КІРІСПЕ

LabVIEW немесе Laboratory Virtual Instrument Engineering Workbench - зертханалық виртуалды құрылғыларды жасау ортасы.

LabVIEWбағдарламалық ортасы радиотехника, телекоммуникация, электроника және автоматтандыру жүйелерінде өте пайдалы әрі қазіргі таңдаоқу жүйесінде, ғылыми зерттеу жұмыстарында және өндірісте кең қолданысқа ие. Сонымен қатар бұл графикалық бағдарламалау ортасы Windows, MacOS, Linux операциялық жүйелеріне орнатылып, виртуалды құрылғылар арқылы компьютердің кез-келген цифрлық кіріс/шығыстарындағы ақпараттармен жұмыс істеуге мүмкіндік береді. Бұл оқулық LabVIEWбағдарламалық ортасында сандық өңдеу және басқару бағыттары бойынша қазақша оқу топтарына негізгі мағлұмат алуға арналған. Оқулықта.....

Мазмұны

Кіріспе..................................................................................................

1.LabVIEW бағдарламасының негізгі ұғымдары және бағдарламаны іске қосу......................................................................

Лабораториялық жұмыс 1. Виртуалды құрылғы құру....................

Тапсырма 1.LabVIEW инструменттері және мәлімет типтері. LABVIEW негізінде виртуалды құрылғының .EXE форматын жасау.....................................................................................................

2.Виртуалды құрылғымен жұмыс істеу негіздері............................

Лабораториялық жұмыс 2. Арифметикалық амалдарды орындау.

Тапсырма 2.Математикалық формулалармен жұмыс....................

3. Массивтер........................................................................................

Лабораториялық жұмыс 3. Электр тізбегіндегі тоқты анықтау.....

Тапсырма 3. Бір өлшемді массивтермен есептер шығару...............

4.For және While циклдері.................................................................

Лабораториялық жұмыс 4. Автоиндексация. Ығыстыру регистрі. Кері байланыс түйіні. Логистикалық бейнелеу. Бейсызық теңдеулерді шешу...............................................................................

Тапсырма 4. Алдын-ала және кейінгі шарттарымен цикл операторлары.......................................................................................

5. Тізбекті және таңдау операторлары..............................................

Лабораториялық жұмыс 5.Цифрлық есептеу құрылғыларын жасау.....................................................................................................

Тапсырма 5. Шартқа не оқиғаға негізделген опрераторлар............

6. Кластерлер.......................................................................................

Лабораториялық жұмыс 6. Кластерді масштабтау. Радио батырмалар...........................................................................................

Тапсырма 6...........................................................................................

7. Мәліметті графикалық бейнелеу әдістері.....................................

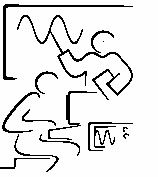
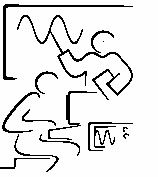
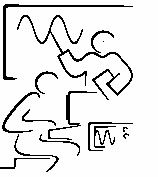
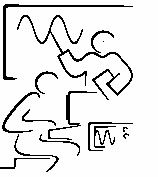
Лабораториялық жұмыс 7..................................................................

Тапсырма 7...........................................................................................

8. Таблицалар. Файлмен жұмыс істеу функциялары......................

Лабораториялық жұмыс 8. Таблицаны іріктеу.................................

Тапсырма 8. Екі өлшемді массивтермен есептер шығару...............

1-бөлім.

**LABVIEW БАҒДАРЛАМАСЫНЫҢ НЕГІЗГІ ҰҒЫМДАРЫ ЖӘНЕ БАҒДАРЛАМАНЫ ІСКЕ ҚОСУ**

1-бөлімдеLabVIEW программалау ортасының негізі ұсынылған. LabVIEW ортасымен жұмыс жасауды бастау үшін ретке келтіру және өңдеу техникасы, программаны іске қосу, палитра типтері, терезелер, терезелер құрылымы, контексті мәзірлержайлы негізгі ақпаратқа ие болу керек.

1.1.**LabVIEW программалау ортасы**

LabVIEW ортасында жазылған программа виртуалды құрылғы (ВҚ) деп аталады. Себебі олар сырттай қарағанда және функционалды түрде нақты (дәстүрлі) құрылғылар іспеттес. Алайда, олар С, Basic, С++ секілді танымал программалау тілдері программасына ұқсайды. Әрі қарай LabVIEW программасын виртуалды құрылғы деп атайтын боламыз. ВҚ нақты қандай да бір физикалық құрылғыларды, мысалы, осциллограф немесе мультиметрді жасап шығарады (симуляциялайды). LabVIEW мәліметтерді сақтау, өңдеу, жинау үшін қажетті барлық құралдар жиынтығынан тұрады.

LabVIEWдегі тұтынушы интерфейсі бұл беттік панель және блог-диаграмма. Беттік панель басқару элементтері (батырмалар, ауыстырып-қосқыштар) көмегімен құрылады. Ал блог-диаграмма басқару элементтерінің көрінісін, бейнесін көрсетеді (графиктер, жарықтық диодтар және т.б.). Осыдан соң ВҚ-ның блог-диаграммасында беттік панельдегі объектілерді басқару үшін функцияны графикалық түрде көрсетуді қолдану арқылы программалау жүзеге асады.

LabVIEW әр түрлі **DAQ** –құрылғыларын программалау, қозғалысты және бейнені басқару жүйесі, **GPIB**, **VXI**, **PXI**, **RS-232** және **RS-485** типті интерфейске ие аппаратты құрылғылар үшін қолданылады. LabVIEW программалау ортасының көмегімен сыртқы құралдармен мәліметті енгізу, талдау, басқару, өлшеу үшін программалық-аппараттық жиынтық жасауға болады. LabVIEW-бұл автономды модульдерді де(**.EXE**) және ортақ қолданылатын динамикалық кітапханаларды да (**.DLL**) жасап шығаратын 32 разрядты компилятор.

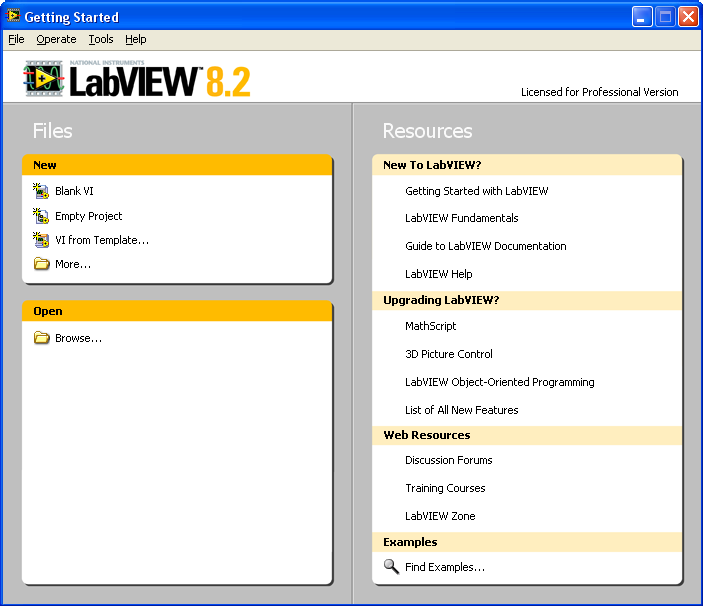
**1.2. LabVIEW программалау ортасын іске қосу**

LabVIEWді іске қосқан сәттежаңа ВҚ құруға болатын жүйе жұмысы үшін іске қосу терезесі (старттық диалогтік терезе) ашылады. Бұл терезедеосы уақытқа дейін құрылып қойылған ВҚларды ашуға, мысалдарды табуға немесе еске түсіру ортасына (LabVIEW Help) қолжетімділікті қалыптастыруды пайдалануға болады. Іске қосу терезесі барлық беттік панельді және блог-диаграммаларды жапқан кезде де пайда болады (1-сурет).Терезе екі бөлікке бөлінген. Терезенің сол жақ бөлігі жаңа виртуалды құрылғы (Blank VI), бос жоба (Empty project), шаблон бойынша ВҚ (VI from Template) құруғамүмкіндік береді. Ал, терезенің оң жақ бөлігі анықтамалық материалдарға қолжетімділікті қамтамасыз етеді.

Іске қосу терезесі келесідей құраушыларға ие:

* Стандартты пунктері бар мәзір панелі, мысалы **File»Exit**
* ВҚны құру және ашу үшін қолданылатын **Files** терезесі. Бұл терезеде **New** және **Open** қосымша беттері орналасқан. **New** қосымша бетін қолдана отырып, сіздер жаңа ВҚ құруыңызға жаңа жоба немесе ВҚ шаблонын (үлгісін) жүктеуіңізге болады. **Open** қосымша бетін қолдана отырып бұрын құрылып қойған ВҚны ашуыңызға да болады.

**Resources** терезесі әр түрлі мәтіндік және Internet ресурстарынан қосымша ақпарат алу үшін, сонымен бірге құрастырылған мысалдарды көруге мүмкіндік береді.



1-сурет .Іске қосу терезесінің (стартты диалогтік терезе) көрінісі

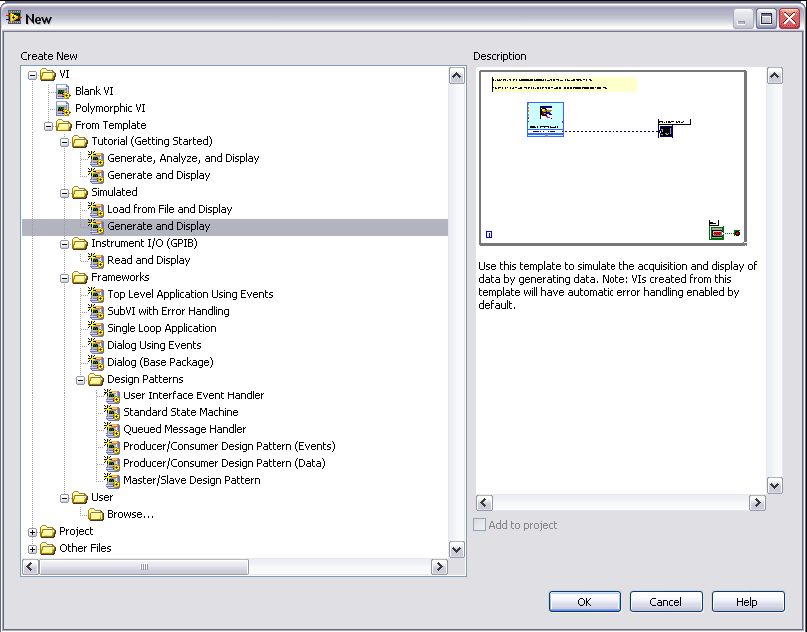
**1.2.1. Жаңа ВҚ құру немесе құрылып қойылған ВҚны ашу**

**New** диалогтік терезесіLabVIEW ортасында қосымшаларды құрастыру кезінде әр түрлі құраушыларды құру үшін қажет. **New** диалогтік терезесі келесідей құраушылардан тұрады:

**VI**- әр түрлі ВҚды құрайды.

**Blank VI** – жаңа беттік панель мен блог-диаграмманы ашады. **New** қосымша бетіндегі **Blank VI** жолын басқаннан кейін экранда беттік панель және бос ВҚның блог-диаграммасы пайда болады.

ВҚ шаблонын ашу үшін **New** қосымша бетінен тышқанның оң жақ батырмасын басып **VI from Template** жолын таңдаңыз. Осыдан кейін экранда **New** диалогтік терезесі ашылады. **Create New (**жаңа ВҚны құру**)** тізімдік терезеден шаблонды таңдап болған соң **Description** секциясында шаблонның сипаттамасы және ВҚ блог-диаграммасы бейнеленеді. ВҚ құру үшін **ОК** батырмасын басыңыз (2-сурет ).



2-сурет. ВҚны құру, **New** қосымша бетінің көрінісі

**Frameworks** – ВҚны құруға қажетті құраушыларды, орындалатын функциялардың арнайы түрлерін құрайтын беттік панель мен блог-диаграмманы ашады.

**Instrument I/O** – компьютерге қосылған сыртқы құралдармен байланыс үшін қажетті құраушыларды құрайтын беттік панель мен блог-диаграмманы ашады.

**Simulated** – құралдардан алынған мәліметтерді модельдеу үшін қажетті құраушылардан тұратын беттік панель мен блог-диаграмманы ашады.

**Tutorial (Getting Started) -** Getting Started нұсқаулығының жаттығуларын орындауға арналған ВҚ құру үшін қажетті құраушыларға ие беттік панель мен блог-диаграмманы ашады.

**User** - бұрын құрылған ВҚның беттік панелі мен блог-диаграммасын ашады.

**Project –**LabVIEWда жоба терезесін ашады.

**Other Files** – кітапханалар, глобальді айнымалыларды, класстарды құруға мүмкіндік береді.

**Description** - ВҚ шаблонындағы **Create New** тізімінен таңдалып алынған блог-диаграмманы бейнелейді.

**1.3. Беттік панель және блог-диаграмма терезесі**

Виртуалды құрылғы негізгі 4 құраушыдан: беттік панель, блог-диаграмма, белгішеден (иконка) және байланыстырғыш панельдерден тұрады.

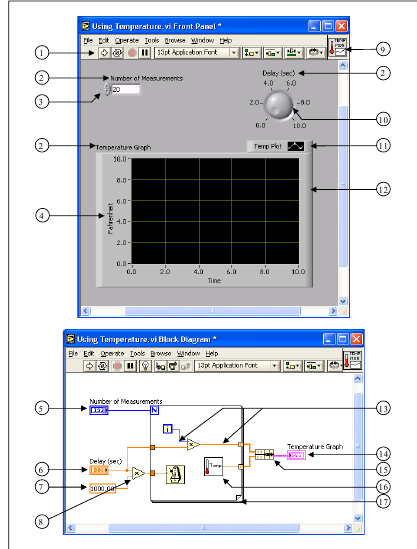
**Blank VI** батырмасын басқаннан кейін беттік панель терезесі ашылады. Беттік панель-бұл ВҚны қолданушы интерфейсі. Беттік панельде болашақ виртуалды құрылғының сыртқы көрінісі жасалынады. Беттік панельдің көрінісі төменде көрсетілген (3-сурет ).

Беттік панель басқару элементтері **(Controls)** палитрасы көмегімен құрылады. Бұл элементтер мәліметтерді енгізу ортасы **(басқару)** немесе мәліметтерді бейнелеу ортасы **(бейнелеу)** болуы мүмкін. **Басқару** элементтері-батырмалар, ауыстырып-қосқыштар, сырғақтар және тағы да басқа мәліметтерді енгізушілерден тұрады. **Бейнелеу** элементтеріне-графиктер, цифрлық тақталар, жарықтық диодтар және т.б. жатады.

ВҚның беттік панеліне енгізілген мәліметтер ВҚның блог-диаграммасына келіп түседі және онда виртуалды құрылғы олармен қажетті амалдарды орындайды. Беттік панельден енгізілген басқару және бейнелеу элементтерінің иконкаларын (белгішелерін) басқару және бейнелеу элементтерінің терминалдары деп атайды. Есептеу нәтижесі ВҚның беттік панеліндегі мәліметті бейнелеу элементіне жіберіледі.

**Басқару** және **Бейнелеу** элементтерін беттік панельге орналастырып болған соң блог-диаграммада олар сәйкесінше өзінің графикалық бейнесіне ие болады.

Келесі терезе ол – блог-диаграмма. Блог-диаграммада болашақ виртуалды құрылғының негізгі коды жасалынады. 3-суретте беттік панель мен блог-диаграмманың әрбір бөлігі ашып көрсетілген.



3-сурет . Беттік панель мен блог-диаграмма көрінісі

1-интрументальді панель; 2-өзіндік белгі; 3-цифрлық басқару элементтері; 4-еркін белгі ; 5-цифрлық басқару элементінің мәліметтер терминалы ; 6-батырмалардың мәліметтер терминалы; 7-сандық тұрақтылар; 8-көбейту функциясы; 9-белгіше (иконка); 10-басқару қаламы; 11-графикті басқару панелі; 12-осциллограммның екікоординатты графигі; 13-мәліметтер сымдары; 14-осциллограмм графигі мәліметінің терминалы; 15-біріктіру функциясы (Bundle); 16-ВҚның ішкі программасы; 17-бекітілген итерациясы бар цикл (For)

Беттік панель мен блог-диаграммадағы бұл бөліктердің өзіндік қызметтері бар. Блог-диаграмма объектілері беттік панель элементтерінің графикалық бейнесінен, амалдарынан, функцияларынан, ВҚның ішкі программасынан, тұрақтылардан, құрылымдардан және блог-диаграммадағы объектілер арасынан мәліметті тарату үшін қолданылатын мәліметтер сымдарынан тұрады. **Байланыстырғыштар** немесе мәлімет сымдары блог-диаграммада басқару және бейнелеу элементтерінің терминалдарын әр түрлі операциялармен, функциялармен байланыстыратын сымдар.

**1.3.1. Беттік панельдегі жабдықтар панелі**

Жабдықтар панелі ВҚны қосу және өңдеу үшін қолданылады. Жабдықтар панелінің көрінісі төменде көрсетілген (4-сурет ).

4-сурет. Беттік панельдегі жабдықтар панелінің көрінісі

1-кесте. Бетік панельдің жабдықтар панеліндегі батырмалар функцияларының сипаттамалары

|  |  |
| --- | --- |
| Батырмалар | Сипаттама |
|  | **Run** іске қосу батырмасы ВҚны іске қосады. |
|  | ВҚмен жұмыс жасау кезінде егер бұл ВҚ жоғары деңгейдегі құрылғы болса **Run** батырмасы сол жақта көрсетілгендей көрініске ие болады. |
|  | Егер ВҚ ішкі программа негізінде жұмыс жасаса, онда **Run**  батырмасы сол жақта көрсетілгендей көрініс табады. |
|  | **Run** батырмасы сынған тіл секілді көрініске ие. Мұндай көріністегі батырма ВҚ іске қосуға жіберілмейді дегенді білдіреді. Бұл батырманы басқаннан соң жіберілген қателіктер тізімін көрсететін **Error list** терезесі ашылады. |
|  | **Run Continuously** үздіксіз іске қосу батырмасы- ВҚны мәжбүрлеп тоқтату моментіне дейін жүзеге асыра береді. |
|  | ВҚ жүзеге асыру кезінде **Abort Execution** батырмасы пайда болады. Бұл батырма ВҚны тез арада тоқтату үшін қолданылады. |
|  | **Pause** батырмасы ВҚның жұмысын тоқтатады. **Pause** батырмасын басқаннан кейінLabVIEWде тоқтату орын алған жерде жарық беріледі. Батырманы қайта басу ВҚ жұмысын қайта жалғастырады. |
|  | **Text Settings-** өлшемін, стилін және түсін қосқандағымәтінді орнататын мәзір. |
|  | **Align Objects** мәзірі объектілерді ось бойынша түзейді (вертикаль бойынша да) |
|  | **Distribute Objects** мәзірі кеңістікте объектілерді түзеуді жүзеге асырады (аралықтар, сығу). |
|  | **Resize Objects** мәзірі беттік панельдегі бірнеше рет қолданылатын объектілерді бір өлшемге алып келеді. |
|  | **Reorder** мәзірі бір-біріне қосылған бірнеше объектілермен жұмыс кезінде қолданылады. Орын алмастыру элементі көмегімен объектілер біреуін белгілеп, Reorder мәзірінен беттік панельдегі оның бейнелену реттілігін таңдау керек**.** |
|  | **Context Help** батырмасы **Context Help** терезесін экранға шығарады. |



**Ескерту.** Программаны тоқтату үшін мүмкіндігінше **Abort Execution**батырмасын қолданбаған жөн. ВҚның нәтижені көрсетуін күту керек немесе программалық жолмен тоқтатуға болады. Мысалы, беттік панельге ВҚны тоқтататын батырманы орналастырып қоюға болады.

**1.3.2. Блог-диаграмманың жабдықтар панелі**

Блог-диаграммада ВҚ іске қосқан сәтте төменде көрсетілгендей

****жабдықтар панелі пайда болады:

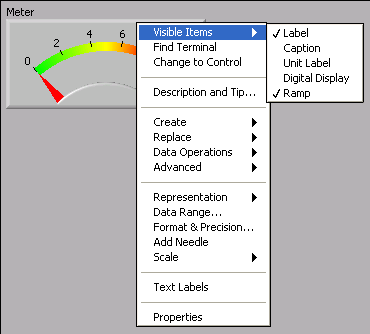
5-сурет . Блог-диаграмманың жабдықтар панелі

2-кесте. Блог-диаграмманың жабдықтар панеліндегі батырмалар функцияларының сипаттамалары

|  |  |
| --- | --- |
| Батырмалар | Сипаттамалары |
|  | **HighlightExecution** батырмасы блог-диаграмма арқылы жүретін мәліметтер ағынын көруге қажет (ретке келтіру немесе дұрыстау режимі). Батырманы екінші рет басу арқылы қызметі тоқтатылады. |
|  | **Retain Wire Values** батырмасы осы моментке дейін сымдар бойынша таратылған мәліметтерді сақтау үшін қолданылады. Бұл батырманы өшіріп, кез келген сым үшін кез келген уақытта сымдар арқылы таратылған мәліметтерді қарауға мүмкіндік бар. |
|  | **Step Into** батырмасы бір түйіннен екінші түйінге дейін рет бойынша орындалатын циклдар, ВҚның ішкі программалары және т.б. үшін қолданылады. Бұл ретте іске қосылуға дайын екендігін көрсету үшін түйіндер жанып –өшіп тұрады. |
|  | **Step Over** батырмасы реттілік циклдарын, ішкі программаны өткізіп жіберу үшін қолданылады. |
|  | **Step Out** батырмасы циклдан, ішкі программадан шығу үшін қолданылады. Түйіннен шығу реттілік режиміндегі түйіннің аяқталғандығын және келесіге ауысқандығын көрсетеді. |
|  | **Warning** батырмасы блог-диаграммада қандай да бір қате пайда болғанда шығады, тек мұның программаның әрі қарай орындалуына кедергісі болмайды. |

**1.4. Контексті мәзір**

Контексті мәзір өте көп қолданылады. LabVIEWдің барлық объектілері беттік панельде, блог-диаграммада да өздерінің контексті мәзіріне ие болады. Констексті мәзір беттік панель мен блог-диаграмманың объектілерінің беталысын өзгерту үшін қолданылады. Оны шақыру амалы: беттік панельде немесе блог-диаграммада объекті үстіне меңзерді апарып, тышқанның оң жақ батырмасын басу арқылыжүзеге асырылады. Контексті мәзірдің мысалы төменде көрсетілген (6-сурет).



6-сурет. Контексті мәзір көрінісі

**1.5. Негізгі мәзір**

ВҚның жоғары бөлігіндегі негізгі мәзір мынадай: Open, Save, Copy, Paste және басқа да қосымшаларға ортақ пунктерді құрайды және LabVIEW мәзіріндегі арнайы пунктерге де ие. Негізгі мәзірдің кейбір пунктері осы пунктерді шақыратын «ыстық» батырмалар жайлы ақпаратқа ие (MacOS). Мәзір экранның жоғары бөлігінде пайда болады.

**Назар аударыңыз.**ВҚны жасау кезінде негізгі мәзірдің кейбір пунктері қолжетімсіз.

Негізгі мәзірді құраушылар:

* **File** мәзір пункті ВҚны ашу, жабу, сақтау және басып шығару үшін қолданылады.
* **Edit** мәзір пункті ВҚның құраушыларына өзгерістер енгізу және іздеу жүргізу үшін қолданылады.
* **View** мәзір пункті LabVIEWде жұмыс жасауға мүмкіндік беретін әр түрлі терезелерді ашу үшін және ВҚ иерахиясында әр түрлі палитралар бейнесін шығару үшін қолданылады.
* **Project** мәзір пункті жобаның файлдық жүйесімен жұмыс жасау үшін қолданылады.
* **Operate** мәзір пункті ВҚны іске қосу, тоқтату, басқа да опциялар үшін қолданылады.
* **Tools** мәзір пункті құралдармен, DAQ құрылғыларымен байланыс үшін, ВҚны салыстыру, қосымшаны құру, LabVIEW конфигурациялары үшін қолданылады.
* **Window** мәзір пункті LabVIEW терезелерінің бейнесін, палитраларын көрсету үшін қолданылады.
* **Help** мәзір пункті палитралар, мәзірлер, жабдықтар, ВҚ және функциялар жайлы ақпараттар алу үшін қолданылады.

**1.6. Палитралар**

LabVIEW ВҚны құру, іске қосу үшін қажетті 3 қосымша палитрадан тұрады: **Tools Palette (Жабдықтар палитрасы)**, **Controls Palette (Элементтер палитрасы), Functions Palette (Функция палитрасы).** Бұл палитраларды экранның кез келген жеріне орналастыруға болады.

**1.6.1. Жабдықтар палитрасы**

ВҚ-ны жасау, өңдеу және күйін келтіру үшін **Tools Palette** (жабдықтар палитрасы) палитрасын қолданамыз. Жабдықтар палитрасын басты бетте және блог-диаграммада қолдана аламыз. Жабдық термині тышқан меңзерінің арнайы операциялық режимін жобалайды (тұспалдайды). Қандай да бір құралды таңдағанда меңзер белгісі сол таңдап алған құрал белгісіне ауысады. Жабдықтар палитрасы негізгі мәзір арқылы **Window»Show Tools Palette** пунктерін орындаған кезде қолжетімді. Жабдықтар палитрасын беттік панель мен блог-диаграмманың кез келген жеріне орналастыруға болады.

Ескертпе. **Tools Palette** жабдықтар палитрасының уақытша нұсқасынбір уақытта **<Shift>** батырмасын және тышқанның оң жақ батырмасын басып экранға шығаруға болады. Жабдықтар палитрасының LabVIEWдегі көрінісі төменде көрсетілген (7-сурет ).



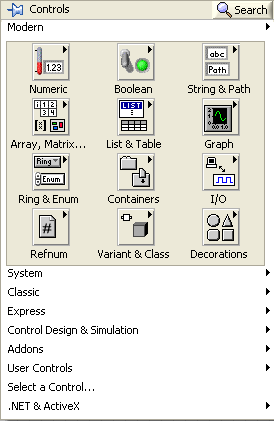
7-сурет . Жабдықтар палитрасының (Tools Palette) LabVieW ортасындағы графикалық бейнесі

3-кесте. Жабдықтар палитрасындағы батырмалар сипаттамалары

|  |  |
| --- | --- |
| Батырмалар | Сипаттамалары |
|  | Егер құралды өздігінен таңдау режимі қосылып тұрса, онда меңзерді басты беттегі немесе блог-диаграммадағы нысанға әкелсек LabVIEW өздігінен **Tools** палитрасынан керекті құрылғыны тандайды. Құрылғыны өздігінен тандау **Жабдықтар (Tools)** палитрасындағы **Automatic Tool Selection** батырмасын басқанда немесе***<Shift-Tab>*** батырмасын басқанда қосылады |
|  | **Басқару** жабдығы мәтінді енгізу немесе басқару элементтерінің мәнін өзгерту үшін қолданылады. Мысалы жолдық басқару элементіне меңзерді апарған кезде жабдықбелгісі мына түрге ауысады: |
|  | **Орын алмастыру жабдығы**объектілерді таңдау, орнын ауыстыру немесе көлемдерді өзгерту үшін қолданылады.  Көлемі өзгеретін объектіге құралды дәлдегенде құралдың белгісі мына белгіге өзгереді: |
|  | **Мәтінді енгізу** жабдығы мәтінді өзгерту үшін және бос белгіні құру үшін қолданылады. Бос белгіні құру кезінде жабдық белгісі мына белгіге ауысады: . |
|  | **Байланыстыру** жабдығы блог-диаграммадағы объектілерді байланыстыра отырып, мәліметтер сымдарын құрады. |
|  | **Контексті мәзірді шақыру** жабдығы сәйкес келетін объекті үшін тышқанның сол жақ батырмасын басу арқылы контексті мәзірді шақырады. |
|  | **Экранды жылдам айналдыру** жабдығы терезені көру үшін айналдыру жолағын қолданбай ақ көруге мүмкіндік береді. |
|  | **Бақылау нүктесін енгізу** құралы бақылау нүктелерін ВҚ, функцияға, түйінге, мәліметтер қорына, құрамына орналастыруға және ішіндегі программалардың орындалуын тоқтатуға мүмкіндік береді. |
|  | **Ретке келтіру индикаторын орналастыру** жабдығы блог-диаграммадағы мәліметтер ағынын зерттеуге мүмкіндік береді. ВҚдағы жұмыс нәтижелері күмәнді немесе күтпеген мән болған кездегі аралық мәндерді көру үшін қолданылады. |
|  | **Түстерді көшіру** жабдығы Бояу құралы арқылы түстерді көшіру үшін қолданылады. |
|  | **Бояу** жабдығы объектінің түсін өзгертуге мүмкіндік береді. Ол сонымен бірге сол мезеттегі алдыңғы жоспарды және фонның түстерін күйге келтіру параметрлерін бейнелейді. |

**1.6.2. Элементтер палитрасы**

Элементтер палитрасы беттік панельге басқару және бейнелеу элементтерін шығару үшін қолданылады. Ол тек беттік панель үшін ғана қолжетімді. Элементтер палитрасын шығару үшін негізгі мәзірден **Window»Show Controls Palette** пунктерін немесе беттік панельдің жұмыс бетінде тышқанның оң жақ батырмасын басу арқылы шақыруға болады. Палитраның сол жақ жоғары бұрышындағы батырманы қолдана отырып оны экранда сақтап қалуға болады. Элемент палитрасының көрінісі төменде бейнеленген (8-сурет).

