



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

Έλεγχος οργάνων με χρήση του LabView

Instruments controlling using LabView

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΩΝ

Παναγιώτου Λεμονιά – Μαρία

Χατζηβασιλείου Ελένη

Βόλος, 2016

Επιβλέποντες: Πλέσσας Φώτιος

Σταμούλης Γεώργιος

(Υπογραφή)

(Υπογραφή)

.....
Πλέσσας Φώτιος

.....
Σταμούλης Γεώργιος

.....
Παναγιώτου Λεμονιά-Μαρία

.....
Χατζηβασιλείου Ελένη

Προπτυχιακή φοιτήτρια
του τμήματος Η.Μ.Μ.Υ του Π.Θ

Προπτυχιακή φοιτήτρια
του τμήματος Η.Μ.Μ.Υ του Π.Θ

©2016 All rights reserved.

Copyright ©Παναγιώτου Λεμονιά-Μαρία, Χατζηβασιλείου Ελένη, 2016.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας εξ ολοκλήρου ή τμήματος αυτής για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρών μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό θα πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τους συγγραφείς και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών .

Περίληψη

Η παρούσα διπλωματική έχει σαν στόχο την επίτευξη του απομακρυσμένου ελέγχου κάποιων συγκεκριμένων οργάνων(την γεννήτρια σημάτων, τον παλμογράφο και τον αναλυτή φάσματος). Αρχικά γίνεται μια εισαγωγή σχετικά με την εξοικείωση του προγράμματος LabVIEW, καθώς απαιτήθηκε και γνώση δικτύωσης των οργάνων. Στην συνέχεια η δυνατότητα του απομακρυσμένου ελέγχου μας έδωσε την έννοια του εικονικού εργαστηρίου. Ακόμα κάνουμε μια περιγραφή για το γενικότερο πρωτόκολλο προγραμματισμού και διασύνδεσης οργάνων VISA. Έπειτα κάνουμε μία σύντομη παρουσίαση των οργάνων που θα απαρτίσουν το εικονικό εργαστήριο. Στα επόμενα κεφάλαια αναλύουμε τα νί όπου έχουμε υλοποιήσει. Πιο συγκεκριμένα ξεκινάμε τη διαδικασία σύνδεσης των προγραμμάτων οδήγησης όπου κάνουμε μια πλήρη περιγραφή για την κατασκευή των νί's καθώς δίνουμε και τα αποτελέσματα όπου θα έχουμε σε κάθε όργανο ξεχωριστά. Τέλος σας παραθέτουμε την διαδικασία σύνδεσης οργάνων μέσω Ethernet/LAN.

Ευχαριστίες

Στα πλαίσια αυτής της διπλωματικής εργασίας θέλουμε να ευχαριστήσουμε όλους όσους μας στήριξαν και μας καθοδήγησαν μέχρι και σήμερα καθ' όλη τη διάρκεια της φοίτησης μας στη σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

Αρχικά θα θέλαμε να εκφράσουμε τις θερμές μας ευχαριστίες στον επιβλέποντα καθηγητή και καθοδηγητή της παρούσας εργασίας **κ. Πλέσσα Φώτιο** για την απρόσκοπτη επικοινωνία και συμβολή των γνώσεών του αλλά και της υποστήριξης του καθ' όλη τη διάρκεια της διεκπεραίωσης της εργασίας. Ακόμη, θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας και τους φίλους μας, που μας στάθηκαν σε όλη τη διάρκεια των προπτυχιακών μου σπουδών μας.

ΠΕΡΙΕΧΟΜΕΝΑ

	Σελ.
1. Εισαγωγή στον έλεγχο οργάνων	6
2. Εισαγωγή στο LabVIEW	
2.1 Τι είναι το Lab View	8
2.2 Εισαγωγή στα VI	8
2.3 Έλεγχος οργάνων με την χρήση του LabVIEW	9
2.4 Δικτύωση με τη χρήση του Lab View	10
3. Το πρωτόκολλο VISA	
3.1 Τι είναι το VISA	11
3.2 Βασικοί στόχοι του πρωτοκόλλου VISA	11
3.3 Βασική ορολογία	11
4. Τα όργανα που πρόκειται να ελέγξουμε	
4.1 Η γεννήτρια σημάτων E4438C της Agilent	14
4.2 Ο παλμογράφος 54831B της Agilent	15
4.3 Ο αναλυτής φάσματος E4440A της Agilent	17
5. Το πρόγραμμα ελέγχου της γεννήτριας (E4438C)	
5.1 Λειτουργίες που θέλουμε να επιτελέσει η γεννήτρια(SignalGenerator)	
.....	18
5.2 E4438CInitialize.vi	19
5.3 Δημιουργία του RFout.vi	28
5.4 Δημιουργία του SetFreq.vi	30
5.5 Δημιουργία του SetLevel.vi	32
5.6 Δημιουργία του Close.vi	34
5.7 Δημιουργία του κορυφαίου Agilent_E4438C.vi.....	35
5.8 Λειτουργίες που θέλουμε να επιτελέσει η γεννήτρια(SignalGenerator) με την χρήση έτοιμων συναρτήσεων	38
5.9 Αποτελέσματα γεννήτριας(E4438C)	42
6. Το πρόγραμμα ελέγχου του παλμογράφου	
6.1 Λειτουργίες που θέλουμε να επιτελέσει ο παλμογράφος(Oscilloscope)	
.....	44
6.2 Ανάλυση των επιμέρους VI	46
7. Το πρόγραμμα ελέγχου του αναλυτή φάσματος	
7.1 Λειτουργίες που θέλουμε να επιτελέσει ο αναλυτής φάσματος (SpectrumAnalyzer)	56
7.2 Ανάλυση των επιμέρους VI	57
7.3 Αποτελέσματα του αναλυτή φάσματος(E4440A)	63
ΠΑΡΑΡΤΗΜΑ	68
ΒΙΒΛΙΟΓΡΑΦΙΑ	70

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή στον έλεγχο οργάνων

Η ανάγκη αποθήκευσης, επαλήθευσης ή και επεξεργασίας των μετρήσεων είναι αρκετά προφανής. Αρχικά, για την κάλυψη αυτής, θεωρήθηκε σωστός ο εξοπλισμός των μετρητικών οργάνων με κάποια μνήμη, στην οποία ο χρήστης είχε τη δυνατότητα να αποθηκεύσει τις μετρήσεις του. Το ήδη υψηλό όμως κόστος των οργάνων, σε συνδυασμό με το κόστος υποστήριξης αυτών των επιπρόσθετων λειτουργιών, οδήγησε γρήγορα στην εγκατάλειψη αυτού του σχεδιασμού. Έτσι, σιγά-σιγά με την εξέλιξη των υπολογιστών, αναζητήθηκαν τρόποι αποθήκευσης των μετρήσεων των οργάνων σε αυτούς. Σήμερα, υπάρχει μία σειρά από πρωτόκολλα και καλώδια διασύνδεσης του υπολογιστή με τα όργανα, με σκοπό την αμφίδρομη μεταφορά δεδομένων, όπως SERIAL, GPIB, VXI κτλ.

Μία άλλη ανάγκη που παρουσιάστηκε πρόσφατα, με την εξέλιξη και γενίκευση της χρήσης των δικτύων, και ιδιαίτερα των ασύρματων δικτύων, αποτελεί ο έλεγχος του οργάνου από μακριά. Κάτι το οποίο βρίσκει πολλές εφαρμογές στον τομέα της τηλεμετρίας, ιδιαίτερα για δυσπρόσιτες περιοχές, αλλά και στη διασύνδεση των οργάνων για εκπαιδευτικούς σκοπούς. Επιπλέον, μπορεί να μειώσει σημαντικά το ποσοστό των οργάνων που καταστρέφονται μέσω μιας σειράς αυτοματοποιημένων ελέγχων τήρησης των κανονισμών ασφαλείας από τον υπολογιστή.

Η προσπάθεια που έγινε σε αυτή τη διπλωματική ήταν η επίτευξη του απομακρυσμένου ελέγχου κάποιων συγκεκριμένων οργάνων (παλμογράφος, γεννήτρια συχνοτήτων και αναλυτής φάσματος) παρέχοντας σημαντική πληροφορία σε πραγματικό χρόνο. Για την εκπόνηση της διπλωματικής απαιτήθηκε γνώση δικτύωσης των οργάνων, κατασκευή προγραμμάτων ελέγχου, εξοικείωση με το πρόγραμμα LabVIEW και γνώση δικτύωσης υπολογιστών.

Στα κεφάλαια που ακολουθούν γίνεται λεπτομερής αναφορά στα παραπάνω και συγκεκριμένα στο πρώτο κεφάλαιο γίνεται μία αναφορά στο πακέτο LabVIEW και στο τρόπο με τον οποίο δουλεύει. Στο δεύτερο κεφάλαιο περιγράφεται το γενικότερο πρωτόκολλο προγραμματισμού και διασύνδεσης οργάνων VISA ενώ στο τρίτο κεφάλαιο κάνουμε μία σύντομη παρουσίαση των οργάνων που θα απαρτίσουν το εικονικό εργαστήριο το οποίο απαρτίζεται από την γεννήτρια σημάτων E4438C της Agilent, τον παλμογράφο 54831B της Agilent και τον αναλυτή φάσματος E4440A της Agilent. Και στα τελευταία κεφάλαια αναλύουμε τα νί(τα εικονικά όργανα) όπου έχουμε υλοποιήσει. Πιο αναλυτικά στο πέμπτο κεφάλαιο ξεκινάμε τη διαδικασία σχεδίασης των προγραμμάτων οδήγησης και την συγχώνευσή τους σε ένα κοινό πρόγραμμα το οποίο θα αποτελεί και το πρόγραμμα διεπαφής του απομακρυσμένου χρήστη(την γεννήτρια σημάτων E4438C της Agilent) καθώς και για τα επόμενα κεφάλαια όπου αναλύουμε τα εικονικά όργανα για τον παλμογράφο 54831B της Agilent και τον αναλυτή φάσματος E4440A της Agilent.

Εισαγωγή στο LabVIEW

Το LabVIEW είναι μια γλώσσα προγραμματισμού για τη συλλογή δεδομένων, την ανάλυση δεδομένων, την προσομοίωση και τον έλεγχο οργάνων και μετρήσεων μέσω υπολογιστή. Στηρίζεται στο γραφικό προγραμματισμό μέσω αντικειμένων και αποτελεί ένα καλό παράδειγμα του αντικειμενοστραφή προγραμματισμού (object oriented programming). Στο γραφικό περιβάλλον του LabVIEW ο προγραμματιστής δεν χειρίζεται κώδικα, αλλά γραφικά αντικείμενα, όπως κουμπιά, ενδείκτες, οθόνες ή τετραγωνίδια που παριστάνουν συναρτήσεις ή εκτελούν συγκεκριμένες λειτουργίες με τη μορφή υπορουτινών. Αυτά τα εικονίδια έχουν εισόδους και εξόδους και επιδέχονται προγραμματισμό των ιδιοτήτων τους.

Το όνομα LabVIEW είναι το ακρωνύμιο των λέξεων **L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench (Σχεδιαστήριο για την Κατασκευή Εργαστηριακών Εικονικών Οργάνων) και αναπτύχθηκε κατά το τέλος της δεκαετίας του 80 από την εταιρία National Instruments.

Προγραμματίζοντας με τα αντικείμενα που μας δίνει το περιβάλλον του LabVIEW δημιουργούμε τα λεγόμενα εικονικά όργανα (Virtual Instruments ή απλώς VIs). Η γραφική γλώσσα που χρησιμοποιεί το LabVIEW για τον προγραμματισμό και τη δημιουργία εικονικών οργάνων ονομάζεται γλώσσα G. Είναι πάντως δυνατό να χρησιμοποιήσει κανείς το LabVIEW σε απλό επίπεδο, χρησιμοποιώντας έτοιμα εικονικά όργανα, χωρίς να μπαίνει σε λεπτομέρειες με τη γλώσσα G.

Ένα εικονικό όργανο μπορεί να προσομοιώνει απλώς μια λειτουργία και να την παρουσιάζει στην οθόνη του υπολογιστή. Όμως, είναι δυνατό να συνδέεται με τις θύρες εισόδου/εξόδου του υπολογιστή ή με επιπρόσθετες κάρτες επέκτασης, προκειμένου να κάνει πραγματική εισαγωγή ή εξαγωγή δεδομένων. Στην περίπτωση αυτή ο υπολογιστής με τη βοήθεια των εισόδων και των εξόδων μετατρέπεται σε ένα ισχυρό εργαλείο μετρήσεων, με πολλές δυνατότητες επεξεργασίας δεδομένων.

Το LabVIEW διαθέτει έναν αριθμό από έτοιμα VIs και ορισμένα εικονίδια συναρτήσεων που επιτρέπουν την επικοινωνία με όλα τα γνωστά πρωτόκολλα μετάδοσης δεδομένων. Έτσι, υπάρχουν έτοιμες λειτουργίες που επιτρέπουν τη συλλογή και μετάδοση δεδομένων μέσω της σειριακής θύρας του υπολογιστή, καθώς και μέσω της παράλληλης θύρας.

Επίσης, υπάρχουν λειτουργίες για την ανταλλαγή δεδομένων με την κάρτα ήχου, καθώς και με κάρτες επέκτασης που χρησιμοποιούν το πρωτόκολλο GPIB ή IEEE 488. Εξάλλου, όλες οι κάρτες συλλογής δεδομένων της εταιρίας National Instruments είναι συμβατές με το LabVIEW με τη βοήθεια ειδικών οδηγών που ενσωματώνονται στο λογισμικό. Το ίδιο συμβαίνει και με σημαντικό αριθμό άλλων οργάνων, για τα οποία κυκλοφορούν οδηγοί συμβατοί με το LabVIEW.

Έτσι, μέσω των πρωτοκόλλων επικοινωνίας η πληροφορία που δημιουργείται στην οθόνη του υπολογιστή συνδέεται με πραγματικά όργανα, μέσω του λογισμικού. Όταν πατούμε ένα εικονικό κουμπί στην οθόνη, ενεργοποιείται ένας πραγματικός διακόπτης σε ένα εργαστηριακό όργανο. Σ'

αυτήν ακριβώς τη δυνατότητα, που επεκτείνει την απλή προσομοίωση ώστε να γίνεται εφικτός ο έλεγχος αληθινών οργάνων, βρίσκεται και η δύναμη του LabVIEW ως λογισμικού μετρήσεων και ελέγχου.

2.1 Τι είναι το LabVIEW

Το LabVIEW είναι ένα περιβάλλον ανάπτυξης προγραμμάτων που διαφέρει από τα άλλα. Ιδιαίτερο χαρακτηριστικό του είναι ο τρόπος με τον οποίο σχεδιάζεται η κάθε εφαρμογή. Το LabVIEW χρησιμοποιεί τη γλώσσα προγραμματισμού G. Με το πρόγραμμα αυτό αντί να γράφουμε σειριακά με τη μορφή εντολών το σώμα του προγράμματος, χρησιμοποιούμε block diagrams για την απεικόνιση απλών εντολών και υποπρογραμμάτων. Σε αντίθεση με τις παραδοσιακές, βασιζόμενες σε κείμενο γλώσσες προγραμματισμού, όπου οι εντολές καθορίζουν την εκτέλεση του προγράμματος, το LabVIEW χρησιμοποιεί προγραμματισμό ροής δεδομένων (dataflow programming) όπου η ροή των δεδομένων καθορίζει την εκτέλεση. Ένα πρόγραμμα σε LabVIEW ονομάζεται VI (Virtual Instrument) επειδή μιμείται την εμφάνιση και λειτουργία ενός πραγματικού οργάνου. Το LabVIEW είναι ένα εργαλείο γενικού προγραμματισμού αλλά περιλαμβάνει πολλές βιβλιοθήκες συναρτήσεων και εργαλεία ειδικά για ανάκτηση δεδομένων και για έλεγχο οργάνων.

Στο LabVIEW το User Interface κατασκευάζεται με τη χρήση ενός συνόλου εργαλείων και αντικειμένων, το οποίο είναι γνωστό ως **Front Panel**. Στη συνέχεια, προστίθεται ο κώδικας με τη χρήση γραφικών αναπαραστάσεων των συναρτήσεων ώστε να ελεγχθούν τα αντικείμενα του Front Panel. Το **Block Diagram** περιέχει αυτόν τον κώδικα. Γενικά, το Block Diagram θα μπορούσε να θεωρηθεί ως ένα είδος διαγράμματος ροής. Το LabVIEW συνεργάζεται πλήρως με υλικό (Hardware) με σκοπό την ανταλλαγή δεδομένων μεταξύ οργάνου και υπολογιστή και έχει επίσης την κατάλληλη υποδομή για υποστήριξη δικτυακών εφαρμογών μέσω του LabVIEW Web Server.

2.2 Εισαγωγή στα VI

Όπως αναφέραμε παραπάνω τα προγράμματα του LabVIEW ονομάζονται VI's – Virtual Instruments - ακριβώς γιατί η εμφάνισή τους και η λειτουργία τους μιμείται το πραγματικό όργανο. Σε αυτή την παράγραφο θα γίνει μία μικρή αναφορά στα επιμέρους τμήματα από τα οποία αποτελείται ένα VI και στον τρόπο με τον οποίο μπορεί να συνταχθεί ένα πρόγραμμα με το LabVIEW.

Ένα VI αποτελείται:

1. **Front Panel:** Το οποίο αποτελεί και το μέσω επικοινωνίας με το χρήστη (user interface). Το Front Panel έχει τα δικά του εργαλεία. Κάνοντας δεξί κλικ έχοντας ανοιχτό το παράθυρο του FrontPanel εμφανίζεται το παράθυρο των εργαλείων που μπορούν να τοποθετηθούν στο Front Panel. Βλέπουμε ότι η παλέτα των Controls

αποτελείται από αριθμητικά (numeric) controls, boolean controls, πίνακες και γενικότερα ότι μπορεί να έχει μια συνηθισμένη γλώσσα προγραμματισμού.

2. **Block Diagram:** Το οποίο περιέχει τον γραφικό, πηγαίο κώδικα που καθορίζει τον τρόπο με τον οποίο λειτουργεί το VI. Κάνοντας δεξιά κλικ έχοντας ανοιχτό το παράθυρο του Block Diagram εμφανίζεται αυτή τη φορά το παράθυρο των συναρτήσεων από το οποίο μπορούμε να επιλέξουμε οποιαδήποτε συνάρτηση για την επεξεργασία των controls του Front Panel. Έτσι για την επεξεργασία των αριθμητικών δεδομένων επιλέγουμε τις συναρτήσεις numeric οι οποίες δεν είναι τίποτε άλλο παρά ένα πακέτο αριθμητικών πράξεων όπως πρόσθεση, πολλαπλασιασμός και πολλά άλλα. Αντίστοιχα για την επεξεργασία των boolean controls επιλέγουμε τις boolean functions οι οποίες έχουν σαν εισόδους boolean μεταβλητές και δίνουν έξοδο boolean. Υπάρχουν και συναρτήσεις σύγκρισης (comparison function) οι οποίες μπορούν να πάρουν σαν είσοδο αριθμητικά δεδομένα και να δώσουν στην έξοδο δεδομένα τύπου boolean.
3. **Τετράγωνο Συνδέσεων και Εικονιδίων:** Σε κάθε VI αντιστοιχίζεται ένα εικονίδιο που ενσωματώνει ένα πίνακα συνδέσεων (connector pane). Όταν το εικονίδιο τοποθετείται σε ένα Block Diagram το VI λειτουργεί όπως οι υπο-ρουτίνες άλλων γλωσσών. Η πρόσβαση στις παραμέτρους εισόδου και εξόδου γίνεται από τις συνδέσεις του connector pane. Δημιουργείται έτσι μία ιεραρχική δομή των VI's. Κάθε VI αποτελείται από άλλα VI's που σ' αυτή την περίπτωση αναφέρονται ως subVI's. Τα subVI's χρησιμοποιούνται όπως και οι υπόλοιπες συναρτήσεις στο Block Diagram συνδέοντας τις εισόδους και εξόδους αναλόγως. Αυτό διευκολύνει το σχεδιασμό της εφαρμογής σε πολλά επίπεδα από το πιο βασικό ως το ανώτατο επίπεδο της εφαρμογής. Από την Functions Palette επιλέγοντας "Select a VI" εμφανίζεται ένα παράθυρο διαλόγου μέσω του οποίου μπορούμε να επιλέξουμε ένα VI από το σημείο ή το φάκελο που το έχουμε αποθηκεύσει και να το τοποθετήσουμε στο Front Panel του VI στο οποίο δουλεύουμε τη συγκεκριμένη στιγμή και να το θέσουμε ως sub-VI. Το γεγονός αυτό μάλιστα μπορεί να φανεί και καλύτερα αν από τη γραμμή εργαλείων του Front Panel επιλέξουμε "View" και στη συνέχεια "Show VI Hierarchy", τότε εμφανίζεται ένα παράθυρο που μας παρουσιάζει την ιεραρχική δομή του προγράμματός μας δείχνοντας τα sub-VI's που έχουμε ορίσει.

2.3 Έλεγχος οργάνων με τη χρήση του LabVIEW

Το Lab View μπορεί να χρησιμοποιηθεί για την επικοινωνία με όργανα ελεγχόμενα από GPIB, VXI, RS-232 και διάφορα άλλα πρότυπα. Η επικοινωνία με τα όργανα γίνεται με τη βοήθεια προγραμμάτων οδήγησης οργάνων (drivers), δηλαδή βιβλιοθηκών από VI που ελέγχουν προγραμματιζόμενα όργανα. Με τον τρόπο αυτό απλοποιείται η απαιτούμενη για τον έλεγχο του οργάνου διαδικασία, αφού δεν χρειάζεται να μάθει κανείς το πρωτόκολλο του κάθε οργάνου. Όταν κάποιος εκτελεί το VI ενός οργάνου ουσιαστικά τρέχει ένα αυτόματα εγκατεστημένο driver, ο οποίος στέλνει αυτόματα τις κατάλληλες για το όργανο εντολές. Τέτοια προγράμματα

υπάρχουν για τα περισσότερα καινούρια όργανα όχι όμως και για τα παλαιότερα. Στα πλαίσια της διπλωματικής αυτής και για την κάλυψη των αναγκών της αναπτύχθηκαν δύο νέοι drivers των οποίων η δομή, η κατασκευή και λειτουργία θα εξηγηθεί αναλυτικά σε επόμενο κεφάλαιο.

2.4 Δικτύωση με τη χρήση του LabVIEW

Το LabVIEW μπορεί να δικτυωθεί με άλλες διαδικασίες, που περιλαμβάνουν ακόμα και εκτέλεση μέσα από άλλες εφαρμογές ή από φυσικά απομακρυσμένους υπολογιστές. Ανάμεσα στους σκοπούς που μπορούν να εξυπηρετηθούν μέσα από το πρόγραμμα αυτό, είναι:

1. Ανταλλαγή δεδομένων ανάμεσα σε VI's που εκτελούνται σε ένα δίκτυο με τη χρήση της τεχνολογίας NI Data Socket.
2. Δυνατότητα μεταφοράς των εικόνων του Front Panel καθώς και της διάθεσης του προγράμματος στο Internet με χρήση HTML.
3. Επικοινωνία των προγραμμάτων με χρήση χαμηλού επιπέδου πρωτοκόλλων όπως TCP, UDP κ.τ.λ.

Από όλα τα παραπάνω γίνεται φανερό ότι το LabVIEW είναι ένα πανίσχυρο προγραμματιστικό πακέτο το οποίο είναι κατάλληλα εφοδιασμένο για να ανταποκριθεί στις σύγχρονες και συνεχώς αυξανόμενες απαιτήσεις. Ιδιαίτερα στον τομέα των εργαστηριακών μετρήσεων, ο έλεγχος των οργάνων γίνεται με περισσότερη ευκολία και άνεση από τη μεριά του προγραμματιστή, ενώ από τη μεριά του χρήστη το περιβάλλον της εκάστοτε εφαρμογής είναι πολύ πιο φιλικό.

Το πρωτόκολλο VISA (Vxl plug and play System Alliance)

3.1 Τι είναι το VISA

Το VISA είναι ένα πρότυπο API(Application Programming Interface) εισόδου και εξόδου για προγραμματισμό οργάνων. Μπορεί να ελέγξει VXI,GPIB,PXI ή σειριακά όργανα, κάνοντας τις κατάλληλες κλήσεις του driver ανάλογα με τον τύπο οργάνου που χρησιμοποιείται.

Το πρωτόκολλα GPIB, serial καθώς και κάποια VXI όργανα χρησιμοποιούν επικοινωνία που βασίζεται σε μηνύματα. Τα προγραμματιζόμενα μέσω μηνυμάτων όργανα προγραμματίζονται μέσω υψηλού επιπέδου Strings εντολών και θέτει τα κατάλληλα bits των καταχωρητών να κάνουν ορισμένες λειτουργίες. Τα παραπάνω όργανα είναι εύκολα να προγραμματιστούν. Για να γίνει πιο εύκολη η διαδικασία προγραμματισμού τους, το πρωτόκολλο SCPI θέτει κάποια πρότυπα στο είδος των εντολών που χρησιμοποιούνται στον προγραμματισμό οποιουδήποτε οργάνου. Έτσι αντί να χρειάζεται να μάθει κάποιος διαφορετικά μηνύματα εντολών για κάθε είδος οργάνου, χρειάζεται να μάθει απλά ένα σύνολο εντολών που είναι οι VISA Write, VISA Assert Trigger, VISA Clear, VISA Read STB.

3.2 Βασικοί στόχοι του πρωτοκόλλου VISA

Ο βασικός στόχος του **VXI plug and play Systems Alliance** είναι να διευκολύνει τη χρήση της VXI τεχνολογίας. Οι προγραμματιστές χρειάζονται μία αρχιτεκτονική που να δίνει τις δυνατότητες της συσκευής και όχι του αγωγού διεπαφής. Επιπλέον πρέπει τα προγράμματα να είναι συμβατά με τις δυνατότητες των συσκευών και των αγωγών διεπαφής τους. Κατανοώντας τους παραπάνω στόχους, προκύπτει ένα πρωτόκολλο που έχει πολύ μικρότερο αριθμό συναρτήσεων που πρέπει να μάθει ο χρήστης.

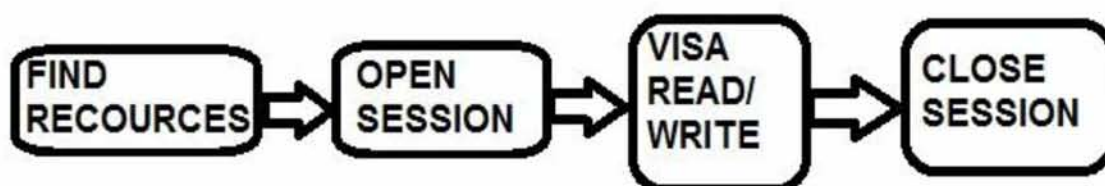
Επιπλέον τα περισσότερα όργανα εξάγουν ένα σύνολο εντολών στις οποίες μπορούν να ανταποκριθούν. Επειδή το όργανο πρέπει να είναι ευέλικτο, αυτές οι εντολές είναι αρκετά «πρωτόγονες» και απαιτείται ένας μεγάλος αριθμός από αυτές, για να μπορέσει η συσκευή να εκτελέσει ακόμα και τις πιο τετριμμένες λειτουργίες. Αυτό έχει ως συνέπεια να απαιτείται μεγάλος όγκος προγραμμάτων για τον έλεγχο των λειτουργιών των συσκευών.

3.3 Βασική ορολογία

Οι τυπικές συσκευές έχουν τη δυνατότητα να στέλνουν και να λαμβάνουν μηνύματα, να απαντούν σε ερωτήσεις των καταχωρητών τους, να ζητούν την εκτέλεση κάποιας υπηρεσίας, να κάνουν επανεκκίνηση κτλ. Συνεπώς για να ανταποκρίνεται το πρωτόκολλο VISA στις βασικές απαιτήσεις δημιουργίας του, πρέπει να επικεντρώνεται στις προαναφερθείσες βασικές λειτουργίες. Αυτή η ικανότητα συγκεντρώνεται σε ένα resource.

Ένα resource είναι απλά μία πλήρης περιγραφή μιας συγκεκριμένης ιδιότητας της συσκευής. Για να αποκτηθεί για παράδειγμα η ικανότητα εγγραφής στη συσκευή, χρειάζεται μία συνάρτηση που μπορεί να χρησιμοποιηθεί για να αποσταλούν τα μηνύματα – στην vi Write. Επίσης πρέπει να καθοριστούν κάποιες λεπτομέρειες, όπως με ποιον χαρακτήρα τερματίζεται η εντολή και πόσο χρόνο θα προσπαθεί η συνάρτηση να επικοινωνήσει με το όργανο πριν λήξει η σύνδεση. Προφανώς στα παραπάνω αναγνωρίζει κάποιος βασικές αρχές του αντικειμενοστραφή προγραμματισμού.

Συμπερασματικά λοιπόν μπορούμε να πούμε ότι η επικοινωνία με το όργανο γίνεται σύμφωνα με ένα γενικό διάγραμμα ροής που ισχύει για όλες τις περιπτώσεις. Πρώτα δημιουργείται μια software σύνδεση με το όργανο, ακολουθεί η εγγραφή και η ανάγνωση δεδομένων στο και από τον hardware δίαυλο και τέλος κλείνει η σύνδεση. Το μοτίβο αυτό συστηματοποιεί τον έλεγχο οποιουδήποτε οργάνου και μόνο η σύνταξη των εντολών πρέπει να λαμβάνεται υπόψη κάθε φορά. Στο σχήμα 3.3.1 απεικονίζεται το γενικό διάγραμμα επικοινωνίας και εξηγούνται τα διαδοχικά βήματα.



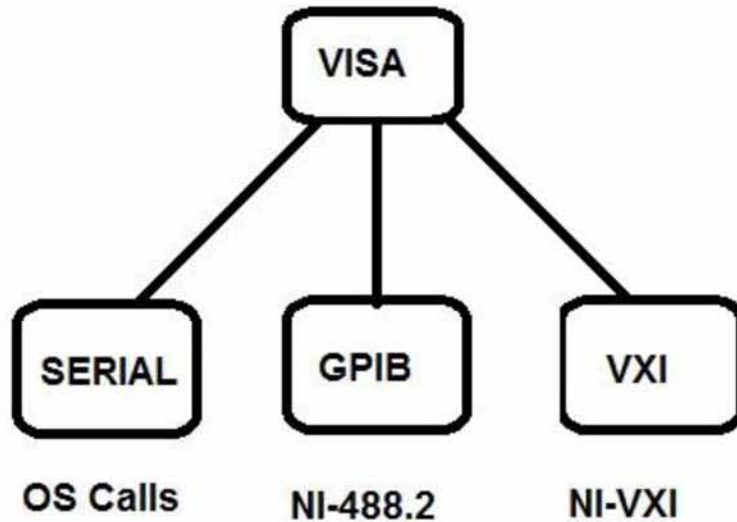
Σχήμα 3.3.1 Γενικό Διάγραμμα VISA.

1. FIND RECOURCES: Σε αυτό το στάδιο αναγνωρίζονται οι συσκευές που είναι διαθέσιμες και ο τύπος του hardware διαύλου που συνδέει καθεμιά. Σε κάθε συνδεδεμένη συσκευή αποδίδεται ένα resource ,μια προγραμματική οντότητα με μια λογική διεύθυνση. Το υπόλοιπο πρόγραμμα επικοινωνεί με το resource που αντιστοιχεί σε κάθε όργανο. Όλα τα ζητήματα hardware ρυθμίζονται σε αυτό το στάδιο.

2. OPENSESSION: Χρησιμοποιώντας την παραπάνω λογική διεύθυνση δημιουργείται μια ενεργή software σύνδεση με το resource, που αντιστοιχεί στην συσκευή. Δημιουργείται μια μεταβλητή αναφοράς που λέγεται VISA Session και περιλαμβάνει όλα τα στοιχεία για τη σύνδεση. Οι διαδικασίες εισόδου – εξόδου χρησιμοποιούν αποκλειστικά αυτή την μεταβλητή.

3. PERFORM I/O: Η επικοινωνία με τη συσκευή γίνεται με εγγραφή και ανάγνωση συμβολοσειρών (strings).Οι εντολές που υποστηρίζονται και η σύνταξή τους αναφέρονται στο εγχειρίδιο προγραμματισμού του οργάνου (manual).

4. CLOSE SESSION: Κλείνει η σύνδεση με τη συσκευή και απελευθερώνονται οι πόροι του συστήματος.



Σχήμα 3.3.2 Δομή VISA

Το σχήμα 3.3.2 φανερώνει τη δομή του πρωτοκόλλου VISA. Από το σχήμα αυτό γίνεται φανερό ότι το VISA είναι ένα πιο γενικό πρωτόκολλο μέσα στο οποίο εντάσσεται και το πρωτόκολλο GPIB. Αυτό θα γίνει καλύτερα κατανοητό από τα προγράμματα του LabVIEW.

Σε όποια από αυτά χρησιμοποιήσαμε το VISA ήμασταν αναγκασμένοι να κάνουμε και ένα Initialize VI το οποίο έκανε το handshaking με το όργανο και ουσιαστικά το κατέτασσε σε κάποια από τις περιπτώσεις του σχήματος. (Στην περίπτωση μας GPIB). Αντίθετα χρησιμοποιώντας το πρωτόκολλο GPIB δεν χρειαζόταν κάτι αντίστοιχο.

Τα όργανα που πρόκειται να ελέγξουμε

4.1 Η γεννήτρια σημάτων E4438C της Agilent

Οι γεννήτριες σημάτων (function ή signal generator) είναι όργανα που παρέχουν σήμα (πληροφορία) συνήθως με την μορφή τάσης ή ρεύματος στο κύκλωμα. Έχουν την δυνατότητα τα παράγουν σήματα με ποικίλα χαρακτηριστικά τα οποία ελέγχονται από τον χρήστη. Τα σήματα εξέρχονται από την γεννήτρια μέσω ομοαξονικού (θωρακισμένου - shielded) καλωδίου το οποίο συνδέεται στην γεννήτρια μέσω ενός BNC βύσματος. Το ομοαξονικό καλώδιο με βύσμα BNC είναι ουσιαστικά δύο καλώδια και είναι κατασκευασμένο ως εξής. Το ένα καλώδιο, το οποίο μεταφέρει το σήμα, είναι στο εσωτερικό και καταλήγει σε ένα pin στο βύσμα BNC. Το δεύτερο καλώδιο αντιστοιχεί στη γείωση, περιβάλλει ομοαξονικά το εσωτερικό καλώδιο και όταν συνδεθεί το BNC βύσμα στη γεννήτρια, καταλήγει στο περίβλημα της γεννήτριας. Στην άλλη άκρη, το καλώδιο καταλήγει σε δύο ξεχωριστά καλώδια εκ των οποίων η κόκκινη απόληξη αντιστοιχεί στο σήμα και η μαύρη στην γείωση. Θωρακισμένο καλώδιο με βύσμα BNC στο ένα άκρο και διπλή απόληξη στο άλλο.

Τα σήματα μπορεί να είναι διαφόρων τύπων, τα πλέον συνηθισμένα είναι:

1. αρμονικά (ημιτονοειδή)
2. τετραγωνικά
3. τριγωνικά

Οι παράμετροι που μπορούν να ρυθμιστούν είναι

1. Η συχνότητα (frequency)
2. Το πλάτος (V_{pp} , V_{rms} , V_p). V_{pp} είναι η διαφορά της μέγιστης από την ελάχιστη τιμή τάσης ενώ V_p είναι το πλάτος του σήματος, πχ. για ένα αρμονικό σήμα το V_{pp} είναι διπλάσιο του V_p
3. Το επίπεδο συνεχούς τάσης (DC offset)
4. Στην περίπτωση των τετραγωνικών σημάτων ο κύκλος εργασίας (% duty cycle). Ως κύκλος εργασίας, ορίζεται το ποσοστό του χρόνου, μέσα σε μια περίοδο, κατά τον οποίο το σήμα έχει την μέγιστη τιμή.
5. Στην περίπτωση των τριγωνικών παλμών η ασυμμετρία (asymmetry). Ως ασυμμετρία ορίζεται το ποσοστό του χρόνου, μέσα σε μια περίοδο, κατά τον οποίο ο παλμός αυξάνει.

