

# Εισαγωγή σε πληροφοριακά μετρητικά συστήματα

## 1.1 Εισαγωγή

Προγραμματιστικά μετρητικά συστήματα όπως το LabView (Laboratory Virtual Instrument Engineering Workbench) είναι εργαλεία τα οποία μπορεί να χρησιμοποιηθούν για την σχεδίαση και την ανάπτυξη πολύπλοκων εφαρμογών, ικανών να αλληλεπιδρούν με το εξωτερικό περιβάλλον. Το βασικό πλεονέκτημα του βασίζεται στον γραφικό προγραμματισμό (Graphical Programming), που χρησιμοποιεί, ο οποίος έχει αναπτυχθεί από την National Instruments. Η γραφική του φύση, το κάνει ιδανικό για εφαρμογές μετρήσεων, αυτοματισμού, ελέγχου οργάνων και ανάλυση δεδομένων. Το αποτέλεσμα όλων αυτών είναι η υψηλή παραγωγικότητα και η ευκολία ανάπτυξης εφαρμογών.

Κατά την παρουσίαση του Εργαστηρίου Συστημάτων Αυτομάτου Ελέγχου (ΣΑΕ) οι ελεγκτές που θα χρησιμοποιηθούν και αναπτυχθούν στις εργαστηριακές ασκήσεις θα εφαρμοστούν σε πραγματικές πειραματικές πειραματικές διατάξεις ενώ η θεωρία που θα χρησιμοποιηθεί, καλύπτει σημαντικό μέρος της σημερινής τεχνολογίας που χρησιμοποιείται από την βιομηχανία.

Για περισσότερες πληροφορίες, οδηγίες χρήσεως και on-line help σχετικά με τα προγράμματα LabView και Matlab καλείστε να επισκεφτείτε τα παρακάτω αντίστοιχα sites:

[www.ni.com](http://www.ni.com)

[www.mathworks.com](http://www.mathworks.com)

## 1.2 Εικονικά όργανα (Virtual Instruments)

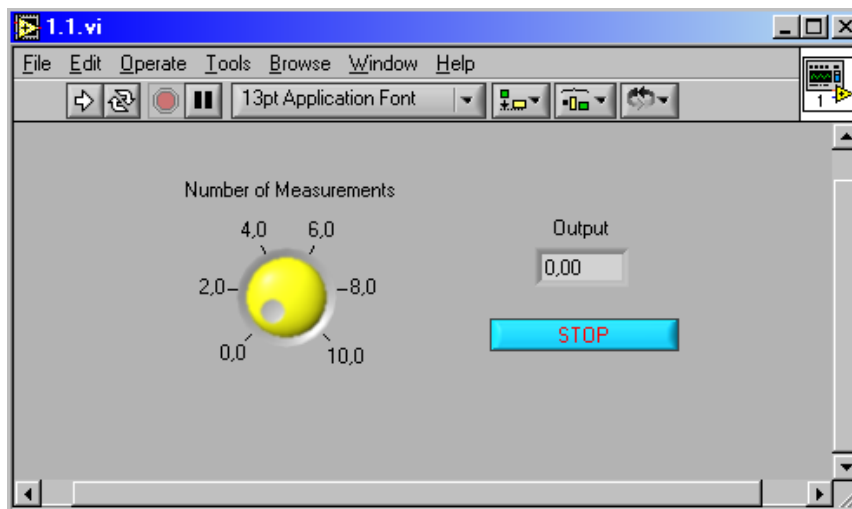
Τα εικονικά όργανα (Virtual Instruments, VI) είναι στοιχεία προγραμματισμού του LabView, τα οποία αποτελούνται από το *front panel* και το *block diagram*. Το *front panel* χρησιμοποιείται για την απεικόνιση στοιχείων τύπου *controls* και *indicators* για τον χρήστη, ενώ στο *block diagram* περιέχεται ο κώδικας για το εικονικό όργανο. Μέσω του *front panel* και των στοιχείων *controls* οι χρήστες μπορούν να θέσουν τιμές στις μεταβλητές να επιλέξουν διαδικασίες και γενικά να ελέγξουν την ροή και τον τρόπο εκτέλεσης του προγράμματος. Μέσω των στοιχείων *indicators* οι χρήστες μπορούν να παρατηρήσουν την

μεταβολή των μεταβλητών, τα αποτελέσματα διεργασιών και γενικά όλη την πληροφορία που παράγεται κατά την εκτέλεση του γραφικού κώδικα στο *block diagram*.

Οι κλασσικές γλώσσες προγραμματισμού (C, Basic) χρησιμοποιούν συναρτήσεις και υπορουτίνες για την ανάπτυξη των εφαρμογών τους. Στο περιβάλλον του Labview οι αντίστοιχες οντότητες ονομάζονται εικονικά όργανα (Virtual Instruments). Πολλαπλά VI μπορούν να χρησιμοποιηθούν για την δημιουργία εφαρμογών μεγάλης κλίμακας. Αντίστοιχα ένα VI μπορεί να χρησιμοποιηθεί για την επικοινωνία με τον χρήστη ή σαν μια υπορουτίνα σε κάποια εφαρμογή.

### 1.3 Front panel εικονικού οργάνου



Στην εικόνα 1.1 απεικονίζεται το front panel ενός εικονικού οργάνου (VI) του Labview. Περιέχει ένα περιστροφικό διακόπτη (knob) για την επιλογή των μετρήσεων, ένα αντικείμενο (indicator) για την απεικόνιση της μετρούμενης τιμής και ένα κουμπί για stop. Front Panel τέτοιου είδους μπορεί να δημιουργηθούν χωρίς ιδιαίτερη προσπάθεια και σε μικρό χρονικό διάστημα.

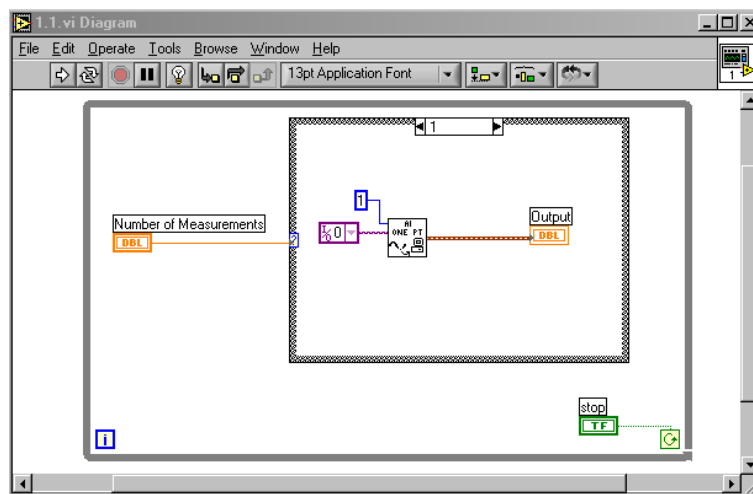


Εικόνα 1.1: Front Panel εικονικού οργάνου

### 1.4 Block diagram εικονικού οργάνου





Στο Σχήμα 1.2 απεικονίζεται το δομικό διάγραμμα, ή ο πηγαίος κώδικας, της εφαρμογής του Σχήματος 1.1. Τα VIs προγραμματίζονται στην γλώσσα-G όπου το πρόγραμμα γίνεται compile από την μηχανή εκτέλεσης του Labview. Ακολουθώς τα VIs μετατρέπονται σε εκτελέσιμο κώδικα που μπορεί να εκτελεστεί από την μηχανή διαδικασιών του Labview. Κάθε φορά που μια αλλαγή συμβαίνει στο VI, το Labview ενημερώνει κατάλληλα και τον πηγαίο κώδικα. Αν το δομικό διάγραμμα δεν είναι σωστά σχεδιασμένο τότε στο σημείο που υπάρχει το πρόβλημα, το καλώδιο σύνδεσης των δύο δομικών

στοιχείων είναι διακεκομμένο, καθώς επίσης το βελάκι εκτέλεσης έχει την επόμενη μορφή . Στην αντίθετη περίπτωση, το βελάκι έχει την επόμενη μορφή  και ο χρήστης είναι έτοιμος να εκτελέσει το VI. Τα δομικά διαγράμματα που σχεδιάζουμε, δεν καθορίζουν και την σειρά με την οποία τα στοιχεία που χρησιμοποιούμε θα εκτελεστούν. Μπορούμε να υποχρεώσουμε μια σειρά εκτέλεσης, με το να απαιτήσουμε κάποιο στοιχείο να λάβει δεδομένα από κάποιο άλλο, πριν από την εκτέλεση του. Επιπλέον χρησιμοποιώντας πιο πολύπλοκες προγραμματιστικές δομές, μπορούμε να καθορίσουμε με ακρίβεια τον τρόπο εκτέλεσης των διαδικασιών. Πιο γενικά κατά την διάρκεια του προγραμματισμού, θα είναι εμφανές ότι ορισμένες λειτουργίες θα πρέπει να πραγματοποιηθούν, πριν από κάποιες άλλες. Είναι ευθύνη του προγραμματιστή να επιλέξει τον τρόπο και να εξασφαλίσει την σωστή αλληλουχία κατά την εκτέλεση των διαδικασιών.




Σχήμα 1.2: Block Diagram του εικονικού οργάνου Σχήματος 1.1

## 1.5 Εκτέλεση των Virtual Instruments

Ένα πρόγραμμα στο Labview, εκτελείται με το πάτημα του βέλους ή το *Run Button*  τα οποία βρίσκονται στην παλέτα, στο πάνω μέρος του παραθύρου. Κατά την διάρκεια της εκτέλεσης το *Run Button* αλλάζει σε ένα μαύρο χρώμα . Θα πρέπει να σημειώσουμε πως κατά την διάρκεια της λειτουργίας τα κουμπιά που εμφανίζονται στην παλέτα είναι το *Continuous Run*, *Stop* και *Pause buttons* . Τα τρία τελευταία κουμπιά  χρησιμοποιούνται για την στοίχιση των αντικειμένων που εισάγουμε. Τα VIs συνήθως εκτελούνται από τα Front Panels, ενώ έχουν την δυνατότητα να εκτελεστούν και από τα Block Diagrams. Αυτό επιτρέπει στον προγραμματιστή να τρέξει το πρόγραμμα και να χρησιμοποιήσει κάποια άλλα εργαλεία που θα αναφερθούν στη συνέχεια και που είναι χρήσιμα κατά την διαδικασία του Debugging.

Αν το βέλος *Run* εμφανιστεί σαν σπασμένο βέλος, τότε αυτό υποδεικνύει ότι το πρόγραμμα δεν μπορεί να εκτελεστεί, λόγω κάποιων λαθών. Όταν όλα τα λάθη έχουν διορθωθεί, το σπασμένο βέλος θα αντικατασταθεί από το κανονικό βέλος που υποδηλώνει ότι το πρόγραμμα είναι έτοιμο για λειτουργία. Κατά την διάρκεια της ανάπτυξης, θα παρατηρήσουμε ότι στην παλέτα απεικονίζεται το σπασμένο βέλος. Αν η ένδειξη αυτή παραμείνει και μετά την ολοκλήρωση του προγραμματισμού, τότε με το πάτημα του σπασμένου βέλους, μπορούμε να δούμε μια λίστα με όλα τα λάθη που έχουν παρουσιαστεί κατά την διάρκεια της ανάπτυξης. Στρατηγικές για την διόρθωση των σφαλμάτων θα παρουσιαστούν σε επόμενο κεφάλαιο.

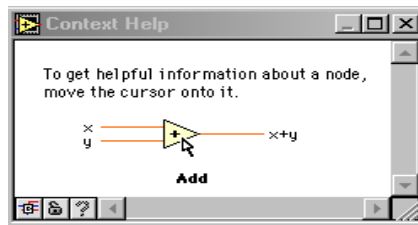
Η παλέτα περιέχει ακόμα τέσσερα κουμπιά  στο δομικό διάγραμμα τα οποία δεν είναι διαθέσιμα στο Front Panel και χρησιμοποιούνται για το debugging μιας εφαρμογής. Το κουμπί με την λάμπα, είναι για βηματική εκτέλεση του προγράμματος, ενώ τα τρία επόμενα είναι για την εκτέλεση βημάτων μέσα στον κώδικα.

## 1.6 Αρχειοθέτηση στο Labview

Το πρόγραμμα Labview, χρησιμοποιεί για την αποθήκευση αρχείων την ονοματολογία filename.vi. Επιπλέον πολλαπλά VIs μπορούν να σωθούν σε ένα αρχείο με την μορφή βιβλιοθήκης και με κατάληξη \*.llb. Οι βιβλιοθήκες, είναι χρήσιμες για την οργάνωση πολλαπλών συσχετιζόμενων VIs. Όταν καλούμε κάποιο VI που εμπεριέχει και κάποια άλλα sub-VIs το πρόγραμμα προσπαθεί να τα εντοπίσει γρήγορα. Η χρήση βιβλιοθήκης, απλοποιεί την όλη διαδικασία, τοποθετώντας όλα τα σχετικά VIs κάτω από το ίδιο directory. Στην περίπτωση που θέλουμε να δημιουργήσουμε μία βιβλιοθήκη με VIs, θα πρέπει να επιλέξουμε την επιλογή Save with Options, από το File Menu.

