1.GAIA: ARAZOAK EBAZTEN

Labview zenbait arazo konpontzeko erabili daitekeen programazio-hizkuntza da. "Arazoak-ebazten" irtenbide horiek aurkitzeko nahitaezko teknikari dagokio. Ordenagailu programatzaileak software programen bidez, software garapen-metodoa erabiltzen dute arazoak ebazteko; metodo hauek programatzaileari laguntzen diote kode bat garatzeko.

Gai honi dagozkion atalak ondorengoak dira:

- A. Software garapen-metodoa.
- B. Arazoaren aurkezpena.
- C. Diseinua.
- D. Implementazioa.
- E. Frogapena.
- F. Mantenimendua.
- G. Kurtsoko proiektua.

A. <u>Software garapen-metodoa:</u>

Labview programak arazoak konpontzeko asmoz, software irtenbideak inplementatzeko erabiltzen duen metodoa da.

Metodo honetan, hurrengo pausuak betetzen dira:

- 1. Arazoaren aurkezpena.
- 2. Algoritmoa edo/eta fluxu-diagrama diseinatu.
- 3. Diseinuaren inplemetazioa: Pausu honen barnean kontroladorez eta adierazlez osaturiko aurreko panela eta funtzioez beteta dagoen bloke-diagrama izango ditugu.
- 4. Implementazioa frogatu eta egiaztatu: Honetarako, bloke-diagraman zenbait tresna gure esku ditugu.
- 5. Implementazioaren mantenimendua eta eguneraketa.

B. <u>Arazoaren aurkezpena:</u>

Software garapen-metodoaren etapa irauten duen bitartean, arazoaren mamia aurkeztuko dugu. Hau gauzatzeko, beharrezkoak izango ditugun zenbait faktore adieraziko dira.

C. <u>Diseinua:</u>

Arazoa adierazi bezain laster, arazoaren analisiaren bitartez irtenbide bat diseinatu dezaket. Analisi honen zati bat softwarearen sarrerak eta irteerak identifikatzea izango da. Sarrerak eta irteerak definitu ondoren, algoritmoa, fluxu-diagrama edo/eta egoeren diagrama sortu ahal izango da, honek software irtenbidera heltzera lagunduko digularik.

Sarreren ezagupena

Sarrerak arazoen ebazpena irauten duen bitartean prozesatu nahi ditudan datuak adierazten dituzte.

Irteeren ezagupena

Irteerak kalkulu edo prozesuen emaitza dira.

Gehigarrien ezagupena

Arazoaren ebazpenean eragin dezaketen beste zenbait faktore dira.

Arazoa konpontzeko algoritmoaren diseinua

Sarrerak, irteerak eta beste zenbait gehigarri adierazi eta gero, algoritmoa sortu daiteke. Algoritmoa sarrerak prozesatu eta irteerak sortzen dituen pausuen multzoa da.

Egoera diagramaren diseinua

Diagrama mota hauek LABVIEW egoeren makinak sortzerakoan gehienbat erabiltzen den algoritmoa da. Egoeren diagrama programaren egoerak argi adierazten digu eta horrek egoera ezberdinez osaturiko programak sortzea ahalbidetzen du. Gainera, borobil formako etiketak erabiltzen ditu egoera konstantea adierazteko; gezia egoera baten trantsizioa delarik.

Egoera baldintza bat betetzen duen programaren zati bat da. Trantsizioa, aldiz, hurrengo egoerara mugitzeko programak sortzen duen baldintza, akzio edo gertaera da. Programaren hasiera borobil solidoz eta amaiera borobil zuri baten bidez ematen da.

D. *Inplementazioa*:

Etapa honetan algoritmoaren edo fluxu-diagramaren kodea sortzen da. Labview programazio-hizkuntza grafikoa da; algoritmoa berdin-berdin lan egiten du.

E. Frogapena:

Frogapena eta egiaztapena software garapen-metodoaren zati garrantzitsua da. Inplementazioa datu mota egokiez osatuta dagoela ziurtatu beharko gara. Datu logikoen frogapenaren bidez, sarrerak espero dugun emaitza sortzen dutela egiaztatzen da. Datu zentzugabeen frogapenaz, aldiz, kodea errore-gestioa eraginkorra duen frogatu daiteke.

F. Mantenimendua:

Mantenimendua programazio-arazoak konpontzeko gaur egungo prozesua da eta arazoaren irtenbidearekiko paralelotasun handia duten aldaketak eransten ditu.

G. <u>Kurtsoko proiektua:</u>

Kurtso honetan zehar, kurtsoko proiektua kontzeptu ezberdinak jakinarazteko balioko digu, bai ariketa praktikoak bai kasuen ikasketa.

<u>Azterketa askea: Froga</u> Lotu software garapen-metodoak deskribatzen duten ondorengo pausu bakoitza bere deskribapen zuzenarekin.

1.	Arazoaren aurkezpena	A. Algoritmo edo fluxu-diagrama esleitu
2.	Diseinua	B. VI egiaztatu
3.	Inplementazioa	C. Arazoa definitu
4.	Frogaketa	D. VI eguneratu
5.	Mantenimendua	E. Sarrera eta irteeren ezagupena

<u>Azterketa askea: Frogaren erantzuna</u> Lotu software garapen-metodoak deskribatzen duten ondorengo pausu bakoitza bere deskribapen zuzenarekin.

1.	C. Arazoaren aurkezpena: Arazoa definitu.
2.	E. Diseinua: Sarrera eta irteeren ezagupena.
3.	A. Inplementazioa: Algoritmo edo fluxu-diagrama esleitu.
4.	B. Frogaketa: VI egiaztatu.
5.	D. Mantenimendua: VI eguneratu.

2.GAIA: LABVIEW ITSASKETA

Gai honek Labview programan itsasketari buruz azalpena ematen digu. Honetarako, hainbat baliagarritasun erabiltzen ditu: menuak, zenbait tresna, paletak, laguntza...Gainera, VI batek nola exekutatzen den ikasteaz gain, bloke-diagrama eta aurreko panelari buruzko ezaguera orokorra lortzen dugu. Gai honen amaieran datuak lortzen, aztertzen eta aurkezten dituen VI sinplea eraikiko da.

Gai honi dagozkion atalak ondorengoak dira:

- A. Instrumentu birtualak (VI-ak).
- B. VI hasiera.
- C. VI baten zati ezberdinak.
- D. Explorer proiektua.
- E. Aurreko panelaren leihoa.
- F. Bloke-diagramaren leihoa.
- G. Kontroladore, VI eta funtzioen bilaketa.
- H. Tresnen aukeraketa.
- I. Datuen fluxua.
- J. VI sinplearen eraikuntza.

A. Instrumentu birtualak(VI-ak):

Labview programak instrumentu birtualak izenaz ezagutzen dira, edo VI-ak, beren itxura eta operazioa instrumentu fisikoak imitazen baitituzte, osziloskopio eta multimetroak adibidez. Labview datuak lortzeko, aztertzeko, adierazteko eta biltzeko, VI eta funtzioen multzoak erabiltzen ditu, bere kodearen arazoak konpontzen laguntzeko dauden tresnak bezalakoak.

B. <u>VI hasiera:</u>

Gure ordenagailuan Labview programa irekitzerakoan, **Getting Started** leihoa azaltzen da. Leiho hau gauza askotarako erabiltzen da: VI eta proiektu berriak eraikitzeko, irekitako Labview artxibo berriak aukeratzeko, adibideak aurkitzeko, eta Labview laguntza bilatzeko. Labview programaren inguruan zenbait gauza ikasten laguntzeko, hainbat informazio eta baliabidetara sartu gaitezke, eskuliburu zehatz, gai laguntzaile eta *ni.com/manuales* errekurtsora adibidez.

Getting Started leihoa artxibo berria edo existitzen den artxiboa irekitzerakoan itxiko da; **View** >> **Getting Started Window** aukera hautatzean, leihoa erakutsi ahal izango delarik.



2-1 Irudia. LABVIEW Getting Started leihoa

Labview moldatzeko aukera izango dugu, berria den VI zuria irekiz. Tools >> Options hautatu, Category zerrendatik Environment hautatu, eta Skip Getting Started window on launch egiaztapen kutxan egiaztapen marka jarri.

<u>Oharra:</u> Getting Started leihoko gaiak aldakorrak izango dira, Labview bertsio eta instalatutako gidaren arabera.

VI edo proiektu baten irekiera edo sorketa

Labview programaren hasiera proiektu edo VI zuri baten bidez egingo dugu, existitzen den VI edo proiektu bat irekiz eta dagozkion aldaketak eginez, edo VI edo proiektu berri bat hasteko plantila baten bitartez.

VI edo proiektu berri baten sorketa

Getting Started leihoaren bidez proiektu berria ireki, New zerrendan Empty Project hautatu. Honela, izenik gabeko proiektu berri bat irekiko da eta artxibo berriak erantsi ahal izango dira, proiektu osoa gordez.

Proiektuarekin erlaziorik ez duen berria den VI zuria irekitzeko, Getting Started leihoan aurkitzen den New zerrendatik Blank VI hautatuko da.

Plantila batetik VI baten sorketa

File >> New aukera hautatuz, VI-en plantilak dituen New leiho bat izango dugu. Hala ere, leiho hau Getting started leihotik New lotura klikeatuz ere lortuko dugu.

eate New		Description	
🖌 🛅 VI	*		
🛃 Blank VI			
🖌 🗁 From Template			
a 🗁 Tutorial (Getting Started)			
📲 Generate, Analyze, and Display		Creates a new Virtual Instrument (VI).	
🦉 🦓 Generate and Display			
a 🛅 Simulated			
- 🌆 Load from File and Display			
Generate and Display			
a 🛅 Instrument I/O (GPIB)			
📲 Read and Display	=		
👍 🧰 Frameworks			
Top Level Application Using Events			
- 📲 SubVI with Error Handling			
- 🍇 Single Loop Application			
📲 Dialog Using Events			
📲 Dialog (Base Package)			
a 🛅 Design Patterns			
📲 User Interface Event Handler			
📲 Standard State Machine	1		
- 🍓 Queued Message Handler			
- 🍇 Producer/Consumer Design Pattern (Events)			
- 🍓 Producer/Consumer Design Pattern (Data)			
📲 Master/Slave Design Pattern			
🖌 🧰 User		Description	
Browse		Description	
🔤 🔜 Polymorphic VI	~		
m	•		



Existitzen den VI baten irekiera

Getting Started leihoan aurkitzen den Open zerrendatik Browse aukeratuko dugu.

<u>Aholkua:</u> Kurtso honetan gauzatuko dugun VI $C:\Exercises\Labview Basics I$ direktorioan aurkituko da.

Loading atalean memorian kargatzen ari diren VI-en subVI-ak azaltzen dira eta momenturarte kargaturiko subVI-ak erakusten ditu. Edozein unean karga anulatu ahal izango da, **Stop** botoian klikeatuz.

Ē	Using Temperature.vi	×
	Loading: E:\labview 2007-08\Solutions\LabVIEW Basics I\Shared Files\Read Voltage.vi	3 Loaded
	Searching: C:\Archivos de programa\National Instruments\LabVIEW 8.0\vi.lib\Platform\Ti	medLoop
	Ignore Item Browse Sto	р

2-3. Irudia. VI kargatzen ari dela adierazten duen elkarrizketa-kutxa

Labview ezin izango du subVI-a berehala aurkitu, VI bilaketa-bidetik direktorio zehatzak bilatzen hasiko gara. Honetarako, Tools >> Options hautatuz eta Category zerrendatik Paths aukeratuko dugu.

Labview **Ignore Item** botoia klikeatuz, subVI-a ez jakinduko du, edo galdutako subVI-a **Browse** botoiaren bidez bilatuko du.

VI-aren gordeketa

File >> **Save** hautatuz, VI berri bat gordeko dugu. Dena den, oraindik ez baldin bada VI-a gorde, **File** >> **Save as** leihoa lortu beharko da; leiho honekin VI baten kopia sortu ahal izango da, edo VI orijinala ezabatu eta berri batekin ordezkatu.

🛃 Save "T8	_P1.vi" As
Original file	
E:\B1\T6\V_7.	1\T8_P1.vi
• _	Copy - create copy on disk
	Substitute copy for original Copy will be in memory. Original will be closed.
	Create unopened disk copy Original will be in memory. Copy will not be opened.
	Open additional copy Both original and copy will be in memory. Copy must have new name.
	Rename - rename file on disk
	Duplicate hierarchy to new location Creates a source distribution build specification for this VI and its hierarchy.
	Continue Cancel Help

2-4 Irudia. Elkarrizketa-kutxa moduan gorde

C. <u>VI baten zati ezberdinak:</u>

Labview programa osatzen duten VI-ak hiru zati garrrantzitsuz bananduta daude: aurreko panelaren leihoa, bloke-diagrama eta paneleko ikur/lotura.

Aurreko panelaren leihoa

Leiho hau VI-rako erabiltzailearen interfazea da. Kontroladore eta adierazleen bidez osatuta dago, hauek VI-aren sarrera eta irteerei dagozkiolarik.



2-5 Irudia. VI baten aurreko panela

Bloke-diagramaren leihoa

Aurreko panela sortu eta gero, funtzioen errepresentazio grafikoa erabiliz kodea erantsiko dugu, panel honen objektuak kontrolatzeko helburuarekin. Gainera, aurreko panelean aurkitzen diren osagaiak, bloke-diagraman terminal bezala azaltzen dira diagramaren sarrerak edo irteerak adieraziz.



2-6 Irudia. Bloke-diagrama

Paneleko ikur/lotura

Zati honek VI-aren identifikazio grafikoa ematen du.

VI-a subVI bezala erabili daiteke. SubVI beste VI batean erabiltzen den VI-a da, programazio-hizkuntzan oinarritutako testu batean ageri den funtzioa baten antzekoa. VI-a subVI-a bezala erabiltzeko, paneleko ikur/lotura izan beharko du.



VI guztiek aurreko panel eta bloke-diagramaren goi-eskuineko aldean ikur bat izaten dute. Ikur honek VI-aren errepresentazio grafikoa ematen du eta testu zein irudiz osatuta egon daiteke. VI-a subVI bezala erabiltzen bada, VI-aren blokediagraman ikurrak subVI-a adieraziko du.

VI-a subVI bezala erabili ahal izateko, paneleko lotura bat eraiki beharko da; honek ikurraren terminal multzoa adierazten du eta VI-aren kontroladore eta adierazleei erreferentzia egiten dio, programazio-hizkuntzan oinarritutako testu batean ageri den parametroen zerrenda baten antzekoa. Ordenagailuko arratoiaren eskuineko botoiarekin aurreko panelean aurkitzen den ikurran klikeatuz, paneleko loturan sartu ahal izango gara (bloke-diagraman dagoen ikurratik ezin izango gara lotura honetan murgildu).

D. <u>Explorer proiektua</u>

Proiektu batek instrumentu birtual ezberdinak bildu (bai Labview bai ez-Labview artxiboak), zenbait espezifikazio eraiki eta artxiboak txartel batean zabaldu edo deskargatzen ditu. Proiektu bat gordetzerakoan, Labview proeiktuaren fitxategiluzapena sortzen du (*.lvproj*), proiektuan artxiboen aipamenak, hedapenaren informazioa... emanez.

Zenbait aplikazio eta liburutegi elkarbanatuak erabiltzeko, proiektua erabili beharko dugu. Gainera, denbora erreal (RT), field-programmable gate array (FPGA), edo digital zerbitzari digitalarekin (PDA) lan egiteko ere erabiliko da.

Explorer proiektuaren leihoa

Leiho hau Labview proiektuak sortu eta editatzeko erabiliko da. **File** >> **New Project** hautatuz, **Project Explorer** leihoa lortuko dugu. Hala ere, **New** elkarrizketa-kutxatik **Project** >> **New Project** edo **Empty Project** aukeratuz ere, leiho honetaz ohartu ahal izango gara.

Explorer proiektuaren leihoaren barnean, ondorengo gaiak aurkituko dira:

- **Project root** Etiketa hau proiektuaren izena dakar.
 - **My computer** Proiektuan ordenagailu lokala txartel baten moduan errepresentatzen du.
 - Dependencies
 - **Build Specifications** Labview Professional Development System edo Application Builder gure esku baldin badugu, stand-alone applications (EXEak), liburutegi elkarbanatuak (DLL-ak), instalatzaileak, eta zip artxiboak konfiguratu ahal izateko **Build Specifications** erabiltzeko aukera izango dugu.

Aholkua: Txartela VI-a exekutatu dezakeen edozein gailuari deitzen zaio.

Proiektuaren barnean beste txartel bat gehitzen duzunean, Labview txartela errepresentatzeko gai gehigarria sortzen du Explorer proiektuan. Honez aparte, txartel bakoitza **Dependencies** eta **Build Specifications** atalak eransten ditu. Txartel bakoitzaren azpian artxiboak gehitu daitezke.

Proiektuarekin erlazionatutako tresnen barra

Standard, Project, Build Specifications eta **Source Control** tresnen barra botoiak Labview proiektuan operazioak egiteko egiteko erabiliko dira. Tresnen barra Explorer proiektuaren goiko aldean erabilgarri izango dugu.



2-7 Irudia. Explorer Proiektuaren leihoa

<u>Aholkua:</u> Labview programan iturriaren kontrolari dagokion konfigurazioa erabiltzeko moduan baldin badugu **Source Control** tresnen barra izateko aukera izango dugu.

View >> Toolbars hautatuz, tresnen barra ezkutaketa edo agerpena lortuko da. Dena den, gauza bera lortzeko beste modu bat dago: ordenagailuaren eskuineko botoian klikeatuz.

Labview proiektu baten sorketa

Proiektu bat sortzeko, hurrengo bide ezberdinak jarraitu daitezke:

- 1. Explorer proiektuaren leihoa azaltzeko, **File** >> **New Project** hautatu. Hala ere, **New** elkarrizketa-kutxatik **Project** >> **Empty Project** ere aukeratuz, leihoa agertzea lortuko da.
- 2. Gai berriak gehitzeko proiektuan sartu ahal izango dira, txartelaren azpian.
- 3. Proiektua gordetzeko: File >> Save Project.

Existitzen diren artxiboak proiektu batean gehipena

Existitzen diren artxiboak proiektuan erantsi ditzakezu. VI-ak edo antzeko testu artxiboak Labview proiektu batean gehitzeko Explorer proiektuaren leihoan aurkitzen den **My Computer** gaia (edo beste txartel bat) erabili beharko duzu.

Hainbat gai proiektu batean eransteko, ondorengo aukerak egongo dira:

- Ordenagailuko arratoiaren eskuineko botoiarekin My Computer hautatu eta gero, Add File aukera hartuko dugu. Honez aparte, badago beste modu bat artxibo bat gehitzeko, Explorer proiektuaren menutik abiatuta: Project >> Add To Project >> Add File.
- Txartela eskuineko botoiarekin klikeatu eta karpeta bat eransteko atzipen zuzeneko menuan Add Folder hautatu beharko da. Helburu bera duen beste aukera bat ondorengoa litzateke: Project >> Add To Project >> Add Folder.

Oharra: Proiektu batekin disko batean karpeta bat gehitu ondoren, karpetan aldaketaren bat gauzatzen bada, Labview ez da proiektuko karpetan berehala eguneratuko.

- Txartela eskuineko botoiarekin klikeatu eta VI berri bat eransteko atzipen zuzeneko menuan New >> VI hautatu beharko da. Helburu bera duen beste aukera bat ondorengoa litzateke: File >> New VI edo Project >> Add To Project >> New VI.
- 5. Aurreko panelaren edo bloke-diagramaren goi-eskuineko aldean kokaturiko VI ikurra hautatu eta ikur hau txartelara arrastratu.
- 6. Zure ordenagailuko artxibo sistematik gai edo karpeta bat hautatu eta txartelara arrastratu.

Proiektu baten gaien ezabaketa

Explorer proiektu baten gaiak ezabatu ahal izateko, hurrengo bide ezberdinak jarraitu daitezke:

- Ezabatu nahi duzun gaia arratoiaren eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Remove** hautatu.
- Ezabatu nahi duzun gaia aukeratu eta <Delete> pultsatu.
- Ezabatu nahi duzun gaia aukeratu eta tresnen barra estandarretik **Delete** botoia klikeatu.

Oharra: Proiektutik gai bat ezabatuz, gai hori ez da diskotik desagertzen.

Proiektu baten gaien antolaketa

Explorer proiektuaren gaiak antolatzeko karpetak erabiltzen dira. Project Root eskuineko botoiarekin klikeatu edo txartela eta karpeta berri bat gehitzeko atzipen zuzenaren menutik **New >> Folder** hautatu.

Proiektu baten artxiboen erakusketa

Labview proiektu batera artxibo bat gehitzen denean, diskoan Labview artxibo bakoitzak erreferentzia bat dauka. Explorer proiektuaren leihoan aurkitzen den artxiboa arratoiaren eskuineko botoiarekin klikeatuz, eta atzipen zuzeneko menutik **Open** hautatuz, artxiboa bere editorean irekitzea lortuko dugu.

Proiektua eskuineko botoiarekin klikeatuz eta atzipen zuzeneko menutik **View** >> **Full Paths** hautatuz, proiektuaren artxiboak diskoan non dauden gordeta ikusiko dugu.

Project File Information elkarrizketa-kutxaren eta Explorer proiektuaren leihoaren bidez, proiektuaren artxiboak diskoan non dauden ikusteko aukera egongo da. **Project File Information** elkarrizketa-kutxa erakusteko, **Project** >> **File Information** aukera hartu beharko da. Dena den, helburu bera lortzeko ere, proiektuaren gainean eskuineko botoiarekin klikeatuz eta atzipen zuzeneko menutik **View** >> **File Information** hautatuz, **Project File Information** elkarrizketa-kutxaren erakusketa lortuko da.

Proiektu baten gordeketa

Labview proiektua gordetzeko hainbat aukera eskuragarri ditugu:

- **File >> Save Project** hautatu.
- **Project** >> **Save Project** hautatu.
- Proiektuaren gainean eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Save** hautatu.
- Proiektuaren tresnen barratik **Save Project** botoia klikeatu.

Proiektua gorde baino lehen, oraindik gordeta ez dauden atxibo edo artxibo berriak proiektuan gorde beharko dira. Proiektu bat gordetzerakoan, Labview ez ditu proiektuko menpeko artxiboak gordeko.

<u>Oharra</u>: Proiektuaren berrikusketa gauzatu behar denean, proiektuaren segurtasun kopia egin beharko da.

E. <u>Aurreko panelaren leihoa:</u>

Existitu edo berria den VI-a irekitzerakoan, VI-aren aurreko panelaren leihoa azaltzen da. Leiho hau VI-a erabiltzailearekin duen interfazea da.



2-8 Irudia. Aurreko panela lanabesekin batera

Kontroladore eta adierazleak

Aurreko panelaren leihoa kontroladore eta adierazleez osatuta dago, guzti hauek VIaren sarrera eta irteerei dagozkiolarik. Kontroladoreak, pultsadoreak, markatzaileak, eta beste zenbait gailu dira. Adierazleak, aldiz, grafikoak, LED-ak, eta beste hainbat motatako erakusleei dagozkie. Kontroladoreak instrumentaziozko sarrerako gailuak simulatu eta VI-aren bloke-diagrama datuaz hornitzen ditu. Adierazleak instrumentaziozko irteerako gailuak simulatu eta bloke-diagraman sortzen diren datuak erakusten ditu.

Erabiltzaileak sarrerako balioak aldatu eta irteerako balioak ikusi ditzake, aurretik aipaturiko zenbait baliabideen bitartez.

Kontroladore edo adierazle guztiek datu-mota zehatz batekin erlazionaturik daude. Gehien erabiltzen den datu-mota zenbakizkoa (numerikoa), boolearra eta string da.

Zenbakizko kontroladore eta adierazleak

Zenbakizko datu mota zenbait motatako zenbakiak adierazi ditzake, zenbaki osoa edo erreala bezalakoa. Zenbakizko objektu arruntenak, zenbakizko kontroladore eta adierazleak dira. Metro eta markatzaile bezalako objektuak zenbakizko datua adierazten dute.



Zenbakizko kontroladore batean balioak sartu edo aldatzeko, funtzionamendu lanabesarekin gehikuntza eta gehikuntzaren kontrako botoien gainean klikeatu edo funtzionamendu lanabesarekin etiketadun zenbakiaren gainean bi bidez klikeatu, zenbaki berria sartu, eta Intro tekla pultsatu.

Kontroladore eta adierazle boolearrak

Datu mota boolearrean datuaek bi egoera soilik adierazi ditzakete: EGIA (True) eta GEZURRA (False) edo ON (Martxan) eta OFF (Geldirik). Balio boolearrak sartu eta erakusteko asmoz, kontroladore eta adierazle boolearrak erabiltzen dira. Objektu boolearrak etengailuak, pultsadore eta LED-ak simulatzen dituzte.



2-10 Irudia. Kontroladore eta adierazle boolearrak

String motatako kontroladore eta adierazleak

Atring datu mota ASCII karaktereen sekuentzia da. Erabiltzailearengandik (pasahitza edo erabiltzailearen izena) testu bat jasotzeko, string kontroladorek erabiltzen dira. Erabiltzailearen testua erakusteko, aldiz, string adierazleak erabiltzen dira. Gehien erabilitako string objektuen artean taulak eta testua sartzeko kutxatilak ditugu.

String Control	Table		
			*
		<u> </u>	
String Indicator			
			T
	*	-	

2-11 Irudia. String kontroladore eta adierazleak

Kontrol paleta

Paleta honek aurreko panela sortzeko erabiltzen diren kontroladore eta adierazleez osatuta dago.

Paleta hau erabili ahal izateko, **View** >> **Controls Palette** aukeratuko dugu. Kontrol paleta zenbait mailatan banatuko da; bakoitzak lortu nahi duen helburuaren arabera, maila bat edo beste hautatzeko aukera egongo da. Kurtso honetan zehar, bereziki **Modern** izeneko mailarekin lan egingo dugu.



2-12 Irudia. Kontrol paleta

Maila hauek ezkutatu edo ikusteko, paletaren barnean aurkitzen den **View** botoia hautatu eta **Always Visible Categories** aukera hartuz edo hartu gabe utziko da.

Atzipen zuzeneko menuak

Labview programan agertzen diren objektu guztiek atzipen zuzeneko menua izaten dute. VI-a sortzerakoan, aurreko panel eta bloke-diagraman aurkitzen diren objektuen itxura eta portaera aldatu ahal izateko, atzipen zuzeneko menua osatzen duten gai ezberdinak erabiliko ditugu. Menu honetara sartzeko, ordenagailuko arrratoiaren eskuineko botoiarekin objektua klikeatu beharko da.

Propietateen elkarrizketa-kutxak

Objektuen portaera edo itxura aldatu ahal izateko, aurreko paneleko objektuek propietateen elkarrizketa-kutxak izaten dituzte. Objektu baten elkarrizketa-kutxa honetara sartzeko, objektua eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Properties** aukera hartu beharko da. Objektu batetarako erabiltzeko moduan dauden propietateen elkarrizketa-kutxaren eta atzipen zuzeneko kutxaren artean dauden aukerak oso antzekoak dira.

Appearance	Data Range	Scale	Format and Precision	Text Labels	Docume <
-Label			Caption		
🗹 Visible			Visible		
Meter					
-Enabled St	ate				
💿 Enable	:d				
🔵 Disable	ed				
🔵 Disable	ed & grayed				
Needle 1	-		A	dd De	lete
			Show digital display(s)		
			Show radix		
Nee	dle color				
Nee	dle color		Show increment/deci	rement buttons	
Nee	dle color		Show increment/deci	rement buttons	
Nee	dle color		Show increment/deci Show value tip strip	rement buttons	
Nee	dle color		Show increment/deci	rement buttons	
Nee	dle color		Show increment/deci	rement buttons	
Nee	dle color		Show increment/deci	rement buttons	

2-13 Irudia. Objektu baten propietateen elkarrizketa-kutxa

Aurreko paneleko leihoaren tresna-barra

Leiho bakoitzak berarekin erlazionatutako tresna-barra dauka. Aurreko paneneko tresna-barra-ean aurkitzen diren botoien bidez VI-a exekutatu eta editatuko dugu.

- VI-a exekutatzeko **Run** botoia klikeatu beharko da. Beharrezkoa bada, Labview VI-a konpilatuko du.
- VI-a exekutatzen dagoen bitartean, **Run** botoia kolore beltza izango du.

- VI-a subVI-a bezala exekutatzen ari bada, **Run** botoia kolore zuria hartzen hasiko da.
- Sortzen edo editatzen ari zaren VI-a erroreak baldin baditu, Run botoia apurtuta azalduko da. Bloke-diagramaren konekzioak egiten amaitzerakoan Run botoia oraindik apurtuta agertzen bada, VI-a apurtuta egongo da eta ezin izan da exekutatu. Errore eta aholku guztien zerrenda duen boti hau klikeatuz, Error list leihoa eskuragarri izango dugu.
- Exekuzioa moztuta edo geldirik ez dagoen bitartean, VI-a exekutatzeko Run Continuously botoia klikeatu beharko da. Berriro ere botoia klikeatuz, exekuzio jarraia desaktibatuko da.
- VI-a exekutatzen dagoen bitartean, **Abort Execution** botoia agertuko da. VI-a gelditzeko beste modu bat ez baldin badago, VI-a berehala gelditzeko, botoi honi eman beharko diogu.
- <u>Kontuz</u>!: VI-a oraingo iterazioa amaitu baino lehen, Abort Execution botoia VI-a berréala gelditzen du. Kanpoko errekurtsoak erabiltzen dituen VI-a mozteko, errekurtsoetatik egoera ezezagun bat atera daiteke. Stop botoia duen VI-a diseinatu beharko da arazo hau ekiditzeko.
- VI-aren exekuzioa gelditzeko Pause botoia klikeatu beharko da. Pause botoia klikeatzerakoan, Labview exekuzioa gelditu egin duen lekua azpimarratzen du bloke-diagraman, eta Pause botoia kolore gorria hartuko du. Pause botoia berriro ere klikeatu VI-a exekutatzen jarraitu dezan.
- Letra iturriaren konfigurazioa aldatzeko, **Text Settings** menua aukeratu beharko da, neurria, kolorea eta mota ere kontutan izanda.
- Ardatzetan zehar objektuak lerrokatzeko, Align Objects menua aukeratu beharko da, bertikalak, goiko ertza, ezkerrera eta horrela hurrenez hurren.
- Aurreko panelean aurkitzen diren zenbait objektuen tamaina aldatu eta guztiek tamaina bera izan dezaten, **Resize Objects** menua aukeratu beharko da.
- Bata bestearen artean elkarezartzen diren objektuak baldin baditugu eta horien artean atzean edo aurrean dagoena zein den definitu nahi denean, Reorder menua aukeratu beharko da. Posizionamendu lanabesarekin objektuetako bat hautatu eta gero Move Forward, Move Backward, Move To Front eta Move To Back artean egokiena aukeratu.
- **Context Help** leihoaren pantaila txandakatzeko, **Show Context Help Window** aukeratu beharko da.
- Zehaztuta dagoen balio bat beste balio batekin ordezkatu ahal dela gogoratzeko, Enter Text azalduko da. Enter Text botoia klikeatu bezain laster desagertuko da, Enter tekla pultsatu, edo aurreko panel edo bloke-diagramaren lan egiteko gunea klikeatu.

F. <u>Bloke-diagramaren leihoa:</u>

Bloke-diagramaren objektuak beste bloke-diagramen objektuen artean datuak transferitzen dituzten terminalez, subVI-ez, funtzioez, konstanteez, egiturez, eta kableez osatuta daude.

🥵 ALU_1.vi Front Panel *	
<u>File E</u> dit <u>V</u> iew <u>P</u> roject <u>O</u> perate <u>T</u> ools <u>W</u> indow	ALU_1
🗘 🐼 🦲 💵 13pt Application Font	
	^
N1 Resultado	
7/0,00	
N2	_
€ 0,00	-
Operación	
	~
< III	▶
😫 ALU_1.vi Block Diagram *	
<u>File E</u> dit <u>V</u> iew <u>P</u> roject <u>O</u> perate <u>T</u> ools <u>W</u> indow <u>H</u> elp	ALU_1
수 🐼 🔘 💵 😵 🕵 🛵 🖻 🔐 13pt Application	
	^
N11	
Resultado	
	=
	Ť

2-14 Irudia. Bloke-diagramaren adibidea eta dagokion aurreko panela

Terminalak

Aurreko panelaren objektuak bloke-diagraman terminal bezala azaltzen dira. Terminalak aurreko panel eta bloke-diagramaren artean informazioa elkartrukatzen duten sarrerako eta irteerako portuak dira. Terminalak programazio-hizkuntzetan testuetan oinarritutako parametro eta konstanteekiko antzekoak dira. Terminal moten artean, kontroladore edo adierazleen terminalak eta nodoen terminalak bereiztu daitezke. Bloke-diagramaren funtzio baten terminalak erakusteko, funtzioa eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Visible Items >> Terminals** hautatuko dugu.

Kontroladoreak, adierazleak eta konstanteak

Kontroladoreak, adierazleak eta konstanteak bloke-diagramako algoritmoaren sarrera eta irteeren portaera bera dute.

Terminalak ikur edo ikur gabe adierazi daitezke.



2-15 Irudia. Triangelu baten azaleraren aurreko panela

Bloke-diagramaren nodoak

Sarrerak eta/edo irteerak dituzten eta VI-a exekutatzen denean operazioak egiten dituzten bloke-diagraman aurkitzen diren objektuak dira. Nodoak programaziohizkuntzetan testuetan oinarritutako deklarazio, funtzio,...ekin konparatuz oso antzekoak dira. Nodoak subVI-ak, egiturak edo funtzioak izan daitezke. Egiturak prozesuak kontrolatzen dituzten elementuak dira, Case egitura, For edo While bukleak besteak beste.

Funtzioak

Funtzioak Labview programaren barnean oinarrizko elementuak dira eta ez dituzte aurreko paneleko leihorik ezta bloke-diagramaren leihorik ere; dena den, paneleko lotura eskuragarri dute. Funtzioaren gainean bi bider klikeatuz, funtzioaren aukeraketa da soilik lortzen duguna. Funtzioen ikurra hori kolorekoa da.

SubVI-ak

SubVI-ak beste VI baten barnean erabiltzeko eraikitzen diren edo funtzioen paletaren bitartez sartu gaitezkeen VI-ak dira.

VI batek subVI modura erabiltzeko lamen nahikoa dauka. Bloke-diagraman aurkitzen den subVI-aren gainean bi bider klikeatuz, honen aurreko panela azaltzen da; panel hau dagozkien kontroladore eta adierazleez osatzen da. Bloke-diagrama kableez, ikurrez, funtzioez, subVI-ez, eta beste zenbait objektuez osatuta dago. Aurreko panel eta bloke-diagramaren goi-eskuineko ertzean VI-aren ikurra erakusten da; VI bat subVI bat bezala bloke-diagraman kokatzen denean, hau da azaltzen den ikurra.

SubVI-ak ere Express VI-ak izan daitezke. Express VI-ak elkarrizketa-kutxekin konfiguratzen direnez, kable minimoak behar dituzten nodoak dira. Neurketa zeregin arruntenak egiteko Express VI-ak erabiltzen dira. Express VI baten konfigurazioa subVI bat bezala gorde daiteke. *Labview Help* ataletik *Express VI* gaiaz baliatuz, Express VI konfigurazio batetik subVI bat sortzeko informazio gehiago lortu ahal izango dugu.

Labview bloke-diagramako Express VI eta gainontzeko VI-en artean ezberdintzeko, kolore ezberdinetako ikurrak erabiltzen du. Express VI-ak kolore urdina eta subVI-ak kolore horia izaten dute bloke-diagraman.

Nodo hedatuak ikurren aurrean

VI-ak eta Express VI-ak ikur edo nodo hedatuak bezala erakutsi daitezke. Nodo hedatuak ikur bezala azaltzen dira baina kolore batez inguratuak daude. SubVI-ak kolore horia eta Express VI-ak kolore urdina izaten dute. Bloke-diagraman leku gehiegi okupatu nahi ez dugunean, ikurrak erabiltzen ditugu. Nodo hedatuak, aldiz, kableatua errazagoa eta bloke-diagramen dokumentazioan laguntzeko baliagarriak dira. Normalean subVI-ak ikur bezala eta Express VI-ak nodo hedatuak bezala azaltzen dira bloke-diagraman. SubVI edo Express VI-a nodo hedatua bezala erakusteko, dagokion VI-a eskuineko botoiarekin klikeatu eta egiaztapen-marka atzipen zuzeneko menuan aurkitzen den **View As Icon** gaiaren ondora mugitu beharko da.

Kableatua errazagoa izateko, nodo hedatua tamainaz aldatu daiteke, hala ere, askotan bloke-diagraman leku gehiegi okupatzen du bloke-diagraman. Ondorengo pausuak bete beharko dira bloke-diagramako nodoaren tamaina aldatzeko:

- 1. Posizionamendu lanabesa nodoaren gainera eraman. Nodoaren goiko eta beheko aldeak tamainaz aldatu.
- 2. Kurtsorea mugitu kurtsore honen tamaina aldatu ahal izateko.
- 3. Tamainuz aldatutako kurtsorea erabili nodoaren beheko aldea beherantz mugitzeko; honela, gehigarrizko terminalak erakutsiko dira.
- 4. Ordenagailuko arratoiaren botoia askatu.

Tamainuz aldatzeko prozedura deuseztatzeko, arratoiaren botoia askatu baino lehen nodoaren ertza bloke-diagramaren ondora mugitu beharko da.

DBL	> 1.23
	DBU

2-16 Irudia. Kontroladore bata adierazteko moduak

<u>Oharra:</u> SubVI edo Express VI-a nodo hedatua bezala ikusi nahi bada, ezin izango dira nodo horren terminalak erakutsi ezta nodo horretarako datu baseen sarrera baimendu.

Kableak

Bloke-diagramaren barnean datuak transferitzeko, kableak erabiliko ditugu. Kable bakoitzak datuen iturri bakarra izango du baina VI eta datuak irakurtzen duten funtzio ezberdinetara konektatu daiteke. Kableak kolore, itxura, eta lodiera ezberdina izaten dute, datu motaren arabera.

Apurtuta agertzen diren kableak gorriz koloreko X bat erdian duten lerro beltz ezjarraiak dira. Kable mota hauek hainbat arrazoiengatik sortzen dira, datu mota ezberdina duten bi objektu lotzen ari direnean adibidez.

2-17 Irudia. Apurtutako kable baten itxura

Labview programan, VI-aren barnean datuak heda daitezen, zenbait terminal elkarren artean konektatzeko kableak erabiltzen dira. Honela, datu mota bateragarria duten sarrera eta irteerak erlazionatu ahal izango ditugu; adibidez, ezin izango da irteera adierazten duen matrize bat zenbakizko sarrera batekin kableatu. Gainera, kableen norabideak ere zuzenak izan beharko dira. Kableak sarrera bakar batera eta gutxienez irteera batera konektatu beharko dira; adibidez, ezin izango dira bi adierazle batera kableatu. Kableen bateragarritasuna zehazten duten konponenteak kontroladore eta/edo adierazleen datu mota eta terminalaren datu mota izaten dute.

Datu motak

Datu motak batera kableatu daitezkeen objektuak, sarrerak eta irteerak adierazten digu. Adibidez, etengailu batek ertz berdea baldin badu, etengailua etiketa berdea duen edozein sarrerarekin konektatutu daiteke Express VI batean. Kableak terminalaren kolore berdina dutela kontutan hartu beharko da.

Automatikoki kableatutako objektuak

Bloke-diagramako objektuak mugitzeko ordenagailuko arratoiaren botoia askatzen denean, Labview kableak automatikoki konektatzen ditu. Dena den, blokediagraman ere objektuak automatikoki konektatu daitezke. Labview programak hoberen egokitzen diren terminalak konektatuta eta bat etortzen ez diren terminalak aske uzten ditu.

Posizionamendu lanabesa erabiliz objektu bat mugitzen ari zaren momentu berean, Intro barra presionatuz kableatu automatikoa aldatuko dugu.

Eskuz kableatutako objektuak

Kableatuaren lanabesa terminalaren gainetik pasatzerakoan, terminalaren izena azaltzen da. Gainera, terminala **Context Help** leihoan ñir-ñir egingo du eta ikurrak terminal egokian konektatu duzula egiaztatzen lagunduko digu. Objektuak elkarrekin konektatzeko, ondorengo prozedura bete beharko da: lehenengo terminaletik kableatuaren lanabesa pasatu, klikeatu, kurtsorea bigarren terminaletik pasatu, eta klikeatu berriz ere. Konekzioa egin eta gero, arratoiaren eskuineko botoiarekin kablea klikeatuz eta atzipen zuzeneko menutik **Clean Up Wire** hautatuz, Labview automatikoki kablearentzako bide bat aukeratuko du. Apurtutako kableak baldin badaude, bloke-diagramako kable guzti hauek ezabatzeko, <Ctrl-B> tekla presionatuko dugu.

Funtzioen paleta

Paleta hau bloke-diagrama sortzeko erabiltzen diren VI-ez, funtzioez eta konstanteez osatuta dago. Bloke-diagramatik funtzioen paletara sartzeko, **View > Functions Palette** aukeratu beharko da. Paleta hau zenbait mailatan zatitu daiteke; beharren arabera maila hauen erakusketa edo ezkutaketa egongo da. Kurtso honetan zehar, batez ere **Programming** izeneko mailarekin lan egingo dugu nahiz eta gainerakoak ere erabiliko ditugun.



2-18 Irudia. Funtzioen paleta

Maila hauek ezkutatu edo ikusteko, paletaren barnean aurkitzen den **View** botoia hautatu eta **Always Visible Categories** aukera hartuz edo hartu gabe utziko da.

Bloke-diagramaren tresna-barra

VI-a exekutatzean, VI-a garbitzeko bloke-diagramaren tresna-barra-ean zenbait botoi azaltzen dira.

• Run botoia zapaltzerakoan, bloke-diagramaren exekuzioan animazioa erakusteko, **Highlight Execution** botoia klikeatu beharko da. Diagram honen datuen fluxua kontutan hartu beharko da. Exekuzio hau desaktibatzeko berriz ere botoi honi emango genioke.

- **Retain Wire Values** botoian klikeatu fluxu exekuzioko puntu bakoitzean kableen baloreak gordetzeko, horrela kablean azterketa bat gauzatzen denean kabletik igarotako datuen azkenengo balioa berehala gelditu ahal izanog du.
- Nodoa ireki eta gelditzeko Step Into botoia klikeatu beharko da. Botoi hau berriro ere zapaltzen baldin bada, lehenengo akzioa exekutatuko litzateke eta subVI edo egituraren hurrengo akzioan geldituko litzateke. <Ctrl> presionatuz eta beheranzko norantza duen geziarekin ere lortu genezake.
- Nodo bat exekutatu eta hurrengo nodoan gelditu dadin, **Step Over** botoia klikeatu beharko da. <Ctrl> presionatuz eta eskuineranzko norantza duen geziarekin ere gauza bera lortu genezake.
- Oraingo nodoa exekutatzen amaitu eta gelditu dadin, Step Out botoia zapaldu beharko da. VI-a exekutatzen bukatzen duenean, botoi hau kolore iluna izango du. <Ctrl> presionatuz eta goranzko norantza duen geziarekin ere lortuko genuke.
- VI-a oharra baldin badu eta **Error List** leihoan dagoen **Show Warnings** egiaztapen-kutxan egiaztapen-marka egiten badugu, **Warning** botoia azalduko da. Oharra bloke-diagramarekin arazo handia dagoela adierazten du, hala ere, nahiz eta oharra izan, exekutatzen ari den VI-a ez da geldituko.

G. Kontroladore, VI eta funtzioen bilaketa:

View >> **Controls** edo **View** >> **Functions** aukeratzerakoan, kontrol eta funtzioen paletak ireki egingo dira; paleta bakoitzaren goiko aldean bi botoi agertuko dira.

- **Search** Paletetan aurkitzen diren kontroladoreak, VI-ak edo funtzioak bilatzeko, paleta search modura aldatu beharko da. Paleta modu honetan dagoen bitartean, modu honetatik aterako **Return** botoiari eman beharko diogu eta honela, berriz ere paletara bueltatuko da.
- View Paleta guztien mailak erakutsiz eta ezkutatuz, eta Text eta Tree formatuetan gaiak alfabetikoki banatuz, momento horretan daukagun paletaren formatua hautatzeko zenbait aukera izango ditugu. View botoia klikeatuz eta atzipen zuzeneko menutik Options hautatuz, Options elkarrizketa-kutxaren Controls/Functions Palettes mailak erakutsi daitezke. Botoi hau paletaren goiezkerraldean aurkitzen den txintxeta soilik klikeatuz aterako da.

Functions	
🔍 Return 🛛 👫 View	
random	
Continuous Random.vi	<u>^</u>
Periodic Random Noise PtByPt.vi Periodic Random Noise PtByPt.vi Periodic Random Noise.vi Random Number (0-1) < <express numeric<br="">Random Number (0-1) <<numeric>> Simulate Signal <<input/>> Simulate Signal <<signal analysis="">> Simulate Signal <<waveform generation=""></waveform></signal></numeric></express>	>>
Search Options	Help

2-19 Irudia. Funtzioen paletan objektu bat bilatzen

Funtzio eta VI-ekin harreman nabaria izan arte, nahi ditugun funtzio edo VI-ak bilatzeko **Search** botoia erabiliko dugu. Adibidez, Random Number funtzioa bilatu nahi baldin baduzu, funtzioen paletan agertzen den **Search** botoia klikeatu eta paletaren goiko aldean Random Number testua idatziz, gure helburua lortuko dugu. Bilaketa emaitzetatik interesatzen zaiguna bloke-diagramara eramanez edo objektua bi bider klikeatuz bilaketa amaitutzat emango da. Objekturen bat maiz erabiltzen baldin bada, **Favorites** mailara eramatea oso ohikoa da; hau paletaren objektua arratoiaren eskuineko botoiarekin klikeatuz eta **Add Item to Favorites** hautatuz lortuko genuke.