Η γεννήτρια που χρησιμοποιήθηκε είναι της εταιρίας Agilent μοντέλο E4438C οι οποίες μπορούν να δώσουν σήματα με συχνότητες έως 10/250kHz.



Σχήμα 4.1.1 - Πρόσωση της γεννήτριας σημάτων

4.2 Ο παλμογράφος 54831B της Agilent

Ο παλμογράφος είναι το πιο χρήσιμο όργανο μελέτης και μέτρησης της εναλλασσόμενης και συνεχούς τάσης και χρησιμοποιείται ευρύτατα σε πολλούς τομείς της επιστημονικής έρευνας και τεχνολογίας. Το όργανο αυτό παρέχει οπτική απεικόνιση πάσης φύσεως κυματομορφής συνεχούς και εναλλασσόμενης τάσης όπως επίσης μικρής διάρκειας χρόνου και διαφορές φάσης μεταξύ δύο ημιτονικών τάσεων.

Τα περισσότερα όργανα που μετρούν την τάση χρησιμοποιούν μηχανικά μέσα, όπως πηνία, δείκτες, κάτροπτα κ.λπ. Τα κινούμενα μηχανικά μέρη του μετρητή έχουν μεγάλη μηχανική αδράνεια και ενώ είναι αξιόπιστα όταν με τα όργανα αυτά μετρούν μία σταθερή στο χρόνο τάση, είναι αναξιόπιστα όταν, λόγω αδράνειας του μετρητή, η τάση μεταβάλλεται με μεγάλη συχνότητα.

Στον παλμογράφο τον ρόλο του δείκτη ή της βελόνας του αναλογικού μετρητή παίζει μία λεπτή δέσμη ηλεκτρονίων, η οποία εκτρέπεται από τη μετρούμενη τάση. Όπως γνωρίζουμε η δέσμη ηλεκτρονίων έχει αμελητέα αδράνεια και επομένως μπορεί να ακολουθήσει πολύ γρήγορες μεταβολές τάσης, που αντιστοιχούν σε μερικές εκατοντάδες MHz.

Ο παλμογράφος 54831 της Agilent μας προσφέρει την δυνατότητα παρακολούθησης τεσσάρων καναλιών. Η ψηφιακή του υπόσταση τον καθιστά ικανό για επικοινωνία με άλλα ψηφιακά όργανα και, το πιο σημαντικό, με τον σύγχρονο ηλεκτρονικό υπολογιστή. Η επικοινωνία αυτή πραγματοποιείται μέσω του δικτύου LAN(αφού ο χρήστης συνδεθεί πρώτα μέσω VPN στο Πανεπιστήμιο Θεσσαλίας). Το Bandwidth του παλμογράφου αυτού φτάνει μέχρι και τα 600MHz. Ως σύγχρονος παλμογράφος προσφέρει όλες εκείνες τις μετρητικές δυνατότητες που έχουμε γνωρίσει στους συμβατικούς παλμογράφους και επιπλέον, λόγω της ψηφιακής του φύσης(digital),είναι σε

θέση να προσαρμόσει το περιβάλλον της διεπαφής (user interface) στις επιθυμίες του κάθε χρήστη.

Ο παλμογράφος 54831B είναι 'συμβατός' με το προγραμματιστικό πακέτο LabVIEW με το οποίο τον ελέγχουμε. Από τις πρώτες εκδόσεις του πακέτου έχουν αναπτυχθεί έτοιμα κατάλληλα προγράμματα τα οποία εξομοιώνουν τον εν λόγω παλμογράφο. Τα προγράμματα αυτά παρουσιάζουν σε ένα γραφικό περιβάλλον την κύρια όψη του οργάνου. Το όνομα μάλιστα αυτών των προγραμμάτων προέκυψε ακριβώς από την ιδιότητά τους αυτή, ονομάζονται Virtual Instruments και οποιοσδήποτε, πολύ εύκολα μπορούμε να πούμε, είναι σε θέση να χειριστεί αυτά τα 'εικονικά όργανα'.

Κατά τη διαδικασία της μέτρησης με το LabVIEW όπως είναι φυσικό παρατηρείται κάποια καθυστέρηση μεταξύ της ένδειξης του οργάνου και της αντίστοιχης στην οθόνη του υπολογιστή. Αυτή η καθυστέρηση εξαρτάται από το μήκος του καλωδίου που συνδέει το όργανο και τον υπολογιστή. Επίσης ακούγεται ένας χαρακτηριστικός θόρυβος που οφείλεται στον ηλεκτρικό θόρυβο των συνδέσεων και των οργάνων.

Στην εφαρμογή που αναπτύξαμε ο παλμογράφος λαμβάνει τις επιθυμητές εξόδους και παρουσιάζει τα αποτελέσματα στην οθόνη του. Με τη χρήση του LabVIEW έρχονται τα παραπάνω δεδομένα στην οθόνη του υπολογιστή. Ο όλος έλεγχος του οργάνου πρέπει να τονίσουμε ότι γίνεται με το LabVIEW αρκεί το όργανο να είναι ανοιχτό και συνδεδεμένο ασφαλώς με τον υπολογιστή. Για τη λήψη μιας νέας μέτρησης πρέπει να ξανατρέξουμε το κατάλληλο πρόγραμμα.

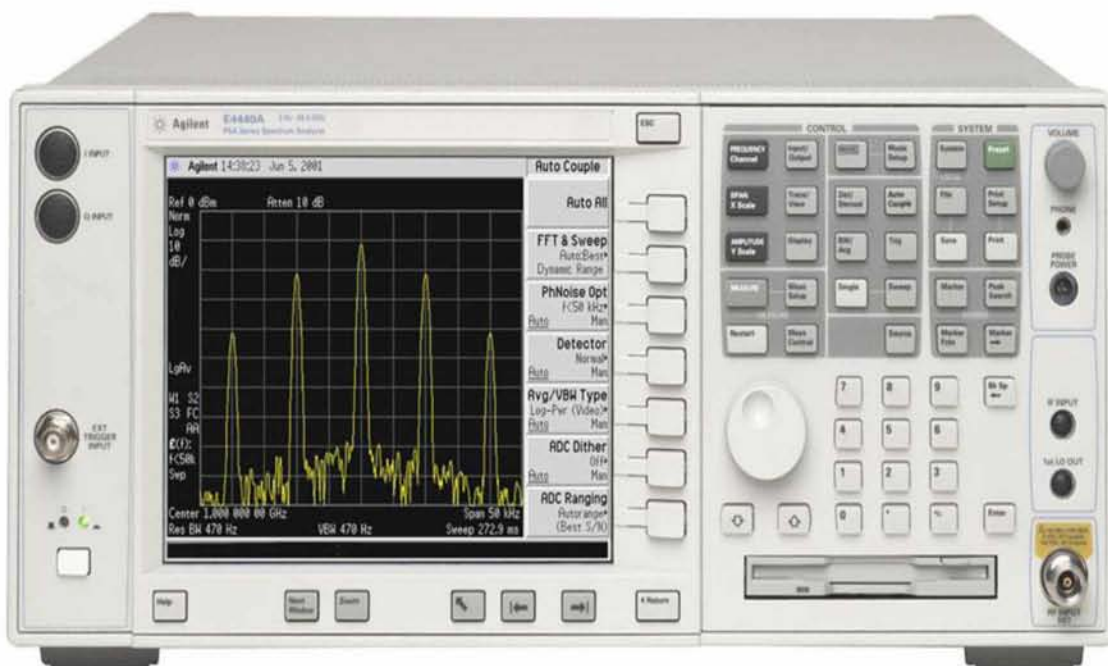


Σχήμα 4.2.1 - Πρόσωση του παλμογράφου

4.3 Ο αναλυτής φάσματος E4440A της Agilent

Αναλυτής Φάσματος ονομάζεται η συσκευή η οποία απεικονίζει την ένταση ενός στιγμιότυπου κάποιου σήματος συναρτήσει της συχνότητάς του. Σε αντίθεση με την εικόνα που παράγει ένας παλμογράφος (η οποία απεικονίζει την ένταση του σήματος συναρτήσει του χρόνου), στην εικόνα

ενός αναλυτή φάσματος είναι ευδιάκριτα χαρακτηριστικά όπως οι αρμονικές συχνότητες, το εύρος συχνοτήτων, η περιοδική αλλαγή αυτών κ.α. Οι δύο βασικοί τύποι αναλυτών φάσματος βασίζονται είτε σε φίλτρα συχνοτήτων μεταβλητού εύρους είτε σε ταχείς μετασχηματισμούς Φουριέ (FFT), με τους τελευταίους να είναι πιο επικρατείς. Επίσης, ο αναλυτής φάσματος είναι μια συσκευή με επιλογή ένδειξης συχνοτήτων που επιτρέπει τον προσδιορισμό των εξόδων του στοιχείου που συνδέεται στην είσοδο του αναλυτή, σε διάφορες συχνότητες που μας ενδιαφέρουν. Έχει τη δυνατότητα σάρωσης επιτρέποντας έτσι την παρουσίαση όλου του φάσματος της εξόδου του στοιχείου που συνδέεται στην είσοδό του. Οι περισσότεροι αναλυτές φάσματος έχουν λογαριθμική έξοδο και συνεπώς προσφέρουν και τη δυνατότητα μέτρησης σε λογαριθμικές μονάδες dBm. Ο αναλυτής φάσματος που χρησιμοποιήσαμε είναι ο E4440A της Agilent με εύρος συχνοτήτων από 110 GHz έως 325 GHz.



Σχήμα 4.3.1 - Πρόσωση του αναλυτή φάσματος

ΚΕΦΑΛΑΙΟ 5

Το πρόγραμμα ελέγχου της γεννήτριας (E4438C)

5.1 Λειτουργίες που θέλουμε να επιτελέσει η γεννήτρια(Signal Generator).

Για τον έλεγχο της γεννήτριας θα μπούμε στη διαδικασία κατασκευής ενός προγράμματος οδήγησης. Υπενθυμίζουμε ότι η επικοινωνία του οργάνου με τον υπολογιστή γίνεται με τη μεταφορά strings ASCII. Τα προγραμματιζόμενα μέσω μηνυμάτων όργανα προγραμματίζονται μέσω υψηλού επιπέδου strings ASCII χαρακτήρων. Το όργανο έχει ένα τοπικό επεξεργαστή, οποίος στοιχίζει τα strings εντολών και θέτει τα κατάλληλα bits των καταχωρητών να κάνουν ορισμένες λειτουργίες.

Σε γενικές γραμμές σε μια γεννήτρια συχνοτήτων θέλουμε να κάνουμε τα εξής:

1. Να δίνεται η δυνατότητα καθορισμού του πλάτους του σήματος.
2. Να δίνεται η δυνατότητα καθορισμού της συχνότητας του σήματος.
3. Να δίνεται η δυνατότητα ενεργοποίησης της εξόδου.

Από το manual του οργάνου λοιπόν βρίσκουμε ότι οι αντίστοιχες εντολές είναι αυτές που συνοψίζονται στον πίνακα 5.1.1 παρακάτω. Εν συνεχεία υλοποιούμε ξεχωριστά για την κάθε εντολή το αντίστοιχο vi με αντικειμενικό σκοπό την τελική επιτυχημένη συνεργασία τους.

Command	Description
FREQ<nrf>	Set the output frequency to <nrf> KHZ.
MVLEV<nrf>	Set the output level to <nrf> in mV
RFON	Switch on RF output.

Πίνακας 5.1.1 Εντολές και περιγραφή λειτουργιών που επιτελούν.

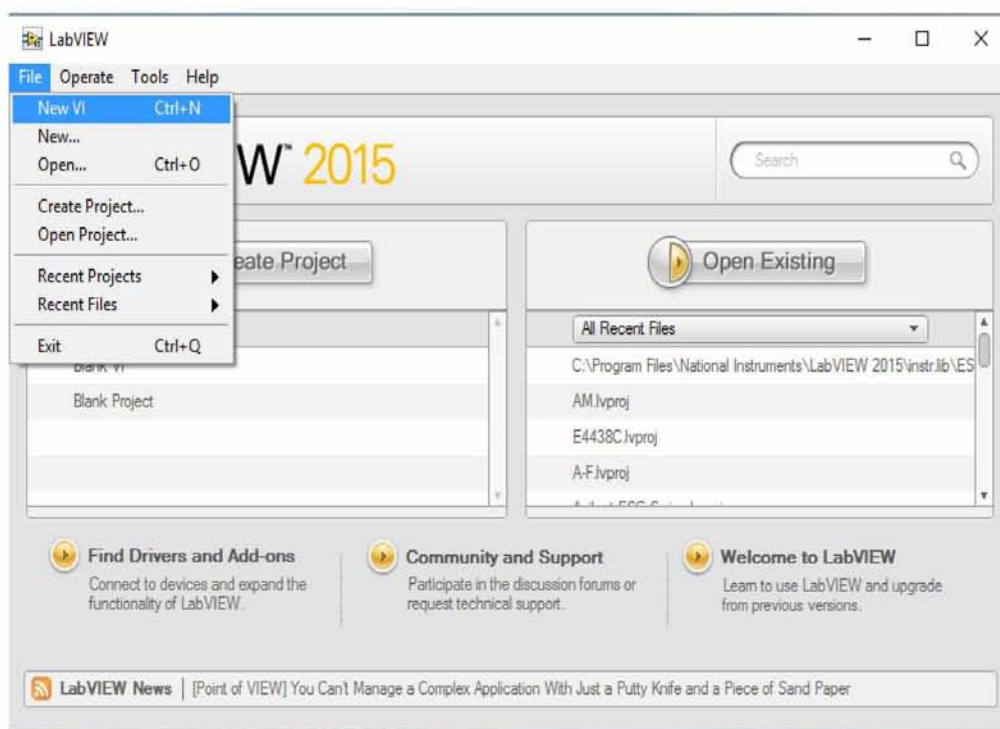
Επιπλέον θα χρειαστούν ακόμα 2 vi's. Το ένα φυσικά θα είναι το initialize.vi το οποίο θα κάνει το handshaking με το όργανο και ένα close.vi το οποίο θα τερματίσει την επικοινωνία με το όργανο και θα ανακαταλείπει τους πόρους του συστήματος. Πρέπει επίσης να τονίσουμε ότι ο προγραμματισμός γίνεται με βάση τις συναρτήσεις VISA και για το λόγο αυτό χρειαζόμαστε τα vi's initialize και close. Ωστόσο το πρωτόκολλο VISA είναι πιο γενικό και υπάρχουν επίσης και κάποιες standard εντολές που ισχύουν για όλα τα όργανα, όπως για παράδειγμα η εντολή *IDN?. Η οποία κάνει το όργανο talker ζητώντας του να προσδιορίσει την ταυτότητα του.

Τέλος μπορούμε να πούμε ότι κατασκευάζοντας ένα (απλό) initialize.vi και ένα close.vi δεν υπάρχει περιορισμός ως προς τα ποια όργανα θα χρησιμοποιηθούν. Θα ισχύουν για όλα τα όργανα όλων των τύπων. Κάνουμε δηλαδή το initialize.vi αλλά και close.vi μία φορά και μπορούμε να τα χρησιμοποιούμε για κάθε φορά που κάνουμε κάποιο πρόγραμμα οδήγησης. Αυτή ακριβώς είναι και η γενικότητα του πρωτοκόλλου VISA όπως αναφέραμε στο 3ο κεφάλαιο. Αντί να χρειάζεται να μάθει κάποιος διαφορετικά μηνύματα εντολών για κάθε είδος οργάνου, χρειάζεται να μάθει απλά ένα σύνολο εντολών που είναι οι VISA Read, VISA Write, VISA Assert Trigger, VISA Clear, VISA Read STB.

5.2 E4438CInitialize.vi

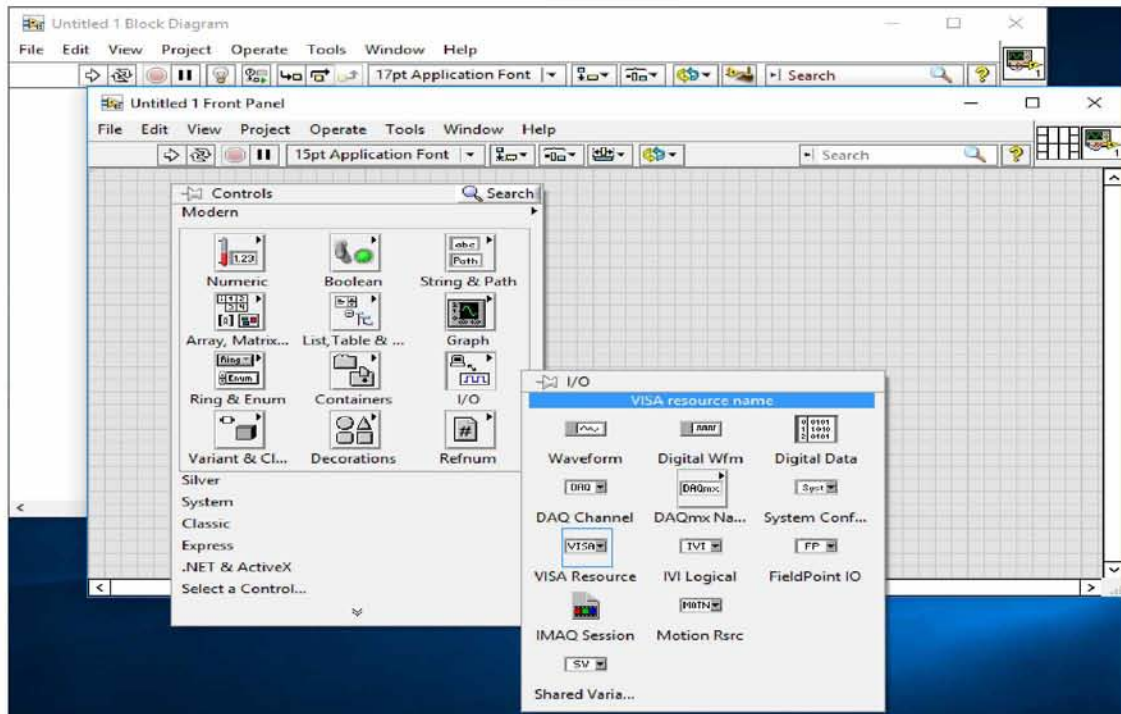
Το E4438CInitialize.vi κατασκευάζεται εύκολα ακολουθώντας τα παρακάτω βήματα:

1. Ανοίγουμε το LabVIEW(Σχήμα 5.2.1) και επιλέγουμε New VI.



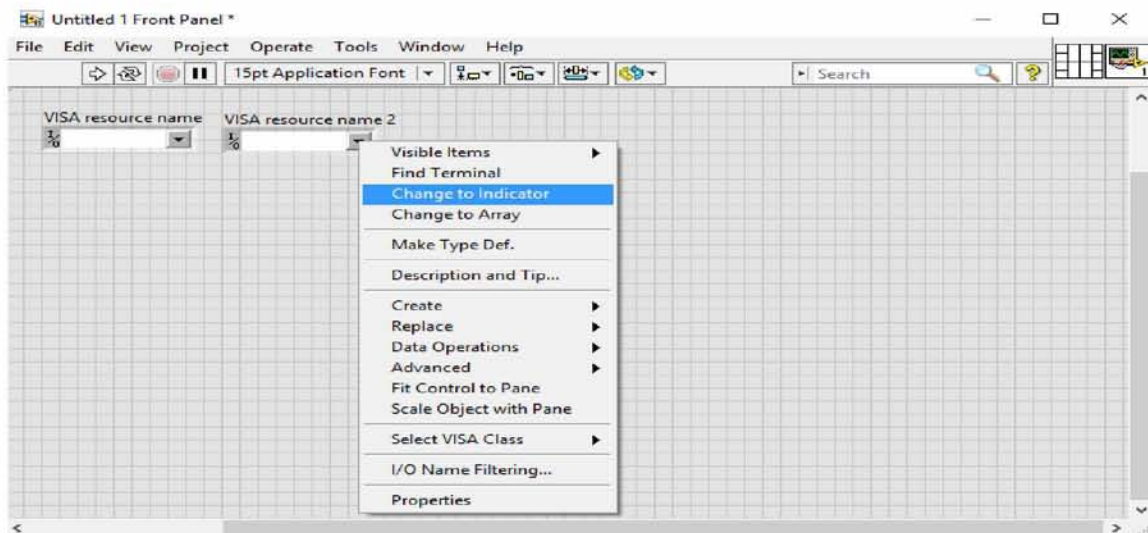
Σχήμα 5.2.1 Παράθυρο έναρξης του LabVIEW.

2. Στο Front Panel κάνουμε τα εξής:
 - Δεξί κλικ -> I/O -> VISA Resource Name -> Drag&Drop στο Front Panel.(Σχήμα 5.2.2)



Σχήμα 5.2.2 Ανοιγμα ενός VISA Session

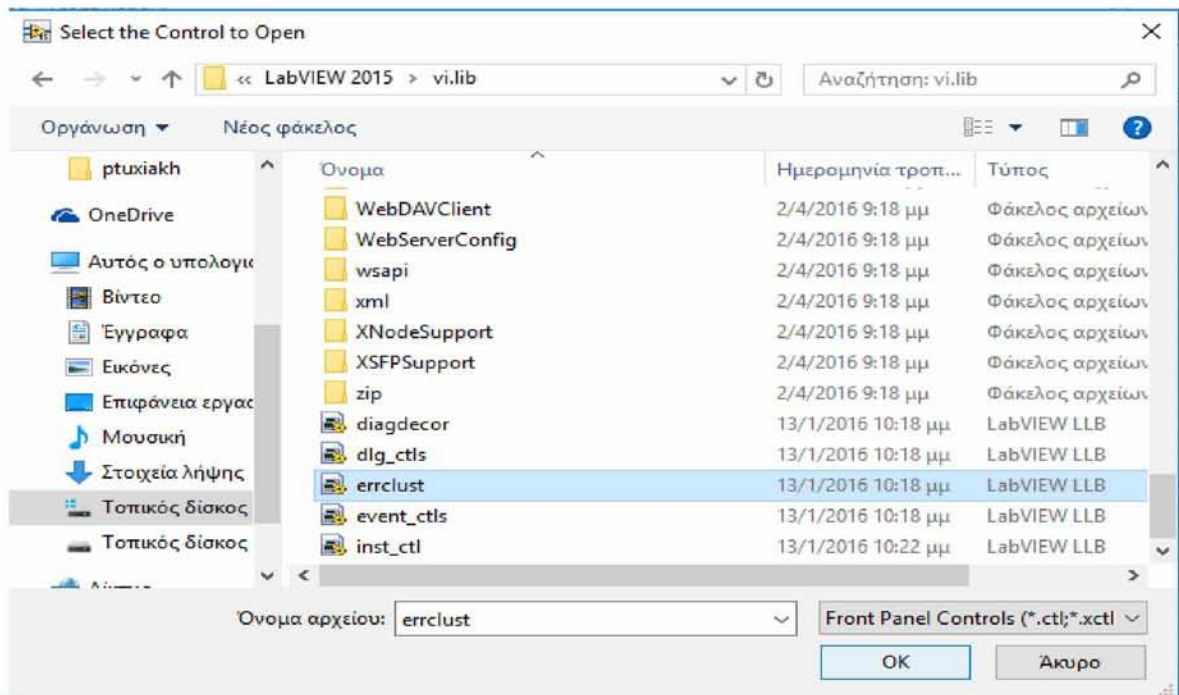
- Δεξί κλικ ->I/O -> VISA Resource Name -> Drag&Drop στο Front Panel ->Δεξί κλικ στο VISA Resource Name2 -> Change to Indicator από το pop-up menu (Σχήμα 5.2.3)



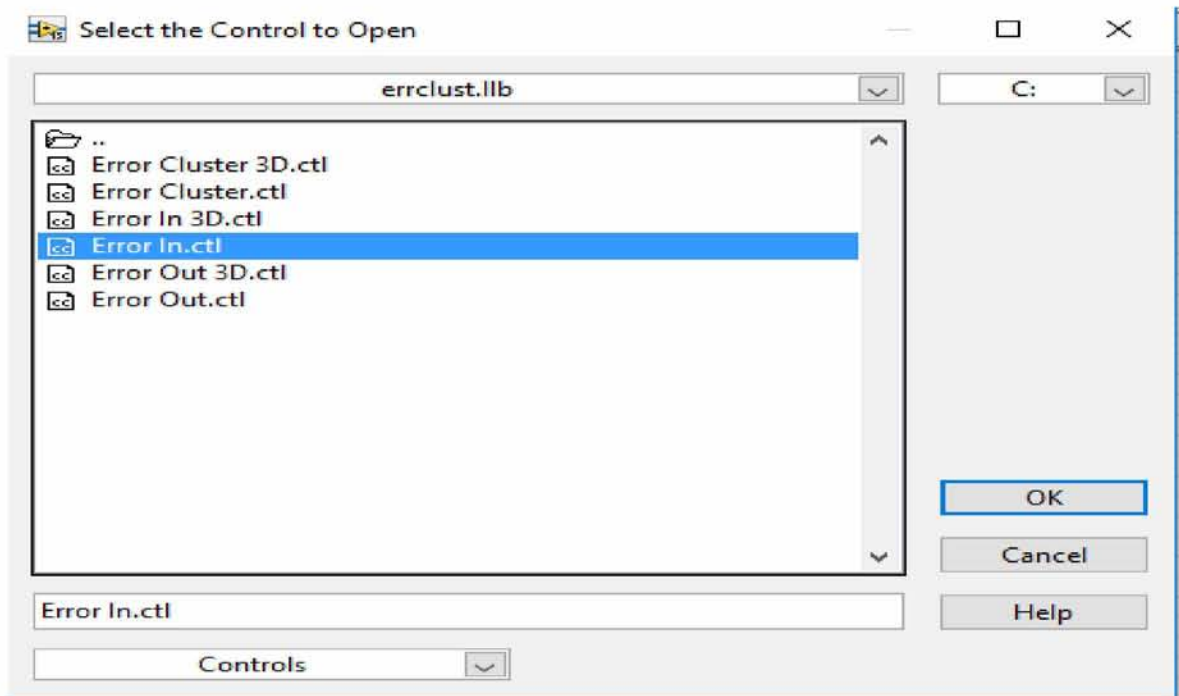
Σχήμα 5.2.3 Αλλαγή σε Indicator

- Δεξί κλικ -> Select a Control -> LabVIEW2015/vi.lib -> errclust.lib από το dialog box (Σχήμα 5.2.4) ->Error In.ctl από

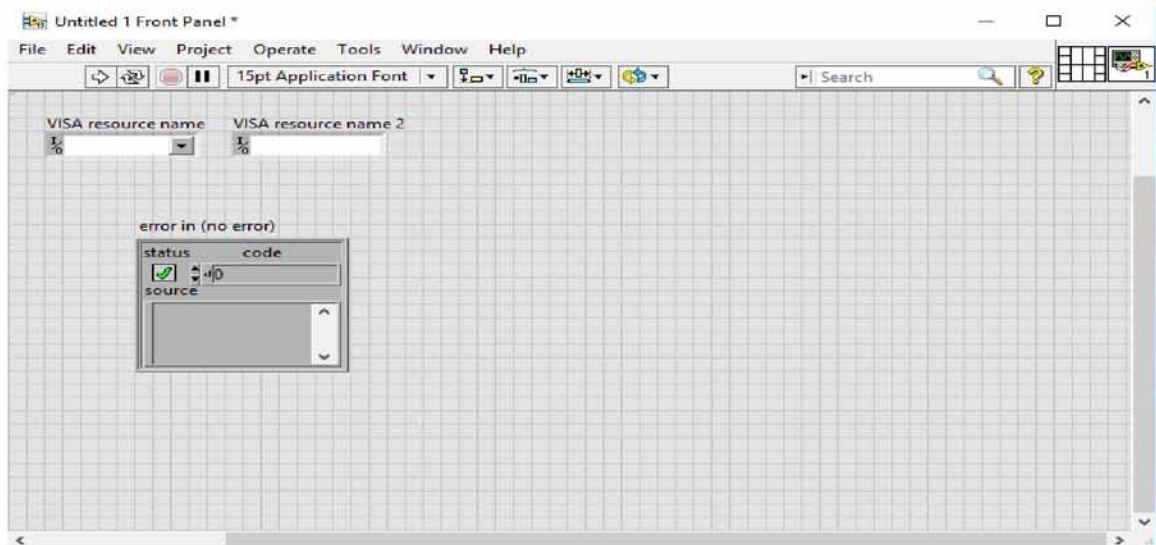
to Select the Control to Open (Σχήμα 5.2.5) ->Drug&Drop στο Front Panel (Σχήμα 5.2.6)



Σχήμα 5.2.4 Επιλογή βιβλιοθήκης errclust.llb

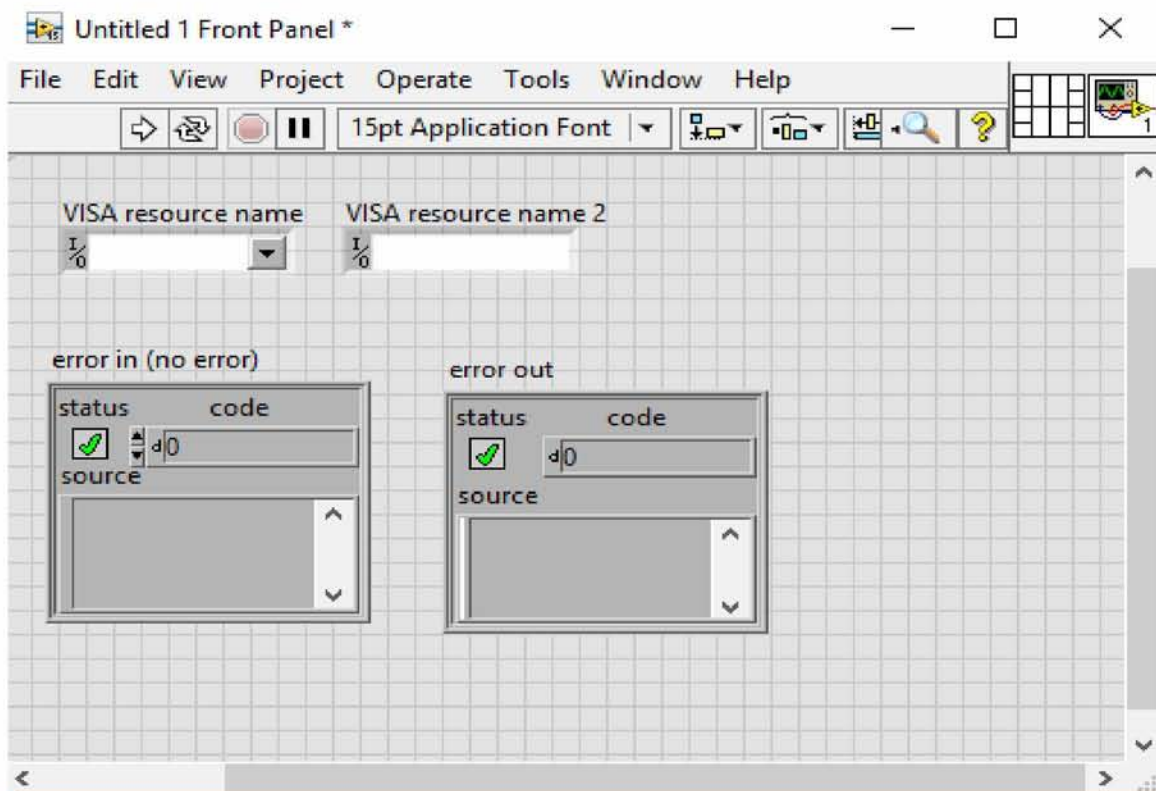


Σχήμα 5.2.5 Επιλογή Error In.ctl



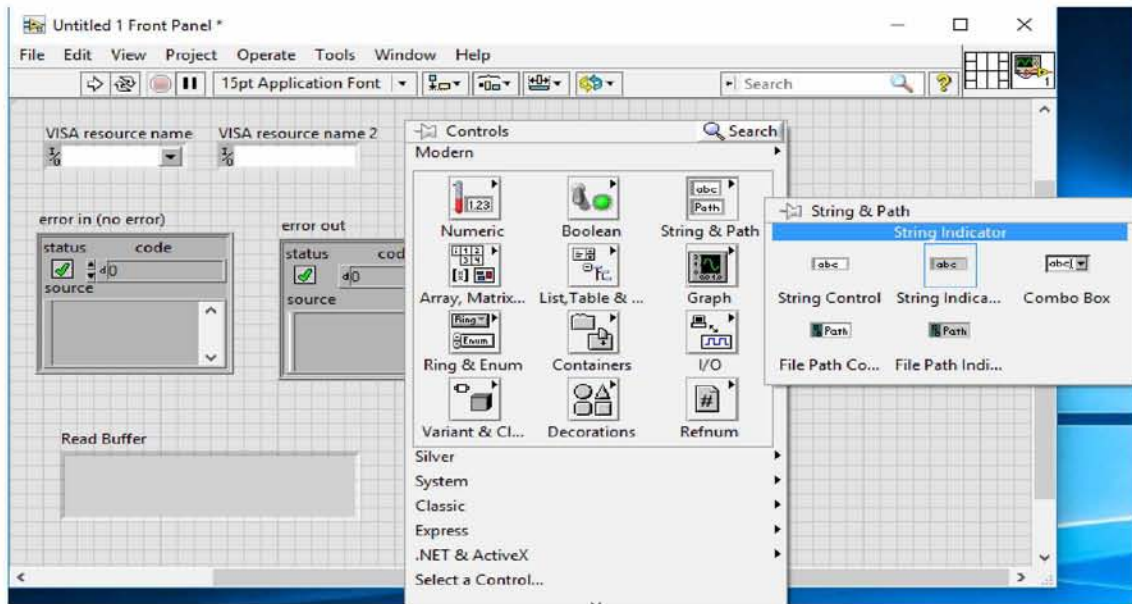
Σχήμα 5.2.6 Front Panel

- Δεξί κλικ -> Select a Control -> LabVIEW 2015/vi.lib -> errclust.lib από το dialog box -> Error Out.ctl από το Select the Control to Open -> Drag&Drop στο Front Panel (Σχήμα 5.2.7).