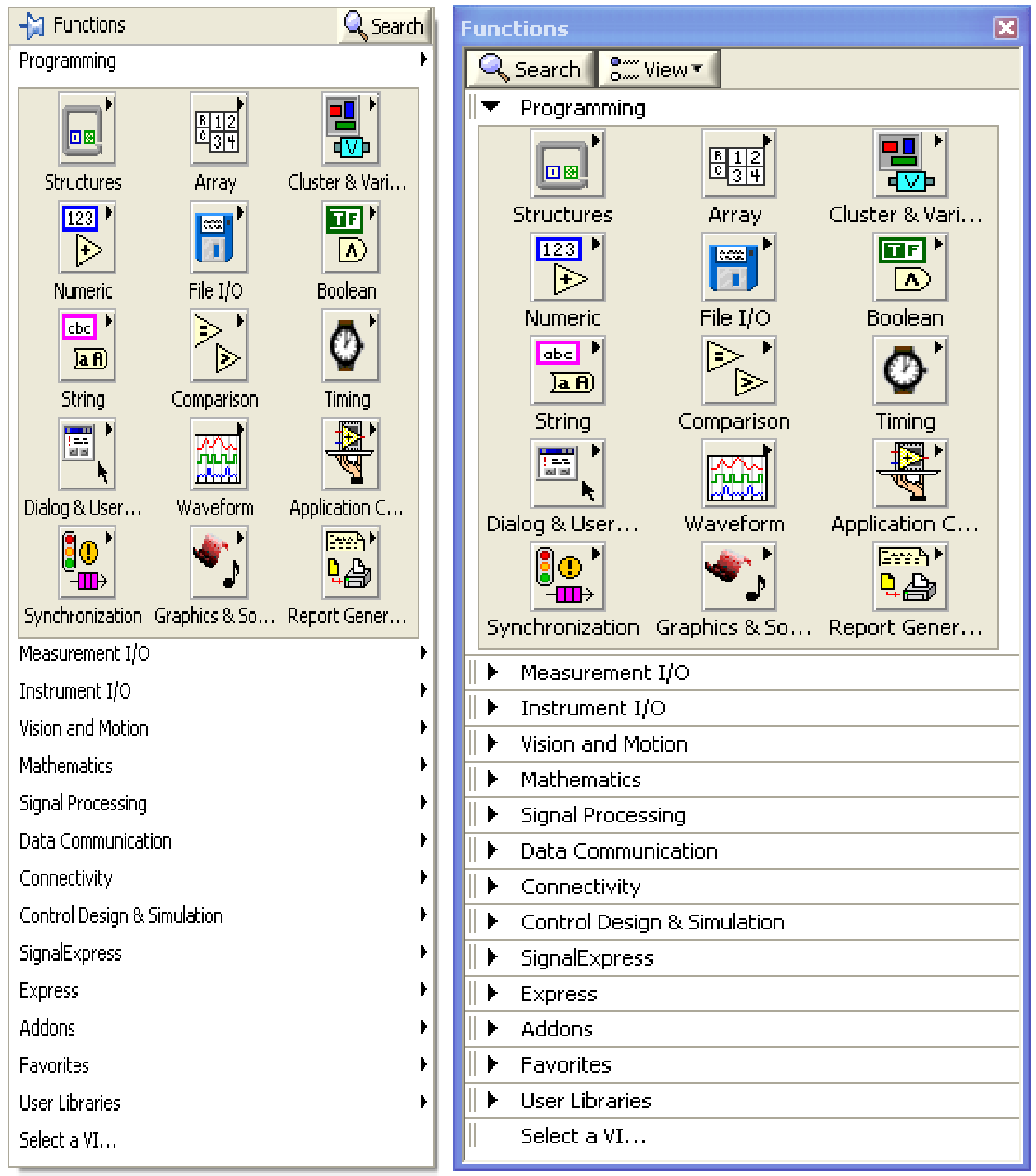


8-сурет. Элементтер палитрасының көрінісі

**1.6.3. Функция палитрасы**

Функция палитрасы блог-диаграмма құру үшін қолданылады. Ол тек блог-диаграммада ғана қолжетімді. Функция палитрасын шығару үшін негізгі мәзірден **Window»Show Functions Palettе** пунктін немесе блог-диаграмманың жұмыс бетінде тышқанның оң жақ батырмасын басу арқылы шақыруға болады. Палитраның жоғары сол жақ бөлігіндегі батырманы басу арқылы экранды сақтап қалуымызға болады. Функция палитрасының көрінісі төменде бейнеленген (9-сурет).

Палитрадағы пункттер категорияларға сәйкес ұйымдастырылған. Сіз палитраның жоғарғы бөлігіндегі **View** батырмасын қолданып палитрада әртүрлі категорияларды (Category(Standard), **Category** (Icons and Text), **Icons, Icons and Text, Text, Tree**) көрсету тәсілдерін таңдай аласыз.



9-сурет. Функция палитрасының көрінісі

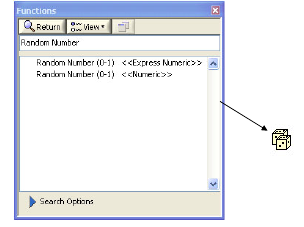
•**Category** (Standard) және **Category**  (Icons and Text) палитраларында пунктерді көрсету тәсілі кезінде Сіз опция категориясына контексті мәзірдегі **Move this Category Up** және **Move this Category Down** опция категорияларын қолдана отырып палитрадағы пункттердің орналасуын өзгерте аласыз. Бұдан бөлек Сіз категорияның сол жағында орналасқан қос сызықты шерту арқылы категорияның орнын ауыстыра аласыз.

• **Text** және **Tree** палитраларында пункттерді көрсету тәсілі кезінде Сіз оларды әліпби тәртібінде реттей аласыз. Ол үшін **View» Sort Alphabetically** батырмасын таңдаңыз.

**Search** батырмасын басқан кезде Сіз қандай да функцияны, түйінді немесе ВҚ-ны атауы бойынша іздеу режиміне өте аласыз.

**1.6.4. Controls (Элементтер) және Functions (Функциялар) палитраларындағы объектілерді табу**

Мысалы, **RandomNumber** (кездейсоқ сандар генераторы) функциясын табу үшін **Functions** (Функциялар) палитрасында Search батырмасын басу керек және мәтінді енгізу жолына «**Random Number**» енгізу керек. LabVIEW экранға атауында енгізілген мәтін кездесетін функциялар мен түйіндерді шығарады. Іздеу нәтижелерінде ізделінген функцияны таңдап оны тышқанның көмегімен блог-диаграммаға орналастырса болады. Бұл процесс көрінісі төменде көрсетілген (10-сурет).



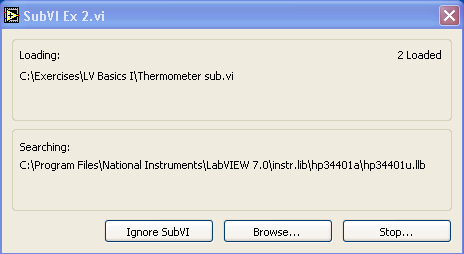
10-сурет.Элементтер және Функция палитрасында объектілерді іздеу көрінісі

Іздеуден шыққан функцияны екі рет басу арқылы оның орналасу орнын көре аласыз.

**1.7. ВҚны жүктеу**

Негізгі меңзердегі **File»Open** пунткі көмегімен диалогтік терезе ашылады. Осы терезе ВҚны таңдауға және оны компьютер жадына жүктеуге мүмкіндік береді. Төменде ВҚны жүктеу кезінде пайда болатын диалогтік терезенің көрінісі көрсетілген (11-сурет ).

Бұл терезеде таңдап алынған ВҚның барлық ішкі программаларының ретімен жадқа жүктелуі көрініп тұрады. Жүктеуді кез келген уақытта тоқтатуға болады. Ол үшін **Stop** батырмасын басу жеткілікті.



11-сурет. ВҚны жүктеу кезіндегі диалогтік терезенің көрінісі

Егер LabVIEW тез арада ішкі программаны таба алмаса **Search Path** файлдарды іздеу жолында жазылған барлық директорияларға іздеу жалғасады. Файлдарды іздеу жолдарын Жабдықтар (**Tools » 0ptions » Paths**) мәзір пунктінқолданып өңдеуге болады. **Ignore Sub VI** батырмасын басып немесе **Browse (шолу)** батырмасын басып ішкі программаға қолдан іздеу жүргізуге мүмкіндік бар.

**Өтілген материалдардың қысқаша мазмұндамасы**

* ВҚ негізгі төрт құраушыдан тұрады – беттік панель, блог-диаграмма, белгіше және байланыстырушы панель.
* Беттік панель – ВҚ қолданушысының интерфейсі.Беттік панельде болашақ виртуалды құрылғының сыртқы көрінісі жасалады.
* Блог-диаграмма – түйіндерден, деректер терминалынан және өткізгіштерден тұратын программаның графикалық бастапқы мәтіні. Блог-диаграммада болашақ виртуалды құрылғының негізгі коды жасалады.
* Жабдықтар палитрасы ВҚ құру және түзету үшін арналған. Басылған **<Shift>** батырмасын ұстап тұрып және тышқанның оң жақ батырмасын басып, экранға Жабдықтар палитрасының уақытша нұсқасын шығаруға болады.
* **Controls** (Элементтер) палитрасы беттік панельдің интерфейсін құру үшін арналған. Экранға **Controls** (Элементтер) палитрасын шығару үшін беттік панельдің ашық кеңістігінде тышқанның оң жақ батырмасын басу қажет. **Functions** (Функция) палитрасы блок-диаграмма жасау үшін арналған. Экранға **Functions** (Функция) палитрасын шығару үшін блок-диаграмманың ашық кеңістігінде тышқанның оң жақ батырмасын басу қажет.
* LabVIEW-дің барлық объектілері, беттік панельдің және блог-диаграмманың бос жұмыс жасау кеңістігі өзінің контексттік мәзіріне ие болады. Қалқыма мәзірді шақыру объектте, беттік панельде немесе блог-диаграммада тышқанның оң жақ батырмасын басумен орындалады. **Help** мәзірі**Context Help** (контекстік анықтама) және **LabVIEW Help** терезелерін экранға шығару үшін қолданылады.

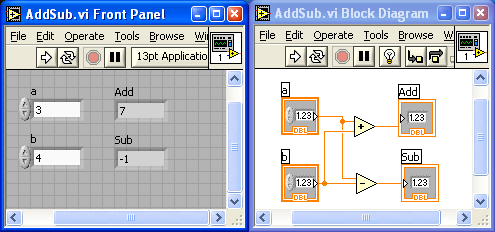
**1-лабораториялық жұмыс.Виртуалды құрылғы құру**

*ажәне b* екі санының қосындысын және айырмасын есептейтін виртуалды құрылғы құру қажет.

Бұл есепті шешу үшін төмендегі суретте көрсетілгендей блог - диаграмма мен беттік панель құрастыру керек**.**

Беттік панель мен блог-диаграмма құрастыру келесідей амалдарды орындау нәтижесінде жүзеге асады:

1. Беттік панельге 4 цифрлық элемент шақырылады. Оның екеуі ажәне b басқару элементтері , ал екеуі а+b және а-b (шығыс мәндер) индикаторлар. Олардың суреттері немесе белгішелері (иконкалары) блог-диаграммада автоматты түрде пайда болады. Блог-диаграммада функция палитрасының ішінен қосу (+) және азайту (-) амалдары шақырылады. Бұл амалдарды шақыру мына алгоритм бойынша іске асырылады: **Arithmetic & Compare => Numeric.**



12-сурет. Екі санның қосындысы мен айырмасын есптейтін виртуалды құрылғының беттік панелі мен блог-диаграмма көрінісі

2.Элементтерді байланыстыру байланыстырғыш немесе катушка көмегімен іске асады. Катушканы элементке алып барамыз, элемент жыпылықтаған сәтте тышқанның оң жақ батырмасын басып, пайда болған сымды келесі индикатор немесе басқару элементіне дейін созамыз. Байланысатын элементпен катушка түйіскен сәтте элементіміз жыпылықтайды. Сол моментте тышқан батырмасын босатамыз.

3.Беттік панельде басқарушы элементтерге сан мәндері беріледі.

4.Виртуалды құрылғыны іске қосу үшін беттік панельден немесе блог-диаграммадан іске қосу батырмасын басамыз. Сол сәтте индикаторлар терезесінен нәтижені көреміз.

1-тапсырма. LabVieW инструменттері және мәлімет типтері. LabVieWнегізіндегі виртуалды құрылғының .EXE форматын жасау.

**Бақылау сұрақтары:**

1.

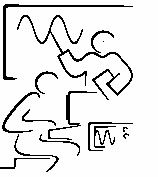
2.

3.

4.

5.

6.

2-бөлім

**ВИРТУАЛДЫ ҚҰРЫЛҒЫМЕН ЖҰМЫС ІСТЕУ НЕГІЗДЕРІ**

**2.1. ВҚ компоненттері**

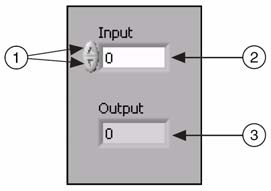
Виртуалды құрылғы дегеніміз LabVieW аппараттық-программалық ортада жазылған программа. ВҚ 4 негізгі компоненттерден тұрады: беттік панель, блог-диаграмма, иконкалар (белгішелер) және біріктіруші панельдер.

**2.1.1.Беттік панель**

Беттік панельде ВҚның кіріс және шығыс мәндерін енгізу үшін басқару және бейнелеу элементтері құрылады. Басқару элементтері-батырмалар, ауыстырып-қосқыштар және тағы да басқа мәліметтер енгізу құралдары жатады. Бейнелеу элементтеріне- графиктер, жарықтық диодтар және тағы басқа да индикаторлар жатады. Басқару элементтері мәліметтерді енгізу құрылғыларын модельдейді және мәліметтерді ВҚның блог-диаграммасына жібереді. Блог-диаграмма беттік панельден енгізілген кіріс мәліметтерді генерациялайды.

**Сандық басқару элементтері және мәліметтерді бейнелеу**

Көбіне екі сандық объект–сандық басқару және мәліметтерді бейнелеу элементі кеңінен қолданылады (13-сурет):

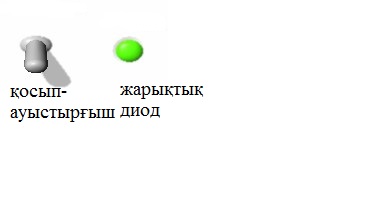


1.Мәліметтерді өзгерту батырмасы; 2-сандық басқару элементі; 3-мәліметтерді бейнелеудің сандық элементі

13-сурет.Сандық басқару элементі мен мәліметтерді сандық бейнелеу элементтерінің көрінісі

**Логикалық басқару элементтері және мәліметтерді бейнелеу**

Логикалық басқару және бейнелеу элементтері логикалық айнымалы арқылы (TRUE/FALSE - ақиқат/жалған) мәліметтерді енгізу және бейнелеу үшін қолданылады. Логикалық объектілер ретінде қосып-ауыстырғыштар, батырмалар және жарықтық диодтар қолданылады. Вертикалды қосып-ауыстырғыш және дөңгелек жарықтық диод көрінісі төменде көрсетілген:



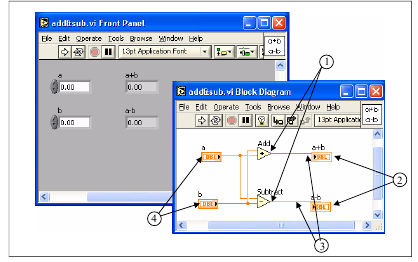
14-сурет. Вертикалды қосып-ауыстырғыш және жарықтық диодтың сыртқы көрінісі

**Басқару элементтерін өңдеу және мәліметтерді бейнелеу**

Мәліметтерді басқару және бейнелеу элементтерінің барлығын дерлік олардың контексті мәзірін қолдана отырып өңдеуге болады. Контексті мәзірден шақыру үшін объектіге тышқан батырмасын алып барып оң жақ бөлігін басу қажет. Мысалы, белгініөңдеу үшін белгі бетіне тышқанды апарып, оң жақ батырмасын басу керек.

**2.1.2. Блог-диаграмма**

Блог-диаграмма түйіннен, терминалдардан және мәліметтер сымдарынан тұрады (15-сурет):



1.Түйіндер; 2-мәліметтерді бейнелеу элементтерінің терминалдары;

3-мәліметтер сымдары; 4-мәліметтерді басқару элементтерінің терминалдары

15-сурет. Блог-диаграмманың көрінісі

**2.1.3. Мәліметтер терминалдары**



Блог-диаграммадағы беттік панельдің объектілері терминалдар түрінде көрініс табады (әріптік-сандық белгісі бар тік бұрышты формадағы графикалық бейне). Мәліметтер терминалындағы әріптік-сандық белгілеу басқару және бейнелеу элементтерінде қолданыла алатын мәліметтер типтерін анықтайды. Мысалы, сол жақта көрсетілген DBL-терминалы берілген басқару элементінің жылжымалы үтірлі екілік дәлдіктегі сан қолданылатынын көрсетеді.

Мәліметтер терминалдары сол жақта көрсетілгендей белгіше (иконка) түрінде бейнеленуі мүмкін. Ол үшін мәліметтер терминалы бетінде тышқанның оң жақ батырмасын басу жеткілікті және контексті мәзірден View as Icon (белгіше (иконка) түрінде бейнелеу) пунктін таңдау керек. Стандартты түрде бейнелеу үшін белгіні алып тастау қажет. Стандартты түрде мәліметтер терминалдарын бейнелеу блог-диаграммада орындарды сақтауға мүмкіндік береді. Мәліметтер терминалы блог-диаграмма мен беттік панель арасында мәліметтер алмасуды қамтамасыз етеді. Олар: мәтіндік программалау тілдеріндегі айнымалы мен тұрақтыларға ұқсайды. Мәліметтер терминалдары мынадай типтерге бөлінеді: басқару элементтері терминалы, мәліметтерді бейнелеу терминалдары және түйіндер терминалдары. Басқару элементтері және бейнелеу элементтері терминалдары беттік панельде мәліметтерді басқару және бейнелеу құралдарына жатады. Беттік панельде берілген мәліметтер (15-суретте берілгендей яғни басқару элементтері а және b), блог-диаграммаға осы терминалдар арқылы түседі. Add (қосу), Subtract (азайту) функциялары есептеуді аяқтаған сәтте шығыста жаңа мәлімет мәні шығады. Бұл мәндер мәліметтерді бейнелеу элементтері терминалдарына түсіп, әрі қарай беттік панельге жіберіледі.

 Сол жақта көрсетілген байланыстырушы панель **Add** (қосу),**Subtract** (азайту) функциялары үш жолдан тұрады. Ол 3 жол мәліметтерді енгізу/шығаружолдары болып табылады. Яғни, екі жолы мәліметтерді енгізу жолы, ал бір жолы мәліметтерді шығару жолы. Байланыстырушы панельді бейнелеу үшін функцияның үстіне тышқан батырмасын апарып, тышқанның оң жақ батырмасын басып, контексті мәзірден **VisibleItems»Terminals** пунктін таңдау керек.

**2.1.4.Түйіндер**

Түйіндер-ВҚдағы алгоритмдік амалдарды орындайтын және бір немесе бірнеше мәліметтер кіріс/шығысынан тұратын блог-диаграммадағы объектілер. 15-суретте түйін сарғылт түсті нүкте түрінде бейнеліп тұр. Бұлар мәтіндік программалау тіліндегі операторлар, функциялар және ішкі программаларға ұқсайды. Түйіндер өзіне функцияларды, ВҚның ішкі программасын және құрылымдарды қосады. ВҚның ішкі программасы дегеніміз басқа ВҚның блог-диаграммасында ішкі программа ретінде қолдануға болатын виртуалды құрылғы. Құрылымдар дегеніміз процесстерді басқару элементтері, мысалы **Case** (нұсқалар), **While** циклі (шарт бойынша) және т.б. Жоғарыдағы блог-диаграммада көрсетілген **Add**  (қосу) және **Subtract** (азайту) түйіндері –функциялар түйіндері болып табылады.

**2.2. Мәліметтер типтері мен өткізгіштері**

LabVIEW ортасында мәлімет өткізгіштері бірнеше мәліметтер терминалдарын байланыстыру үшін қолданылады. Кіріс/шығыс жолдары өткізгіштер бойымен жіберіліп жатқан мәліметтер типімен сәйкес болу керек. Мысалы, кіріс элемент массив болса оны сандық типтегі элементпен байланыстыруға болмайды. Өткізгіштер тек бір ғана мәлімет көзіне байланысу керек. Тым болмағанда мәліметтер кірісінің біреуіне байланысу қажет. Мысалы, екі бейнелеу элементтерін біріктіруге болмайды. Байланыстың үйлесімділігін анықтайтын құраушыларға элементтерді басқару және бейнелеу мәлімет типтері және кіріс/шығыс мәлімет типтері жатады.

**Мәліметтер типтері**

Осы курста мынадай мәліметтер типтері қолданылады:

**\*Numeric** (сандық тип)

**\*Floating point –** жылжымалы үтірлі сан, қызғылт сары түсті терминалмен бейнеленеді. **Single** (32 bit), **double** (64-bit) немесе **extended** (128-bit),**precision** (бірлік, екілік немесе кеңейтілген дәлдіктегі) түрінде көрсетілуі мүмкін. Жылжымалы үтірлі сан комплексті болуы мүмкін.

\***Integer —** бүтін сандық тип, көк түсті терминал түрінде бейнеленеді. Бүтін сан 3 түрде: 8,16, 32 бит түрінде көрсетілуі мүмкін. Егер сан бүтін болса, онда бір бит санның белгісіретінде қолданылуы мүмкін.

**\*Boolean —** логикалық тип, жасыл түстегі терминалмен бейнеленеді. Логикалық тип тек екі мәнді ғана қабылдай алады: 0 (Жалған) немесе 1 (ақиқат).

**\* String** — жолдық тип, алқызыл түстегі терминалмен бейнеленеді. Жолдық мәлімет типі ASCII форматтағы мәтінді құрайды.

**\* Path –** файлға жол, терминал түрінде бейнеленеді. Файлға жол жолдық типке жақын, алайда LabVIEW қолданылатын платформа үшін стандартты синтаксистерді қолдана отырып оны форматтайды.

**\* Array** — массивтер құраушы элементтердің мәлімет типтерін құрайды және соған сәйкес түстерге ие болады.

**\* Сluster** — кластерлер әр түрлі мәлімет типтерінен тұрады. Егер оның барлық элементтері сандық болса мәліметтердің кластерлік типі қоңыр түспен бейнеленеді. Егер де кластердегі элементтер әр түрлі типте болса, онда кластер түсі қызғылт түсті болады.

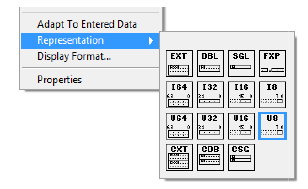
**\* Waveform -** сигналдық мәлімет типі уақыттың бастапқы мәніне ие және өлшеу арасындағы уақыт интервалы мәліметтерін құрайтын элементтер кластері болып табылады.

**\* Dynamic** — динамикалық тип, қоңыр көк түсті терминалмен бейнеленеді. Динамикалық тип сигнал мәліметтерімен қоса қосымша ақпараттарға да ие. Мысалы, сигнал атауы, күні оны алудың уақыты.Көптеген экспресс-ВҚлар динамикалық тип мәндерін қабылдайды немесе қайтарады. Динамикалық типтегі мәліметтерді кез келген бейнелеу элементтеріне бағыттауға болады.

Мәлімет типтерімен жұмыс жасау кезіндегі LabVIEW ортасының ерекшелігі полиморфизм қасиетінің болуы. Полиморфизм деп әр түрлі мәлімет типтерін өңдейтін функция қабілеттілігін айтады. Функция кіріске қандай мәліметтерді беру керек екекнін өзі түсінеді және оны сәйкесінше өңдейді.

3-кесте. LabVIEWде қолданылатын мәлімет типтерінің сипаттамалары

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Басқару  элементі | Индикатор | Мәлімет типі | Түсі | Бастапқы қалпындағы мәні |
|  |  | Кеңейтілген дәлдіктегі жылжымалы үтірлі сан | Қызғылт сары | 0 |
|  |  | Екілік дәлдіктегі жылжымалы үтірлі сан | Қызғылт сары | 0 |
|  |  | Бірлік дәлдіктегі жылжымалы үтірлі сан | Қызғылт сары | 0 |
|  |  | Бекітілген нүктелі сан | Күлгін | 0 |
|  |  | Белгісі бар 64 разрядты бүтін сан | көк | 0 |
|  |  | Белгісі бар 32 разрядты бүтін сан | көк | 0 |
|  |  | Белгісі бар 16 разрядты бүтін сан | көк | 0 |
|  |  | Белгісі бар 8 разрядты бүтін сан | көк | 0 |
|  |  | Белгісі жок 64 разрядты бүтін сан | көк | 0 |
|  |  | Белгісі жок  32 разрядты бүтін сан | көк | 0 |
|  |  | Белгісі жок 16 разрядты бүтін сан | көк | 0 |
|  |  | Белгісі жок 8 разрядты бүтін сан | көк | 0 |
|  |  | Бірлік дәлдіктегі жылжымалы үтірлі комплексті сан | Қызғылт сары | 0+i0 |
|  |  | Екілік дәлдіктегі жылжымалы үтірлі комплексті сан | Қызғылт сары | 0+i0 |
|  |  | Кеңейтілген дәлдіктегі жылжымалы үтірлі комплексті сан. | Қызғылт сары | 0+i0 |
|  |  | Массив- квадраттық жақша ішіне мәлімет типін жинайды және сол мәлімет типіне сәйкес түске ие болады. | сұр | - |
|  |  | Санап шығу типі | көк | - |
|  |  | Логикалық | жасыл | FALSE |
|  |  | Жолдық | Алқызыл (қызғылт) | Бос жол |
|  |  | Жол | Теңіз толқыны | Жол емес (not a path) |
|  |  | Кластер-әр түрлі мәлімет типтерін жинайды | Қоңыр немесе алқызыл (қызғылт) | - |



16-сурет. Мәлімет типі көрінісінің контексті мәзірі

**2.3. Мәліметтер өткізгіштері**

Блог-диаграммадағы объектілер арасындағы мәліметтер байланыстырушы сызықтар бойымен-мәліметтер өткізгіштерімен таратылады (жіберіледі).

