

Εργαστήριο Αναλογικού και Ψηφιακού Ελέγχου Ι



LabVIEW



Εισαγωγή

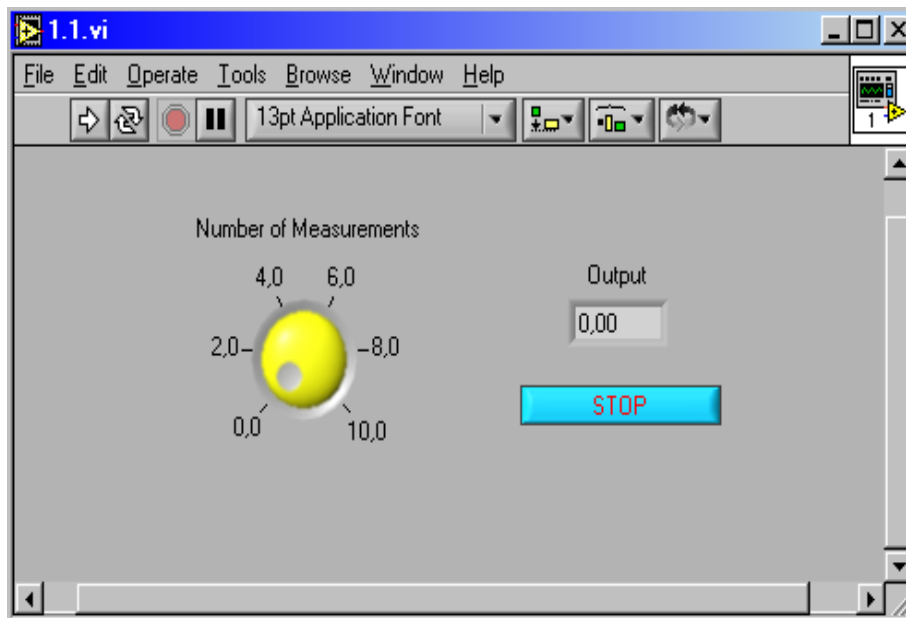
Το LabVIEW αποτελεί εργαλείο που μπορεί να χρησιμοποιηθεί για τη σχεδίαση και την ανάπτυξη πολύπλοκων εφαρμογών, ικανών να αλληλεπιδρούν με το εξωτερικό περιβάλλον. Το βασικό του πλεονέκτημα είναι ότι βασίζεται στον γραφικό προγραμματισμό (Graphical Programming) και αυτό το κάνει ιδανικό για εφαρμογές μετρήσεων, αυτοματισμού, ελέγχου οργάνων και ανάλυση δεδομένων.

Εικονικά όργανα (Virtual Instruments)

Τα VIs (εικονικά όργανα) είναι στοιχεία προγραμματισμού του LabVIEW, τα οποία επιτελούνται από το front panel και το block diagram.



Στην Εικόνα 1 απεικονίζεται το front panel ενός εικονικού οργάνου (VI) του LabVIEW. Το front panel χρησιμοποιείται για την απεικόνιση στοιχείων τύπου controls και indicators για το χρήστη. Μέσω αυτού και των στοιχείων controls, οι χρήστες μπορούν να θέσουν τιμές στις μεταβλητές, να επιλέξουν διαδικασίες και γενικά να ελέγξουν τη ροή και τον τρόπο εκτέλεσης του προγράμματος. Μέσω των στοιχείων indicators, οι χρήστες μπορούν να παρατηρήσουν την μεταβολή των μεταβλητών, τα αποτελέσματα διεργασιών και γενικά όλη την πληροφορία που παράγεται κατά την εκτέλεση του γραφικού κώδικα στο block diagram.

