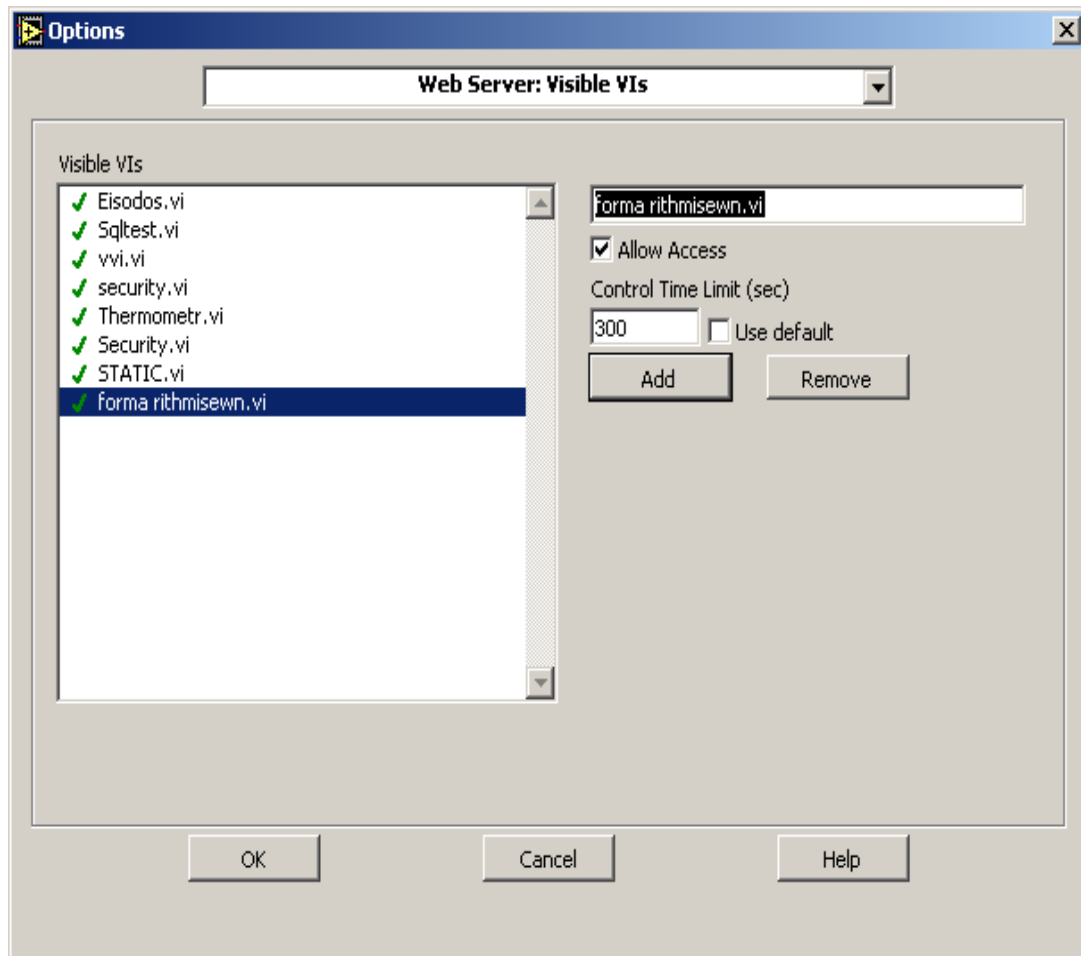


Στην συνέχεια από την λίστα του Drag and Drop επιλέγουμε **“Web Server : Browser Access”**



Στο παράθυρο που εμφανίζεται ελέγχουμε εάν έχουμε τσεκάρει την επιλογή **“Allow Viewing and Controlling”** και πατάμε μετά το **“OK”**.