Мәліметтер өткізгіші мәтіндік программалау тіліндегі айнымалыға ұқсайды. Мәліметтердің әрбір өткізгіші жалғыз мәлімет көзіне ие бола алады, бірақ оларды көптеген ВҚға және функцияға жіберуі мүмкін. Мәліметтер типтері тарататын мәлімет типтеріне байланысты түсімен, стилімен және сызықтың қалыңдығымен өзгешеленеді. Мәліметтер өткізгішінің негізгі типтеріне мысалдар төмендегі кестеде көрсетілген:

4-кесте. Мәлімет өткізгіштерінің типтері

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мәлімет өткіз  гішінің типі | Бір мән | Бірөлшемді (1D) массив | Екі өлшемді  (2D) массив | Түсі |
| Сандық |  |  |  | Сарғылт(жылжымалынүктемен), көк (бүтінсан) |
| Логикалық |  |  |  | Жасыл |
| Жолдық |  |  |  | Қызғылт |

**Объектілерді автоматты түрде мәліметтер өткізгіштері арқылы байланыстыру**

LabVIEW ортасында объектілер мәліметтер өткізгіштері арқылы байланысады да, содан кейін блог-диаграммаға орналасады. Автоматты режимде LabVIEW ортасы біршама үйлесімді кіріс/шығыс мәліметтері жолын байланыстырады, ал үйлесімсіз жолдар байланыспай қалады. Егер таңдап алынған объект блог-диаграммадағы басқа объект төңірегінде орналасса онда LabVIEW ортасы пунктирлі уақытша мәліметтер өткізгішін көрсетеді. Тышқан батырмасын жіберген кезде LabVIEW автоматты түрде мәліметтер өткізгішін таңдап алынған объект мәліметінің кіріс/шығыс жолына қосады. Өткізгіштерді автоматты қосу параметрін түзеу негізгі мәзірдегі мына пунктер арқылы іске асады: **Tools>>Options>>Block Diagram**.

**Объектілерді мәліметтер өткізгіші арқылы қолдап байланыстыру**

Объектілерді мәліметтер өткізгішімен қолдап байланыстыру **Байланыстыру** құралы көмегімен іске асады. Мәліметтерді енгізу және шығару жолына Байланыстыру элементін орналастырғаннан кейін экранда өткізгіштің қосу орнын нақтылау үшін қолдануға мүмкіндік беретін кеңес беріледі.

**2.4. LabVIEWдағы базалық функциялар**

LabVIEW**Programming, Mathematics, Signal Processing, Vision and Motion, Connectivity** категориялары бойынша функция палитрасында орналасқан біріктірілген функциялар санын құрайды. Көп сандар және функциялар күрделілігіне орай функция палитрасын екі топқа бөлген ыңғайлы: базалық және қосымша. Базалық функция қатарына мынадай палитра ішіндегі палитраларды: Сандық (Numeric), логикалық (Boolean), жолдық (String), салыстыру (Comparison)жатқызады.

**2.4.1.Сандық функциялар**

Mathematics палитрасында орналасқан математикалық функциялар сандық мәліметтермен тригонометриялық, арифметикалық, логарифмдік амалдарды орындау және сандық мәліметтерді түрлендіру үшін қолданылады.

Математикалық функциялар палитрасы төмендегідей бірнеше палитрадан тұрады:

Сандық амалдар:

-Типтерді түрлендіру;

-Биттік манипуляция;

-Комплекстік сандар;

-Масштабтау;

-Бекітілген нүктелі сандар;

-Математикалық тұрақтылар;

Элементар функциялар:

-Тригонометриялық функциялар;

-Логарифмдік функциялар;

-Гиперболалық функциялар;

-Дискретті математика;

-Сызықты алгебра;

-Аппроксимация;

-Интерполяция және экстраполяция;

-Интегралдау және дифференциалдау;

-Статистика;

-Дифференциалдық теңдеуді шешу;

-Геометрия;

-Полиномдар.

Жоғарыда аталып өтілген палитраларды және функциялар комбинациясын қолдана отырып сандарға көптеген математикалық амалдарды орындауға болады.

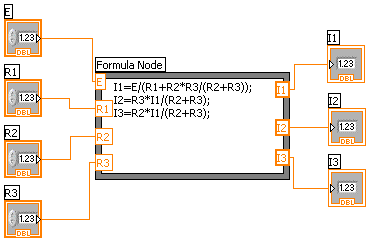
Төменде LabVIEWдағы негізгі математикалық функциялардың сипаттамалары жазылған. Барлық қарастырылған функциялар полиморфты. Олардың кірісіне скалярлы өрнектерді, массивтарды, комбинацияларды беруге болады.

5-кесте. Математикалық функция сипаттамасы

|  |  |  |
| --- | --- | --- |
| Атауы | Пиктограмма | Сипаттама |
| Add |  | Екі операндтың суммасын береді. |
| Subtract |  | Екі операнд айырымын береді. |
| Multiply |  | Екі операндтың көбейтіндісін береді. |
| Divide |  | Екі операндтың айырымын береді. |
| Increment |  | Кіріс мәнге бір мәнін қосып береді. |
| Decrement |  | Кіріс мәннен бір мәнін азайтып тастайды. |
| Quotient&  Remainder |  |  |
| Absolute  Value |  | Кіріс мәнінің модулін береді. |
| Square |  | Кіріс мәнінің квадратын береді. |
| Negate |  | Кіріс мәнін -1 мәніне көбейтеді. |
| Sine |  | Кірісінде мән радиан бойынша беріледі, кіріс мәннің синусын шығарады. |
| Cosine |  | Кірісінде мән радиан бойынша беріледі, кіріс мәннің косинусын шығарады. |
| Tangent |  | Кірісінде мән радиан бойынша беріледі, кіріс мәннің тангенсін шығарады. |
| Power of X |  | х мәннің у дәрежесін шығарады. |
| Power of 2 |  | 2-нің х дәрежесін шығарады. |
| Logarithm  Base 10 |  | Кіріс мәнінен ондық логарифм мәнін шығарады. |
| Random Number |  | 0-ден 1-ге дейінгі кездейсоқ сандарды береді. |

Жоғарыдағы қарастырылған математикалық функциялармен қатар Formula Node құрылымы да мәтіндік программалау тілінің негізі болып табылады. Formula Node құрылымында математикалық өрнектер және амалдар мәтіндік түрде жазылады. Бұл құрылымды өте көп айнымалылар және күрделі есептер кезінде қолданған ыңғайлы.

Formula Node жұмыс жасау үшін міндетті түрде кіріс, шығыстарын анықтап алу керек. Кіріс/шығысты енгізу үшін құрылымның сол жақ бөлігіне (кіріс), оң жақ бөлігіне (шығыс) тышқан батырмасын апарып, оң жағын басу арқылы контексті мәзірден Add input және add output пунктерін таңдау қажет. Құрылым рамкаларына кіріс/шығыс элементтерінің белгісін енгізу керек. Кіріс/шығыс айнымалылары құрылым ішіндегі айнымалылармен бірдей болу міндетті.



17-сурет. Formula Node құрылымының көрінісі

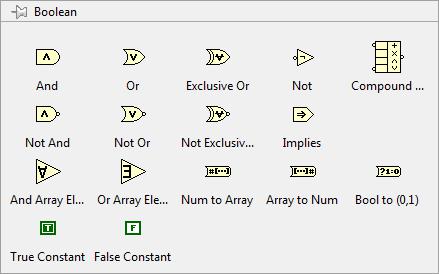
6-кесте. Formula Node құрылымындағы амалдар сипаттамалары

|  |  |
| --- | --- |
| Атауы | Сипаттамасы |
| \*\* | Дәрежеге шығару |
| +, -, !, ~, ++,– – | Унарлық қосу, азайту, логикалық емес операторы, биттік толықтыру, алдыңғы және кейінгі инкремент, алдыңғы және кейінгі декремент. |
| \*, /, % | Көбейту, бөлу, модуль (қалдық) |
| >>және<< | Арифметикалық жылжыту (оңға және солға) |
| >, <, >=, <= | Үлкен, кіші, үлкен немесе тең, кіші немесе тең |
| != ,== | Эквивалентті емес, эквивалентті |
| & | Биттік ЖӘНЕ |
| ^ | Оқшауланған НЕМЕСЕ логикалық элементі |
| | | Биттік НЕМЕСЕ |
| && | ЖӘНЕ логикалық элементі |
| || | НЕМЕСЕ логикалық элементі |
| ?: | Шартты өрнек |
| abs(x) | х-тің абсолюттік мәні |
| acos(x) | х-тің арккосинусы (рад) |
| asin(x) | х-тің арксинусы (рад) |
| atan(x) | х-тің арктангенсі (рад) |
| ceil(x) | Үлкен санға дейін дөңгелектеу |
| cos(x) | Косинус х |
| cosh(x) | Гиперболалық косинус |
| cot(x) | Котангенс х (рад) |
| csx(x) | Косеканс х |
| exp(x) | Экспонента х |
| floor(x) | Кіші бүтін х-ке дөңгелектеу |
| pow(x,y) | х-тің у дәрежесін шығару |
| sec(x) | Секанс х |
| sin(x) | Синус х |
| sinh(x) | Гиперболалық х |
| sqrt(x) | Квадраттық түбір х |
| tan(x) | Тангенс х |
| tanh(x) | Гиперболалық тангенс х |

**2.4.2. Логикалық функциялар**

Логикалық функциялар (18-сурет) қарапайым логикалық шамалармен және сол шамалар массивтері мәндерімен логикалық амалдарды орындауға қолданылады. ЖӘНЕ (And), НЕМЕСЕ (Or) , ЕМЕС (Not), оқшауланған НЕМЕСЕ (Exclusive Or), НЕМЕСЕ-ЕМЕС ( Not And), НЕМЕСЕ-ЖӘНЕ(Not Or) , оқшауланған НЕМЕСЕ-ЕМЕС(Not Exclusive Or), оқшауланған (Implies) функциялар полиморфты болып табылады. Бұл функциялардың екі кірісі де логикалық және сандық мәндерді де иелене алады. Логикалық функциялар Programming-Boolean палитрасында орналасқан.

7-кестеде LabVIEWдегі негізгі логикалық функциялардың сипаттамасы берліген.



18-сурет. Логикалық амалдар палитрасы

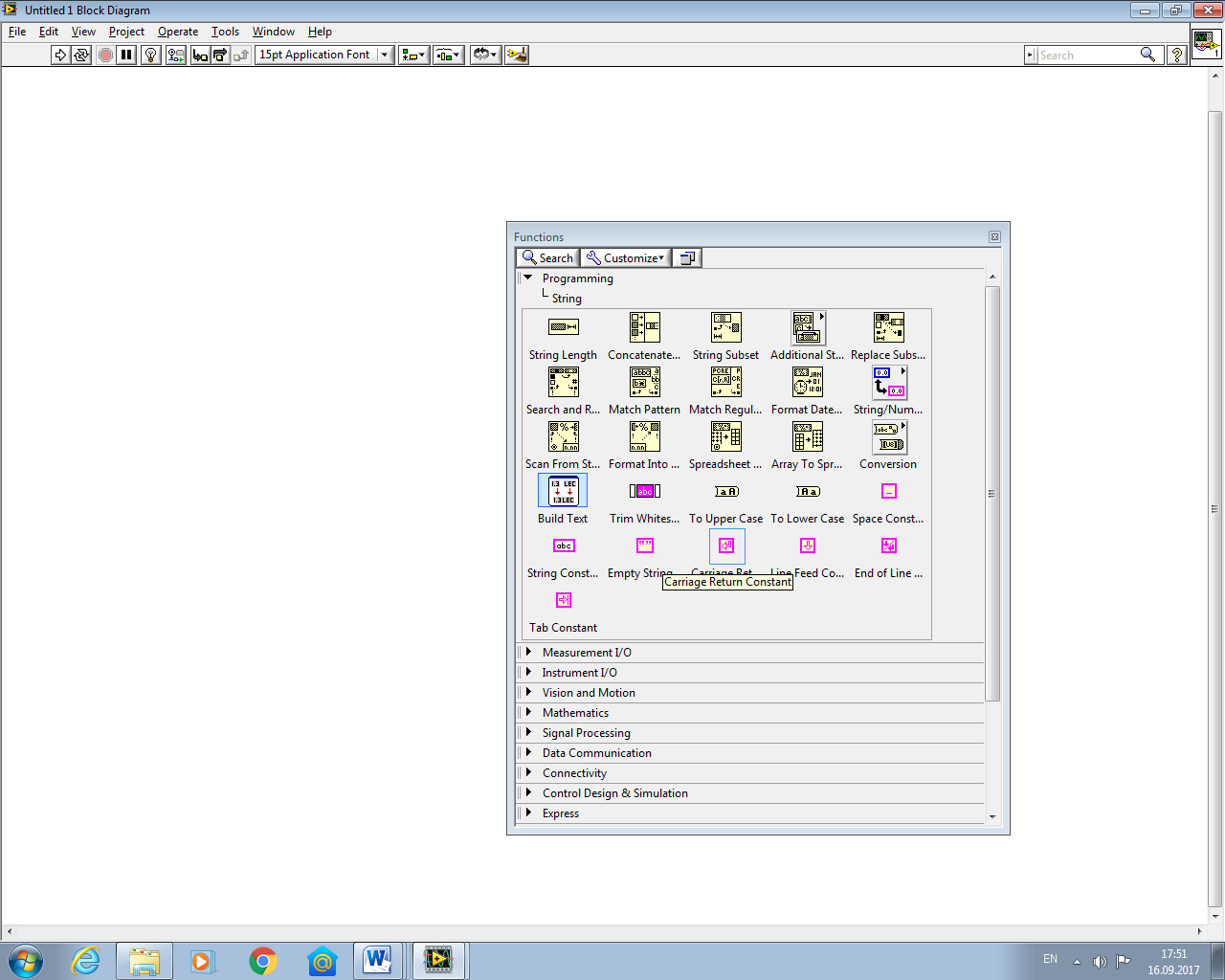
7-кесте. Логикалық функциялар сипаттамасы

|  |  |  |
| --- | --- | --- |
| Атауы | Пиктограмма | Сипаттама |
| And | and | Егер екі кірісіне де ақиқат (1) мәнін беретін болсақ, онда шығысына ақиқат (1) мәнін шығарады. Әйтпесе жалған (0) мәнін қайтарады. |
| Or | or | Егер екі кірісіне жалған (0) мәнін берген жағдайда шығысында жалған (0) мәнін шығарады. Әйтпесе ақиқат (1) мәнін қайтарады. |
| Exclusive Or | ExOr | Егер екі кірісіне жалған (0) мәнін берген жағдайда шығысында жалған (0) мәнін немесе ақиқат (1) мәнін шығарады. Әйтпесе ақиқат (1) мәнін қайтарады. |
| Not | not | Кірісіне ақиқат (1) мәнін берген кезде шығысында жалған (0) мәнін шығарады. |
| Not And | notAnd | Екі кірісіне бірдей жалған (0) мәнін берген кезде шығысына ақиқат (1) мәнін шығарады, әйтпесе жалған (0) мәнін қайтарады. |
| Not Or | notOr | Егер екі кірісіне бірдей ақиқат (1) мәнін берсек, онда шығысы жалған (0) мәнін шығарады. Әйтпесе жалған (0) мәнін қайтарады. |
| Implies | impl | х мәнін инверттейді, содан кейін у кірісімен НЕМЕСЕ логикалық амалын орындайды. Егер у кірісіндегі мән ақиқат (1) болса, онда функция жалған мәнін қайтарады. Әйтпесе ақиқат мәнін қайтарады. |

**2.4.3. Жолдық функциялар**

Жолдар ASCII стандартындағы бейнеленетін және бейнеленбейтін символдар тізбектілігін көрсетеді. Әдетте жолдар қарапайым мәтіндік хабарламалар түрінде қолданылады. құрылғыларды басқару процесінде сандық мәліметтер символдық жолдар түрінде жіберіледі. Жолдық функциялар Programming-String палитрасында орналасқан.

Жолдық басқару элементтері және индикаторлар (strings)мәтіндік мәліметтерді демонстрациялайды. Жолдық мәліметтер өтетін жолдық терминалдар және өткізгіштер алқызыл түске боялған (19-сурет).



19-сурет. Жолдық функция палитрасы

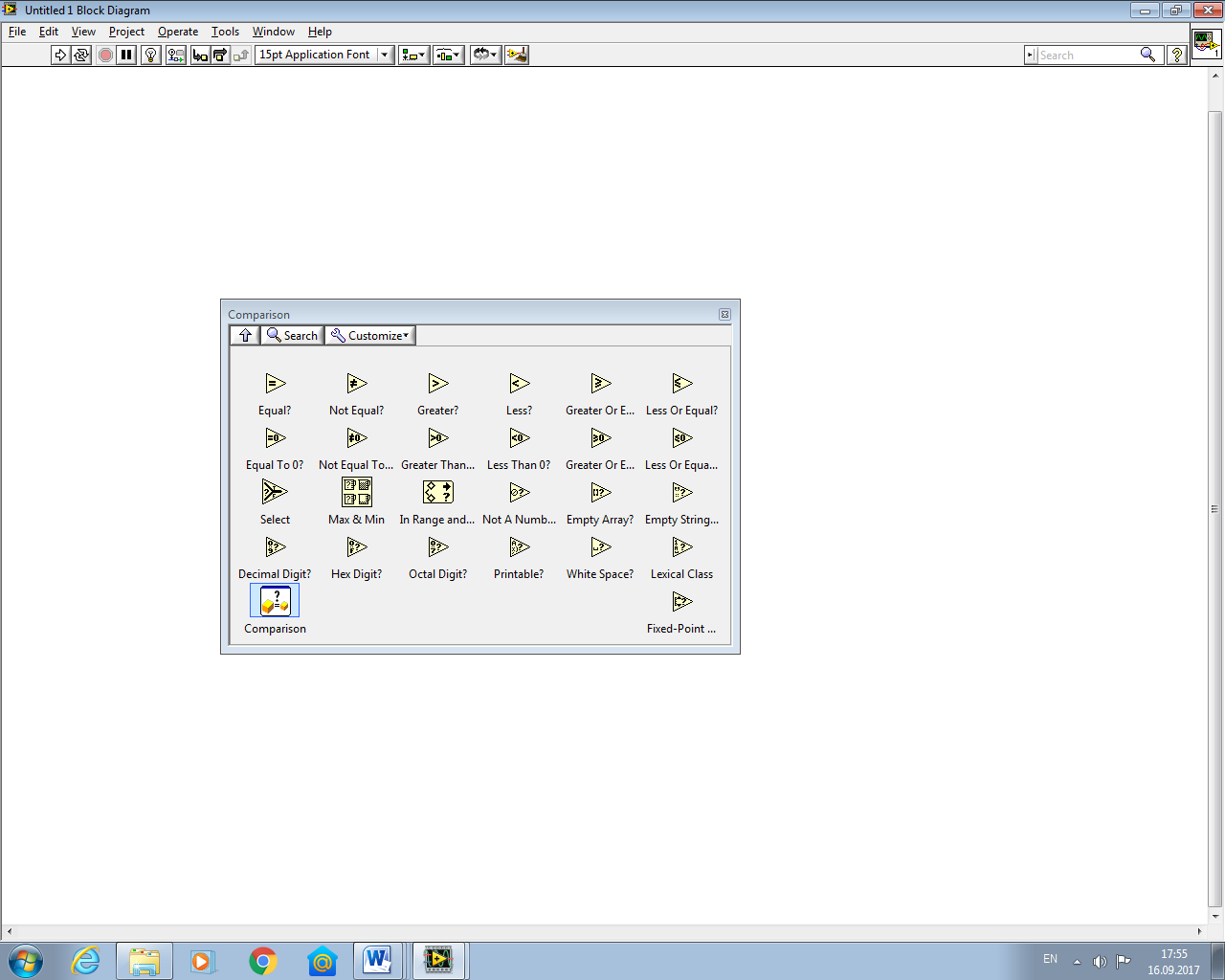
8-кесте. Жолдық функция сипаттамасы

|  |  |  |
| --- | --- | --- |
| Атауы | Пиктограмма | Сипаттама |
| String Length |  | Функция жолдағы символдар санын қайтарады. Жол (string) кірісіне жол, кластер, жолдық кластер масиивтері берілуі мүмкін. |
| Concatenate string |  | Функция кіріс жолдарды және бір өлшемді жолдық массивтерді жалғыз ғана шығыс жолына біріктіреді. Жолға біріктірілген жолдық массивке массивтің әрбір элементі кіреді. |
| String Subset |  | Функция жылжытудан (щааыуе) және кірісінде ұзындығына (length) берілген символдан тұратын кіріс жолдың (string) бір бөлігін ғана қайтарады. |
| Search and Replace String |  | Функция жол ішіндегі жолдың барлық үлгілерін немесе бір үлгісін қандай да бір келесі жол ішіндегі жолмен алмастырады. |
| Number To Decimal String |  | Функция кірісінде еніне (width) орнатылған мән негізінде санды (number) ондық сан жолына (decimal integer string) түрлендіреді. |
| Decimal String To Number |  | Функция жылжытудан (Offset) бастап жолдардың сандық символдарын (string)  бүтін ондық сандарға (number) түрлендіреді. |
| String To Byte Array |  | Функция жолдарды таңбасы жоқ байттық массивтерге түрелендіреді. Массивтегі әрбір байт жолдағы символға сәйкес келетін ASCII код болып табылады. |
| Byte array To String |  | Функция таңбасы жоқ байттық массивтерді жолдарға түрлендіреді. |

**2.4.4. Салыстыру функциясы**

Салыстыру функциясы скалярлы және векторлық айнымалылар және тұрақтылар арасындағы әр түрлі қатынастарды (тең, тең емес, үлкен, кіші, 0-ден үлкен, 0-ден кіші) тексеруге мүмкіндік береді. Сонымен бірге әр түрлі ақпараттар болуын тексеру, екі мәннің біреуін таңдау, максималды және минималды мәндерді таңдау, берілген диапазонда санның болуын анықтау сияқты функцияларды да атқарады.Салыстыру функциясы Programming-Comparison палитрасы ішінде орналасқан.

Барлық салыстыру функциялары полиморфты болып табылады. Скаляр, массивтер, кластерлер, тұрақтылар, логикалық, жолдық, сандық типтегі айнымалылар, жолдар, сілтемелер және құрылымдар, кестелерді салыстыру осы функция негізінде жүзеге асады. Тең функциясының жұмыс принципін қарастырайық. Егер х мәні у-ке тең болса, онда тең функциясы ақиқат мәнін қайтарады. Әйтпесе жалған мәнін қайтарады. Массивтерді және кластерлерді салыстыру кезінде элементтерді салыстыру режимінің (Compare Elements) мүмкіндігі және салыстыру жиынтығы (Compare Aggregates)қарастырылған. Режимді таңдау салыстыру жолы (Comparision Mode)көмегімен Элементтердің салыстыру режимін таңдау кезінде х=у? шығысы массив немесе логикалық скаляр кластерлерітүрінде көрініс табады.



20-сурет. Салыстыру функциясының палитрасы

9-кесте.Салыстыру функциясының сипаттамасы

|  |  |  |
| --- | --- | --- |
| Атауы | Пиктограмма | Сипаттама |
| Equal? |  | Егер х=у болса функция ақиқат мәнін қайтарады, әйтпесе жалған мәні қайтып келеді. |
| Not Equal? |  | Егер х≠y болса, онда функция ақиқат мәнін қайтарады, әйтпесе жалған мәні қайтып келеді. |
| Greater? |  | Егер x>y болса, онда функция ақиқат мәнін қайтарады, әйтпесе жалған мәні қайтып келеді. |
| Less? |  | Егер x<y болса, онда функция ақиқат мәнін қайтарады, әйтпесе жалған мәні қайтып келеді. |
| Equal To 0? |  | Егер x=0 болса ақиқат мәні қайтарылады, әйтпесе жалған мәні қайтарылады. |
| Not Equal To 0? |  | Егер х≠0 болса, онда функция ақиқат мәнін қайтарады, әйтпесе жалған мәні қайтарылады. |
| Select |  | Функция мәндерді s кірісінің күйіне байланысты t және f кірістеріне қайтарады. Егер s кірісіне ақиқат мәні орнатылса, онда функция t кірісіне енгізілген мәнді қайтарады.Егер де s кірісіне жалған мәні орнатылса, онда функция функция f кірісіне енгізілген мәнді қайтарады. Функция полиморфты болып табылады. |
| In Range and Coerce |  | Функция х шамасының жоғары шек (Upperlimit) және төменгі шек (lowerlimit)кірістеріне берілген мән аралығында жататынын және жатпайтынын анықтайды. Қосымша белгіленген шекте шығыс мәнін шектейді. |

**2.5.ВҚны өңдеу**

Беттік панель және блог-диаграмма объектілерін өңдеудің бірнеше әдістері бар.

**Объектіні құру**

Элементтер палитрасы көмегімен беттік панель мен блог-диаграммада объектіні құруда қосымша ретінде басқару және бейнелеу, тұрақтылар элементтерін олардың түйіндеріне тышқанды апарып, бір рет басып жіберу арқылы да шығарады. Ол үшін контексті мәзірден **Create** пунктін таңдау қажет.

**Constant** – тұрақтыны құру, блог-диаграммада ғана бейнеленеді.

**Control -** ВҚның беттік панелінде басқару элементін құру.

**Indicator –** беттік панельде бейнелеу элементін құру.

**Объектіні бөліп алу**

Блог-диаграммада, беттік панельде объектіні бөліп алу Орын алмастыру элементі арқылы жүзеге асады.

Объект бөлініп алынған кезде немесе белгіленген кезде оны пунктирлі сызықтар қоршап тұрады. Бірнеше объетіні бөліп алу үшін оларды белгілеп жатқан моментте **<Shift>** батырмасын басып тұру керек. Бірнеше объектіні меңзер арқылы біріктіріп бөліп алуға болады.

**Объетінің орнын ауыстыру**

Объектінің орнын ауыстыру Орын ауыстыру құралы арқылы жүзеге асады. Пернетақтадағы нұсқа (стрелка) көмегімен де объекті орнын ауыстыруға болады. Бірнеше пикселді қадаммен объектінің орнын ауыстырғанда **<Shift>**батырмасын басып тұру керек. Объектінің орнын ауыстыру кезінде **<Shift>**батырмасын басып тұрып таңдап алынған объектінің бағытын шектеп қоюға да (мысалы, тек горизонталь немесе вертикаль бойынша ғана) болады. Алғашында таңдалып алынған қозғалыс бағыты (горизонтальді немесе вертикальді) объектінің орын ауыстыруының бағытын анықтайды.

**Объектіні жою**

Объектіні жою үшін Орын ауыстыру элементі көмегімен объектіні белгілеп алып, содан соң пернетақтадағы**<Delete>**батырмасын басу керек немесе негізгі мәзірден **Edit»Clear** пунктін таңдау қажет.

**Әрекетті болдырмау және қайта қалпына келтіру**

Егер де ВҚны өңдеу кезінде қателік кеткен болса онда әрекетті болдырмауға немесе қайта қалпына келтіруге болады. Ол үшін негізгі мәзірден **Edit** (өңдеу) пунктін таңдап, әрі қарай **Undo** (болдырмау) және **Redo** (қайта қалпына келтіру) пунктерін таңдау қажет. Объектілерге жүргізілген болдырмау және қайта қалпына келтіру әрекеттерінің санын орнату үшін негізгі мәзірде **Tools»Options** пунктінде жүзеге асады. Ол үшін шыққан мәзірден Block Diagram бөлігін таңдап алу керек. Қайталаудың аз мәні компьютер жадының ресурстарын сақтап қалады.