Functions	×
🔍 Search 🛯 🗂 View 🚽	
Programming	-
Instrument I/O	
I ▼ Mathematics	
Add Subtract	=
Multiply Divide	
Quotient & R Conversion	
Increment Decrement	
Add Arra Add Item to Eavorites	
Examples	
Compound Ar Data Manipul	
Absolute Value Round To Ne	
Round To -In Round To +I	~

2-20 Irudia. Objektu bat Favorites mailara eramango da

H. <u>Tresnen aukeraketa:</u>

VI-en sorketa, aldakutza eta garbiketa Labview programak eskainitako lanabesen bitartez lortuko dugu. Lanabesa kurtsorearen funtzionamenduaren modu berezia da; modu hau aukeratutako lanabesaren ikurrari dagokio.

t	Too	ls –	×	X
E	X			_1
L	∿•	Þ.	Ąĭ	Ы
E	*	×.	$\langle n \rangle$	
	۱	ŧ₽	A	
		<u> </u>		
H				~
1				

2-21 Irudia. Lanabesen paleta

<u>*Aholkua:*</u> Lanabesen paletan nahi dugun lanabesa eskuz aukeratu daiteke. **View** >> **Tools Palette** hautatuz, **Tools** paleta gure esku izango dugu.

Funtzionamendu lanabesa

Arratoiaren kurtsorea esku baten ituxrako ikur batean aldatzen denean, funtzionamendu lanabesa erabiltzen ari garela esaten da. Lanabes mota hau kontroladoreen balioa aldatzeko erabiltzen da.

Funtzionamendu lanabesa aurreko paneleko leihoan erabiltzen da gehienbat, hala ere, bloke-diagraman ere erabili daiteke gehikuntza eta gehikuntzaren kontrako botoiak zapaldu eta kontroladoreen baliok aldatzeko.

Posiozionamendu lanabesa

Ordenagailuko arratoiaren kurtsorea gezi baten itxura hartzen duenean, lanabes mota hau erabiltzen ari dela esaten da. Posizionamendu lanabesa objektuak hautatu edo tamainuz aldatzen ditu. Objektu bat aukeratu ondoren, mugitu, kopiatu edo ezabatu daiteke.

Lanabes hau bai aurreko panelean bai bloke-diagraman erabili daiteke.

Etiketen lanabesa

Ordenagailuko arratoiaren kurtsorea ondorengo itxura hartzen duenean, etiketen lanabesa erabiltzen ari garela esaten da. Lanabes mota hau zenbait gauzetarako erabilgarria izango da: kontroladoreetan testua sartzeko, testua editatzeko eta etiketa askeak sortzeko.

Modu zehatz bat sartzea derrigortzen ez duen aurreko paneleko edo bloke-diagramako gune konkretu batean gaudenean, kurtsorea gurutzea baten itxura izaten du. Gurutze modu hau aktibatuta dagoenean, etiketen lanabesa sartzeko bi bider klikeatu eta etiketa askea sortzea lortuko da.

Kableatuaren lanabesa

Arratoiaren kurtsorea haridun bobina baten itxura hartzen duenean, kableatuaren lanabesa erabiltzen ari da. Lanabes mota hau bloke-diagraman objektuak elkarren artean konektatzeko erabiltzen da.

Kableatuaren lanabesarekin lan egiten ari garenean, normalean bloke-diagraman lan egiten ari gara baina paneleko lotura sortuta badago, aurreko panelean ere erabiltzen da.

Paletan sartuta dauden beste zenbait lanabes mota

Lanabes paletaren bitartez, funtzionamendu, posizionamendu, kableatu eta etiketen lanabesetara zuzenean sartu gaitezke, modu automatikoko hautaketa lanabesa erabili beharrean.



2-22 Irudia. Lanabesen paleta

- Lanabes paletaren goiko aldean modu automatikoko hautaketa lanabesa dago. Modu hau aukeratzean, Labview zuzenean kurtsoreari kokapenaren arabera dagokion lanabes mota hautatu dezake. Dena dela, lanabes mota hau itzali daiteke; lanabes hau hautatu gabe utziz edo paneleko beste gai bat hartuz. Honez aparte, badaude beste lanabes gehigarri batzuk:
- Objektuaren atziopen zuzeneko menuaren lanabesa erabili arratoiaren eskuineko botoiarekin objektuaren atzipen zuzeneko menura sartzeko.
- Desplazamendu barra erabili gabe leiho ezberdinen artean desplazatzeko, desplazamendu lanabesa erabili beharko da.
- Puntuaren lanabesa VI, funtzio, nodo, kable, eta egituretan puntua finkatua dagoen eremuan bertan exekuzioa gelditzeko erabili beharko da.
- Azterketa lanabesa bloke-diagramako kableetan ikerketak sortzeko erabili beharko da. Lanabes mota hau espero ez diren emaitzak sortzen duten VI-aren bitarteko balioak egiaztatzeko baliogarria izango da.
- Koloreen lanabesarekin baliatuz, koloreak kopiatu eta gero horiek itsasteko, koloreak kopiatzeko lanabesa erabili beharko da.
- O Objektu bat koloreztatzeko koloreen lanabesa erabiliko da.

I. <u>Datuen fluxua:</u>

Labview kableen bidez konektatutako nodo exekutableetan datu-fluxuaren programazio sistema jarraitzen du, non nodoen artean aurkitzen diren kableak nodo batek sortutako datuak beste nodo batek erabiltzen duela adierazten du. Nodoak beharrezko sarrerako datuak jasotzean exekutatu daitezke eta baita, bloke-diagraman irteerako datuak produzitu beste nodo batzuei.

Visual Basic, C++, JAVA, eta testuan oinarritutako programazio-hizkuntza gehienak programaren kontrol fluxuaren modeloa jarraitzen dute. Kontrol fluxuan, programako elementuen orden sekuentziala, programaren exekuzio ordena zehazten du.



2-23 Irudia. Datuen fluxuaren adibidea

J. <u>VI sinplearen eraikuntza:</u>

Labview programako VI gehienak hiru zeregin izaten dituzte: zenbait datu mota eskuratu, eskuratutako datuen analisia eta lortutako emaitzaren aurkezpena. Zati hauetariko bakoitza oso sinplea denean, VI osoa bloke-diagraman oso objektu gutxi ipinita osatu daiteke. Express VI-ak batez ere honetarako diseinatzen dira, operazioak sarritan erabiltzen direlarik. Atal honetan, zenbait Express VI ikasiko ditugu, hauetariko bakoitza ondorengo hiru mailatan: eskuraketa, analisia eta aurkezpena. Ondoren, hiru zati ezberdin hauek erabilita, VI sinplearen eraikuntza nola gauzatzen den ikasiko dugu.

Funtzioen paletan Express VI guztiek batera taldekatzen dira; **Express** izeneko mailan hain zuzen ere. VI mota hauek datu mota dinamikoa erabiltzen dute elkarren artean informazioa igarotzeko.

Eskuraketa

Eskuraketaren zeregina gauzatu ahal izateko, Express VI-ak ondorengo baliabideak erabiltzen dituzte: DAQ Assistant, Instrument I/O Assistant, Simulate Signal eta Read from Measurement File.

DAQ Assistant

DAQ Assistant datuen eskuraketa dispositiboaren arabera datuak eskuratzeko gai da. Kurtso honetan zehar termino hau hainbat kasutan erabiliko da. Horrexegatik, datuen eskuraketari buruz zerbait gehiago ikasi arte, datuen eskuraketa dispositiboan kanal bakarra erabiliko da, CH0. Kanal hau DAQ Signal Accessoryan tenperatura sentsorearekin konektatuta egongo da; sentsore honen balioa aldatuz, tenperaturaren irakurketa aldaketak pairatuko ditu.

Instrument I/O Assistant

Instrument I/O Assistant instrumentuen datuen kontrola eskuratzen dute, normalean GPIB edo serie interfaze baten bidez baliatzen dira.

Simulate Signal

Simulate Signal Express VI-a datu simulatuak sortzen ditu, sinu seinale bat bezala.

Read From Measurement File

Read From Measurement File Express VI-a Write To Measurement File Express VI-aren bidez sortutako artxiboa irakurtzen du. Hau normalean LVM edo TDM formatua duten artxiboak irakurri ditzake, Express VI honek ASCII artxiboak ezin ditu irakurri.

Analisia

Analisiaren zeregina gauzatu ahal izateko, Express VI-ak ondorengo baliabideak erabiltzen dituzte: anplitude eta mailaren neurketa, estadistikak, tonuen neurketa,..

Amplitude and Level Measurements

Amplitude and Level Measurements Express VI-a seinaleen tentsio neurketa egiten dute. Honek DC, rms, tontor maximoa, tontor minimoa, tontorraren tontorra, zikloaren batezbestekoa, eta zikloaren rms neurketak egiten ditu.

Statistics

Statistics Express VI-a uhinaren datu estadistikoak kalkulatzen ditu. Honek batezbestekoa, batuketa, desbiazio tipikoa eta muturreko balioak lortzen ditu.

Spectral Measurements

Spectral Measurements Express VI-a uhinari dagokion espektruaren neurketa gauzatzen du, magnitude eta potentziaren dentsitate espektrala bezala.

Tone Measurements

Tone Measurements Express VI-ak maiztasun edo anplitude maximoko tonuaren bilaketa gauzatzen du. Hala ere, tonu bakarraren maiztasun eta anplitudea ere aurkitu dezake.

Filter

Filter Express VI-a iragazki eta leihoen bitartez seinalea prozesatzen du. Honetarako erabiltzen diren iragazkiak Highpass, Lowpass, Bandpass, Bandstop eta Smoothing eskuragarri izaten dute. Leihoek, aldiz, Butterworth, Chebyshev, Inverse Chebyshev, Elliptical eta Bessel.

Aurkezpena

Azkenengo zeregin honen helburu nabarmena ondorengoa izango litzateke: Write To Measurement File Express VI-a bezalako funtzioa gauzatzen edo aurreko panelean datuak aurkezten dituen Express VI-a erabiliz emaitzen aurkezpena lortzea da. Zeregin honetarako normalean erabiltzen diren adierazleak Waveform Chart, Waveform Graph eta XY Graph izaten dira. Gehien erabilitako Express VI-en artean Write To Measurement File Express VI-a, Build Text Express VI-a, DAQ Assistant eta Instrument I/O Assistant daude. Kasu honetan, DAQ Assistant eta Instrument I/O Assistant ordenagailutik DAQ dispositibo edo kanpoko instrumentu batetara irteerako datuak eskaintzen dizkio.

Write To Measurement File

Write To Measurement File Express VI-a LVM edo TDM formatua duen artxiboa idazten du.

Build Text

Build Text Express VI-a testua sortzen du, aurreko panelean erakutsi edo artxibo edo instrumentu batera esportatzeko.

VI-aren exekuzioa

- Express VI-ak konfiguratu eta denak batera konektatu ostean, VI-a exekutatu daiteke; VI-a eraikitzen bukatzerakoan, VI-a exekutatu dadin tresna-barra-ean aurkitzen den **Run** botoia klikeatu beharko da.
- VI-a exekutatzen ari den bitartean, **Run** botoiaren gezia kolore beltza hartuko du. Dena den, exekuzioa bete ondoren, botoi hau berriro ere itxura arruntera itzuliko da, eta aurreko paneleko adierazleak datuak izango dituzte.

Run botoiaren erroreak

VI-a exekutatzen ez baldin bada, apurtuta dagoela adieraziko du. Sortzen edo aditatzen ari garen VI-a erroreak dituenean, **Run** botoia apurtuta agertuko da.

Botoi hau bloke-diagraman konekzioak egiten bukatzerakoan eraindik apurtuta mantentzen bada, VI-a apurtuta dagoela adieraziko du eta beraz, ezin izango da exekutatu.

Normalean honek sarrerak elkarren artean konektatuta ez daudela edo kablea apurtuta dagoela esatera eramango gaita. Broken run botoia zapaltzean, **Error List** elkarrizketa-kutxara sartzeko aukera izango dugu. Kutxa honek erroreak aipatu eta arazoen deskribapena ematen du.

Azterketa askea: Froga

Hurrengo irudia kontutan izanda, froga honi dagozkion ondorengo galderak erantzun:



2-24 Irudia. Datuen fluxuaren galderak

- 1. Funtzio hauen artean, zein exekutatzen da lehena?
 - a. Batuketa.
 - b. Kenketa.
 - c. Ezezaguna.
- 2. Funtzio hauen artean, zein exekutatzen da lehena?
 - a. Sinua.
 - b. Zatiketa.
- 3. Funtzio hauen artean, zein exekutatzen da lehena?
 - a. Ausazko zenbakia.
 - b. Zatiketa.
 - c. Batuketa.
 - d. Ezezaguna.
- 4. Funtzio hauen artean, zein exekutatzen da azkena?
 - a. Ausazko zenbakia.
 - b. Kenketa.
 - c. Batuketa.
 - d. Ezezaguna.
- 5. Zeintzuk dira VI baten hiru zatiak?
 - a. Aurreko panelaren leihoa.
 - b. Bloke-diagramaren leihoa.
 - c. Proiektua.
 - d. Paneleko ikur/lotura.

Azterketa askea: Frogaren erantzuna



- 1. Funtzio hauen artean, zein exekutatzen da lehena?
 - a. Batuketa.
 - b. Kenketa.
 - c. Ezezaguna.
- 2. Funtzio hauen artean, zein exekutatzen da lehena?
 - a. Sinua.
 - b. Zatiketa.
 - c. Ezezaguna.
- 3. Funtzio hauen artean, zein exekutatzen da lehena?
 - a. Ausazko zenbakia.
 - b. Zatiketa.
 - c. Batuketa.
 - d. Ezezaguna.
- 4. Funtzio hauen artean, zein exekutatzen da azkena?
 - a. Ausazko zenbakia.
 - b. Kenketa.
 - c. Batuketa.
 - d. Ezezaguna.
- 5. Zeintzuk dira VI baten hiru zatiak?
 - a. Aurreko panelaren leihoa.
 - b. Bloke-diagramaren leihoa.
 - c. Proiektua.
 - d. Paneleko ikur/lotura.

<u>3. GAIA: VI BATEN ARAZOEN KONPONBIDEA ETA</u> <u>GARBIKETA</u>

VI-a exekutatu ahal izateko, erabilitako subVI, funtzio eta egitura guztien terminalak datu mota zuzenarekin konektatuta egon beharko dira. Hala ere, batzuetan VI-aren exekuzioa edo datuen lorpena espero ez dugun moduan egiten da. Labview programak VI-aren exekuzioa nola konfiguratzen den edo bloke-diagramako organizazio edo bloke-diagramaren bidez igarotako datuen arazoak izendatzen ditu.

Gai honi dagozkion atalak ondorengoak dira:

- A. Labview laguntzaren erabilpena.
- B. Apurtutako VI-en zuzenketa.
- C. Garbiketa egiten duten teknika ezberdinak.
- D. Definitu gabeko edo espero ez diren datuak.
- E. Erroreen egiaztapena eta gestioa.

A. *Labview laguntzaren erabilpena:*

VI bat sortu eta editatzeko hiru aukera ditugu: **Context Help** leihoa, *Labview Help* eta NI Example Finder.

Context Help leihoa

Objektu bakoitzaren gainetik arratoiaren kurtsorea mugitzen denean, **Context Help** leiho honek Labview objektuen oinarrizko informazioa erakusten du. **Context Help** leihoaren erakusketa txandakatzeko, **Help** >> **Show Context Help** aukeratu, <Ctrl-H> tekla pultsatu edo tresna-barra-en **Show Context Help Window** botoia klikeatu beharko da.

Aurreko panela eta bloke-diagramaren objektuen gainetik kurtsorea mugitzerakoan, **Context Help** leihoak subVI, funtzio, konstante, kontroladore, eta adierazleen ikurra erakutsiko du, terminal bakoitzera kablea bat konektatuta egonik. Elkarrizketa-kutxaren aukeren gainetik kurtsorea mugitzen denean, **Context Help** leihoak aukera hauen deskripzioa ematen du.

Context Help leihoan, beharrezkoak diren terminalen etiketak beltz nabarmena ("negrita"), gomendagarriak diren terminalen testua kolore argia, eta hautapenezko terminalak kolore ahula izango dute. Azken terminal mota hauen etiketa **Hide Optional Terminals and Full Path** botoia klikeatuz, ez agertzea lortuko da.

• **Context Help** leihoaren behe-ezkerrekaldean kokatutako **Hide Optional Terminals and Full Path** botoia klikeatuz, panelaren loturaren hautapenezko terminalak eta VI-aren bide osoa ikusteko aukera izango dugu.



3-1 Irudia. Context Help leihoa

- Lock Context Help botoia klikeatuz, momento horretan irekita daukagun Context Help leihoa ixtearen aukera izango dugu. Leihoa itxita dugularik, leihoaren edukia aldatzen ez duen objektuaren gainetik kurtsorea mugituko dugu. Leihoa berriro ere irekitzeko, bigarren aldiz botoiari emango genioke. Dena dela, aukera honetara Help menuaren bitartez ere sartu zaitezke.
- Objektu bati dagokion *Labview Help* gaia existitzen baldin bada, Context Help leihoaren barnean kolore urdina duen Clic here for more help lotura azalduko da; More Help botoia ere erabili daiteke. Ondorioz, objektuari buruz informazio

gehiago *Labview Help* baliatuz eskuratu ahal izateko, botoia edo lotura klikeatu beharko da.

Labview laguntza

Laguntza mota hau erabili ahal izateko, **Context Help** leihotik More Help aukera klikeatu eta **Help** >> **Search the Labview Help** hautatuko dugu, edo **Context Help** leihoan aurkitzen den **Clic here for more help** aukeratuz ere helburu bera lortuko genuke. Hala ere, objektuaren gainean eskuineko botoiarekin klikeatuz eta atzipen zuzeneko menutik Help atal aukeratuz ere gauza bera lortuko genuke.

Labview Help paleten, menuen, lanabesen, VI-en eta funtzioen deskribapen zehatzak ematen du. Laguntza mota honek Labview programari buruzko pausuz-pausuzko instrukzioak emateko ahalmena du eta ondorengo errekurtsoetarako zenbait lotura izaten ditu:

- *Labview Documentation Resources* Labview manual bertsio guztien PDF-a dauka.
- Nazional instrumentazioko web-aren errekurtso teknikoak, NI Developer Zone, KnowledgeBase, eta Product Manuals Library besteak beste.

NI adibideen topaketa

New elkarrizketa-kutxak VI-ak sortzen hasteko Labview VI-en plantila ugari dauzka. Hala ere, VI-en plantila hau Labview-k dituen hainbat adibideetatik azpimultzo txiki bat soilik osatzen du. VI adibideetatik edozein aldaketa gauzatu daiteke edo adibide batetik momentu horretan sortzen ari zaren VI-a eraikitzeko kopiatu eta itsatsi agindua ere erabiltzea gauza ohikoa da.

Gainera, Labview programaren bitartez bidali daitezkeen VI-en adibideez aparte, NI Developer Zone atalean aurkitzen den ni.com/zone bitartez, beste zenbait adibide ere lor genitzake. NI Example Finder-a instalatuta dauden adibide guztien sarrera da eta adibideak NI Developer Zone-an bila daitezke.

NI Example Finder laguntza mota gure esku izateko, bi aukera ezberdin ditugu: **Help** >> **Find Examples** hautatu edo **Getting Started** elkarrizketa-kutxatik **Find Examples** atala bilatu.

rowse	Search	Submit	Double-click an example to open it.	Information
			Analyzing and Processing Signals	
Browse according to: Task		g to:	Building User Interfaces	
			Communicating with External Applications	
_	O Directory Structure		Distributing and Documenting Applications	
O Dire			🧰 Favorites	
			🧾 🔚 Fundamentals	
\sim		1111111111111111	🔚 Hardware Input and Output	
			e 📴 Industry Applications	
0			TY Solution Streent Street Str	
-		Networking		
(Learning Article	Articles	New Examples for LabVIEW 8.0	
Conten		Ciano Optimizing Applications		
	Discussion Besou	Resources	Printing and Publishing Data	
~			Programmatically Controlling VIs	
0	Code (8)	Se User	Colkits and Modules	
~	onuning	uroups	Colkits and Modules Not Installed	
Student Cornor				Requirements
	Visit LabVIE	W Zone		
] Inclu ni.co	de ni.com e× m query time	amples		
dware				
	Find bardia	210		

2-2 Irudia. NI adibideen topaketa

B. <u>Apurtutako VI-en zuzenketa:</u>

VI-a exekutatzen ez baldin bada, apurtutako edo exekutatu gabeko VI-a izango da. Kasu hau daukagunean, sortutako edo editatutako VI-a erroreak izango ditu eta hau honela izanik, **Run** botoia apurtuta agertuko da.

Bloke-diagramako konekzio guztiak bukatu eta gero botoia oraindik apurtuta azaltzen bada, VI-a apurtuta dagoela esango da eta ezin izango da exekutatu.

Apurtutako VI-en zergaitien bilaketa

Oharrak ez dute balio apurtutako VI-ak zuzentzeko, VI-aren arazo larrien galarazketan laguntzeko diseinatzen dira. Honez aparte, erroreak ere VI-ak apurtu ditzakete, horrexegatik, VI-a exekutatu baino lehenago, erroreak konpondu beharko dira.

VI-aren apurketaren zergaitia topatzeko, **Run** apurketa botoia edo **View** >> **Error List** klikeatu beharko da. **Error List** aukerak dauzkagun erroreen ezagupena izateko balio du. **Items with errors** atalak, aldiz, memorian aurkitzen diren gai guztien izenak adierazten digu, erroreak dituzten VI eta proiektuaren liburutegiak besteak beste. Bi edo gai gehiago izen berbera baldin badute, atal honek gai bakoitzaren aplikazioaren adibidea azaltzen du. **Errors and warnings** atalak, **Items with errors** atalean aukeratutako errore eta oharren zerrendaren berri ematen digu. **Details** atalak erroreen deskribapenaz osatuta dago eta zenbait kasutan, errore hauek nola zuzendu. Erroreen deskribapen zehatza eta helburu hau lortzeko pausuz pausuzko instrukzioak ematen duen *Labview Help*-etik gai bat erakusteko, **Help** botoia klikeatu beharko da.

Erroreak dituen bloke-diagrama edo aurreko paneleko eremua nabarmentzeko, errorearen desbribapenaren gainean bi bider klikeatu edo **Show Error** botoia zapaltzeko aukera egongo da.

🔋 Error list					
Items with errors					
No VIs with errors					^
				_	•
0 errors and warnings				5how Warnings	
Details					•
Choose a VI from the list of names to	ee its error:	s. Click any erro	or in the list	t to display mor	e 🔺
details about it here. Double-click any Error button, to highlight the object ca	front panel (using the er	or block diagram ror.	n error, or	click the Show	
					×
Clo	se	Show Erro	or	Help	

2-3 Irudia. Error List elkarrizketa-kutxaren adibidea
Apurtutako VI-en zergaiti arruntak

Ondorengo zerrendak VI-a editatzen zauden bitartean honen apurketaren arrazoi arruntak adierazten dizkigu:

• Datu moten egokitasun eta konekzio falta dela eta, bloke-diagramako konekzioak egiteko erabiltzen diren kableak apurtuta agertuko dira.

Labview Help-ean aurkitzen den Correcting Broken Wires gaiaren bidez, apurtutako kableak konpontzeko informazio gehiago lortzeko aukera egongo da.

Beharrezkoa den bloke-diagramaren terminala konektatu gabe badago.

Labview Help-ean lortu daitekeen Using to Link Block Diagram Objects gaia erabiliz, VI-aren sarrera eta irteerak modu egokian konektatu ahal izango dira.

• SubVI-a apurtuta edo paneleko lotura editatu ondoren bloke-diagraman VI-aren ikurra kokatzen denean.

SubVI-ari buruz informazio gehiago nahi baldin bada, *Labview Help*-etik *Creating SubVIs* gaia erabiliko da.

C.Garbiketa egiten duten teknika ezberdinak:

VI-a apurtuta ez badago, baina espero ez diren datuak lortzen baldin badira, segituan aipatzen diren zenbait tekniken bidez, VI-arekin edo bloke-diagramako datuen fluxuarekin arazoak ezagutu eta konpontzea lortuko da:

- VI eta funtzioen behekaldean error in eta error out parametroak lotu beharko dira. Parametro hauek bloke-diagraman aurkitzen diren nodo bakoitzaren erroreak detektatu eta errorea non dagoen adierazten dute eta eraikitako VI-etan erabili ahal izango dira.
- Funtzionamendu lanabesaren bitartez kablea hiru bider klikeatuz, bide osoa nabarmendu eta kable guztiak terminal egokian konektatu direla bermatzen da.
- Bloke-diagrama osatzen duten funtzio eta subVI bakoitzaren balioak egiaztatzeko, **Context Help** leihoa erabili beharko da.
- Exekuzioaren nabarmenketa eginez, bloke-diagramatik zehar aurkitzen diren datuen mugimendua ikusteko aukera egongo da.
- Bloke-diagramako VI-aren akzio bakoitza, VI-aren pausu bakar baten bidez ikusiko da.
- Datuen bitarteko balioa eta VI eta funtzioen irteerako erroreak egiaztatzeko, probe lanabesa erabiliko da.
- Exekuzioa gelditzeko apuketa puntuak erabiliko dira.

- Kontroladore eta adierazleen balioak editatzeko exekuzioa eten egingo da, exekutatzen zenbat denbora igarotzen den edo subVI-aren exekuzioaren hasierara itzultzeko asmoz.
- Erabilitako funtzio edo subVI baten datuak definitu gabe dauden zehaztu beharko da. Kasu hau askotan zenbakiekin gertatzen da. Adibidez, VI-aren puntu bateko operazio batean zenbaki bat zerorekin zatitzean emaitza Inf (infinitua) bezala ikusten da, aurreko funtzio edo subVI-ak zenbakiak itxaroten zutela kontsideratuz.
- VI-a espero dena baino astiroago exekutatzen baldin bada, subVI nabarmenen bidez exekuzioa itzaltzen duzula baieztatu beharko da. Momentu zehatz batean erabiltzen ez diren aurreko paneleko eta bloke-diagramako subVI-en leihoak itxi behar dira, beharrezkoak ez ditugun leihoak irekita egotea exekuzioaren abiaduran eragiten baitu.
- Kontroladore eta adierazleen errrepresentazioa egiaztatu beharko da gainezka jasotzen ari zaren ikusteko, zenbaki flotantea zenbaki oso batean bihurtua izan daitekeelako edo zenbaki oso bat txikia den zenbaki oso batean. Adibidez, 16 biteko zenbaki oso bat 8 biteko zenbaki osoa onartzen duen funtzio batekin soilik konektatu daiteke. Honek 16 biteko zenbaki osoa 8 biteko zenbaki oso batean soilik bihurtzen duen funtzioa eragiten du, datuen galera ekartzen duelarik.
- Edozein For egitura zero iterakzio exekutatzzen dituen zehaztu eta hutsik dagoen matrizea produzitu beharko da.
- Desplazamentu-erregistroa modu egoki batean hasieratu duzula egiaztatu beharko da, zure asmoa exekuzio batetik beste baten egituraren bidez datuak gordetzea izan ezean.
- Iturri baten eta helburu bezala dugun puntuaren cluster elementuen ordena egiaztatu beharko da. LABVIEW datu mota eta cluster-en tamainari dagokionez, denboran desegokitasuna editatzen du, hala ere, ez du datu mota berbera duten elementuen desegokitasuna detektatzen.
- Nodoen exekuzio ordena egiaztatu beharko da.
- VI-a ezkutatuta aurkitzen diren subVI-ak ez dituela egiaztatu beharko da. SubVI bat beste nodo baten gainean kokatuz edo egituraren tamaina txikituz eta subVIa ikusi gabe utzita ere, subVI-a ezkutatuta aurki daiteke.

Exekuzioaren nabarmenketa

Highlight Execution botoia bloke-diagramaren goiko aldean aurkitzen da. Botoi honek bloke-diagramatik zehar datuen mugimendua erakusten du, nodo batetik bestera mugitzeko kableetan zehar desplazatzen diren burbuilak errepresentatzen direlarik. Exekuzio nabarmenketa honek, beraz, indarkeri bakar batekin multzokatuz VI-aren nodo batetik besterako datuen balioak ezagutzeko balioko luke.

Oharra: Exekuzio nabarmenketak VI exekuzioaren abiadura asko murrizten du.

Single-Stepping

Honen bidez exekuzioan zehar bloke-diagraman aurkitzen den VI-aren akzio bakoitza ikusteko aukera izango dugu. Botoi hauek (step into, step over eta step out) pausu bakarreko moduan aurkitzen den VI edo subVI baten exekuzioan eragiten du.

Bloke-diagramako tresna-barra-ean dauden Step Over edo Step Into botoiak klikeatuz, pausu bakarreko moduan sartuko gara. Kurtsorea Step Over, Step Into edo Step Out botoien gainean kokatean, botoi hauetariko bat zapaltzen bada, hurrengo pausua deskribatzen duen zinta baten muturra ikusiko dugu. Pausu bakarreko modua subVI edo normalean exekutatuz lortu daiteke.

Exekuzio nabarmenketa erabiltzen ari den VI-ak pausu bakarreko modua baldin badu, subVI-en ikurren artean momentu hartan exekutatzen ari dela azaltzen da.

Probe lanabesak

VI-a exekutatzen ari den momentuan kableatik zehar aurkitzen diren bitarteko balioak egiaztatzen ditu.

Lanabes mota hau eragiketa ugariz osaturiko bloke-diagrama korapilatsua dugunean erabiliko da, eragiketa hauen emaitza datu okerra eman dezaketelarik. Exekuzio nabarmenketa, single-stepping eta apurketa puntuak dituen **probe** lanabesa, datu oker hauek non dauden zehazteko erabiliko da. Datua dagoeneko prest baldin badago, single stepping edo apurketa puntu batekin gelditzen bada, **probe** berehala gaur eguneratuko da. Single-stepping edo apurketa puntu baten bidez nodo batean exekuzioa gelditzen denean, justu momentu horretan exekutatzen ari den kablea **probe** dezakezu, kabletik zehar hedatzen ari den dtaua ikusi ahal izateko.

Probe motak

Probe orokorra, datua ikusteko **kontrol** paletan aurkitzen den adierazlea erabiliz, probe berri bat sortuz, probe hornituak erabiliz edo ohiturazko probe erabiliz, VI-a exekutatzen ari den bitartean kabletik zehar dauden bitarteko balioak egiaztatu daitezke.

Orokorra:

Kabletik zehar pasatzen ari den datua ikusteko, probe orokorra erabiliko da. Arratoiaren eskuineko botoiarekin kablea klikeatuz eta atzipen zuzeneko menutik **Custom Probe** >> **Generic Probe** hautatuz, probe orokorra lortuko genuke.

Probe orokorrak datuak erakusten ditu; baina datuari erantzuteko ezin izango da probe orokorra konfiguratu.

Labview programak eskuineko botoiarekin kablea klikeatuz eta **Probe** aukeratuz, probe orokorra erakusten du, hala ere, jada datu motaren probe zehaztua baldin badago ez litzateke hau gertatuko.

Probe bat VI baten antzera garbitu daiteke. Dena dela, ezin izango du probe bere bloke-diagrama ezta bere subVI-en bloke-diagrama ere ez. Probes garbitzeko, beraz, probe orokorrak erabiliko dira.

Datua ikusteko adierazlearen erabilpena:

Zenbait kasutan adierazle bat erabili daiteke kableetatik zehar pasatzen ari diren datuak ikusi ahal izateko. Adibidez, zenbakizko datu bat ikusten baldin bada, probaren barnean chart grafikoa datua ikusteko kokatu daiteke. Arratoiaren eskuineko botoiarekin kablea klikeatu, atzipen zuzeneko menutik Custom Probe >> Controls hautatu eta erabili nahi den adierazlea aukeratu beharko da. Hala ere, Controls paletatik Select a Control ikurra ere klikeatu daiteke.

Aukeratutako adierazlearen datu mota eskuineko botoiarekin klikeatutako kablearen datu motarekin ezberdintasuna baldin badu, Labview programak ezin izango du kable honen gainean adierazle hau ipini.

Horniketa:

Probe hornituak kableetatik zehar pasatzen ari diren datuei buruzko informazio osoa erakusten duten VI-ak dira. Adibidez, VI Refnum Probe VIaren izena, VI-aren bidea eta erreferentziaren balio hexadezimalari buruzko informazioa itzultzen du. Kabletik zehar hedatzen ari diren datuetatik erantzun ahal izateko, probe hornituak erabiltzen dira. Adibidez, error cluster baten errorearen egoera (status), kodea (code), iturria (source) eta deskribapena jasotzeko Error probe erabiliko da eta errore edo ohar bat izaten baldin badugu, baldintzazko apurketa puntua jarri nahi izango du.

Probe hornituak atzipen zuzeneko menuaren goiko aldean aurkitzen den **Custom Probe**-n azaltzen da. Probe hornituak lortu ahal izateko, arratoiaren eskuineko botoiarekin kablea klikeatu eta atzipen zuzeneko menutik **Custom Probe** hautatuko dugu. Atzipen zuzeneko menuan kablearen datu motarekin konektatu daitezkeen probes bakarrik agertuko dira.

Probe hornituen erabileraren adibideetako bat labview/examples/general/probes.llb atalean aurki dezakegu.

Ohiturazkoak:

Existitutako probe batean oinarritutako probe bat edo probe berri bat sortzeko, Custom Probe Wizard erabiliko da. Arratoiaren eskuineko botoiarekin kablea klikeatuz eta atzipen zuzeneko menutik **Custom Probe** >> **New** hautatuz, Custom Probe Wizard erakutsiko da. Labview programak kableetatik zehar hedatzen ari diren datuak nola probes –ari buruz informazio gehiago izan nahi badu, probe bat sortu beharko da. Probe berri bat sortzen denean, probe-aren datu mota eskuineko botoiarekin klikeatutako kablearen datu motarekin koinziditu beharko du. Gainera, bakoitzak sortutako probe-a editatu nahi baldin bada, gorde izan genuen direktoriotik ireki behar izango dugu.

Atzipen zuzeneko menuaren **Custom Probe** aukeratik probe-a hautatu ondoren, **Select a Control** paletaren botoia erabiliz itsasoratu edo Custom Probe Wizard erabiliz probe berri bat sortuko dugu. **Probe** hau datu motari dagokion probe bihurtzen da eta Labview probe hori kargatu egiten du arratoiaren eskuineko botoiarekin kablea klikeatu eta atzipen zuzeneko menutik Probe hautatzerakoan. Hortaz, Labview kablearen datu mota bera duen eta berarekin konektatu daitekeen probe-ak ditugunean soilik kargatu ahal izango dira.

<u>**Oharra:</u>** Datu mota zehatza duen ohiturazko probe bat nahi badugu, probe-a user.lib/_probes/default direktorioan gordeko dugu. Probes ezin izango dira vi.lib/_probes direktorioan gorde Labview-k artxiboak gainidazten baititu gaur eguneratu edo berriz ere instalatzean.</u>

Apurketa puntuak

Apurketa puntua lanabesa erabili, apurketa puntua bloke-diagramako VI-an, nodoan, edo kablean kokatzeko eta exekuzioa leku berean gelditzeko.

Apurketa puntua kablean kokatzerakoan, datua kabletik zehar pasatu ostean exekuzioa geldituko da. Bloke-diagramaren barnean apurketa puntua kokatzerakoan, aldiz, bloke-diagramako nodo guztiak exekutatu eta gero exekuzioa geldituko da.

VI-a apurketa puntuaren bitartez gelditzen baldin bada, Labview-k bloke-diagrama aurreko panelera eramango du eta apurketa puntua daukan nodo edo kablea nabarmenduko du. Kurtsorea existitzen den apurketa puntuaren gainetik mugitzean, apurketa puntua lanabesa kurtsorearen eremu beltza kolore zuriaz agertuko da.

Exekuzio bitartean apurketa puntura heltzen garenean, VI-a gelditzen da eta **Pause** botoia kolore gorriz azaltzen da. Ondorengo neurriak hartu daitezke:

- Exekutatzen ari den bitartean, single-stepping botoiak erabiliz, pausu bakarrekoa izatea lortuko dugu.
- Probe kableak bitarteko balioak egiaztatzeko.
- Aurreko paneleko kontroladoreen balioen aldaketa.
- **Pause** botoia klikeatu, hurrengo apurketa punturarte exekutatzen jarraitu dezan edo VI-a exekutatzen bukatu arte.

Exekuzioaren etendura

SubVI baten exekuzioaren etendura zenbait arrazoiengatik izan dezakegu: kontroladore eta adierazleen balioak editatzeko, subVI-a zenbat aldiz exekutatzen den kontrolatzeko edo subVI-aren exekuzioaren hasierara itzultzeko. Exekuzioa eten egin dela, subVI bati egin dakiokeen dei guztiak edo subVI bati dei zehatz bat eten dakioke.

SubVI baten dei guztiak etentzeko, subVI-a ireki eta **Operate** >> **Suspend when Called** hautatu. SubVI-a beste VI bat deitzerakoan, berehala eten egingo da. Singlestepping daukagula menuan aurkitzen den gai hau aukeratzen baldin bada, subVI ez da segituan etenduko; subVI-a dei egiten diogunean eten egingo da.

SubVI zehatz baten deia etentzeko, bloke-diagramako subVI-aren nodoa eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **SubVI Node Setup** aukeratuko dugu. **Suspend when called** egiaztapen-kutxan egiaztapen-marka bat kokatu beharko da, honela, exekuzioa eten egingo da baina momentu zehatz horretararte soilik.

File >> **VI Hierarchy** hautatuz, **VI Hierarchy** leihoa erakustea lortuko da; honek VI-a eten edo geldi dagoen adieraziko du. Beheko aldean azaltzen den ikurra VI-a eregularki exekutatzen ari dela esan nahi du edo single-stepping.

- Pause ikurrak, eten edo gelditutako VI irudiaren berri ematen digu.
- Kolore berdeko edo betegabeko kolore zuri eta beltzeko ikurrak, deitzerakoan gelditu egiten den VI-a adierazten du. Pause gorridun edo kolore zuri eta beltzeko solidozko ikurrak, momentu horretan geldirik dagoen VI-a adierazten du. Harridura ikurrak, aldiz, subVI-a eten dagoela esaten digu.
- VI bat momentu berean eten eta geldirik egon daiteke.

SubVI-aren momentu hartako unearen zehazpena

SubVI bat gelditzerakoan, tresna-barra-en **Call list** menu hedatua goiko VI behean aurkitzen den subVI-aren deien katea zerrendatzen du. Zerrenda hau ez da **Browse** >> **This VI''s Callers** hautatzean lortutako zerrenda bera izango, honek deitutako VI guztien zerrenda ematen du, exekutatzen ari direla edo ez kontutan hartu gabe. Bloke-diagramak une bat baino gehiago baldin badu, **Call list** menua momentu hartako unea zehazteko erabiliko da. **Call list** menutik VI bat hautatzean, dagokion bloke-diagrama ireki, eta Labview-k momentu hartako unea nabarmentzen du.

C. <u>Definitu gabeko edo espero ez diren datuak:</u>

Definitu gabeko datuek, NaN (zenbaki bat ez) edo Inf (infinitua), eragiketa guztien baliotasuna anulatzen dute. Puntu flotanteko eragiketak, kalkuluen emaitzak zentzugabe edo akastunak adierazten duten ondorengo bi balio sinbolikoak itzultzen dute:

- NaN (zenbaki bat ez) sortutako eragiketak anulatzen dituen puntu flotanteko balioa errepresentatzen du, zenbaki negatibo baten erro karratua hartzea besteak beste.
- Inf (infinitua) sortutako eragiketak anulatzen ez dituen puntu flotanteko balioa errepresentatzen du, zenbaki bat zerorekin zatitzean besteak beste.

Labview-k balio osoen gainezka edo gainezkaren kontrako baldintzak ez ditu egiaztatzen. Puntu flotanteko zenbakien gainezka eta gainezkaren kontrakoa dena, IEEE 754-rekin ados dago, *Standard for Binary Floating-Point Arithmetic*.

Puntu flotanteko eragiketekk fidagarriak diren NaN eta Inf hedatzen ditu. NaN edo Inf explizituki edo inplizituki balio oso edo boolear batean bihurtzen duzunean, baliaok zentzugabeak bihurtzen dira. Adibidez, 1 balioa zerorekin zatitzean, emaitza bezala Inf izango dugu. Inf 16 biteko zenbaki oso batean bihurtzean 32.767 balioa sortzen du, balio normal bat dela dirudi.

Datu bat datu mota osoan bihurtu baino lehen, puntu flotanteko bitarteko balioak egiaztatzeko Probe lanabesa erabiliko da. **Comparison** funtzioa, Not a

Number/Path/Refnum?, guretzako baliagarria ez den balio batekin konektatuz, NaN konprobatzea lortuko dugu.

VI batek definitua ez dauden datuak sortzen dituen edo ez zehazteko, ez da NaN, Inf edo hutsik dauden matrizeak bezalako balio berezietan oinarritzen. Honez aparte, VIaren txosten bat egitean definitu gabeko datuak sortzearen une bat duen errorea aurkitzen baldin bada, VI-a definitutako datuak produzitzen dituela baieztatuko da.

Adibidez, For egitura bat autoindexatzeko matrize bat erabiltzen duen VI bat sortzen baduzu, matrizearen sarrera hutsik dagoenean VI-a zer egin nahi duen zehaztu beharko da. Hala ere, irteerako errore kode bat sortzen badu, definitutako datuak egiturak sortzen duen balioarekin ordezkatu beharko da, edo matrizea hutsik dagoenean For egitura exekutatzen ez duen Case egitura bat erabili beharko da.

E.Erroreen egiaztapena eta gestioa:

Erroreak egiaztatzeko mekanismorik gabe, VI-ak ondo lan egiten ez duela soilik jakin dezakegu. Erroreen egiaztapenak erroreak non eta zergatik sortzen diren esaten digu.

Erroreen gestio automatikoa

Errore bakoitzak zenbakizko kode bat dauka eta bakoitzari errore mezu bat dagokio.

Labview-k zuzenean edozein errore maneiatzen du etendura exekuzioaren bidez VI-a exekutatzen denean, errorea gertatzen den subVI edo funtzioan nabarmendu egiten du eta errorearen elkarrizketa-kutxa bat erakusten du.

Erroreen gestio automatikoa oraingo VI-agatik desaktibatzeko, File >> VI Properties eta Category menu hedatuan aurkitzen den Execution hautatu beharko da. Errore gestio automatikoa berria den edozeingatik desaktibatzeko, VI zuria sortu beharko da, horretarako, Tools >> Options eta Category zerrendatik Block Diagram aukera hartuko da. Errore gestio automatikoa VI baten barnean aurkitzen den subVI edo funtzio batez desaktibatzeko, bere error out parametroa beste subVI edo funtzio baten error in parametroarekin edo adierazle baten error out-ekin konektatuko dugu.

Erroreen eskuzko gestioa

Aurretik aipatutakoaz aparte, erroreen gestioa egiteko beste metodo bat ere erabili daiteke. Adibidez, bloke-diagramako I/O VI bat denboratik kanpo baldin badago, errore elkarrizketa-kutxa gekditu eta erakusteko aplikazio osoa ez genuke nahi izango.

Erroreak maneiatu ahal izateko, **Dialog & User Interface** paletan aurkitzen diren Labview-ren erroren gestioa egiteko VI-ak eta funtzioak eta VI eta funtzio guztien **error in** eta **error out** parametroak erabiliko dira. Adibidez, Labview programak errore bat aurkitzen badu, mota ezberdinetako elkarrizketa-kutxetan errorearen mezua erakutsi daiteke. Erroreen gestioa garbiketa lanabesarkein batera erabili daiteke erroreak aurkitu eta maneiatu ahal izateko.

Bi modu hauetariko bat erabiliz, zenbakizko erroreen kodeekin edo error cluster-arekin, VI-ak eta funtzioak erroreak itzultzen dute. Normalean, funtzioak zenbakizko erroreen kodeak eta VI-ak error cluster-a erabiltzen dute, errore sarrera eta errore irteerekin.

Edozein motatako sarrera edo irteera sortzerakoan (I/O), erroreak gertatu daitezkeen posibilitatea dugu. I/O ia funtzio guztiek errorearen informazioa itzuli egiten dute. VIetan erroreen egiaztapena dauka, bereziki I/O eragiketak (artxiboa, instrumentazioa, datuen eskuraketa eta komunikazioa) eta erroreak modu egokian maneiatzeko mekanismo bat ematen du.

Errroreak maneiatzeko, Labview-k VI-en, funtzioen eta parametroen erroreen gestioa erabiltzen du. Adibidez, Labview-k errore bat aurkitzen badu, elkarrizketa-kutxa batean errore mezua erakutsi daiteke. Edo programatikoki errorea zehaztu ostean, subVI edo funtzioaren error out irteera Clear Errors VI error in sarrerarekin konektatuz errorea borratu daiteke. Erroreen gestioa garbiketa lanabesarekin erabiliz, erroreak aurkitu eta maneiatu ditzakegu. National Instrument-ak erroreen gestioa erabiltzea gomendatzen du.

Error clusters

SubVI-etan errore sarrerak eta irteerak sortzeko, error cluster kontroladore eta adierazleak erabiltzen dira.