Αμέσως μετά από την λίστα Drag and Drop επιλέγουμε **“Web Server : Visible VIs”**

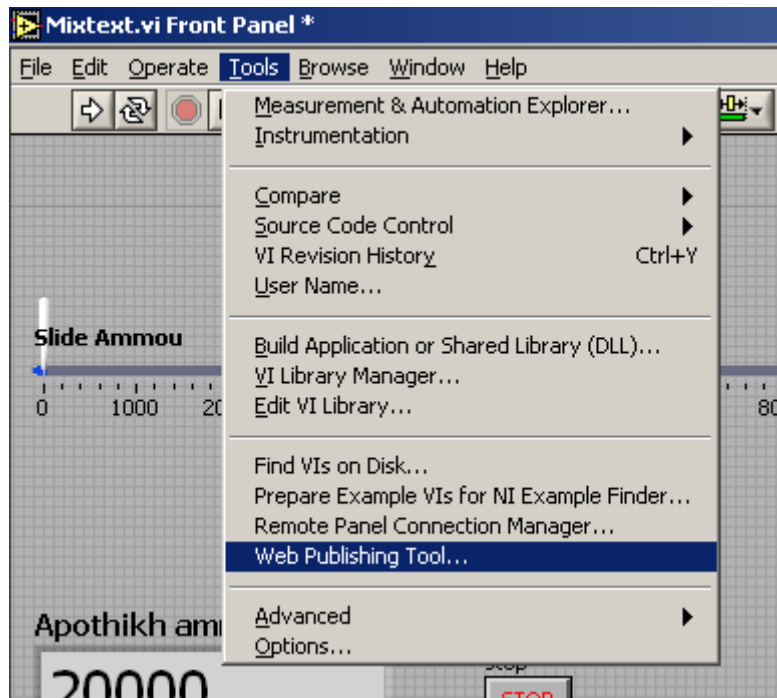


Εδώ τσεκάρουμε την επιλογή **“Allow Access”** . Δηλώνουμε ένα **“Control Time Limit”** και εφ’ όσον κάνουμε τα παραπάνω πατάμε **“Add”** και γράφουμε το όνομα του .vi το οποίο θα φαίνεται και με το οποίο θα δουλέψουμε.

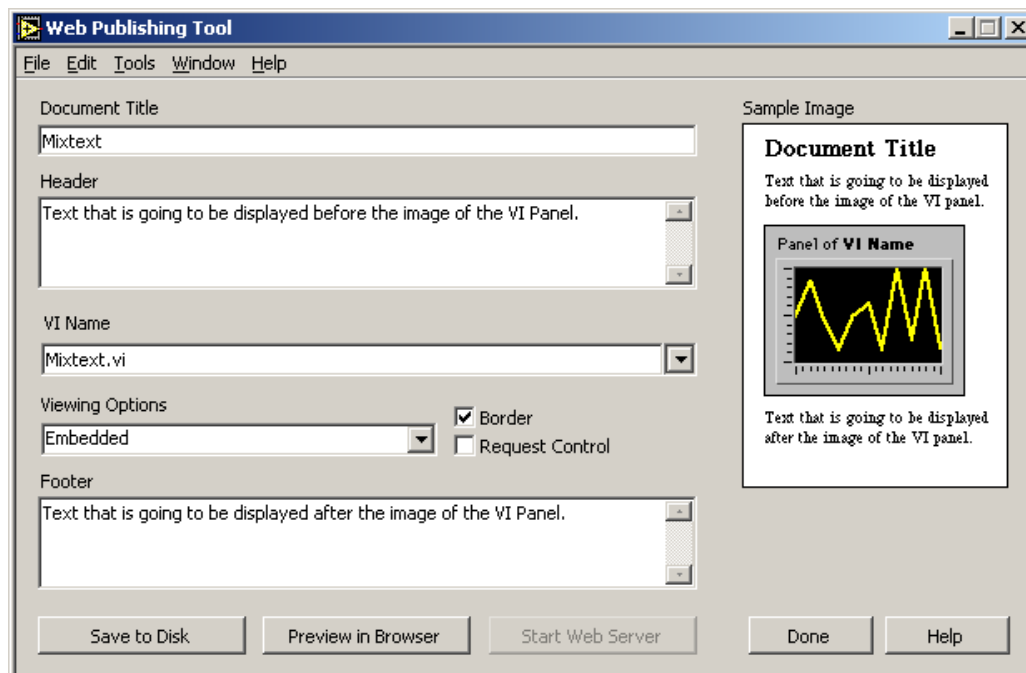
Μετά από τις ρυθμίσεις του Web Server αυτό που μας μένει είναι η δημιουργία ενός HTML αρχείου το οποίο θα καλεί ο Client για να έχει πρόσβαση στην εφαρμογή.