**Объектіні көшіру**

Таңдап алынған объектінің орын ауыстыра отырып және бір уақытта **<Ctrl>**батырмасын басып көпшілік объектіні көшіруге болады. **(MacOS) <Option>** батырмасын содан соң **(Sun)** батырмасын,келесі **<Meta**> батырмасын, одан кейін **(Linux)** батырмасын басу керек. Таңдап алынған объектіні жаңа орынға орналастырып болған соң ең бірінші тышқан батырмасы содан кейін **<Ctrl>**батырмасы жіберіледі. Жаңа орында объект көшірмесі пайда болады.

Объектілерді стандартты әдіс бойынша да көшіруге болады. Ол үшін негізгі меңзерден **Edit>>Copy** және содан кейін **Edit>>Paste** пунктерін таңдаңыз**.**

**2.5.1. Объект белгілері**

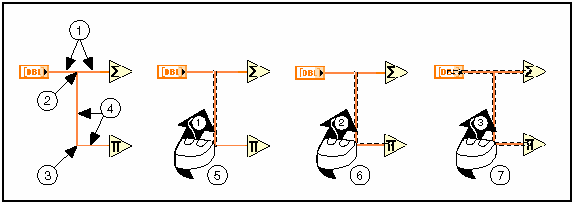
Белгілер объектілерді сәйкестендіру үшін қолданылады. LabVieW ортасында екі белгі қолданылады: бос және меншікті. Меншікті белгілер объектіге ғана тиесілі және тек сол объектімен бірге қозғалады, тек соны ғана суреттейді. Меншікті белгіні объектіден тәуелсіз орнын ауыстыруға болады. Бірақ объектінің орнын ауыстыру кезінде объектімен қоса белгі де орнын ауыстырады. Бос объектілер объектіге тиесілі емес, оларды құруға, орнын ауыстыруға, айналдыруға немесе жойып тастауға болады. Яғни, бос белгі объектіге тәуелсіз. Олар объектіні суреттеу, беттік панельге және блог-диаграммаға түсініктеме енгізу үшін қолданылады.

Бос белгіні құру үшін Мәтінді енгізу құралын қолдану керек. Бұл құралды белгілеп алып, беттік панель немесе блог-диаграмманың бос орнынан тышқан батырмасын басып, мәтінді енгізуге болады. Мәтінті енгізіп болған соң белгіден тыс кеңістікте меңзерді орналастыру керек немесе жабдықтар панеліндегі <Enter> батырмасын басуға болады.

**2.5.2. Мәліметтер өткізгіштерін жою және бөліп көрсету**

Мәліметтер өткізгіштері сегменті – бұл өткізгіштің жеке горизонтальді немесе вертикальді бөлігі. Екі сегменттің байланысу орны –бұл мәлімет өткізгіштерінің бұрылған жері. Екі, үш немесе одан да көп мәліметтер өткізгіштерінің түйіскен жері байланысу нүктесі деп аталады.

Мәліметтер өткізгіштері байланысу нүктесі арасындағы, мәліметтер терминалдары арасында және байланыс нүктелері, егер байланыс нүктесі жоқ болса, онда терминалдар арасындағы сияқты барлық сегменттерді құрайды. Сегментті белгілеу үшін Орын ауыстыру құралы қолданылады. Тышқан батырмасын екі рет басу мәлімет өткізгіштерін бөліп көрсетеді, ал үш рет басу-мәлімет өткізгіштер жиынын бөліп көрсетеді (сурет 21).

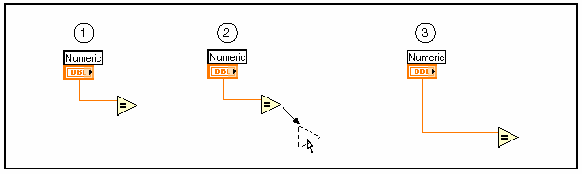


1-Сегмент; 2-байланысу нүктесі; 3-мәлімет өткізгіштерінің бұрылуы; 4-мәліметтер өткізгіші; 5-бөліп көрсетілген сегмент; 6-Бөліп көрсетілген мәлімет өткізгіші; 7-мәліметтер өткізгіші жиынын бөліп көрсету.

21-сурет . Мәліметтер өткізгіштерін жою және бөліп көрсету

**2.5.3. Мәлімет өткізгіштерін автомасштабтау**

Төмендегі суреттемеде көрсетілгендей (22-сурет) объектілердің орнын ауыстыру мәліметтер өткізгішін жоюға алып келмейді.



22-сурет.Мәлімет өткізгіштерін автомасштабтау

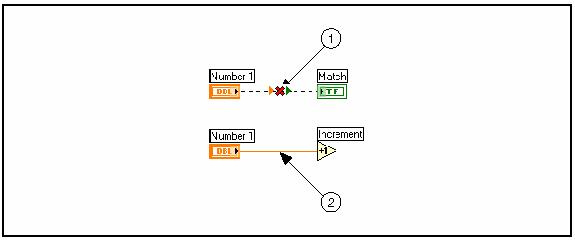
**2.5.4. Үзіліп қалған мәлімет өткізгіштері**

Үзіліп қалған мәлімет өткізгіші дәл ортасында қызыл кресті бар қара штрихті сызық түрінде көрсетіледі (сурет 23). Мәлімет өткізгіштерінің үзілуі әр түрлі жағдайлар әсерінен орын алады. Мысалы, әр түрлі мәлімет типтеріндегі объектілерді байланыстыру жағдайы 23-суретте көрсетілген.

Мәліметтер өткізгіштерінің үзіліп қалуын сипаттайтын негізгі себептер өткізгішке Байланыстыру құралын енгізген кезде терезеде қалқымалы ойға салу жазуы түрінде көрсетіледі. Өткізгішті Орын ауыстыру құралымен үш рет басу және **<Delete>**батырмасын басу белгіленген өткізгішті жояды. Барлық үзілген өткізгіштерді жою негізгі мәзірдегі Edit»Remove BrokenWires пунктерін орындаған кезде іске асады.



**Назар аударыңыз.** Негізгі мәзірдегі Remove Broken Wires пунктін қолдану қандай да бір қырағылықты талап етеді. Кей кезде өткізгіштер блог-диаграмма соңына дейін құрылып бітпеген жағдайда үзіліп кетуі мүмкін.



1-бұзылған өткізгіш; 2-дұрыс өткізгіш

23-сурет . Мәлімет өткізгіштерінің көрінісі

**2.5.5.Объекті өлшемін өзгерту**

Беттік панельдегі көптеген объектілердің өлшемін өзгертуге болады. Объектіні өлшемін өзгертуге дайындау үшін оған Орын ауыстыру құралын енгізу керек. Сол жақ суретте көрсетілгендей объектінің бұрыштарынан маркерлер пайда болады. Содан кейін меңзерді осы екі маркердің біреуіне орнату керек және тышқанның сол жақ батырмасын басып маркерді орналастыру керек, бұл жағдайда шрифт өлшемі өзгермейді. Өзгеретін өлшемнің аралық шектері штрихті сызықпен белгіленеді. Элемент қажетті өлшемге ие болған кезде тышқан батырмасын бос жіберу керек. Маркерлердің орнын ауыстыру кезінде <Shift> батырмасын басып тұру объектінің қатынасын сақтайды.

Блог-диаграммадағы объектілердің де (мысалы: құрылымдар және тұрақтылар) өлшемін өзгертуге болады.

**2.5.6. Кеңістікте объектілерді үлестіру және теңестіру**

Ось бойынша объектілерді теңестіру жабдықтар панеліндегі **Align Objects** пункті арқылы іске асырылады. Кеңістікте объектілерді бірқалыпты үлестіру үшін **Distribute Objects** пунктін пайдалану қажет.

**2.5.7. Беттік панельдің жұмыс бетінде объектінің орнын нақтылау және объектілерді топқа біріктіру, объектіні орналастыру тәртібін орнату**

Объектілер бір-бірін жауып қалған жағдайда объектілерді орналастыру тәртібін орнатуға болады. Ол үшін обектіні Орын алмастыру құралы көмегімен белгілеп алуға болады. Сосын негізгі мәзірдегі**Reorder** жабдықтық панелінің ішінен қажетті орнатуларды таңдау керек. Мысалы: **Move Forward** (алдыңғы бөлікке орнату), **Move Backward** (артқы бөлікке орнату), **Move To Front** (алдыға жылжыту), **Move To Back** (артқа жылжыту).

Беттік панельдің жұмыс кеңістігінде объектілерді топқа біріктіру және олардың орнын беттік панельде орнықтыру үшін **Reorder** мәзірінен **Group** (топтау), **Ungroup** (топтты бөліп жіберу), **Lock** (бұғаттау), **Unlock** (бұғаттан шығару) жабдықтарын таңдаған жөн.

**2.5.8. Бірнеше объектіні бір өлшемге келтіру**

Бірнеше объектіні бір өлшемге келтіру **Resize Objects** (объектінің өлшемін өзгерту) мәзірі арқылы іске асады. Таңдап алынған барлық объектінің өлшемін өзгерту ені немесе биіктігі, ені/биіктігі ішінен үлкен биіктіктегі немесе үлкен енді объектілерді өзгерту сияқты амалдар алдын-ала қарастырылған. Сонымен қоса, барлық таңдап алынған объекті үшін пиксельде өлшем беру мүмкіндігі бар.

Жеке объектілер объектінің өлшемін тек вертикаль немесе горизонталь бойынша ғана өзгертуге мүмкіндік береді. Мысалы: сандық басқару және бейнелеу элементтері. Кейбір объектілер өлшемін өзгерткен кезде де өз қатынасын сақтап қалады. Мысалы, егер өлшемі өзгертілетін объектілер ішінде, мысалы биіктігі бойынша өлшемі өзгеретін объект таңдалынып алынса, онда сандық тұрақты қатысады, LabVIEW оның өлшемін өзгертпейді, басқа объектілер өлшемін өзгертіп қана қояды. Басқа қолданымдар арасында немесе ВҚ арасында объектілерді көшіру

Бір ВҚдан екінші бір ВҚға объектіні көшіру үшін негізгі менюден **Edit»Copy** және **Edit»Paste** пунктерін таңдаймыз. Егер де екі ВҚда бір уақытта бір экран бетінде ашық тұрса, онда объектіні таңдап алып бір ВҚдан екінші ВҚға орнын ауыстырамыз.

**2.5.9. Объектілерді бояу**

ВҚдағы объектілердің түсін өзгертуге болады. Мысалы, блог-диаграммадағы мәліметтер терминалдары, мәліметтер өткізгіштері мәліметтер көрсетілген типке қатысты тек нақты бір түске ие болады. Жұмыс кеңістігінің немесе объектінің түсін өзгерту үшін Бояу құралын қолдану керек. Ол үшін таңдап алынған элемет үстінде тышқанның оң жақ батырмасын басып **Tools>>Options** мәзірінің ішінен **Colors** пунктін таңдайсыз. Сонымен бірге объектіні түссіз етуіңізге де болады. Ол үшін **Colors** менюінен **T** әрпін таңдайсыз.

**2.6. Программалау стилі**

LabVIEW ортасындағы блог-диаграмма басқа программалау тілдеріне қарағанда кодты бейнелейтін бірден-бір құрал болып табылады. Мәліметтер ағыны принципі программаны өткізгіштер, терминалдар, құрылымдар, түйіндер секілді элементтерде құрады. Әрбір элементтің өз тағайындалуы және бірегей мүмкіндіктері бар. Олар программисттің қиялын және мүмкіндіктерін ешқандай шектемейді. Нәтижесінде LabVIEWде программа жазудың бірнеше стильдері пайда болды.

Бір есепті бірнеше әдістермен шешуге болады. Бірақ, программалар функционалды түрде эквивалентті болады. Бір қарағанда, артықшылық, программалау және рәсімдеу (тағайындау)-әркімнің жеке жұмысы болып көрінеді. Шын мәнінде мұның барлығы тікелей программа сапасына әсер етеді.

LabVIEWдағы программалау стилі қолданудың қарапайымдылығы, эффективтілігі, сенімділігі және оқуға ыңғайлылығы, барлық қосымшалар кодтарының құрылымдылығымен тікелей байланысты. LabVIEWда программа жазу кезінде жақсы стильдің негізгі ережелерін қолданған жөн:

-Объектілерді тығыз орналастыру керек.

-Бір құрылғы ішіндегі құрылғының блог-диаграммасы өлшемін ені және биіктігі бойынша шектеу.

-Әрбір құрылғы ішіндегі құрылғы үшін белгіше (иконка) және түсінікті сипаттамалар болу керек.

-Өткізгіштер иілісінен және ілмектелінуін болдырмау керек.

-Параллель өткізгіштер бірдей қашықтықта өту керек.

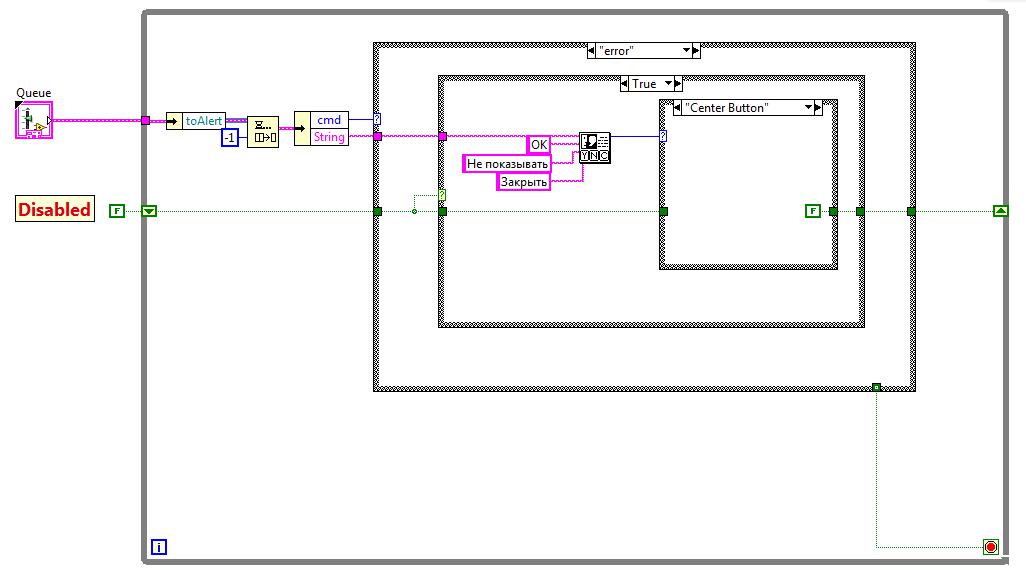
-Блог-диаграммада объектілердің беттесуінен алшақ болу қажет.

-Өткізгіш ұзындығын шектеу керек. Оның басы және аяғы экранда көрініп тұру қажет.

-Ұзын өткізгіштер үшін белгілер енгізу керек.

-Байланысқан мәліметтерді кластерге біріктіру.

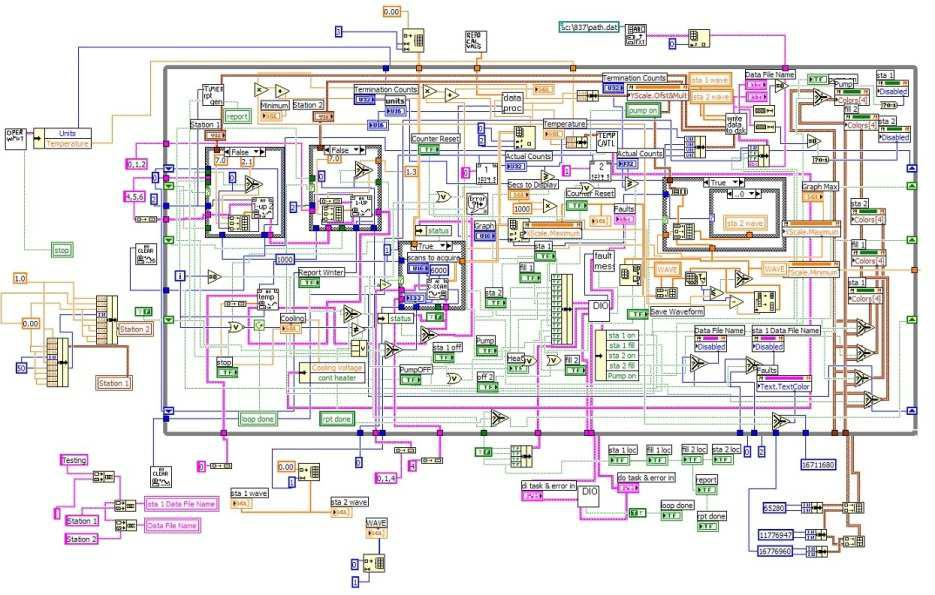
-Әрқашан мәлімет ағындарын солдан оңға қарай бағыттау қажет.



24-сурет. Жақсы программалау стилінің мысалы

Жақсы программалау стилінің көрінісі (24-сурет) жоғарыда көрсетілген. 24-суреттен көріп тұрғаныңыздай программа түсінікті әрі жинақы жасалынған.Программаны түсіну, түсіндіру өте оңай және ыңғайлы. Ал нашар программалау стилі көрінісі 25-суретте көрсетілген. Мұнда программа жинақсыз, түсініксіз формада жасалған.

Жоғарыдағы айтылған ережелерді сақтау программаны оқу және болашақта оны қолдану бойынша уақытты үнемдеу мүмкіндігін арттырады.



25-сурет. Нашар (жаман) программалау стилінің мысалы

**2-Лабораториялық жұмыс. Арифметикалық амалдарды орындау**

Төмендегі мысалдарда LabVIEW ортасында қарапайым электр тізбегі үшін арифметикалық амалдарды орындау қарастырылған. Тізбек үшін токтарды анықтау басты мақсат.

*Мысал 2.1*

*Электр энергия көзінен және қабылдағышынан тұратын виртуалды құрылғы құру қажет. Ол үшін:*

1.Энергия көзі ретінде беттік панельде басқару элементтерін таңдау қажет **Controls *=>*Numeric => Digital Control**. Ал қабылдағыш ретінде индикаторларды таңдаған жөн: **Controls => Numeric => Digital Indicator**.

2. Блог диаграмма бетінде бұл элементтердің иконкалары (бейнелері) пайда болады.

3.Басқару элементтері (энергия көзі) мен индикаторларды (қабылдағыш) байланыстыру үшін катушканы қолданамыз. Катушканы басқару элементіне алып барамыз, элемент жыпылықтап жанған кезде тышқанның сол жақ батырмасын басып, сымды келесі байланыстыратын элементке немесе индикаторға жалғаймыз. Катушка жалғанушы элементке жеткен кезде жалғанушы элемент жыпылықтайды. Сол моментте тышқанның батырмасын бос жібереміз. Егер байланыс дұрыс орнатылса, онда сымдарымыз қызыл түске ие болады. Ал қате жалғанғанда пунктирлі сызықтармен белгіленеді. Беттік панельде басқару элементтеріне мәндерін береміз.

4.Виртуалды құрылғыны іске қосу үшін беттік панельден іске қосу батырмасын (**Run**) басамыз. Сәйкесінше шығыс индикаторлар басқару элементіне берілген мәнді шығару керек.

*Мысал 2.2*

26-суретте электр тізбегінің сұлбасы берілген. Мұндағы ,, резисторлар (кедергілер), Е энергия көзі. Оны белгілі шама деп алайық.

Барлық тармақтағы токтарды анықтауды қарастырайық.



26-сурет. Электр тізбегінің көрінісі

Есептеулер Ом және Кирхгоф заңдарын қолдана отырып жүргізіледі:

;

;

.

Бұл есепті шығаруда мынадай операциялар орындалу қажет:

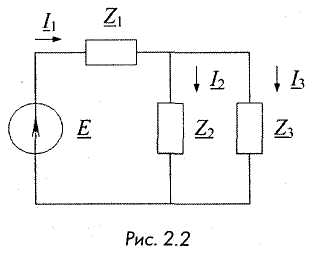
1.Беттік панельге тізбектегі кедергілер мәнін, электр қозғаушы күшін (ЭҚК), яғни басқару элементтерін шығару керек. Ал шығыс мәндері ретінде токтар мәнін көрсететін индикаторларды орналастыру қажет.

2.Содан соң есептеуді жүргізу үшін блог-схеманы жинау қажет. Ол үшін арифметикалық операцияларды орындаймыз. Арифметикалық операторларды шақыру мына амалдармен орындалады: **Functions => Numeric.** Басқару элементтерін арифметикалық оператордың кірістеріне жалғаймыз, ал арифметикалық оператор шығыстарына индикаторлар жалғанады. Блог схеманы құрастырып біткеннен соң виртуалды құрылғыны іске қосамыз.

*Есеп 2.1*

24-суреттегі электр тізбегі үшін токтарды есептеу қажет. Бірақ мұнда кедергілер мәні комплексті түрде болсын. Оның мәндерін өздеріңіз бере аласыздар.

Негізі 27-сурет жоғарыдағы 26-суретпен бірдей, тек кедергілер мен токтар мәні комплексті түрде шығу керек. Осы мақсатта барлық элементтердің типтерін өзгерту керек. Беттік панельден немесе блог диаграммадан басқарушы элемент және индикаторға курсорды алып барып, тышқанның оң жақ батырмасын басып **Representation => CDB** амалын орындаймыз. **Мұнда CDB (комплексті сан), ал DBL** (нақты сан). Полярлық жүйеге көшу үшін **All Functions => Complex=> Complex To Polar** таңдаймыз.



27-сурет. Комплексті сандар негізіндегі электр тізбегінің көрінісі

**2-Тапсырма. Математикалық формулалармен жұмыс.**

**Өтілген материалдар үшін қысқаша түсініктеме, кеңестер мен құпиялар**

**Көптеген құпиялар мен кеңестер** <Ctrl> батырмасын пайдалануды ұсынады. **(MacOS)** <Ctrl> батырмасының орнына <Option> батырмасын . <Meta> батырмасының орнына**(Sun) батырмасын**. <Alt> батырмасының орнына **(Linux)** .

Әрекет:

* Мәзірдегі көп қолданылатын негізгі пунктер ыстық батырмаларға эквивалентті.
* Мысалы, ВҚны сақтау үшін негізгі мәзірден File»Save пунктіне кіру керек немесе пернетақтадан <Ctrl-S> батырмасын басу қажет. Негізгі ыстық батырмалар:

**-<Ctrl-R>**, ВҚны іске қосу.

-**<Ctrl-E>**, блог-диаграмма мен беттік панель арасындағы ауыстыру.

**-<Ctrl-H>,** Context Help (контексті анықтама) терезесін экранға шығарады және жасырып қояды.

- **<Ctrl-B>** Үзілген мәлімет өткізгіштерін жою.

-**<Ctrl-F>** ВҚны, глобальді айнымалыларды, функцияларды, мәтіндерді немесе ВҚ жадындағы басқа да объектілерді іздейді.

-**<Ctrl-.>**Программаның орындалуын тоқтатады.

**-<Ctrl-V>**Объектіні орналастырады.

**-<Ctrl-Sh-F>**Іздеу нәтижесін шығарады.

-**<Ctrl-Z>**Соңғы әрекетті жою.

-**<Ctrl-Sh-Z>**Соңғы әрекетті қайталау.

-**<Ctrl-L>**Қателіктер тізімін шығару.

-**<Ctrl-T>**Беттік панель мен блог-диаграмманы бір экранда көрсету

-**<Ctrl-Q>**LabVieWдан шығу.

-**<Ctrl-S>**Программаны сақтау.

-**<Ctrl-W>**Файлды жабу.

-**<Ctrl-O>**Файлды ашу.

-**<Ctrl-I>**Программа қасиеттерін экранға шығару.

-**<Ctrl-/>**Активті түрде қолданылатын экранды толық өлшеміне дейін созады.

3-бөлім.

**Массивтер**

Бұл бөлімде бір мәлімет типіндегі элементтерді массивке біріктіру айтылған.

**3.1. Массивтер**

Массивтер бір типтегі мәліметтерді біріктіреді. Массив-бұл нақты өлшемдегі элементтер жиыны. Массив элементтері деп массив объектілерін құраушы топтарды айтады. Массивтің өлшемі-бұл жол (ұзындық) мен баған (биіктік), сонымен бірге тереңдік жиынтығы болып табылады. Массив бір өлшемді және одан да көп өлшемді иелене алады. Бірақ массив өлшемінің шекті мәні (2^31-1).

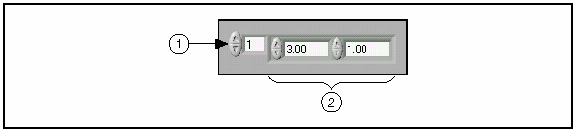
Массивтерді құрайтын мәліметтер әр түрлі типте болуы мүмкін: Нақты сандар, логикалық және жолдық. Сонымен қоса массив кластерлер және мәліметтерді графикалық бейнелеу элементтерінен де тұрады. Массивтерді бір типтегі мәлімет топтарымен жұмыс жасау кезінде және қайталанатын есептеулерден кейінгі мәліметтер жиынына қолданған ыңғайлы. Массивтер графиктен алынған немесе цикл жұмысы кезінде жиналған мәліметтерді сақтау үшін өте кең қолданылады.

Массивтерден тұратын массив құруға болмайды. Алайда кластер массив жасауға болады. Бірақ әрбір кластер бір немесе одан да көп массивтен тұрады.

Массивтің барлық элементтері реттік номерге (индекске) ие. Оларға осы реттік номері бойынша жүгінеміз. Массив элементтерінің нумерациясы әрқашан да 0-ден басталады. Олай болса массив индекстері 0 мен (n-1) диапазоны аралығында жатады. Мұнда n-массивтегі элементтер саны. Мысалы, массивте күн жүйесінің 9 планетасы бар делік. 9 планетаның массивтегі индекстері 0-ден 8-ге дейінгі шекте жатыр. Күн жүйесінің үшінші планетасы жердің массивтегі индексі 2-ге тең.

**3.1.1. Басқару және бейнелеу элементі массивін құру**

Басқару және бейнелеу элементі массивін құру үшін төменде мысалда көрсетілгендей (28-сурет) массив шаблонын **Controls >> Modern >>Matrix&Cluster** палитрасынан таңдап алып, беттік панельге орналастырасыз. Содан кейін массив шаблонының ішіне басқару және бейнелеу элементтерін қоясыз.

