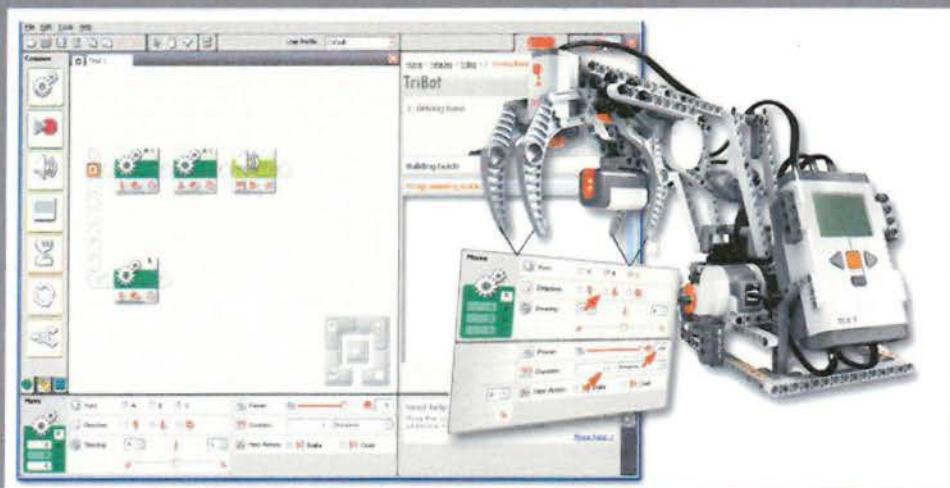


ΗΥ/Σ  
114



## ΤΕΙ ΠΕΙΡΑΙΑ ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ  
ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ  
ΣΤΑ Σ.Α.Ε.



ΒΙΒΛΙΟΘΗΚΗ  
ΤΕΙ ΠΕΙΡΑΙΑ

ΚΑΝΑΤΑ ΠΗΝΕΛΟΠΗ 34063  
ΠΟΥΚΑ ΕΛΕΟΝΩΡΑ 36176  
ΥΠΕΥΘΥΝΗ ΚΑΘΗΓΗΤΡΙΑ Α.ΒΕΛΩΝΗ



## **ΠΕΡΙΛΗΨΗ**

Στην παρούσα πτυχιακή εργασία γίνεται μελέτη και περιγραφή του προγράμματος LabView της εταιρίας National Instruments. Στόχος αυτής της εργασίας είναι να εισάγει σε πρώτο βαθμό τον αναγνώστη- χρήστη στο προγραμματιστικό περιβάλλον της γραφικής γλώσσας 'G' που χρησιμοποιεί το LabView (από την έκδοση LabView 7 και μετά).

Σε κάθε κεφάλαιο παρουσιάζονται αναλυτικά τα βασικά χαρακτηριστικά του περιβάλλοντος LabView και στη συνέχεια ακολουθούν παραδείγματα. Αξίζει να σημειωθεί ότι έχει γίνει λεπτομερής ανάλυση της διαδικασίας δημιουργίας των απαραίτητων προγραμμάτων, ώστε να μπορεί ο αναγνώστης να κατανοήσει και να αξιοποιήσει πλήρως τη σχεδίαση που υλοποιήθηκε.

## ΠΕΡΙΕΧΟΜΕΝΑ

1.1 Εισαγωγή στο Labview.....	6
1.2 Εικονόγραφο (Virtual Instrument).....	6
1.3 Ξεκίνημα του Labview.....	12
1.4 Τα παράθυρα Front Panel και Block Diagram.....	13
1.4.1 Εργαλεία του Front Panel.....	16
1.4.2 Εργαλεία του Block Diagram.....	17
1.4.3 Μενού Pop-Up.....	18
1.4.4 Μενού Pull- Down.....	23
1.4.5 Παράθυρο Γρήγορης Βοήθειας.....	30
1.5 Ιδιότητες των VIs.....	31
1.6 Εξερεύνηση Παλετών.....	37
1.6.1 Παλέτα εργαλείων (Tools).....	38
1.6.2 Παλέτες Αντικειμένων (Controls) και Λειτουργιών (Functions).....	40
1.6.2.1 Παλέτα Αντικειμένων (Controls).....	41
1.6.2.2 Παλέτα Λειτουργιών (Functions).....	43
1.6.3 Μορφοποίηση Παλετών.....	51
1.7 Ανοίγοντας VIs.....	53
1.8 Front Panel και Block Diagram.....	54
1.8.1 Front Panel.....	54
1.8.1.1 Επεξεργασία Αντικειμένων.....	53
1.8.2 Block Diagram.....	60
1.8.2.1 Δημιουργία – Διαγραφή Συνδέσεων.....	62
1.9 Προγραμματισμός με το Labview.....	63
1.9.1 Χρήση των SubVIs.....	64
1.9.2.1 Λειτουργία Κονέκτορα.....	71
2.1 Δομές.....	74
2.1.1 Η Δομή While Loop και η λειτουργία της.....	75
2.1.2 Η Δομή For Loop και η λειτουργία της.....	80
2.1.3 Η Δομή Case και η λειτουργία της.....	81
2.1.4 Η Δομή Sequence και η λειτουργία της.....	83
2.1.5 Η Δομή Event και η λειτουργία της.....	86
2.1.6 Global.....	88
2.1.7 Local.....	91
2.1.8 Formula Node.....	92
2.2 Shift Registers.....	92
3.1 Πίνακες.....	95
3.1.1 Δι- διάστατοι Πίνακες.....	97
3.1.2 Επεξεργασία Πινάκων.....	99

# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

3.1.3 Πολυμορφισμός.....	105
3.2 Συστάδες (Clusters).....	105
3.2.1 Bundle.....	106
3.2.2 Unbundle.....	106
3.2.3 Λειτουργίες Συστάδων (Clusters).....	107
4. Προγραμματιστικές Λειτουργίες.....	110
4.1 Χαρακτήρες (Strings).....	110
4.1.1 Τύποι Χαρακτήρων.....	107
4.1.2 Λειτουργίες Χαρακτήρων.....	112
4.2 Αριθμητικές Λειτουργίες.....	116
4.2.1 Μετατροπή Συστημάτων Αριθμών.....	120
4.2.2 Τριγωνομετρικές Συναρτήσεις.....	121
4.2.3 Λογαριθμικές Συναρτήσεις.....	124
4.2.4 Μιγαδικές Συναρτήσεις.....	126
4.2.5 Αριθμητικές Σταθερές.....	127
4.2.6 Λειτουργίες Σύγκρισης.....	128
4.2.7 Λογικές Λειτουργίες.....	132
4.2.8 Λειτουργίες Χρόνου, Διαλόγου και Λάθους.....	134
4.2.9 Λειτουργίες Γραφημάτων και Ήχου.....	137
4.2.10 Διαγράμματα και Γραφήματα.....	140
4.2.11 Γραφήματα Charts.....	143
4.2.12 Γραφήματα Graphs.....	144
4.2.13 Λειτουργία Ανάλυσης.....	146
4.2.14 Λειτουργίες Ελέγχου Εφαρμογών.....	149
4.2.15 Λειτουργίες Αρχείων.....	150
5. Παρουσίαση Express VIs.....	153
6. Εφαρμογές στα Σ.Α.Ε.....	174
Labview Control Design Toolkit.....	174
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	197

## ΠΡΟΛΟΓΟΣ



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Το θέμα αυτής της πτυχιακής εργασίας είναι η μελέτη και περιγραφή του προγράμματος LabView της εταιρίας National Instruments

Στο **Κεφάλαιο 1** γίνεται μια εισαγωγή και παρουσίαση του προγραμματιστικού περιβάλλοντος **LabView** της **National Instruments**. Περιγράφονται τα βασικά του χαρακτηριστικά, τα πλεονεκτήματα που το καθιστούν ένα πανίσχυρο λογισμικό σχεδίασης. Περιγράφεται η έννοια του **Εικονοργάνου** και στη συνέχεια παρουσιάζεται αναλυτικά το παράθυρο ελέγχου του προγράμματος (**Front Panel**) με όλες τις δυνατότητες που περιλαμβάνει, αλλά και το αντίστοιχο **Block Diagram** με βάση το οποίο εξηγείται αναλυτικά όλη η διαδικασία προγραμματισμού. Στη συνέχεια ακολουθεί περιγραφή και ανάλυση των παλετών Εργαλείων, Αντικειμένων και Λειτουργιών του περιβάλλοντος LabView.

Στο **Κεφάλαιο 2** γίνεται αναφορά στις **Δομές ελέγχου While Loop, For Loop, Case, Sequence, Event Structure** καθώς και των μεταβλητών **Global, Local, Formula Node**, και τέλος στα **Shift Registers**. Ακολουθούν παραδείγματα για την κατανόηση της λειτουργίας τους.

Στο **Κεφάλαιο 3** εξετάζονται οι **Πίνακες** (Μονοδιάστατοι και διδιάστατοι), καθώς και η περιγραφή των βασικότερων λειτουργιών επεξεργασία πινάκων. Στη συνέχεια ακολουθεί η 'έννοια του Πολυμορφισμού. Ακόμα γίνεται αναφορά στις **Συστάδες (Bundle & Unbundle)** καθώς και στις λειτουργίες τους.

Στο **Κεφάλαιο 4** περιγράφονται λεπτομερώς οι **Προγραμματιστικές λειτουργίες**. Όπως οι λειτουργίες χαρακτήρων, οι αριθμητικές, τριγωνομετρικές, λογαριθμικές, λειτουργίες σύγκρισης, γραφημάτων και ήχου καθώς και διαγράμματα και γραφήματα και τέλος λειτουργίες αρχείων.

Στο **Κεφάλαιο 5** εξετάζουμε την λειτουργία των **Express VIs** και πραγματοποιούνται κάποια παραδείγματα βημα-βήμα για την καλύτερη κατανόηση τους.

Στο **Κεφάλαιο 6**, το οποίο είναι και το τελευταίο της πτυχιακής γίνεται μελέτη και σχεδίαση Ψηφιακών Σ.Α.Ε. με χρήση του **LABVIEW Control Design and Simulation Module**.

Κλείνοντας θα θέλαμε να ευχαριστήσουμε θερμά την κα. Α. Βελώνη καθηγήτρια του τμήματος Η/Υ Συστημάτων του ΤΕΙ ΠΕΙΡΑΙΑ για την εμπιστοσύνη που έδειξε στο πρόσωπο μας με την ανάθεση της παρούσας πτυχιακής εργασίας, για την πολύτιμη καθοδήγηση και για τις συμβουλές που μας προσέφερε. Τέλος οφείλουμε ένα μεγάλο ευχαριστώ στους γονείς μας που μας στήριξαν και παρότρυναν την ολοκλήρωση των σπουδών μας.

## 1.1 Εισαγωγή στο Labview

Το **Labview** (Laboratory Virtual Instrument Engineering Workbench) είναι μία δυνατή γλώσσα προγραμματισμού μετρήσεων, ελέγχου και ανάλυσης για υπολογιστές.

Αφορά τα λειτουργικά συστήματα Win (3.1, 9x, 2000, NT, XP), Solaris, Macintosh, HP-UX. Το **Labview** ξεφεύγει από το παραδοσιακό περιβάλλον των γλωσσών προγραμματισμού που κάνουν χρήση λεκτικών εντολών, εισάγοντας τον χρήστη σε ένα γραφικό περιβάλλον που περιέχει εργαλεία για συλλογή μετρήσεων, έλεγχο αυτόνομων οργάνων, ανάλυση και παρουσίαση των μετρήσεων.

Με αυτή την γραφική γλώσσα προγραμματισμού που καλείται “**G**”, ο χρήστης μπορεί να προγραμματίσει με διαγραμματικά μπλοκ (**block diagram**). Αφού δημιουργηθούν τα διαγράμματα, το **Labview** τα μεταφράζει σε κώδικα μηχανής. Το **Labview** αποτελεί ένα ενιαίο σύστημα συλλογής μετρήσεων, ανάλυσης, ελέγχου και παρουσίασης. Για τη συλλογή μετρήσεων και τον έλεγχο οργάνων το **Labview** υποστηρίζει τα πρωτόκολλα **RS-232/422/485**, **IEEE488(GPIB)**, **VISA**, **PXI**, **VXI**, καθώς και κάρτες δειγματοληψίας. Μια πλήρης βιβλιοθήκη από οδηγούς λογισμικού (**drivers**) για όργανα διευκολύνει τον έλεγχο αυτόνομων οργάνων. Για την ανάλυση δεδομένων υπάρχουν **ρουτίνες Επεξεργασίας Σήματος, Φίλτρων, Στατιστικής, Γραμμικής Άλγεβρας** κ.ο.κ. Κλείνοντας, το **Labview** διαθέτει πολλά πλεονεκτήματα ως σύστημα παρουσίασης αποτελεσμάτων για τον χρήστη καθώς είναι γραφικό από την φύση του.

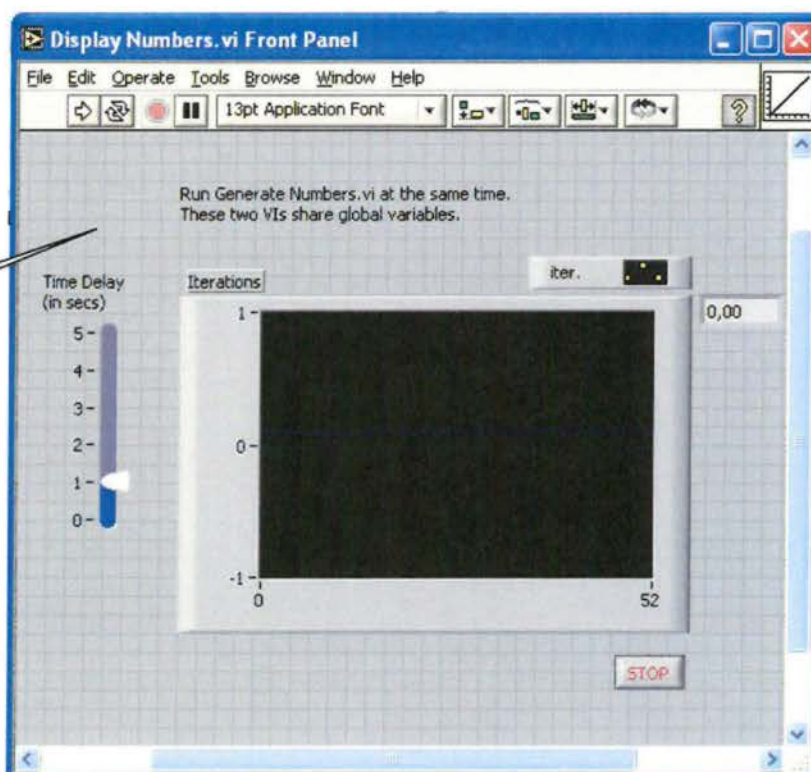
## 1.2 Εικονόγραφο (Virtual Instrument)

Η εικονική ενοργάνωση ορίζεται ως συνδυασμός του υλικού και του λογισμικού με τα βιομηχανικά πρότυπα στις τεχνολογίες υπολογιστών για την δημιουργία καθορισμένων από τον χρήστη λύσεων ενοργάνωσης. Τα προγράμματα εφαρμογών, όπως **Labview**, **LabWindows/CVI**, **Component Works**, διαθέτουν δυνατότητες προγραμματιστικού επιπέδου και ανάλυσης δεδομένων, στοιχεία που είναι απαραίτητα για την εικονική ενοργάνωση. Η εικονική ενοργάνωση μπορεί να χρησιμοποιηθεί για να δημιουργήσουμε ένα σύστημα μετρήσεων και ελέγχου, επεκτείνοντας το ρόλο τους στον τομέα της βιομηχανικής αυτοματοποίησης με συνδυασμό διαφορετικών τμημάτων υλικού και λογισμικού.

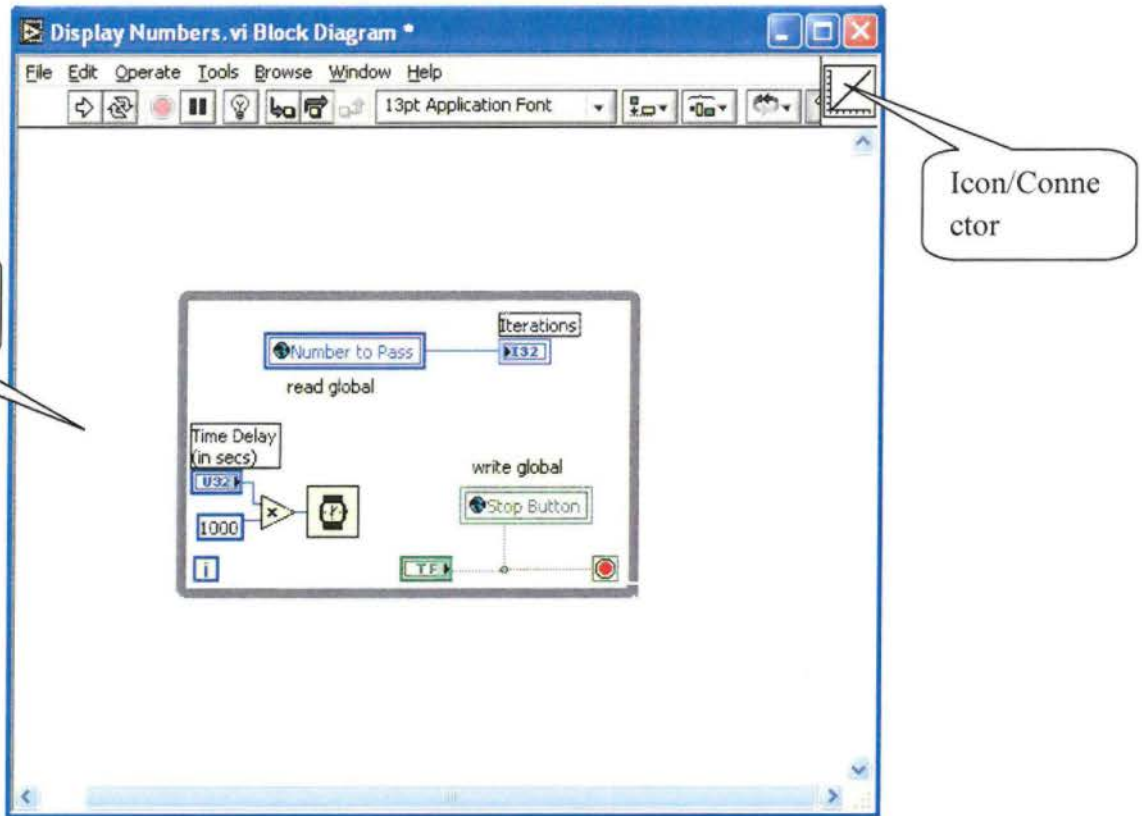
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Καλούμε **Εικονόγραφο (Virtual Instrument** ή όπως θα το λέμε από δω και πέρα **VI** οποιοδήποτε πρόγραμμα έχει γραφεί στη γλώσσα **Labview** (εικ. 1.1). Κάθε **VI** αποτελείται από :

- 1) Το μιμικό παράθυρο ή παράθυρο γραφικών (**Front Panel**)
- 2) Το τμήμα του διαγράμματος (**Block Diagram**)
- 3) Το παράθυρο του Κονέκτορα και της Εικόνας (**Icon/Connector**)



# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

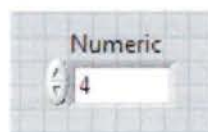


Εικόνα 1.1: Εικόργανο ή VI

Το **front panel** αποτελεί το τμήμα εκείνο όπου ο χρήστης αναγνωρίζει τις εισόδους και εξόδους του προγράμματος του.

Έστω ότι έχουμε μπροστά μας ένα πραγματικό όργανο, όπου έχουμε στην διάθεση μας πλήκτρα, διακόπτες, γράφους, και άλλες ενδείξεις που μας δίνουν την δυνατότητα να προγραμματίσουμε το όργανο. Τα ίδια ακριβώς αντικείμενα παρέχονται στο **front panel** του **Labview** και τα διακρίνουμε σε στοιχεία εισόδου και εξόδου.

**Control** ονομάζουμε κάθε τύπο στοιχείου προγραμματισμού που εισάγει δεδομένα στον κώδικα (εικ. 1.2)

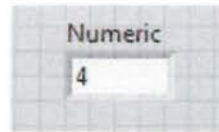


Εικόνα 1.2 Control



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

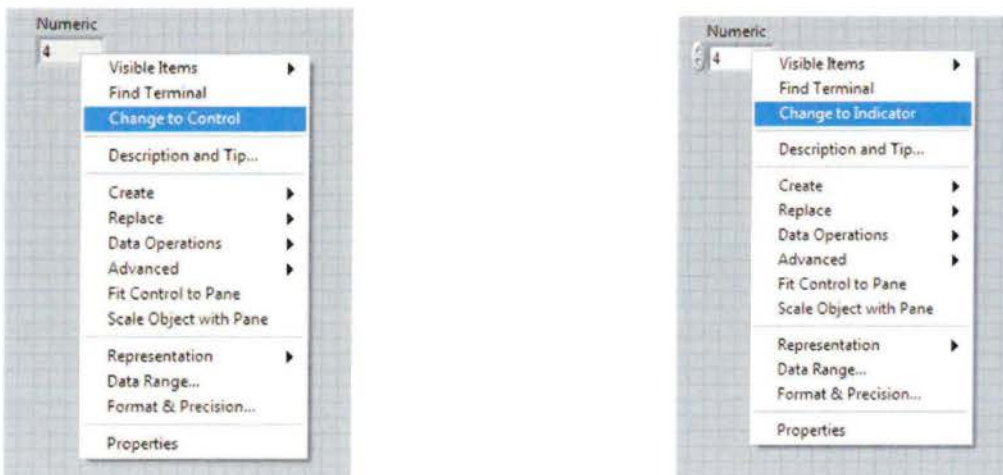
**Indicator** ονομάζουμε κάθε τύπο στοιχείου που απεικονίζει δεδομένα εξόδου που αποκτά ή παράγει το VI. (εικ. 1.3)



Εικόνα 1.3:Indicator

Παρακάτω θα αναφέρουμε παράδειγμα για την καλύτερη κατανόηση των αντικειμένων **controls**.

Το **Labview** επιτρέπει στον χρήστη να μετατρέπει ένα στοιχείο από **control** σε **indicator** επιλέγοντάς το με τον κέρσορα, όταν βρίσκεται πάνω από το αντίστοιχο αντικείμενο πατώντας το δεξί πλήκτρο από το ποντίκι και επιλέγοντας την ιδιότητά του όπως φαίνεται στην συνέχεια (εικ. 1.4)



Εικόνα 1.4: Μετατροπή στοιχείου από Indicator Control και αντίστροφα

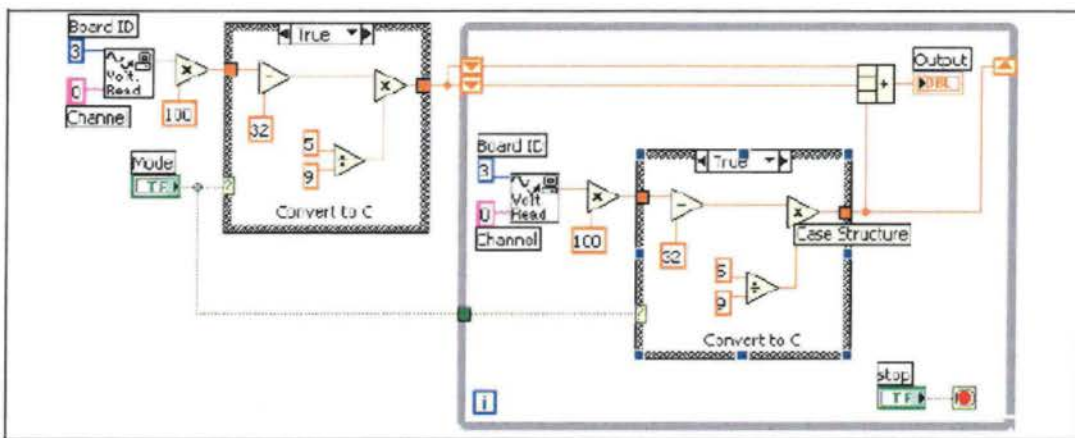
Το **front panel** ενός **VI** προσομοιώνει την εικόνα ενός φυσικού οργάνου και είναι αυτό που μας συνδέει με τον κώδικα του **VI**.

Το **front panel** διαθέτει διακόπτες, ποτενσιόμετρα (**είσοδοι** του προγράμματος), γραφήματα και δείκτες (**εξόδοι** του προγράμματος) και πολλά άλλα εξαρτήματα ελέγχου. Δίνοντας δεδομένα με το ποντίκι ή το πληκτρολόγιο τα αποτελέσματα προκύπτουν από το πρόγραμμα μας στην οθόνη σε μορφή γραφημάτων, αρχείων, χαρακτήρων και αριθμών.

Με κάθε **front panel** εμφανίζεται και ένα **block diagram** το οποίο είναι ο κώδικας του **VI**.

Κατασκευάζουμε το διάγραμμα/κώδικα με την βοήθεια την γραφικής γλώσσας προγραμματισμού “G” που αναφέραμε παραπάνω.

Θεωρούμε το διάγραμμα που ακολουθεί (εικ. 1.5) ως τον κώδικα μιας οποιασδήποτε γλώσσας προγραμματισμού με αριθμητικές ρουτίνες, κόμβους, εισαγωγές και εξαγωγές δεδομένων κ.α., στον οποίο συνδέουμε τα στοιχεία του **front panel** και του **block diagram** μεταξύ τους προσδιορίζοντας έτσι τη ροή των δεδομένων μέσα στο πρόγραμμα.

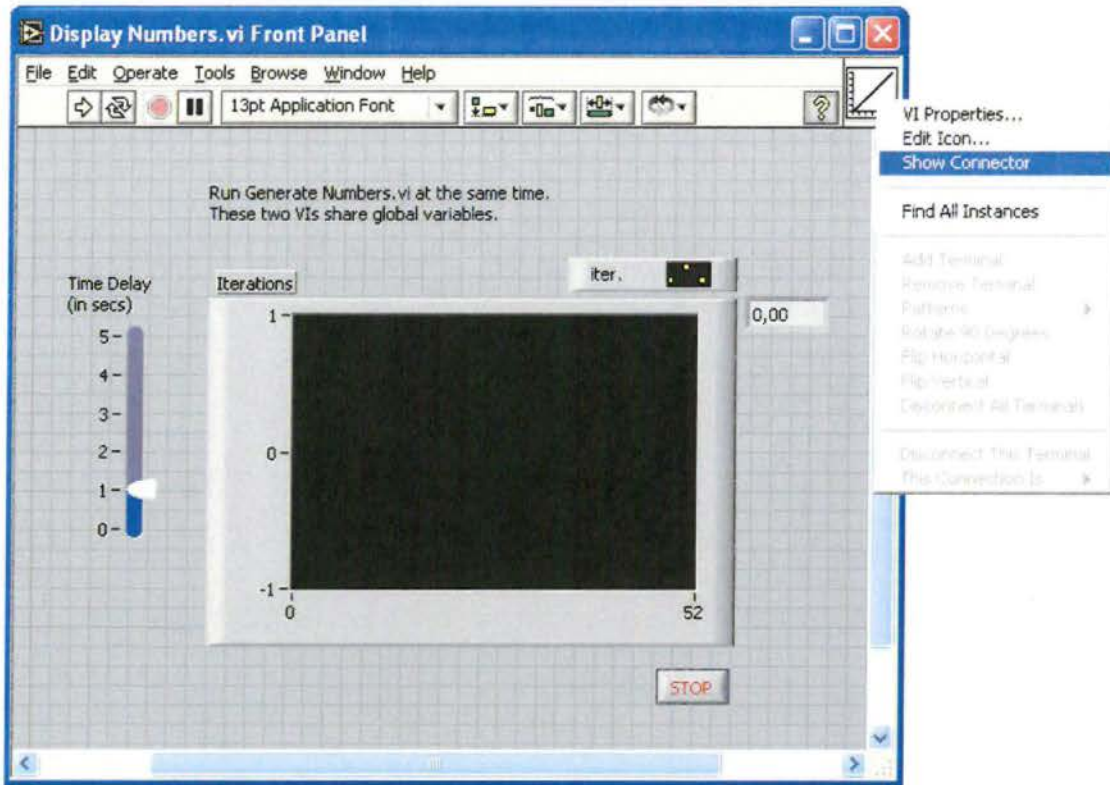


Εικόνα 1.5: Block diagram ενός VI

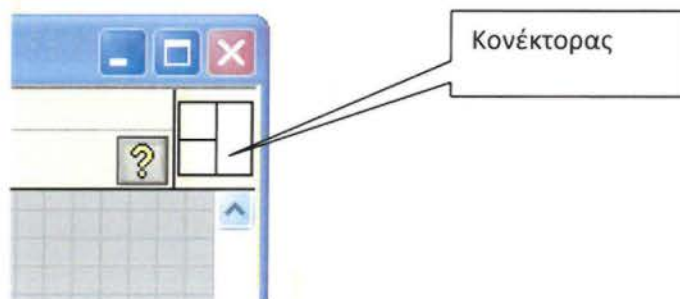
Τα VIs έχουν ιεραρχικό χαρακτήρα. Ο όρος ιεραρχικός θα κατανοηθεί καλύτερα παρακάτω, όπου περιγράφεται αναλυτικά η λειτουργία των υπορουτινών VIs (subVIs) και ο ρόλος τους στο Labview. Τα SubVIs είναι VIs που χρησιμοποιούνται από άλλα VIs. Τοποθετούνται μόνο στο **block diagram** με τη μορφή εικονιδίου και συμπεριφέρονται όπως τα άλλα αντικείμενα. Ανήκουν στην κατηγορία κόμβων διαθέτουν δηλαδή τερματικά σημεία (**terminal points**) για εισόδους και εξόδους και έχουν μια ξεχωριστή λειτουργία μέσα στο VI. Μπορούμε να τους δώσουμε τη θέση που έχουν οι υπορουτίνες άλλων γλωσσών προγραμματισμού. Χρησιμοποιούμε την Εικόνα/Κονέκτορα ώστε να μπορέσουμε να μετατρέψουμε το VI σε μία υπορουτίνα, έναν **subVI**, τον οποίο μπορούμε με αυτό τον τρόπο να καλέσουμε στο διάγραμμα οποιουδήποτε άλλου προγράμματος. Η εικόνα αναπαριστά γραφικά το **subVI** στο **block diagram** άλλων VIs, ενώ οι ακροδέκτες του κονέκτορα προσδιορίζουν που πρέπει να συνδέσουμε τις εισόδους και τις εξόδους (**terminals**) της εικόνας. Θεωρούμε τους ακροδέκτες ως τις παραμέτρους λειτουργίας του **subVI**, τις οποίες πρέπει να προσδιορίσει ο χρήστης όταν καλεί τον **subVI** από κάποιο άλλο πρόγραμμα. Οι ακροδέκτες αντιστοιχούν στα **controls** και **indicators** του **front panel** του VI. Ο

# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

κονέκτορας είναι συνήθως κρυμμένος κάτω από την εικόνα (μπορούμε να επιλέξουμε να τον δούμε με δεξιά κλικ επάνω στην εικόνα) (εικ 1.6)



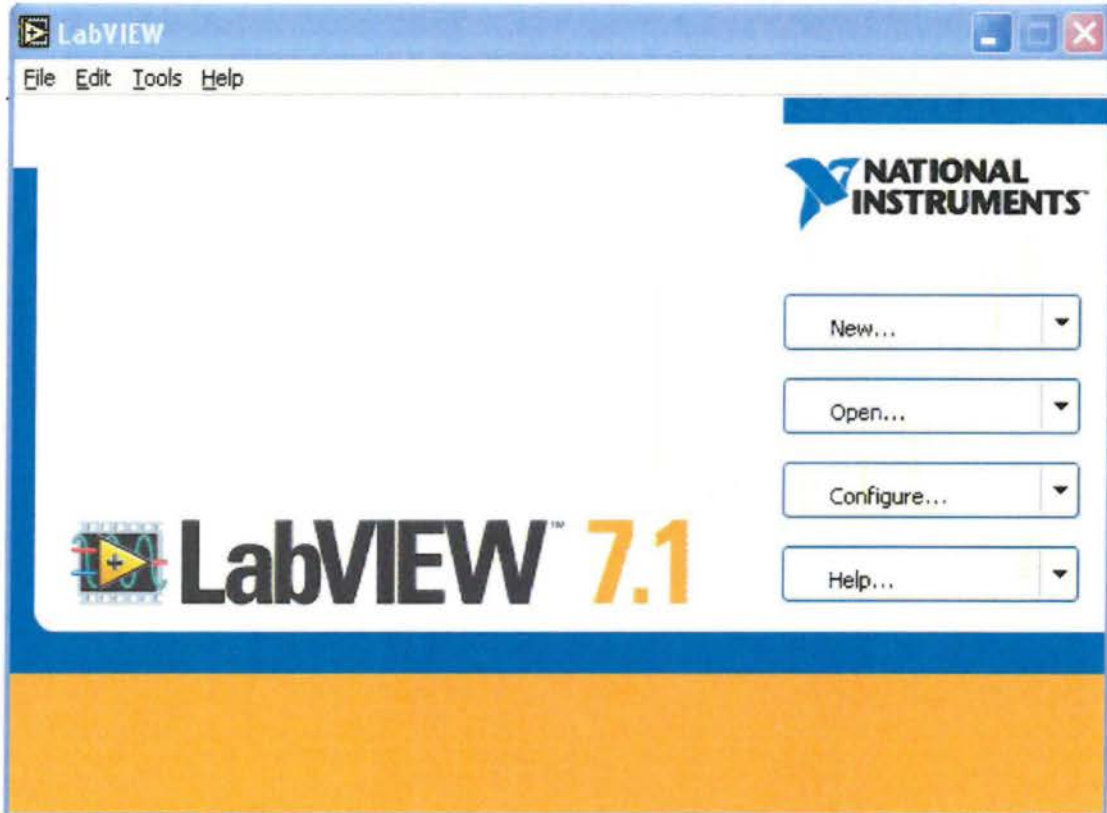
Εικόνα 1.6: Εμφάνιση Connector





### 1.3 Ξεκίνημα του Labview

Κατά την εκκίνηση του **Labview**, εμφανίζεται η παρακάτω εικόνα 1.7 για την έκδοση 7.1 .



Εικόνα 1.7: Οθόνη εκκίνησης Labview 7.1 1

Στις περισσότερες εκδόσεις οι επιλογές που εμφανίζονται είναι οι ακόλουθες:

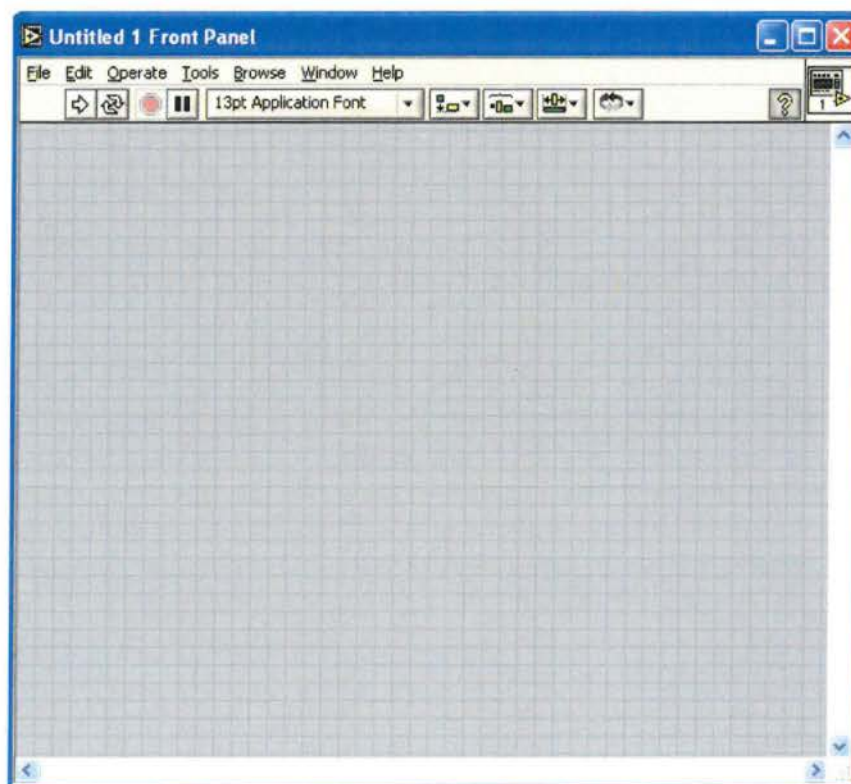
- **New VI:** Δημιουργία καινούργιου **VI**.
- **Open VI:** Άνοιγμα ενός υπάρχοντος **VI**.
- **DAQ Solution Wizard:** Το **Labview** μπορεί να κατασκευάσει μια εφαρμογή **DAQ**.
- **Configure:** Δημιουργία εφαρμογής **DAQ** με χρήση **Measurements & Automation Explorer (MAX)** και καθορισμός ιδιοτήτων των **VI**.
- **Search Examples:** Παρουσιάζει λίστα με όλα τα υπάρχοντα διαθέσιμα παραδείγματα.
- **LabVIEW Tutorial:** Δυνατότητα καλύτερης κατανόησης του **Labview** on-line.
- **Exit:** Έξοδος από το **Labview**.
- **Quick Tip:** Γρήγορα παραδειγματάκια προγραμματισμού.



## 1.4 Τα παράθυρα Front Panel και Block Diagram

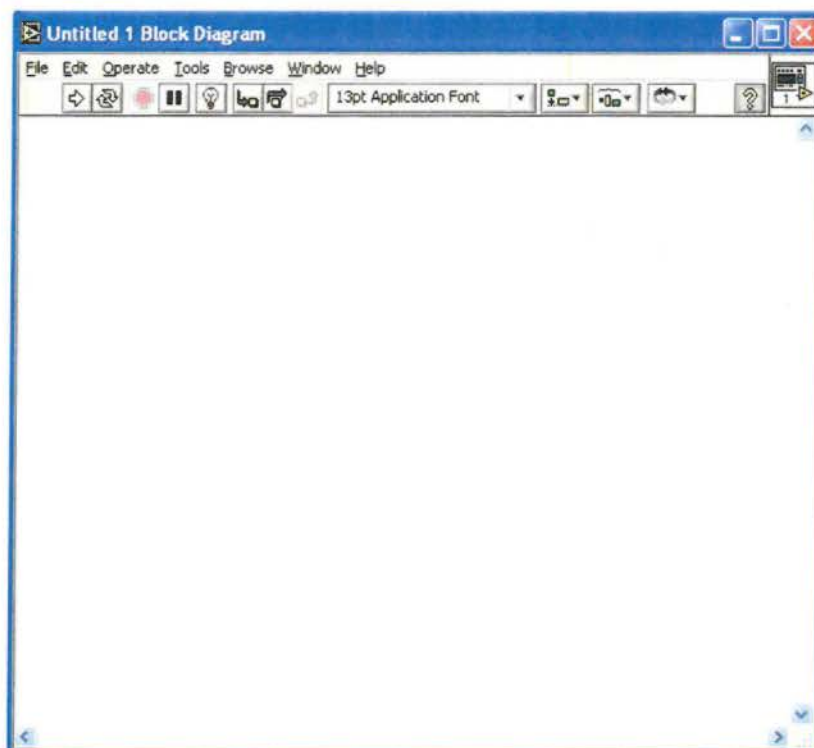
Όταν διαλέξουμε την επιλογή **New VI** από το μενού επιλογών, ανοίγει ένα νέο παράθυρο γκρι χρώματος, το οποίο είναι το **front panel** του νέου μας **VI** (εικ. 1.8). Το άλλο παράθυρο που ανοίγει μαζί και έχει λευκό χρώμα είναι το παράθυρο που θα περιέχει το **block diagram**, (εικ. 1.9).

Το **front panel** διαθέτει ποικίλους τύπους αντικειμένων, καθένα από τα οποία αντιστοιχεί σε ένα στοιχείο (**terminal**) στο **block diagram**.



Εικόνα 1.8: Παράθυρο front panel

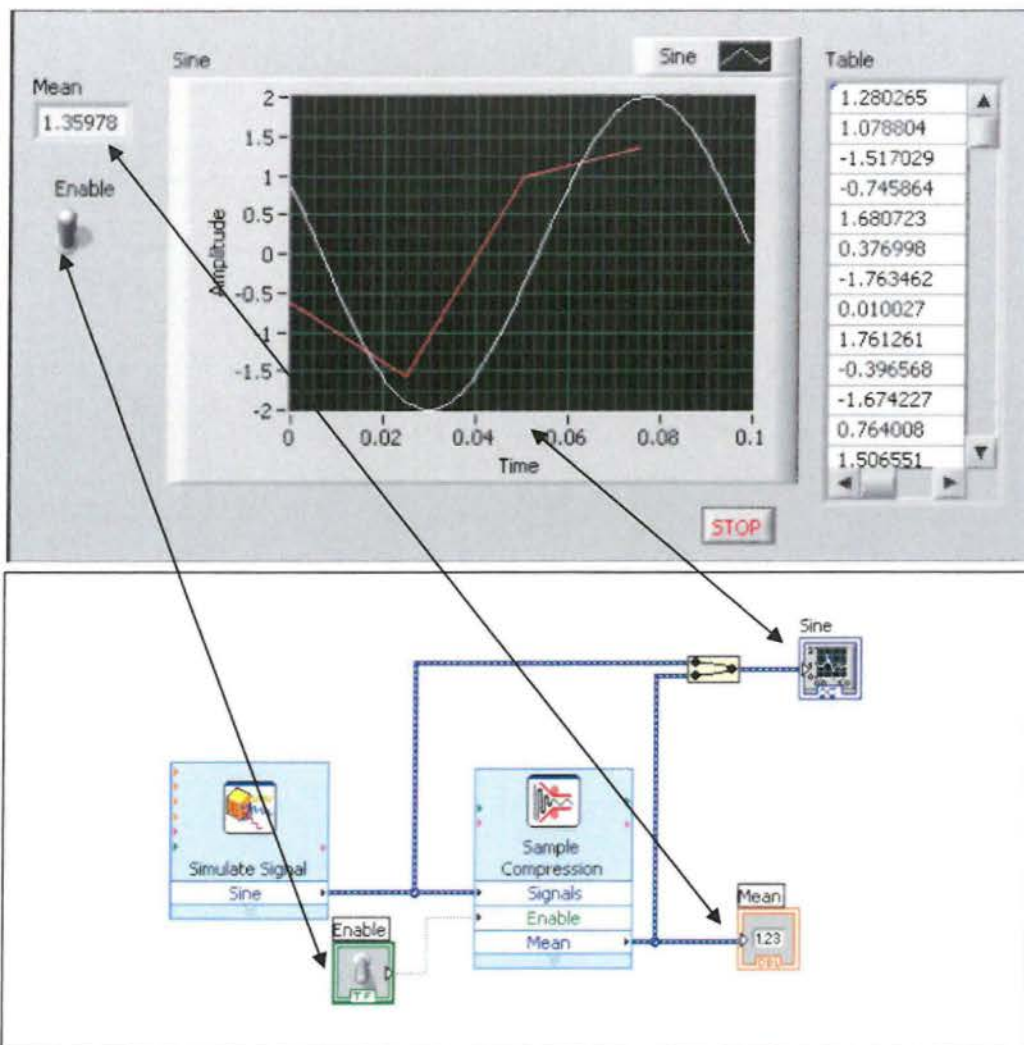
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.9: Παράθυρο block diagram

Εκτός από διάφορα τερματικά στοιχεία στο **block diagram**, μπορούμε να προγραμματίσουμε και διάφορες σταθερές, ρουτίνες, αλγόριθμους, **subVIs** και καλώδια τα οποία θα μεταφέρουν τα δεδομένα από τον ένα κόμβο στον άλλο, δίνοντας πληροφορίες για τη διαγραμματική ροή του προγράμματος μας, όπως φαίνεται και στο **VI** που ακολουθεί (εικ 1.10)

**ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.**



**Εικόνα 1.10: Συσχετισμός τερματικών 1**

### 1.4.1 Εργαλεία του Front Panel

Τόσο το **front panel**, όσο κι το **block diagram** διαθέτουν εργαλεία με τα οποία μπορούμε να ελέγξουμε τη λειτουργία των **VIs**. Η εικόνα που ακολουθεί παρουσιάζει τα εργαλεία που βρίσκουμε στην κορυφή του **front panel**.(εικ.1.11)



Εικόνα 1.11: Απεικόνιση εργαλείων στο πάνω μέρος του Front Panel



#### **Πλήκτρο Εκκίνησης-Λειτουργίας(Run).**

Με το συγκεκριμένο πλήκτρο ξεκινάμε την λειτουργία του VI.

Όταν το VI εκτελείται, το πλήκτρο αλλάζει σε:



εάν το VI είναι ο κύριος VI ή



εάν πρόκειται για έναν subVI το οποίο καλείται από κάποιο VI σε ανώτερο επίπεδο.



#### **Πλήκτρο Τερματισμού (Stop).**

Με την εκτέλεση του VI, εμφανίζεται το συγκεκριμένο πλήκτρο. Πατώντας το μπορούμε να τερματίσουμε το πρόγραμμα πάραυτα.

Παρατήρηση: Αποφεύγουμε να χρησιμοποιούμε το πλήκτρο τερματισμού (**Stop**) κατά την λειτουργία ενός VI. Είτε αφήνουμε το VI να τρέξει μέχρι να ολοκληρώσει την λειτουργία που του έχουμε ορίσει είτε βρίσκουμε μια μέθοδο ώστε να σταματά προγραμματιστικά. Και αυτό γιατί είναι καλό να είναι γνωστή η κατάσταση στην οποία το πρόγραμμα τερμάτισε τη λειτουργία του.



#### **Πλήκτρο Σπασμένης Λειτουργίας (Broken-Run)**

Εμφανίζεται όταν το VI έχει παρουσιάσει κάποιο σφάλμα στο διάγραμμα και δεν μπορεί να συνεχίσει την λειτουργία του. Αντικαθιστά το πλήκτρο Run. Πατώντας το συγκεκριμένο πλήκτρο έχουμε την δυνατότητα να διακρίνουμε ποια ακριβώς είναι τα σφάλματα.



#### **Πλήκτρο Συνεχούς Λειτουργίας (Continuous Run).**

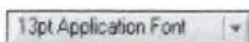
Πατώντας το, το VI εκτελείται συνεχώς.





#### Πλήκτρο Προσωρινής Διακοπής (Pause).

Πατώντας το, επαναφέρουμε το VI στην κανονική του λειτουργία.



#### Πλήκτρο γραμματοσειράς.

Με το συγκεκριμένο πλήκτρο έχουμε την δυνατότητα να επιλέξουμε τον τύπο, το μέγεθος, το στυλ αλλά και το χρώμα της γραμματοσειράς που επιθυμούμε να χρησιμοποιήσουμε.



#### Πλήκτρο ευθυγράμμισης.

Με αυτό το πλήκτρο μπορούμε να επιλέξουμε την μέθοδο ισοστοίχισης των αντικειμένων, δηλαδή κάθετα, οριζόντια, κ.ο.κ.



#### Πλήκτρο ισοστοίχιση αντικειμένων.

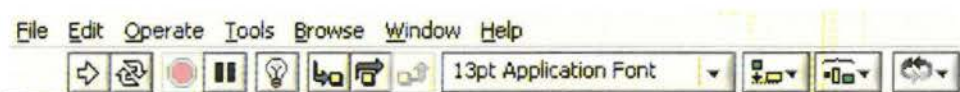


#### Πλήκτρο τοποθέτησης αντικειμένων σε επίπεδα χώρου.

Επιλέγουμε αυτό το πλήκτρο όταν αντικείμενα επικαλύπτουν το ένα το άλλο και θέλουμε να ορίσουμε ποίο θα είναι μπροστά ή πάνω από το άλλο.

## 1.4.2 Εργαλεία του Block Diagram

Σε αυτή την ενότητα θα αναλύσουμε την λειτουργία των εργαλείων που βρίσκονται στο **block diagram**.



#### Πλήκτρο Αργής Κίνησης (Execution Highlighting).

Μπορούμε να δούμε τη ροή των δεδομένων μεταξύ των κόμβων, σε αργή κίνηση, δίνοντας μας έτσι την δυνατότητα να διαγνώσουμε λάθη στον προγραμματισμό του διαγράμματος. Σε περίπτωση που δεν διέρχονται δεδομένα από τα καλώδια τότε αυτά αλλάζουν χρώμα όπως και τα VIs.



#### Πλήκτρο εκτροπής (Single Step/Step Over).

Τρέχουμε το πρόγραμμα από κόμβο σε κόμβο. Φτάνοντας σε έναν κόμβο μπορούμε να αποφύγουμε να τον τρέξουμε, για να κερδίσουμε χρόνο

πατώντας το πλήκτρο υπέρβασης (**Step over**) παραδείγματος χάρη ένα **Loop** 500 επαναλήψεων.



### Πλήκτρο Βήματος Μετάβασης (**Step Intro**).

Αντίθετα με το πλήκτρο βήματος υπέρβασης (**Step Over**), μας δίνει την δυνατότητα να εισχωρήσουμε στον κόμβο και να παρακολουθήσουμε την λειτουργία του.

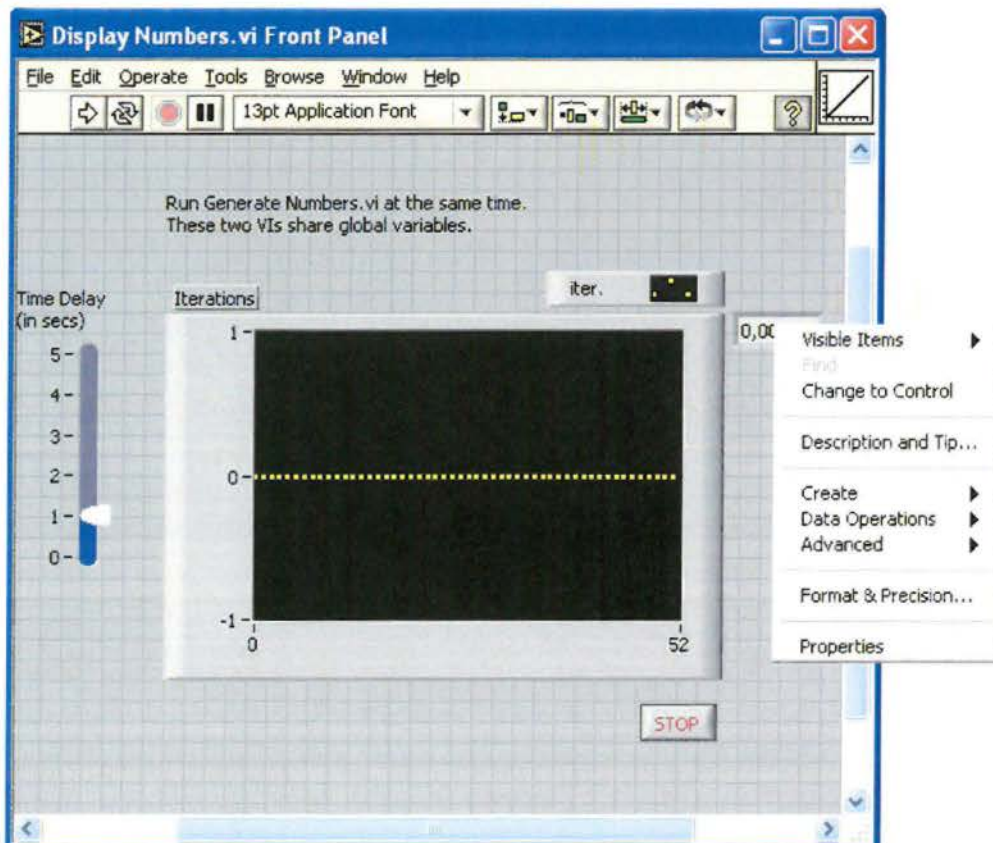
### ☛ Πλήκτρο Βήματος Υπέρβασης

Με το συγκεκριμένο πλήκτρο έχουμε την δυνατότητα, έχοντας εισχωρήσει μέσα στον κόμβο να εξέλθουμε από αυτόν.

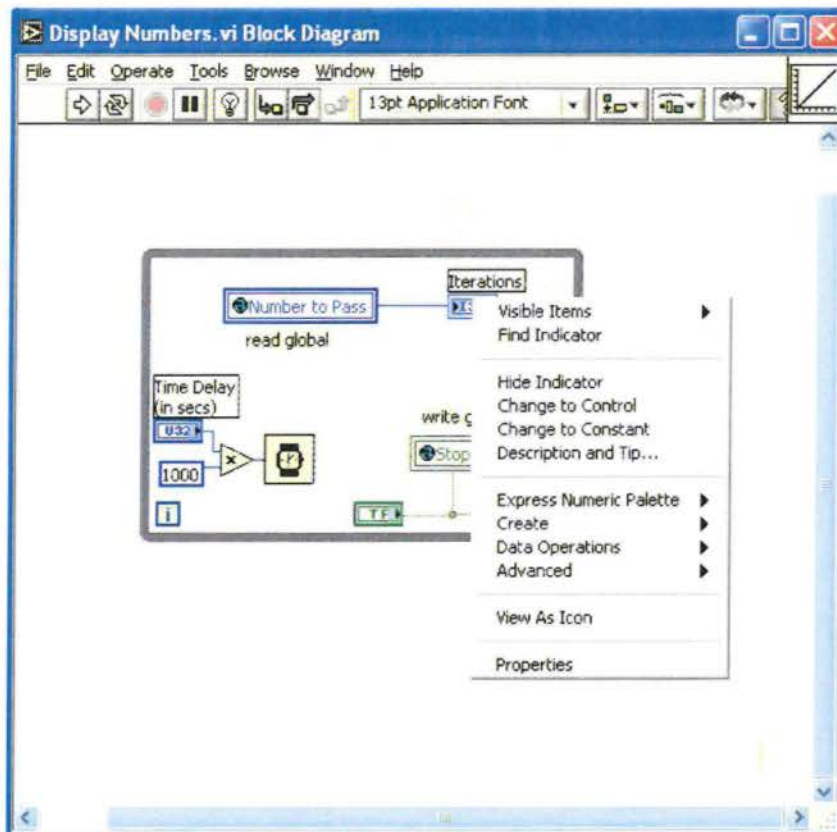
## 1.4.3 Μενού Pop-Up

Σχεδόν όλα τα αντικείμενα που χρησιμοποιούμε για την κατασκευή ενός **VI** έχουν **Pop-Up** μενού.

Τα **Pop-Up** μενού εμφανίζονται όταν ο κέρσορας βρίσκεται πάνω από το αντίστοιχο αντικείμενο ή παράθυρο και πατήσουμε το δεξί πλήκτρο του ποντικιού, όπως φαίνεται και στις εικόνες που ακολουθούν (εικ. 1.12)

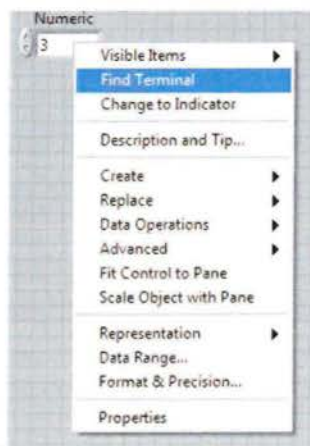


## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.12: Εμφάνιση Pop-Up μενού στο περιβάλλον του Front Panel & Block diagram αντίστοιχα

Οι εντολές των **Pop-Up** μενού μπορεί να διαφέρουν στο **Front Panel** από εκείνες του **Block Diagram**. Κάποια στοιχεία μπορεί να έχουν διαφορετικό μενού με περισσότερες εντολές ή και ακόμη διαφορετικές από κάποια άλλα. Η εικόνα 1.13 παρακάτω μας δείχνει το **Pop-Up** μενού ενός αριθμητικού στοιχείου.



Εικόνα 1.13: Απεικόνιση Pop-Up μενού αριθμητικού στοιχείου

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Οι εντολές που ακολουθούν μπορούμε να τις βρούμε είτε σε κάποιο **control** είτε σε κάποιο **indicator**.

### ➤ **Visible Items**

Εμφανίζει μια λίστα αντικειμένων που μπορούν να εμφανιστούν ή να κρυφτούν όποτε το επιθυμούμε. Αυτά τα αντικείμενα μπορεί να είναι είτε η ετικέτα είτε η επικεφαλίδα.

### ➤ **Find Terminal**

Δείχνει σε ποιο στοιχείο του **block diagram** είναι τοποθετημένο το τερματικό στοιχείο του αντικειμένου.

### ➤ **Change to Indicator**

Αλλάζει το **control** σε **indicator**. Αντίστοιχα όταν το αντικείμενο είναι **indicator**, η εντολή αλλάζει σε **Change to control**.

### ➤ **Description and Trip**

Με αυτή την εντολή μπορούμε να εισάγουμε περιγραφή και σχόλια για ένα αντικείμενο μέσω ενός πίνακα.

### ➤ **Create**

Εισάγει μεταβλητή ή και στοιχείο **control**.

### ➤ **Replace**

Αντικαθιστά το αντικείμενο με ένα άλλο, χωρίς να χαθούν στοιχεία που έχουμε εισάγει όπως ετικέτα ή τιμές.

### ➤ **Data Operations**

Δίνει πρόσβαση σε κάποιες εντολές από τα μενού **Operate** και **Edit**

### ➤ **Advanced:**

#### • **Key Navigation**

Ορίζει το πλήκτρο συντόμευσης για την λειτουργία του **control**.

#### • **Customize**

Διαμορφώνει το **control** στην επιθυμητή μορφή.

#### • **Hide Control**

Εξαφανίζει το **control**. Για να το εμφανιστεί ξανά, από το **Short Cut** μενού του τερματικού επιλέγουμε **Show control**.

#### • **Enable State**

Ορίζει την κατάσταση του **control**, η οποία μπορεί να είναι είτε ενεργή είτε ανενεργή.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Εξειδικευμένες εντολές:

➤ **Representation**

Αλλάζει τον τύπο δεδομένων του **control** ή του **indicator**.

➤ **Data Range**

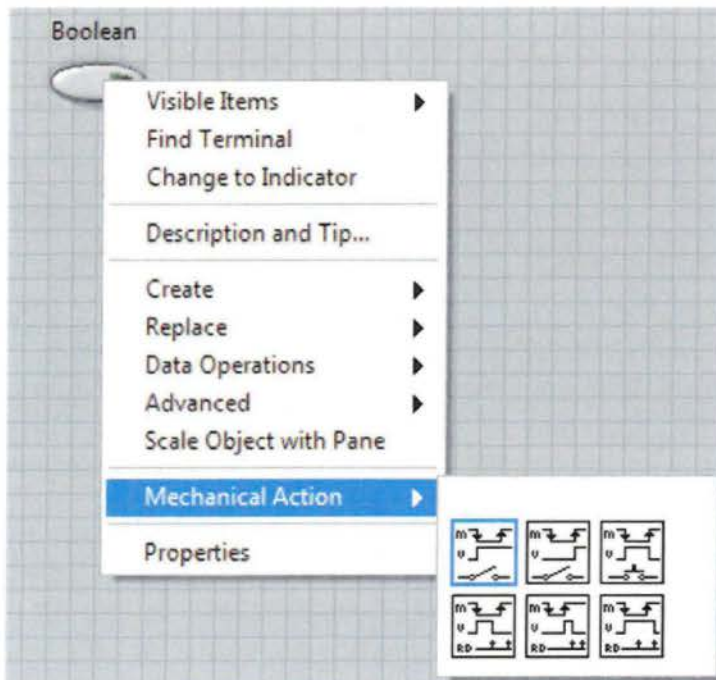
Ορίζει το είδος των τιμών.

➤ **Format & Precision**

Διαμορφώνει τη μορφή των ψηφίων της τιμής.