Error in eta error out cluster-ak hiru konponente ezberdinez osatuta daude:

- **Status** (egoera) errorea baldin badago emaitza bezala TRUE (egia) ematen duen balio boolearra da.
- **Code** (kodea) zenbakien bidez errorea identifikatzen duen 32 biteko eta zeinua duen zenbaki osoari dagokio.
- Source (iturria) errorea noiz gertatzen den identifikatzen duen string-a da.

Labview programan erroreen gestioak datuen fluxuaren modelua jarraitzen du.

VI-a exekutatzean, Labview-k nodo exekuzio bakoitzaren erroreak aztertzen ditu; errorerik topatzen ez baldin badu, nodoa ohikoa den bezala exekutatuko da, hala ere, errorea detektatzen baldin badu, nodoak errorea hurrengo nodora pasatuko dio kodeari dagokion zati hori exekutatu gabe geratuz. Exekuzioaren amaneran, Labview programak erroreen txostena garatuko du.

Errorearen azalpena

Errorea gertatzen denean, cluster-aren muturraren barnean eskuineko botoiarekin klikeatuz eta atzipen zuzeneko menutik **Explain Error** hautatuz, **Explain Error** elkarrizketa-kutxa irekiko da; kutxa honek erroreari buruzko informazioa ematen du. Atzipen zuzeneko menuak **Explain Warning** izeneko aukera bat dauka eta VI-a oharrak dituen adierazteko balio du, ez erroreak.

Explain Error elkarrizketa-kutxa honetara **Help** >> **Explain Error** menutik ere sartu gaitezke.

Bi modu hauetariko bat erabiliz, zenbakizko erroreen kodeekin edo error clusterarekin, VI-ak eta funtzioak erroreak itzultzen dute. Normalean, funtzioak zenbakizko erroreen kodeak eta VI-ak error cluster-a erabiltzen dute, errore sarrera eta errore irteerekin.

Azterketa askea: Froga

1. Erroreen gestio automatikoa nola desaktibatzen da?

a. **Operate >> Disable Error Handling** hautatuz.

- b. Aktibatuta dagoen exekuzioa nabarmendu.
- c. SubVI-aren error out cluster terminala beste subVI baten error in cluster terminalarekin konektatu.
- d. **Error List** elkarrizketa-kutxan aurkitzen den **Show Warnings** egiaztapen kutxan egiaztapen marka bat kokatu.
- 2. Lau aukera hauetatik zein da error cluster-aren edukia edo mamia? (Erantzun bat baino gehiago egon daiteke)
 - a. Egoera: Boolearra.
 - b.Errorea: String.
 - c. Kodea: 32-bit oso.
 - d. Iturria: String.

Azterketa askea: Frogaren erantzuna

1. Erroreen gestio automatikoa nola desaktibatzen da?

- a. Operate >> Disable Error Handling hautatuz.
- b. Aktibatuta dagoen exekuzioa nabarmendu.

c. SubVI-aren error out cluster terminala beste subVI baten error in cluster terminalarekin konektatu.

d. Error List elkarrizketa-kutxan aurkitzen den Show Warnings egiaztapen kutxan egiaztapen marka bat kokatu.

2. Lau aukera hauetatik zein da error cluster-aren edukia edo mamia? (Erantzun bat baino gehiago egon daiteke)

a. Egoera: Boolearra.

- b. Errorea: String.
- c. Kodea: 32-bit oso.
- d. Iturria: String.

4.GAIA: VI BATEN INPLEMENTAZIOA

Gai honek Labview programa batean kodea nola inplementatzen den erakusten digu. Gainera, teknika honek erabiltzailearen arteko interfazea dakar, datu motaren aukeraketa, kodearen dokumentazioa, While eta For bezalako zenbait egituren erabilpena, koderako denboran oinarritutako softwarea, datuen erakusketa, eta Case egitura erabiliz kodearekiko erabakiak hartu beharra, hain zuzen ere.

Gai honi dagozkion atalak ondorengoak dira:

- A. Aurreko panelaren leihoaren diseinua.
- B. Labview datu motak.
- C. Dokumentaziorako kodea.
- D. While egiturak.
- E. For egiturak.
- F. VI-aren denbora erlazioak.
- G. Datu iteraktiboen transferentzia.
- H. Datuen marrazketa.
- I. Case egiturak.
- J. Nodoen formula.

A. <u>Aurreko panelaren leihoaren diseinua:</u>

Software-garapen metodoaren diseinuko fasean, arazoaren sarrera eta irteerak identifikatu genituen. Identifikazio honek aurreko panelaren leihoaren diseinua zuzenean eramango du.

Ondorengo metodoak erabiliz, arazoaren sarrerak berreskuratu daitezke:

- Dispositibo baten bidezko eskuraketa, datuen eskuraketa dispositiboak edo multimetroak besteak beste.
- Artxibo batetik zuzenean irakurketa egin.
- Kontroladoreen manipulazioa.

Arazoaren irteerak adierazleen bidez erakutsi edo irteera hauek artxibo batean erregistratu daitezke. Hala ere, seinale sortzailea erabiliz, dispositibo baten irteerako datuak ere lortzeko aukera dago.

Datuen eskuraketa, seinale sortzailea eta artxiboen erregistroari buruzko zenbait kontzeptu, kurtso honetan zehar ere agertuko dira.

Kontroladore eta adierazleen diseinua

Kontroladore eta adierazleen aukeraketa gauzatzen dugunean, burutuko ditugun zenbait eginkizunetarako egokiak direla ziurtatu beharko gara. Adibidez, sinu uhin baten maiztasuna zehaztu nahi dugunean, dial kontroladore mota aukeratzen dugu, edo tenperatura zehaztu nahi dugunean, aldiz, termometro baten portaera duen adierazlea hautatuko dugu.

Etiketak eta azpitituluak

Kontroladore eta adierazleentzako etiketak sortzen direnean, modu argi batean egiten dela ziurtatu beharko gara. Etiketa hauetariko bakoitza erabiltzaileek kontroladore eta adierazle bakoitzaren funtzioa identifikatu dezaten lagunduko die. Gainera, modu argi batean dauden etiketak bloke-diagramako kodearen dokumentazioan laguntzen dute. Kontroladore eta adierazleen etiketak bloke-diagramako terminalen izenei dagokie.

Azpitituluek, aldiz, aurreko panelean aurkitzen diren kontroladoreen deskribapena gauzatzera laguntzen dute; hauek ez dira bloke-diagraman agertzen. Azpitituluak erabiliz, erabiltzailearekin interfazearen dokumentazioa egitera baimentzen dio, bloke-diagrama izen luzeekin oztopatu gabe. Adibidez, estazio meteorologikoan tenperaturaren mailarako muga handiago bat eman beharko da. Tenperatura igoera maila honen gainetik jartzen bada, estazio meteorologikoak berotasunaren oharra adieraziko du. Kasu honetan kontroladoreari, *Upper Temperature Limit (Celsius)* deitu dezakegu. Dena dela, etiketa honek bloke-diagraman beharrezkoa ez den eremua okupatu dezake. Honen ordez, *Upper Temperature Limit (Celsius)* kontroladorerako azpititulua eta etiketa, aldiz, bloke-diagramako deskribapen laburra sortzeko erabiliko da, *Upper Temp* besteak beste.

Kontroladore eta adierazleen aukera ezberdinak

Kontroladoreentzako defektuzko balioak jarri daitezke. Defektuzko balio hauek lortzeko, exekuzio denbora bitartean erabiltzaileak beste balio bat zehazten ez baldin badu, VI-arentzako balio arrazoitua esleitu genezake. Honetarako, hurrengo pausuak jarraitu beharko dira:

- 1. Desiratzen dugun balioa sartu.
- 2. Arratoiaren eskuineko botoiarekin kontroladorea klikeatu eta atzipen zuzeneko menutik **Data Operations >> Make Current Value Default** hautatu.

Upper Temperature Limit (Celsius)	Upper Temperature Limit (Celsius)
-----------------------------------	-----------------------------------

4-1 Irudia. Defektuzko balioen lorpena aurreko panel eta bloke-diagraman

Gainera, kontroladore eta adierazleen gaiak ezkutatu edo erakutsi daitezke. Azpititulua soilik ikustearekin nahikoa bada, etiketa ezkutatzeko arratoiaren eskuineko botoiarekin kontroladorea klikeatu eta **Visible Items >> Label** hautatuko genuke.

Untitled 1 From Edit <u>V</u> iew Pro	n <mark>t Panel *</mark> ject <u>O</u> perate <u>T</u> ools <u>W</u> indo	w <u>H</u> elp	
¢ & ●	13pt Application Font		
Upper T	emperature Limit (Celsius)		
y 33 .	Visible Items Find Terminal Change to Indicator Description and Tip Create Replace Data Operations Advanced Scale Object with Pane	✓ Label Caption Unit Label Radix ✓ Increment/Decrement	
	Representation Data Range Format & Precision		

4-2 Irudia. Aurreko paneleko kontroladore baten etiketaren ezkutaketa

Koloreen erabilpena

Kolore egokiaren erabilpena erabiltzailearen interfazearen itxura eta funtzionalitatea hobetu dezake. Kolore ugarien erabilpenak, aldiz, ez da egokia ikusiko.

Labview-k koloreen aukeraketan lagundu dezaketen koloreen hautatzaileak eskuragarri ditu. Koloreen lanabesa hautatu eta kolore hautatzailea erakusteko, objektua edo lan egiteko erabiltzen ari garen eremua arratoiaren eskuineko botoiarekin klikeatuko dugu. Kolore hautatzailearen goiko aldean gris koloreen eskala aurkitzeaz aparte, objektu gardenak sortzeko erabiltzen den kutxa ageri da. Bigarren espektrua, sakoneretan eta aurreko paneleko objektuetan erabiltzeko kolore apalak ditu. Hirugarren espektrua, aldiz, nabarmenketak gauzatzeko aproposak diren koloreez osatuta dago.

Aurrerantzean aurkitzen diren aholkuak koloreen hautaketarako oso baliagarriak izango dira:

- Labview-k eskaintzen dizkigun koloreak erabil itzazu. Ordenagailuan kolorea ez baldin badago, Labview-k itxura berdintsuena duenagatik ordezkatuko luke. Gainera, aurreko paneleko leihoaren itxura VI-a exekutatzen ari den koloreen sistema batean egokitzeko, koloreen sistema erabili daiteke.
- Kolore grisekin hasiera eman. Gris tonuetatik bat edo bi hautatu eta sakonerarekin kontraste handia daukan kolore nabarmenak aukeratu.
- Kolore nabarmenak neurritasunarekin gehitu. Objektu txikiak objektu handiak baino kolore distiratsuagoak eta kontrastedunak izan beharko dira.
- Kontraste-aldaketak kolore-aldaketak baino maizazo erabil itzazu.
- Koloreetan bat etortzen diren objektuen multzoak gauzatu ordez, egokitasun eta okupatutako lekuaren arabera multzokatu.
- Koloreei buruzko ikasketak gauzatzeko, hardware instrumentuen panelak, mapak eta aldizkariak oso leku aproposak dira.
- Aurreko paneleko kontroladoreak koloreen sistema erabiltzea nahi baldin baduzu, **kontrol** paletan aurkitzen den **System** mailatik objektuak aukeratu.

Lekutasuna eta linealtasuna

Eremu zuria eta linealtasuna taldeak egiteko eta sakabanatzeko teknikarik garrantsitzuenak dira. Zuzen batetik gure begiak antzeman ditzakeen elementuetatik garbitasuna eta koherentzia daude. Gaiak zuzen batean daudenean, gure begiek zuzen hori ezkerretik eskuinera eta goitik behera jarraitzen dute. Hau direkzio komandoen sekuentziarekin erlazionatuta dago. Nahiz eta kultura batzuk iritzia eskuinetik ezkerrera eman, gehienek goitik behera egiten dute.

Aurreko panelaren diseinua egiterakoan, erabiltzaileak VI-a eta kontroladore eta adierazle taldearekin nola interakzionatzen duen kontutan hartu beharko da. Kontroladore ugari beraien artean erlazionaturik badaude, cluster batean sar itzazu.

Testu eta iturriak

Testu txukuna erakusten denean, irakurtzeko eta ulertzeko askoz ere errazagoa izaten da. Labview-k zenbait iturri erabiltzen ditu. Iturri ugari erabiliz, aurreko paneleko leihoa hankaz gora ikusiko litzateke. Horren ordez, iturri berdineko bi edo hiru tamainu ezberdin erabiltzea komenigarriena da, honela, atal batzuk beste batzuetatik ezberdinduko lirateke eta emaitza bezala, testua txukun-txukuna izango genuke.

Erabiltzaileren interfazearen aholku eta lanabesak

Labview programak aurreko paneleko leihoan dituen lanabesen artean, kontroladoreen sistemak, kontroladoreen fitxak, apaingarriak, menuak, eta aurreko paneleko objektuen tamainu aldaketa automatikoa aurkitzen dira.

Kontroladoreen sistemak

Erabiltzaileren interfazeen artean teknikarik arruntena erabiltzailearekin interakzionatzen dagoen momentu aproposa erakusten diguten elkarrizketa-kutxak dira. VI bat elkarrizketa-kutxa baten portaera izateko, pausu hauek jarraitu beharko dira: File >> VI Properties aukeratu, Window Appearance maila hautatu eta Dialog aukera hartu.

Guk sortutako eta **System** paletan aurkitzen den kontroladore eta adierazleen sistema erabiliko da. VI-a zein plataforman exekutatzen duzun arabera kontroladoreen sistema aldatzen denez, plataforma bakoitzean dauden kontroladoreen itxura honen arabera aldatuko da. VI-a plataforma ezberdinetan exekutatzean, kontroladoreen sistemak kolorea eta itxura moldatzen dute plataforma horretako oinarrizko elkarrizketa-kutxaren kontroladoreekin konektatu ahal izateko.

Kontroladoreen sistemak normalean kolore gardenaz aparte, kolore guztiak ez jakintzat uzten dituzte. Aurreko paneleko leihoaren barnean grafiko bat kokatzen badugu, ertz ezkutu batzuekin edo sistemaren koloreekin antzekotasuna duten koloreak hautatuz, elkarren artean konektatu ahal izango dira.

Kontroladoreen fitxak

Instrumentu fisikoek normalean erabiltzaileekin interfazea egokia izaten dute. Bere diseinu nagusia nahiz eta eskatu, kontroladore txiki eta zehatzagoak ditu, kontroladoreen fitxak besteak beste. Kontroladore mota hauek aurreko paneleko kontroladore eta adierazleak eremu txiki batean gainezartzeko erabiltzen dira.

Kontroladorearen fitxa batean beste orri bat gehitzeko, fitxa arratoiaren eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Add Page Before** edo **Add Page After** hautatuko dugu. Etiketen lanabesarekin berriz ere fitxak etiketatu, eta aurreko paneleko objektuak orri egokian kokatu beharko dira. Objektu hauen terminalak bloke-diagraman eskuragarri egongo dira, beste edozein aurreko panelaren objektuen terminalak bezalaxe (apaingarriak izan ezik).

Kontroladore hauetan zenbakien bidez izendatutako kontrol terminalak Case egituraren hautatzailearekin konektatu daitezke, bloke-diagrama txukunagoa lortzeko helburuz. Metodo honekin, fitxa kontroladoreen orri bakoitza azpidiagramarekin erlazionatu daiteke edo Case egiturarekin. Beraz, kontroladore mota honen orri bakoitzaren kontroladore eta adierazle terminalak Case egiturako azpidiagraman kokatu beharko dira.

Apaingarriak

Decorations paletan aurkitzen diren apaingarriak, aurreko paneleko objektuak kutxa, zuzen edo geziekin taldekatu edo sakabanatzeko erabiltzen dira. Objektu hauek apaingarri bezala soilik balio dute eta ez dute daturik erakusten.

Menuak

Aurreko paneleko funtzionalitatea era ordenatu batean eta eremu txiki batean aurkezteko menuak erabiltzen dira. Leku kantitate txikia erabiliz, aurreko panelean kontroladore eta adierazle kritikoentzako, hasiberri guztien gaientzako, produkziorako beharrezkoak diren gaientzako eta menuetan modu egoki batean sartzen ez direnentzako lekua utzi beharko da. Hala ere, menuen gaientzako teklatuaren bide laburra sortu daiteke.

Aurreko paneleko objektuentzako exekuzio-denbora atzipen zuzeneko menua sortzeko, aurreko paneleko objektua arratoiaren eskuineko botoiarekin klikeatu eta Advances >> Run-Time Shortcut Menu >> Edit hautatuko dugu. VI-rako exekuzio-denbora menua sortzeko, aldiz, Edit >> Run-Time Menu aukera hartu beharko da.

Aurreko paneleko objektuen tamainu aldaketa automatikoa

Leiho baten tamainu minimoa finkatzeko, VI Properties >> Window Size aukerak erabili beharko dira, pantailak neurri bera mantendu beharko du aldaketak egiten diren bitartean, eta aurreko paneleko objektuak zehaztu beharko dira tamainu aldaketa bi modu ezberdinez egiteko. VI-a diseinatzerakoan, aurreko paneleko leihoa pantailako erreboluzio ezberdinekin ordenagailuetan erakutsi daitekeen kontutan hartu beharko da. Aurreko paneleko leihoa pantailako erreboluzioekin era proportzional batean mantentzeko, File >> VI Properties hautatu, Category menu hedatuan aurkitzen den Window Size aukera hartu eta egiaztapen-marka Maintain Proportions of Window for Different Monitor Resolutions egiaztapen-kutxan kokatu beharko da.

B. Labview datu motak:

Datuentzako datu mota ugari existitzen dira. *Navigating Labview* izeneko 2. gaian zehar, zenbakizko, boolear eta string datu motei buruzko ikasketa gauzatu genuen. Bestelako datu motak, zenbakiz izendatutako datu mota, datu dinamikoa eta horrelakoak izaten dira. Zenbakizko datu motan ere, datu mota desberdinak aurki daitezke, zenbaki osoak edo zatiki moduan adierazitako zenbakiak besteak beste.

Terminalak

Bloke-diagramako terminalek erabiltzaileari errepresentatutako datu motari buruzko informazioaren berri ematen diote.



4-3 Irudia. Datu motaren terminalaren adibidea

Zenbakizko datu mota

Zenbakizko datu motak tipo ezberdinetako zenbakiak adierazten ditu. Zenbaki baten errepresentazio mota aldatzeko, arratoiaren eskuineko botoiarekin kontroladore, adierazle edo konstantean klikeatu eta **Representation** hautatu beharko genuke.

Errepresentazio ezberdineko zenbakizko sarrera bi edo gehiago funtzio batekin konektatzerakoan, funtzioak irteera formatu handiago edo luzeago batean itzultzen du. Exekuzioa baino lehen, funtzioak errepresentazio zabalen aurrean errepresentazio txikiak behartzen dute eta bihurketa gauzatzen denean Labview-k gogorkeria kokatzen du ematen den terminaleko puntuan.

🐏 Untitled 1 Block Diagram *		_ 2 🛛
<u> Eile E</u> dit <u>V</u> iew <u>P</u> roject <u>O</u> perate <u>T</u> ools <u>W</u> indow <u>H</u> elp		
수 🐼 🔘 💵 😵 🕵 🏎 🗗 🔐 13pt Application For		2 5
		*
Numeric		
Visible Items		
Find Control		
Hide Control		
Change to Indicator		
Change to Constant		
	-	
Numeric Palette		
Data Operations		=
Advanced ►		
Representation 🕨	Double Precision	
View As Icon		
Properties		
	U64 U32 U16 U8 1 = 100 1 = 100 1 = 100 7 = 100	
		→ <u></u>

4-4 Irudia. Zenbakizko errepresentazioa

Zenbakizko datu motak erresentazioan ondorengo azpimailak izan dezake: puntu flotanteko zenbakiak, zeinua duten zenbaki osoak, zeinugabeko zenbaki osoak, eta zenbaki konplexuak.

Puntu flotanteko zenbakiak

Puntu flotanteko zenbakiak zatiki moduan adierazitako zenbakiak dira. Labview programan, zenbaki mota hauek kolore laranjez adierazten dira.

- **Single-precision (SGL)** --- Prezisio sinplea, puntu flotanteko zenbakiak 32 bitetako IEEE prezisio sinpleko formatua dute. Prezisio sinplea erabiliz, puntu flotanteko zenbakiek memoria gorde eta zenbakien tarteen gainezka eragozten dute.
- Double-precision (DBL) --- Prezisio bikoitza, puntu flotanteko zenbakiak 64 bitetako IEEE prezisio bikoitzeko formatua dute. Zenbakizko objektuentzako defektuz erabiltzen den formatua prezisio bikoitzekoa da. Egoera gehienentzako, prezisio bikoitzekoa erabiltzen da, puntu flotanteko zenbakiak.
- **Extended-precision (EXT)** --- Memorian, mota honetako zenbakien tamainua eta prezisioa plataformaren arabera aldatzen dira. Windows-ean, hauek 80 bitetako IEEE prezisio hedatuko formatua dute.

Zenbaki osoak

Zenbaki osoak zenbaki guztiak adierazten dute. Zeinua duten zenbaki osoak positibo edo negatiboak izan daitezke; zeinugabeko zenbaki oso dtau mota, aldiz, zenbakia ezagutzen duzunean, beti positiboa izango da. Labview programan, zenbaki osoak kolore urdinarekin errepresentatzen dira.

Labview-k puntu flotanteko zenbakiak zenbaki osoetan bihurtzen ditu, VI-a zenbaki bikoititik hurbilago aurkitzen da. Adibidez, 2.5 daukagunean 2 izanda eta 3.5 izatean 4.

- **Byte** (**I8**) --- Byte zenbaki osoak 8 bitetako biltegiratzea dute.
- Word (I16) --- Word zenbaki osoak 16 bitetako biltegiratzea dute.
- Long (I32) --- Long zenbaki osoak 32 bitetako biltegiratzea dute.
- Quad (I64) --- Quad zenbaki osoak 64 bitetako biltegiratzea dute.

Zenbaki konplexuak

Zenbaki konplexuak elkarren artean erlazionatutako bi balioekin adierazten dira memorian: hauetako bat zenbakiaren zati osoa eta bestea, zati irudikaria errepresentatzen du. Labview programan, zenbaki konplexuak puntu flotanteko zenbakiak direnez, zenbaki mota hauek ere kolore laranjaz adierazten dira.

• **Complex Single** --- Prezisio sinpleko konplexua, puntu flotanteko zenbakiak balio erreal eta irudikariz osatutako 32 bitetako IEEE prezisio sinpleko formatua dute.

- **Complex Double** --- Prezisio bikoitzeko konplexua, puntu flotanteko zenbakiak balio erreal eta irudikariz osatutako 64 bitetako IEEE prezisio sinpleko formatua dute.
- **Complex Extended** ---- Prezisio hedatuko konplexua, puntu flotanteko zenbakiak balio erreal eta irudikariz osatutako IEEE prezisio hedatuko formatua dute. Memorian, mota honetako zenbakien tamainua eta prezisioa plataformaren arabera aldatzen dira. Windows-ean, hauek 80 bitetako IEEE prezisio hedatuko formatua dute.

Balio boolearrak

Labview programak datu boolearra 8 bitetako balioetan biltegiratzen du. 8 bitetako balio hau zero bada, balio boolearra GEZURRA (False) izango da, honela, zero ez den beste edozein baliok EGIA (Egia) adieraziko du. Labview-n, kolore grisa datu boolearra errepresentatuko du.

Balio boolearrek berarekin erlazionatutako akzio mekanikoa izaten dute. Bi akzio nagusien artean, kisketa eta etengailua daude. Kisketa akzioa, atearen txirrinaren antzekoa da, etengailu akzioak argia duen etengailuaren antzera duen bitartean. Kisketa edo etengailua funtzionatzen hasten den momentua definitu daiteke: pultsatuz, itzaliz edo itzaki arte.

Akzio mekanikoei buruz, NI Example Finder atalean Mechanical Action of Booleans VI-rekin gehiago ikasteko aukera izango dugu.



4-5 Irudia. Boolearraren akzio mekanikoa

String (kateak)

String ASCII karaktereen sekuentzia bat da. Datu mota honek informazio eta datuari buruzko plataforma askeko formatua eskaintzen du.String aplikazio arrunt batzuek ondorengoa izaten dute:

- Testu sinpleko mezuak sortzen dituzte.
- Zenbakizko datua string karaktereak bezala instrumentuetara pasatzen dira eta ondoren, string-ak zenbakizko balioetan bihurtuko dira.
- Zenbakizko datua diskoan biltegiratzen da. Zenbakizko datua ASCII artxiboan biltegiratzeko, diskoaren artxiboan datua idatzi baino lehenago, lehendabizi zenbakizko datua string-ean bihurtu beharko da.
- Elkarrizketa-kutxarekin erabiltzaileari instrukzioak eman edo galdetuko diogu.

Aurreko paneleko leihoan, string-ak taulak, testuz betetako kutxatilak eta etiketak bezala azaltzen dira. Labview eraikitako VI eta funtzioen bitartez, string manipulatuak erabili daitezke, kateen formatua, kateen analisia eta beste zenbait edizio gehitzen zaizkielarik.

Labview programan, string-ak kolore larrosarekin adierazten dira.

Aurreko panelean aurkitzen den string kontroladore edo adierazlea arratoiaren eskuineko botoiarekin klikeatuz, bistaratze mota ezberdinak lortu daitezke:

Bistaratze mota	Deskribapena	Mezua	
Bistaratze arrunta	Kontroladoreen iturria erabiliz, inprimatu daitezkeen karaktereak erakusten dira. Ikusten ez diren karaktereak, aldiz, normalean kutxetan azaltzen dira.	There are tour display types. \ is a backslash.	
`\´ kodeen bistaratzea	Ikusi ezin diren karaktere guztien barra-kodea erakusten du.	There\sare\sfour\sdisplay stypes.\n\\\sis\sa\sbackslash.	
Pasahitza bistaratzea	Espazio karaktere bakoitza izartxo (*) batekin erakusten da.	********	
Bistaratze hexadezimala	Karaktere bakoitzaren ASCII balioa hexadezimalean erakusten du, berezko karakterearen ordez.	5468 6572 6520 6172 6520 666F 7572 2064 6973 706C 6179 2074 7970 6573 2E0A 5C20 6973 2061 2062 6163 6B73 6C61 7368 2E	

Labview-k string-ak punteroak bezala biltegiratzen ditu, 4 bitetako luzera duen balio bati 8 bitetako zenbaki osoez osatutako 1D matrizea jarraituz osatzen delarik.

Zenbakien bidez adierazitakoak

Zenbakien bidez adierazitako kontroladore, adierazle edo konstanteak datu moten konbinazio bat da. Mota honetako zenbakiak balio bikoitza adierazten dute, string eta zenbakizkoa, non zenbakien bidez adierazitako hau zerrendaren balioetako bat delarik. Adibidez, Month izeneko zenbakien bidez adierazitako datu mota sortzen badugu, Month-ari dagokion balio bikoitza posiblea Urtarrila-0, Otsaila-1, eta horrela jarraituko genuke Abendua-11-ra heldu arte. Aurrerantzean, kontroladore baten kasurako dagozkion balio bikoitzak adierazten duen **Properties** elkarrizketa-kutxa dago.

Items	Digital Display		Trank	_
January	0		Insert	
February	1		Delete	
March	2	Ξ	Doloco	_
April	3		Move Up	
May	4			
June	5		Move Down	
July	6			
August	7	•		

4-6 Irudia. Zenbakiz adierazitako Enum kontroladorearen propietateak

Datu mota hau baliagarriagoa da, bloke-diagramako zenbakiak string-ean manipulatzeko errazagoa baita. Ondorengo irudian mota honetako **Month** kontroladorearen balio bikoitzen aukeraketa eta honi dagokion bloke-diagramako terminala aurkitzen da.



4-7 Irudia. Zenabkiz adierazitako kontroladorea aurreko panel eta bloke-diagraman

Dinamikoa

Datu mota dinamikoak Express VI baten bidez sortutako edo eskuratutako informazioa biltegiratzen du. Datu mota hau urdin iluneko terminalarekin azaltzen da. Express VI gehienek datu mota dinamikoa onartu eta/edo itzuli egiten dute.

Datu mota dinamikoa zenbakizko, uhinezko edo boolear datua onartzen duen adierazle edo sarrerarekin konektatu daiteke. Adierazleak graph edo chart grafikoak edo zenbakizko adierazleak izaten dituzte.

Bestelako VI eta funtzioek ez dute datu mota dinamikoa onartzen. Datu mota honek izaten dituen datuak analisatu edo prozesatzeko, eraikitako VI edo funtzioa erabiltzeko, datu mota dinamikoaren bihurketa egin beharko da.

)∞<mark>(⊪(</mark>)

Beste VI eta funtzioez erabilia izan dadin, datu mota dinamikoa zenbakizko, boolear, uhin, eta matrize datu motan bihurtzeko, Convert from Dynamic Data Express VI erabiltzen da. Convert from Dynamic Data Express VI blokediagraman kokatzeko, **Configure Convert from Dynamic Data** elkarrizketakutxa azaltzen da. Convert from Dynamic Data Express VI itzulitako datuak formateatzeko, **Configure Convert from Dynamic Data** elkarrizketakutxa erakutsitako aukerak ditugu.

Datu mota dinamikoa matrize motako adierazle bati konektatzerakoan, Labview Convert from Dynamic Data Express VI automatikoki bloke-diagraman kokatzen da. **Configure Convert from Dynamic Data** elkarrizketa-kutxa irekitzeko, Convert from Dynamic Data Express VI bi bider klikeatuko dugu; honela, datua matrizean nola azaltzen den kontrolatuko da.

Express VI-ez erabilia izan dadin, datu mota dinamikoa zenbakizko, boolear, uhin, eta matrize datu motan bihurtzeko, Convert from Dynamic Data Express VI erabiltzen da. Convert from Dynamic Data Express VI bloke-diagraman kokatzeko, **Configure Convert from Dynamic Data** elkarrizketa-kutxa azaltzen da. Datu mota dinamikoan bihurtzeko, datu mota aukeratzeko helburuz, elkarrizketa-kutxa hau erabiliko da.

C. <u>Dokumentaziorako kodea:</u>

VI-ak mantendu eta aldatzeko pertsona profesionalek dokumentazio egokiaren balioa dakite. Kodearen etorkizuneko aldaketa errazteko, bloke-diagrama modu zuzen batean dokumentatu beharko da. Gainera, VI-aren amaiera eta aurreko paneleko objektuak azaltzeko, aurreko paneleko leihoa modu egoki batean dokumentatu beharko da.

Aurreko paneleko leihoak dokumentatzeko, zinten ertzak, deskribapenak, VI Propietateak eta diseinu egokia erabiltzen dira.

Zinten ertzak eta deskribapenak

Zinten ertzak kontroladore edo adierazleen gainean klikeatzerakoan erakusten diren azalpenak dira. Adibidez, tenperatura Celsius graduetan edo sarrera algoritmo baten barnean nola lan egiten duen azaltzeko zinten ertza gehituko genuke.

Deskribapenak kontroladore eta adierazle zehatzen informazio gehigarria ematen du. Kontroladoreetan zinten ertzak eta deskribapenak gehitzeko, kontroladore edo adierazlea arratoiaren eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Description and Tip** hautatuko da.

VI Propietateak

VI-en deskribapenak sortu eta VI-etatik HTML artxiboetara lotzeko edo laguntza ematen duten artxiboak konpilatzeko, **VI Properties** elkarrizketa-kutxatik dokumentazioaren konponentea erabiliko dugu. VI Propietateak erakusteko, aurreko panel edo bloke-diagraman aurkitzen den VI-aren ikurra arratoiaren eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **VI Properties** hautatu edo **File** >> **VI Properties** aukera hartu beharko da. Ondoren, **Categories** menu hedatutik **Documentation** aukeratuko da. VI exekutatzen dagoen bitartean, ezin izango da elkarrizketa-kutxa honetara sartu.

Orri hau hurrengo konponenteekin osatuta dago:

- VI description --- VI ikurraren gainetik kurtsorea mugitzen bada, Context Help leihoan azaltzen den testua izango du. Deskribapenean, hizki beltzetan (negrita) formateatu nahi badugu, edozein testuaren inguruan eta etiketak erabili beharko ditugu. Hala ere, era programatu batean VI-aren deskribapena editatu nahi izanez gero, VI Description propietateak erabili ahal izango dira.
- Help tag --- HTML artxiboaren izena edo laguntza konpilatua duen artxibo bati erlazionatu nahi diren gaiaren hitz kodearen aurkibidea izaten du barnean. Dena den, Help erabili dezakezu: era programatu batean laguntza etiketa ipintzeko, etiketen propietateak dokumentatuko dira.
- Help path --- Context Help leihora lotu nahi dugun HTML artxiboaren bidea edo laguntza konpilatua duen artxiboa izango du. Eremu hau hutsik baldin badago, Detailed help lotura ez da Context Help leihoan azalduko eta Detailed help botoia iluna egongo da.
- **Browse** --- HTML artxibo batetik zehar itsasoratu edo laguntza konpilatua duen artxiboa Help path bezala erabiltzeko, honek elkarrizketa-kutxa artxiboa erakusten du.

Kontroladore eta adierazleen izendapena

Kontroladore eta adierazleen izen logiko eta deskriptiboak emanez, aurreko panelari baliagarritasuna gehitzen diogu. Adibidez, kontroladore bat *Temperature* izena esleitzen badiogu, erabiltzailea ezin izango du jakin ze unitate erabiltzen ditugun. Hala ere, *Temperature* $^{\circ}C$ bezala izendatuz, aurreko panelean informazio gehiago erantsiko genuke. Orain badakigu unitate metrikoetan tenperaturak sartzen.

Programazio grafikoa

Labview programaren modu grafikoak bloke-diagramaren dokumentazioan laguntzen duen bitartean, aparteko azalpenak etorkizunean VI-en aldaketak egiten direnean, oso lagungarria da. Bloke-diagramari dagozkion azalpenen artean, bi mota bereiz daitezke: algoritmoen funtzio edo eragiketa deskribatzen duen azalpenak eta kableen artean pasatzen diren datuen helburua azaltzeko erabiltzen diren azalpenak, hain zuzen ere. **Functions >> Programming >> Structures >> Decorations** azpipaletatik etiketa arruntak, etiketen lanabesa edo etiketa askeak sartu daitezke. Defektuz, etiketa askeek kolore horia izaten dute.

Functions Search String Vi	ew	×				
♥ Programming U Structures U Decoration	าร					
A1	/	1				
Free Label	Thin Line	Thin Line with				
	/	1				
Flat Frame	Thick Line	Thick Line wit				
► Instrument I/C)					
▶ Mathematics						
🕨 🕨 Signal Processi	ing					
🕪 Data Communi	Data Communication					
↓ Connectivity						
► Express						
► Favorites						
Select a VI						

4-8 Irudia. Etiketa arruntak, etiketen lanabesak eta etiketa askeak

VI-ak azaltzeko, ondorengo arauak bete beharko dira:

- Bloke-diagraman azalpenak erabili, egiten ari den kodea azaltzeko.
- Labview kodea grafikoa delako dokumentatuta dagoen bitartean, blokediagramako funtzioak deskribatzeko etiketa askeak erabili.
- Funtzio eta subVI-en deietan etiketak ez erabili, luzeak eta maneiatzeko zailak izaten baitira.
- Beren erabilpena identifikatu nahian kable luzeak etiketatzeko, kolore zuriko sakonera duten etiketa aske eta txikiak erabili. Kableak etiketatzea desplazamendu erregistroetatik datozen eta bloke-diagrama osoa okupatzen duten kableentzako baliagarria da. Desplazamendu erregistroei buruz informazio gehiago lortzeko, gai honen *Case Structures* atala daukagu.

- Estrukturen funtzionalitate nagusiena zehazten duten estrukturak etiketatu.
- Konstanteen jatorria adierazten duten konstanteak etiketatu.
- Bloke-diagraman erabilitako algoritmoak dokumentatzeko, etiketa askeak erabili.

D. <u>While egiturak:</u>

Testuan oinarritutako programazio-hizkuntzan ditugun Do Loop edo Repeat-Until Loop-en antzera, While egiturak baldintza gertatu arte azpidiagrama bat exekutzen du.

Ondorengo irudia, Labview-rekin programatutako While egitura bati dagokio.

🔁 L	Intitil	ed 1 B	lock D	iagram *	i:			
Eile	<u>E</u> dit	⊻iew	<u>P</u> roject	<u>O</u> perate	<u>T</u> ools	<u>W</u> indow	<u>H</u> elp	1
		\$ ₽		8 😰	40 f	a n 1	3pt Applie	ation
								*
l c								- L
				Numeri	c			
				1.231	Ţ			=
				DBL				
					_			
							net?	
	i.				Ľ		······	
•		_			1			* }

4-9 Irudia. While egitura

While egitura **Structures** izeneko paletan aurki dezakegu. Honela, paletatik While egitura hautatu ostean, kurtsorearen bidez errepikatu nahi den bloke-diagramako atalaren inguruan paletatik lortutako ikurra arrastratu beharko da. Arratoiaren botoia askatzerakoan, While egitura adierazteko erabiltzen den laukia aukeratutako eremua itxiko du.

While egituran bloke-diagramako objektuak gehituz, egitura honen barnean ipintzeko aukera izango dugu.

Aholkua: While egitura behin gutxienez exekutatuko da.

While egiturak baldintza terminala, sarrera terminala, balio boolear zehatza jaso arte azpidiagrama exekutatuko du. Baldintza terminalaren portaera eta itxura normalean

Stop if true (egia bada, gelditu) izaten da; honela izanda, While egituraren baldintza terminalak EGIA (true) balioa jaso arte azpidiagrama exekutatzen jarraituko du.

Egituraren terminala edo ertza arratoiaren eskuineko botoiarekin klikeatuz eta atzipen zuzeneko menutik **Continue if true** aukera hartuz, bere portaera eta itxura aldatu ahal izango da. Bigarren modu honetan dagoenean, While egituraren baldintza terminalak GEZURRA (false) balioa jaso arte azpidiagrama exekutatzen jarraituko du. Dena dela, funtzionamendu lanabesa erabiliz baldintza terminala klikeatuz ere, egiturari dagokion baldintza aldatzeko aukera izango genuke.

Iterazioen kontaketa zero-n hasiko da beti; lehenengo iterazioan zehar, iterazioaren terminalak **0** balioa itzuliko du.

Aurrerantzean aurkezten den bloke-diagraman, subVI-aren irteera 10.00 baino handiagoa edo berdina eta **Enable** kontroladorea *Egia* (True) izan arte While egitura exekutatuko da. Eta (And) funtzioak bi sarrerak egia izanda soilik, *egia* itzuliko luke. Gainontzeko kasuetan, *Gezurra* itzuli luke.

Hurrengo adibidean, egitura infinitu baten probabilitatearen handipena azaltzen da. Orokorrean, egitura gelditzeko erabili izan zen baldintzetako bat eukitzea nahi izango genukeen portaera izango litzateke, bete behar diren baldintzak hainbeste eskatu ordez.



4-10 Irudia. Infinitu egitura posiblea

Tunelen estruktura

Tunelak egituren barne eta kanpo datuak sartzen ditu. Tunela While egituraren ertzean solidozko bloke bat bezala ageri da. Datua egitura terminalen ostean, egituratik kanpo pasatzen da. Tunelak datua egituratik pasatzen duenean, datua tunelera heldu arte egitura ez da exekutatuko.

Ondorengo bloke-diagraman, iterazio terminala tunelera konektatuta dago. While egitura exekutatzeari bukatu arte, tunelaren balioa ezin izango da **Iterations** izeneko adierazlera pasatu.



4-11 Irudia. While egituraren tunela

Iterations adierazlean, iterazio terminalaren azken balioa soilik erakutsiko da.

Erroreen gestioa While egiturak erabiliz

While egiturako iterazioa gelditzeko, error cluster-a While egituraren baldintza terminalera konektatu daiteke. Error cluster-a, baldintza terminalera lotzerakoan, terminaletik error cluster-aren **status** parametroaren EGIA edo GEZURRA balioa soilik pasatuko da. Honela, errorea gertatzen baldin bada, While egitura geldituko litzateke.

Error cluster-a baldintza terminalera konektatuta dagoenean, atzipen zuzeneko gaien artean aurkitzen diren **Stop if True** eta **Continue if True**, **Stop on Error** eta **Continue while Error**-engatik aldatuko lirateke.

Hurrengo irudian, egitura noiz geldituko den zehazteko, error cluster eta stop botoia batera erabiltzen ari dira. Hau egitura gehienak gelditzeko gomendatzen den metodoa da.



4-12 Irudia. While egitura baten geldiketa

E. For egiturak:

For egiturak, azpidiagrama zenbait alditan exekutatzen du. Behean agertzen den irudiak Labview programaren bidez errepresentatutako For egitura aurkezten du. For egituraren fluxu-diagrama, While-enarekin konparatuz, aberatsagoa da eta flexibilitate gehiago onartzen du.

100 N		
]	

4-13 Irudia. Labview For egitura

Egitura mota hau **Programming** >> **Structures** paletan aurki daiteke. While egitura For egituragatik aldatzeko, While egituraren ertza eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Replace with For Loop** hautatu beharko da. Kontaketa terminalaren balioak (sarrerako terminala), azpidiagrama zenbat aldiz errepikatzen den adierazten du.

Iterazio terminala (irteerako terminala), iterazio osoen zenbaketaz arduratzen da.

Iterazioen kontaketa zero-an hasiko da beti; lehenengo iterazioan zehar, iterazioaren terminalak **0** balioa itzuliko du.

For While egiturarekin konparatuz duen diferentzia, For egitura zenbait auditan exekutatzen dela izango litzateke. While egitura baldintza terminalaren balioa existitzen bada soilik, azpidiagrama exekutatzen bukatuko du.

Hurrengo For egiturak 100 segundutan zehar segunduro zenbaki aleatorioa sortzen du eta zenbaki aleatorio honen erakusketa zenbakizko adierazlean ematen da.



4-14 Irudia. For egituraren adibidea

Zenbakizko bihurketa

Labview-k zenbakizko datu mota, zeinudun edo zeinugabeko zenbaki osoa bezala, puntu flotanteko zenbakien balioak bezala, edo zenbaki konplexuen balioak bezala errepresentatu daiteke, hau gai honetako *Labview Data Types* atalean azaltzen da. Normalean, funtzio baten sarreretara datu mota ezberdinak konektatzerakoan, funtzioak irteera formatu handiago edo luzeago batean itzultzen du. Labview-k bit gehien

erabiltzen dituen errepresentazioa aukeratzen du. Biten kopurua berdina bada, programa honek zeinugabekoa aukeratuko luke. Adibidez, biderketa gauzatzen duen funtzio bati DBL eta I32 konektatzen baditugu, emaitza I32 izango da. 64 bitetako zeinudun zenbaki osoa behartua egongo da, prezisio bikoitzeko puntu flotanteko zenbakia baino bit gutxiago erabiltzen baititu. Biderketa funtzioaren sarrera txikienak puntu grisa erakutsiko du.



4-15 Irudia. Zenbakizko bihurketaren adibidea

Hala ere, For egituraren kontaketa terminalak beste era batera lan egiten du. Prezisio bikoitzeko puntu flotanteko zenbakia 64 bitetako kontaketa terminalera konektazen badiozu, Labview programak zenbakizko balio luzea 32 bitetako zeinudun zenbaki oso batean bihurtuko luke. Bihurketa hau bihurketa arrunt estandarraren kontrakoa izan arren, beharrezkoa da, For egitura zenbaki oso batez soilik exekutatu daitekeelako.



4-16 Irudia. For egituraren bihurketa

F. <u>VI-aren denborazko erlazioa:</u>

Egitura iterakzio bat exekutatzen bukatzen duenean, berehala hurrengo iterazio exekutatzen hasiko da, stop gelditzeko baldintza ailegatu ezean. Kasu gehienetan, iterazioaren denbora edo maiztasuna kontrolatzea beharrezkoa izaten da. Adibidez, datuak eskuratzen ari bazara eta 10 segunduetatik behin datua eskuratu nahi baduzu, egituraren iterazioen denbora kontrolatzeko modua nahitanahiezkoa izango da.

Gainera, exekuzioa maiztasun konkretu batean sortzea beharrezkoa ez bada, prozesadoreak beste eginkizun batzuk betetzeko denbora izan beharko du, erabiltzailearen interfazea prozesatzean bezalaxe. Atal honek egiturak zenbatzeko zenbait metodo aurkezten ditu:

Itxaroteko funtzioak

Denbora zehatz batean VI-a lo egin zezan, egituraren barnean itxaroteko funtzioa kokatuko genuke. Honen bidez, itxaroteko denboran zehar prozesadoreak beste eginkizun batzuk bidaltzeko aukera izango luke. Itxaroteko funtzioak funtzionamendu sistemaren milisegunduko erlojua erabiltzen du.

Ð

Wait Until Next ms Multiple funtzioa milisegundu kontadorea monitorizatu eta kontadore honek guk zehazturiko kantitatearen multiplo batera ailegatu arte itxaroten du. Funtzio hau aktibitateak sinkronizatzeko erabiltzen da. Egituraren exekuzio maiztasuna kontrolatu ahal izateko, funtzio hau erabiltzen da. Egituraren lehenengo iterazioaren exekuzio-maiztasuna ez dago zehaztua.

1

Wait (ms) funtzioak milisegundo kontadorea guk zehazturiko sarreraren kantitate berdina zenbatu arte, itxarongo du. Funtzio honek egituraren exekuziomaiztasuna behintzat guk zehazturiko sarrera kantitatea dela bermatzen du.

Oharra: Time Delay Express VI Wait (ms) funtzioaren antzera konportatzen da.