Σχήμα 5.2.7 Front Panel

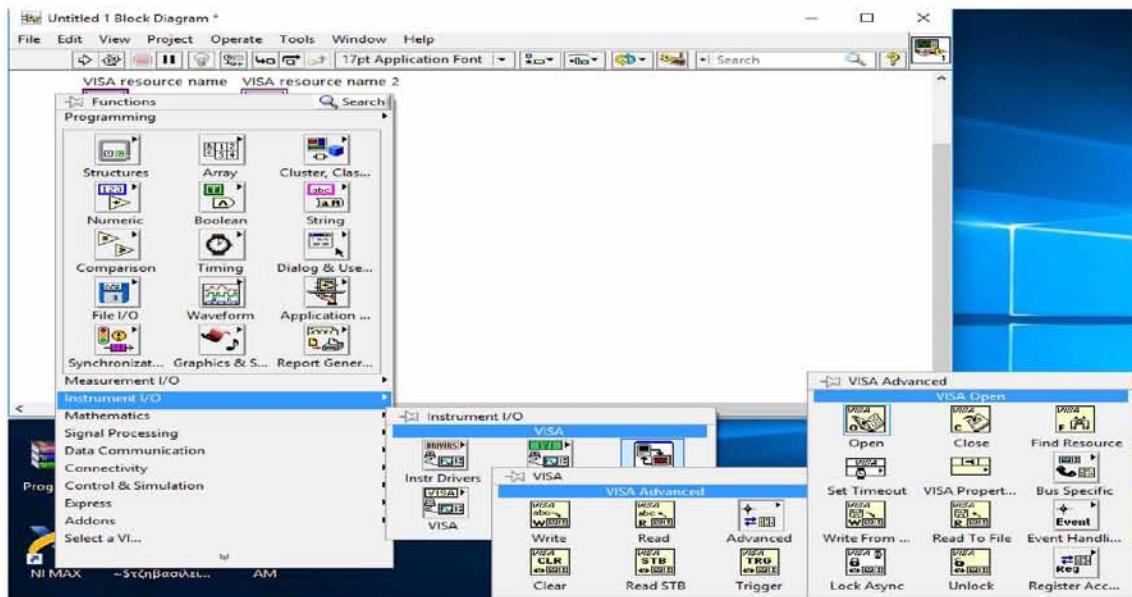
- Δεξί κλικ -> String and Path -> String Indicator -> Drag&Drop στο Front Panel -> Δεξί κλικ στην ετικέτα string και γράφουμε read buffer (Σχήμα 5.2.8).



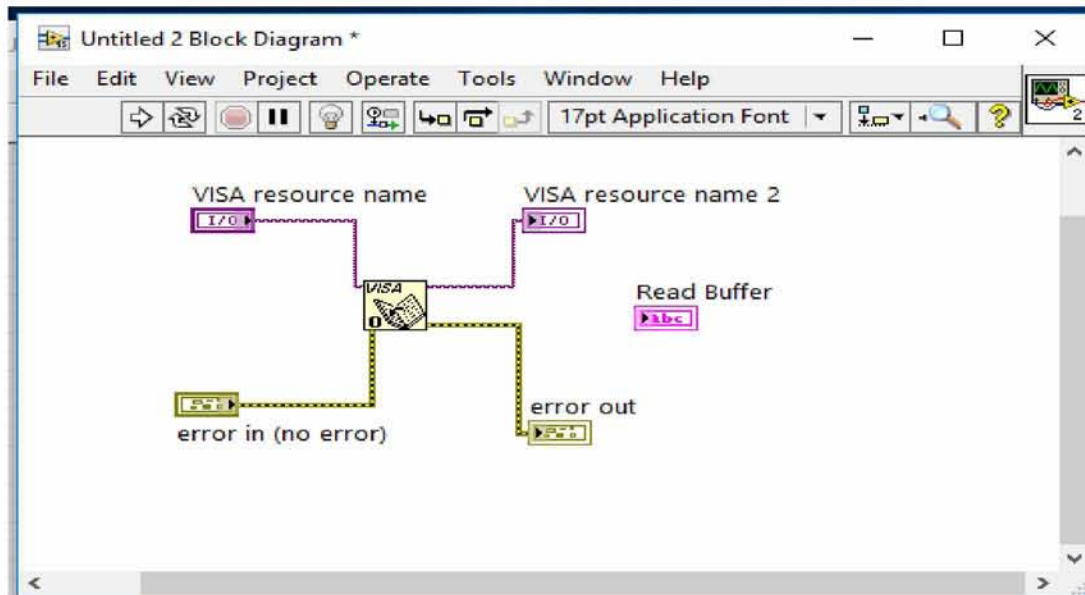
Σχήμα 5.2.8 Front Panel

3. Στο Block Diagram κάνουμε τα εξής:

- Δεξί κλικ -> Instrument I/O -> VISA -> VISA advanced -> VISA open (Σχήμα 5.2.9) -> Drag&Drop στο Block Diagram και ενώνουμε τα τερματικά όπως φαίνεται στο Σχήμα 5.2.10.

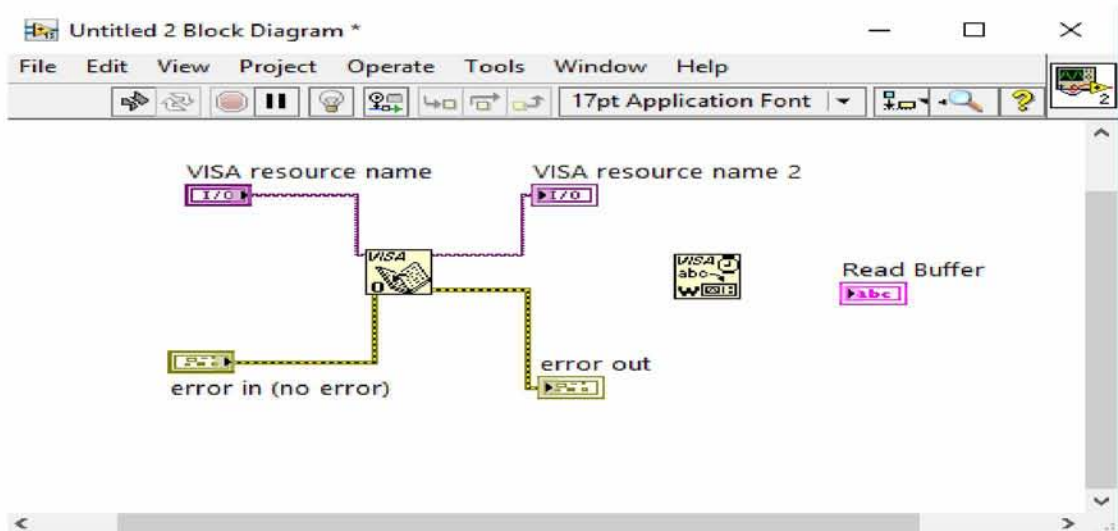


Σχήμα 5.2.9 Path του VISA Open

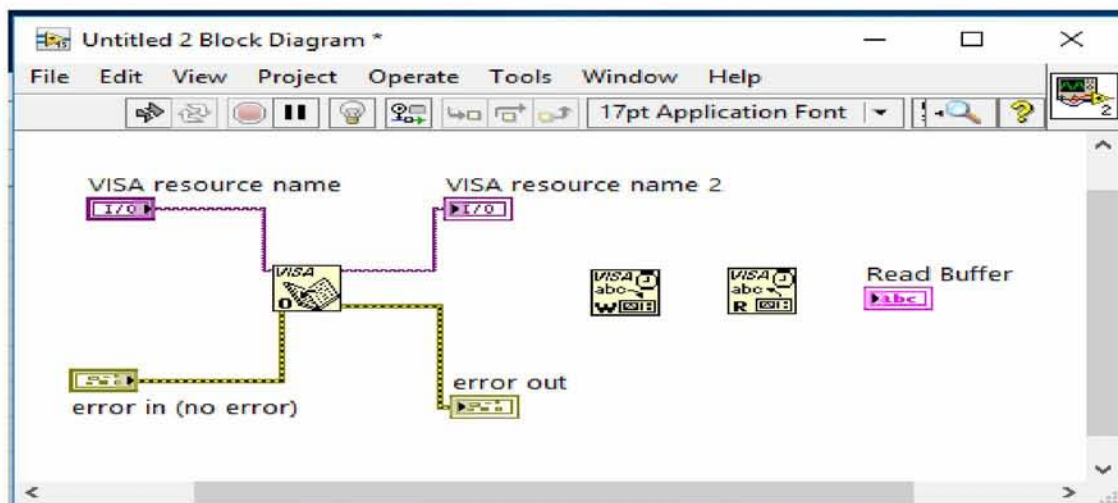


Σχήμα 5.2.10 Τρόπος που ενώνονται τα τερματικά

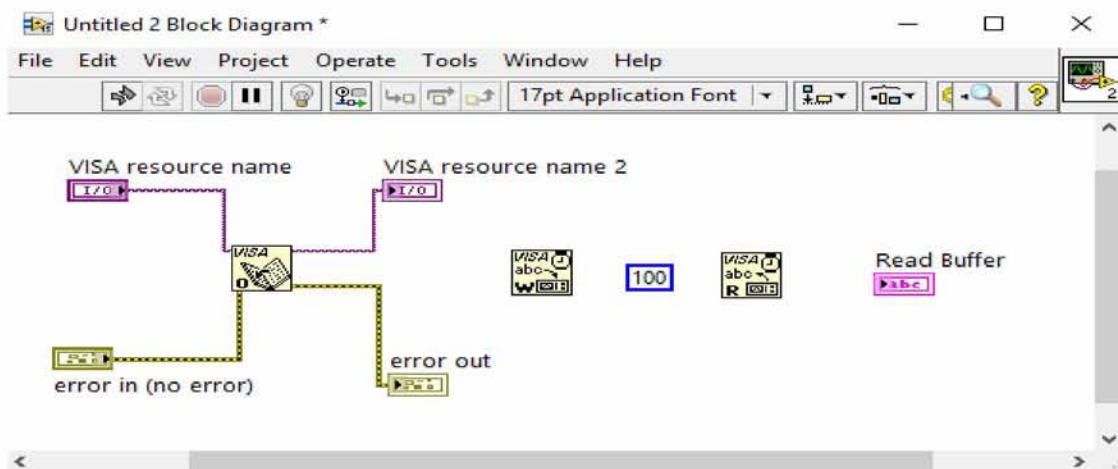
- Δεξί κλικ -> Instrument I/O -> VISA -> VISA Write -> Drag&Drop στο Block Diagram (Σχήμα 5.2.11)
- Δεξί κλικ -> Instrument I/O -> VISA -> VISA Read -> Drag&Drop στο Block Diagram (Σχήμα 5.2.12)
- Δεξί κλικ -> Numeric -> Numeric Constant -> Drag&Drop στο Block Diagram -> Γράφουμε μέσα στη Numeric Constant 100 (ή όποιον αριθμό της αρεσκείας μας - κατά προτίμηση όχι μεγαλύτερο από 100 για να διαβάζει πιο γρήγορα). Με την συνάρτηση αυτή λέμε στο όργανο να μας διαβάσει τους 100 πρώτους χαρακτήρες όταν θα γίνει Talker. (Σχήμα 5.2.13)
- Δεξί κλικ -> String -> String Constant -> Drag&Drop στο Block Diagram -> Γράφουμε μέσα στη String Constant *IDN? και ενώνουμε τα τερματικά όπως φαίνεται στο σχήμα 5.2.14. Με αυτή την συνάρτηση γράφουμε στο όργανο την εντολή *IDN? και αποκωδικοποιώντας την το όργανο γίνεται Talker. Στη συνέχεια το όργανο δίνει την ταυτότητά του και η απάντηση αυτή καταγράφεται σε μορφή συμβολοσειράς (string) στον string indicator με την ονομασία write buffer (Διαβάζοντας τους 100 πρώτους χαρακτήρες).



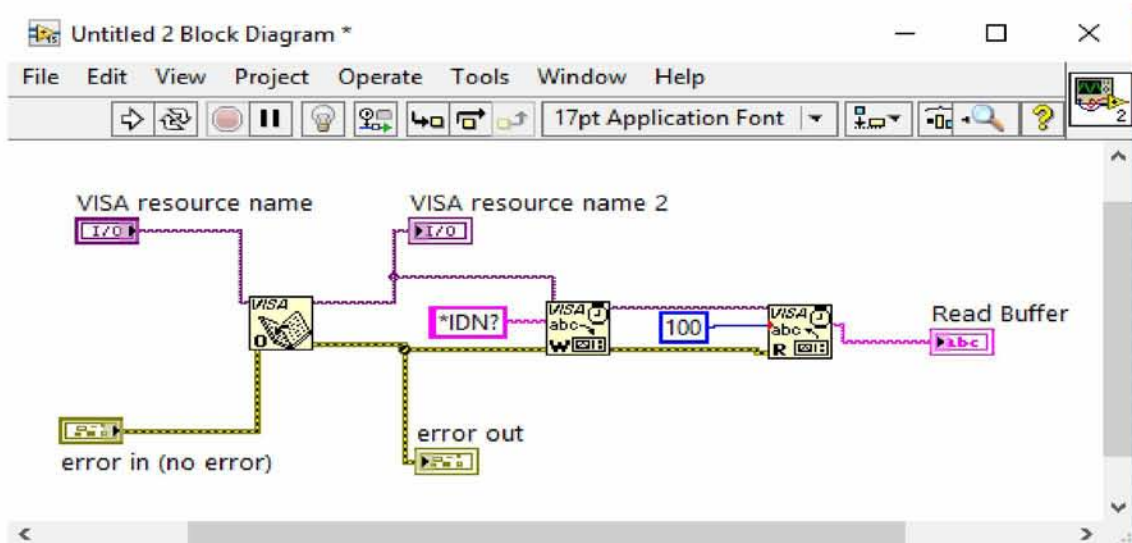
Σχήμα 5.2.11 VISA Write στο Block Diagram



Σχήμα 5.2.12 VISA Read στο Block Diagram

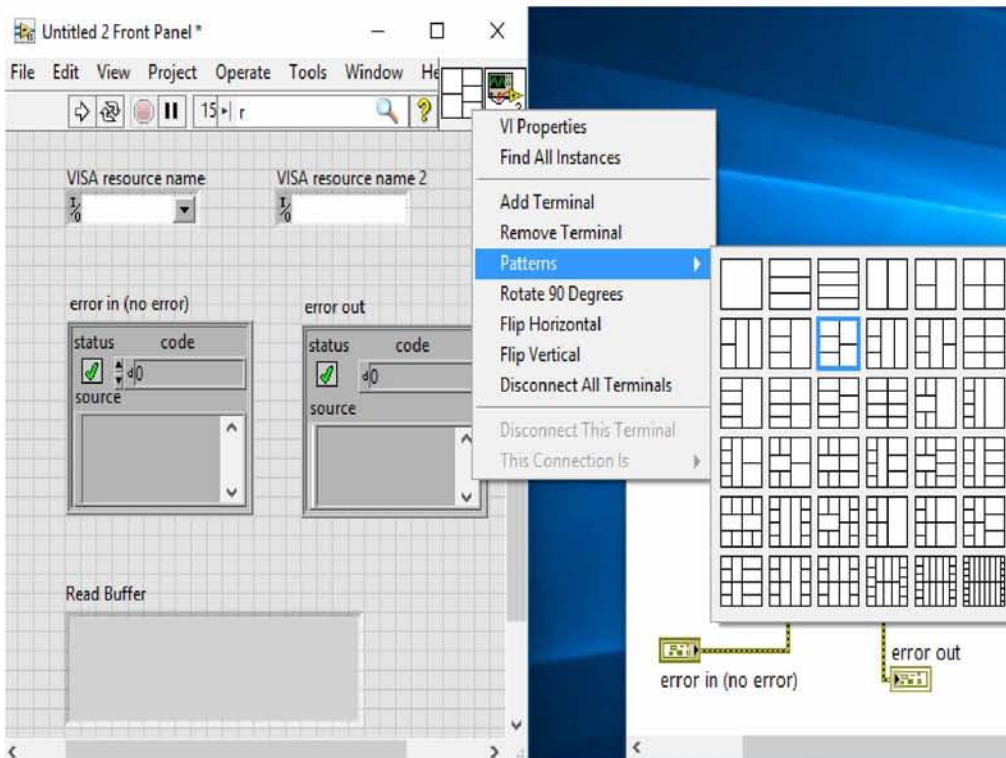


Σχήμα 5.2.13 Τοποθέτηση της Numeric Constant

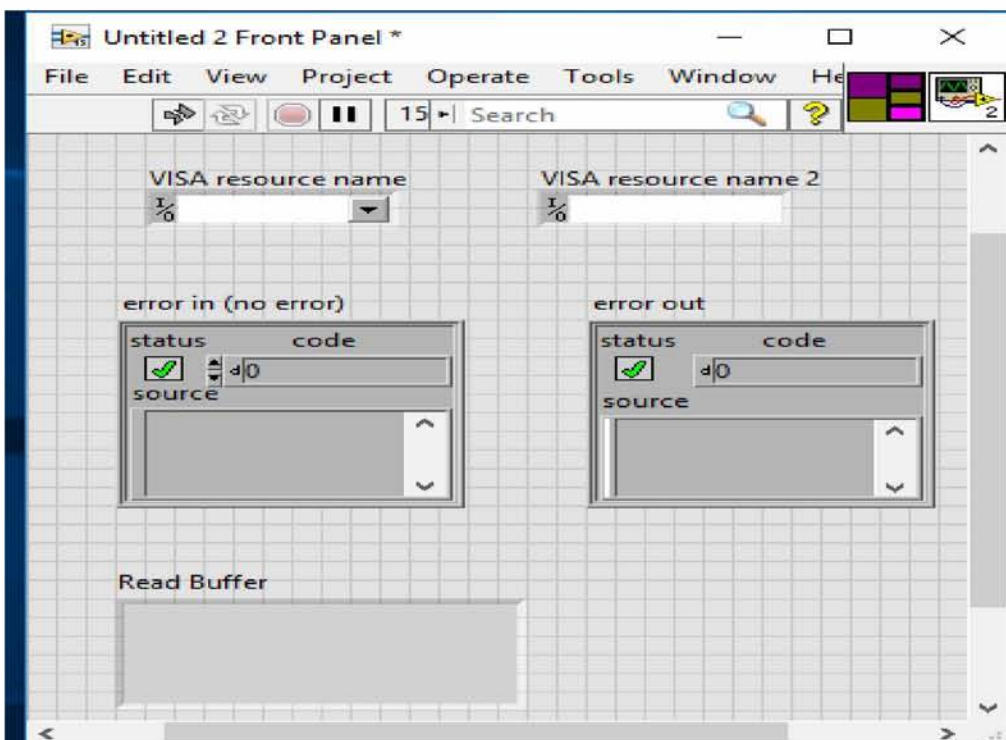


Σχήμα 5.2.14 Τοποθέτηση της String Constant και ένωση τερματικών

4. Επανερχόμαστε τώρα στο Front Panel και κάνουμε δεξί κλικ στη εικόνα που βρίσκεται στο πάνω δεξιά μέρος του παραθύρου του Front Panel. Από το pop-up επιλέγουμε το Patterns όπως φαίνεται στο Σχήμα 5.2.15. Από την επιλογή αυτή θα φανούν οι ακροδέκτες του νί που θα συνδέουμε τερματικά όταν θα θελήσουμε να το χρησιμοποιήσουμε ως subνί . Στο σημείο αυτό και για την διευκόλυνση ακολουθούμε κάποιους άγραφους κανόνες στο προγραμματισμό με LabVIEW για την ευκολία ανάγνωσης και δημιουργίας προγραμμάτων. Έτσι τα controls (VISA resource name, error in) αντιστοιχίζονται στα connectors που βρίσκονται στο αριστερό μέρος του icon ενώ τα indicators (VISA resource name 2, error out, κτλ) αντιστοιχίζονται στα connectors που βρίσκονται στο δεξί μέρος του icon. Επιπλέον τα VISA sessions ενώνονται στα πάνω connectors του icon ενώ τα error in και error out ενώνονται στα κάτω connectors του icon. Πρώτα κάνουμε κλικ στο connector του icon που θέλουμε να συνδέσουμε και στη συνέχεια στο control ή στο indicator που θέλουμε να αντιστοιχεί. Κατόπιν βλέπουμε ότι το connector από λευκό παίρνει το χρώμα της μεταβλητής στην οποία αντιστοιχεί. Δηλαδή αν η μεταβλητή είναι numeric το connector γίνεται μπλε, αν είναι string γίνεται ροζ κ.ο.κ. (Σχήμα 5.2.16). Από το pop-up επιλέγουμε το edit icon. Από την επιλογή καθορίζουμε με ποιο σχήμα θέλουμε να εμφανίζεται το νί αυτό ως subνί.



Σχήμα 5.2.15 Connectors του icon



Σχήμα 5.2.16 Σύνδεση των Connectors

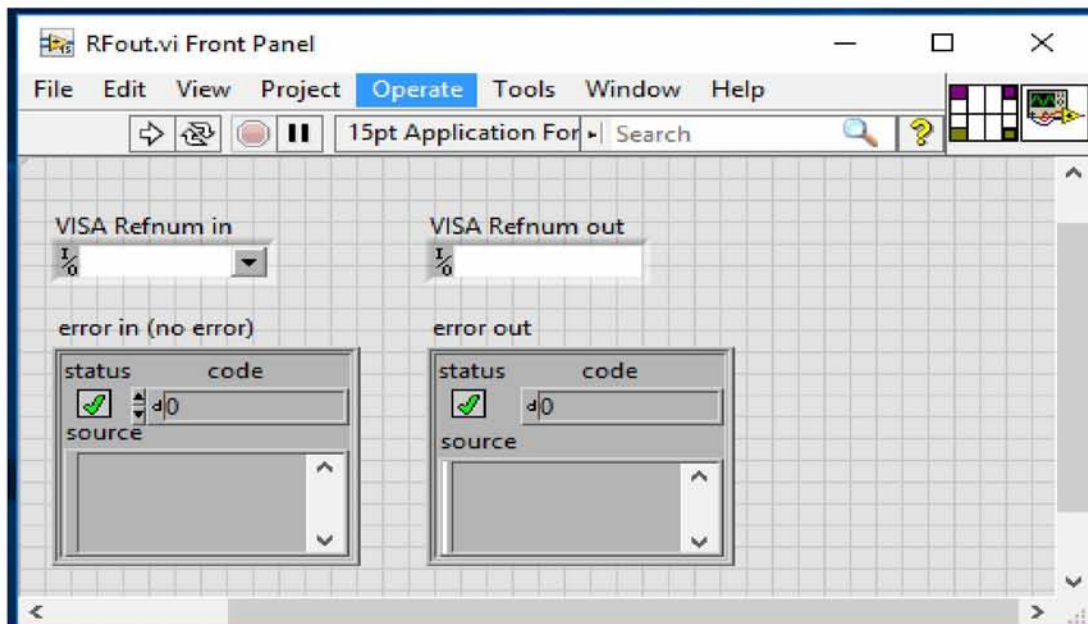
5. Σώζουμε το vi με την ονομασία Initialize.vi δημιουργώντας μία νέα βιβλιοθήκη ως εξής:

- Από το παράθυρο διαλόγου που εμφανίζεται μετά την επιλογή του “Save As...” επιλέγουμε “New VI Library” και δίνουμε το όνομα στη βιβλιοθήκη. Εμείς την ονομάζουμε E4438C και τη σώζουμε στην τοποθεσία LabView2015/instr.lib
- Μέσα σε αυτή σώζουμε και το Initialize.vi

5.3 Δημιουργία του RFout.vi

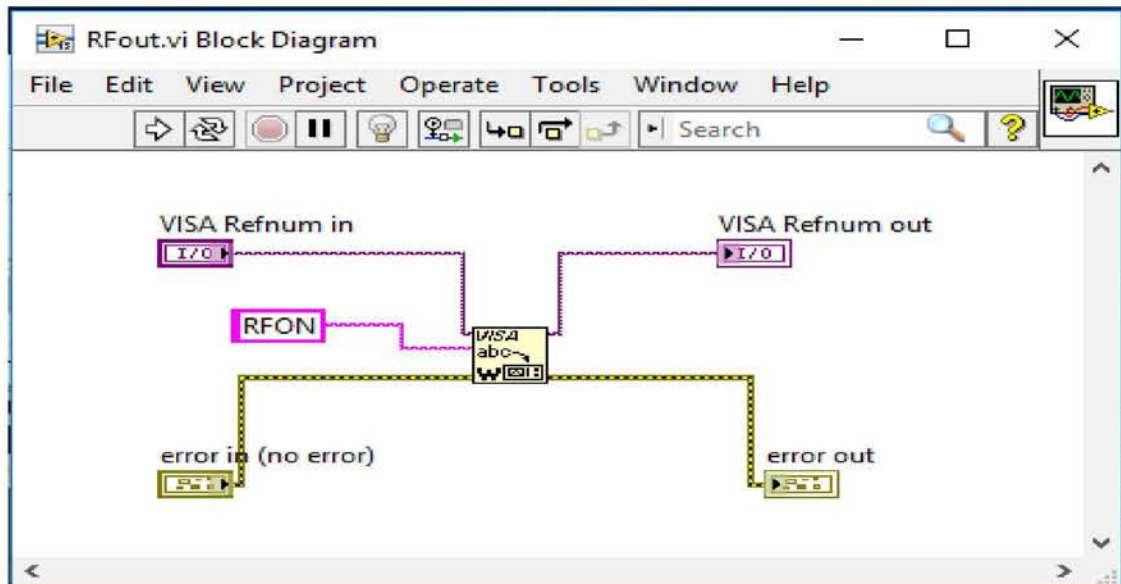
1. Στο Front Panel τοποθετούμε τα controls και indicators (Σχήμα 5.3.1) όπως περιγράφηκε παραπάνω:

- Error In.ctl
- Error Out.ctl
- VISA Resource Name (Control)
- VISA Resource Name (Indicator)



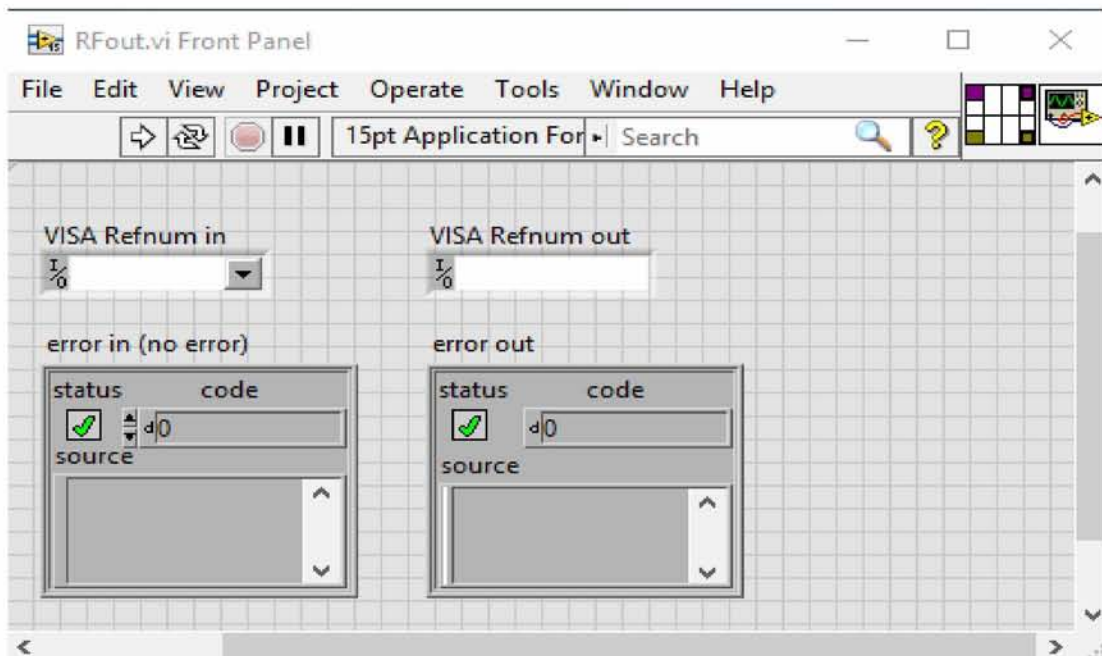
Σχήμα 5.3.1 Front Panel RFout.vi

2. Στο block diagram τοποθετούμε τη συνάρτηση VISA Write και τη σταθερά String Constant μέσα στην οποία γράφουμε την εντολή RFON με την οποία όπως προαναφέραμε ενεργοποιείται η έξοδος. Στη συνέχεια ενώνουμε τα τερματικά σύμφωνα με το Σχήμα 5.3.2.



Σχήμα 5.3.2 Block Diagram RfOut.vi

3. Επανερχόμαστε στο Front Panel και αντιστοιχίζουμε τα controls και τα indicators στα connector του icon σύμφωνα με το Σχήμα 5.3.3. Αν θέλουμε μπορούμε να εισάγουμε και μία εικόνα για τον τρόπο που θα φαίνεται σαν subvi από τον icon editor. Στη συνέχεια σώζουμε το vi με το όνομα RfOut.vi μέσα στη βιβλιοθήκη E4438C.llb.

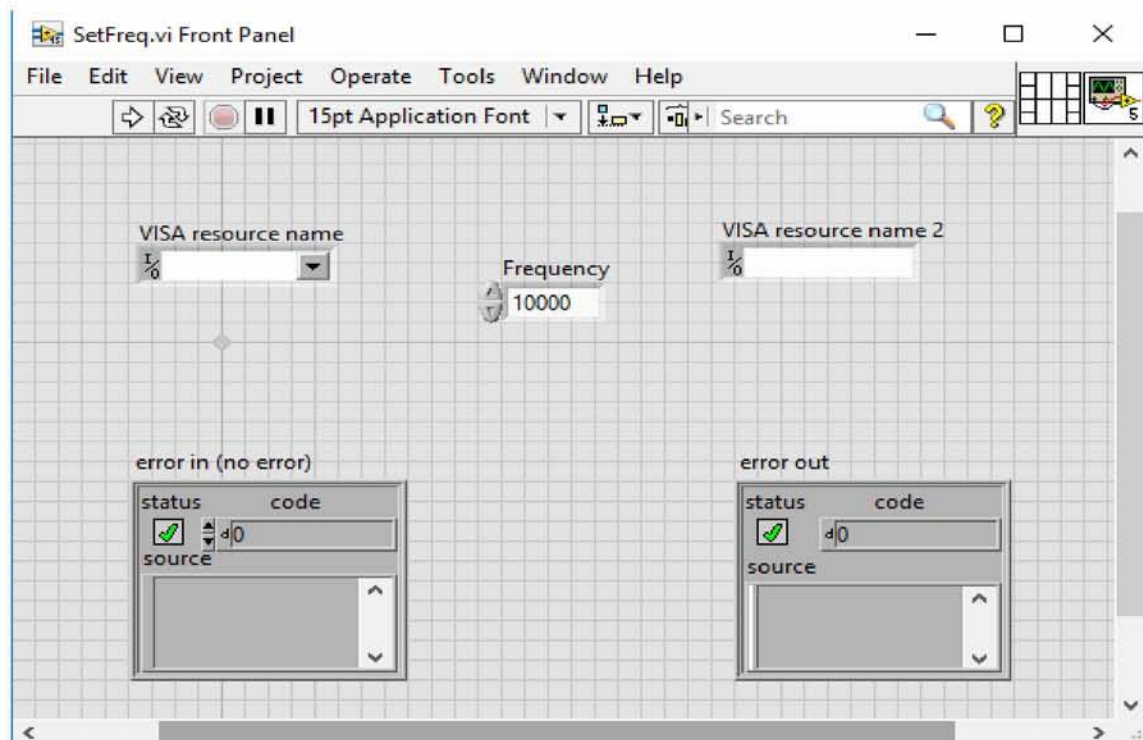


Σχήμα 5.3.3 Σύνδεση των Connectors

5.4 Δημιουργία του SetFreq.vi

1. Στο Front Panel τοποθετούμε τα controls και indicators (Σχήμα 5.4.1) όπως περιγράφηκε παραπάνω:

- Error In.ctl
- Error Out.ctl
- VISA Resource Name (Control)
- VISA Resource Name (Indicator)
- Δεξί κλικ -> Numeric -> Digital Control(Το ονομάζουμε Frequency).



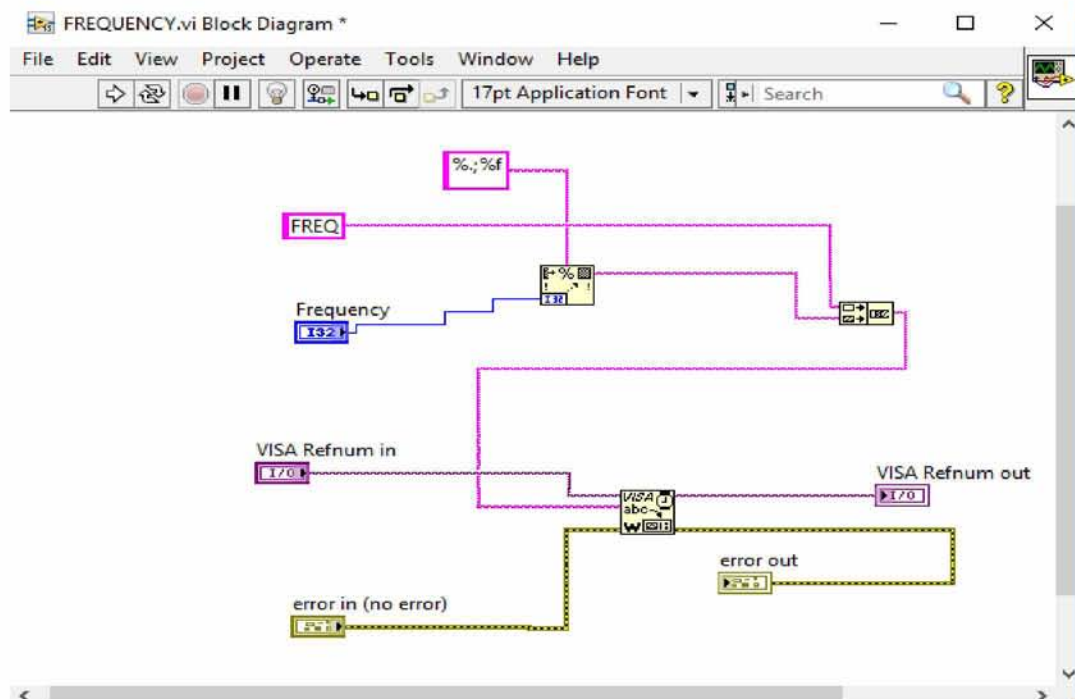
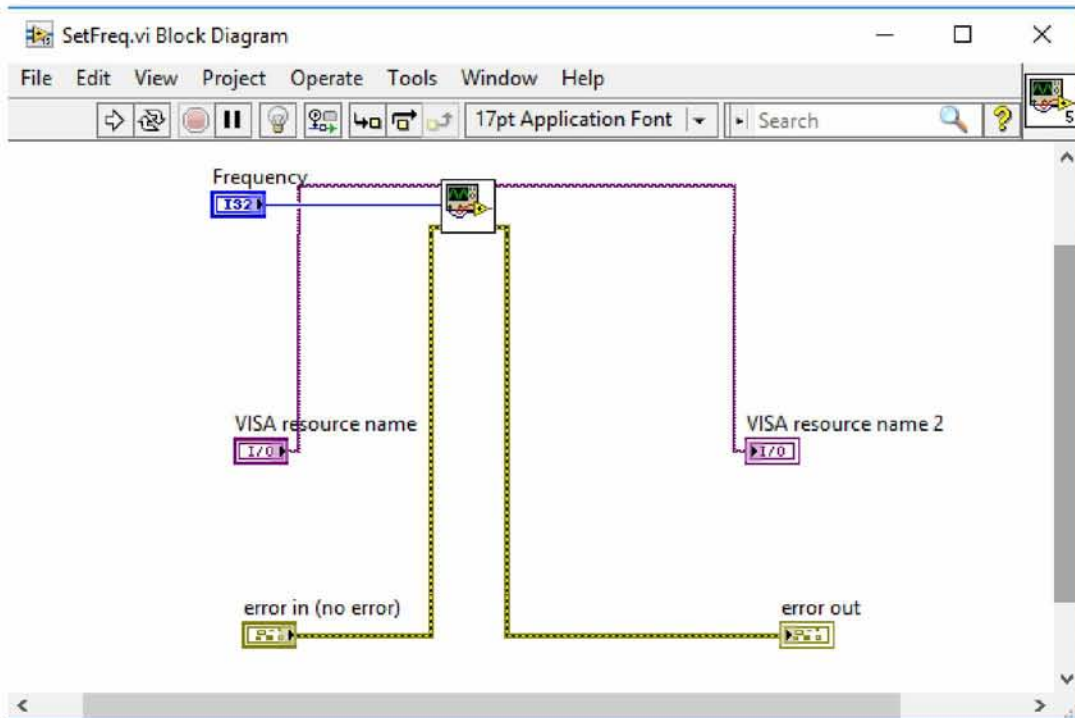
Σχήμα 5.4.1 FrontPanel του SetFreq

2. Κάνοντας δεξί κλικ στο control Frequency επιλέγω από το pop-up menu Data Range και θέτω τη μέγιστη και την ελάχιστη τιμή της συχνότητας που θέλω να δίνει η γεννήτρια. Φυσικά επιλέγουμε την τιμή της συχνότητας να κυμαίνεται μέσα στα όρια του πραγματικού εύρους της γεννήτριας.