Για την δημιουργία αυτού του αρχείου ακολουθούμε την εξής διαδικασία:

Tools → Web Publishing Tool...



Στην επιλογή αυτή θα εμφανιστεί το παρακάτω παράθυρο.

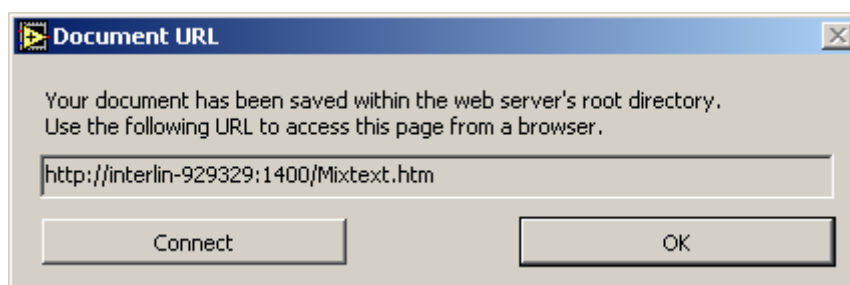


Στο παράθυρο αυτό έχουμε:

- **Document Title:** είναι ο τίτλος του εγγράφου
- **Header:** Είναι το κομμάτι του κειμένου που θα εμφανίζεται πάνω από το VI...
- **VI Name:** Εδώ γράφουμε το όνομα του VI του οποίου θα δημιουργήσουμε το HTML
- **Viewing Options:** Επιλέγουμε τον τρόπο εμφάνισης του VI στο browser...
- **Footer:** Το κομμάτι του κειμένου που θα φαίνεται κάτω από το VI.

Πριν τελειώσουμε τις ρυθμίσεις δεν πρέπει να ξεχάσουμε **να τσεκάρουμε την επιλογή “Request Control”** εάν θέλουμε στην εφαρμογή εκτός από την παρακολούθησή της και τον έλεγχό της.

Τελειώνοντας τις ρυθμίσεις αυτές πατάμε το **“Save to Disk”** έτσι ώστε να αποθηκεύσουμε το αρχείο στο δίσκο. Μόλις το αρχείο σωθεί εμφανίζεται η παρακάτω οθόνη



Στην οθόνη αυτή πληροφορούμαστε από το σύστημα για τον τρόπο που θα επικοινωνήσουμε με την εφαρμογή μέσα από το διαδίκτυο χρησιμοποιώντας έναν browser. Όταν το PC στο οποίο είναι φορτωμένη η εφαρμογή βρίσκεται σε τοπικό δίκτυο στο οποίο το ίντερνετ διαμοιράζεται μέσω ενός modem/Router, τότε το όνομα (URL) που μας δίνει το LabVIEW είναι το local ip του Η/Υ, η θύρα και το όνομα του αρχείου που δημιουργήσαμε. Για να συνδεόμαστε με ένα Η/Υ, που είναι συνδεδεμένοι σε τοπικό δίκτυο LAN, πρέπει να κάνουμε forward την θύρα από την οποία θέλουμε να επικοινωνήσουμε στο modem/router.

Τον τρόπο με τον οποίο κάνουμε forward μια θύρα θα τον δούμε στο Manual του Router μιας και ο τρόπος διαφέρει ανάλογα με το μοντέλο.