Igarotako denbora

Kasu batzuetan, VI-aren momentu konkretu batean zenbat denbora igaro den zehazteko balio du. Beraz, Elapsed Time Express VI hasierako momentua espezifikatu ondoren igarotako denbora adieraziko luke. VI honek VI-a exekutatzen jarraitzen duen bitartean, denbora ez galtzea uzten du. Funtzio honek ez du prozesadoreak beste eginkizun batzuk betetzen uzten.



G. <u>Datu iteraktiboen tranferentzia:</u>

Egiturekin programatzen dugunean, egituraren aurretiko iterazioetatik datuak sartu daitezke. Adibidez, egituraren iterazio bakoitzean datuen zati bat eskuratzen ari bazara, eta datuen bost zatietako batezbestekoa egin nahi bada, egituraren aurretiko iterazioen datuak gogoratu beharko dira. Desplazamendu erregistroak egitura baten iterazio batetik hurrengo iteraziora balioak trasferitzen ditu.

<u>*Oharra*</u>: Berrelikatutako nodoak aurretiko iterazioetatik informazioa gordetzeko Labview programan erabilitako beste metodo bat da.

Desplazamendu erregistroak testuan oinarritutako programazio-hizkuntzan erabiltzen diren aldagai estatikoen antzekoak dira.

Egituraren bidez, aurretiko iterazio batetik hurrengo iteraziora balioak pasatzeko desplazamendu erregistroak erabiltzen dira. Terminal bikoitzen bidez adierazten dira, egituraren bertikaleko ertzean eta bata bestearekiko kontrakoa izanik.



4-17 Irudia. For egitura baten desplazamendu erregistroa

Egituraren eskuin aldeko terminala gora begira dagoen gezi bat dauka eta iterazioan lortutako balioa izaten du: Labview-k eskuin aldeko erresgistroarekin konektatuta dauden datuak hurrengo iteraziora transferitzen ditu. Egitura exekutatu ostean, egituraren eskuin aldeko terminalak desplazamendu erregistroan biltegiratutako azkenengo balioa itzultzen du.

Desplazamendu erregistro bat sortzeko, egituraren eskuin edo ezker ertzean arratoiaren eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Add Shift Register** hautatu beharko da.

Desplazamendu erregistroa edozein motatako datua transferitu eta automatikoki lehenengo aldiz desplazamendu erregistrora konektatutako objektuaren datu motara aldatuko da. Terminal bakoitzera kableatutako datua mota berdinekoa izan beharko litzateke.

Egitura batera desplazamendu erregistro bat baino gehiago gehitu daitezke. Eragiketa ezberdinak egin behar baditugu, datuen balioak biltegiratzeko desplazamendu erregistro bat baino gehiago erabiliko genuke.



Desplazamendu erregistroen hasieraketa

VI-a exekutatzean, egituraren lehenengo iteraziotik pasatzen den desplazamendu erregistroaren balioa erreseteatuz, desplazamendu erregistroaren hasieraketa lortuko genuke.



Ondorengo irudian, For egitura bost bider exekutatzen da, desplazamendu erregistroa duen balioa kasu bakoitzean bat gehituz. For egituraren bost iterazioen ostean, desplazamendu erregistroa azken balioa pasatuko dio adierazleari, 5. VI-a exekutatzen den bakoitzean, desplazamendu erregistroa *0* balioarekin hasieratuko da.

Desplazamendu erregistroa hasieratzen ez baldin bada, egiturak idatzitako balioa desplazamendu erregistrorako erabiltzen du egituraren azkenengo exekuzioan edo datu motaren defektuzko balioa egitura inoiz ez bada exekutatu.

VI baten ondorengo exekuzioen artean informazioaren egoera gordetzeko, hasieratu gabeko desplazamendu erregistroa erabili. Hurrengo irudiak hasieratu gabeko desplazamendu erregistroa erakusten digu.



Aurretiko irudian, For egitura bost bider exekutatzen da, desplazamendu erregistroa duen balioa kasu bakoitzean bat gehituz. VI-a lehenengo aldiz exekutatzean, desplazamendu erregistroa 0 balioarekin hasieratzen da, 32 bitetako zenbaki oso baten defektuzko balioa delarik. For egituraren bost iterazioen ostean, desplazamendu erregistroa azken balioa pasatuko dio adierazleari, 5. VI-a hurrengoko aldiz exekutatzean, desplazamendu erregistroa 5 balioarekin hasieratuko da, aurretiko exekuzioaren azkenengo balioa delarik. For egituraren bost iterazio eta gero, desplazamendu erregistroak azkenengo balioa, 10, adierazleari pasatuko dio. VI-a berriz ere exekutatzen baldin bada, desplazamendu erregistroa 10 balioarekin hasieratuko litzateke eta horrela, prozesu bera betetzen egongo da etengabe. Hortaz, VI-a itxi artea, desplazamendu erregistroek aurretiko iterazioaren balioa gordeko lukete.

Desplazamendu erregistroen pilaketa

Desplazamendu erregistro mota hauek egituraren aurretiko iterazioetatik datuak sartzea ahalbidetzen digu. Desplazamendu erregistroen pilaketak aurretiko iterazioen balioak gogoratu eta balio horiek hurrengo iteraziora eramaten ditu. Desplazamendu erregistroen pilaketa sortzeko, ezkerreko terminala arratoiaren eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Add Element** aukera hartuko dugu.

Desplazamendu mota hau egituraren ezkerreko aldean soilik gerta daiteke, eskuineko terminalak oraingo iteraziotik hurrengo iteraziora sortutako datuak soilik transferitzen baititu, ondorengo irudian ikus daitekeen bezala.



Aurretiko irudiko ezkerreko aldean beste elementu bat gehitzen bada, azkenengo bi iterazioen balioak hurrengo iteraziora eramaten dira, desplazamendu erregistroaren goiko aldean biltegiratutako iterazio berrienaren balioarekin. Terminalaren beheko aldeak aurreko iteraziotik pasatutako datuak biltegiratzen ditu.

H. <u>Datuen marrazketa:</u>

Datu sinpleak marrazteko grafikoak erabiltzen dira. Atala honek grafikoen erabilera eta pertsonalizatzeari buruzko zenbait azalpen emango ditu.

Chart grafiko mota

Grafiko mota hau maiztasun konstante batez eskuratutako datuen marrazki bat edo gehiago erakusten dituen zenbakizko adierazleen mota berezia da. Chart grafikoak, marrazki bat edo bat baino gehiago adierazi dezake. Beheko irudian bi marrazki ezberdin dituen chart grafikoa adierazten da: Sawtooth Waveform eta Triangle Waveform.



4-18 Irudia. Chart grafiko mota

- Datu berria erakusteko, chart nola gaur eguneratzen den konfiguratu daiteke. Chart gaur eguneraketa modua finkatzeko, arratoiaren eskuineko botoiarekin chart grafikoa klikeatu eta atzipen zuzeneko menutik Advanced >> Update Mode hautatuko dugu. Grafiko mota hauek, datuak erakusteko hiru modu ezberdin erabiltzen dituzte:
- Strip Chart --- Datuak jarraitasunez exekutatzen ari dira, grafikotik zehar ezkerretik eskuinera desplazatuz eta datu zaharrak ezkerrean eta datu berriak eskuineko aldean daudelarik.
- Scope Chart --- Datuen zati bat soilik erakusten du, pultsu edo uhin bat bezalakoa, grafikotik zehar ezkerretik eskuinera desplazatuz. Balio berri bakoitzerako, grafiko honek azkenengo balioaren eskuineko aldean marrazten du. Marrakia marrazketa eremuaren eskuienko ertzera ailegatzean, Labview-k marrazkia ezabatu eta ezkerreko aldetik berriro ere marrazten hasiko da. Scope Chart grafikoaren pantailaren deskribapena osziloskopioaren antzekoa da.
- Sweep Chart --- Honek aurreko moduaren antzera lan egiten du baina datu zaharrak eskuineko aldean eta datu berriak ezkerreko aldean egongo dira, zuzen bertical baten bitartez banaturik egongo direlarik. Marrazkia marrazketa eremuaren eskuineko aldean dagoenean Labview-k ez du sweep chart –ean dagoen marrazkia ezabatzen. Sweep chart EKG erakusketaren oso antzekoa da.

Hurrengo irudiak chart grafiko motaren modu bakoitzaren adibide bana aurkezten du. Scope chart eta sweep chart pantailen deskribapenak osziloskopio bati dagokionaren antzekotasun nabarmena dute. Marrazki baten deskribapenak gainkarga gutxiago suposatzen duenez, scope chart eta sweep chart-en marrazkien erakusketak strip chartetakoak baino askoz ere azkarragoak dira.



4-19 Irudia. Datuak erakusteko modu ezberdinak

Chart grafiko moten kableaketa

Chart grafiko motari irteera eskalarra zuzenean kableatu dakioke. Ondorengo irudian chart grafikoaren terminala sarrerako datu motarekin konektatuta agertzen da.



4-20 Irudia. Zenbakizko kontroladore bat Chart grafiko batera konektatu

Cluster & Variant paletan aurkitzen den Bundle funtzioa erabiliz, chart grafiko motek marrazki bat baino gehiago erakutsi dezakete. Hurrengo irudian, Bundle funtzioak hiru VI-en irteerak batera lotzen ditu chart grafiko batean erakusteko helburuarekin.



4-21 Irudia. Chart grafikoari marrazki ezberdinen kableaketa

Bundle funtzioaren irteera lotzeko, chart grafikoaren terminalak bere itxura aldatzen du. Gainera, marrazki gehiago gehitu ahal izateko, Bundle funtzioa tamainuz aldatzen duen posizionamendu lanabesa erabiliko da.

Graph grafiko mota

Mota honetako grafikoak dituzten VI-ak normalean datuak matrize baten bildu eta ondoren, graph batean datuak marrazten ditu.



4-22 Irudia. Graph grafiko mota

Graph eta graph XY grafiko motak dituen Graph Adierazleen paletan graph ezberdinak aurki daitezke. Graph grafiko motak balio sinpleko funtzioak soilik marrazten ditu, y = f(x) bezala, puntuak x ardatzean zehar uniformeki sakabanatzen direlarik, eskuratutako denbora-aldakorreko grafikoak bezalaxe. XY grafikoak edozein puntu multzoa erakusten du, uniformeki laginduta edo ez.

Marrazki ezberdinak erakusteko, marrazkiaren legenda tamainuz aldatu beharko da. Aurreko panelean lekua gordetzeko eta marrazkien artean konparaketak egiteko, marrazki ezberdinak erabiltzen dira. Graph eta graph XY grafiko motak automatikoki marrazki ezberdinetara moldatzen dira.

Marrazki bakarreko graph grafiko motak

Graph grafiko motak marrazki bakarreko graph grafiko moten zenbait datu mota onartzen ditu. Graph-k balioen matrize sinplea onartzen du, datuak graph-aren puntuak bezala interpretatzen ditu eta x = 0 –tik hasten den batetik x indizea gehitzen du. Graph-k hasiera bezala x balioa duen cluster-a onartzen du, Ax, eta y datuen matrize bat. Graph-k grafiko moten datu mota ere onartzen du, datuak eraman, denbora hasi eta grafikoaren At-z arduratzen delarik.

Graph grafiko motak onartzen dituen datu moten adibideak lortzeko, *labview\examples\general\graphs\gengraph.llb* –ean aurkitzen den Waveform Graph VI-ra abiatu zaitez.

Marrazki ezberdineko graph grafiko motak

Graph grafiko motak marrazki ezberdinetako graph grafiko moten zenbait datu mota onartzen ditu. Graph grafikoak 2D balioen matrizea onartzen du, non matrizearen ilara bakoitza marrazki simple bat adierazten duen. Graph-k datuak graph-aren puntuak bezala interpretatzen ditu eta x = 0 –tik hasten den batetik x indizea gehitzen du. Graph eskuineko botoiarekin klikeatuz eta atzipen zuzeneko menutik **Transpose Array** hautatuz, 2D matrize datu mota graph batera konektatu ahal izango da, matrizearen zutabe bakoitza marrazki bat bezala gestionatzeko. Hau DAQ dispositibo baten kanal ezberdinak lagintzeko batez ere oso baliagarria izaten da, dispositiboak datua banatutako zutabe bat bezala kanal bakoitzera biltegiratutako 2D matrizeak bezala itzuli dezakeelako.
Ondorengo datu mota onartzen duen graph-aren adibide bat lortzeko, *labview\examples\general\graphs\gengraph.llb* –ean aurkitzen den Waveform Graph VI-a topatu eta (Y) Multi Plot 1 graph-era abiatu zaitez.

Graph-k hasiera bezala x balioa duen cluster-a ere onartzen du, Ax balioa, eta y datuen 2D matrize bat. Graph-k y datuak graph-aren puntuak bezala interpretatzen ditu eta x = 0 –tik hasten den batetik x indizea gehitzen du. Honelako datu mota maiztasun berean laginduta dauden seinale ezberdinak erakusteko oso baliagarria da.

Ondorengo datu mota onartzen duen graph-aren adibide bat lortzeko, *labview\examples\general\graphs\gengraph.llb* –ean aurkitzen den Waveform Graph VI-a topatu eta (Xo =10, Dx = 2, Y) Multi Plot 2 graph-era abiatu zaitez.

Graph grafiko motak cluster-ak dituen marrazkien matrizea onartzen du. Cluster bakoitzak y datuez osatutako 1D matrizea izaten du. Barneko matrizeak marrazkiaren puntuak deskribatzen ditu, eta kanpoko matrizeak marrazki bakoitzeko cluster bat dauka. Aurrerantzean agertzen den aurreko panelak matrize honen y cluster-a azaltzen du.

Marrazki bakoitzean elementu bakoitzaren zenbakia ezberdina bada, 2D matrizearen ordez, marrazkien matrizea erabili beharko da. Adibidez, kanal bakoitzaren kantitate ezberdinak erabiliz kanal ezberdinen datuak lagintzen baditugu, 2D matrizearen ilara bakoitza elementu kopuru berdina izan behar duenez, 2D matrizearen ordez datu egitura hau erabili beharko da. Cluster matrize baten barneko matrizeen elementuen kopurua aldatu daiteke. Ondorengo datu mota onartzen duen graph-aren adibide bat lortzeko, *labview\examples\general\graphs\gengraph.llb* –ean aurkitzen den Waveform Graph VI-a topatu eta (Y) Multi Plot 2 graph-era abiatu zaitez.

Graph grafiko motak x balioko hasiera, delta x balioa eta cluster-ak dituen matrizea onartzen du. Cluster bakoitzak y datuez osatutako 1D matrizea izaten du. Bundle funtzioa matrizeak clurter-etan lotzeko eta Build Array funtzioa lortutako cluster-ak matrizean eraikitzeko erabiliko dira. Gainera, Build Cluster Array funtzioak guk zehazturiko sarrerak dituen cluster-en matrizeak sortzen ditu. Ondorengo datu mota onartzen duen graph-aren adibide bat lortzeko, *labview\examples\general\graphs\gengraph.llb* –ean aurkitzen den Waveform Graph VI-a topatu eta (Xo =10, Dx = 2, Y) Multi Plot 3 graph-era abiatu zaitez.

Graph-k x balioko hasiera, delta x balioa eta y datuez osaturiko matrizea duen cluster-en matrize bat onartzen du. Hasiera puntu bakarra eta marrazki bakoitzaren x eskalarentzako gehikuntza adierazi daitekeenez, marrazki ezberdineko graph datu moten artean, hau orokorrena da. Ondorengo datu mota onartzen duen graph-aren adibide bat lortzeko, *labview\examples\general\graphs\gengraph.llb* –ean aurkitzen den Waveform Graph VI-a topatu eta (Xo =10, Dx = 2, Y) Multi Plot 1 graph-era abiatu zaitez.

Graph grafiko motak Express VI-ekin erabilia izaten den datu mota dinamikoa ere onartzen du. Gainera, seinalarekin erlazionatutako datua, datu mota dinamikoek seinaleari buruzko informazioa ematen duten ezaugarriak izaten dituzte, seinalearen izena edo datua eskuratua izen den data eta denbora besteak beste. Ezaugarriek graph grafiko motan seinalea nola azaltzen den zehazten dute. Datu mota dinamikoa kanal ezberdinak baditu, graph grafikoak kanal bakoitzetik marrazkia erakusten du eta automatikoki marrazkiaren legenda eta x eskalaren denbora marka formateatzen da.

Marrazki bakarreko XY graph grafiko mota

XY graph-k marrazki sinpleko XY graph baten hiru datu onartzen ditu. Grafiko mota honek x eta y matrizea dituen cluster-a onartzen du. Ondorengo datu mota onartzen duen graph-aren adibide bat lortzeko, *labview\examples\general\graphs\gengraph.llb* –ean aurkitzen den XY Graph VIa topatu eta (X and Y arrays) Single Plot graph-era abiatu zaitez.

XY graph-k puntuen matrize bat ere onartzen du, non puntua x eta y balioa duen cluster-a izango den. . Ondorengo datu mota onartzen duen graph-aren adibide bat lortzeko, *labview\examples\general\graphs\gengraph.llb* –ean aurkitzen den XY Graph VI-a topatu eta (Array of Pts) Single Plot graph-era abiatu zaitez. Grafiko mota honek, datu konplexuen matrizea ere onartzen du, zati erreala x ardatzean eta zati irudikaria y ardatzean marraztuta daudelarik.

Marrazki ezberdineko XY graph grafiko mota

Marrazki ezberdinak erakusteko, XY graph-k hiru datu mota onartzen ditu. Grafiko honek marrazkien matrize bat onarzten du, non marrazkia x eta y matrizea duen cluster-a den. Ondorengo datu mota onartzen duen graph-aren adibide bat lortzeko, *labview\examples\general\graphs\gengraph.llb* –ean aurkitzen den XY Graph VI-a topatu eta (X and Y arrays) Multi Plot graph-era abiatu zaitez.

XY graph-ak marrazkien cluster matrize bat ere onartzen dute, non marrazkia puntuak dituen marrazkia bat den. Puntua x eta y balioa duen cluster-a da. Ondorengo datu mota onartzen duen graph-aren adibide bat lortzeko, *labview\examples\general\graphs\gengraph.llb*—ean aurkitzen den XY Graph VI-a topatu eta (Array of Pts) Multi Plot graph-era abiatu zaitez. Grafiko mota honek marrazkien cluster matrize bat ere onartzen dute, non marrazkia datu konplexuekin osatutako matrize bat den, zati erreala x ardatzean eta zati irudikaria y ardatzean marraztuta daudelarik.

I. Case egiturak:

Case egiturak bi azpidiagrama edo gehiago ditu. Azpidiagrama bakarra soilik momentuan ikusgai izango da eta egitura momentuan behin soilik exekutatuko da. Sarrerako balio batek zein azpidiagrama exekutatzen den adieraziko digu. Case egitura testuan oinarritutako programazio-hizkuntzan aurkitzen diren to switch statements edo if...then...else statements-en oso antzekoak dira.

Case egituraren goiko aldean dagoen kasuaren etiketa hautatzailea, kasua erdiko aldean eta gehikuntza eta gehikuntzaren kontrako geziak honen alde bietan duen hautatzailearen balioaren izena dauka.



Gehikuntza eta gehikuntzaren kontrako gezietan klikeatuz, eskuragarri ditugun kasu guztietatik desplazatzeko aukera izango dugu. Hala ere, kasuaren ondoan dagoen beheko gezia klikeatuz, eta menu hedatutik kasu bat hautatuz ere lortu genezake.

Zein kasua exekutatuko den zehazteko, sarrerako balio bat, edo hautatzailea, hautatzailearen terminalera konektatu beharko da.

Hautatzailearen terminalera zenbaki osoa, balio boolearra, string, edo zenbakiz adierazi daitekeen balioa konektatu beharko litzateke. Hautatzailearen terminala Case egiturako ezkerreko aldearen edozein lekutan kokatu daiteke. Hautatzailearen terminalaren datu mota boolearra baldin bada, egiturak EGIA eta GEZURRA kasuak izango ditu. Hautatzailearen terminala zenbaki osoa, string, edo zenbakiz adierazi daiteken balioa bada, egituraren kasu kopurua edozein izan daiteke.

Balio-tarteen kanpo dauden balioak gestionatzeko, Case egituraren defektuzko kasua zehaztuko da. Beste modu batean, sarrerako balio guztiak esplizituki zerrendatu beharko litzateke. Adibidez, hautatzailea zenbaki osoa baldin bada eta 1, 2 eta 3-rako kasuak zehaztuz gero, sarrerako balioa 4 edo espezifikatu gabeko bestelako zenbaki osoa bada, defektuzko kasua zehaztu beharko genuke exekutatzeko.

Kasuak batu, biderkatu, ezabatu edo berantolatzeko, Case egituraren ertzean eskuineko botoiarekin klikeatu eta defektuzko balioa hautatu beharko da.

Kasuaren aukeraketa

Aurrerantzean ikus daitekeen irudia, erabiltzaileak nahi duen tenperatura adierazteko Celsius edo Fahrenheit unitatea aukeratzen duen arabera, VI honek kode ezberdinak exekutatzeko Case egitura erabiltzen du. Goiko bloke-diagramak Egia kasua erakusten du lehenengo planoan. Erdiko bloke-diagraman, aldiz, Gezurra kasua aukeratzen da. Kasu bat aukeratzeko, balioak editatzeko, identifikatutako hautatzaile kasuan balioa sartu beharko da eta balioak editatzeko etiketen lanabesa erabiliko litzateke. Beste kasu bat hautatu eta gero, kasu hori aurretik azalduko da, beheko aldean aurkitzen den irudian ikusten den bezala.

Hautatzailearen balioa sartu eta hautatzailearen terminalarekin konektatutako objektuaren datu mota bera ez baldin badu, balioa kolore gorriz agertuko da, egitura exekutatu baino lehen balioa ezabatu edo editatu behar dela adierazteko, eta VI-a ez da exekutatuko. Gainera puntu flotanteko zenbakiak ezin izango dira kasuaren hautatzailearen balio bezala erabili. Puntu flotanteko balio bat kasuaren hautatzailearen etiketan sartuz gero, balioa kolore gorriz agertuko da, egitura exekutatu baino lehen balioa ezabatu edo editatu behar dela adierazteko.

Sarrera eta irteerako tunelak

Case egitura batean sarrera eta irteera tunel ezberdinak sortu daitezke. Sarrerak kasu guztietarako eskuragarri daude baina kasuak sarrera bakoitza erabiltzea ez dute behar. Dena dela, kasu bakoitzerako edozein irteera tunela definitu beharko genuke.

Adibide hau kontutan hartuko dugu: Bloke-diagramako Case egitura irteera tunel bat dauka baina gutxienez kasu hauetako batean irteerako balioa ez dago tunelarekin konektatuta. Kasu hau exekutatzen baldin bada, Labview-k ez du jakingo zein balio kokatzen den irteeran. Labview programak errore hau tunelaren erdia zuriz utzita adieraziko du. Konektatu gabeko kasua agian ez da momentu horretan bloke-diagraman ikusgai dugun kasua izango.

Errore hau zuzentzeko, konektatu gabeko irteera tunelak dituzten kasuetara mugitu eta tunelera irteera bat kableatu ezazu. Hau lortzeko, irteera tunela eskuineko botoiarekin klikeatu eta atzipen zuzeneko menutik **Use Default If Unwired** hautatuz, tunel datu motaren defektuzko balioa konektatu gabeko tunel guztietarako erabili ahal izango da. Irteera modu egoki batez kasu guztietan kableatuta badago, irteera tunelak kolore solidoa izango du.

Hala ere, **Use Default Unwired** aukera erabiltzea ekiditu beharko da. Aukera hau erabiliz, bloke-diagrama ez da era egoki batean dokumentatuko eta kodea erabiliz beste programadore batzuk okertu daitezke. Aukera hau erabiltzen hautatzen baldin bada, erabilitako defektuzko balioa tunelera konektatutako datu motaren defektuzko balioa dela kontziente izan beharko zara. Adibidez, tunela datu mota boolearra bada, defektuzko baloa Gezurra izango da.

Datu mota	Defektuzko balioa
Zenbakizkoa	0
Boolearra	Gezurra
String	Hutsik ("")

4-23 Irudia. Defektuzko balioen datu mota

Adibideak

Ondorengo adibideetan, zenbakizko balioak tuenelen bidez Case egiturara pasatzen dira eta batuketa edo biderketa dago, hautatzailera terminalera konektatzen den balioaren arabera.

Case egitura boolearra

Hurrengo irudia Case egitura boolearrari dagokio. Ilustrazioa sinplifikatzeko, kasuak elkarren artean gainezartzen dira.



Kontroladore boolearra hautatzailearen terminalera konektatuta dagoela egia bada, VI-ak zenbakizko balioak gehituko ditu. Beste kasuetan, VI-ak zenbakizko balioak kenduko ditu.

Zenbaki osoen Case egitura

Hurrengo irudia zenbaki osoen Case egiturari dagokio.

Integer (zenbaki osoa) Controls >> Text Controls paletan aurkitzen den testu eraztunaren kontroladorea da; zenbakizko balioak testuzko gaiekin erlazionatzen ditu. Hautatzailearen terminalera konektatutako testu eraztunaren kontroladorea 0 bada (batuketa), VI-ak zenbakizko balioak batuko ditu. Balioa 1 bada (kenketa), VI-ak zenbakizko balioak kenduko ditu. Testu eraztunaren kontroladorea aurreko bi balioen ezberdina bada, VI-ak zenbakizko balioak batuko ditu, hori defektuzko kasua baita.

String Case egitura

Hurrengo irudia string Case egiturari dagokio.

String-a batuketa baldin bada, VI-ak zenbakizko baliaok batuko ditu. String kenketa bada, aldiz, VI-ak zenbakizko balioak kenduko ditu.

Zenbakien bidez adierazitako Case egitura

Hurrengo irudia zenbakien bidez adierazitako Case egiturari dagokio.

Zenbakien bidez adierazitako kontroladore mota aukeratu egin behar duen gaien zerrenda ematen dio erabiltzaileari. Zenbakien bidez adierazitako kontroladore motak zenbakizko balio eta string etiketei buruzko informazioa izaten du. Case egituraren atzipen zuzenetik Add Case For Every Value aukera hartzean, kasuaren hautatzaileak zenbakien bidez adierazitako kontroladore motan gai bakoitzaren string etiketa erakusten du. Case egiturak zenbakien bidez adierazitako kontroladore motaren momentu hartako gaietan oinarritutako azpidiagrama kasu egokia exekutatzen du. Aurreko bloke-diagraman **Enum** batuketa baldin bada, VI-ak zenbakizko balioen batuketa egiten du. **Enum** kenketa baldin bada, aldiz, VI-ak zenbakizko balieon kenketa gauzatzen du.

Erroreen gestioa Case egitura erabiliz

Ondorengo irudian Case egiturari dagokio, non kasuak definitzen dituen error cluster-a ikus daiteke.

Error cluster-a Case egituraren hautatzailearen terminalera konektatzerakoan, kasuaren hautatzailearen etiketak bi balio erakusten ditu --- Errorea eta errorerik ez --- eta Case egituraren ertza kolorez aldatzen da --- gorria errorerako eta berdea errorerik ez dagoenean. Errore bat gertatzen baldin bada, Case egiturak Errore azpidiagrama exekutatuko du.

Error cluster-a aukeratze terminalera konektatzean, Case egiturak cluster-aren egoera (status) boolearra soilik ezagutuko du.

J. Nodoen formula:

Labview programan ekuazio korapilatsua erabili nahi duzunean, bloke-diagramako funtzio aritmetiko ezberdinak ez dira batera konektatu behar. Ekuazioak ingurune familiar, matematiko batean garatu eta aplikazio batean sartu beharko dira.

Nodoen formula bloke-diagraman eragiketa matematikoak egiteko testuan oinarritutako nodo komenigarria da. Ezin izango da kanpoko edozein kode edo aplikaziora sartu edo behe-mailako funtzio aritmetikoak konektatu ekuazioak sortzeko. Gainera, testuan oinarritutako ekuazioen adierazpenak, nodoen formulak if statements, while egiturak, for egiturak, eta do egiturak, C programadoreekin familiarrak diren testuan oinarritutako bertsio hauek onartu dezake. Programazio elementu hauek C programan zer aurkitzen duzunarekin oso antzekoa da baina ez dira berdinak izango.

Nodoen formula korapilatsuak edo aldagai asko dituzten ekuazio eta existitzen den testuan oinarritutako kode bat bezala erabiltzeko oso baliagarria da. Grafikoki berriro ere sortu beharrean, existitzen den testuan oinarritutako kodea nodoen formulan kopiatu eta itsatsi daiteke.

Nodoen formulako sarrera eta irteerako terminalak sortzeko, nodoaren ertza arratoiaren eskuineko botoiarekin klikeatu eta **Add Input** edo **Add Output** atzipen zuzeneko menutik hautatuko dugu, gero aldagaiaren balioa sarrera edo irteeran sartu beharko da. Ekuazioa egituran sartu; ekuazio bakoitzaren egitura koma komarekin (;) bukatu beharko da.

Erabakiak hartzerakoan ere, nodoen formula erabili ahal izango dugu. Aurrerantzean agertzen den bloke-diagramak nodoen formulan if-then egitura erabiltzeko bi modu ezberdin erakusten ditu. Egitura biek emaitza bera sortzen dute.



Nodoen formula eragiketa ezberdin asko gauzatu ditzake. Nodoen formularen funtzio, eragiketa eta sintaxiari buruz informazio gehiago lortzeko, *Labview Help* atala kontsultatu beharko dugu.

<u>**Oharra:**</u> Formula Express VI-ak formula matematikoak sortzeko kalkulagailuaren interfazea erabiltzen du. Oinarrizko kalkulagailuak gauzatzen dituen funtzio kopuru handiagoa egin ahal izateko, Express VI hau oso baliagarria izango da. Formula Express VI-ei buruzko informazio gehiago lortzeko, *Labview Help* atalera abiatuko gara.

Azterketa askea: Froga

1. Hauetatik zein identifikatzen du bloke-diagrama batean aurkitzen den kontroladore edo adierazlea?

- a. Azpititulua
- b. Lekua.
- c. Etiketa.
- d. Balioa.

2. Egitura hauetatik, zein exekutatuko litzateke behin gutxienez?

- a. While egitura.
- b. For egitura.

3. Hauetatik zein aurkitzen da bloke-diagraman SOILIK?

- a. Kontroladorea.
- b. Konstantea
- c. Adierazlea.
- d. Paneleko lotura.

4. Zein akzio mekanikoa egiten du gezurra (false) egoeran dagoen boolearra egia (true) izatera pasatzea, bere gainean klikeatzen eta askatu arte egia (true) egoera mantendu eta Labview-k balioa irakurri behar duenean?

a.	Askatua izan arte aldatu.
b.	Askatzerakoan aldatu.
c.	Askatua izan arte mantendu
d.	Askatzerakoan mantendu.

Azterketa askea: Frogaren erantzuna

1. Hauetatik zein identifikatzen du bloke-diagrama batean aurkitzen den kontroladore edo adierazlea?

- a. Azpititulua.
- b. Lekua.
- c. Etiketa.
- d. Balioa.

2. Egitura hauetatik, zein exekutatuko litzateke behin gutxienez?

a. While egitura.

b. For egitura.

3. Hauetatik zein aurkitzen da bloke-diagraman SOILIK?

- a. Kontroladorea.
- b. Konstantea.
- c. Adierazlea.
- d. Paneleko lotura.

4. Zein akzio mekanikoa egiten du gezurra (false) egoeran dagoen boolearra egia (true) izatera pasatzea, bere gainean klikeatzen eta askatu arte egia (true) egoera mantendu eta Labview-k balioa irakurri behar duenean?

a.	Askatua izan arte aldatu.
b.	Askatzerakoan aldatu.
с.	Askatua izan arte mantendu.
d.	Askatzerakoan mantendu.

Gai honi dagozkion zenbait adibide jarraian adierazitakoak dira:

Temperature Analysis.vi:

Adibide honen algoritmoa tenperaturaren 40 lagin eskuratzea izango da, beraz, datuen eskuraketaren bidez errepresentatua egongo da. Tenperatura lagin kopuruen arabera adierazia dugu.

Ts laginketarako erabilitako laginketa denbora da, hau da, lagin bat zenbatero hartzen den. Denbora hau hardware edo softwarearen bidez zehaztu daiteke; hardawarearekin (txartela erabiliz) askoz ere zehatzagoa da nahiz eta gehienetan softwarea erabiltzen den. Adibide honetan sotwarea erabiltzen ari da, 250 milisegunduro 40 lagin hartzea esaten du, beraz, Ts = 250mseg.

Lagin honek **for egituraren** bitartez, matrize batean gordetzen du. **For egitura** honek exekutatzeari amaitzen duenean, lagin hauekin zenbait gauza egin genitzake:

- Funtzio polinomiko baten bidez modelizatu, Labview-k antzematen du.
- Batezbestekoaren balioa, bariantza, desbiazio tipikoa, balio maximo eta minimoa... adierazten diguten estadistikako zenbait balio daude.





Graph Circle.vi:

Zirkunferentzia baten ekuazio matematikoak ondorengoak dira:

x = RSen (x) y = RCos (y) Ax = RSin (Af)Ay = RCos (Af)





Cursor Legend kurtsore bat egiten denean balioen adierazlea da. Kurtsorea askea edo grafikara moldatzea egin dezaket.

Ariketa honen aldaketa bezala, pantailan zenbait zirkulu kontzentrikoak errepresentatzea izango litzateke, bakoitzari kolore bat esleituz edo banan-banan etengabe agertzen joatea.

EUITI Bilbao

Graph Circle_1.vi:

Adibide honetan, bere ekuazioagatik zentraturiko elipsea sortu egin da. Honetan egin daitezkeen aldaketen artean, inklinazio edo aldapak egitea izango litzateke.



Graph Circle_1A.vi:

Ariketa honetako bloke-diagraman paraleloki, hau da, batera edo modu konkurrentean, bi **for egitura** lan egiten ari dira. Zenbait kasutan, **bi for egitura** hauek **while** baten barnean egongo dira, egoera hau askotan emango da. Hala ere, **for egitura** bat bestearekin lotzen dituzten kablerik ez dago, bestela murrizketa (restricción) egongo litzateke; modu konkurrentean lan egiteko hauen artean ezin izango da inongo konekziorik egon.



Ikus daitekeenaz, grafiko batean bi maximo baldin badaude, maiztasun bikoitza izango du.



Calculator.vi:

Balioen eta aukeratzen denaren funtzio, A eta B kontroladoreen arteko batuketa, kenketa, biderketa edo zatiketa gauzatuko du. Hautatze hau "function" aldagaiak kontrolatuko du:

- Function = 0 --- Batuketa.
- Function = 1 --- Kenketa.
- Function = 2 --- Biderketa.
- Function = 3 --- Zatiketa.



Prozesu hau jarraiki errepikatu dadin, while egitura batean sartuko nuke guztia.

Case_1.vi:

Case egitura select egituraren eboluzio bat da. Azkenengo kasi honetan, elementua egia bada, goikoa beteko litzateke eta gezurra bada, aldiz, behekoa.

Labview programan **case egitura** bloke-diagraman aurkitzen da: View – Functions Palette – Structures – Case Structure.





Adibide honetan bi kasu ezberdin bakarrik ditugunez, egia gezurra, batuketa kenketa, **case egitura** egitea ez du merezi; **select egitura** bat jartzea egokiagoa izango litzateke.

Case 2.vi:

Ariketa honetan, batuketa kenketa eragiketarako **text ring** erabili izan da; honen bidez zein eragiketa egin behar den aukeratzen da. Menuetan kontzeptu hau oso erabilia izaten da.





Case 2A.vi:

Adibide honek ALU_1 izeneko subVI-a dauka. Era modular batean lan egitea komenigarria da eta beti kode edo funtzioen berrerabilpena eta mantenimendua egin beharko da. Horrela, subVI gehien erabiltzea saiatuko gara.

🔁 case_2A.vi Front I	anel *		
<u>File E</u> dit <u>V</u> iew <u>P</u> roject	<u>O</u> perate <u>T</u> ools	<u>W</u> indow <u>H</u> elp	
수 🕹 🔵 🗉	13pt Applicatio	n Font 🛛 🔫 🖳	
			^
N1		N2	
4,0 6,0 4	2,44	4,0 6,0	() 10,00
2,08,0	2,0-	8,0	
0,0 10,0	0,	,0 10,0	E
ON/OFF	Resultado	Operación	
\bigcirc	0,24	Dividir	3
			· · · · · · · · · · · · · · · · · · ·



"Operación " aldagaia kolore urdinez adierazita dagoenez, emango duen balioa zenbaki oso bat izango da eta zein eragiketa gauzatuko den zehaztuko du.

Case 3.vi:

Ariketa honetan string datu mota da erabiltzen dena, hau da, testuak zein balio duen arabera, akzio edo funtzio ezberdinak egingo dira. Ondorengo adibideetan informazioa testu moduan bidaltzen dela ikusi daiteke:

- String: Zatitu.
- String: Batu.
- String: Ezer.

😫 case_3	.vi Front P	anel *		
<u>Eile E</u> dit	view Project	Operate <u>T</u>	ools <u>W</u> indo	
_	<u>· & </u>		cation Font	
N1		N2		
4,0	6,0	4,0	6,0	
2,0-	-8,0	2,0-	-8,0	
	1	-		
0,0	10,0	0,0	10,0	
5,78		9 4,6	/	=
String		Res	ultado	
sumar		11	,46	
ON/OFF				
				~
I < □				
Case_3.v	vi Block Diag	ram *	Window Hol	
Eile Edit Vie	vi Block Diag >>> Project 이 장 (ram * perate <u>T</u> ools	 Window Help	
to case_3.v Eile Edit ⊻ie ₽	vi Block Diag w Project O & () III (perate <u>T</u> ools	Window Help	
Eile Edit ⊻ie	21 Block Diag 3w Project Q 2관 (● III (ram * perate <u>T</u> ools 2 9 - C	Window Help	
to case_3.v Eile Edit ⊻ie ₽	vi Block Diag w Project O 2 () () (ram * perate <u>T</u> ools 2 9 9 9 9 9 7	Window Heli	
Eile Edit Vie	21 Block Diag	ram * perate <u>T</u> ools	Window Hel	
Eile Edit Vie	vi Block Diag	perate Tools	Window Heli	
Eile Edit Vie	21 Block Diag aw Project O	ram * perate <u>T</u> ools	Window Help 13pt A Resultado	
Eile Edit Vie	vi Block Diag aw Project O C III (C III) (C IIII) (C III) (C I	ram * perate <u>T</u> ools	Window Heli Mindow Heli 13pt 4	
File Edit Vie E E E N1 DBI N2 DBI String String	vi Block Diag	ram * perate <u>T</u> ools	Window Help	
Eile Edit Vie	21 Block Diag	ram * perate <u>T</u> ools	Window Heli 13pt 4 13pt 4 Resultado DBL ON/OFF	
File Edit Vie File Edit Vie N1 N2 String abc	vi Block Diag	ram * perate <u>T</u> ools	Window Hel 13pt A 13pt A Resultado DBL ON/OFF	
Eile Edit Vie	VI Block Diag	ram * perate <u>T</u> ools	Window Hel 13pt 4 13pt 4 Resultado DBL ON/OFF	
Eile Edit Vie	vi Block Diag	ram * perate <u>T</u> ools	Window Hel 13pt A 13pt A Resultado DBL ON/OFF	

Beraz, string batekin ez duzu aukeratzen egin nahi duzuna, kasu honetan idatzi beharko duzuna baizik. Hau normalean bloke-diagraman gauzatzen da. Desegokia den termino bat idaztean, defektuz jarrita dagoena egingo luke; hau programa ez blokeatzeko helburutzat egiten da.

Case 3A.vi:

1240 zenbakia jartzea eta "1240" zenbakizko string-a ipintzea ez da gauza bera izango; testu formatuan doanean, karaktereak (letrak) ASCII formatuan bidaltzen ditugulako eta hau dela eta, ez dute berdina okupatzen.





Case_5.vi:

Adibide hau oso garrantzitsua da. **While egitura** bat daukagu eta egitura hau bere barnean beste hiru blokez osatuta dago. Profile jartzen duen Labview irtengunean barneko hiru blokeen zenbait ezaugarri azaltzen dira: zenbat denbora behar duten exekutatzeko...

🔁 case_5.vi Front Panel *			
<u>File E</u> dit <u>V</u> iew <u>P</u> roject <u>O</u> perate	<u>T</u> ools <u>W</u> indow <u>H</u> elp		
다 🐼 🔘 💵 13pt App	lication Font 🛛 🔹 🚛 🕶 🗐		? 3≯
N1 4.0 6.0	N2	RESULTADO	^
2,0- 1 -8,0	4,0 6,0	7,55	_
0,0 10,0	2,00,0	Boolean 2	
5,46	÷) 2,09	Boolean Boolean 3	
OldVal 0,00			E
error out	error in (no error)	error in (no error) 2	
status code	status code	status code	
	√ ⊕ 43	√ → 4	
source	source	source	
VER PALETA DE CLUSTER			•
▲	III		►:

While egituraren barnean hiru estruktura daude, modu konkurrente edo paraleloan lan egiten dutenak. Hiru hauen artean ez dago haririk elkArren artean lotzen dituenak. Honez aparte, cluster bat ere ikus dezakegu. Ikuspuntu grafikotik cluster honek bus baten funtzioa egiten du.

Blokeetako bi **Case egiturei** dagokie, hala ere, azkenengoak **Event egitura** aurkezten du. Labview-k **Event egiturei** programazio grafikoaren itxura ematen die eta programa honetan oso erabiliak izaten dira. Gertaera (evento) gauzatu arte, ez du ezer egiten; kasu honetan konkretuki, errorea dagoen edo ez kontutan izanda, gauza bat edo beste egingo luke, hortaz, prozesu zehatz batean errore bat gertatu dela ohartarazten digu. Adibide honetako **Event egituran** hiru gertaera bereiz daitezke:

- 1) X segundu ezer egin gabe igarotzen badira, _____ gertatuko da.
- 2) N1 etengailuan aldaketaren bat baldin badago, ______.
- 3) Errorea baldin badago, _____.

LabView Manuala



Determine Warning 4 6.vi:

Adibide hau Labview 8.0 eskuliburuaren 4.gaiko 6.ariketari dagokio.

Erabiltzaileak sarrerak bezala tenperatura, tenperatura minimoa eta tenperatura maximoa erabiltzen ditu VI bat sortzeko. String oharra sarreretako erlazioaren arabera zehaztua egongo da. Dena dela, gerta daiteke VI-a era oker batean lan egitea; erabiltzaileak tenperatura minimoa baino txikiagoa den tenpetura maximoa sartu dezake. VI-a aldatu beharko da, erabiltzaileari errorea dagoela adierazteko beste string bat sortuko da: "Upper Limit < Lower Limit". "Warning" boolearraren egoera egia bada, errorea dagoela adieraziko du.

😫 Determine Warnings	=4_6.vi Front Panel	
<u>File E</u> dit <u>V</u> iew <u>P</u> roject <u>O</u>	perate <u>T</u> ools <u>W</u> indow <u>H</u> elp	
수 & ● ■	13pt Application Font 🛛 👻 🚛 🖬 🖓	
Current Temp		^
÷) 30		
May Temp	Warning Text	
() 30	Heatstroke Warning	=====
Min Temp	Warning?	
÷) 10		~
I I I I I I I I I I I I I I I I I I I	m	▶

Erreferentziazko adibide bat dela esan dezakegu, algoritmoa mantenduz inplementazioa aldatu daiteke; hau da, **Select egitura** batetik **Case egitura** batera pasatzen da baina beste guztia berdina izaten jarraitzen du.



Determine More Warnings.vi:

Ariketa honetan, **While egitura** batetik **Case egitura** ezberdinak paraleloki lan egiten egon daitezkeela erakusten digu. Kasu honetan egia eta gezurrarekin (V/F) osatutako **Case egiturak** ditugu (**Select egituragatik** ordezkatu daitezkeelarik). Hauek paraleloki lan egiteko, beraien artean ez da kablerik existitu behar; kable honek datuen fluxu erlazioa emango liguke eta honek guztia aldatzea ekarriko luke.



Determine More Warnings 4 9.vi:

VI honek "Heatstroke Warning" edo "Freeze Warning" gertatu den adieraziko digu, sarrera bezala dituen tenperatura sarrerak kontutan hartuta. VI hau hedatu beharko dugu "High Wind Warning" haizearen abiaduraren irakurketan edo haizearen abiadura maximoaren doiketan oinarrituta gertatu den arabera.

🔁 Determine M	lore Warnings_	4_9.vi Front P	anel *				
<u>File E</u> dit <u>V</u> iew <u>F</u>	Project Operate	<u>T</u> ools <u>W</u> indow <u>H</u>	lelp				
中國	i 13pt App	lication Font 🛛 🔫	₽ ▼ •		¢0-	3	WARMING
stop							*
STOP							
							100
Curre	ent Temp	Current Wind Spe	ed				
78	And the lot of the lot	J 60					
							-
Max	Tomp	May Wind Speed		Warnin	a Text		E
() 30	remp	() 80		Freeze	e Warning		
Min T	[emp			Warnin	ia?		
() 10				0			
Taala	Due file						
100IS	Prome						
1.4		n	1.2.				1.11

Oharrak string sinpleak bezala erakusten dira. Adibidez, "Heatstroke Warning" eta "High Wind Warning" gertatzen badira, string-ak "Heatstroke eta High Wind Warning" irakurri beharko luke.



Temperature Control.vi:

Ariketa honek **tot-down diseinua** aurkezten du. Instrumentu birtual baten egitura arrunta errepikatzen da: desplazamendu erregistroa, hasieraketak, eta **while egitura**ren barnean zenbait egitura sartzen ditu, guzti horiek era konkurrente eta paraleloan lan egiten daudelarik.