3. Στο Block Diagram του Freq.vi κάνουμε τα εξής:

- Δεξί κλικ -> String -> String Constant -> Μέσα στη String Constant γράφω "FREQ"
- Δεξί κλικ -> String -> String/Number Conversion -> Number To Decimal String (Μετατρέπουμε δηλαδή τον δεκαδικό αριθμό σε String)

- Δεξί κλικ -> String -> Concentrate Strings (Για να κάνουμε σύμπτυξη των 2 strings σε ένα, το οποίο θα το δώσουμε σαν μέσω φυσικά μιας VISA Write εντολή στο όργανο σύμφωνα με το manual του κατασκευαστή) .
- Επιλέγουμε μία VISA Write συνάρτηση.
- Ενώνω τα τερματικά σύμφωνα με το Σχήμα 5.4.2 - 5.4.3.



Σχήμα 5.4.2-5.4.3 Control Panel του SetFreq.vi -FREQUENCY.vi

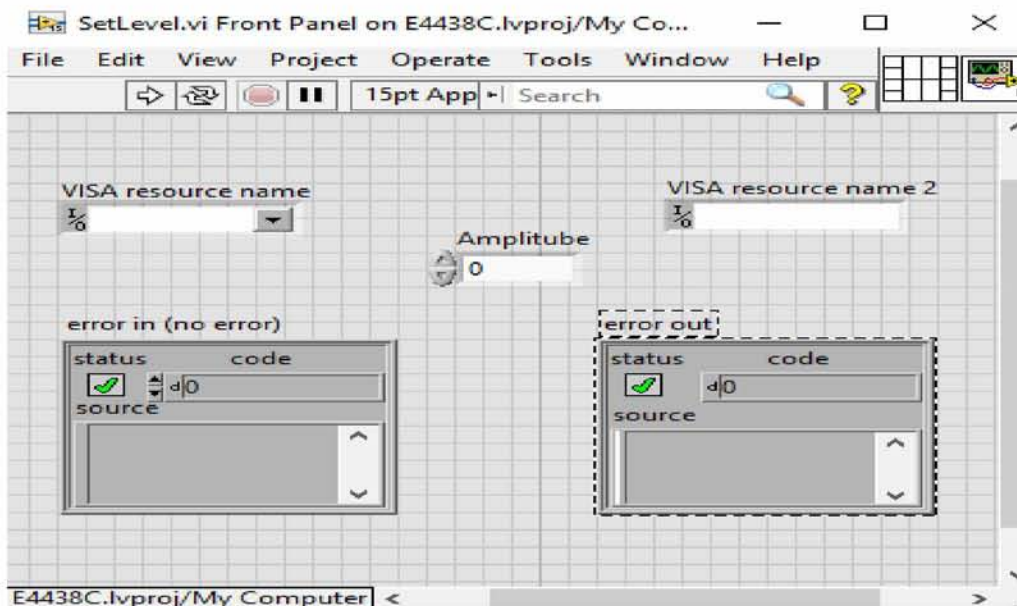
4. Στο Front Panel κάνουμε τα εξής:

- Δεξί κλικ στο Icon -> Show Connector από το pop-up menu
- Αντιστοιχίζουμε τα controls και τα indicators στους connectors
- Διπλό κλικ στο Icon και εισάγουμε την επιθυμητή εικόνα.

5.5 Δημιουργία του SetLevel.vi

1. Στο Front Panel τοποθετούμε τα controls και indicators (Σχήμα 5.5.1) όπως περιγράφηκε παραπάνω:

- Error In.ctl.
- Error Out.ctl.
- VISA Resource Name (Control).
- VISA Resource Name (Indicator).
- Δεξί κλικ -> Numeric -> Digital Control (Το ονομάζουμε Amplitude).
- Δεξί κλικ στο Icon -> Show Connector από το pop-up menu.
- Αντιστοιχίζουμε τα controls και τα indicators στους connectors.
- Διπλό κλικ στο Icon και εισάγουμε την επιθυμητή εικόνα.

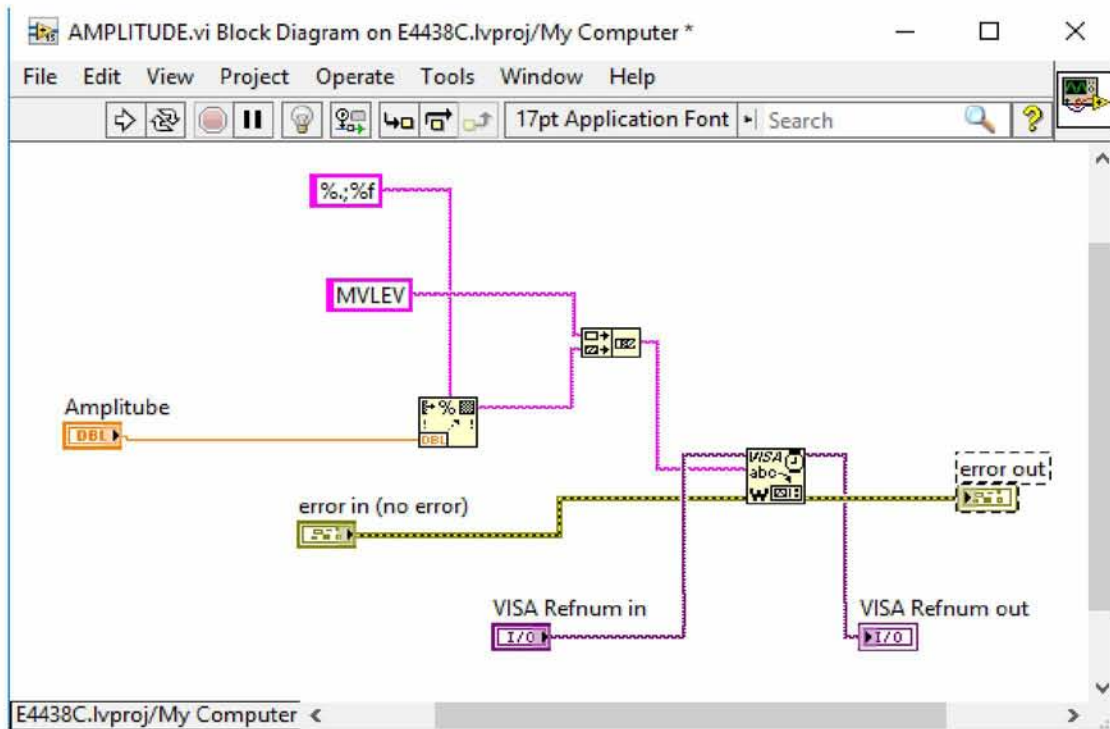
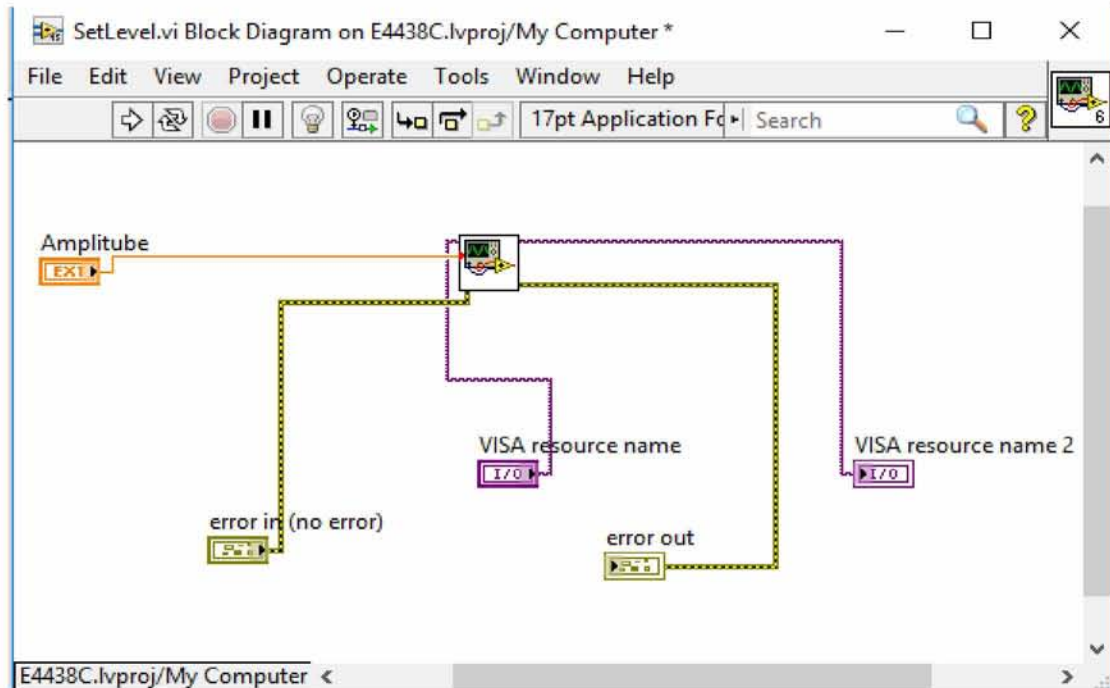


Σχήμα 5.5.1 Front Panel του SetLevel.vi

2. Στο Block Diagram του SetFreq.vi κάνουμε τα εξής:

- Δεξί κλικ -> String -> String Constant -> Μέσα στη String Constant γράφω "MVLEV".
- Δεξί κλικ -> String -> String/Number Conversion -> Number To Decimal String (Μετατρέπουμε δηλαδή τον δεκαδικό αριθμό σε String).
- Δεξί κλικ -> String -> Concentrate Strings (Για να κάνουμε σύμπτυξη των 2 strings σε ένα, το οποίο θα το δώσουμε μέσω

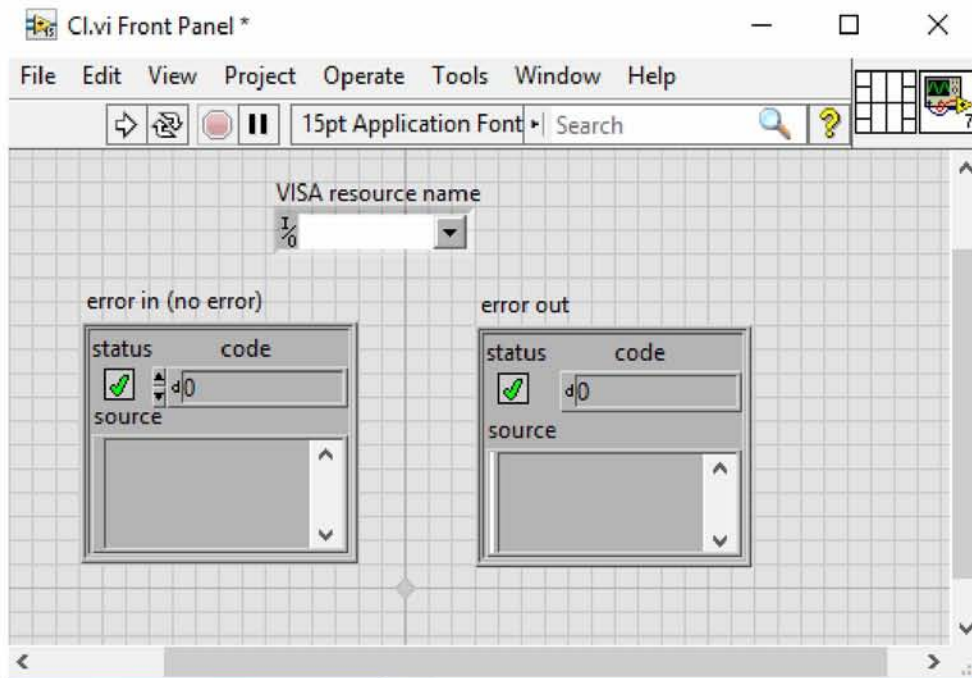
- φυσικά μιας VISA Write εντολή στο όργανο σύμφωνα με το manual του κατασκευαστή).
- Επιλέγουμε μία VISA Write συνάρτηση.
 - Ενώνω τα τερματικά σύμφωνα με το Σχήμα 5.5.2 – 5.5.3.
 - Σώζουμε το vi με το όνομα SetLevel.vi στη βιβλιοθήκη E4438C.



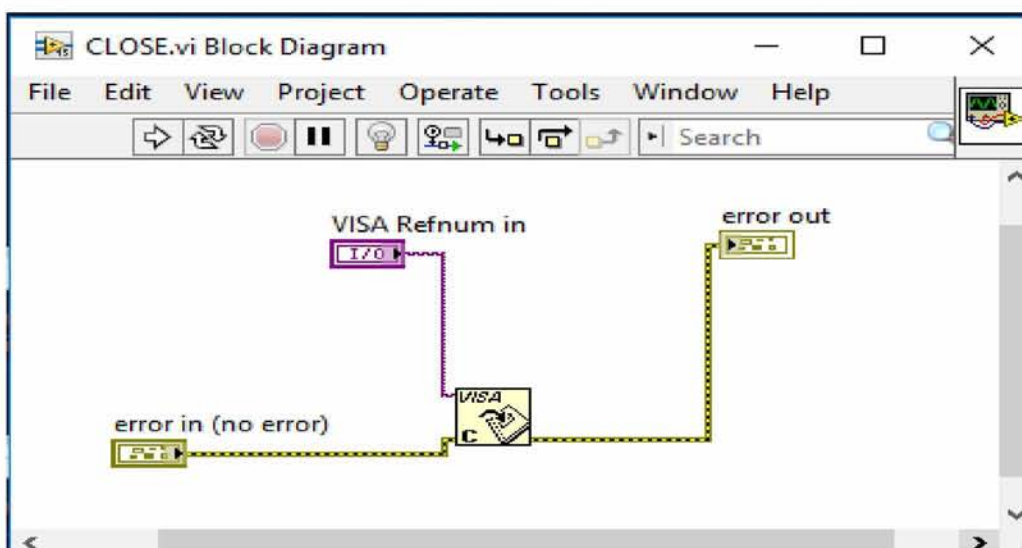
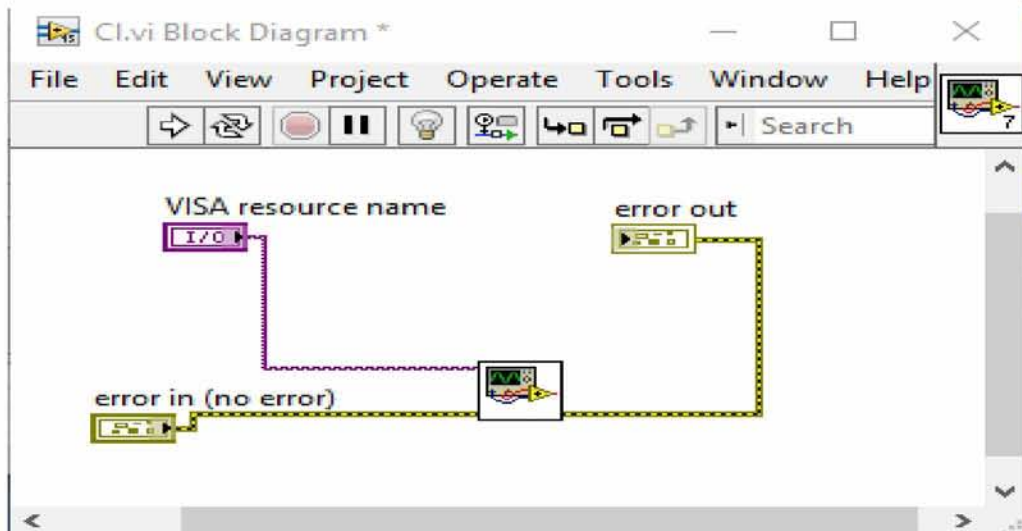
5.5.2-5.5.3 Block Diagram του SetLevel.vi-AMPLITUDE.vi.

5.6 Δημιουργία του Close.vi

1. Στο Front Panel τοποθετούμε τα controls και indicators (Σχήμα 5.6.1) όπως περιγράφηκε παραπάνω:
 - Error In.ctl.
 - Error Out.ctl.
 - VISA Resource Name (Control).
 - Αντιστοιχίζουμε τα controls και τα indicators στους connectors.
 - Διπλό κλικ στο Icon και εισάγουμε την επιθυμητή εικόνα.
2. Στο Block Diagram του Close.vi κάνουμε τα εξής:
 - Δεξί κλικ -> Instrument I/O -> VISA -> VISA Advanced -> VISA Close -> Drag&Drop στο Block Diagram.
 - Ενώνω τα τερματικά σύμφωνα με το Σχήμα 5.6.2 – 5.6.3.
 - Σώζουμε το vi με το όνομα TGR1040Close.vi στη βιβλιοθήκη E4438C.llb



Σχήμα 5.6.1 Front Panel Close.vi

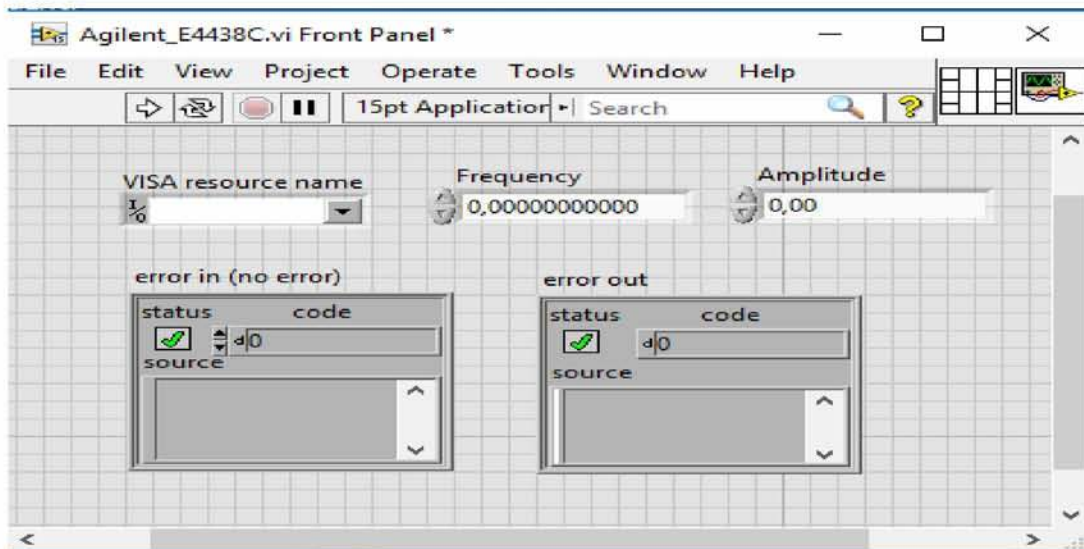


Σχήμα 5.6.2 – 5.6.3 Block Diagram Cl.vi -Close.vi

5.7 Δημιουργία του κορυφαίου Agilent E4438C.vi

1. Στο Front Panel τοποθετούμε τα controls αλλά και τα indicators σύμφωνα με όσα περιγράψαμε μέχρι τώρα και σύμφωνα με το Σχήμα 5.7.1. Τονίζουμε το γεγονός ότι το vi αυτό αποτελεί και το Interface του με το χρήστη οπότε και δεν χρειάζεται να ρυθμίσουμε connectors.

- Error In.ctl.
- Error Out.ctl.
- VISA Resource Name (Control).
- Numeric -> Digital Control το οποίο ονομάζουμε Amplitude.
- Numeric -> Digital Control το οποίο ονομάζουμε Frequency.
- Δεξί κλικ στο Amplitude και ρυθμίζουμε το Data Range.
- Δεξί κλικ στο Frequency και ρυθμίζουμε το Data Range.

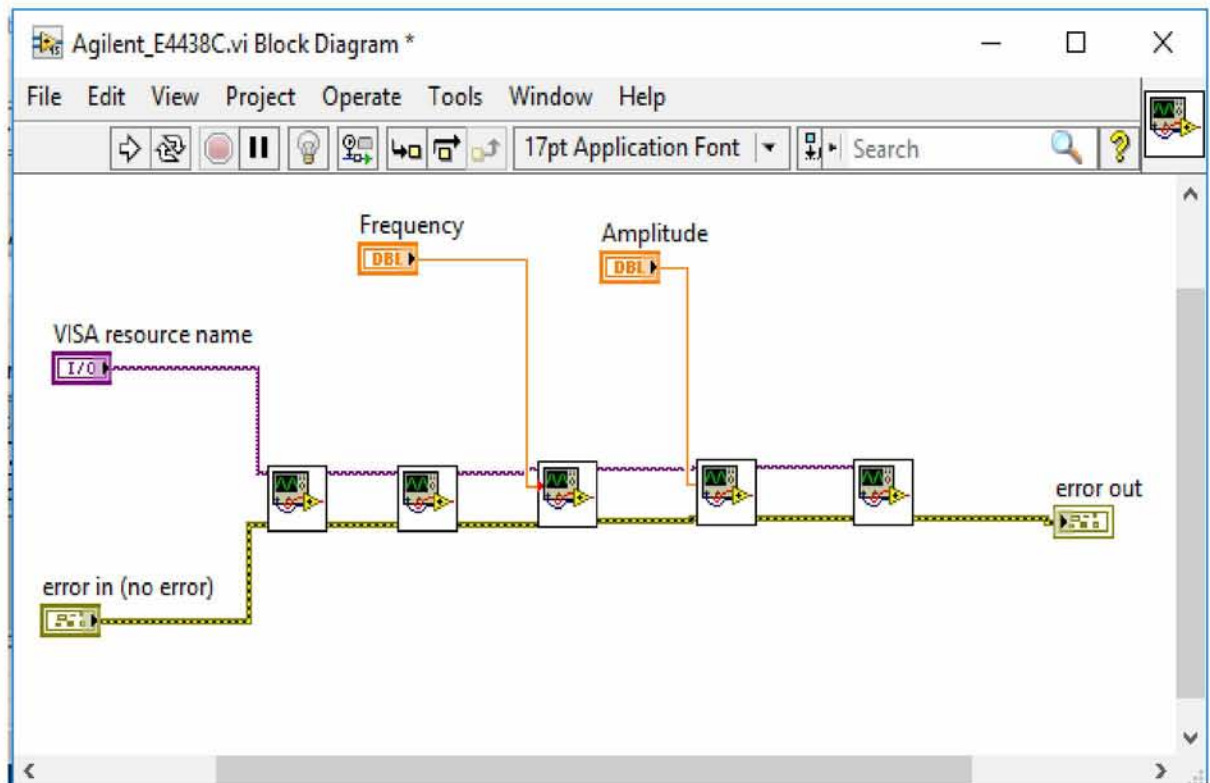


Σχήμα 5.7.1 Front Panel του Agilent_E4438C.vi

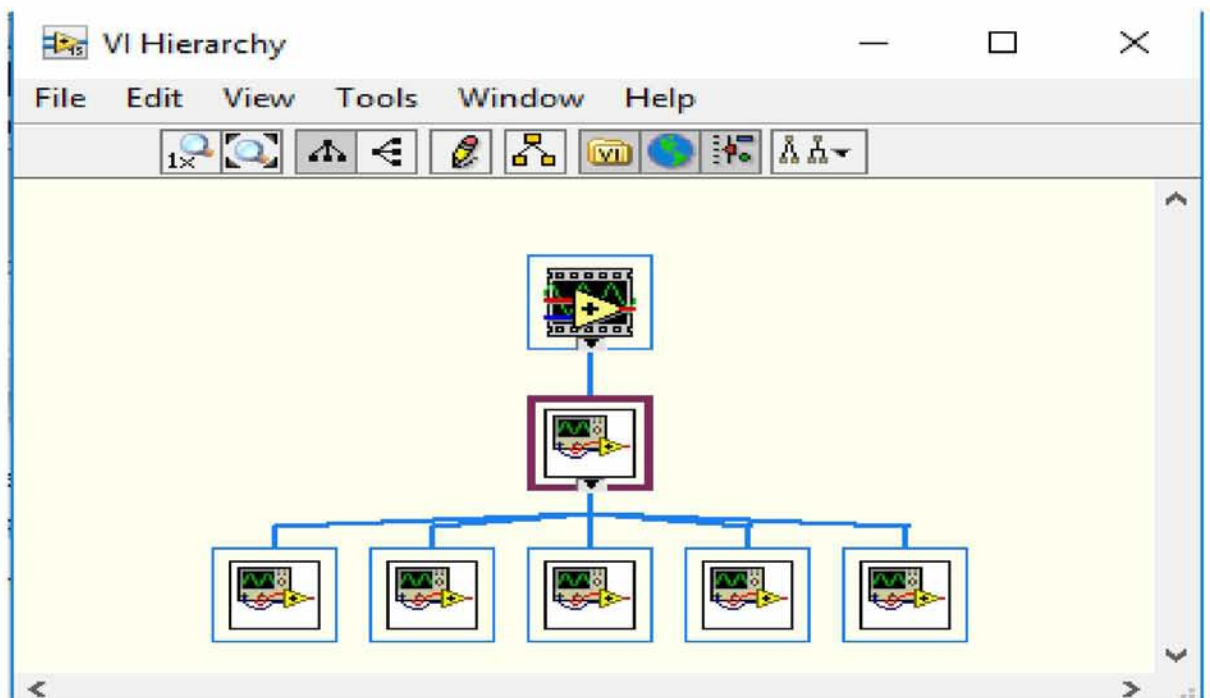
2. Στο Block Diagram κάνουμε τα εξής:

- Δεξί κλικ -> Select a VI -> LabVIEW 2015/instr.lib ->E4438C.llb -> Initialize.vi. (Χρησιμοποιούμε δηλαδή το Initialize.vi ως subvi στο E4438C.vi).
- Δεξί κλικ -> Select a VI -> LabVIEW 2015/instr.lib ->E4438C.llb ->RFout.vi. (Χρησιμοποιούμε δηλαδή το RFout.vi.vi ως subvi E4438C.vi).
- Δεξί κλικ -> Select a VI -> LabVIEW 2015/instr.lib -> E4438C.llb >>SetFreq.vi. (Χρησιμοποιούμε δηλαδή το SetFreq.vi ως subvi στο E4438C.vi).
- Δεξί κλικ -> Select a VI -> LabVIEW 2015/instr.lib ->E4438C.llb ->SetLevel.vi. (Χρησιμοποιούμε δηλαδή το SetLevel.vi.vi ως subvi στο E4438C.vi).
- Δεξί κλικ -> Select a VI -> LabVIEW 2015/instr.lib ->E4438C.llb ->Close.vi. (Χρησιμοποιούμε δηλαδή το Close.vi ως subvi στο E4438C.vi).

Στη συνέχεια ενώνουμε τα τερματικά σύμφωνα με το σχήμα 5.7.2. Σε αυτήν την περίπτωση φαίνεται η χρησιμότητα των connectors που θέταμε καθ' όλη τη διάρκεια της δημιουργίας των συνι's. Συνεπώς είναι πολύ εύκολο να καταλάβει κανείς ότι το τερματικό με το όνομα Frequency θα συνδεθεί στον ακροδέκτη με το όνομα Frequency του subvi SetFreq. Εντελώς ανάλογα θα συνδεθεί και το τερματικό Amplitude στον αντίστοιχο ομώνυμο ακροδέκτη του subvi SetLevel. Τέλος από τη γραμμή εργαλείων του Front Panel (ή και του Block Diagram) του E4438C.vi επιλέγοντας Browse >> VI Hierarchy θα εμφανιστεί το διάγραμμα του σχήματος 5.7.3 στο οποίο φαίνεται ότι στο πρώτο επίπεδο έχουμε το κορυφαίο E4438C.vi και σε δεύτερο επίπεδο (δηλαδή subvi's) τα Initialize, RFout, SetLevel, SetFreq και close.



Σχήμα 5.7.2 Block Diagram με τα subvi's



Σχήμα 5.7.3 Ιεραρχία των VI's.

5.8 Λειτουργίες που θέλουμε να επιτελέσει η γεννήτρια(Signal Generator) με την χρήση έτοιμων συναρτήσεων

Το LabVIEW διαθέτει έναν αριθμό από έτοιμα VIs και ορισμένα εικονίδια συναρτήσεων που επιτρέπουν την επικοινωνία με όλα τα γνωστά πρωτόκολλα μετάδοσης δεδομένων. Έτσι, υπάρχουν έτοιμες λειτουργίες που επιτρέπουν τη συλλογή και μετάδοση δεδομένων μέσω της σειριακής θύρας του υπολογιστή, καθώς και μέσω της παράλληλης θύρας. Επίσης, το πρόγραμμα για τον έλεγχο της γεννήτριας δίνεται από την κατασκευάστρια εταιρία του LabView. Αγνοώντας το Initialize.vi και το Close.vi-των οποίων η χρησιμότητα όπως είπαμε είναι το άνοιγμα του διαδρόμου επικοινωνίας και το κλείσιμο αυτού αντίστοιχα-εστιάζουμε το ενδιαφέρον μας στο Configure.vi.(όπου περιέχει τις συναρτήσεις για το FM και το AM). Παρατηρώντας το Block diagram θα δούμε ότι αποτελείται από 3 subvi's. Αυτά είναι:

- 1) Initialize.vi
- 2) Configure.vi
- 3) Close.vi

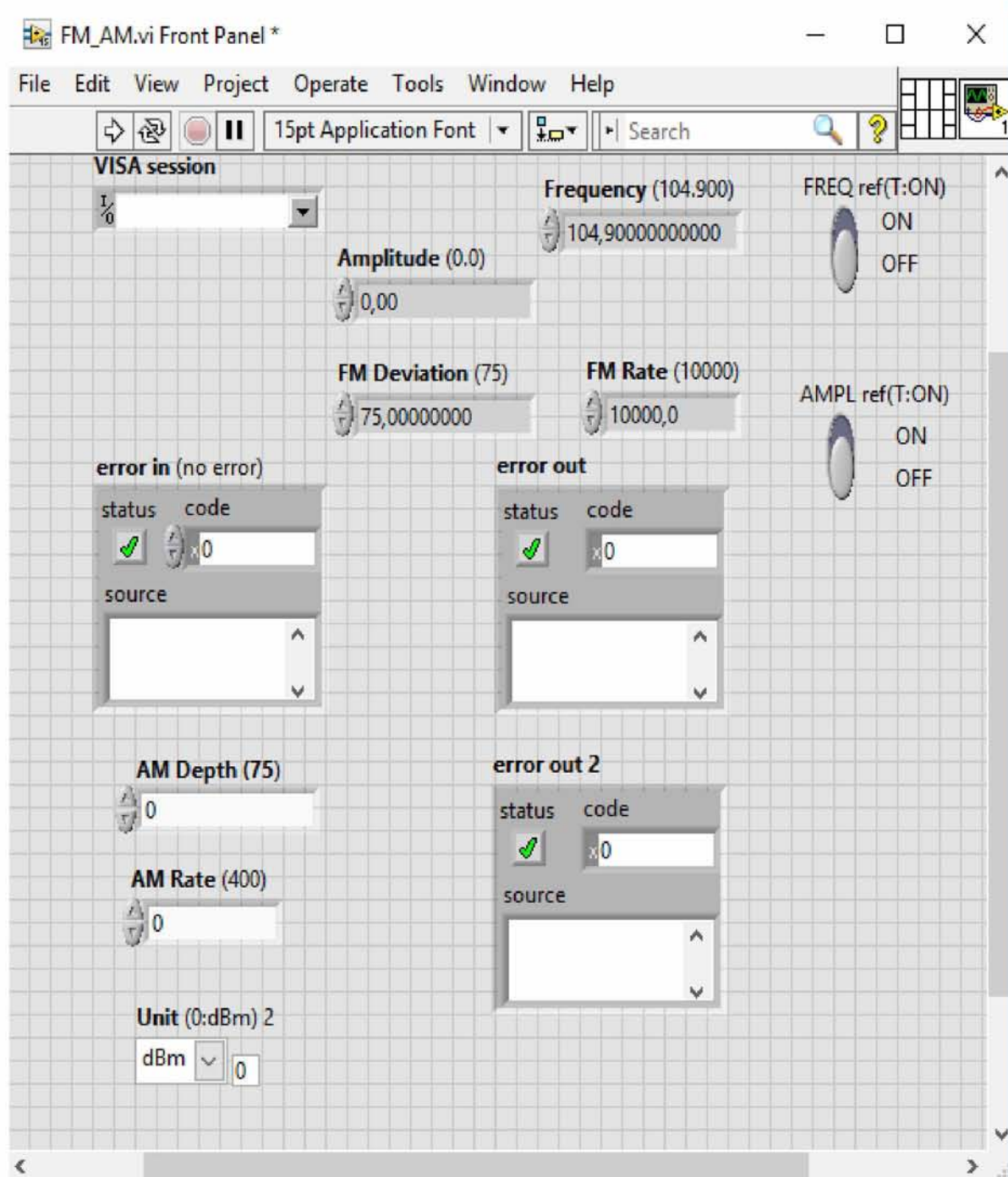
του E4438C. Το Front Panel και το Block Diagram για το FM-AM φαίνονται στα παρακάτω Σχήματα(5.8.1 - 5.8.5).

Σε γενικές γραμμές σε μια γεννήτρια συχνοτήτων θα θέλουμε να κάνουμε τα εξής:

1. Να δίνεται η δυνατότητα καθορισμού του πλάτους του σήματος.
2. Να δίνεται η δυνατότητα καθορισμού της συχνότητας του σήματος.
3. Να δίνεται η δυνατότητα ενεργοποίησης της εξόδου.

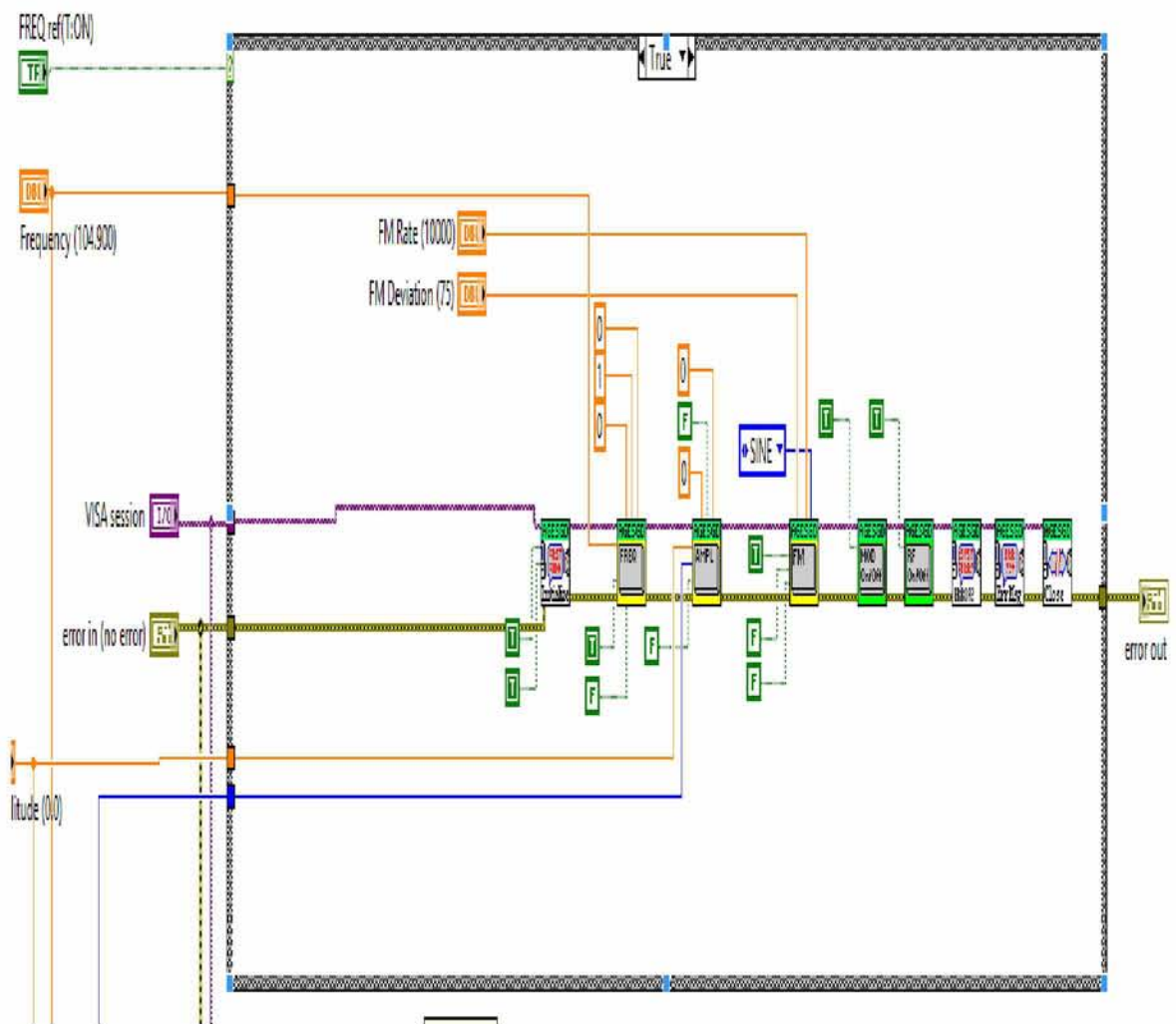
Δημιουργία του κορυφαίου FM-AM.vi με τις έτοιμες συναρτήσεις του LabVIEW.