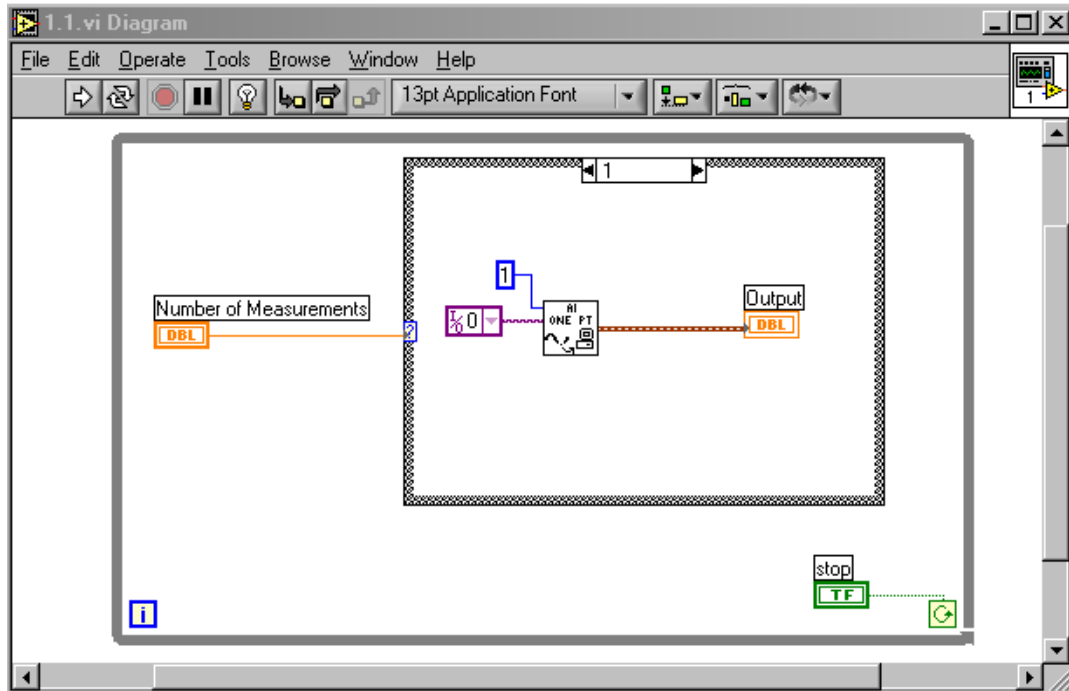
Στην παρακάτω εικόνα, λοιπόν, υπάρχει ένας περιστροφικός διακόπτης για την επιλογή των μετρήσεων, ένα αντικείμενο (indicator) για την απεικόνιση της μετρούμενης τιμής και ένα κουμπί για stop.



Εικόνα 1. Front Panel εικονικού οργάνου

Στην Εικόνα 2 απεικονίζεται το block diagram (δομικό διάγραμμα) ή ο πηγαίος κώδικας της εφαρμογής της Εικόνας 1. Τα VIs προγραμματίζονται στη γλώσσα-G όπου και το πρόγραμμα γίνεται compile από τη μηχανή εκτέλεσης του LabVIEW. Ακολούθως, τα VIs μετατρέπονται σε εκτελέσιμο κώδικα που μπορεί να εκτελεστεί από τη μηχανή διαδικασιών

του LabVIEW. Κάθε φορά που μία αλλαγή συμβαίνει στο VI, το LabVIEW ενημερώνει κατάλληλα και τον πηγαίο κώδικα. Αν το δομικό διάγραμμα δεν είναι σωστά σχεδιασμένο τότε, στο σημείο που υπάρχει το πρόβλημα, το καλώδιο σύνδεσης των δύο δομικών στοιχείων είναι διακεκομμένο, καθώς, επίσης, το βελάκι εκτέλεσης έχει την επόμενη μορφή . Στην αντίθετη περίπτωση, το βελάκι έχει τη μορφή  και ο χρήστης είναι έτοιμος να εκτελέσει το VI.



Εικόνα 2. Block Diagram του εικονικού οργάνου της Εικόνας 1

Στο Block Diagram μπορούμε να προγραμματίσουμε την συμπεριφορά του εικονικού οργάνου. Κάθε αντικείμενο στο Front Panel, έχει την δική του οντότητα στο Block Diagram και αλληλεπιδρά με το πρόγραμμα που αναπτύσσεται εκεί.

Στο Block Diagram έχουμε αντικείμενα τύπου nodes και terminals. Τα nodes είναι εκτελέσιμα στοιχεία προγραμματισμού και έχουν λειτουργία ανάλογη με αυτή των ορισμάτων, των υπορουτινών και των συναρτήσεων σε μια κοινή γλώσσα προγραμματισμού.

- Υπάρχουν τέσσερις τύποι nodes:

- 1) functions,
- 2) subVI nodes,
- 3) structures και
- 4) Code Interface Nodes (CINs).

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

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

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

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





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


- 1) control-indicator terminals
- 2) node terminals.