Honen analisia tenperatura bi puntuen artean (maximo eta minimoa) dagoen egiaztatzea izango da.

While egituraren goiko aldean aurkitzen den desplazamendu erregistroak hiru balio ezberdinen (oraingo balioa eta aurreko biak) batezbestekoa gauzatzen du eta bundle funtzioa duen chart motako grafiko bakar batekin errepresentatzen da.

Case egitura baten aplikazio garrantzitsuena programazioaren barnean izango litzateke; egitura honetan irteera guztiak zerbaitera kableatuta egon beharko dira bestela errorea emango luke, sarreretan, aldiz, hau ez da beharrezkoa.

Gainera, guzti honetaz aparte, **Case egitura** baten barnean dagoen desplazamendu erregistro bat dago. Honek oraingo balioa balio maximoa izatea egingo du.



Adibide honetan momentu batean irakurtzen dena, aurreko iterazioan irakurritakoarekin konparatzen ari da:

- Oraingo balioa < aurreko balioa ----- Balioa mantentzen da.
- Oraingo balioa > aurreko balioa ----- Aldaketa dago.

Square Root.vi:

Adibide honek zenbaki erreal baten erro karratua aurkitzea izan du helburu. Elkarrizketa-kutxa baten sarreraren ideia agertzen da. Mezu bat sortzeko, "String Constante" erabiliko da, berriro dimentsionatu eta nahi dena idazten da. Zenbaki negatibo bat agertzen baldin bada, hau adieraziko digun mezu bat azalduko da.



Praktiketan gauza bera baina zenbaki errealekin egin beharrean, zenbaki konplexuekin egitea oso aproposa izango litzateke.

Square Root 2.vi:

Hemen aurreko adibideko algoritmo berbera baina inplementazio ezberdinekin izango genuke.

"Formula Node" erabiltzen da, beraz, C programetako lexiko antzekoa erabiliz, bere barnean kodea testu eran idazten da, sententzia guztiak puntu eta komarekin bukatzen dira (;) eta testu eran erabilitako funtzio matematikoak C-n erabilitakoak izango dira. Sarreretan aldagaiak zeintzuk diren definitu beharko dira eta hauek deklaratu. Hauez aparte, deklaratu beharko ditugun bitarteko aldagaiak ere dauzkagu; aldagai global eta lokalekin ez dute inolako zerikusirik eta programazioan laguntzen dute.

😫 Square Root 2.vi	Front Panel *
<u>File E</u> dit <u>V</u> iew Project	Operate Tools Window Help
수 & 🔵 🗉	13pt Application Font
	^
Value	Square Root Value
16,00	4,00
digital control	*digital indicator*
	-99999.00 indicates an error
	free label
• • • • • • • • • • • • • • • • • • •	m >:

Kasu honetan **"Formula Node"** honen barnean **Select egitura**ren testu bertsioa dugu C lengoaian idatzita:

- $x \ge 0$ bada ------ y = raíz cuadrada de x.
- Bestela balio bat esleitzen da.

🔁 s	quar	e Roc	t 2.vi 8	lock Dia	gram	*			×
Eile	<u>E</u> dit	⊻iew	Project	<u>O</u> perate	<u>T</u> ools	<u>W</u> indow	<u>H</u> elp		-]
	C	\$ ֎		8 🕵	40 5	1 🗘 🚺	3pt Applicat	:iq 🖸	
	Value DBL)	Forn Y 	mula Node = (x>=0) 99999;	e) ? sqrt (x):	Į	Squ 7 J	are Root Va	lue	•
•				m				•	•

Beraz, if then egitura erabili beharrean, C lengoaia agertzen da.

Formula Node Exercise.vi:

Ariketa honen enuntziatua ondorengoa izango litzateke: Ekuazio berdina balioztatu baina 200 puntuentzako eta y aldagaia 0 eta 200 balioan artean egonik.



Adibide honetan, "A" izeneko bitarteko aldagaia dugu, **"Formula Node"** egituraren ezker edo eskuinean (sarrera eta irteera) jartzen delarik. Funtzio bat inplementatu nahi dugu eta funtzio osoa idatzi beharrean, bitarteko aldagaia eskuinean kokatzen da.

Graph batean errepresentatzearen aukera gehitu ahal izango genuke: **"Formula Node"-** tik informazioa puntu puntu ateratzen da eta hau grafiko batean ikusi nahi bada, bloke batean joan beharko da.

Beraz, informazioa matrize batean bihurtu beharko dugu. Bihurketa hau gauzatzeko, "**autoindexamiento**"-aren bitartez, **for egitura** baten barnean lortuko genuke. Grafikoa era jarrai batean ateratzen den arren, 200 laginetan zatituta dago; 200 balio edo puntu dituen matrize unidimentsionala sortzen da. "Plot"-aren bidez ere, **graph**-aren puntu ezberdinak ikusi ahal izango ditugu.

Adibide honi dagokion bloke-diagrama ondorengoa izango litzateke:



Gainera, **XY grafiko** bat ere aurkitzen da baina zenbakizko balioa izan ordez, intentsitatezko balioa dauka; intentsitatezko grafikoari dagokio.

Build Array funtzioa erabili izan da, non y matrize eta x matrizearen bitartez, matrize bidimentsionala lortzen den.

Equations.vi:

Adibide hau trakzio eta konpresio esfortzuak antzemateko erabiltzen da.

Bi ekuazio lineal ditugu (x eta y aldagaiei balio batzuk ematen zaie eta bukatzen da) sarrerak, irteerak...nola doazen ikusteko batez ere. Hala ere, nahi diren ekuazioak jarri daitezke: planoen ekuazioak...Honela, x eta y aldagaiak balioz aldatuz, emaitza ezberdinak lortuko dira. Dena den, prozesu hau era jarrai batean exekutatzea lortu nahi badugu, kontroladore batekin **While egitura** baten barnean sartuko genuke. Exekutatutakoan, froga guztiak egin arte jarraiki funtzionatzen egongo da.

Tools Window Help
plication Font 🔽 🕮 🥵 🦿 👔
y1 26,87 9 y2 5,20



Array Over Threshold.vi:

Ariketa hau matrize, desplazamendu erregistro eta **Case egitura** batez osatuta dago. Honez aparte, **for egitura** bat ere aurkitzen da.

Hala ere, ariketa honetatik gehien interesatzen zaiguna honako hau da: desplazamendu erregistroa egiten ari dena, zein objektua adierazle bat izan ordez, matrize bat den.

😫 Array Over	Threshold.vi Front Pa	nel *		
<u>F</u> ile <u>E</u> dit <u>V</u> iew	<u>Project Operate T</u> ools <u>y</u>	<u>W</u> indow <u>H</u> elp		T
수 관	13pt Application F	iont 👻 🚛 🖬		<u></u>
Ş	Threshold		Nº de Iteraciones 17	^
nI	nput Array	Output Array	Inicial	
.	3,00	2 4,00	3 1 0,00	
	()-4,00	15,00	0,00	
	() 5,00	3,00	0,00	
	4,00	5,00	0,00	
	-2,00	5,00	0,00	
	15,00	0,00	0,00	
	-2,00	0,00	0,00	= =
	1,00	0,00	0,00	
	3,00	0,00	0,00	
	-10,00	0,00	0,00	
	1.00	0,00	0,00	
	15.00	0,00	0,00	
	-4.00	0,00	0,00	
	4,00	0,00	0,00	
	0.00	0,00	0,00	
	0,00	0,00	0,00	
•				→



5.GAIA:DATU ESKURAKETA

Batzuetan oso onuragarria da beraien artean erlazionaturik dauden datuak taldekatzea. Labview-n "arrays" eta "cluster" erabiltzen dira erlazionatutako datuak taldekatzeko. "ARRAYS" taldekatzen ditu datu estruktura berdinean eredu berdineko datuak; eta "CLUSTERS" taldekatzen ditu data estruktura berdinean eredu ezberdineko datuak. "TYPE DEFINITIONS" arrays eta cluster- en erabilpena definitzeko erabiltzen da.

Gai honi dagozkion atalak ondorengoak dira:

- A. Arrays.
- B. Clusters.
- C. Type Definitions.

A. Arrays:

Array bat, elementuak eta dimentsioak definitzen dute. Elementuak, array-a osatzen duten datuak dira. Dimentsioak,ordea, array-aren luzeera, altuera eta sakonera dira. Array bat dimentsio asko izan ditzake, eta memoriak jasan ditzakeen elementu kopurua (elementu bana dimentsioko).

Array-ak edozelan eraiki daiteke: numerikoekin, booleanoekin, grafikoekin, clusterrekin eta abar. Array-ekin lan egitea komenigarria da, antzeko datuak daudenean eta eragiketa berdinak behin eta berriro errepikatzen direnean. Baina bere erabilpen baliagarriena, grafikoetatik edo "loop"-etatik lortutako datuak gordetzeko da, non loop bakoitzak eragindako iterakzio bakoitza array-aren elementu bat izango da.

Murrizketak

Ezin dira sortu array-en array-ak. Hala ere, erabili daiteke array multidimentsionala edo sortu cluster-ez osaturiko array-a, non cluster bakoitzak bat edo array gehiago izan ditzake. Gainera, ezta ezin dira sortu kontrol subpaneleko array-ak, "tab" kontrolekoak, "NET" kontrolekoak, "ActiveX" kontrolekoak, edo XY grafikokoak.

Array sinple baten adibide bat eguzki sistemaren bederatzi planetak zerrendatzen dituen array-a da. Labview-k, bederatzi elementuko dimentsio bateko "string" array bat bezala errepresentatuko du.

Array-en elementuak ordenaturik daude. Array-ak indize bat erabiltzen dute edozein elementura iristeko. Indizea zeroan hasieratuta dago, eta 0 - n-1 tartean mailakatuta dago, non *n* array-aren elementu kopurua den.

Hurrengo adibidean numerikoez osatutako array bat dago. Array-an erakutsitako lehenengo elementua (1) indize 1-ean dago; eta bigarren elementua (2) indize 2-an. Indizean aukeratutako elementua, beti agertuko da array-aren goikaldean eta eskumaldean.

	Array		
÷) o	$\left(\frac{k}{\tau}\right)$ 1	2	

Kontrol eta adierazle array-ak sortzeko

Panel frontalean kontrol edo adierazle array bat sortzeko, kontro paletatik array aukeratu eta panel frontalean jarri. Ondoren, array barruan objetuak sartu, hauek adibidez numerikoak, booleanoak, string-ak cluster-ak, eta abar izan daitezke.

Debekaturik dagoen elementuren bat sartzen saiatu ezkero, adibidez XY grafiko bat, ezinezkoa izango da, ez duelako utziko.

Beharrezkoa da elementaren bat array-ean sartzea, bloke diagraman array-a erabili baino lehen. Bestela, array terminala beltza agertuko da, kortxete huts batekin, eta ez du edukiko inolako daturik erlazionaturik.

Bi dimentsioko array-ak

Aurreko adibideak, dimentsio bateko array-ak aurkezten dituzte. Bi dimentsioko arrayak elementuak hezi moduko batean gordetzen ditu. Beraz beharrezkoak dira bi indize, bata lerroa aukeratzeko eta bestea zutabea, biak zeroan hasieratuta daudelarik.

Array multidimentsional bat sortzeko panel frontalean, klikeatu ezkerreko botoia indizearen gainean eta aukeratu "add dimension".

Array-ak hasieratu daitezke edo ez. Array-a hasieratu gabe dagoenean, dimentsio bakoitzeko elementu kopurua aukeratzen da eta elementu bakoitzeko edukiera. Hasieratu gabe dagoen array bat, dimentsio kopuru finkoa dauka, baina ez elementu kopuru finkoa. Hemen dago hasieratu gabeko bi dimentsioko array bat.

	Array			
÷) o	÷) o	0	A) O	
÷) o	- e) o	<u>/</u> 0		

Eta hemen hasieratutako bi dimentsioko array bat.

0	Array					
÷) o	$\left(\frac{\lambda}{\tau}\right)$ 1	÷2	-2	÷) 5	(<u>x</u>)	
÷) o	9	()-5	()3	e (
	2	-4	2	e ((<u>/</u>)0	
		12	÷) o	e (
	<u>(</u>)4	÷ 5	-6	e) ((A) 0	

Array konstantea sortzeko

Bloke-diagraman array konstante bat sortzeko, aukeratu "array constant" funtzio paletan eta jarri bloke diagraman. Barnean, konstante numerikoa, string konstantea edo cluster konstantea jarri daiteke. Array konstante bat erabili daiteke datu konstanteak gordetzeko, edo oinarri bat bezala beste array batekin konparatzeko. Array konstantea ere oso erabilgarria da datuak subVI batera pasatzeko.



Auto-indexatutako sarrera array-ak

Lotzen bada array bat For Loop edo While Loop batekin, gorde daiteke loop-en iterakzio bakoitza array-aren elementu bat bezala, auto-indexamendua aktibatuz.

Array-a eta loop-a lotzerakoan lotura lauki bete bat da; ordea lotura horretan ezkerreko botoiaren bidez "enable indexing" aukeratuz laukian bi kortxete agertuko dira, auto-indexamendua adieraziz.

<u>Sarrera array-ak</u>: Auto-indexamendua aukeratzen badugu array eta For Loop sarrera baten loturaren artean, Labview zuzenean erlazionatzen ditu kontaketa terminala eta array-aren tamaina, beraz, kontaketa terminala ez da beharrezkoa kableatzea.

For Loop-ak erabili daitezke prozesatzeko array elementu bat momento oro, arrazoi honengatik, For Loop eta array bat lotzerakoan Labview-k zuzenean auto-indexamendua aktibatzen du. Auto-indexamendua ez bada nahi, ezkerreko botoiarekin "disable auto-indexing" aukeratzearekin nahikoa da.

Hurrengo irudian ikusten den bezala, For Loop-a burutu egingo da array-aren elementu kopuruaren hainbat aldiz. Normalean, For Loop-aren kontaketa terminala konektatuta ez badago zirkuitua irekita egongo da eta ez du funtzionatuko. Ordea, kasu honetan bai.



Auto-indexamendua lotura bat baino gehiagotan aukeratzen bada, edo kontaketa terminala kableatzen bada, kontaketa aukera guztien txikiena izango da. Adibidez, bi array auto-indexatuak loop-era konektaturik badaude, 10 eta 20 elementuekin bakoitzak, eta kontaketa terminala 15 balioarekin kableatzen bada, loop-a 10 aldiz burutuko da, eta loop-ak soilik indexatuko ditu bigarren array-aren lehenengo 10 elementuak.

<u>Irteera array-ak</u>: Array bat eta loop baten irteeraren arteko lotura indexatzen bada, irteerako array-a elementu berri bat jasoko du loop-aren iterakzio bakoitzeko. Beraz, auto-indexatutako irteerako array-aren tamaina, iterakzio kopuruaren berdina izango da.

	Array
÷) o	0,0820728
	0,972135
	0,585047
	0,732743
	0,815253


Bi dimentsioko array-ak sortzeko

Erabili daitezke bi For Loop bata bestearen barnean, bi dimentsioko array-ak sortzeko. Kanpoko For Loop-a lerroak sortuko ditu, eta barneko For Loop-a zutabeak.



	Array				
÷Ο	0,0388541	0,227772	0,844211	0,382	0,493588
÷)o	0,164103	0,175908	0,317174	0,654277	0,209567
	0,491562	0,914595	0,59597	0,479513	0,218326
	0,589723	0,80918	0,687781	0,225751	0,10459

B. <u>Clusters:</u>

Cluster-etan taldekatzen dira mota ezberdineko datu elementuak. Cluste adibide bat da Labview-ren errore cluster-a, non bateratu egiten dira balio boolearra, balio numerikoa eta string-a.

Cluster batean elementu ezberdina elkartzeak, eliminatzen ditu hainbat kable konekzio bloke-diagrama, hau errazagoa eta sinpleagoa geratuz, eta gainera murrizten ditu subVI batek behar dituen konekzio terminal kopurua.

Cluster eta array elementuak ordenaturik egon arren, banandu behar dira cluster-aren elementuak "unbundle by name" funtzioaren bitartez, beraietara banan-banan iristeko. Cluster-ak, array-ak ez bezala, tamaina zehatza daukate. Array-ak bezala, Cluster-ak kontrol bat edo adierazle bat izan daiteke. Baina, cluster batean ezin dira nahastu kontrolak eta adierazleak.

Kontrol eta adierazle clusterrak sortzeko

Panel frontalean cluster bat sortzeko, kontrol paletatik cluster-a aukeratu eta atera. Ondoren, bere barnean edozelako elementua sartu baiteke: numerikoa, boolearra, stringa, array-a, cluster-a, eta abar.



Cluster konstanteak sortzeko

Cluster konstante bat sortzeko, bloke-diagraman funtzio paletan "cluster constant" aukeratu eta bloke-diagramara atera. Ondoren, bere barnean konstante numerikoa, constante booleanoa, string konstantea, cluster konstantea eta abar sartu daiteke. Cluster konstante bat erabili daiteke datu konstanteak gordetzeko edo, oinarri bezala beste cluster batekin konparatzeko.

Panel frontalean kontrol edo adierazle cluster bat badaukagu eta. Sortu nahi badugu cluster konstante bat bloke diagraman elementu berdinekin, panel frontaleko cluster-a bloke diagramara eraman daiteke, panel frontaleko cluster-ean ezkerreko botoia sakatuz eta CREATE-CONSTANT auketatuz.

Cluster-aren ordena

Cluster-aren elementuen ordena bertan sartu izanarena da, hau da, lehenengo sartu dena elementu 0 izango da, bigarrena elementu 1 eta abar. Elementu bat ezabatzen bada, ordena aldatu egiten da automatikoki. Cluster-aren ordena finkatu egingo du blokediagraman "bundle" edo "unbundle" funtzioetan elementuen ordena. Cluster-aren ordena ikusi eta aldatu daiteke ezkerreko botoia sakatuz eta "reorder controls in cluster" aukeratuz.



Hemen ikusi dezakegu cluster-aren ordena aldatzeko pantaila. Elementu bakoitzaren kutxatxo beltza, bere oraingo posisioa adierazten du; eta kutxatxo zuria, posizio berria. Cluster-aren elementuen posisio berria adierazteko, "clic to set to" kutxatilan orden berria jarri eta ondoren elementua aukeratu. Elementuaren orden berria jarriko da eta beste elementuen ordena automatikoki egokituko da. Aldaketak gordetzeko "OK" sakatu, edo aurreko ordena mantentzeko "X".

Cluster-ak beraien artean kableatzeko, elementu kopuru berdina izan behar dute. Gainera, elementuen ordena ere berdina izan behar da, konekzioak datu mota berdinen artean izateko, bestela konekzioa ezinezkoa izango litzateke.

Cluster funtzioen erabilpena

Cluster funtzioak, cluster-ak eratu eta eraldatzeko erabiltzen dira. Adibidez, ondorengo akzioak egin daitezke:

- Cluster baten elementu bat atera.
- Elementu bat gehitu cluster-era.
- Cluster-a apurtu bere elementuetan.

Bunble funtzioarekin cluster-a lotzeko; **bundle** eta **bunble by name** funtzioekin cluster-a eraldatu daiteke; **unbundle** eta **unbundle by name** funtzioekin cluster-a banatu daiteke.

Bunble, bundle by name, unbundle eta unbundle by name funtzioak bloke-diagraman jarri daitezke cluster-aren gainean ezkerreko botoiaz "cluster & varian palete"

aukeratuz. Bundle eta unbundle funtzioak zuzenean dituzte beharrezkoak diren terminal kopuruak. Bundle by name eta unbundle by name funtzioak ordea agertzen dira lehenengo elementuarekin bakarrik, ondoren, "positioning tool" bitartez handitu egiten da beharrezkoak diren elementuak agertzeko.

Cluster-a lotzeko

Bundle funtzioaren bidez lotu egiten dira elementuak cluster batean, edo elementu baten balioa aldatzeko, elementu guztien balioak aldaku gabe. "positioning tool" bitartez funtzioaren tamaina aldatu daiteke edo, sarrera elementu baten gainean ezkerreko botoiaz "add imput" aukeratuz.



Cluster-a eraldatzeko

Cluster sarrera kableatzen bada, cluster hori aldatu daiteke kableatuz bakarrik aldatu nahi den elementua.



Bundle by name funtzioa erabiliz, ere eraldatu daiteke cluster-a, baina oraingoan, elementuen izenak agertuko dira. Bundle by name funtzioak, bundle funtzioaren modu berean lan egiten du, baina elementuak beraien ordenagatik agertu beharrean, beraien izenaren bidez adierazten dira. Kasu honetan, sarreren kopurua ez du zertan izan behar irteera cluster-aren elementu kopuruaren berdina.

Hemen ikusten da aurreko adibide berdina baina bundle by name funtzioarekin. Panel frontala berdina izango da.



Cluster-a banatzeko

Unbundle funtzioaren bidez banatu daiteke cluster-a bere elementu guztietan. Unbundle by name funtzioarekin, ere banatu daitezke cluster-aren elementuak, baina soilik aukeratutako izenenak, hau da, ez da beharrezkoa sarrera elementu guztiak banatzea.



Errore cluster-ak

Labview badauka erroreak adierazteko cluster-ak. Errore cluster batek ondorengo informazioa ematen du:

- <u>Status</u>: balio boolearra, true adieraziko duena errore bat bagoenean.
- <u>Code</u>: balio numerikoa, errorea zenbakatu egiten duena.
- <u>Source</u>: string bat, errorea non gauzatu den adierazten duena.

status code	
source	
	-

C. <u>Type definitions:</u>

Type definitions erabili daiteke custom array eta cluster-ak definitzeko.

Diseinatuta kontrolak (Custom controls)

Custom kontrola eta adierazleak erabiltzen dira panel frontalean eskuragarri dauden objektuen kopurua handitzeko. Sortu ahal dira konponente ezberdinak, jadanik labviewk dituen kontrol eta adierazletaz aparte. Sortutako custom kontrola edo adierazlea gorde daiteke direktorio batean edo LLB batean, eta ondoren erabili beste aplikazio batzuetarako. Custom kontrola edo adierazlea ikono bat bihurtu daiteke eta "controls" paletan sartu.

"Control editor" leihoarekin, kontrolak eta adierazleak pentsonalizatu daitezke. Adibidez, aldatu daiteke existitzen den kontrol edo adierazle baten tamaina, kolorea, konekzioak... eta kontrol edo adierazle berri bat bezala gorde.

Era ezberdinak daude "control editor" leihora ailegatzeko:

- Kontrola edo adierazlearen gainean ezkerreko botoia sakatuz eta advancedcustomize aukeratuz.
- Panel fontaleko kontol edo adierazle bat aukeratuz, edit-customize control aukeratuz.
- File-new bidez, custom control aukeratu.

"Control editor"-ak lan egiteko aukera bi ematen ditu: "edit mode" eta "customize mode". "Edit mode" erabili daiteke kontrol edo adierazle baten tamaina edo kolorea aldatzeko. Ordea, "customize mode" kontrol edo adierazlean aldaketa handiak egiteko erabiltzen da, kontrol eta adierazlearen atal guztiak indibidualki aldatu daitezkeelako.

🔁 Control 5 Control *	
Elle Edit View Project Operate Tools Window Help	
🖉 Control 🛛 🔽 13pt Application Font 🔽 🏪 🏧 🕮 🧉	<u>> 3 * ></u>
	<u>^</u>
Numeric	
	~
\$	>

Custom kontrol bat sortu ondoren, gorde daiteke beste edozein aplikazio baterako erabiltzeko. Hasiera batean kontrolak .*ctl* moduan gordetzen dira.

Gai honi dagozkion zenbait adibide jarraian adierazitakoak dira:

Graph_1.vi:

Simulazioko sistema bat (elementu birtuala) instrumentu erreal batera lagintzeko esaten digun datuen eskuraketa sistema bat daukagu, hau da, txartel errealera DAQ-aren bitartez pasa daiteke.

Laginketaren maiztasuna PC-agatik zehaztuta dago.



For egiturak "autoindexamiento" lanabesa dauka. Lanabesa honek iterazio guztien datuak puntuz puntu gordetzen ditu, hau da, iterazio bakoitzari sortzen den balioa elkartzen dio eta hau lortuta, pixkanaka matrize bat sortzen joaten da. **For egitura** defektuz egokitua dago, egitura hau mugatua delako (gauzatuko diren iterazio kopurua badakigulako). **While egituran**, aldiz, hasieran defektuz ez dago egokitu egongo, iterazio kopurua ezin baita jakin. Beraz, egokitu beharko da.

Graph batean jatorria eta desplazamendua aldatu daiteke. Gainera, graph cluster batekin ere alda genezake, blokeen bidez aldaketa egin dezakegulako. Chart batean, aldiz, ezin izango da hau egin, puntuz puntu aldatzea beharrezkoa baita.

🥵 Graph_1.vi Block Diagram *	\mathbf{X}
File Edit View Project Operate Tools Window Help Image: Comparison of the text of te	
V ^o Iteraciones	*

Graph_2.vi:

Graph bakar batean grafiko bat baino gehiago errepresentatu nahi baldin badira, sarrerak matrizeak direnez, **build array** funtzioa erabiliko genuke. Kasu honetan bi matrize sartu eta bi dimentsioko bat ateratzen da.





Graph_3.vi:

Adibide honetan aurretik adierazitako gauza bera azpimarratuko genuke.





Graph 4.vi:

Gai honetan zehar adierazitako adibide guztien artetik, adibide hau osotasun gehiena duena da.

Desberdintasun txiki bat dago: hemen matrize bat eta zatiketa bat dauzkagu eta dimentsio bakarreko matrize bat jatorrizko puntuan adierazi behar da.

Gainera, **build array** funtzio bat dago, non muga baten barnean sartuta, nahi ditugun sarrera guztiak ipintzeko aukera izango dugu. Grafiko bat baino gehiago izatean, **build array** nahi ditugun sarrerekin gehituko da.





Xo jatorria eta Dx gehikuntzarekin errepresentatu beharko den dimentsio bakarreko matrize bat daukagu Honez aparte, "cluster de array" bat dago eta honen atzetik "array de clusters" jarraitzen dio.

Graph grafiko motak parametrizatzeko, aurreko panelean kokatutako propietateak (properties) erabiliko dira. Aukera honen bidez, grafikoaren forma, kolorea... aldatu ahal izango ditugu; dena den, hauen artean oso garrantzitsua den kontzeptu bezala, kurtsoreak aurkitzen dira.

Intensity Graph Example.vi:

Ariketa hau intentsitatezko grafiko batez (koloreen grafikoa) osatuta dago. Mota honetako grafikoak zenbakizko balio batentzako koordenatu batzuk zehaztu beharrean, zenbakizko balio bati kolore bat dagokio. Horrexegatik, mapa koloreztatuei dagozkien grafikoak dira; parametro ezberdinen funtzio (tenperatura...), kolore ezberdin bat daukagu.

🛃 Int	ensity Gra	ph Exam	ple.vi Fr	ont Pane	1*							
<u>Eile E</u>	dit ⊻iew <u>P</u> r	oject <u>O</u> pe	erate <u>T</u> ool:	s <u>W</u> indow	Help							
	수 & (3pt Applicati	on Font			\$ 7 -					2 IHTEHSTY
	User Data				Intensity Graph							^
- <u>^</u>) o	() 10,00	(4) 6,00	(4) 7,00	2,00	100-	~		***		•	-2,0	
$=\left(\frac{\lambda}{\tau}\right)$	3,00	7,00	4,00	(-) 10,00	80 -							
Ť	÷1,00	5,00	() 10,00	0,00	ک ۵۰ ۵۰ -	86	88					
	7,00	3,00	(-)1,00	5,00	e uen 🕻	900	12.2	28 H				
	(0,00	€)0,00	(-)0,00	0,00	은 40-	005		11	7773		8	
	0,00	0,00	0,00	0,00	20-		200	8 L.	e)	£		= =
					0-			84	H11	1	2.0	
	F	Plot			– ó	20	40 Tiro	60	вo	100		
		🔍 User [Data		Time	8	 [<u>8.88] کار</u>		→ ,⊕	1		
) Interf	erence Patt	ern	Frequency		<u>-</u> <u>-</u> <u>-</u> <u>-</u> <u>-</u> <u>-</u> <u>-</u> <u>-</u> <u>-</u> - <u>-</u> - - - -					
		\sim			Amplitude		LZ Z.ZZ					
					Amplicude							
L												

Grafiko mota hauetan, **propietatezko nodoak** erabiliak izaten dira. Hau da, objektuak kolorea... bezalako zenbait propietate dituzte, hauek programan zehar prozesua moztu gabe **"property nodes"** aukeraren bitartez aldatu daitekeelarik. Honela, aurreko paneleko intentsitatezko grafikoa klikeatuz, "create" aukera aterako da eta honen barnean "property nodes" izango dugu.

Adibide honi dagokion bloke-diagrama ondorengoa izango litzateke:



Intensity Graph 1.vi:

Oso arrunta izaten den for anidatua daukagu:



Informazio hau **zenbakizko graph batean** errepresentatuko dugu eta gero, informazio bera koloreen bidez ere adieraziko dugu. **Write Spreadsheet** batean ere gorde daiteke. Azkenengo hau datuen taula bat da eta honen bidez, datuak testu formatu batean gorde ditzakegu. Dena dela, ondorengo hau kontutan izan beharko da: **while egitura** baten barnean dagoenean ezin izango da gorde, egituraren iterazio bat egiten den bakoitzean fitxeroaren izena eskatuko duelako.



"Indexamiento" kontzeptua, barnean dagoen eskalare bati zein iterazio dagokion erlazionatzea izango da. Termino hau matrizea sortzeko erabiltzen da; matrizea jada sortuta baldin badago, **"indexamiento"** kontzeptua matrizea non dagoen jakiteko soilik balioko du.

While egitura exekutatzeari bukatzen duenean, "bloc de notas"-ean azkenengo iterazioaren datuak soilik izango ditut, hau da, fitxero honetan gordetzen den matrizea azkenengo iterazioari dagokiona da; aurreko iterazioei buzo ez da ezer agertuko.

Array Exercise.vi:

Adibide honek **Build Array** eta **Array Subset** funtzioen operatibitatea ikustearen helburua dauka.

Bi matrizeen sorketa daukagu.

Goiko aldean matrize bat sortzen da. Matrize honen elementu guztiak eskala faktore ("Scalling Factor") batez biderkatzen dira eta biderketa honen emaitza "Final Array_1" aldagaian azaltzen da.

🥵 Array Exercise.vi Front Panel *
File Edit View Project Operate Tools Window Help
Random Array_1
0,534 0,949 0,844 0,527 0,885 0,082 0,013 0,485 0,298 0,343
1D Digital Indicator Array
Scalling Factor s
Digital Control
Final Array_1
0 1,068 1,898 1,688 1,053 1,770 0,165 0,026 0,970 0,595 0,686
1D Digital Indicator Array
Start Subset_1 # of Elements_1
€2
Digital Control *Digital Control*
Subset Array 1
Random Array_2
0 0,658 0,537 0,507 0,769 0,644 0,429 0,629 0,255 0,891 0,825
0,000 0,007 0,007 0,0044 0,429 0,029 0,200 0,891 0,825

Array Subset funtzioa erabiltzen da. Funtzio honek matrize bat izanda, matrize horren zenbait elementu ematen digu; horretarako, zein zenbaki eta zenbat elementu ematen digun adierazi beharko dugu. Matrizearen elementu hauek indize batzuen bidez adierazten dira.

Select egitura ere dago, egia eta gezurra aukeren artean hautatzeko aukera ematen digularik. **Select** kasu honetan aurreko panelean lan egiten du nahiz eta gehienetan era konkurrente moduan agertzen den.

Build Array funtzioak bi matrize unidimentsionalen bidez, matrize bidimentsionala sortzen du.



Ariketa honi dagokion algoritmoa ondorengoa da:

- "Random Array_1" sortu.
- "Random Array_2" sortu.
- "Array_1"-en subset-a.
 S = egia bada ----- Subset_1 = A.
 Bestela "Random Array_1 = A.
- A eta A2-rekin matrize bidimentsionala sortu.

Array Pair Multiplier.vi:

Ariketa honetan **Decimate** kontzeptua daukat. Adibidez:

Array A: 0, 4, 3, 5, -1, 7, 6, 8 0, 3, -1, 6 ------ Decimate1 funtzioa (bikoitiak) 4, 5, 7, 8 ------ Decimate2 funtzioa (bakoitiak) Array B: 0, 4, 3, 5, -1, 7, 6, 8, 10, -1, 14 0, 5, 6, -1 4, -1, 8, 14 ------ Hiruren Decimate funtzioa (hirunaka) 3, 7, 10

Hau egiten den zenbait aplikazio daude: tenperaturazko hiru tranduktore (T0, T1, T2),... Kasu honetan datu guztiak seriean bidaltzen ditut.





"Multiplied Array" aldagaian "Decimated Array_1" eta "Decimated Array_2" sarreren emaitzak azaltzen dira.

Die Roller.vi:

Dado bat errepresentatzen du. Dadoaren sei aurpegi daude eta jaurtiketak simulatu egiten dira. Beraz, dadoa zenbat aldiz jaurtitzen den kontzeptuarekin erlazionaturik dago.

🔛 Die Rol	ller.v	i Front	Panel					
<u>F</u> ile <u>E</u> dit	<u>V</u> iew	<u>P</u> roject	<u>O</u> perate	<u>T</u> ools	<u>W</u> indov	w <u>H</u> elp		2
ا	› 관		13pt Ap	plication	Font			•
								^
Numbe	er of ro	lls		Dice	Array			n
] 100	_		, 4	3,0	10			
					_			_
# 01 . ≙la	15	25	35	45	55	65	-	=
540			U					11
								Ŧ
I .								►:

Index Array funtzioa erabiltzen du, funtzio honek guztia zerora hasieratu dezagun esaten digu.



<u>R1.vi:</u>

Zenbaki oso, erreal eta konplexuez osaturiko matrizeak existitzeaz gain, **boolearrez** osaturiko matrize bat ere aurki daiteke. Panelezko instrumentazioan, matrize mota hauek berezi egiten dien zerbait izaten dute.

Instrumentu birtual bat sortzen den bakoitzean, defektuz aldagai bat jarri beharko da eta hasieratu.





Replace Array Subset funtzioa erabiltzen da. Hau desplazatzen ari den funtzioari dagokio; matrizeak eta desplazamendu erregistroak eta gero, **Case egitura** batekin osatu daiteke.

<u>R2.vi</u>:

Aurreko adibidearen matrize boolearra daukagu baina kasu honetan biratzen ari da etengabe.





<u>R3.vi:</u>

Ariketa osotasunez azaltzen da, aurreko biek zati bat soilik aurkeztu dute.

Kasu honetan ere matrize boolearra dugu eta modu ezberdinetan bira egitea esaten zaio: ezkerretik eskuinera, eskuinetik ezkerrera... Hala ere, behin soilik biratzen du eta berehala gelditzen da.



Hasiera eta bukaera lotuz, balio global, lokal...-en bidez desplazamendu erregistro bat egiten da.



Cluster Exercise.vi:

"Cluster Inicial" –ean zenbakizko balio bat, bi zenbaki boolear eta "Slide" bat daukagu. **Unbundle** funtzioaren bitartez, cluster honen zenbait elementu lortzen dira.

Beste alde batetik, "Boolean 1" eta kontroladore bezala lan egiten duen "Slide", "Cluster Reducido" deituriko beste cluster bat osatzen dute.

Unbundle By Name funtzioa, funtzio bat hartu eta inkrementatzen du.

Bundle funtzioa sarrerak bezala "Cluster Inicial" eta cluster honetatik hartu eta prozesatu diren beste bi elementu ditu.



Beraz, **Build Array** funtzioa gehitzen duen bitartean, **Bundle** funtzioa ordezkatu egiten du, hau da, hasieratzean ordezkatu egiten da, ez da gauza gehiagorik gehitzen.

Azkenean,"Cluster Modificado" lortzen dugu, ordena errespetatuz noski. **Bundle** edo **Unbundle** funtzioen ordena aldatzen bada, zuzenean detektatuko luke.



"Cluster Inicial" aldatzean, "Cluster Reducido" ere aldaketak izango lituzke, horrexegatik, batek maestro eta besteak esklabu bezala konportatuko dira.

Cluster Scaling.vi:

Konstanteen bidez osaturiko matrizeak existitzen diren bezala, **cluster konstantea** ere aurki dezakegu. Cluster hau, bere izenak dioen bezala, konstanteen osatuta egongo da eta beste cluster motak bezala lan egingo du, ez da inolako aldaketarik suposatuko.

😫 Cluster Scaling.vi I	Front Panel *						
<u>File E</u> dit <u>V</u> iew <u>P</u> roject	<u>O</u> perate <u>T</u> ools <u>W</u> i	ndow <u>H</u> elp					
수 & 🖲 🗉	13pt Application For	nt 🖃 🏪 🎰	· 🔤 - 🔅 - 👔				
Raw Data (Datos crudos) Datos Escalados							
V (Temperatura)	temperature (deg C)	flow rate (liters/sec)	pressure (N/m^2)				
4,00	500	100-					
	400	80 -	40000 60000				
V (Flujo)	300	60 -	20000	-			
15,00	200	40 -	-20000				
V (Presión)	100-	20 -	100000				
A 8.00	0-	0-					
W (0)00	200,40	153,00	80000,0C				
STOP							
				- 1 .131			

Adibide honetan zehar bloke-diagramako programazioa errazteko, cluster-en erabilera ikusten da.



Bi cluster sortzen ditut: hasierako balioez eta konstanteen bidez osaturikoak. Honela, sortutako bi cluster-en arteko biderketa lortzea izango da ariketa honen helburua.

<u>C1.vi:</u>

Bundle funtzioaren bitartez, cluster bat lortzen da. Cluster batek mota ezberdinetako objektuak onartzen ditu baina bere barnean ezin izango dira kontroladoreak eta adierazleak nahastu.

😫 C1.vi Front Panel *		
File Edit View Project Operate 다 중 중 🔘 💵 13pt Ap	Tools Window Help	
stop Nuevo Comando STOP batuketa Nueva Funcion F2 Warning	Indicador Funciones 3	output cluster Nuevo Comando batuketa Nueva Funcion 3 Warning
•	III	······································

String-a ASCII formatuan testua onartzen du. ASCII formatuan letra bakoitza 8 bitez osatutako dago.

🔁 C1.vi Block Diagram *	. 🗆 🔀
File Edit View Project Operate Tools Window Help Image: Second	,
Nuevo Comando Bundle output cluster Nueva Funcion UICP Indicador Funciones Warning TEP Stop TEP Warning	E
< III	

<u>C2.vi:</u>

Bloke-diagraman cluster-aren elementu baten aldaketa egiten da.

"Input Cluster" eta N1_1 sarrerek "Output Cluster" irteera osatzen dute.



Bundle funtzioak ordezkatu egiten du eta sarrerako elementuren bat aldatzearen aukera ere uzten du, inplementatuz eta aldatutako balio berria irteeran agertuz.



<u>C3.vi:</u>

Ariketa honetan konkretuki datuen oinarriak (bases de datos) sortzeko, **cluster** bat erabiltzen ari da.

🔁 C3.vi Front Panel 📃 🗖	
Eile Edit View Project Operate Tools Window Help	
🗘 🐵 🔲 🚹 13pt Application Font 🖃 🚛 🖬 🕮	2
Cluster Erabilera	-
Izena	
oihane	
Adina	
21	
Nº 55	
16097995	
	= =
Langue	
Ez Bai	



Cluster Error 1.vi:

Mota askotako erroreak egon daitezke eta errore guzti hauek cluster batean agertzen dira. Cluster-a errorea ematen duenean, gelditu eta testuzko ohar bat azalduko da errorearen kodearekin.

🔁 Cluster Error_1.	vi Front Panel	
File Edit View Projec	t Operate Tools Window H	
error code (0)		
4	error in (no error)	error out
		status code
	source	source
		Cluster Error_1.vi
error code (0) 2		
÷)8		•
1	111	•

Unbundle By Name funtzioaren **status** (egoera) aukera, ondorengo funtzionamendua izango luke: errorerik ez baldin badago, ondo funtzionatuko du; bestela, geldituko da, stop baten moduan.



Cluster 1.vi:

Adibide honetan cluster eta matrizearen arteko erlazioa ageri da.

VI honi hasiera emateko, 5 elementuz osatutako cluster-a egongo da, hauen artean, bi matrize, bi zenbakizko balio analogiko eta balio boolear digital bakarra aurkitzen dira.

"Output Cluster" A1 eta A2 matrizeez osatuta dago. Cluster mota hau sarrerakoaren berdina da, sarrerako cluster-a aldatzen bada, irteerakoa ere aldaketa sufrituko luke.



Array To Cluster funtzioarekin "Cluster" sarreraren elementuak ateratzeko aukera egongo da. **Cluster To Array** funtzioaren bitartez, aldiz, N1 eta N2 zenbakizko balioak matrize batean bihurtuko dira; N1 edo N2 aldatzen baldin badira, matrize baten posizioan sartzen joango dira. Beraz, bi funtzio hauen arteko ezberdintasuna, informazio bera baina modu ezberdinean errepresentatzen dutela da.



Matrize bat ezin izango da pieza ezberdinetan zatitu; cluster batek, aldiz, **Unbundle** funtzio batera konektatuz lortuko luke. Hala ere, matrize batean **Index Array** funtzioa dago, lortu nahi duzun balioa adierazten diozularik.

6.GAIA:NEURKETA DATUEN GORDEKETA

Gai honi dagozkion atalak ondorengoak dira:

- A. I/O artxiboak.
- B. Maila altuko I/O artxiboak.
- C. Maila baxuko I/O artxiboak.

A. <u>I/O artxiboak:</u>

I/O artxiboak, datuak artxibo batean gorde edo irakurtzen ditu.

I/O artxiboen ekintza tipiko bat ondorengo pausuak jarraitzen ditu:

- 1. Artxibo bat sortu edo ireki. Adierazi non dagoen existitzen den artxiboa edo non gorde nahi den sortuko den artxibo berria. Artxiboa irekitzen denean, erreferentziazko zenbaki bat adierazten du artxiboa.
- 2. Irakurri edo idatzi artxiboan.
- 3. Itxi artxiboa.

Artxibo formatuak

Labview erabili edo sortu ditzake ondorengo artxibo formatoak: Binarioak, ASCII, LVM, eta TDM.

- Binarioak: artxibo binarioak, beste artxibo formatuen oinarria da.
- ASCII: ASCII artxiboa, artxibo binarioen mota bat da, oso erabilia programa gehienengatik. ASCII kodean oinarritzen da. Artxibo hauek, testu artxiboak bezala ere ezagutzen dira.
- LVM: Labview-ren datu neurketa artxiboa (*.lvm*). Testu artxibo bat da kalkulu orriarekin edo testu editore batekin ireki daitekeena. Artxibo mota honek, datuen beste espezifikazio batzuk ere gordetzen ditu, data eta ordua adibidez. Mota hau, ASCII artxibo motaren aldaera bat da Labview-rentzat sortua.
- TDM: Artxibo mota honek, artxibo binarioen aldaera bat da National Instrument produktuetarako sortua. Gaur egun bi artxibo ezberdinez osatuta dago: XML sekzio bat datuen informazioarekin, eta artxibo binario bat grafikoetarako.

Labview datu direktorioa

"Labview data" direktorioa erabili daiteke, Labview-k sortutako datu artxiboak gordetzeko, adibidez *.lmv* edo *.txt* artxiboak. Labview-k "Lavbiew data" direktorioa instalatu egiten du instalakuntza orokorra egiterakoan, Labview-k sortutako datu artxiboak erosotasunez gorde eta aurkitzeko.

B. <u>Maila altuko I/O artxiboak:</u>

I/O artxiboko Instrumentu birtual (VI) batzuk, I/O artxibo baten prozezuaren hiru pausuak jarraitzen dituzte: ireki, irakurri/idatzi, eta itxi. VI batek hiru pausuak jarraitzen baditu, honek maila altuko VI bat izango da. Artxibo baten loop batean idazteko, maila baxuko I/O artxiboko VI bat erabili. Ordea, eragiketa bakarreko artxibo batean idazteko, hobe maila altuko I/O artxiboko VI bat erabilizea.

Maila altuko I/O artxiboko VI-ak ondorengoa dauka:

• Kalkulu orri artxibo batean idazteko: dimentsio bateko edo biko array bat, zenbakiak baino ez dituenak, testu string batean bihurtzen du, eta string-a idatzi egiten du ASCII artxibo berri batean edo jadanik existitzen den artxibo batean. VI-ak artxiboa ireki edo sortzen du bertan idatzi baino leen, eta ondoren ixten

du. VI hau erabili daiteke ulergarriak diren testu artxiboak sortzeko, kalkulu orrien aplikazio desberdinak erabiliz.

- Kalkulu orri artxibo batetik irakurtzeko: irakurri egiten ditu testu artxibo numeriko baten lerro edo zutabe zenbaki zehatz bat, karaktere partikular batetik hasita, eta datuak bi dimentsioko zenbakizko array batean bihurtzen ditu. VI-ak artxiboa irekitzen du irakurri baino lehen eta, ondoren ixten du. VI hau erabili daiteke testu formatuan gordetako kalkulu orri artxibo bat irakurtzeko.
- Neurketa artxibo batean idazteko: VI Expres bat datuak, testuan oinarritutako neurketa artxibo batean (*.lvm*) edo binarioa den neurketa artxibo batean (*.lvm*), idazten dituenak. Gordeketa metodoa aukeratu daiteke.
- Neurketa artxibo batekik irakurtzeko: VI Expres bat datuak, testuan oinarritutako neurketa artxibo batetik (*.lvm*) edo binarioa den neurketa artxibo batetik (*.lvm*), irakurtzen dituenak. Zehaztu daiteke artxiboaren izena, artxiboaren formatua eta segmentuaren tamaina.