➤ **Mechanical Action**

Ορίζει τον τρόπο που λειτουργεί ο διακόπτης (εικ 1.14)

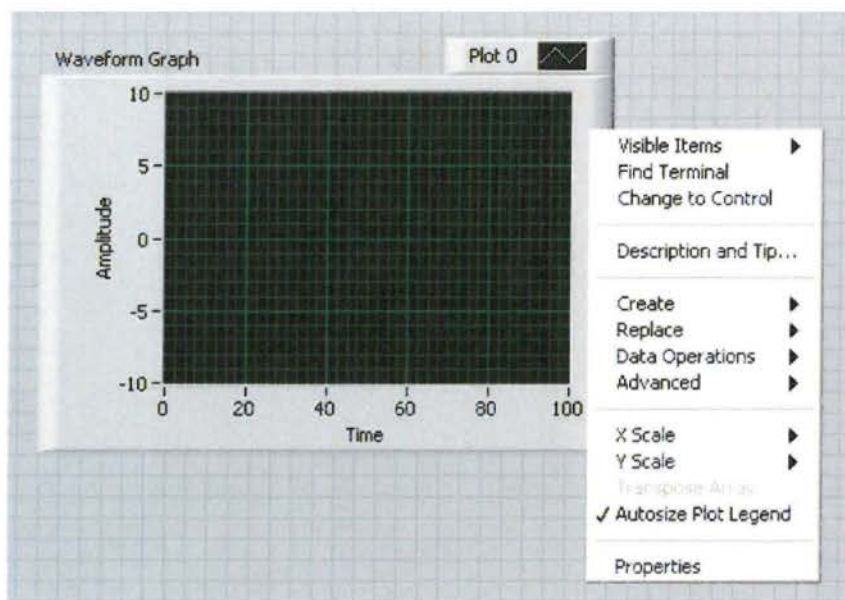


Εικόνα 1.14: Απεικόνιση Pop-Up μενού για control Boolean

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

### ➤ X scale / Y scale

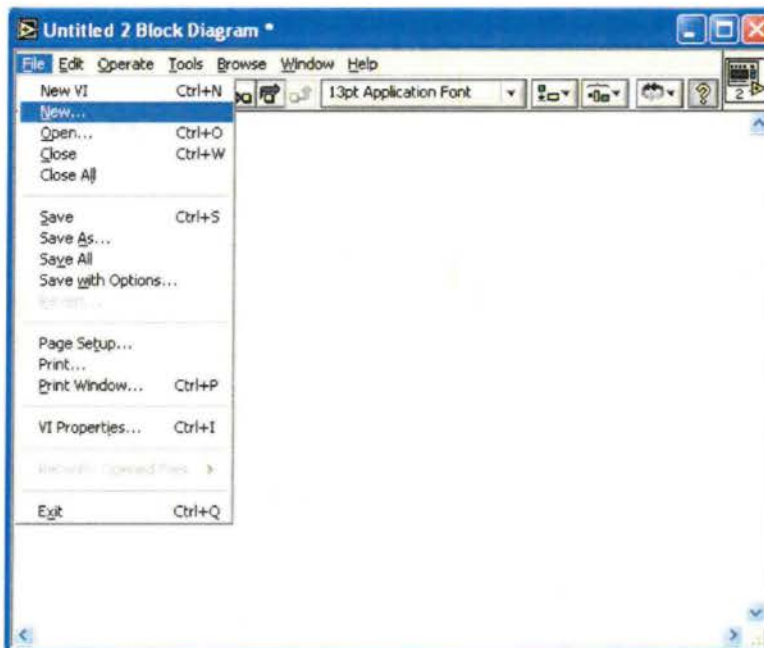
Ρυθμίζει τα χαρακτηριστικά των αξόνων X / Y. (εικ. 1.15)



Εικόνα 1.15: Απεικόνιση Pop-Up μενού για για control Graph

### 1.4.4 Μενού Pull- Down

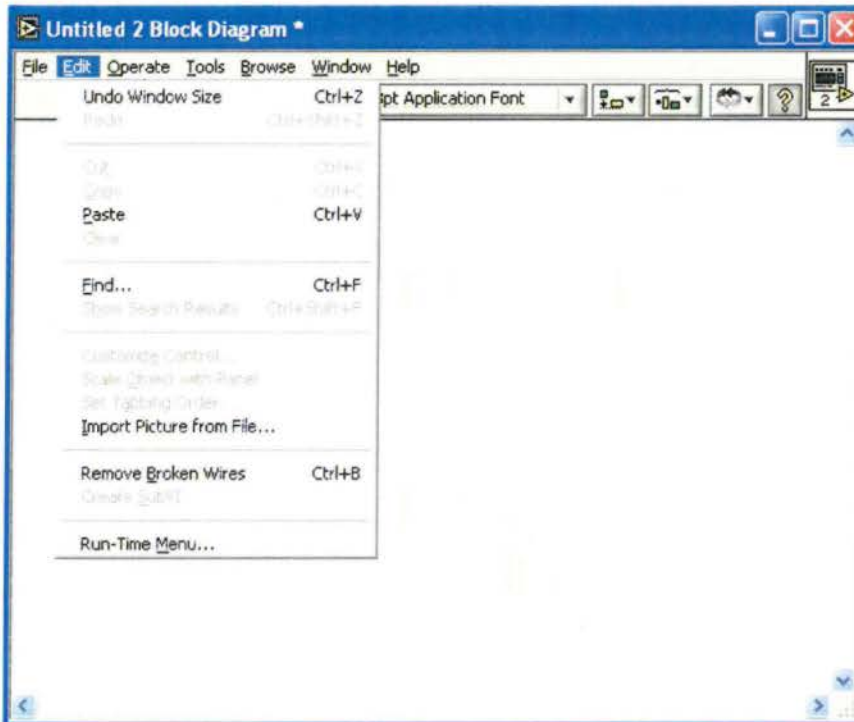
Στο πάνω μέρος του παραθύρου **VI** είναι διαθέσιμα τα μενού **Pull-Down**. Μόλις επιλέξουμε ένα μενού από την μπάρα επιλογής, εμφανίζεται μια λίστα από εντολές. Παρακάτω ακολουθούν μερικές εντολές οι οποίες είναι οι πιο κοινές στις διάφορες εκδόσεις **Labview** που κυκλοφορούν.



- **New:** Δημιουργία καινούργιου VI.
- **Open:** Άνοιγμα ενός ήδη υπάρχοντος VI.
- **Close:** Κλείσιμο του VI που χρησιμοποιούμε.
- **Save:** Αποθήκευση του VI.
- **Save As:** Αποθήκευση του VI με νέο όνομα.
- **Page Setup:** Δυνατότητα της μορφής της σελίδας που θέλουμε να εκτυπώσουμε καθώς και τροποποίηση της εκτύπωσης.
- **Print:** Εκτύπωση της σελίδας.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

- **VI Properties:** Εμφανίζει τις ιδιότητες του VI όπως τη μνήμη RAM που χρησιμοποιεί, τον κωδικό πρόσβασης στο Labview, τις διαστάσεις και την εμφάνιση του παραθύρου εργασίας.
- **Recently opened files:** Μπορεί να διαθέσει τα τελευταία 10 αρχεία που έχουν ανοιχθεί.
- **Exit:** Έξοδος από το Labview.

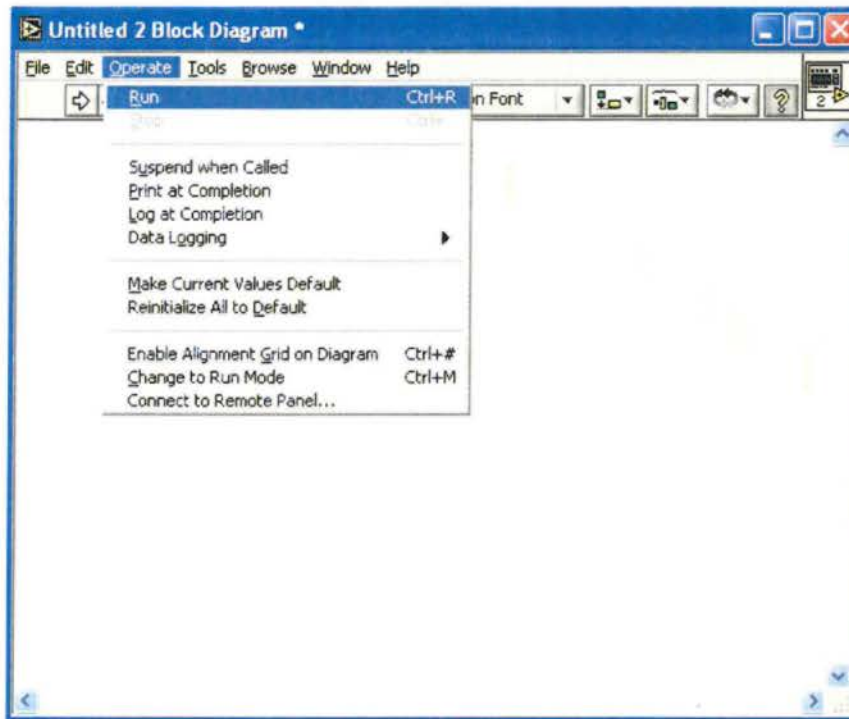


- **Undo Delete :** Αναίρεση της τελευταίας ενέργειας.
- **Redo:** Ακύρωση αναίρεσης που έχουμε πραγματοποιήσει.
- **Cut:** Αποκοπή.
- **Copy:** Αντιγραφή.
- **Paste:** Επικόλληση του αντιγραμμένου
- **Find:** Εύρεση αρχείου VI.
- **Import picture from file:** Εισαγωγή εικόνας στο VI από αρχείο.
- **Remove broken wires:** Διαγραφή των λανθασμένων συνδέσεων του block diagram.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

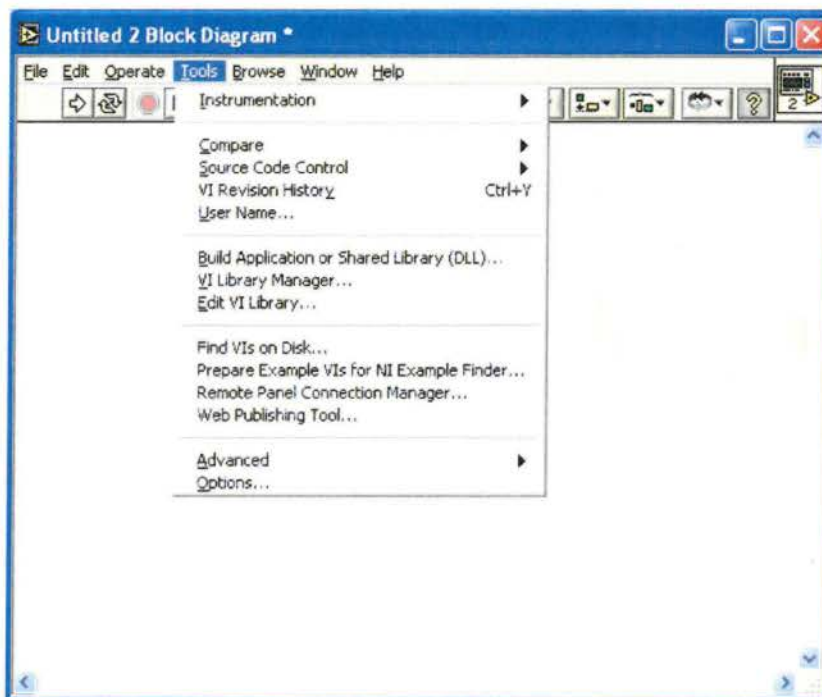
- **Create subVI:** Δημιουργεί ένα subVI από το επιλεγμένο αντικείμενο.
- **Run time menu:** Μας δίνει την δυνατότητα να φτιάξουμε ένα μενού με τις εντολές που εμείς έχουμε ανάγκη.



- **Run:** Τρέχει το VI.
- **Stop:** Σταματάει την εκτέλεση του VI. Προτείνεται να μην χρησιμοποιείται συχνά γιατί υπάρχει πιθανότητα να κάνει το σύστημα ασταθές. Για να αποφευχθεί κάτι τέτοιο συνίσταται η χρήση διακοπών τερματισμού.
- **Breakpoints:** Προσπαθεί να ανακαλύψει στο VI τα σημεία που προκαλούν τη διακοπή της εξομοίωσης του.
- **Suspend when called:** Αναβάλλει την εκτέλεση του VI όταν ζητηθεί να χρησιμοποιηθεί ως **subVI**.
- **Print at completion:** Εκτυπώνει τα αποτελέσματα του **front panel** όταν τελειώσει η εκτέλεση του VI.
- **Log at completion:** Καταχωρεί τα δεδομένα όταν τελειώσει η εκτέλεση του VI.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

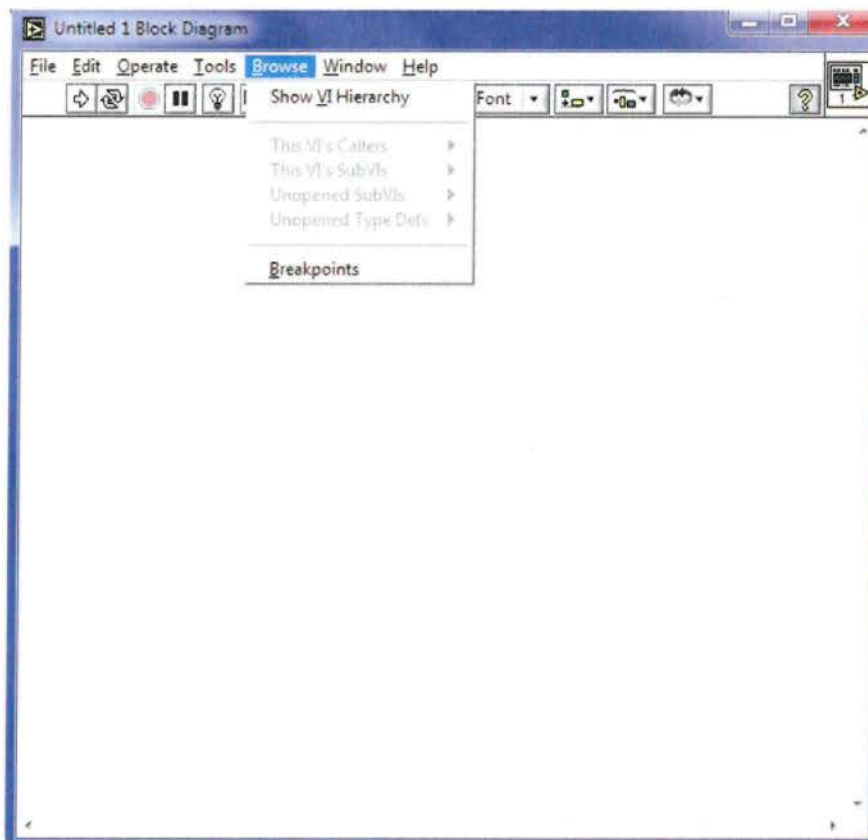
- **Data logging:** Δημιουργείτε ένα αρχείο για να αποθηκεύει τα δεδομένα του VI. Με αυτή την εντολή έχουμε πρόσβαση σε αυτό το αρχείο όποτε επιθυμούμε.
- **Change to run mode:** Θέτει το VI σε εκτελέσιμη κατάσταση. Σε αυτή την κατάσταση δεν μπορούμε να τροποποιήσουμε το VI. Όταν το VI βρίσκεται σε **Run Mode**, η εντολή αλλάζει σε **Edit Mode**.



- **Measurement and automation explorer:** Ανοίγει το **MAX**, το οποίο το χρησιμοποιούμε όταν θέλουμε να σχηματίσουμε τα όργανα / εργαλεία από τις εγκατεστημένες συσκευές στο σύστημα μας. Επιπλέον προσθέτει καινούργια κανάλια και VIs στο σύστημα.
- **Instrumentation:** Διαλέγοντας αυτή την επιλογή μπορούμε να συνδεθούμε με την ιστοσελίδα του λογισμικού της **National Instruments**.
- **Compare:** Κάνει σύγκριση μεταξύ δύο VIs και στο τέλος της σύγκρισης εμφανίζει τις διαφορές.
- **VI ReVision history:** Εμφανίζει ένα πίνακα με σχόλια του VI. Μπορούμε να προσθέτουμε σχόλια ανά πάσα στιγμή.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

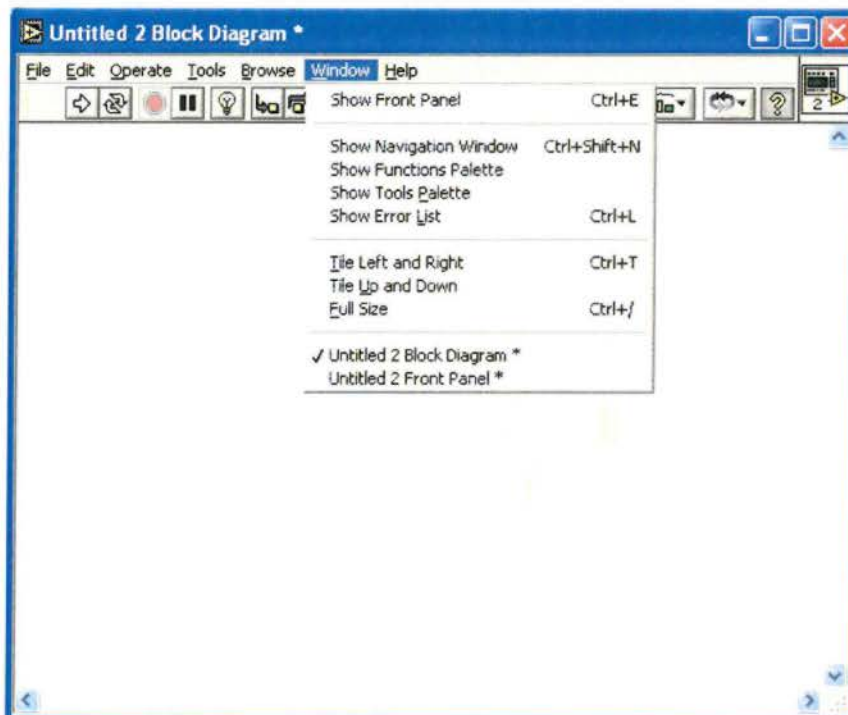
- **Options:** Παρουσιάζει τα χαρακτηριστικά του Labview.
- **User name:** Ονοματίζουμε τον χρήστη του Labview ή αλλάζουμε τον ήδη υπάρχοντα



- **Show VI hierarchy:** Εμφανίζει το παράθυρο ιεραρχίας των VI. Παρουσιάζει δηλαδή το κάθε **subVI** μέσα σε ποιο VI βρίσκεται.
- **This VI's callers:** Εμφανίζει μια λίστα με τα VIs που καλούν το συγκεκριμένο VI και το χρησιμοποιούν ως **subVI**.
- **This VI's subVIs:** Εμφανίζει μια λίστα με τα subVIs που καλεί το VI.
- **Unopened subVIs:** Εμφανίζει τη λίστα με τα **subVIs** που δεν είναι ανοιχτά εκείνη τη χρονική στιγμή.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

- **Breakpoints:** Ανιχνεύει το VI για τα σημεία που προκαλούν τη διακοπή της εξομοίωσης του VI

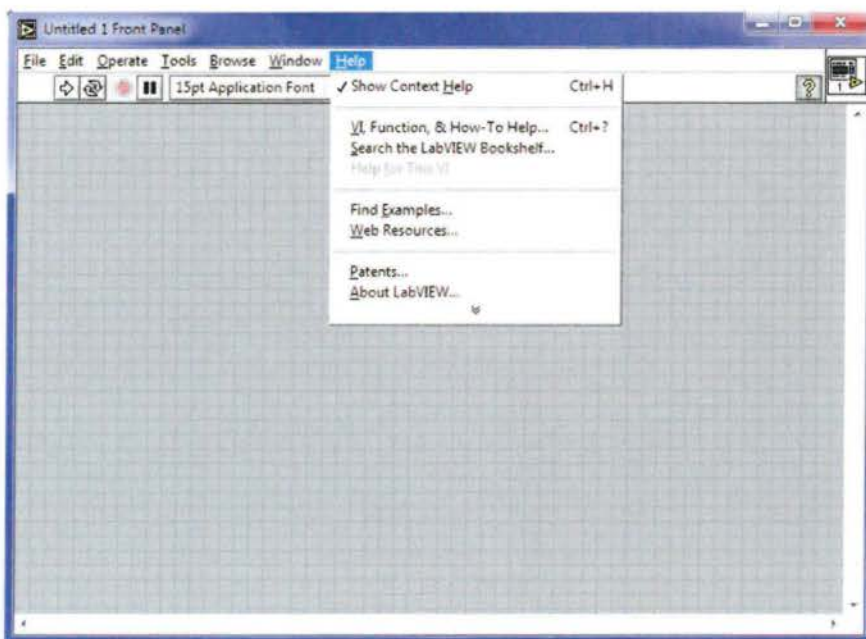


- **Show Front Panel:** Κάνει μετάβαση στο **Front Panel**. Ενώ όταν είμαστε στο **Front Panel**, η εντολή αλλάζει σε **Show Block diagram** και κάνει μετάβαση στο **block diagram** αντίστοιχα.
- **Show Controls Palette:** Εμφανίζει την παλέτα με τα **control**.
- **Show Tools palette:** Έχουμε πρόσβαση στην παλέτα με τα εργαλεία.
- **Show Error list:** Εμφανίζει το παράθυρο διαλόγου με τη λίστα που περιλαμβάνει τα λάθη του VI.
- **Tile left and right:** Ανοίγει το **front panel** στο μισό αριστερό της οθόνης ενώ το **block diagram** στο άλλο δεξί.
- **Tile up and down:** Κάνει ανάλογη λειτουργία με την προηγούμενη εντολή μόνο που επιλέγοντας αυτή την εντολή το **front panel** εμφανίζεται στο πάνω μέρος της οθόνης και το **block diagram** στο κάτω.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

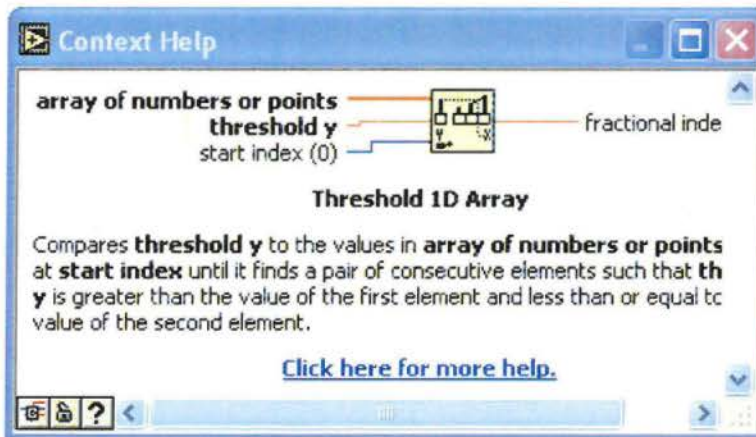
- **Full size:** Μεγιστοποιεί το παράθυρο.



- **Search the LabVIEW Help:** Εμφανίζει το παράθυρο βοήθειας του Labview. Έχουμε την δυνατότητα να επιλέξουμε ώστε η βοήθεια να είναι συγκεκριμένη για το VI που είναι ανοιχτό ή μπορούμε να εμφανίσουμε πλήρη λίστα βοήθειας.
- **Explain error:** Εξηγεί τα λάθη του VI.
- **Find Examples:** Εμφανίζει μια λίστα από VI ως παραδείγματα.
- **About LabVIEW:** Διαθέτει πληροφορίες για το Labview.

### 1.4.5 Παράθυρο Γρήγορης Βοήθειας

Για να εμφανίσουμε το παράθυρο γρήγορης βοήθειας, επιλέγουμε **Show Context Help** από το μενού **Help** ή πατάμε **ctrl-H**. Με την τοποθέτηση οποιουδήποτε από τα εργαλεία από την παλέτα εργαλείων (**Tools**) επάνω στα αντικείμενα, όπως **subVIs**, κόμβοι, λειτουργίες, σταθερές, στο διάγραμμα το παράθυρο **Help** παρουσιάζει την αντίστοιχη εικόνα και τα καλώδια που πρέπει να συνδεθούν σε κάθε ακροδέκτη. Οι ακροδέκτες οι οποίοι πρέπει απαραίτητα να συνδεθούν σημειώνονται με πορτοκαλί. Η εικόνα που ακολουθεί (εικ.1.16 ) απεικονίζει ένα παράθυρο επιλογής **Απλής / Αναλυτικής Βοήθειας (Simple / Complex Diagram Help)**.



Εικόνα 1.16: Παράθυρο επιλογής Απλής / Αναλυτικής Βοήθειας (Simple/Complex Diagram Help)

#### **Απλή βοήθεια.**

Με αυτή την επιλογή δίνουμε έμφαση στις απαραίτητες συνδέσεις. Οι υπόλοιποι ακροδέκτες απλά φαίνονται, γνωστοποιώντας τον χρήστη ότι υπάρχουν και άλλες συνδέσεις. Αντίθετα η αναλυτική βοήθεια παρουσιάζει όλους τους ακροδέκτες και ο χρήστης μπορεί να την επιλέξει από το μενού **Help**.

#### **Κλείδωμα του παραθύρου Γρήγορης Βοήθειας.**

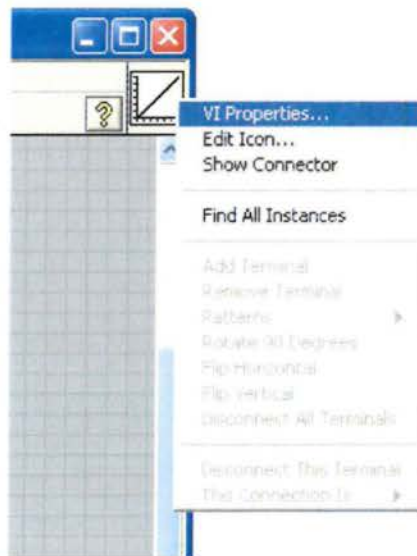
Διαλέγοντας αυτή την δυνατότητα μπορούμε να κλειδώσουμε το περιεχόμενο του παραθύρου της βοήθειας πάνω στο συγκεκριμένο αντικείμενο, για να μην αλλάζει όταν μετακινούμε το ποντίκι επάνω σε άλλα αντικείμενα. Ξεκλειδώνουμε το παράθυρο είτε μέσα από το μενού **Help** είτε ενεργοποιώντας ξανά την επιλογή κλείδωμα (**Lock**) στο κάτω μέρος του παραθύρου της βοήθειας.

### Βοήθεια On-line

Με αυτό τον τρόπο έχουμε απευθείας πρόσβαση **On-line** στα εγχειρίδια χρήσης του **Labview** και τις αναλυτικές περιγραφές για κάθε αντικείμενο.

## 1.5 Ιδιότητες των VIs

Μπορούμε να ορίσουμε τις ιδιότητες των λειτουργιών ενός VI παντώντας το δεξί πλήκτρο από το ποντίκι επάνω στο εικονίδιο του κοννέκτορα και επιλέγοντας την εντολή **VI Properties** όπως φαίνεται στην εικόνα που ακολουθεί.(εικ 1.17)



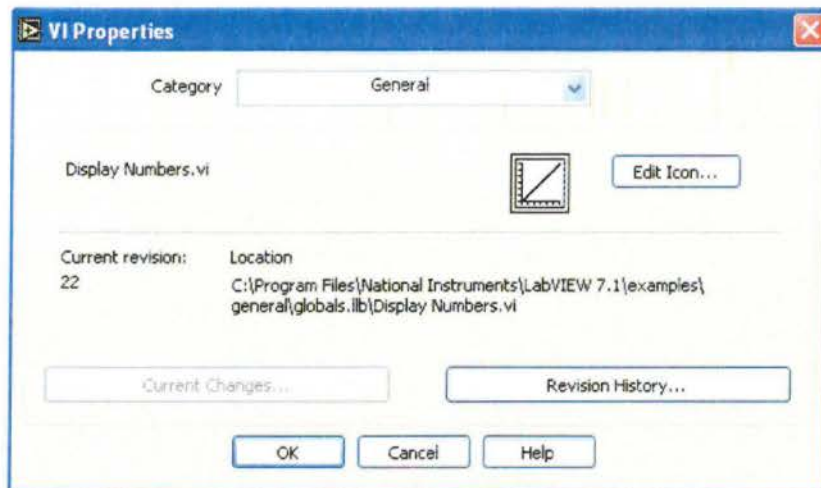
Εικόνα 1.17: Εμφάνιση ιδιοτήτων VI

Διαλέγοντας την επιλογή **VI Properties** έχουμε πρόσβαση στις κατηγορίες των ιδιοτήτων των VIs που ακολουθούν:

### ➤ Γενικές Ιδιότητες(General)

Με αυτή την κατηγορία μπορούμε να διαμορφώσουμε την εικόνα του κοννέκτορα και να δούμε που αποθηκεύεται το VI (εικ 1.18)

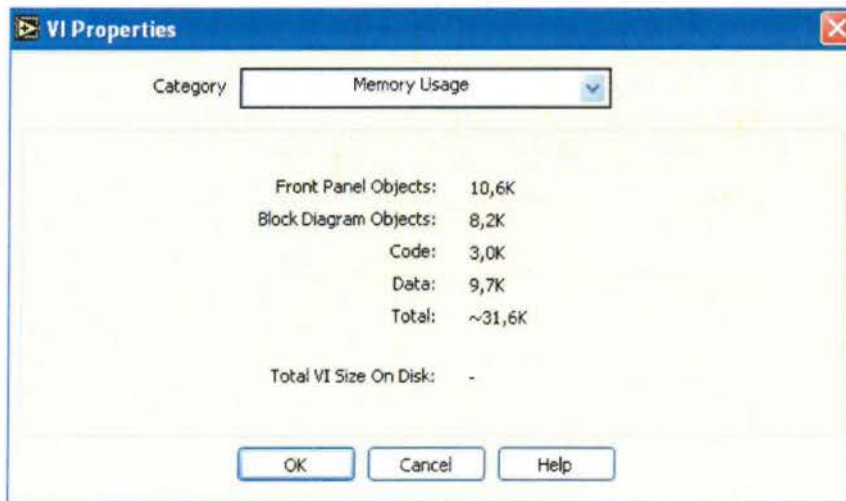
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.18: Παράθυρο γενικών ιδιοτήτων VI

### ➤ Χρήση Μνήμης (Memory Usage)

Επιλέγοντας αυτή την κατηγορία μπορούμε να δούμε τα ποσοστά μνήμης που καταλαμβάνουν τα επιμέρους τμήματα κώδικα του VI (εικ 1.19).



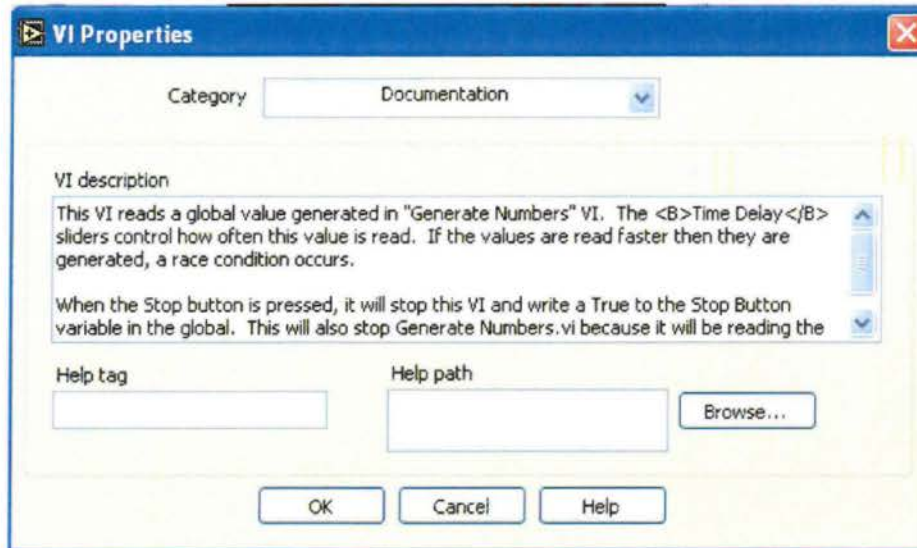
Εικόνα 1.19: Παράθυρο χρήσης μνήμης VI



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

### ➤ Λειτουργία Περιγραφής Κειμένου VI (Documentation)

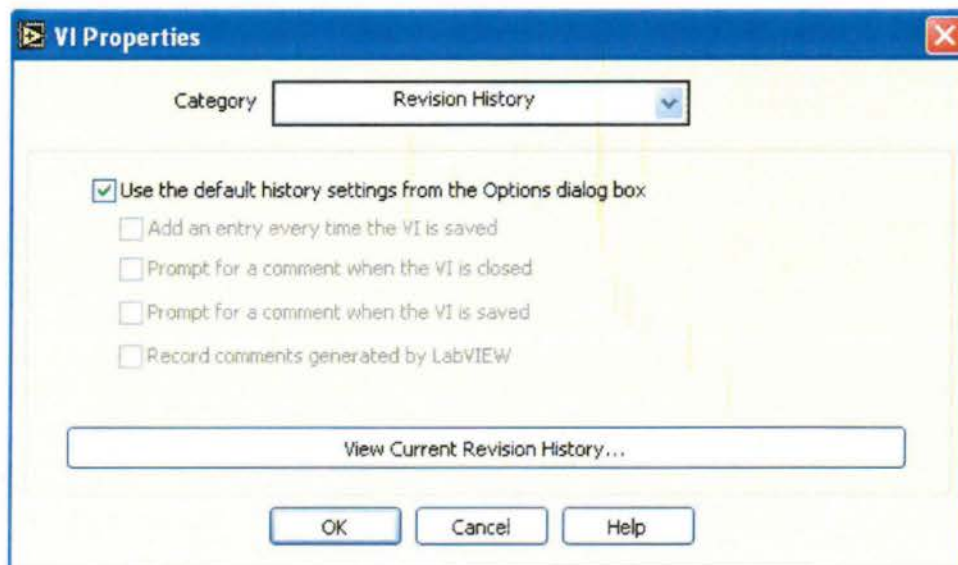
Στη λειτουργία περιγραφής κειμένου VI έχουμε την δυνατότητα να δώσουμε την γενική περιγραφή λειτουργίας του VI και περισσότερες πληροφορίες προς τους χρήστες (εικ 1.20).



Εικόνα 1.20: Παράθυρο περιγραφής κειμένου VI

### ➤ Λειτουργία Καταγραφής Ιστορικού (Revision History)

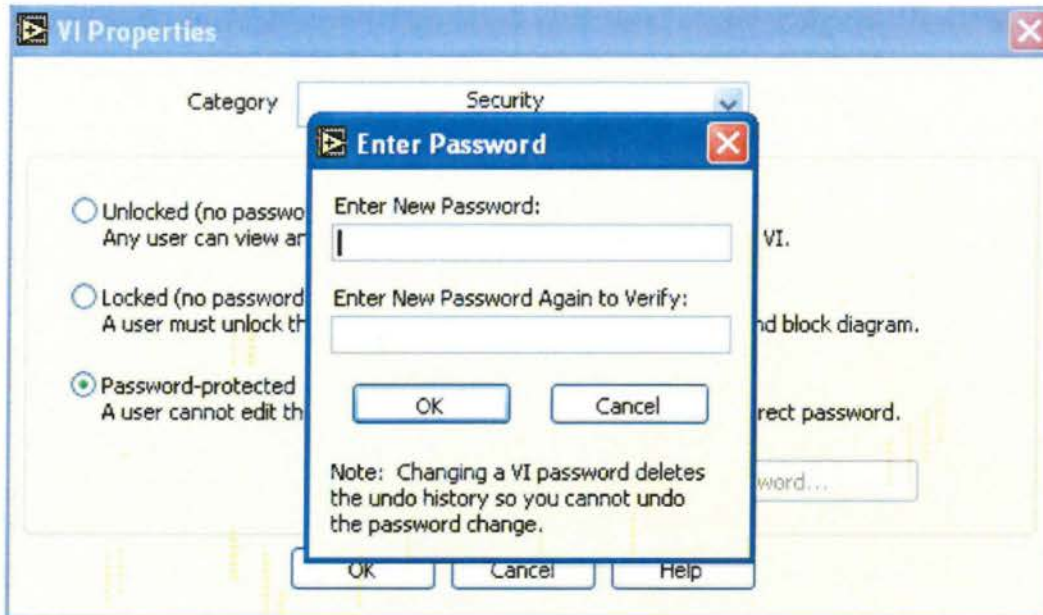
Σε αυτή την λειτουργία ορίζουμε πόσο συχνά θέλουμε να γίνεται καταγραφή του ιστορικού (εικ 1.21).



Εικόνα 1.21: Παράθυρο καταγραφής ιστορικού VI

➤ **Λειτουργία Ασφαλείας (Security)**

Το Labview μας παρέχει την δυνατότητα χρήσης κωδικού πρόσβασης για την χρήση του VI με την εντολή **Password-protected** (εικ 1.22).

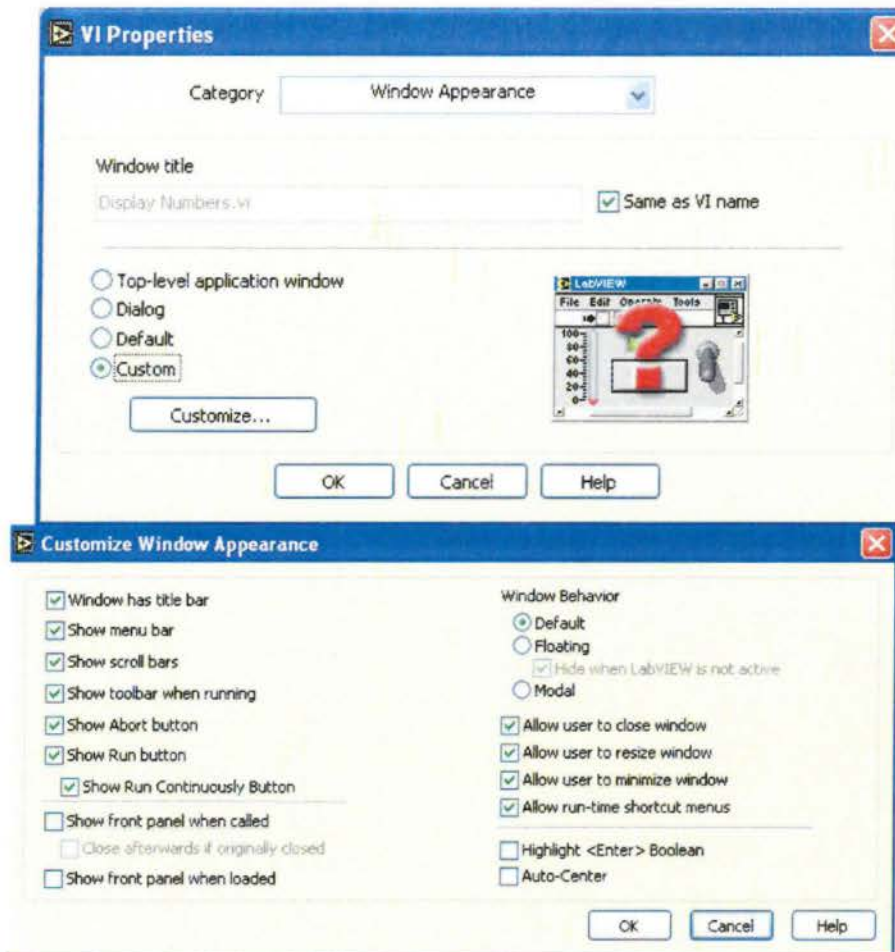


Εικόνα 1.22: Παράθυρο λειτουργίας ασφαλείας VI

➤ **Λειτουργία Εμφάνισης Παραθύρου (Window Appearance)**

Με αυτή την λειτουργία μπορούμε να δηλώσουμε τον τρόπο εμφάνισης του εκτελέσιμου VI. Για παράδειγμα να εμφανίζεται ή να αποκρύπτεται το μενού επιλογών του Labview. Πιο συγκεκριμένα επιλέγοντας την κατηγορία **Custom** όπως φαίνεται στην εικόνα που ακολουθεί (εικ 1.23) μπορούμε να διαμορφώσουμε την εμφάνιση του εκτελέσιμου VI επιλέγοντας τα στοιχεία που θέλουμε να εμφανίσουμε στο παράθυρο του VI.

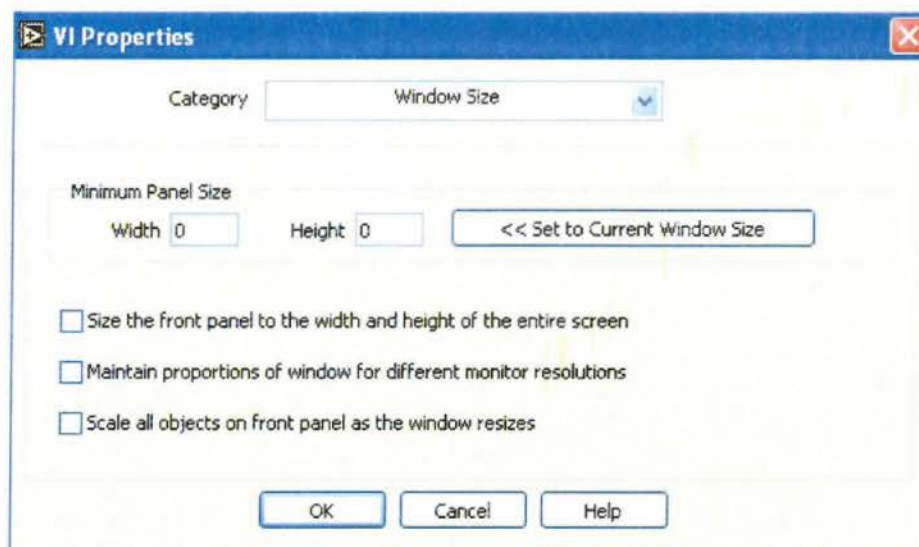
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.23: Ιδιότητες παράθυρο εμφάνιση VI

### ➤ Λειτουργία Αλλαγής Διαστάσεων Παραθύρου (Window Size)

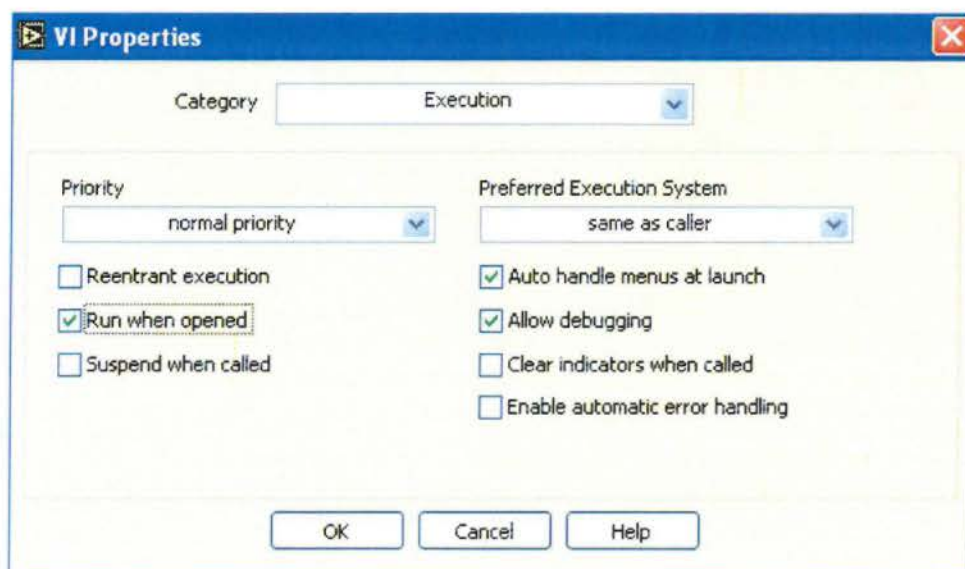
Σε αυτή την λειτουργία έχουμε την δυνατότητα να τροποποιήσουμε το πλάτος αλλά και το ύψος του παραθύρου εμφάνισης του VI (εικ 1.24).



Εικόνα 1.24: Τροποποίηση διαστάσεων παραθύρου εμφάνισης VI

#### ➤ Τρόπος Εκτέλεσης της Εφαρμογής VI (Execution)

Στην λειτουργία εκτέλεσης της εφαρμογής VI μπορούμε να επιλέξουμε τον τρόπο εκτέλεσης του, όταν αυτό καλείται. Δηλαδή έχουμε την δυνατότητα, να θέτουμε σε αυτόματη εκκίνηση το VI όταν το επιλέξουμε (εικ 1.25).

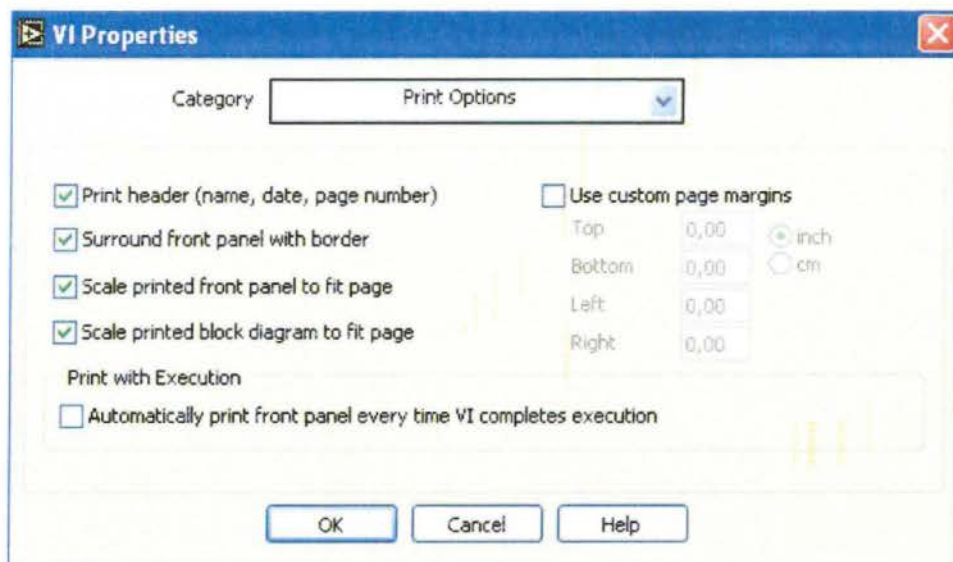


Εικόνα 1.25: Ιδιότητες τρόπου εκτέλεσης VI

#### ➤ Λειτουργία Εκτύπωσης VI (Print Options)

Αν θέλουμε να αλλάξουμε τα στοιχεία, τα όρια, καθώς και οτιδήποτε έχει να κάνει με την εκτύπωση ενός αρχείου VI αυτή είναι η λειτουργία που πρέπει να επιλέξουμε.(εικ 1.26).





Εικόνα 1.26: Τρόπος εκτύπωσης VI

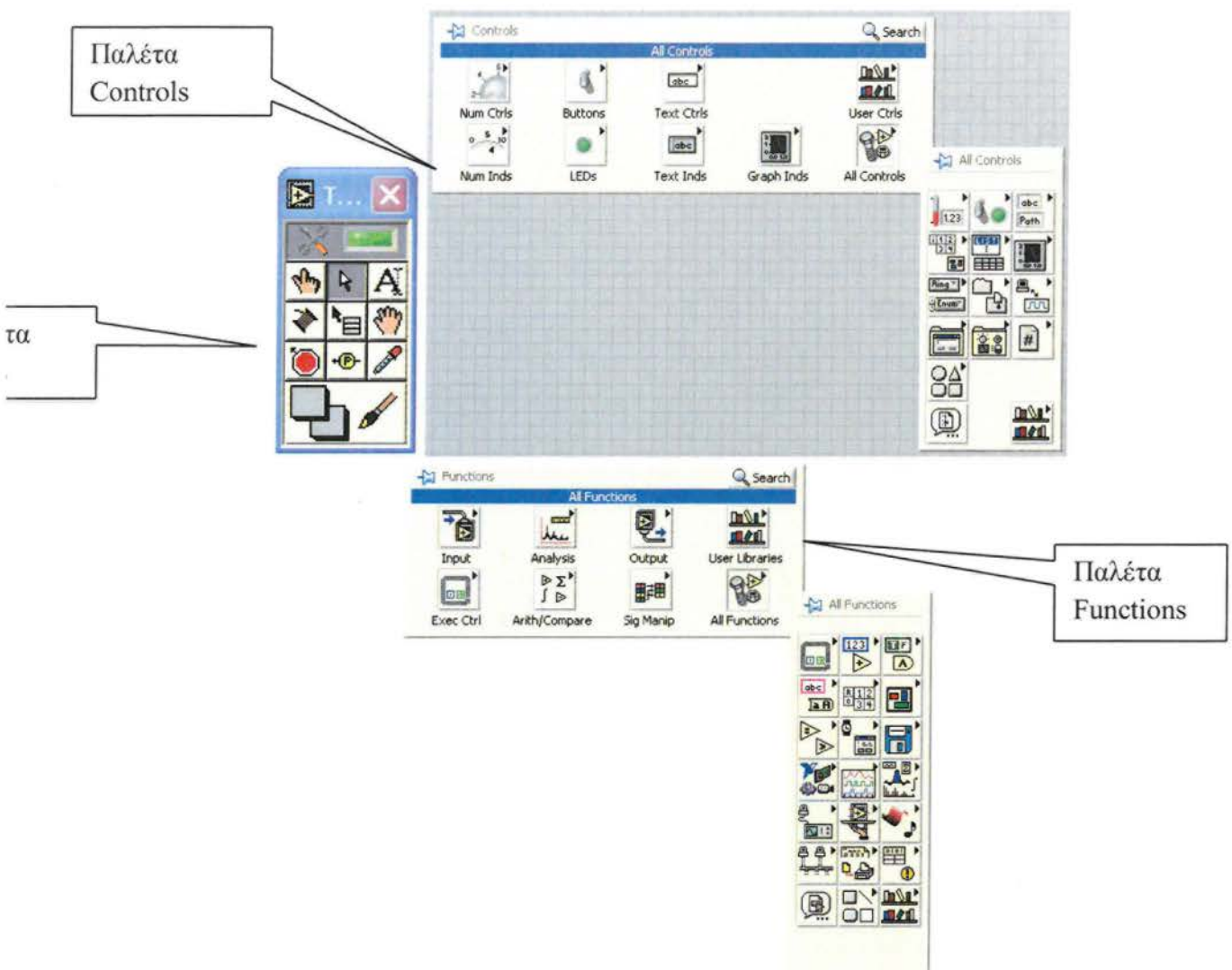
## 1.6 Εξερεύνηση Παλετών

Το **Labview** μας δίνει την δυνατότητα χρήσης τριών διαφορετικών γραφικών παλετών για να γίνεται πιο εύκολα η σχεδίαση αλλά και η λειτουργία των VIs (εικ 1.27).

Οι παλέτες είναι οι ακόλουθες:

1. Παλέτα Tools (Εργαλείων).
2. Παλέτα Controls (Αντικειμένων).
3. Παλέτα Functions (Συναρτήσεων/Λειτουργιών).

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.27: Απεικόνιση Παλετών

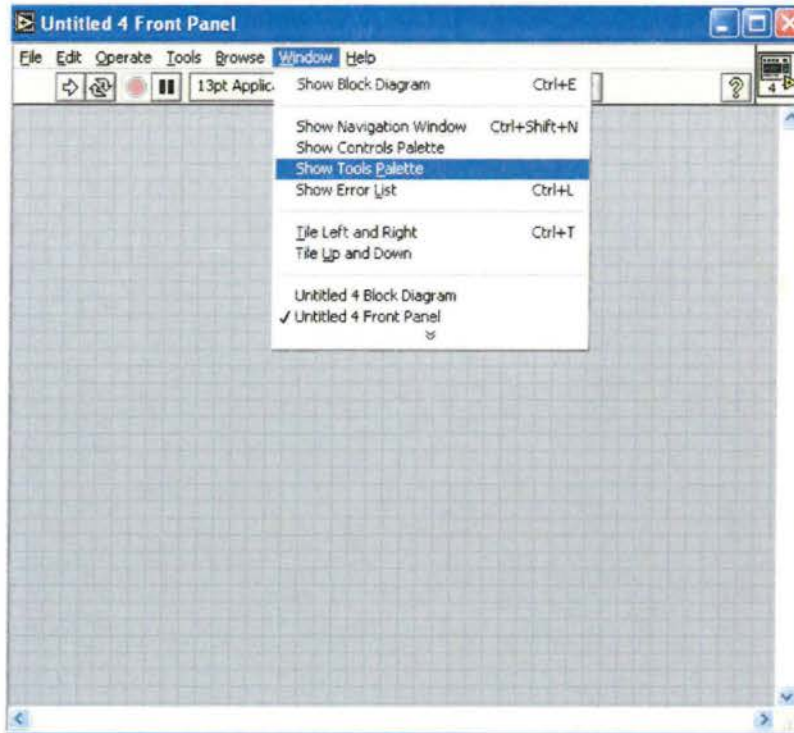
Στην συνέχεια θα ακολουθήσει περιγραφή των παλετών καθώς και επεξήγηση των εργαλείων που διαθέτουν.

### 1.6.1 Παλέτα εργαλείων (Tools)

Με την χρήση αυτών των εργαλείων από την παλέτα **Tools** μπορούμε να σχεδιάσουμε και να αλλάξουμε τα VIs. Για να εμφανίσουμε την συγκεκριμένη παλέτα, επιλέγουμε **Show Tool Palette** από το μενού **Window** (εικ 1.28). Επιλέγοντας ένα εργαλείο ο κέρσορας παίρνει το ανάλογο σχήμα. Ακουμπώντας με το ποντίκι οποιοδήποτε εργαλείο από την παλέτα πάνω από subVIs ή σε λειτουργίες του **block diagram**, βλέπουμε την βοήθεια (**On-Line**) για το εν λόγω αντικείμενο. Χρησιμοποιούμε τα εργαλεία για να εκτελέσουμε συγκεκριμένες

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

λειτουργίες. Την παλέτα μπορούμε να την μεταφέρουμε σε οποιοδήποτε σημείο της οθόνης επιθυμούμε.



Εικόνα 1.28: Εμφάνιση παλέτας εργαλείων

Επεξήγηση των εργαλείων της παλέτας



### **Automatic Tool Selection:**

Δίνει την δυνατότητα της αυτόματης εναλλαγής εργαλείων κατά το σχεδιασμό.



### **Operating:**

Με αυτό το κουμπί μπορούμε να αλλάζουμε τιμές στα στοιχεία **controls** και **indicators**. Επιπλέον να θέτουμε σε λειτουργία τα κουμπιά, τους διακόπτες και άλλα αντικείμενα.



### **Positioning:**

Επιλέγει, μετακινεί και αλλάζει το μέγεθος των αντικειμένων.



### **Labeling:**

Επιλέγοντας το μπορούμε να γράψουμε πάνω στο κενό χώρο είτε του **block diagram** είτε του **front panel**.



### **Wiring:**

Συνδεεί τα αντικείμενα στο **block diagram**.



**Menu:**

Εμφανίζει το **sort cut** μενού του αντικειμένου.



**Scrolling:**

Μας επιτρέπει να μετακινούμε το παράθυρο, χωρίς να κάνουμε χρήση των μαπαρών ολίσθησης που βρίσκονται στο κάτω μέρος αλλά και στα δεξιά του παραθύρου.



**Breakpoint:**

Σταματάει την λειτουργία VIs, αντικειμένων, συνδέσεων, αλλά και βρόγχων. Είναι χρήσιμο όταν θέλουμε να σταματήσουμε την λειτουργία κάποιου στοιχείου για λίγο.



**Probe:**

Έλεγχος των δεδομένων που μεταφέρονται σε συγκεκριμένη σύνδεση. Συνήθως χρησιμοποιείται όταν το VI δεν δίνει τα αναμενόμενα αποτελέσματα και θέλουμε να ελέγξουμε σε ποιο σημείο βρίσκονται οι λανθασμένες τιμές.



**Color copy:**

Αντιγράφει το χρώμα που θα επικολληθεί με το εργαλείο **Color**.



**Color:**

Δίνει χρώμα στο αντικείμενο ή το φόντο.

## 1.6.2 Παλέτες Αντικειμένων (Controls) και Λειτουργιών (Functions)

Οι παλέτες αυτές αποτελούνται από υποπαλέτες, καθεμιά από τις οποίες εσωκλείει μέσα της πλήθος από προγραμματιζόμενα αντικείμενα/εικόνες που είναι απαραίτητα για την κατασκευή των VIs. Η πρόσβαση σε αυτά τα αντικείμενα γίνεται με την ενεργοποίηση της εικόνας της κάθε υποπαλέτας. Σε περίπτωση, που



θέλουμε να επιλέγουμε από διάφορες υποπαλέτες, μας δίνεται η δυνατότητα να τις καρφώσουμε, (εικ 1.28) ώστε να παραμείνουν στο χώρο του **front panel** ή του **block diagram**.

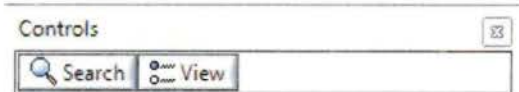


Εικόνα 1.28: Απεικόνιση “καρφώματος” παλετών

### 1.6.2.1 Παλέτα Αντικειμένων (Controls)

Στο **front panel** μπορούμε να βρούμε είτε **controls** είτε **indicators** να συνυπάρχουν μαζί. Τα **controls** είναι εξαρτήματα που αντιπροσωπεύουν τυπικές συσκευές εισόδου και μπορούμε να τις εντοπίσουμε σε ένα κοινό όργανο, όπως για παράδειγμα διακόπτες και κουμπιά. Είναι κατάλληλα όταν θέλουμε να παρέχουμε ή να προβάλλουμε δεδομένα.

Αντίθετα τα **indicators** αντιπροσωπεύουν απεικονίσεις εξόδου, που δείχνουν τα δεδομένα που αποκτά ή παράγει το VI. **Indicators** μπορούμε να θεωρήσουμε τα γραφήματα, τα θερμόμετρα, αλλά και τους μετρητές. Όταν τα τοποθετήσουμε στο **front panel**, το **Labview** δημιουργεί ταυτόχρονα ένα αντίστοιχο στοιχείο στο **block diagram**, το τερματικό στοιχείο.

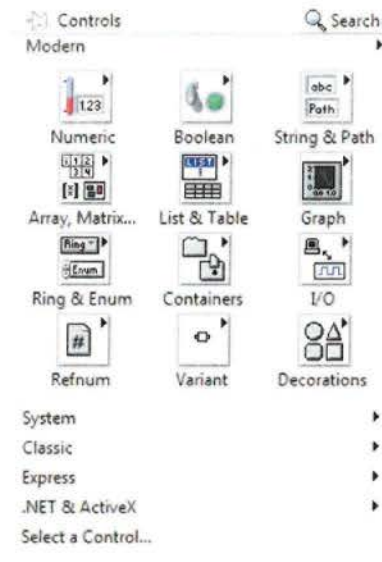


Παρατήρηση: Η διαφορά που έχουν τα τερματικά στοιχεία των **controls** από αυτά των **indicators** είναι ότι αυτά έχουν παχύ περίγραμμα ενώ αντίθετα αυτά των **indicators** έχουν λεπτό.

Τα χαρακτηριστικά των αντικειμένων στο **front panel** και των τερματικών στοιχείων στο **block diagram** μπορούν να αλλάξουν με τις εντολές που διαθέτει το **Pop-Up** μενού. Οι εντολές για να επεξεργαστούμε τα αντικείμενα και γενικότερα το VI βρίσκονται στα **Pull Down** μενού, που βρίσκονται στην κορυφή του παραθύρου.

Αυτή η παλέτα μας παρέχει την δυνατότητα να τοποθετούμε **controls** και **indicators** αντικείμενα στο **front panel**, επιπλέον διαιρείται σε συγκεκριμένα υποπαλέτες που περιέχουν καθορισμένους τύπους αντικειμένων (εικ 1.29). Εάν η παλέτα δεν είναι ορατή επιλέγουμε **Show Controls Palette** από το μενού **Window**. Διαφορετικά μπορούμε να εμφανίσουμε την παλέτα αντικειμένων πατώντας το δεξί πλήκτρο από το ποντίκι οπουδήποτε στο χώρο του **front panel**.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.29: Απεικόνιση control palette

Παρατήρηση: Η παλέτα των αντικειμένων είναι διαθέσιμη μόνο όταν το **front panel** είναι ενεργό.

Οι παλέτες που ακολουθούν είναι οι πιο κοινές και βρίσκονται στην υποπαλέτα **Modern**.



### Παλέτα Numeric.

Διαθέτει controls και indicators για αριθμητικά δεδομένα.



### Παλέτα Boolean.

Διαθέτει controls και indicators για λογικά / ψηφιακά δεδομένα.



### Παλέτα String & Path.

Διαθέτει controls και indicators για κείμενο υπό μορφή ASCII.



### Παλέτα Array, Matrix & Cluster.

Διαθέτει controls και indicators για ομαδοποίηση δεδομένων.



### Παλέτα List & Table.

Διαθέτει controls και indicators για την δημιουργία λιστών και πινάκων.



### Παλέτα Graph.

Διαθέτει indicators για γραφική απεικόνιση δεδομένων.



**Παλέτα Ring & Enum.**

Διαθέτει controls και indicators για δημιουργία μενού εναλλακτικών επιλογών.