3.2 BROWSER –Απομακρυσμένος έλεγχος

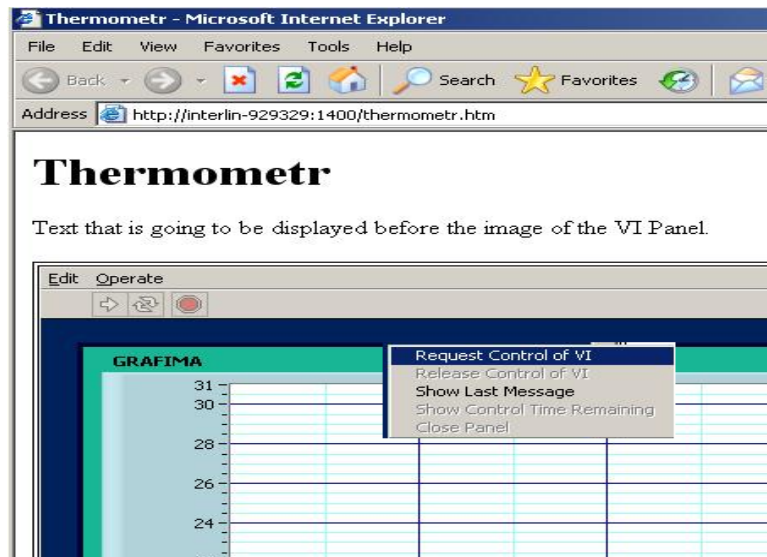
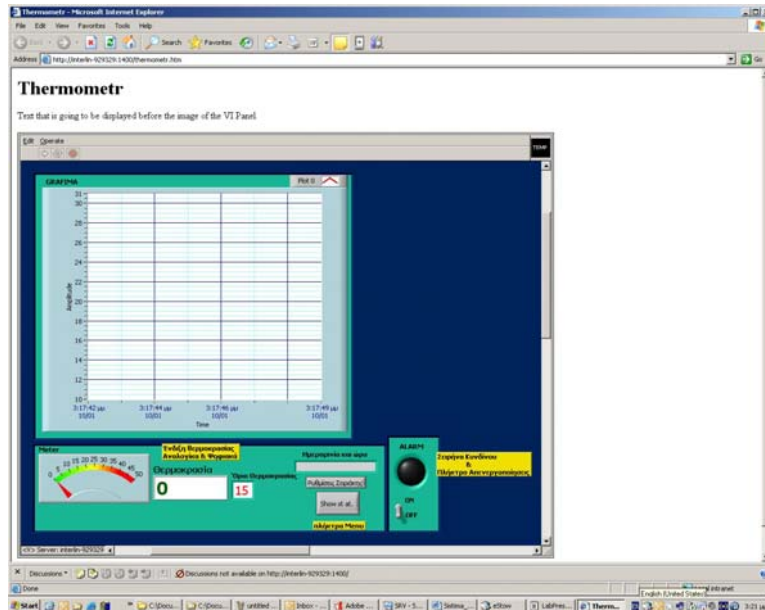
Ο απομακρυσμένος έλεγχος της εφαρμογής, ο οποίος είναι και ο κύριος στόχος αυτού του project, επιτυγχάνεται με την χρήση ενός απομακρυσμένου PC, ο οποίος βέβαια, πρέπει να είναι συνδεδεμένος στο διαδίκτυο και να χρησιμοποιεί έναν οποιοδήποτε browser.

Ανοίγοντας λοιπόν έναν browser πληκτρολογούμε την διεύθυνση που μας δίνεται απο το LabVIEW όταν δημιουργήσαμε το HTML αρχείο και μπορούμε να παρακολουθήσουμε την εφαρμογή αλλά και να την ελέγξουμε.



Εφ' όσον πρώτα ο browser εγκαταστήσει στο σύστημα το απαραίτητο λογισμικό που χρειάζεται ένας υπολογιστής για να τρέχει τις εφαρμογές του LabVIEW, φορτώνει την εφαρμογή μας.

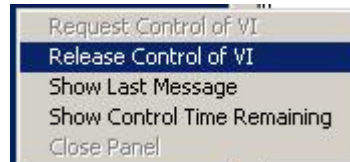
Όπως φαίνεται και στο παράθυρο η εφαρμογή μας είναι φορτωμένη σε έναν browser, και συγκεκριμένα στον explorer.