C. <u>Maila baxuko I/O artxiboak:</u>

I/O maila baxuko artxiboen VI eta funtzioak, I/O artxiboen prozezuaren atal bat baino ez dute burutzen. Adibidez, funtzio bat dago ASCII artxibo bat irekitzeko, beste bat ASCII artxibo bat irakurtzeko, eta beste bat ASCII artxibo bat ixteko. Maila baxuko funtzioak oso gomendagarriak dira I/O artxiboan loop-ekin lan egiten denean.

Maila baxuko funtzioen bidezko "disk streaming"

I/O artxibo funtzioen bidez ere, disk streaming eragiketak egin daitezke, memoria baliabideak gordetzen dituenak, baina funtzioa eta operazio sistemaren arteko iterakzio kopurua artxiboa ireki eta ixteko murriztuz. Disk streaming, idazketa eragiketak egiten diren bitartean, artxiboak irekita mantentzeko teknika da, adibidez, loop baten bitartean. "Path" kontrol edo konstante bat, "write to text file", "write to binary file", edo "write to spreedseheet file" funtzio batekin kableatzen bada, funtzioa edo VI-a exekutatzen den bakoitzean artxiboa ireki eta itxi behar da. VI-ak eragingarriagoak dira, artxibo berdina behin eta berriro ireki eta ixtea ekiditen bada.

Artxibo berdina ireki eta ixtea ekiditeko, loop-aren barneko artxiboari erreferentziazko zenbaki bat eman behar zaio. Artxibo bat irekitzean, Labview-k erreferentziazko zenbaki bat sortzen du, artxibo horretara lotuta. Irekitako artxibo bakoitzean burututako eragiketak, erreferentziazko zenbaki bat erabiltzen du objetu bakoitza identifikatzeko.

Ondorengo adibidean, ez da "disk streaming" erabiltzen, eta VI-ak artxiboa ireki eta itxi behar du loop-aren iterazio bakoitzean.



Bigarren adibide honetan "disk streaming" erabiltzen du, VI-ak operazio sistemarekin artxiboa ireki eta ixteko iterakzio kopurua txikitzeko. Horrela, artxiboa ireki egiten da loop-a hasi baino lehen eta itxi artxiboa amaitzean, beraz, bi artxibo eragiketa gorde egiten dira loop-aren iterakzio bakoitzeko.



Gai honi dagozkion zenbait adibide jarraian adierazitakoak dira:

<u>ST 1.vi:</u>

Ariketa honetan sarrerako zenbakizko balioa string-ean bihurtzen da. Zenbaki bat string-ean bihurtzeko era, **Format Into String formatua**rekin ezagutzen dugu.

Gainera, propietateen bidez konfiguratzen den Express funtzioa dago. Komandoak sortzeko, Express funtzio bat denez, ireki eta exekutatu beharko da.

🐱 ST_1.vi Front Panel *														
Eile	<u>E</u> dit	⊻iew	Project	<u>O</u> perate	<u>T</u> ools	<u>W</u> indo	w <u>H</u> e	lp				[V B	
	C	\$ ֎		13pt Ap	plicatior	n Font	ìt IT ╬┱ ा विष थिप 😵							
													^	
\mapsto														
	stop		a tensior	nes:										
	1010		-2 0 2				Result							
		-4		$\begin{pmatrix} 4 \\ -6 \end{pmatrix}$	9,22		-El valo 9,2200	r de la te)00 Voltic	ension e Ds	es de: -			Ξ	
		-87	0	~8										
		-10		10			j							
													v	
							1						► 13	

Testua sartzeko, Configure Built Text modua erabiltzen da.



<u>ST_1C.vi:</u>

Zenbakiak string-ean bihurtzeko, **Number To Fractional** funtzioa ere erabiltzen da. Hala ere, funtzio honen alderantzizkoa ere existitzen da, **Fract/Exp String To Number** izenekoa.



Beste alde batetik, funtzioak ere lotu daitezke, hau da, bata bestearen atzean jarri. Horretarako, desplazamendu erregistroak erabiltzen baditugu, aurreko balioak jarri ditzakegu.
<u>ST 2.vi:</u>

Adibide honetan, Scan From String funtzioa erabiltzen da.





<u>T8 P1.vi:</u>

Adibide honetan, Former Into File erreferentzia bezala hartzen den funtzioa da.

Kasu honetan informazioa **"hoja de cálculo" formatuan** gordetzeko modu ezberdinak ikusiko ditugu.

T8_P1.vi Front Panel			
<u>File E</u> dit <u>V</u> iew <u>P</u> roject <u>O</u> perate <u>T</u> ools	<u>W</u> indow <u>H</u> elp		
🖒 🐼 🛑 💵 13pt Applicatio	n Font 🖃 🚛 🕇	━━▾	<u></u>
		error out	· · · · ·
file path	Nº de Iteraciones	status code	
	(+) 20	source	
function (open:0)			
open 0			_
	sta	code source	
		0	~
			►i

VI honen egiturari dagokionez, zenbait zatitan bereiz daitezke: programa nagusia **While** edo For egituraren barnean dago, baina egitura hauetako batean sartu baino lehenago, hasieraketa egoten da; kasu honetan, fitxeroa ireki edo ixtea izango litzateke.

Hasieraketa egiteko, **Open Create Replace File** funtzioa erabiltzen da. Funtzio honek informazioa gordeko den eremua sortzen du. **While edo For egitura** bukatzen denean, fitxeroa ixten da. "Cluster de error" batek dituen hiru elementuak errorea zein den adierazten digun string adierazlean banatzen dira.



Fitxero batean gordetzeko, lehenengo sortu beharko da, bestela ez digu gordetzen utziko. Horrexegatik, fitxeroa existitzen bada, irekitzen da; bestela "fitxeroa aukeratu" aukerarekin sortuko da. **Open edo Create** hautatzen badugu, exekutatzen da eta sortu denez, azkenik gorde ahal izango da.

Hiru mota ezberdinetako informazioa biltegiratzen da. Balioen artean tabuladore bat sartzen da, elkarren artean banakatzeko, eta azkenengotik lehenengora itzultzen da berriro ere. Fitxeroan informazioa gordetzeari bukatzen denean, itxi eta errorea agertuko da, erroreren bat egotean soilik.

Spreadsheet Converter.vi:

Erabili egiten da elementuren bat beste batengatik aldatzeko. Ondorengo adibidean, tarte bakoitza aldatu egiten da koma batengatik.



Lehenengo, **Concatenate String** eragiketa erabiliz zenbaki zerrenda bat sortzen da, zenbakien artean tarte bat utziz eta, hiru zenbakiero lerroz aldatuz. Irteera **string indicator** batera konektatzen da, tab spreadsheet string izena izango duena.

Ondoren, concatenate string funtzioaren irteera, **Search And Replace String** eragiketaren sarrerara konektatzen da. Eragiketa honekin, sarreran jarritako string-ean aldaketa batzuk egiten ditu. Erabiltzaileak adierazten du zein de aldatu nahi duen elementua, eta irteeran agertuko da string berdina baina aldaketekin. Kasu honetan esaten zaio aldatzean zenbakien arteko zenbakia koma batengatik. Eta irteeran string bat agertuko da coma spreedsheet string deitutakoa, tarteen ordez komak izango dituena.

Spreadsheet Example.vi:

Adibide honetan, seinale batzuen adierazpena egiten da, grafikoa eta taula moduan.



Lehenengo pausua seinaleak sortzea da **Sine Pattern.vi** eta **Uniform White Noise.vi** eragiketen bidez. Ondoren, **Built Array** funtzioarekin sortutako hiru seinaleen balioak batu egiten ditu array bakarrean. Sortutako array hau **Waveform Graph** batean ikustarazten da, non hiru seinaleak grafiko bakarrean adieraziko dira.

Beste bide batetik, hiru seinaleekin sortutako array-a ondo agertzeko gero ikustarazten denean, beharrezkoa da aldatzea, aldatua gordetzen delako eta ez bada berriro aldatzen alderantziz agertuko litzateke. Beraz, **Transpose Array** funtzioarekin eraldatu egiten da array-a (zutabeak lerro bihurtzen dira).

Ondoren, array-a string formatura aldatu egiten da Number To Frictional String funtzioarekin, azkenik taula batean ikuztarazteko.

<u>T8_P2.vi</u>:

Adibide honetan maila altuko funtzio bat erabili egiten da datuak idazteko neurketa artxibo batean.



Write Labview Measurement File, Express VI bat da, maila altuko funtzio bat. Funtzio hau, beste express VI-ak bezala, propietateen bidez parametrizatu daiteke. Seinaleak eskuratzen ditu eta artxibo batean idazten ditu. Martxan jartzerakoan beharrezkoa da gordetzea *.lvmf* bezala, eta ondoren artxibo hori irekitzerakoa "block de notas" programarekin, datuak agertuko dira zutabeetan, kalkulu orri formatuan.

<u>T8 P2B.vi</u>:

Ariketa honetan, alde batetik bi seinale ikustarazten dira **Waveform Graph** batean, eta bestetik express VI baten bidez artxibo batean datuak idazten dira.



Grafikoan bi seinale adieraziko dira, senoidal bat eta eskuratutako bat. Bakoitzak anplitude ezberdina duenez, grafika berean adierazi ahal izateko neurketa bakoitzerako menu flotantea jarri beharko da, horrela eskala ezberdina duten seinaleak grafika beheran irudikatu ahal izango dira. Seinale ezberdinak chart batean irudikatzeko **Bundle** funtzioa erabiltzen da; eta graph berdinean seinale ezberdinak irudikatzeko **Built Array** funtzioa erabiltzen da. Kasu honetan seinaleak **Waveform Graph** batean irudikatu nahi direnez, **Built Array** bat jarri beharko da. **Built array** funtziora ailegatzen diren seinaleak gainera dimentsio ezberdinekoak dira, tenperatura eskuraketa unidimentsionala da, eta sinua bidimentsionala. Hau gertatzerakoan, **Built Array**-aren irteeran array bidimentsionala agertuko da.

Write Labview Measurement File, maila altuko funtzioa da eta datuak artxibo batean idazten ditu. Datu horiek kalkulu orri formatuan gordetzen dira.

T8 P2D.vi:

Oraingoan, Read From Measurement File maila altuko funtzioaren adibide bat daukagu.



Read From Measurement File funtzioan beharrezkoa da adieraztea zein artxibotik irakurri behar den **file name** sarreraren bidez. Irakurtzeko artxiboa binarioa edo *.lvmf* motatakoa izan daiteke. **Signals** irteeran seinaleak adieraziko dira **Waveform Graph** batean; eta deskription irteera string batean, non irakurri beharreko artxiboaren hasierako informazioa agertuko da.

More Spreedsheet.vi:

Adibide honetan **For loop** bi erabiliz bi dimentsioko array bat sortu egiten da, alde batetik artxibo batean gorde egiten dena, eta bestetik grafiko batean adierazten dena.



Write To Spreadsheet File funtzioarekin sortutako bi dimentsioko array-a artxibo batean idazten da, ondoren kalkulu orri formatoan gordeko dena. Concatenate String funtzioarekin waveform 1, 2, eta 3 ko string bat sortzen da haien artean tarte bat utziz. Ondoren, sortutako string hau artxibo batean gorde egiten da Write Characters To File funtzioarekin.

Azkenik, "block de notas" programarekin gordetako artxiboa irekitzen badugu, ondorengoa da ikusiko duguna:

577 🧾	- Bloc d	le notas			_		X
Archivo	Edición	Formato	Ver	Ayuda			
<pre>wavefc 0,162 0,264 0,200 0,200 0,200 0,200 0,202 0,242 0,394 0,465 0,378 0,378 0,378 0,378 0,422 0,527 0,265 0,037 0,711 0,424 0,836 0,173 0,173 0,150 0,150 0,150</pre>	orm 1 0,99 0,44 0,97 0,50 0,50 0,59 0,59 0,59 0,59 0,59 0,59	Wa 56 0, 42 0, 56 0, 74 0, 74 0, 76 0, 76 0, 78 0, 98 0, 98 0, 98 0, 98 0, 98 0, 98 0, 98 0, 93 0, 94 0, 93 0, 93 0, 93 0, 94 0, 95 0, 94 0, 95 0, 97 0, 93 0, 93 0, 94 0, 93 0, 93 0, 93 0, 93 0, 93 0, 93 0,	vef(611 683 612 103 041 981 722 208 981 722 208 199 722 208 199 722 208 199 458 819 458 692 957 553 957 553 9103 061 489 964 814	orm 2	waveform	3	
<						2	:

Hiru zutabe eta 100 lerro izango ditu, guk diseinatutakoaren arabera. Zutabe bakoitzaren goikaldean sortutako string-a agertzen da.

Irakurri 2.vi:

Ariketa honetan artxibo batetik informazioa irakurtzen da eta bi dimentsioko array batean bihurtzen da. Ondoren, array honekin hainbat ekintza egiten dira.



Read From Spreedsheet File funtzioarekin, artxibo bateko informazioa irakurri eta bi dimentsioko array batean bidurtzen du. Gughenmein **Waveform**-ean hasierako array hau irudikatu egiten da inolako aldaketarik egin gabe. **Number To Frictional String** funtzioarekin, zenbakiak string batean bihurtzen ditu, ondoren taula batean adierazi ahal izateko.

Transpose 2D Array funtzioarekin, bi dimentsioko array-a aldatu egiten da; eta **Index Array** funtzioarekin, array bat zatitu egiten da erabiltzaileak adierazitako subarray edo dimentsioan. Ondoren, **Built Array** funtzioarekin subarray-ak berriro elkartu egiten dira, azkenik, elkartutako array hau beste **Waveform** batean adierazteko.

Read Write.vi:

Adibide honetan elkartu egiten dira VI berdinean idaztearen eta irakurtzearen funtzioak, aurretik bakoitza indibidualki egin direnak. Elkartu egiten dira **Case estruktura** baten bidez.

e <u>E</u> c	lit ⊻iew	Project Q	perate <u>T</u> o	ols <u>W</u> inde	ow <u>H</u> elp								SPR
	中國		13pt Applica	ition Font			₩ ~ Ø-	7				2	SHE
			Table	LEER	Boolean	ESCRIBIR					Sina		
			Table	Sine	Noice	Cosine	T				Noice		
			0		0.18	1.00	^				Cooler		
			1	0.05	0,94	1.00	-	Waveform G	raph		Cosine		-
			2	0,10	0,99	1,00		1,0-					
			3	0,15	0,47	0,99			X =		1. 1/		
			4	0,20	-0,67	0,98		0.5-	Λ				
			5	0,24	0,41	0,97		-,-	(. Nil)	L II NORAL			
			6	0,29	0,49	0,96		de de	W HAN	N N N N N N	111		
			7	0,34	0,98	0,94		· 클 0,0-		lína INÉ III	r Mit I	1/	
			0	0,38	0,56	0,92		Æ	T DANK	WIIIK (A	инн		
			10	0,43	-0,90	0,90		-0.5-		MI I YI I 🕅	11/11/ V	W	
			11	0.51	0.32	0.86			Y PLAT	N 111 W	1111 8		
			12	0,56	0,07	0,83		02/22			11/1	1	
			13	0,60	0,08	0,80	T	-1,0-	20 4		100	120 140	
					1	1	r	•	20 1	0 00 00	100	120 110	
								au da a mari a 1		subarray 2		subarray	3
							÷)0	subarrayi	() 0	-0.56	÷) o	1	
								0.049		0.309		0,999	
	Array			-	-		·	0,000		0.57		0,995	
0	0	0,049	0,098	0,147	0,195	0,243	0,29	0,098		-0.107		0,989	T
0	-0,56	0,309	0,57	-0,107	-0,704	-0,162	0,354	0,14/		-0,107		0,981	Ē
	1	0,999	0,995	0,989	0,981	0,97	0,957	0,195		-0,704		0.97	f
								0,243		-0,162		0.057	
								0,29		0,354		0,957	
								0,337		-0,851		0,942	
								0,383		0,193		0,924	
								0.428		-0,651		0,904	
								0.471		-0,995		0,882	
								pojiri		P		0,858	
												0,831	٦
												0,803	٦



Case estruktura erabili egiten da, non boolear kontrol batekin bi leiho irekitzen dira: True (idatzi) eta False (irakurri). Horrela kontrol boolearra aldatuz funtzio bat edo beste aukeratu ditzakegu.

Idatzi funtzioa jadanik aurretik azalduta dago, non **Write To Spreadheet File** funtzioarekin eratutako array-a gorde egiten da, eta "block de notas" programarekin irekitzerakoan informazioa kalkulu orri formatuan agertzen da.

rrrr 🧾	- Bloc de n	otas			×
Archivo	Edición For	mato	Ver	Ayuda	
0,000 0,049 0,049 0,145 0,243 0,290 0,383 0,428 0,471 0,514 0,556 0,634 0,672 0,707 0,741 0,773 0,831 0,858 0,882 0,904 0,957	0,182 0,945 0,945 0,470 -0,675 0,412 0,486 0,976 0,561 -0,959 -0,370 0,073 0,023 0,031 0,031 -0,110 -0,851 -0,564 -0,564 -0,564 -0,564 -0,564 -0,564 -0,564 -0,634		000 999 999 999 999 999 999 999 999 999		

Irakurri funtzioa ere aurretik eginda dago:



Read From Spreedsheet File funtzioarekin artxibo bateko informazioa irakurtzen da.

<u>EF_1.vi:</u>



Adibide honetan lehenengo **Open/Create/Replace File** funtzioa daukagu, artxibo bat ireki, sortu edo jadanik existitzen den bat ordezkatu egiten duena. Ondoren, txartelean dagoen tenperatura sentsoretik eskuratutako datuak **Thermometer.vi** funtzioarekin, **Number To Decimal String** funtzioarekin string bihurtzen dira, eta **Write To Text File** funtzioarekin artxibo batean idazten dira. Azkenik, artxiboa ixten da **Close File** funtzioaren bidez.

Dena **While loop** baten barnean dago, amaituko dena erabiltzailea stop botoia sakatzerakoan edo erroreren bat gertatzerakoan. Loop-a bukatzerakoan, erroreren bat gertatu bada, ikustarazi egingo da.

<u>EF_2.vi</u>:



Adibide hau aurrekoaren berdina da, baina **while loop**-aren kokapena ezberdina da. Oraingoan, artxiboa irekitzen da (edo sortu, edo ordezkatu) **Open/Create/Replace File** funtzioarekin. Ondoren **while loop**-aren barnean artxiboan tenperatura datuak gordetzen egongo dira, erabiltzaileak gelditu arte edo errore bat gertatu arte. **While loop**-a amaitzen denean, artxiboa itxiko da, eta erroreak ikustaraziko dira, egotekotan.

Temperature Log.vi:

Adibide honetan txarteletik tenperatura eskuratu egiten da, tenperatura hau gorde eta grafiko batean ikustarazten da.



Lehenengoz artxibo bat ireki, sortu edo ordezkatu egiten da **Open/Create/Replace File** funtzioarekin. Ondoren, txarteletik tenperatura neurketak egiten dira. Alde batetik tenperatura hau string batean bihurtzen da **Number To Frictional String** funtzioarekin eta artxibo batean idazten **Write To Text File** funtzioarekin.

Beste alde batetik, **While loop**-a eta desplazamendu erregistroaren bidez, oraingo tenperatura eta aurreko neurketen tenperatura izanik, posiblea da batez besteko tenperatura adieraztea. Batzen badira oraingo tenperatura, eta desplazamendu erregistroatik lortutako aurreko tenperaturak, eta ondoren zatitzen badira 3-rekin (kasu honetan aurreko bi balio eta oraingoko balioa dauzkagulako), orduan batez besteko balioa izango dugu. **Waveform Chart** bakarrean oraingoko tenperatura eta batez bestekoa adierazteko, **Bundle** funtzioa erabili behar da bi grafikoak elkartzeko grafiko bakarrean.

Erabiltzaileak power botoia eragitean edo errore bat gertatzekotan, **While loop-**a geldituko da, artxiboa itxiko da, eta egotekotan, erroreak ikustaraziko dira.

Inicio.vi:

Adibide honetan **Property Node** funtzioa lantzen da.



Lehenengo kontrol bat jartzen da, numerikoa (knob), eta kableatu egiten da adierazle batekin, ere numerikoa (meter). Ondoren, bakoitzari **Property Node** bat sortuko zaio (objetuaren gainean, ezkerreko botoiaz →create→property node). Honen bidez edozein objetuaren ezaugarriak aldatu daitezke: egin dezakegu desagertzea, itxarotea aldaketaren bat...

Kasu honetan knob kontrolak hiru ezaugarri ditu: **visible** (desagertzea egin dezakeena), **disabled** (zegun zenbakia erabilgarria izango da edo ez: 0-eskuragarria da, 1-ez da eskuragarria, 2-ez da eskuragarria eta iluntzen da), **keyfocus** (posiblea da balioa aldatzea teklatuarekin). Meter adierazleak, ordea, soilik ezaugarri bat dauka, visible.

Property Node Exercise.vi:

Oraingoan ere Property Node erabilpena erakusten da.



Random Number bat, zenbaki aleatorioak sortzen dituena, biderkatu egiten da 10 zenbakiarekin, eta erabiltzaileak finkatutako zenbaki limite batekin konparatzen da. **Select** funtzioarekin, segun zenbakia limitea baino handiagoa edo txikiagoa den, tankearen kolorea aukeratzeko. Honetarako, **Property Node** bat eratzen da, **fill color** deitutakoa, tankearen kolorea aldatzeko lehen aipatutako moduaz, segun zenbakia limitea baino handiagoa edo txikiagoa den.

Bestalde, tankearen posizio bertikala eta horizontala ere aldatu daiteke. Horretarako, Horizontal Pointer Slide eta Vertical Pointer Slide kontrol bi jartzen dira, Bundle

batekin elkartuko direnak, eta **Position** Property Node-ra konektatuz tankea bertikalean eta horizontalean mugitu ahal izango da.

Eta azkenik, disable boolear kontrolaren balioaren arabera, tankea eta beste kontrol guztiak iluntzea posiblea da, **disabled** Property Nodearen bidez.

Temperature limit 1.vi:

Ariketa honetan, tenperatura neurketa bat egiten da, ikustarazten da grafiko batean, eta tenperatura limite batekin konparatu egiten da.



Lehenengo **Secuence** estrukturan, grafikoaren bi **Property Node** definitzen dira. Alde batetik, **Scale Multipler**, x eskalaren balioak finkatzen dituena; eta beste alde batetik **histoy**, VI-a amaitzerakoan grafikoa garbitzen duena, horrela berriro hasieratzerakoan 0-tik hasiko da irudikatzen, ez du aurretik egindako neurketaren grafikoa gordetzen. Gainera, ordua eta data adierazteko string bat sortzen da. Honetarako, **Get Date/Time String** funtzioa erabiltzen da (seconds boolear sarrerarekin ordua adierazterakoan segunduak adierazten diren edo ez kontrolatzen da). **Format Into String** funtzioarekin lortutako ordu eta data datuak string formatura aldatzen dira, mezua adierazteko modua adierazte.

While loop-ean, tenperatura neurketa bat egiten da txarteletik. Tenperatura hau erabiltzaileak ezarritako limite batekin konparatzen da. Bi tenperatura hauek grafiko batean irudikatzen dira, beraz, beharrezkoa da lehenengo **Bundle** funtzioaren bidez elkartzea. Tenperatura, limitea baino handiagoa bada boolear batez adieraziko da out of range deitua. Boolear honi property node bat egiten zaio, piztekotan ilunagoa agertzeko, **blinding**.

While loop-aren barnean **case** estruktura bat daukagu. Tenperatura maximoa baino handiagoa bada, seinalea gorria izango da, eta soinu batez adieraziko da. Kolore aldaketa egiteko grafikoari beste property node bat jartzen zaio, **plot color** deitua. Eta tenperatura maximoa baino txikiagoa bada, ez da soinurik egongo, eta seinalearen kolorea horia izango da, oraingoan ere plot color propietatea erabiliz.

Azkenik, while loop-a bukatu egingo da edo "abort" botoia sakatuz, edo loop kopurua erabiltzaileak finkatutako eskuratzeko puntuak, berdintzen direnean.

<u>E2_2.vi</u>:

Ariketa honetan **Secuence** estruktura bat daukagu **While loop** baten barnean.



Boolear kontrola true dagoen bitartean **While loop**-a funtzionatzen jarraituko du. **Secuence** estruktura bat daukagu, lau sekuentzia dituenak: test 1, test 2, test 3, eta amaituta. **Secuence** bakoitzaren barnean testu konstante bat dago, eta kanpoan string adierazle bat. Tenporizazio bat dago egingo duena sekuentzia bakoitzaren iraupena 5 segundukoa izatea. Azkenengo sekuentziara heltzerakoan, ez bada boolearraren egoera aldatzen, 5 segundu igaro ostean, berriro lehenengo sekuentziara joango da.

Orduan, **Secuence** estrukturarekin orden bat finkatzen da, eta **While** bukleak erritmoa markatzen du. Baina dagoen arazoa da, derrigorrez sekuentzia batetik bestera joan behar dela, ezin da nahi den beste orden bat jarraitu. Beste orderen bat jarraitzeko, programa gehiago jarri behar dira aldi berean lan egiten, eta aldagai lokalen bidez elkartu.

Time to match.vi:

Adibide honetan, sistemaren erlojua harturik, zenbaki bat aurkitzeko denbora kalkulatzen du.





Secuence estruktura bat daukagu, lehenengo zenbaki bat aurkitzen du, eta gero zenbaki hori aurkitzeko pasatu den denbora kalkulatzen da. Lehenengo sekuentziaren barnean While loop bat daukagu, biraka egongo dena erabiltzaileak finkatutako zenbakia aurkitu arte. Horretarako, **Random Number** bat dago zenbaki aleatorioak sortzen dituenak. Zenbaki hauek, banan-banan konparatu egiten dira erabiltzailearen zenbakiarekin, berdina aurkitu arte. Momentu horretan hurrengo sekuentziara pasatuko da, non konparatuko dira hasierako erlojua eta oraingokoa, tardatutako denbora kalkulatzeko.

7. GAIA: APLIKAZIO MODULARREN GARAPENA

Gai honi dagozkion atalak ondorengoak dira:

- A. Modulazioa ulertuz.
- B. Ikono eta konektore orria eraikiz.
- C. SubVI-en erabilpena.

A. Modulazioa ulertuz:

Modulazioa definitu egiten du nola programa bat modulu diskretuetan banaturik dagoen, horrela modulu batean aldaketak beste moduluetan eragina ia nulua da. Moduluak labview-n, subVI-ak deitzen dira.

VI bat beste VI batekin, subVI-a deitzen da. SubVI bat subrutina baten antzekoa da testuan oinarritutako programazio lengoaietan. SubVI bat klikeatzean, panel frontal bat eta bloke-diagrama bat agertuko da, non aukera ezberdinak konfiguratu daitezke. Panel frontala kontrolak eta adierazleak ditu; eta bloke-diagraman, konekzioak, panel frontaleko ikonoak, funtzioak, subVI-ak, eta beste edozein VI batean egongo diren objektuak egongo dira.

SubVI-ak erabiliko dira funtzio baterako operazio berdina errepikatzen denerako. Horrela, ez da beharrezkoa behin eta berriro berdina errepikatzea, subVI-a jartzearekin nahikoa izango da.

B. Ikono eta konektore orria eraikiz:

VI-aren panel frontala eta bloke-diagrama eratu ostean, ikono eta konektore orria eratzen da, horrela VI-a subVI-a bezala erabili daiteke. Ikono eta konektore orria, testuan oinarritutako programazio lengoaiekin konparatuz, prototipo funtzioaren berdina da. VI guztiek ikono bat daukate bloke-diagrama eta panel frontalaren artean; honelako itxurakoa:

l	AAB
l	to the
L	V 1

Ikono bat, VI baten adierazpen grafikoa da. Testua, irudiak, edo bien arteko konbinazioa euki ditzake. VI bat subVI modura erabiltzen bada, ikonoak, subVI-a identifikatuko du VI-aren bloke-diagraman. Ikonoan klikeatuz, subVI-an aldaketak egin daitezke.

Konektore orri bat eraiki behar da VI bat subVI bat bezala erabiltzeko. Honelakoa izaten da:

H			Н
H	Н	H	Н
F			Н

Konektore orria, terminal multzo bat da, VI-aren kontrol eta adierazleei dagokiena. Konektore orriak, VI-arekin konektatu daitezkeen sarrerak eta irteerak definitzen ditu, subVI bat bezala erabiltzeko. Konektore orriak informazioa bere sarrera terminalen bidez jasotzen du; datuak bloke-diagramara panel frontaleko kontrolen bidez pasatzen ditu; eta panel frontaleko adierazleen bidez, bere irteera terminaletan emaitzak jasotzen ditu.

Ikono bat sortuz

Defekuzko ikonoa, zenbaki baten bidez adierazten du zenbat VI berri sortu diren, Labview-a zabaldu denetik. Beste ikono berri bat sortzeko defektuzko bat ordezkatzeko, ikonoan ezkerreko botoiarekin "edit icon" aukeratu. "Icon editor" leihatilarekin, ikonoa sortu, kolorea, tamaina...aldatu daiteke.



"Edit" menuaren bidez, irudiak moztu, kopiatu edo itsatsi daitzezke ikonoan. Ikonoaren atal bat aukeratzen badugu eta irudi bat itsatsi, Labview-k irudiaren tamaina eraldatzen du auketatutako atalean sartzeko. "Copy from" aukerarekin, irudia aldatu daiteke koloretik zuri-beltzera, edo alderantziz. Ikonoa guztiz diseinatu ostean, OK botoia sakatu.

Ondorengo lanak burutu daitezke "icon editor" barnean:

Color copy tool": Ikonoaren elementu baten kolorea kopiatzeko.

"Pencil tool": Pixelka marraztu edo ezabatzeko.

"Rectangle tool": errektangelu bat marrazteko ikonoan. Birritan klikeatu, ikono osoa mugatzeko.

"Line tool": Lerro zuzenak marrazteko. Lerro bertikalak, horizontalak eta diagonalak marrazteko, "shift" tekla sakatu tresna hau erabiltzen den bitartean kurtsorearen posizioa aldatzeko.

Diffill tool": Ikonoaren eremu bat aurreko planoko kolorearekin betetzeko.

"Filled rectangle tool": Errektangelu bat marrazteko aurreko planoko kolorearekin eta, atzeko planoko kolorearekin betetzeko. Birritan klikeatu, ikono osoa aurreko planoko kolorearekin mugatzeko eta, atzeko planoko kolorearekin betetzeko.

"Select tool": Ikonoaren eremu bat aukeratzeko, hura mozteko, kopiatzeko, mugitzeko, edo beste edozein aldaketa egiteko. Birritan klikeatu eta "delete" tekla sakatu ikono osoa ezabatzeko.

A "Text tool": Ikonoan testua sartzeko. Birritan klikeatu mota ezberdina aukeratzeko. "Small fonts" aukera oso erabilgarria da ikonoetan.

"Foreground/background tool": Atzeko edo aurreko planoko koloreak erakusteko. Aukeratu errektangelu bat kolore paleta bat agertzeko, non kolore berriak aukeratu daitezke. Goikaldeko errektangelua, aurreko planoko kolorea adierazten du; eta behekaldeko errektangelua, atzeko planoko kolorea.

Ezkerraldeko aukerak erabili ondorengo lanak burutzeko:

- "Show terminals": Konektore orriaren terminalaren eredua aurkezten du.
- "OK": Ikonoaren marrazketa gordetzen du, eta panel frontalera bueltatzen da.
- "Cancel": Panel frontalera bueltatzen da inolako aldaketak gorde gabe.

"Edit" menuan beste edizio aukera batzuk daude: moztu, kopiatu, itsatsi...

Konektore orria antolatzeko

Konektore orri bat definitzeko, panel frontaleko goiko ezkerreko aldean dagoen ikonoan ezkerreko botoia sakatu eta, "show connector" aukeratu, konektore orria ikuskarazteko. Konektore orria agertuko da ikonoaren ordez. Konektore orria lehenengo aldiz ikusten denean, konektore eredu bat agertzen da. Eredu hau aldatzeko, ezkerreko botoiaz konektore orria sakatu eta "patterns" aukeratu. Konekzio era asko agertuko dira eta diseinuaren arabera (sarrera eta irteera kopurua), bat edo beste aukeratu.



Konektore orriaren errektangelu bakoitza terminal bat adierazten du. Errektangeluak erabili sarrerak eta irteerak esleitzeko. Konektore orria normalean, terminal bat dauka panel frontaleko kontrol eta adierazle bakoitzeko. Aurreratzekotan aldaketak egongo direla VI-an sarrera edo irteera berriekin, terminal gehiago jarri konektatu gabe.

Ondorengo panel frontalean lau kontrol eta adierazle bat dago, beraz, konektore orrian lau sarrera terminal eta irteera terminal bat egongo dira.

🔛 ບ	ntitled 1	Front Par	nel *							צו
Eile	<u>E</u> dit <u>V</u> iew	Project	<u>O</u> perate	<u>T</u> ools <u>V</u>	<u>V</u> indow <u>H</u>	<u>H</u> elp			F	
	\$ €	2 🔍 💷	13pt Ap	plication F	ont 🛛 🔫		- -	5	2	
										^
		Num E 0 Num E 0	eric eric 3	28	Numeric 2 0 Numeric 4 0	2	Num	eric 5		
										~
<										> 13

Terminal ereduak aukeratu eta aldatu

Konektore orrian ezkerreko botoiaz "pattern" aukeratuz, nahi den terminal kopuruko eredua aukeratu daiteke, diseinuaren sarrera eta irteeren arabera. Terminal gehiagoko eredu bat aukeratu daiteke eta, soberako terminalak konektatu gabe utzi, horrela malgutasun handia dago VI-a aldatzeko hierarkia asko aldatu gabe. Ere egon daitezke kontrol edo adierazle gehiago terminalak baino.

Gehienez, 28 terminal egon daitezke konektore orri batean.

Gehien erabiltzen den eredua ondorengoa da, estandar bezala agertzen dena. Ezkerreko terminalak sarrerakoak dira, eta eskumakoak irteerakoak. Normalean goiko sarrera eta irteerak erreferentziak jartzeko erabiltzen dira, eta beheko sarrera eta irteerak erroreak konektatzeko.



Kontrol eta adierazleen terminalen esleipena

Konektore orriaren eredua aukeratu ostean, konekzioak definitu behar dira, panel frontaleko kontrol eta adierazle bakoitzari konektore orriaren terminal bati esleituz. Kontrola eta adierazleak konektore orriarekin konektatzerakoan, jarri ezkerraldean sarrerak eta eskuinaldean irteerak, akatsak ekiditeko. Terminalak eta panel frontaleko kontrola edo adierazlea konektatzeko, lehenengo klikeatu konektore orriaren terminal bat, eta ondoren terminal orri esleitu nahi zaion panel frontaleko kontrola edo adierazlea. Terminalaren kolorea konektatutako datu motaren arabera aldatuko da, konekzioa eginda dagoela adierazteko. Alderantziz ere egin daiteke, hau da, lehenengo kontrola edo adierazlea aukeratu eta ondoren terminala.

C. <u>SubVI-en erabilpena:</u>

SubVI bat bloke-diagraman jartzeko, funtzio paletan "select-a VI" aukeratu, eta dauden subVI-en artean erabili nahi dena aukeratu eta bloke-diagraman jarri.

Irekitako VI bat, beste irekitako VI baten bloke-diagraman jar daiteke. Horretarako, subVI bezala erabili nahi den panel frontalaren goikaldean eskumaldean dagoen ikonoa sakatu, eta beste VI-aren panel frontalera eraman.

Subvi-ak ireki eta eraldatu

VI batean dagoen SubVI baten panel frontala irekitzeko, bloke-diagraman, subVI-an birritan klikEatu. Ordea, subVI-aren bloke-diagrama irekitzeko, "ctrl" sakatu eta birritan klikatu, bloke-diagraman, subVI-a. Horrela, subVI-a eraldatu daiteke, eta gordetzerakoan, aldaketak subVI-a erabiltzen den aldiro eragina izango dute.

Beharrezkoak, gomendatutakoak eta aukerazko sarrera eta irteeren prestakuntza

"Context help" leihatilan, beharrezko terminalen izenak, beltztuta agertzen dira; gomendatutako terminalen izenak, testu normalarekin; eta aukerazko terminalen izenak, zirriborraturik. Aukerako terminalen izenak ez dira agertuko, "hide optional terminals and full path" botoia sakatu egiten bada "context help" lehiatilan.

ē

Adierazi daiteke sarrerak edo irteerak beharrezkoak, gomendatutakoak, edo aukerazkoak diren, erabiltzaileak ez ahazteko terminalak konektatzeaz. Honetarako, konektore orriaren terminal bat aukeratu, ezkerreko botoia sakatu eta, "this connection is" aukeratu. Ondoren, "required" (beharrezkoa), "recommended" (gomendatutakoa), edo "optional" (aukerazkoa) aukeratu.

Sarrera terminalentzako, beharrezkoa adierazi nahi du, subVI-a jarri den blokediagrama ezingo duela funtzionatu, beharrezko terminalak kableatuta ez badaude. Irteerako terminalak ezin dira beharrezkoak izan. Sarrera eta irteera terminalentzako, gomendatua edo aukerazkoa adierazi nahi du, subVI-a jarri den bloke-diagrama funtzionatu dezakeela, gomendatutako edo aukerazko terminalak kableatuta egon ez arren. Terminalan kableatzen ez badira, VI-ak ez ditu sortuko inolako oharrik.

VI-aren sarrerak eta irteerak *vi.lib*- ean jadanik badaude, beharrezkoak, gomendatutakoak edo aukerazkoak bezala gordeta. Labview-ak, VI baten sarrerak eta irteerak gomendatutakoak bezala definitzen ditu, hau aldatzeko aurretik azaldutako pausuak jarraitu behar dira.

Subvi bat sortu existitzen de VI batetik

VI baten bloke-diagrama sinplifikatu daiteke, bloke-diagramaren sekzio batzuk subVIetan bihurtuz. VI baten sekzio bat subVI batean bihurtzeko, lehenengo bloke diagramaren sekzioa aukeratu behar da, eta ondoren "edit"-"create subVI" aukeratu. SubVI berriaren ikono bat agertuko da bloke-diagramaren sekzioaren lekuan. Labviewk zuzenean, subVI berriaren kontrol eta adierazleak sorten ditu; konektore orria konfiguratzen du, sarrera eta irteera kopuruaren arabera; eta konekzioak egiten ditu VIaren beste objektuekin. SubVI berria, konektore orriaren eredu orokorra erabiltzen du, eta ikono orokorra. Hau aldatzeko, birritan klikeatu subVI-an eta nahi diren aldaketak egin.



Gai honi dagozkion zenbait adibide jarraian adierazitakoak dira:

Grafica.vi:

Bi array-en bidez, grafika bat sortuko da, eta exekutatzen egongo da fin botoia sakatu arte:





Sarrera biak, X eta F(X), bi array dira, elkartuko direnak **Bundle** funtzioarekin, ondoren biak adieraziak izateko **XY Graph** berdinean. Exekutatzerakoan beste pantaila batean grafikoa agertuko da, eta **fin** botoia sakatzerakoan pantaila berri hau itxiko da. Hau lortu egoten da **While loop**-aren bidez.

Sin & Cos.vi:

Ondorengo adibidean senoa edo kosenoa adierazten duen grafiko bat agertzen da:





Erabiltzaileak, goiko limitea, beheko limitea, eta puntu kopurua jarri beharko ditu. Ondoren **For buclearen** bidez grafikoaren puntuak adierazten dituen array bat sortuko du, autoindexamiento bidez. **Sine & Cosine** funtzioarekin, array horretako puntuekin, seinalearen sinua eta kosinua sortzen du.

Case estruktura batekin, boolear kontrol baten bidez, aukeratu ahal da true edo false, irteeran funtzio bat edo beste lortzeko. Horrela, boolearean coseno aukeratzekotan, **Case estruktura**ren false kasua aukeratuko da, eta hau **Sine & Cosine** funtzioaren kosenoarekin kableatuta dagoenez, grafikoan kosenoa irudikatuko da. Eta boolearrean sinua aukeratzekotan, **Case estruktura**ren true kasua aukeratuko da, **Sine & Cosine** funtzioaren funtzioaren sinuarekin kableatuta dagoena. Kasu honetan grafikoan sinua irudikatuko da.

XY grafiko batean irudikatzeko, beharrezkoa da y ardatzaren puntuen array-a, eta x ardatzaren puntuen array-a definitzea, beraz, **Bundle** funtzio bat jarri behar da bi array hauek elkartzeko.

Sin & Cos & Tag.VI:

Adibide hau oso interesgarria da, funtzio asko erabiltzen direlako, eta subVI-en erabilpena diagrama errazteko egiten da.

😰 sen&cos&tag.vi Front Panel *	_ 🗆 🗙
Eile Edit View Project Operate Tools Window Help	
🗘 🔲 13pt Application Font 🗐 🚛 🖬 🐨	8 4
	<u>^</u>
Seno Coseno Tangente	
SALIR	
View> VI Hierarchi	


Adibide honetan, senoa, kosenoa, eta tangentea egiteko aukera ematen duen funtzio bat daukagu, ondoren limiteak finkatzen dira, eta azkenik grafiko batean adierazten du.

Lehenengoz, hiru boolear adierazle bidez aukeraketa egiten da, sinu, kosenu edo tangente. Hiru boolear hauek **Compound Arithmetic** funtzioarekin elkartu egiten dira, horrela jakiteko zein den aukeraketa. **Compound arithmetic** funtzioaren irteera, **Case estruktura**ren aukeraketa sarrera izango da. Ez bada inolako grafikarik zehazten, **Case estruktura**ren false aukeratzen da, ez da ezer gertatuko, "salida" botoia sakatu arte geldi egongo da. Aukeraketa bat egiten bada, orduan **Case**-ren true aukeratzen da, eta hainbat eragiketa egingo dira.

Case estrukturaren barruan **Secuence estruktura** bat dago. **Secuence estruktura** hau hiru sekuentzia izango ditu: lehenengoan limiteak ezartzen dira; bigarrenean sinua, kosenua edo tangentea eraikiko da; hirugarrenean grafika batean irudikatuko da.

Secuence estrukturaren lehenengo sekuentzian, sinu, kosenu nahiz, tangente aukeratu, **limites subVI**-ra deituko zaio, grafikoetarako limiteak finkatzeko.

Bigarren sekuentzian, beste **Case estruktura** bat agertzen da. **Case estrukturaren** aukeraketa sarrera tangente aukera da. Tangentea aukeratu ez bada, **Case**-an false aukeratuko da eta, **Sen & Cos SubVI**-ra deituko zaio. Egindako aukeraketaren arabera, sinua edo kosenua, subVI-honek sinua edo kosenua emango du. Ordea, tangentea aukeratu bada, **Case**-ren true aukeratuko da, eta tangentea lortzeko funtzioa egingo da.

Tangentea lortzeko, bi **Sin & Cos SubVI** erabiltzen dira eta, haien arteko zatiketa eginez tangentea lortzen da.

Azkenengo sekuentzian, **grafica subVI**-ra deitzen zaio, aukeratutakoaren irudia egiteko.

Erabilitako limites subVI-a ondorengoko itxura dauka:

Edit View Proje	ect <u>O</u> perate	Loois Windo	w <u>H</u> elp					
\$	13pt A	pplication Font	- -	-0		¢,		?
1 for the Course day								
Limite Superior	-12,00							
	-2,00							
lúmero de Puntos	J10000							
	Acortas	7						
	Aceptar							
Límite Superior	2,00	7						
Límite Inferior	-2,00	Ī						
Número de Puntos	10000							
I limites.vi Bl Eile Edit ⊻iew	ock Diagra Project Op	m erate <u>T</u> ools	Window He	lp Applicatio	n Font			ه،]
Imites.vi Bl Eile Edit ⊻iew	ock Diagra Project Op	m erate Iools	Window He	lp Applicatio	n Font		v	<u>ښ۰</u>
Dimites.vi Bl	ock Diagra Project Op	m erate <u>T</u> ools	Window He	lp Applicatio	n Font	v t	· ·Ūm ·	<u>ڻ</u> ۔
Imites.viBl Elle Edit View \$	ock Diagra Project Op	m erate Iools	Window He I I 3pt /	lp Applicatio	n Font	v	• • • • •	<u></u>
D limites.vi Bl Elle Edit ⊻lew	ock Diagra Project Op	m erate <u>I</u> ools	Window Hel	lp Applicatio	n Font		•	⇔ -
D limites.vi Bl Elle Edit View	ock Diagra Project Op	m erate <u>I</u> ools 2 2 Jun T	Window Hei	lp Applicatio	n Font	.	v in i	<u>\$</u>
D limites.vi Bl Elle Edit ⊻lew \$	ock Diagra Project Op	m erate Iools	Window Hel	lp Applicatio	n Font	Límite Su	Perior	ه.]
D limites.vi Bl Elle Edit ⊻lew	ock Diagra Project Op	m erate Iools 2 2 4 4 6	Window Hei	lp Applicatio	n Font	Emite Su	Perior	⊕- -
D limites.vi Bl Elle Edit ⊻lew	ock Diagra Project Op	erate Iools	Window Hel	lp Applicatio	n Font	Límite Sc Þölt	perior	<u>ښا</u>
Ele Edit View	ock Diagra Project Op III	m erate Iools	Window Hel Carl 13pt / Superior	lp Applicatio	n Font	Emite So For the Solution of t	perior erior	¢-
Delimites.vi Bl File Edit View €	ock Diagra Project Op III	m erate Iools 2 2 4 5	Window Hel	lp Applicatio	n Font	Limite Su Limite Inite Limite Inite	perior erior	<u></u>
D limites.vi Bl Elle Edit ⊻lew	ock Diagra Project Op III	m erate Iools 2 2 4 C	Window Hei	lp Applicatio	in Font	Limite Su Limite Ini Imite Init	perior erior	⊕- -
Imites.vi Bl Elle Edit ⊻ew €	ock Diagra Project Op	m erate Iools 2 2 Im To Imite I I Imite I I Imite I	Window Hei Superior nferior a de Puntos	Ip Applicatio	n Font	Émite Su Emite Su POBL Númer Númer	perior o de Puntos	⇔ -

Gai honi dagozkion atalak ondorengoak dira:

- A. Hardware-aren erabilpena.
- B. Hardware-arekin komunikazioa.
- C. DAQ Device baten simulazioa.
- D. Sarrera analogikoaren neurketa.
- E. Irteera analogikoaren sortaraztea.
- F. Kontadoreen erabilpena.
- G. I/O digitalen erabilpena.