1. Στο Front Panel για το project μας FM-AM τοποθετούμε τα controls αλλά και τα indicators σύμφωνα με όσα περιγράψαμε μέχρι τώρα και σύμφωνα με το Σχήμα 5.8.1

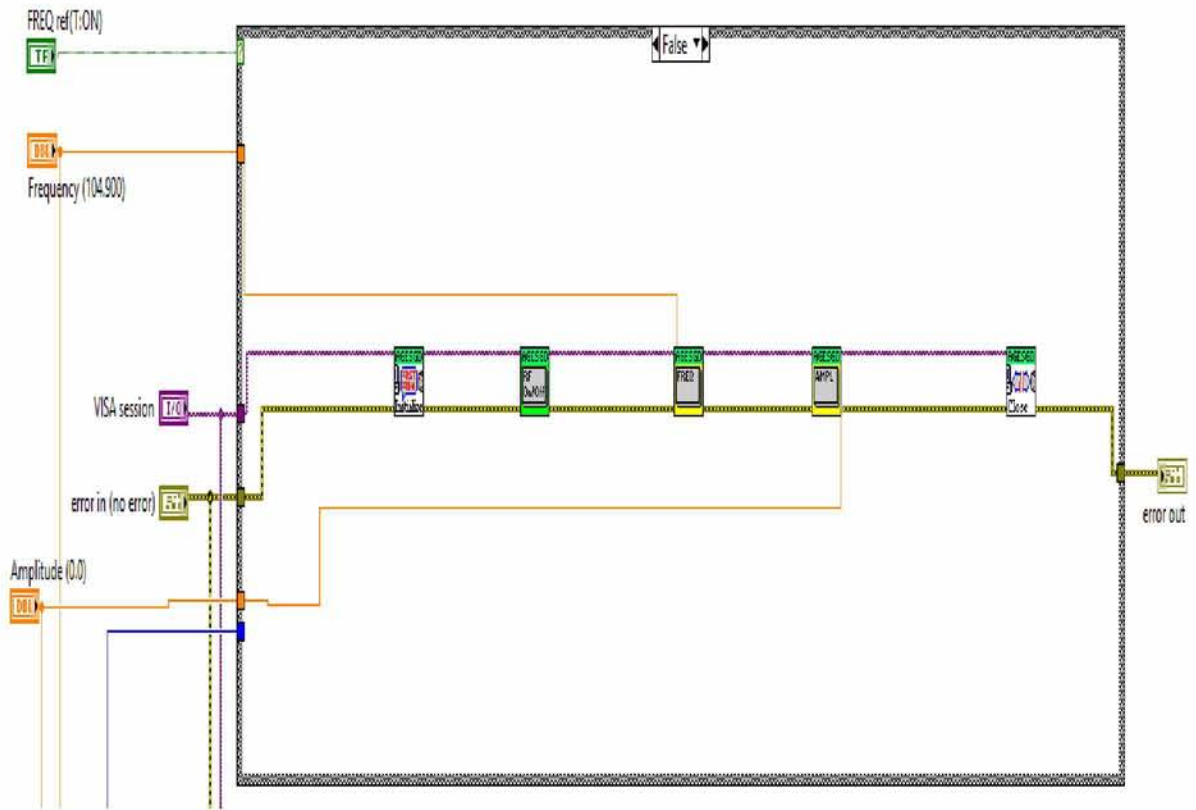


Σχήμα 5.8.1 Front Panel του FM-AM.vi

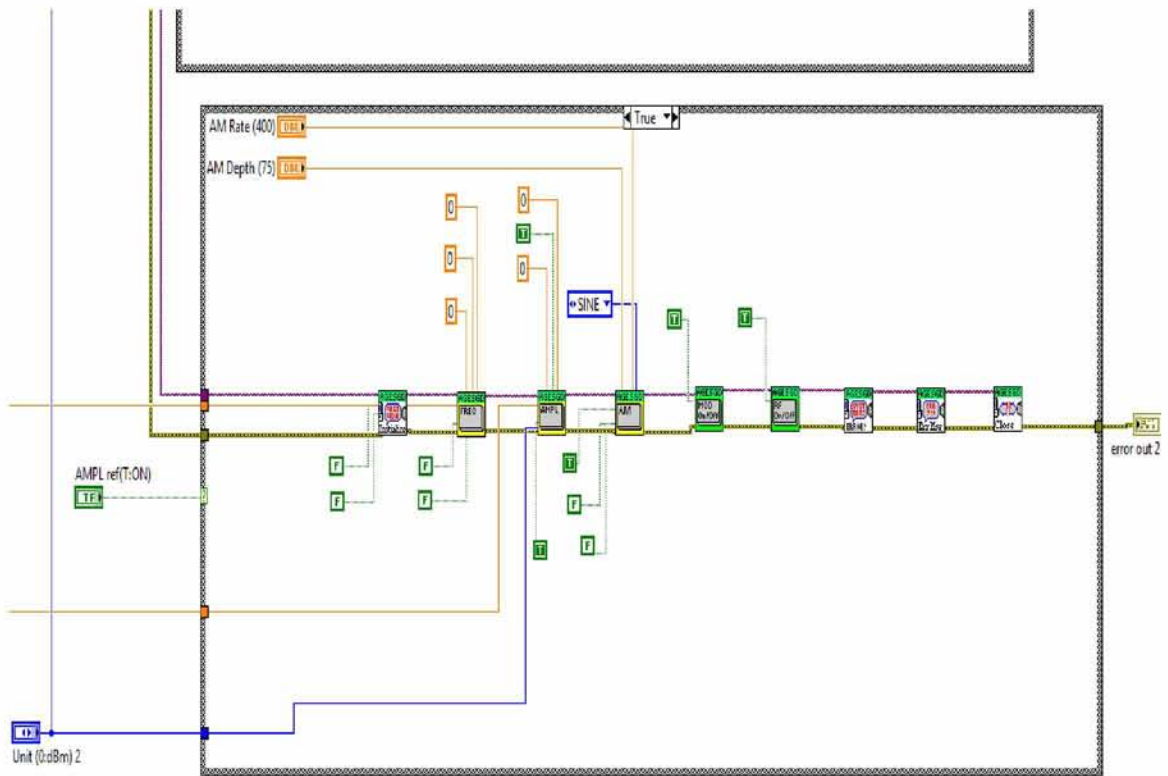
2. Στο Block Diagram για το project μας FM-AM η διαφορά με πριν είναι ότι χρησιμοποιούμε τους έτοιμους drivers σύμφωνα με το Σχήματα 5.8.2-5.8.3-5.8.4-5.8.5.



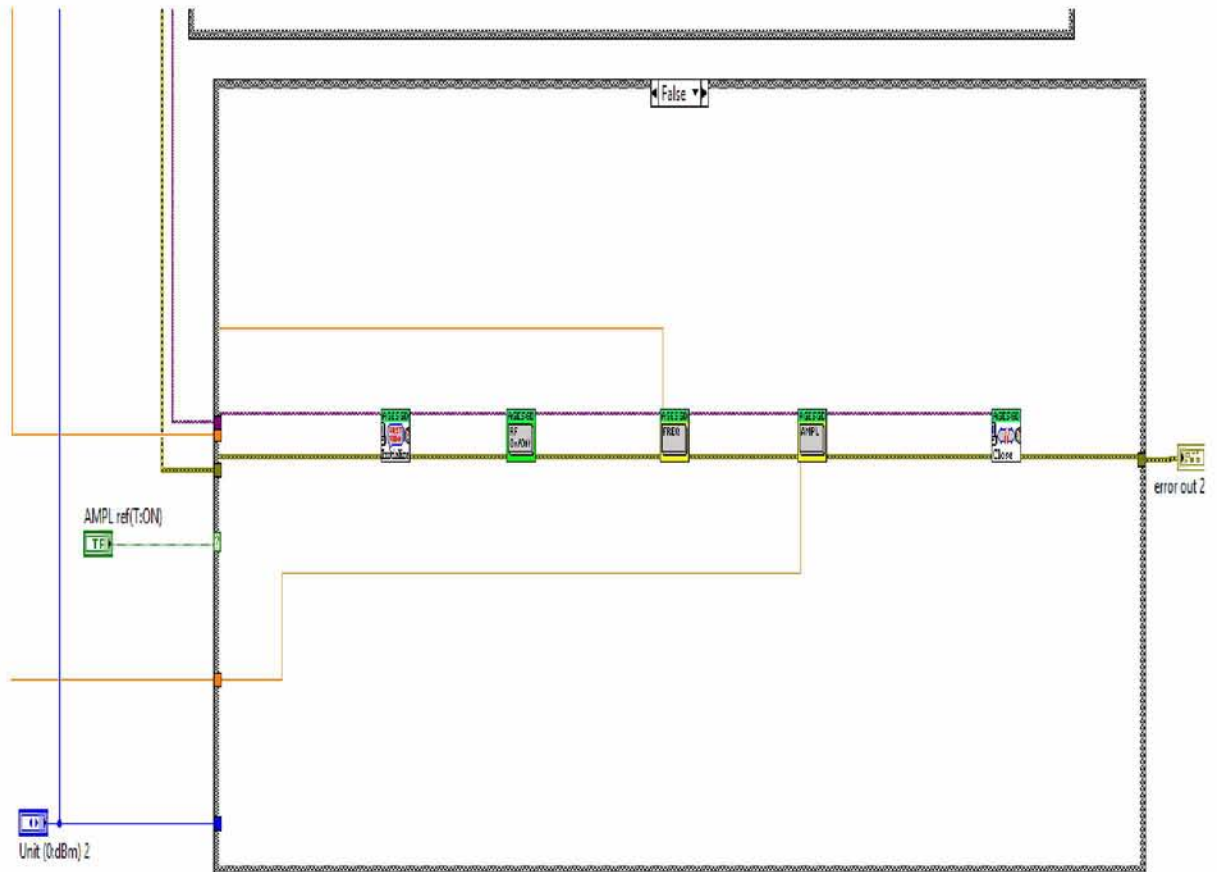
Σχήμα 5.8.2 Block Diagram για Freq=True



Σχήμα 5.8.3 Block Diagram για Freq=False



Σχήμα 5.8.4 Block Diagram για Ampl=True



Σχήμα 5.8.5 Block Diagram για Ampl = False

5.9 Αποτελέσματα γεννήτριας(E4438C)

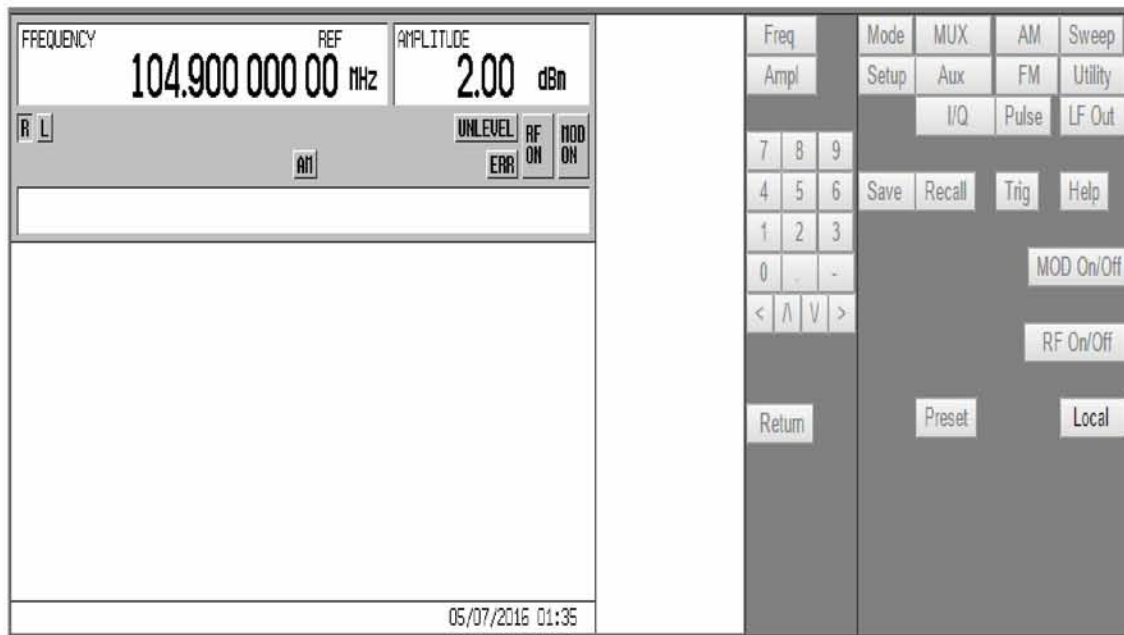
Τα αποτελέσματα που θα δούμε στην γεννήτρια μας είναι τα εξής:

- Αν εκτελέσουμε για Freq = true έχουμε το εξής αποτέλεσμα στην γεννήτρια μας (Σχήμα 5.9.1).

FREQUENCY		REF	AMPLITUDE		Freq	Mode	MUX	AM	Sweep			
104.900 000 00		MHz	0.00		dBm	Ampl	Setup	Aux	FM	Utility		
R	L	UNLEVEL		RF ON	T:ON	I/Q		Pulse	LF Out			
Fill						7	8	9	Save	Recall	Trig	Help
						4	5	6				
						1	2	3				
						0	.	-	MOD On/Off			
						<	∧	>	RF On/Off			
						Return		Preset		Local		
05/07/2016 01:30												

Σχήμα 5.9.1 Freq = True

- Αν εκτελέσουμε για Ampl = true έχουμε το εξής αποτέλεσμα στην γεννήτρια μας(Σχήμα5.9.2).



Σχήμα 5.9.2 Ampl = True

ΚΕΦΑΛΑΙΟ 6

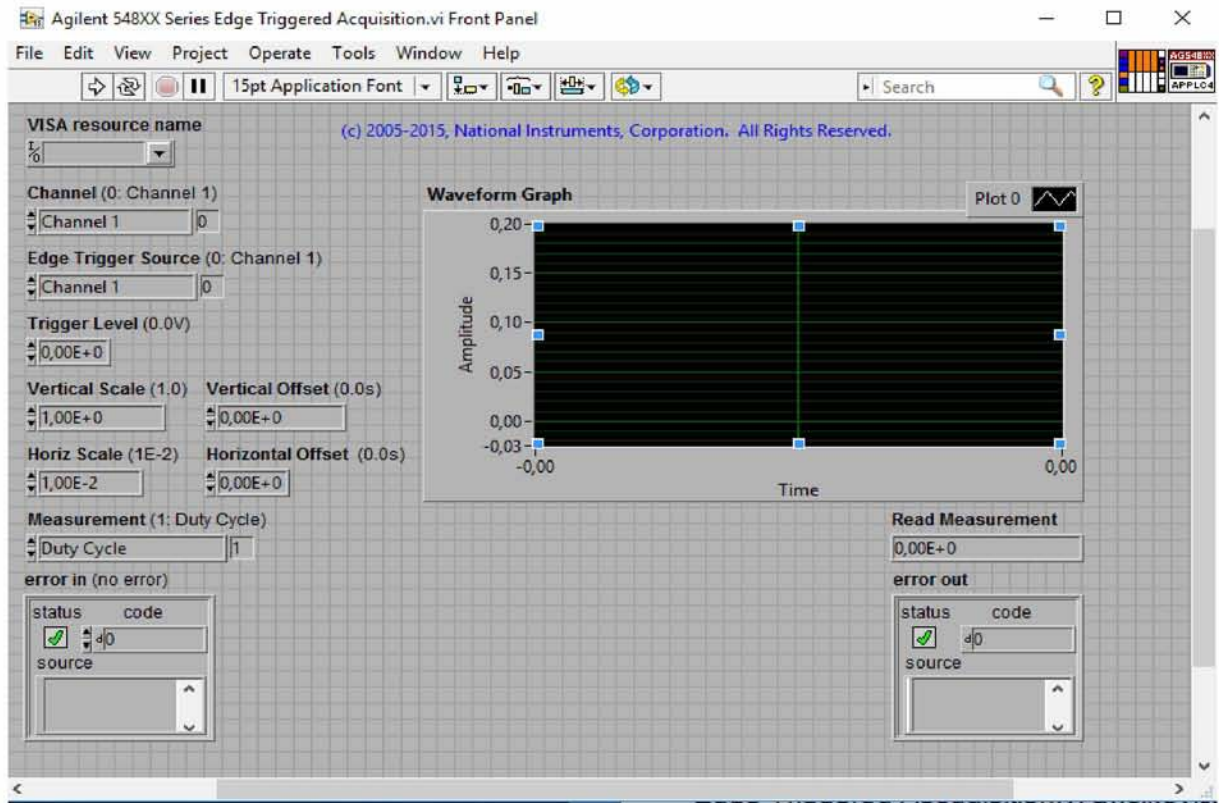
Το πρόγραμμα ελέγχου του παλμογράφου

6.1 Λειτουργίες που θέλουμε να επιτελέσει ο παλμογράφος (Oscilloscope)

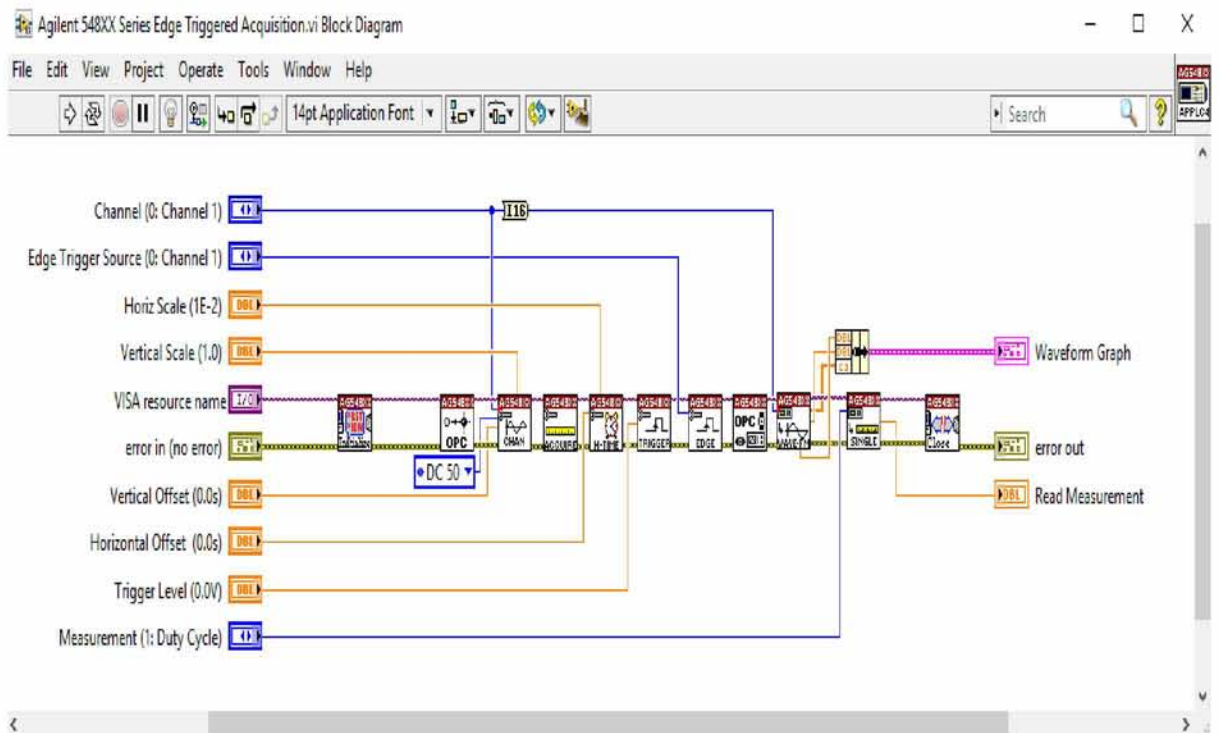
Το πρόγραμμα για τον έλεγχο του παλμογράφου δίνεται από την κατασκευάστρια εταιρία του LabVIEW και βρίσκεται στην τοποθεσία: C:\Program Files\National Instruments\LabVIEW2015\instr.lib\Agilent 548XX Series. Το παρακάτω vi είναι μια απλή εφαρμογή και παρουσιάζει πώς μπορεί να χρησιμοποιηθεί αυτός ο driver. Στην ουσία παίρνει μετρήσεις στην θετική ή στην αρνητική ακμή του ρολογιού (edge trigger), καθορίζει τις οριζόντιες και κατακόρυφες κλίμακες και μετατοπίσεις έτσι ώστε να σχηματιστεί η κυματομορφή και να απεικονίσει μια μέτρηση. Το Agilent 54831B-Series EdgeTriggeredAcquisition.vi είναι ένα vi της βιβλιοθήκης και παρατηρώντας το block diagram θα δούμε ότι αποτελείται από subvi's. Αυτά είναι:

- 1) Initialize.vi
- 2) Actn Begin Wait for OP Comp.vi
- 3) Conf Channel.vi
- 4) Conf Measurement Acquisition.vi
- 5) Conf Horz Timebase.vi
- 6) Conf Trigger.vi
- 7) Conf Trigger (Edge).vi
- 8) Actn End Wait for OP Comp.vi
- 9) Data Read Waveform.vi
- 10) Data Read Meas (Single).vi
- 11) Close.vi

Αγνοώντας το Initialize.vi και το Close.vi-των οποίων η χρησιμότητα όπως είπαμε είναι το άνοιγμα του διαδρόμου επικοινωνίας και το κλείσιμο αυτού αντίστοιχα-εστιάζουμε το ενδιαφέρον μας στα υπόλοιπα vi τα οποία θα αναλύσουμε παρακάτω. Το Front Panel και το Block Diagram του Agilent 54831B-SeriesEdgeTriggeredAcquisition.vi φαίνονται στα παρακάτω σχήματα. Το front panel απεικονίζεται στο σχήμα6.1.1 ενώ το block diagram απεικονίζεται στο σχήμα 6.1.2.



Σχήμα 6.1.1 Front Panel του Agilent 54831B- Series Edge Triggered Acquisition.vi

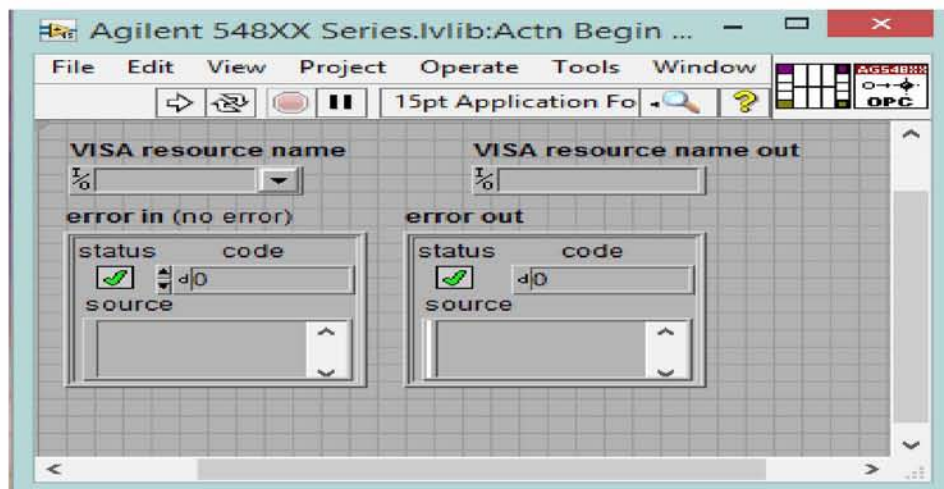


Σχήμα 6.1.2 Block Diagram του Agilent 54831B- Series Edge Triggered Acquisition.vi

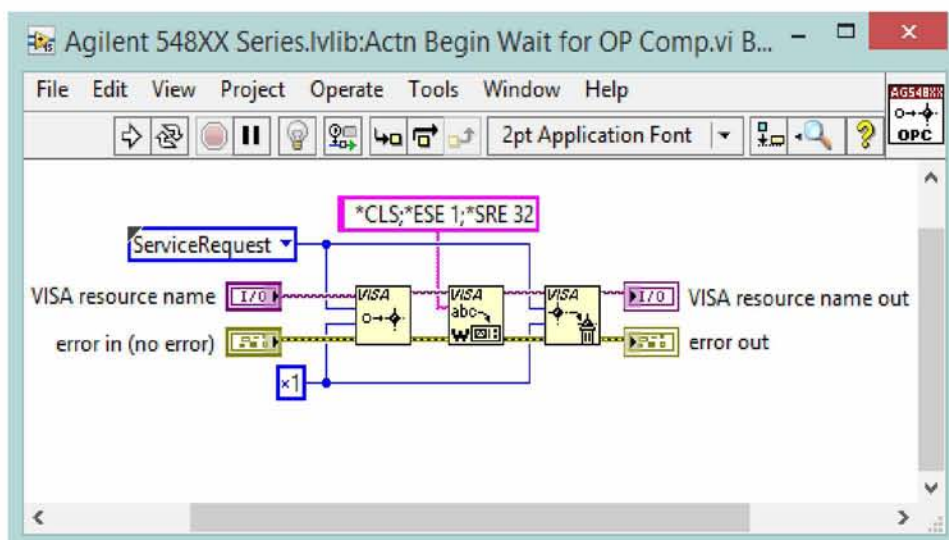
6.2 Ανάλυση των επιμέρους VI

➤ Actn Begin Wait for OP Comp.vi

Αυτό το vi επιτρέπει το συγχρονισμό του οργάνου ενεργοποιώντας το service request και χρησιμοποιείται σε συνδυασμό με το ActnEndWaitforOPComp.vi. Στα σχήματα που ακολουθούν διακρίνονται το Front Panel (σχήμα 6.2.1) και το Block Diagram (σχήμα 6.2.2).



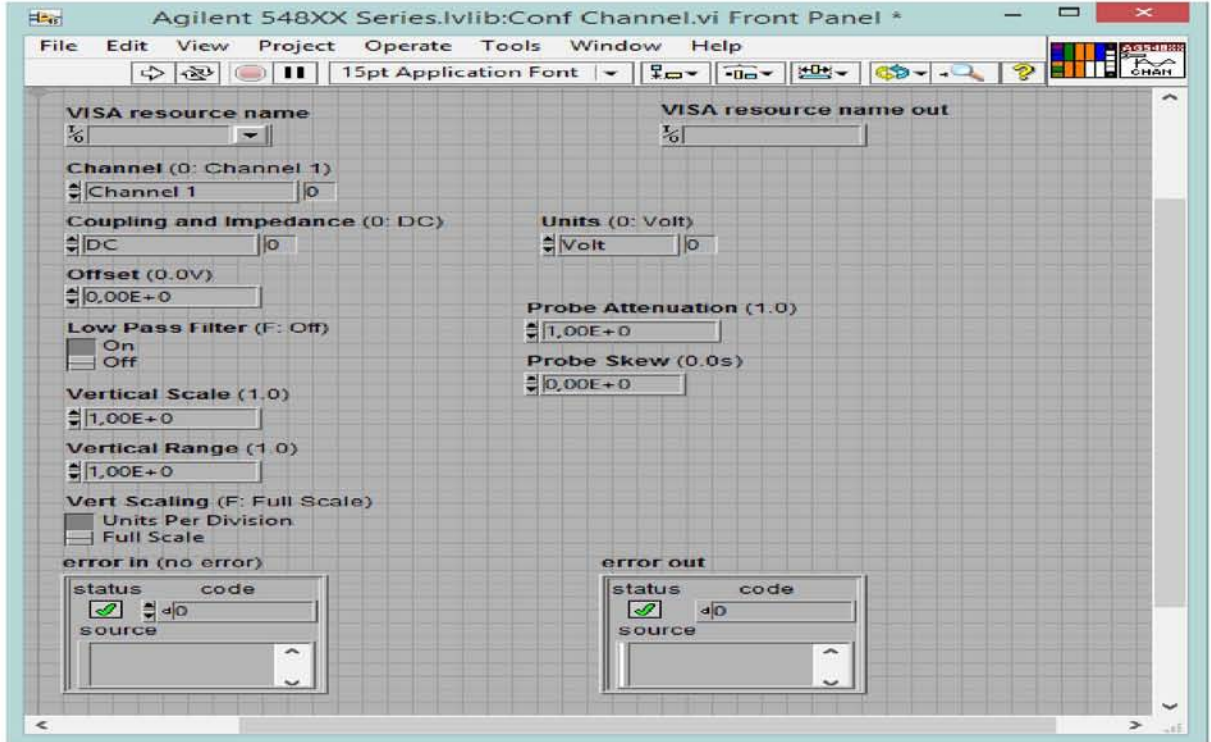
Σχήμα 6.2.1 Front Panel του Actn Begin Wait for OP Comp.vi



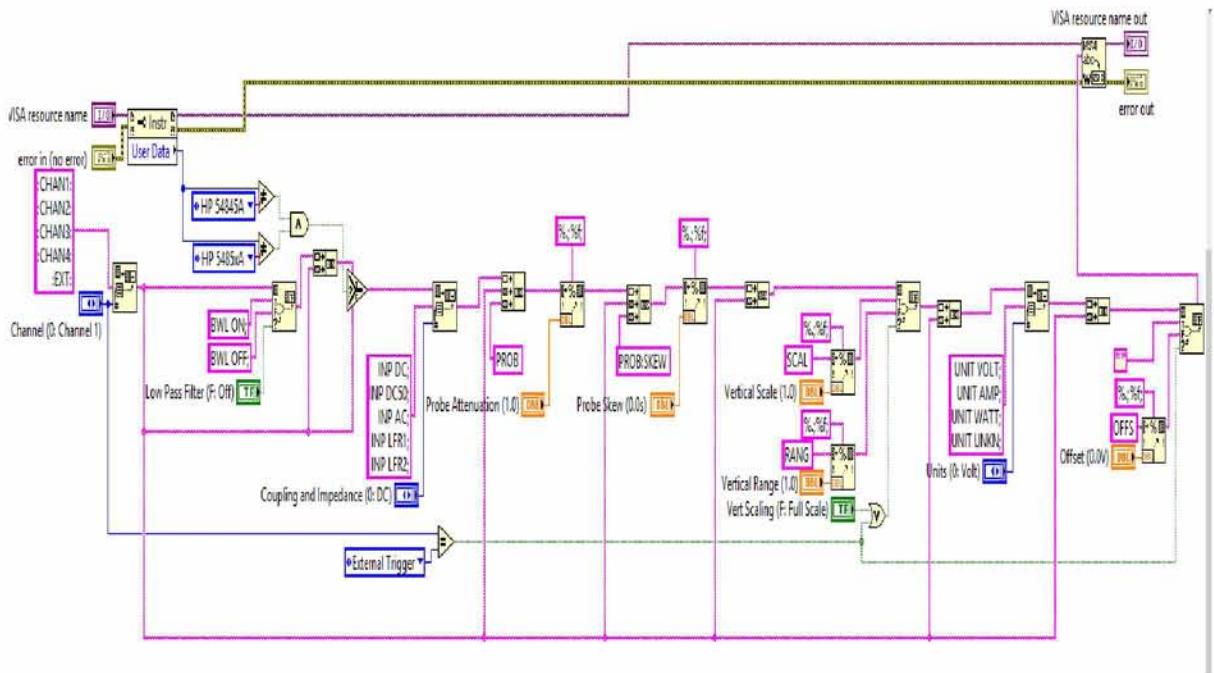
Σχήμα 6.2.2 Block Diagram του Actn Begin Wait for OP Comp.vi

➤ **Conf Channel.vi**

Αυτό το vi ρυθμίζει όλες τις συναρτήσεις των καναλιών του παλμογράφου στον άξονα Υ. Όπως για παράδειγμα τα coupling, offset και scaling.