Με δεξί κλικ πάνω στην εφαρμογή παρουσιάζεται ένα παράθυρο το οποίο έχει την επιλογή “Request Control Of Vi” διαλέγοντας την οποία, αποκτούμε τον έλεγχο του VI.

Μην ξεχνάμε ότι για να υπάρχει αυτή η επιλογή που μόλις εξηγήσαμε πρέπει να έχουμε τσεκάρει την επιλογή **“Request Control”** στις ρυθμίσεις. Όταν ο έλεγχος της εφαρμογής είναι ήδη σ’ αυτό το PC τότε το δεξί κλικ εμφανίζει μια άλλη επιλογή.

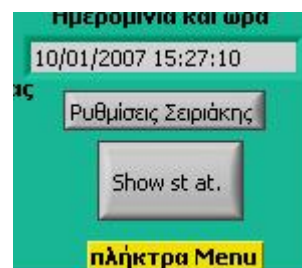
Η επιλογή **“Release Control of VI”** απελευθερώνει τον έλεγχο του VI από το PC αυτό.



Όταν λέμε ότι ένα PC έχει τον έλεγχο της εφαρμογής, σημαίνει ότι ελέγχει εξ’ ολοκλήρου την εφαρμογή



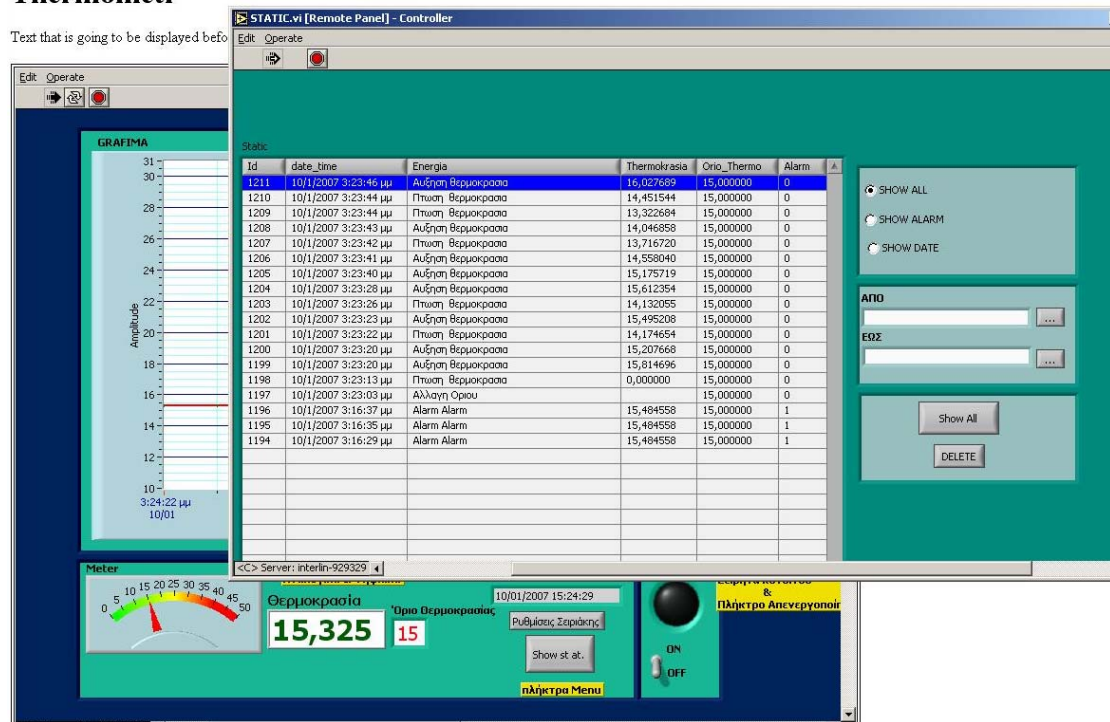
Πατώντας το **“Show stat”**



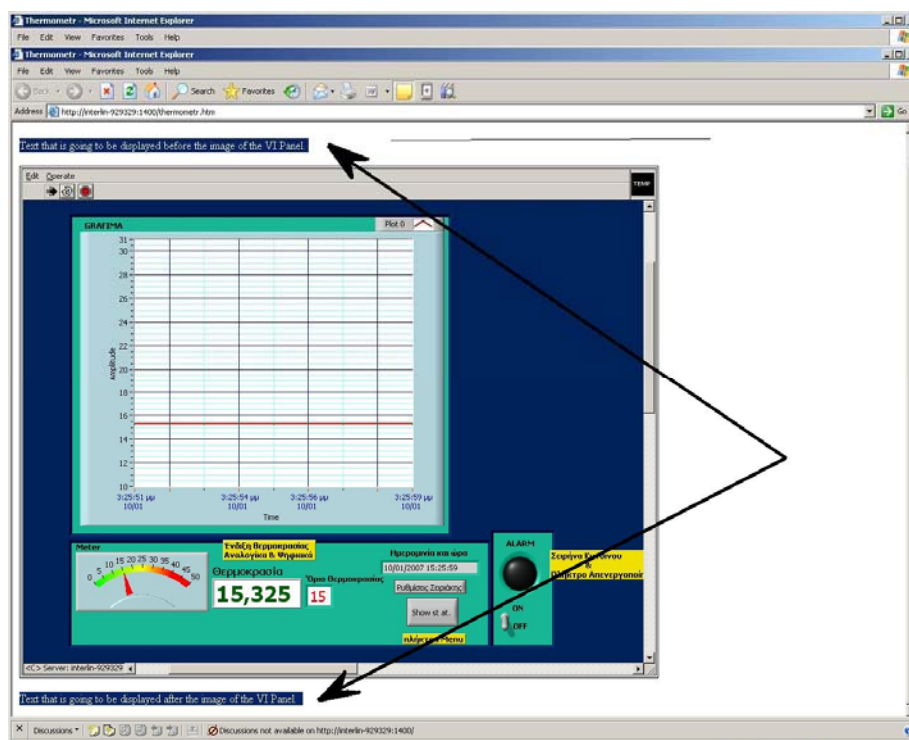
Εμφανίζεται η παρακάτω καρτέλα

Thermometr