**Παλέτα Containers.**

Δίνει την δυνατότητα εισαγωγής μπαρών ολίσθησης στο front panel.



**Παλέτα Refnum.**

Διαθέτει controls και indicators για επεξεργασία αρχείων.



**Παλέτα Variant**

Μπορούμε να εισάγουμε στοιχεία Variant.



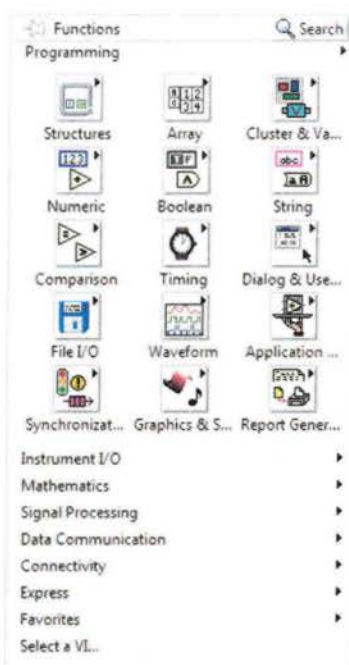
**Παλέτα Decorations.**

Διαθέτει γραφικά για σχεδίαση περιγραμμάτων στο front panel.

### 1.6.2.2 Παλέτα Λειτουργιών (Functions)

Για την κατασκευή του **block diagram** είναι απαραίτητη η χρήση της παλέτας λειτουργιών (εικ. 1.30). Αντίστοιχα με την προηγούμενη παράγραφο και αυτή η παλέτα υποδιαιρείται σε υποπαλέτες. Αν η παλέτα δεν είναι ορατή, επιλέγουμε το **Functions Palette** από το μενού **View** ή μπορούμε να ανοίξουμε την παλέτα **Function** πατώντας οπουδήποτε στο κενό χώρο του **block diagram**.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.30: Απεικόνιση functions palette

Παρατήρηση: Η παλέτα λειτουργιών είναι διαθέσιμη μόνο όταν το **block diagram** είναι ενεργό.



### Παλέτα Structures.

Διαθέτει προγραμματιστικές δομές, όπως While & For Loops.



### Παλέτα Array.

Με αυτή την παλέτα μπορούμε να επεξεργαστούμε πίνακες.



### Παλέτα Cluster & Variant.

Διαθέτει λειτουργίες για την επεξεργασία δομών από αναμειγυρισμένα στοιχεία.



### Παλέτα Numeric.

Διαθέτει λειτουργίες αριθμητικές, λογαριθμικές, τριγωνομετρικές.



### Παλέτα Boolean.

Διαθέτει VIs για λογικές πράξεις.



### Παλέτα String.

Διαθέτει VIs για επεξεργασία κειμένου σε μορφή ASCII.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



### Παλέτα Comparison.

Διαθέτει λειτουργίες για τη σύγκριση στοιχείων, τα οποία μπορεί να είναι αριθμητικά, λογικά, χαρακτήρες.



### Παλέτα Time.

Χρησιμοποιείται για χρονισμό.



### Παλέτα File I/O.

Χρησιμοποιείται για καταχώρηση δεδομένων και επεξεργασία αρχείων.



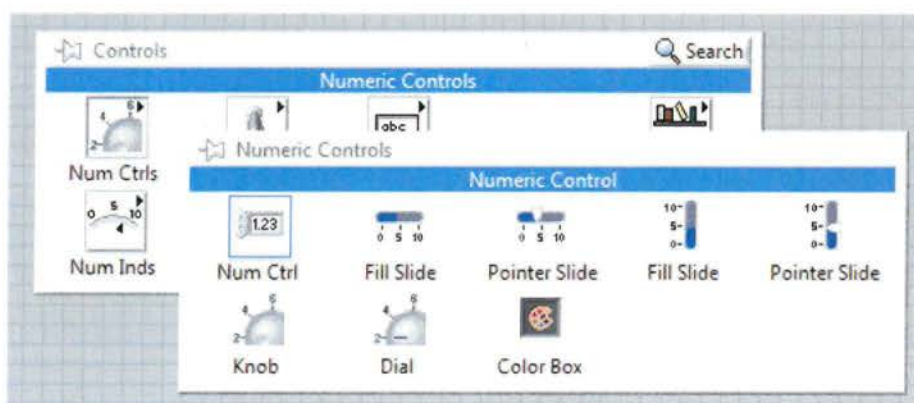
### Παλέτα Waveform.

Περιέχει εντολές για χρήση γραφημάτων.

Σε αυτό το σημείο έχοντας αναλύσει τις βασικότερες λειτουργίες των παλετών είμαστε σε θέση να δημιουργήσουμε ένα παράδειγμα σύγκρισης δύο αριθμών.

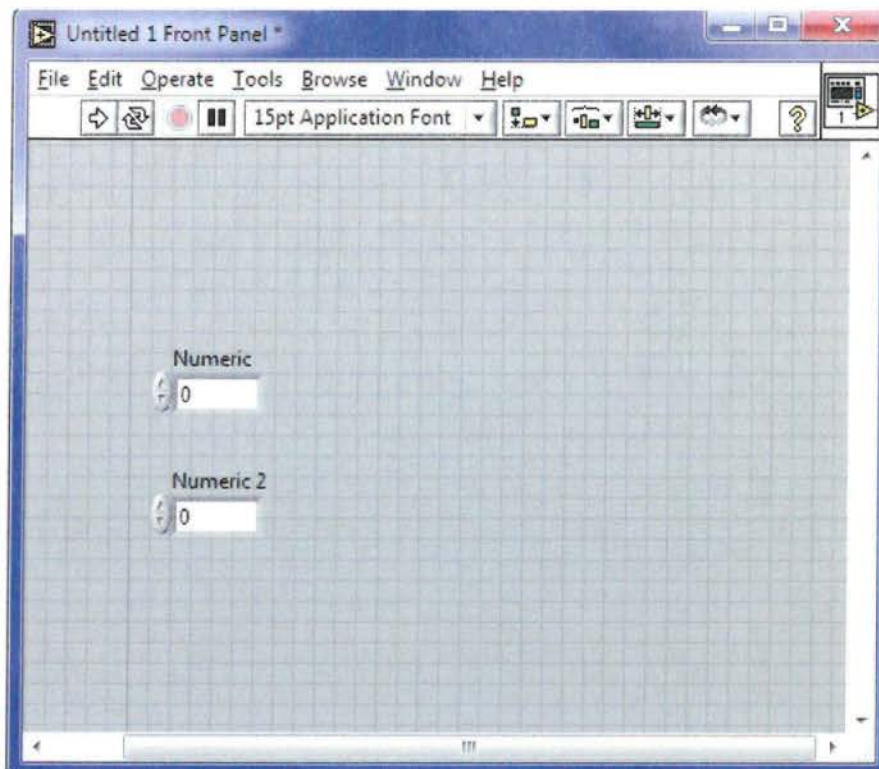
### Βήμα 1:

Ενώ έχουμε ανοιχτό το παράθυρο του **front panel** από την παλέτα **Controls** επιλέγουμε δύο αριθμητικά αντικείμενα (**controls**) (εικ. 1.31) και τα τοποθετούμε όπως φαίνεται στο παράθυρο που ακολουθεί (εικ. 1.32). Θα χρησιμοποιήσουμε τα στοιχεία **controls** για να εισάγουμε τους αριθμούς που θα συγκρίνουμε.



Εικόνα 1.31 : Επιλογή Controls

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

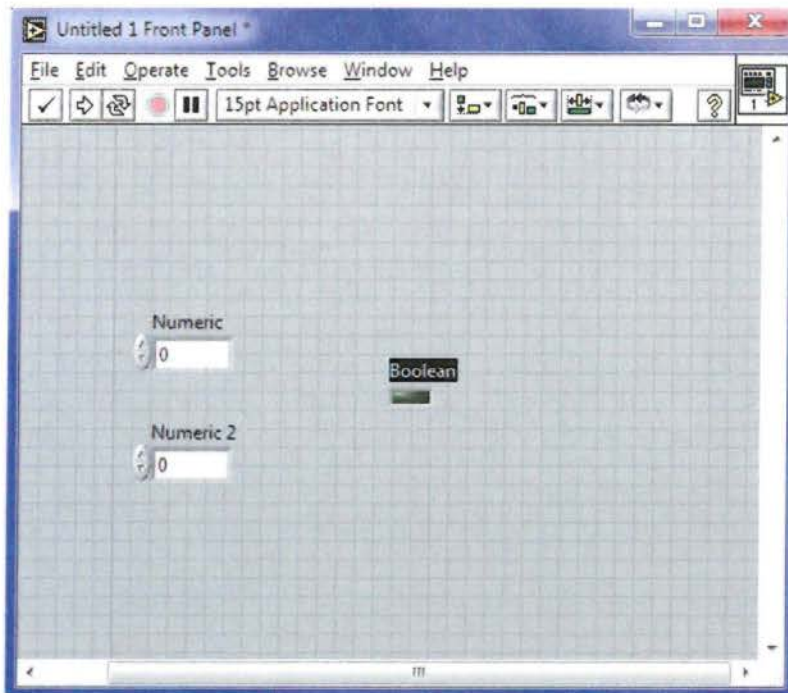


Εικόνα 1.32 : Τοποθέτηση Controls στο Front Panel

### Βήμα 2:

Στην συνέχεια επιλέγουμε από την υποπαλέτα λογικών λειτουργιών στοιχείο **Square Led** και το τοποθετούμε στο παράθυρο του **front panel**.(εικ. 1.33)

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.33: Τοποθέτηση Square Led στο Front Panel

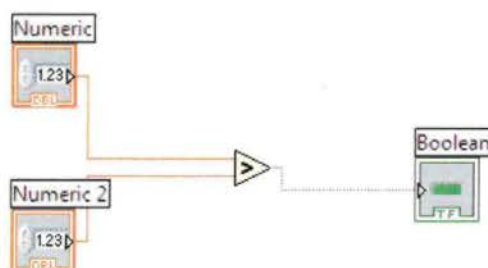
Παρατηρούμε ότι με την κατασκευή κάθε αντικειμένου **control/indicator** αυτόματα δημιουργείτε ο αντίστοιχος ακροδέκτης στο **block diagram**. Τα σύμβολα στους ακροδέκτες υποδηλώνουν τον τύπο των δεδομένων που λαμβάνει. Το σύμβολο **DBL**, για παράδειγμα σημαίνει ότι χρησιμοποιούμε αριθμό διπλής ακρίβειας, ενώ ο τύπος **TF** στο στοιχείο **Led** δηλώνει **ON-OFF** αντικείμενα.

### Βήμα 3:

Ενώ βρισκόμαστε στο **block diagram** επιλέγουμε από την υποπαλέτα συγκρίσεων του μενού λειτουργιών την λειτουργία **Greater?**. Η λειτουργία αυτή συγκρίνει δύο αριθμούς και επιστρέφει **TRUE** αν ο πρώτος αριθμός (**Numeric**) είναι μεγαλύτερος από τον δεύτερο (**Numeric 2**).

### Βήμα 4:

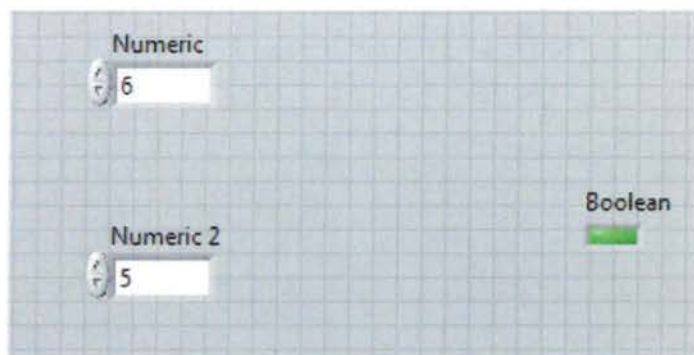
Από την παλέτα των εργαλείων επιλέγουμε το εργαλείο σύνδεσης (**Wiring**) για να συνδέσουμε τα εικονίδια μεταξύ τους. (εικ. 1.34)



Εικόνα 1.35: Σύνδεση εικονιδίων με την χρήση του εργαλείου Wiring

### Βήμα 5:

Τρέχουμε το πρόγραμμα έχοντας εισάγει αριθμούς στα αντικείμενα του **front panel**. Παρατηρούμε την ορθή λειτουργία του παραδείγματος ανάβοντας το **Led**. (εικ. 1.36)



Εικόνα 1.36 : Ορθή λειτουργία του προγράμματος

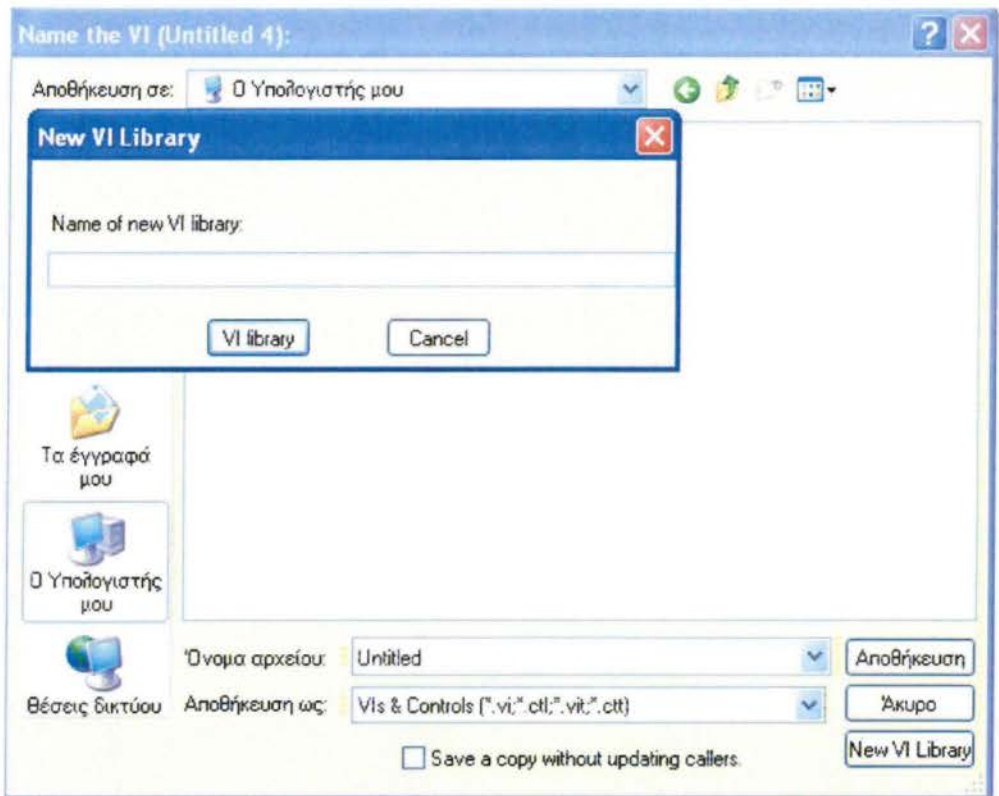
### Δημιουργία βιβλιοθήκης από VIs στο Block Diagram.

Για να δημιουργήσουμε μια καινούργια βιβλιοθήκη από VIs, διαλέγουμε από το μενού **File** στο πάνω μέρος του παραθύρου την επιλογή **Save as...** και πατάμε το κουμπι **New LLB**.

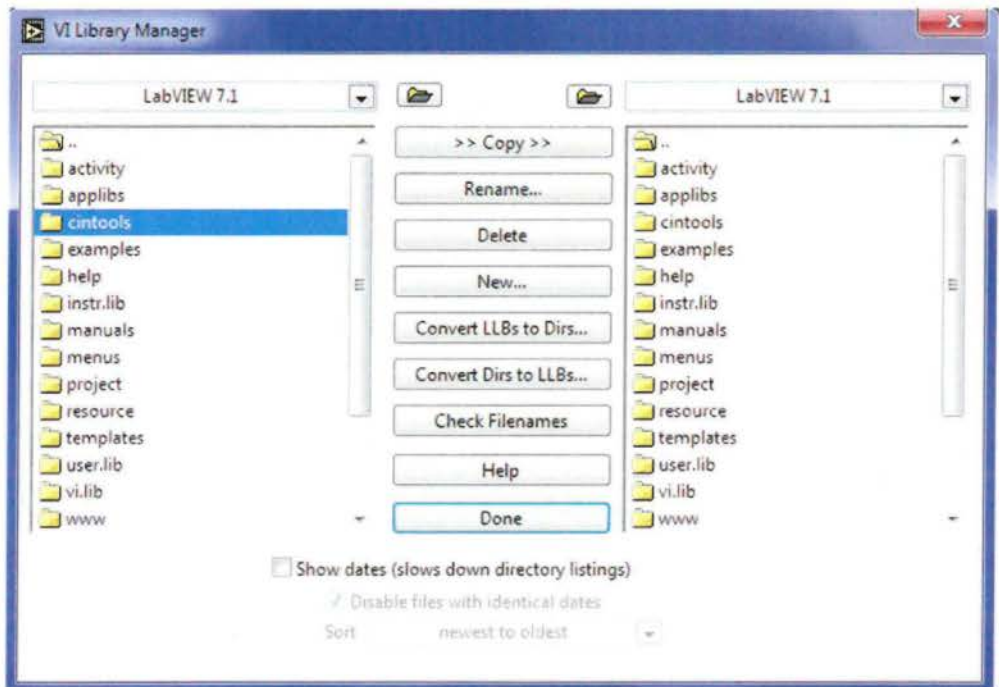
Στο διαλογικό παράθυρο που εμφανίζεται (εικ 1.37) δίνουμε όνομα στην καινούργια μας βιβλιοθήκη και στην συνέχεια πατάμε το κουμπι **LLB**. Δεν χρειάζεται να γράψουμε την κατάληξη **.llb** στην βιβλιοθήκη μας γιατί το **Labview** την προσθέτει αυτόματα. Για να διαγράψουμε μια βιβλιοθήκη από το μενού **Tools** διαλέγουμε την επιλογή **VI Library Manager**. Στο παράθυρο που ανοίγει στην οθόνη επιλέγουμε την βιβλιοθήκη που θέλουμε να αφαιρέσουμε και πατώντας το δεξί πλήκτρο του ποντικιού μας επιλέγουμε το **Delete**. (εικ 1.38)



**ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.**



**Εικόνα 1.37: Δημιουργία βιβλιοθήκης από VIs στο block diagram**



**Εικόνα 1.38: Διαγραφή βιβλιοθήκης**

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Γιατί όμως να χρησιμοποιούμε βιβλιοθήκες από VIs;

Στην συνέχεια ακολουθούν τρεις καλοί λόγοι σε αυτή την ερώτηση:

1. Με τα VIs βιβλιοθήκης έχουμε την δυνατότητα να χρησιμοποιήσουμε έως 255 χαρακτήρες για να δώσουμε όνομα στα VIs μας.
2. Τα VIs βιβλιοθήκης συμπιέζουν τα VIs και εξοικονομούν χώρο στο σκληρό δίσκο.
3. Έχοντας πολλά VIs μέσα σε ένα μοναδικό αρχείο είναι πιο εύκολη η μεταφορά αρχείων μεταξύ υπολογιστών.

Παρατηρήσεις:

- Τα VIs βιβλιοθήκης δεν είναι ιεραρχικά. Δεν γίνεται, δηλαδή, να υπάρχει ένα VI βιβλιοθήκης μέσα σε ένα άλλο.
- Η αποθήκευση VIs απευθείας στο δίσκο είναι πιο γρήγορη από ότι μέσα σε ένα VI βιβλιοθήκης.

### 1.6.3 Μορφοποίηση Παλετών

Με το **Labview** μπορούμε να μορφοποιήσουμε το μενού της παλέτας λειτουργιών (**functions**) και αντικειμένων (**controls**) δημιουργώντας νέες υποπαλέτες ή αλλάζοντας τις ήδη υπάρχουσες. Πατώντας το αριστερό πλήκτρο από το ποντίκι μας επάνω στο πλήκτρο λειτουργιών όπως δείχνει η εικόνα παρακάτω (εικ 1.39)



Εικόνα 1.39: Απεικόνιση Πλήκτρου Λειτουργιών

Από το πλαίσιο επιλογών που εμφανίζεται μπορούμε να ορίσουμε τον τύπο παλέτας στον οποίο επιθυμούμε να εργαστούμε, καθώς και τον τρόπο που θέλουμε να εμφανίζονται τα εργαλεία στην οθόνη μας. Είτε δηλαδή σε μορφή εικονιδίων είτε κειμένων κ.α. (εικ 1.40)

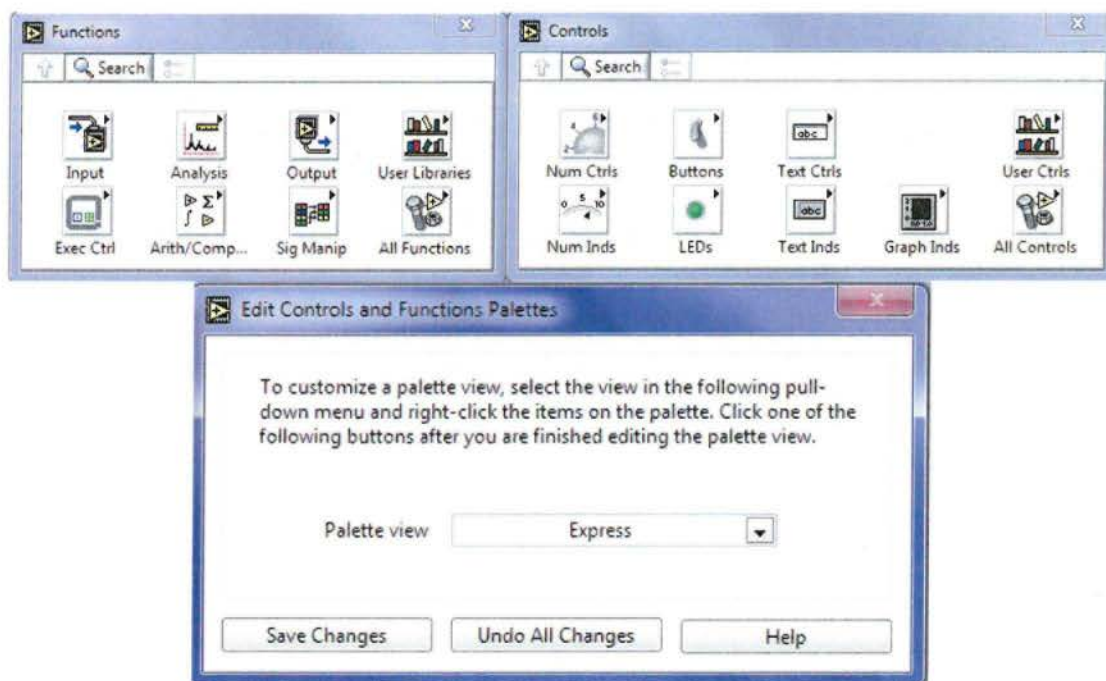
Αν θέλουμε να τροποποιήσουμε τις παλέτες θα πρέπει να επιλέξουμε από το μενού **Tools** στο πάνω μέρος της οθόνης, την εντολή **Advanced** και στην συνέχεια την εντολή **Edit Palette Views**. Στην οθόνη εμφανίζεται ένα παράθυρο όπως φαίνεται στην εικόνα 1.41 με το οποίο μπορούμε να διαμορφώσουμε τις παλέτες μας προσθέτοντας ή αφαιρώντας ολόκληρες βιβλιοθήκες ή ανεξάρτητα VIs.

Σε περίπτωση που θέλουμε να προσθέσουμε, να διαγράψουμε ή να μετανομάσουμε μια βιβλιοθήκη ή ένα VI, επιλέγουμε με δεξί πάτημα του πλήκτρου από το ποντίκι επάνω στο εικονίδιο της υποπαλέτας ή του VI που θέλουμε να αλλάξουμε και επιλέγουμε τις εντολές που είναι απαραίτητες.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.40: Ορισμός τρόπου εμφάνισης παλέτας



Εικόνα 1.41: Τροποποίηση Παλέτας

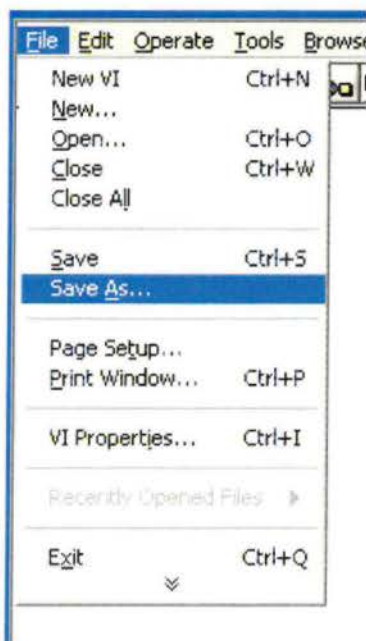


## 1.7 Ανοίγοντας VIs

Ανοίγουμε ένα VI στη μνήμη επιλέγοντας **Open** από το μενού **File** στο πάνω μέρος του παραθύρου. Βιβλιοθήκες από VIs αλλά και τα VIs εύκολα γίνονται αντιληπτά στο διαλογικό παράθυρο που εμφανίζεται, καθώς οι βιβλιοθήκες VIs μοιάζουν με φακέλους με την εικόνα VI γραμμένη επάνω τους, ενώ τα VIs διατηρούν την κατάληξη .vi.

Για να ανοίξουμε μια βιβλιοθήκη VI ενεργούμε με τον ίδιο τρόπο που θα ανοίγαμε έναν οποιοδήποτε φάκελο είτε επιλέγοντας την και ενεργοποιώντας το πλήκτρο OK είτε ενεργοποιώντας την με διπλό πάτημα από το ποντίκι. Καθώς φορτώνεται ο VI, ένα παράθυρο εμφανίζεται στην οθόνη και μας ενημερώνει για τους **subVIs** που φορτώνονται μαζί με το κύριο πρόγραμμα στη μνήμη.

Σε περίπτωση που το **Labview** δεν μπορεί αυτόματα να βρει ένα συγκεκριμένο **subVI**, ψάχνει σε όλους τους φακέλους που έχουμε προσδιορίσει στην ανίχνευση (**VI Search Path**). Μπορούμε να επιλέξουμε να αγνοήσουμε το **subVI**, θέτοντας ενεργό το **Ignore SubVI**. Η καταχώριση ενός VI μπορεί να γίνει σε ένα φάκελο ή βιβλιοθήκη από VIs επιλέγοντας **Save, Save As..., Save a Copy As...** από το μενού **File** (εικ 1.42)



Εικόνα 1.42: Αποθήκευση VI

## 1.8 Front Panel και Block Diagram

### 1.8.1 Front Panel

Προσθέτοντας αντικείμενα μπορούμε να δημιουργήσουμε το **front panel**. Τα αντικείμενα αυτά μπορεί να είναι είτε **controls** είτε **indicators**. Υπάρχουν διαθέσιμα πολλά είδη **controls** αλλά και **indicators**. Τοποθετούμε τα αντικείμενα αυτά στο παράθυρο γραφικών, επιλέγοντας τα μέσα από τις διάφορες υποπαλέτες των αντικειμένων. Σε περίπτωση που το μενού των γραφικών δεν είναι ορατό υπάρχουν οι ακόλουθοι δυο τρόποι για να εμφανιστεί:

- Να πατήσουμε το δεξί πλήκτρο του ποντικιού, οπουδήποτε στο παράθυρο των γραφικών.
- Να επιλέξουμε **Show Control Palette** από το μενού **Window** στο πάνω μέρος του παραθύρου.

Με τους αντίστοιχους τρόπους εμφανίζεται το μενού γραφικών στο **Block Diagram** με την μόνη διαφορά ότι επιλέγουμε **Show Functions Palette** από το μενού **Window**.

#### Αριθμητικά Controls Indicators (Numeric).

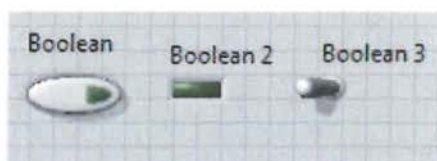
Οι δύο πιο συνηθισμένοι τύποι στο **Labview** είναι οι **digital control** και **digital indicator** όπως φαίνονται στην εικόνα που ακολουθεί (εικ 1.37). Για εισάγουμε κάποιο νούμερο, μπορούμε απλά να αυξομοιώσουμε τον αριθμό με τα πλήκτρα αυξομοίωσης ή να εισάγουμε τον αριθμό απευθείας, χρησιμοποιώντας το εργαλείο δείκτη (**Operating Tool**).



Εικόνα 1.43: Αριθμητικά Controls Indicators

#### Ψηφιακά Controls και Indicators (Boolean).

Η χρήση αυτών των αντικειμένων είναι για την καταχώριση και παρουσίαση των τιμών **Αλήθεια – Ψεύδους / ON – OFF** για αυτό τον λόγο εμφανίζονται στην οθόνη με την μορφή διακοπών, πλήκτρων και **Leds** (εικ 1.44)



Εικόνα 1.44: Ψηφιακά Controls και Indicators

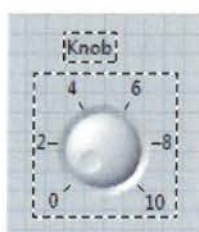
### 1.8.1.1 Επεξεργασία Αντικειμένων

Προκειμένου να γίνει η επεξεργασία των αντικειμένων απαραίτητη είναι η χρήση των εργαλείων από την παλέτα εργαλείων και εντολές από το **Pop-up** και **Pull-Down** μενού.

#### Επιλογή Αντικειμένων

Για να επιλέξουμε οποιοδήποτε αντικείμενο θέλουμε πατάμε το αριστερό πλήτρο του ποντικιού, χρησιμοποιώντας το εργαλείο θέσης (**positioning**). Με την επιλογή ενός αντικειμένου, το **Labview** δημιουργεί μια διακεκομμένη γραμμή η οποία το περικυκλώνει (εικ 1.45)

Η επιλογή των αντικειμένων μας δίνει πολλές επιλογές για να επεξεργαστούμε τα αντικείμενα. Μερικές από αυτές είναι να τα διαγράψουμε, να τα αντιγράψουμε, να τα μετακινήσουμε ή απλά να αλλάξουμε το μέγεθος τους.



Εικόνα 1.45: Απεικόνιση επιλεγμένου αντικειμένου

Σε περίπτωση που θέλουμε να επιλέξουμε περισσότερα του ενός αντικείμενα, το **Labview** μας δίνει την επιλογή να κρατάμε πατημένο το **Shift** για κάθε επιπλέον αντικείμενο που θέλουμε να επιλέξουμε. Εναλλακτικά μπορούμε να πατήσουμε το εργαλείο θέσης σε μια ανοιχτή περιοχή, στη συνέχεια να το σύρουμε έχοντας πατημένο το ποντίκι σχεδιάζοντας έτσι ένα παραλληλόγραμμο, ώστε τα αντικείμενα που θέλουμε να επιλέξουμε να βρίσκονται μέσα του. Όταν αφήσουμε το ποντίκι, το παραλληλόγραμμο εξαφανίζεται και εμφανίζονται τα επιλεγμένα αντικείμενα με τη διακεκομμένη γραμμή γύρω τους.

#### Μετακίνηση Αντικειμένων

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Για να μετακινήσουμε ένα αντικείμενο θα πρέπει πρώτα να το έχουμε επιλέξει με τον τρόπο που αναφέραμε παραπάνω. Έτσι αφού έχουμε επιλέξει το αντικείμενο το μετακινούμε χρησιμοποιώντας το εργαλείο θέσης, μετακινώντας το στην επιθυμητή θέση και κρατώντας πατημένο το ποντίκι.

Μας δίνεται η δυνατότητα να το μετακινήσουμε κάθετα ή παράλληλα έχοντας πατημένο το πλήκτρο **Shift**. Επιπλέον, για πιο ακριβής μετακίνηση μπορούμε να χρησιμοποιήσουμε τα βελάκια του πληκτρολογίου.

Με την μετακίνηση συνδεδεμένων αντικειμένων, η σύνδεση επιμηκύνεται αυτόματα. Οι επιμηκύνσεις συνδέσεων αντικειμένων δεν συνιστώνται γιατί συχνά προκαλούν λάθη στις συνδέσεις. Για να τις αφαιρέσουμε από το κύκλωμα επιλέγουμε την εντολή **Remove bad wires** από το μενού **Edit**.

### Αντιγραφή Αντικειμένων

Για να αντιγράψουμε το αντικείμενο που θέλουμε, αρχικά το μαρκάρουμε και στην συνέχεια διαλέγουμε **Copy** από το μενού **Edit**. Έπειτα, τοποθετούμε τον κέρσορα στο σημείο που θέλουμε να βρίσκεται το νέο μας αντικείμενο και διαλέγουμε **Paste** από το ίδιο μενού.

Το **Labview** σε ορισμένες εκδόσεις δεν επιτρέπει να αντιγράψουμε τερματικό στοιχείο από το **block diagram**. Σε περίπτωση όμως που επιτρέπεται, γίνεται πολύ εύκολα πηγαίνοντας στο **panel** και αντιγράφοντας το αντίστοιχο **control**.

### Αντικατάσταση Αντικειμένων

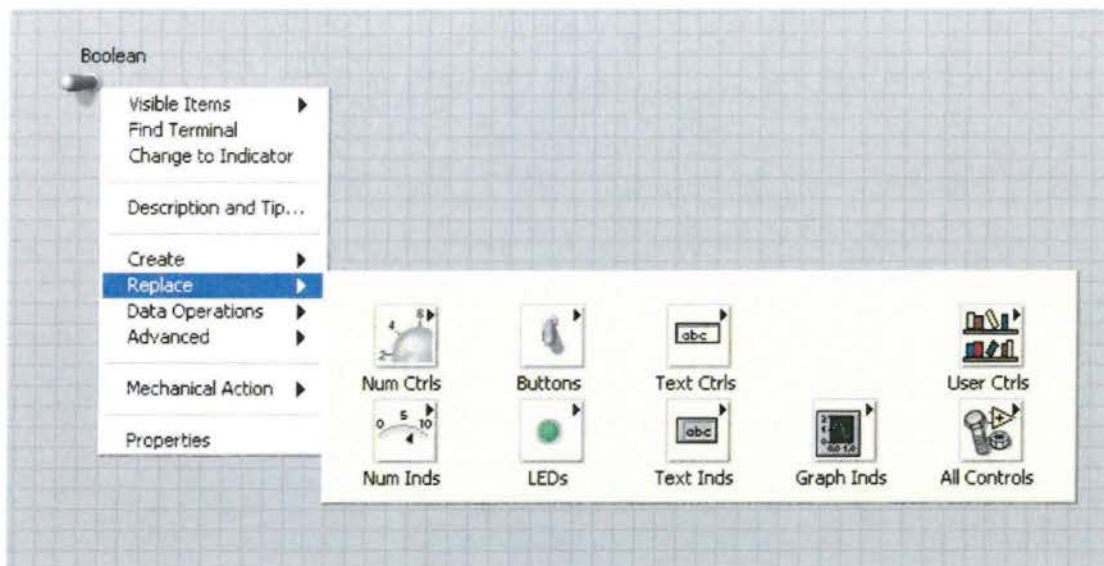
Η αντικατάσταση των αντικειμένων επιτυγχάνεται ανοίγοντας το **Short Cut** μενού του αντικειμένου που θέλουμε να αντικαταστήσουμε και επιλέγοντας την εντολή **Replace** (εικ 1.46). Από την παλέτα που εμφανίζεται επιλέγουμε ένα αντικείμενο το οποίο αυτόματα αντικαθιστά το αυθεντικό. Με την αντικατάσταση αυτή το **Labview** φροντίζει να αντικαταστήσει και τις πληροφορίες του αυθεντικού.

Πρέπει όμως να δώσουμε προσοχή στα κάτωθι:

- Κάποιες πληροφορίες δεν είναι δυνατόν να αντικατασταθούν και αυτό γιατί δεν έχουν όλα τα αντικείμενα τις ίδιες ιδιότητες.
- Επιπλέον, αντικαθιστώντας και τα τερματικά στοιχεία των αντικειμένων, οι συνδέσεις παραμένουν, αλλά επειδή κάποια αντικείμενα έχουν διαφορετικούς τύπους και δεδομένα κατά την αντικατάσταση, οι συνδέσεις “σπάνε”.



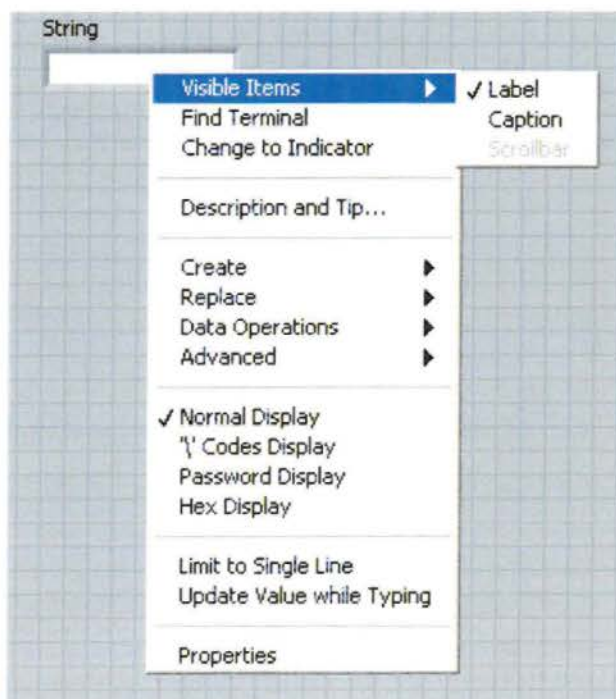
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.46: Αντικατάσταση αντικειμένου

### Απόκρυψη – Εμφάνιση στοιχείων Αντικειμένων

Για να αποκρύψουμε ή να εμφανίσουμε στοιχεία του αντικειμένου, όπως για παράδειγμα την επιγραφή και την επεξήγηση, ανοίγουμε το **Sort Cut** μενού και από την επιλογή **Visible Item** επιλέγουμε τα στοιχεία που θέλουμε να επεξεργαστούμε (εικ 1.47).

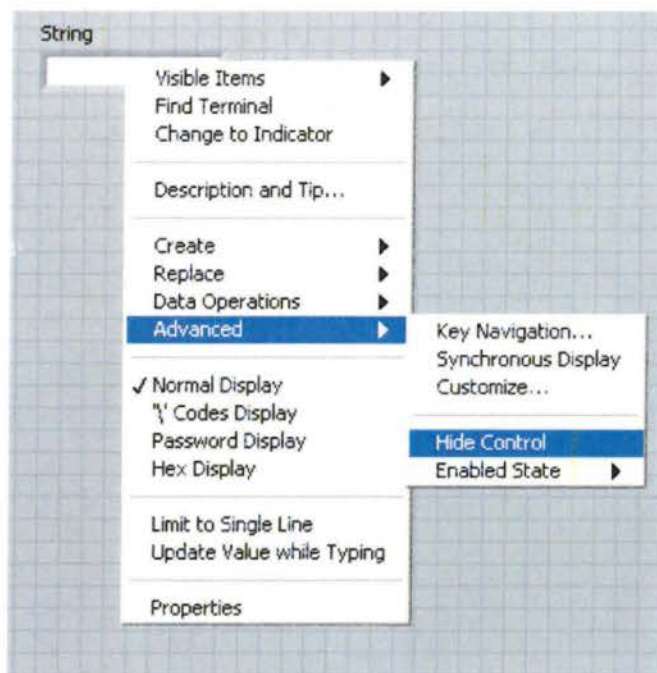


Εικόνα 1.47: Απόκρυψη – Εμφάνιση στοιχείων Αντικειμένων

Στο **front panel** υπάρχει η δυνατότητα να αποκρύψουμε ολόκληρο το αντικείμενο αν επιλέξουμε από το **Sort Cut** μενού την εντολή **Advance** και στην συνέχεια την επιλογή **Hide Control** όπως φαίνεται στην εικόνα 1.48. Για την

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

επανεμφάνιση του αντικειμένου πηγαίνουμε στο **block diagram** και από το **Sort Cut** μενού του αντίστοιχου τερματικού στοιχείου επιλέγουμε το **Show control**.



Εικόνα 1.48: Απόκρυψη αντικειμένου

### Διαγραφή Αντικειμένων

Αν επιθυμούμε να διαγράψουμε ένα αντικείμενο θα πρέπει να πατήσουμε την εντολή **Delete** από το μενού **Edit** ή να διαλέξουμε το αντικείμενο και να πατήσουμε το πλήκτρο **Delete** από το πληκτρολόγιο μας. Σε ορισμένες εκδόσεις του **Labview** δεν επιτρέπεται να διαγράψουμε τερματικό στοιχείο από το **block diagram**. Σε αυτή την περίπτωση, πρέπει να το διαγράψουμε από το **panel**.

### Δημιουργία Πλήκτρων Συντόμευσης

Το **Labview** μας παρέχει την δυνατότητα να δημιουργούμε πλήκτρα συντόμευσης για τα αντικείμενα που έχουμε χρησιμοποιήσει, για να χειριζόμαστε το **front panel** δίχως την βοήθεια του ποντικιού. Αυτό επιτυγχάνεται αν από το **Short Cut** μενού επιλέξουμε την εντολή **Advanced** και στην συνέχεια την εντολή **Key Navigation** (εικ 1.49). Στην συνέχεια από το παράθυρο διαλόγου που ανοίγει επιλέγουμε το πλήκτρο συντόμευσης (εικ 1.50).

Όταν το VI είναι σε λειτουργία με τα πλήκτρα που έχουμε επιλέξει ελέγχουμε τα **controls** με τους ακόλουθους τρόπους:

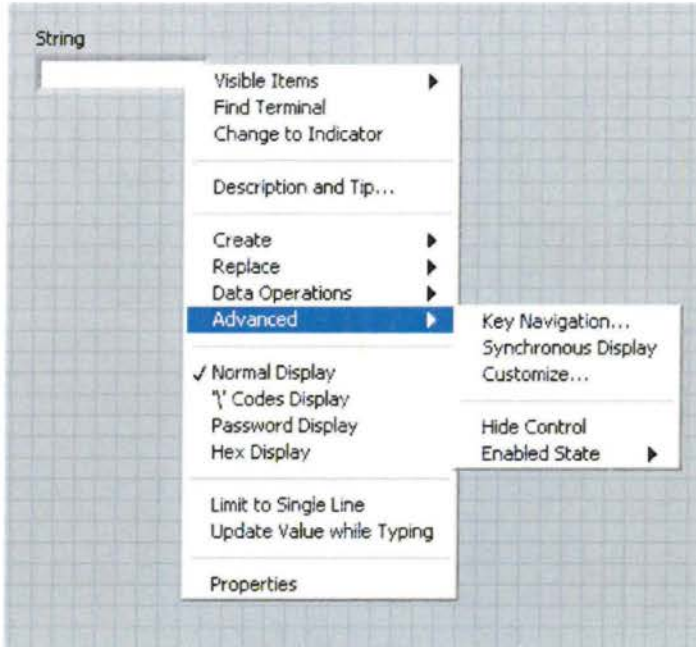
- σε περίπτωση που το **control** είναι **control** κειμένου, τότε το **Labview** εστιάζει το κείμενο για να γράψουμε
- ενώ εάν είναι λογικής κατάστασης, αλλάζει την λογική τιμή.

Την εντολή για να αλλάξουμε τις τιμές μπορούμε να την αναζητήσουμε και στο μενού **Edit** χρησιμοποιώντας την εντολή **Set Tabbing Order**. Σε αυτή την

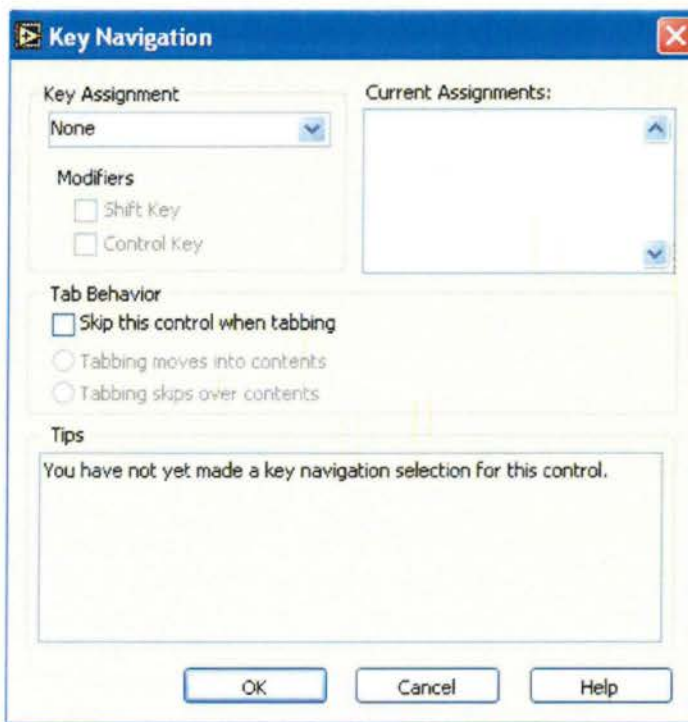
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

περίπτωση όμως χρησιμοποιούμε το πλήκτρο **Tab** για να επιλέξουμε το **control** και το **Enter** για να αλλάξουμε τις τιμές.

Παρατήρηση: Πλήκτρα συντόμευσης μπορούμε να κατασκευάσουμε μόνο για τα **controls**, και αυτό γιατί τα **indicators** είναι έξοδοι και δεν μπορούμε να τροποποιήσουμε τις τιμές τους, κάτι που είναι εφικτό στις εισόδους.



Εικόνα 1.49: Δημιουργία πλήκτρου συντόμευσης

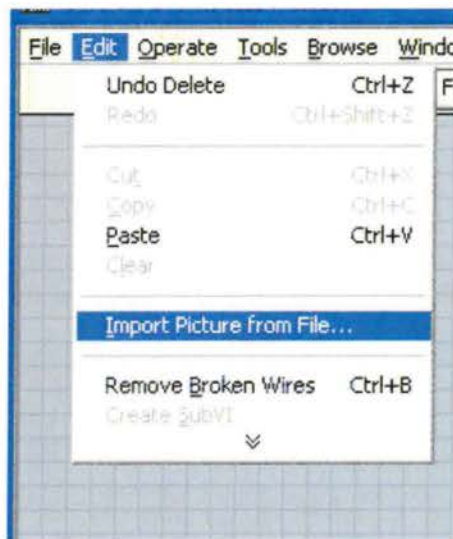


Εικόνα 1.50: Παράθυρο διαλόγου πλήκτρου συντόμευσης



### Εισαγωγή Εικόνας-Διακόσμηση Panel

Από την παλέτα διακόσμησης (**Decorations**), μπορούμε να διαλέξουμε διακοσμητικά παράθυρα και γραμμές για να κάνουμε πιο ευχάριστο στο μάτι το VI μας. Επιπλέον από το μενού **Edit** επιλέγοντας την εντολή **Import Picture From File** μπορούμε να εισάγουμε εικόνα από αρχείο και να την χρησιμοποιήσουμε για φόντο. Προσοχή όμως γιατί το **Labview** δεν υποστηρίζει όλους τους τύπους εικόνων! (εικ 1.51)



Εικόνα 1.51: Εισαγωγή εικόνας

## 1.8.2 Block Diagram

Στο **block diagram** κανείς θα συναντήσει **κόμβους (nodes)**, **ακροδέκτες (terminals)** και **καλώδια (wires)**. Για κάθε αντικείμενο που προσθέτουμε στο **front panel** έχουμε και έναν αντίστοιχο ακροδέκτη στο **block diagram**.

Οι κόμβοι είναι σημεία εκτέλεσης προγράμματος και χωρίζονται στις ακόλουθες τέσσερις κατηγορίες:

- **Λειτουργίες/Συναρτήσεις**
- **SubVIs**
- **Προγραμματιστικές δομές και**
- **CINs**

Οι **Λειτουργίες/Συναρτήσεις** είναι οι εντολές που προϋπάρχουν στο **Labview** για αριθμητικές πράξεις, ανάλυση, καταγραφή στοιχείων στο δίσκο κ.α.

Τα **SubVIs** είναι υπάρχοντα προγράμματα τα οποία καλεί κανείς ως υπορουτίνες σε κάποιο κύριο πρόγραμμα.

Οι **Προγραμματιστικές δομές**, όπως για παράδειγμα το **For** και το **While Loops**, ελέγχουν τη ροή των δεδομένων στο πρόγραμμα.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Τα **CINs** είναι ένας τρόπος για να συνδεθεί εξωτερικά ένας γραμμένος κώδικας με το **block diagram** του Labview, επειδή συμβαίνει πολλοί χρήστες να έχουν τις δικές τους ρουτίνες σε **γλώσσα C** ή **Pascal** και χρησιμοποιώντας **CINs** έχουν πρόσβαση σε αυτές. Κάθε **control-indicator** στο **front panel** αντιστοιχεί σε έναν δικό του **ακροδέκτη (κουτάκι)** στο **block diagram**, τον οποίο μπορεί να φανταστεί κανείς ως την πόρτα μέσω της οποίας δεδομένα μεταφέρονται μεταξύ του **block diagram** και του παραθύρου των γραφικών. Κάθε φορά που ο χρήστης ή κάποιο ανώτερου επιπέδου VI εισάγει δεδομένα στα αντικείμενα του **front panel**, αυτά καταλήγουν στο **block diagram** μέσα από αυτούς τους ακροδέκτες.

Όταν το VI φτάσει στο τέλος της λειτουργίας του, τα δεδομένα περνούν από το **block diagram** πίσω στο **front panel**, πάλι μέσα από τους κατάλληλους ακροδέκτες. Οι ακροδέκτες αυτοί δημιουργούνται ή διαγράφονται αυτόματα με τη δημιουργία ή την διαγραφή ενός **control** ή **indicator** αντικειμένου στο **front panel**.

Στο **block diagram** υπάρχουν τρεις κατηγορίες αντικειμένων:

### ➤ **Κόμβοι (Nodes)**

Εκτελούν μια συγκεκριμένη εργασία

### ➤ **Τερματικά (Terminals)**

Είναι τα αντικείμενα που συνδέουν το front panel με το block diagram

### ➤ **Καλώδια (Wires)**

Είναι οι οδοί από τις οποίες περνούν τα δεδομένα

Στο **front panel** κάθε αντικείμενο συνδέεται με το **block diagram** με την μορφή των τερματικών στοιχείων. Έχουμε την δυνατότητα να αλλάξουμε τις ιδιότητες των τερματικών για καλύτερη χρήση τους. Σε περίπτωση που θέλουμε να τροποποιήσουμε τα τερματικά αυτό γίνεται εύκολα με την χρήση εντολών από το **Pop-up** και **Pop-Down** μενού.

Στο **block diagram** είναι απαραίτητη η σύνδεση των εξαρτημάτων για να ξεκινήσει η λειτουργία του VI. Οι συνδέσεις είναι οι οδοί δεδομένων ανάμεσα στα τερματικά στοιχεία πηγής και προορισμού. Για να μην υπάρχουν προβλήματα με τις συνδέσεις θα πρέπει να είμαστε προσεκτικοί κατά την χρήση τους. Για παράδειγμα δεν μπορούμε να συνδέουμε μια πηγή (**control terminal**) με μια άλλη. Επιπλέον, δεν μπορούμε να συνδέσουμε ένα στοιχείο προορισμού (**indicator terminal**) με ένα άλλο. Φυσικά μπορούμε όμως να συνδέσουμε ένα **control terminal** με ένα **indicator terminal**.

Κάθε σύνδεση έχει διαφορετικό στυλ ή χρώμα ανάλογα με τον τύπο δεδομένων που ρέουν μέσω αυτής. (εικ 1.52)

- Το πορτοκαλί χρώμα αντιπροσωπεύει την ροή αριθμητικών δεδομένων
- Το πράσινο τα λογικά δεδομένα
- Το μωβ τους χαρακτήρες

	Βαθμοτή	Πίνακας 1Δ	Πίνακας 2Δ	
Αριθμός	—————	—————	—————	Πορτοκαλί
Boolean	.....	~~~~~	~~~~~	Πράσινο
String	~~~~~	○○○○○○○○	~~~~~	Μοβ

Εικόνα 1.52: Βασικά στυλ των καλωδίων στα block diagrams

### 1.8.2.1 Δημιουργία - Διαγραφή Συνδέσεων

Για να δημιουργήσουμε συνδέσεις στο **block diagram** χρησιμοποιούμε το εργαλείο **Wiring Tool** (εικ 1.53). Τοποθετούμε το εργαλείο σύνδεσης πάνω στο αντικείμενο που θέλουμε να συνδέσουμε και μόλις αρχίσει να αναβοσβήνει πατάμε το αριστερό πλήκτρο από το ποντίκι και το μετακινώντας το πραγματοποιούμε τη σύνδεση.

Σε περίπτωση που θέλουμε να αφαιρέσουμε τη σύνδεση, πατάμε το δεξί πλήκτρο του ποντικιού. Επιλέγουμε ένα κομμάτι της σύνδεσης πατώντας το αριστερό πλήκτρο του ποντικιού επάνω του με το εργαλείο θέσης. Με διπλό κλικ επιλέγουμε έναν κλάδο, ενώ με τριπλό ολόκληρη τη σύνδεση. Για να διαγράψουμε το επιλεγμένο κομμάτι πατάμε το πλήκτρο **Delete** ή από το **Short Cut** μενού επιλέγουμε την εντολή **Clean Wire Branch**.



Εικόνα 1.53: Απεικόνιση εργαλείου Σύνδεσης

#### Διακεκομμένα Καλώδια (Bad Wires)

Ο πιο σύνηθες τρόπος για την εύρεση λαθών προγραμματίζοντας με το **Labview** είναι η εύρεση διακεκομμένων καλωδίων. Όταν η σύνδεση είναι σωστή, το καλώδιο είναι συνεχές. Στην αντίθετη περίπτωση, ένα διακεκομμένο καλώδιο εμφανίζεται (εικ 1.54).

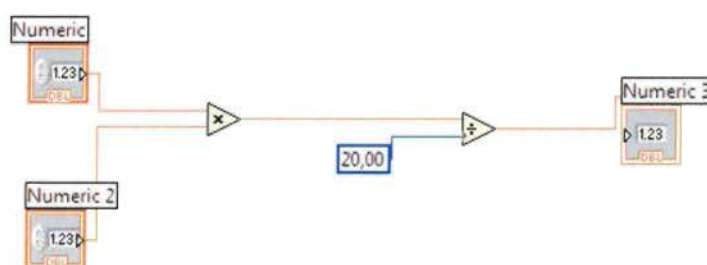
Όπως αναφέραμε και παραπάνω, για να διαγράψουμε διακεκομμένα καλώδια, τα επιλέγουμε και πατάμε το πλήκτρο **Delete** ή από το **Short Cut** μενού επιλέγουμε την εντολή **Clean Wire Branch**.



Εικόνα 1.54: Απεικόνιση λανθασμένης και σωστής σύνδεσης

## 1.9 Προγραμματισμός με το Labview

Η εφαρμογή οποιουδήποτε προγράμματος στο **Labview** βασίζεται σε ένα πολύ απλό κανόνα, την ομαλή ροή των δεδομένων. Μόνο όταν όλα τα δεδομένα από προηγούμενους κόμβους φτάσουν στους ακροδέκτες εισόδου ενός κόμβου, τότε μόνο αυτός τρέχει. Στην συνέχεια, ο κάθε κόμβος όταν εκτελεστεί διαθέτει όλα τα απαραίτητα δεδομένα στους ακροδέκτες εξόδου του. Άρα η σημαντική διαφορά, μεταξύ του **Labview** και άλλων γλωσσών προγραμματισμού, είναι ότι σε άλλες γλώσσες οι εντολές εκτελούνται η μια μετά την άλλη, όπως δηλαδή έχουν γραφτεί. Στο παράδειγμα που ακολουθεί απεικονίζεται το **block diagram** ενός εικονόργανου που πολλαπλασιάζει δύο αριθμούς και διαιρεί 20 από το αποτέλεσμα (εικ 1.55)



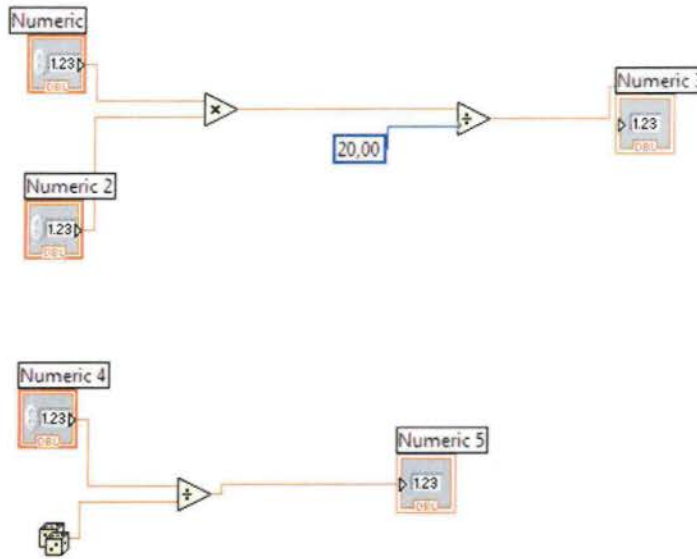
Εικόνα 1.55: Block diagram παραδείγματος

Το **block diagram** εκτελείται από αριστερά προς τα δεξιά, αφού για να πραγματοποιηθεί η διαίρεση θα πρέπει πρώτα να εκτελεστεί ο πολλαπλασιασμός.

Το επόμενο παράδειγμα ανεβάζει λίγο τον βαθμό δυσκολίας, καθώς δεν γνωρίζουμε από την αρχή ποιο μέρος του κώδικα θα πραγματοποιηθεί πρώτο, ο πολλαπλασιασμός ή η διαίρεση εφόσον τα δεδομένα είναι γνωστά στους ακροδέκτες εισόδου και των δύο λειτουργιών από την αρχή του προγράμματος (εικ 1.56). Σε τέτοια παραδείγματα, όταν ένα ένα τμήμα του κώδικα πρέπει να εκτελεστεί πριν από

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

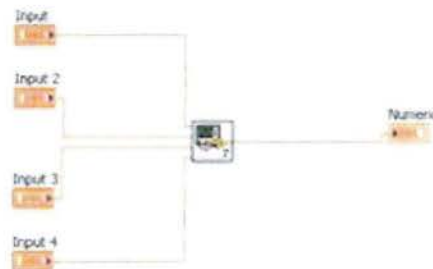
κάποιο άλλο, δίχως να υπάρχει εξάρτηση, γίνεται χρήση της προγραμματιστικής δομής **Sequence** για την οποία θα μιλήσουμε εκτενέστερα παρακάτω.



Εικόνα 1.56: Block diagram παραδείγματος

### 1.9.1 Χρήση των SubVIs

Η επικράτηση του **Labview** παρουσιάζεται στην ιεραρχική φύση των VIs. Σκεδιάζοντας ένα VI, μπορούμε να το χρησιμοποιήσουμε ως **subVI** σε ένα υψηλότερου επιπέδου VI (εικ 1.57). Μπορούμε να χρησιμοποιήσουμε όσα **subVIs** επιθυμούμε.

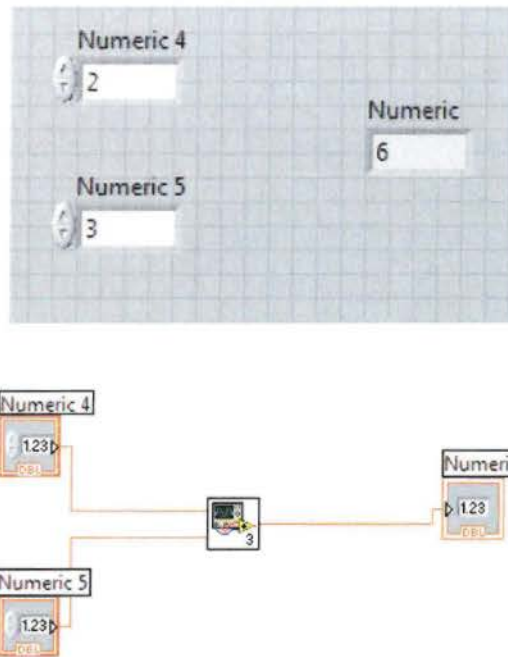


Εικόνα 1.57: Απεικόνιση subVI μέσα σε VI

Τα **subVIs** μπορούμε να τα βάλουμε μόνο στο **block diagram** με την μορφή εικονιδίου λειτουργούν όπως τα υπόλοιπα αντικείμενα. Επιπλέον ένα **subVI** μπορεί να καλεί άλλα **subVIs**, και αυτά με την σειρά τους μπορούν να καλούν άλλα.