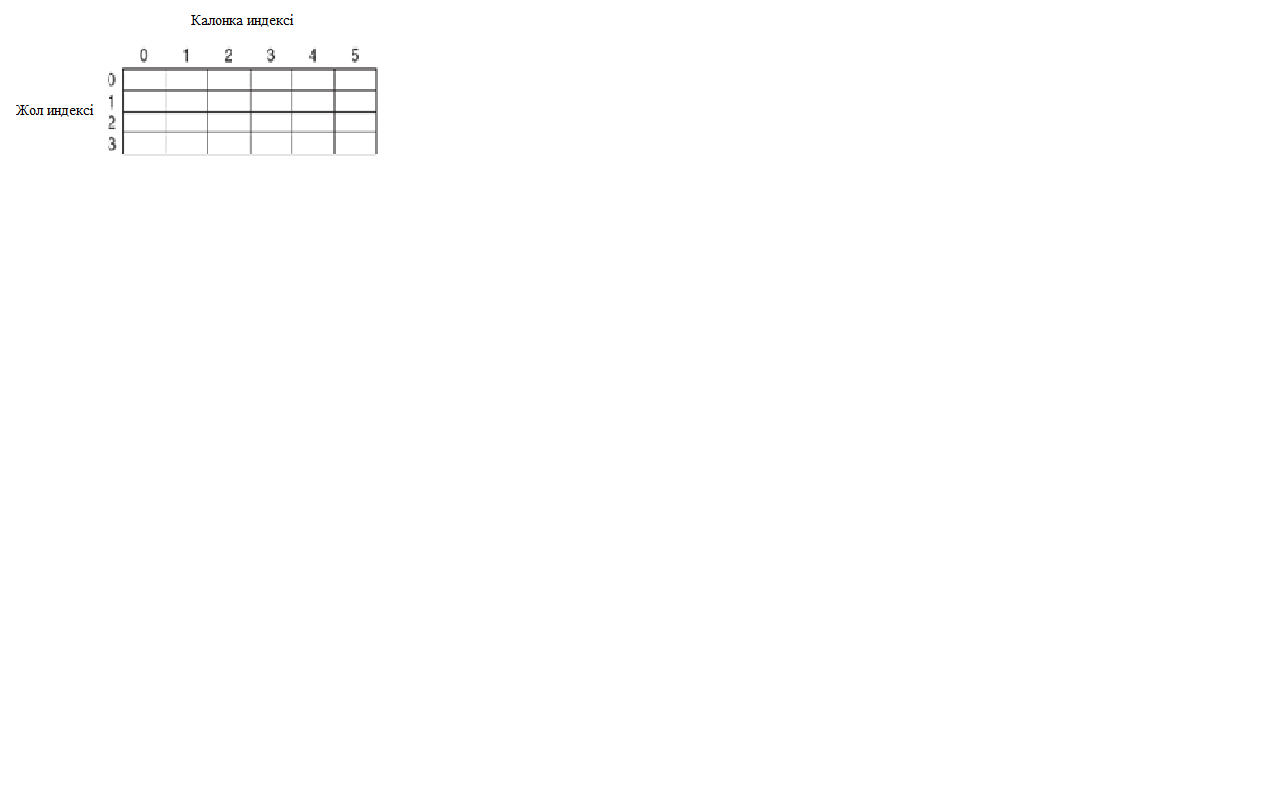


28-сурет. 1-массив элементінің индексі; 2-массив элементтерінің мәндері

Массив шаблонына объектілерді орнатуды массивті блог-диаграммада қолдануға дейін жасап қою керек. Егер бұлай жасамасаңыз, онда массив шаблоны инициализацияланбайды және оны қолдану мүмкін болмайды.

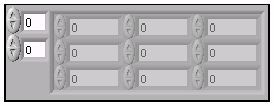
**3.1.2. Екі өлшемді массивтер**

Екі өлшемді массивте (2D) элементтер матрица түрінде сақталады. Осылайша элементтерді енгізу үшін жол мен баған индексін көрсету керек. Төменде 29-суретте 6 баған (ұзындық) және 4 жолдан (биіктік) тұратын екі өлшемді массив бейнеленген. Массивтегі элемент саны**-**24 (6\*4=24).



29-сурет. Массивке баған мен жолды беру көрінісі

Массивтің өлшемін көбейту үшін элемент индексіне тышқанды апарып,оң жақ батырмасын басып контексті меню ішінен Add Dimension пунктін таңдау керек. Сонымен бірге Орын алмастыру элементін қолдануға болады. Ол үшін элемент индексінің өлшемін өзгерту қажет. Төменде 30-суретте инициализацияланбаған басқару элементтерінен тұратын екі өлшемді массивтің көрінісі көрсетілген:

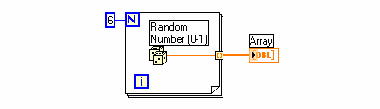


30-сурет. Массивтің беттік панельдегі көрінісі

**3.1.3. Тұрақты массивті құру**

Тұрақты массивті блог-диаграммада Function >> Programming >> Array палитрасының ішінен Array Constant шаблонын таңдап, оның ішіне сандық тұрақтыларды енгізу арқылы құрады. Тұрақты массивті ВҚның ішкі программаларына мәліметтерді тарату үшін қолдану ыңғайлы.

**3.2. Автоматты индексация**

For және While циклдары массивтерді автоматты түрде жинайды және олардың индексациясын өздерінің шекарасында жүргізеді. For және While циклдары жайлы 4-бөлімде толық қарастырылған. Бұл қасиет автоиндексация деп аталады. Массив мәліметтерінің терминалдарын циклдан шығу терминалына байланыстырғаннан кейін циклдің әрбір итерациясы массивте жаңа элемент тудырады. 31-суретте көрсетіліп тұрғандай массив мәліметтерінің терминалын циклдің шығыс терминалымен байланыстырып тұрған мәлімет өткізгішінің қалыңдай түскені көрініп тұр. Ал циклдан шығу терминалының өзі массив мәліметі терминалының түсіне сәйкес болады.

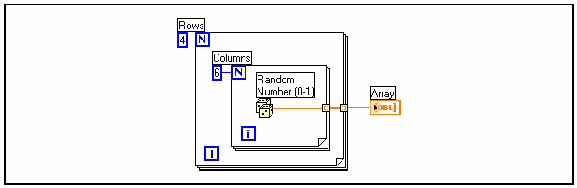
31-сурет. Массивтің блог-диаграммадағы жұмыс істеу принципінің көрінісі

Автоиндексация циклдан кіру / шығу терминалына тышқан батырмасын апарып, оң жақ батырманы басып контексті менюден Disable Indexing пунктін таңдау арқылы өшіруге болады. Автоиндексацияны тек соңғы мәнді білу қажет болған кезде ғана өшіру керек.

For циклі циклдармен жұмыс кезінде қолданылатын болғандықтан, бұл цикл үшін автоиндексация автоматты түрде іске қосылады. Автоиндексацияны іске қосу үшін циклдан кіру/шығу терминалына тышқанды апарып, оң жақ батырмасын басып контексті менюден Enable Indexing батырмасын басу қажет.

**3.2.1. Екі өлшемді (2D) массивті құру**

Екі өлшемді массивті құру үшін бірінің ішінде бірі орналасқан екі For циклін қолдану керек. 32-суретте көрсетілгендей ішкі цикл массив элементтерін жолға, ал сыртқы цикл массив элементтерін бағанға біріктіреді.

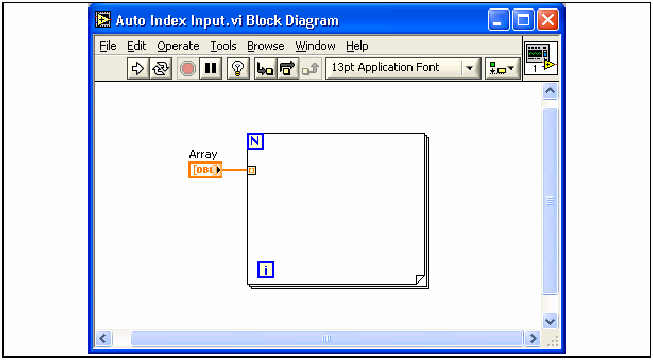


32-сурет. Екі For циклі көмегімен массив құру көрінісі

**3.2.2. Циклдің итерация саны терминалы мәнін орнату үшін автоиндексацияны қолдану**

For циклінің кіріс терминалына қосылған массивтің автоиндексациясы қосылулы тұрған кезде LabVIEW цикл итерация саны терминалының мәнін (N) автоматты түрде массив өлшеміне тең етіп алады. Сондықтан N терминалына мән берудің қажеті болмай қалады.

Келесі мысалда For циклі массивте қанша элемент болса, сонша рет орындалады. Егер де Run батырмасы ажыратылған бейнеге ие болса, онда циклдің итерация саны терминалы қосылмай тұр деген сөз. Алайда біз қарастыратын мысалда нұсқа еш ажыратылмай, бүтін қалпында (33-сурет).



33-сурет. Циклдің итерация саны терминалы мәнін орнату үшін автоиндексацияны қолдану көрінісі

Мысал үшін, егер циклдің кіріс терминалына 10 элементтен тұратын массивті байланыстырып, ал итерация санын 15ке тең десек, онда цикл 10 итерация орындайды (33-сурет)..

**3.3. Массивтермен жұмыс жасайтын функциялар**

Массивтерді құру және басқару үшін функция палитрасында орналасқан (**Function >> Programming >> Array**) функциялар қолданылады. Өте көп қолданылатын функциялар:

**Array Size** –массивтегі әр өлшемді элементтердің санын көрсетеді. Егер массив n-өлшемді болса, онда Array Size функциясының шығысында n-элементтен құралған массив шығады. Мысалы, төменде көрсетілген Array Size функциясы 3 өлшемді массивті береді.

|  |  |  |
| --- | --- | --- |
| 7 | 4 | 2 |



**Initialize Array**- **element**деген атауға ие мәліметтерді енгізу жолындағы мәндерді әрбір элементпен инициализациялайтын n-өлшемді массивті құрайды. Массивтің өлшемін ұлғайту үшін мәліметтерді енгізу жолын созсаңыз жеткілікті. Мысалы, егер **Initialize Array** функциясы үшін параметрлердің келесідей мәндері берілсе: **element** жолына 4 мәні беріледі, ал **dimension size** жолына (егер ол біреу болса) онда шығысында төменде көрсетілгендей массив шығады.

|  |  |  |
| --- | --- | --- |
| 4 | 4 | 4 |

**Build Array**- n-өлшемді массивке элементтерді енгізеді немесе бірнеше массивтерді біріктіреді. Функцияның өлшемін өзгерту мәліметтерді енгізу жолының санын арттырады. Бұл сәйкесінше енгізетін элементтердің санын көбейтеді. Мысалы, алдыңғы екі массивті біріктіру үшін **Build Array** функциясы қолданылады. Біріккен массивтің графикалық бейнесі төменде көрсетілген:

|  |  |  |
| --- | --- | --- |
| 7 | 4 | 2 |
| 4 | 4 | 4 |

Кіріс мәліметтерді дәл сондай өлшемдегі ұзын массивке біріктіру үшін төменде көрсетілгендей функцияға тышқан батырмасын апарып, оң жағын басу арқылы контексті мәзірден**Concatenate Inputs** пунктін таңдау қажет.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | 4 | 2 | 4 | 4 | 4 |



**Array Subset**- **index** жолына келіп түскен индекстен және **length** жолында көрсетілген ұзындықтан бастап массив бөліктерін көрсетеді. Мысалы, алдыңғы массивтің кіріс жолына **Array subset** функциясын, ал **index** жолына 2, **length** жолына 3 мәнін берсек, онда мәліметтерді енгізу жолында төменде көрсетілгендей мәндер пайда болады:

|  |  |  |
| --- | --- | --- |
| 2 | 4 | 4 |



**Index Array** - **index** мәліметтерді енгізу кірісіне енгізілген, индексіне сәйкес келетін элементтерді береді. Мысалы , алдыңғы массивтер үшін егер де мәліметтерді енгізу кірісі **indexке** 0 мәнін берсек, онда **Index Array** функциясы 2 мәнін көрсетеді. **Index Array** функциясын екі өлшемді массивтен жолдарды және бағандарды белгілеу үшін және массив ішіндегі массивті бейнелеу үшін қолданады. Ол үшін екі өлшемді массивті функцияның мәліметтерді енгізу жолына беру керек. **Index Array** функциясы екі **index** жолынан тұру қажет. **Indexтің** жоғары бөлігі жолды көрсетсе, ал төменгі бөлігі бағанды көрсетеді. **Indexтің** екі жолын да пайдалануға болады. Ол кезде бір уақытта жолды, бағанды немесе жеке бір элементті таңдауға болады. Мысалы, функцияның мәліметтерді енгізу жолына төменде көрсетілген массивті енгізсек:

|  |  |  |
| --- | --- | --- |
| 7 | 4 | 2 |
| 4 | 4 | 4 |

Егер де **index** жолына 0 саны берілсе, онда **Index Array** функциясы мәліметтерді енгізу жолына келесідей массивті береді:

|  |  |  |
| --- | --- | --- |
| 7 | 4 | 2 |

**3.4.Полиморфизм**

**Function >> Programming >> Numeric** палитрасының ішінде орналасқан арифметикалық функциялар полиморфты болып табылады. Бұл функцияның кіріс жолына әр түрлі типтегі мәліметтер (скалярлы шамалар, массивтер) келіп түсетінін көрсетеді. Мысалы, Add функциясын массивке скалярлы шаманы енгізу үшін және екі массивті қосу үшін қолдануға болады.

1-массив:

|  |  |  |
| --- | --- | --- |
| 1 | 3 | 2 |

Add функциясының бір кірісіне скалярлы шама 2-ні, ал екінші кірісіне 1-массивті қоссаңыз, онда төменде көрсетілген массив түзіледі және функция массивтің әрбір элементіне 2 санын қосып шығады, нәтижесінде массив мынадай түрге ие болады:

2-массив:

|  |  |  |
| --- | --- | --- |
| 3 | 5 | 4 |

Add функциясының кірісіне алдыңғы екі массивті берсек, онда функция бірінші массивтің әрбір элементін екінші массивтің сәйкес келетін элементтерімен қосады және төменде көрсетілген массивті шығарады:

3-массив:

|  |  |  |
| --- | --- | --- |
| 4 | 8 | 6 |

Егер де Add функциясының көмегімен екі әр түрлі өлшемдегі массивті қоссақ, мысалы, алдыңғы (3-массив) және төменде көрсетілген массивті қоссақ:

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | 1 | 2 | 3 |

Онда функция бірінші массивтің әрбір элементін екінші массивтің сәйкес келетін элементеріне қосады және екі массив өлшемінен екі есе кіші өлшемдегі келесі бір массивті шығарады:

|  |  |  |
| --- | --- | --- |
| 7 | 9 | 8 |

Арифметикалық амалдар кластерлермен де осылайша жұмыс жасайды. Кластерлер жайлы келесі бөлімнен білуге болады.

**3.5. Формула түйіні**

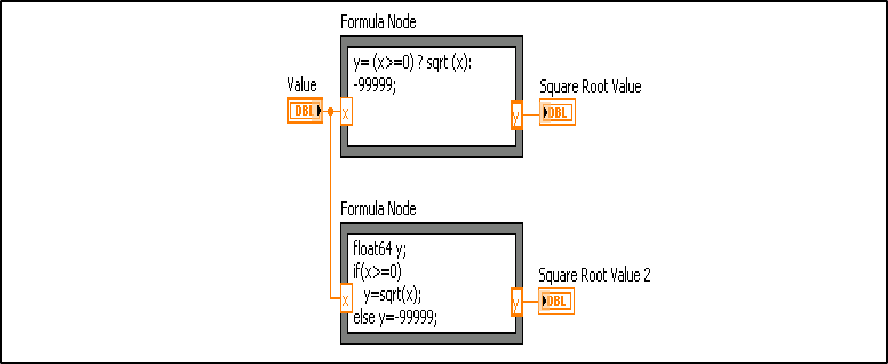
Формула түйіні (**Formula Node**) блог-диаграммада математикалық амалдарды мәтіндік түрде орындауға арналған. Формула түйінін мына жағдайда: өрнек бірнеше айнымалыға ие болса немесе жеткілікті күрделі болған жағдайда қолданған ыңғайлы. Алгоритмді құру процесін жеделдету үшін жазылып қойылған мәтіндік математикалық кодтарды көшіріп, формула түйіні ішіне орналастыруға болады.

Формула түйінінің кіріс және шығыс терминалдарын құру үшін меңзерді формула түйінінің кез келген жеріне апарып, тышқан батырмасының оң жақ батырмасын басу қажет. Контексті мәзірден міндетті түрде **Add Input** немесе **Add Output** пунктерін таңдау керек. Содан кейін кіріс және шығыс айнымалыларын бересіз. Негізгі теңдеу құрылымның жұмыс ортасына жазылады. Әрбір теңдеу соңы нүктелі үтір (;) белгісімен аяқталу қажет.

Сонымен бірге формула түйіні шешім қабылдау үшін де қолданылуы мүмкін. Төмендегі блог-диаграммада формула түйінінде **if-then** операторын қолданудың екі эквивалентті әдісі көрсетілген.

Формула түйіні әр түрлі математикалық операцияларды орындауға мүмкіндік береді.

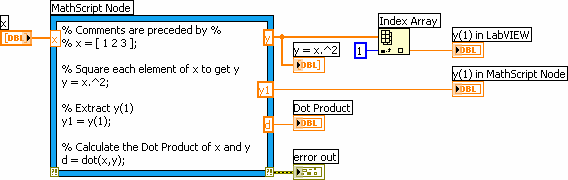
Ескерту. **Function>>Express>>Arith/Compare** палитрасында орналасқан **Formula Express** виртуалды құрылғысы, шындығында, LabVIEWдегі ғылыми калькулятор болып табылады. Ол формула түйіні орындай алатын бірнеше амалдарды орындайды. Алайда бір рет жұмыс жасағанда тек бір амалды ғана орындай алады.



34-сурет. Формулалық түйін

**3.6. Математика түйіні (MathScript Node)**

Формула түйініне ұқсас Математика түйіні блог-диаграммада мәтіндік түрде математикалық амалдарды орындауға арналған. Формула түйінінен айырмашылығы математика түйіні көбірек кең мүмкіндікке ие. Математика түйінін қолдана отырып, сіз LabVIEWге LabVIEW MathScript, MatLab, Xmath ортасында жазылған кодтарды көшіріп, тасымалдай аласыз.

Төменде математика терезесінің LabVIEW ортасындағы көрінісі көрсетілген:

35-сурет. Математика түйіні

MathScript Node түйінін MatLabта жазылған кодтарды құру, жүктеу және өңдеу үшін қолдануыңызға болады. Мұнда назар аударарлық жайт, егер сіздің компьютеріңізде Матлаб программасы орнатылмаған жағдайда да бұл операциялар орындалады. Алайда MathScript функциясы Матлаб иеленетін барлық функцияларды иелене алмайды.

**LabVIEW MathScript терезесінің қолданылуы**

LabVIEWде математикалық кодты құру үшін LabVIEW MathScript Window терезесін қолданыла ласыз. Бұл терезені ашу үшін жабдықтар сызғышында орналасқан Tools>>MathScript Window пунктін таңдаңыз. LabVIEW MathScript Window сіз анықтайтын айнымалылар тізімін, шақырылатын командалар тарихын және шығыстағы мәнді генерациялайды. Таңдап алынған айнымалыларды бейнелейді.

**Өтілген материалдардың қысқаша мазмұны, кеңестер және құпиялар**

* Массивтер бір типтегі мәлімет тобын біріктіреді. Массивті құрайтын мәліметтер кез келген типте бола алады: нақты сандық, логикалық және жолдық. Массив сонымен бірге сигналдық мәліметтерді және кластерлерді құрай алады.
* Массивтің бірінші элементінің индексі әрқашанда 0-ге тең. Олай болса массив индекстері 0-ден (n-1)-ге дейінгі диапазонда жатыр. Мұндағы n- массивтегі элементтер саны.
* Массив шаблонына объектіні орнатуды беттік панельде блог-диаграммаға массивті орнатпастан бұрын жасау керек. Әйтпесе блог-диаграммада массив шаблоны инициализацияланбайды.
* Басқару және бейнелеу элементтерінен тұратын массивті құру үшін оның шаблонын **Controls >> Modern >> Array, Matrix & Cluster** палитрасы ішінен таңдап аламыз. Оны беттік панельге орналастырамыз. Содан соң массив шаблоны ішіне басқару немесе бейнелеу элементтерін орнатамыз.
* Егер де массивті **For** немес **While** циклінің кіріс терминалымен байланыстырсақ, онда өшірілген автоиндексация кезінде массивтің әрбір элементін оқуға және өңдеуге болады.
* Массивтерді құру және басқару үшін **Function >> Рrogramming >> Array** палитрасының ішіндегі функциялар қолданылады.
* LabVIEW автоиндексацияны автоматты түрде жүзеге асырады. **For** циклі үшін автоиндексация таңдау бойынша қосылған. Ал **While**  циклі үшін автоиндексация таңдау бойынша ажыратылған.
* Полиморфизм ұғымы функцияның мәліметтерді енгізу жолы, мәліметтер құрылымы бойынша әр түрлі болуы мүмкін (скалярлы шамалар, массивтер) екендігін көрсетеді.

3-Лабораториялық жұмыс. Электр тізбегіндегі токты анықтау

*3.1 – мысал . Формулалық түйінді қолданып тізбектегі токты анықтау.*



36-сурет. Электр тізбегінің көрінісі

1. 36-суреттегі тізбек үшін токтарды Ом және Кирхгоф заңдарын пайдаланып есептеуге болады:

;

; (1)

.

Мұнда ,, кедергілердің мәндері және Е электр қозғаушы күшінің мәні таңдалынып алынады.

2. Бұл формулаларды формулалық түйінді қолданып есептеуге болады. Формулалық түйін блог панельге **All Functions => Structures => Formula Node** пунктерінтаңдау арқылы шақырылады. Пайда болған формулалық рамканы бізге қажетті өлшемге дейін созуға болады. Және сол рамка ішіне (1)-формуланы жазамыз. Белгісіз шама формулалық түйіннің сол жағына жазылады. Әрбір формула жеке-жеке қатарға жазылады және әрбір формула соңында нүктелі үтір қойылады.

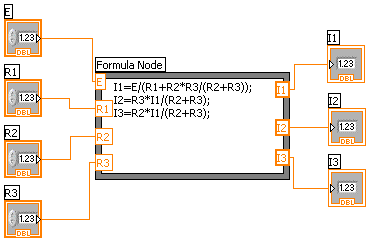
3. Енді формулалық түйін ішінде жазылған формуланы кіріс және шығыс элементтерімен байланыстыру үшін курсорды формулалық түйін рамкасына апарып тышқанның оң жақ батырмасын басып **Add Input (кірісті енгізу)** амалын таңдаймыз. Ал шығыс мән үшін **Add Output (шығысты енгізу)** амалын орындаймыз. Пайда болған рамка ішіне мәндер белгіленіп жазылады.

4. Кіріс рамкаға цифрлық басқарушы элементтері қосылады, ал шығысқа- индикаторлар. Кіріс, шығыс мәндерді рамканың кез-келген жеріне орналастыра берсе болады. Рамканың атаулары формулалық түйін атауларымен сәйкес болу керек. Кіріс және шығыс шама үшін бір белгілеуді қолдануға мүмкіндік беріледі.

5. Басқарушы элементтер көмегімен мәндер енгізіледі. Виртуалды іске қосу батырмасы арқылы виртуалды құрылғыны іске қосамыз.

Формулалық түйіннің көрінісі 37- суретте көрсетілген.

Ескерту. Формулалық түйін комплексті сандар үшін қолданылмайды**.**



37-сурет. Формулалық түйіннің блог-диаграммадағы көрінісі

*3.2-мысал. Алгебралық теңдеуді матрицалық формада шешу*



38-сурет. 3.2-мысал үшін қажетті электр тізбегі

38-суреттегі тізбек үшін токтарды есептеуде Ом және Кирхгоф заңдары қолданылып, теңдеулер құрылады:

** (2)

Бұл теңдеулерді матрицалық формада жазуға болады.

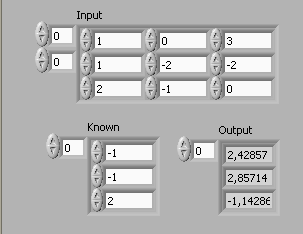
**.**  (3)

LabVIEW ортасында сызықты алгебралық теңдеулерді шешу үшін **All Functions => Analyze => Mathe­matics => Linear Algebra => Solve Linear Equatіons** жолдары бойынша шақырылатын«сызықты теңдеулерді шешу» элементі қолданылады. Олардың терминалын ашып қою үшін белгіше (иконка) бетіне меңзерді апарып, тышқанның оң жақ батырмасын басып, әрі қарай пайда болған мәзірден **Visible Items => Terminals** пунктін таңдайсыз.Терминалы ашып көрсетілген элемент көрінісі 39-суретте көрсетілген.

linAlg

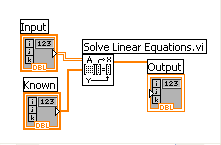
39-сурет.Терминалы ашып көрсетілген элемент көрінісі

Терминалдар тағайындалуын анықтау үшін белгіше бетінен тышқанның оң жақ батырмасын басу және **Open Front Panel** алдыңғы беттік панельді ашу қажет.Алдыңғы панель 40-суретте көрсетілгендей түрге ие болады. (2)-теңдеу мынадай негізде қорытып шығарылады: **Input Matrix** (теңдеудің бірінші матрицасы) коэффициенттер матрицасы - сол жақ жоғары бөліктегі терминалға беріледі, **Known Vector** (оң жақ) берілген әсер ету векторлары -сол жақтағы орта терминалға жүргізіледі. Ал **Solution Vector** (ізделініп отырған токтар векторы) нәтижені шығарушы вектор оң жақ жоғары терминалдан алынады.



37-сурет. Беттік панель көрінісі

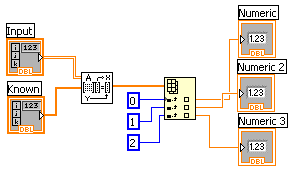
Матрицаны беттік панельге шығару үшін **Array** (массивті құру) элементі шақырылады.Элемент беттік панельге мына операцияларды орындау арқылы шығарылады: **All Controls => Array&Cluster => Array.** Сандық басқару элементі (коэффициенттер матрицасы және басқарылатын әсер векторлары үшін) немесе сандық индикаторды (ізделініп отырған токтар векторлары үшін) енгізетін бос ұяшық пайда болады. Содан кейін элемент қажетті өлшеміне дейін созылады. Ол үшін жылжыту (орын ауыстыру) инструменті қолданылады. Басқарушы элементтер және индикаторлар терезелері сұры түсті иеленеді. Оған сандық мәліметтерді енгізгеннен кейін түсі ақ болады.**Array** элементін беттік панельге енгізгеннен кейін оның терминалы блог-диаграммада пайда болады. Барлық қажетті байланыстар орындалатын блог-диаграмма және беттік панель көрінісі 41-суретте көрсетілген.



41-сурет. Блог-диаграммада жиналатын блог-схема көрінісі

Операцияларды орындау нәтижесінде **Output**  шешімдер векторы шығады. Бұл бағанда токтар мәні көрсетілген: =3A, =2A, =1A .

Бұл бағаннан белгісіздерді жеке белгілеп алуға болады. Ол үшін массив элементін шығару функциясы **All Functions => Array => Index Array** қолданылады.Функцияның шығыстары индикация элементтерімен қосылады (42-сурет). Программаны іске қосқаннан кейін индикаторлар кіріс вектор жолдарының мәндеріне сәйкес келетін мәндерді көрсетеді.



42-сурет. **Index Array**функциясының көрінісі

**Solve Linear Equation** программалық инструменті әмбебап қасиетке ие екенін айта кету керек. Яғни, оның бір қасиеті асыра анықталған және анықталмаған теңдеулер жүйесін шеше алады. Бірінші жағдайда теңдеуді қанағаттандыратындай шешім , ал екінші жағдайда мүмкін болатын шешімдердің біреуі іздеп табылынады.

3-тапсырма. Бір өлшемді массивтермен есептер шығару

**Қосымша жаттығу.**

1. Массивтегі әрбір элементтің ретін өзгертетін ВҚ құрыңыз. Массивтегі элементтер саны 100 кездейсоқ сан болсын. Мысалы, 0 индексті массив элементі 99 индексті болатын массив элементіне айналсын. Ал 1 индексті массив элементі 98 индексті массив элементіне айналады және тағы да сол сияқты.

**Кеңес.** Массивтегі мәліметердің ретін өзгерту үшін **Function >> Programming >> Array** палитрасындағы **Reverse ID Аrray** функциясын қолданған дұрыс.

2. Кездейсоқ сандар генераторынан анықталатын мәліметтер реттілігі үшін (санедар үшін) сол моменттегі орта, максимум және минимум мәндерді анықтайтын ВҚны құрыңыз.

3.Термометр көмегімен температура мәні массивін жинайтын ВҚ құрыңыз. Массив өлшемін беттік панельдегі басқару элементі көмегімен анықтаңыз. **Initialize Array** функциясының көмегімен дәл бірінші массив өлшеміндегідей массивті құрыңыз. Бірақ бұл массив элементтерінің барлығының мәні 10-ға тең болсын. Екі массивті қосыңыз. Пайда болған массивтің өлшемін анықтаңыз. Оның орта мәнін табыңыз. Беттік панельге температура мәні алынған массивті, инициализацияланған массивті, соңғы массивті және орта мәндер массивтерін шығарыңыз.