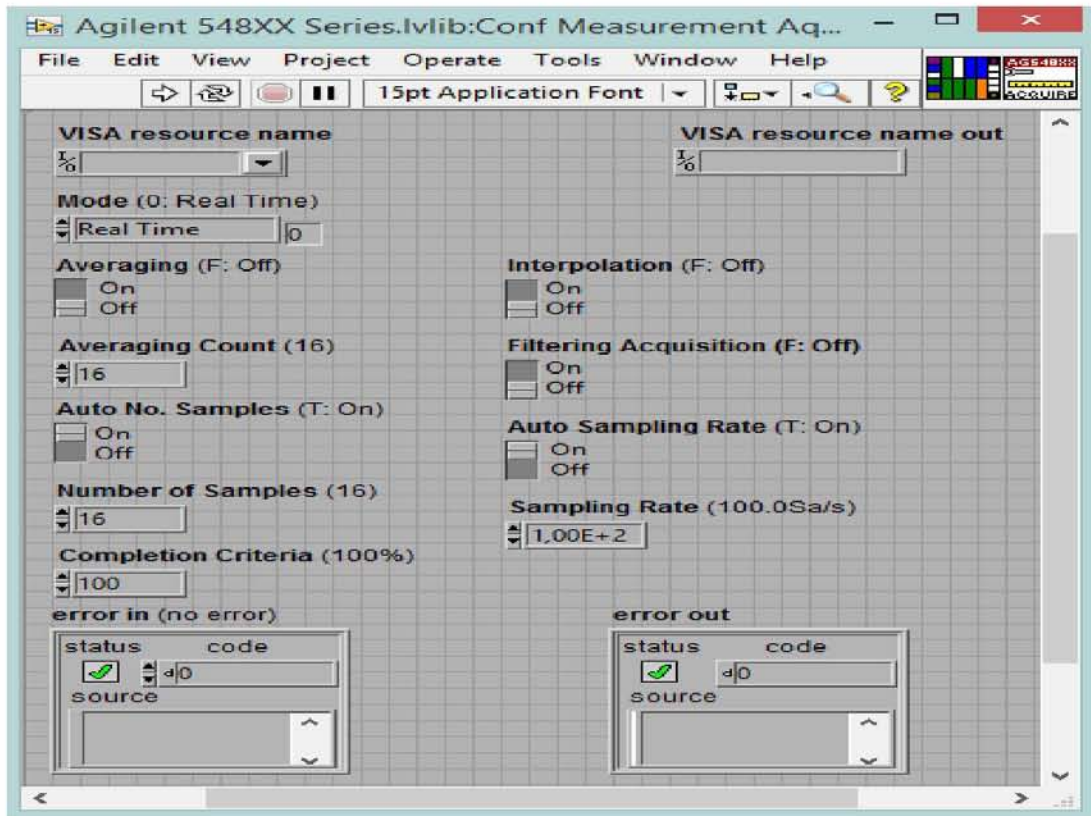
Σχήμα 6.2.3 Front Panel του Conf Channel.vi



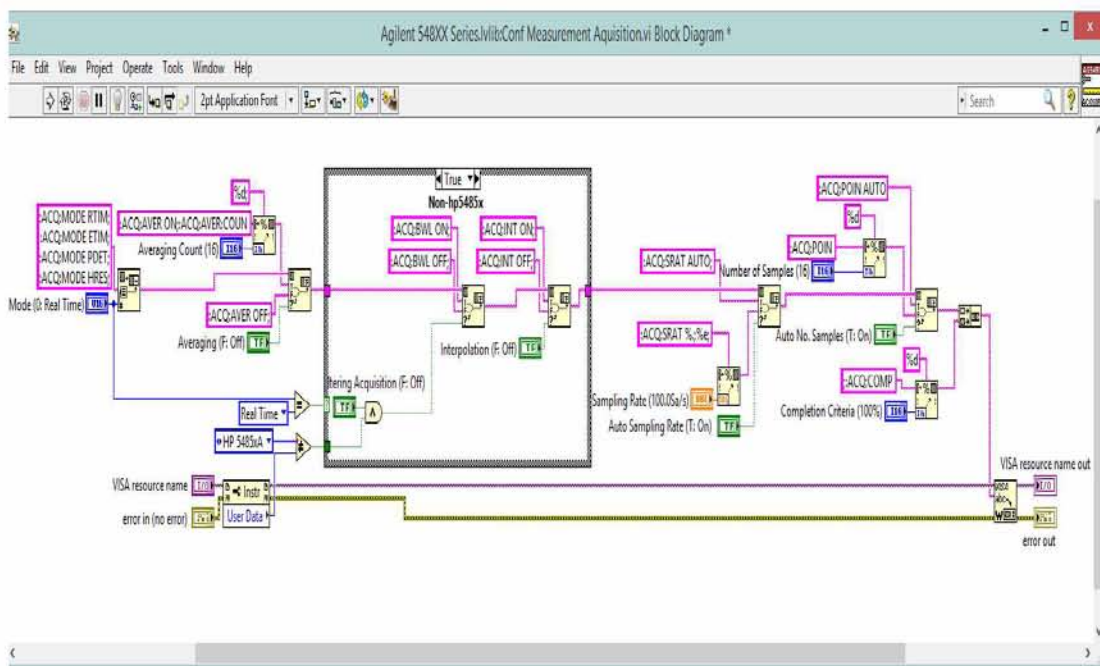
Σχήμα 6.2.4 Block Diagram του Conf Channel.vi

➤ Conf Measurement Acquisition.vi

Αυτό το vi καθορίζει τον τρόπο απόκτησης των μετρήσεων από τον παλμογράφο.



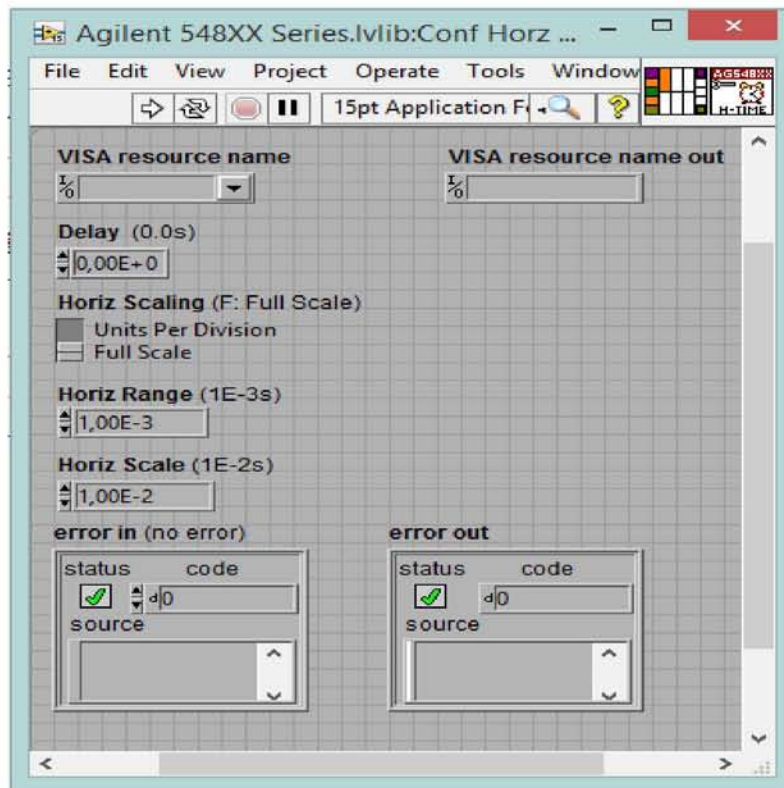
6.2.5 Front Panel του Conf Measurement Acquisition.vi



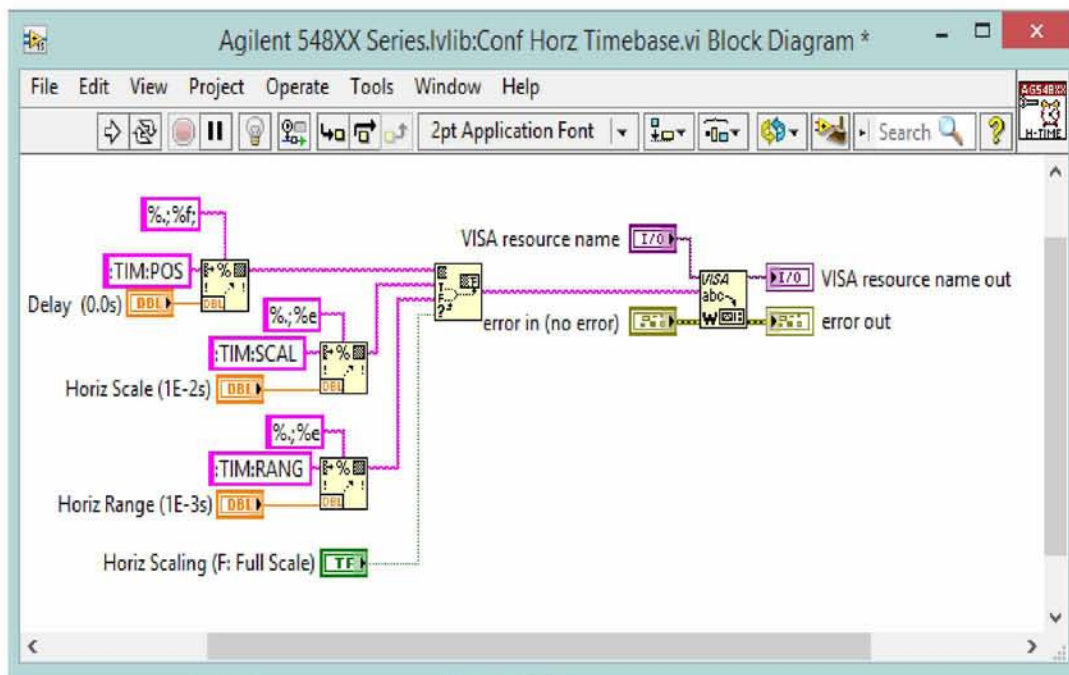
6.2.6 Block Diagram του Conf Measurement Acquisition.vi

➤ Conf Horz Timebase.vi

Αυτό το νί ρυθμίζει τις συναρτήσεις στον άξονα Χ. Όπως για παράδειγμα το delay και το horizontal scale.



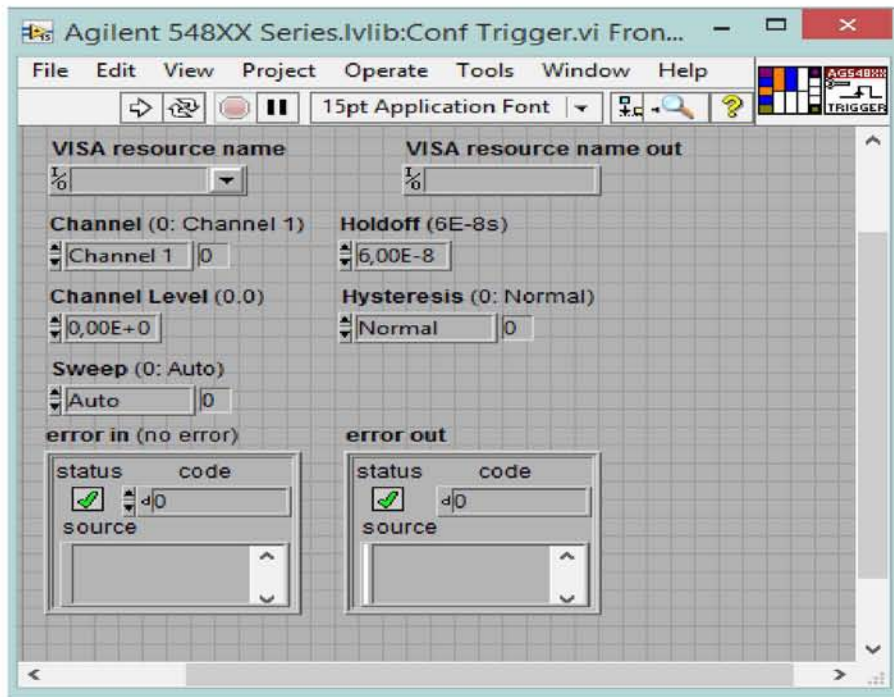
6.2.7 Front Panel του Conf Horz Timebase.vi



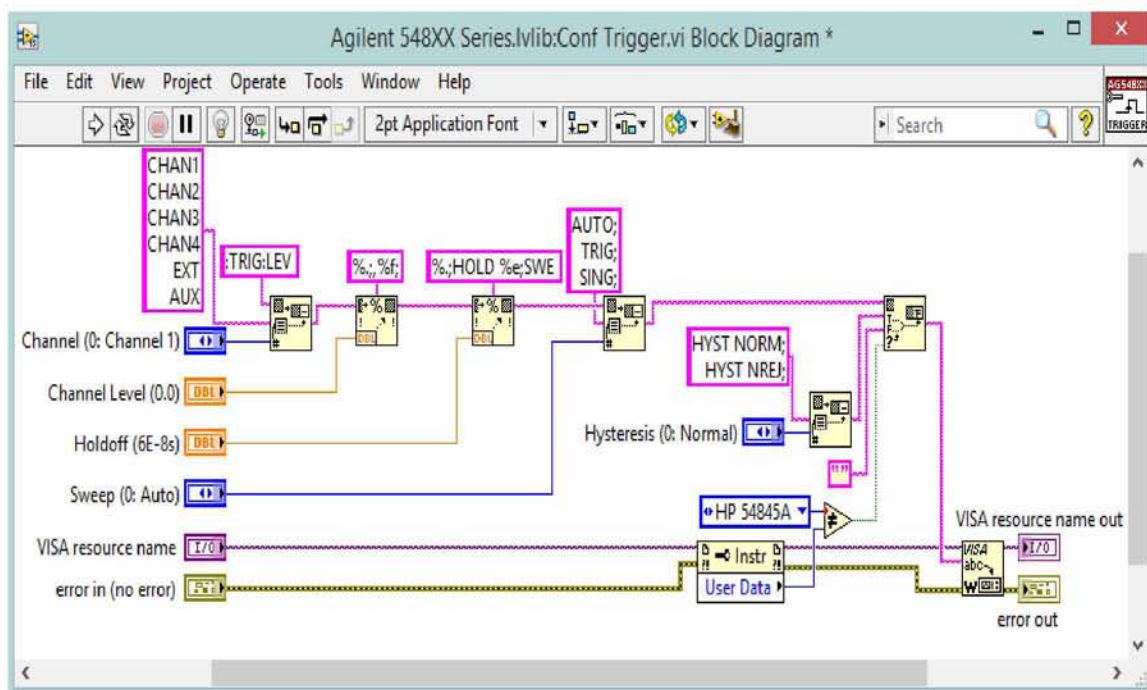
6.2.8 Block Diagram του Conf Horz Timebase.vi

➤ Conf Trigger.vi

Αυτό το vi διαμορφώνει τις γενικές παραμέτρους για το trigger.



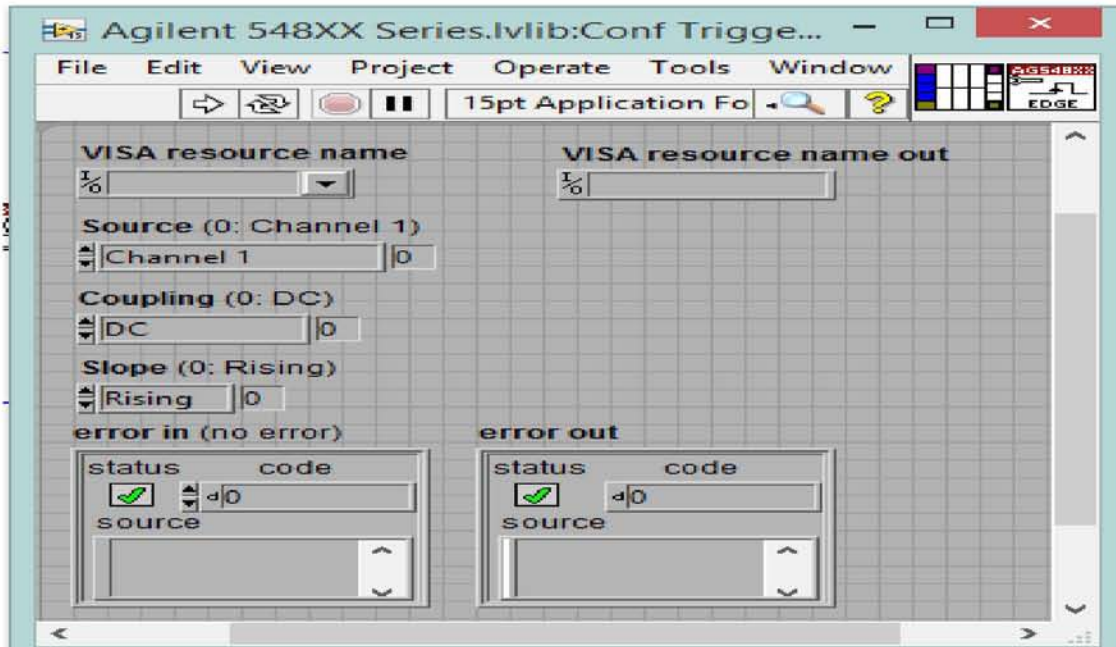
6.2.9 Front Panel του Conf Trigger.vi



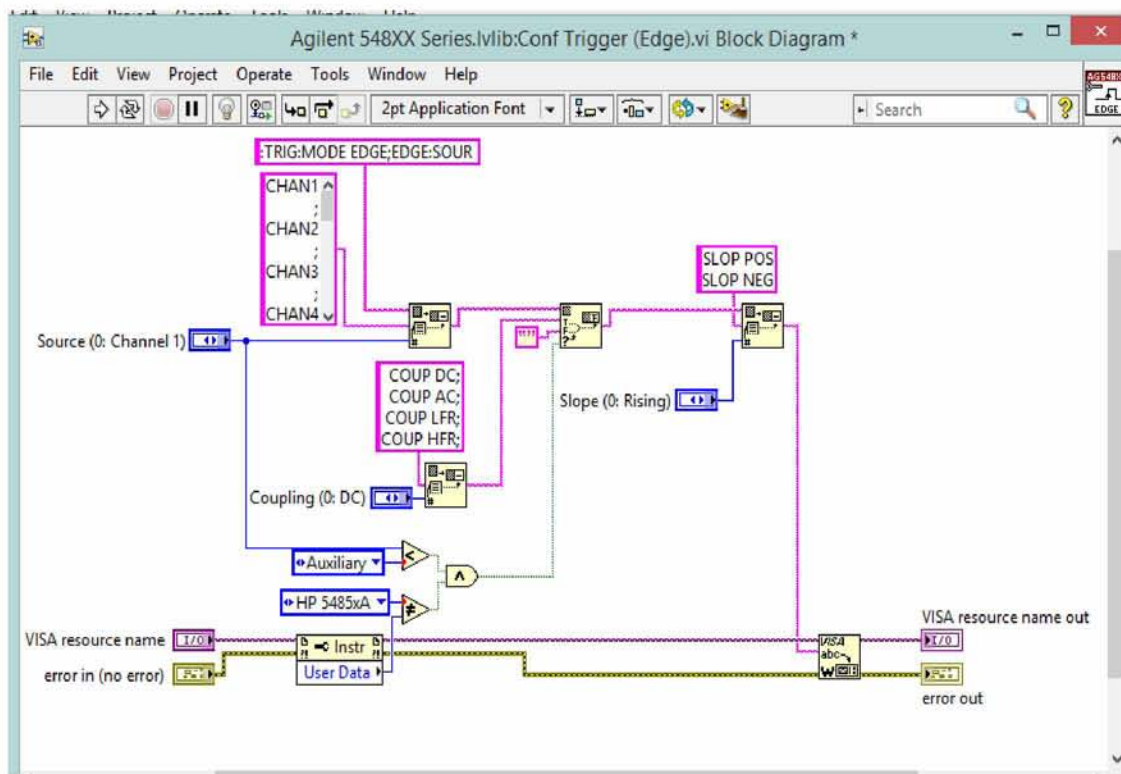
6.2.10 Block Diagram του Conf Trigger.vi

➤ Conf Trigger (Edge).vi

Αυτό το vi διαμορφώνει και ενεργοποιεί το edge trigger.



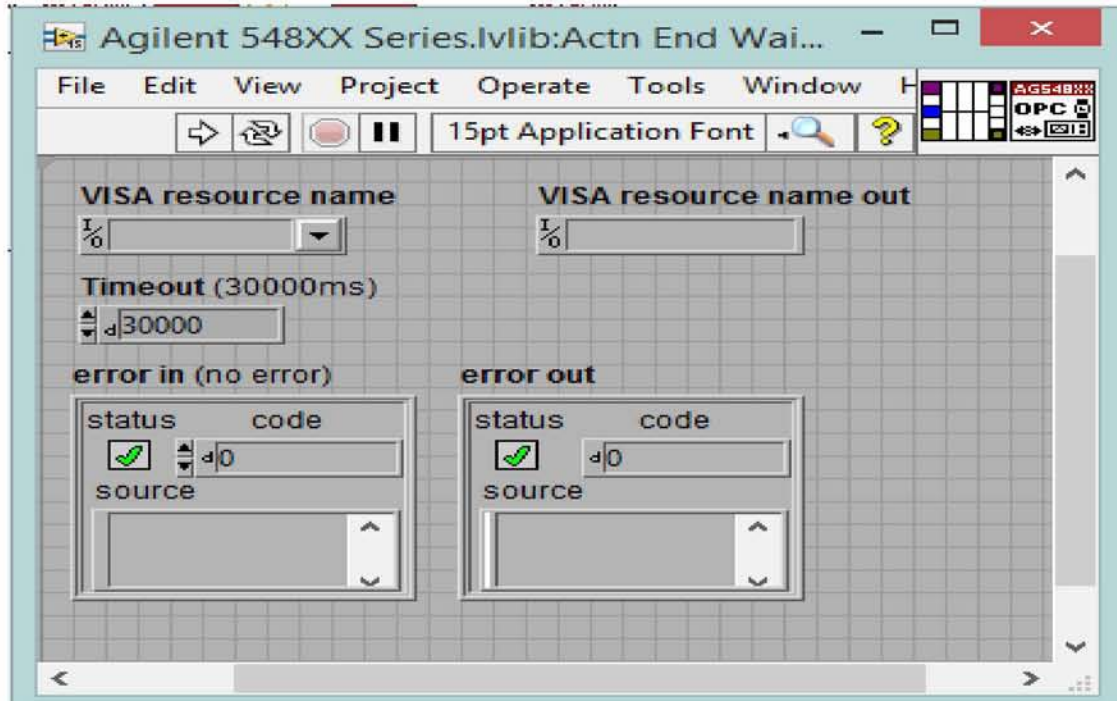
6.2.11 Front Panel του Conf Trigger (Edge).vi



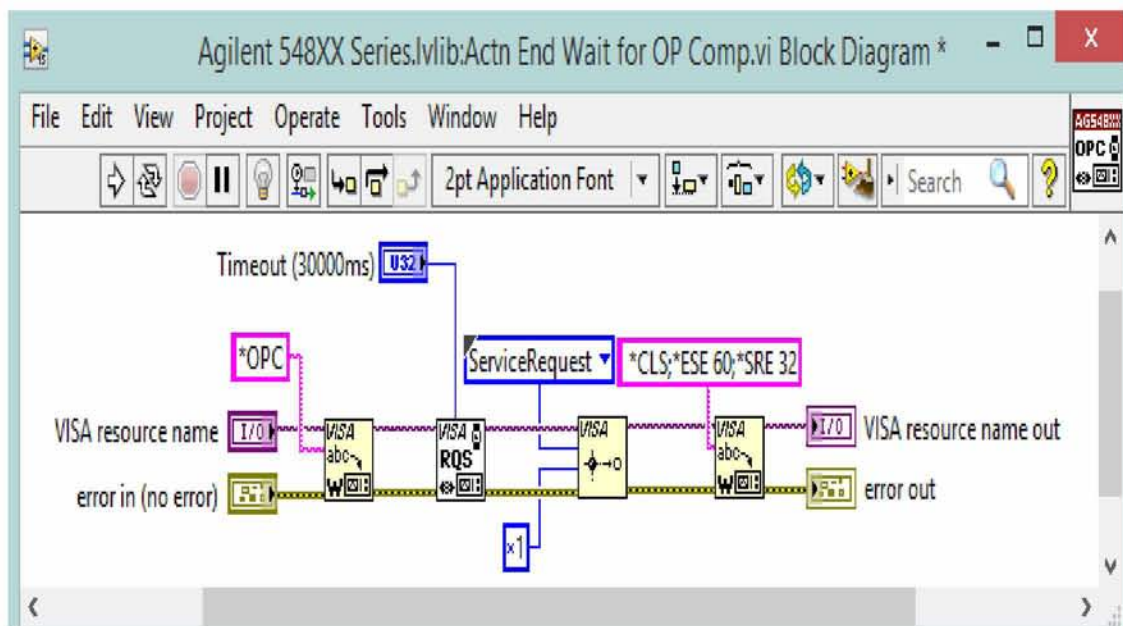
6.2.12 Block Diagram του Conf Trigger (Edge).vi

➤ **Actn End Wait for OP Comp.vi**

Αυτό το vi περιμένει να ολοκληρωθεί η διαδικασία έτσι ώστε να απενεργοποιήσει το service request. Όπως αναφέρθηκε παραπάνω λειτουργεί σε συγχρονισμό με το ActnBeginWaitforOPComp.vi.



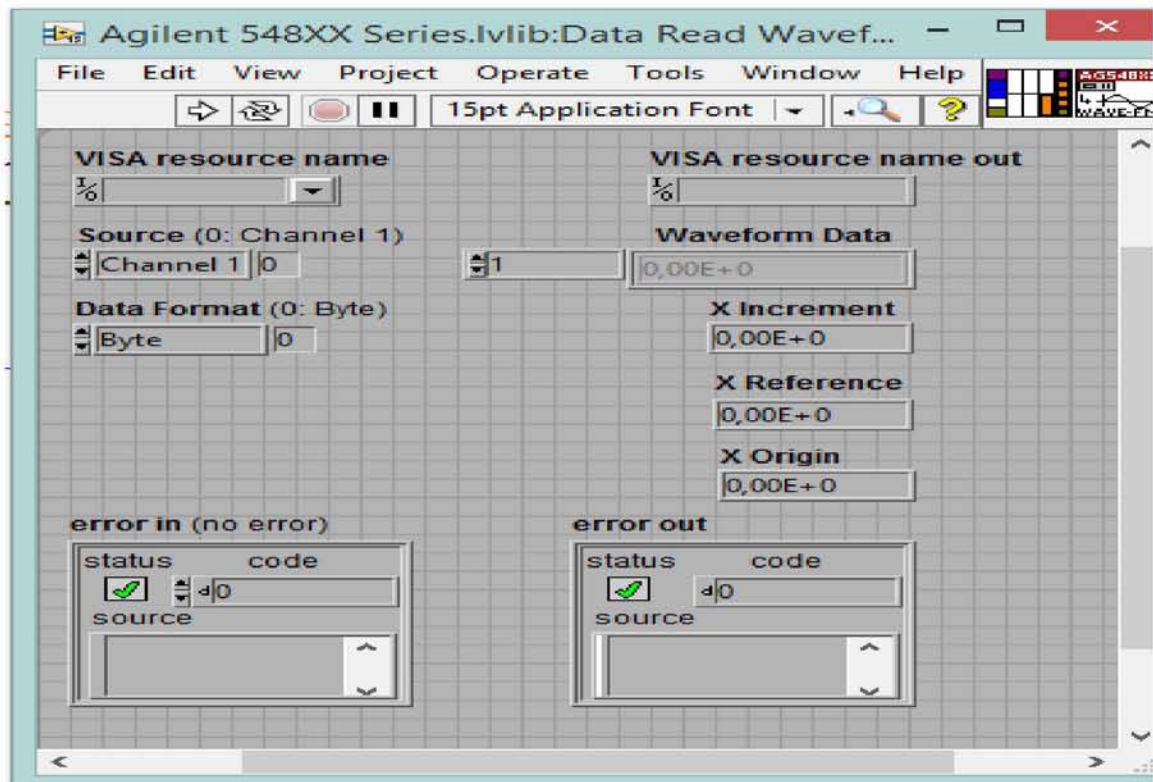
6.2.13 Front Panel του Actn End Wait for OP Comp.vi



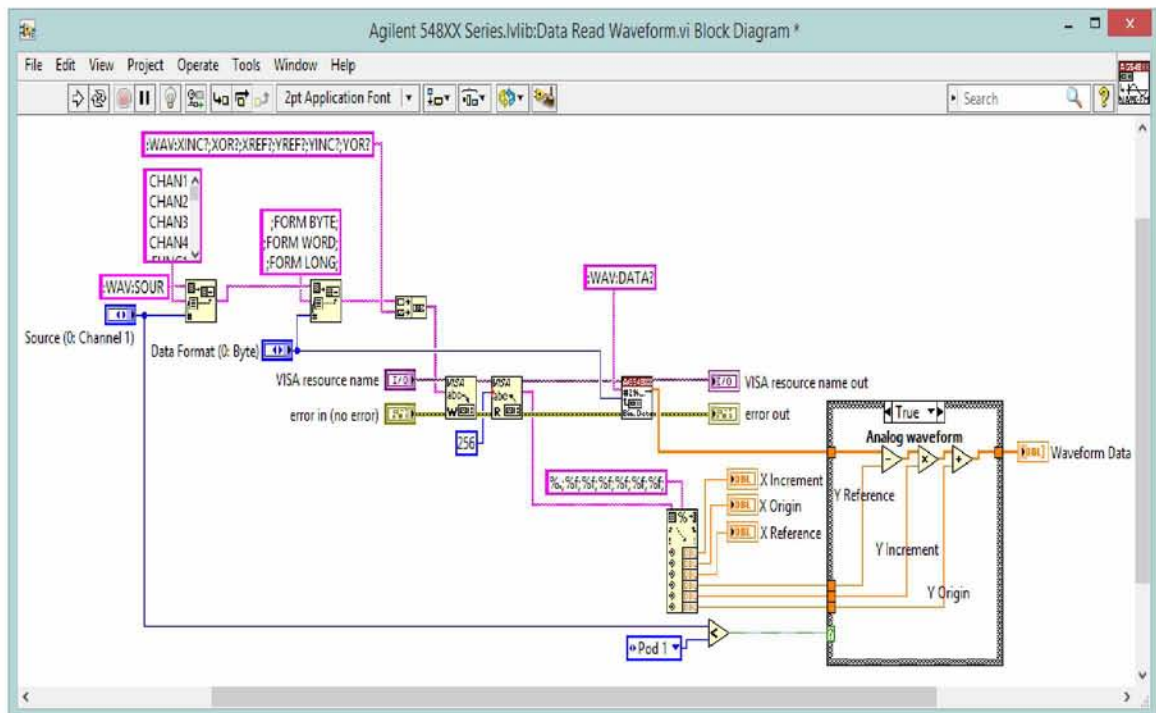
6.2.14 Block Diagram του Actn End Wait for OP Comp.vi

➤ Data Read Waveform.vi

Αυτό το vi διαβάζει την κυματομορφή από τον παλμογράφο.



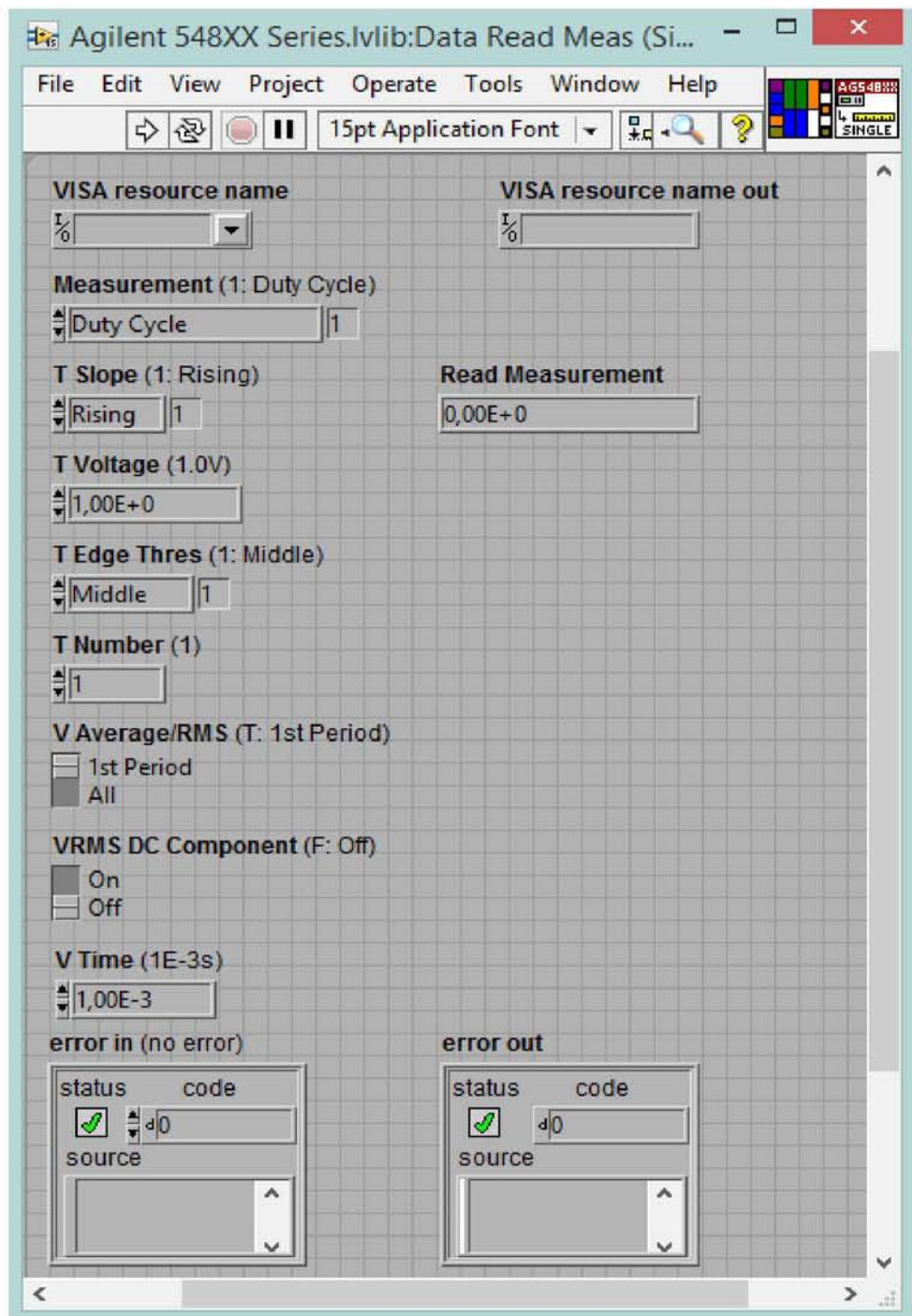
6.2.15 Front Panel του Data Read Waveform.vi



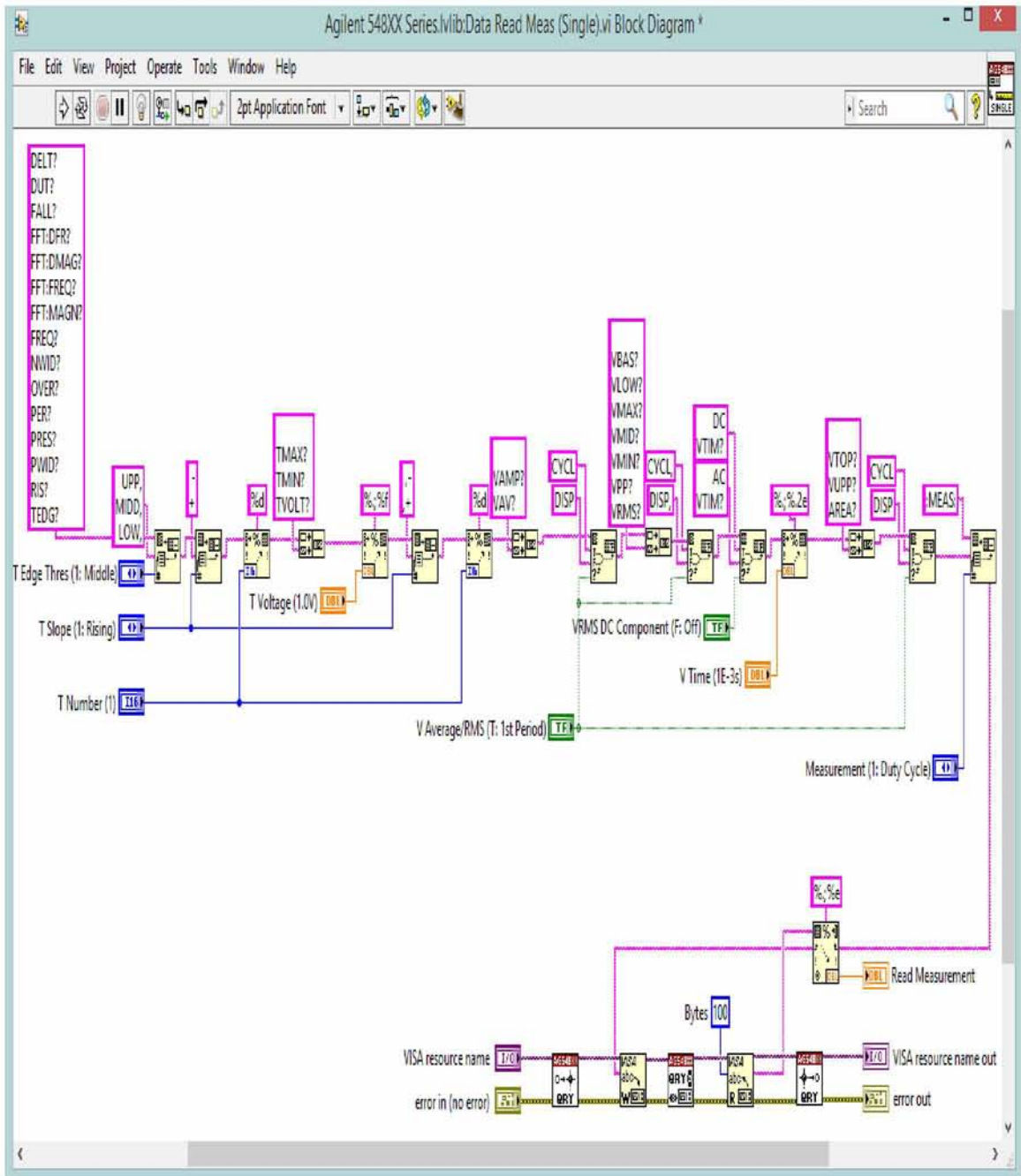
6.2.16 Block Diagram του Data Read Waveform.vi

➤ **Data Read Meas (Single).vi**

Αυτό το vi επιστρέφει μια συγκεκριμένη μέτρηση που έχει ρυθμιστεί από το ConfMeasurement.vi.