Text that is going to be displayed before



Στο παρακάτω παράθυρο απεικονίζονται τα πεδία “Header” και “Footer” για τα οποία είχαμε μιλήσει στη παράγραφο 3.1 (Ρυθμίσεις Web Server) σελ.20



ΚΕΦΑΛΑΙΟ 4^ο

HARDWARE

Πιστεύω πως μετά από αυτή την λεπτομερειακή ανάλυση στο λογισμικό κομμάτι μπορούμε να δώσουμε μια εικόνα του hardware που χρησιμοποιήσα για την επίτευξη του project.

Το υλικό που χρησιμοποιήσα είναι της εταιρείας IspDas (www.ispdas.com).

Ξεκινάω με το module I-7060 το οποίο έχει ως χαρακτηριστικό ότι είναι D/O(digital output), δηλαδή στέλνει την εντολή από το PC στο μέσο που θέλουμε να ελέγχουμε.

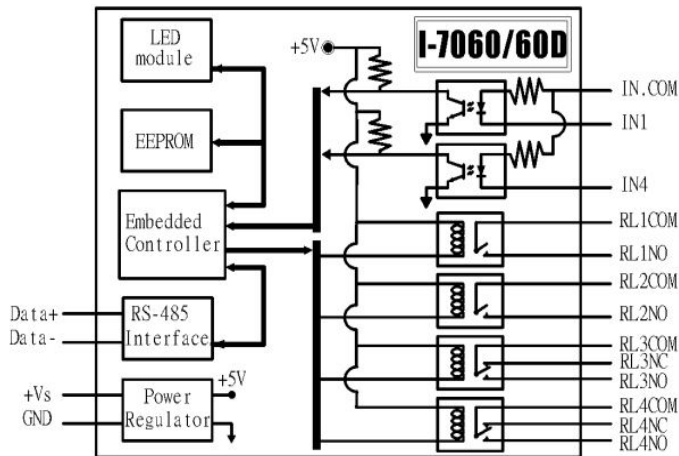
Ο λόγος που χρησιμοποιήσα το υλικό αυτής της εταιρείας είναι το χαμηλό κόστος αλλά και η πολύ εύκολη κατανόηση του τρόπου λειτουργίας του.