Στην εικόνα 1.58 φαίνεται το **front panel** ενός VI, το οποίο εκτελεί έναν πολλαπλασιασμό και στην αμέσως επόμενη εικόνα φαίνεται το διάγραμμα VI που χρησιμοποιεί **subVI**.





Εικόνα 1.58: Front panel παραδείγματος και block diagram που φαίνεται η χρήση του subVI

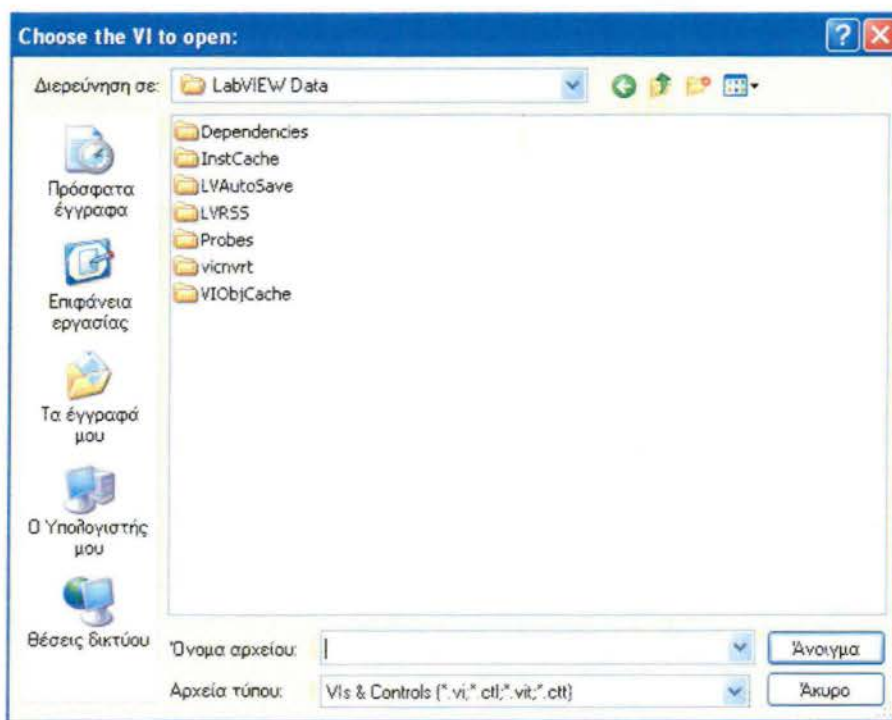
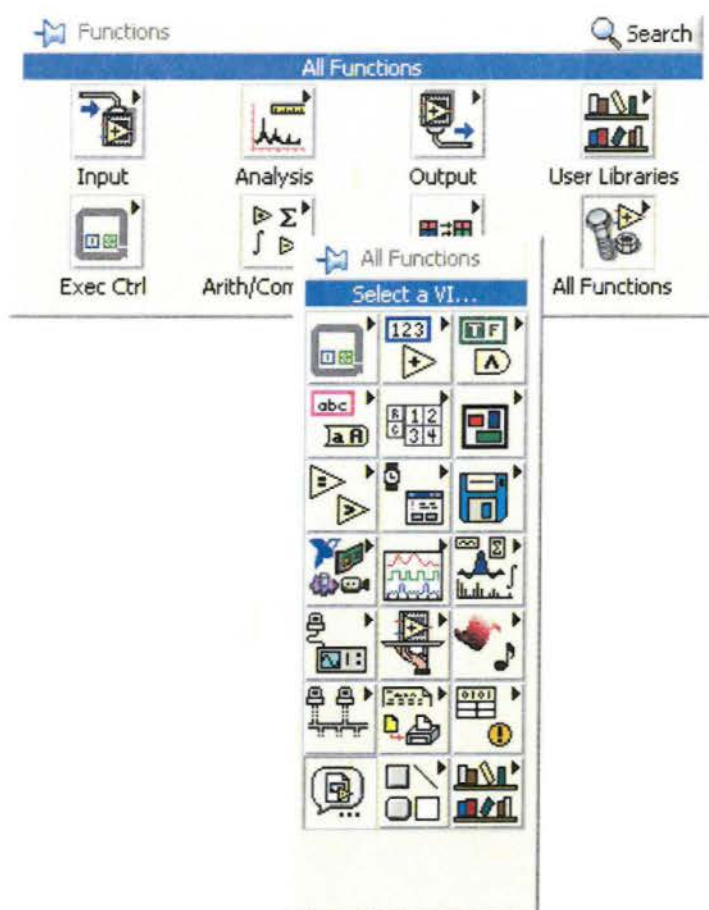
### Δημιουργία subVI

Για να δημιουργήσουμε subVI υπάρχουν δύο τρόποι:

#### 1. Δημιουργώντας subVI από ένα VI.

Αν θέλουμε να βάλουμε ένα **subVI** στο **block diagram**, το οποίο βρίσκεται αποθηκευμένο κάπου στον σκληρό μας, ανοίγουμε την παλέτα λειτουργιών (**Functions**) και επιλέγουμε την εντολή **Select a VI...** Στη συνέχεια ανοίγει ένα παράθυρο αναζήτησης αρχείων, από το οποίο εντοπίζουμε το VI που θέλουμε να χρησιμοποιήσουμε και το τοποθετούμε στο **block diagram**. (εικ 1.59)

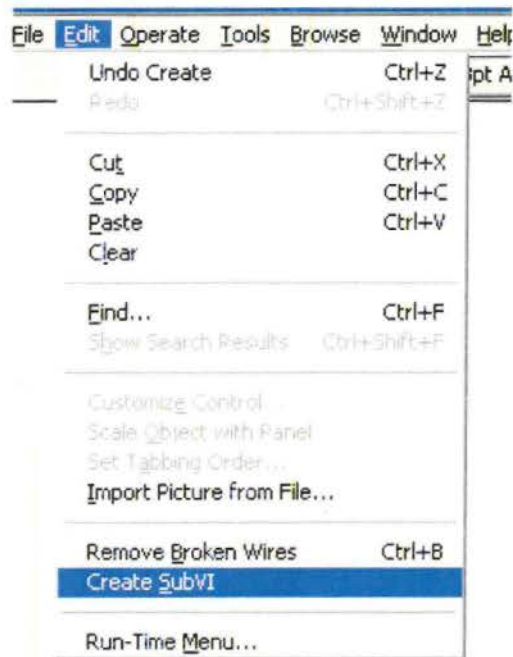
# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.59: Παράθυρο αναζήτησης SubVI

## 2. Δημιουργία SubVI από επιλεγμένα αντικείμενα.

Με αυτό τον τρόπο μπορούμε να δημιουργήσουμε **subVIs** αν επιλέξουμε τα αντικείμενα από ένα VI και στην συνέχεια διαλέξουμε την εντολή **Create SubVI** από το μενού **Edit** (εικ 1.60). Όλα τα επιλεγμένα τερματικά τοποθετούνται σε ένα **subVI** με εικονίδιο το προεπιλεγμένο αυτό του **Labview**. Σε περίπτωση που επιλεγούν τερματικά στοιχεία τα οποία είναι συνδεδεμένα με κάποια άλλα, κατά τη δημιουργία του **subVI** μεταφέρονται τα επιλεγμένα στοιχεία και αντιγράφονται αυτά που είναι συνδεδεμένα με τα επιλεγμένα. Οι συνδέσεις διατηρούνται, ενώ τα **controls** του **subVI** συνδέονται αυτόματα με τους συνδέσμους των τερματικών. Έτσι μπορούμε να συναρμολογήσουμε πιο εύκολα ένα πολύπλοκο VI.



Εικόνα 1.60: Δημιουργία subVI από το μενού Edit

Στο παράδειγμα που ακολουθεί θα κατασκευάσουμε ένα **subVI** με την μέθοδο **Create subVI** και θα το ονομάσουμε **Thermometer.vi**. Έτσι κατασκευάζουμε ένα εικονόργανο – θερμόμετρο χρησιμοποιώντας την λειτουργία **Read Voltage.vi** ώστε να καλεί ένα **subVI** που μετατρέπει την θερμοκρασία από βαθμούς Κελσίου σε βαθμούς Fahrenheit.

### Βήμα 1:

Ανοίγουμε ένα καινούργιο **front panel**.

Τοποθετούμε έναν **indicator** – **Θερμόμετρο**.

Επιλέγουμε την εντολή **Thermometer** από την υποπαλέτα αριθμητικών λειτουργιών.

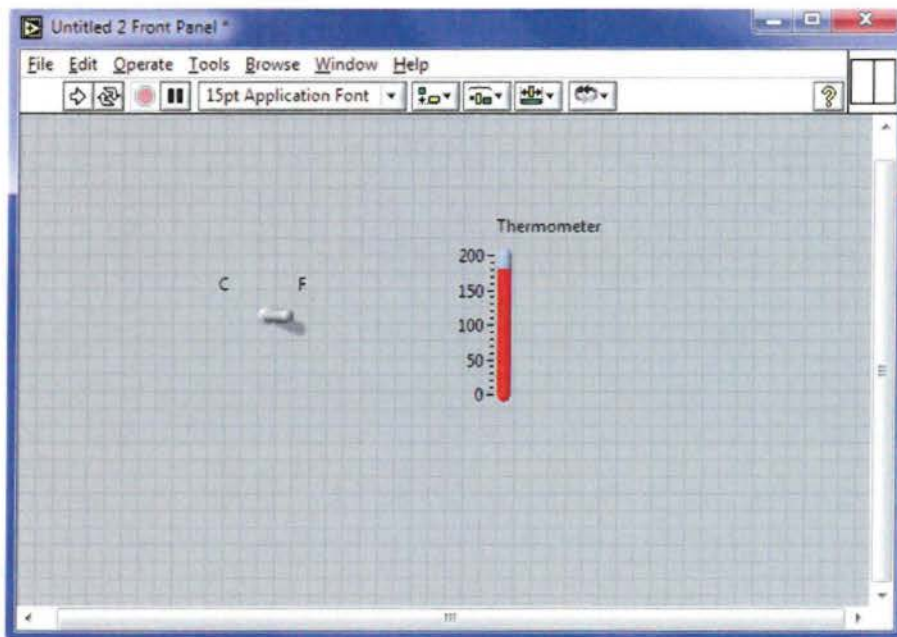
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

### Βήμα 2:

Προσαρμόζουμε την κλίμακα του Θερμομέτρου, για να μετράει την θερμοκρασία στο διάστημα 0 έως 200. Πατάμε δεξί κλικ πάνω στο εικονίδιο του θερμομέτρου και επιλέγοντας την εντολή **Properties** από το **Pop-Up** μενού. Στο παράθυρο διαλόγου που θα εμφανιστεί επιλέγουμε την καρτέλα **Scale** και δίνουμε την τιμή 200 στην επιλογή **maximum**.

### Βήμα 3:

Τοποθετούμε ένα διακόπτη στο **front panel** επιλέγοντας **Toggle Switch** από την υποπαλέτα **controls**. Με το εργαλείο ονοματοθέτησης τοποθετούμε δύο ετικέτες πάνω από τον διακόπτη και τις ονομάζουμε C και F αντίστοιχα.



Παρατήρηση:

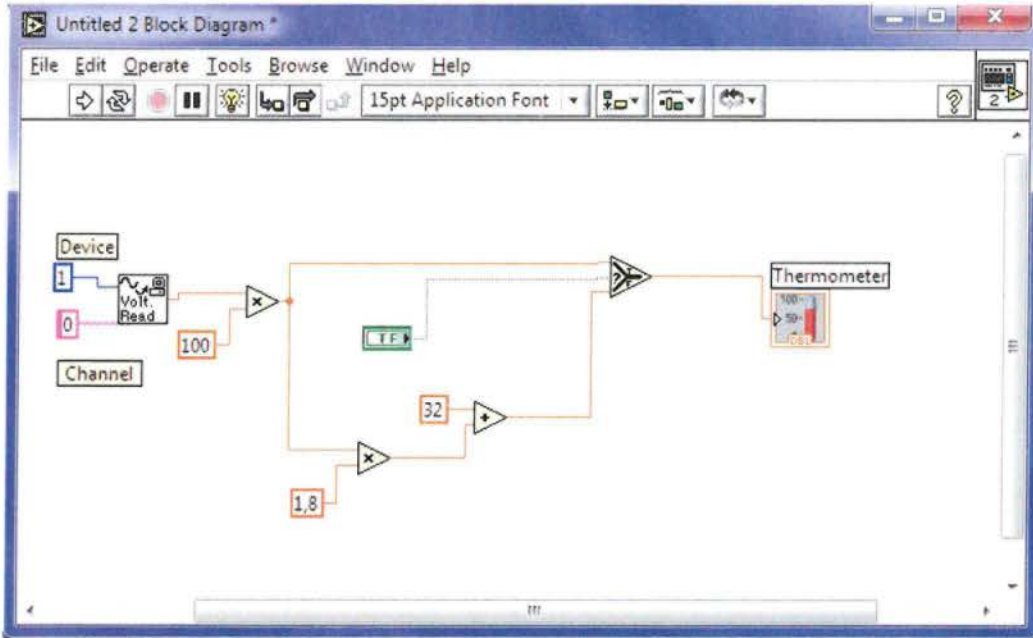
Μας δίνετε η δυνατότητα να περιγράψουμε την λειτουργία κάθε αντικείμενου του **front panel** πατώντας δεξί κλικ πάνω σε οποιοδήποτε αντικείμενο και επιλέγοντας την εντολή **Description and Tip**....

### Βήμα 4:

Κατασκευάζουμε το **block diagram** που ακολουθεί.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Αρχικά πολλαπλασιάζουμε την τάση εξόδου της λειτουργίας **Volt. Read** επί 100 για να πραγματοποιηθεί η μετατροπή σε βαθμούς Κελσίου. Για την μετατροπή σε βαθμούς Fahrenheit πολλαπλασιάζουμε 1,8 και στην συνέχεια προσθέτουμε την σταθερά 32 στην θερμοκρασία Κελσίου.

Συνδέουμε στην είσοδο **True** της λειτουργίας **Select** το αποτέλεσμα που έχει προκύψει σε βαθμούς Κελσίου ενώ στην είσοδο **False** συνδέουμε το αποτέλεσμα της μετατροπής σε βαθμούς Fahrenheit. Τέλος στην μεσαία είσοδο της **Select** ανάλογα με την θέση του διακόπτη μας εμφανίζει στην έξοδο την τιμή που παίρνει σε μια από τις δύο εισόδους.

### Βήμα 5:

Δοκιμάζουμε να τρέξουμε το VI πατώντας τον διακόπτη συνεχούς λειτουργίας **Run**. Στην συνέχεια αποθηκεύουμε το εικονόργανο με το όνομα **Thermometer.vi**. Το VI που μόλις κατασκευάσαμε θα το μετατρέψουμε σε **subVI** με την μέθοδο που θα ακολουθήσει προκειμένου να χρησιμοποιηθεί στο **block diagram** άλλων εικονογράφων.

### Βήμα 6:

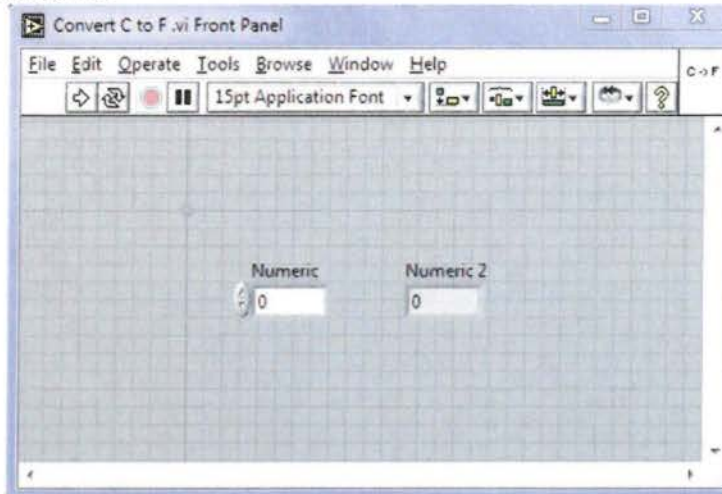
Μετατρέπουμε το παραπάνω παράδειγμα με σκοπό να καλεί ένα **subVI** για να μετατρέπει την θερμοκρασία από Βαθμούς Κελσίου σε Fahrenheit. Με το εργαλείο θέσης επιλέγουμε τις λειτουργίες πολλαπλασιασμού και άθροισης. Στην συνέχεια, επιλέγουμε το **Create SubVI** από το μενού **Edit**.

### Βήμα 7:

Απεικονίσουμε το **subVI** που δημιουργήσαμε με μια χαρακτηριστική εικόνα. Πατώντας διπλό κλικ πάνω στο **subVI** μας εμφανίζει το **front panel** του **subVI**. Πατώντας πάνω στο εικονίδιο του κοννέκτορα επιλέγουμε την εντολή **Edit Icon** και

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

δημιουργούμε την εικόνα της αρεσκείας μας η οποία θα αντιπροσωπεύει το πως θα εμφανίζεται το **subVI**.



Παρατήρηση:

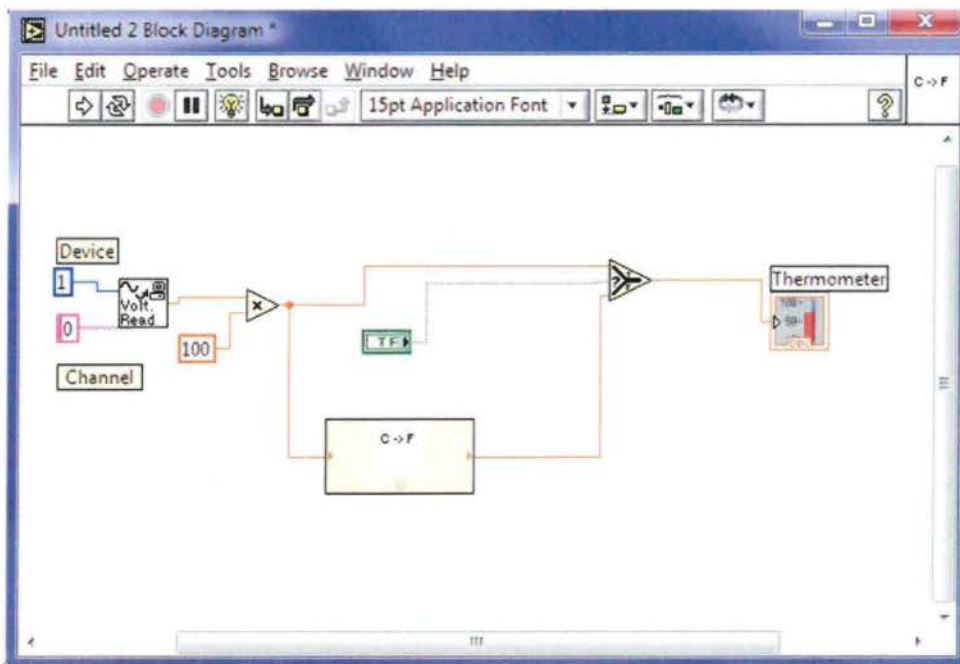
Κατασκευάζουμε ένα **subVI** επιλέγοντας **Create SubVI** ο κωννέκτορας και οι συνδέσεις έχουν ήδη γίνει.

### Βήμα 8:

Αποθηκεύουμε το VI με την ονομασία **Convert C to F.vi**.

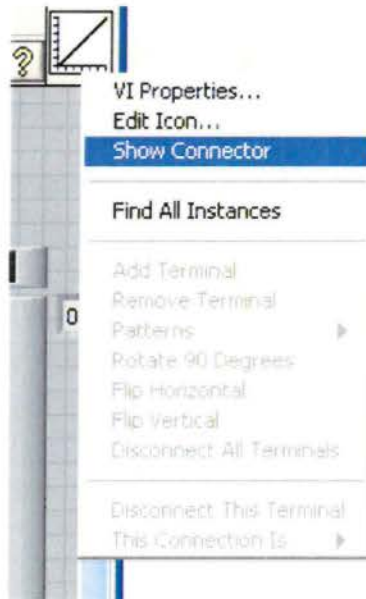
Κατόπιν αποθηκεύουμε πλέον το τροποποιημένο **Thermometer.vi** σε **New Thermometer.vi**

Το **New Thermometer.vi** περιλαμβάνει τώρα στο **block diagram** το **subVI Convert C to F.vi**.



### 1.9.2.1 Λειτουργία Κονέκτορα

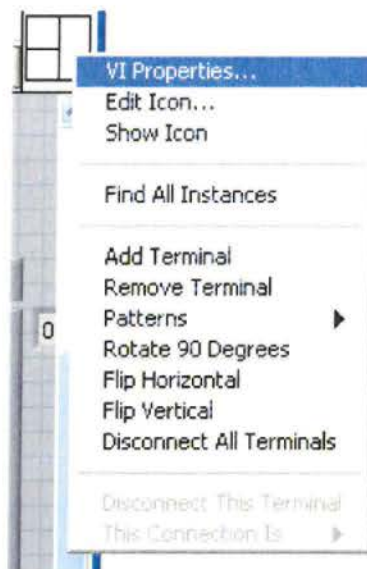
Με τη χρήση ενός κονέκτορα (**συνδετήρας**) μπορεί κανείς να επεξεργαστεί τα δεδομένα ενός **subVI**. Αν θέλουμε να εισάγουμε δεδομένα από και προς αντικείμενα **controls/indicators** ενός **subVI**, θα πρέπει να έχουμε δηλώσει ακροδέκτες στον κονέκτορα γι' αυτά τα αντικείμενα. Ξεκινώντας, θα πρέπει να διαλέξουμε τον κονέκτορα πατώντας το δεξί πλήκτρο του ποντικιού στο εικονίδιο στην πάνω δεξιά μεριά του **front panel** και στην συνέχεια επιλέγουμε την εντολή **Show Connector** (εικ 1.61). Με αυτόν τον τρόπο ορίζουμε τον αριθμό των ακροδεκτών για το συγκεκριμένο εικονόργανο του παραδείγματος.



Εικόνα 1.61 : Επιλογή κονέκτορα

Στην συνέχεια, κάνουμε σύνδεση μεταξύ των ακροδεκτών των αντικειμένων **controls/indicators** του **front panel** και αυτών με τα οποία αντιστοιχούν. Ο κονέκτορας μπορεί να οριστεί μόνο στο **front panel** ενός εικονοργάνου. Με την εντολή **Show Connector** γίνεται επιλογή ενός συνόλου αριθμού ακροδεκτών, το οποίο αντιστοιχεί στο συνολικό αριθμό αντικειμένων στο **front panel**.(εικ 1.62)

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.62 : Επιλογές κονέκτορα

Ο κονέκτορας στην ουσία είναι ένα σύνολο από ακροδέκτες, που χρησιμοποιούνται για τη σύνδεση των **controls** και των **indicators**, τις εισόδους και εξόδους του **subVI**, με τα τερματικά του VI που το καλεί. Μέχρι και 28 εισοδοί και εξοδοί είναι διαθέσιμοι. Με την εντολή **Show Patterns** από το **Pop-Up** μενού, μπορούμε να δούμε τους τρόπους με τους οποίους υπάρχει δυνατότητα να τοποθετήσουμε ένα κονέκτορα. Με την εντολή **Patterns** έχουμε την δυνατότητα, να αλλάξουμε τον αριθμό των ακροδεκτών στο σύστημα μας. Ενώ αν θέλουμε να αλλάξουμε το χωρικό προσανατολισμό των ακροδεκτών θα πρέπει να διαλέξουμε μια από τις εντολές **Flip Horizontal**, **Flip Vertical**, **Rotate 90°**.

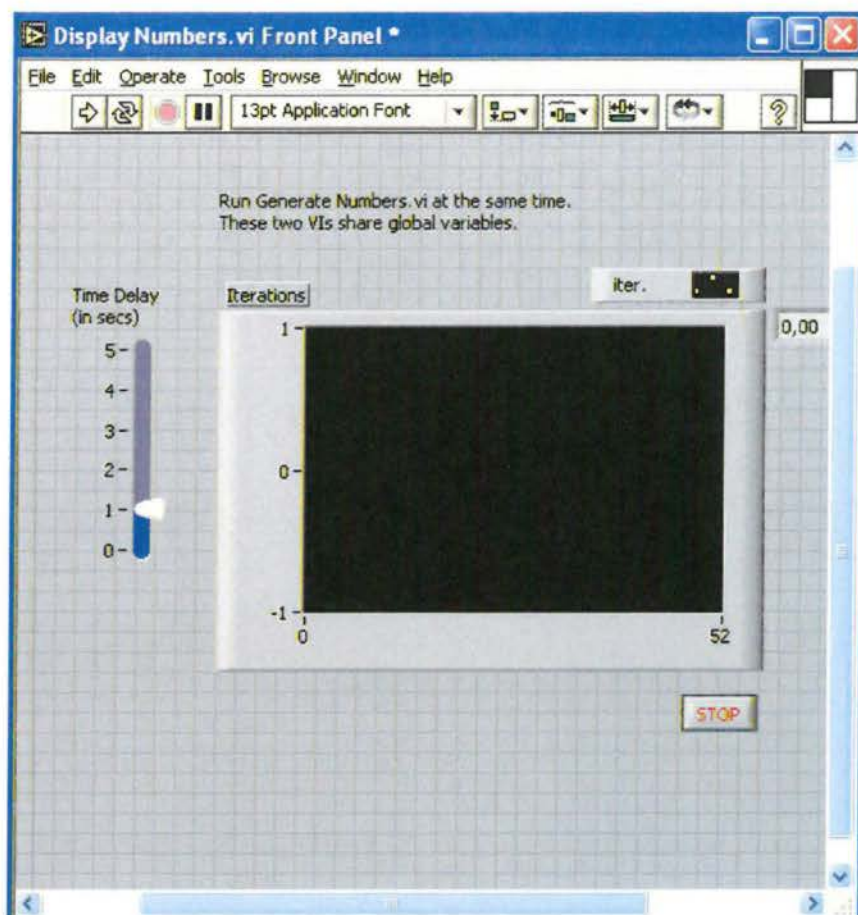
Συνδέουμε τους ακροδέκτες στα αντικείμενα του **front panel** με το εργαλείο σύνδεσης γνωστό και ως **Wiring Tool**.

Ακολουθεί παράδειγμα για την καλύτερη κατανόηση των ανωτέρων.

Πατάμε το αριστερό πλήτρο του ποντικιού πάνω σε έναν από τους ακροδέκτες και αυτόματα ο ακροδέκτης γίνεται μαύρος (εικ 1.63)



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 1.63: Αλλαγή χρώματος ακροδέκτη

Στην συνέχεια πατάμε το αριστερό πλήκτρο πάνω στο αντικείμενο το οποίο θέλουμε να αντιστοιχίσουμε με το συγκεκριμένο ακροδέκτη. Το αντικείμενο περιβάλλεται από ένα διακεκομμένο περίγραμμα. Επιλέγοντας ένα άλλο αντικείμενο ή έναν άλλο ακροδέκτη, η διακεκομμένη γραμμή δεν εμφανίζεται πλέον και ο επιλεγμένος ακροδέκτης χρωματίζεται, γεγονός που σημαίνει ότι η σύνδεση ήταν σωστή

Επαναλαμβάνουμε τις παραπάνω ενέργειες για όλες τις συνδέσεις. Υπάρχει η δυνατότητα, να επιλέξουμε το αντικείμενο πριν από τον ακροδέκτη για τη σύνδεση. Επιπλέον, μπορούμε να αφήσουμε ασύνδετους ακροδέκτες στον κονέκτορα ή και ακόμα ασύνδετα αντικείμενα στο **front panel** μην δημιουργώντας πρόβλημα στο εικονόγραφο.

## 2.1 Δομές

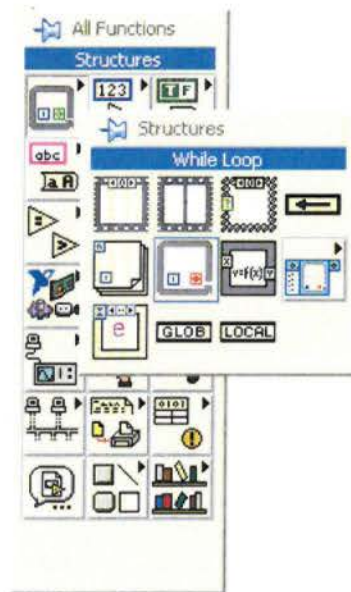
Οι δομές (**structures**) χρησιμοποιούνται για την εκτέλεση ροής σε ένα VI. Συγκεκριμένα όταν θέλουμε να επαναληφθεί μια λειτουργία , μέχρις ότου μια συνθήκη χαρακτηριστεί αληθής, ή για έναν προκαθορισμένο αριθμό επαναλήψεων εντολών. Οι δομές τις οποίες συναντάμε στο **LabVIEW** είναι οι εξής:

- **While Loop**
- **For Loop**
- **Case**
- **Sequence**
- **Event Structure**

Στην κατηγορία των δομών ανήκουν και οι τύποι των μεταβλητών **Global** και **Local Variable** καθώς και η λειτουργία **Formula Node**.

### 2.1.1. Η Δομή While Loop και η λειτουργία της

Τοποθετούμε την **While** στο **block diagram** επιλέγοντάς την από την υποπαλέττα **Structures**.(εικ. 2.1) Μόλις την επιλέξουμε μας δίνεται η δυνατότητα να ορίσουμε το μέγεθός της. Αυτό επιτυγχάνεται καθώς με την βοήθεια του αριστερού ποντικιού, περικυκλώνουμε με την θηλιά το τμήμα του Διαγράμματος που μας ενδιαφέρει να επαναληφθεί.





Εικόνα 2.1: Έυρεση δομής While Loop

Η **While Loop** έχει δύο ακροδέκτες: τον ακροδέκτη απαρίθμησης (**iteration terminal i**) ο οποίος μετράει της επαναλήψεις και τον υπο συνθήκη ακροδέκτη (**conditional terminal**) ο οποίος αποτελεί την είσοδο της δομής.(εικ.2.2) Η **While Loop** εκτελεί τον κώδικα που βρίσκεται στα όρια του και κάθε φορά που ολοκληρώνεται μία εκτέλεση το **LabVIEW** ελέγχει την τιμή στον υπο συνθήκη ακροδέκτη. Εάν η τιμή στον υπο συνθήκη ακροδέκτη είναι **αληθής (true)**, η διαδικασία επαναλαμβάνεται έως ότου η τιμή αλλάξει σε **ψευδής (false)**, τότε σταματάει η επανάληψη.

**ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.**



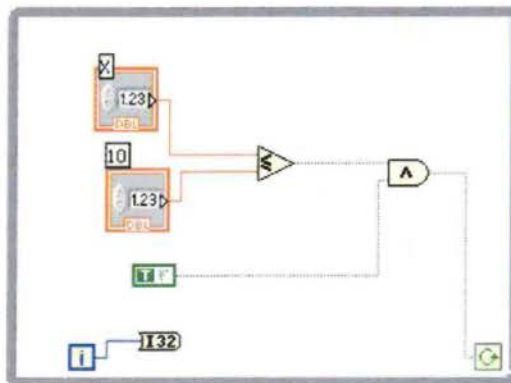
Εικόνα 2.2: Δομή While Loop

-  ακροδέκτη απαρίθμησης
-  υπο συνθήκη ακροδέκτης

Σημείωση: ο κώδικας που βρίσκεται μέσα στα όρια της **While** θα εκτελεστεί τουλάχιστον μία φορά, αυτό συμβαίνει διότι το **LabVIEW** ελέγχει την τιμή του υπο συνθήκη ακροδέκτη αν είναι **true** ή **false** στο τέλος κάθε εκτέλεσης.

**Παράδειγμα 1:**

Για να κατανοήσουμε καλύτερα την λειτουργία της **While** θα παρουσιάσουμε ένα απλό παράδειγμα στο οποίο ο βρόχος **While** εκτελεί τον κώδικα για όσο ο X είναι μικρότερος του 10 και εάν ο διακόπτης έχει τιμή αλήθειας. ο αριθμός των επαναλήψεων εμφανίζεται σε ένα **indicator**. (εικ. 2.3)



Εικόνα 2.3: Παράδειγμα συνθήκης While



### Παράδειγμα 2:

Στο ακόλουθο παράδειγμα θα χρησιμοποιήσουμε το εικονόργανο **Digital Thermometer.vi** το οποίο κατασκευάσαμε στην προηγούμενη ενότητα και κάνοντας χρήση της δομής **While loop** θα καταγράψουμε την θερμοκρασία δωματίου.

#### Βήμα 1:

Τοποθετούμε ένα διακόπτη στο **front panel**. Με το εργαλείο ονοματοθέτησης το ονομάζουμε “ΔΙΑΚΟΠΤΗ”.

#### Βήμα 2:

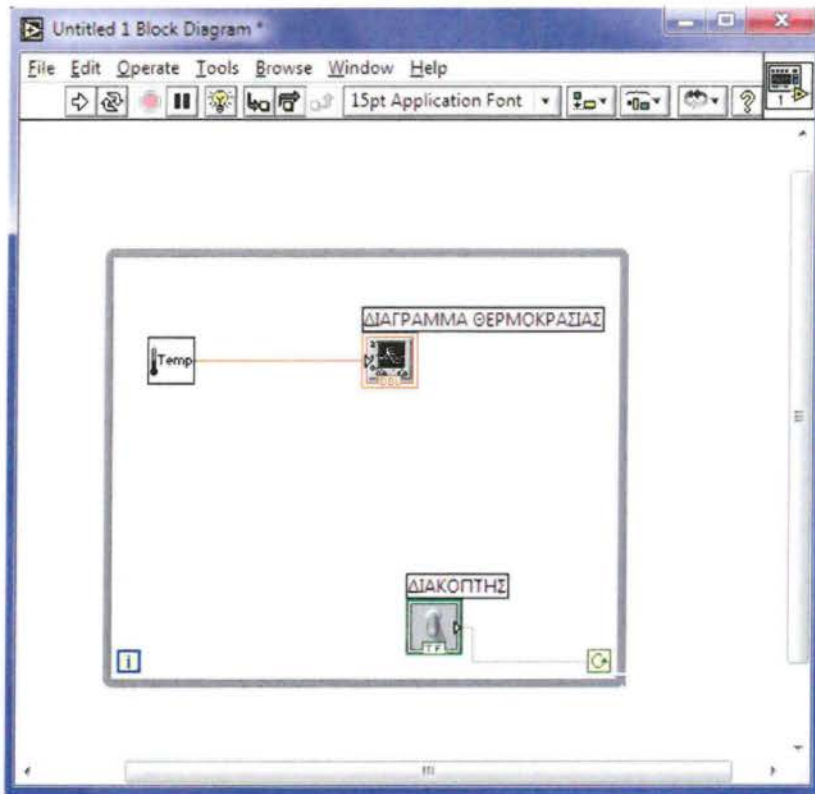
Τοποθετούμε ένα διάγραμμα κυματομορφών και το ονομάζουμε “ΔΙΑΓΡΑΜΜΑ ΘΕΡΜΟΚΡΑΣΙΑΣ”. Επίσης μετονομάζουμε και την ετικέτα για το γράφημα σε “ΘΕΡΜ/ΣΙΑ”. Ακολούθως αλλάζουμε την κλίμακα μέτρησης του διαγράμματος με το εργαλείο ονοματοθέτησης ,που χρησιμοποιήσαμε και παραπάνω, από 0.0 σε 90.

#### Βήμα 3:

Στο **block diagram** επιλέγουμε την δομή **While** και μέσα της τοποθετούμε τους δύο ακροδέκτες που αντιστοιχούν στα αντικείμενα του **front panel**. Από την υποπαλέτα **functions** επιλέγουμε την εντολή **Select a VI...**. Στην συνέχεια διαλέγουμε το εικονόργανο **Digital Thermometer.vi**.

Κάνουμε την συνδεσμολογία των αντικειμένων όπως φαίνεται στην εικόνα που ακολουθεί.

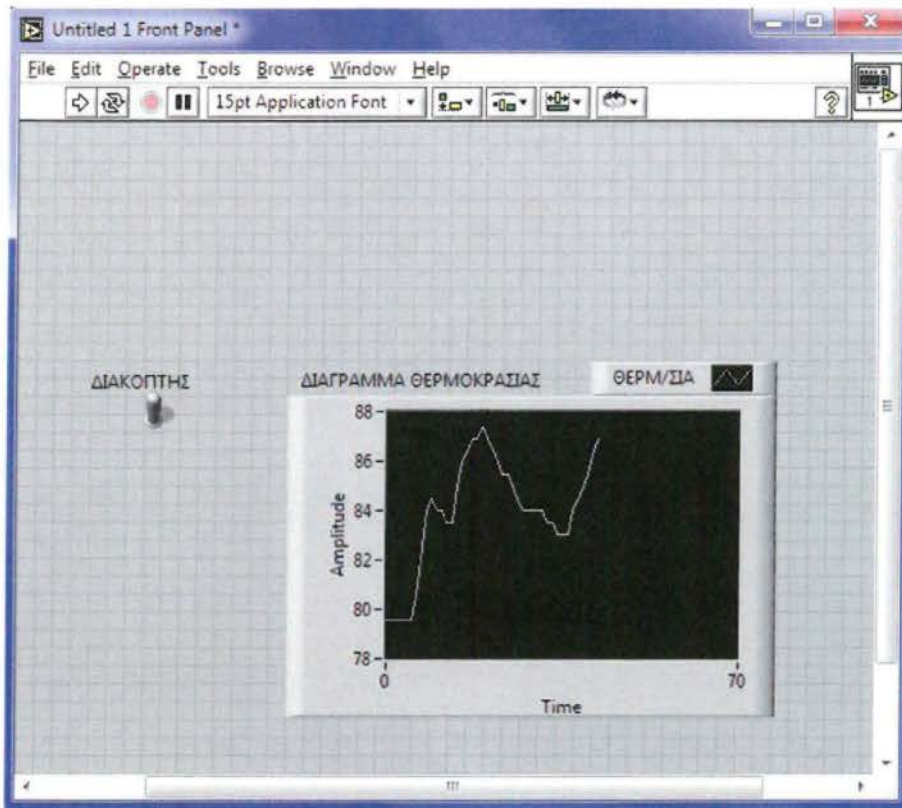
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



### Βήμα 4 :

Πατάμε το κουμπί **Run** από το μενού επιλογών στο πάνω μέρος του παραθύρου και ύστερα ενεργοποιούμε τον διακόπτη στη θέση αληθής (**True**). Η **While** θα εκτελείται όσο η συνθήκη (διακόπτης) είναι αληθής και θα εμφανίζει στο ΔΙΑΓΡΑΜΜΑ ΘΕΡΜΟΚΡΑΣΙΑΣ τις μετρήσεις από το εικονόργανο **Digital Thermometer.vi**.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



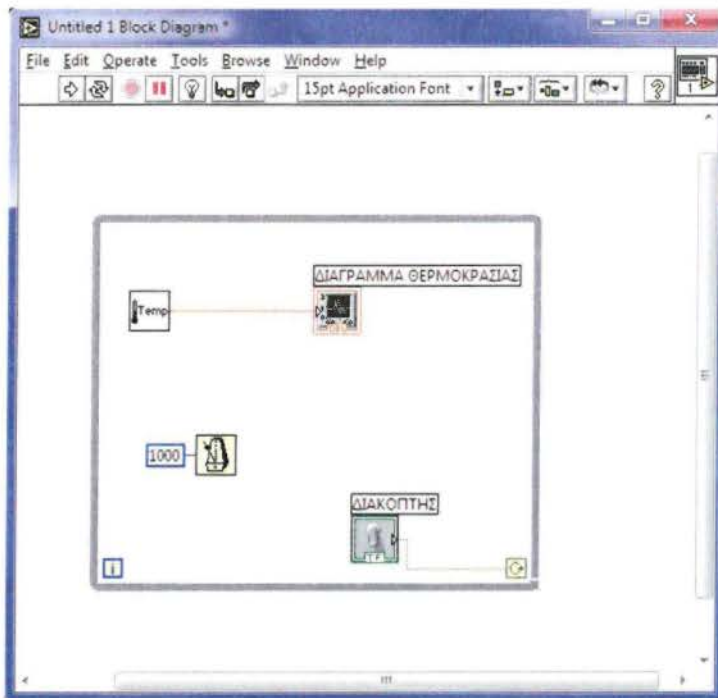
### Βήμα 5:

Για να συλλέξουμε δεδομένα ανά τακτά χρονικά διαστήματα θα πρέπει να ορίσουμε κάποιον χρονικό περιορισμό όσο αφορά τη συχνότητα εκτέλεσης της δομής **While loop**. Αυτό πραγματοποιείται με την χρήση της λειτουργίας **Wait Until Next ms Multiple**. Η συγκεκριμένη λειτουργία ορίζει την επανάληψη κάθε κύκλου της δομής στο χρονικό διάστημα που έχουμε ορίσει σε χιλιοστά του δευτερολέπτου.

### Βήμα 6:

Θέτουμε τη δειγματοληψία του παραδείγματος κάθε δευτερόλεπτο όπως φαίνεται και στην εικόνα που ακολουθεί.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



### Βήμα 7:

Αποθηκεύουμε το εικονόργανο με την ονομασία **ΜΕΤΡΗΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ.vi**

### Βήμα 8:

Για να εμπλουτίσουμε το παράδειγμα μας αντικαθιστούμε στο **front panel** τον διακόπτη με ένα στοιχείο **Knob** το οποίο καθορίζει στην χρονική καθυστέρηση στο διάστημα 0 έως 100sec.

ΑΠΟΘΗΚΕΥΟΥΜΑΙ ΤΟ ΠΑΡΑΔΕΙΓΜΑ ΩΣ SIGNAL.VI

## 2.1.2. Η Δομή For Loop και η λειτουργία της

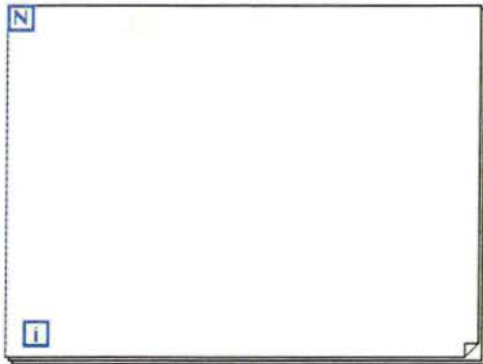
Τοποθετούμε την **For** στο **block diagram** επιλέγοντάς την από την υποπαλέττα **Structures**. Μόλις την επιλέξουμε μας δίνεται η δυνατότητα να ορίσουμε το μέγεθός της. Αυτό επιτυγχάνεται καθώς με την βοήθεια του αριστερού ποντικιού, περικυκλώνουμε με την θηλιά το τμήμα του Διαγράμματος που μας ενδιαφέρει να επαναληφθεί.

Η διαφορά της For με την While είναι ότι η For πραγματοποιεί όσες επαναλήψεις έχουμε ορίσει εμείς, ενώ η While επαναλαμβάνει την εκτέλεση έως ότου μία συνθήκη γίνει ψευδής.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Η δομή **For** έχει δύο ακροδέκτες, τον συντελεστή επανάληψης (**Count Terminal**) **N** με τον οποίο ορίζουμε πόσες φορές θα επαναληφθεί η εκτέλεση και τον απαριθμητή (**iterational terminal**) **I** ο οποίος περιέχει τον αριθμό επαναλήψεων που έχουν γίνει.

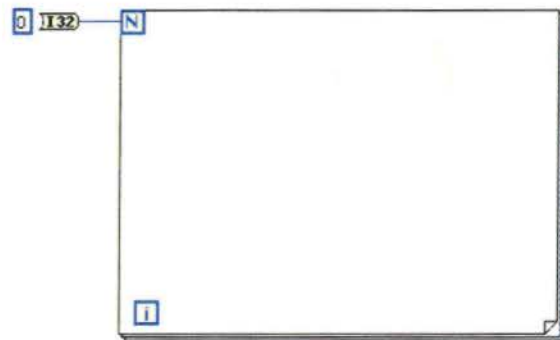


**N** συντελεστής επανάληψης

**i** απαριθμητής

Εικόνα 2.4: Δομή For Loop

Για να πάρει τιμή ο **συντελεστής N** τον ενώνουμε καλωδιακά με κάποια εξωτερική τιμή, αυτό επιτυγχάνεται χρησιμοποιώντας αριθμητική σταθερά ή στοιχείο **control** από την παλέτα αντικειμένων επιλέγοντας **Create constant** από το **Pop-Up** μενού του **For Loop**. Ο **απαριθμητής I** στην πρώτη εκτέλεση παίρνει την τιμή 0 και στην τελευταία N-1.(εικ. 2.5)



Εικόνα 2.5: Ορισμός τιμού επαναλήψεων δομής For

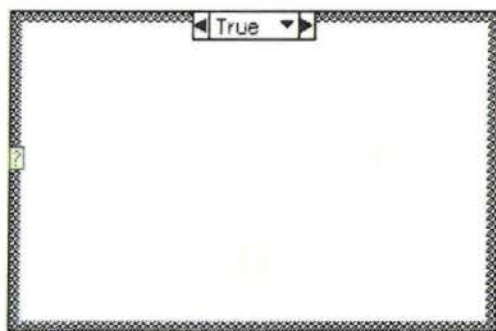
### 2.1.3. Η Δομή Case και η λειτουργία της

Τοποθετούμε την **Case** στο **block diagram** επιλέγοντάς την από την υποπαλέττα **Structures**.(εικ.2.6) Μόλις την επιλέξουμε μας δίνεται η δυνατότητα να ορίσουμε το μέγεθός της. Αυτό επιτυγχάνεται καθώς με την βοήθεια του αριστερού ποντικιού, περικυκλώνουμε με την θηλιά το τμήμα του Διαγράμματος που μας ενδιαφέρει να επαναληφθεί.

Με την **Case** εκτελούμε μία συγκεκριμένη συνθήκη. Η δομή **Case** αποτελείται από έναν **δείκτη** και έναν ακροδέκτη επιλογής (**selector**). Θέτουμε δεδομένα στον

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

ακροδέκτη επιλογής και ανάλογα με το είδος των δεδομένων αλλάζει και ο αριθμός των περιπτώσεων (**Cases**). Αν ο ακροδέκτης δεχτεί λογική τιμή, τότε οι **Cases** θα είναι **αληθής (true)** ή **ψευδής (false)**. Κάθε **Case** έχει το δικό της υποδιάγραμμα και μπορούμε να προβάσουμε μόνο μία κάθε φορά χρησιμοποιώντας τα βελάκια που βρίσκονται στο δείκτη.

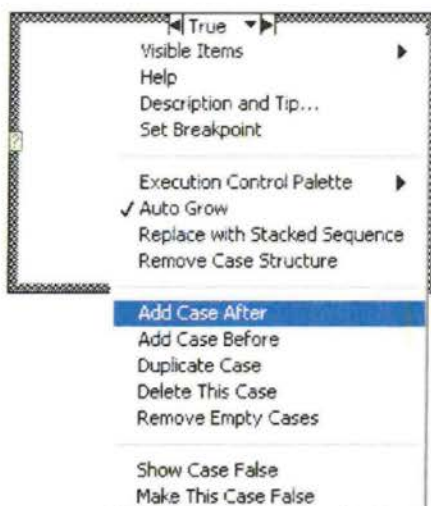


Εικόνα 2.6: Δομή Case

Κάθε φορά εκτελείται ένα υποδιάγραμμα, είναι αυτό που συμφωνεί με την συνθήκη η οποία ορίζεται στον επιλογέα. Στη συνέχεια συγκρίνει τους δείκτες των υποδιαγραμμάτων. Αν κάποιος δείκτης συμφωνεί τότε εκτελείται το αντίστοιχο υποδιάγραμμα. Αν δεν συμφωνεί τότε εκτελείται το προεπιλεγμένο (**default**) **Case**.

Σημείωση: ο τύπος δεδομένων που εισάγουμε στον δείκτη θα πρέπει να συμφωνεί με τον τύπο του στοιχείου που είναι συνδεδεμένο στον επιλογέα, αλλιώς το **LabVIEW** δεν θα μπορεί να κάνει την σύγκριση.

Μπορούμε να προσθέσουμε **Cases** από το **Pop-Up** μενού, με τις επιλογές **Add Case After** και **Add Case Before** οι οποίες προσθέτουν **Cases** μετά και πριν αντίστοιχα (εικ.2.7). Με την επιλογή **Duplicate Case** αντιγράφουμε **case** και διαγράφουμε με την επιλογή **Delete This Case**. Για να προεπιλέξουμε **Case** επιλέγουμε το **Make This Case Default**.



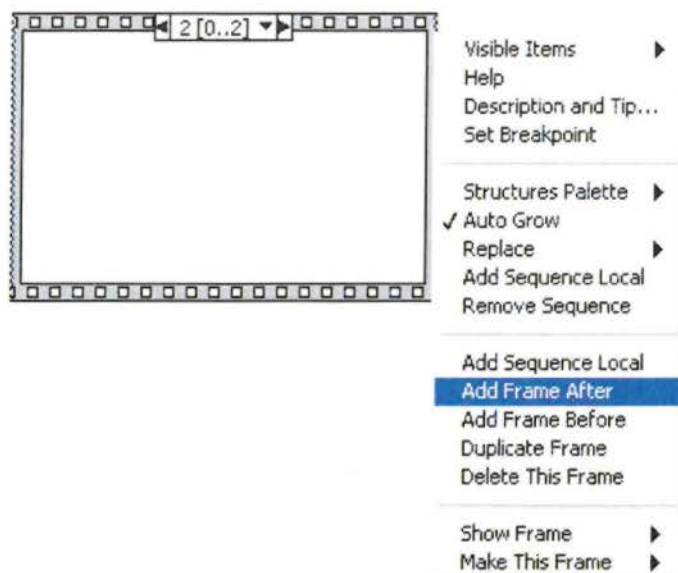
Εικόνα 2.7: Pop-Up μενού

#### 2.1.4. Η Δομή Sequence και η λειτουργία της

Τοποθετούμε την **Sequence** στο **block diagram** επιλέγοντάς την από την υποπαλέττα **Structures**. Μόλις την επιλέξουμε μας δίνεται η δυνατότητα να ορίσουμε το μέγεθός της. Αυτό επιτυγχάνεται καθώς με την βοήθεια του αριστερού ποντικιού, περικυκλώνουμε με την θηλιά το τμήμα του Διαγράμματος που μας ενδιαφέρει να επαναληφθεί.

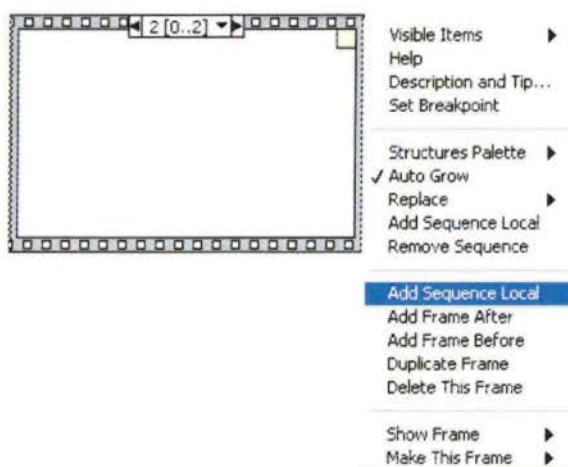
Η **Δομή Sequence** αποτελείται από πλαίσια (**frames**). Αρχικά έχει μόνο ένα πλαίσιο. Εμείς μπορούμε να προσθέσουμε κι άλλα πλαίσια από το **Pop-Up** μενού επιλέγοντας **Add frame After** ή **Add frame Before** (εικ.2.8). Τα πλαίσια εκτελούνται πάντα με την σειρά, αρχίζοντας από το 0. Τα δεδομένα μεταφέρονται από το ένα πλαίσιο στο άλλο κάνοντας χρήση τοπικής μεταβλητής. Αφού εκτελεστεί και το τελευταίο πλαίσιο μπορούν τα δεδομένα να εξέλθουν από την δομή.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 2.8: Δομή Sequence

Για να προσθέσουμε μία τοπική μεταβλητή επιλέγουμε **Add Sequence Local** από το **Pop-Up** μενού και μας εμφανίζεται ένα κίτρινο πλαίσιο.(εικ. 2.9)

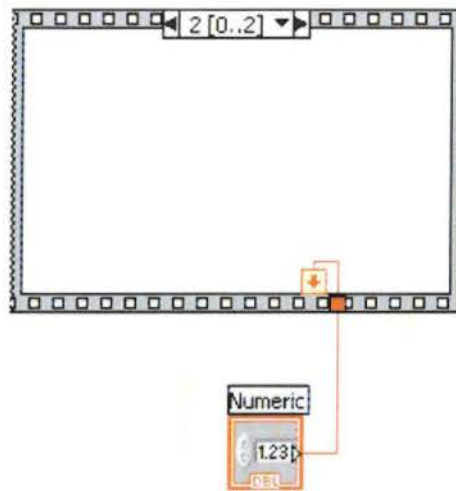


Εικόνα 2.9: Εισαγωγή μεταβλητής



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Μόλις συνδέσουμε μία πηγή δεδομένων , εμφανίζεται ένα βέλος προς τα έξω, η τοπική μεταβλητή αποθηκεύει την τιμή του **control** και την μεταφέρει στα πλαίσια όπου το βέλος του ακροδέκτη δείχνει προς τα μέσα. Στα πλαίσια που δεν θέλουμε να χρησιμοποιήσουμε την τοπική μεταβλητή, απλά το αφήνουμε ασύνδετο.(εικ. 2.10)



Εικόνα 2.10: Έξοδος δεδομένων

## 2.1.5. Η Δομή Event και η λειτουργία της

Τοποθετούμε την **Event** στο **block diagram** επιλέγοντάς την από την υποπαλέττα **Structures**. Μόλις την επιλέξουμε μας δίνεται η δυνατότητα να ορίσουμε το μέγεθός της. Η **Event** αποτελείται από ένα σύνολο γεγονότων. Αρχικά έχει μόνο ένα γεγονός. Μπορούμε να προσθέσουμε κι άλλα γεγονότα από το **Pop-Up** μενού επιλέγοντας **Add Event Case**. (εικ 2.11)

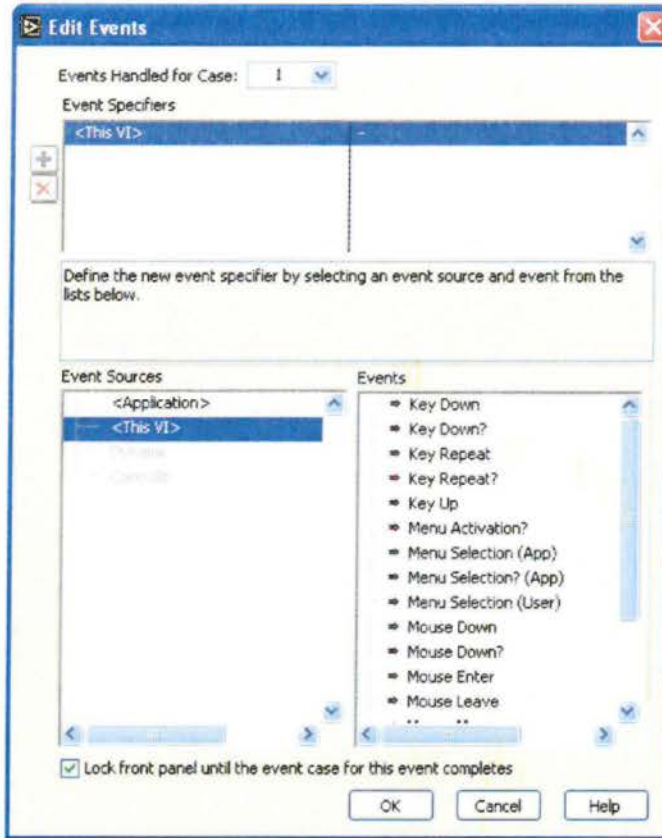


Εικόνα 2.11: Δομή Event

Στην κορυφή του πλαισίου υπάρχει ένας δείκτης, που δείχνει τον αριθμό του γεγονότος που προβάλλεται εκείνη τη στιγμή. Μέσα στις αγκύλες δείχνει το σύνολο των γεγονότων.

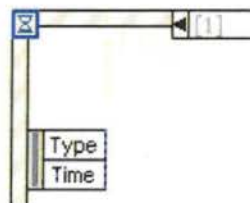
Η **Event** περιμένει έως ότου ενεργοποιηθεί ένα γεγονός από επιλεγμένο αντικείμενο στο **front panel** του **VI** και κατόπιν εκτελεί τον κώδικα του πλαισίου χρησιμοποιώντας γεγονότα δράσης, όπως μετακίνηση του ποντικιού. Σε κάθε πλαίσιο τις δομής υπάρχει στην αριστερή εσωτερική πλευρά ένας κόμβος γεγονότων (**Event Data Node**) που την ενημερώνει για το γεγονός που την κάλεσε.

Η επιλογή του είδους του γεγονότος για κάθε συγκεκριμένο πλαίσιο γίνεται από το παραθυρικό περιβάλλον επιλέγοντας την εντολή **Edit Events Handles by This Case** από το **Pop-Up** μενού (εικ. 2.12).



Εικόνα 2.12: Επιλογή είδους γεγονότος

Υπάρχει ένας χρονιστής ο οποίος ορίζει την καθυστέρηση εκτέλεσης πλαισίου **Timeout** σε περίπτωση που δεν εντοπιστεί κάποια προγραμματισμένη δράση (εικ.2.13). Δίνοντας την τιμή -1 οι λειτουργίες του πλαισίου **timeout** δεν εκτελούνται.



Εικόνα 2.13: Απεικόνιση χρονιστή

Σημείωση: Η δομή **Event** πρέπει να χρησιμοποιείται μέσα σε μία **While Loop** για τον συνεχή εντοπισμό γεγονότων.

## 2.1.6. Global

Η μεταβλητή **Global** βρίσκεται στην παλέτα **Function >> Structures**. Μας δίνεται η δυνατότητα να εισάγουμε ένα πλήθος μεταβλητών ως δεδομένα εισόδου μέσα στον κώδικα ενός **VI**.

Η μεταβλητή **Global** αποτελεί ένα ανεξάρτητο **VI** που έχει μόνο **front panel**.

Πατάμε το δεξί πλήκτρο από το ποντίκι επάνω από τη μεταβλητή και επιλέγουμε **Change to Write** και **Change to Read** για να ορίσουμε εάν η μεταβλητή **Global** διαβάζει ή καταχωρεί την τιμή ενός αριθμητικού ή λογικού δεδομένου.(εικ.2.14)

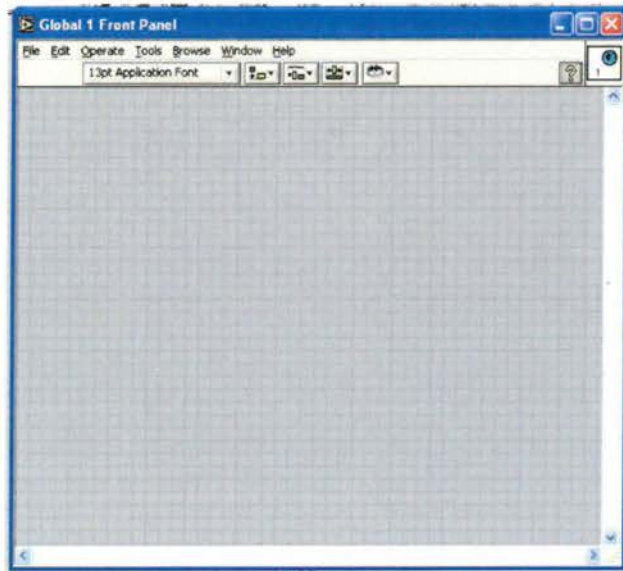


Εικόνα 2.14: Μεταβλητή Global



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

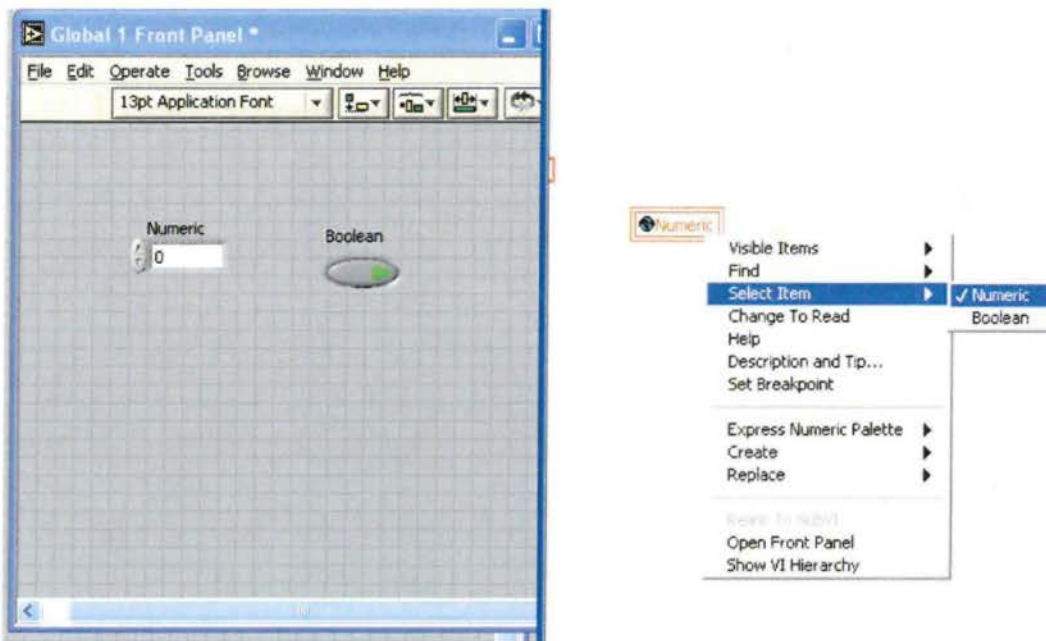
Ο προγραμματισμός μιας μεταβλητής **Global** επιτυγχάνεται με διπλό αριστερό κλικ επάνω στο εικονίδιο της και αυτομάτως ανοίγει το **front panel Global 1** για να εισάγουμε τους τύπους των μεταβλητών με μορφή αριθμητικών και λογικών **Controls**. (εικ.2.15)



Εικόνα 2.15: Front Panel μεταβλητής Global

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Η επιλογή της μεταβλητής που θέλουμε να χρησιμοποιήσουμε στον κώδικα του **block diagram** γίνεται πατώντας το δεξί πλήκτρο και επιλέγοντας την εντολή **Select Item** του **Pop-Up** μενού και κατόπιν τη συγκεκριμένη μεταβλητή που θέλουμε από τον πίνακα του πλήθους που εμφανίζεται.(εικ.2.16).