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

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

Εκτέλεση των VIs

Ένα πρόγραμμα LabVIEW εκτελείται με το πάτημα του *Run Button*  τα οποία βρίσκονται στην παλέτα, στο πάνω μέρος του παραθύρου. Κατά τη διάρκεια της εκτέλεσης το *Run Button* αλλάζει, και γίνεται . Θα πρέπει να σημειώσουμε πως, κατά τη διάρκεια της λειτουργίας, τα κουμπιά που εμφανίζονται στην παλέτα είναι το *Continuous Run*, *Stop* και *Pause buttons* . Τα τρία τελευταία κουμπιά  χρησιμοποιούνται για την στοίχιση των αντικειμένων που εισάγουμε.

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

Οι παλέτες εργασίας

Οι τρεις διαθέσιμες παλέτες εργασίας είναι:

Tools Palette

Η παλέτα αυτή περιέχει εργαλεία που είναι κατάλληλα για τη δημιουργία και την εκτέλεση των εικονικών οργάνων (VIs). Εάν η παλέτα δεν είναι ορατή, τότε για την εμφάνισή της επιλέγουμε Show Tools Palette από το Window menu. Μετά την επιλογή ενός εργαλείου από αυτό το menu, ο κέρσορας παίρνει τη μορφή του επιλεγμένου εργαλείου.



Tools Palette

Controls Palette

Μπορούμε να εισάγουμε κουμπιά ελέγχου και απεικόνισης στο front panel από την Controls Palette. Κάθε επιλογή σε αυτή την παλέτα, εμπεριέχει μία υπο-επιλογή με επιπλέον κουμπιά ελέγχου και απεικόνισης σχετικά με την προηγούμενη επιλογή. Σε περίπτωση που η παλέτα δεν είναι εμφανής, μπορούμε να την ανοίξουμε, επιλέγοντας, Show Controls Palette από το Window menu. Επιπλέον, μπορούμε να ανοίξουμε την Controls Palette, κάνοντας δεξί click πάνω σε μια άδεια περιοχή του front panel. Προσοχή θα πρέπει να δώσουμε στο γεγονός ότι η παλέτα αυτή είναι ενεργή μόνο όταν βρισκόμαστε στο front panel.



Controls Palette

Functions Palette


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

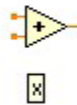


Functions Palette

Διασύνδεση (Wiring)

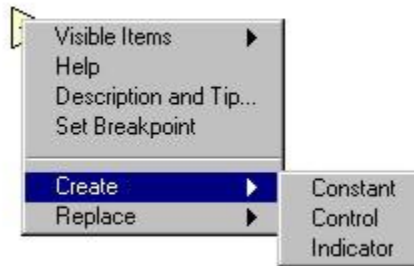
Τα καλώδια (wires) είναι ο δρόμος μεταγωγής των δεδομένων μεταξύ των αντικειμένων. Διαφορετικοί τύποι καλωδίων αντιστοιχούν σε διαφορετικούς τύπους δεδομένων (π.χ. ακέραιοι αριθμοί κινητής υποδιαστολής). Σε μια έγχρωμη εικόνα, κάθε τύπος δεδομένων, απεικονίζεται και με ένα διαφορετικό χρώμα για μεγαλύτερη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

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

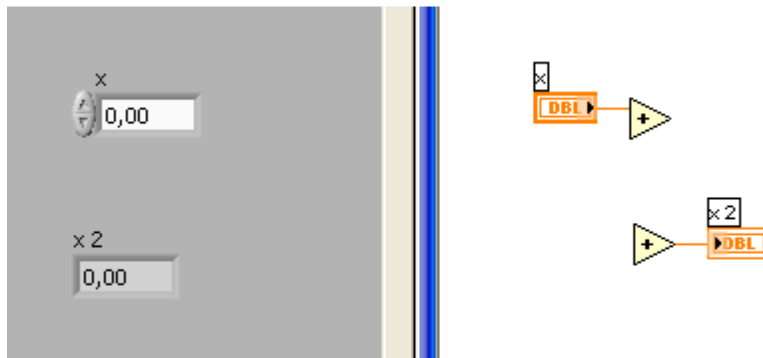


Όταν το wiring tool βρίσκεται πάνω από την περιοχή ενός Terminal τότε η περιοχή αυτή αναβοσβήνει, υποδηλώνοντας ότι ένα click θα συνδέσει το καλώδιο σε αυτό το Terminal. Δεν χρειάζεται να έχουμε πατημένο το mouse κατά την διάρκεια της μετακίνησης του wiring tool από το ένα Terminal στο άλλο.