A. <u>Hardware-aren erabilpena:</u>

DAQ sistema tipiko bat, hiru harware mota ditu: bloke terminala, kablea, eta DAQ tresna. DAQ sistema baten pausuak ondorengoak dira:

- 1. Seinalea.
- 2. Terminal blokea.
- 3. Kablea.
- 4. DAQ trenza.
- 5. Ordenagailua.

Fenomeno fisiko bat neurtu daitekeen seinale batean bihurtu ostean, seinale izaera izan edo ez, seinale hori eskuratu behar da. Seinale bat eskuratzeko, terminal bloke bat, kablea, DAQ tresna, eta ordenagailu bat beharrezkoak dira. Hardware konbinazio hau, ordenadore estandar bat neurketa edo automatizazio sistema batean bihurtu dezake.

Terminal blokea eta kablearen erabilpena

Terminal blokea, seinaleak konektatzeko lekua ematen du. Seinaleak konektatzeko terminal taldea, eta konektore bat kable bat konektatzeko, terminal blokea eta DAQ tresna lotzeko, osatzen dute terminal blokea. Terminal blokeak, 100, 68, edo 50 terminalak dituzte. Terminal blokearen mota bi faktoreen araberakoa da: trena, eta neurtzen hari diren seinaleen kopurua. 68 terminal dituen terminal blokeak, lurrerako terminal gehiago konektatzeko aukera ematen ditu, 50 terminal dituen terminal blokeak baino. Lurrerako terminal gehiegi edukitzeak ekiditen du kableen arteko konekzio asko egitea lurrerako konekzioa lortzeko, interferentziak gutxituz.

Terminal blokeak, babestuak edo ez-babestuak izan daitezke. Babestutako terminal blokeak, babes hobeagoa ematen dute zarataren aurrean. Terminal bloke batzuk, beste aukera bereziak eskaintzen dituzte, adibidez, elkarketa beroaren konpentsazioa, beharrezkoa dena termopareen neurketa egokia egiteko.

Kableak, seinalea terminal blokeatik DAQ tresnara garraiatu egiten du. Kableak, 100, 68, edo 50 pineko konfigurazioa izan dezakete. Konfigurazioa aukeratu egiten da terminal blokearen eta erabiltzen den DAQ tresnaren arabera. Kableak, terminal blokeak bezala, babestuak edo ez-babestuak izan daitezke.

DAQ seinale eranskina

DAQ seinale eranskina, bloke terminal bat da ikasketarako diseinatuta. Hiru kable konekzio ezberdinak ditu, DAQ tresna ezberdinekin erabiltzeko, eta terminalak seinaleak konektatzeko. Sarrera analogikoko hiru kanal eskuragarri daude, bat tenperatura sentsoreara konektatuta dago, eta beste biak irteera analogikoko bi kanaletara.

DAQ seinale eranskinak funtzio sorgailu bat barneratzen du, etengailu bat duena seinalearen frekuentzia tartea aukeratzeko. Funtzio sorgailuak, uhin sinusoidala edo uhin karratua sortu ditzake. Lurrerako konekzioa, uhin sinusoidalaren terminala eta uhin karratuaren terminalaren artean kokatuta dago.

Trigger digital botoiak, TTL pultso bat sortzen du, trigger sarrera edo irteera analogikorako. Trigger botoia sakatzerakoan, seinalea +5V-tik 0V-ra doa, eta +5V-ra bueltatzen da botoia askatzerakoan. Lau LED DAQ tresnaren lehenengo lau digital mailak adierazten dituzte. LED-ak alderantzizko logika erabiltzen dute, beraz, maila altua denean LED-a amatatuta egongo da, eta maila altuarekin piztuko da.

DAQ seinale eranskinak, koadraturazko kodifikadore bat dauka, bi pultsu trenak sortzen dituenak kodifikadorearen kontrola biratzerakoan. Terminalak, sarrera eta irteera sinalea daukate. Azkenik, DAQ seinale eranskinak, aurretik aipatutako instrumentuez aparte beste elementu batzuk ditu aipatzekoak: errele bat, termopare sarrera bat, eta mikrofono jack bat.

DAQ tresnaren erabilpena

DAQ tresna gehienak lau elementu estandarrak izan ohi dituzte: sarrera analogikoak, irteera analogikoak, I/O digitalak, eta kontadoreak.

DAQ tresnarekin neurtutako seinalea, ordenadorera transferitu daiteke bus estruktura barietate ezberdinen bidez.

B. <u>Hardware-arekin komunikazioa:</u>

National Instrument-en datu eskuraketa tresnak, driver motor bat dauka, tresna eta software aplikazioa komunikatzeko. Driver motor bi daude aukeratzeko: NI-DAQmx eta ohiko NI-DAQ. Labview erabili daiteke driver motoreekin komunikatzeko. Izan ere, jadanik erabili da Labview-ko DAQ asistentea, datu eskuraketa tresnarekin komunikatzeko. DAQ asistentea, VI Express bat da NI-DAQmx-ekin komunikatzeko.

Gainera, National Instrument, datu eskuraketa tresna konfiguratzeko oso erabilgarria den aplikazio bat dauka: Measurement & Automation Explorer (MAX).

NI-DAQ erabilpena

NI-DAQ 7.0 bi NI-DAQ driver-ak ditu: ohiko NI-DAQ eta NI-DAQmx (bakoitzak bere programazio interfaze aplikazioa (API), hardware konfigurazioa, eta software konfigurazioa duelarik).

- Ohiko NI-DAQ-ak, NI-DAQ 6.9.x NI-DAQ-ren aurreko bertsioaren eguneraketa bat da. Ohiko NI-DAQ, NI-DAQ 6.9.x-en VI eta funtzio berdinak ditu, eta era berdinean funtzionatzen du. Ohiko NI-DAQ erabili daiteke NI-DAQmx erabiltzen den ordenagailu berdinean erabili daiteke, NI-DAQ6.9.x bertsioarekin egin ahal ez zena.
- NI-DAQmx, azkenengo NI-DAQ driver-a da, VI, funtzio eta neurketa instrumentuak kontrolatzeko tresna berriekin. Hurrengoak dira NI-DAQmx –en abantailak, NI-DAQ –ren aurreko bertsioen aurrean: DAQ asistentea kanalak eta neurketa lanak konfiguratzeko; errendimendua handitzen du, I/O analogiko azkarragoak eta multi-irakurketa; eta API sinple bat, DAQ aplikazioak sortzeko, funtzio eta VI gutxiago erabiliz, NI-DAQ-ren aurreko bertsioak baino.

Ohiko NI-DAQ eta NI-DAQmx-a instrumentu eta tresna ezberdinak erabiltzen dituzte. NI neurketa tresna programatzerakoan, NI software aplikazioak erabili daitezke, adibidez: Labview, LadWindows/CVI, edo Measurement Studio. NI software aplikazioa erabiliz, datu eskuraketa eta kontrol aplikazioen garapen denbora asko txikitu egiten da, erabiltzen den programa kontutan izan gabe:

- Labview-k datu eskuraketa Labview DAQ VI-en bidez egiten du, VI batzuk NI neurketa tresnekin programaturik.
- C bilakatzaileentzako, LabWindows/CVI, ANSI C ingurunearen integratu bat da, LabWindows/CVI datu eskuraketa liburutegia eskaintzen duena, NI neurketa tresnak programatzeko.
- Measurement Studio garapen tresneria, Microsoft Visual Studio. NET-an testak eta neurketa softwareak diseinatzeko erabiltzen da. Measurement Studio baditu tresnak Visual C#, Visual Basic. NET, eta Visual C++. NET-en erabiltzeko.

DAQ harware konfigurazioa

Datu eskuraketa tresna erabili baino lehen, baieztatu behar da softwarea tresnarekin komunikatu daitekeela.

✓ <u>Windows</u>

"Windows Configuration Manager" gorde egiten du ordenagailuan instalatutako hardware guztia, National Instrument DAQ tresna barne. Plug & Play (PnP) tresna badauka, adibidez, E Series MIO tresna, "Windows Configuration Manager"-ak automatikoki detektatu egiten du eta konfiguratzen du. PnP tresna ez badauka, trena eskuz konfiguratu beharko da, kontrol paneleko Add New Hardware aukerarekin.

Windows konfigurazioa egiaztatu daiteke Device Manager-en bidez. Data Acquisition Devices ikusi daiteke, non egongo dira ordenagailuan instalatuta dauden DAQ tresna guztiak. Birritan klikeatuz DAQ tresna, elkarrizketa koadro bat ireki egingo da orri ezberdinekin. "General" orrian, informazio orokorra agertuko da, tresna kontutan hartu gabe. "Resources" orrian, tresnaren sistema baliabideak zehaztu egiten dira: eten mailak, DMA, eta software konfigurazio tresnen oinarrizko helbideak. "NI-DAQ Information" orrian, DAQ tresnaren bus mota finkatu egiten da. "Driver" orrian, DAQ tresnaren driver bertsioa eta kokapena zehaztu egiten da.

✓ <u>Measurement & Automation Explorer</u>

Labview-k MAX instalatu egiten du, tresna eta kanal konfigurazio parametroak finkatu egiten dituenak. DAQ tresna ordenagailuan instalatu ostean, konfigurazio lanabes hau martxan jarri behar da. MAX, Device Manager-ak Windows Registry – an grabatutako informazioa irakurtzen du, eta zenbaki bat esleitzen dio DAQ tresna bakoitzari. Zenbaki hau erabiltzen da, tresna bakoitza identifikatzeko Labview barruan. MAX-ra heldu daiteke birritan klikeatuz goiko aldean dagoen ikonoan, edo aukeratuz Tools-Measurement & Automation Explorer, Labview-n. Ondorengo leihatila MAX leihatila nagusia da.



✓ <u>Eskalak</u>

Neurketetarako eskalak konfiguratu daitezke. Hau oso erabilgarria da sentsoreekin lan egiterakoan. Aplikazioan eskalatutako balio bat ematea onartzen du. Adibidez, kurtso honetan tenperatura sentsore bat erabiltzen da errepresentatzen duena tenperatura boltaiarekin. Tenperaturaren bihurketa ekuazioa ondorengoa da: *voltaia x 100 = Celsius*. Eskala konfiguratu ostean, edozein aplikazio programan erabili daiteke, tenperatura balio bat lortuz, boltaia baino.

C. <u>DAQ</u> device baten simulazioa:

NI-DAQmx simulazio tresnak eratu daitezke NI-DAQmx 7.4 bertsioan edo beranduagoko batean. NI-DAQmx simulazio tresna erabiliz, NI produktuak probatu daitezke edozein aplikazioan inolako hardware-aren beharrarekin. Beranduago hardwarea lortzerakoan, NI-DAQmx simulazio tresna konfigurazioa barneratu daiteke tresna fisiko batean, MAX Portable Configuration Wizard erabiliz. NI-DAQ mx simulazio tresnarekin, tresna fisikoaren konfigurazioa esportatu daiteke sistema batera, non tresna fisikoa instalatuta ez dago.

NI-DAQmx simulazio tresna eratzeko

NI-DAQmx simulazio tresna sortzeko, ondorengo pausuak jarraitu behar dira:

- 1. "Devices and Interfaces" ireki.
- 2. "NI-DAQmx Devices" ezkerreko botoiaz sakatu eta Create New NI-DAQmx Device » NI-DAQmx Simulated Device aukeratu.
- 3. "Choose Device" elkarrizketa koadratuan, aukeratu simulatu nahi den tresna, eskuragarri dauden tresna familien artean.
- 4. Aukeratu tresna eta OK klikeatu.

- 5. PXI tresna aukeratzekotan, txasis zenbakia eta PXI slot zenbakia jarri beharko da.
- 6. SCXI txasisa aukeratzekotan, SCXI konfigurazio panela irekiko da.

NI-DAQmx simulazio trenza ezabatzeko

NI-DAQmx simulazio tresna ezabatzeko, ondorengo pausuak jarraitu behar dira:

- 1. Devices and Interfaces » NI-DAQmx Devices ireki.
- 2. Ezkerreko botoia sakatu ezabatu nahi den NI-DAQmx simulazio tresnaren gainean.
- 3. "Delete" sakatu.

D. Sarrera analogikoaren neurketa:

Sarrera analogikoa, seinale analogiko beten neurketa prozesua da, eta neurketaren transferentzia ordenagailu batera analisirako, ikustarazteko edo, gordetzeko. Seinale analogiko bat, etengabe aldatzen ari den seinale bat da. Sarrera analogikoa, boltaia edo korrontea neurtzeko erabiltzen da gehien bat. Tresna mota asko erabili daitezke sarrera analogikoa adierazteko: DAQ multifuntzio tresna (MIO), abiadura handiko digitalizatzaileak, multimetro digitalak (DMMs) eta, Dynamic Signal Acquisition (DSA) devices.

Analogiko-digital bihurketa

Ordenagailu batekin seinale analogiko bat eskuratzeko, analogiko-digital bihurketa egitea beharrezkoa da, seinale elektriko bat hartzen duena eta datu digitalean bihurtu egiten du, ordenagailuak prozesatu ahal izateko. Analogiko-digital bihurgailuak (ADC), zirkuitu batzuk dira, boltaiaren maila "0" eta "1"-etan bihurtu egiten dituenak.

ADC-ak seinalearen laginak hartzen dituzte erloju baten goranzko edo beheranzko flankoarekin sinkronizaturik. Ziklo bakoitzean, ADC-ak seinale analogikoaren lagin bat hartzen du, horrela seinalea neurtua eta digital balio batean bihurtua izateko. Laginketa erloju batek kontrolatu egiten du, zenbatero hartu egiten diren sarrerako seinalearen laginak.

Sinkronizazio lanabesaren erabilpena

Sarrera analogikoarekin lan egiterakoan, konfiguratu daiteke lagin bat eskuratzeko, n lagin eskuratzeko edo, etengabe laginak eskuratzeko.

✓ <u>Lagin bat eskuratzeko</u>

Lagin bat eskuratzea, denon esku dagoen eragiketa bat da. Driver-ak sarrera kanaletik balio bat eskuratzen du eta zuzenean balioa ematen du. Eragiketa hau ez du behar inolako hardware tenporizariorik. Adibidez, periodikoki tanke baten fluidoaren maila monitorizatzeko, datu puntualak eskuratu beharko liratezke. Konektatu beharko liratezke fluidoaren mailaren araberako boltaia ematen duen transduktorera eta, neurketa tresnaren kanal bat, eta kanala hasieratu datu eskuraketa puntuala egiteko fluidoaren maila ezagutu nahi denean.

✓ <u>N lagin eskuratzeko</u>

Era bat kanal baten edo askoren lagin ugari eskuratzeko, modu errepikakor batean lagin bat hartzea da. Hala ere, kanal bat edo askoren lagin datu bat behin eta berriro eskuratzea ez da eragingarria eta denbora asko kontsumitzen du. Gainera, ezin da kontrol zehatza eraman lagin edo kanalen artean pasatutako denbora tarteaz. Horren ordez, hardware sinkronizazioa erabili daiteke, ordenagailu memorian buffer bat erabiltzen duena datuak modu eraginkorragoan eskuratzeko. Programatzerakoan, sinkronizazio funtzioa gaineratu behar da, eta zehaztu "sample rate" eta "sample mode".

NI-DAQmx-ekin, kanal askotako datuak multzokatu daitezke. Adibidez, tankearen fluidoaren maila eta tenperatura monitorizatzeko. Kasu honetan, tresnaren bi kanaletara bi transduktore konektatu behar dira.

✓ <u>Etengabe laginak eskuratzeko</u>

Lagin talde bat ikusi, prozesatu edo gordetzeko eskuratuta izaten diren momentuan, beharrezkoa da laginak etengabe eskuratzea. Aplikazio mota hauentzako, laginketa modua "continuous" izan behar da.

Trigger lanabesaren erabilpena

NI-DAQmx-ek kontrolatutako tresna batek, tresna, akzio espezifiko bat egitera behartzen duen estimulu baten aurrean erreakzionatzen du. NI-DAQmx akzio guztiak estimulu edo kausa bat behar du. Ohizko bi akzio, laginak sortzea eta, grafiko eskuraketaren hasieraketa dira. Estimuluaren ostean, akzioa burutu egiten da.

Akzioetarako kausak "trigger" deritzo. Adibidez, "start trigger" datu eskuraketa hasieratzen du; "Preference trigger" sarrera lagin talde batean erreferentzia puntua finkatzen du; "Pretrigger data" erreferentziazko puntuaren aurreko datu eskuraketa da eta "Posttrigger data" erreferentziazko puntuaren ondorengo datu eskuraketa.

E. <u>Irteera analogikoaren sortaraztea:</u>

Irteera analogikoa, ordenagailuatik irteera seinalea sortzearen prozesua da. Irteera analogikoa, digital-analogiko (D/A) bihurketak burutzeaz sortu egiten da. Eskuragarri dauden irteera analogiko motak edozein aplikazioarako, boltaia eta korrontea dira.

Boltai edo korronte lan bat burutzeko, bateragarria den tresna instalatuta egon behar da, mota horretako seinalea sortu ditzakeena.

Sinkronizazio lanabesaren burutzea

Irteera analogikoa burutzerakoan, lana sinkronizatuta egon daiteke lagin bat sortzeko, n lagin sortzeko edo, etengabe laginak sortzeko.

✓ <u>Lagin bat sortzeko</u>

Lagin bateko eguneraketa egin ohi da, seinalearen maila erritmoa baino garrantzitsua denean. Adibidez, momentu oro lagin bat sortzen da, DC edo seinale konstante bat sortzeko. Software sinkronizazioa, tresnak noiz sortzen duen seinale bat kontrolatzeko erabili daiteke.

Eragiketa honek ez du inolako hardware sinkronizaziorik behar. Adibidez, jakina den boltai bat sortu behar bada tresna bat estimulatzeko, lagin bateko eguneraketa lanabes egokia izango litzateke.

✓ <u>N laginak sortzeko</u>

Era bat kanal baten edo askoren lagin ugari sortzeko, modu errepikakor batean lagin bat sortzea da. Hala ere, kanal bat edo askoren lagin datu bat behin eta berriro sortzeak ez da eragingarria eta denbora asko kontsumitzen du. Gainera, ezin da kontrol zehatza eraman lagin edo kanalen artean pasatutako denbora tarteaz. Horren ordez, hardware sinkronizazioa erabili daiteke, ordenagailu memorian buffer bat erabiltzen duena datuak modu eraginkorragoan sortzeko.

Software sinkronizazioa edo hardware sinkronizazioa erabili daiteke, seinale bat noiz sortua izaten den kontrolatzeko. Software sinkronizazioarekin, laginak sortuak izaten diren erritmoa, software eta operazio sistemaren araberakoa da, neurketa tresnaren araberakoaren ordez. Hardware sinkronizazioarekin, laginen sorketa erritmoa, TTL seinale batek (tresna baten erloju bat adibidez) kontrolatzen du. Hardware erloju bat, software loop bat baino azkarragoa izan daiteke. Gainera hardware erloju bat, software loop bat baino zehatzagoa da.

Programatzerakoan, sinkronizazio funtzioa gaineratu behar da, eta zehaztu "sample rate" eta "sample mode". Beste funtzio batzuekin bezala, lagin ugari sortu daitezke kanal bakarrerako edo askotarako.

N laginen sorketa erabili, denbora konstantea duten seinaleak sortzeko, adibidez, AC sinu uhina.

✓ <u>Etengabe laginak sortzeko</u>

Etengabeko sorketa, n laginen sorketaren antzekoa da baina, akzio bat gertatu behar da sorketa gelditzeko. Etengabe seinaleak sortzeko, adibidez, AC sinu uhin infinitua, "continuous" sinkronizazio modua jarri behar da.

Trigger lanabesaren sortzea

NI-DAQmx-ek kontrolatutako tresna batek zerbait egiten duenean, akzio bat sortzen du. Ohizko bi akzio, lagina ekoiztea eta, sorketa hasieratzea dira. NI-DAQmx akzio guztiak estimulu edo kausa bat behar du. Estimulua gertatzerakoan, akzioa sortzen da. Akzioetarako kausak "trigger" deritzo. "Start trigger" sorketa hasieratzen du.

Digital-analogiko bihurketa sortzea

Digital-analogiko bihurketa, analogiko-digital bihurketaren kontrakoa da. Digitalanalogiko bihurketan, ordenagailuak sortzen du datua. Datua, sarrera analogikotik eskuratua izan daiteke, edo ordenagailuan bertan software bidez sortua izan. Digitalanalogiko bihurgailuak (DAC), datua hartzen du eta irteera pinaren boltaia aldatzeko erabiltzen du. DAC-ak seinale analogiko bat sortzen du, ondoren beste tresna edo zirkuituetara bidali daitekeena.

DAC-ak eguneraketa erloju bat dauka, DAC-ri balio berri bat noiz sortzeko adierazten diona. Eguneraketa erlojuaren funtzioa digital-analogiko bihurgailu batean, laginketa erlojuaren funtzioa analogiko-digital bihurgailu baten antzekoa da. Erlojuaren ziklo

bakoitzean, DAC-ak digital balio bat boltai analogiko balio batean bihurtzen du eta, irteera bat sortzen du pin bateko boltai moduan. Abiadura handiko erloju erabiltzen bada, DAC-ak ia konstantea dirudien seinale bat sortu ditzake.

F. <u>Kontadoreen erabilpena:</u>

Kontadorea digital sinkronizazio tresna bat da. Normalean kontadoreak kontatzeko, maiztasuna neurtzeko, periodoa neurtzeko, posizioa neurtzeko eta, pultso sortzaile bezala erabili ohi dira.

✓ <u>Kontaketa erregistroa</u>

Kontadorearen oraingo kontaketa gordetzen du. Kontaketa erregistroa software bidez gorde daiteke.

✓ <u>Iturria</u>

Sarrrera seinalea, kontaketa erregistroaren oraingo kontaketa gordeketa aldatu dezakeena. Kontadoreak, iturri seinalearen goranzko edo beheranzko flankoen bila dago. Kontaketa, goranzko edo beheranzko flankoan alda daiteke, eta hau software bidez aukeratzen da. Aukeratutako flanko mota, seinalearen flanko aktiboa izango da. Iturri seinalean flanko aktiboa gertatzen denean, kontaketa aldatzen da. Flanko aktiboarekin kontaketa goranzkoa edo beheranzkoa den, software bidez aukeratzen da ere.

✓ <u>Atea</u>

Sarrera seinale bat erabakitzen duena iturriaren flanko aktibo batek kontaketa aldatzen duen edo ez. Kontaketa gerta daiteke atea maila altuan, maila baxuan edo, goranzko eta beheranzko flankoen konbinazio batzuen artean dagoenean. Atearen konfigurazioa software bidez egiten dira.

✓ <u>Irteera</u>

Irteera seinale bat pultsoak edo pultsu trenak sortzen dituena.

Kontadorea kontaketa sinplerako konfiguratzen denean, kontaketa gehitzen da iturrian goranzko flanko bat gertatzerakoan. Kontaketa goranzko flankoarekin gehitzeko, kontadorea hasieratuta egon behar da. Kontadoreak, kontaketa zenbaki finko bat dauka, kontadorearen erresoluzioa bezala definiturik. Adibidez, 24 bit-eko kontadoreak kontatu dezake: 2(counter resolution) - 1 = 2(24) - 1 = 16,777,215

Beraz, 24 bit-eko kontadore bat 16, 777, 215 baliora heltzen denean, kontaketa terminalera heldu da. Hurrengo flanko aktiboarekin kontadorea berriro hasieratu beharko da eta 0 posizioan jarri.

G. <u>I/O digitalen erabilpena:</u>

Seinale digitalak, datu digitalak kable batetik bidaltzen dituzten seinale elektrikoak dira. Seinale hauek soilik bi egoera dituzte: on/off, altua/baxua/, 0/1... Seinale digital bat kable batetik bidaltzerakoan, bidaltzaileak boltai bat ematen dio kableari, eta hartzaileak boltaiaren maila interpretatzen du bidalitakoaren balioa ezagutzeko. Balio digital bakoitzaren boltai tartea, erabilitako boltai maila estandarraren araberakoa da. Seinale digitalak erabilpen ugari dituzte; erabilpen ohikoena, digital edo egoera finitudun tresnen kontrola edo neurketa da, etengailu edo LED-etan adibidez. Seinale digitalak ere erabili daitezke datuak bidaltzeko, tresnak programatzeko edo, tresnen arteko komunikazioa egoteko. Gainera, seinale digitalak, erloju edo trigger modura erabili daitezke, beste neurketak kontrolatzeko edo sinkronizatzeko.

Lerro digitalak, DAQ tresna batean erabili daiteke balio digitalak eskuratzeko. Eskuraketa mota hau software sinkronizazioan oinarrituta dago. Tresna batzuetan, lerroak konfiguratu daitezke, lagin digitalak neurtu edo sortzeko. Lerro bakoitza lanabesaren kanal bati dagokio.

Portu digitalak, DAQ tresna batean lerro digital batzuen balio digitala eskuratzeko erabili daiteke. Eskuraketa mota hau software sinkronizazioan oinarrituta dago. Portuak konfiguratu daitezke, lagin digitalak neurtu edo sortzeko. Portu bakoitza lanabesaren kanal bati dagokio.

9.GAIA:DATU ESKURAKETAREN SOFTWAREA ETA HARDWAREA

Gai honi dagozkion atalak ondorengoak dira:

- A. DAQ gailuaren osagaiak.
- B. Konfigurazioa.
- C. DAQ gailua simulatuz.
- D. Lurrerako konekzioen ingurukoak.
- E. Neurketa sistema motak.
- F. DAQ softwarea.
- G. NI-DAQ.
- H. Meassurement & Automation Explorer (MAX).
- I. NI-DAQmx VI-en orokortasunak.

A. <u>DAQ gailuaren osagaiak:</u>

Interfaze-ak

DAQ gailu tipikoak hiru interfaze ditu seinaleak jaso eta bidaltzeko: I/O konektorea, ordenagailuaren I/O interfaze zirkuitua, eta Real Time System Integration (RTSI) busa.

- **I/O konektorea:** I/O konektorea, seinaleak DAQ gailura sartu edo irtetzeko bidea da. I/O konektoreak 100, 68 edo 50 pin izan ditzake, gailuaren arabera. Kablearen amaiera bat I/O konektorera konektatzen da, eta beste amaiera terminal blokera.
- **Ordenagailuaren I/O interfaze zirkuitua:** Ordenagailuaren I/O interfaze zirkuituak, DAQ gailuaren eta ordenagailuaren arteko datu transferentzia ahalbidetzen du. Zirkuitua ezberdina izan daiteke erabilitako bus protokoloaren arabera.
- RTSI busa: RTSI busaren bidez, ordenagailu baten DAQ gailu ezberdinen arteko informazio trukaketa, edo seinale guztien sinkronizazioa ahalbidetzen du. Adibidez, bi gailuk sarrera analogiko bat maiztasun berdinean adieraztea nahi bada, RTSI busaren bidez erloju seinale bat erdibanatu dezakete, horrela bi gailuek erloju seinale berdina erabiltzeko. RTSI busa erabiltzen da, beraz, gailuak elkarkonektatzeko, PXI plataforma erabiltzen ez bada. PXI txasisean gailuak konektatzen dira RTSI bus baten antzerako lana burutuz.

Sarrera analogiko zirkuitua

Sarrera analogikoko seinaleak, I/O konektoretik igaro ondoren, sarrera analogiko zirkuitutik pasatzen dira, analogiko-digital bihurgailura (ADC) joan baino lehen. Sarrera analogiko zirkuituak, multiplexadore bat eta instrumentazio anplifikadore bat osatzen dute.

- Multiplexadorea: Multiplexadorea, edo mux, etengailu bat da sarrera kanal guztietatik soilik bat konektatzen duena instrumentazio anplifikadorearekin. Datuak kanal askoetatik eskuratzen direnean, multiplexadoreak txandakatzen da datu bakarra konektatuz momentu bakoitzean anplifikadorearekin. Labview-k kontrolatu egiten du zein ordenean konektatu egiten ditu multiplexadoreak seinaleak anplifikadorera.
- **Instrumentazio anplifikadorea:** Instrumentazio anplifikadoreak, anplifikatu egiten du jasotako seinalea. Anplifikazioaren asmoa, seinalearen anplitudea, ADC-ren anplitudera gehien hurbiltzea da. Anplifikadoreak seinale bat anplifikatzen duenean, irabazpen bat gehitzen dio. Irabazpen hau, seinaleak zenbat anplifikatuko den izango da. Adibidez, biko irabazpena seinalea bikoizten du, eta 0.5-eko irabazpena, seinalea erdibitzen du.

Analogiko-digital bihurgailua (ADC)

ADC-ak gailu elektroniko bat da, boltai analogikoa zenbaki digital batean bihurtzen duena, horrela ordenagailura bidali daiteke honek informazioa ulertu eta azaldu dezake ordenagailuaren I/O interfaze zirkuituaren bidez. Sarrera analogiko zirkuitua eta ADC

bidez, seinale analogiko bat eskuratzen da, eta seinalearen maila, itxura edo maiztasuna neurtu daiteke.

Digital-analogiko bihurgailua (DAC)

DAC-ak ADC-ren aurkako lana burutzen du. Ordenagailutik ordenagailuaren I/O interfaze zirkuitura bidalitako zenbaki digital bat hartzen du eta, seinale analogiko batean bihurtzen du, I/O konektoretik kanporatuko dena. DAC erabilia izaten da DC seinaleak sortzeko (maila), maila espezifikoak sortzeko (maiztasunak), eta grafikoak (itxura).

Digital I/O zirkuitua

Digital I/O zirkuituak bai sarrera zein irteera funtzioak bete ditzake. E serieko DAQ tipiko batek digital I/O tenporizazio software bat dauka zortzi lerro digitalekin, seinaleak eskuratu edo sortzeko. Lerro digitalak ez dute tenporizazio zirkuiturik eta ez daude erabiltzailearen esku. Beraz, erabilgarriak dira seinale digital baten maila neurtzeko, baina ez erritmoa (maiztasuna). M serieko DAQ tipiko batek digital I/O tenporizazio hardware bat dauka 28 lerro digitalekin, seinaleak eskuratu edo sortzeko. Digital I/O erlojua hardware tenporizazioa da, beraz, M serieko gailuek erloju iturri finko, iraunkor bat daukate. M serieko DAQ gailu baten lerro digital ezberdinak, ere sinkronizatuta izan daitezke kanpo erloju baten bidez, 10MHz baino txikiagokoa izan beharko dena. Digital I/O zirkuitua erabili daiteke etengailu bat monitorizatzeko, egoera aldatu duen ikusteko.

Kontadore zirkuitua

Kontadoreak, seinale digitalak sortu eta eskuratu egiten dituzte. Beraien tenporizazio seinaleak, "timebases" deituak, idealak egiten dituzte seinale digital baten erritmoa (maiztasuna) neurtzeko. DAQ gailu baten kontadore zirkuitua erabili ohi da motore baten ardatzaren maiztasuna neurtzeko, motorea kontrolatzeko maiztasun zehatz bateko pultsu tren bat sortzeko.

B. <u>Konfigurazioa:</u>

Sarrera eta irteera analogikoen hainbat ikuspegi, erlazio handia daukate DAQ gailuaren konfigurazioarekin.

- ✓ ADC-ren anplitudea eta erresoluzioa.
- ✓ Instrumentazio anplifikadorearen irabazpena.
- ✓ Anplitudea, erresoluzioa eta irabazpenaren konbinazio bat "code width value" ezaugarria kalkulatzeko.

Erresoluzioa

Seinale analogiko bet errepresentatzeko erabilitako bit kopurua definitu egiten du ADC baten erresoluzioa. DAQ gailu baten erresoluzioa, erregela baten marken antzekoa da. Erregela gero eta marka gehiago izan, neurketa gero eta zehatzagoa izango da. DAQ gailu batean erresoluzioa gero eta altuagoa izan, handiagoa izango da sistema batek ADC-aren anplitudea banatutako dibisio kopurua, eta orduan, detektatutako aldaketa txikiagoa izango da. Hiru bit-eko ADC batek, anplitudea 23 edo 8 dibisioeta banatzen

du. Kode binario edo digital batek 000 eta 111 bitartean dibisio bakoitza adieraziko du. ADC-ak seinale analogikoaren neurketa bakoitza dibisio digital batean bihurtzen du.

Ondorengo irudian, 5kHz-ko sinu uhin digital bat agertzen da 3 bit-eko ADC batez eskuratutakoa. Seinale digitalak ez du oso ondo adierazten seinale originala, bihurgailuak dibisio digital gutxi dituelako seinale analogikoaren boltai aldaketa adierazteko. Ordea, erresoluzioa 16 bit-eraino handituz, ADC-ren dibisio kopurua 8-tik 65,536-raino handitzen da, horrela 16 bit-eko ADC-ak seinale analogikoaren adierazpen oso zehatza lortzen du.



Gailu anplitudea

Anplitudeak adierazten du, ADC-ak digitalizatu ditzakeen seinale analogikoaren mailen maximoak eta minimoak. DAQ gailu gehienak aukeratzeko anplitudeak dituzte, normalean 0 eta 10V edo -10 eta 10V. ADC anplitudea aukeratu daiteke erresoluzioaren arabera seinalearen neurketa egokiena egiteko.

Anplifikazioa

Seinale baten anplifikazioa seinalea digitalizatu baino lehen egin behar da, seinalearen adierazpena hobetzeko. Seinale bat anplifikatuz, ADC baten sarrera anplitudea txikitu daiteke, horrela ADC-ak eskuragarri dauden dibisio digital gehien erabili ditzake seinalea adierazteko.

DAQ gailu batean eskuragarri dagoen anplitudea, erresoluzioa, eta anplifikazioa, definitu egiten dute boltai sarrera batean somatu daitekeen aldaketarik txikiena. Boltaiaren aldaketa hau balio digitalaren LSB (Least Significant Bit) adierazten du, eta ere deitzen da "Code Width".

Code width

Code Width, sistema batek somatu dezakeen seinale baten aldaketarik txikiena definitzen du. Ondorengo formularen bidez kalkulatzen da:

Code width=

Amplification x $2^{\text{resolution in bits}}$

Code width gero eta txikiagoa, gailuak gero eta zehatzago adierazi ahal izango du seinalea. Formulan ikusi daiteke aurretik erresoluzio, anplitude eta irabazpen ataletan ikasitakoa:

- ✓ Erresoluzio handiagoa = code width txikiagoa = seinalearen adierazpen zehatzagoa.
- ✓ Anplifikazio handiagoa = code width txikiagoa = seinalearen adierazpen zehatzagoa.
- ✓ Anplitude handiagoa = code width txikiagoa = seinalearen adierazpen zehatzagoa.

DAQ gailua aukeratzeko orduan code width datua ezagutzea oso garrantzitsua da.

C. <u>DAQ gailua simulatuz:</u>

NI-DAQmx gailu simulatuak sortu daitezke NI-DAQmx 7.4 bertsioan edo ondorengoko batean. NI-DAQmx gailu simulatua erabiliz, NI produktuak frogatu daitezke hardwarea nahiz eta hardwarea ez izan. NI-DAQmx gailu simulatua, gailu erreal baten berdina da, MAX-en "create new" menuan "NI-DAQmx Simutaled Device" aukeratuz sortu egiten dena, funtzio edo programa bat burutzeko hardwarea gabe. Driver-a eta erabilitako programak jadanik ordenagailuan daude. Beranduago hardwarea eskuratzen bada, NI-DAQmx gailu simulatuaren konfigurazioa gailu fisikora eraman daiteke "MAX Portable Configuration Wizard" bidez. NI-DAQmx gailu simulatuarekin ere, gailu fisiko baten konfigurazioa gailu fisikoa instalaturik ez duen sistema batera eraman daiteke.

NI-DAQmx gailu simulatua sortuz

NI-DAQmx gailu simulatu bat sortzeko ondorengo pausuak jarraitu behar dira:

- ✓ Devices and interfaces-en Create»New aukeratu.
- ✓ Lista batean hainbat gailu gertuko dira aukeratzeko NI-DAQmx Simulated Device aukeratu, eta Finish klikeatu.
- ✓ Choose Device elkarrizketa koadroan simulatuko den gailuaren familia aukeratu.
- ✓ Gailua hautatu eta **OK** klikeatu. MAX-en konfigurazio zerrendan NI-DAQ mx gailu simulatuen ikonoa horia da. Ordea, gailu fisikoen ikonoa berdea.

- ✓ PXI gailua aukeratzekotan, txasis zenbaki bat eta PXI Slot zenbaki bat definitu beharko dira.
- ✓ SCXI txasis bat aukeratzekotan SCXI konfigurazio panela irekiko da.

NI-DAQmx gailu simulatua ezabatzeko

NI-DAQmx gailu simulatu bat ezabatzeko ondorengo pausuak jarraitu behar dira:

- ✓ Devices and interfaces»NI-DAQmx Devices ireki.
- ✓ Ezkerreko botoiaz sakatu ezabatu nahi den NI-DAQmx gailu simulatuan.
- ✓ Delete klikeatu.



D. Lurrerako konekzioen ingurukoak (grounding issues):

Zehatzak eta zaratarik gabeko neurketak egiteko, beharrezkoa da seinale sorgailuaren portaera ezagutzea, DAQ gailuaren konfigurazioa egokia izatea, eta kableaketa garbia egitea. Ondorengo irudian, datu eskuraketaren eskema tipikoa ikusi dezakegu:



Aplikazio mota asko aukeratu ahal izateko, DAQ gailu gehienek, sarrera analogikoaren konfigurazioan malgutasun handia eskaintzen dute. Baina malgutasun honek prezio bat dauka, sarrera konfigurazioa eta aplikazioaren arteko nahasketa bat egon daitekeelako. Gai honetan, DAQ gailuan eskuragarri dauden sarrera konfigurazio ezberdinak argitu egiten dira, egokiena aukeratzeko dagokion aplikazioarako.

Seinale iturri motak

Sarrera analogikoen eskuraketan, lurrera konektatutakoak eta flotanteak diren seinale iturriak erabiltzen dira.

✓ Lurrera konektatutako seinale iturriak

Iturri mota hauetan, tentsio seinaleak lurrera erreferentziaturik daude, ondorengo irudian ikusi daitekeen bezala.



Lurrera konektatutako seinaleak lurrerako sistema bat erabiltzen dute, beraz, neurketa gailuaren lurrerako konekzio berdina erabiltzen dute.

Independenteak diren lurrera konektatutako bi seinale iturrien lurrak normalean ez daude potentzial berdinera. Lur sistema berdinera konektaturik dauden bi instrumentuen lurren arteko potentzial diferentzia normalean 10Mv eta 200Mv tartekoa izaten da. Diferentzia handiagoa izan daiteke zirkuituen korronte banaketa ez bada egokia. Honek "**Ground Loop**" deitutako fenomenoa sortzen du.

✓ <u>Seinale iturri flotanteak</u>

Seinale iturri mota hauetan, tentsio seinalea ez dago inolako lur komunera erreferentziaturik, ondorengo irudian ikusi daitekeen bezala.



Seinale iturri flotanteen adibide bateriak, termopareak, transformadoreak, eta anplifikadore isolatzaileak dira. Irudian ikusi daiteke nola iturriaren terminala ez dago sistemaren lurrera konektatuta, terminal independientea da.

E. <u>Neurketa sistema motak:</u>

Normalean edozein zirkuituan aztertu, neurtu edo sortu ohi den seinalea, tentsio seinalea da. Beste fenomeno elektriko batera eraldaketa, korrontera edo maiztasunera adibidez, egokia izaten da, seinalea kable oso luzeen bidez distantzia handietara ingurugiro latzetan bidali behar denean. Hala ere, eraldatutako seinale hori, tentsio seinalera bihurtua izaten da berriro neurtua izan baino lehen. Beraz, oso garrantzitsua da tentsio sorgailua oso ondo ulertzea.

Tentsio seinalea, neurtua izaten da bi puntuen arteko potentzial diferentzia bezala, ondorengo irudian ikusten den bezala.



Neurketa sistema konfiguratu egiten da erabiltzen de hardwarea eta neurketa kontutan hartuz.

Diferentzial neurketa sistema

Neurketa sistema hau, seinale iturri flotanteen antzekoa da, non neurketa lurra flotante baten arabera egiten den, neurketa sistemaren lurraren desberdina dena. Diferentzial neurketa sistema baten sarrerak ez daude erreferentzia finko bati lotuta. Neurketa sistema honen adibide dira bateriaz elikatutako tresnak, edo instrumentazio anplifikadoreak dituzten DAQ gailuak. National Instrument-eko gailu tipiko batek, zortzi kanaleko diferentzial neurketa sistema bat erabiltzen du, ondorengo irudikoa bezalakoa. Multiplexadore analogikoak erabiliz, neurketa kanalen zenbakia handitzen da, instrumentazio anplifikadore bakarra dagoenean. Ondorengo irudian, lurrerako sarrera analogiko (AIGND) pin-a lurreko neurketa sistema da.



Diferentzial neurketa sistema idealak, bere bi terminalen (positiboa+ eta negatiboa-) arteko potentzial diferentziaren esanetara dago. Modu komuneko tentsioa, instrumentazio anplifikadorearen lurrarekiko neurtutako tentsioa da, anplifikadorearen bi sarreretan egongo dena. Diferentzial neurketa sistema ideal batek ez du modu komuneko tentsioa neurtuko. Hau oso erabilgarria da, izan ere modu komuneko tentsioaren bidez zirkuiturako oso kaltegarria izan daitekeen zarata sartu ohi da.

Hala ere, hainbat parametro, modu komuneko tentsioaren anplitudea, edo modu komunaren errefusatze ratioa (CMRR), mugatzen dute diferentzial neurketa sistemaren gaitasuna modu komuneko tentsioa ez barneratzeko.

✓ <u>Modu komuneko tentsioa</u>

Modu komuneko tentsioaren anplitudeak mugatu egiten du neurketa sistemaren sarreran egon daitekeen tentsio anplitudea. Hau errespetatzen ez bada, neurketan erroreak agertu daitezke, eta gainera, gailuaren osagairen bat apurtu daiteke. Ondorengo ekuazioak definitu egiten du modu komuneko tentsioa (Vmk):

$$Vmk = (V + V -)/2$$

non V+ neurketa sistemaren sarrera ez-alderanztailearen tentsioa da, eta V- neurketa sistemaren sarrera alderanztailearen tentsioa.

✓ Modu komunaren errefusatze ratioa

CMRR-ak neurtu egiten du, diferentzial neurketa sistema baten gaitasuna modu komuneko tentsioa errefusatzeko. CMRR-ak maiztasunaren funtzioa da, normalean txikitu egiten da maiztasuna handitzen den heinean. CMRR-a gero eta handiagoa, anplifikadoreak hobeto ezabatu ahal izango du modu komuneko tentsioarekin batera sartutako zarata. DAQ gailu gehienek CMRR-a definitzen dute korronte lerroaren maiztasunaren arabera (normalean 60Hz-koa izaten dena). Ondorengo ekuazioak definitu egiten du CMRR-a (DB-etan):

CMRR = 20log Modu komuneko irabazkena

Erreferentziadun eta ez-erreferentziadun bukaera bakarreko neurketa sistema

Mota honetako neurketa sistemak, lurrerako konekzioko iturrien antzekoak dira, izan ere, neurketa lurrerarekiko egiten baita. Lurrarekiko erreferentziadun bukaera bakarreko neurketa sistema (GRSE), edo erreferentziadun bukaera bakarreko neurketa sistema (RSE), tentsioa neurtzen dute lurrarekin erreferentziatuz, AIGND, zuzenean lotuta dagoena neurketa sistemaren lurrarekin. Ondorengo irudian, zortzi kanaleko erreferentziadun bukaera bakarreko neurketa sistema bakarreko neurketa sistemaren da.



DAQ gailua batzuetan ez-erreferentziadun bukaera bakarreko neurketa sistema erabiltzen du (NRSE), edo seudodiferentzial neurketa, aurrekoaren aldaera bat dena. Ondorengo irudian NRSE sistema ageri da:



NRSE neurketa sistema batean, neurketa guztiak nodo bakarreko sarrera seinale batekiko (AISENSE E serietako eta M serietako gailuetan) egiten dira, baina nodo honen potentziala neurketa sistemaren lurrarekin (AIGND) aldatu daiteke. Kanal bakarreko NRSE neurketa sistema, kanal bakarreko diferentzial neurketa sistemaren berdina da.

F. <u>DAQ</u> softwarea:

DAQ sistema bat osoa egoteko azken osagaia softwarea da. Ordenagailuak DAQ gailutik datuak jasotzen ditu. Erabiltzaileak aukeratutako aplikazioak datuak aldatu egiten ditu erabiltzaileak ulertzeko. Softwareak ere DAQ sistema manipulatu ditzake, DAQ gailua konfiguratuz datuak eskuratzeko orduan. Beraz, DAQ softwareak, programatu daitekeen interfaze independientea da, DAQ hardware gailuaren sarrera analogikoa, irteera analogikoa, I/O digitala eta, kontadoreak/tenporizadoreak programatzeko.