## 1.7 Χρήση βοήθειας στο Labview

Το LabView, παρέχει έναν on-line οδηγό παροχής βοήθειας για όποια λειτουργία θέλουμε να πραγματοποιήσουμε. Το κάθε εικονοόργανο, μπορεί να χρειάζεται έναν μεγάλο αριθμό και διαφορετικά είδη συνδέσεων. Ο αρχάριος χρήστης και ειδικά ο αρχάριος θα πρέπει να χρησιμοποιεί την δυνατότητα για on-line help. Η λειτουργία αυτή ενεργοποιείται από το Help Menu επιλέγοντας Show Context Help. Η ενεργοποίηση της λειτουργίας αυτής εμφανίζει ένα παράθυρο που εμπεριέχει όλες τις απαραίτητες πληροφορίες για τον τρόπο διασύνδεσης του εικονικού οργάνου. Αφήνοντας το mouse pointer ακίνητο πάνω από ένα εικονικό όργανο, το παράθυρο με τις οδηγίες θα απεικονίσει τις σχετικές με αυτό το όργανο πληροφορίες. Ένα τυπικό παράθυρο βοήθειας απεικονίζεται στο Σχήμα 1.3. Η χρήση του είναι σχεδόν αναγκαία με σκοπό την εύκολη και επιτυχή διασύνδεση και θα πρέπει σχεδόν πάντα να ενεργοποιείται.



Σχήμα 1.3: Παράθυρο βοήθειας σε περιβάλλον Labview

## 1.8 Οι παλέτες εργασίας στο Labview



Το LabView περιέχει γραφικές παλέτες εργασίας σε μορφή menu με σκοπό την περαιτέρω βοήθεια στη δημιουργία και εκτέλεση των εικονικών οργάνων. Οι τρεις διαθέσιμες παλέτες είναι **Tools**, **Controls** και **Functions**. Στη συνέχεια θα γίνει η παρουσίαση της κάθε μιας επιμέρους παλέτας εργασίας.









### *Tools Palette*

Η παλέτα αυτή περιέχει εργαλεία που είναι κατάλληλα στην δημιουργία και εκτέλεση των εικονικών οργάνων. Εάν η παλέτα δεν είναι ορατή, τότε για την εμφάνιση της επιλέγουμε Show Tools Palette από το Windows menu. Μετά την επιλογή ενός εργαλείου από αυτό το menu, ο κέρσορας παίρνει την μορφή του επιλεγμένου εργαλείου. Τοποθετώντας οποιοδήποτε επιλεγμένο εργαλείο πάνω από ένα sub-VI ή συνάρτηση, εμφανίζονται στο Help παράθυρο πληροφορίες σχετικές με το sub-VI ή την συνάρτηση. Για αυτή την λειτουργία, θα πρέπει να έχει προεπιλεγεί η εμφάνιση του παραθύρου βοήθειας, με τον τρόπο που προαναφέρθηκε. Η παλέτα εργασίας παρουσιάζεται στο Σχήμα 1.4, ακολουθούμενη από την λειτουργικότητα των επι-μέρους υπό-εργαλείων.



Σχήμα 1.4: Παλέτα εργαλείων Labview

	<p><i>Operating Tool:</i> για να χειριστούμε κουμπιά ελέγχου και απεικόνιση στο Front Panel.</p>
	<p><i>Positioning Tool:</i> για να επιλέξουμε, μετακινήσουμε ή να αλλάξουμε το μέγεθος. Στην περίπτωση της αλλαγής μεγέθους, το αντικείμενο, θα πρέπει να θεωρηθεί ότι συμπεριφέρεται σαν παράθυρο.</p>
















	<i>Labelling Tool:</i> για να εισάγουμε κείμενο σε ετικέτες (Labels) αντικειμένων.
	<i>Wiring Tool:</i> για να διασυνδέσουμε αντικείμενα στο Block Diagram.
	<i>Menu Tool:</i> για να εκτελούμε διαδικασία Pop-up πάνω σε ένα menu ενός οργάνου κάνοντας αριστερό click, αντί για δεξί.
	<i>Scrolling Tool:</i> για την μετακίνηση μέσα στο παράθυρο, χωρίς να χρησιμοποιούμε τα scroll bars των Windows.
	<i>Breakpoint Tool:</i> για να εισάγουμε breakpoints μέσα στην διαδικασία εκτέλεσης ενός εικονικού οργάνου.
	<i>Probe Tool:</i> για να εισάγουμε probes σε καλώδια στο δομικό διάγραμμα και να βλέπουμε την τιμή των μεταβλητών που μεταφέρουν.
	<i>Color Copy Tool:</i> για να αντιγράφουμε ένα χρώμα με σκοπό την εφαρμογή του από το Color Tool.
	<i>Color Tool:</i> για να βάφουμε ένα αντικείμενο, είτε στην μπροστινή ή οπίσθια όψη.

### **Controls Palette**

Μπορούμε να εισάγουμε κουμπιά ελέγχου και απεικόνισης στο Front Panel από την Controls Palette. Κάθε επιλογή σε αυτή την παλέτα, εμπεριέχει μια υπο-επιλογή με επιπλέον κουμπιά ελέγχου και απεικόνισης σχετικά με την προηγούμενη επιλογή. Σε περίπτωση που η παλέτα δεν είναι εμφανής, μπορούμε να την ανοίξουμε, επιλέγοντας, Show Controls Palette από το Windows menu. Επιπλέον μπορούμε να ανοίξουμε την Controls Palette, κάνοντας δεξί click στο mouse πάνω σε μια άδεια περιοχή του front panel. Προσοχή θα πρέπει να δώσουμε στο γεγονός ότι η παλέτα αυτή είναι ενεργή μόνο όταν βρισκόμαστε στο front panel. Η Controls παλέτα, απεικονίζεται στο Σχήμα 1.5 ακολουθούμενη από την λειτουργικότητα των υπό εργαλείων της.



Σχήμα 1.5: Controls Παλέτα στο Labview







	Numeric sub-palette: Περιέχει κουμπιά ελέγχου και απεικόνιση για αριθμητικά δεδομένα
	Boolean sub-palette: Περιέχει κουμπιά ελέγχου και απεικόνιση για Boolean δεδομένα
	String sub-palette: Περιέχει κουμπιά ελέγχου και απεικόνιση για ASCII προτάσεις και πίνακες με χαρακτήρες.
	Array & Cluster sub-palette: Περιέχει λειτουργίες σχετικές με την δημιουργία πινάκων και δομών δεδομένων.
	List & Ring sub-palette: Περιέχει κουμπιά ελέγχου και απεικόνιση για την δημιουργία menu και πινάκων επιλογής.
	Graph sub-palette: Περιέχει εργαλεία απεικόνισης γραφημάτων ή δεδομένων πραγματικού χρόνου
	Ring & Enum: Περιέχει εργαλεία για την δημιουργία διατάξεων επιλογής διαδικασιών μέσω menu
	I/O: Περιέχει λειτουργίες για την επικοινωνία με συσκευές
	RefNum: Περιέχει λειτουργίες σχετικές με τον ορισμό του path των αρχείων.
	Dialog Controls: Απεικονίζει ένα κουτί διαλόγου με την εφαρμογή.
	Classic Controls: Περιέχει κουμπιά ελέγχου και απεικόνιση που είχαν οι προηγούμενες εκδόσεις του Labview.
	ActiveX: Περιέχει σχετικές λειτουργίες για την εκτέλεση του ActiveX μηχανισμού.
	Decorations: Περιέχει γραφικά αντικείμενα για την προσαρμογή και διακόσμηση των front panels
	Select Control: Περιέχει κουμπιά ελέγχου και απεικόνιση, κατασκευασμένα από τον χρήστη
	User Lib: Περιέχει ειδικά κουμπιά ελέγχου που μπορεί να δημιουργήσει ο χρήστης.

## Functions Palette
















Με τις λειτουργίες της Function Palette, δημιουργούμε το Block diagram. Κάθε επιλογή στην παλέτα, περιέχει έναν αριθμό από υπό παλέτες, σχετικές με την προηγούμενη επιλογή. Σε περίπτωση που η παλέτα δεν είναι εμφανής, μπορούμε να την ανοίξουμε, επιλέγοντας, Show Function Palette από το Windows menu. Επιπλέον μπορούμε να ανοίξουμε την Function Palette, κάνοντας δεξί click στο mouse πάνω σε μια άδεια περιοχή του block diagram. Προσοχή θα πρέπει να δώσουμε στο γεγονός ότι η παλέτα αυτή είναι ενεργή μόνο όταν βρισκόμαστε στο block diagram. Η Function παλέτα, απεικονίζεται στο Σχήμα 1.6 ακολουθούμενη από την λειτουργικότητα των υπό εργαλείων της.





Σχήμα 1.5: Functions παλέτα στο Labview

	Structures sub-palette: Περιέχει προγράμματα ελέγχου εκτέλεσης του κώδικα, όπως for, while loops.
	Numeric sub-palette: Περιέχει τριγωνομετρικές, λογαριθμικές και άλλες μαθηματικές συναρτήσεις.
	Boolean sub-palette: Περιέχει λογικές και Boolean συναρτήσεις.
	String sub-palette: Περιέχει συναρτήσεις για την επεξεργασία προτάσεων από χαρακτήρες.
	Array sub-palette: Περιέχει συναρτήσεις για την επεξεργασία πινάκων.
	Cluster sub-palette: Περιέχει συναρτήσεις για την επεξεργασία συμπλεγμάτων (clusters).

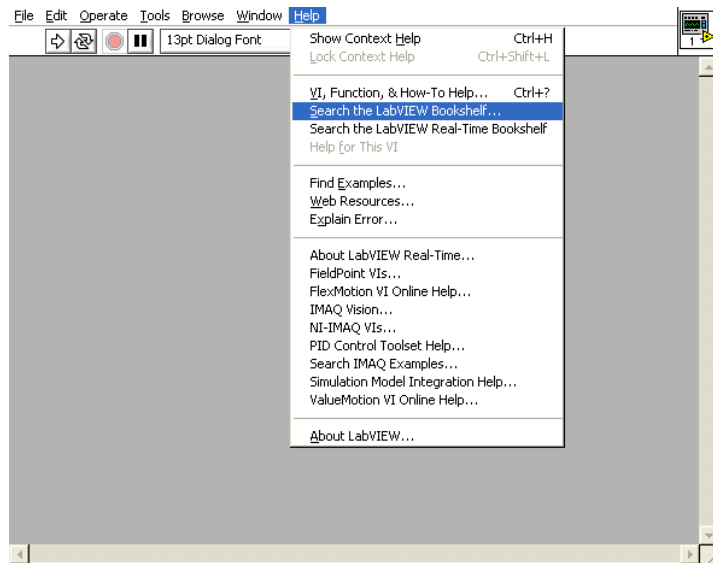


	Comparison sub-palette: Περιέχει συναρτήσεις για την σύγκριση αριθμών, λογικών μεταβλητών και προτάσεων από χαρακτήρες.
	Time & Dialog sub-palette: Περιέχει συναρτήσεις για την δημιουργία παραθύρων διαλόγου, χρονισμού και χειρισμού λαθών.
	File I/O sub-palette: Περιέχει συναρτήσεις και εικονικά όργανα για την εκτέλεση I/O λειτουργιών.
	Data Acquisition sub-palette: Περιέχει εικονικά όργανα για την σύνδεση με κάρτες συλλογής δεδομένων.
	Waveform sub-palette: Περιέχει συναρτήσεις για την επεξεργασία και χρησιμοποίηση κυματομορφών.
	Analyze sub-palette: Περιέχει συναρτήσεις για την ανάλυση των δεδομένων.
	Instrument I/O sub-palette: Περιέχει εικονικά όργανα για τον έλεγχο GLIB, serial και VISA οργάνων.
	Imaq Vision sub-palette: Περιέχει συναρτήσεις για ψηφιακή επεξεργασία εικόνας.
	Mathematics sub-palette: Περιέχει πιο προηγμένες συναρτήσεις για την εκτέλεση μαθηματικών υπολογισμών με πίνακες και δεδομένα.
	Communication sub-palette: Περιέχει εικονικά όργανα για λειτουργίες TCP, DDE, Apple Events και OLE.
	Application Control sub-palette: Περιέχει εικονικά όργανα, κατάλληλα για την εκτέλεση και έλεγχο εικονικών οργάνων σε άλλους υπολογιστές.
	Graphics & Sound sub-palette: Περιέχει συναρτήσεις εισαγωγής και επεξεργασίας εικόνων και μουσικής
	Tutorial sub-palette: Περιέχει εικονικά όργανα που παρουσιάζονται στα εγχειρίδια χρήσεως του Labview.
	Report Generation sub-palette: Περιέχει εικονικά όργανα για την γρήγορη και εύκολη δημιουργία αναφορών, για εφαρμογές που έχουν σχεδιαστεί στο Labview.
	Advanced sub-palette: Περιέχει προηγμένες συναρτήσεις για

	το κάλεσμα προγραμμάτων γραμμένων σε C, DLLs και επεξεργασία δεδομένων.
	Select a VI sub-palette: Εισαγωγή ενός έτοιμου εικονικού οργάνου, στην επιφάνεια εργασίας του Labview.
	User Libraries sub-palette: Περιέχει μία λίστα από εικονικά όργανα που έχουν σχεδιαστεί από τον χρήστη.