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



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



Σε αυτήν την λειτουργία όπου τα controls/indicators δημιουργούνται αυτόματα, η μορφή τους είναι δεδομένη (default) και αντιστοιχεί σε αυτή του απλού control/indicator που παρουσιάζεται στο προηγούμενο σχήμα.

Με σκοπό την αντικατάσταση των control/indicator που έχουν δημιουργηθεί, από αυτά της αρεσκείας μας, και διατηρώντας πάντα τις διασυνδέσεις (καλωδιώσεις) που έχουν αυτόματα δημιουργηθεί, θα πρέπει στο front panel να κάνουμε δεξί click πάνω στο control/indicator που μας ενδιαφέρει και να επιλέξουμε την εντολή replace. Η διαδικασία αυτή απεικονίζεται στο επόμενο σχήμα.

Στο σχήμα αυτό μπορούμε να παρατηρήσουμε όλες τις ενέργειες που μπορούμε να εκτελέσουμε σε κάθε control/indicator.

Οδηγίες για την παράδοση των αναφορών

Κάθε εικονικό όργανο (VI) που σχεδιάζετε στα πλαίσια της άσκησης, να το αποθηκεύετε δίνοντας του όνομα που ακολουθεί την παρακάτω φόρμα:

Ο(μαδα)#_Α(σκηση)#_Ε(φαρμογη)#.vi

π.χ. *O1_A5_E3.vi*

Το συμπιεσμένο αρχείο (.rar ή .zip) που θα παραδίδετε θα πρέπει να έχει όνομα που θα ακολουθεί την παρακάτω φόρμα:

Ο(μαδα)#_Α(σκηση)#.rar (ή .zip)

π.χ. *O1_A5.rar*

και θα πρέπει να περιλαμβάνει:

- Την γραπτή αναφορά της άσκησης (σε μορφή .pdf ακολουθώντας τους παραπάνω κανόνες ονομασίας π.χ. *O1_A5.pdf*) όπου θα παρουσιάζετε αναλυτικά τα VI's που δημιουργήσατε, κάνοντας τα απαραίτητα σχόλια για την επεξήγηση τους και απαντώντας σε ερωτήσεις που συμπεριλαμβάνονται στις εκφωνήσεις των ασκήσεων ή που θα σας τεθούν κατά τη διάρκεια εκτέλεσης του εργαστηρίου.
- Όλα τα VI αρχεία που δημιουργήσατε στα πλαίσια της άσκησης και αντιστοιχούν στις εφαρμογές της άσκησης αυτής.
- Οτιδήποτε άλλο σας ζητηθεί κατά την διάρκεια εκτέλεσης του εργαστηρίου.

Οι αναφορές θα παραδίδονται μέσω της πλατφόρμας eclass. Περαιτέρω πληροφορίες για το θέμα της παράδοσης θα σας δοθούν στο εργαστήριο.

Άσκηση 1

Εισαγωγή στο λογισμικό LabVIEW

Σκοπός της άσκησης είναι η εισαγωγή στο περιβάλλον του λογισμικού LabVIEW και η κατασκευή απλών εικονικών οργάνων που αφορούν βασικές αριθμητικές και λογικές πράξεις.