Normalean DAQ softwarea, driver-ak eta software aplikazioak izaten ditu. Driver-ak gailu motaren araberakoak dira, eta gailuak onartzen dituen komandoak izaten ditu. Software aplikazioa, Labview adibidez, driver komandoak bidaltzen ditu. Software aplikazioak ere, eskuratutako datuak ikustarazi eta aztertu egiten ditu.

NI neurketa gailuak NI-DAQ driver softwarea (VI asko neurketa gailua konfiguratzeko, datuak eskuratzeko, datuak bidaltzeko...) barne dauka.

Neurketa sistema batek ondorengo software aplikazioak ditu:

- ✓ **NI-DAQ**: DAQ gailu kontrolatzen duen softwarea.
- ✓ Measurement & Automation Explorer (MAX): Labview eta NI-DAQ bitarteko elkarrizketa egiten duen softwarea.
- ✓ **Labview**: Erabiltzaileak erabilitako softwarea driver-era komandoak bidaltzeko aplikazioa sortzeko, eta datuak eskuratzeko, aztertzeko eta, aurkezteko.

G. <u>NI-DAQ:</u>

NI-DAQ bi NI-DAQ driver-ak ditu, ohiko NI-DAQ (Legacy), eta NI-DAQmx, bakoitzak bere interfaze programazio aplikazioarekin (API), hardware konfigurazioarekin, eta software konfigurazioarekin. NI-DAQ softwarea erabiltzen da National Instrument DAQ gailuekin komunikatzeko.

NI-DAQ bateragarria da ondorengo software aplikazioekin eta programazio lengoaiekin:

- ✓ National Instrument Labview.
- ✓ National Instrument Real-Time Module.
- ✓ National Instrument LabWindows/CVI.
- ✓ National Instrument Measurement Studio.
- ✓ Microsoft Visual C/C++.
- ✓ Microsoft C#.NET.
- ✓ Microsoft Visual Basic.NET.
- ✓ ANSI C.

Ohizko NI-DAQ

Ohizko NI-DAQ (Legacy), NI-DAQ 6.9 bertsioaren ondorengokoa da, izan ere, VI eta funtzio berdinak dituzte, eta era berdinean lan egiten dute. Baina, ohizko NI-DAQ, NI-DAQmx instalatuta dagoen ordenagailu berdinean erabili daiteke, NI-DAQ 6.9 ,ordea, ez.

NI-DAQmx

NI-DAQmx, azkenengo NI-DAQ driver-a da VI, funtzio, eta tresneria berriarekin neurketa gailuak kontrolatzeko. NI-DAQmx ondorengo abantailak ditu NI-DAQ aurreko bertsioen aurrean:

- ✓ DAQ asistentea, NI-DAQmx-en lanak, kanalak, eskalak...konfiguratzeko interfaze grafikoa da, ondoren Labview-n erabiltzeko. DAQ asistentea erabiltzen da NI-DAQmx kodea sortzeko lanak eta kanalak burutzeko, edo NI-DAQmx kodea beste DAQ sistema batean erabiltzeko. Labview edo MAX erabiltzen dira DAQ asistentea konfiguratzeko.
- ✓ Adierazpena zabaltzen du.
- ✓ API sinple bat, DAQ aplikazioak sortzeko funtzio eta VI gutxiago erabiliz NI-DAQ ren aurreko bertsioak baino.
- ✓ Labview-ren funtzioak handitzen ditu, Property Nodeak barneratuz adibidez eta, adierazpen grafikoa hobetzen du.
- ✓ API eta funtzionalitate antzekoak denentzako: ANSI C, LabWindows...

H. <u>Measurement & Automation explorer (MAX):</u>

MAX, Windows-en oinarritutako aplikazio bat da NI-DAQ instalazio bitartean zuzenean instalatzen dena. MAX erabili ohi da NI software eta softwarea konfiguratzeko eta probatzeko, kanal eta interfaze berriak sartzeko, sistema miaketak burutzeko, eta sistemara konektatuta dauden gailu eta tresnak ikustarazteko. MAX erabili beharko da, ohiko NI-DAQ edo NI-DAQmx-ekin programazioan gailuak konfiguratzeko. MAX ondorengo funtzio mailak ditu:

- ✓ Data Neighborhood.
- ✓ Devices and Interfaces.
- ✓ Historical Data.
- ✓ Scales.
- ✓ Software.
- ✓ VI Logger Tasks.
- ✓ IVI Drivers.
- ✓ Remote Systems

Data neighborhood



Data neighborhood bidez era erraz batean konfiguratu daitezke sistemaren kanal fisikoak, haien artean DAQ kanal eta lan birtualak. Maila honek ere, kanal birtual horiek testeatzeko eta birkonfiguratzeko aukera ematen du. Baita ere, data neighborhood-en bitartez DAQ gailura heldu daiteke eta, sortu edo konfiguratu kanal eta lan birtualak.

✓ <u>DAQ asistentea</u>

DAQ asistentea, interfaze grafikoa da neurketa kanalak eta lanak sortu eta konfiguratzeko.

• Kanalak: NI-DAQmx kanalaren konfigurazio informazioa, eskala edo sarrera limiteak, adibidez, kanal fisiko jakin baterako. Kanalaren konfigurazio informazioa atera daiteke, eta aldi berean, kanalari izen adierazgarri bat eman (hau oso garrantzitsua da, izan ere, labview-k objetuak izenaren arabera identifikatzen ditu). Ondoren, izen horren bidez, kanal horretara eta bere konfigurazioara heltzea posiblea izango litzateke. Kanalaren deskribapena egin daiteke, kanalak erabiliko duen transkonduktorea aukeratu, anplitudea definitu (irabazpena finkatu), lurrera konektatzeko modua aukeratu, kanal birtualaren eskala esleitu, eta

kanalari izen adierazgarri bat eman kanalaren zenbakia ordezkatuz, dena aldi berean.

Adibidez, demagun DAQ-ren 0 kanala tenperatura sentsore batekin kableaturik dagoela. Kanal birtual bat eratu daiteke 0 kanalerako eta Tenperatura Sentsorea deitu. Kanal birtuala eratu daitezke I/O analogikoetarako, I/O kontadoreentzako, eta I/O digitalentzako. Kasu honetan, kanalari izen bat jarri zaionez, errazagoa izango da jakitea kanalak egiten zuena, zenbakiarekin uzten bada baino.

Select the measurement type for your task. To have multiple measurement types within a single task, you must first create the task with one measurement type. After you create the task, click the Add button to add a new measurement type to the task.	Analog Input Analog Output Counter Input Counter Output Digital I/O TEDS	KANALAK (channels)
---	--	-----------------------



• Lanak: NI-DAQmx lana (tarea), tenporizazio eta trigger berdina daukaten kanal birtualen taldea da. Teorikoki, lan bat, bururu nahi den

neurketa edo sortutako objektu bat adierazten du. Lana osatzen duten kanalak, beste edozein lan batean erabiliak izan daitezke (global channel), edo lan bakar baterako esleitu (local channel). Ere, lan bat sortzen den bitartean kanal berriak eratu daitezke, edo lan bat burutu daiteke DAQ asistentearen bidez sortutako kanalekin.



Devices and interfaces



Kategoria honek adierazi egiten du instalatutako eta detektatutako hardwarea. Devices and interfaces kategoria honetan ere agertuko dira ondorengo lanabesak gailuak konfiguratu eta testeatzeko:

\checkmark <u>Self test</u>

Self test lanabesak, DAQ gailuan barne testeo bat hasieratzen du, iturri eta objektu guztiak ondo konektatuta dauden jakiteko, eta gailua ondo konfiguratuta dagoen ikusteko.

\checkmark <u>Test panels</u>

Test panel lanabesak, DAQ gailuaren I/O analogikoen, I/O digitalen eta kontadorearen I/O-en funtzionamendua testeatzen du. Lanabes hau erabili ohi da

gailuaren funtzionamendua eta sistemaren konfigurazioa gainbegiratzeko NI-DAQmx-etik zuzenean. Gailua modu egokian ez badabil testeo panelean, ere ez du ondo funtzionatuko labview-n. Datuen eskuraketan arazoak badaude labview-n, self test eta test panels erabili arazoak aurkitu eta konpontzeko.



✓ <u>Reset</u>

Reset lanabes honekin DAQ gailua ezabatu egiten du, bare hasierako egoerara bueltatuz.

✓ <u>Properties</u>

Properties lanabesarekin, DAQ gailuarekin erabiltzen diren RTSI konfigurazioa eta osagai gailua ikusi eta aldatu daitezke. Sistemaren baliabideak gailurako, memoria amplitudea eta IRQ maila adibidez, MAX-en konfigurazioan ezkerraldean agertutako lehioan Attributes atalean agertzen dira.

✓ <u>Self calibrate</u>

Self calibrate lanabes honek, DAQ gailuaren barne kalibrazio bat egiten du.

Scales



Kategoria honetan, momentu horretan jadanik konfiguratuta dauden eskalak zerrendatzen dira, eta ematen du aukera eskala horiek testeatzeko edo aldatzeko. Eskalak ere DAQ asistentera sartzeko posibilitatea dauka, horrela eskala berriak sortzeko.

✓ DAQ assistant

DAQ asistentearen bitartez eskala berriak sortu daitezke, ondoren erabiliak izango direnak existitzen diren kanal birtualen informazioa akotatzeko. Diseinatu edo sortutako eskala bakoitza bere izen eta deskripzioa izan ditzake, errazagoa izateko identifikatzeko. Eskala bat ondorengo mota batekoa izan daiteke:

- Linear: Formula hau erabiltzen duten eskalak dira, y=mx + b.
- Map Ranges: Eskalak non x eta y ardatzen balioak proportzionalak diren.
- Polinomial: Formula hau erabiltzen duten eskalak, y=a0 + (a1*x) + (a2*x2) + ... + (an*xn)
- Table: Eskala non taula batean sartu egiten da x eta y ardatzen balioa.

Software

Kategoria honetan, instalatuta dagoen NI software guztia agertuko da. Kategoria honen barnean **Software Update Agent** atala ere agertuko da, zeinen xedea instalatutako NI softwarea azkenengo bertsioa den aztertzea da. Instalatutako softwarea azkenengo bertsioa ez bada, **Software Update Agent**-ek *ni.com* web orrialdea irekitzen du azkenengo bertsio hori eskuratu ahal izateko.



I. <u>NI-DAQmx vi-en orokortasunak:</u>

✓ DAQ mx izen kontrolak

DAQmx name controls paletak, lanentzako izenak, kanalentzako izenak, kanal fisikoak, terminalak, eskalentzako izenak, gailuen zenbakia, eta etengailu kontrolak ditu. Normalean lanak eta kanalak MAX-ean konfiguratu ohi direnez, labview-n egin beharrean, gehien erabiltzen direnak lanentzako izena eta kanalentzako izena kontrolak dira. Kanal fisikoaren, terminalaren, eskalaren, eta gailu zenbakiaren balioak MAX-ean konfiguraten dira, lan edo kanal bat sortu edo konfiguratzen denean. Konfiguratutako lan edo kanalak zuzenean sartu egiten dira kanal eta lanen zerrendara.

✓ DAQmx datu eskuraketa VI



NI-DAQmx VI-ak erabiltzen dira NI-DAQ hardware gailuekin instrumentazioa, eskuraketa eta, kontrola aurrera eramateko.

DAQmx datu eskuratze paletak ondorengoak barne ditu:

- ✓ <u>Constants</u>
 - **DAQmx lan izen konstantea**: Zerrendatu egiten ditu DAQ asistentearen bidez sortutako eta gordetako lan guztiak. Konstantean ezkerreko botoiaz sakatuz **I/O Name Filtering** aukeratu, horrela lana limitatu egiten da konstanteak ikustarazten duenera, eta ere limitatu egiten da konstantean jarri daitekeena.



• **DAQmx kanal globaleko konstantea**: Zerrendatu egiten ditu DAQ assistentearekin sortutako eta gordetako lan guztiak. Browse aukeraren bidez dauden kanal guztien artean aukeratu daiteke. Ezkerreko botoia sakatuz **I/O Name Filtering** aukeratu, horrela kanala limitatu egiten da konstanteak ikustarazten duenera, eta ere limitatu egiten da konstantean jarri daitekeena.



- ✓ <u>VIs</u>
 - DAQmx create virtual channel VI: Kanal birtualak sortu edo martxan jartzen ditu, eta lan batean sartzen ditu. VI honek, kanalaren I/O mota (sarrera analogikoa, irteera digitala, edo kontadore irteera adibidez), adierazteko neurketa edo sorkuntza (tenperatura neurketa, tentsio sorgailua, edo lan kontadorea adibidez), eta kasu batzuetan, erabilitako sentsorea (termoparea, edo RTD adibidez), definitzen dute. Funtzioak, MAX-en kanalak sortzen direnean egindako konfigurazio berdina egiten du. Funtzio hau erabiltzen da, programaren berezitasunengatik kanal fisikoen konekzioak aldatu behar direnean, baina ez beste parametrorik, adibidez terminalen konfigurazio mota, edo aplikatutako eskala. Kanal fisikoaren menua erabili DAQ gaiaren zenbaki gailua adierazteko, eta seinaleak konektatuta dagoen kanal fisikoa finkatzeko.



• **DAQmx Read VI:** Adierazitako kanal edo lanaren laginak irakurtzen ditu. VI hau konfiguratu daiteke erabiltzaileak egin nahi duen irakurketaren arabera, beraz, aukeratu daiteke: bueltatutako laginaren formatua, lagin bakarra irakurtzea momentu bakoitzean edo lagin asko, eta kanal bakarretik edo askotik irakurri behar den. Konfigurazioa aukeratu daiteke objektuaren gainean ezkerreko botoia sakatuz eta **Select Type** aukeraren bitartez.



• **DAQmx Write VI:** Adierazitako kanal edo lanaren laginak idazten ditu. VI hau konfiguratu daiteke erabiltzaileak egin nahi duen irakurketaren arabera, beraz, aukeratu daiteke: idatzi beharreko laginaren formatUa, lagin bakarra idaztea momentu bakoitzean edo lagin asko, eta kanal bakarretik edo askotik idatzi behar den. Konfigurazioa aukeratu daiteke objeKtuaren gainean ezkerreko botoia sakatuz eta **Select Type** aukeraren bitartez.



• **DAQmx Wait Until Done VI:** Itxaroten du neurketa edo sorkuntza bukatu arte. VI mota hau erabiltzen da ziurtatzeko eragiketa bukatuta dagoela lana gelditu baino lehen.



• **DAQmx Timing VI:** Eskuratu edo sortu beharreko lagin kopurua konfiguratu egiten du, eta beharrezkoa denean buffer bat sortzen du. Konfigurazio hau, lanean erabilitako tenporizazio motaren araberakoa da.



• **DAQmx Trigger VI:** Trigger funtzioa konfiguratzen du edozein lanerako. Konfigurazioa, lana eta trigger motaren araberakoa izango da. VI mota hauetan, ez da ezer gertatuko konfiguratutako gertakari bat gertatu arte.



• **DAQmx Start Task VI:** Lana hasieratzen du, neurketa edo sorketa berriro hasteko. VI mota hau beharrezkoa da hainbat aplikazioetarako, eta beste batzuetan aukerazkoa da.



• **DAQmx Stop Task VI:** Lana gelditzen du eta Start Task erabili baino lehenagoko egoerara bueltatzen du.



• **DAQmx Clear Task VI:** Lana garbitzen du. Garbitu baino lehen, beharrezkoa bada, lana gelditzen du. Lan bat garbitu ostean, ezin da berriro erabili, egin daitekeen bakarra da lana berriro sortzea.



- ✓ <u>Property Nodes</u>
 - DAQmx Channel Property Node: DAQmx kanal mota aukeratuta dagoeneko ezaugarri multzoa. Ezkerreko botoia sakatuz, Property Node-an Select Filter aukeratu, horrela soilik agertuko dira sisteman instalatutako gailuaren ezaugarriak.



• **DAQmx Timing Property Node:** DAQmx tenporizazio mota aukeratuta dagoeneko ezaugarri multzoa. Ezkerreko botoia sakatuz, Property Node-an **Select Filter** aukeratu, horrera soilik agertuko dira sisteman instalatutako gailuaren ezaugarriak.



• DAQmx Trigger Property Node: DAQmx trigger mota aukeratuta dagoeneko ezaugarri multzoa. Ezkerreko botoia sakatuz, Property Node-an Select Filter aukeratu, horrera soilik agertuko dira sisteman instalatutako gailuaren ezaugarriak.

Pro	perty Node	
В ?!	DAQmx Trigger	D ?!
۲	Start.TrigType	

• **DAQmx Read Property Node:** DAQmx irakurri mota aukeratuta dagoeneko ezaugarri multzoa. Ezkerreko botoia sakatuz, Property Node-an **Select Filter** aukeratu, horrela soilik agertuko dira sisteman instalatutako gailuaren ezaugarriak.

Pr	operty Node	
Б ?!	➡ DAQmx Read	6 ?!
۲	RelativeTo	

• **DAQmx Write Property Node:** DAQmx idatzi mota aukeratuta dagoeneko ezaugarri multzoa. Ezkerreko botoia sakatuz, Property Node-an **Select Filter** aukeratu, horrela soilik agertuko dira sisteman instalatutako gailuaren ezaugarriak.

Pr	operty Node
D ?!	🗝 DAQmx Write 🦉
۲	RelativeTo

- ✓ DAQ assistant
 - **DAQ Assistant Express VI:** NI-DAQmx erabiliz lanak sortu, konfiguratu, eta martxan jartzen ditu.



✓ <u>SUBPALETTES</u>

• **DAQmx Real Time:** Denbora errealeko eragiketak konfiguratu eta adierazteko.



- **DAQmx Device Configuration:** hardware konfigurazio eta kontrolerako.
- **DAQmx Advanced Task Options:** Lanen konfigurazio eta kontrol aurreratuentzako.
- **DAQmx Advanced:** NI-DAQmx-en gai aurreratuetara sartzeko.

10.GAIA:SARRERA ANALOGIKOA

Gai honi dagozkion atalak ondorengoak dira:

- A. Buffer sarrera analogikoa.B. Eskuraketa jarraiaren fluxu diagrama.

A. Puntu askotako (buffer) sarrera analogikoa:

Puntu asko eskuratzeko momentu berean, DAQ Read VI-ren barnean lagin asko irakurtzeko aukera jarri behar da. DAQ Read VI konbinatu egingo da DAQ Timing VI, DAQ Start Task VI, eta DAQ Stop Task VI-rekin tenporizazio hardware buffer eskuraketa VI bat sortzeko.

- ✓ <u>Tenporizazio hardware eskuraketa</u>: Hardware seinale batek, laginketa erlojua edo AI bihurketa erlojua adibidez, eskuraketaren erritmoa (maiztasuna) finkatu egiten du. Hardware erloju bat software loop bat baino azkarragoa da, beraz, laginen eskuraketa maiztasuna handiagoa izan daiteke. Hardware erloju bat gainera, software loop bat baino zehatzagoa da. Software loop baten erritmoa aldatuta ikusi daiteke beste eragin batzuengatik, adibidez, ordenagailuan beste programa bat irekitzeak, baina hardware erlojua beti konstante mantenduko da.
- ✓ <u>Buffer eskuraketa</u>: Lagin asko eskuratzen ditu gailuaren dei bakarrarekin. Informazioa, bitarteko buffer batera pasatuko dira labview-ra heldu baino lehen.

DAQmx timing VI

DAQmx Timing VI-ak laginak eskuratzeko maiztasuna eta eskuratu edo sortu beharreko lagin kopurua konfiguratzen du, eta beharrezkoa denean buffer bat sortzen du. Konfigurazio hau lanean erabiltzen den tenporizazio motaren araberakoa izango da. Eskuragarri dauden tenporizazio aukerak: laginketa erlojua, Handshaking, Implicit, Use Waveform, eta Change Detection.

Laginketa erlojuak ondorengo parametroak ditu konfiguratzeko:

- ✓ <u>Sample mode</u>: Definitzen da laginketa infinituki egiten den, edo denbora finitu baten bitartean.
- ✓ <u>Samples per channel</u>: Finkatu egiten du sarrera edo irteera lagin kopurua, "sample mode finite mode" baldin bada. Balio honek definituko du bitarteko bufferraren tamaina, datuen gordeketa egiten duena informazioa labview-ra pasatu baino lehen.
- ✓ <u>Rate</u>: Laginketa erritmoa finkatzen du, laginketa segundoko unitateaz. Laginketarako kanpo erloju bat erabiltzen bada, balio hau erloju horretaz espero daitekeen maximoarekin hasieratuko da.
- ✓ <u>Source</u>: Laginketa erlojuaren iturri terminala finkatzen du. Gailuaren barne erlojua erabiltzen bada, terminal haun konektatu gabe geratuko da.
- ✓ <u>Active edge</u>: Erlojuaren zein flankoarekin gertatuko den lagin neurketa definitzen du. Goranzko edo beheranzko flankoaren artean aukeratu beharko du erabiltzaileak.
- ✓ <u>Task/Channels in</u>: Lanaren izena edo kanalek zerrenda finkatzen du. Kanal zerrenda bat definitzen bada, NI-DAQmx-ak zuzenean lan bat eratuko du.
Handshaking parametroak, gailua eta kanpo gailua arteko lotura bat egotekotan eskuratu edo sortu beharreko lagin kopurua definitzen du.

Implicit parametroak, eskuratu edo sortu beharreko lagin kopurua definitzen du tenporizarioa aipatu gabe. Normalean parametro hau definitu egiten da, lanean tenporizaziorik behar ez denean, adibidez, lanak non kontadoreak erabiltzen dira neurketa frekuentzia definitzeko, edo buffer bidez kontrolatzen direnean neurketak, edo pultsu trenak sortzerakoan.

Waveform parametroak, waveform sarreraren dt parametroa erabiltzen du laginketa erlojuaren erritmoa (maiztasuna) definitzeko. Dt parametroak laginen arteko denbora adierazten du, segundutan. "Sample node finite samples" bada, NI-DAQmx-ak adierazitako lagin kopuruko grafikoa sortuko du. VI honek ez du inolako laginik kanporatzen, beharrezkoa da DAQmx Write VI batera kableatzea lagina eratzeko.

Change detection parametroak, lana konfiguratzen du datuen eskuraketa aukeratutako portu edo lerroaren goranzko edo beheranzko flankoarekin izateko. Portu edo lerro baten bai goranzko zein beheranzko flankoak detektatzeko, portu edo lerro horren izena kableatu beharko da goranzko flanko kanal fisikora eta baheranzko flanko kanal fisikora.

Buffer eskuraketaren fluxu diagrama

Ondorengo fuxu diagramak era erraz batean adierazi egiten du buffer eskuraketa. Buffer eskuraketan lagin kopuru zehatz bat eskuratu egiten da maiztasun jakin batean. DAQmx Timing VI erabili egiten da tenporizazioa eta buffer-a konfiguratzeko. DAQmx Start Task VI-a erabili eskuraketa hasieratzeko. DAQmx Read VI-ak itxaroten du kanal bakoitzeko lagin guztiak eskuratuta egon arte datua emateko. DAQmx Stop Task VI-ak lana gelditzen du eta, gailura lotuta dauden baliabideak askatzen ditu. Error Handle VI-ak prozesuan gertatutako arazoren bat ikustarazten du.



Buffer eskuraketaren adibidea

Ondorengo adibideak erakusten du nola sortu buffer eskuraketa VI bat. DAQmx Timing VI-ak lana/kanala, tenporizazioa eta, kanaleko laginak (buffer tamaina), finkatzen du. Ondoren, DAQmx Start Task VI-ak eskuraketa hasieratzen du. Programak itxarongo du DAQmx Read VI-an buffer-a betetzen den arte. Buffer-a beteta dagoenean, DAQmx Read VI-ak datua emango du buffer-etik. DAQmx Stop VI-ak eskuraketa gelditzen du, eta DAQmx Error Handle VI-ak errorerik egotekotan, errorea ikustarazten du.



DAQmx Read VI-ren **Number Of Samples Per Channel** sarrera kableatu gabe dagoenez, NI-DAQmx-ak automatikoki zehaztu egiten ditu irakurri beharrekolagin kopurua, DAQmx Timing VI-an konfiguratutakoaren arabera. NI-DAQmx-ak automatikoki bere balio zehazten du, eta **Number Of Samples Per Channel** sarreran balio hori -1 bezala finkatzen da. DAQmx Read VI-ak bi dimentsioko array bat bueltatzen du, grafiko batera kableatuta izan daitekeena. Array-ak ez du tenporizazio informaziorik ematen.

Azkenik, error cluster-a beti kableatu behar da. Error in sarrerak errore bat detektatzen badu DAQmx Start Task VI, DAQmx Read VI, edo DAQmx Stop Task VI-an, error out irteeratik errorearen informazioa emango du eta, gurutzapena geldituko du.

Zer da benetan gertatzen dena?

Benetan ulertzeko zer gertazen den eta nola funtzionatzen duen buffer eskuraketa batek, oinarrizko maila batera jaistea da egokiena, ondorengo irudian ikusten den bezala:



Jakina da datuak eskuratzen direnean, informazioa instrumentazio anplifikadoretik ADC-ra pasatzen dela. Baina beharbada ezagutzen ez dena da zer gertatzen den seinale horrekin ondoren. Datuak **First In First Out** (FIFO) buffer batera pasatzen dira, non gordeta gelditzen dira ordenagailura pasatu arte. Ondoren datuak pasatu egiten dira gailutik ordenagailuaren buffer batera **Direct Memory Access** (DMA), edo **Imterrupt Request** (IRQ) bidez.

Ordenagailuaren buffer-ak datuak gordetzen ditu gailuatik irtetzen direnean. DAQmx Timing VI-aren **Number Of Samples Per Channel** sarrerak (edo DAQmx Configure Input Buffer VI-aren **buffer size** sarrerak), PC buffer-a konfiguratu egiten du, datuak gordeko dituenak DAQmx Read VI-ak berreskuratzeko prest egon arte. DAQmx Read VI-ak datuak Labview-ko buffer batera transferituko ditu, horrela, panel frontalak ikusi ahal izango dira. Labview buffer=ak datuak era ezberdinetan adierazi ditzake: grafiko batean, array batean... DAQmx Read VI-aren konfigurazioaren arabera, eta bere irteeraren kableaketaren arabera.

Buffer transferentzia

PC buffer eta Labview buffer arteko datuen transferentzia oso garrantzitsua da sarrera analogiko eragiketetan. DAQmx Timing VI-aren **Number Of Samples Per Channel** sarrerak PC buffer-a esleitzen du. Buffer eskuraketa bat egiteko, eskuraketa hasiko da DAQmx Start Task VI-a deitzerakoan. Eskuraketa hasterakoan, PC buffer-a datuaz betetzen hasten da, eta betetan egon arte datuen eskuraketa jarraituko du. PC buffer-a zein erritmoan beteko den, DAQmx Timing VI-an konfiguratutako erritmoak (maiztasuna) definituko du. Buffer-a beteta dagoenean, DAQmx Read VI-ak datuak transferituko ditu PC buffer-etik Labview buffer-era.

B. Eskuraketa jarraiaren fuxu diagrama:

Oinarrizko ezberdintasuna buffer eskuraketa eta eskuraketa jarraiaren artean, eskuratutako puntu (lagin) kopurua da. Buffer eskuraketarekin, definitutako lagin kopuru finitu bat eskuratu egiten da; eskuraketa jarraiarekin, ordea, infinitu lagin eskuratu daitezke. Ondorengo fluxu diagraman eskuraketa jarraia ikusi daiteke:



Lehenego hiru pausuak berdinak dira buffer eskuraketan eta eskuraketa jarraian. DAQmx Timing VI-arekin gailuaren konfigurazioa egiten da, DAQmx Start Task VIarekin eskuraketa hasieratzen da, eta DAQmx Read VI-arekin datuen irakurketa prestatzen da. Datuak etengabe eskuratzen direnez, beharrezkoa da datuak etengabe irakurtzea. Beraz, DAQmx Read VI-a loop baten barnean egongo da. Loop-a bukatu egingo da errore bat gertatzerakoan edo, panel frontaletik erabiltzaileak gelditzerakoan. Loop-a eragiten den bakoitzean, DAQmx Read VI-ak datu bat ematen du. Loop-a amaitzerakoan, DAQmx Stop Task VI-ak lana bukatzen du. Simple Error Handler VI-ak ikustaratzen du prozesuan gertatutako erroreren bat.

Buffer eskuraketa jarraia

Ondorengo irudian buffer eskuraketa jarraiaren bloke diagrama ikusi daiteke, buffer eskuraketaren bloke diagramaren antzekoa dena hurrengo aldaketekin:

- DAQmx Read VI-a While loop baten barnean dago.
- Number Of Samples Per Channel sarrera definituta dago erabiltzaileagatik. Eskuraketa finituan, ordea, NI-DAQmx-ak automatikoki definitzen zituen zenbat lagin irakurri. Number Of Samples Per Channel sarrera kableatu gabe uzten bada, NI-DAQmx-ak buffer-ean eskuragarri dauden lagin kopurua irakurriko du.
- Kanal bakoitzean eskuragarri dauden laginak ikustarazi daitezke (backlog).



Buffer eskuraketa jarraiaren lehenengo pausuan tenporizazioa, laginketa modua, irakurtzeko lagin kopurua kanaleko (buffer), eta erritmoa (maiztasuna) konfiguratu egiten dira DAQmx Timing VI-arekin. DAQmx Start Task VI-ak eskuraketa hasieratzen du. DAQmx Read VI-a, While loop baten barnean dagoena, buffer eskuraketaren datuak irakurtzen ditu.

Buffer-ak gainezka egitea (overflow) ekiditeko, **Number Of Samples Per Channel To Read** ezin da buffer tamaina baino handiagoa izan. Buffer eskuraketa jarraian, egokia izaten da **Number Of Samples Per Channel To Read** sarrera buffer tamainaren laurdena edo erdia izatea. Labview-k etengabe datuak bidaltzen dituenez buffer-era, buffer-ean eskuragarri dauden lagin kopurua kanaleko monitorizatzea garrantzitsua da, buffer-a era egokian betetzen edo husten dela ziurtatzeko.

Eskuragarri dauden lagin kopurua kanaleko oso azkar handitzen bada, buffer-a gainezka egin ditzake (overflow), eta errore bat sortu. While loop-a, non DAQmx Read VI-a dago, gelditu daiteke bai erabiltzaileak bloke diagramako botoia sakatu duelako, edo bai errore bat gertatu delako DAQmx Read VI-an, adibidez, buffer-ak gainezka egitea.

While loop-a gelditu ostean, DAQmx Stop Task VI-ak lana gelditu egiten du, eta Simple Error Handler VI-ak erroreak ikustarazten ditu.

Buffer zirkularra

Buffer eskuraketa jarraia eragiketa zaila da, ordenagailuak buffer bakarra erabiltzen duelako, eta buffer-ak hartu ditzakeen lagin gehiago eskuratzen direlako. Buffer-ak hartu ditzakeen lagin kopurua baino lagin gehiago eskuratzeko, buffer zirkularra erabiltzen da. Ondorengo irudiak buffer zirkular batek nola funtzionatzen duen erakusten du:



Buffer zirkular bat buffer normal baten antzekoa da, baina kasu honetan amaierara heltzerakoan gelditu beharrean berriro hasieratu egiten da. DAQmx Timing VI-aren **samples per channel** sarrerak PC buffer-a definitzen du. DAQmx Start Task VI-a eskuraketa hasieratzen duenean, PC buffer-a datuaz betetzen hasten da. Eskuraketa While loop-aren barnean gertatzen da.

Demagun, Number Of Samples Per Channel To Read sarrera, PC buffer tamainaren laurdenarekin edo erdiarekin definitzen dela. PC buffer-aren lagin kopurua kanaleko, Number Of Samples Per Channel To Read kopuruaren berdina denean, DAQmx Read VI-ak lagin kopuru hori transferitu egingo du PC buffer-etik Labview buffer-era. DAQmx Read VI-ak ertz bat sortzen du Current Sample Position deitua, buffer-aren oraingo posizioa adierazten duena horrela irakurtzen jarraitu daiteke lehen utzitako lekuan.

Bitartean, PC buffer-ak datuaz betetzen jarraitu egiten da. DAQmx Read VI-ak datuak transferitzen jarraitzen du PC buffer-etik Labview buffer-era, PC buffer-a betetzen den bitartean. Datuen amaiera PC buffer-aren amaiera adierazten duenean, datu berria buffer-aren hasieran idatziko da. Lagin bukaera marka eta **Current Sample Position** arteko desberdintasuna, eskuragarri dauden lagin kopurua kanaleko izango da (backlog). Labview-k buffer-etik datuak oso azkar hartu behar ditu, lagin bukaera marka **Current Sample Position** heltzera ekiditeko. Horrela gertzekotan, datu berriak aurreko datuen gainean idatziko dira, eta errore bat sortuko da.

✓ <u>Gain-idaztearen errorea</u>

Buffer zirkular batean agertu daitekeen errore arruntena, gain-idaztearena da. Errore hau gertatzen da lagin bukaera marka **Current Sample Position-**era heltzen denean, eta datuak batzuk besteen gainean idazten dira. Eta honen arrazoia da Labview-k ez dituela datuak beharrezko abiaduraz hartzen PC buffer-etik. Aukera asko daute arazo hau konpontzeko, baina guztiak ezin dira egoera guztietan erabili, eta batzuk besteak baino hobeagoak dira:

- Kanaleko lagin kopurua (buffer tamaina) handitu DAQmx Timing VI-aren bidez. Baina buffer tamaina handitzeak ez du arazoa konponduko buffer-a ez bada azkar hustutzen. Gainera gogoratu behar da, egokiena dela lagin kopurua kanaleko buffer tamainaren laurdena edo erdia izatea. Buffer tamaina handitzea konponbide bat izan daiteke azkenengo hau betetzen bada.
- Buffer-a azkarrago hustutzea **Number Of Samples Per Channel To Read** handituz. Baina Number Of Samples Per Channel To Read ezin da oso handia jarri, bestela DAQmx Read VI-an asko itxaron beharko da buffer-aren kanaleko lagin kopurua Number Of Samples Per Channel To Read berdindu arte. Itxaroten den denbora laginak buffer-a bete arte, erabilia izan daiteke buffer-a hustutzeko.
- DAQmx Timing VI-an kanaleko laginen erritmoa jaitsi. Horrela, laginen bidalketa abiadura buffer-era murrizten da, baina, ez da oso erabilgarria izango erritmo (maiztasuna) jakin bat lortu nahi denean.
- Ekiditu loop-aren abiadura motela izatea beharrezkoak ez diren eragiketa edo analisiak eginez.

✓ Gainezka (overflow) errorea

Buffer eskuraketa jarraian aurkitu daitekeen beste errore bat gailuaren FIFO bufferak gainezka egitea da. Errore hau ez da hain arrunta gain-idazte errorea bezala, eta ez da hain erraza konpontzeko. Arazoa gertatzen da FIFO buffer-a ez denean behar beste azkar hustutzen. FIFO buffer-ak DMA edo IRQ bidez pasatuko du informazioa FIFO buffer-etik PC buffer-era. FIFO buffer-ak oso geldo hustutzen bada, orduan arazoa konpontzeko aukerak gutxi dira:

> • Ziurtatu DMA erabiltzen dela DMA aukera eskuragarri dagoenean. DMA IRQ baino azkarragoa da, eta transferentzia hobeagoa izan daiteke.

- DAQmx Timing VI-an kanaleko laginen erritmoa jaitsi.
- Aukeratu gailu bat FIFO buffer handiagoa duena. Hala ere, konponbide hau arazoa desagertu baino handitu dezake.
- Aukeratu ordenagailu bat bus azkarrago batekin, datuak azkarrago transferitzeko FIFO buffer-eti PC buffer-era.

11.GAIA:IRTEERA ANALOGIKOA

Gai honi dagozkion atalak ondorengoak dira:

- A. Buffer AO VI-ak.
- B. Buffer sorkuntza finitua.
- C. Buffer sorkuntza jarraia.

A. Buffer AO (Analog Output)VI-ak:

Irteera analogikoko lagin asko sortzeko, DAQmx Write VI-a lagin askotarako konfiguratu behar da. Lagin askotako sorkuntza erabili ohi da, denboran aldatzen den seinale bat sortzeko, adibidez, korronte alternoko sinu uhina. Buffer irteera analogikoa finitua edo jarraia izan daiteke. Bi buffer prozedurak ondorengo pausuak jarraitzen dituzte:

- 1. Buffer-ean laginak idatzi. Puntuak (laginak) Labview-tik hartzen dira eta bitarteko memoria buffer batean jartzen dira gailura bidali baino lehen. Buffer sorkuntza email oso bat aldi berean bidaltzearen antzekoa da, hizki bakoitza banan banan bidali beharrean.
- 2. Laginak transferitu buffer-etik DAQ gailura. Laginak transferitzeko erritmoa tenporizazio konfigurazioaren araberakoa da. Lagin bakarreko sorkuntzan bezala, software zein hardware tenporizazioa egin daiteke.

Hardware tenporizazioko sorkuntza batean, hardware seinale bat eguneraketa erlojua deitua, sorkuntzaren erritmoa jartzen du. Hardware erloju bat software loop bat baino azkarragoa izan daiteke, beraz, frekuentzia eta mota gehiagoko seinaleak sortu daitezke. Gainera hardware erloju bat zehatzagoa da software loop bat baino. Software loop batean kanpoko beste gai batzuk egin dezakete erritmoa aldatzea, adibidez beste programa baten irekiera; hardware erloju bat ordea, konstantea da.

B. <u>Buffer sorkuntza finitua:</u>

Ondorengo irudia buffer sorkuntza baten fluxu diagrama ageri da:



Ondorengo irudian buffer sorkuntza finitu baten adibidea erakusten da, laginketa erlojua erabiliz, eta irteera datuak gordetzeko array-arekin.



Waveform datu mota erabili daiteke tenporizazioa eta lagin datua definitzeko, ondorengo irudian bezalaxe:



Ezberdintasun batzuk daude bi buffer sorkuntza finituen artean, laginketa erlojua eta array-a erabiltzen duena; eta waveform datu mota erabiltzen duena laginketa erlojua eta laginak adierazteko. DAQmx Timing VI-aren konfigurazioa aldatzen da, eta ere, DAQmx Write VI-aren **data** sarreran sartutako datu mota. Baina biak egitura berdina jarraitzen dute.

DAQmx Create Virtual Channel VI erabiltzen da irteera analogikoko kanal birtual bat sortzeko. Kanal edo lan birtuala jadanik sortuta badago, DAQ assistant erabiliz MAXean, VI hau saltatu daiteke eta zuzenean kableatu kanala/lana hurrengo VI-arekin, DAQmx Timing VI.

DAQmx Timing VI-ak bi aukera ditu irteera analogikoekin erabili daitezkeenak: Sample Clock eta Use Waveform. Lagin kopuru finitu bat sortzen denez, **Sample Mode** sarreran finite samples aukera jarri behar da. "Sample Clock" aukeratzen bada, sorkuntza **Rate** (errintmoa, maiztasuna) eta **Number Of Samples** (lagin kopurua) definitu behar dira. Lagin kopuruak buffer tamaina finkatuko du. "Use waveform" aukeratzen bada, waveform (uhina) datua **waveform** sarrerara kableatu. Datu honek erritmoa eta lagin kopurua definituko ditu.

DAQmx Write VI-ak datuak bidaltzen ditu PC buffer-era. Aukeztu daiteke irteerako datuak waveform datuak izatea, edo array batean biltzea. Use waveform auketatu bada

aurreko pausuan, Write VI-an irteera waveform modukoa izatea aukeratu. Timing VI-an kableatutako Waveform berdina kableatu Write VI-aren **data** sarreran. "Sample clock" aukeratu bada aurreko pausuan, Write VI-an irteera array bat izatea aukeratu. Sortu nahi den array-a kableatu Write VI-aren **data** sarreran.

DAQmx Start Task VI-ak sorkuntza hasieratzen du. DAQmx Wait Until Done VI-ak lana bukatuta egon arte itxaroten du. Lana bukatzen denean, DAQmx Stop Task VI-ak lana gelditzen du. Eta azkenik, aurrekotan ikusi den bezala errore VI bat dago amaieran, errorik gertatzekotan hura ikustarazteko.

Waverform (uhin itxuradun) irteera maiztasuna

Waveform irteeraren maiztasuna, laginketa erritmoa, maiztasuna (update rate) eta buffer-ean uhinak duen ziklo kopuruaren araberakoa da, ondorengo irudian ikusi daitekeen moduan:



Irteera seinalearen maiztasuna kalkulatzeko formula ondorengoa da:

Seinale maiztasuna= (ziklo kopurua*laginketa maiztasuna)/lagin kopurua

ADIBIDEAK:

Lagin kopurua=1000. Ziklo kopurua=1. Laginketa maiztasuna=1 kHz. Seinale maiztasuna= (1*1000)/1000=1 Hz.

Lagin kopurua=1000. Ziklo kopurua=1. Laginketa maiztasuna=2 kHz. Seinale maiztasuna=(1*2000)/1000=2 Hz.

Lagin kopurua=1000. Ziklo kopurua=2. Laginketa maiztasuna=1 kHz. Seinale maiztasuna=(2*1000)/1000=2 Hz.



Ta: seinale analogikoaren periodoa. Ts: laginketa periodoa. Fa: seinale analogikoaren maiztasuna. Fs: laginketa maiztasuna. N: lagin kopurua, buffer tamaina.

Ta=N*Ts. Fa=1/Ta=1/(N*Ts)=Fs/N.

Buffer tamaina=20000. Ziklo kopurua=4. Laginketa maiztasuna=1/Ts=1000 Hz.

Ta= (N/ziklo kopurua)*Ts= (20000/4)*(1/1000)=5. Fa= 1/Ta=(Ziklo kopurua/N)*Fs=1/5=0,2 Hz.

C. <u>Buffer sorkuntza jarraia:</u>

Buffer sorkuntza finitua eta buffer sorkuntza jarraiaren arteko ezberdintasun garrantzitsuena, sortutako lagin kopurua da. Buffer sorkuntza finituan, lagin kopuru zehatza bat sortzen da; buffer sorkuntza jarraian, ordea, lagin kopuru infinitua sortu daiteke.

Ondorengo irudian buffer sorkuntza infinituaren fluxu diagrama aztertu daiteke:



Irudia aurrekoaren antzekoa da hurrengo aldaketekin:

- DAQmx Timing VI-en sample mode sarreran **Continuous Samples** aukera jartzen da.
- DAQmx Is Task Done VI-a erabiltzen da DAQmx Wait Until Done VI-aren ordez While loop-aren barnean.



Lehenengoz kanal birtuala eta tenporizazioak konfiguratu egiten dira, Create Virtual Channel VI eta Timing VI bitartez. Ondoren buffer-ean idazten da Write VI-arekin eta lana hasieratzen da Start Task VI-arekin.

While loop-ak lana testeatzen du DAQmx Is Task Done VI-arekin, jakiteko bukatu duen. Sorkuntza bukatuko da, erabiltzaileak stop botoia sakatzen duenean, edo erroreren bat gertatzerakoan. While loop-a bukatzerakoan, DAQmx Stop Task VI-ak lana gelditzen du, eta errorerik egotekotan ikustaraziko dira.

Gai honi dagozkion zenbait adibide jarraian adierazitakoak dira:

Variable Servo Fan_Beta.vi:

Ondorengo adibidean seinale analogiko bat sortzen da, kanpo txartelaren funtzio batzuk erabiliz.





Lehenengoz, **DAQmx Create Virtual Channel** funtzioarekin, kanal birtual bat sortu egiten da tentsioa sortzeko. Kasu honetan adierazi egiten zaio zein kanal fisikorako sortzen de kanal birtual berria, eta zein izango den kanal berriaren izena. Ondoren, **DAQmx Start Task** funtzioarekin, lana hasieratzen da. **While loop**-aren barnean, **DAQmx Write** funtzioa dago, fitxero batean tentsio seinale analogikoaren datuak gordeko dituena. **While loop**-atik irtetzerakoan, **DAQmx Stop Task** funtzioarekin lana itxi egingo da, eta azkenik **Simple Error Handle** funtzioarekin errorean ikustaraziko dira, erroreren bat gertatzekotan.

Eraikitako VI hau asko gogoratzen du open/create/replace funtziora, non, fitxero bat irekitzen da, bertan eragiketan burutzen dira, eta ixten da.

Voltmeter.vi:

Adibide honetan, kanal fisiko batetik tenperatura neurketa bat egiten da, eta kanal digital bat sortu ostean, tenperatura hau ikustarazten da.





Lehenengoz, **DAQmx Create Virtual Channel** funtzioarekin, kanal fisiko batekiko kanal birtual bat sortzen da. Ondoren, **DAQmx Start Task** funtzioarekin, lana hasieratzen da. **While loop**-aren barnean **DAQmx Write** funtzioa daukagu, kanalaran tenperatura neurketa bat irakurriko duena, eta numeriko batean adierazi. **While loop**-atik irtetzerakoan (erabiltzaileak jarritako tenporizazioa kontutan harturik), **DAQmx Stop Task** funtzioarekin lana bukatu egingo da, eta **Simple Error Handle** funtzioarekin sortutako erroreen ikustaraztea egingo da, inolako errorerik sortzen bada.

Hauxe bera **DAQ** Assistant batekin egitea posiblea da, txarteletik tanperatura, tentsio... neurketa bat egin, DAQmx read funtzioarekin irakurri, eta ikustarazi numeriko edo grafiko batean.