4. 3 жол және 10 бағаннан тұратын кездейсоқ сандарды генерациялайтын екі өлшемді массивті құратын ВҚ құрыңыз. Осыдан кейін ВҚ барлық жолдарды индекстейді және әрбірінің осциллограммасын гарфикте көрсетеді. Беттік панельде 3 график болу керек.

5. Бір өлшемді массивті генерациялайтын және жұптары бойынша 0 индексінен бастап элементтерді көбейтетін (0 және 1 индексті және әрі қарай ), содан соң нәтижені бейнелеу элементі көмегімен құрылған массив түрінде көрсететін вқны құрыңыз. Мысалы, кіріс массиві {1,23,10,5,7,11} мәндерін иеленеді, ал нәтижесінде {23,50,77} мәндері бар массив пайда болады.

**Кеңес.Function >> Programming >> Array** палитрасында орналасқан **Decimate ID array** функциясын қолданыңыз.

6. 3х3 өлшемді екі матрицаны жеке-жеке элементтері бойынша көбейтетін ВҚ құрыңыз.

**4-бөлім.**

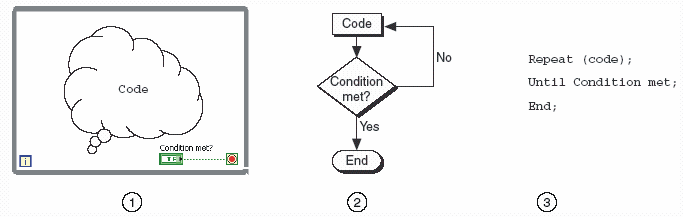
**Бірнеше рет қайталану және циклдер. For және while циклдері**

Құрылымдар циклдық операторлардың және мәтіндік программалау тілдеріндеқолданылатын **Case** (нұсқалар) операторлардың графикалық көрінісі болып табылады. Блог-диаграммадағы құрылымдар мәліметтер ағындарымен қайта-қайта операция жүргізуді орындау үшін және операция орындалу үшін шарттарды орнату үшін қолданылады. LabVIEW ортасы келесідей құрылымдардан тұрады:  **While** циклі (шарт бойынша), **For** циклі (белгіленген итерация санымен), **Case** құрылымы (нұсқа), **Sequence** құрылымы (тізбектілік), **Event** құрылымы (оқиғалар), сонымен бірге **Formula Node** (формулалық түйін), **MathScript Node** (математикалық түйін).

Бұл бөлімде While циклі (шарт бойынша), **For** циклі (белгіленген итерация санымен), сонымен бірге осы құрылымдармен өте көп қолданылатын **Shift Register** (ығысу регистрі) және **Feedback Node** (кері байланыс түйіні) қарастырылған.

**4.1. While циклі (шарт бойынша)**

**While (While Loop)**циклі мәліметтермен әр түрлі амалдардылогикалық шарт орындалмайынша бірнеше реторындайды.**While** циклі мәтіндік программалау тілінде қолданылатын **Do** және **Repeat Until** циклдеріне ұқсайды. Төмендегі иллюстрация LabVIEW ортасындағы **While** циклін (1), **While** циклінің эквивалентті блог-схемасын (2), **While** циклі жұмысының мәтіндік аналогты кодын (3) демонстрациялайды (43-сурет):

****

43-сурет. 1-LabVIEWортасындағы While циклі ; 2-Блог-схемасы; 3-Мәтіндік аналог коды

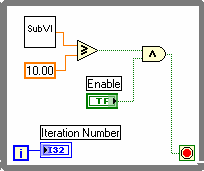
**While** циклі функция палитрасының ішінде орналасқан. Блог-диаграммада жұмыс бетінің кез келген жерінен тышқан батырмасының оң жағын басу арқылы функция палитрасын ашып, мына пункті таңдау керек: **Function >>Programmming>>Stuctures** . Функция палитрасынан циклді таңдап болған соң меңзер көмегімен цикл ішіне қажетті блог-диаграмманың бөлігін бөліп қою керек. Тышқан батырмасын босатқаннан кейін блог-диаграмманың белгілеп алған аумағы цикл денесінің ішіне орналасады. Блог-диаграмма объектілерін цикл ішіне енгізу объектіні көшіру арқылы іске асады.



**Кеңес.While** циклі ең көп дегенде 1 рет қана орындалады.

**While** циклінің блог-диаграммасы циклден шығу шарты орындалмағанша жұмыс жасай береді. Әдепкі қалпы бойынша шығу шарты терминалы сол жақта көрсетілгендей көрініске ие. Бұл дегеніміз шығу шарты терминалына **TRUE** мәні келіп түскенше цикл орындала береді деген сөз. Бұл жағдайда шығу шарты терминалы **Stop if TRUE** (тоқтау егер ақиқат болса) терминалдары деп аталады.

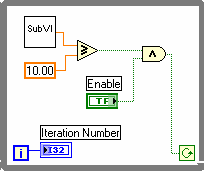
Итерацияны есептеуіш терминалы, сол жақта көрсетілгендей орындалған итерация сандары мәнін көрсетеді. Терминалдың бастапқы мәні әрқашанда нөлге тең. 44-суретте көрсетілген блог-диаграммадағы While циклінен шығу шарты ВҚның ішкі программасының мәні 10,00 және **Enable** басқару элементі терминалы күйімен сипатталатын шығыс параметрі мәнімен анықталынады. **And** (логикалық Және) функциясы егер мәліметтер кірісіне кірген екі функция мәні **TRUE** (ақиқат) болса, онда шығысында **TRUE** мәнін қабылдайды. Басқа жағдайда шығысындағы функция **FALSE** мәнін шығарады және цикл жұмысы тоқтатылады.



44-сурет. While циклінің блог-диаграммадағы көрінісі

Алдыңғы мысалда циклдің үздіксіз орындалуына ықтималдылық жоғары болды. Әдетте бір уақытта орындалатын екі шарттан гөрі жалғыз циклдан шығу шартын алуға тырысады.

Шарт терминалына тышқан батырмасын апарып оң жақ батырмасын басу арқылы контексті мәзірді шақырып, оның ішінен **Continue if True** (ақиқат болса әрі қарай жалғастыру) пуктін таңдау керек. Сонымен бірге басқару құралын пайдалануға да болады. Басқару құралын пайдалансаңыз, онда ішінен **Continue if True** (ақиқат болса әрі қарай жалғастыру) элементі сол жақта көрсетілген элементке ауысады. Нәтижесінде циклден шығу шарты болып шығыс терминалына келіп түскен шарт **False** мәні алынады. Мұның блог-диаграммадағы көрінісі мынадай (45-сурет):

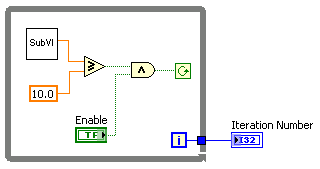


45-сурет. Циклдан шығу шартын берудің графикалық көрінісі

While циклі ВҚның ішкі программасының шығысы10 санынан кіші болмайынша орындала береді.

**4.1.1.Циклдің кіріс/шығыс мәліметтер терминалдары**

Мәліметтер **While** цикліне циклдің кіріс/шығыс мәліметтер терминалдары көмегімен кіруі немесе шығуы мүмкін. Циклдің кіріс/шығыс мәліметтер терминалдары құрылымға мәліметтерді жеткізеді немесе құрылымнан мәліметтерді алады. Циклдің кіріс/шығыс мәліметтер терминалдары While циклінің аумағында тұтас тіктөртбұрыштар тәріздес бейнеленеді. Тіктөртбұрыш түсі терминал арқылы таратып жатқан мәлімет типіне байланысты болады. Егер мәліметтер циклге кіріс/шығыс терминалдарынан келіп түссе, онда циклдің орындалуы терминалға мәлімет түскен сәттен басталады. Келесі блог-диаграммада итерацияны есептеу терминалы циклдің шығыс терминалымен байланысқан. Циклден шығу терминалының мәні бейнелеу элементіне шарт орындалғанша дейін түспейді немесе шығарылмайды.

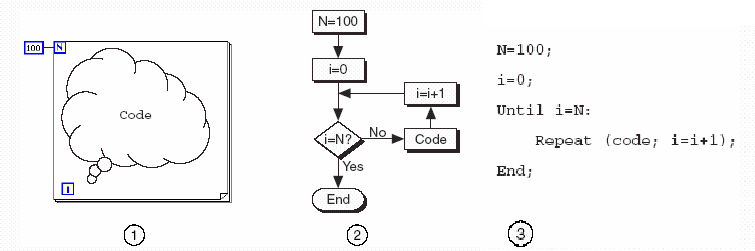


46-сурет. While циклдық оператордың жұмысын тоқтату көрінісі

Бейнелеу элементінде тек итерацияның соңғы мәні ғана бейнеленеді (46-сурет).

**4.2. For циклі (белгіленген итерация санымен)**

For (For Loop) циклі (белгіленген итерация санымен) қандай да бір ақпарат ағындарымен нақты бір санға тең қайталанатын операцияларды орындайды. For циклінің LabVIEW программалау ортасындағы көрінісі, эквивалентті блог-схемасы, мәтіндік аналогтік коды төменде көрсетілген (47-сурет):



47-сурет. 1. For циклінің LabVIEW ортасындағы көрінісі;

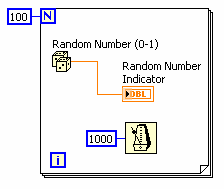
2-Блог-схемасы; 3-Мәтіндік аналогтік коды



For циклі функция палитрасының ішінде (Function >> Programming >> Structure) орналасқан. For цикліндегі N максималды итерация саны терминалының мәні ақпараттар ағынымен орындалатын операциялардың максималды санын анықтайды. Оның графикалық көрінісі сол жақ бетте көрсетілген.

Итерацияны есептегіш терминалы, сол жақ бетте көрсетілгендей орындалған итерация сандарының мәндерін көрсетеді. Итерация есептегішінің бастапқы мәні әрқашанда 0-ге тең.

For циклінің While циклінен айырмашылығы N белгіленген итерацияның максималды мәнін орындап барып, жұмысты тоқтатады. Ал While циклі циклдан шығу шартын орындап барып, жұмысты тоқтатады.

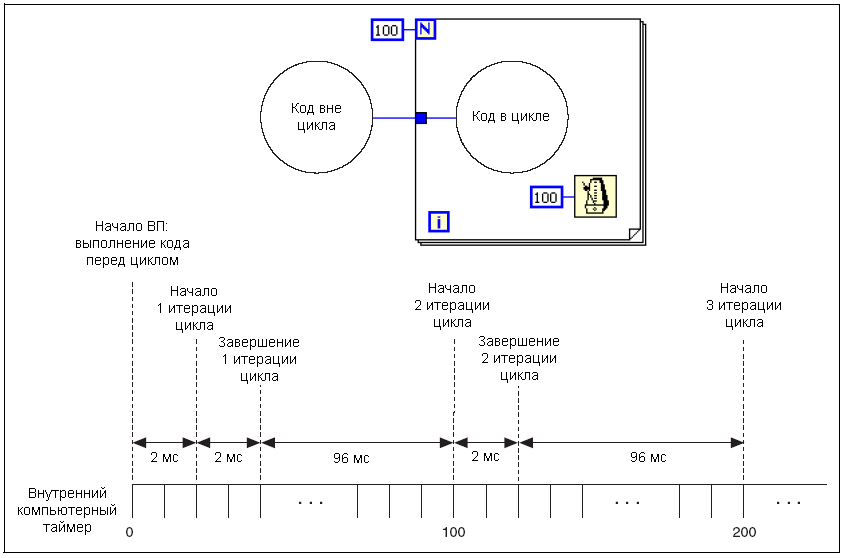
Төменде (48-сурет) көрсетілген For циклі әр секунд сайын 60 рет кездейсоқ сандарды генерациялайды және оларды мәліметтерді бейнелеу элементіне шығарады.

48-сурет. For циклі негзіндегі блог-диаграмма көрінісі

**4.2.1. Күту функциялары**



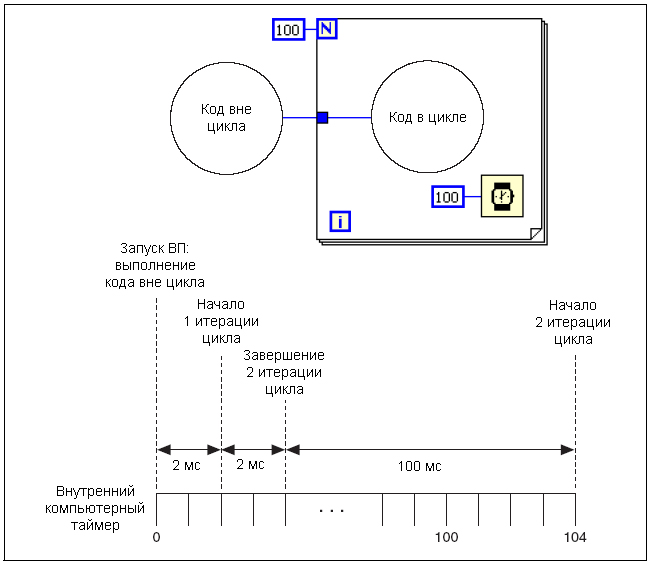
**Wait Until Next ms Multiple** функциясы көрсетілген қысқа мәнге жету үшін компьютердің ішкі таймеріне қажетті уақыт интервалына тең итерация аралығындағы интервалды қамтамасыз етеді. Төменде көрсетілгендей (49-сурет) циклдің бірінші периоды қысқа болатыны жөнінде ықтималдылық бар.



49-сурет. Күту функциясы қызметін көрсететін блог-диаграмма

Wait (ms) функциясы (сол жақ бұрышта көрсетілген) күту уақытын программаның орындалу уақытына қосып отырады. Егер программаның орындалу уақыты айнымалы болса, онда күту уақытын программаның уақытына қосу қиындық тудырады.

Күту функциясы Function >> Programming >> Timing ішінде, яғни функция палитрасында орналасқан (50-сурет).



50-сурет. Waitфункциясының қолданылу көрінісі

**4.2.2. Мәліметтер типтерін түрлендіру**

LabVIEW әр түрлі амалдарды мынадай мәліметтер типтерімен, мысалы: нақты сандық тип**: byte, word, long**, жылжымалы үтірлі сан: **single, double, extended precision,** комплекстік сандар: **single, double, extended precision** орындайды.Функция мәндерінің кіріс жолына әр түрлі типтегі операндтар келіп түскенде, функция шығысындағы мән едәуір кең диапазон форматындағы мәліметті қабылдайды. Бұл ретте LabVIEW типтерді өзгертуді автоматты түрде жүзеге асырады.Ал өткізгіштерді терминалмен байланыстыру орнында сұры түсті нүткенің бейнесі пайда болады. Мысалы, For циклінің итерацияның максималды сан терминалы N екілік дәлдіктегі нақты сандық типті (long integer) қабылдайды. Оған жылжымалы үтірлі, екілік дәлдікті форматтағы мәліметтер келіп түседі. Итерация саны терминалында сұр түсті нүкте пайда болады. Егер тек бір ғана типте жұмыс жасайтын функцияның мәліметтер кірісі жолына екі түрлі типтегі мәлімет келіп түскенде LabVIEWде бірінші терминал өзінің мәлімет типін екінші терминалға қатысты өзгертеді. LabVIEW көп биттер санына ие мәліметтер типтерін таңдайды. Егер типтер орын алған биттер санына эквивалентті болса, онда LabVIEW белгісіз мәлімет типін таңдайды.

Блог-диаграмма объектілерінің мәлімет типтерін өзгерту үшін сол объектіге тышқанды апарып, оң жақ батырмасын басу арқылы Representation пуктін таңдау керек.

**4.3.Циклдағы алдыңғы итерация мәндеріне қолжетімділікті ұйымдастыру**

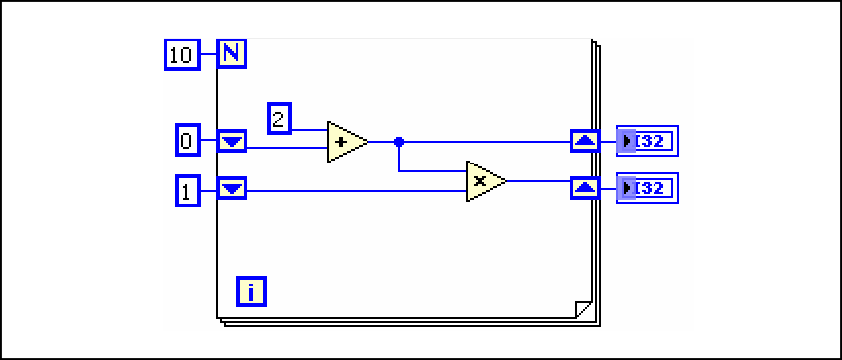
Циклдармен жұмыс жасау кезінде циклдің алдыңғы итерация мәндеріне қолжетімділікті қаматамасыз етуге болады. Мысалы, температураны өлшейтін ВҚ жағдайында өлшеу нәтижелерін графикке шығару кезінде алдыңғы итерацияда алынған мәндерді білу маңызды. Бұл мәліметтерге қолжетімділікті екі жол арқылы жүзеге асыруға болады: бірінші ығыстырушы регистр (Shift Register), екінші кері байланыс түйіні (Feedback Node) арқылы.

**4.3.1. Ығыстырушы регистрлер**

Ығыстырушы регистрлер циклдардаағымдағы цикл итерациясының мәндерін келесі итерацияға жіберу үшін қолданылады. Ығыстырушы регистрлер мәтіндік программалау тіліндегі статистикалық айнымалыларға ұқсайды. Ығыстырушы регистр жұптық терминалдар секілді көрініске ие. Оның графикалық көрінісі сол жақ бұрышта көрсетілген. Ығыстырушы регистр жоғары және төмен бағытталған екі нұсқадан тұрады. Оң жақтағы терминал нұсқасы жоғарыға бағытталған және ол ағымдағы итерация біткеннен кейінгі мәліметтерді сақтайды. LabVIEW осы регистрден мәліметтерді келесі цикл итерациясына жібереді. Ығыстырушы регистрді блог-диаграммада цикл ішіне тышқанды басу арқылы контексті мәзірден Add Shift Register пуктін таңдау арқылы орналастырасыз.

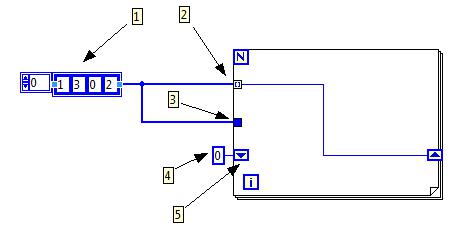
Ығыстырушы регистр кез келген типтегі мәліметтерді таратады, өзіне бірінші кеп түскен мәлімет типін автоматты түрде қабыл алады. Ығыстырушы регистрдің типі және өлшемі ығыстырушы регистрдің оң жақ (кіріс) терминалына келіп түскен мәліметтердің типі мен өлшемі арқылы анықталынады. Ығыстырушы регистрдің сол жақ терминалының разрядтар санын (циклдер кідірісін) орын алмастыру инструментін қолданып немесе контексті мәзірден элементті енгізу (Add Element) және элементті жою (Remove Element)опцияларын таңдау арқылы өзгертуге болады. Ығыстырушы регистр арқылы таратылатын мәлімет типтері бірдей болу керек. Ығыстырушы регистрді жүктеу үшін оның сол жақ терминалына цикл сыртынан қандай да бір мәндерді жіберу қажет. Егер ығыстырушы регистрді жүктегіңіз келмесе, онда цикл орындалған кездегі регистрге жазылған соңғы мәндерді қолданады. Жүктелмеген ығыстырушы регистрін қолданатын цикл іске қосылған ВҚдан алынған мәнді соңғы ығыстырушы регистрдің шығыс мәніне меншіктеу үшін қолданылады. Бір виртуалды құрылғыдан кейінгі екінші виртуалды құрылғы арасындағы күйлер жайындағы ақпаратты сақтау үшін терминалдың сол жақ кірісін анықталмаған ету керек. Цикл орындалып біткен соң регистрге жазылған соңғы мән оң жақ терминалға жазылады. Циклдан мәліметтерді оң жақ терминал арқылы келесі бөлікке жіберу кезінде регистрге жазылған соңғы мән жіберіледі.

Виртуалды құрылғыны бірінші рет іске қосқан кезде ығыстырушы регистрдің сол жақ терминалының мәні регистрдің мәлімет типі үшін әдепкі қалпы бойынша тең мәнге ие.Егер де бұл терминалдар қосылмаса, онда келесі іске қосу кезінде алдыңғы виртуалды құрылғыдағы мәндер келіп түседі. Ығыстырушы регистрдің сол жақ терминалын инициализациялау үшін тұрақтыны, басқару элементін немесе массивтер инициализациясы функцияларын қосу қажет. Инициализацияланбаған ығыстырушы регистрі бар циклдер ығыстырушы регистрдің шығысына соңғы орындалған виртуалды құрылғының мәндерін беру үшін қолданылады.

Бір цикл құрылымында бірнеше ығыстырушы регистрлерді қолдануға мүмкіндік бар. Егер бір циклда бірнеше операция орындалса, онда мәліметтерді сақтау үшін бірнеше терминалы бар ығыстырушы регистрлерді қолдану қажет болады. Төменде екі ығыстырушы регистрдің қолданылу схемасы көрсетілген (51-сурет):

51-сурет. Екі ығыстырушы регистрді қолдану сүлбесі

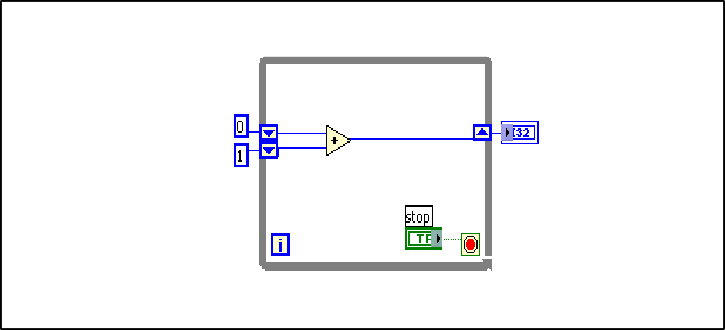
Ығыстырушы регистрдің жұмыс жасау принципін түсіну мақсатында мына мысалды қарастырайық. Төменде 52-суретте барлық қарастырылған терминалдар циклдары көрсетілген. Бұл мысалда Foor Loop құрылымына (1) массив жалғанған. Массивтің құрылым қабырғасымен байланысқан жерінде автоиндекстелетін терминал (2) пайда болады. Сонымен қоса құрылымға индекстелмейтін кіріс (3), ығыстырушы регистр (5) кіреді. Ығыстырушы регистрдің инициализацияланатын кірісіне инициализацияланатын мән (4) енгізіледі. Мәннің шамасы 0-ге тең. Әрбір итерация сайын ығыстырушы регистрдің кірісіне (1,3,0,2) массивтер мәндері беріледі. Автоиндекстелетін кіріс бар кезінде құрылымға итерацияның максималды мәнін (N) берудің қажеті жоқ. Себебі массив өлшемі арқылы автоматты түрде анықталып отырады.



52-сурет. Цикл құрылымының терминалдары

4.3.2. **Ығыстырушы регистрлер стегі (жиыны)**

Ығыстырушы регистрлер стегін құру үшін ығыстырушырегистрдің сол жақ терминалына тышқан батырмасын апарып,сол жақ батырмасын басып контексті мәзірден Add Elementпунктін таңдау қажет. Ығыстырушы регистрлер стегі циклдіңалдыңғы итерация мәндеріне қолжетімділікті қаматамасыз етеді.Ығыстырушы регистрлер стегі алдыңғы итерацияның мәндерінсақтайды және сол мәндерді келесі итерацияға жібереді. Ығыстырушы регистрлер стегі циклдің тек сол жақ бөлігінде ғана бола алады. Себебі оң жақ терминал мәліметтерді сол мезеттегі итерациядан келесі итерацияға жіберіп қана отырады (сурет 53).



53-сурет. Ығыстырушы регистрлерлер жиыны

Сол жақ терминалға тағы да екі ығыстырушы регистрді қосқанда

соңғы үш итерацияның мәліметтері келесі итерацияға жіберіледі.Сонымен қоса соңғы итерация мәні ығыстырушы регистрдің ең жоғарысына келіп орнығады. Екінші терминал алдыңғы итерацияда берілген мәнді сақтайды. Ал соңғы (төменде орналасқан) терминал екі итерация кейін алынған мәнді сақтайды.

**4.3.3. Кері байланыс түйіні**



Сол жақ бетте көрсетілгендей кері байланыс түйіні

While және For циклдерінде ВҚның ішкі программаларындағы мәліметтердің шығысын байланыстыру кезінде автоматты түрде пайда болады. Ығыстырушы регистр секілді кері байланыс түйіні сол мезеттегі итерация аяқталған уақыттағы кез келген типтегі мәліметтерді сақтайды және бұл мәндерді келесі итерацияға жібереді. Кері байланыс түйінін қолдану мәліметтер өткізгішінің, байланысның санын азайтады.

Кері байланыс түйінін While және For циклдерінің ішіне орналастырады. Орналастыру үшін Structure (структура) палитрасына кіріп Feedback Node-ты таңдайсыз.

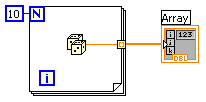
**Өтілген тақырыптарға қысқаша материалдар, кеңестер және құпиялар**

* Блог-диаграммада құрылымдар мәліметтер ағынымен қайталанатын операциялар орындау үшін қолданылады.
* While циклі шығу шарты терминалына циклдан шығу логикалық мәні берілмегенше орындала береді. Әдепкі қалпы бойынша бұл True логикалық мәні болып табылады.
* For циклі нақты бір санға тең рет орындалады.
* Циклды құру үшін цикл ішінде орналасатын элементтерді белгіліп алып тышқанның батырмасын жіберсеңіз жеткілікті.
* Wait Until Next ms Multiple функциясы цикл итерациясының орындалуына қажетті берілген уақыт интервалын қамтамасыз етеді.
* Wait функциясы нақты бір уақытты күтеді.
* Мәліметтер өткізгіштерін мәліметтер терминалдарымен байланыстырған сәтте сұр түсті нүктелер пайда болады.
* Мәліметтерді циклдің бір итерациясынан келесі итерациясына жіберу үшін ығыстырушы регистрлер қолданылады.
* Ығыстырушы регистрді құру үшін циклдің шетіне тышқанды апарып, оң жақ батырмасын басып контексті мәзірден Add Shift Register пунктін таңдайсыз.
* Циклдің алдыңғы бірнеше итерация мәндерін келесі ығыстырушы регистрге керек итерацияға жіберу үшін қосымша терминалдар енгізу керек. Оны енгізу үшін циклға меңзерді апарып, тышқан батырмасының оң жақ батырмасын басып, контексті мәзірден Add Element пунктін таңдайсыз.
* Кері байланыс түйіні сол моменттегі итерацияның кез келген типтегі мәліметін сақтайды және ол мәнді келесі итерацияға жібереді.
* Кері байланыс түйіні байланыс санын азайту үшін қолданылады.

**4-лабораториялық жұмыс. Автоиндексация. Ығыстыру регистрі. Кері байланыс түйіні. Логистикалық бейнелеу. Ығыстыру регистрі стекі. Бейсызық теңдеулерді шешу**

*1. Автоиндексация*

For циклі кірісінде массивті элементтерге жіктейді және шығысында элементтерден массивті құрастырады (біріктіреді). Бұл автоиндексация деп аталады. 54-суретте көрсетілген блог-диаграмманы құрастырыңыз.