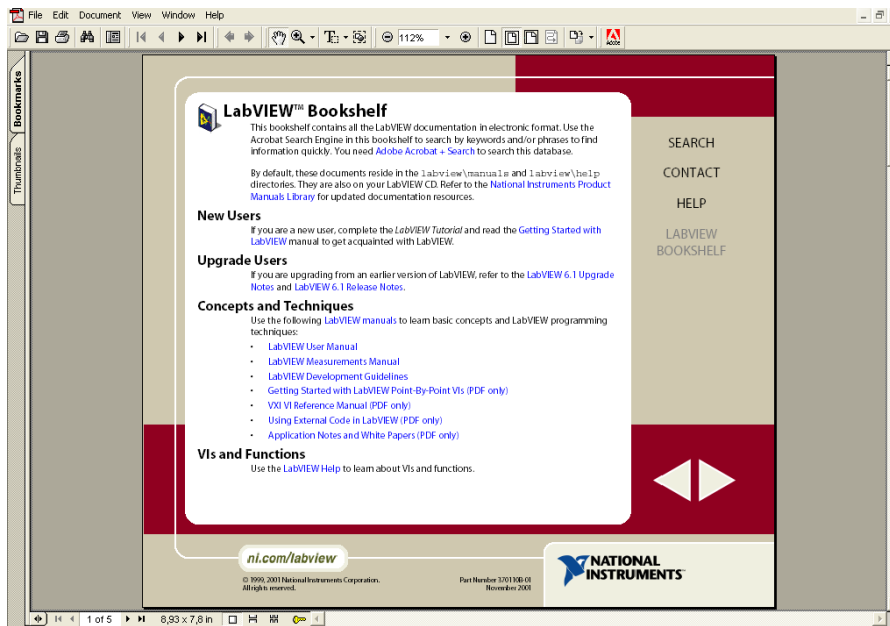
Οι περισσότερες από τις παρουσιαζόμενες συναρτήσεις θα αναλυθούν περισσότερα στα επόμενα κεφάλαια, μέσα από την αναλυτική παρουσίαση του τρόπου δημιουργίας μερικών απλών εικονικών οργάνων.

Ο ενδιαφερόμενος αναγνώστης, μπορεί να ανατρέξει στο σύνολο των βοηθητικών εγγράφων (σε ηλεκτρονική μορφή) που συνοδεύουν το Labview. Για να έχουμε πρόσβαση στα έγγραφα αυτά, θα πρέπει στο front panel από το μενού να επιλέξουμε *Search the LabView bookshelf*, όπως φαίνεται στο Σχήμα 1.6.



Σχήμα 1.6: Έγγραφα βοήθειας στο LabView

Αυτόματα η επιλογή αυτή θα ανοίξει το Acrobat Reader ([www.adobe.com](http://www.adobe.com)) όπου και θα βρεθούμε στην κεντρική σελίδα των εγγράφων βοήθειας, όπως απεικονίζεται στο Σχήμα 1.7. Από εκεί μπορούμε να ανατρέξουμε στο κεφάλαιο που θέλουμε να εμβαθύνουμε, τις σχετικές με το LabView γνώσεις μας.



Σχήμα 1.8: Η κεντρική οθόνη περιήγησης στην βοήθεια του Labview

Από τα εγχειρίδια αυτά, οι συγγραφείς θεωρούν απαραίτητο για μια εις βάθος και ουσιαστική γνώση του LabView, την ανάγνωση των παρακάτω:

- Getting Started with LabVIEW
- LabVIEW User Manual
- LabVIEW Measurements Manual

# Σχεδιασμός μετρητικών εικονικών οργάνων

## 2.1 Σκοπός της άσκησης

Σκοπός της άσκησης είναι η εκμάθηση των βασικών προγραμματιστικών αρχών της αντικειμενοστραφούς γλώσσας G που χρησιμοποιείται από το Labview, για τον σχεδιασμό μετρητικών εικονικών οργάνων.

## 2.2 Κύρια δομικά διαγράμματα

### *To Front Panel*

Στο Front Panel σχεδιάζουμε εικονικά όργανα με ένα συνδυασμό από κουμπιά ελέγχου και απεικόνισης. Τα κουμπιά ελέγχου είναι ο τρόπος παροχής δεδομένων στα εικονικά όργανα. Τα κουμπιά απεικόνισης έχουν την δυνατότητα να απεικονίζουν πληροφορίες που παράγονται από το εικονικά όργανα. Υπάρχουν πολλοί τύποι κουμπιών ελέγχου και απεικόνισης, τα οποία μπορούν να εισαχθούν στο Front Panel από τις διάφορες υπό-παλέτες της Controls παλέτας.

Ως παράδειγμα τα δύο κυρίως χρησιμοποιημένα αριθμητικά αντικείμενα είναι τα digital control και digital indicator. Για να εισάγουμε ή να αλλάξουμε την τιμή σε ένα digital control, είτε μπορούμε να κάνουμε click πάνω στα κουμπιά αυξομείωσης με το operating tool, είτε να κάνουμε διπλό click πάνω στον αριθμό με το labeling tool ή το operating tool.

### *Block Diagram*

Στο Block Diagram μπορούμε να προγραμματίσουμε την συμπεριφορά του εικονικού οργάνου. Κάθε αντικείμενο στο Front Panel, έχει την δική του οντότητα στο Block Diagram και αλληλεπιδρά με το πρόγραμμα που αναπτύσσεται εκεί. Στο Block Diagram έχουμε αντικείμενα τύπου nodes και terminals. Τα nodes είναι εκτελέσιμα στοιχεία προγραμματισμού και έχουν λειτουργία ανάλογη με αυτή των ορισμάτων, των υπορουτινών και των συναρτήσεων σε μια κοινή γλώσσα προγραμματισμού.

Υπάρχουν τέσσερις τύποι nodes, οι οποίοι είναι οι: α) functions, β) subVI nodes, γ) structures και δ) Code Interface Nodes (CINs).

Τα functions είναι έτοιμες συναρτήσεις για την εκτέλεση βασικών λειτουργιών όπως είναι η πρόσθεση αριθμών, file I/O, μορφοποίηση και χειρισμός χαρακτήρων.

Τα subVIs είναι εικονικά όργανα τα οποία σχεδιάζουμε και στη συνέχεια μπορούμε να τα ξανακαλέσουμε σε άλλα εικονικά όργανα, ως κομμάτι αυτών.

Τα structures όπως είναι οι βρόχοι For, Loop, While και Case ελέγχουν την ροή του προγράμματος.

Τα CINS είναι ένας τρόπος επικοινωνίας του Block Diagram και κώδικα αναπτυγμένο από τον χρήστη σε γλώσσα C.

Τα terminals είναι στοιχεία αντίστοιχα με τις παραμέτρους και τις σταθερές, μέσα από τα οποία τα δεδομένα περνάνε από το Block Diagram στο Front Panel. Υπάρχουν δύο τύποι Terminals: τα control-indicator terminals και τα node terminals.

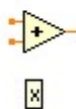
Τα control-indicator terminals ανήκουν στο Front Panel. Οι τιμές των μεταβλητών που τοποθετεί ένας χρήστης κατά την εκτέλεση ενός εικονικού οργάνου, περνάνε από το Front Panel στο Block Diagram μέσω από αυτά τα Terminals. Όταν εκτελεστεί ο κώδικας στο Block Diagram, η παραγόμενη πληροφορία, περνά στο Front Panel, μέσα από τα indicators Terminals.

Τα control-indicator Terminals δημιουργούνται και διαγράφονται αυτόματα όταν δημιουργούμε ή διαγράφουμε ένα κουμπί ελέγχου ή απεικόνισης από το Front Panel.

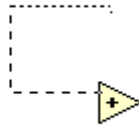
### ***Διασύνδεση (Wiring)***

Τα καλώδια (wires) είναι ο δρόμος μεταγωγής των δεδομένων μεταξύ των αντικειμένων. Διαφορετικοί τύποι καλωδίων αντιστοιχούν σε διαφορετικούς τύπους δεδομένων (π.χ. ακέραιοι αριθμοί κινητής υποδιαστολής). Σε μια έγχρωμη εικόνα, κάθε τύπος δεδομένων, απεικονίζεται και με ένα διαφορετικό χρώμα για μεγαλύτερη έμφαση.

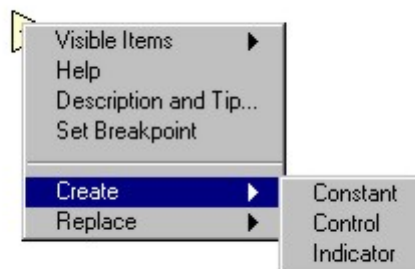
Για να συνδέσουμε ένα αντικείμενο με ένα άλλο χρησιμοποιούμε το wiring tool στο πρώτο terminal και στη συνέχεια μετακινούμε το εργαλείο στο επόμενο terminal και κάνουμε click σε αυτό. Στο επόμενο σχήμα απεικονίζονται τα terminals ενός αθροιστή.



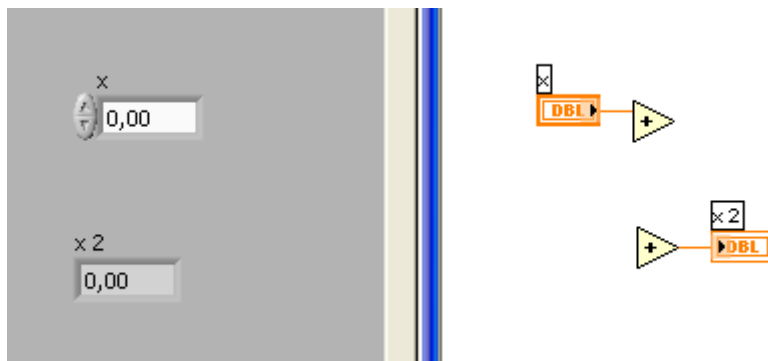
Όταν το wiring tool βρίσκεται πάνω από την περιοχή ενός Terminal τότε η περιοχή αυτή αναβοσβήνει, υποδηλώνοντας ότι ένα click θα συνδέσει το καλώδιο σε αυτό το Terminal. Δεν χρειάζεται να έχουμε πατημένο το mouse κατά την διάρκεια της μετακίνησης του wiring tool από το ένα Terminal στο άλλο.



Με οποιοδήποτε εργαλείο του Labview μπορούμε να ανοίξουμε το pop-up menu πάνω σε οποιαδήποτε συνάρτηση ή subVI και να επιλέξουμε Create Control, Create Constant ή Create Indicator. Με αυτό τον τρόπο τα αντίστοιχα control, constant και indicator θα δημιουργηθούν αυτόματα, με έτοιμες τις καλωδιώσεις τους.

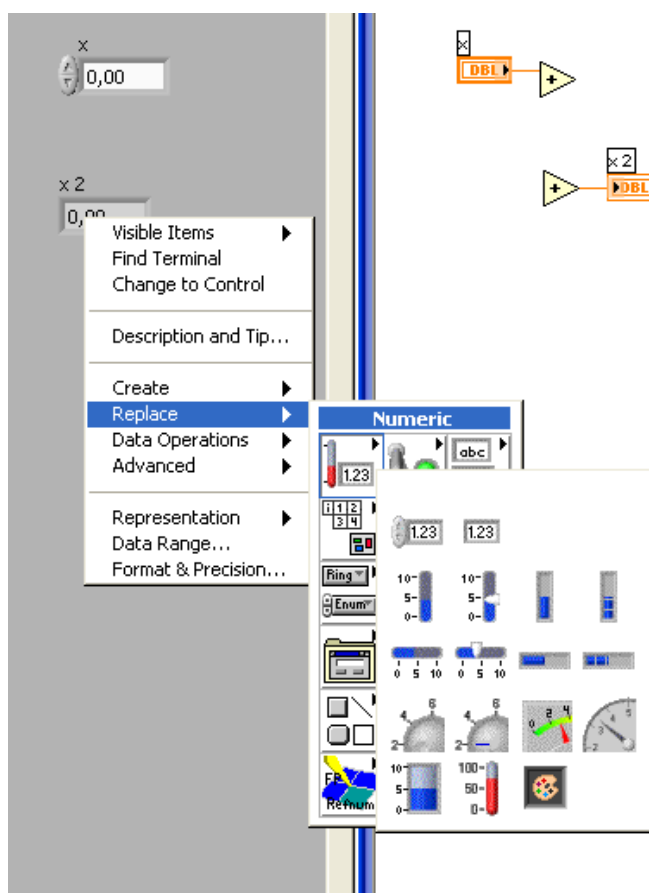


Εάν επιλέξουμε την δημιουργία ενός control/indicator αυτό θα έχει σαν αποτέλεσμα την δημιουργία του σχετικού control/indicator αυτόματα και στο front panel. Την λειτουργία αυτή απεικονίζει το επόμενο σχήμα, αναφορικά με την δημιουργία ενός control και ενός indicator.



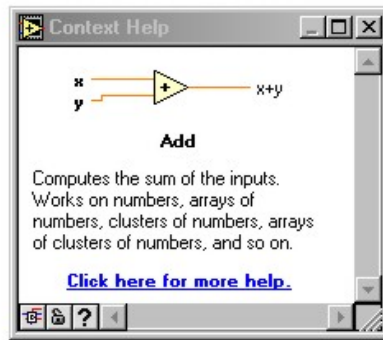
Σε αυτήν την λειτουργία όπου τα controls/indicators δημιουργούνται αυτόματα, η μορφή τους είναι δεδομένη (default) και αντιστοιχεί σε αυτή του απλού control/indicator που παρουσιάζεται στο προηγούμενο σχήμα. Με σκοπό την αντικατάσταση των control/indicator που έχουν δημιουργηθεί, από αυτά της αρεσκείας μας, και διατηρώντας πάντα τις διασυνδέσεις (καλωδιώσεις) που έχουν αυτόματα δημιουργηθεί, θα πρέπει στο front panel να κάνουμε δεξί click πάνω στο control/indicator που μας ενδιαφέρει και να επιλέξουμε την εντολή replace. Η διαδικασία αυτή απεικονίζεται στο επόμενο σχήμα. Στο σχήμα αυτό μπορούμε να παρατηρήσουμε όλες τις ενέργειες που μπορούμε να εκτελέσουμε σε κάθε

control/indicator. Οι πιο βασικές από τις ενέργειες αυτές θα αναλυθούν μέσα στο παρόν εγχειρίδιο, ενώ παράλληλα οι φοιτητές μπορούν να πειραματιστούν με την λειτουργικότητα των υπολοίπων ενεργειών.