Εικόνα 2.16: Επιλογή Μεταβλητής

### 2.1.7. Local

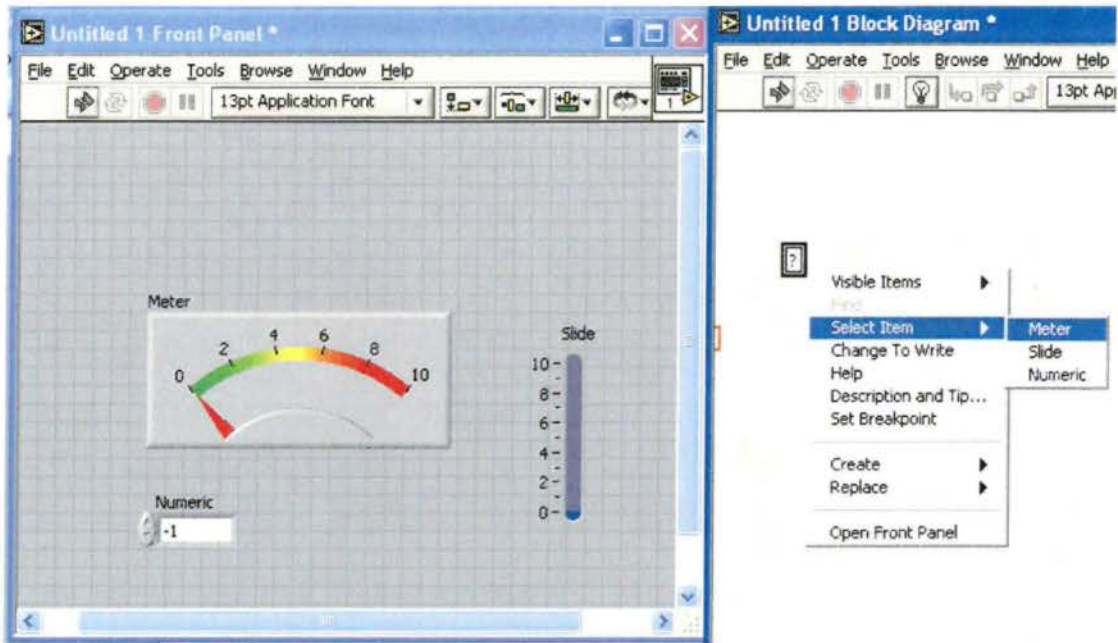
Η μεταβλητή **Local** βρίσκεται στην παλέτα **Function >> Structures**. Πατάμε το αριστερό πλήκτρο από το ποντίκι επάνω από τη μεταβλητή και επιλέγουμε **Change to Write** και **Change to Read** για να ορίσουμε εάν η μεταβλητή **Local** διαβάζει ή καταχωρεί την τιμή ενός αριθμητικού ή λογικού δεδομένου.(εικ.2.17)



Εικόνα 2.17: Μεταβλητή Local

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Επιλέγουμε την μεταβλητή που θέλουμε να χρησιμοποιήσουμε στον κώδικα του **block diagram** πατώντας το δεξί πλήκτρο από το ποντίκι και επιλέγοντας την εντολή **Select Item** του **Pop-Up** μενού και κατόπιν τη μεταβλητή από τον πίνακα των στοιχείων **control** του **front panel** που εμφανίζεται (εικ.2.18).

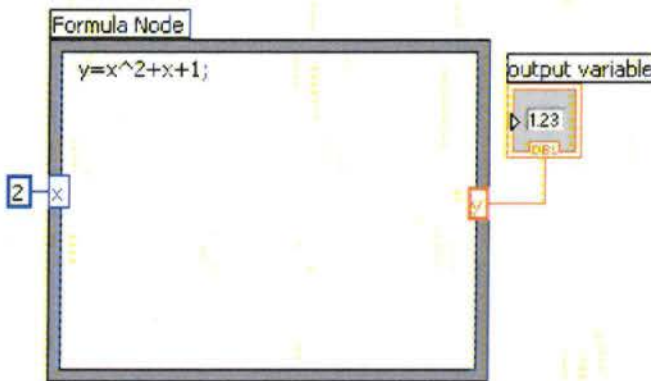


Εικόνα 2.18: Επιλογή στοιχείου μεταβλητής



### 2.1.8. Formula Node

Τοποθετούμε την **Formula Node** στο **block diagram** επιλέγοντάς την από την υποπαλέττα **Structures**. Μόλις την επιλέξουμε μας δίνεται η δυνατότητα να ορίσουμε το μέγεθός της. Η **Formula Node** μας δίνει τη δυνατότητα να εκτελούμε αλγεβρικές εξισώσεις. Η σύνταξη είναι παρόμοια με άλλες γλώσσες προγραμματισμού. Με τη βοήθεια του εργαλείου **ονοματοθέτησης (labeling)** γίνεται η εισαγωγή του κώδικα στη δομή. Στο τέλος κάθε γραμμής βάζουμε το ερωτηματικό σύμβολο (;)



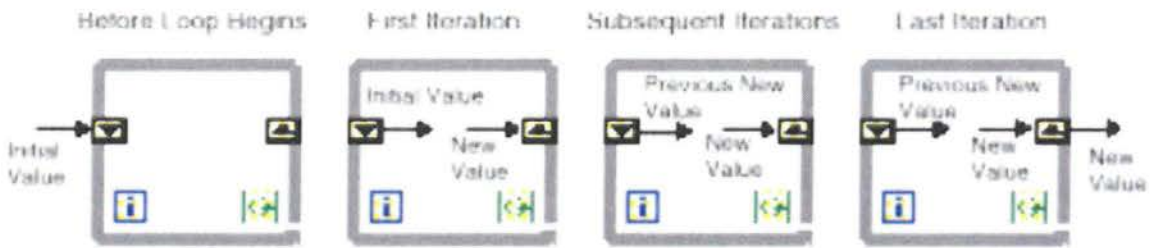
Εικόνα 2.19: Formula Node

## 2.2. Shift Registers

Οι **Shift Registers** είναι στοιχεία μνήμης όπου μπορούμε να αποθηκεύσουμε δεδομένα. Η χρησιμότητά τους έγκειται στη μεταφορά στοιχείων μεταξύ επαναλήψεων στις δομές **While Loop** και **For Loop**. Για να δημιουργήσουμε **Shift Register** πατάμε στο περίγραμμα της δομής το δεξί πλήκτρο από το ποντίκι και επιλέγουμε **Add Shift Register**.

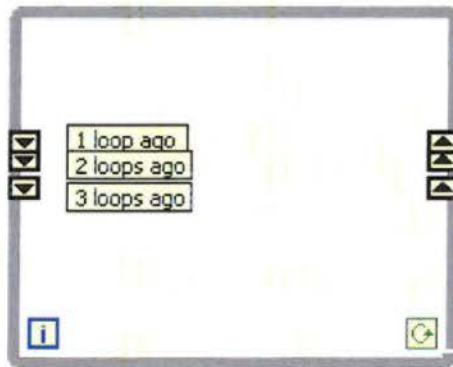
Ο **Shift Register** αποτελείται από δύο ακροδέκτες, έναν αριστερά και έναν δεξιά της δομής. Τα στοιχεία αποθηκεύονται στον δεξιό ακροδέκτη στο τέλος της κάθε επανάληψης και ύστερα μεταφέρονται στον αριστερό ακροδέκτη για να είναι διαθέσιμα για την επόμενη επανάληψη. (εικ.2.20)

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 2.20: Δομή Shift Registers

Μπορούμε να δημιουργήσουμε επιπλέον ακροδέκτες για να έχουμε πρόσβαση σε αποτελέσματα προηγούμενων επαναλήψεων. Αν για παράδειγμα τοποθετήσουμε τρεις ακροδέκτες στην αριστερή πλευρά, έχουμε πρόσβαση σε στοιχεία από τις τρεις προηγούμενες επαναλήψεις. (εικ. 2.21)



Εικόνα 2.21: Παράδειγμα Προσθήκης Ακροδεκτών

Για να προσδιορίσουμε μία συγκεκριμένη αρχική τιμή, καλωδιώνουμε την τιμή αυτή στον ακροδέκτη της αριστερής πλευράς, έξω από τη δομή ώστε να διαβαστεί πριν από την έναρξη της δομής.

**Σημείωση:** Το LabVIEW διατηρεί τις τιμές που έχει ένας **Shift Register**, μέχρι να κλείσουμε το εικονόργανο. Αν, λοιπόν εκτελούμε το εικονόργανο χωρίς να έχουμε προσδιορίσει αρχικές τιμές για τους **Shift Registers** πολλαπλές φορές, τότε οι

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

αρχικές τιμές για τις επόμενες επαναλήψεις θα καθορίζονται από τις προηγούμενες εκτελέσεις του εικονόργανου.

### 3.1. Πίνακες

Ένας πίνακας απαρτίζεται από δεδομένα του ίδιου τύπου. Για να προσπελάσουμε κάποιο δεδομένο ή στοιχείο ενός πίνακα θα πρέπει να μας είναι γνωστή η ακριβή θέση του στοιχείου στον πίνακα. Με άλλα λόγια χρειαζόμαστε έναν **δείκτη θέσης** του στοιχείου. Ένας πίνακας μπορεί να έχει πολλαπλές διαστάσεις, όπου κάθε διάσταση μπορεί να περιέχει έως  $2^{31}$  στοιχεία οποιουδήποτε τύπου. Αν υποθέσουμε ότι έχουμε  $N$  στοιχεία ανά διάσταση, τότε ο δείκτης θέσης κυμαίνεται στο διάστημα  $(0 \dots N-1)$ . Για τον μονοδιάστατο πίνακα ο δείκτης του πρώτου στοιχείου είναι μηδέν (εικ.3.1).

Δείκτης                    0    1    2    3    4    5    6    7    8    9

Στοιχεία	2.7	1.5	3.2	8.7	2.9	5.8	6.2	4.6	9.5	7.4
----------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

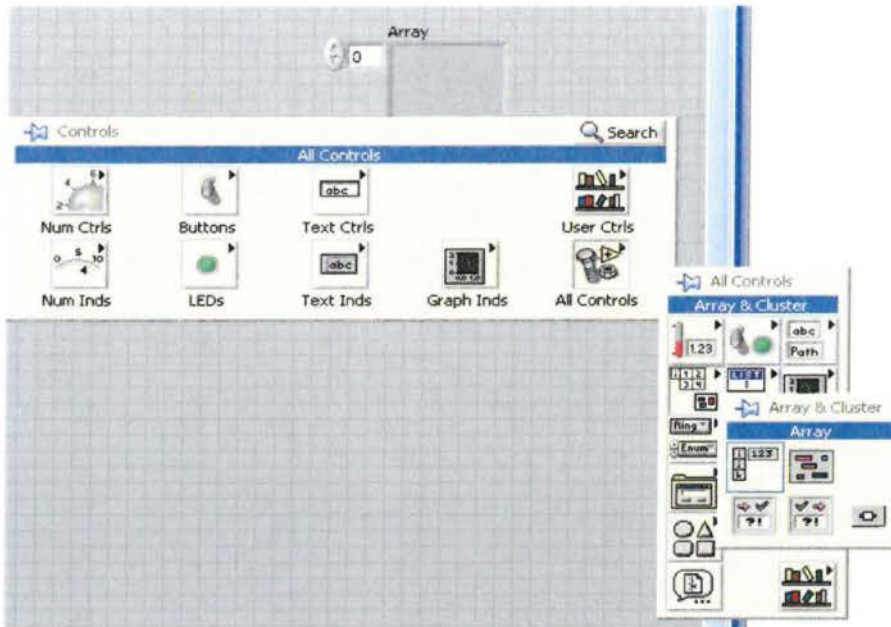
Εικόνα 3.1: Μονοδιάστατος πίνακας 1

#### Κατασκευή πίνακα:

##### 1η Κίνηση

Επιλέγουμε από την παλέτα πινάκων και συστάδων ένα κέλυφος πίνακα και το τοποθετούμε στο παράθυρο των γραφικών (εικ.3.2).

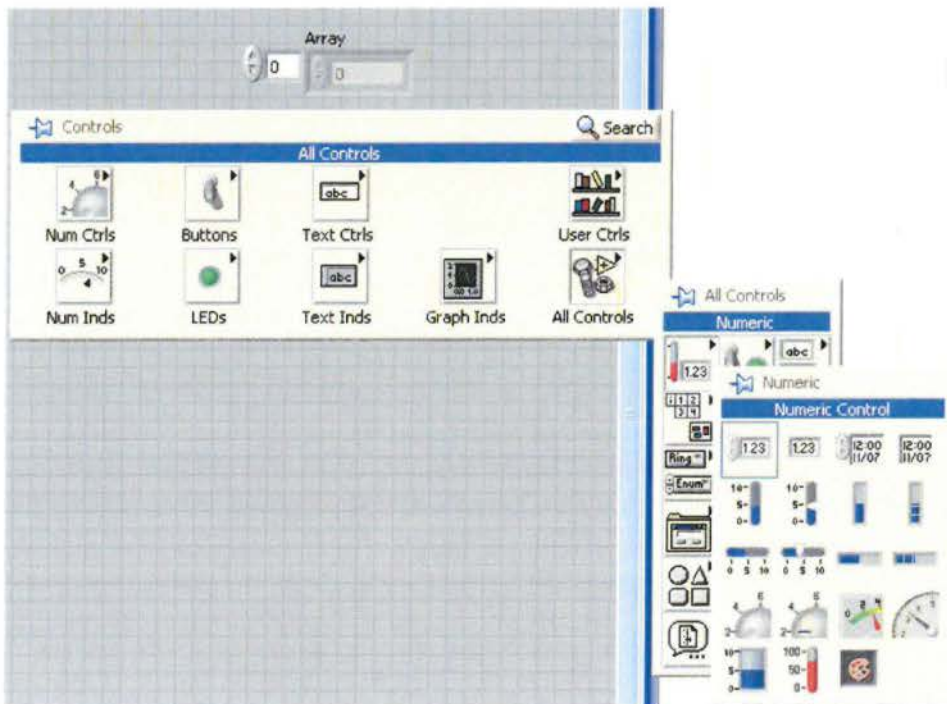
# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 3.2: Κατασκευή πίνακα

## 2<sup>η</sup> Κίνηση

Για την ολοκλήρωση της κατασκευής του πίνακα, πατάμε το δεξί πλήκτρο από το ποντίκι στο άδειο κέλυφος, επιλέγουμε τον τύπο των δεδομένων από τις υποπαλέτες και τοποθετούμε το αντικείμενο πάνω στο κέλυφος. (εικ.3.3).



Εικόνα 3.3: Επιλογή στοιχείων πίνακα



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Σημείωση: Απαιτείται να έχουμε προσδιορίσει τον τύπο των δεδομένων που μεταφέρει ο πίνακας, προτού τον χρησιμοποιήσουμε στο **block diagram**.

### 3.1.1. Δι-διάστατοι Πίνακες

Στη περίπτωση που έχουμε έναν δι-διάστατο πίνακα απαιτούνται δύο δείκτες θέσης για κάθε στοιχείο. Ο ένας δείκτης καθορίζει σε ποια σειρά βρίσκεται το στοιχείο και ο άλλος δείκτης σε ποια στήλη βρίσκεται το στοιχείο.

Βλέπουμε στην διπλανή εικόνα ένα πίνακα 4 σειρών και 5 στηλών.

Μπορούμε να προσθέσουμε διαστάσεις σε έναν πίνακα με δεξί πλήκτρο του ποντικιού πάνω στο αντικείμενο και επιλογή **Add Dimension**.

0    1    2    3    4

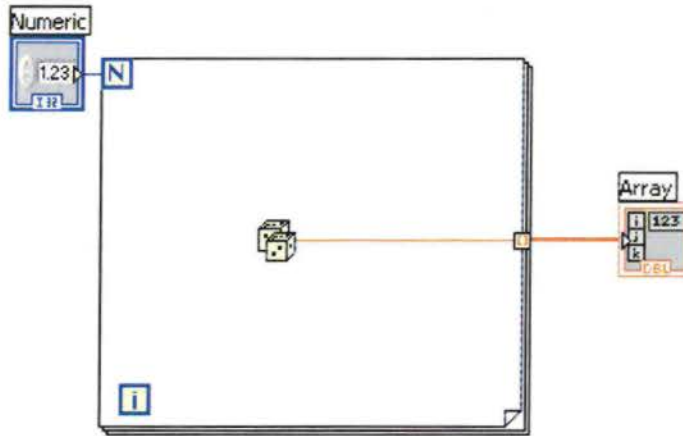
0				
1				
2				
3				

Εικόνα 3.4: Πίνακας δύο διαστάσεων

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

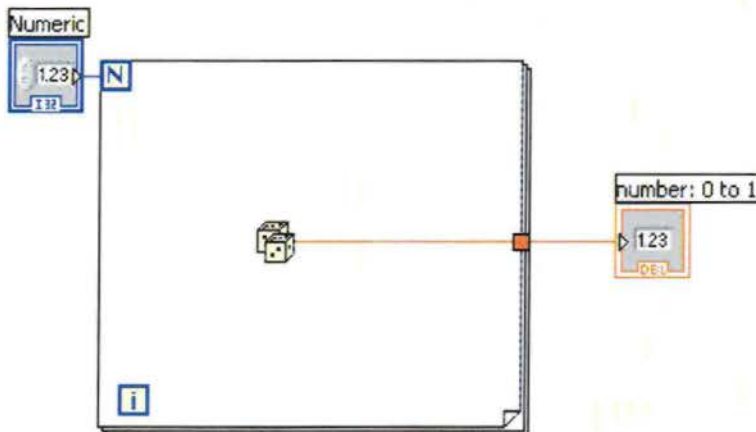
Οι δομές **For Loop** και **While Loop** μπορούν αυτόματα να δημιουργήσουν πίνακες στα άκρα τους, μια ιδιότητα γνωστή ως αυτόματος δείκτης (**Auto-Indexing**).

Βλέπουμε στη παρακάτω εικόνα το διάγραμμα με τη δομή **For** η οποία σε κάθε επανάληψη δημιουργεί ένα νέο στοιχείο του πίνακα. Όταν ολοκληρωθεί η εκτέλεση της δομής τα στοιχεία του πίνακα εξέρχονται από τη δομή. Παρατηρούμε την αλλαγή του πάχους της σύνδεσης καθώς από το καλώδιο μεταφέρεται ολόκληρος ο πίνακας.



Εικόνα 3.5: Δημιουργία πίνακα με For

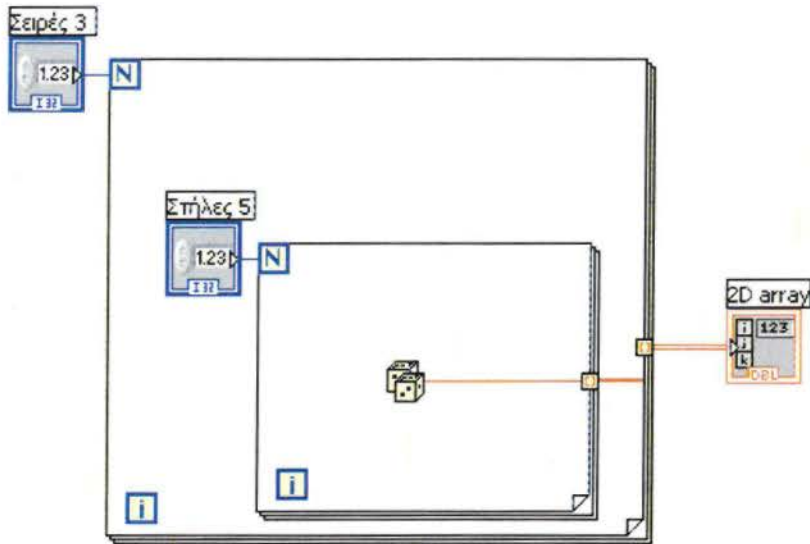
Αν θέλουμε μόνο το τελευταίο στοιχείο να μεταφέρουμε αντί ολόκληρο τον πίνακα, θα πρέπει να απενεργοποιήσουμε την ιδιότητα αυτόματου δείκτη πατώντας δεξί πλήκτρο πάνω από το καλώδιο και επιλέγοντας **Disable Indexing**. (εικ.3.6)



Εικόνα 3.6: Απενεργοποίηση αυτόματου δείκτη

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Χρησιμοποιώντας μια **For** μέσα σε μία άλλη **For** κατασκευάζουμε έναν δι-διάστατο πίνακα. Η εξωτερική **For** φτιάχνει τον αριθμό των σειρών και η εσωτερική **For** φτιάχνει τον αριθμό των στηλών (εικ.3.7).

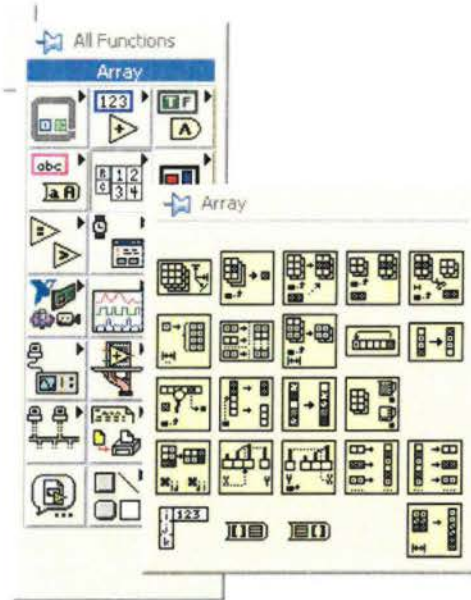


Εικόνα 3.7: Κατασκευή δι-διάστατου πίνακα με χρήση δύο For

Σημείωση: Όταν περνάμε έναν πίνακα μέσα σε μία **For** δεν χρειάζεται να συνδέσουμε τον δείκτη επανάληψης λόγω της ιδιότητας αυτομάτου δείκτη. Η δομή καθιστά τον αριθμό επαναλήψεων ίσο με το μέγεθος του πίνακα. Αν περάσουμε δύο πίνακες διαφορετικού μεγέθους ο αριθμός επαναλήψεων ισούται με το μέγεθος του μικρότερου πίνακα.

### 3.1.2. Επεξεργασία Πινάκων

Στην υποπαλέτα **Array** υπάρχουν διαθέσιμες λειτουργίες για επεξεργασία πινάκων (εικ.3.8).

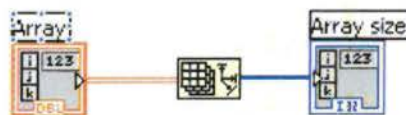


Εικόνα 3.8: Υποπαλέτα Array

Ας εξετάσουμε τις βασικότερες λειτουργίες επεξεργασίας πινάκων:

#### ➤ Μέγεθος Πίνακα

Η λειτουργία αυτή εμφανίζει τον αριθμό των στοιχείων του πίνακα. Στην είσοδο της λειτουργίας συνδέουμε τον πίνακα και στην έξοδο της λειτουργίας συνδέουμε έναν αριθμητικό **indicator** ή έναν πίνακα με δύο στοιχεία αν πρόκειται για **2D πίνακα**.



#### ➤ Ευρετήριο Πίνακα

Η λειτουργία αυτή μας επιτρέπει να προσπελάσουμε τα στοιχεία του πίνακα. Το ευρετήριο παίρνει της διαστάσεις του πίνακα μόλις τον συνδέσουμε. Χρησιμοποιούμε



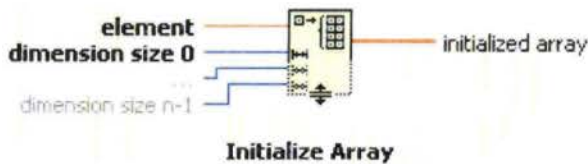
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

ένα **control** για να επιλέξουμε το στοιχείο και έναν **indicator** για να προβάλουμε την τιμή του στοιχείου.

Σημείωση: Αν συνδέσουμε πίνακα με **n διαστάσεις**, τότε συνδέουμε **n αριθμούς ευρετηρίου**. Ένα για κάθε διάσταση.

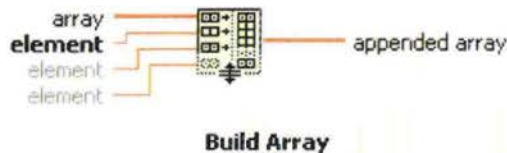
### ➤ Αρχικοποίηση πίνακα

Η λειτουργία αυτή μας επιτρέπει να εισάγουμε στοιχεία σε έναν πίνακα με συγκεκριμένη τιμή. Έχει δύο εισόδους, στη μία ορίζουμε τον αριθμό των στοιχείων του πίνακα και στην άλλη την τιμή που θέλουμε να εισάγουμε στα στοιχεία



### ➤ Δημιουργία πίνακα

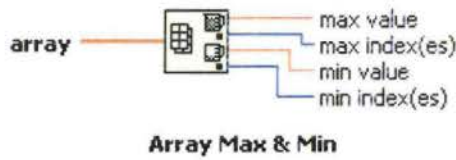
Η λειτουργία αυτή πραγματοποιεί συγχώνευση στοιχείων πινάκων σε ένα πίνακα. Στην είσοδο της λειτουργίας μπορούμε να συνδέσουμε πίνακες ή άλλα **controls**. Τα στοιχεία των πινάκων πρέπει να είναι και αυτά **controls**. Στη περίπτωση που στην είσοδο συνδέσουμε και άλλα αντικείμενα εκτός από πίνακες τότε όλοι οι πίνακες εισόδου και εξόδου πρέπει να είναι μονοδιάστατοι. Η τοποθέτηση των στοιχείων υλοποιείται με την σειρά ξεκινώντας με τα στοιχεία του πρώτου πίνακα.



### ➤ Μέγιστο και Ελάχιστο

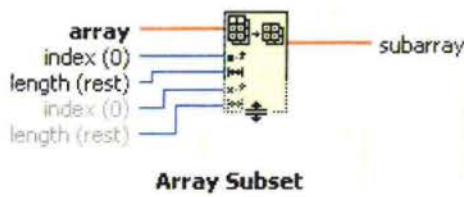
Αναζητά τη μέγιστη και ελάχιστη τιμή σε έναν αριθμητικό πίνακα και επιστρέφει τον δείκτη όπου βρίσκεται. Εάν πρόκειται για μονοδιάστατο πίνακα τα αποτελέσματα των δεικτών είναι κλιμακωτοί ακέραιοι αριθμοί. Αν πρόκειται για δι-διάστατο πίνακα τα αποτελέσματα είναι μονοδιάστατοι πίνακες που περιέχουν τις τιμές των δεικτών.

# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



## ➤ Υποσύνολο Πίνακα

Επιστρέφει ένα σύνολο από τον πίνακα με πρώτο στοιχείο το στοιχείο που δηλώνει ο δείκτης (**Index**) έως τα στοιχεία που δηλώνει η παράμετρος μήκος (**length**).



## ➤ Πίνακας σε Συστάδα

Η λειτουργία αυτή μετατρέπει έναν μονοδιάστατο πίνακα σε μία συστάδα στοιχείων του ίδιου τύπου με τα στοιχεία του πίνακα.



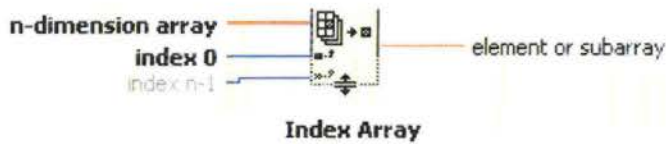
## ➤ Συστάδα σε Πίνακα

Η λειτουργία αυτή μετατρέπει μία συστάδα όμοιων στοιχείων με έναν μονοδιάστατο πίνακα στοιχείων του ίδιου τύπου.



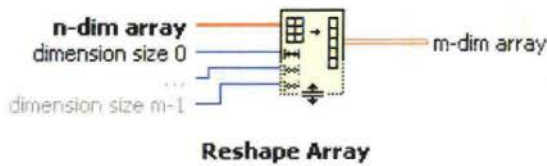
## ➤ Στοιχείο Πίνακα με Δείκτη

Η λειτουργία αυτή επιστρέφει το στοιχείο του πίνακα που δηλώνει ο δείκτης (**index**).



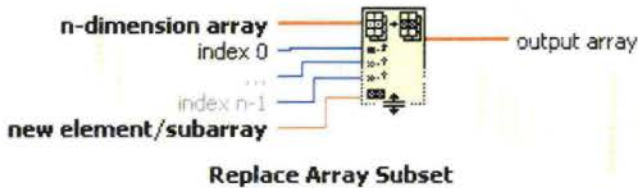
➤ **Αναδιαμόρφωση Πίνακα**

Η λειτουργία αυτή μας επιτρέπει να αλλάξουμε έναν μονοδιάστατο πίνακα σε δι-διάστατο και αντίστροφα. Μπορούμε ακόμα να την χρησιμοποιήσουμε για να αυξήσουμε και να μειώσουμε το μέγεθος ενός μονοδιάστατου πίνακα.



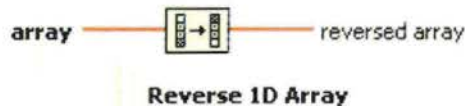
➤ **Αντικατάσταση Ενός Στοιχείου του Πίνακα**

Η λειτουργία αυτή αντικαθιστά ένα στοιχείο του πίνακα που δηλώνει ο δείκτης (**index**) με ένα νέο στοιχείο (**new element**).



➤ **Αντίστροφη 1D Πίνακα**

Η λειτουργία αυτή αντιστρέφει τη σειρά των στοιχείων ενός μονοδιάστατου πίνακα.



➤ **Περιστροφή 1D Πίνακα**

Η λειτουργία αυτή περιστρέφει τα στοιχεία σε έναν μονοδιάστατο πίνακα προς την κατεύθυνση που υποδεικνύεται από το δείκτη **n**.



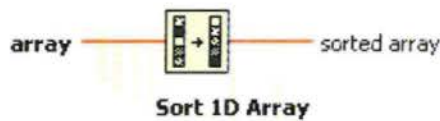
➤ Αναζήτηση σε 1D Πίνακα

Πραγματοποιεί αναζήτηση στοιχείου σε έναν μονοδιάστατο πίνακα ξεκινώντας από τη σειρά που δηλώνει ο δείκτης έναρξης (start index).



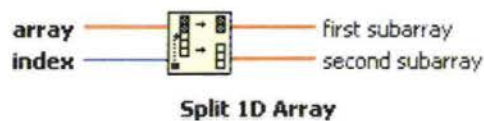
➤ Ταξινόμηση 1D Πίνακα

Πραγματοποιεί ταξινόμηση των στοιχείων του πίνακα κατά αύξουσα σειρά.



➤ Διαίρεση 1D Πίνακα

Πραγματοποιεί διαίρεση του πίνακα στη σειρά του στοιχείου που δηλώνει ο δείκτης και επιστρέφει δύο σύνολα.

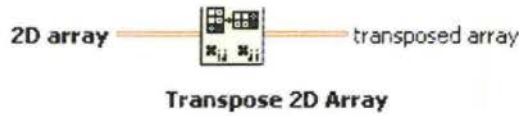


➤ Αντιμετάθεση σε 2D Πίνακα



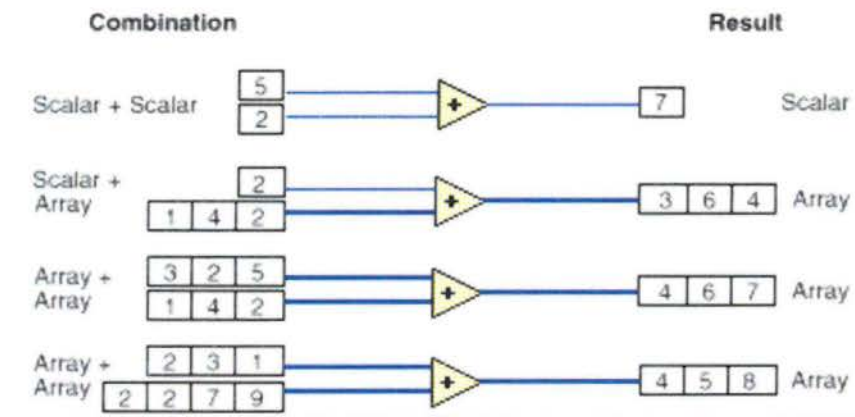
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Ρυθμίζει εκ νέου τα στοιχεία ενός δι-διάστατου πίνακα με τη μετατιθέμενη μορφή από πίνακα  $[ I, j ]$  σε πίνακα  $[ j, I ]$ .



### 3.1.3. Πολυμορφισμός

Όλες οι αριθμητικές ρουτίνες του **LabVIEW** είναι πολυμορφικές. Μπορούμε επομένως να προσθέσουμε ένα διαβαθμισμένο μέγεθος σε ένα διάνυσμα ή να προσθέσουμε δύο πίνακες μαζί, χωρίς να τροποποιήσουμε την λειτουργία.



Εικόνα 3.10: Παράδειγμα πολυμορφισμού

Στην πρώτη περίπτωση, το αποτέλεσμα είναι ένα βαθμωτό μέγεθος. Στην δεύτερη, η βαθμωτή είσοδος προστίθεται σε κάθε στοιχείο του διανύσματος. Στην τρίτη, κάθε στοιχείο του ενός διανύσματος προστίθεται στο αντίστοιχης θέσης στοιχείο του άλλου διανύσματος. Αν τα διανύσματα δεν είναι του ίδιου μεγέθους το αποτέλεσμα συμφωνεί σε μέγεθος με το μικρότερο διάνυσμα.(εικ 3.10)

## 3.2. Συστάδες (Clusters)

Το **LabVIEW** μας δίνει την δυνατότητα να ομαδοποιήσουμε δεδομένα διαφορετικού τύπου σε ένα αντικείμενο με τη χρήση συστάδων (**Cluster**). Οι βασικές δομές των

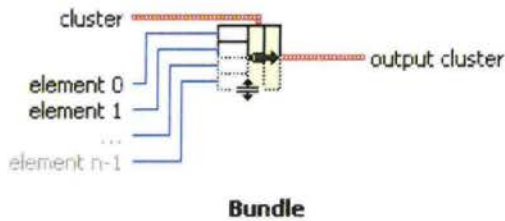
συστάδων είναι δύο: η **Bundle** που ομαδοποιεί στοιχεία σε μία έξοδο και η **Unbundle** η οποία διαχωρίζει τα στοιχεία από μία συστάδα.

### 3.2.1. Bundle

Η συστάδα **Bundle** έχει μία έξοδο στην οποία βρίσκονται ομαδοποιημένα τα στοιχεία που δέχεται στις εισόδους της (εικ 3.11). Στην έξοδο συνδέουμε μία συστάδα ή ένα γράφημα.

Προσθέτουμε εισόδους από το **Pop-Up** μενού επιλέγοντας **Add Input**.

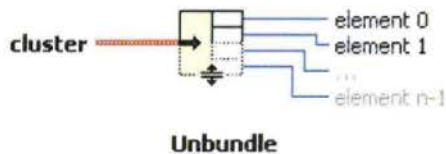
Εκτός από τις εισόδους στα αριστερά έχει και τη μεσαία είσοδο για συστάδα. Η είσοδος αυτή χρησιμεύει στη περίπτωση που θέλουμε να αλλάξουμε την τιμή ενός στοιχείου της συστάδας, χωρίς να επηρεάσουμε τα άλλα, το συνδέουμε με το **Bundle** και στη συνέχεια συνδέουμε ένα **control** στη είσοδο που αντιστοιχεί στο στοιχείο που θέλουμε να ελέγχουμε.



Εικόνα 3.11: Απεικόνιση Συστάδας Bundle

### 3.2.2. Unbundle

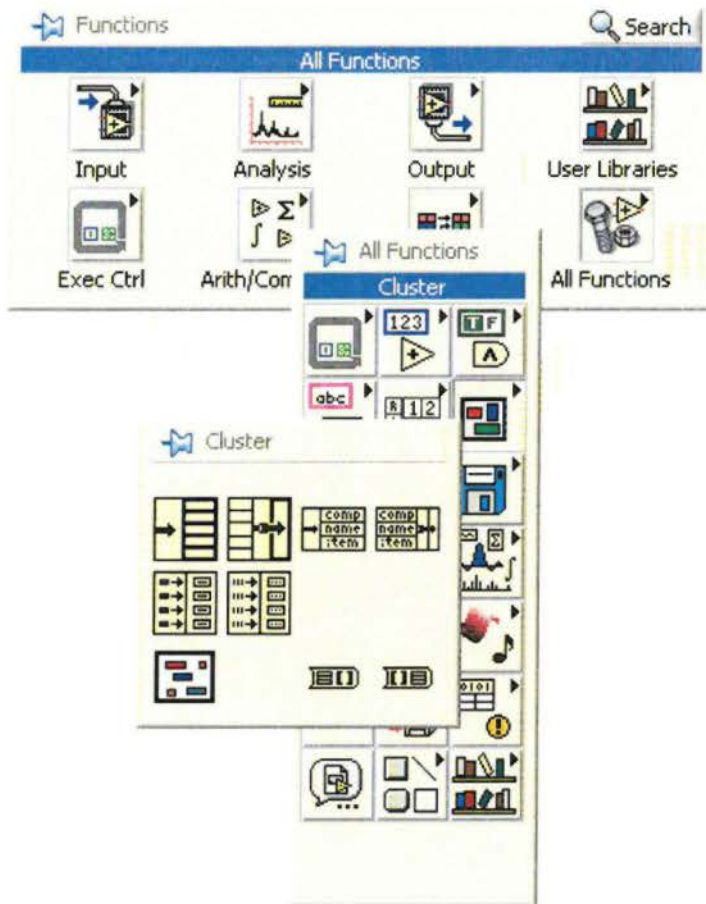
Η λειτουργία **Unbundle** διαχωρίζει τα στοιχεία από μία συστάδα (εικ. 3.12). Θα πρέπει να ορίσουμε τόσες εξόδους όσα είναι τα στοιχεία στη συστάδα. Προσθέτουμε εξόδους από το **Pop-Up** μενού επιλέγοντας **Add Element**.



Εικόνα 3.12: Απεικόνιση Συστάδας Unbundle

### 3.2.3. Λειτουργίες Συστάδων (Clusters)

Στη παρακάτω εικόνα 3.13 βλέπουμε τις διαθέσιμες λειτουργίες συστάδων.



Εικόνα 3.13: Παλέτα Συστάδων

#### Περιγραφή των Λειτουργιών Συστάδων

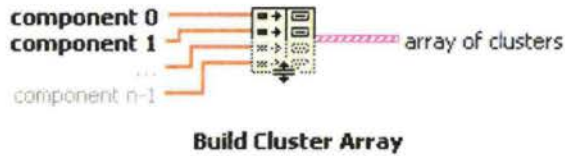
##### ➤ Πίνακας σε συστάδα

Μετατρέπει έναν μονοδιάστατο πίνακα σε μία συστάδα στοιχείων του ίδιου τύπου.



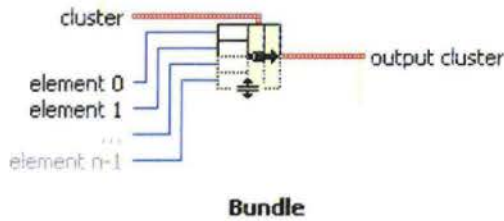
➤ **Δημιουργία Συστάδας**

Ομαδοποιεί όλα τα στοιχεία σε έναν πίνακα συστάδων.



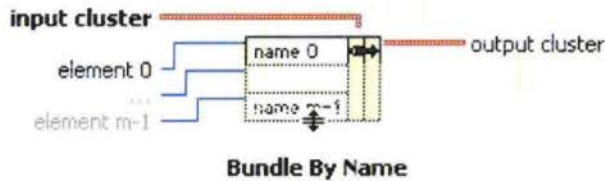
➤ **Δέσμη**

Προσθέτει τα στοιχεία μίας συστάδας στο σύνολο μιας νέας.



➤ **Δέσμη με Όνομα**

Η ομαδοποίηση στη συστάδα γίνεται με χρήση ταυτότητας ονόματος του κάθε στοιχείου εισαγωγής σε αυτή.



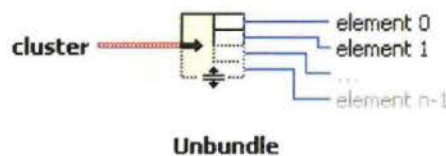
➤ **Συστάδα σε Πίνακα**

Μετατρέπει μια συστάδα όμοιων στοιχείων σε ένα μονοδιάστατο πίνακα .



➤ **Από-Ομαδοποίηση Συστάδας**

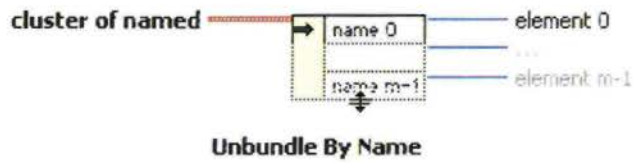
Από-Ομαδοποιεί μια συστάδα στα επιμέρους στοιχεία της (Components).





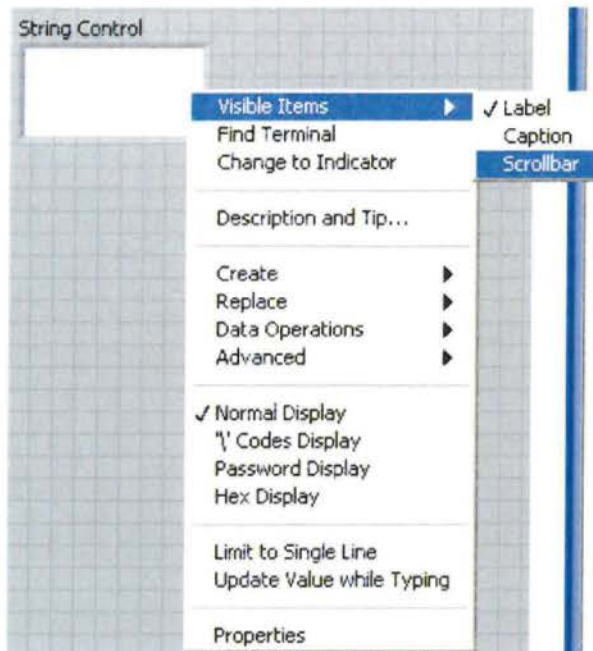
➤ **Από-Ομαδοποίηση με Όνομα**

Από-ομαδοποιεί τα στοιχεία της συστάδας με επιλογή της ταυτότητας του ονόματος με το οποίο ομαδοποιήθηκε.



## 4.1 Χαρακτήρες (Strings)

Μια ακολουθία χαρακτήρων (**String**) είναι μια συλλογή **ASCII** χαρακτήρων. Για να εισάγουμε χαρακτήρες στο **control** κειμένου, χρησιμοποιούμε το εργαλείο ονοματοθέτησης ( **Labeling Tool**). Για να μεγαλώσουμε τον χώρο του **control** κειμένου δημιουργούμε την κυλιόμενη μπάρα από το **Pop-Up** μενού επιλέγοντας **Visible Items >> Scrollbar**.(εικ.4.1)



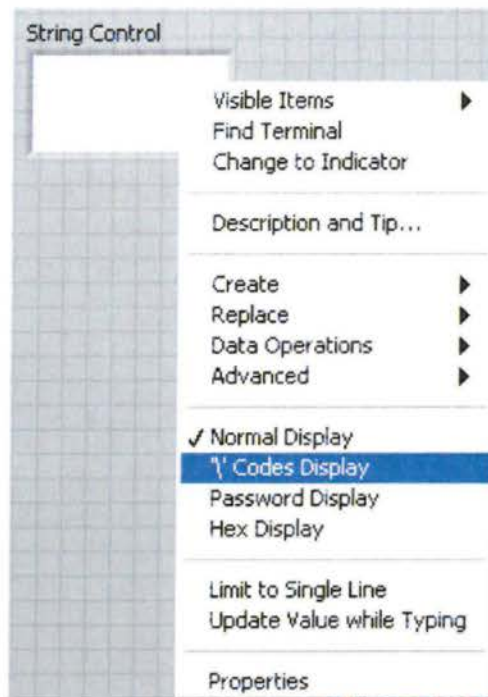
Εικόνα 4.1: Εμφάνιση κυλιόμενης μπάρας

### 4.1.1. Τύποι Χαρακτήρων

Στο **control** κειμένου μπορούμε να εισάγουμε χαρακτήρες διαφόρων τύπων, όπως δεκαεξαδικούς αριθμούς, μυστικούς κωδικούς και μη ορατούς χαρακτήρες, όπως το **Tab**, το **Enter** και το **Space**.

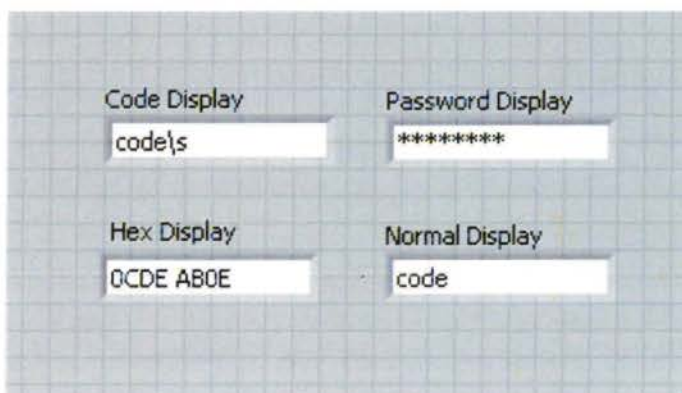
Επιλέγουμε των τύπο χαρακτήρων από το **Pop-Up** μενού.(εικ.4.2)

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 4.2: Εμφάνιση παλέτας λειτουργιών χαρακτήρων

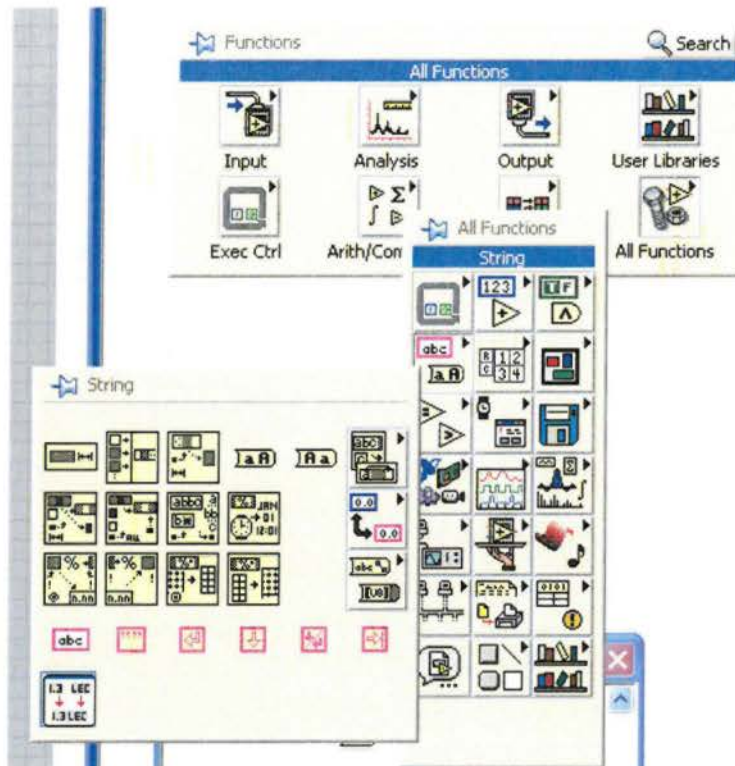
- Η επιλογή **Password Display** (εμφάνιση κώδικα) εισάγει χαρακτήρες στο **control** κειμένου με την μορφή (αστεράκι) ' \* ', ενώ το **LabVIEW** τους διαβάζει κανονικά.
- Η επιλογή **Code Display** (μη ορατοί χαρακτήρες) εμφανίζονται όταν τελειώνουμε και βγαίνουμε από το **control** κειμένου και όχι όταν πληκτρολογούμε. Για παράδειγμα το **Enter** παρουσιάζεται ως \n και το **space** \s.
- Η επιλογή **Hex Display** εμφανίζει τους χαρακτήρες σε δεκαεξαδική παράσταση.



Εικόνα 4.3: Μορφές εμφάνισης χαρακτήρων

## 4.1.2. Λειτουργίες Χαρακτήρων

Στη παρακάτω εικόνα παρουσιάζεται η παλέτα λειτουργιών χαρακτήρων.(εικ.4.4)



Εικόνα 4.4: Εμφάνιση παλέτας λειτουργιών χαρακτήρων

### Περιγραφή Λειτουργιών Χαρακτήρων

#### ➤ Παράσταση Λογιστικού Φύλλου (Spreadsheet)

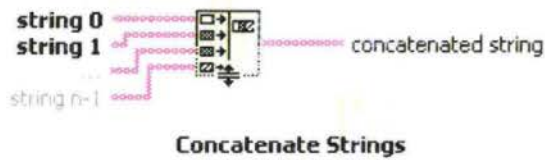
Μετατρέπει τα στοιχεία ενός πίνακα σε χαρακτήρες που θα χρησιμοποιηθούν σε παράσταση λογιστικού φύλλου (**Spreadsheet**).



#### ➤ Τοποθέτηση Χαρακτήρων σε Σειρά

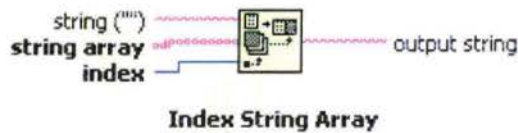
Συνδέει τους χαρακτήρες σε μορφή μονοδιάστατου πίνακα





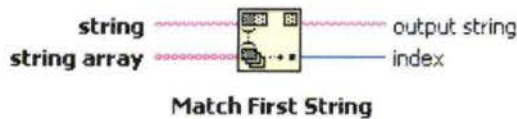
➤ **Εύρεση και Αντικατάσταση Χαρακτήρων**

Αντικαθιστά τον χαρακτήρα που ορίζεται από τον δείκτη του πίνακα με το νέο.



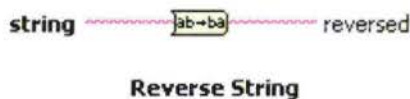
➤ **Εύρεση Δείκτη**

Συγκρίνει κάθε χαρακτήρα του πίνακα με το χαρακτήρα εύρεσης και τον εξάγει σε κάθε αντιστοιχία με τον δείκτη τοποθέτησης του μέσα από το πίνακα.



➤ **Αντιστροφή Χαρακτήρων**

Αντιστρέφει τους χαρακτήρες μιας σειράς



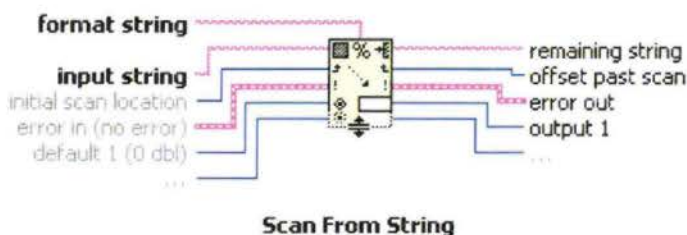
➤ **Περιστροφή Χαρακτήρων κατά μία Θέση**

Μετατοπίζει τους χαρακτήρες μιας σειράς κατά μία θέση μπροστά. Ο πρώτος χαρακτήρας μπαίνει στη τελευταία θέση.



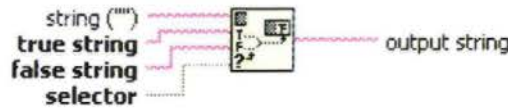
➤ **Αντικατάσταση Μορφής (Format) σε Σειρά Χαρακτήρων**

Μετατρέπει τη μορφή κάθε χαρακτήρα σύμφωνα με το format επιλογής μας.



➤ **Επιλογή True και False Χαρακτήρα**

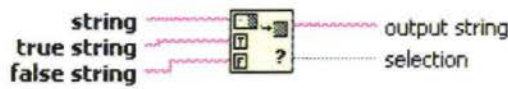
Επιλέγει ως έξοδο είτε τη ψεύτικη είτε την αληθινή σειρά χαρακτήρων σύμφωνα με τον λογικό (**boolean**) επιλογή.



**Append True/False String**

➤ **Αληθής ή Ψευδής Χαρακτήρας**

Εξετάζει την αρχική σειρά χαρακτήρων και ελέγχει εάν ταιριάζει με την αληθινή ή την ψεύτικη σειρά χαρακτήρων. Επιστρέφει μια λογική αλήθειας ή ψεύδους στην επιλογή.



**Match True/False String**

➤ **Μήκος Χαρακτήρων**

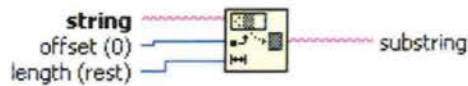
Επιστρέφει τον αριθμό των χαρακτήρων από μία σειρά.



**String Length**

➤ **Υποσύνολο Σειράς**

Επιστρέφει ένα σύνολο χαρακτήρων από μια σειρά από τον χαρακτήρα που δηλώνει η συνθήκη **offset** έως το μήκος **length**.



**String Subset**

➤ **Μετατροπή Κεφαλαίων Χαρακτήρων σε Πεζούς**

Μετατρέπει τους κεφαλαίους αλφαβητικούς χαρακτήρες σε πεζούς.



**To Lower Case**

➤ **Μετατροπή Πεζών Χαρακτήρων σε Κεφαλαίους**

Μετατρέπει τους πεζούς αλφαβητικούς χαρακτήρες σε κεφαλαίους.



**To Upper Case**

➤ **Μετατροπή και Επισύναψη**

Μετατρέπει έναν αριθμό σε χαρακτήρα σύμφωνα με την επιλογή **format** και τον επισυνάπτει στη σειρά χαρακτήρων



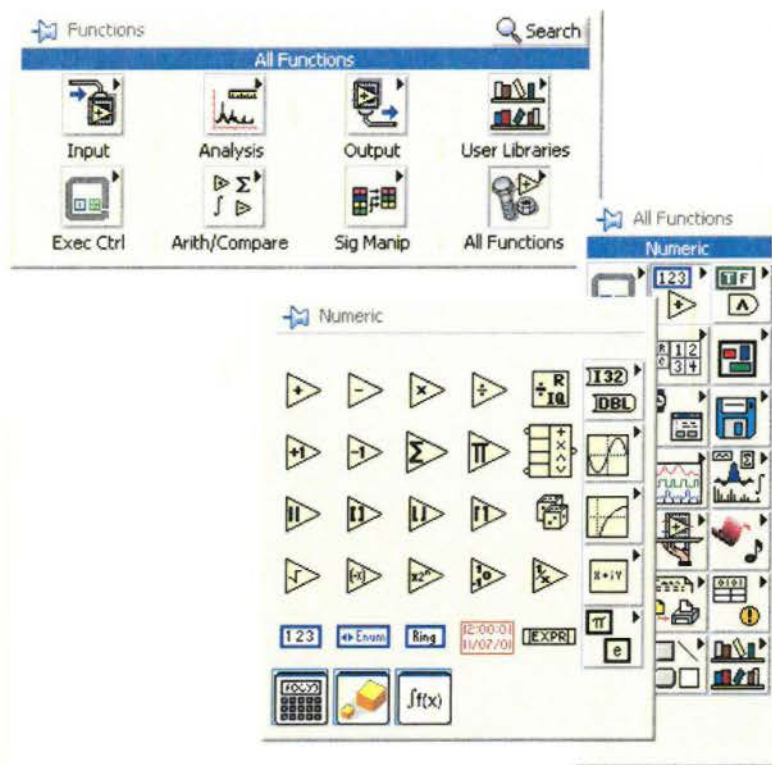
➤ **Σταθερά Σειράς**

Εισάγει σειρά χαρακτήρων με την πληκτρολόγηση τους. Πατώντας το δεξί πλήκτρο από το ποντίκι επάνω στο εικονίδιο μπορούμε να αλλάξουμε τη μορφή της εμφάνισης σε μορφή συνθήματος (**password**) ή δεκαεξαδικών χαρακτήρων.



## 4.2. Αριθμητικές Λειτουργίες

Στην ακόλουθη εικόνα βλέπουμε τις διαθέσιμες λειτουργίες της αριθμητικής παλέτας.(εικ.4.5)



Εικόνα 4.5: Εμφάνιση Αριθμητικής παλέτας

### Περιγραφή Αριθμητικών Λειτουργιών

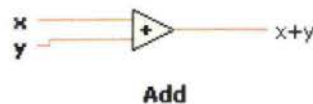
#### ➤ Απόλυτη τιμή

Επιστρέφει την απόλυτη τιμή της εισόδου [abs(x)].



#### ➤ Άθροισμα

Υπολογίζει το άθροισμα των εισόδων (x+y).





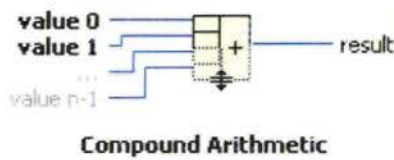
➤ **Άθροισμα των Στοιχείων ενός Πίνακα**

Επιστρέφει το άθροισμα όλων των στοιχείων ενός πίνακα.



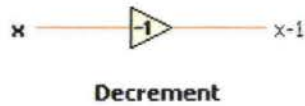
➤ **Αριθμητικές Πράξεις**

Εκτελεί αριθμητικές πράξεις (πολλαπλασιασμού, πρόσθεσης, αφαίρεσης) σε δύο ή περισσότερους αριθμούς ή σε εισόδους λογικών τιμών.



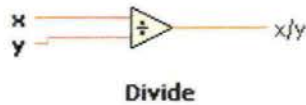
➤ **Μείωση**

Αφαιρεί 1 από την τιμή εισόδου.



➤ **Διαίρεση**

Υπολογίζει το πηλίκο των εισόδων ( $x/y$ ).



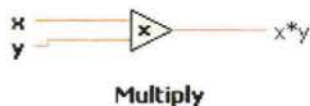
➤ **Αύξηση**

Προσθέτει 1 στην τιμή εισόδου.



➤ **Γινόμενο**

Υπολογίζει το γινόμενο των εισόδων ( $x*y$ ).



- **Πολλαπλασιασμός των Στοιχείων ενός Πίνακα**  
Υπολογίζει το γινόμενο των στοιχείων ενός πίνακα.



- **Πρόσημο**  
Αντιστρέφει το πρόσημο της μεταβλητής εισόδου.



- **Πηλίκο και Υπόλοιπο**  
Υπολογίζει το πηλίκο των εισόδων και δίνει το υπόλοιπο.



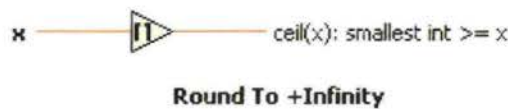
- **Τυχαίος Αριθμός (0-1)**  
Παράγει έναν τυχαίο αριθμό μεταξύ 0 – 1.