6.2.17 Front Panel του Data Read Meas (Single).vi



6.2.18 Block Diagram of Data Read Meas (Single).vi

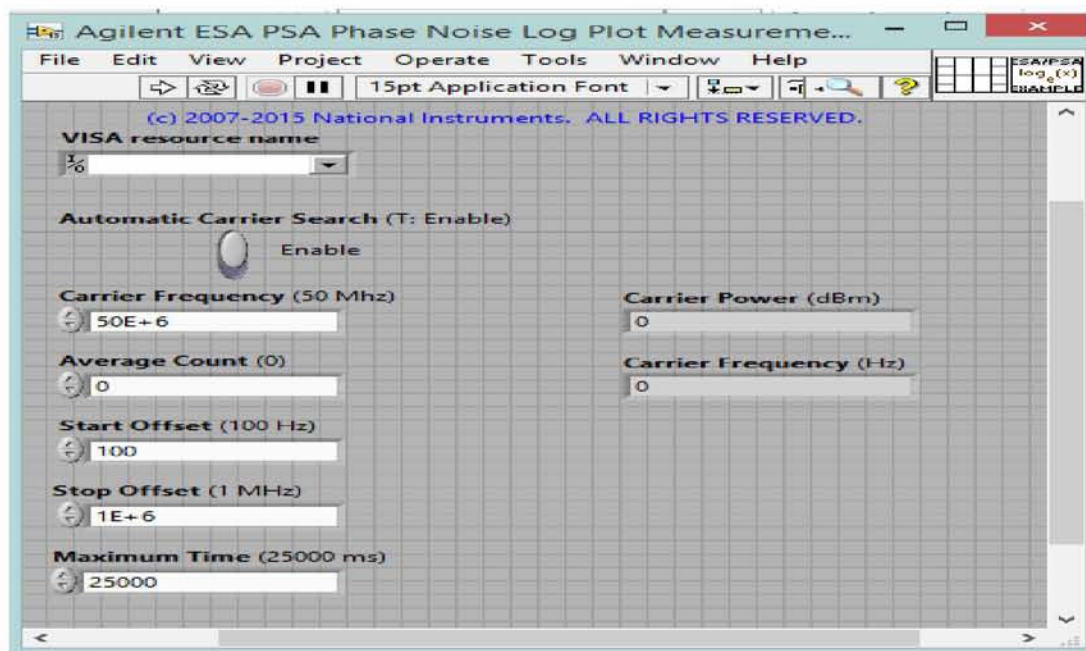
Το πρόγραμμα ελέγχου του αναλυτή φάσματος

7.1 Λειτουργίες που θέλουμε να επιτελέσει ο αναλυτής φάσματος (Spectrum Analyzer)

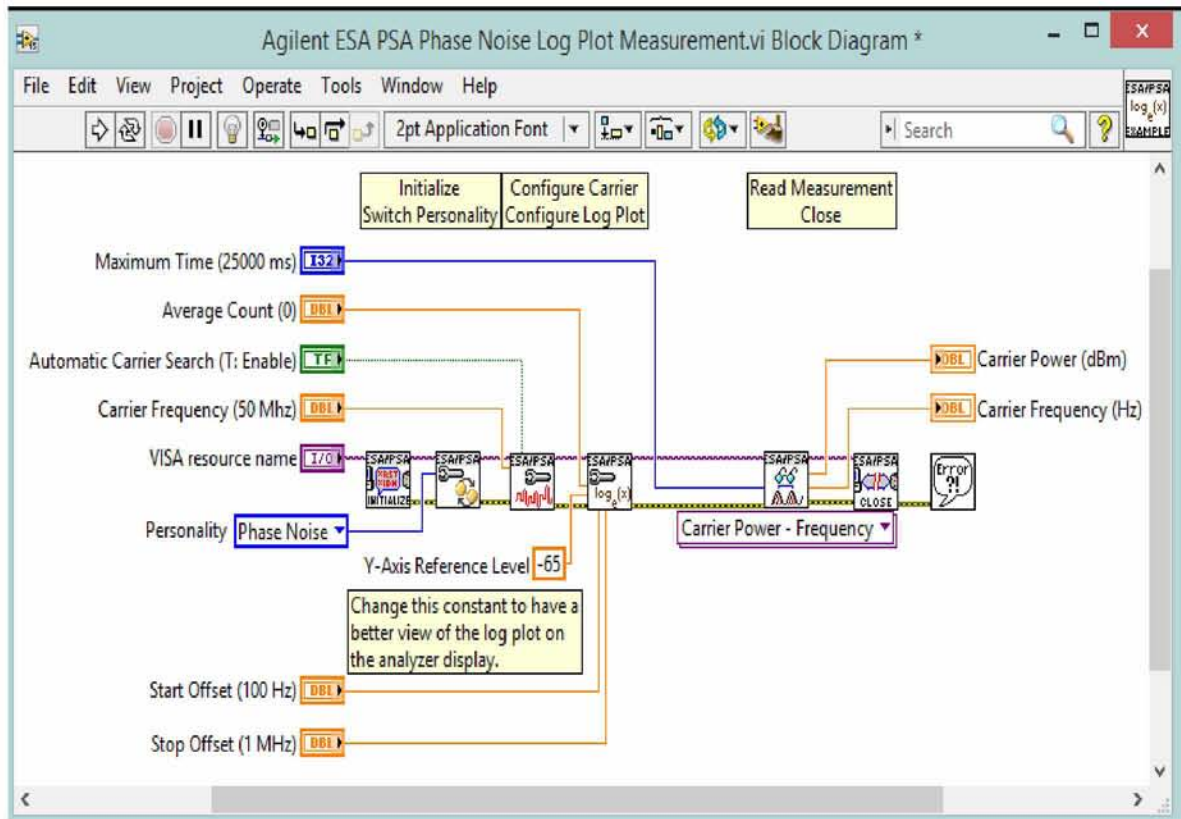
Το πρόγραμμα για τον έλεγχο του παλμογράφου δίνεται από την κατασκευάστρια εταιρία του LabVIEW και βρίσκεται στην τοποθεσία: C:\ProgramFiles\NationalInstruments\LabVIEW2015\instr.lib\AgilentESAPSASeries. Το παράδειγμα αυτό δημιουργεί μια γραφική παράσταση χρησιμοποιώντας το personality του θορύβου του σήματος. Το Agilent E4440A – PhaseNoiseLogPlotMeasurement.vi είναι ένα vi της βιβλιοθήκης και παρατηρώντας το block diagram θα δούμε ότι αποτελείται από subvi's. Αυτά είναι:

- 1) Initialize.vi
- 2) Select Personality.vi
- 3) Configure Carrier Signal.vi
- 4) Configure Log Plot.vi
- 5) Read Measurement (Carrier Power -Frequency).vi
- 6) Close.vi

Αγνοώντας το Initialize.vi και το Close.vi-των οποίων η χρησιμότητα όπως είπαμε είναι το άνοιγμα του διαδρόμου επικοινωνίας και το κλείσιμο αυτού αντίστοιχα-εστιάζουμε το ενδιαφέρον μας στα υπόλοιπα vi τα οποία θα αναλύσουμε παρακάτω. Το Front Panel και το Block Diagram του Agilent E4440A – PhaseNoiseLogPlotMeasurement.vi φαίνονται στα παρακάτω σχήματα. Το front panel απεικονίζεται στο σχήμα 7.1.1 ενώ το block diagram απεικονίζεται στο σχήμα 7.1.2.



Σχήμα 7.1.1 Front Panel του Agilent E4440A – Phase Noise Log Plot Measurement.vi

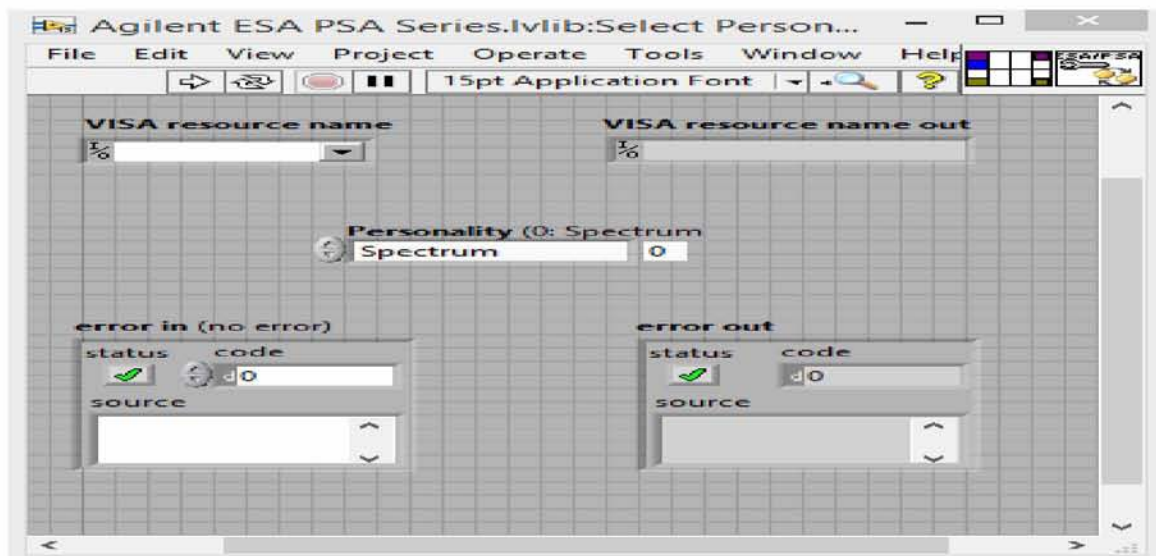


Σχήμα 7.1.2 Block Diagram του Agilent E4440A – Phase Noise Log Plot Measurement.vi

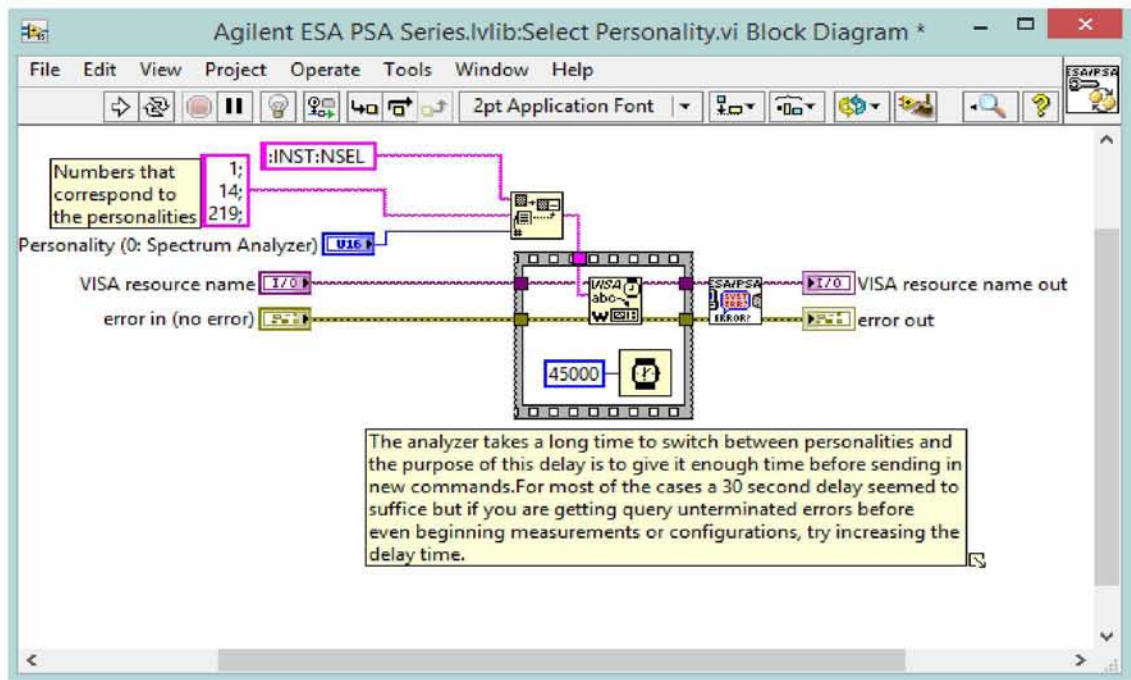
7.2 Ανάλυση των επιμέρους VI

➤ Select Personality.vi

Μέσω αυτού του vi γίνεται η επιλογή του personality που θα χρησιμοποιηθεί για τις μετρήσεις.



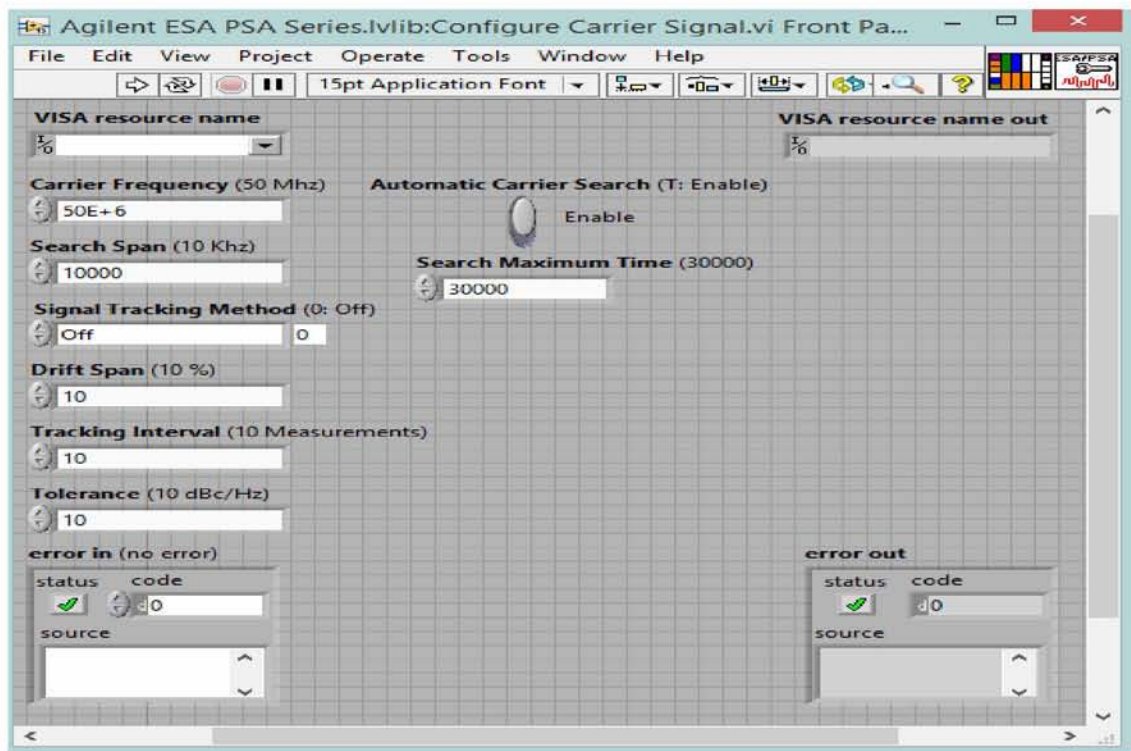
Σχήμα 7.2.1 Front Panel του Select Personality.vi



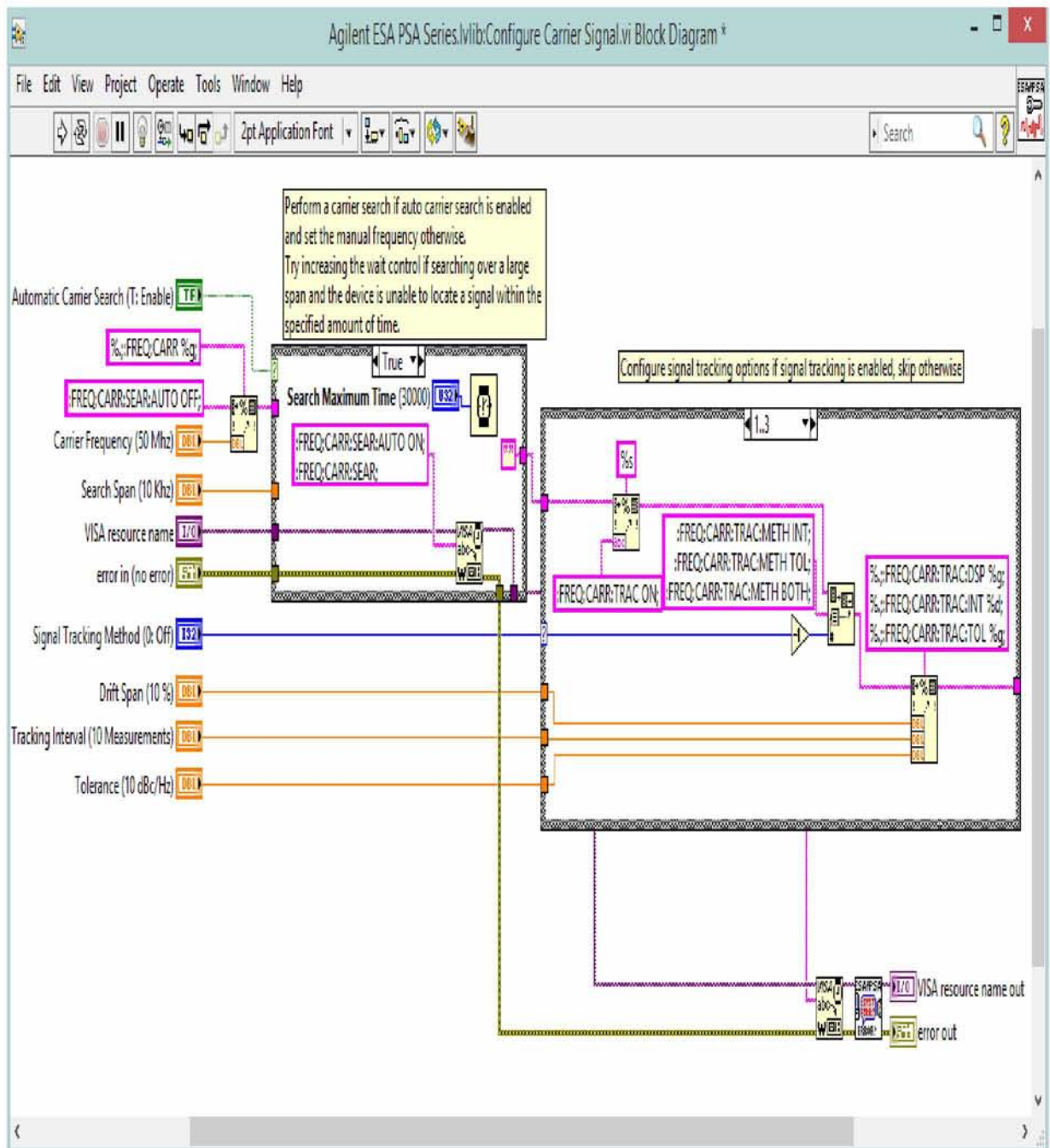
Σχήμα 7.2.2 Block Diagram του Select Personality.vi

➤ **Configure Carrier Signal.vi**

Αυτό το vi ρυθμίζει τις ιδιότητες του φέροντος σήματος.



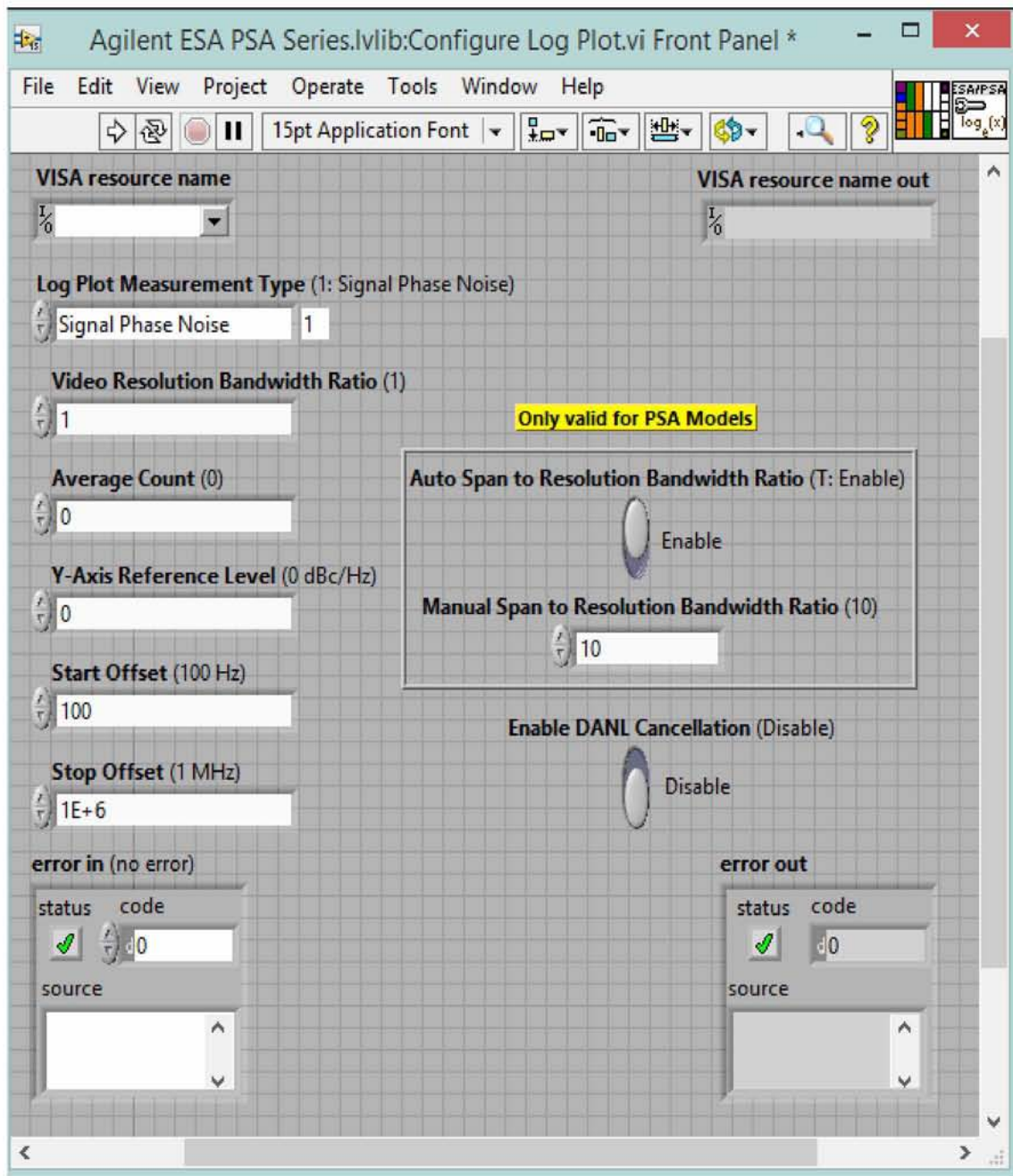
Σχήμα 7.2.3 Front Panel του Configure Carrier Signal.vi



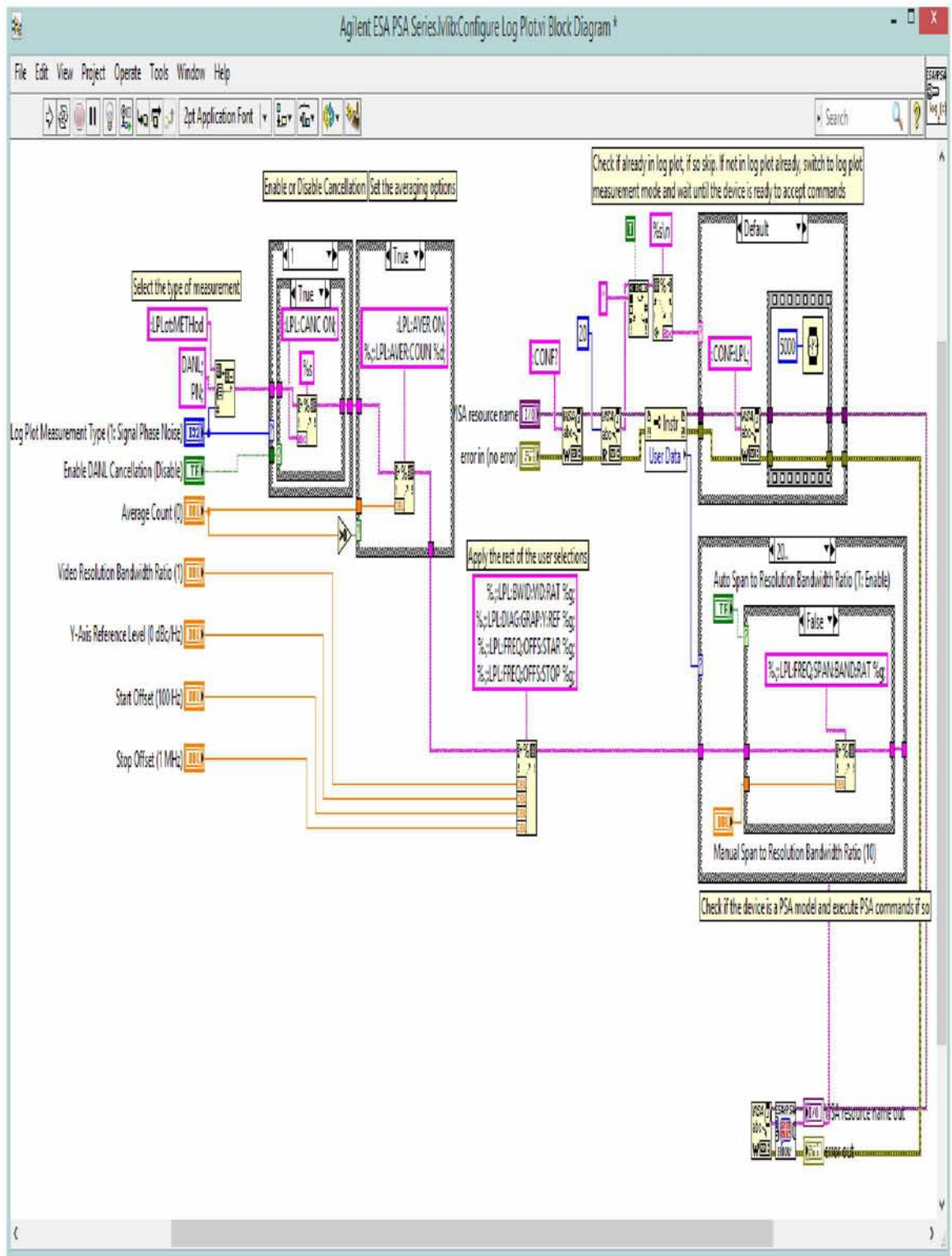
Σχήμα 7.2.4 Block Diagram του Configure Carrier Signal.vi

➤ **Configure Log Plot.vi**

Αυτό το vi διαμορφώνει τις επιλογές των μετρήσεων τις γραφικής παράστασης του personality του θορύβου.



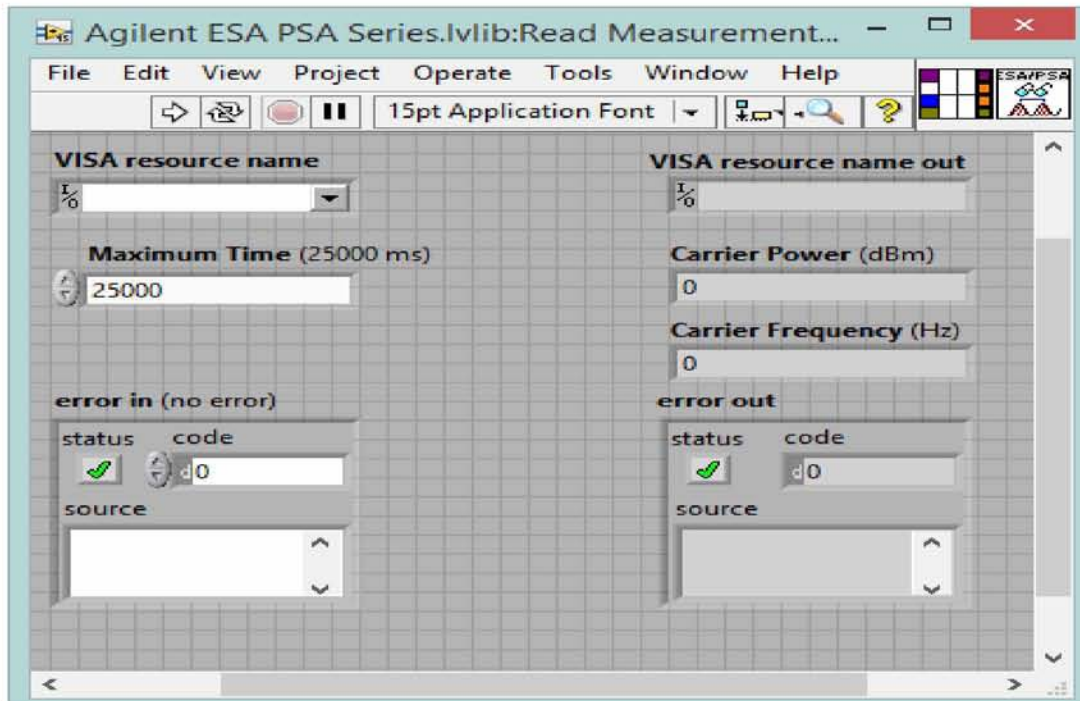
Σχήμα 7.2.5 Front Panel του Configure Log Plot.vi



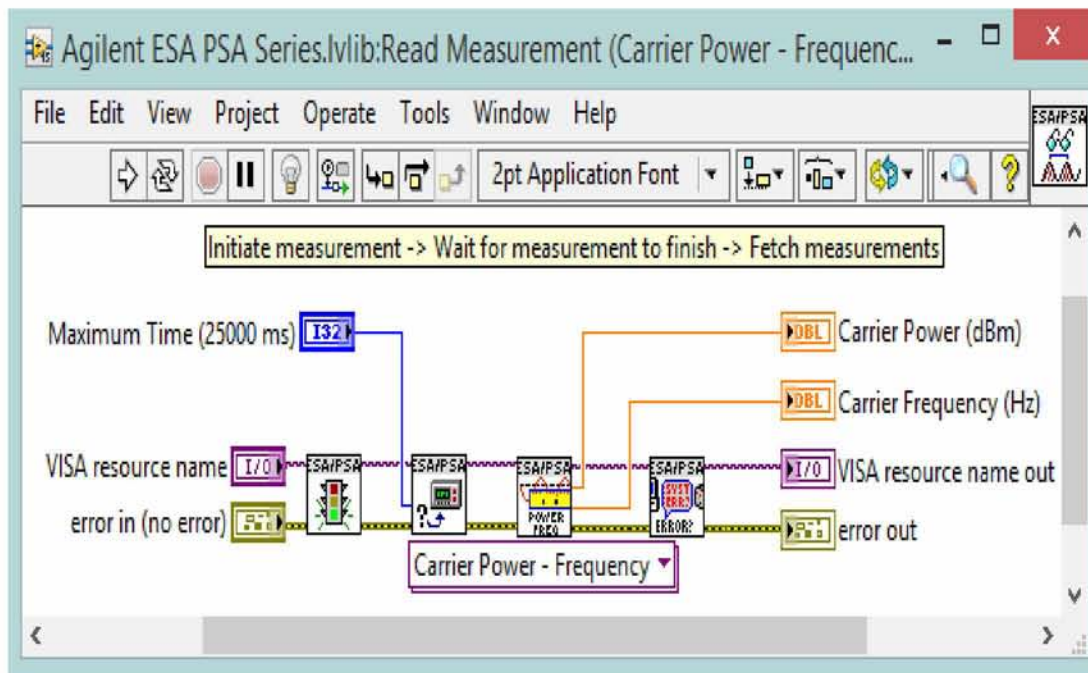
Σχήμα 7.2.6 Block Diagram του Configure Log Plot.vi

➤ **Read Measurement (Carrier Power -Frequency).vi**

Αυτό το vi διαβάζει τις τιμές των carrier power και carrier frequency έτσι ώστε να δημιουργήσει το ζητούμενο διάγραμμα(log plot, spot frequency, monitor spectrum).