Εφαρμογές

- 1.1 Σχεδιασμός εικονικού οργάνου (VI) για την αφαίρεση δύο ακέραιων αριθμών από το -2 έως το 9.
- 1.2 Σχεδιασμός εικονικού οργάνου (VI) για τον πολλαπλασιασμό δύο τυχαίων αριθμών από το 0 μέχρι το 13 και το αποτέλεσμα να εμφανίζεται ως ακέραιος. (Υπόδειξη: χρήση *Random Number (0-1)*)
- 1.3 Σχεδιασμός εικονικού οργάνου (VI) για τον υπολογισμό του μέσου όρου τριών αριθμών.
- 1.4 Σχεδιασμός εικονικού οργάνου (VI) για τον υπολογισμό και την απεικόνιση της λογικής πράξης *OR* δύο λογικών μεταβλητών.
- 1.5 Σχεδιασμός εικονικού οργάνου (VI) που θα συγκρίνει δύο νέους τυχαίους αριθμούς κάθε 1.5sec και να ενεργοποιεί Boolean Indicator ανάλογα με το αποτέλεσμα της σύγκρισης. (Υπόδειξη: χρήση *While Loop, Wait until Next ms Multiple*)
- 1.6 Σχεδιασμός εικονικού οργάνου (VI) για τη σύγκριση δύο ακέραιων αριθμών από το 0 έως το 15 και την απεικόνιση του μικρότερου. (Υπόδειξη: χρήση *Comparison Function → Select*)
- 1.7 Σχεδιασμός εικονικού οργάνου (VI) για τη σύγκριση δύο αριθμών από το -12 έως το 22 και την απεικόνιση του μικρότερου και του μεγαλύτερου. (Υπόδειξη: χρήση *Case Structure*)
- 1.8 Σχεδιασμός εικονικού οργάνου (VI) για τη σύγκριση δύο τυχαίων αριθμών από το 0 έως το 11 και την δυνατότητα επιλογής απεικόνισης του μεγαλύτερου ή του μικρότερου, κάθε φορά, αριθμού. (Υπόδειξη: χρήση *Case Structure, Tab Control*)
- 1.9 Τροποποιήστε το εικονικό όργανο (VI) του ερωτήματος 1.8 χρησιμοποιώντας το *Enum Constant* αντί του *Tab Control*. Ποιά είναι η διαφορά τους;
- 1.10 Σχεδιασμός εικονικού οργάνου (VI) για τη δημιουργία διανύσματος 12 στοιχείων με τυχαίους αριθμούς από το 0 έως το 70. (Υπόδειξη: χρήση *For-Loop, κατασκευή Array Indicator*) Στη συνέχεια υπολογίστε το μέσο όρο των τιμών του διανύσματος που δημιουργήσατε. (Υπόδειξη: χρήση *Add Array Elements, Array Size*)

- 1.11 Αφού απεικονίσετε τα στοιχεία του διανύσματος του ερωτήματος 1.10 σε ακέραια μορφή, να εξάγετε το 1^ο και 4^ο στοιχείο και να τα συγκρίνετε. Το 10^ο στοιχείο του διανύσματος να αντικατασταθεί από το μικρότερο εκ των δύο προαναφερθέντων στοιχείων. Στο *Front Panel* να απεικονίζονται (σε ακέραια μορφή): το διάνυσμα πριν την αντικατάσταση, το 1^ο και 4^ο στοιχείο του, το νέο 10^ο στοιχείο και το διάνυσμα μετά την αντικατάσταση. (Υπόδειξη: χρήση *Index Array*, *Select Function*, *Replace Array Subset*)
- 1.12 Να επαναληφθεί το ερώτημα 1.10 κάνοντας χρήση των *Shift Register* και *Build Array*. Τα στοιχεία του διανύσματος να απεικονίζονται σε ακέραια μορφή.
- 1.13 Σχεδιασμός εικονικού οργάνου (VI) για τη δημιουργία διδιάστατου (2D) πίνακα 6x5 με τυχαίους αριθμούς από το 0 έως το 15. Τα στοιχεία του διανύσματος να απεικονίζονται σε ακέραια μορφή. (Υπόδειξη: χρήση *For-Loop*, κατασκευή *2D Array Indicator*)
- 1.14 Στο εικονικό όργανο (VI) που δημιουργήσατε για το ερώτημα 1.13 κάντε χρήση του *Wait (ms)* δίνοντας του κάποια λογική τιμή π.χ. 400. Τι παρατηρείτε; Για ευκολία δημιουργήστε *Indicators* στα *Loop Iterations* των *For Loop* που χρησιμοποιήσατε. Πως επηρεάζει η θέση του *Wait Function* την εκτέλεση του VI; Τι διαφορά έχει το *Wait until Next ms Multiple* που χρησιμοποιήσατε στο ερώτημα 1.5 από το *Wait*;
- 1.15 Βασιζόμενοι στο VI του ερωτήματος 1.13, να βρείτε τον ανάστροφο του πίνακα που δημιουργήσατε. Έπειτα, δημιουργήστε νέο πίνακα που θα αποτελείται από τις 2 τελευταίες γραμμές και τις 4 τελευταίες στήλες του ανάστροφου. Στο *Front Panel* να απεικονίζονται: ο αρχικός, ο ανάστροφος και ο νέος πίνακας. Η απεικόνιση των στοιχείων τους να γίνει σε ακέραια μορφή. (Υπόδειξη: χρήση *Transpose 2D Array*, *Array Subset*)
- 1.16 Ένας αισθητήρας θερμοκρασίας δίνει ως έξοδο σήμα τάσης 0 έως 2V το οποίο «μεταφράζεται» σε εύρος μέτρησης -30 έως 80 βαθμών Celsius. Να σχεδιαστεί εικονικό όργανο (VI) που θα αναλαμβάνει την μετατροπή (*Scaling*) της τάσης σε θερμοκρασία. Έπειτα, τροποποιήστε το VI ώστε να μετατρέπει την θερμοκρασία από Celsius σε Fahrenheit. Για τη διευκόλυνση σας στην αναγωγή (*Scaling*) των μεγεθών από τη μία κλίμακα στην άλλη, συμβουλευτείτε το αντίστοιχο κομμάτι του Παραρτήματος. (Υπόδειξη: χρήση *Thermometer Indicator*)