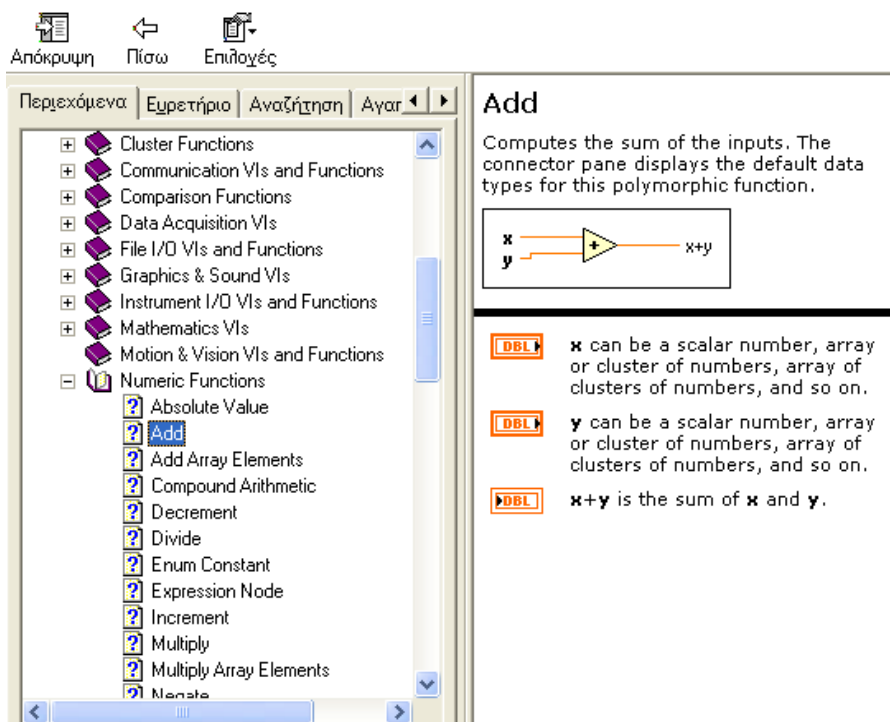


### **Χρησιμοποίηση βοήθειας**


Όλα τα ενσωματωμένα εικονικά όργανα και συναρτήσεις στο Labview περιέχουν πλήρη βοήθεια για την λειτουργικότητα αυτών. Όταν χρησιμοποιούμε ένα άγνωστο εικονικό όργανο ή μια συνάρτηση, μπορούμε να το τοποθετήσουμε στο Block Diagram, να ανοίξουμε το pop-up μενού, κάνοντας δεξί click και να επιλέξουμε On-line Help για μια πλήρη περιγραφή της λειτουργικότητας του και των συνδέσεων αυτού. Στην περίπτωση που χρειαζόμαστε μια γρήγορη εύρεση βοήθειας για να μας θυμίσει τον τρόπο λειτουργίας του εικονικού οργάνου καθώς και την χρήση των εισόδων και εξόδων αυτού, μπορούμε να επιλέξουμε show help από το help menu. Ακολούθως ένα παράθυρο βοήθειας εμφανίζεται σχετικά με το εικονικό όργανο που βρίσκεται κάτω από τον κέρσορα. Στο επόμενο σχήμα παρουσιάζουμε τις πληροφορίες που εμφανίζονται στο παράθυρο βοήθειας για την χρήση του αντικειμένου που εκτελεί την πρόσθεση μεταξύ δύο αριθμών.



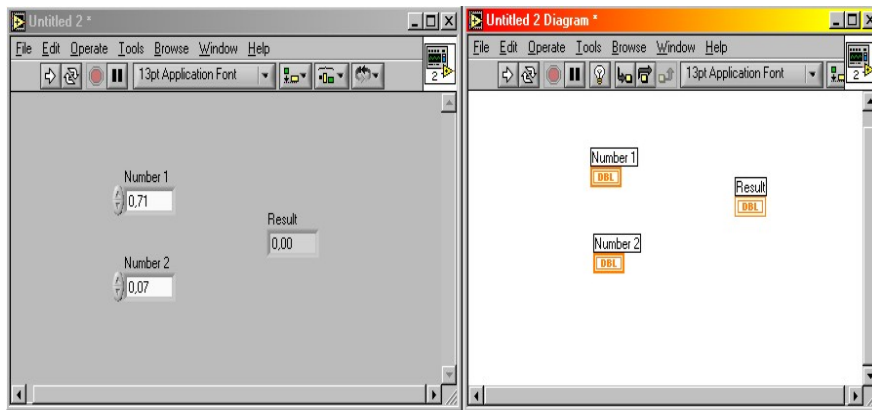
Εάν επιλέξουμε το «Click here for more help» τότε θα εμφανιστεί το κλασσικό παράθυρο βοήθειας των Windows όπου και θα αναφέρονται πιο εκτενείς πληροφορίες για την χρήση του κάθε οργάνου. Το περιγραφέν παράθυρο βοήθειας παρουσιάζεται στη συνέχεια.








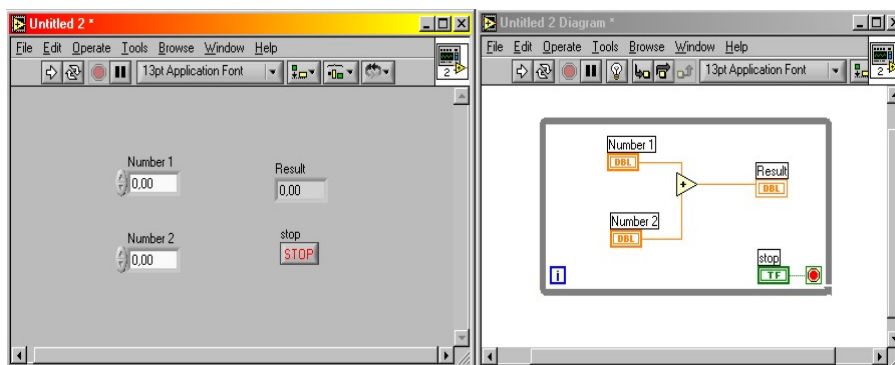
### 2.3 Σχεδιασμός ενός εικονικού οργάνου για την πρόσθεση δύο αριθμών


Με σκοπό τον σχεδιασμό ενός εικονοοργάνου για την πρόσθεση δύο αριθμών, αρχικά εισάγουμε στο front panel, δύο Digital Controls και ένα Digital Indicator. Με το Labeling tool  αλλάζουμε τα ονόματα, Numeric, Numeric2 και Numeric 3, αντίστοιχα σε Number 1, Number 2 και Result. Μετά την τοποθέτηση των controls και των indicators στο block diagram εμφανίζονται τα αντίστοιχα αντικείμενα.












Στο block diagram, εισάγουμε από το Function palette/numeric  το σύμβολο add που χρησιμοποιείται για την πρόσθεση αριθμών / πινάκων. Με την βοήθεια του Wiring tool  εκτελούμε τις διασυνδέσεις όπως εμφανίζονται στο παρακάτω διάγραμμα. Από το Function palette/Structures , επιλέγουμε το while loop και περικυκλώνουμε με αυτό όλο τον προηγούμενο σχεδιασμό μας. Στη συνέχεια στο  του while loop, εκτελούμε δεξί click και επιλέγουμε create control. Αυτόματα δημιουργείται ένα Boolean control button (στο front panel) με αρχική με τιμή false και ένδειξη stop. Κάνουμε ξανά δεξί Click πάνω στο  και επιλέγουμε Stop if true. Μετά από τις παραπάνω ενέργειες, θα πρέπει να έχουν δημιουργηθεί τα παρακάτω διαγράμματα.

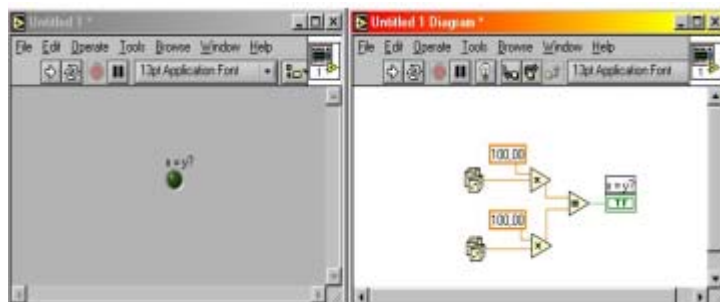


Το εικονοόργανο που σχεδιάσαμε, μπορεί να προσθέτει δύο αριθμούς συνεχώς (while loop) μέχρι μια λογική συνθήκη ελέγχου να παράγει μια τιμή true. Στο πρόγραμμα μας, η τιμή αυτή παράγεται με το πάτημα του κουμπιού stop. Ο δείκτης  μας πληροφορεί τις πόσες φορές ο βρόχος while, έχει εκτελεστεί.

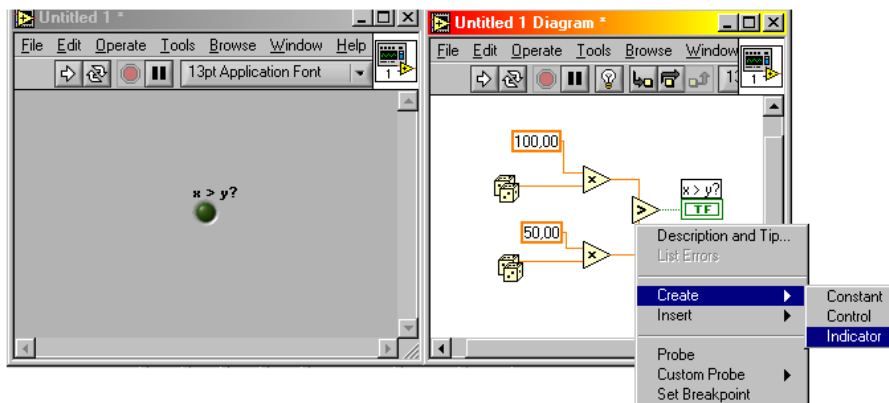
Στο σημείο αυτό, είμαστε σε θέση να αρχίσουμε την εκτέλεση του εικονικού οργάνου, για την πρόσθεση δύο αριθμών, πατώντας το κουμπί run . Το αποτέλεσμα εμφανίζεται στο πεδίο Result. Το πρόγραμμα αυτό θα εκτελείται μέχρι να πατήσουμε το κουμπί stop .



## 2.4 Σχεδιασμός ενός εικονοργάνου για την σύγκριση δύο τυχαίων αριθμών

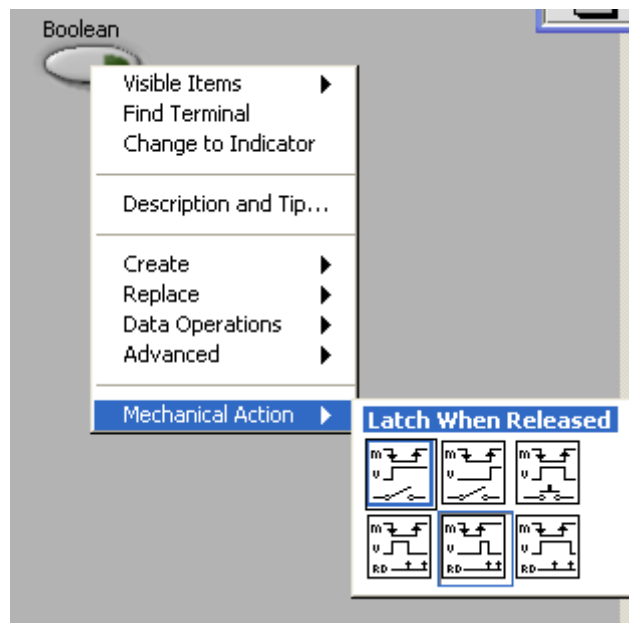
Στο Block diagram, επιλέγουμε από το Functions/numeric  δύο αντικείμενα «random number[0-1]» (Η γραφική απεικόνιση ισοδυναμεί με ένα ζάρι ) και τα εισάγουμε. Τα αντικείμενα αυτά μπορούν να μας δώσουν τυχαίους αριθμούς μέσα στο αριθμητικό διάστημα [0-1]. Με σκοπό να αυξήσουμε το διάστημα από [0-100], από το Functions/numeric εισάγουμε δύο αντικείμενο multiply . Με το Wiring tool συνδέουμε τα “ζάρια” με την μία άκρη ενός πολλαπλασιαστή και στη συνέχεια κάνοντας δεξί click, στα ελεύθερα άκρα, επιλέγουμε create constant και δίνουμε την τιμή 100. Με αυτόν τον τρόπο έχουμε τώρα τυχαίες τιμές, μέσα στο διάστημα (0-100). Από το Function/comparison  επιλέγουμε την συνάρτηση greater και με το wiring tool ενώνουμε τις πολλαπλασιασμένες εξόδους (με τον αριθμό 100), από τα “ζάρια” στις εισόδους της συνάρτησης greater . Η συνάρτηση αυτή, χρησιμοποιείται για να συγκρίνει δύο αριθμούς, και επιστέφει Boolean τιμή true ή false. Στην έξοδο της συνάρτησης με το wiring tool, κάνουμε δεξί click και επιλέγουμε create indicator. Η ενέργεια αυτή θα εμφανίσει στην οθόνη του Control Panel ένα σβηστό λαμπάκι με πράσινο χρώμα.