Σχήμα 7.2.7 Front Panel του Read Measurement (Carrier Power -Frequency).vi

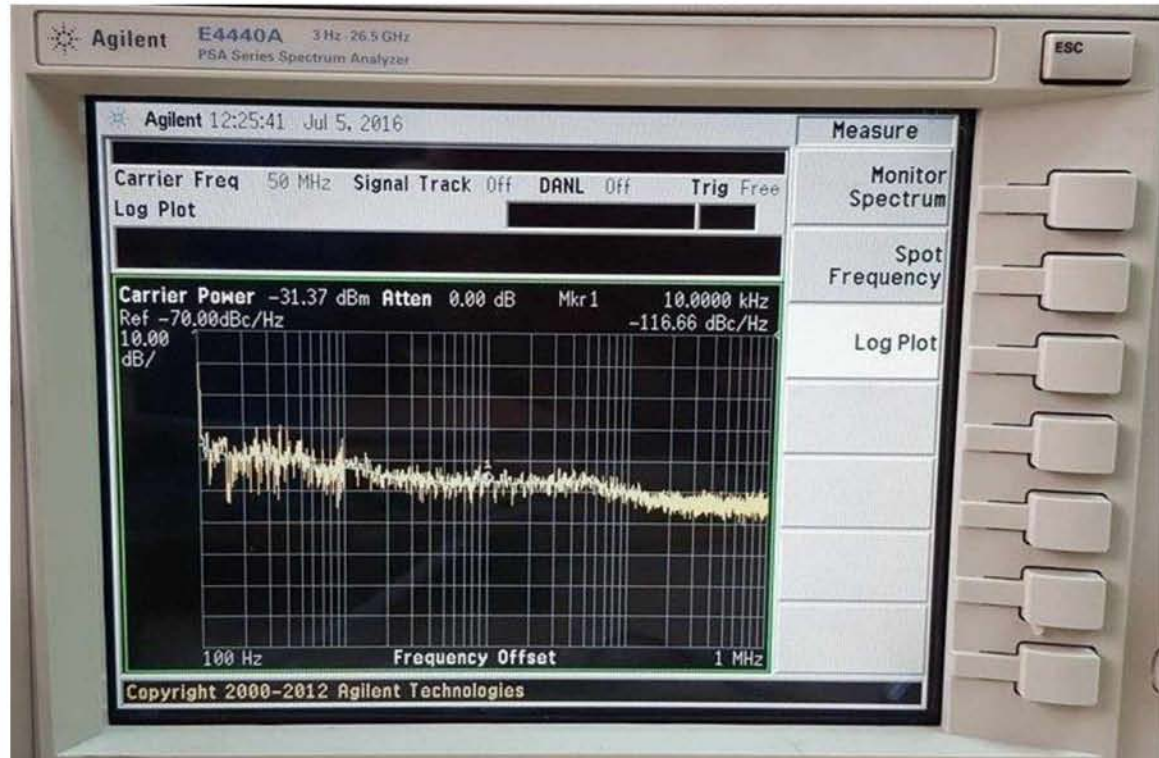


Σχήμα 7.2.8 Block Diagram του Read Measurement (Carrier Power - Frequency).vi

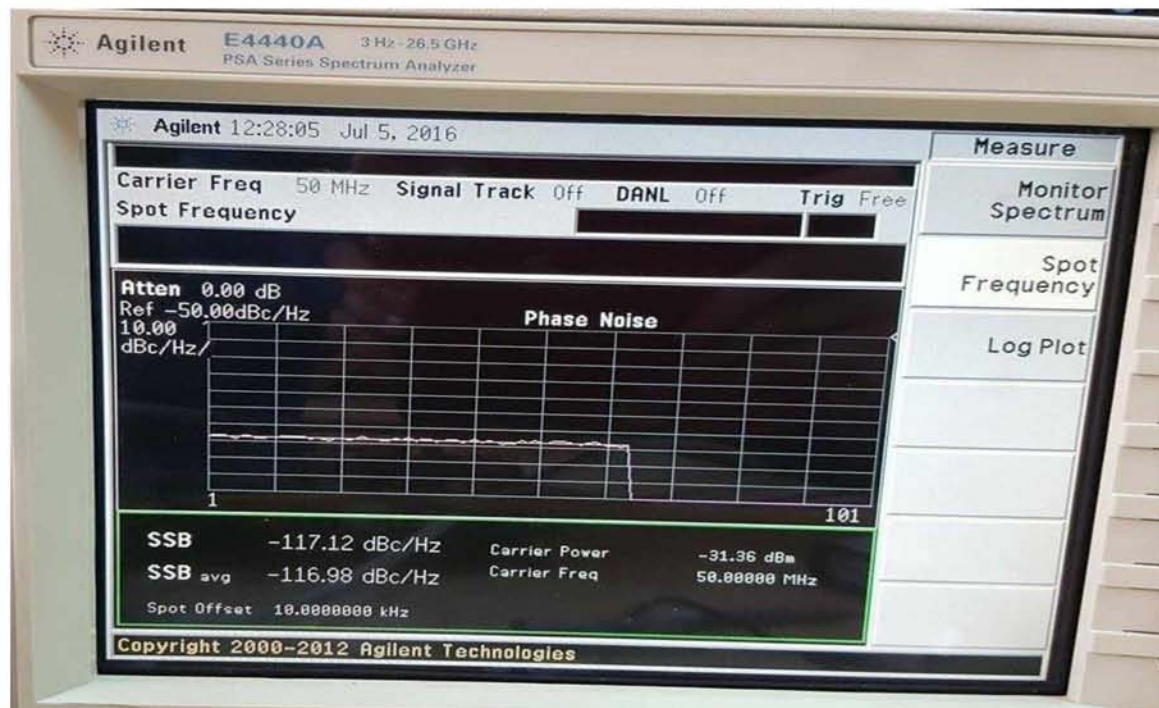
7.3 Αποτελέσματα του αναλυτή φάσματος(E4440A)

Τα αποτελέσματα που θα δούμε στον αναλυτή φάσματος είναι τα εξής:

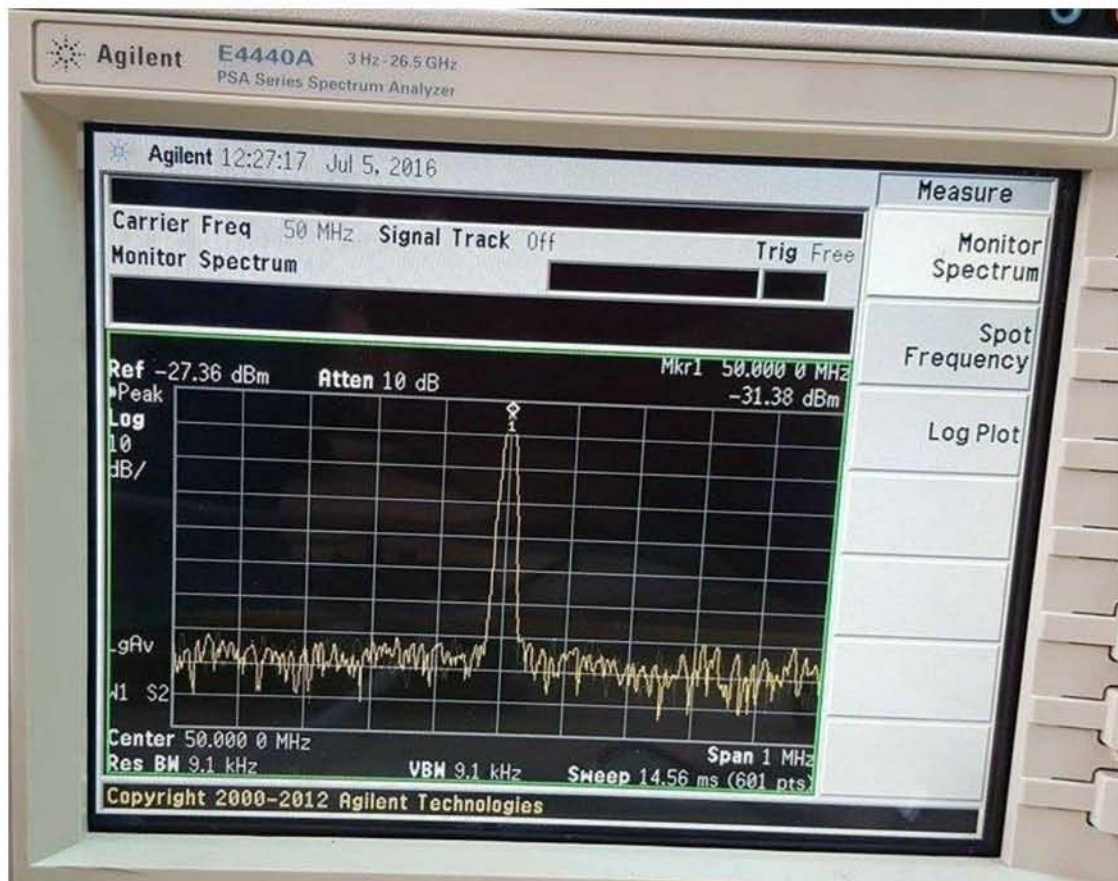
- Για Carrier Frequency = 50 MHz έχουμε:



Σχήμα 7.3.1 Log Plot

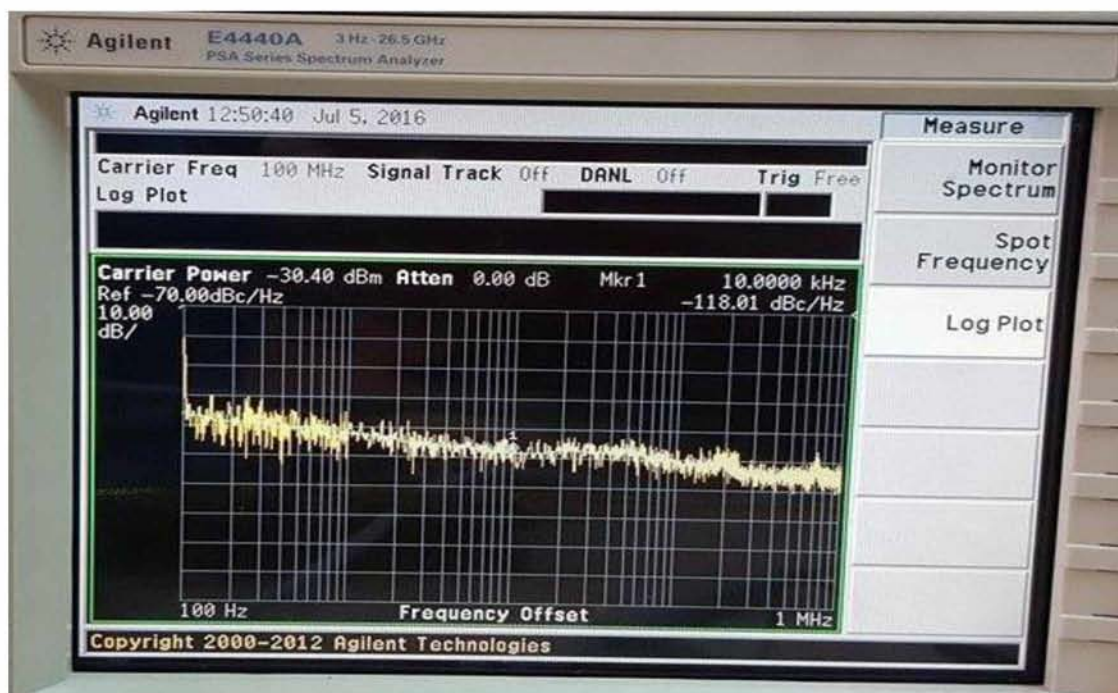


Σχήμα 7.3.2 Spot Frequency

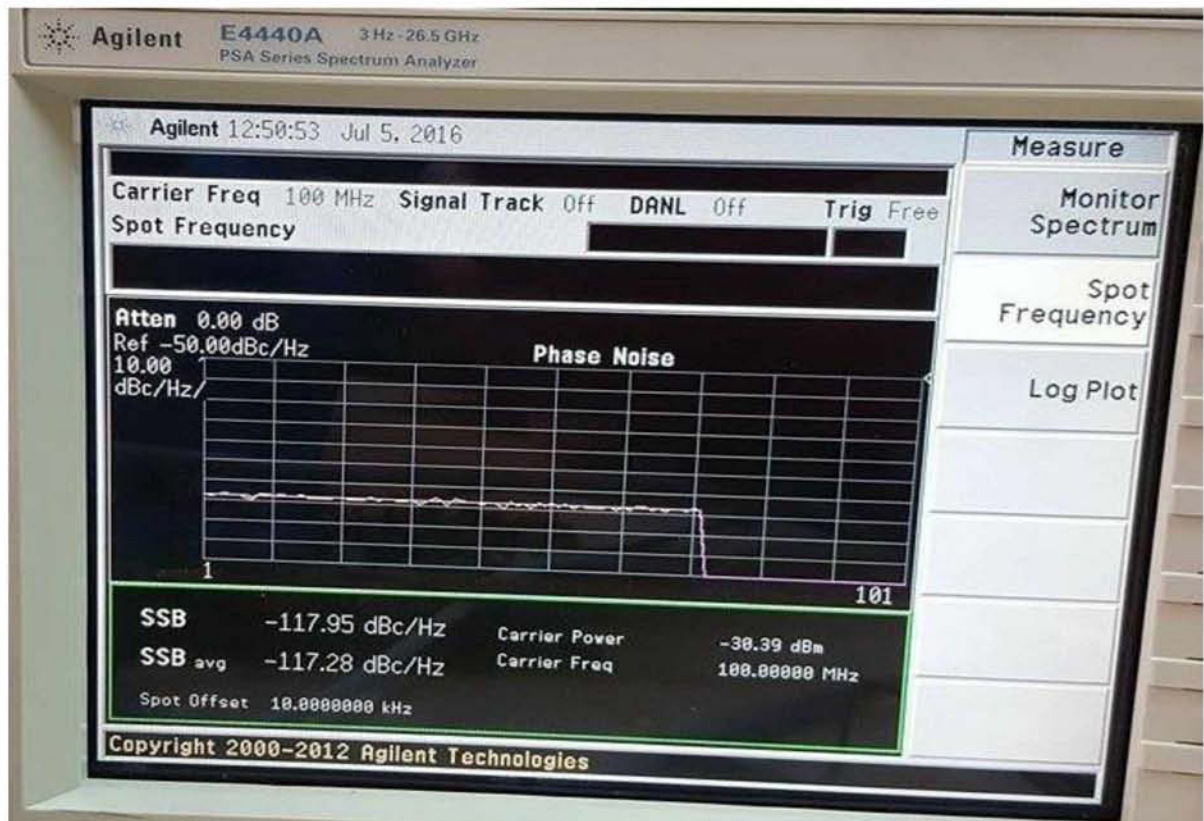


Σχήμα 7.3.3 Monitor Spectrum

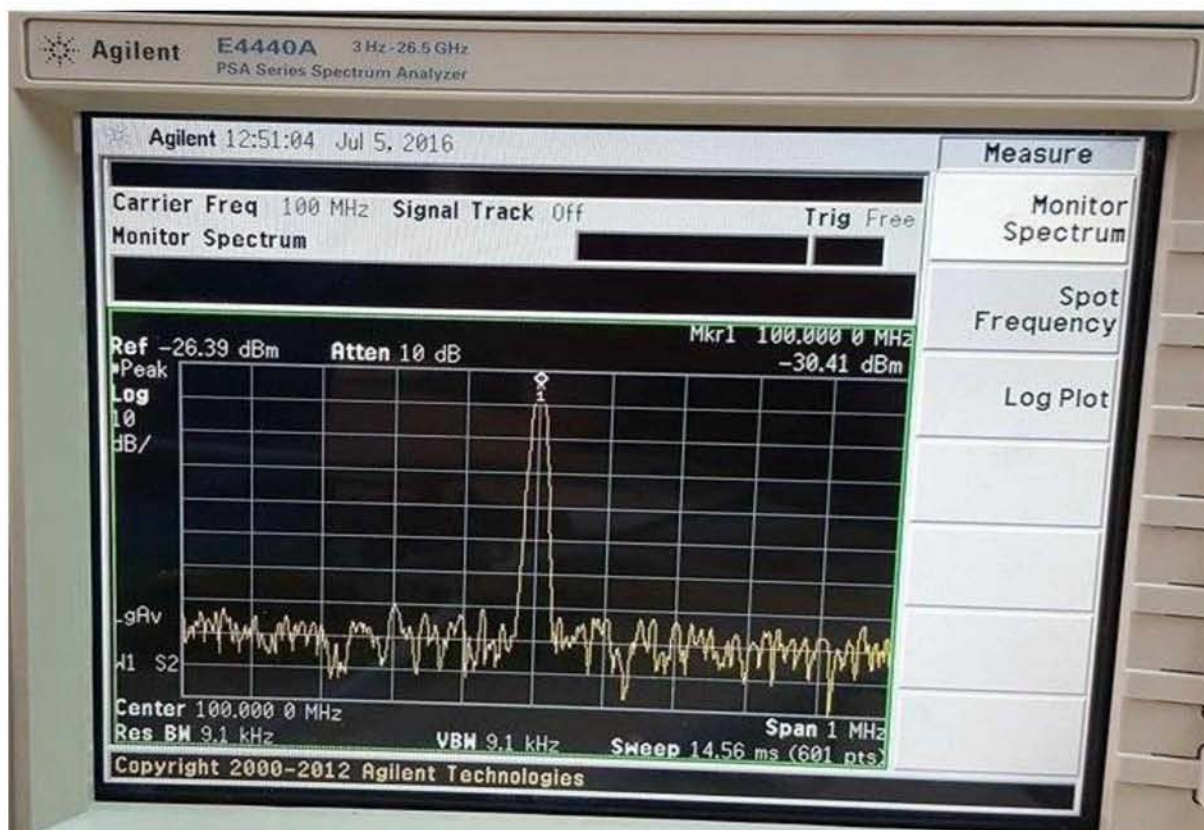
➤ Για Carrier Frequency = 100 MHz έχουμε:



Σχήμα 7.3.4 Log Plot

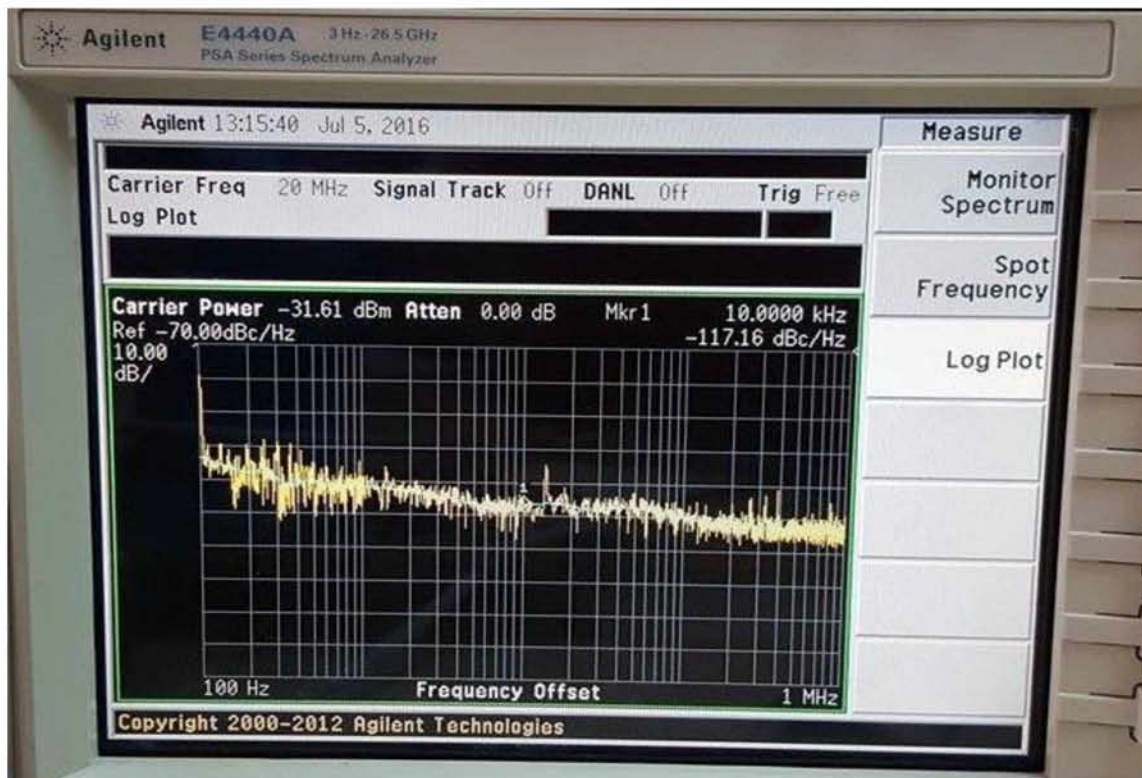


Σχήμα 7.3.5 Spot Frequency

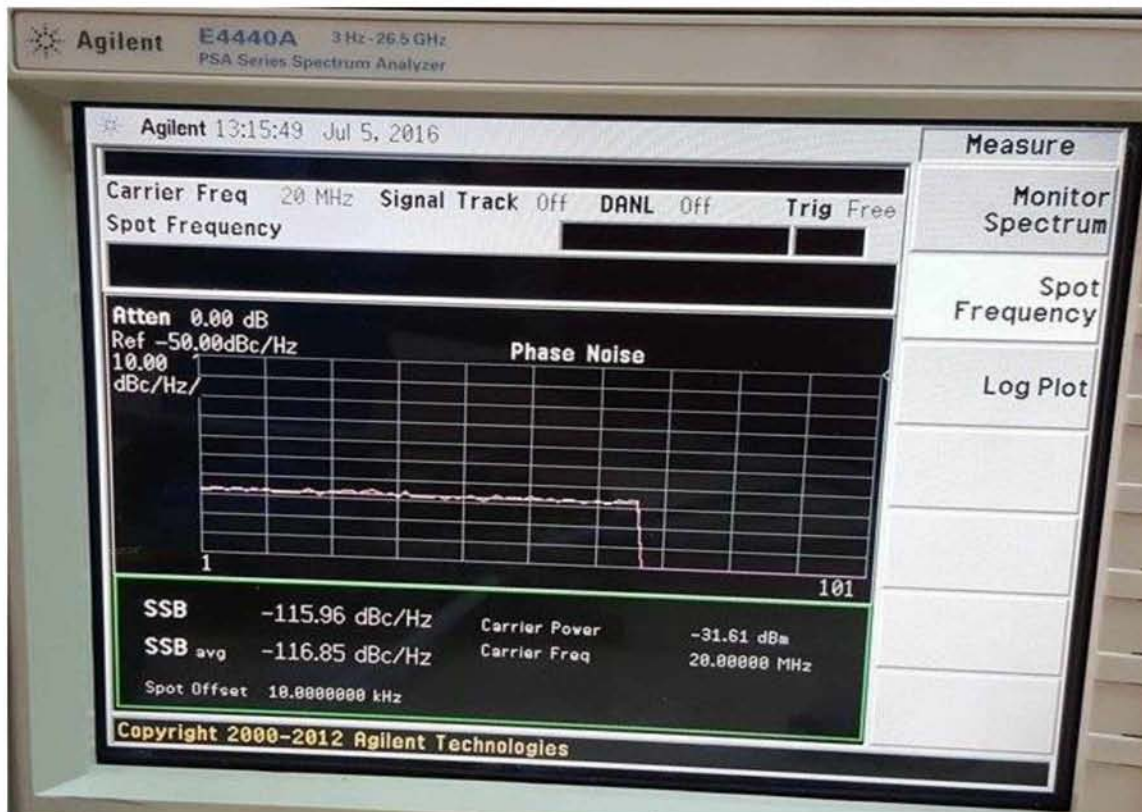


Σχήμα 7.3.6 Monitor Spectrum

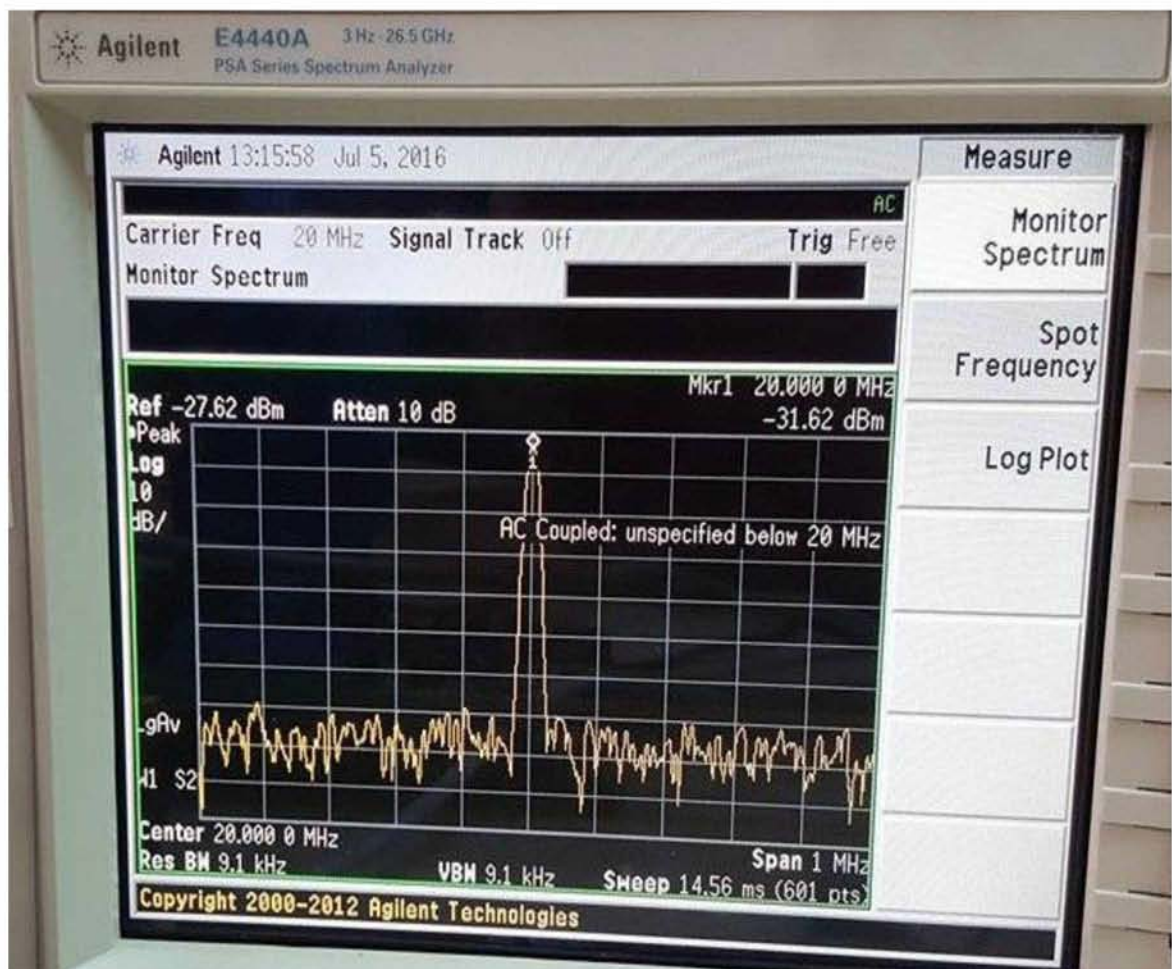
➤ Για Carrier Frequency = 200 MHz έχουμε:



Σχήμα 7.3.7 Log Plot



Σχήμα 7.3.8 Spot Frequency



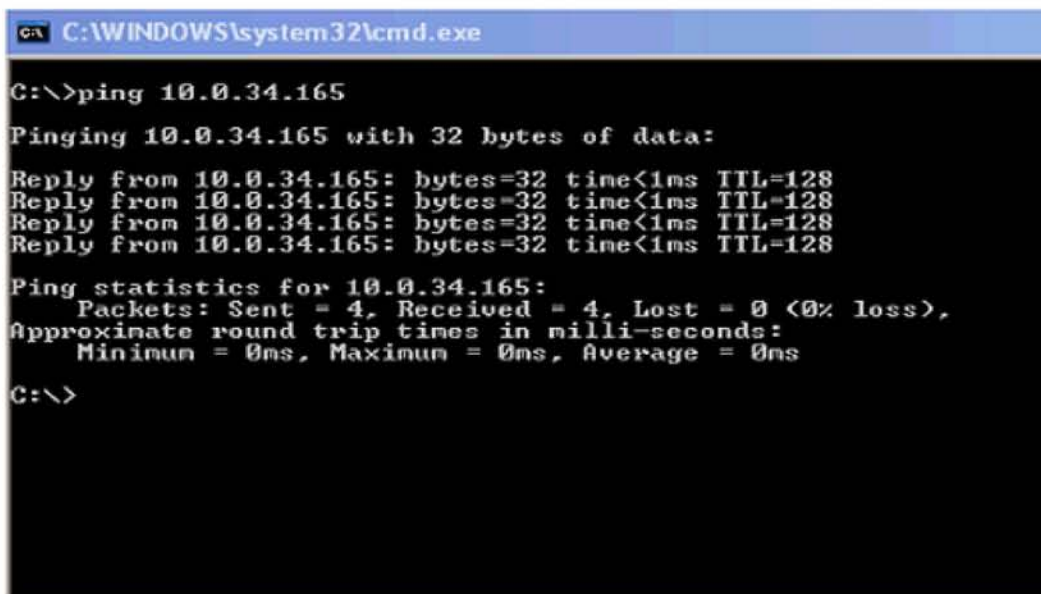
Σχήμα 7.3.9 Monitor Spectrum

Σύνδεση οργάνων μέσω Ethernet/LAN

Το LabVIEW προσφέρει τη δυνατότητα στους χρήστες του να συνδέσουν τα όργανα που χρησιμοποιούν με ποικίλους τρόπους. Στην παρούσα διπλωματική επιλέξαμε να πραγματοποιήσουμε τη σύνδεση μέσω καλωδίου Ethernet, εφόσον είχε προηγηθεί το στήσιμο ενός τοπικού δικτύου με τα όργανα του εργαστηρίου. Για να μπορέσει να επιτευχθεί η επικοινωνία του LabVIEW με τα επιθυμητά όργανα θα πρέπει να ακολουθηθούν οι παρακάτω ενέργειες.

1. Έλεγχος σύνδεσης με τη συσκευή:

Αρχικά θα πρέπει να έχει γίνει λήψη και εγκατάσταση του NI-VISA. Στη συνέχεια θα πρέπει να ελεγχθεί αν έχει πραγματοποιηθεί η σύνδεση του υπολογιστή με τη συσκευή κάτι το οποίο γίνεται μέσω της εντολής ping στη γραμμή εντολών των Windows. Συγκεκριμένα, πληκτρολογούμε "ping xxx.xxx.xxx.xxx" (όπου xxx.xxx.xxx.xxx είναι η IP διεύθυνση του οργάνου). Εφόσον η σύνδεση είναι επιτυχημένη θα εμφανιστεί κάτι αντίστοιχο με την παρακάτω εικόνα.

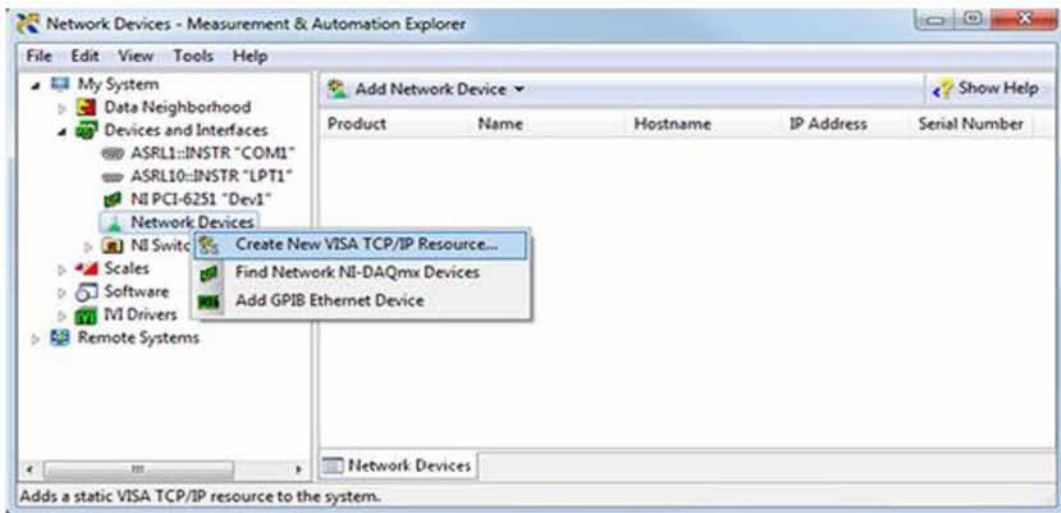


```
C:\WINDOWS\system32\cmd.exe
C:\>ping 10.0.34.165
Pinging 10.0.34.165 with 32 bytes of data:
Reply from 10.0.34.165: bytes=32 time<1ms TTL=128
Reply from 10.0.34.165: bytes=32 time<1ms TTL=128
Reply from 10.0.34.165: bytes=32 time<1ms TTL=128
Reply from 10.0.34.165: bytes=32 time<1ms TTL=128
Ping statistics for 10.0.34.165:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
```

Σχήμα 1 –Αποτέλεσμα της εντολής ping

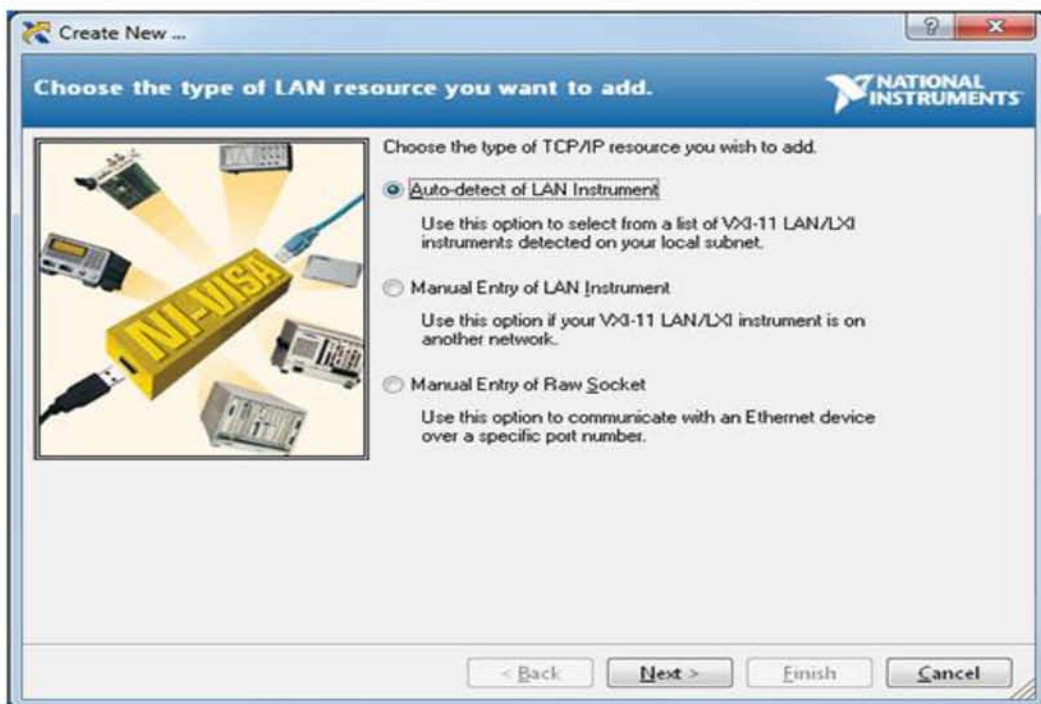
2. Διαμόρφωση της συσκευής στο NIMAX

Στο σημείο αυτό θα πρέπει να ανοίξουμε το πρόγραμμα NI MAX και να κάνουμε κλικ στο "MySystem" ώστε να εμφανιστούν περαιτέρω επιλογές ανάμεσα στις οποίες είναι και το "Devices and Interfaces". Επεκτείνοντας το Devices and Interfaces εμφανίζεται η επιλογή "NetworkDevices" μέσω της οποίας με δεξί κλικ μπορούμε να δημιουργήσουμε ένα νέο πόρο για τη συσκευή μας (CreateNewVISATCP/IPResource).



Σχήμα 2 – Δημιουργία VISA Resource

Στο παράθυρο που ανοίγει (Σχήμα 3) επιλέγουμε “Auto-detect of LAN Instrument” καθώς και οι τρεις συσκευές που προσθέτουμε ανήκουν στο τοπικό δίκτυο. Πατώντας το κουμπί “Next” ολοκληρώνεται η δημιουργία του πόρου και είναι εφικτός ο χειρισμός του οργάνου από τον υπολογιστή μας.



Σχήμα 3 – Επιλογή LAN Resource

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Κωνσταντίνος Καλοβρέκτης, “LabVIEW για Μηχανικούς Συστήματα συλλογής Δεδομένων”, Εκδόσεις Τζιόλα.
2. National Instruments, “LabVIEW Student Edition User’s Guide”
3. National Instrments, “LabVIEW Data Acquisition Manual”
4. National Instruments, “LabView User Manual”
5. National Instruments, “NI VISA Programmer Reference Manual”
6. National Instruments, “NI VISA User Manual”
7. National Instruments, “LabView Instrument Driver Standards”
8. <http://www.ni.com/>
9. <http://www.agilent.com>
10. E4438C ESG Vector Signal Generator(Keysight Technologies):
<http://www.keysight.com/en/pd-1000004297%3Aepsg%3Apro/esg-vectorsignalgenerator?cc=GR&lc=eng&lsrch=true&searchT=e4438c>
11. 54831B Oscilloscope(Keysight Technologies) :
<http://www.keysight.com/en/pd-1000004328%3Aepsg%3Apro-pn-54831B/infiniium-dso?cc=GR&lc=eng>
12. E4440A PSA Spectrum Analyzer(Keysight Technologies):
<http://www.keysight.com/en/pdx-x201707-pn-E4440A/psa-spectrum-analyzer-3hzto265ghz?cc=GR&lc=eng&lsrch=true&searchT=e4440a>
13. Connecting Instruments via Ethernet/LAN : <http://www.ni.com/getting-started/set-up-hardware/instrument-control/ethernet-connect>
14. Wikipedia Oscilloscope : <https://en.wikipedia.org/wiki/Oscilloscope>
15. Wikipedia Vector Signal Generaton:
https://en.wikipedia.org/wiki/Signal_generator#Vector_signal_generator

16. Wikipedia Spectrum Analyzer:

https://en.wikipedia.org/wiki/Spectrum_analyzer

17. Agilent PSA Series Spectrum Analyzers Phase Noise Measurement
Personality : <http://cp.literature.agilent.com/litweb/pdf/5988-3698EN.pdf>