- **Πηλίκο 1/x**  
Υπολογίζει την τιμή εισόδου 1/x.



- **Στρογγυλοποίηση Υψηλότερης Τάξης**  
Στρογγυλοποιεί την τιμή της εισόδου x στον επόμενο μεγαλύτερο ακέραιο αριθμό.



- **Στρογγυλοποίηση Χαμηλότερης Τάξης**  
Στρογγυλοποιεί την τιμή της εισόδου x στον επόμενο μικρότερο ακέραιο αριθμό.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



**Round To -Infinity**

### ➤ Δύναμη του 2

Υψώνει τον αριθμό  $x$  σε δύναμη του 2 με εκθέτη  $n$ .



**Scale By Power Of 2**

### ➤ Τετραγωνική Ρίζα

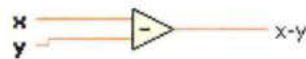
Υπολογίζει την τετραγωνική ρίζα της τιμής εισόδου  $x$  [ $\text{sqrt}(x)$ ].



**Square Root**

### ➤ Διαφορά

Υπολογίζει την διαφορά των εισόδων.



**Subtract**

### ➤ Αριθμητική Σταθερά

Εισάγει μια σταθερή αριθμητική τιμή ως δεδομένο.

123

**Numeric Constant**

### ➤ Απαριθμημένη Σταθερά

Η απαριθμημένη σταθερά δημιουργεί πίνακα με στοιχεία σταθερών αριθμών. Πατώντας δεξί πλήκτρο πάνω στο εικονίδιο επιλέγουμε **Add Item Before** για πρόσθετη εισαγωγή στοιχείου στο σύνολο της.

Enum

**Enum Constant**

### ➤ Μετατροπή Μονάδας

Μετατρέπει έναν φυσικό αριθμό (αριθμός σε μονάδα) σε έναν καθαρό αριθμό ή έναν καθαρό αριθμό σε φυσικό.



**Convert Unit**

## 4.2.1 Μετατροπή Συστήματος Αριθμών

### Περιγραφή Εντολών Μετατροπής Συστήματος Αριθμών

#### ➤ Λογικό (01)

Μετατρέπει το λογικό αληθές ή ψευδές σε ακέραιο αριθμό 1 και 0 αντίστοιχα.



**Boolean To (0,1)**

#### ➤ Μετατροπή Αριθμού σε Λογικό Πίνακα

Μετατρέπει έναν ακέραιο αριθμό στοιχείων σε λογικό πίνακα 8 ..16, ή 32 στοιχείων.



**Number To Boolean Array**

#### ➤ Πίνακας Λογικών τιμών σε Αριθμό

Μετατρέπει έναν πίνακα λογικών τιμών σε ακέραιο αριθμό.



**Boolean Array To Number**

#### ➤ Μετατροπή σε 8-bit Ακέραιο αριθμό

Μετατρέπει τον αριθμό εισόδου σε 8-bit ακέραιο αριθμό.



**To Byte Integer**

#### ➤ Μετατροπή σε 16-bit Ακέραιο αριθμό

Μετατρέπει τον αριθμό εισόδου σε 16-bit ακέραιο αριθμό.



**To Word Integer**

#### ➤ Μετατροπή σε 32-bit Ακέραιο αριθμό

Μετατρέπει τον αριθμό εισόδου σε 32-bit ακέραιο αριθμό.



**To Long Integer**



- **Μετατροπή σε 8-bit Απροσήμαστο (Unsigned) Ακέραιο αριθμό**  
Μετατρέπει τον αριθμό εισόδου σε 8-bit απροσήμαστο ακέραιο αριθμό.



**To Unsigned Byte Integer**

- **Μετατροπή σε 32-bit Απροσήμαστο (Unsigned) Ακέραιο αριθμό**  
Μετατρέπει τον αριθμό εισόδου σε 32-bit απροσήμαστο ακέραιο αριθμό.



**To Unsigned Long Integer**

- **Μετατροπή σε 16-bit Απροσήμαστο (Unsigned) Ακέραιο αριθμό**  
Μετατρέπει τον αριθμό εισόδου σε 16-bit απροσήμαστο ακέραιο αριθμό.



**To Unsigned Word Integer**

- **Διπλασιασμός Ακρίβειας**  
Μετατρέπει τον αριθμό εισόδου σε σύνθετο αριθμό διπλής- ακρίβειας.

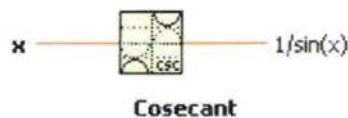


**To Double Precision Complex**

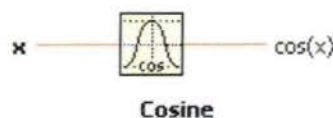
## 4.2.2. Τριγωνομετρικές Συναρτήσεις

### Περιγραφή Εντολών Τριγωνομετρικών Συναρτήσεων

- **Συντέμνουσα (Cosecant)**  
Υπολογίζει τη συντέμνουσα του  $x$  (σε ακτίνια).



- **Συνημίτονο**  
Υπολογίζει το συνημίτονο του  $x$  (σε ακτίνια).



➤ **Συνεφαπτομένη**

Υπολογίζει τη συνεφαπτομένη του  $x$  (σε ακτίνια).



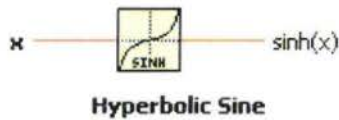
➤ **Υπερβολικό Συνημίτονο**

Υπολογίζει το υπερβολικό συνημίτονο του  $x$  (σε ακτίνια).



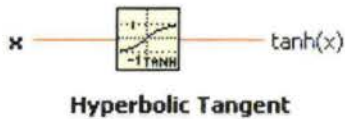
➤ **Υπερβολικό Ημίτονο**

Υπολογίζει το υπερβολικό ημίτονο του  $x$  (σε ακτίνια).



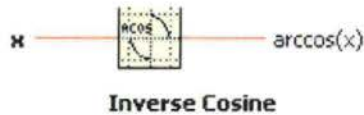
➤ **Υπερβολική Εφαπτομένη**

Υπολογίζει την υπερβολική εφαπτομένη του  $x$  (σε ακτίνια).



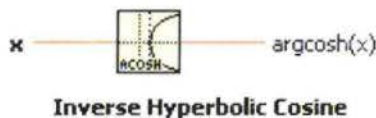
➤ **Αντίστροφο Συνημίτονο**

Υπολογίζει το arccosine του  $x$  (σε ακτίνια).



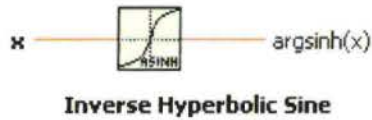
➤ **Αντίστροφο Υπερβολικό Συνημίτονο**

Υπολογίζει το υπερβολικό arccosine του  $x$  (σε ακτίνια).



➤ **Αντίστροφο Υπερβολικό Ημίτονο**

Υπολογίζει το υπερβολικό  $\operatorname{argsinh}$  του  $x$  (σε ακτίνια).



➤ **Αντίστροφη Υπερβολική Εφαπτομένη**

Υπολογίζει το υπερβολικό  $\operatorname{argtanh}$  του  $x$  (σε ακτίνια).



➤ **Αντίστροφο Ημίτονο**

Υπολογίζει το  $\operatorname{arcsin}$  του  $x$  (σε ακτίνια).



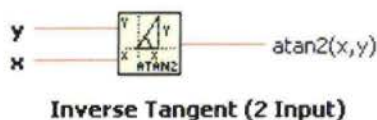
➤ **Αντίστροφη Εφαπτομένη**

Υπολογίζει το  $\operatorname{arctan}$  του  $x$  (σε ακτίνια).



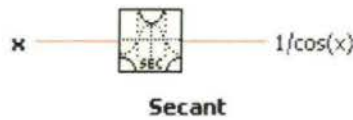
➤ **Αντίστροφη Εφαπτομένη (x,y)**

Υπολογίζει τον  $\operatorname{arctan}$   $y/x$



➤ **Τέμνουσα (Secant)**

Υπολογίζει την τέμνουσα του  $x$  (σε ακτίνια).



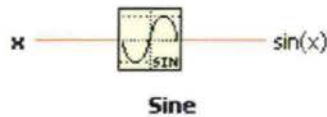
➤ **Ημίτονο x προς x**

Υπολογίζει το ημίτονο του x που διαιρείται με το x



➤ **Ημίτονο**

Υπολογίζει το ημίτονο του x (σε ακτίνια).



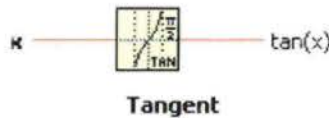
➤ **Ημίτονο και Συνημίτονο**

Υπολογίζει και το ημίτονο και το συνημίτονο του x (σε ακτίνια).



➤ **Εφαπτομένη**

Υπολογίζει την εφαπτομένη του x (σε ακτίνια).



### 4.2.3. Λογαριθμικές Συναρτήσεις

#### Περιγραφή Εντολών Λογαριθμικών Συναρτήσεων

➤ **Εκθετικός (exp)**

Υπολογίζει το  $e^x$ .



**ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.**



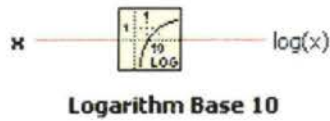
➤ **Λογάριθμος με Βάση 2**

Υπολογίζει το λογάριθμο του x με βάση 2



➤ **Λογάριθμος με Βάση 10**

Υπολογίζει το λογάριθμο του x με βάση 10



➤ **Λογάριθμος με Βάση x**

Υπολογίζει το λογάριθμο του y με βάση x



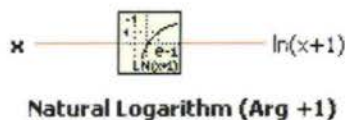
➤ **Φυσικός Λογάριθμος**

Υπολογίζει το φυσικό λογάριθμο του x [ $\ln(x)$ ].



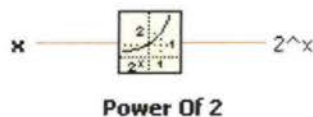
➤ **Φυσικός Λογάριθμος (Arg +1)**

Υπολογίζει το φυσικό λογάριθμο του  $\ln(x + 1)$ .



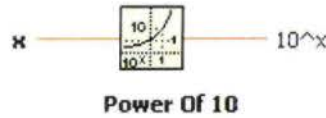
➤ **Δύναμη του 2**

Υπολογίζει τη δύναμη του 2 με εκθέτη x



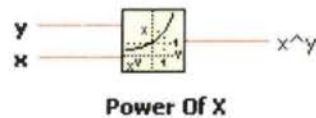
➤ **Δύναμη του 10**

Υπολογίζει τη δύναμη του 10 με εκθέτη x



➤ **Δύναμη του x**

Υπολογίζει την παράσταση x^y.



### 4.2.4. Μιγαδικές Συναρτήσεις

#### Περιγραφή Εντολών Μιγαδικών Συναρτήσεων

➤ **Αντίστροφος μιγαδικός**

Υπολογίζει τον αντίστροφο μιγαδικό του  $x + iy$



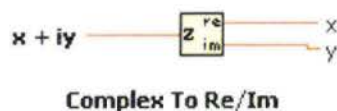
➤ **Πολικές συντεταγμένες**

Υπολογίζει τις πολικές συντεταγμένες  $\theta$  και  $r$



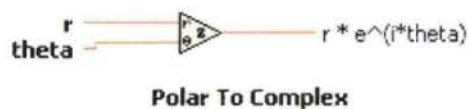
➤ **Μετατροπή μιγαδικού αριθμού σε πραγματικό και φανταστικό μέρος**

Μετατρέπει έναν μιγαδικό αριθμό σε ορθογώνιο σύστημα αριθμών.



➤ **Μετατροπή πολικού αριθμού σε μιγαδικό**

Υπολογίζει το μιγαδικό αριθμό από πολικές συντεταγμένες.



➤ **Μιγαδικός αριθμός**

Δημιουργεί έναν μιγαδικό αριθμό



## 4.2.5. Αριθμητικές Σταθερές

### Περιγραφή Αριθμητικών Σταθερών



επιστέφει την τιμή 2.7182818..



επιστρέφει τον φυσικό λογάριθμο του π (1.14472..)



φυσικός λογάριθμος του 2



φυσικός λογάριθμος του 10



επιστρέφει την τιμή του π (3.1415..)



επιστρέφει την τιμή της σταθεράς Planck (J / Hz)



επιστρέφει την τιμή της μοριακής σταθεράς αερίου (J / mol K)



επιστρέφει την τιμή της ταχύτητας του φωτός στο κενό (m / sec)



Η σταθερά διαχείρισης συμβόλου εισάγει ένα αριθμημένο σύμβολο στον κώδικα και χρησιμοποιείται για σύγκριση των μεταβλητών με χρήση συμβόλου στο **front panel**.



Η σταθερά χρώματος θέτει το είδος του χρώματος σε παραμέτρους λειτουργιών Vis διαχείρισης εικόνας.

**Error**

Η σταθερά διαχείρισης σφαλμάτων χρησιμοποιείται για την σύγκριση και αποφυγή τερματισμού της λειτουργίας ενός VI από εμφάνιση σφάλματος χαμηλού επιπέδου.

## 4.2.6. Λειτουργίες Σύγκρισης

Υπάρχουν δύο τρόποι λειτουργίας σύγκρισης πινάκων: Μέθοδος συνόλου και μέθοδος στοιχείων. Στη μέθοδο συνόλου όταν συγκρίνουμε δύο πίνακες χαρακτήρων ή δυο συστάδες, η λειτουργία επιστρέφει μία τιμή. Με τη μέθοδο που συγκρίνει τα στοιχεία χωριστά, επιστρέφει τον πίνακα ή τη συστάδα με το κάθε λογικό αποτέλεσμα σύγκρισης ξεχωριστά για κάθε στοιχείο τιμών.

Οι πίνακες πρέπει να αποτελούνται από τον ίδιο αριθμό στοιχείων και τον ίδιο τύπο.

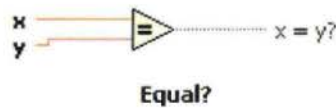
Επίσης θα πρέπει να έχουν τον ίδιο αριθμό διαστάσεων και κάθε διάσταση να έχει το ίδιο μέγεθος.

Μπορούμε να επιλέξουμε τη μέθοδο σύγκρισης με δεξί πλήκτρο από το ποντίκι επάνω στο εικονίδιο του συγκριτή.

### Περιγραφή Εντολών Σύγκρισης.

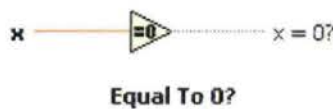
#### ➤ Ίσος

Εάν ο  $x$  είναι ίσος με τον  $y$  επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



#### ➤ Ίσος με 0

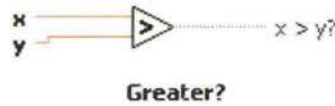
Εάν ο  $x$  είναι ίσος με το 0 επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



#### ➤ Μεγαλύτερος

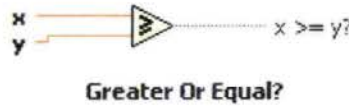
Εάν ο  $x$  είναι μεγαλύτερος από το  $y$  επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.





➤ **Μεγαλύτερος ή Ίσος**

Εάν ο x είναι μεγαλύτερος ή ίσος με τον y επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



➤ **Μεγαλύτερος ή Ίσος με 0**

Εάν ο x είναι μεγαλύτερος ή ίσος με το 0 επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



➤ **Μεγαλύτερος από 0**

Εάν ο x είναι μεγαλύτερος από το 0 επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



➤ **Δεκαεξαδικό Ψηφίο**

Εάν ο χαρακτήρας είναι ψηφίο δεκαεξαδικού συστήματος επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.

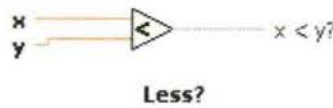


➤ **Διάστημα**

Εάν ο x είναι μεγαλύτερος ή ίσος με την τιμή του lo και μικρότερος από την τιμή hi επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.

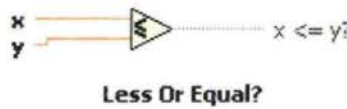
➤ **Μικρότερος**

Εάν ο x είναι μικρότερος από το y επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



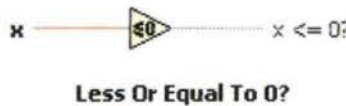
➤ **Μικρότερος ή Ίσος**

Εάν ο x είναι μικρότερος ή ίσος με το y επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



➤ **Μικρότερος ή Ίσος με 0**

Εάν ο x είναι μικρότερος ή ίσος με το 0 επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



➤ **Μικρότερος του 0**

Εάν ο x είναι μικρότερος από το 0 επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



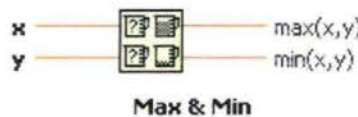
➤ **Κατηγοριοποίηση Χαρακτήρων**

Επιστρέφει τον αριθμό κατηγορίας (class number) στον οποίο ανήκει ο χαρακτήρας.



➤ **Ανώτατος και Ελάχιστος**

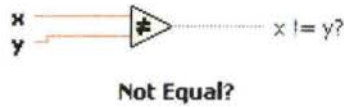
Συγκρίνει της εισόδους x και y και επιστρέφει τη μεγαλύτερη και τη μικρότερη τιμή στην έξοδο.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

### ➤ Όχι Ίσος

Εάν ο  $x$  δεν είναι ίσος με τον  $y$  επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



### ➤ Όχι Ίσος με 0

Εάν ο  $x$  δεν είναι ίσος με το 0 επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



### ➤ Οκταδικό ψηφίο

Εάν ο χαρακτήρας είναι ένα οκταδικό ψηφίο επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



### ➤ Δεκαδικό ψηφίο

Εάν ο χαρακτήρας είναι ένα δεκαδικό ψηφίο επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



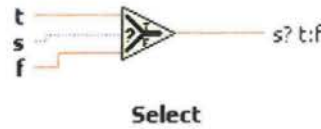
### ➤ Εκτυπώσιμος

Εάν ο χαρακτήρας είναι εκτυπώσιμος χαρακτήρας ASCII επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



### ➤ Επιλογή

Εάν το  $s$  είναι αληθές, επιστρέφει την τιμή που συνδέεται με την είσοδο  $t$ . Εάν το  $s$  είναι λογικό ψευδές, επιστρέφεται η τιμή που συνδέεται με την είσοδο  $f$ .



➤ **Λευκός Χαρακτήρας**

Εάν ο χαρακτήρας είναι διάστημα, ετικέτα, αλλαγή γραμμής κ.α. επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει λογικό ψευδές.



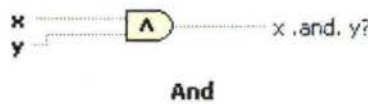
**4.2.7. Λογικές Λειτουργίες**

Οι λογικές λειτουργίες αποκτούν είτε λογικά είτε αριθμητικά στοιχεία εισόδου.

**Περιγραφή Λογικών Εντολών**

➤ **AND**

Υπολογίζει το λογικό ΚΑΙ των εισόδων x,y.



➤ **AND σε Στοιχεία Πίνακα**

Εάν όλα τα στοιχεία του πίνακα είναι αληθή, επιστρέφει λογικό αληθές. Διαφορετικά επιστρέφει ψευδές.



➤ **Λογικός Πίνακας σε Αριθμό**

Μετατρέπει έναν πίνακα λογικών στοιχείων σε ακέραιο αριθμό ως προς το συμπλήρωμα του προς δύο, με το μηδενικό στοιχείο του πίνακα να είναι το λιγότερο σημαντικό δυαδικό ψηφίο.

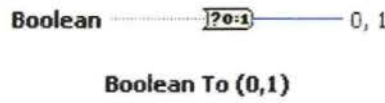




# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

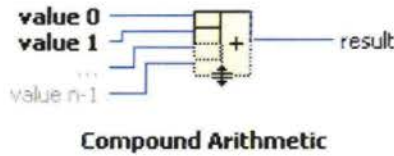
## ➤ Λογικό σε Δεκαδικό (0,1)

Μετατρέπει μια λογική τιμή σε ακέραιο αριθμό 0 και 1 για τις τιμές εισόδου αλήθειας και ψεύδους αντίστοιχα.



## ➤ Λογικές Πράξεις

Εκτελεί αριθμητικές και λογικές πράξεις σε δύο ή περισσότερες μεταβλητές.



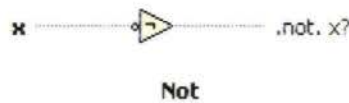
## ➤ Αποκλειστικό OR

Υπολογίζει το αποκλειστικό OR των εισόδων x, y.



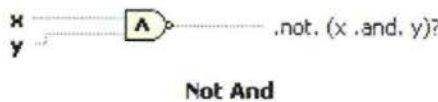
## ➤ NOT

Υπολογίζει τη λογική άρνηση της εισόδου x.



## ➤ NOT AND

Υπολογίζει το λογικό NAND των εισόδων x, y.



## ➤ XNOR

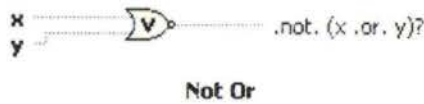
Υπολογίζει τη λογική άρνηση του αποκλειστικού OR των εισόδων x, y.



## ➤ NOT OR

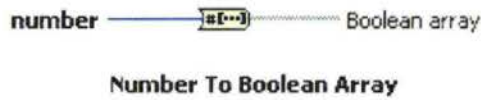
Υπολογίζει το λογικό NOR των εισόδων x, y.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



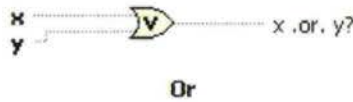
### ➤ Αριθμός σε Λογική τιμή

Μετατρέπει έναν δεκαδικό αριθμό σε παράσταση 8 έως 16, ή 32 στοιχείων του δυαδικού συστήματος.



### ➤ OR

Υπολογίζει το λογικό OR των εισόδων x, y.




### ➤ OR σε Στοιχεία Πίνακα

Επιστρέφει λογικό ψευδές εάν όλα τα στοιχεία στον πίνακα είναι ψευδή. Διαφορετικά επιστρέφει λογικό αληθές.



### ➤ Λογική Σταθερά

Θέτουμε μία τιμή λογικής αλήθειας ή ψεύδους επιλέγοντας T ή F της σταθεράς με τη χρήση του εργαλείου δείκτη. 

## 4.2.8. Λειτουργίες Χρόνου, Διαλόγου και Λάθους.

Με τις λειτουργίες χρόνου, διαλόγου και λάθους έχουμε πρόσβαση σε λειτουργίες συγχρονισμού, σε εφαρμογές που απαιτούν να γνωρίζουμε τον τρέχοντα χρόνο της, την παρεχόμενη περίοδο ενός γεγονότος ή το χρόνο αναστολής μιας λειτουργίας για μια συγκεκριμένη χρονική περίοδο.

### Περιγραφή Λειτουργιών Χρόνου, Διαλόγου και Λάθους.

#### ➤ Ημερομηνία / Χρόνος σε Δευτερόλεπτα

Μετατρέπει μια συστάδα πληροφοριών του χρόνου (ώρα, ημέρα, μήνας, έτος) σε 32-bit ακέραιους αριθμούς μορφής δευτερολέπτων.



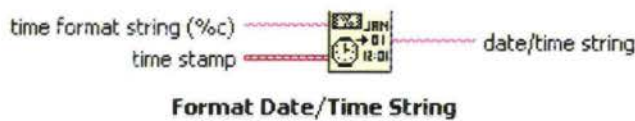
➤ **Δευτερόλεπτα σε Ημερομηνία / Χρόνο**

Μετατρέπει έναν 32-bit ακέραιο αριθμό μορφής δευτερολέπτων σε μια συστάδα πληροφοριών του χρόνου (ώρα, ημέρα, μήνας, έτος).



➤ **Διαμόρφωση μορφής εμφάνισης Ημερομηνίας / Χρόνου**

Δίνει την πληροφορία της Ημερομηνίας / Χρόνου σε μορφή παρουσίασης όπως την διευκρινίζουμε με την επιλογή διαμόρφωσης μορφής εμφάνισης (**time format string**).



➤ **Πλαίσιο Διαλόγου**

Εμφανίζει ένα πλαίσιο διαλόγου που περιέχει ένα μήνυμα και ένα πλήκτρο επιλογής **OK** για τον τερματισμό του. Η εισαγωγή του μηνύματος γίνεται από την επιλογή **message**, και η ονοματοθέτηση του πλήκτρου μπορεί να τροποποιηθεί στην επιλογή **button name ("OK")**.



➤ **Πλαίσιο διαλόγου δύο κουμπιών**

Εμφανίζει ένα πλαίσιο διαλόγου που περιέχει ένα μήνυμα και δύο πλήκτρα επιλογής για το τερματισμό του.



➤ **Μέτρηση (ms)**

Η λειτουργία χρησιμοποιείται για τον υπολογισμό του χρόνου ενός γεγονότος σε χιλιοστά του δευτερολέπτου.

# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



## ➤ Περιμένετε (ms)

Η λειτουργία περιμένει τον διευκρινισμένο χρόνο σε χιλιοστά του δευτερολέπτου που έχουμε εισάγει στην επιλογή **milliseconds to Wait** και κατόπιν αναγκάζει το VI μέσα στο οποίο βρίσκεται να εκτελέσει τις λειτουργίες του. Χρησιμοποιείται ως χρονικό καθυστέρησης μέσα στον κώδικα.



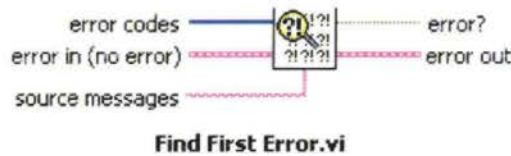
## ➤ Περιμένετε έως..

Η λειτουργία τερματίζει τη λειτουργία ενός βρόχου μετά από συγκεκριμένο χρόνο που έχουμε ορίσει στην επιλογή **millisecond multiple**, ώστε να συγχρονιστεί η έξοδος του με αποτελέσματα άλλων λειτουργιών ή δομών.



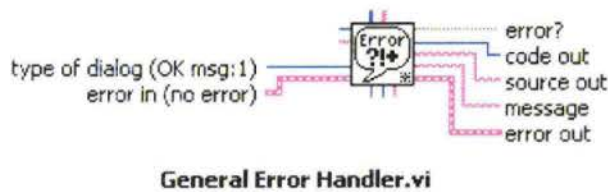
## ➤ Εύρεση Πρώτου Λάθους

Εξετάζει τη θέση λάθους ενός ή περισσότερων χαμηλού επιπέδου λειτουργιών ή **subVIs** που παράγουν έναν αριθμητικό κώδικα λάθους.



## ➤ Γενικός Χειριστής Λάθους

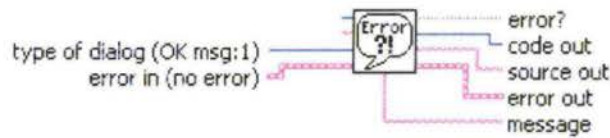
Η λειτουργία αυτή δημιουργεί μέσα από τις προεπιλογές της τον τρόπο εμφάνισης και περιγραφής του λάθους σε ένα πλαίσιο διαλόγου.



## ➤ Απλός Χειριστής Λάθους

Η λειτουργία αυτή εμφανίζει την περιγραφή του λάθους σε ένα πλαίσιο διαλόγου παρέχοντας λιγότερες δυνατότητες επιλογής της μορφής εμφάνισης στον χρήστη.





Simple Error Handler.vi

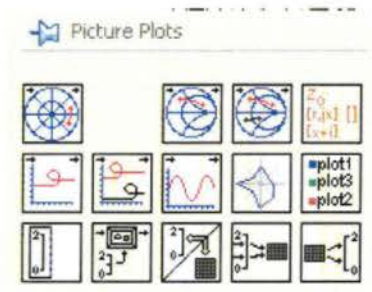
#### 4.2.9. Λειτουργίες Γραφημάτων και Ήχου

##### Περιγραφή Υποπαλετών Γραφημάτων και Ήχου

- Η Υποπαλέτα **3D Graphs** περιέχει εργαλεία επεξεργασίας 3D διαγραμμάτων. Παραμετροποιεί τα διαγράμματα στη μορφή που θέλουμε.

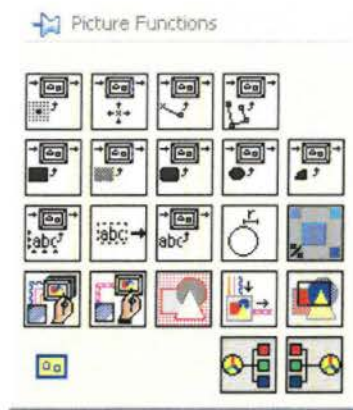


- Η Υποπαλέτα **Picture Plots** περιέχει εργαλεία για κατασκευή 2D διαγραμμάτων, πολικών διαγραμμάτων και χάρτη Smith.



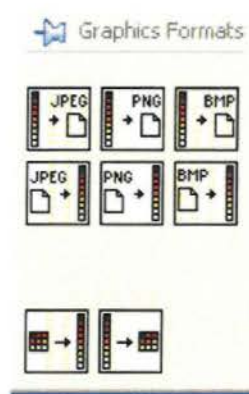
- Η Υποπαλέτα **Picture Function** περιέχει εργαλεία διαμόρφωσης και επεξεργασίας εικόνων.

**ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.**



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

- Η **Υποπαλέτα Graphics Format** διαμορφώνει την εικόνα στον τύπο που επιθυμούμε. Οι λειτουργίες της έχουν τη δυνατότητα να διαβάσουν από ένα αρχείο μια νέα εικόνα και να την αποθηκεύσουν σε αυτό.

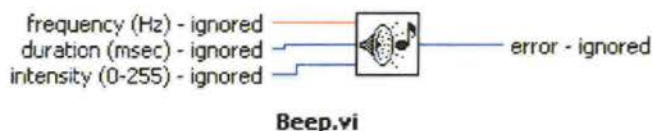


- Η **Υποπαλέτα Sounds** περιέχει λειτουργίες παραγωγής και επεξεργασίας ήχου. Υπάρχει δυνατότητα καταγραφής εξωτερικού ήχου από συσκευή σε αρχείο παραγωγής ήχου και αναπαραγωγής από αρχείο.



### ➤ **Ήχος Τόνου**

Έχουμε τη δυνατότητα να ρυθμίσουμε τη συχνότητα, τη διάρκεια και την ένταση του τόνου ώστε να τον χρησιμοποιήσουμε ως ηχητική ένδειξη σε μία συνθήκη.



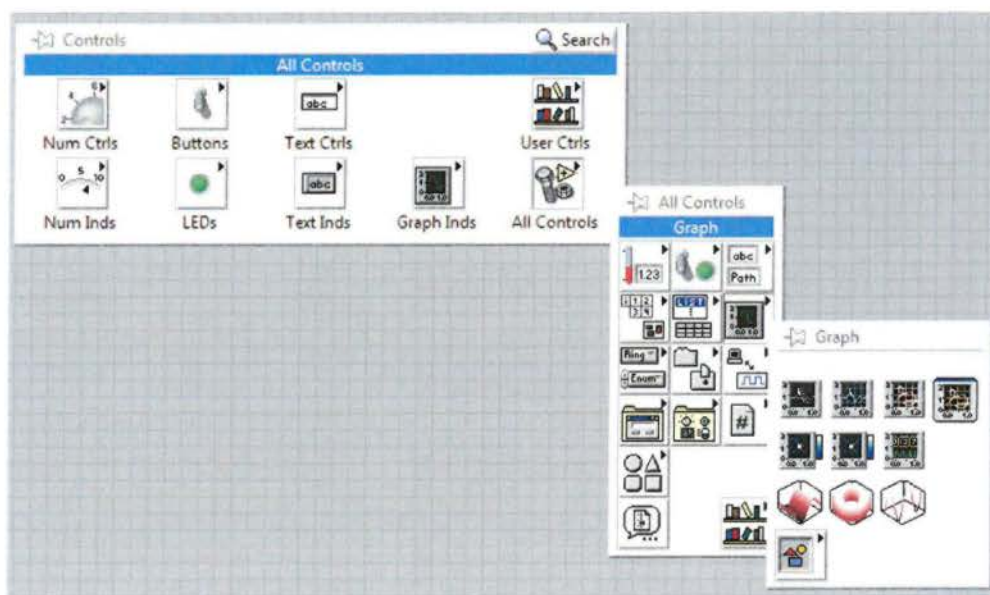
## 4.2.10 Διαγράμματα & Γραφήματα

Γράφημα ονομάζεται η δι-διάστατη απεικόνιση ενός ή περισσότερων κυματομορφών. Στο Labview μπορούμε να βρούμε δύο είδη γραφημάτων:

- γραφήματα XY τα οποία απεικονίζουν ένα διάνυσμα X ως προς ένα διάνυσμα Y και
- γραφήματα κυματομορφών όπου θεωρείται ότι ο άξονας X είναι ο άξονας του χρόνου.

Γίνεται χρήση των γραφημάτων (**graphs**) και των διανυσμάτων (**charts**) για να απεικονίσουμε τιμές σε μορφή γραφικών παραστάσεων ή γραφήματος, όπως στον παλμογράφο. Η βασική διαφορά τους βρίσκεται στο γεγονός ότι τα διαγράμματα απεικονίζουν συνεχώς τα δεδομένα που είναι διαθέσιμα, ενώ τα γραφήματα απεικονίζουν τα στοιχεία έτοιμων πινάκων στην μορφή αξόνων X-Y.

Με την τοποθέτηση τους στο **front panel**, οι άξονες είναι βαθμολογημένοι με προεπιλεγμένη τιμή, η οποία όμως μπορεί να διαφοροποιηθεί με το εργαλείο ονοματοθέτησης. Επιπλέον μπορούμε να επιλέξουμε αυτόματη βαθμολόγηση για τον άξονα X με το **X Scale >> Autoscale X** από το **Pop-Up** μενού και με το **Y Scale >> Autoscale Y** για τον άξονα Y. Η παλέτα γραφημάτων (εικ. 4.6) διαθέτει και τα 3D γραφήματα δίνοντας δυνατότητα τρισδιάστατης απεικόνισης του γραφήματος.

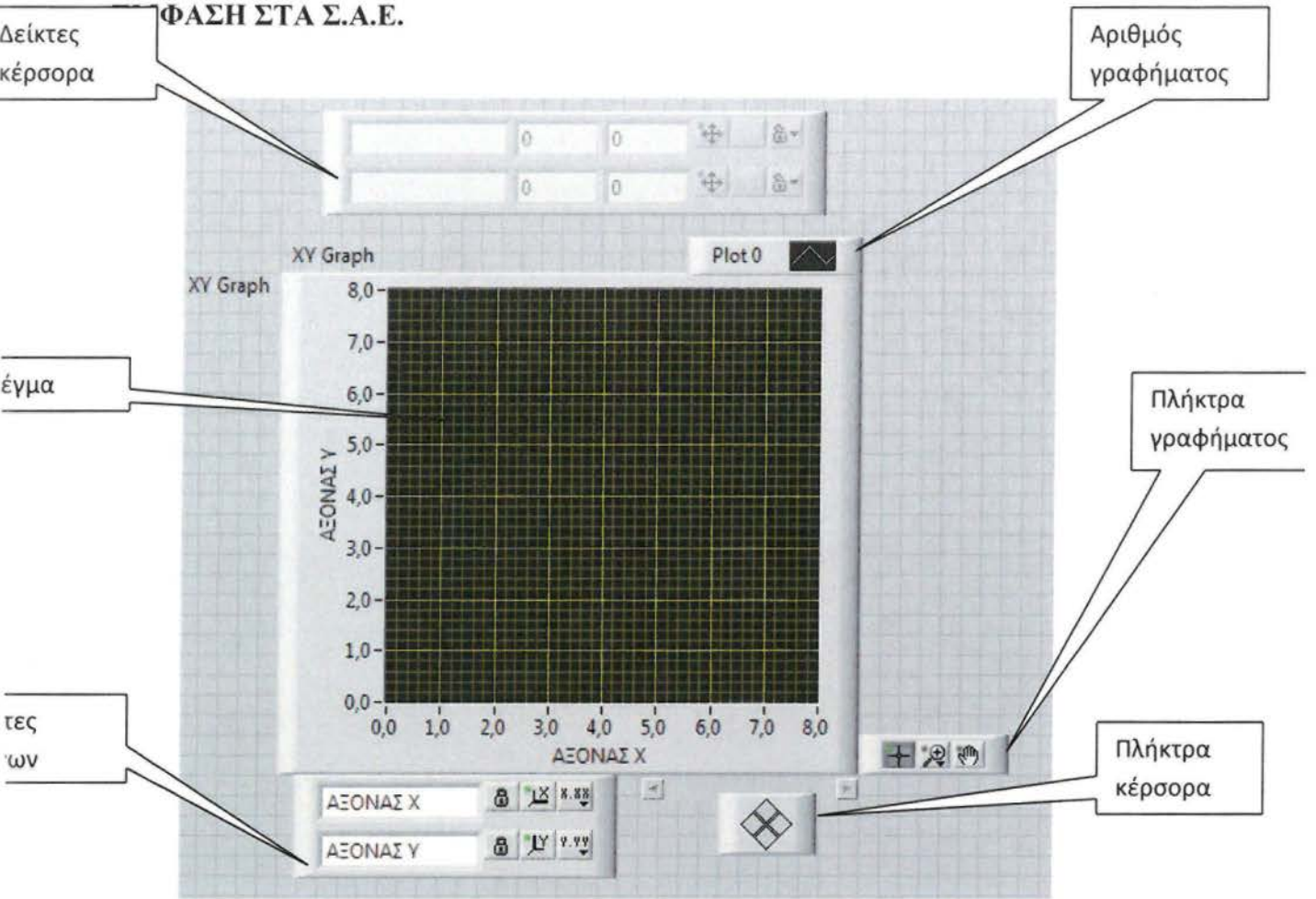


Εικόνα 4.6: Απεικόνιση 3D γραφημάτων

Μας δίνεται επίσης η δυνατότητα να τροποποιήσουμε τη συμπεριφορά των δρομέων των γραφικών παραστάσεων και την μορφή των αξόνων. Η εικόνα που ακολουθεί απεικονίζει τα στοιχεία μιας γραφικής παλέτας.(εικ 4.7)



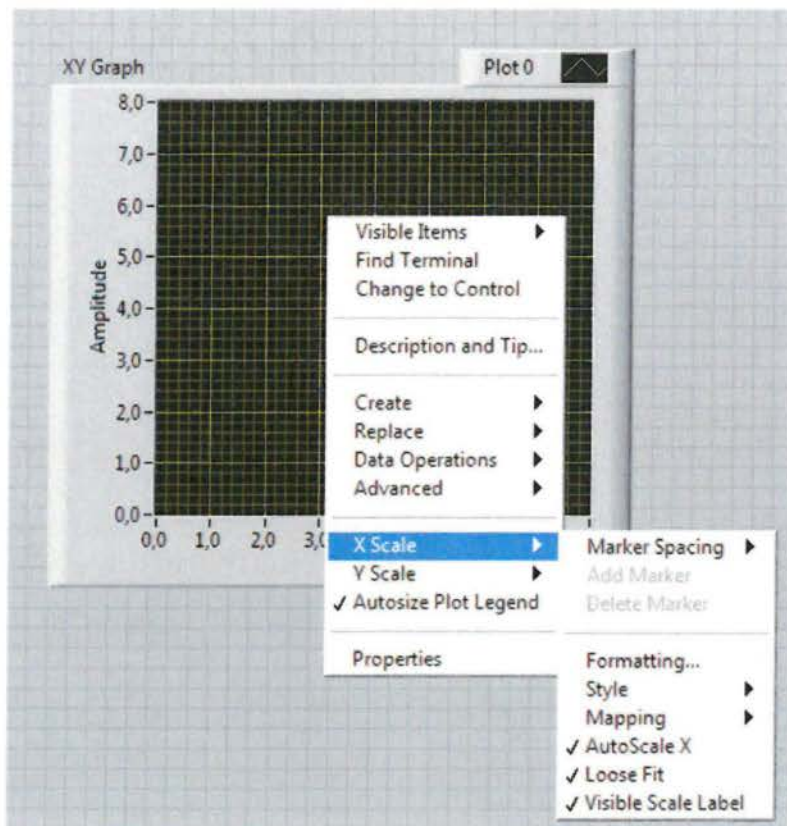
ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ  
ΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 4.7: Αντικείμενα παλέτας γραφήματος

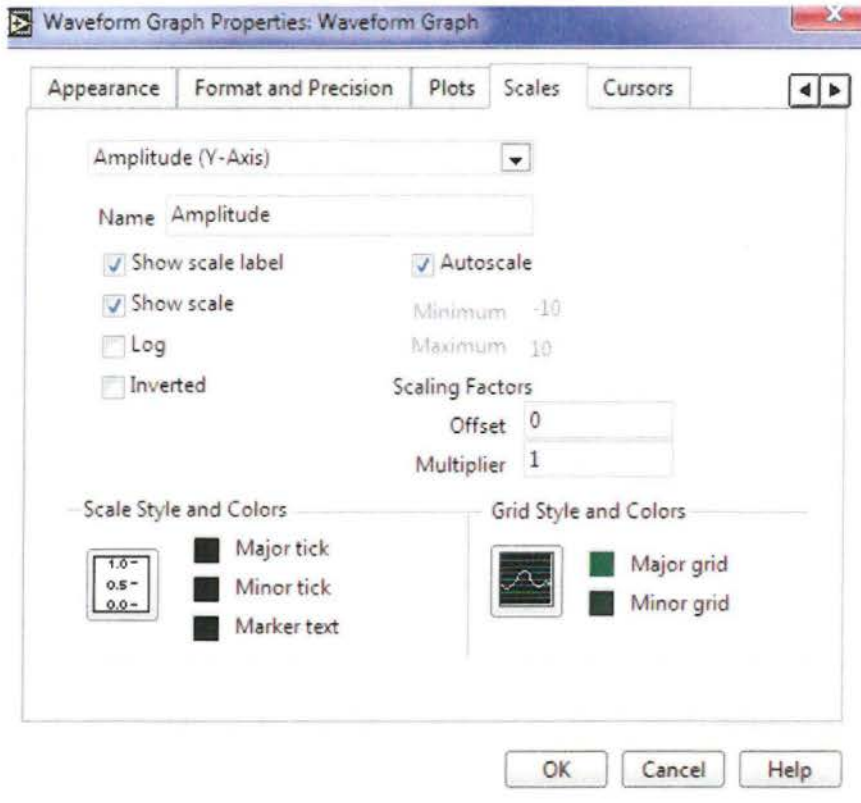
### Επειξεργασία Αξόνων

Πατάμε το δεξί πλήκτρο από το ποντίκι πάνω στο γράφημα (εικ 4.8) και διαλέγουμε τον άξονα (X Scale/ Y Scale ) που θέλουμε να μορφοποιήσουμε.



Εικόνα 4.8: Μορφοποίηση αξόνων

Από το παράθυρο επιλογών (εικ. 4.9) που εμφανίζεται μπορούμε να ορίσουμε πως να απεικονίζεται ο άξονας και να ορίσουμε τις παραμέτρους ανάλυσης και απεικόνισης της κυματομορφής μας ( $X_0$  και  $dX$ ).



Εικόνα 4.9: Παράθυρο τροποποίησης αξόνων

## 4.2.11 Γραφήματα Charts

Στο **Labview** μπορούμε να βρούμε τις τρεις ακόλουθες μορφές απεικόνισης γραφημάτων **charts**:

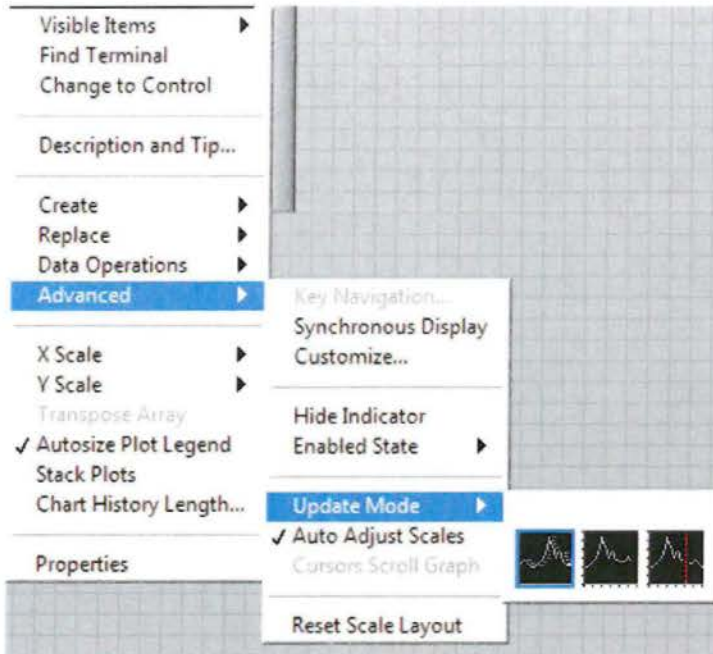
- Ταινίας (Strip)
- Παλμογράφου (Scope)
- Σάρωσης (Sweep)

Στα διαγράμματα ταινίας (**Strip**) η κυματομορφή φαίνεται ότι κυλάει προς τα δεξιά απεικονίζοντας τις νέες τιμές στο δεξί άκρο.

Στον παλμογράφο (**Scope**) οι τιμές του διαγράμματος ξεκινούν από αριστερά και μόλις φτάσουν στο δεξί άκρο ξεκινάει καινούργιο διάγραμμα από τα αριστερά.

Τα διαγράμματα σάρωσης (**Sweep**) είναι παρόμοια με τα διαγράμματα ταινίας, με την διαφορά όμως ότι εδώ μπορούμε να ορίσουμε από ποιο σημείο του διαγράμματος θα ξεκινήσει η απεικόνιση των νέων τιμών. Μπορούμε να ορίσουμε το είδος του διαγράμματος με την εντολή από το **Pop-Up** μενού **Advanced >> Update mode** (εικ. 4.10)

Με το γράφημα **Chart** μπορούμε να απεικονίσουμε πολλά διαγράμματα ταυτόχρονα. Αυτό επιτυγχάνεται, με την λειτουργία **Bundle**.



Εικόνα 4.10: Επιλογή μορφής απεικόνισης

## 4.2.12 Γραφήματα Graphs

Το σύνολο των στοιχείων ενός πίνακα απεικονίζονται με τα γραφήματα **Graphs**. Το **Labview** έχει τις μορφές ακόλουθες μορφές γραφημάτων:

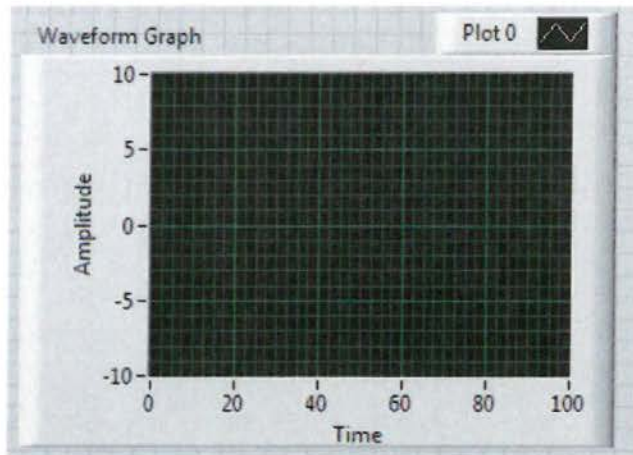
- Waveform Graphs
- XY Graphs

Η απεικόνιση τους στο **front panel** είναι ίδια, λειτουργικά όμως διαφέρουν σημαντικά. Η διαφορά τους βρίσκεται στο διαφορετικό τύπο δεδομένων που δέχονται στην είσοδο τους.

Τα **Waveform Graphs** (εικ. 4.11) είναι κατάλληλα να σχεδιάζουν γραφήματα κυματομορφών από πίνακες με ομοιόμορφα σημεία. Στον άξονα X βρίσκονται τα στοιχεία ενώ στον άξονα Y η τιμή του στοιχείου. Σε δι-διάστατο πίνακα το **Waveform Graph** σχεδιάζει δύο γραφήματα ταυτόχρονα. Στην είσοδο τους υπάρχει λειτουργία δημιουργίας πίνακα.



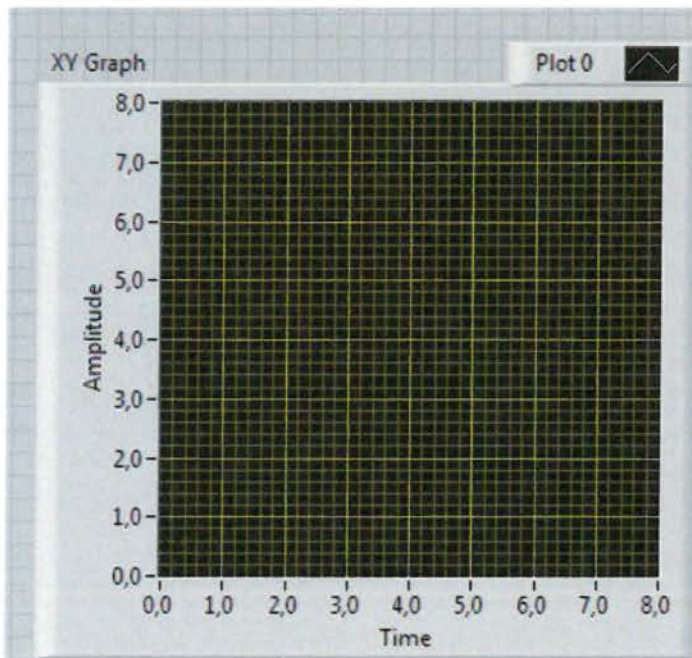
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 4.11: Απεικόνιση Waveform Graphs

Τα **XY Graphs** προορίζονται να σχεδιάζουν καρτεσιανά γραφήματα. Είναι κατάλληλα για δείγματα σε ακαθόριστα διαστήματα ή για σημεία που προέρχονται από διαφορετικές μεταβλητές εξαρτημένες μεταξύ τους (εικ. 4.12). Τα δεδομένα συνήθως τα δίνουμε μέσω λειτουργίας **Bundle**.

Σε γραφήματα **XY Graphs** μας δίνεται η δυνατότητα να αντιστοιχίσουμε κυματομορφές μεταξύ τους.



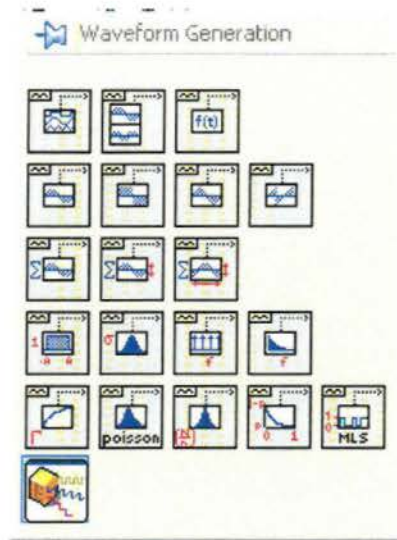
Εικόνα 4.12: Απεικόνιση XY Graphs

### 4.2.13. Λειτουργίες Ανάλυσης

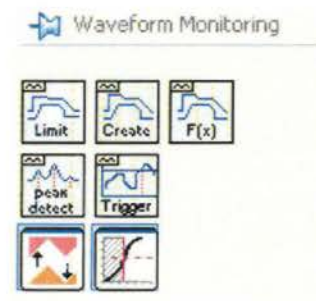
Η παλέτα ανάλυσης περιέχει λειτουργίες για παραγωγή και επεξεργασία συναρτήσεων αναλογικών και ψηφιακών μεγεθών.

#### Περιγραφή Υποπαλετών Ανάλυσης

- Η **Υποπαλέτα Waveform Generator** παρέχει συναρτήσεις ημιτονικών, τετραγωνικών, τριγωνικών και πολυμορφικών αναλογικών σημάτων, καθώς και συναρτήσεις λευκού και γκαουσιανού θορύβου.

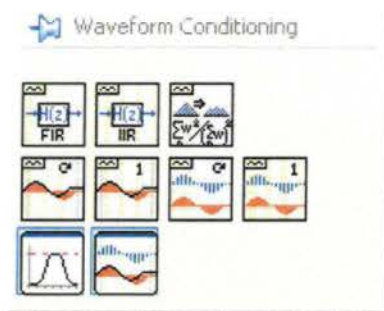


- Η **Υποπαλέτα Waveform Monitoring** παρέχει λειτουργίες σύγκρισης σημάτων. Για παράδειγμα στη λειτουργία ορίου θέτουμε ανώτατο και κατώτατο όριο σήματος. Εάν η πληροφορία του σήματος βρίσκεται μέσα στα όρια, τότε η λειτουργία ορίου δίνει τιμή αληθής.



- Η **Υποπαλέτα Waveform Conditioning** παρέχει συναρτήσεις για λειτουργίες ψηφιακών φίλτρων FIR, IIR.

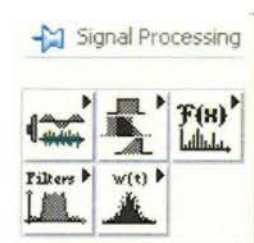
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



- Η **Υποπαλέτα Waveform Measurement** παρέχει λειτουργίες μέτρησης ηλεκτρικών μεγεθών των σημάτων, όπως μετρήσεις ενεργούς τάσης RMS, φάσης, πλάτους συχνότητας κ.ά.



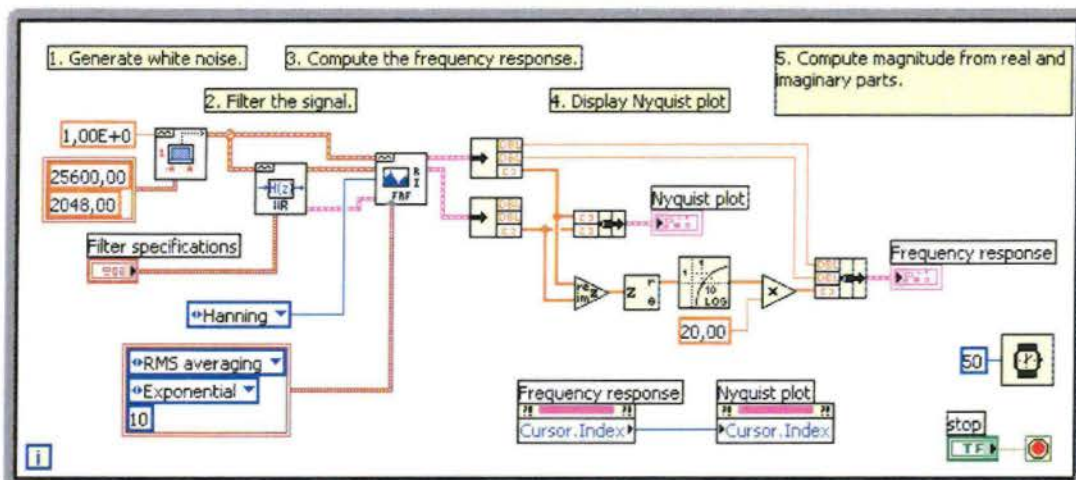
- Η **Υποπαλέτα Signal Processing** παρέχει λειτουργίες γεννητριών ψηφιακού σήματος, θορύβου, καθώς και συναρτήσεις φίλτρων για δημιουργία VIs επεξεργασίας ψηφιακού σήματος.



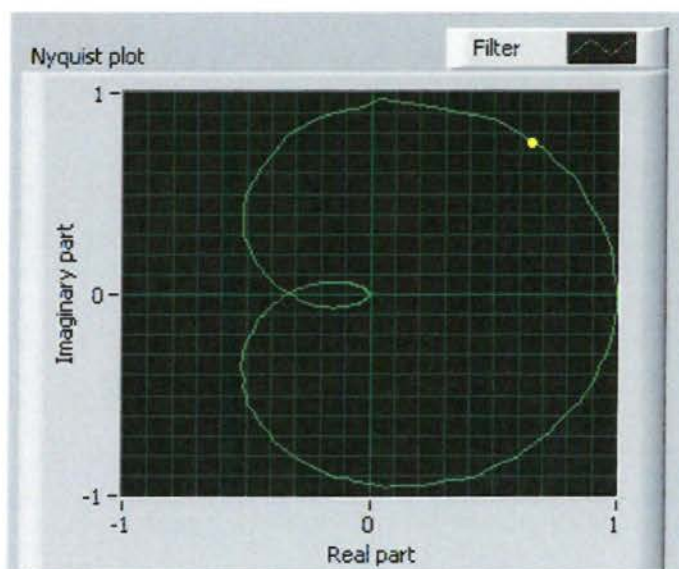
- Η **Υποπαλέτα Mathematics** παρέχει μαθηματικές εξισώσεις για τη μαθηματική ανάλυση των σημάτων, καθώς και μαθηματικές φόρμες για την παραγωγή νέων σημάτων. Το **block diagram** που ακολουθεί (εικ.4.6) αποτελεί ενδεικτικό

# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

παράδειγμα των λειτουργιών που περιλαμβάνονται στην παλέτα ανάλυσης για την παραγωγή γραφήματος Nyquist σε φίλτρο συχνοτήτων.



Εικόνα 4.6: Block diagram που κάνει χρήση της παλέτας ανάλυσης





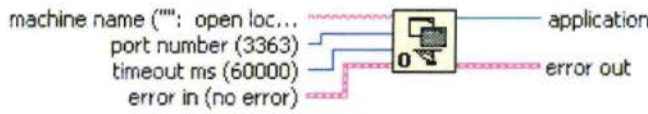
### 4.2.14. Λειτουργίες Ελέγχου Εφαρμογών

Η παλέτα **Ελέγχου Εφαρμογών (Application Control)** παρέχει λειτουργίες που αφορούν τις δυνατότητες κλήσεων των VIs σε ένα δίκτυο και των γενικότερων λειτουργιών εκτέλεσης ενός VI.

#### Περιγραφή Λειτουργιών Ελέγχου Εφαρμογών.

##### ➤ Open

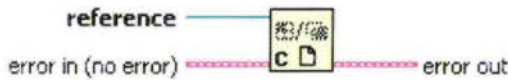
Καλεί VI από H/Y του δικτύου. Η διεύθυνση που δίνεται είναι το TCP/IP του H/Y.



**Open Application Reference**

##### ➤ Close

Κλείνει το VI που έχει κληθεί.



**Close Reference**

##### ➤ Stop

Διακόπτει την εκτέλεση του VI αν στην είσοδο της λειτουργίας εντοπιστεί λογική τιμή True. Εάν δεν εντοπιστεί τιμή, το VI θα τερματιστεί με το τέλος της εκτέλεσης του βρόχου στον οποίο βρίσκεται.



**Stop**

##### ➤ Exit

Σταματάει την εκτέλεση του VI και κλείνει τον συγκεκριμένο τομέα του LabVIEW.



##### ➤ Print from Panel

Εκτυπώνει το from panel του συγκεκριμένου VI.

➤ **Help**

Περιέχει λειτουργίες για τα παράθυρα βοήθειας.



Help

### 4.2.15. Λειτουργίες Αρχείων

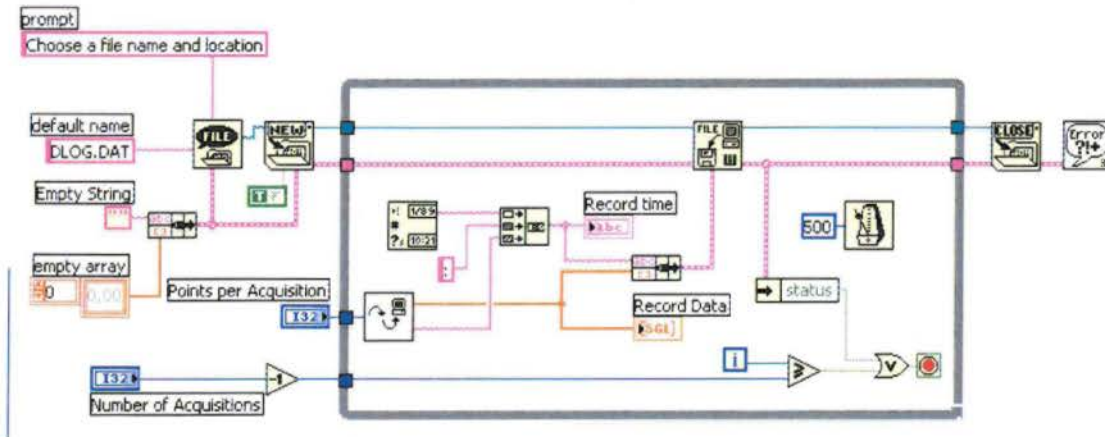
Οι λειτουργίες που είναι διαθέσιμες στην παλέτα λειτουργιών **Αρχείου Εισόδου / Εξόδου (File I/O)** μας παρέχουν δυνατότητα εγγραφής και ανάγνωσης δεδομένων με μορφή χαρακτήρων και λογιστικού φύλλου.