Στην εικόνα βλέπουμε το Module I-7060 και θα δείξουμε με ποιο τρόπο το module αυτό επικοινωνεί και συνδέεται με το υπόλοιπο Υλικό.



Εδώ βλέπουμε τις εισόδους και εξόδους το I-7060

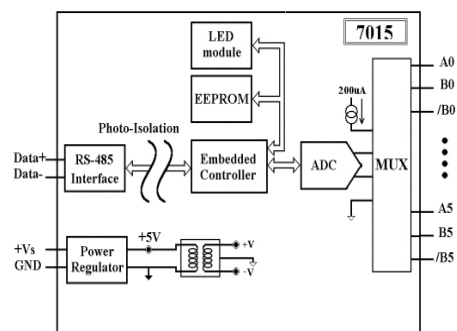
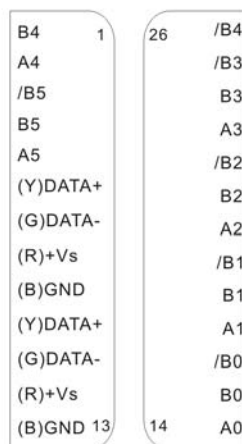


Και εδώ βλέπουμε τον εσωτερικό τρόπο κατασκευής του module.



Το επόμενο module είναι το I-7015. Το χαρακτηριστικό του module αυτού είναι ότι δέχεται πληροφορία (Analog Input A/I)

Στις παρακάτω εικόνες βλέπουμε τις εισόδους και εξόδους του I-7015 και το εσωτερικό.