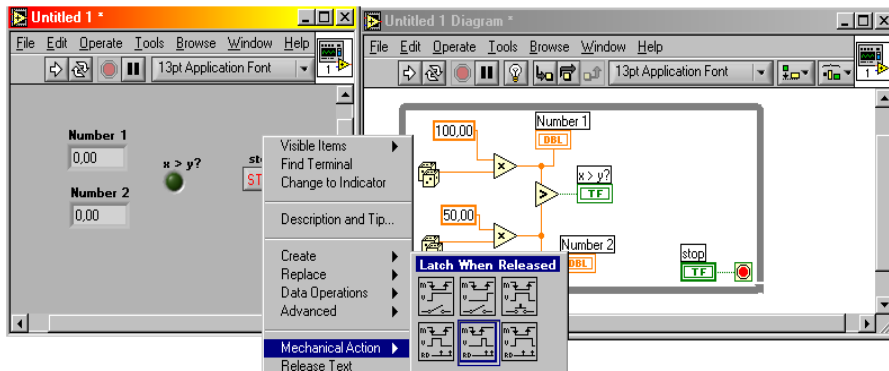
Επιπλέον για να βλέπουμε σε κάθε χρονική στιγμή ποιους αριθμούς συγκρίνουμε, με το wiring tool, πατάμε δεξί click πάνω στις εξόδους των πολλαπλασιαστών και επιλέγουμε create indicator.




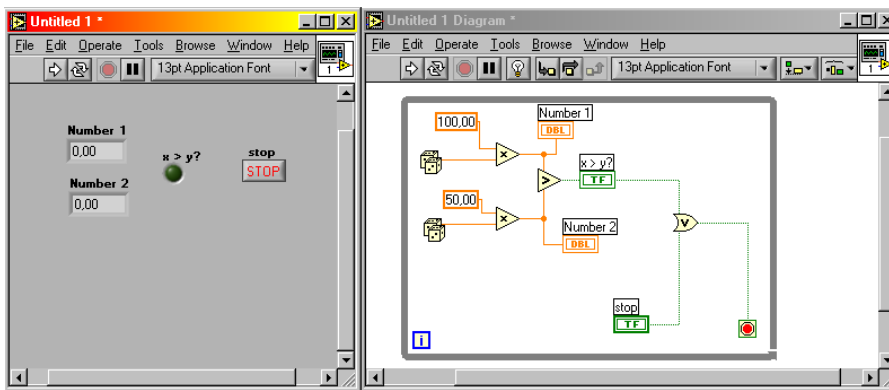
Με το Labeling Tool πηγαίνουμε στο Front Panel και μετονομάζουμε τα δύο Indicators σε Number 1 και Number 2, ενώ το  $x > y$  το μετονομάζουμε σε greater. Με σκοπό η διαδικασία αυτή να επαναλαμβάνεται συνεχώς περικλείουμε το πρόγραμμα μας με έναν βρόχο while. Ο βρόχος αυτός βρίσκεται κάτω από το Functions\structures. Στη συνέχεια στο  του while loop, εκτελούμε δεξί click και επιλέγουμε create control. Αυτόματα δημιουργείται ένα Boolean control button (στο front panel) με αρχική με τιμή false και ένδειξη stop. Κάνουμε ξανά δεξί Click πάνω στο  και επιλέγουμε Stop if true. Στο Front panel, πατάμε δεξί click πάνω στο button και επιλέγουμε Mechanical Action/latch when released. Επίσης με το Labeling tool, του δίνουμε την ονομασία Stop.



Με σκοπό το πρόγραμμα να σταματά μόνο αν  $a > b$  ή αν πατήσουμε εμείς το Stop κουμπί, θα πρέπει να προγραμματίσουμε τις επόμενες αλλαγές στο ήδη υπάρχον πρόγραμμα.




Αρχικά από το μενού Functions/Boolean εισάγουμε μια πύλη OR, στις οποίες τις εισόδους ενώνουμε την έξοδο από το equal και το κουμπί Stop. Τέλος την έξοδο της πύλης την συνδέουμε στο .





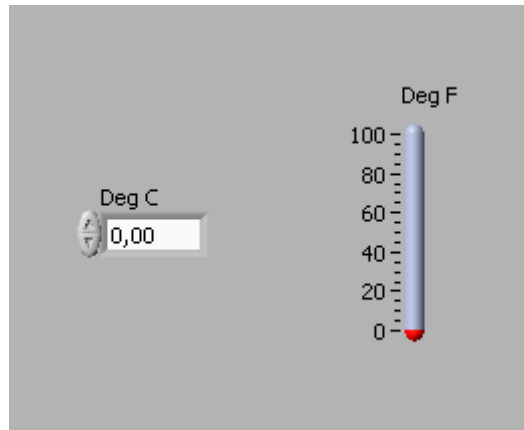
Η πύλη OR χρησιμοποιήθηκε με σκοπό να σταματά η εκτέλεση του προγράμματος όταν μία από της δύο συνθήκες που έχουμε συνδέσει γίνει αληθείς. Οι συνθήκες ελέγχου είναι α) να γίνει ο αριθμός 1 μεγαλύτερος του αριθμού δύο (2) και β) να πατήσει ο χρήστης το κουμπί τερματισμού του προγράμματος.




## 2.5 Σχεδιασμός εικονικού οργάνου για την μετατροπή βαθμών θερμοκρασίας από Celcius σε Farrenheit

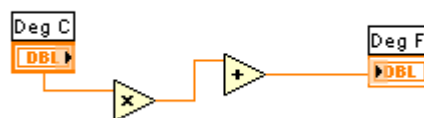
Σκοπός της παρούσας άσκησης, είναι η δημιουργία ενός εικονικού οργάνου, ικανού να μετατρέπει τους βαθμούς θερμοκρασίας C σε βαθμούς F. Για τον σκοπό αυτό αρχικά θα πρέπει να εισάγουμε στο front

panel ένα από την παλέτα numeric ένα digital control . Με την βοήθεια του labeling tool αλλάζουμε το όνομα του control από numeric σε Deg C. Αυτό θα είναι το control από το οποίο θα εισάγει

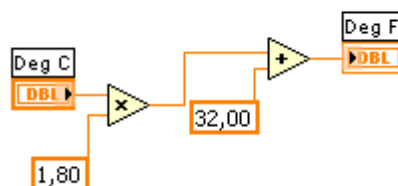
ο χρήστης τους βαθμούς C. Στην συνέχεια θα εισάγουμε από την παλέτα numeric  ένα digital indicator και πιο συγκεκριμένα αυτό που έχει την μορφή θερμομέτρου . Με το labeling tool πηγαίνουμε και αλλάζουμε το όνομα thermometer σε Deg F. Μέχρι αυτό το σημείο το VI που έχουμε σχεδιάσει θα πρέπει να έχει την παρακάτω μορφή.



Στο σημείο αυτό θα περάσουμε στο block diagram. Από το function palette/numeric  θα εισάγουμε έναν πολλαπλασιαστή  και έναν αθροιστή . Μετά την εισαγωγή των παραπάνω πραγματοποιούμε τις καλωδιώσεις που εμφανίζονται στο επόμενο σχήμα:




Με το wiring tool πηγαίνουμε στον πολλαπλασιαστή και στο ελεύθερο άκρο του επιλέγουμε create constant. Στην τιμή αυτή αναθέτουμε με το labeling tool την τιμή 1.80. Κατά τον ίδιο τρόπο με το wiring tool και εκτελώντας δεξί click στο ελεύθερο άκρο του, επιλέγουμε create constant, και στην σταθερά αυτή αναθέτουμε την τιμή 32.00. Μετά το πέρας των ενεργειών αυτών το πρόγραμμα μας είναι έτοιμο και έχει την παρακάτω μορφή:




Στο σημείο αυτό μπορούμε να εκτελέσουμε το VI που έχουμε σχεδιάσει. Αφού βάλουμε μια τιμή στο control Deg C και επιλέξουμε run τότε θα δούμε την αντίστοιχη τιμή σε βαθμούς F στο θερμομέτρο.

## 2.6 Σχεδιασμός εικονικού οργάνου παρακολούθησης θερμοκρασίας


Σκοπός της παρούσας εργασίας είναι να δημιουργήσουμε ένα VI το οποίο εικονικά (σε περιβάλλον εξομοίωσης) θα συλλέγει ένα σήμα ως να ήταν πραγματικό και εν συνεχεία θα το απεικονίζει σε ένα

γράφημα. Για το σκοπό αυτό θα εισάγουμε στο front panel από το μενού graph  αρχικά ένα

waveform chart . Με το labeling tool αλλάζουμε το όνομα waveform chart σε Temperature History.

Στη συνέχεια αλλάζουμε τον άξονα x του γραφήματος σε Time(sec) και τον άξονα y σε Temp (Deg F).

Επιπλέον εκτελούμε δεξί click πάνω στο γράφημα και επιλέγουμε Autoscale για τον άξονα y. Στο block

diagram επιλέγουμε το functions . Μετά την επιλογή μας κάνουμε navigate στον φάκελο

c:\Program Files\National instruments\Labview 6.1\Activity και επιλέγουμε το subvi Digital

Thermometer. Η επιλογή μας αυτή έχει σαν αποτέλεσμα να εισάγουμε στο block diagram μία ρουτίνα

subvi που παράγει τιμές θερμοκρασίας ως να τις δειγματοληπούσαμε σε ένα πραγματικό σύστημα. Εάν

κάνουμε διπλό click πάνω σε αυτό το subvi μπορούμε να παρατηρήσουμε το front panel και το block

diagram της ρουτίνας αυτής. Στην συνέχεια με το wiring tool ενώνουμε την έξοδο του Digital

Thermometer με το Waveform Chart. Επίσης περικλείουμε τον όλον προγραμματισμό μας με έναν βρόχο

While με σκοπό να εκτελείται το πρόγραμμα μας συνεχώς μέχρι να επιλέξουμε εμείς τον τερματισμό του.

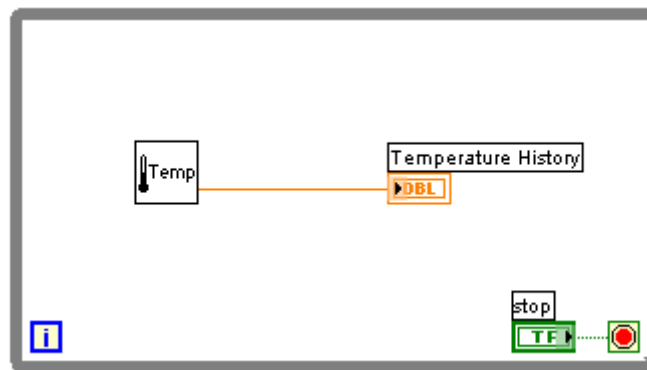
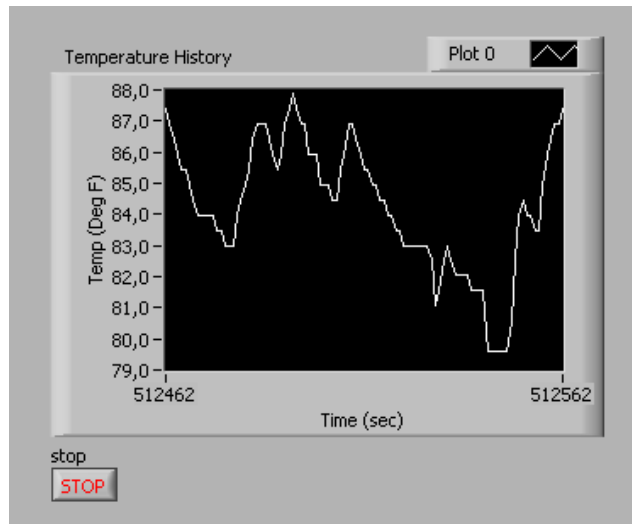
Στη συνέχεια στο  του while loop, εκτελούμε δεξί click και επιλέγουμε create control. Αυτόματα

δημιουργείται ένα Boolean control button (στο front panel) με αρχική με τιμή false και ένδειξη stop.

Κάνουμε ξανά δεξί Click πάνω στο  και επιλέγουμε Stop if true. Στο Front panel, πατάμε δεξί click

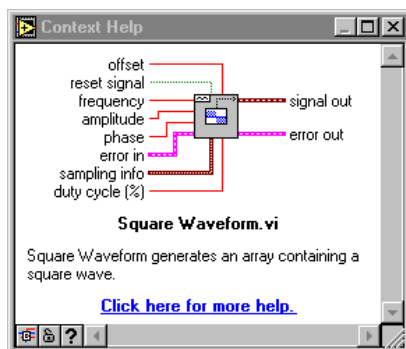
πάνω στο button και επιλέγουμε Mechanical Action/latch when released. Επίσης με το Labeling tool, του

δίνουμε την ονομασία Stop. Στο σημείο αυτό το πρόγραμμα μας θα έχει την παρακάτω μορφή:



## 2.7 Σχεδιασμός εικονικής γεννήτριας συχνοτήτων

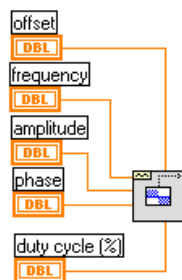
Αφού ξεκινήσουμε το Labview, επιλέγουμε από το menu/window/Tile left and right(ctrl+T). Η επιλογή αυτή μας τοποθετεί το Front panel και το Block diagram σε δύο παράθυρα που καλύπτουν όλη την οθόνη και μπορούν να τα βλέπουμε ταυτόχρονα. Στο Block diagram επιλέγουμε από Function Palette\ Analyze\





Waveform Generation  \ Square Generation . Στο σημείο αυτό είναι πολύ χρήσιμο να εμφανίσουμε την επιπρόσθετη βοήθεια του Labview (ctrl+H), από όπου μπορούμε να πάρουμε

πληροφορίες για τον τρόπο που πρέπει να γίνουν οι διάφορες διασυνδέσεις με το όργανο που παράγει την κυματομορφή.