#### Εγγραφή σε Αρχείο

Η εγγραφή σε αρχείο προϋποθέτει τις εξής ενέργειες:

- **Δημιουργία ( Create),**
- **Εγγραφή (Write),**
- **Κλείσιμο αρχείου (Close).**

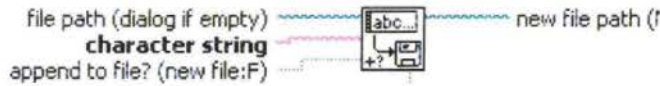
Στο **block diagram** παρουσιάζεται η σειρά της δομής εγγραφής δεδομένων σε κώδικα VI. (εικ. 4.7)



Εικόνα 4.7.: Λειτουργία εγγραφής σε αρχείο

Για ταχύτερο και πιο εύρηστο προγραμματισμό του κώδικα VI μπορούμε να χρησιμοποιήσουμε τη λειτουργία **Write Characters To File** η οποία αποτελεί μορφή **subVI** των προηγούμενων εντολών εγγραφής αρχείου.

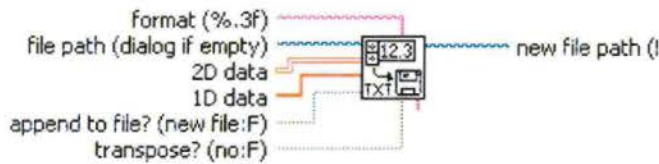
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



**Write Characters To File.vi**

file path: επιλογή διαδρομής και ονοματοθέτηση αρχείου  
character string: εισαγωγή δεδομένων  
append to file?: επιλογή πρόσθεσης δεδομένων στο ίδιο αρχείο ή αντικατάσταση του αρχείου με νέο σε κάθε εισαγωγή δεδομένου στη λειτουργία.

Για εγγραφή δεδομένων σε αρχείο λογιστικού φύλλου (**Spreadsheet file**), η παλέτα αρχείων παρέχει τη λειτουργία **Write To Spreadsheet File**.



**Write To Spreadsheet File.vi**

Η λειτουργία **Write To Spreadsheet File** αποτελεί επέκταση δομής της **Write Characters To File** ενσωματώνοντας πρόσθετες λειτουργίες για τη μετατροπή αριθμητικών δεδομένων σε χαρακτήρες πριν από την αποθήκευσή τους.

Σημείωση: τα δεδομένα στη λειτουργία **Write To Spreadsheet File** πρέπει να εισάγονται πάντοτε με μορφή πίνακα.

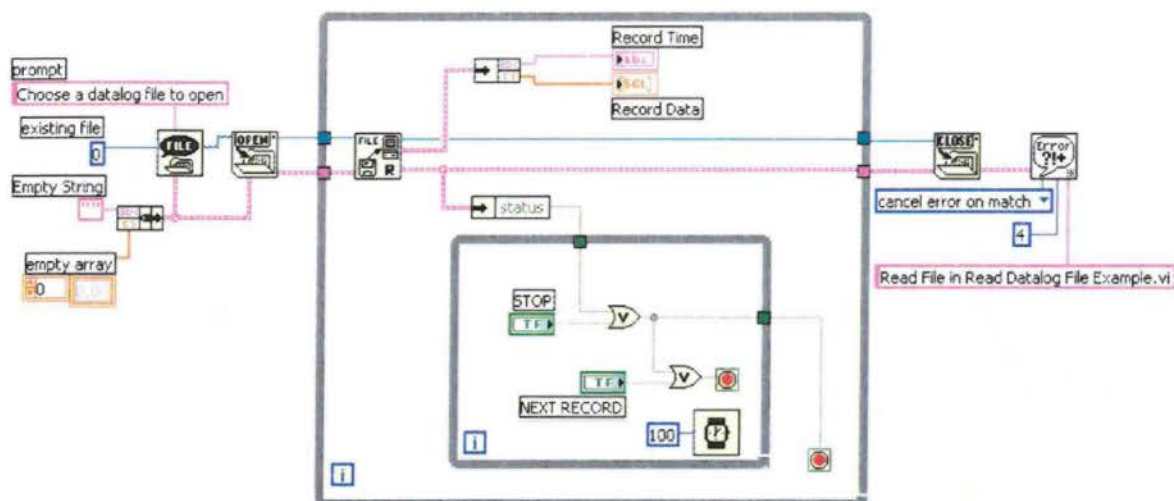
### Ανάγνωση από Αρχείο

Η ανάγνωση από αρχείο προϋποθέτει τις εξής ενέργειες:

- **Ανοιγμα (Open),**
- **Εγγραφή (Write),**
- **Κλείσιμο (Close).**

Στο **block diagram** παρουσιάζεται η σειρά της δομής ανάγνωσης δεδομένων σε κώδικα VI.(εικ. 4.8)

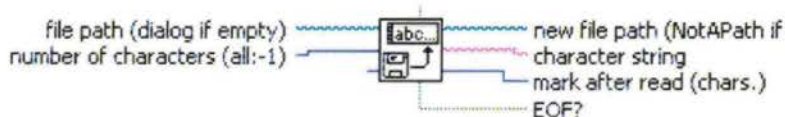
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Εικόνα 4.8.: Λειτουργία ανάγνωσης από αρχείο

Σημείωση: οι λειτουργίες **Open** και **Close** πρέπει πάντοτε να βρίσκονται έξω από τη δομή While Loop, για να μην ανοίγει το ίδιο αρχείο σε κάθε κύκλο της.

Για ταχύτερο και πιο εύρηστο προγραμματισμό του κώδικα VI μπορούμε να χρησιμοποιήσουμε τη λειτουργία **Read Characters From File**, η οποία αποτελεί μορφή **subVI** των προηγούμενων εντολών εγγραφής αρχείου.



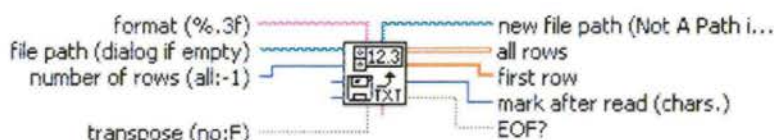
**Read Characters From File.vi**

character string: εξαγωγή των χαρακτήρων δεδομένων

file path: επιλογή διαδρομής και ονοματοθέτησης αρχείου

number of characters: επιλογή αριθμού ανάγνωσης συνολικών αριθμών των δεδομένων

Για ανάγνωση δεδομένων από αρχείο **λογιστικού φύλλου (Spreadsheet file)** η παλέτα αρχείων παρέχει τη λειτουργία **Read from Spreadsheet File**.



**Read From Spreadsheet File.vi**

Η λειτουργία **Read from Spreadsheet File** αποτελεί επέκταση δομής της **Read Characters From File** ενσωματώνοντας λειτουργίες για τη μετατροπή των χαρακτήρων σε αριθμητικά δεδομένα και την ανάγνωση συγκεκριμένων αριθμών γραμμών δεδομένων. Η λειτουργία **Read Lines From File** πραγματοποιεί την ανάγνωση ανά γραμμή περιλαμβάνοντας τη δομή της **Read Characters From File**.



## 5. Παρουσίαση Express VIs

### Εικονικά Όργανα (Virtual Instrument) ή VIs

Εικονικά Όργανα (**Virtual Instrument** ή **VIs**) καλούνται τα προγράμματα του **LabVIEW** διότι μιμούνται την εμφάνιση και την λειτουργία φυσικών οργάνων, όπως είναι η παλμογράφος και τα πολύμετρα. Στο **LabVIEW** δημιουργούμε VIs με στοιχεία ελέγχου (**control**) και στοιχεία ένδειξης (**indicator**). Για να ολοκληρώσουμε τον κώδικα χρησιμοποιούμε **VIs** και δομές για να ελέγχουμε τα αντικείμενα του **front panel**. Το **block diagram** περιέχει αυτόν τον κώδικα. Το **LabVIEW** περιλαμβάνει επίσης εργαλεία για απόκτηση, ανάλυση, προβολή και αποθήκευση δεδομένων, καθώς επίσης και εργαλεία ανίχνευσης λάθους στον κώδικα.

Σημείωση: Έχουμε ήδη αναφέρει ότι τα στοιχεία ελέγχου χρησιμοποιούνται για είσοδο δεδομένων στο πρόγραμμα μας ενώ τα στοιχεία ένδειξης χρησιμοποιούνται για έξοδο των αποτελεσμάτων.

Στο κεφάλαιο αυτό θα δούμε κάποιες ασκήσεις για να εξοικειωθούμε με το γραφικό περιβάλλον προγραμματισμού του **LabVIEW** και τα βασικά χαρακτηριστικά του, καθώς και να διδαχθούμε πώς να αναπτύξουμε βασικές εφαρμογές σε **LabVIEW**.

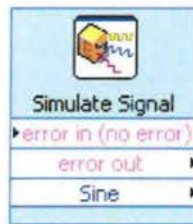
Ας ξεκινήσουμε με την πρώτη άσκηση. Θα χτίσουμε ένα **VI (Virtual Instrument)** που παράγει ένα σήμα και θα παρουσιάζει αυτό το σήμα σε μία γραφική παράσταση.

## Γεννήτρια Προσομοίωσης Σημάτων

1<sup>ο</sup> παράδειγμα:

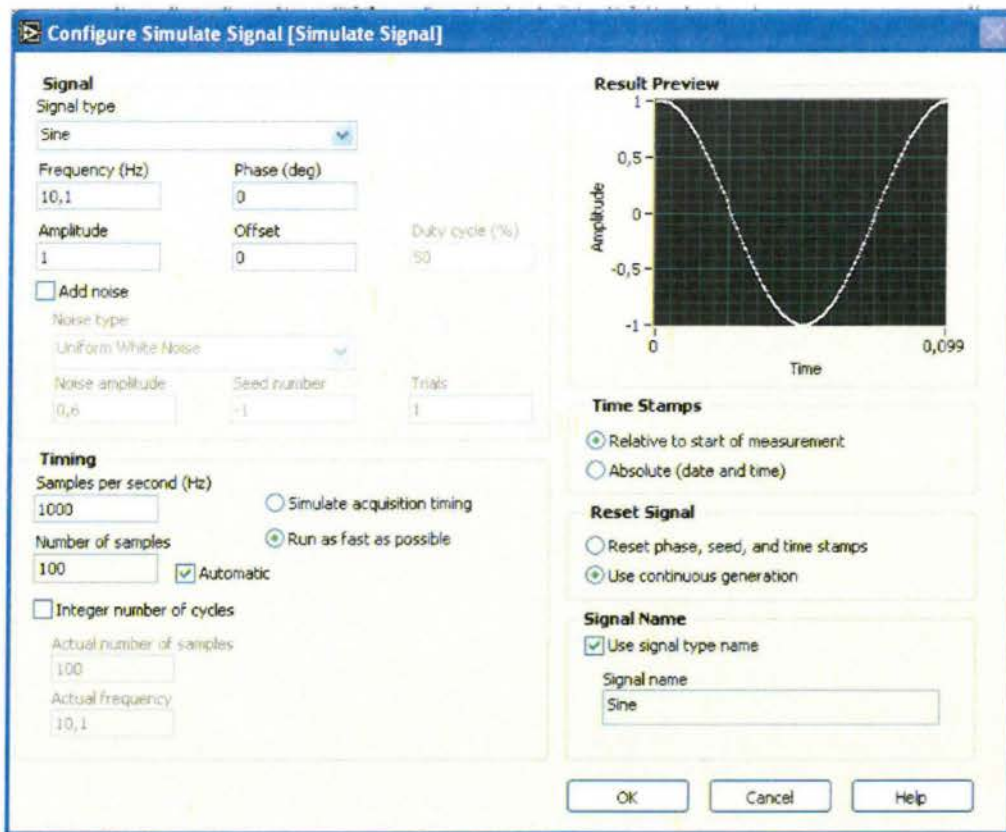
### Βήμα 1:

Επιλέγουμε από την παλέτα εισόδου (input) το **Express VI Simulate Signal**. Το **Express VI Simulate Signal** είναι μια γεννήτρια προσομοίωσης παραγωγής σήματος.



Με την τοποθέτηση του **Express VI** στο **block diagram** αυτόματα ανοίγει το παράθυρο ιδιοτήτων **Configure Simulate Signal**. Ένας άλλος τρόπος για να εμφανίσουμε το παράθυρο ιδιοτήτων είναι με δεξί πλήκτρο του ποντικιού επάνω στο εικονίδιο.


## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



### Βήμα 2:

Αφού έχουμε ρυθμίσει κατάλληλα τις παραμέτρους και τη μορφή του σήματος (**Signal type**), ενεργοποιούμε το πλήκτρο OK και κλείνουμε το παράθυρο ιδιοτήτων.

### Βήμα 3:

Εισάγουμε ένα στοιχείο **control (Knob)** στο **front panel** και με το εργαλείο ονοματοθέτησης  το ονομάζουμε **Amlitude**. Θα το χρησιμοποιήσουμε για τον έλεγχο του πλάτους της κυματομορφής.

### Βήμα 4:

Χρησιμοποιώντας το εργαλείο σύνδεσης  για να καλωδιώσουμε το στοιχείο **control** στην είσοδο **Amlitude** του **Express VI**.

### Βήμα 5:

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Με δεξί πλήκτρο του ποντικιού στο εικονίδιο του **Simulate Signal** από το μενού ανάπτυξης της παραμέτρου Έξοδος Σήματος επιλέγουμε **create >> graph indicator** για δούμε την έξοδο της κυματομορφής.

### Βήμα 6:

Τοποθετούμε άλλο ένα στοιχείο **control** για τον έλεγχο της συχνότητας του παραγόμενου σήματος εξόδου.

### Βήμα 7:

Εκκινούμε τη λειτουργία του VI και παρατηρούμε την αυξομείωση του πλάτους και τις συχνότητας του σήματος, κάνοντας χρήση των ποτενσιόμετρων.

### Βήμα 8:

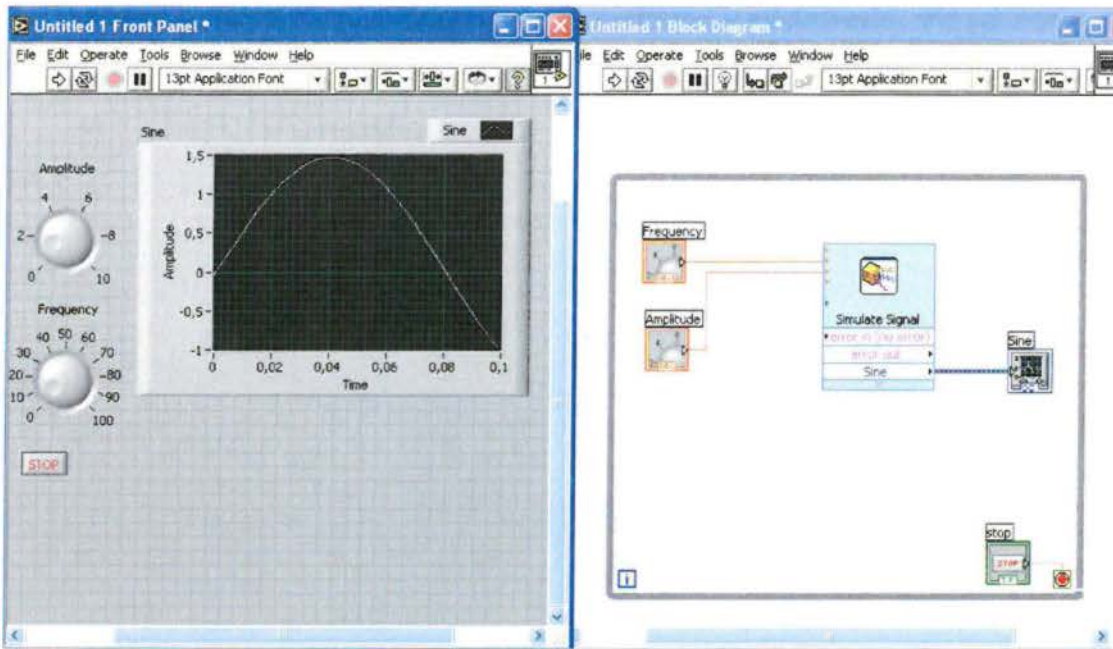
Μπορούμε να αλλάξουμε την κλίμακα για το ποτενσιόμετρο κάνοντας δεξί κλικ πάνω στο στοιχείο και από την επιλογή ιδιότητες μας εμφανίζεται το παράθυρο ιδιοτήτων του **control**. Στη καρτέλα **scale** ρυθμίζουμε το εύρος των τιμών για τη συχνότητα του σήματος.

### Βήμα 9:

Εισάγουμε τα στοιχεία σε μια **While Loop** και εκτελούμε το κώδικα.



# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



## Βήμα 10:

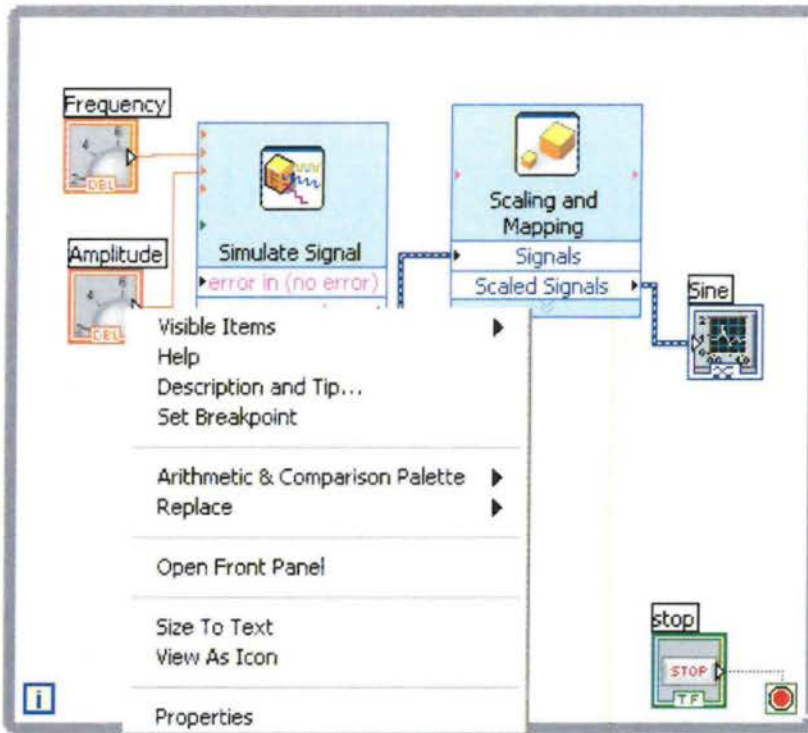
Για την διαμόρφωση της επισκόπησης του σήματος στο καταγραφικό επιλέγουμε από την παλέτα **Αριθμητικών λειτουργιών και συγκρίσεων (Arithmetic & Comparison)** το **Express VI Scaling and Mapping**.



## Βήμα 11:

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

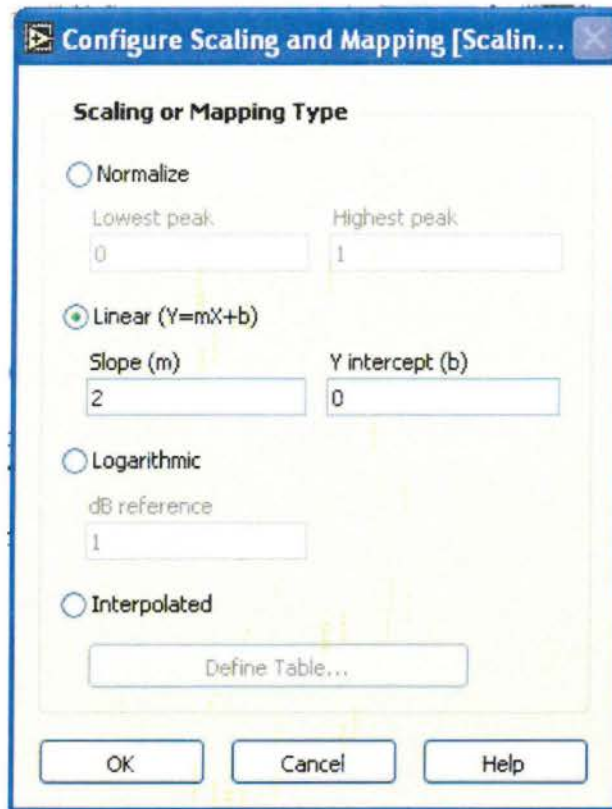
Με δεξί κλικ επάνω στο εικονίδιο της λειτουργίας **Scaling and Mapping** επιλέγουμε ιδιότητες (**properties**).



### Βήμα 12:

Από το παράθυρο διαλόγου του **Express VI** μπορούμε να διαφοροποιήσουμε την κλίμακα ως προς τον τύπο της (γραμμική ή λογαριθμική) καθώς και ως προς το πλάτος της.

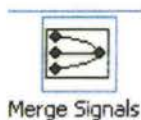
Με την τιμή  $m=2$  στην επιλογή **Linear** ( $Y=mX+b$ ) το πλάτος της κυματομορφής διπλασιάζεται.



## Συστάδα Express VI

Για τη δημιουργία συστάδας σημάτων χρησιμοποιούμε τη λειτουργία **Merge Signals** από την παλέτα **function >> Signal Manipulation**.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Καθώς εκκινούμε το τροποποιημένο VI, παρατηρούμε ότι στο ίδιο γράφημα έχουμε την απεικόνιση των δύο κυματομορφών.

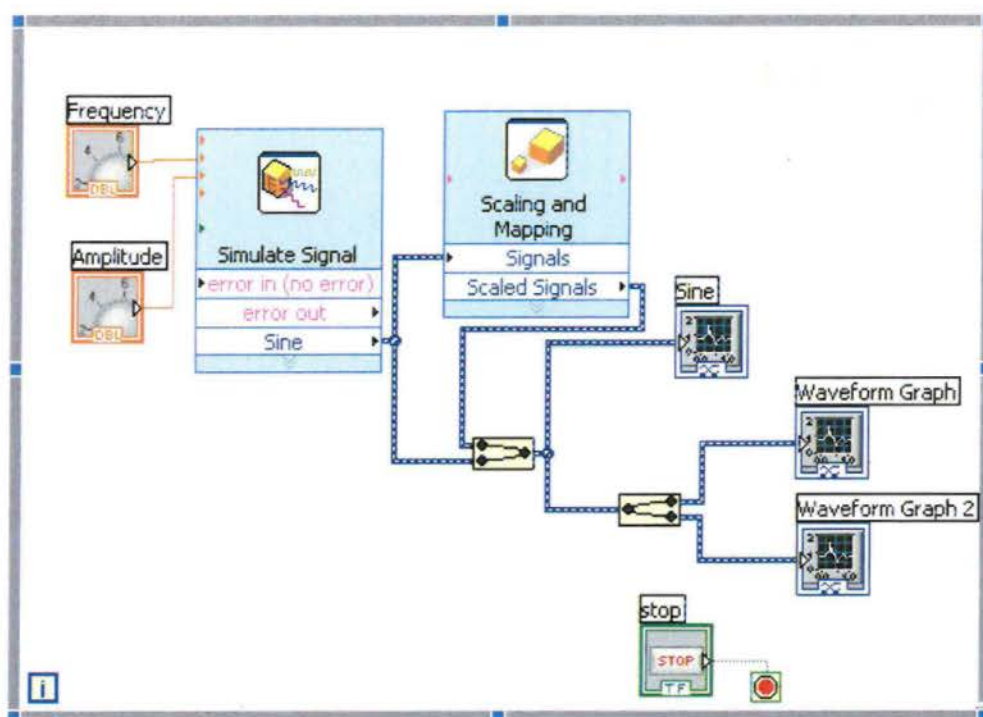
Για το διαχωρισμό της συστάδας των σημάτων χρησιμοποιούμε τη λειτουργία **Split Signals**.





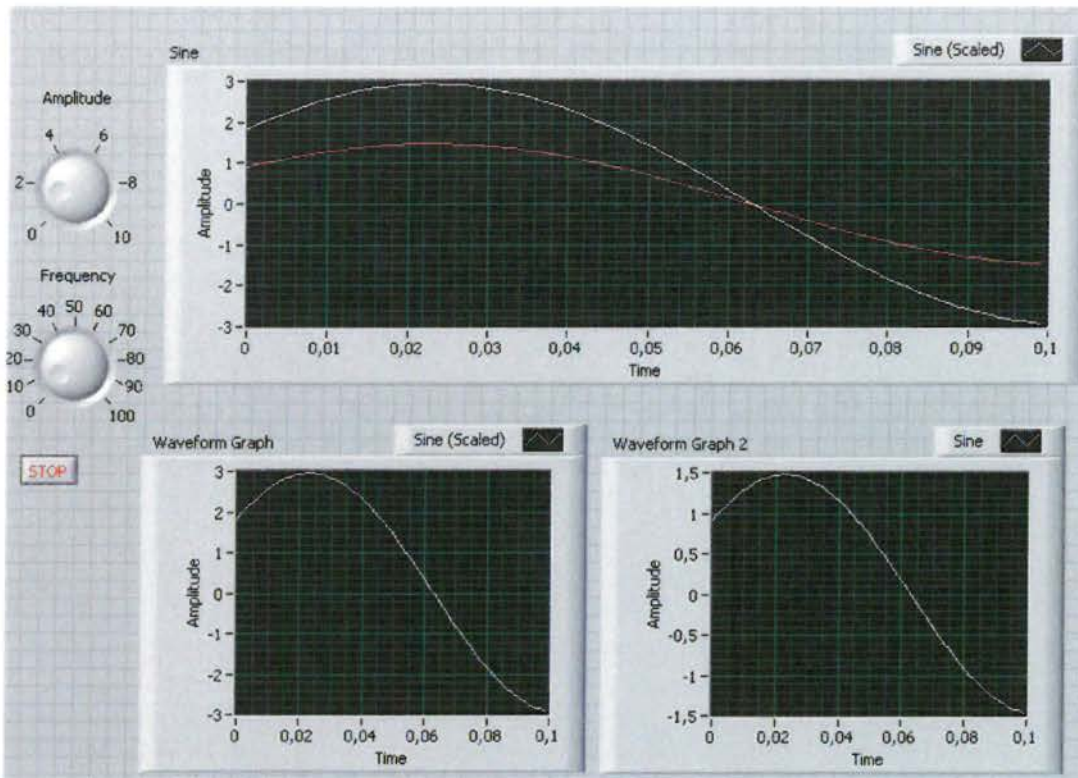
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Τροποποιούμε το VI με τον ακόλουθο τρόπο.



Θέτουμε σε λειτουργία το VI και παρατηρούμε τις κυματομορφές.

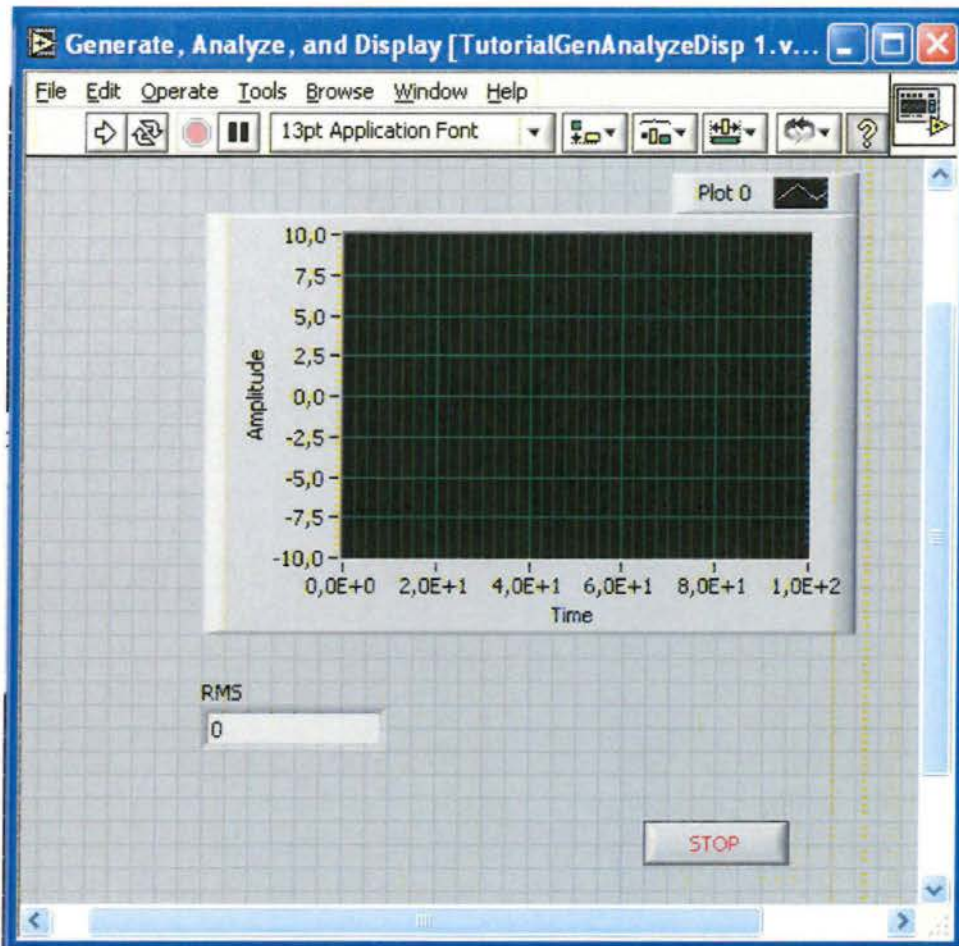
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



### Ανάλυση και Επεξεργασία Σήματος με Express VIs

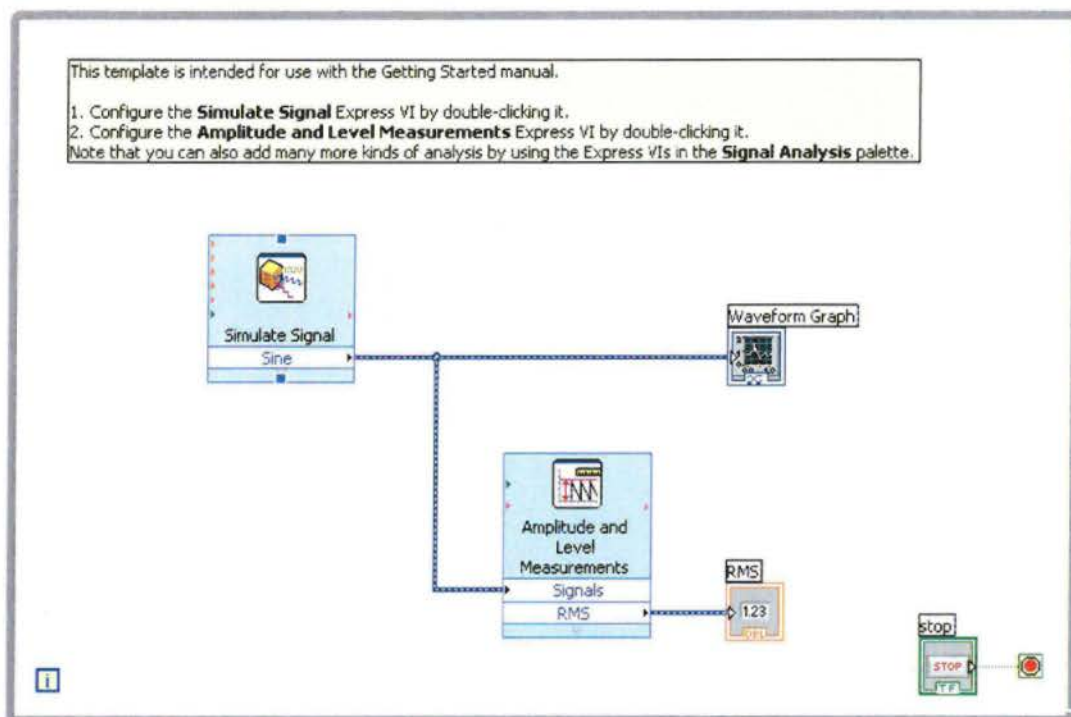
Ανοίγουμε το παράδειγμα **Generate, Analyze, and Display.vi** που βρίσκεται στην επιλογή **NEW** κατά την εκκίνηση του **LabVIEW** για να εξετάσουμε τη μέθοδο προγραμματισμού ανάλυσης και επεξεργασίας σήματος με χρήση των **Express VIs**.

**ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.**



**Front Panel παραδείγματος Generate, Analyze and Display.vi**

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Block diagram του VI

### Βήμα 1:

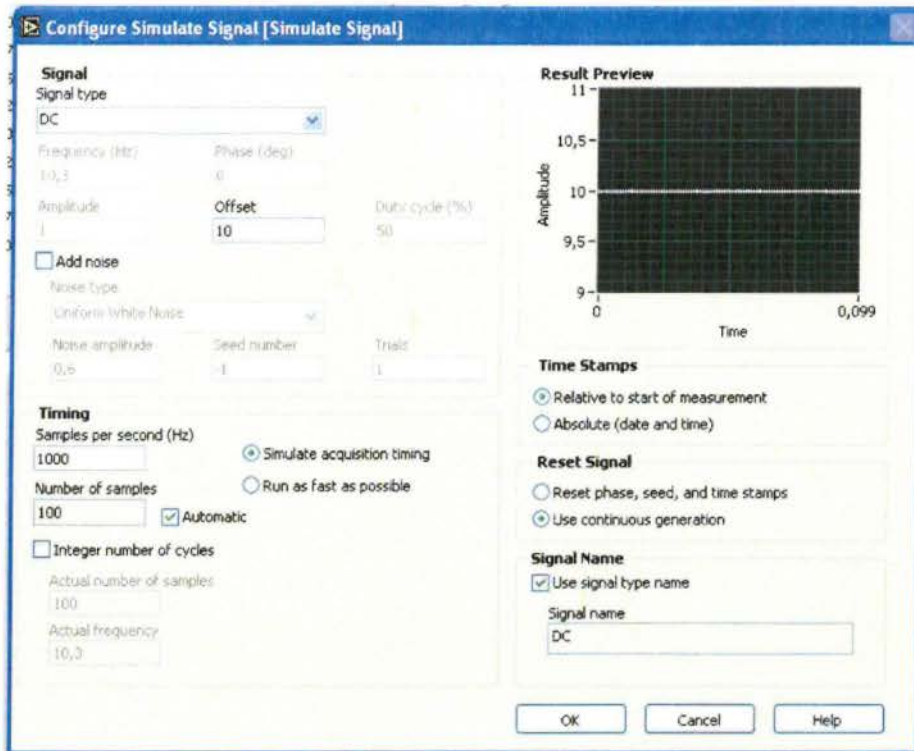
Κάνουμε δεξί κλικ επάνω στο εικονίδιο του **Express VI Simulate Signal** και επιλέγουμε ιδιότητες (**Properties**).

### Βήμα 2:

Στην επιλογή **Signal Type** του παραθύρου διαλόγου ορίζουμε μορφή σήματος **DC** και τοποθετούμε την τιμή **offset 10**.



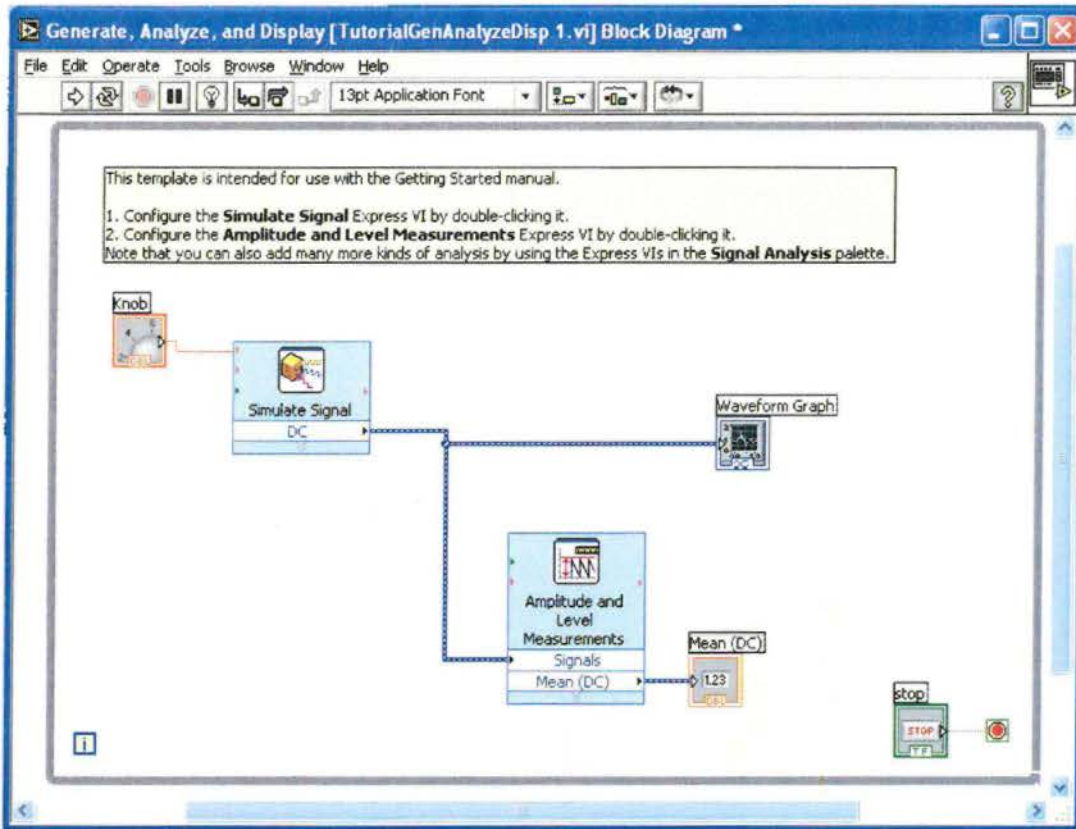
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



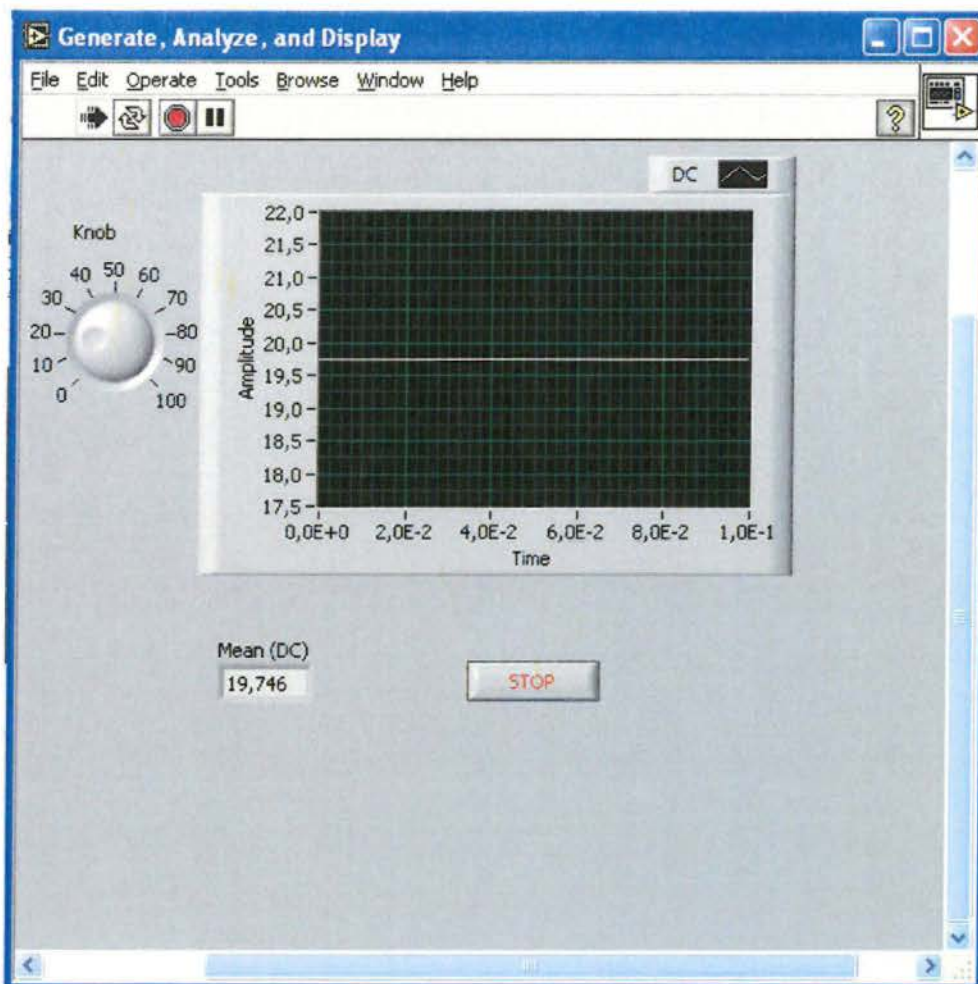
### Βήμα 3:

Συνδέουμε το ποτενσιόμετρο για να έχουμε την δυνατότητα να μεταβάλλουμε την τιμή **offset** του VI και επιλέγουμε έξοδο **DC** από τις ιδιότητες του **Express VI Amplitude and Level Measurements** όπως φαίνεται στη παρακάτω εικόνα.

# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



### Βήμα 4:

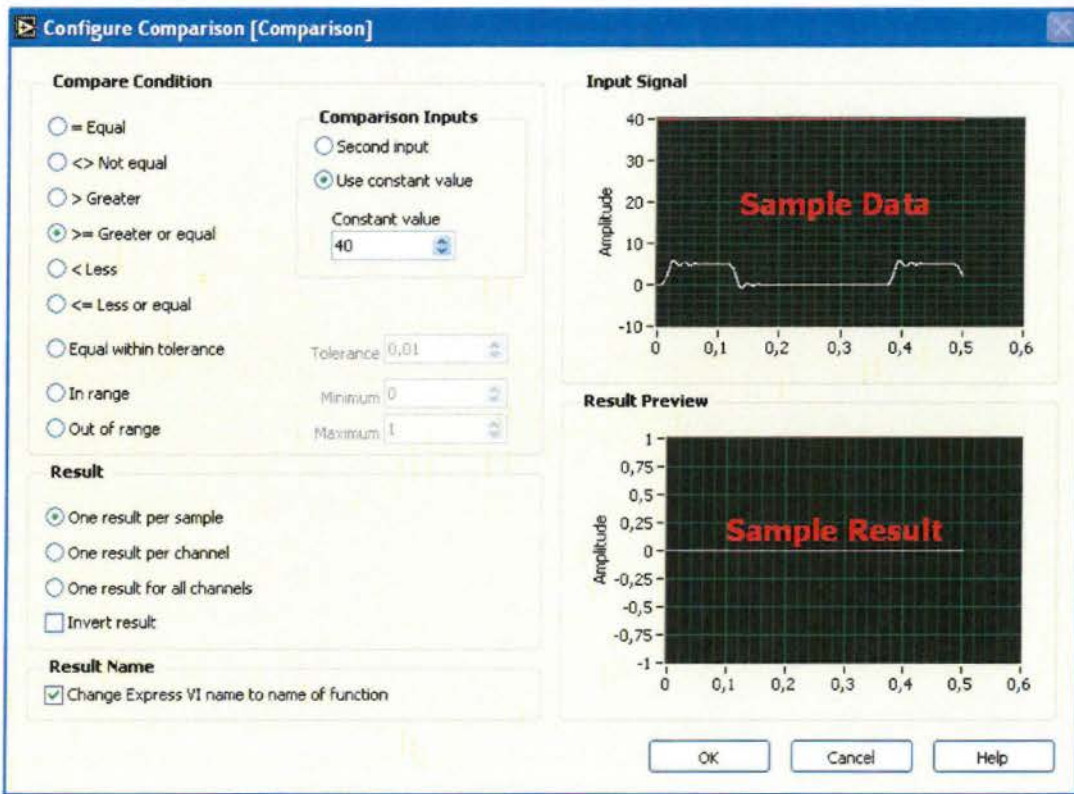
Τοποθετούμε στοιχείο **LED** στο **front panel** του VI. Από την παλέτα **controls** επιλέγουμε την ένδειξη **round LED**. Χρειαζόμαστε το **LED** για να έχουμε ένα οπτικό σύνθημα που δείχνει όταν υπερβαίνει μια τιμή ένα προκαθορισμένο όριο. Μεταβάλλουμε την τιμή του ποτενσιόμετρου και παρατηρούμε τη λειτουργία του **LED**.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

### Βήμα 5:

Για να καθορίσουμε την τιμή στην οποία το LED να ανάβει χρησιμοποιούμε τη λειτουργία του **Express VI Comparison** που βρίσκεται στη παλέτα Αριθμητικών λειτουργιών και Συγκρίσεων (**Arithmetic & Comparison**).

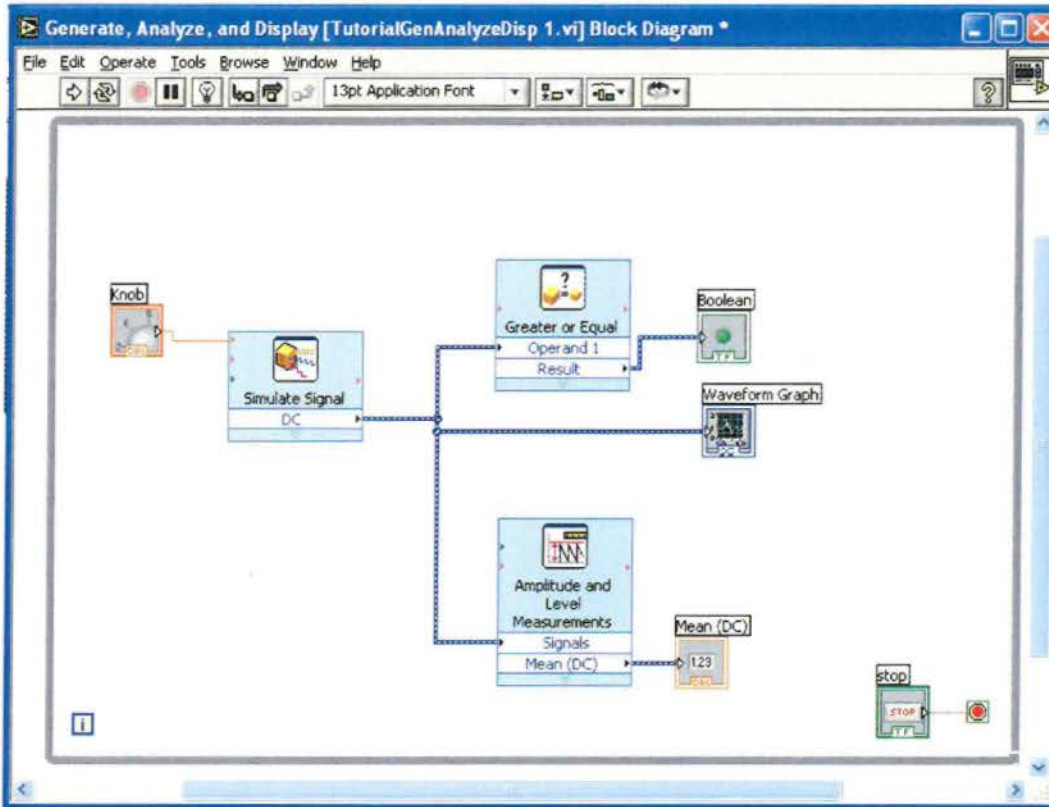
Από τις ιδιότητες του συγκριτή επιλέγουμε την εντολή **Greater or Equal** και ορίζουμε την τιμή 50 .





## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

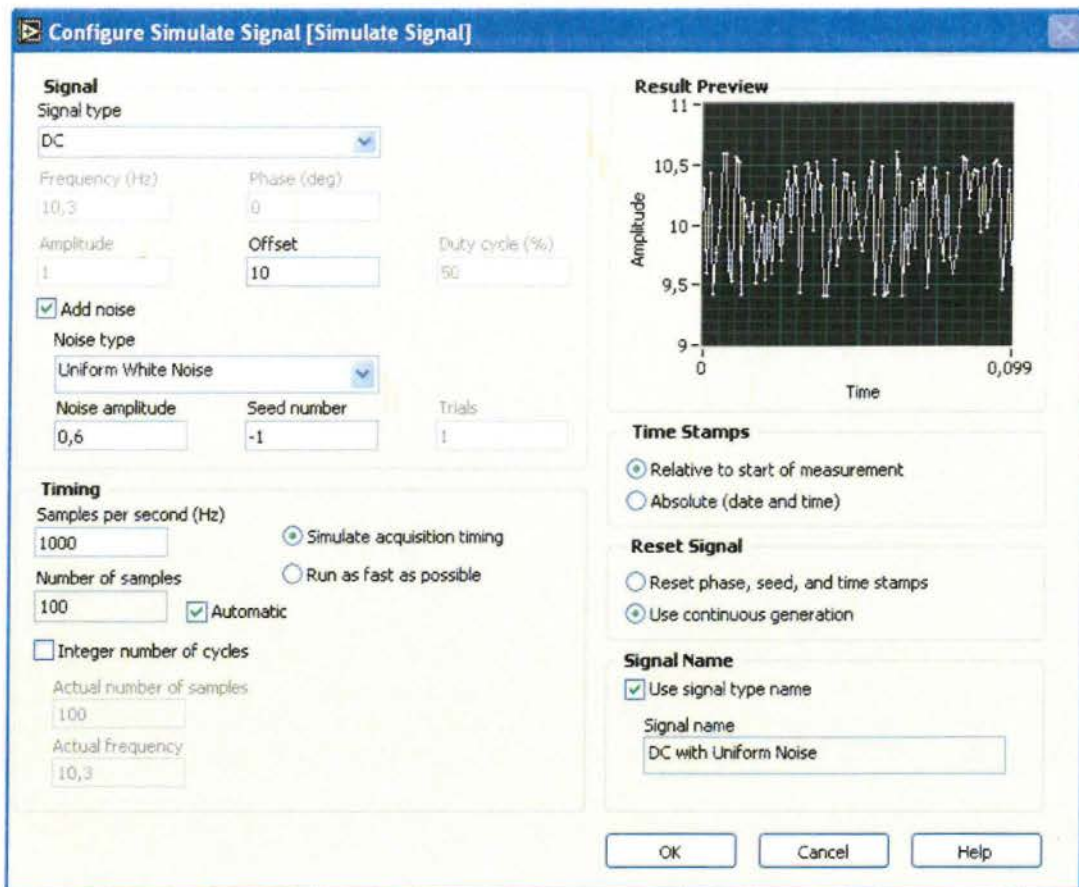
Συνδέουμε το **Express VI** όπως φαίνεται στη παρακάτω εικόνα.



### Βήμα 6:

Μπορούμε να συνεχίσουμε την επεξεργασία του σήματος με την λειτουργία του **Simulation Signal** τσεκάροντας στο παράθυρο ιδιοτήτων **Add noise** και επιλέγοντας **Uniform White Noise**. Με αυτή την ιδιότητα προσθέτουμε μέγεθος θορύβου στην πληροφορία του σήματος της **DC** συνιστώσας.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



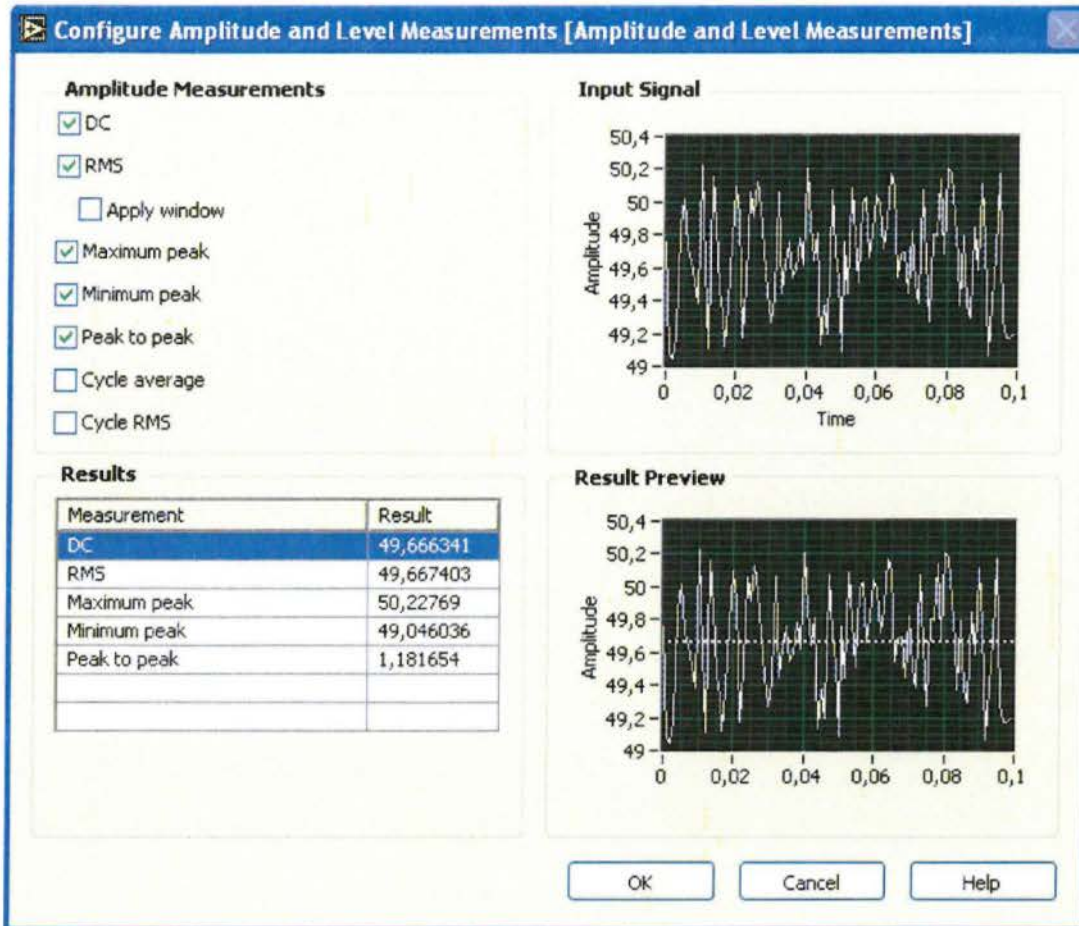
### Βήμα 7:

Τρέχουμε το VI και τοποθετούμε την τιμή του ποτενσιόμετρου κοντά στην τιμή 50. Παρατηρούμε το LED να αναβοσβήνει εξαιτίας της υπέρβασης ορίου λόγω του θορύβου.

### Βήμα 8:

Μπορούμε να προσθέσουμε στο **front panel** ενδείξεις μετρήσεων όπως: η Μέγιστη κορυφή σήματος (**Maximum Peak**), Ελάχιστη κορυφή σήματος (**Minimum Peak**) καθώς και μέτρηση σήματος από κορυφή σε κορυφή (**Peak to Peak**). Για να καθορίσουμε αυτές της μετρήσεις επιλέγουμε τις ιδιότητες του **Express VI Amplitude and Level Measurements** όπως βλέπουμε στην παρακάτω εικόνα:

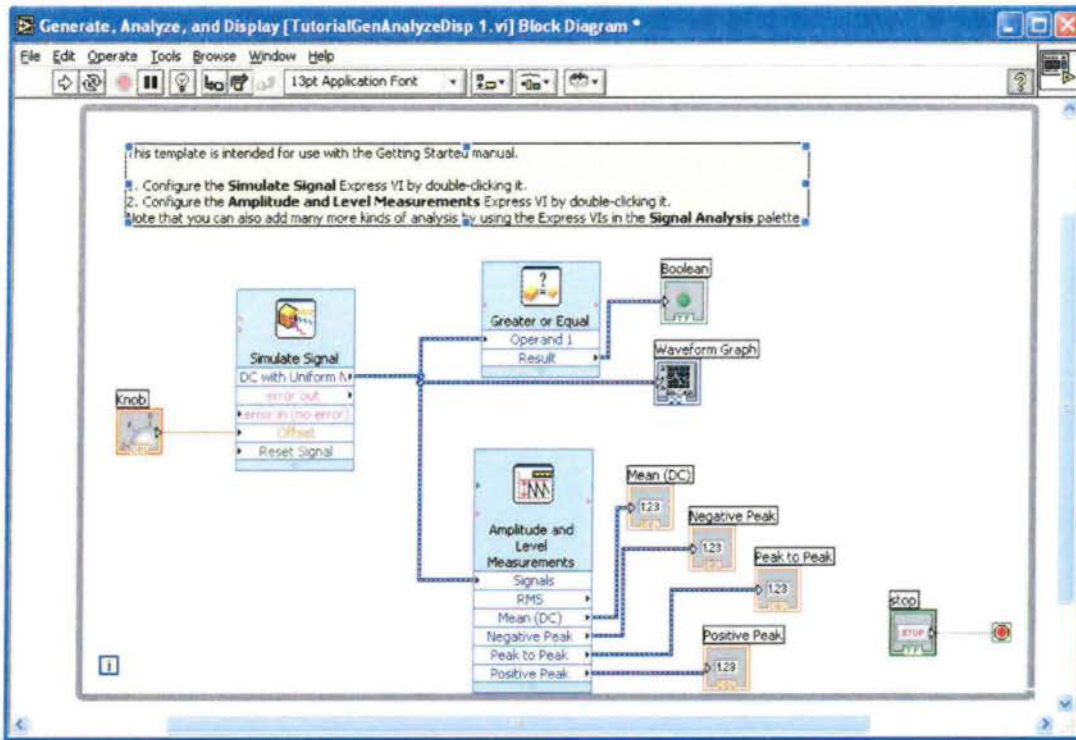
ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



**Βήμα 9:**

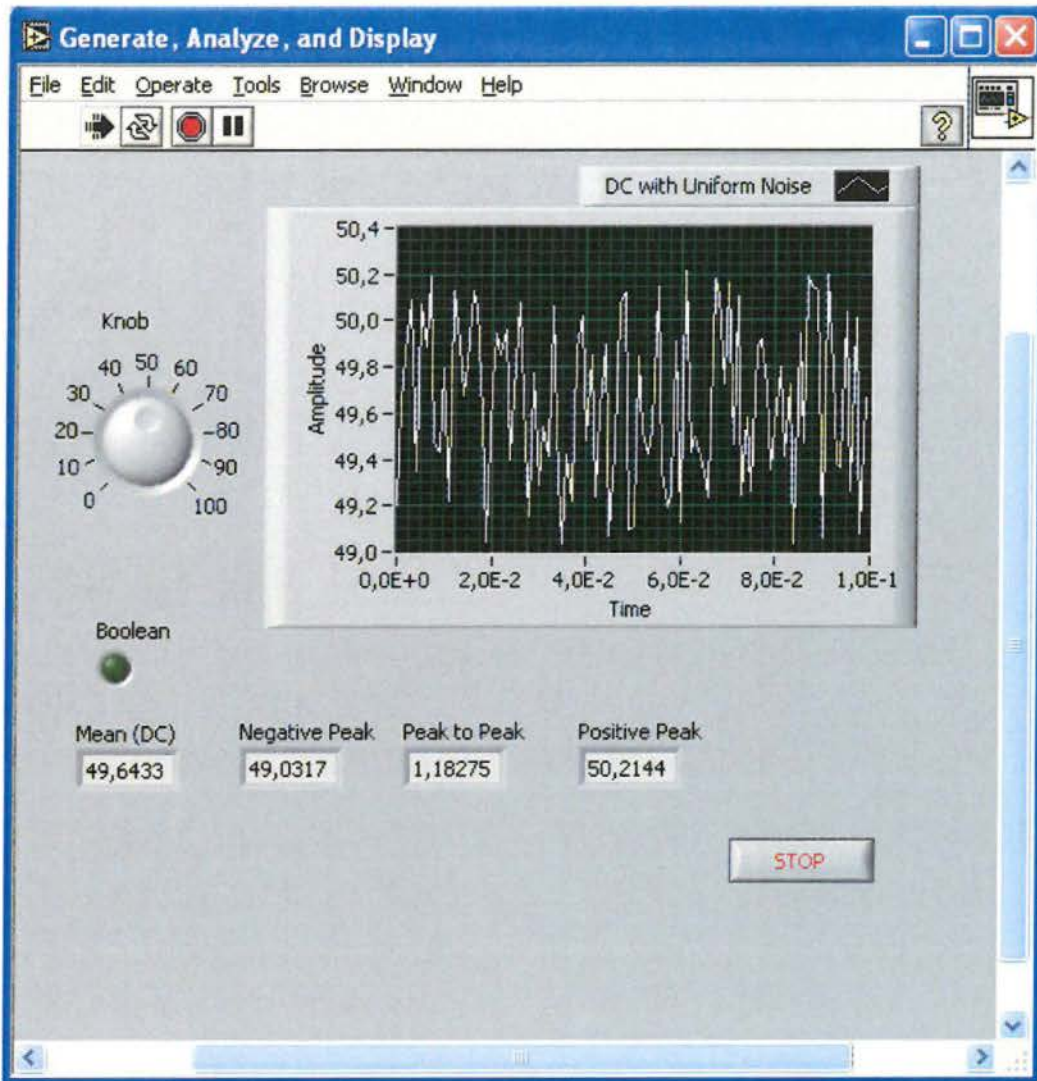
Σε κάθε μια από της 3 εξόδους που δημιουργήσαμε στο **Express VI Amplitude and Level Measurements** πατάμε δεξί κλικ και από το μενού επιλέγουμε **Create >> Numeric Indicator**. Τροποποιούμε το **block diagram** και το **front panel** όπως στη παρακάτω εικόνα.

# ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.





ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



## 6. ΕΦΑΡΜΟΓΕΣ Σ.Α.Ε.

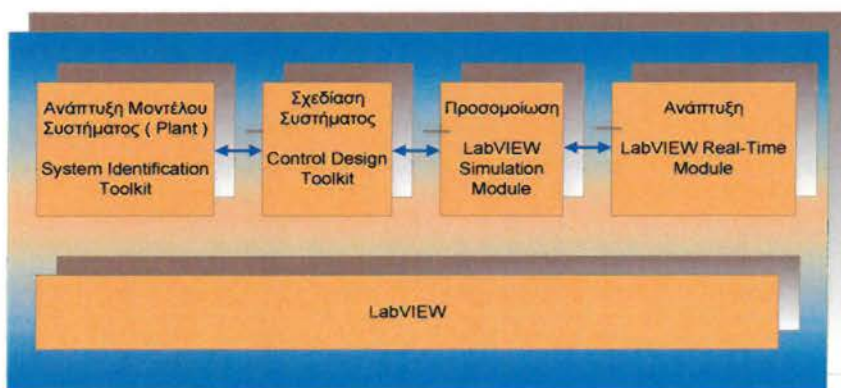
### Labview Control Design Toolkit

Σε αυτό το κεφάλαιο θα δούμε πως μπορούμε να αναλύσουμε συστήματα αυτόματου ελέγχου με την χρήση του **Labview Control Design Toolkit** καθώς θα πούμε λίγα λόγια για το γραφικό του περιβάλλον.

Με τη χρήση γραφικών εργαλείων μπορούμε να δημιουργήσουμε μοντέλα δυναμικών συστημάτων. Το **Control Design Toolkit** υποστηρίζει γραμμικά, χρονικά αμετάβλητα, συνεχή και διακριτά μοντέλα που μπορούν να αναπαρασταθούν ως μοντέλα συνάρτησης μεταφοράς, χώρου κατάστασης ή κέρδους- πόλων-μηδενικών.

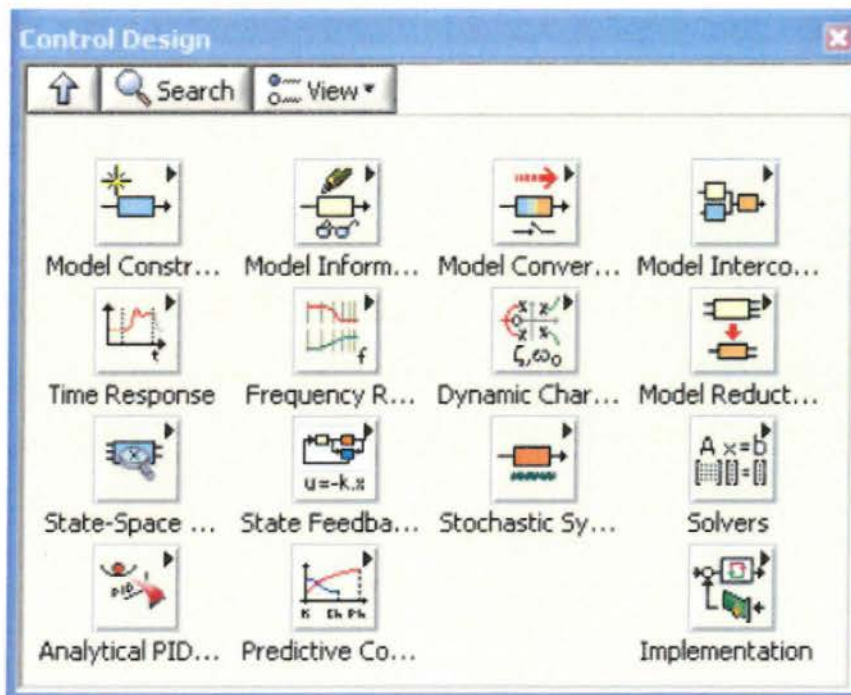
Με τα εργαλεία που παρέχει μπορούμε να πραγματοποιήσουμε μετατροπές ανάμεσα στις διάφορες μορφές των μοντέλων καθώς και να κάνουμε σύνδεση μεταξύ των μοντέλων σε σειρά, παράλληλα, με ανάδραση κ.α. Μπορούμε να κάνουμε ανάλυση της χρονικής απόκρισης, της απόκρισης συχνότητας και των χαρακτηριστικών ενός συστήματος όπως είναι η ευστάθεια, το κέρδος και ο συντελεστής απόσβεσης, χρησιμοποιώντας ενσωματωμένα **VI's (Virtual Instruments-Εικονικά Όργανα)** όπως είναι το **CD Bode.vi**, **CD Nyquist.vi** και το **CD Root Locus.vi**.

Στο εικόνα που ακολουθεί φαίνεται πως χρησιμοποιείται το Labview στη μελέτη και σχεδίαση ενός συστήματος ελέγχου.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Το **Control Design Toolkit** παρέχει το διαδραστικό εργαλείο **Control Design Assistant** που είναι διαθέσιμο από το **Tools / Control Design Toolkit** και μια **βιβλιοθήκη με VIs**, για τη μελέτη και σχεδίαση ενός συστήματος αυτομάτου ελέγχου. Όταν εγκαθίσταται το **Control Design Toolkit**, η **Control Design Palette** είναι διαθέσιμη από την παλέτα **Functions** και φαίνεται στην εικόνα 6.1.



Εικόνα 6.1: Απεικόνιση Control Design Palette

Ακολουθούν μερικοί από τους βασικότερους τύπους των δομικών στοιχείων στην εργαλειοθήκη:

- **Model Construction palette**, με τις παρακάτω συναρτήσεις και υποπαλέτες:
  - Construct State-Space Model
  - Construct Transfer Function Model
  - Construct Zero-Pole-Gain Model
  - Construct Random Model

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

- Construct Special Model
  - Draw Transfer Function Equation
  - Draw Zero-Pole-Gain Equation
  - Read Model From File
  - Write Model From File
  - Model Information palette
- 
- **Model Conversion palette, με τις παρακάτω συναρτήσεις:**
    - Convert to State-Space Model
    - Convert to Transfer Function Model
    - Convert to Zero-Pole-Gain Model
    - Convert Delay with Pade Approximation
    - Convert Delay to Poles at Origin
    - Convert Continuous to Discrete (με διάφορες μεθόδους όπως Tustin, zero order hold κ.α)
    - Convert Discrete to Discrete
    - Convert Discrete to Continuous
    - Convert Control Design to Simulation
    - Convert Simulation to Control Design
- 
- **Model Interconnection palette, με τις παρακάτω συναρτήσεις:**
    - Serial
    - Parallel
    - Feedback
    - Append
- 
- **Time Response palette, με τις παρακάτω συναρτήσεις:**
    - Step Response
    - Impulse Response
    - Initial Response
    - Linear Simulation
    - Get Time Response Data



**ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ  
ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.**

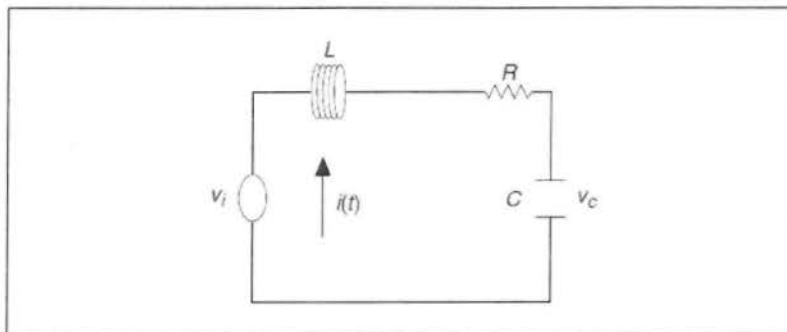
- **Frequency Response palette, με τις παρακάτω συναρτήσεις:**
  - Bode Nyquist
  - Nichols
  - Singular Values
  - All Margins
  - Gain and Phase Margin
  - Evaluate at Frequency
  - Bandwidth
  - Get Frequency Response Data
  
- **The Dynamic Characteristics palette, με τις παρακάτω συναρτήσεις:**
  - Root Locus
  - Pole-Zero Map
  - Damping Ratio and Natural Frequency
  - DC Gain
  - Stability
  - Norm
  - Covariance Response
  - Total Delay
  - Distribute Delay
  - Parametric Time Response
  
- **The State Space Model Analysis palette, με τις παρακάτω συναρτήσεις:**
  - Controllability Matrix
  - Observability Matrix
  - Grammians
  - Canonical State-Space Realization
  - Balance State-Space Model (Diagonal)
  - Balance State-Space Model (Grammians)
  - Controllability Staircase
  - Observability Staircase

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

- State Similarity Transform

Στην συνέχεια θα ακολουθήσουν μερικά παραδείγματα στα οποία θα περιγραφεί αναλυτικά, βήμα προς βήμα, η υλοποίηση τους για την κατανόηση της χρήσης του **Control Design Toolkit**.

Στο ακόλουθο σχήμα απεικονίζεται ένα κύκλωμα RLC σειράς . Οι τιμές της αντίστασης είναι  $R=20\Omega$ , του πηνίου  $L=50\text{mH}$  και του πυκνωτή  $C=10\mu\text{F}$ .



Η συνάρτηση μεταφοράς ανάμεσα στην τάση εισόδου  $V_i$  και την τάση του πυκνωτή  $V_c$

είναι η εξίσωση: 
$$H(s) = \frac{1}{s^2 + \frac{Rs}{L} + \frac{1}{LC}}$$

Η υλοποίηση στο Labview γίνεται εφαρμόζοντας τα παρακάτω βήματα:

### Βήμα 1:

Από την οθόνη “Getting Started” ανοίξτε ένα νέο “Blank VI”. Πηγαίστε στο block diagram (Ctrl+E).

### Βήμα 2:

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Πατήστε δεξί-κλικ οπουδήποτε στο block diagram για να εμφανιστεί η παλέτα **Functions**. Για να μη φύγει και χρειαστεί να επαναλάβετε αυτό το βήμα, πατήστε την πινέζα που βρίσκεται πάνω αριστερά, μπροστά από την λέξη **“Functions”**.

### Βήμα 3:

Εισάγετε το “CD Construct Transfer Function Model.vi” στο block diagram. Από την παλέτα Functions πηγαίνετε στο **Control Design & Simulation >> Control Design >> Model Construction >> CD Construct Transfer Function Model.vi**, επιλέξτε το και τοποθετήστε το στο block diagram. Από το μενού που υπάρχει κάτω από το vi επιλέξτε **Single-Input Single-Output (Symbolic)**.

### Βήμα 4:

Δεξί-κλικ πάνω στο **CD Construct Transfer Function Model.vi >> Visible Items >> Label**, για να εμφανιστεί η ετικέτα με το όνομα του vi.

### Βήμα 5:

Δεξί-κλικ στην είσοδο **Symbolic Numerator** του CD Construct Transfer Function Model.vi και επιλέξτε **Create >> Control**.

### Βήμα 6:

Δεξί-κλικ στην είσοδο **Symbolic Denominator** του CD Construct Transfer Function Model.vi και επιλέξτε **Create >> Control**.

### Βήμα 7:

Δεξί-κλικ στην είσοδο **Variables** του CD Construct Transfer Function Model.vi και επιλέξτε **Create >> Control**.

### Βήμα 8:

Εισάγετε το “CD Draw Transfer Function Equation.vi” στο block diagram. Από την παλέτα Functions πηγαίνετε στο **Control Design & Simulation >> Control Design >> Model Construction >> CD Draw Transfer Function Equation.vi**, επιλέξτε το και τοποθετήστε το στο block diagram. Συμβουλευτείτε το βήμα 5 για να εμφανιστεί η ετικέτα.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

### Βήμα 9:

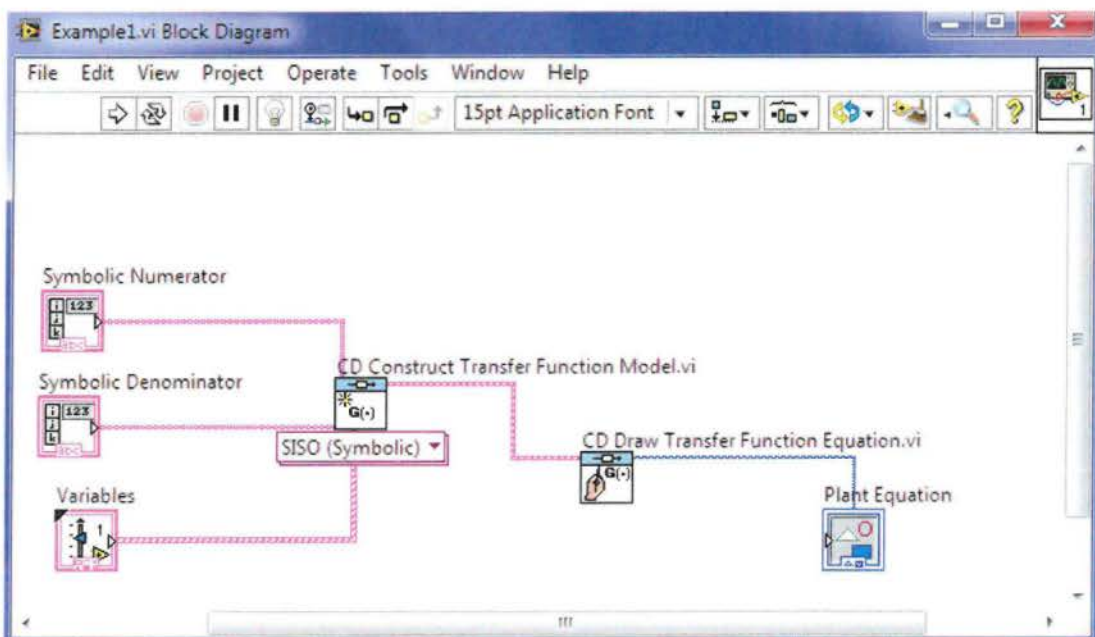
Συνδέστε την έξοδο **Transfer Function Model** του CD Construct Transfer Function Model.vi στην είσοδο **Transfer Function Model** του CD Draw Transfer Function Equation.vi.

### Βήμα 10:

Δεξί-κλικ στην έξοδο **Equation** του CD Draw Transfer Function Equation.vi και επιλέξτε **Create >> Indicator**.

### Βήμα 11:

Διπλό-κλικ στην ετικέτα του Indicator και την μετονομάστε την σε **Plant Equation**. Το block diagram πρέπει να έχει τη μορφή του της εικόνας.



### Βήμα 12:



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Επιλέξτε File >> Save As.. και σώστε το VI με το όνομα “Example1”.

### Βήμα 13:

Αλλάξτε στο front panel του VI (Ctrl+E).

### Βήμα 14:

Δεξί-κλικ στο κελί Value, μέσα στο Variables, πατήστε **Display Format...**, και επιλέξτε **Automatic Formatting**. Αυτό επιτρέπει να φαίνονται οι μικροί αριθμοί σωστά.

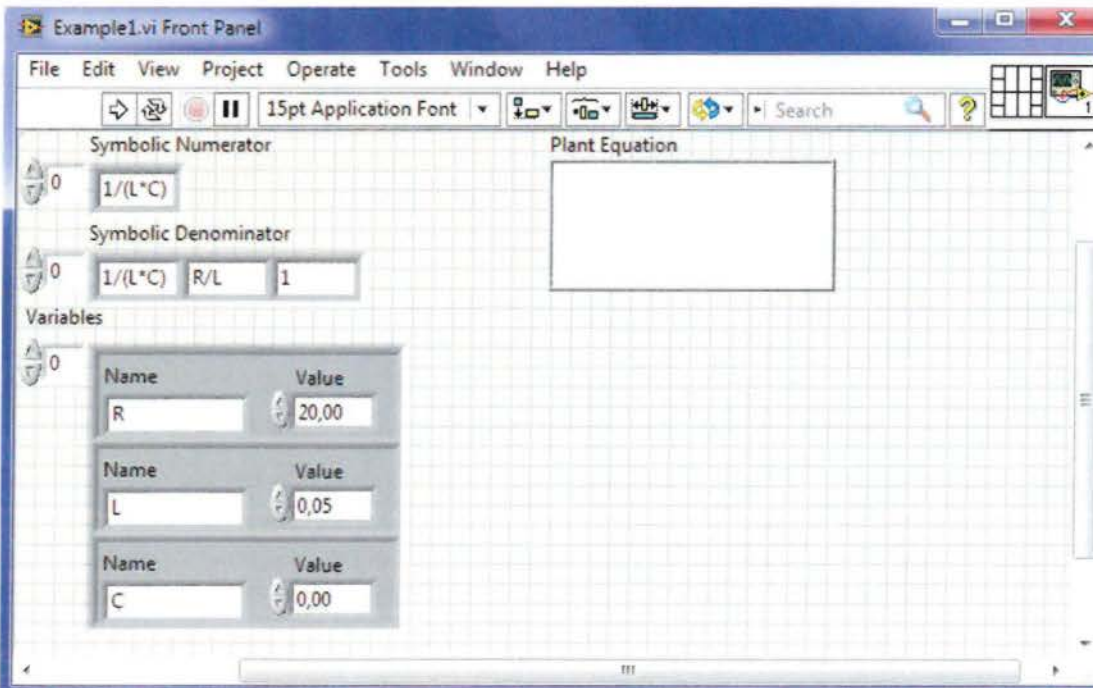
### Βήμα 15:

Δώστε στα **Symbolic Numerator**, **Symbolic Denominator** και **Variables** τις τιμές τους σύμφωνα με την εξίσωση και τα δεδομένα που δίνονται στην αρχή του παραδείγματος.

### Βήμα 16:

Ορίστε τις τιμές ως προεπιλογή. Επιλέξτε **Edit >> Make Current Values Default**.

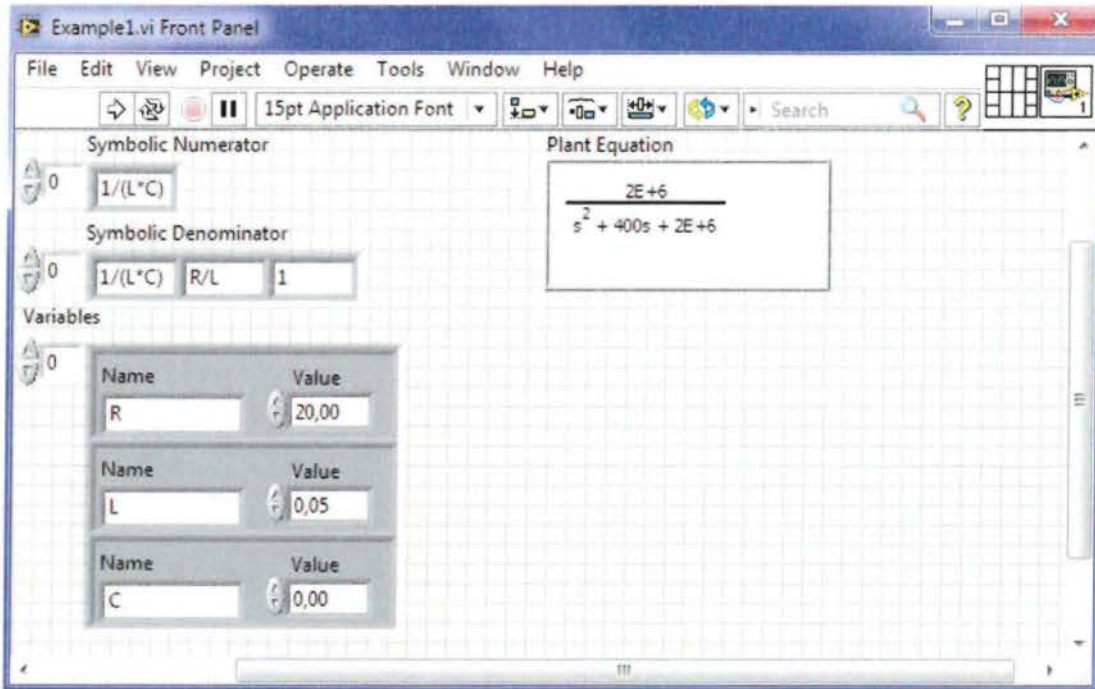
Το front panel πρέπει να έχει τη μορφή της εικόνας που ακολουθεί.



### Βήμα 17:

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Τρέχοντας το VI (Ctrl+R) παρατηρούμε την συνάρτηση μεταφοράς μέσα στο Plant Equation.

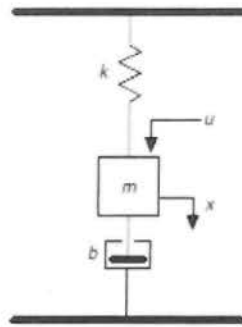


### Βήμα 18:

Αποθηκεύστε το VI (Ctrl+S).

Με το παρακάτω VI προσομοιώνονται η βηματική και κρουστική απόκριση της μετατόπισης του συστήματος μάζας-ελατηρίου

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Μπορούμε να αναπαραστήσουμε το παραπάνω σύστημα με το ακόλουθο μοντέλο στο χώρο κατάστασης:

$$\dot{x} = Ax + Bu = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

$$y = Cx + Du = [1 \quad 0]x + [0]u = x$$

Για το παράδειγμα αυτό, θεωρήστε τις ακόλουθες τιμές:

$$k = 50 \frac{kN}{cm} \quad m = 10kg \quad b = 20 \frac{kN \cdot s}{cm}$$

Θα δείξουμε αναλυτικά πώς με την χρήση του LabVIEW Control Design Toolkit κατασκευάζονται:

- Ένα μοντέλο στο χώρο κατάστασης.
- Το γράφημα βηματικής απόκρισης.
- Το γράφημα κρουστικής απόκρισης.

Να εμφανίζονται τα παραμετρικά δεδομένα της βηματικής απόκρισης.

Η υλοποίηση στο LabVIEW γίνεται εφαρμόζοντας τα παρακάτω βήματα:

### Βήμα 1:

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Από την οθόνη “Getting Started” ανοίξτε ένα νέο “Blank VI”. Πηγαίνατε στο block diagram (Ctrl+E).

### **Βήμα 2:**

Πατήστε δεξί-κλικ οπουδήποτε στο block diagram και για να εμφανιστεί η παλέτα **Functions**.

### **Βήμα 3:**

Εισάγετε το “CD Construct State-Space Model.vi” στο block diagram. Από την παλέτα Functions πηγαίνατε στο **Control Design & Simulation >> Control Design >> Model Construction >> CD Construct State-Space Model.vi**, επιλέξτε το και τοποθετήστε το στο block diagram. Από το μενού που υπάρχει κάτω από το vi επιλέξτε **Symbolic**.

### **Βήμα 4:**

Δεξί-κλικ πάνω στο **CD Construct State-Space Model.vi >> Visible Items >> Label**, για να εμφανιστεί η ετικέτα με το όνομα του vi.

### **Βήμα 5:**

Δεξί-κλικ στην είσοδο **Symbolic A** του CD Construct State-Space Model.vi και επιλέξτε **Create >> Control**.

### **Βήμα 6:**

Δεξί-κλικ στην είσοδο **Symbolic B** του CD Construct State-Space Model.vi και επιλέξτε **Create >> Control**.

### **Βήμα 7:**

Δεξί-κλικ στην είσοδο **Symbolic C** του CD Construct State-Space Model.vi και επιλέξτε **Create >> Control**.

### **Βήμα 8:**

Δεξί-κλικ στην είσοδο **Symbolic D** του CD Construct State-Space Model.vi και επιλέξτε **Create >> Control**.

### **Βήμα 9:**

Δεξί-κλικ στην είσοδο **Variables** του CD Construct State-Space Model.vi και επιλέγουμε **Create >> Control**.

### **Βήμα 10:**

Εισάγετε το “CD Step Response.vi” στο block diagram. Από την παλέτα Functions πηγαίνατε στο **Control Design & Simulation >> Control Design >> Time Response >> CD Step Response.vi**, επιλέξτε το και τοποθετήστε το στο block diagram.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

### **Βήμα 11:**

Συνδέστε την έξοδο **State Space Model** του **CD Construct State-Space Model.vi**, στην είσοδο **State Space Model** του **CD Step Response.vi**.

### **Βήμα 12:**

Δεξί-κλικ στην έξοδο **Step Response Graph** του **CD Step Response.vi** και επιλέξτε **Create >> Indicator**.

### **Βήμα 13:**

Εισάγετε το “**CD Impulse Response.vi**” στο block diagram. Από την παλέτα **Functions** πηγαίνετε στο **Control Design & Simulation >> Control Design >> Time Response >> CD Impulse Response.vi**, επιλέξτε το και τοποθετήστε το στο block diagram.

### **Βήμα 14:**

Συνδέστε την έξοδο **State Space Model** του **CD Construct State-Space Model.vi**, στην είσοδο **State Space Model** του **CD Impulse Response.vi**.

### **Βήμα 15:**

Δεξί-κλικ στην έξοδο **Impulse Response Graph** του **CD Impulse Response.vi** και επιλέξτε **Create >> Indicator**.

### **Βήμα 16:**

Εισάγετε το “**CD Parametric Time Response.vi**” στο block diagram. Από την παλέτα **Functions** πηγαίνετε στο **Control Design & Simulation >> Control Design >> Time Response >> CD Parametric Time Response.vi**, επιλέξτε το και τοποθετήστε το στο block diagram.

### **Βήμα 17:**

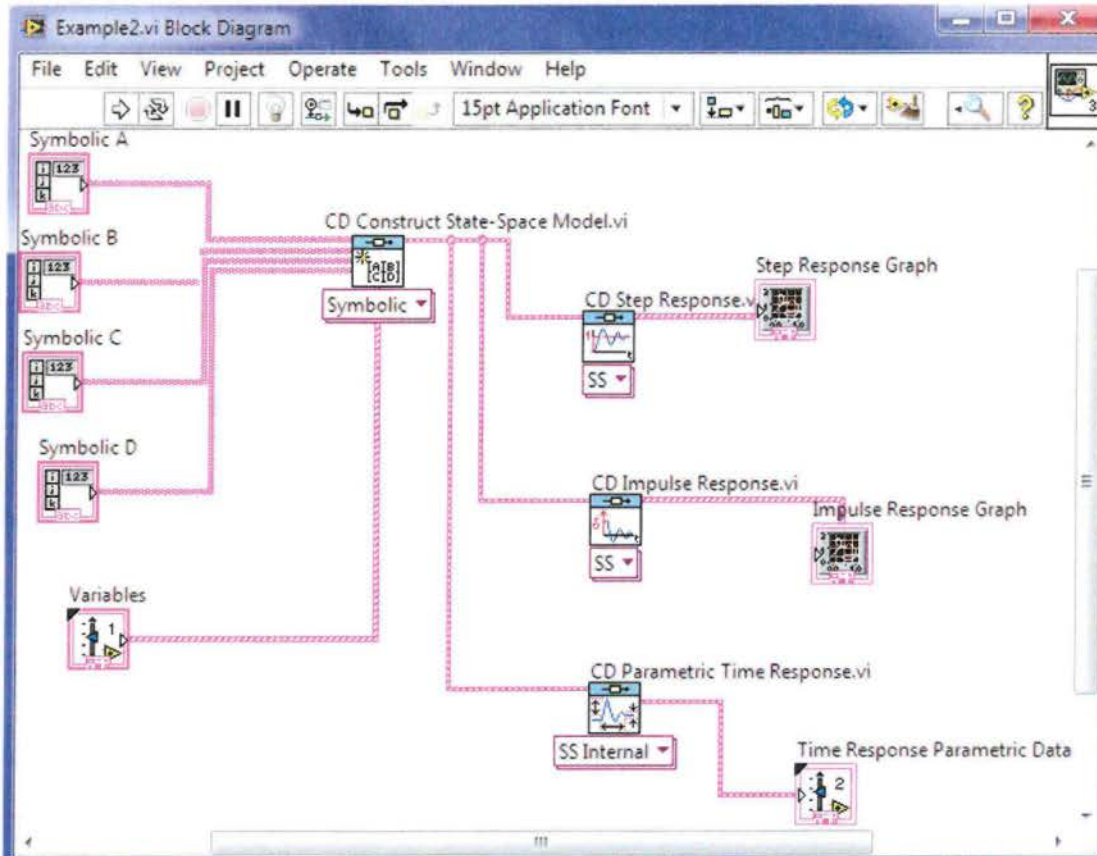
Συνδέστε την έξοδο **State Space Model** του **CD Construct State-Space Model.vi**, στην είσοδο **State Space Model** του **CD Parametric Time Response.vi**.

### **Βήμα 18:**

Δεξί-κλικ στην έξοδο **Time Response Parametric Data** του **CD Parametric Time Response.vi** και επιλέξτε **Create >> Indicator**.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Στο σχήμα που ακολουθεί απεικονίζεται το block diagram μετά την ολοκλήρωση των παραπάνω βημάτων.



### Βήμα 19:

Επιλέξτε File >> Save As.. και σώστε το VI με το όνομα “Example 2”.

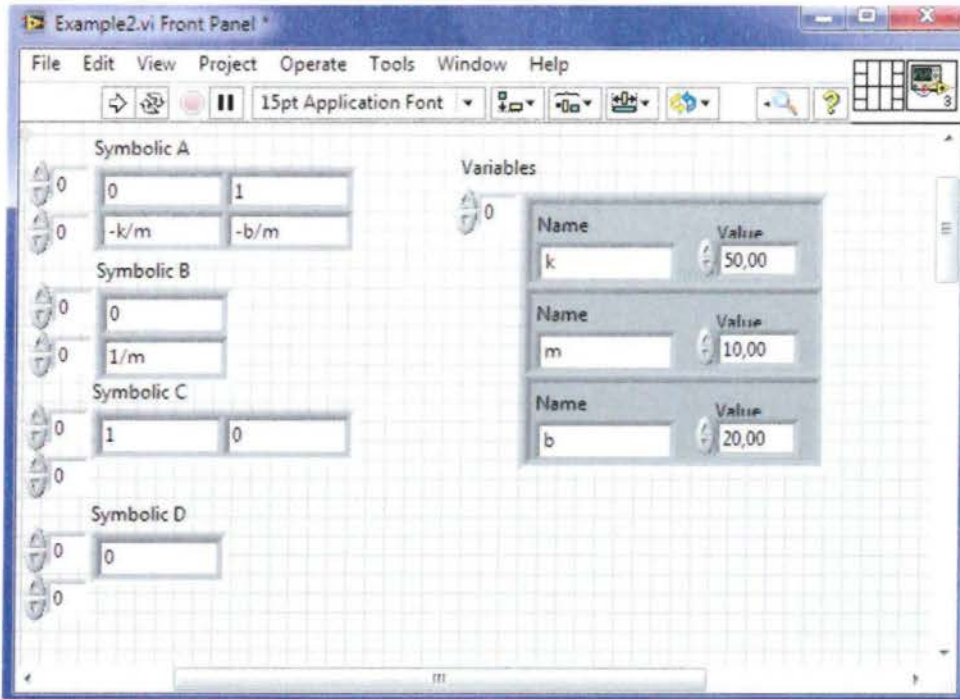
### Βήμα 20:

Αλλάξτε στο front panel του VI (Ctrl+E).

Δώστε στα Symbolic A, Symbolic B, Symbolic C, Symbolic D και Variables τις τιμές τους σύμφωνα με την εξίσωση και τα δεδομένα που δίνονται στην αρχή του παραδείγματος.

Το front panel θα έχει την ακόλουθη μορφή.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



### Βήμα 21:

Ορίστε τις τιμές ως προεπιλογή. Επιλέξτε **Edit >> Make Current Values Default**.

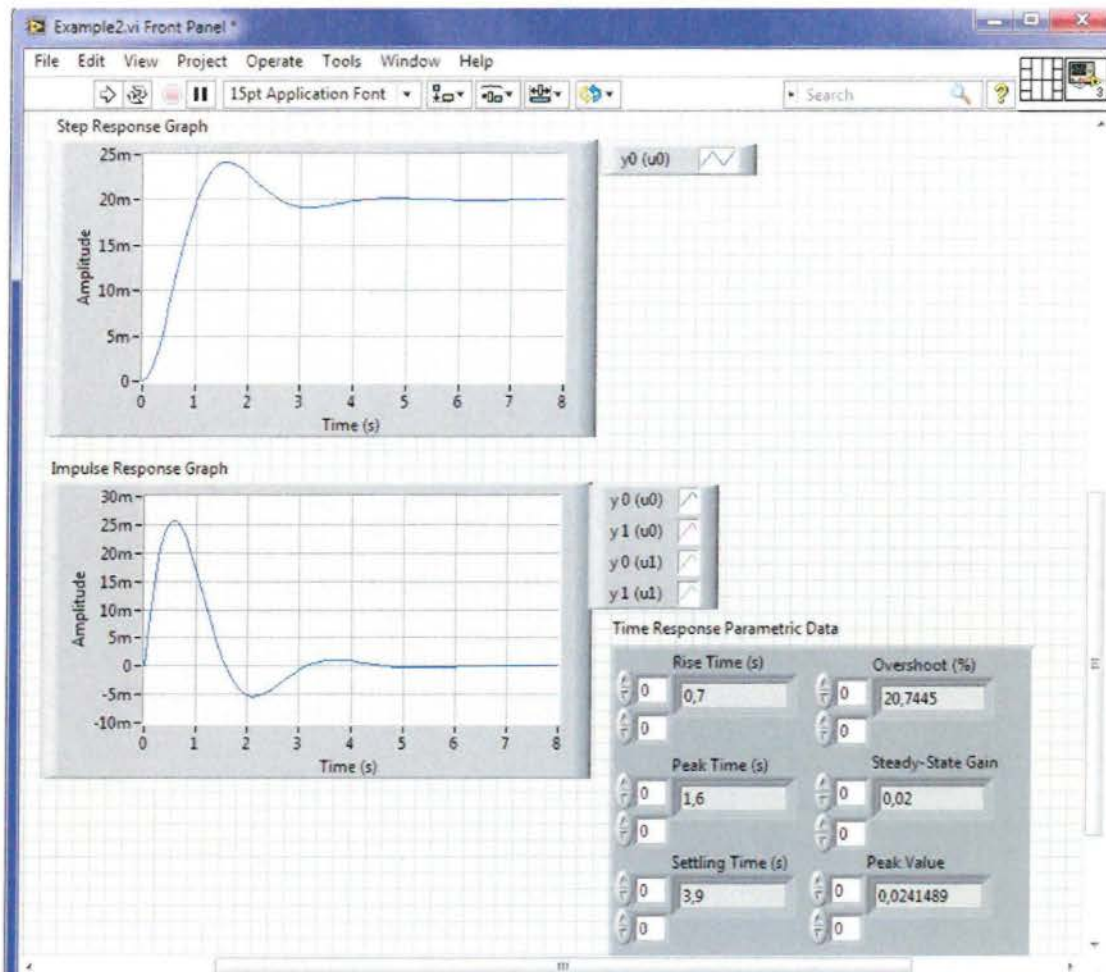
### Βήμα 22:

Για να αλλάξετε τη μορφή των αριθμών στους άξονες των δύο γραφημάτων κάντε δεξί-κλικ μέσα στο γράφημα >> **X Scale >> Formatting >> Automatic Formatting**. Την ίδια διαδικασία και για το Y Scale.

### Βήμα 23:

Όταν τρέξουμε το VI με την εντολή (Ctrl+R) παίρνουμε τα ακόλουθα αποτελέσματα.

## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



### Βήμα 23:

Αποθηκεύστε το VI (Ctrl+S).



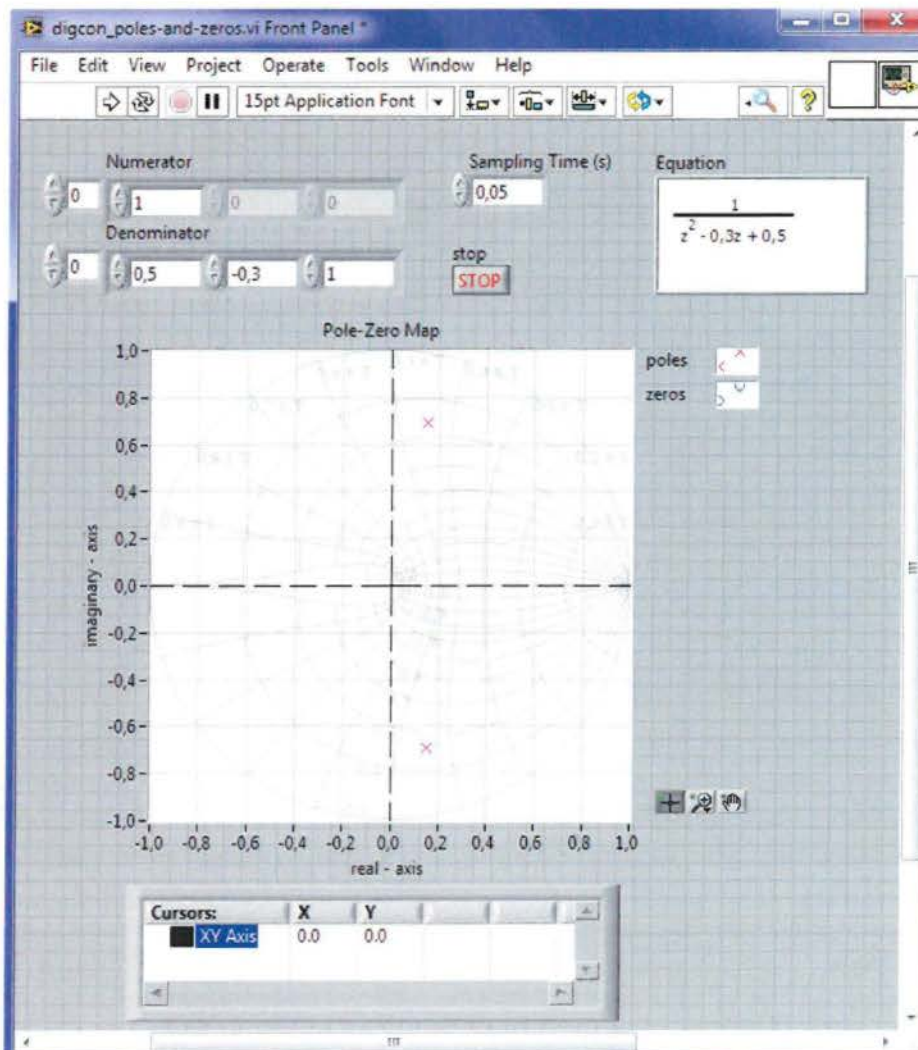
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Στο παράδειγμα που ακολουθεί θα σχεδιάσουμε τους πόλους και τα μηδενικά ενός ψηφιακού συστήματος.

Έστω η συνάρτηση μεταφοράς διακριτού χρόνου  $\frac{1}{z^2 - 0.3z + 0.5}$

Μπορούμε να σχεδιάσουμε τους πόλους και τα μηδενικά της συνάρτησης μεταφοράς χρησιμοποιώντας το **CD Pole-Zero Map VI**.

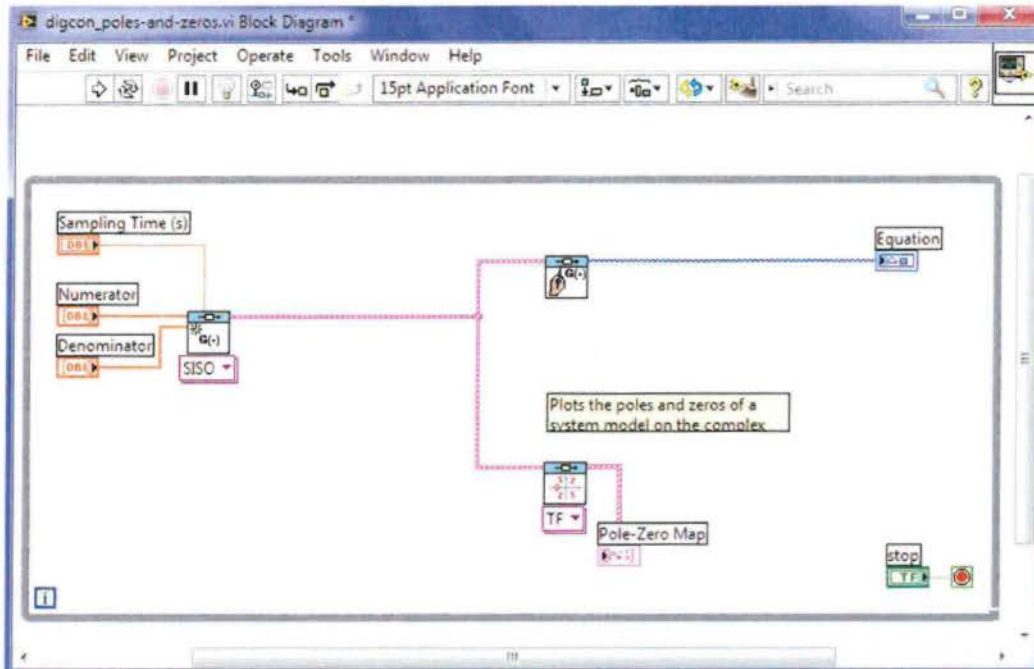
Ανοίξτε το αρχείο **digcon\_poles-and-zeros.vi**. Τρέχοντας την προσομοίωση προκύπτει το ζητούμενο διάγραμμα πόλων και μηδενικών όπως φαίνεται στο front panel που ακολουθεί.





## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Παρακάτω απεικονίζεται το αντίστοιχο **block diagram** του υλοποιημένου συστήματος στο Labview.

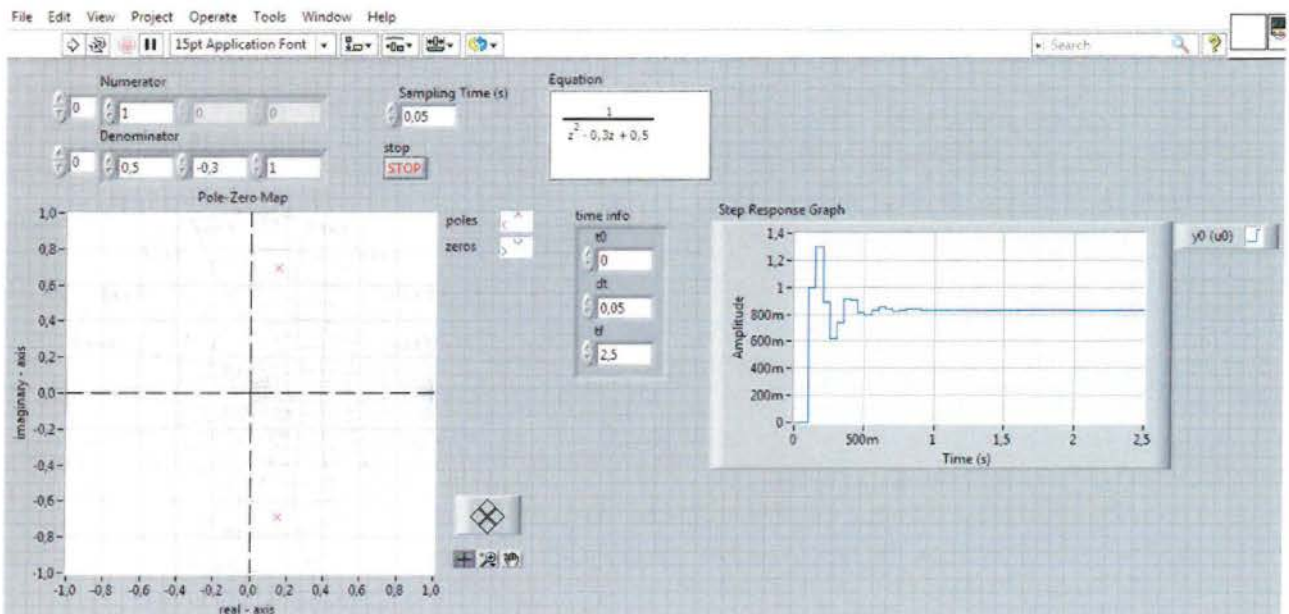


## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

Σε αυτό το παράδειγμα θα μπορέσουμε να σχεδιάσουμε τη βηματική απόκριση του ψηφιακού συστήματος με συνάρτηση μεταφοράς  $\frac{1}{z^2 - 0.3z + 0.5}$ .

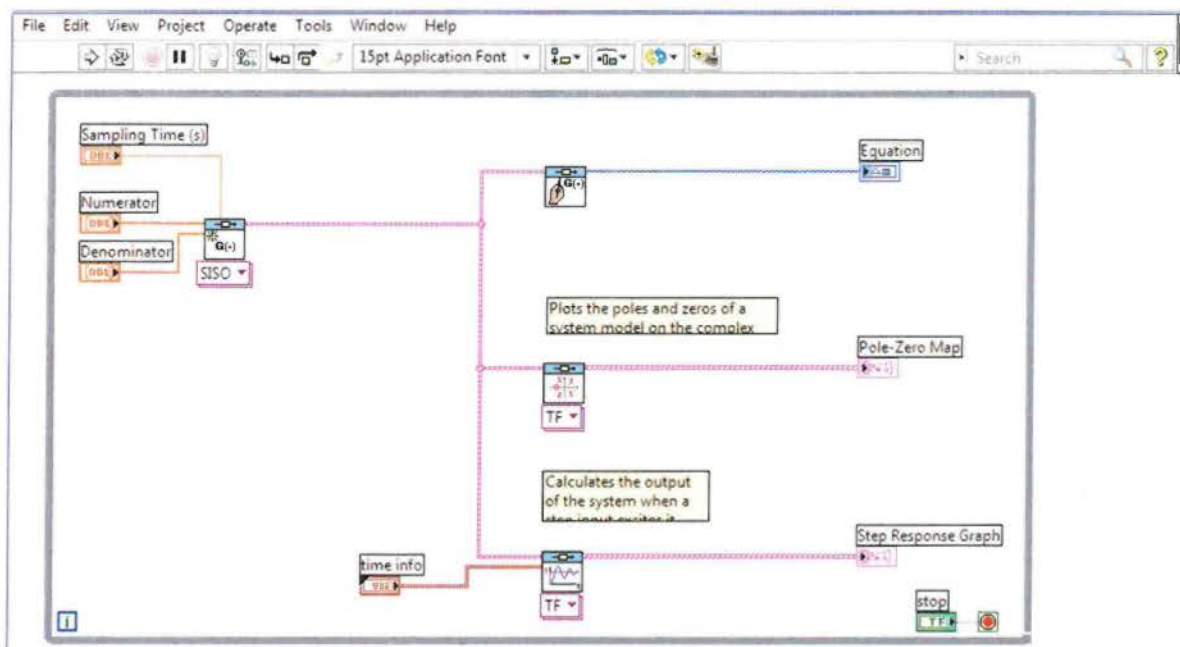
Ανοίξτε το αρχείο **digcon\_step-response.vi**.

Το front panel στο οποίο απεικονίζεται η βηματική απόκριση θα έχει την ακόλουθη μορφή



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

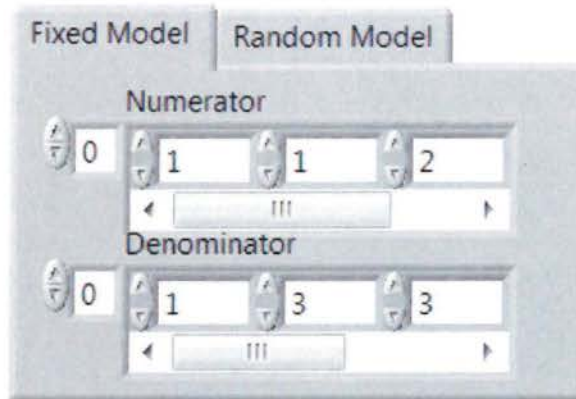
Το block diagram του παραδείγματος θα είναι το παρακάτω



Στο ακόλουθο παράδειγμα θα μετατρέψουμε το αναλογικό σύστημα σε ψηφιακό χρησιμοποιώντας διάφορες μεθόδους διακριτοποίησης

Έστω η συνάρτηση μεταφοράς αναλογικού συστήματος την οποία επιθυμούμε να διακριτοποιήσουμε με διάφορες μεθόδους (Zoh - Tustin- Prewarp - Forward - Backward - Z-Transform - Foh - Matched) και να σχεδιάσουμε την απόκριση συχνότητας, τη βηματική απόκριση, το διάγραμμα πόλων-μηδενικών και την κρουστική απόκριση του προκύπτοντος ψηφιακού συστήματος συγκρίνοντας το με το αρχικό αναλογικό σύστημα του οποίου ο αριθμητής και παρονομαστής δίνονται παρακάτω.

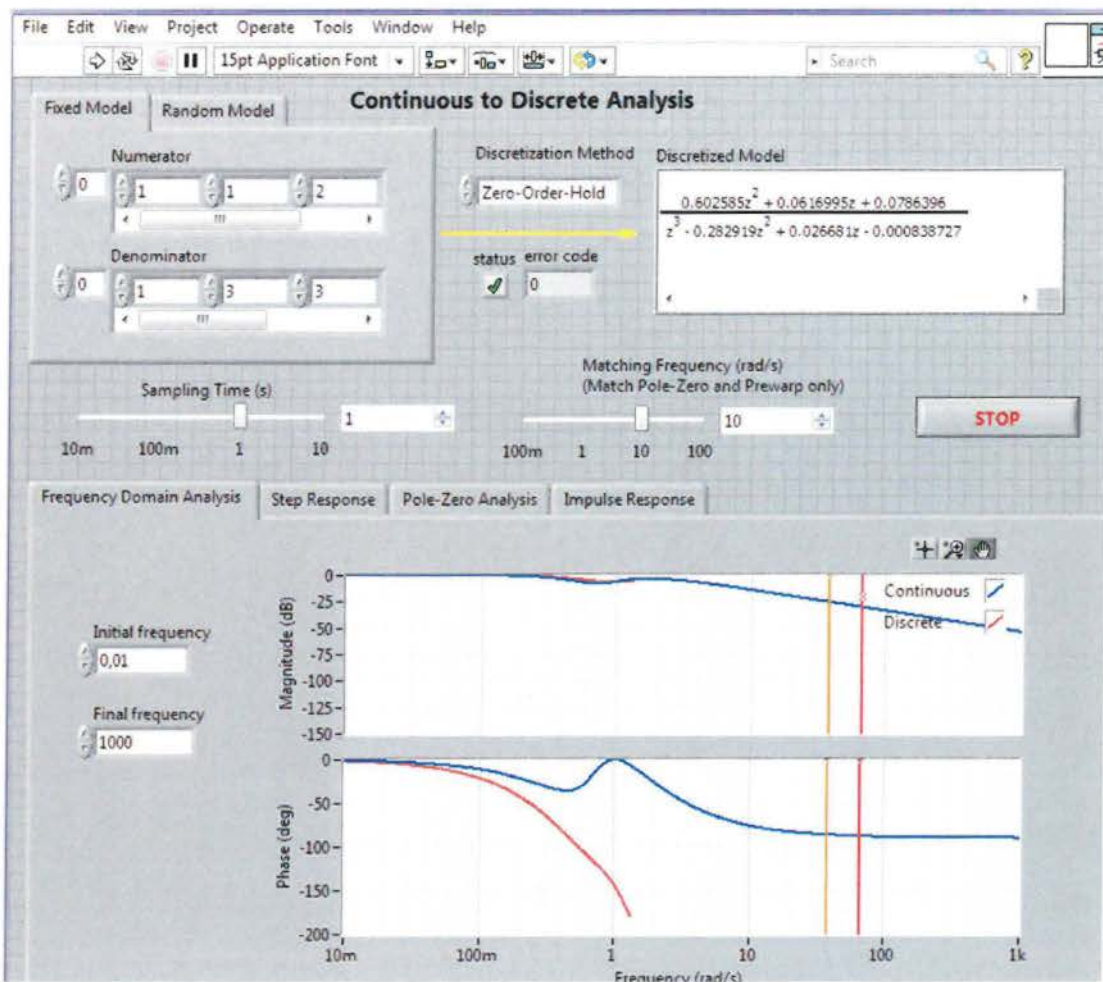
**ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.**



Ανοίξτε το **Continuous to Discrete Conversion.VI** και τρέξτε τις προσομοιώσεις .

Παρακάτω απεικονίζεται το **front panel** με τα στοιχεία του αναλογικού συστήματος και η διακριτή απόκριση συχνότητας με τη μέθοδο **zoh**.

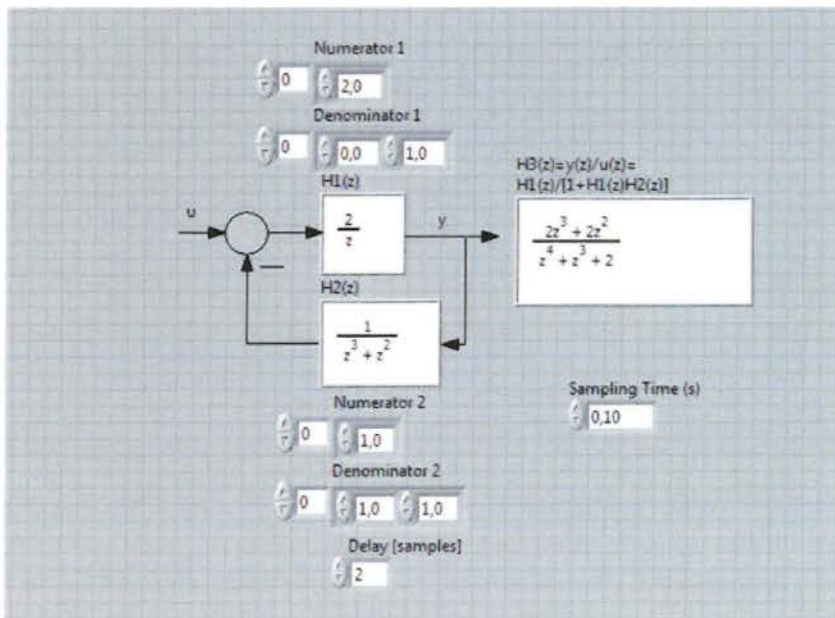
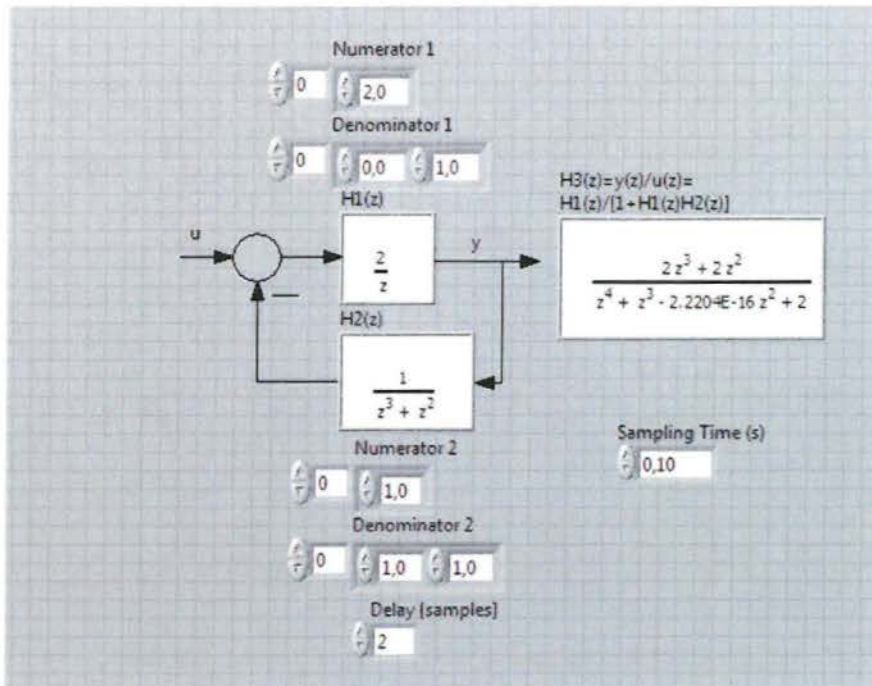
## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



Στο επόμενο παράδειγμα με τη βοήθεια του `feedback_connection_discrete.vi` δύο συστήματα διακριτού χρόνου συνδέονται σε ανατροφοδότηση και προκύπτει η συνολική συνάρτηση μεταφοράς κλειστού βρόχου.

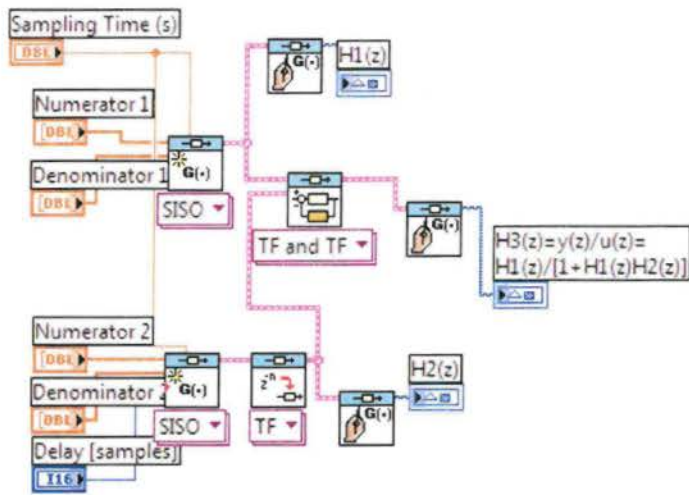


**ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.**



Στην εικόνα που ακολουθεί απεικονίζεται το block diagram του παραδείγματος.

ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.



## ΔΗΜΙΟΥΡΓΙΑ ΕΓΧΕΙΡΙΔΙΟΥ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ LABVIEW ΜΕ ΕΜΦΑΣΗ ΣΤΑ Σ.Α.Ε.

### Ελληνική Βιβλιογραφία

- LabView για μηχανικούς, Κωνσταντίνος Ιπ. Καλοβρέκτης , Τζιόλα, 2007
- LabVIEW στην τεχνική εκπαίδευση , Κωνσταντίνος Καλοβρέκτης, Αντώνιος Γκοτσίνας, Conceptum , 2009
- LabVIEW για μηχανικούς: Συστήματα συλλογής δεδομένων, Καλοβρέκτης, Κωνσταντίνος, Τζιόλα, 2014
- Συστήματα αυτόματου ελέγχου Λυμένες Ασκήσεις, Αναστασία Βελώνη Παπασωτηρίου, 1997
- Συστήματα αυτόματου ελέγχου Ανάλυση και προσομοίωση, Αναστασία Βελώνη, Τζιόλα, 2011

### Ξένη Βιβλιογραφία

- LabVIEW Graphical Programming, Gary W. Johnson, Richard Jennings, McGraw-Hill Education - Europe , 2006
- Labview Digital Signal Processing and Digital Communications, McGraw-Hill Education - Europe. 2005
- Labview Introduction to Data Acquisition, Robert H. King, McGraw-Hill Education - Europe, 2008
- Hands-on Introduction to Labview for Scientists and Engineers, John Essick, Oxford University Press Inc, 2012
- Labview Advanced Programming Techniques, Rick Bitter, Taqi Mohiuddin, Matt Nawrocki, Taylor & Francis Inc, 2006
- Labview for Engineers, Ronald W. Larsen, Pearson Education (US), 2010
- Image Acquisition and Processing with LabVIEW, Christopher G. Relf, Taylor & Francis Inc, 2003
- Labview for Everyone Graphical Programming Made Easy and Fun, Jim Kring, Jeffrey Travis, Pearson Education (US), 2005

### Διευθύνσεις Διαδικτύου

- <http://www.ni.com>
- <http://dsae.teipir.gr>