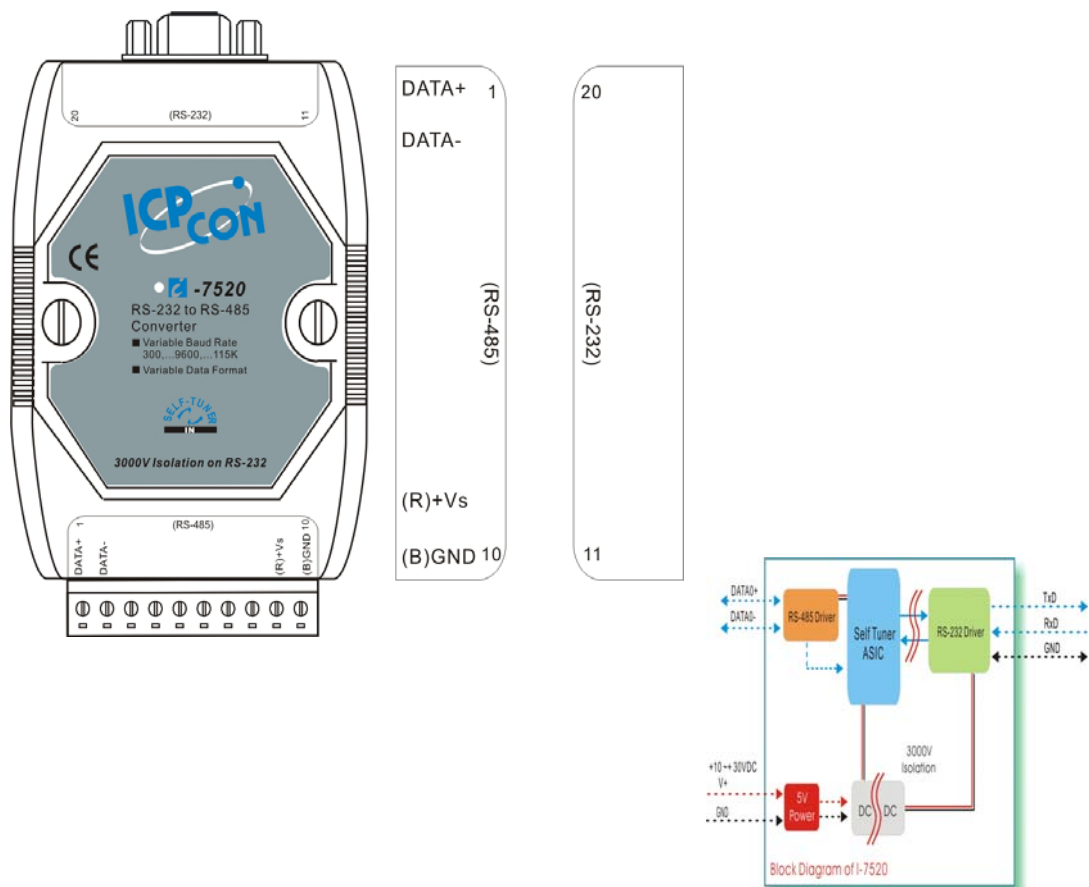


Χρησιμοποιούμε ένα ακόμα module το I-7520R το οποίο κάνει την μετατροπή από RS232 σε RS485. Συνεπώς χρησιμοποιούμε το module αυτό για να μπορούμε να ελέγχουμε το Hardware μέσα από την σειριακή θύρα.

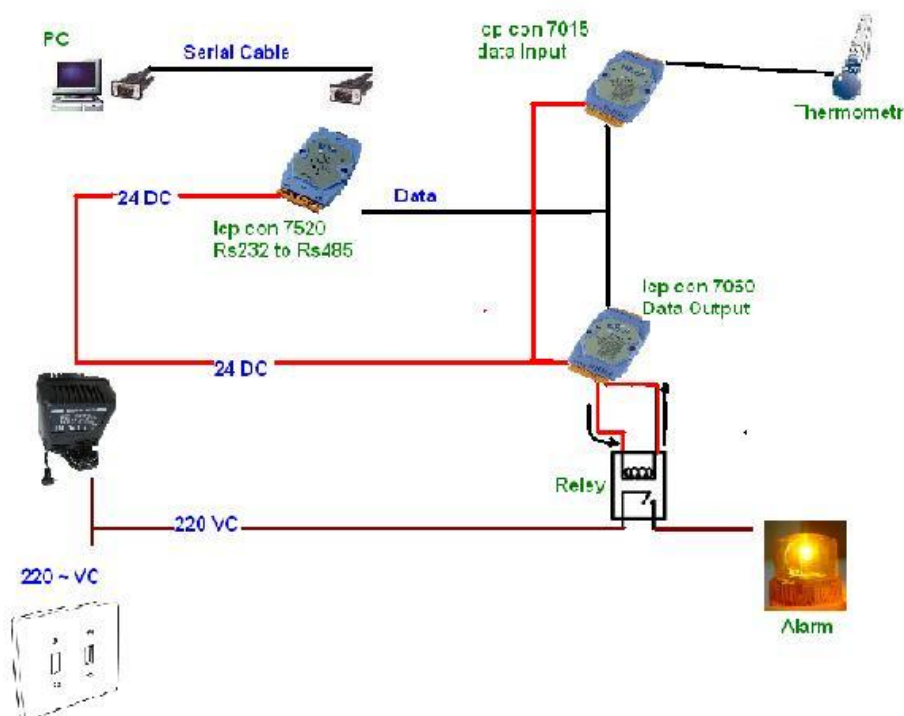


Δίπλα βλέπουμε την φωτογραφία του I-7520R

Παρακάτω παρουσιάζω την συνδεσμολογία του module I-7520R και την κατασκευή του



Χρησιμοποιώντας λοιπόν αυτό το υλικό φτιάχνω το κύκλωμα με το οποίο δουλεύω.



Ο τρόπος συνδεσμολογίας και τροφοδοσίας του υλικού έγινε με βάση το Manual των modules στο:

<http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/7000/manual/7520.pdf>

<http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/7000/manual/7000dio.pdf>