Στη συνέχεια και για τις συνδέσεις που φαίνονται στο επόμενο σχήμα, πηγαίνουμε με το wiring tool και εκτελούμε δεξί click, create control. Το σήμα signal out, είναι τύπου waveform. Τα σήματα αυτά περιλαμβάνουν τρία είδη πληροφορίας. Το πρώτο αναφέρεται στον χρόνο  $t$ , το δεύτερο στον χρονικό διάστημα  $dt$  που ακολουθεί το επόμενο δείγμα της εξόδου και στο τρίτο πεδίο, όπου και υπάρχουν οι αντίστοιχες (με τις χρονικές στιγμές του πρώτου πεδίου) τιμές της εξόδου.



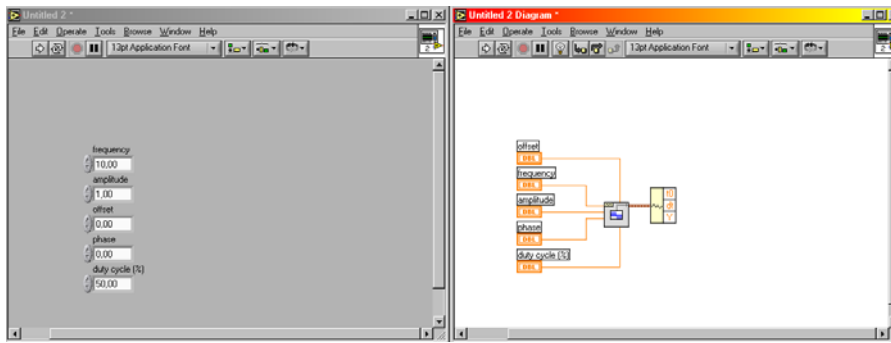
Η εικονική γεννήτρια παλμών που θα δημιουργήσουμε, παράγει τα σήματα, όπως κάνει και μια κανονική παλμογεννήτρια. Η μόνη διαφορά έγκειται στο γεγονός ότι στην μεν πρώτη περίπτωση, οι παλμοί παράγονται έμμεσα από software, ενώ στην δεύτερη περίπτωση, τους παλμούς τους παράγουν άμεσα ηλεκτρονικά κυκλώματα. Με σκοπό να οδηγήσουμε του παλμούς αυτούς σε κάποια διάταξη, θα πρέπει από το signal out να πάρουμε μόνο την πληροφορία που σχετίζεται με τα επίπεδα τάσεως του παλμού και όχι στις χρονικές στιγμές που συμβαίνουν αυτά. Για το σκοπό αυτό, συνδέουμε στο signal out το Get



waveform components  το οποίο βρίσκεται κάτω από το Functions\waveforms . Με την





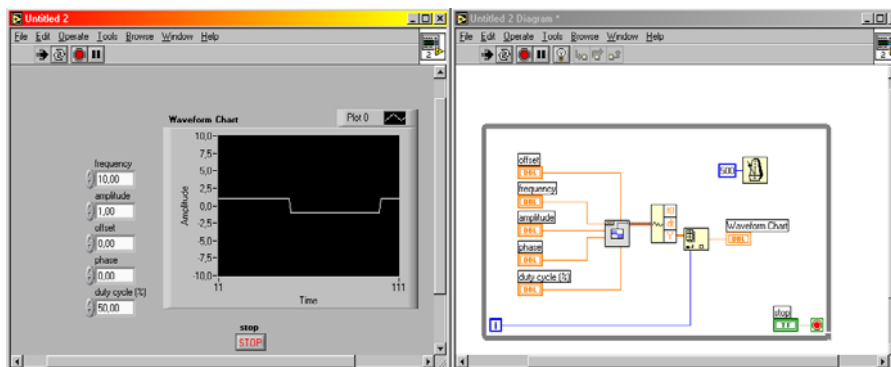
βοήθεια του positioning tool, αυξάνουμε το μέγεθος (σαν παράθυρο Windows) του Get waveform components, όπως φαίνεται και στο παρακάτω σχήμα. Από την ενέργεια μας αυτή, είναι τώρα εμφανή τα στοιχεία που συνθέτουν ένα σήμα Waveform, όπως είναι το signal out. Το  $t_0$  είναι ο αρχικός χρόνος της κυματομορφής, το  $dt$  αποτελεί τον χρόνο μέχρι να εμφανιστεί το επόμενο δείγμα από την κυματομορφή, ενώ το  $Y$  είναι ένα διάνυσμα που περιέχει όλες τις τιμές της κυματομορφής.







Στο Front panel, εισάγουμε από το μενού Controls\Graphs  ένα Waveform chart . Το Y που εμφανίζεται είναι ένα διάνυσμα (πίνακας μεγέθους  $[n, 1]$ ) με όλες τις τιμές του παλμού σε μια περίοδο. Με σκοπό να πάρουμε τις τιμές αυτές από την πρώτη μέχρι και την τελευταία θα πρέπει από το

Functions/array  να επιλέξουμε index array . Με το wiring tool, στην είσοδο του index array/n-dimension array συνδέουμε τον πίνακα Y. Στην έξοδο αυτού συνδέουμε το Waveform chart που εισάγαμε. Όλο το πρόγραμμα μας τον περικλείουμε με ένα while loop και το index του While loop, το συνδέουμε με το index του index array. Με αυτόν τον τρόπο κάθε φορά που θα εκτελείται το while loop θα παίρνουμε και το επόμενο δείγμα από το διάνυσμα με την παλμοσειρά. Επιπλέον δημιουργούμε έναν έλεγχο για την εκτέλεση του loop όπως έχουμε κάνει και στα προηγούμενα παραδείγματα.

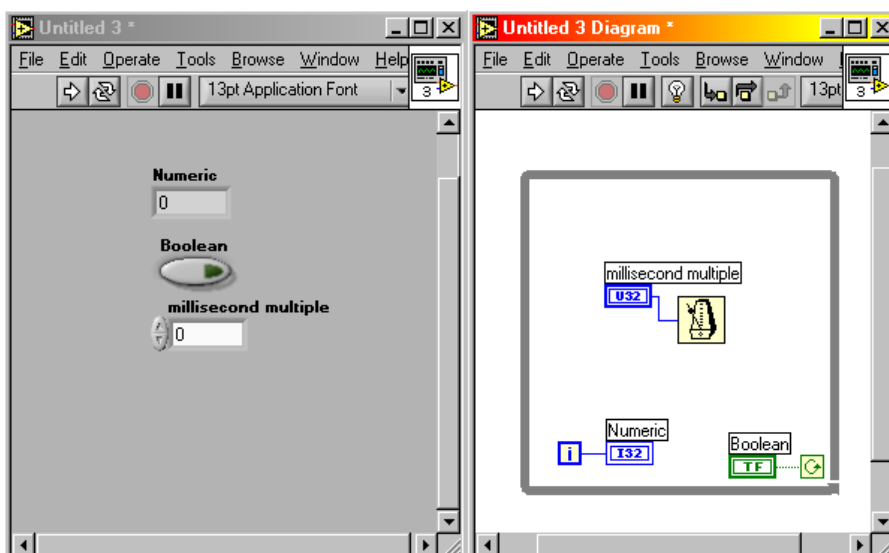


Τέλος εισάγουμε από το Functions\Time&Dialog  το vi wait until next multiple , με σκοπό να ρυθμίσουμε την ταχύτητα με την οποία παράγεται η παλμοσειρά. Με αυτόν τον τρόπο, μπορούμε να βλέπουμε τις τιμές και το είδος του παλμού, με τον οποίο μπορούμε να τροφοδοτήσουμε κάποια συσκευή ή διάταξη.

Στο σημείο αυτό οι φοιτητές, θα πρέπει να πειραματιστούν με την δημιουργία κυματομορφών διαφορετικού τύπου, καθώς και να εξετάσουν την λειτουργικότητα των κουμπιών ελέγχου που έχουν τοποθετηθεί στο Front panel καθώς και τις αλλαγές στην μορφή της προκύπτουσας κυματομορφής.

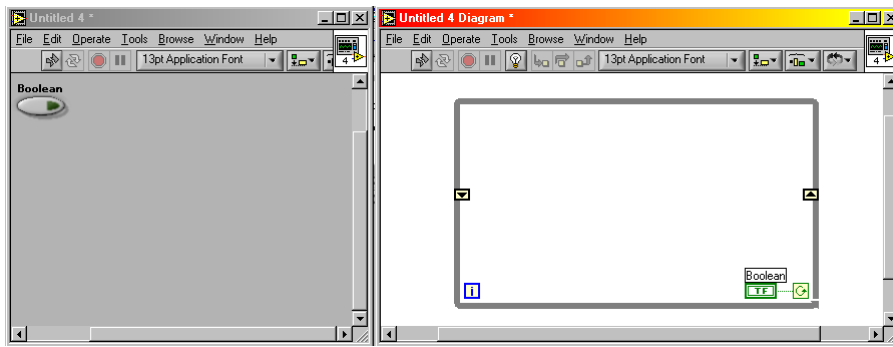
## 2.8 Συγχρονισμός εκτέλεσης των VIs



Με σκοπό ο κώδικας που δημιουργούμε, να μην εκτελείται με την μέγιστη ταχύτητα αλλά με την επιθυμητή που θα την καθορίζει ο χρήστης, θα πρέπει να κάνουμε χρήση του VI Wait Until Next ms multiple, το οποίο μπορούμε να το βρούμε στην Function παλέτα\Time & Dialog. Σε οποιοδήποτε Βρόχο εκτέλεσης του προγράμματος, μπορούμε να το εισάγουμε και να επιλέξουμε, το χρονικό διάστημα καθυστέρησης. Για λόγους επίδειξης, θα χρησιμοποιήσουμε ένα απλό βρόχο While και θα δημιουργήσουμε έναν Indicator του αριθμού των επαναλήψεων του loop. Με το wiring tool, πηγαίνουμε πάνω στο Wait Until Next ms Multiple και επιλέγουμε την δημιουργία ενός control για τον έλεγχο της χρονοκαθυστέρησης. Επίσης δημιουργούμε άλλο ένα control για τον έλεγχο της έναρξης και παύσης λειτουργίας του προγράμματος μας. Μετά την ολοκλήρωση του προγραμματισμού μας, από το front panel και αλλάζοντας το μέγεθος της χρονοκαθυστέρησης, μπορούμε να δούμε την μεταβολή της ταχύτητας του προγράμματος μας.

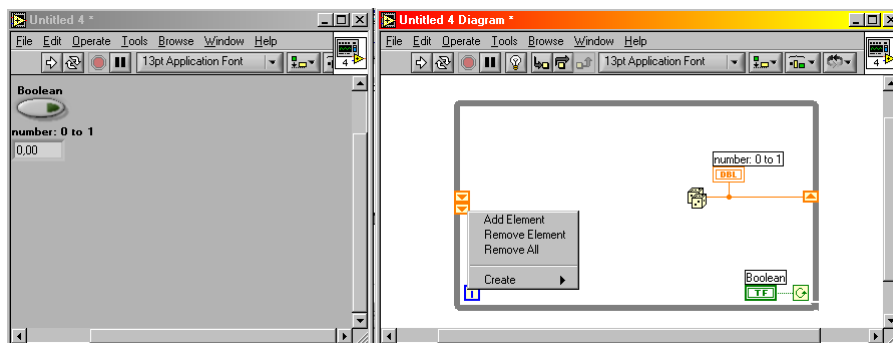


## 2.9 Χρήση Shift Registers

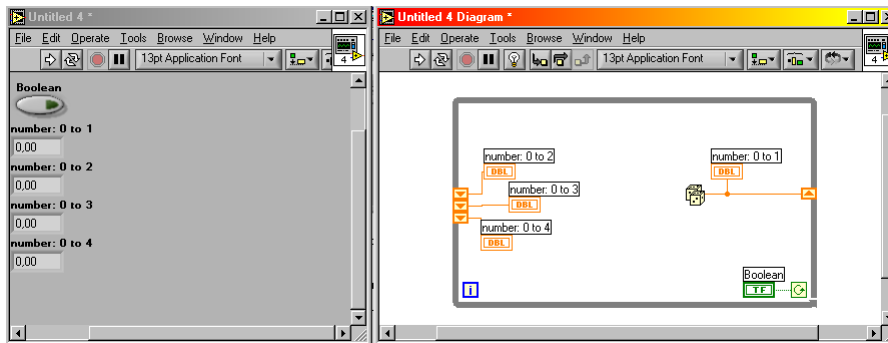
Ο βασικός σκοπός της παρούσας ασκήσεως, είναι η εξοικείωση του χρήστη με την χρήση μνήμης στο περιβάλλον προγραμματισμού του Labview. Σε αυτό το περιβάλλον η μνήμη ορίζεται σαν ένας register. Ανάλογα με τις ανάγκες της εφαρμογής, ο χρήστης μπορεί να χρησιμοποιήσει πολλαπλούς registers. Τους shift registers μπορούμε να τους εισάγουμε στον προγραμματισμό μας μόνο με την ταυτόχρονη χρήση κάποιου λογικού βρόχου (while/for etc.). Η εισαγωγή γίνεται πηγαίνοντας στην άκρη του βρόχου και εκτελώντας δεξί click\ insert shift register. Για την περίπτωση του while loop, και εκτελώντας την παραπάνω διαδικασία θα έχουμε τον shift register που εμφανίζεται στο παρακάτω σχήμα.