54-сурет. Блог-диаграмма көрінісі

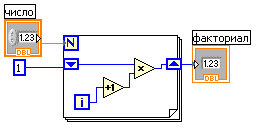
1. Беттік панельде сандық индикаторлардан тұратын массив құрыңыз. Ол үшін беттік панельден басқару палитрасы ішінен мына амалдарды орындаңыз: **Controls=> Array& Clusters=>Array,** сосын массив шаблонына цифрлық индикаторларды орналастырыңыз **Controls=>Numeric=>Numeric Indicator.**
2. Блог-диаграмма терезесінде For структурасын құрыңыз. Ол үшін мына амалдарды орындаңыз: **Functions=>Structures=>For Loop**.
3. **Functions=>Numeric=>Numeric Constant амалдарын орындап,** константа (тұрақтыны) құрыңыз. Итерация санын 10-ға тең деп алыңыз.
4. Кездейсоқ сандар генераторын қойыңыз (КСГ). Бұл For структурасының ішінде орналасады (**Functions=>Numeric=>Random Number (0-1)**). Кездейсоқ сандар генераторынан сымды алып шығып құрылымның оң жағына дейін апарыңыз. Сонда LoopTunnel пайда болуы қажет.
5. Loop Tunnelден массивке дейін сымды қосыңыз.
6. Run батырмасы көмегімен виртуалды құрылғыны іске қосыңыз.

Әрбір итерация массивке жаңа элементті қосып отырады. Циклдің жұмысы тоқтаған кезде шығысында N элементтен тұратын массив аламыз. Шынымен де шығыста массивтің шығатынын сымның қалыңдығы көрсетеді.

*2.* ***Ығыстырушы регистр***

Санның факториалын есептеу үшін ығыстырушы регистрді қолдану.

1. 55-суретте көрсетілген блог диаграммаға қарап беттік панельді құрыңыз.
2. Блог диаграммаға For структурасын құрыңыз. Структура ішіне инкремент функциясын қойыңыз **Functions=>Numeric=>Increment**, содан соң көбейту функциясын орналастырыңыз.
3. Ығыстырушы регистрді орналастырыңыз: **RMB=>Add Shift Register**.
4. Сандық тұрақты (константа) көмегімен ығыстырушы регистрдің бастапқы мәнін 1-ге тең деп беріңіз.
5. Барлық сымдарды 55-суретте көрсетілгендей етіп байланыстырыңыз.



55-сурет. Блог-диаграмма көрінісі

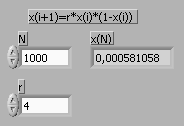
1. Виртуалды құрылғыны іске қосыңыз.

***3.Кері байланыс түйіні***

Алдыңғы мысал үшін ығыстырушы регистрді кері байланыс түйініне (Feedback Node) өзгертіңіз. Ол үшін ығыстырушы регистрдің контексті мәзірінен**Replace with Feedback Node**ты таңдаңыз. Сымдардың қалай байланысқанына назар аударыңыз.

***4. Логистикалық бейнелеу.***

Логистикалық бейнелеу формуласы: . Ығыстырушы регистрді қолдана отырып логистикалық бейнелеуге N-дік итерация мәнін беретін виртуалды құрылғы құрыңыз. Беттік панельдің көрінісі 56-суретте көрсетілген.

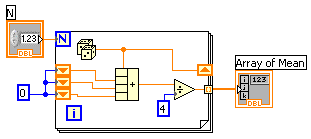


56-сурет. Беттік панель көрінісі

Бірнеше сандарды көбейту үшін **Functions=>Numeric=>Compound Arithmetics** функциясын қолданыңыз. Бұл элементтің көрінісі: CompoundArithMult . Контексті мәзір ішінен көбейту режимін таңдаңыз (**Change mode=>Multiply**).

***5. Ығыстырушы регистрлер стекі***

57-суретте көрсетілген блог-диаграманы жинаңыз. Ығыстырушы регистрлердің стекін құру үшін ығыстырушы регистрдің контексті мәзірінен Add Element пунктін таңдаңыз.

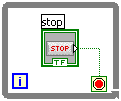


57-сурет. Блог-диаграмма көрінісі

Ретке келтіру режимінде виртуалды құрылғыны іске қосыңыз. Ол үшін **Highlight Execution** батырмасынбасыңыз және құрылғыны **Step Into, Step Over, Step Out** батырмаларымен қосыңыз. Мәндерді бақылаңыз. Бұл виртуалды құрылғының қызметін түсіндіріңіз?

*1. While циклі*

Шартты циклді қолданудың ең қарапайым мысалы функция палитрасының ішінде жиналған: **Functions => Execution Control => While Loop**. Осы циклді блог-диаграммаға орналастырыңыз. 58-суретте көрсетілгендей құрылым пайда болады.



58-сурет. While Loop циклін блог-диаграммаға орнату

**While** құрылымының өзінен бөлек цикл ішінде циклді тоқтату батырмасы орналасқан. Тап осы жағдайда ол батырма ішінде жиналған блог-диаграмманың жұмысына жауап береді. Іске қосылған программаны беттік панельдегі **STOP** батырмасын басу арқылы тоқтатуға болады. Жалпы барлық программаны **While** циклінің ішінде жинауға кеңес беріледі. Интрументалды панельде программаны үздіксіз іске қосу батырмасы **Run Continuously** және программаны дереу, тез өшіру **Abort Execution** батырмасы бар екені белгілі. Дәл осы мысал программаны үздіксіз іске қосу және тоқтатуға негізделген. Алайда осы екі әдіс арасында өзгешеліктер бар. **While** циклін және **STOP** батырмасын қолдану жағдайында цикл ішіндегі блог диаграмма барлық тиісті операцияларды орындайды. Орындалып жатқан программаны дереу ө тоқтату кезінде программаны орындау тез арада тоқтап қалады. Бұл өз кезегінде күтпеген қателіктерге алып келеді.

*2. Бейсызық теңдеулерді шешу*

Бейсызық дифференциалдық теңдеуді бисекция (дәл ортадан бөлу) әдісі арқылы шешетін виртуалды құрылғы құрыңыз.

Әдістің жалпы сипаттамасы: Жалпы бейсызық теңдеуді бисекция әдісімен шешу мынадай принциптерге негізделген. Қандай да бір  аралығын аламыз және осы аралықта *f(x)* = 0 теңдеудің тек бір ғана түбірі жатады. Мұндай жағдайда аралық шегіндегі функцияның мәні қарама-қарсы таңбаға ие болады. Егерде осы аралықтың ортасын тапсақ және осы нүктеде функция мәні қандай таңбаға ие болатынын анықтау керек. Бұдан теңдеудің түбірі қай бөлікте жатқанын анықтауға болады. Осылайша жаңа аралық пайда болады. Процедура пайда болған бөліктің ұзындығы оның алдындағы аралық ұзындығынан кіші болғанша жалғаса береді. Теңдеудің түбірі ретінде соңғы бөліктің орта мәні алынады.

1. Бастапқы деректер ретінде *f(x)* өрнегі, аралығы және е саны алынады. *f(x)* өрнегі(**String Control**) жолы арқылы беріледі, ал қалған барлық сандар сандық басқару элементтерімен беріледі. Бұлардың бәрін беттік панельге орналастыру керек.

2. a) *f(xmjn)*  және *f(xmax)* функцияларының мәндерін алу үшін**AllFunctions => Analyze => Mathematics => Formula => Advanced Formula Parsing => Eval Single-Variable Scalar** функциясын қолданыңыз.

EvalSingle-Var%20Scalar

59-сурет. Функцияның максимум және минимум мәнін беру көрінісі

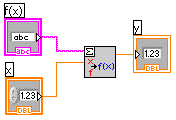
Бұл функцияның кірісіне формуласы жазылған жол (string formula ) және х-тің мәні (x value) беріледі. Ал шығысында х нүктесіндегі функция мәнін аламыз (y value) (59-сурет)*.*

b) Егер сізде **Advanced Formula Parsing** палитрасы болмаса, онда **Numeric => Expression Node** функциясын қолданыңыз.

Expression%20node

60-сурет. Expressin Node функциясы

Бұл жағдайда сізге блог- диаграммада өрнек түйінінің ішіне формулалық жолға формула жазу керек. **Single-Variable Scalar** және **Expression** Node функцияларын салыстырыңыздар (61-сурет).

EvalSingle-Var%20ScalarEx2Expression%20node%20Ex

61-сурет. Блог-диаграмма көрінісі

3.Блог-диаграммаға **While** циклін орналастырыңыз : **All Functions=>Structures=>While Loop**.

4. Бір итерациядан екінші итерацияға өткен кезде бөліктің шегі өзгеріп отыратын болғандықтан ығыстырушы регистрді қолдану қажет. Циклде бөлік шегі үшін екі ығыстырушы регистр құрыңыз. Цикл ішінде бөлік ұзындығын, бөліктің ортасын және бөлік ортасындағы функция мәнін анықтаңыз.

5. Циклдан шығу шартын анықтаңыз : бөлік ұзындығы алдыңғы берілген *е* санынан кіші болу керек. Бұл шартты өздеріңіз іске асыруға тырысыңыз.

6. Ағымдағы бөлікті таңдау процесін жүзеге асыру қажет. Егер *х*min нүктесіндегі және бөлік ортасындағы функция мәні бір уақытта нөлден үлкен немесе тең болса, онда теңдеудің түбірі басқа бөлікте жатқаны. Егер бұл нүктеде функциялар мәні әр түрлі болса, онда олар жаңа бір бөлікті тудырады.

Ендеше цикл алдында шектердің біреуіндегі функцияның мәндерін анықтау керек. Ол үшін функция палитрасының ішіндегі **Functions => Comparisonнен Greater Then 0?** операциясын қолданыңыз. Пайда болған логикалық айнымалыны цикл кірісіне беріңіз. Алынған логикалық айнымалыны циклдің кірісіне беріңіздер. Есептің шартына сәйкес қазіргі бөлікте түбір бар және соған қоса тек жалғыз, онда оң шек әрқашан теріс немесе оң болады. Әрі қарай, цикл ішінде сол сәттегі бөліктің ортасы үшін функцияның таңбасын анықтау керек. Түбірдің бөліктің оң немесе сол жағында орналасуына орай функция мәні өзгеріп отырады. Бұл логикалық айнымалыларды салыстыру міндетті. Егер де бұл екеуі де шындық және жалған болса, онда бөліктің оң жақ бөлігін таңдау керек. Ал егер де екеуінің біреуі шындық ал екіншісі жалған болса онда бөліктің сол жақ бөлігін таңдау керек. Жалған мәнін беретін бірінші жағдай үшін және шындық мәнін беретін екінші жағдай үшін мынадай логикалық операцияны қолданыңыз: **Exclusive Or.** Блог-диаграммада функция палитрасының ішінде орналасқан: **Functions => Boolean**.

7. **Functions => Comparison** палитрасының ішінде бізге қажетті мынадай функция орналасқан. Ол **Select** деп аталады. Екі **Select** қолданыңыз.Бұл жаңа бөліктің оң және сол шектерін анықтауға арналған. Екі жағдайда да*s-тің кірісіне* **Exclusive Or** шығысындағы мәндерді беру қажет.Егер **Exclusive Or** шығысындағы айнымалы шындық мәніне ие болса, онда оң шек сол кездегі мәнді сақтайды. Олай болса*t* кірісіне ығыстырушы регистрге сәйкес келетін оң шекті байланыстырыңыз.  *f* кірісін бөліктің ортасымен байланыстырыңыз. Екінші **Selectте** бөліктің ортасына*t* кірісін, ал *f-*ті сол жақ шекке байланыстырыңыз.**Select** функциясының шығысын ығыстырушы регистрлермен қосыңыз.

8. Бөліктің ортасын циклдің оң жақ шегіне апарамыз. Индикация элементі байланысатын туннель пайда болады.

9. Беттік панельге көшіңіздер. Қандай да бір функция енгізіңіз. Мысал үшін: қарапайым *f*(*х*) = *exp(x)-1-cos(x)* функциясын алсаңыз болады. Шектерді *0* мен  *1* деп беріңіз, сандық шешімнің дәлдігін беріңіз, мысалы0,0001. Программаны іске қосыңыз. Егер блог-диаграмма дұрыс құрылса, онда индикация элементінде мына сан шығу керек 0,601349.

10. Программаның орындалу қадамын бақылаңыз. Ығыстырушы регистрлер мен туннельдің функционалды айырмашылығына назар аударыңыздар.

Сұрақтар:

1. **WhileLoop**құрылымының жұмысын сипаттап беріңіз.
2. **Single-Variable Scalar**-дің **Expression Node-**тан айырмашылығы неде?
3. Циклдан шығудың шартын қалай беруге болады? Қандай функцияларды қолдануға болады?
4. Дәл осы есепте келесі шекті табу қалай іске асырылды? Әдісін және қолданған функцияларын айтып беріңіз?
5. Ығыстырушы регистр мен туннельдің айырмашылығы неде? Бұл лабораториялық жұмыста бұларды қалай қолдандыңыз?

4-тапсырма. Алдын-ала және кейінгі шарттарымен цикл операторлары

**Қосымша жаттығулар**

*1. ax 2* + *bx* + *c* = *0* квадраттық теңдеудің түбірін есептейтін виртуалды құрылғы құрыңыз. a,b,c коэффициенттерінің мәні беттік панельден беріледі. Олардың мүмкін болатын мәндерінің диапазонын +/-10 аралығымен шектеңіз. Егер квадраттық теңдеудің дискременанты нөлден үлкен болса, яғни түбір бар болатын болса, онда беттік панельде жарықтық диод жану қажет.

2.Stop батырмасын басқаннан кейін жұмысын тоқтататын және селектр күйінен тәуелді жағдайдағы екі санның қосындысын, айырымын, көбейтіндісін, бөліндісін есептейтін виртуалды құрылғы құрыңыз. Санды нөлге бөлуге болмайтындығы жайлы қолданушыға хабарлама беру жағдайын қарастырыңыз.

**Кеңес.** Нөлге бөлу кезіндегі хабарламаны шығару үшін **Programming >> Dialog&User Interface** палитрасында орналасқан **One Button Dialog** функциясын қолданыңыз.

3. 7 қадам ішінде , сізге “Сіз таңдап алған сан>50?” және т.б. типтегі сұрақтар қоятын 1-ден 100-ге дейінгі сіз жасырған санды анықтайтын виртуалды құрылғы құрыңыз. Сұрақтарға жауап беру кезінде міндетті түрде виртуалды құрылғы дұрыс немесе бұрыс дегенжауапты алу керек.

**Кеңес.** Сұрақ қою үшін **Programming >> Dialog&User Interface** палитрасында орналасқан **Two Button Dialog** функциясын қолданыңыз.

4. n! факториалды есептейтін виртуалды құрылғы құрыңыз. Оны подпрограмма ретінде рәсімдеңіз (пиктограмма және байланыстырушы панельді құрыңыз). Осы подпрограмма көмегімен мына өрнекті есептеңіз: .

5-бөлім

**Тізбекті және таңдау операторлары**

Бұл бөлімде ВҚларда шешімді қабылдау, сонымен қоса әр түрлі математикалық амалдардың орындалуын жеңілдететін құрылымдармен жұмыс жасау механизмдері айтылады. Негізінен мынадай құрылымдар қарастырылады: **Select** функциясы, **Case** құрылымы, **Sequence** құрылымы, **Event**құрылымы,формулалық түйін (**Formula Node**), математикалық түйін ( **MathScript Node** ). Сонымен қоса бұл бөлімде **Formula Express VI** виртуалды құрылғысының мүмкіндіктері де қарастырылған.

**5.1. Select функциясы және шешім қабылдау**

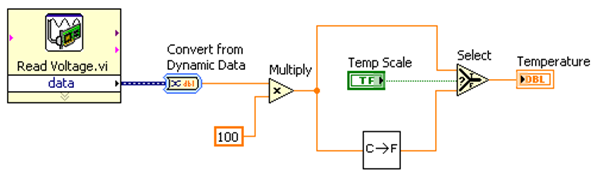
Осы бөлімге дейін әрбір виртуалды құрылғы мәліметтер ағыны арқылы анықталатын реттілік бойынша орындалады деп айтылған болатын. Алайда программа арасында шешім қабылдау қажет болатын жағдайлар да болады. Мысалы, егер де А оқиғасы болып жатса, онда В оқиғасын жасау міндетті, ал егер С оқиғасы болып жатса, онда D оқиғасын жасау міндетті.

Мәтіндік программалау тілінде жазылған программада бұл есеп мына операторлар көмегімен шешіледі: **if, else, case, switch** және т.б. LabVIEWде шешім қабылдаудың бірнеше әдістері ойластырылған. Оның ішіндегі ең қарапайымы ол **Select** функциясы.

**Select функциясы**

**Select**функциясы**Function>>Programming>> Comparison**  палитрасы ішінде орналасқан.Кірісіндегі логикалық екі мәннің біреуіне ие бола алады. Егер логикалық кірісіне TRUE мәні берілсе, онда функцияның шығысы Т кірісіне мәнді жібереді. Ал егер де логикалық кіріске False мәні берілсе, онда F жолындағы мән қайтып келеді.

Select функциясының қолдану мысалы ретінде төмендегі (62-сурет) блог-диаграммаға назар аударсақ болады. Термометр ВҚсын жасауға қолданған блог-диаграмма көрінісі температураның шкаласын анықтау мақсатында құрылған. Мұндағы Select құрылымының негізгі міндеті операторды таңдау.

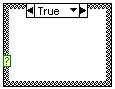


62-сурет. Select құрылымы қолданылған блог-диаграмма

Бұдан да күрделі шешімдерді қабылдау кезінде **Case**құрылымын қолданған ыңғайлы.

**5.2. Case құрылымы**

**Case** құрылымының графикалық бейнесі 63-суретте көрсетілген. Екі немесе одан да көп нұсқалар поддиаграммасына ие. Белгілі бір уақыт мезетінде тек бір ғана нұсқа поддиаграммасы көрініп тұрады. Сол құрылымның орындалуы кезінде тек бір ғана поддиаграмма орындалады. Case құрылымы мәтіндік программалау тіліндегі case операторына немесе if, then, else операторларына ұқсайды. Поддиаграмма дегеніміз диаграмма ішіндегі диаграмма.



63-сурет. Case құрылымының графикалық бейнесі

 Caseқұрылымының селекторы Case құрылымының графикалық бейнесінде жоғары бөлікте орналасқан. Case құрылымының селекторы қақ орта бөлігінде нұсқалар мәндерін көрсетушіден, екі жағынан айналдырғыш нұсқардан тұрады. Бұл нұсқар мүмкін болатын нұсқаларды көру үшін қолданылады.

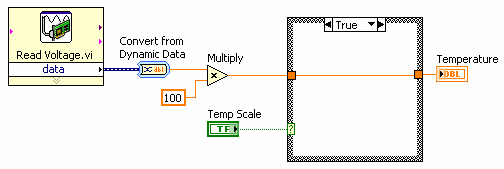
Нұсқалар терминалы құрылымның қандай поддиаграммасы немесе нұсқасы орындалатынын анықтап отырады. Бұл терминалға тек нақты, логикалық, жолдық типтегі мәндерді беру рұқсат етілген. Нұсқалар терминалы Case құрылымының сол жақ бөлігінің кез келген жерінде орналаса алады. Егер нұсқалар терминалы логикалық типті болса, онда құрылым екі логикалық нұсқалардан тұрады: TRUE және FALSE. Егер нұсқалар терминалы мынадай мәлімет типтеріне ие болса: нақтысандық, жолдық және т.б. онда нұсқалар саны -1 -ге дейінгі мәндерді ғана қабылдай алады.

Case құрылымын қолдану үшін нұсқаны әдепкі қалпы бойынша белгілеу қажет. Нұсқалар терминалының мәні диапазоннан асып кеткен жағдайда немесе нұсқалар терминалының мүмкін болатын мәндері үшін нұсқалар болмаса, онда әдепкі қалыптағы нұсқа немесе поддиаграмма орындалады.

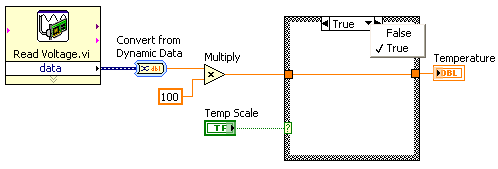
Case құрылымында тышқан батырмасының оң жағын басу нұсқаларды көшіруге, орнын ауыстыруға, нұсқаларды енгізуге немесе жоюға , сонымен қоса әдепкі қалпы бойынша нұсқаны белгілеуге мүмкіндік береді.

**Нұсқаны таңдау**

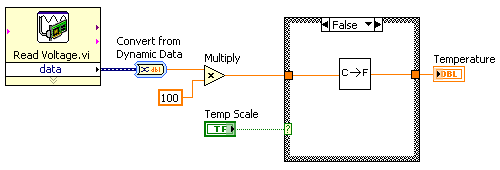
Мысал ретінде Select функциясының орнына Case құрылымың қолданылуы келтірілген (64-сурет). Case құрылымының алдыңғы бөлігінде TRUE логикалық нұсқасы көрсетілген (61-сурет). Нұсқаны анықтау Case құрылымы селекторындағы мәнді таңдаумен немесе Мәтінді Енгізу құралы көмегімен анықталынады (62-сурет).



64-сурет. True логикалық нұсқасы



65-сурет. Нұсқаны таңдау көрінісі

Қандай да бір нұсқаны таңдау кезінде келесі блог-диаграммада көрсетілгендей нұсқа анықталынады.

66-сурет. Нұсқасы таңдалынған блог-диаграмма көрінісі

Нұсқа селекторының мәні нұсқалар селекторы терминалына берілген мәлімет типімен сәйкес болу керек. Қызыл түспен боялған нұсқалар селекторының мәні оны тез арада өңдеу қажет немесе жою керек дегенді білдіреді. Олай жасамаса онда ВҚ іске қосылмайды. Бұл амалды жасау міндетті. Нұсқалар селекторы терминалына жылжымалы үтірлі санды беруге болмайды. Себебі санды жуықтау кезінде қателіктер болуы мүмкін. Егер де жылжымалы үтірлі санды нұсқалар селекторы терминалына берсек, онда LabVIEW ол санды ең жақын жұп санға жуықтайды.

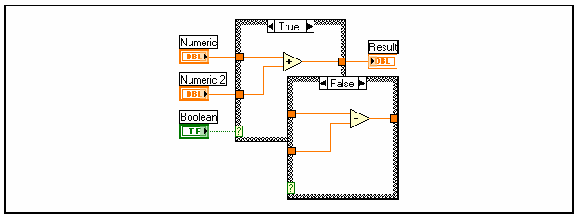
**Кіріс, шығыс терминалдары**

Case құрылымында кіріс және шығыс мәлімет терминалдары қолданылады. Кіріс мәлімет терминалдары барлық поддиаграмма үшін қолжетімді. Бірақ оның құрылымның поддиаграммасымен қолданылуы міндетті емес. Бір поддиаграммада шығыс мәлімет терминалын құру келесі поддиаграммада дәл сол шектерде пайда болуына алып келеді. Егер тым болмаса бір поддиаграммада шығыс терминал анықталмаса, онда бұл терминал жолы ақ түске боялады. Бұл құрылымның дұрыс құрылмағанын көрсетеді. Бүкіл нұсқалар үшін шығыс терминалдары мәндерін анықтау міндетті. Сонымен қоса шығыс терминалдар сәйкес келетін типтегі мәндерді иелену керек. Шығыс терминалдың мәнін анықтау үшін терминалға меңзерді апарып, тышқан батырмасының оң жағын басып контексті мәзірден**Create >> Constant**пунктін таңдау қажет.

Мысалдар.

Келесі мысал Case құрылымының кіріс терминалының мәні нұсқа терминалы мәнінен тәуелді қосыла ма әлде азайтылады ма соны көрсетеді.

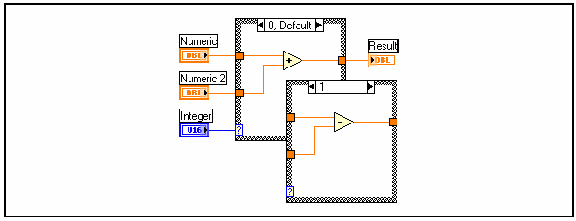
**Логикалық Case құрылымы**

Төменде логикалық Case құрылымының мысалы көрсетілген. Құрылым нұсқалары иллюстрацияны қысқарту мақсатында бір-біріне беттескен.

67-сурет. Логикалық Case құрылымының мысалы

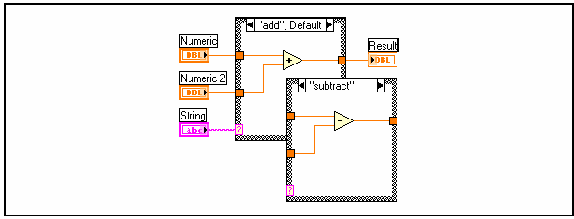
Нұсқалар селекторы терминалының мәлімет өткізгішімен байланысқан логикалық басқару элементі терминалына TRUE мәні берілсе, онда қосу амалы орындалады; ал егер FALSE мәні берілсе азайту амалы орындалады.

**Бүтін сандық Case құрылымы**

Төменде,68-суретте бүтін сандық Case құрылымы көрсетілген. Integer терминалы **ring control** (циклдік артық мәндер тізбегі ) басқару элементіне сәйкес келеді. Ол **Controls >> Modern >> Ring & Еnum** палитрасында орналасқан. Егер де **ring control** басқару элементі 0 санына тең болса, онда қосу амалын орындау керек, ал 1 санына тең болса, азайту амалы орындалады.

68-сурет. Бүтін сандық Case құрылымы

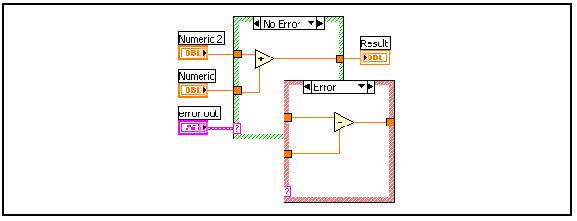
**Жолдық Case құрылымы**

Төменде жолдық Case құрылымының көрінісі көрсетілген (66-сурет):

69-сурет. Жолдық Case құрылымының көрінісі

Егер де басқару элементіне**add** жолы енгізілсе, онда ВҚ екі санның қосындысын орындайды. Ал **subtract** мәні енгізілсе, онда екі санның айырымын орындайды.

**Қателік кластері үшін Case құрылым**

Төменде суретте қателік кластері үшін Case құрылымының көрінісі бейнеленген:

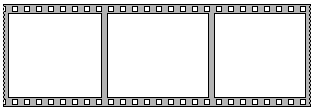
70-сурет. Қателік кластері үшін Case құрылымының көрінісі

Бұл мысалда, Case құрылымының селектор терминалына error out қателік кластері берілген (70-сурет). Бұл жағдайда құрылымның тек екі нұсқасы ғана бар: қате және қате жоқ. Бұл нұсқаға сәйкес құрылым шектері қызыл және жасыл түсті иеленеді. Case құрылымы нұсқаларды қате болу мүмкіндігі жайлы ақпараттың болуына сүйене отырып орындайды. Case құрылымы қателік кластерінің тек **status** логикалық айнымалысына ғана жауап береді.

**5.3.Sequence тізбектілік құрылымы**

Тізбектілік құрылымы қатар орындалатын бір немесе бірнеше диаграммаларды (кадрлар) құрайды. Тізбектілік құрылымының екі типі бар: ашық тізбекті құрылым және бірнеше қабатты тізбекті құрылым.

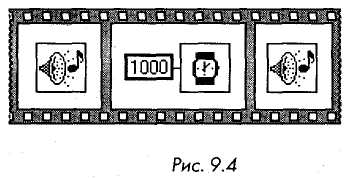
Тізбектілік құрылымын қолдану программаның орындалу реттілігін басқаруға мүмкіндік береді. Егер де сіз программаның кейбір бөлігі қандай да бір бөлігінен бұрын орындалсын десеңіз, онда тізбектілік құрылымы қолданылады. Тізбектілік құрылымының бейнесі 71-суретте көрсетілген.