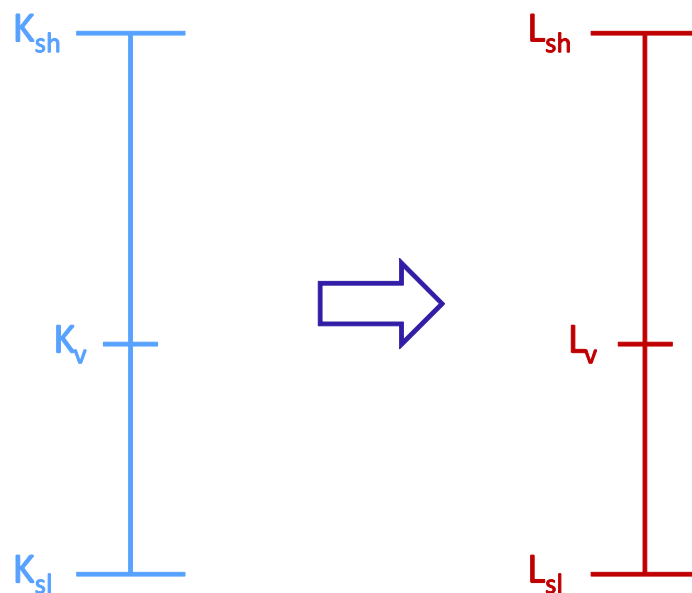
Παράρτημα

Αναγωγή κλίμακας (Scaling)

Η χρήση αναλογικών αισθητήρων συχνά περιλαμβάνει μετατροπές και αναγωγές των σημάτων που επιστρέφουν και περιέχουν τη μέτρηση που μας ενδιαφέρει. Οι περισσότεροι αισθητήρες επιστρέφουν τη μέτρηση σε μορφή σήματος τάσης ή ρεύματος, την οποία καλούμαστε να μετατρέψουμε στις μονάδες των μετρούμενων μεγεθών (π.χ. θερμοκρασία, απόσταση).

Για να γίνει η δυνατή η μετατροπή αυτή, θα πρέπει να γνωρίζουμε το εύρος μέτρησης του αισθητήρα σε μονάδες τάσης ή ρεύματος, καθώς και το εύρος στις φυσικές μονάδες στο οποίο αντιστοιχεί.

Παράδειγμα: ένας αισθητήρας θέσης μπορεί να μετρήσει απόσταση 30mm έως 300mm και τη μέτρηση την επιστρέφει ως σήμα τάσης εύρους 0V έως 10V. Έστω ότι η στιγμιαία μέτρηση που μας επέστρεψε ήταν 3V. Για να υπολογίσουμε την απόσταση στην οποία αντιστοιχούν τα 3V, κάνουμε χρήση του παρακάτω απλού τύπου:



$$L_v = \frac{(L_{sh} - L_{sl})(K_v - K_{sl})}{K_{sh} - K_{sl}} + L_{sl}$$

Όπου K_{sh}, K_{sl} είναι το εύρος μέτρησης (high, low, αντίστοιχα) σε μονάδες τάσης ή ρεύματος, L_{sh}, L_{sl} το εύρος μέτρησης (high, low, αντίστοιχα) σε φυσικές μονάδες και K_v, L_v οι τρέχουσες μετρήσεις σε μονάδες τάσης ή ρεύματος και φυσικές μονάδες, αντίστοιχα. Στην περίπτωση μας, με $K_{sh} = 10, K_{sl} = 0, L_{sh} = 300, L_{sl} = 30$, θα έχουμε:

$$L_v = \frac{(300 - 30)(K_v - 0)}{10 - 0} + 30 \Rightarrow L_v = 27K_v + 30$$

Επομένως, η ένδειξη $K_v = 3V$ αντιστοιχεί σε $L_v = 111mm$.