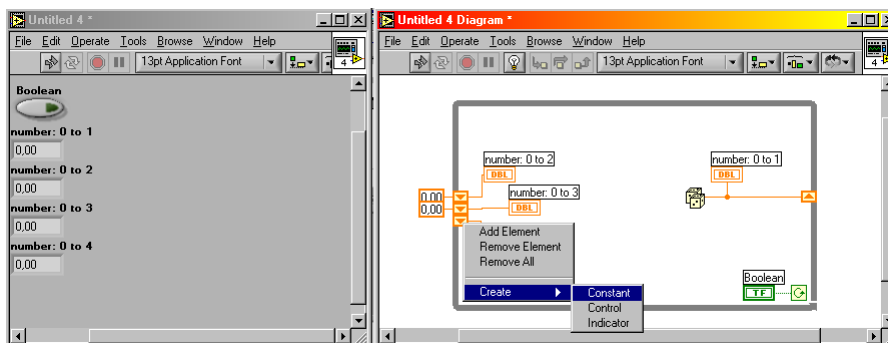
Στο σημείο αυτό θα εισάγουμε και μια γεννήτρια τυχαίων αριθμών. Ο τελικός σκοπός του προγράμματος που αναπτύσσουμε, είναι να μπορούμε να παρακολουθούμε στο front panel, την τωρινή και τις τρεις περασμένες τιμές από την γεννήτρια τυχαίων αριθμών. Κατά την λειτουργία του shift register, η τιμή που περνάμε στο δεξιό του τμήμα , θα βρίσκεται στην διάθεση μας, την επόμενη φορά που θα εκτελεστεί ο βρόχος, από το δεξιό τμήμα του register  αυτή την φορά. Στη συνέχεια συνδέουμε την έξοδο της γεννήτριας τυχαίων αριθμών (ζάρια) στο δεξιό τμήμα του register και δημιουργούμε έναν indicator για αυτήν την έξοδο της γεννήτριας τυχαίων αριθμών. Με σκοπό να βλέπουμε μέχρι και τις τρεις προηγούμενες τιμές από την γεννήτρια τυχαίων αριθμών, στο δεξιό τμήμα του register και με το wiring tool εκτελούμε δεξί click και επιλέγουμε add element. Την διαδικασία αυτή την επαναλαμβάνουμε τρεις φορές, και με αυτόν τον τρόπο, έχουμε δημιουργήσει την μνήμη που θέλαμε. Θα πρέπει να έχουμε πάντα υπόψη μας ότι η μετακίνηση της πληροφορίας σε αυτήν την μνήμη, γίνεται με σειριακό τρόπο και μπορεί να χρησιμοποιηθεί για όλους τους τύπους μεταβλητών που χρησιμοποιεί το Labview.



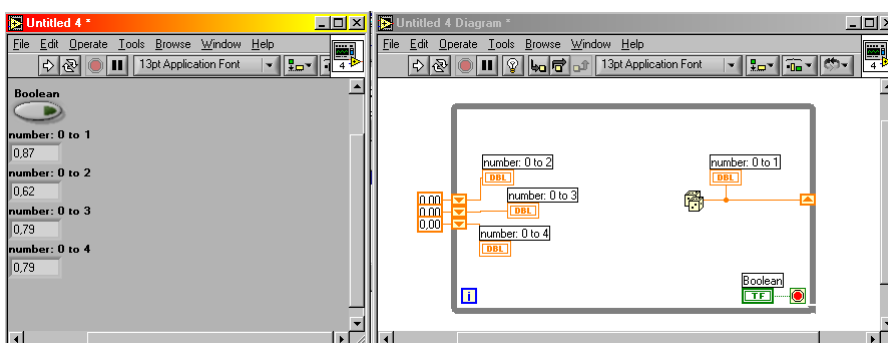
Σε κάθε ένα βελάκι στο δεξιό τμήμα του register, δημιουργούμε και από έναν indicator.



Με το wiring tool και εκτελώντας πάλι δεξί click πάνω στο δεξιό τμήμα του shift register, επιλέγουμε create constant και αναθέτουμε στην σταθερά την τιμή 0. Την διαδικασία αυτή την εκτελούμε τρεις φορές, για κάθε βέλος του register, και με τον τρόπο αυτό μπορούμε να αρχικοποιήσουμε την μνήμη στις τιμές που εμείς επιθυμούμε. Στην περίπτωση μας, η αρχικοποίηση έγινε στην τιμή 0.



Για την ολοκλήρωση του προγράμματος, εισάγουμε μια χρονοκαθυστέρηση 100ms, όπως περιγράφηκε στην ενότητα 2.5 και επιπλέον επιλέγουμε με δεξί click να σταματά η εκτέλεση του while loop μόνο όταν η τιμή του Boolean control είναι true.

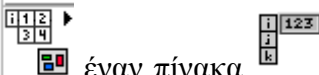



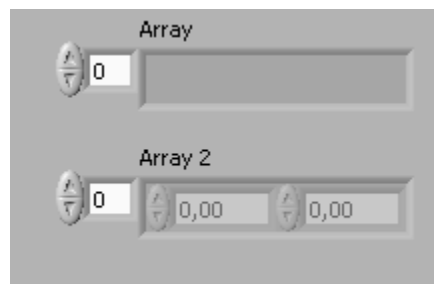
Στο σημείο αυτό είμαστε έτοιμοι να εκτελέσουμε το VI και να παρατηρήσουμε πως οι τιμές της γεννήτριας τυχαίων αριθμών μεταβάλλονται σειριακά. Σε κάθε χρονική στιγμή μπορούμε να παρατηρούμε την τωρινή και τις τρεις προηγούμενες τιμές της γεννήτριας τυχαίων αριθμών.

## 2.10 Δημιουργία πινάκων με την χρήση loops

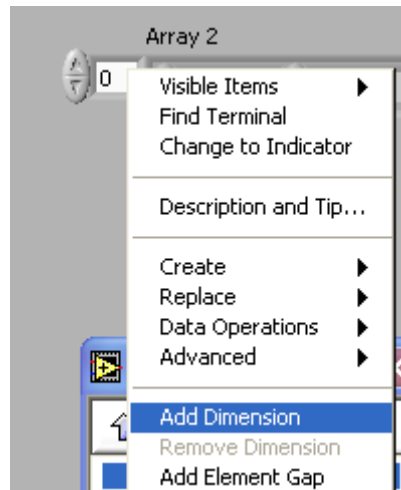
Σκοπός της παρούσης ασκήσεως, είναι η δημιουργία πινάκων με την χρήση loops. Ένας πίνακας περιέχει στοιχεία του ίδιου τύπου. Ο πίνακας αποτελείται από στοιχεία και διαστάσεις. Τα στοιχεία είναι τα δεδομένα που περιέχονται στον πίνακα. Η διάσταση είναι το μήκος, ύψος ή το βάθος του πίνακα, ανάλογα με την χρήση του. Ο πίνακας μπορεί να έχει περισσότερες διαστάσεις από μία. Κάθε διάσταση μπορεί να περιέχει μέχρι  $2^{31}-1$ , εφόσον αυτό επιτρέπεται από την μνήμη. Τους πίνακες τους χρησιμοποιούμε όταν δουλεύουμε με σετ από όμοια δεδομένα και όταν εκτελούμε επαναληπτικούς υπολογισμούς. Οι πίνακες είναι ιδανικοί όταν τους χρησιμοποιούμε για την αποθήκευση δεδομένων που συλλέγονται από κυματομορφές ή δεδομένα που παράγονται μέσα από loops. Θα πρέπει να θυμόμαστε ότι δεν μπορούμε να δημιουργήσουμε πίνακες που να περιέχουν ως στοιχεία άλλους πίνακες.

Με σκοπό την δημιουργία ενός πίνακα control ή indicator επιλέγουμε από το front panel\Array &



Cluster  έναν πίνακα  . Στην συνέχεια με μία διαδικασία drag and drop εισάγουμε μέσα στον πίνακα ένα control ή ένα indicator. Στην περίπτωση που προσπαθήσουμε να σύρουμε ένα αντικείμενο που δεν είναι δυνατόν να περιέχεται μέσα στον πίνακα, η δυνατότητα ενσωμάτωσης του αντικειμένου μέσα στον πίνακα δεν είναι δυνατή. Προσοχή θα πρέπει να δοθεί στο γεγονός ότι θα πρέπει να σύρουμε έναν αντικείμενο στο πλαίσιο του πίνακα πριν αρχίσουμε να τον χρησιμοποιούμε στο δομικό διάγραμμα. Στην αντίθετη περίπτωση το αντικείμενο του πίνακα στο δομικό διάγραμμα εμφανίζεται μαυρισμένο με ένα άδειο πλαίσιο.

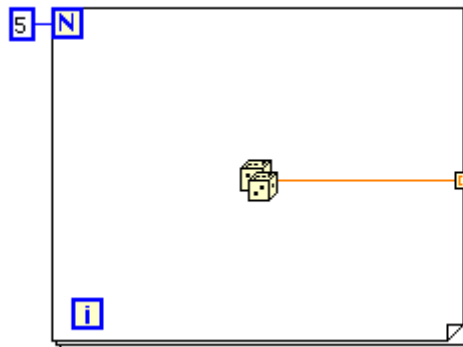



Οι δυδιάστατοι πίνακες αποθηκεύουν τιμές σε ένα πλέγμα. Για τον σκοπό αυτό απαιτούνται δύο τιμές, που περιγράφουν τον αριθμό των σειρών και των στηλών. Οι δείκτες στους πίνακες ξεκινούν από την τιμή 0. Για να προσθέσουμε μια διάσταση σε ένα πίνακα, εκτελούμε δεξί click πάνω στον πίνακα και επιλέγουμε Add Dimension

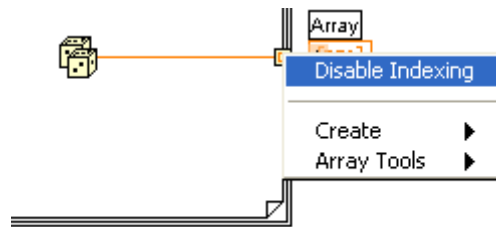


Στην συνέχεια θα δείξουμε την μέθοδο κατά την οποία μπορούμε να αποδώσουμε σε έναν πίνακα τιμές αυτόματα με την χρήση ενός loop. Για τον σκοπό αυτό θα χρησιμοποιήσουμε από την παλέτα με τα

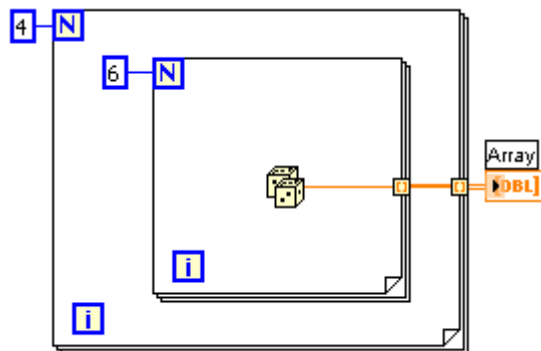
function\structures  ένα for loop . Στην τιμή  $N$  αποδίδουμε την τιμή 5. Μέσα στο for loop εισάγουμε μία γεννήτρια τυχαίων αριθμών. Την έξοδο από την γεννήτρια την ενώνουμε στο πλαίσιο του for loop.



Με αυτόν τον τρόπο έχουμε επιλέξει την δυνατότητα auto-indexing σύμφωνα με την οποία το for loop εκτελείται και κάθε νέα τιμή που παράγεται από την γεννήτρια τυχαίων αριθμών εισέρχεται μέσα στον πίνακα. Για να δούμε τον πίνακα θα πρέπει να πάμε στο λεγόμενο input tunnel  να εκτελέσουμε δεξί click και να δημιουργήσουμε ένα indicator. Αυτόματα θα δημιουργηθεί και ο πίνακας που θα περιέχει τις παραγόμενες τιμές. Για την περίπτωση που δεν θέλουμε η έξοδος να είναι πίνακας αλλά να βγάζει το loop την τελευταία τιμή της εκτέλεσης του loop θα πρέπει το input tunnel να εκτελέσουμε δεξί click και να επιλέξουμε disable indexing. Θα πρέπει να παρατηρήσουμε πως η διάσταση του πίνακα είναι ίση με τον αριθμό των επαναλήψεων του for loop. Στην περίπτωση μας ο αριθμός αυτός είναι ίσος με 5.

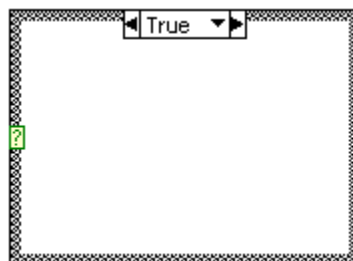


Για την δημιουργία δισδιάστατων πινάκων μπορούμε να χρησιμοποιήσουμε δύο for loops, το ένα μέσα στο άλλο. Το εξωτερικό loop δημιουργεί τα στοιχεία των γραμμών, ενώ το εσωτερικό δημιουργεί τα στοιχεία των στηλών. Στο παρακάτω σχήμα χρησιμοποιήσαμε 2 for loops για να δημιουργήσουμε έναν δισδιάστατο πίνακα διαστάσεων 4x5.



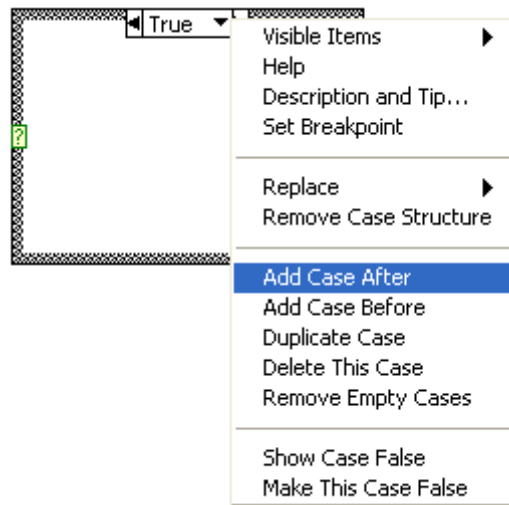
## 2.11 Χρήση λογικών δομών CASE

Μία δομή CASE έχει δύο ή περισσότερα subdiagrams ή cases. Σε κάθε στιγμή μόνο ένα από αυτά είναι ορατό αλλά και ενεργό, δηλαδή πραγματοποιείται ο κώδικας μέσα σε αυτό.

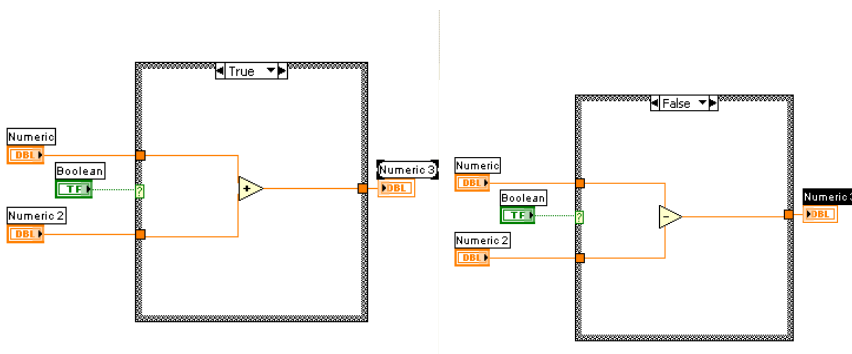


Η τιμή μιας μεταβλητής εισόδου καθορίζει ποίο από όλα τα subdiagrams εκτελείται κάθε φορά. Η δομή case είναι παρόμοια με το case statement «if...then...else» στις τυπικές γλώσσες βασισμένες σε κείμενο. Πάνω στην δομή του case βρίσκεται η ένδειξη του case όπου βρισκόμαστε . Μπορούμε να χρησιμοποιήσουμε τα πάνω και κάτω βέλη για να περιηγηθούμε ανάμεσα στα διάφορα προγραμματισμένα cases. Για να ελέγξουμε το case που θα εκτελεστεί θα πρέπει να συνδέσουμε στην είσοδο (case selector)  μια τιμή, η οποία θα δείχνει ποιο case θα εκτελεστεί. Οι τύποι που μπορούμε να συνδέσουμε στο case selector είναι Boolean, string ή επιλεκτικού τύπου. Ο μέγιστος αριθμός cases για την περίπτωση που έχουμε συνδέσει έναν integer είναι  $2^{31}-1$ . Για να έχουμε δυνατότητα αλλαγής στα

cases, δηλαδή να προσθέσουμε, να αφαιρέσουμε, να αντιγράψουμε κλπ αρκεί να εκτελέσουμε δεξί click πάνω στο case selector.

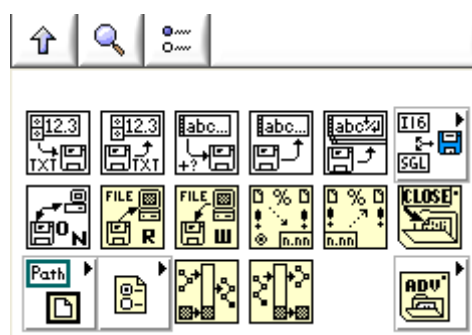


Στο επόμενο σχήμα παρουσιάζουμε μία δομή case κατά την οποία ο χρήστης έχει την δυνατότητα να επιλέξει μεταξύ της αφαίρεσης και της πρόσθεσης μεταξύ δύο αριθμών. Ο επιλογέας, επειδή το case έχει δύο καταστάσεις, είναι τύπου case.



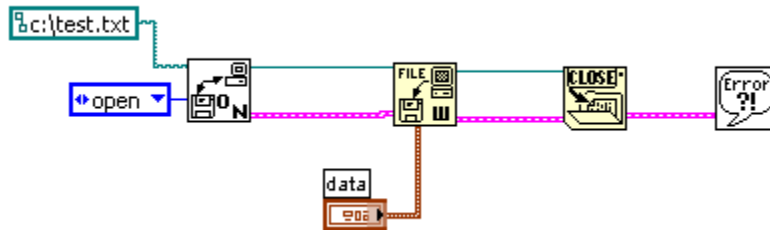
## 2.12 Εγγραφή - ανάγνωση δεδομένων σε / από αρχεία





Σκοπός της άσκησης αυτής είναι ο σχεδιασμός εικονικών οργάνων που θα αποθηκεύουν και θα ανακτούν δεδομένα από αρχεία. Για τον σκοπό αυτό υπάρχουν δύο κατηγορίες από συναρτήσεις: α) οι συναρτήσεις υψηλού επιπέδου και β) συναρτήσεις χαμηλού επιπέδου.

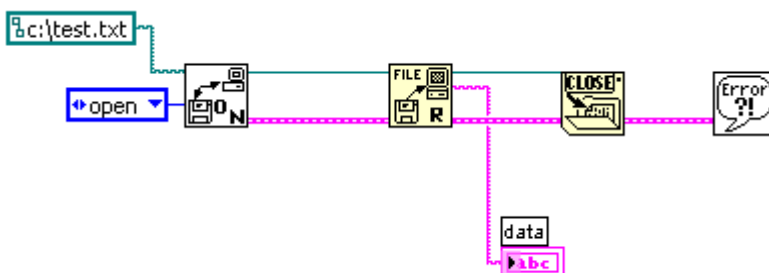




Στη συνέχεια θα παρουσιάσουμε τον τρόπο σχεδιασμού εγγραφής και ανάγνωσης αρχείων με την χρήση συναρτήσεων του δεύτερου επιπέδου για πιο αποδοτικά αποτελέσματα. Ο τρόπος εγγραφής παρουσιάζεται στο παρακάτω σχήμα.



Το VI  χρησιμοποιείται για να καθορίσουμε το path και το όνομα του αρχείου που θα δημιουργήσουμε. Επίσης μπορούμε να καθορίσουμε αν θα δημιουργηθεί άλλο αρχείο, αν η εγγραφή θα συνεχιστεί από το ήδη υπάρχον αρχείο, σε ποιο σημείο κλπ. Το VI  χρησιμοποιείται για να περάσουν τα επιθυμητά δεδομένα στο αρχείο που θέλουμε, ενώ το VI  χρησιμοποιείται για να κλείσουμε το αρχείο που ανοίξαμε και περάσαμε τις τιμές σε αυτό. Τέλος το VI  χρησιμοποιείται για την εμφάνιση τυχόν λαθών σε κάποιο από τα στάδια της εγγραφής. Στην περίπτωση που θα θέλαμε να διαβάσουμε το αρχείο που μόλις δημιουργήσαμε θα έπρεπε να χρησιμοποιήσουμε τον συνδυασμό από VIs που εμφανίζονται στο επόμενο σχήμα:






## 2.13 Αναλογική Είσοδος / Έξοδος




Ένα από τα πιο βασικά πλεονεκτήματα του Labview, είναι η ευκολία που μας παρέχει στην μέτρηση και άσκηση πραγματικών σημάτων (Analog Input-Analog Output). Η δυνατότητα αυτή μπορεί να


πραγματοποιηθεί σε τρία επίπεδα: α) με την χρήση απλών VIs όπου όλες σχεδόν οι λειτουργίες είναι από πριν προγραμματισμένες, β) με την χρήση πιο πολύπλοκων VIs όπου χρειάζεται ένας μικρός προγραμματισμός από τον χρήστη και γ) με την χρήση πολύ πολύπλοκων VIs όπου και μπορούμε να επιτύχουμε βέλτιστους ρυθμούς δειγματοληψίας και απόδοσης. Στην τελευταία περίπτωση απαιτείται και μια ιδιαίτερη γνώση του χρήστη για τον τρόπο που λειτουργεί το hardware, χρονισμούς, AD/DA μετατροπή κλπ. Στην παρούσα παράγραφο θα γίνει η περιγραφή της πρώτης περίπτωσης, των απλών VIs που θα χρησιμοποιηθούν στους ελεγκτές που θα σχεδιάσουμε στα επόμενα κεφάλαια.


Το παράδειγμα που θα δώσουμε είναι αυτό κατά το οποίο θα δημιουργήσουμε ένα σήμα ελέγχου, δηλαδή μια τάση θα την στείλουμε σε ένα συγκεκριμένο κανάλι της κάρτας και στην συνέχεια με θα οδηγήσουμε με καλώδιο το σήμα αυτό σε ένα κανάλι εισόδου της ίδιας κάρτας, από όπου θα πάμε στη συνέχεια και θα το δειγματοληψίσουμε. Κατά την εκτέλεση της ασκήσεως αυτής, οι φοιτητές δεν θα πραγματοποιήσουν καλωδιώσεις αλλά θα χρησιμοποιήσουν τις ήδη υπάρχουσες.

Αρχικά θα πρέπει να σχεδιάσουμε ένα βρόχο while όπου θα στέλνεται το σήμα ελέγχου συνεχώς στην έξοδο. Ο ρυθμός με τον οποίο θα εκτελείται ο βρόχος δεν έχει μεγάλη σημασία. Στο παρόν παράδειγμα

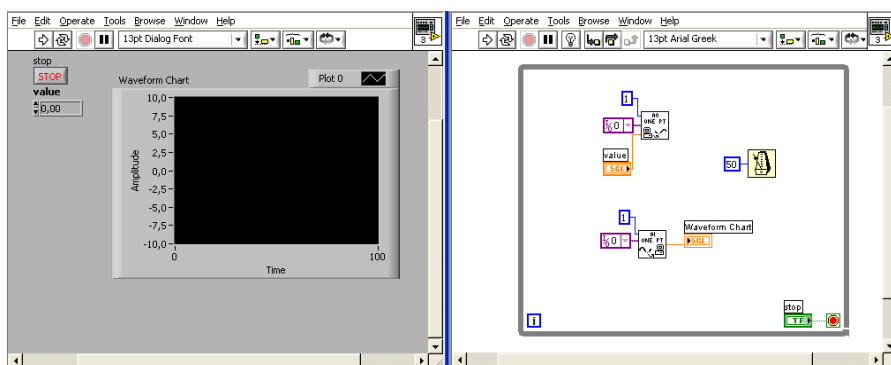
θα επιλέξουμε την τιμή 50 msec. Από την παλέτα Data acquisition  επιλέγουμε το Analog Input  και εν συνεχεία επιλέγουμε το AI sample channel . Αυτό το VI εκτελεί δειγματοληψία σε ένα

μόνο κανάλι εισόδου. Εκτελώντας την ίδια διαδικασία από την παλέτα Data acquisition  επιλέγουμε το Analog Output  και εν συνεχεία επιλέγουμε το AO sample channel . Το VI αυτό εκτελεί την μετατροπή ενός ψηφιακού σήματος σε αναλογικό και το προωθεί σε ένα μόνο κανάλι εξόδου.

Αρχικά στο AO sample channel  με δεξί click θα επιλέξουμε να δημιουργήσουμε ένα constant για το Device. Το device αναφέρεται στην κάρτα που θα χρησιμοποιήσουμε. Για μία κάρτα επιλέγουμε την τιμή 1. Στην συνέχεια πάλι με δεξί click θα πρέπει να αποδώσουμε μια τιμή στο channel που είναι και το κανάλι όπου θα οδηγηθεί το σήμα ελέγχου. Ας επιλέξουμε την τιμή 0. Τέλος στο τελευταίο connector θα δημιουργήσουμε ένα control μέσω του οποίου θα επιλέγουμε την τάση που θέλουμε να στείλουμε στην έξοδο του συστήματος μας.

Για το AI sample channel  θα ακολουθήσουμε την ίδια διαδικασία και θα επιλέξουμε και εδώ για το device την τιμή 1. Για το κανάλι θα επιλέξουμε την τιμή 0. Με αυτή την επιλογή το καλώδιο στο connector block θα πρέπει να συνδέει το κανάλι 0 της εξόδου με το κανάλι 0 της εισόδου. Σε περίπτωση

που έχει πραγματοποιηθεί μια άλλη συνδεσμολογία θα πρέπει να ληφθεί μέριμνα για τον τρόπο επιλογής των καναλιών που θα χρησιμοποιήσουμε, ώστε να παίρνουμε σωστές μετρήσεις. Στο τέλος του προγραμματισμού το VI θα πρέπει να έχει την παρακάτω μορφή:



Κατά την λειτουργία του μπορούμε να επιλέγουμε μία τάση από το control στο front panel να την παράγουμε σε κάποιο κανάλι της κάρτας και εν συνεχεία οδηγώντας το σήμα αυτό σε μια είσοδο της κάρτας να το ψηφικοποιούμε και να το αναπαριστούμε στο waveform chart.