71-сурет. Sequence құрылымының көрінісі

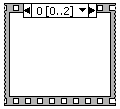
**Ашық тізбекті құрылым (Flat Sequence Structure)**

Ашық тізбекті құрылым солдан оңға қарай кадрдан келесі кадрға өту арқылы орындалады. Тізбектілік ретінен кадрды жоюға немесе қосуға мүмкіндік бар. Ол үшін контексті мәзірді қолданасыз. Кадрды қосқан немесе жойған кезде құрылым өзінің өлшемін автоматты түрде өзгертеді. Мысалы 72-суретте сигнал арасындағы кідіріс 1 секундқа тең екі дыбыстық сигналды беретін ашық тізбекті құрылым суреттелген. Replace контексті мәзір пунктін қолдана отырып құрылым типін бірнеше қабаттыға алмастыруға болады.



72-сурет. **Flat Sequence Structure** ашық тізбекті құрылым бейнесі

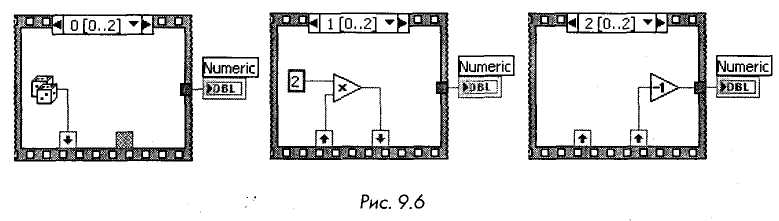
**Бірнеше қабатты тізбекті құрылым (Stacked Sequence Structure)**



73-сурет. Бірнеше қабатты тізбекті құрылым бейнесі

Бірнеше қабатты тізбекті құрылым реті бойынша орындалатын нумерацияланған поддиаграммалардан тұрады (73-сурет). Блог-диаграммада (ашық тізбекті құрылымға қарағанда) бір уақытта сіз тек бір поддиаграмманы көре аласыз. Бір поддиаграммадан келесі поддиаграммаға өту тізбекті құрылым селекторы көмегімен іске асады. Егер блог-диаграммада бос орындарды үнемдегіңіз келсе бірнеше қаббатты тізбекті құрылымды қолданыңыз.

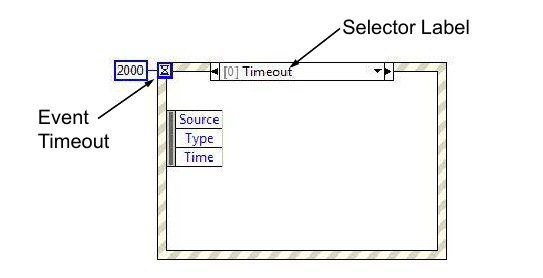
Бір поддиаграммадан келесі бір поддиаграммаға мәліметтерді беру локальді айнымалылар терминалдары арқылы іске асады. Терминалдарды құру үшін құрылым рамкасына меңзерді апарып, контексті мәзірден Add Sequence Local пунктін таңдау қажет. Мәлімет көздерін локальді айнымалылар терминалына байланыстырып болған соң онда сыртқа қарай бағытталған тіл (стрелка) пайда болады. Бұл тіл терминалдың мәліметті қабылдаушы екендігін көрсетеді (74-сурет). Ал келесі поддиаграммадағы терминал мәлімет көздері болып табылады. Ал ондағы тіл бағыты ішке қарай бағытталады. Бұдан бұрыңғы орындалып кеткен кадрлардағы мәліметтер көздері терминалы шрихталған формаға ауысады және оны сіз қолдана алмайсыз. Контесті мәзірдегі Replace пунктін қолдана отырып құрылымның типін өзгертуге болады. Мысалы бірнеше қабатты тізбекті құрылымды ашық тізбекті құрылымға алмастыруға болады. Тіпті құрылымның өзін қандай да бір басқа құрылымға алмастыруға мүмкіндік бар. Мысалы Sequence құрылымын Case құрылымына деген секілді.



71-сурет. Бірнеше қабатты тізбекті құрылымның әрбір поддиаграммасының (кадрларының) көріністері

**5.4.Оқиғалар құрылымы (Event Structure)**

Оқиға құрылымы блог-диаграмманың орындалуымен беттік панельдегі тұтынушының іс-әрекеттерінің синхронизациясы кезінде қолданылады. Бұл құрылымды қолдану тұтынушы әр жолы қандай да бір іс-әрекетті (батырманы басу, басқару элементінің мәнін өзгерту, тышқан батырмасын басу) орындаған кезде анық бір поддиаграмманы орындауға арналған. Оқиға құрылымын қолданбаған кезде беттік панельдегі объектілердің күйлерін тексеріп, қандай да бір өзгерістер болмағандығын бақылауға тура келеді. Беттік панельдегі объектілер кұйлерін сұрау өте көп уақытты алады және тіпті өзгеріс болғанның өзінде оның болу уақытының тез болғаны соншалық өзгеріс жайлы ақпарат ала алмауыңыз мүммкін.Оқиғалар құрылымының LabVieW ортасындағы графикалық көрінісі 72-суретте көрсетілген.



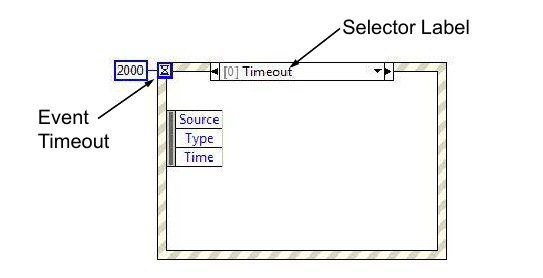
72-сурет. Оқиғалар құрылымының графикалық көрінісі

Оқиғалар құрылымының элементтері:

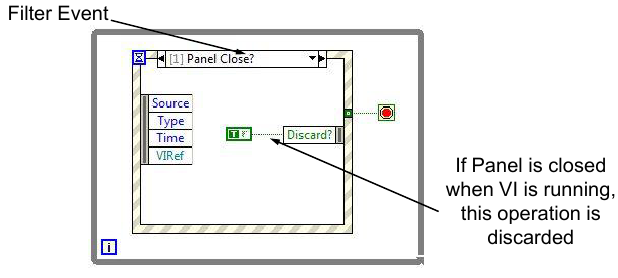
1.Оқиғалар құрылымының селекторы (Selector Label) –поддиаграмма арасындағы өткел қызметін атқарады.

2.Күту уақытының терминалы (Event Timeout)- Event құрылымының оқиғаның аяқталу уақытын анықтайтын уақытын көрсетеді.

3.Оқиғалар мәліметтерінің түйіні – оқиға болған кезде поддиаграммаға мәліметтер осы түйін арқылы жіберіледі. Оның графикалық көрінісі:



73-сурет. Оқиғалар мәліметтерінің түйіні

4.Оқиғаны фильтрлеу түйіндері –атауында сұрақ белгісі бар оқиғалар үшін қолданылады. Графикалық бейнесі:  .

5. Динамикалық оқиға түйіні –әдепкі қалып бойынша құрылым бетіне шығып тұрмайды. Бейнесін шығару үшін контексті мәзірден Show Dynamic Event Terminals пунктін басу керек.

Оқиғалар құрылымы нұсқалар құрылымы сияқты жұмыс жасайды. Қандай оқиға болып жатқанына байланысты блог-диаграмманың бірінші немесе екінші нұсқасы орындалады. Әрбір нұсқа үшін бір немесе бірнеше оқиғаларды орнатуға болады. Виртуалды құрылғыны іске қосу үшін LabVieW ортасы көрсетілген оқиғаның орындалғанын күтеді. Содан соң сәйкес поддиаграмманың орындалуы іске асады.

Оқиғалар түрлері:

**Application Exit, Application Exit?**-LabVieWдан шығу.

**Timeout**-Күту уақытын арттыру.

**Key Down,Key Down?-**Беттік панельдің кез келген жерінен кез келген батырманы басу.

**Key Repeat,Key Repeat?**- Беттік панельдің кез келген жерінен кез келген батырманы басу және ұстап тұру.

**Key Up**-Пернетақтадағы батырманы бос жіберу.

**Menu Activation?**-мәзірді батырмалар көмегімен шақыру. Мысалы:Ctrl+C деген сияқты.

**Menu Selection (App),Menu Selection ? (App)**-контексті мәзір ішінен қандай да бір пункті таңдау.

**Menu Selection (User)**-Тұтынушылық мәзірден пункті таңдау.

**Mouse Down,Mouse Down?**-Тышқан батырмасын басу.

**Mouse Enter**-Беттік панель ішінде меңзер орнын ауыстыру.

**Mouse Leave**-Беттік панельден тыс аумақта меңзердің орнын алмастыру.

**Mouse Up**-Тышқан батырмасын бос жіберу немесе босату.

**Panеl Close,Panel Close?**-Виртуалды құрылғының беттік панелін жабу.

**Panel Resize**-Беттік панельдің өлшемін өзгерту.

**Өтілген материалдар бойынша қысқаша түсіндірме, кеңестер мен құпиялар**

* Select функциясы үшінші логикалық кірістегі мәнге тәуелді екі кіріс мәннің біреуін қайтарады.
* Casе құрылымы екі не одан да көп нұсқаларды иеленуі мүмкін (поддиаграмма). Блог-диаграммада тек бір ғана поддиаграмма көрініп тұрады және 1 ғана поддиаграмма жұмыс жасай алады.
* Егер нұсқалар терминалы логикалық типті болса, онда құрылым екі логикалық нұсқадан тұрады. Олар: True және False . Егер нұсқалар терминалы нақты сандық, жолдық типке ие болса, онда нұсқалар саны 231-1 мәнді иелене алады.
* Егер де подпрограмманың орындалуынан құтылу керек болса, онда кіріс қателік кластерін селектор терминалына байланыстыру керек.
* Арифметикалық өрнектеулер көп болған кезде немесе өте күрделі болған жағдайда Формула түйінін қолдану ыңғайлы. Формула түйінінің жұмыс ортасында жазылған әрбір өрнек соңынан нүктелі үтір (;) белгісі қойылады.
* Тізбекті және таңдау құрылымдары программаның орындалу жолдарын басқаруға ммүкіндік береді.таңдау құрылымы әр түрлі айнымалылар мәндерінде (логикалық, сандық, жолдық және т.б.) әр түрлі программа кодтарын орындау қажеттілігі кезінде қолданылады. Тізбекті құрылым программа аумағын анықталған реттілікте орындау үшін қолданылады.
* Ашық тізбекті құрылым бірнеше қабатты тізбекті құрылымға қарағанда ыңғайлы. Себебі, бір уақытта блог-диаграмманың барлық бөлігі көрініп тұрады.

**Қосымша жаттығулар**

1. Келесідей теңдеулерді есептеуге арналған виртуалды құрылғы құрыңыз: . Екі теңдеуді де формула түйініне енгізіңіз.
2. Калькулятор секілді қызмет ететін виртуалды құрылғы құрыңыз. Сандық мәндерді енгізу үшін сандық басқару элементерін, ал арифмеикалық операцияның нәтижесін шығару үшін сандық бейнелеу элементін қолданыңыз. Арифметикалық операцияны таңдау үшін slide control басқару элементін қолдану ұсынылады.
3. 8.1-жаттығуда (квадрат түбірді шығару) виртуалды құрылғы формула түйіні көмегімен барлық арифметикалық және логикалық амалдарды қолданатындай етіп өзгертіңіз.
4. Екі басқару элементі (атаулары Шек және Кіріс массив ) және бір бейнелеу элементі (Шығыс массив ) бар виртуалды құрылғы құрыңыз. Шығыс массив атауына ие шығыс индикатор шек атауына ие элемент мәндерінен асып кететін кіріс массив элементтерінен тұруы керек. Виртуалды құрылғыға шектен асып кету атауын беріңіз. Келесі басқа бір виртуалды құрылғы құрыңыз. Мұнда вқ кездейсоқ сандар (0-ден 1-ге дейінгі сандар) массивін генерациялайды. 0,5 мәніне ие шектен асп кететін элементтер мәндерінің шығыс массивтерін құру үшін подпрограмма негізінде шектен асып кету атауына ие вқны қолданыңыз.
5. Мәліметтердің екі массивін генерациялайтын виртуалды құрылғы құрыңыз. Оның біреуі кездейсоқ сандар, екіншісі гармоникалқ сигнал. Екеуінің суммасын графикте бейнелеңіз. Екінші графикке қосынды сигнал қуатының спектральді тығыздығын бейнелеңіз және негізгі жиілікті есептеңіз.

Кеңес. Қуаттың спектарльдық тығыздығын есептеу үшін Function>>Express палитрасында орналасқан Spectral Measurements экспресс виртуалды құрылғысын қолданыңыз.

**5-Лабораториялық жұмыс. Цифрлық есептеу құрылғыларын жасау**

*5.1. Case құрылымымен жұмыс . Пароль енгізу*

Сіздің программадағы қандай да функцияға қолжетімділікті үлестірудегі *Case* құрылымын қолдану мысалын қарастырамыз*.*

1. Беттік панельге жолдық басқару элементін орналастырыңыз: **String (String & Path => String Control).** Оның белгісін«Пароль енгізіңіз» деп өзгертіңіз.

2. Блог-диаграммаға **Case** құрылымын орналастырыңыз **(Structures *=>*Case Structure).**

3. **String** басқару терминалын **Case** құрылымының нұсқалар селекторымен байланыстырыңыз.

4. Әдепкі қалпы бойынша құрылымда екі нұсқа болады: **«True»**  және **«False», Default.** Осы екі нұсқаның береуін таңдауға сәйкес беттік панельдегі жолдық басқару элементіне жазған парольміз блог-диаграммадағы бірінші нұсқа орындалғанда ғана, **True**-дің орнына сіз дұрыс деп тапқан нұсқаны берген кезде орындалады. Ал екінші нұсқа басқа жағдайлар үшін оырндалады. Сондықтан блог-диаграммада **True** орнына 123 деген сияқты сіз дұрыс деп тапқан пароліңізді жазыңыз.

5. Екі блог-диаграмма үшін бір батырмадан тұратын диалогтік терезені орналастырыңыз **(Time & Dialog => One Button Dialog).**

6. Екі диалогтік терезе үшін де **message** кірісіне жолдық константаны жалғаңыз. Ол үшін диалогтік терезенің контексті мәзірінен сіз беретін кіріс жанынан **Create => Constant-**ты таңдап алыңыз. **123** паролі жазылған блог-диаграмма үшін **«Пароль дұрыс»** хабарламасын енгізіңіз. Ал екінші блог-диаграмаға **«пароль дұрыс емес»** деп енгізіңіз.

Егер беттік панельдегі жолдық басқару элементіне 123 паролін берсеңіз, онда блог-диаграммадан **«Пароль дұрыс»** деген хабарламаны аласыз**. Ал 123-**тен басқа парольді беттік панельдегі басқару элементіне берсеңіз, блог-диаграммадан **«пароль дұрыс емес»** деген хабарламаны аласыз.

*5.2. Калькулятор*

Калькулятор қызметін атқаратын виртуалды құрылғы құрыңыз. Кіріс мәндерін екі басқару элементі арқылы беріңіз. Арифметикалық операцияны таңдағанда байланысты шығысында бұл екі сан қосылады, азайтылады, көбейтіледі және бөлінеді.

1. Беттік панельде а және b екі басқару элементін құрыңыз.

2. Нөмірленген басқару элементін құрыңыз: **Controls => Ring&Enum =>Enum.** Оның қасиеттеріне кіріп **(**оның қасиеттеріне кіру үшін контексті мәзірден **Properties** пунктін таңдаңыз)**Edit Items** пунктерді редакциялауға өтіңіз.Кестедегі **Label** белгісіне 4 жолды енгізіңіз:«a+b», «a-b», «a\*b», «а/b». Әрбір белгіге 0-ден 3-ке дейінгі сандар сәйкес келеді. Бұл сіз қолданып отырған арифметикалық амалдың жалпы санын көрсетеді.

3. Блог-диаграммаға өтіп, онда **Case** таңдау құрылымын орналастырыңыз.

4. Басқару элементін **Enumды** беттік панельге орналастырған кезде блог-диаграммадан сәйкесінше терминалын көресіз. Басқару элементін нұсқалар селекторына жалғаймыз. Сол кезде ол өз түсін көк түске өзгертеді. Ал құрылыс селекторында **«True»** және **«False»** нұсқаларының орнына«а+b» және «а-b»-ны орнатамыз. Тағы да басқа арифметикалық амалдарды енгізу үшін **Case** таңдау құрылымының контексті мәзірінен**Add Case For Every Value** пунктін таңдаймыз.

LabVIEWде **Enum** басқару элементі блог-диаграммадағы **Case** нұсқаны таңдау құрылымы үшін әрбір пункт бойынша нұсқаларды беріп отырады. Сондықтан да құрылым селекторындағы нұсқалар **Enum** басқару элементінен алынып отырады.

5. Программаны іске асу үшін ең шешуші рөлді 0,1,2 немесе 3 нұсқалар нөмері атқарады. Сондықтан да **Enum** басқару элементінің терминалы, мәліметтер сымы мен нұсқалар селекторы терминалы көк түсті болады. Алайда қолдануға оңай болу үшін нұсқалар атымен маркіленеді. Құрылымдағы әрбір поддиаграмма үшін сәйкес келетін блог-диаграмманы алыңыз және нәтижені индикация элементіне шығарыңыз. Бұл жерде айта кету керек құрылымдағы барлық поддиаграммаларда нұсқалар бір құрылым ішінде орналасады.

6.Программаны іске қосыңыз. Басқару элементіндегі барлық нұсқаларды тексеріңіз.

Сұрақтар:

1. **Case**құрылымының жұмысын айтып беріңіз.
2. **Enum**басқару элементінің жұмысын айтыңыз.
3. **Enum**басқару элементі үшін**Case**құрылымына нұсқаларды қалай қосуға болады?

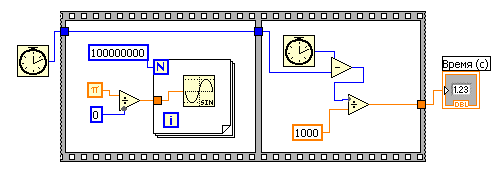
*5.3. Тізбекті құрылым (SequenceStructute). Программаның орындалу уақыты*

Қандай да бір күрделі емес операцияның, мысалы синус -ны есептеуге, орындалуына кеткен уақытты есептейтін виртуалды құрылғы құрыңыз. Ол үшін сол моменттегі уақытты алуға көмектеметін миллисекундты көрсететін **Functions => Time & Dialog => Tick Count (ms)** функциясын екі рет қолданыңыз. Біреуін синустың есептелуіне дейінгі және екіншісін есептелуінен кейін қолданыңыз.

 тұрақтысы үшін **All Functions=> Numeric => Additional Numeric Constants => Pi** қолданыңыз. Барлық тригонометриялық функциялар **All Functions=> Numeric =>Trigonometric** ішінде орналасқан.

Заманауи машиналарда синусты есептеу уақыты соншалықты аз, сондықтан бір реттік есептеу жүргізу бұл операцияның орындалуы үшін орындалуы мүмкін болмайды. Кейбір аргументтің синусын есептеуді мақсатты түрде бірнеше рет орындаған дұрыс, мысалға 108 рет. Мұндай әдіс арқылы уақытты есептеу қандай да бір күрделі емес операцияның орындалу уақытын есептеп береді.

Мұндай виртуалды құрылғыны өздеріңіз жасап көріңіздер. 74-суретте көрсетілген блог-диаграмманы жинаңыз. Виртуалды құрылғыны іске қосыңыздар. Сіздің виртуалды құрылғыңыз қандай нәтижелерді көрсетіп тұр?



74-сурет. Блог-диаграмма көрінісі

*5.4.Stacked Sequence Structure*

1. Жоғарыдағы есепті көпқабатты тізбектілік құрылымын қолданып шығарыңыз (Stacked Sequence Structure).

2. Ол үшін S**equence** құрылымындағы контексті мәзірден**Replace with Stacked Sequence** командасын қолданыңыз.

3. Мәліметтердің бір кадрдан екінші кадрға (Frame) қалай берілетіндігіне назар аударыңыз?

4. Бірінші кадрдағы синус мәнін шығыс индикатордың кірісімен байланыстырып көріңіз? Неліктен LabVIEW қатені көрсетіп тұрады?

5. **Sequence** құрылымының шығысында синустың соңғы мәнін қалай алады?

Сұрақтар:

1. Sequence құрылымының жұмысын айтыңыз?
2. Flat Sequence Structureнің Stacked Sequence Structureден айырмашылығы ?
3. Екінші тапсырмадағы 4 және 5 пункт сұрақтарына жауап беріңіз?

*5.5. Секундомер*

Қарапайым секундомер қызметін атқаратын ВҚ (виртуалды құрылғы) құру. Ол үшін төмендегідей амалдарды орындаңыз:

1.Блог диаграммада **Structures => Event Structures** таңдап, блог диаграммаға оқиғалар құрылымын орналастырыңыз.

2.Оқиғалар құрылымы контексті менюінен Edit **Events Handled by This Case-**ті таңдаңыз. **Event Sources** оқиғалар көзі бөлігінен This VI осы виртуалды құрылғы дегенді таңдаңыз.

Event оқиғалар бөлігінен Mouse Down-ды таңдаңыз. Дәл сол моменттегі Case варианты көрсетілген құрылым беттік панельге мышка батырмасын бір басқанда орындалады.

Lock front panel until the event case for this event completes-ден жалаушаны алып тастаңыз, мұның мағынасы көрсетілген Case орындалғанша беттік панельмен жұмыс жасауға мүмкіндік береді.

3. **Functions=>Execution Control**палитрасын қолданып, блог- диаграммаға **While** циклін орналастырыңыз (программаны тоқтату кнопкасы бар).

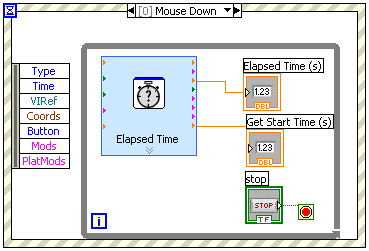
**4. Time & Dialog => Elapsed Time-**ды таңдап виртуалды құрылғы ішіне жұмсалған уақытты енгізіңіз. Пайда болған диалогтық терезеден  **Automatically reset after time target –**ден жалаушаны алып тастаңыз. Бұл функция уақыт есептегішін қандай да бір уақытқа жеткенде түсіріп тастайды. Біздің жағдайда бұл функция қажет емес.

**5.Elapsed Time (s), Get start Time (s)** шығыстары үшін индикациялар элементін құрыңыз. Ол үшін контексті менюден екі жағдайда да **Create => Indicator** таңдайсыз.

6.Беттік панельге өтіңіз. Индикация элементтері қасиеттерінен **Format and Precision** қосымша беттеріне өтіңіз. Ол үшін **Elapsed Time (s)** индикация элементінің контексті менюінен **Properties**-ді таңдаңыз. Сандарды бейнелеу форматындағы таңдау жолында **Relative Time**-ді таңдаңыз. **Get start Time (s)** индикация элементі үшін бейнелеу форматы негізінде **Absolute Time-**ді таңдау керек. Бұл бізге үйреншікті форматтағы уақытты бақылауға мүмкіндік береді.

**7.** Виртуалды құрылғыны іске қосыңыз. Беттік панельде тышқан батырмасын батыру арқылы **Elapsed Time (s)** уақытты есептейді,  **Get start Time (s)** есептеуді бастау моментін бейнелейді. Секундомерді **Stop**батырмасын басу арқылы тоқтатуға болады.

Блог-диаграмма 75-суретте көрсетілгендей көрініс алады:



75-сурет. Блог-диаграмма көрінісі

**5.6.Индикатор үшін оқиғаларды өңдеу.**

Берілген диапазонда нақты санды табуды тексеретін виртуалды құрылғы құрастырыңыз. Сонымен қоса виртуалды құрылғыны жапқан кезде виртуалды құрылғы оқиғаларды өңдеу қажет.

1. (Value) бақылау санын, (Max) максималды және (Min) минималды мәндер диапазонын енгізу үшін үш кіріс элементін құрастырыңыз (**Numeric Control**).

2.Нәтижені көру үшін беттік панельде **All Controls=>Boolean-**ге кіріп,(In Range?) **Round LED** индикаторын таңдаңыз. Сонымен қоса виртуалды құрылғы жұмысын тоқтату үшін **STOP** батырмасын орналастырыңыз.

3.Блог-диаграммада **While Loop** структурасын құрыңыз және бұл структура үшіне оқиғаларды өңдеу структурасы **Event Structure**-ні орналастырыңыз.

**4.Event Structure**-нің контексті менюінен **Edit Events Handled by This Case-**ті таңдаңыз.Үш индикатор үшін мәндер өзгерісін бақылаушы (Value Change) қызметін атқаратын 1 жағдай (оқиға) өңдеушіні құрыңыз. Жағдайды (оқиғаны) енгізу үшін **Edit Events** терезесінен «+» батырмасын басыңыз.

**5.**Диапазонда сандардың болуын тексеру үшін **All Functions=>Comparison** палитрасы ішінен **In Range and Coerce** функциясын таңдаңыз. Тексеріс нәтижесі **In Range?**индикаторында бейнеленуі керек: жанып тұрған лампа –диапазондағы сан, сөніп тұрған лампа- диапазоннан тыс сан.

6.**STOP** батырмасын басқан кезде программаны жабуды өңдеу үшін жаңа жағдай (оқиға) өңдеуші құрастырыңыз. Ол үшін **Case Structure** контексті менюінен **Add Event Case-**ті таңдаңыз. Бұл жағдай үшін блог-диаграммада қолданушыдан программаны жабуды растауды сұрайтын екі батырмалы диалогтық терезені құрыңыз. Қолданушының теріс жауабы кезінде программа өз жұмысын жалғастыру қажет.

7.**Application Exit?** оқиғасы үшін LabVIEW-ді жабу жағдайында диалогтық терезесі бар оқиғаларды өңдеушіні құрыңыз.

**8.**Виртуалды құрылғыны іске қосыңыз. Индикатордағы мәндерді өзгерте отырып, виртуалды құрылғының жұмысын тексеріңіз.**STOP** батырмасын басу арқылы және программаны толық беттік панельден виртуалды құрылғыны жапқан кездегі оқиғаларды өңдеудің дұрыстығын тексеріңіз.

6-бөлім.

**Кластерлер**

Бұл бөлімде кластер ішіне әр түрлі мәлімет типіндегі элементтерді біріктіру процесі жөнінде айтылады.

**6.1.Кластерлер**

Кластерлер әр түрлі типтегі мәліметтерді біріктіреді. Кластерлер программалау тіліндегі құрылымдар рөлін атқарады. Бірнеше мәліметтер тобын кластерге біріктіру блог-диаграммадағы бейберекетсіздіктің алдын алады және ВҚның ішкі программаларына қажетті енгізу/шығару мәліметтерінің жолдары санын азайтады. ВҚдағы енгізу/шығару элементтірінің максимум мәні 28-ге тең. Егер беттік панель 28-ден жоғары элементтен тұратын болса, онда ол элементтерді кластерге біріктіру тиімді. Беттік панель тек 28 элементті қабылдайды, ал 29-шы элементтен бастап қалған басқа элемнеттерді кластерге біріктіру қажет болады. Массивтер сияқты кластер басқару және бейнелеу кластерлері бола алады. Алайда бір кластербірдей басқару және бейнелеу элементтерінен тұра алмайды. Тек басқару немесе бейнелеу элементтерінен ғана тұра алады. Сонымен қоса, массивтердегідей кластерлерде де барлық элементер реттілікпен орналасады. Бірақ оларға индекстері бойынша сілтеме жасауға болмайды. Оларды алғашында жіктеп алу керек. Ол үшін **Unbundle By Name** функциясы қолданылады. Кластердегі қандай да бір элементке аты бойыншасілтеме жасайды.

**6.1.1.Басқару және бейнелеу элементтері көмегімен кластер құру**

Басқару және бейнелеу элементтерінен тұратын кластер құру үшін **Controls >> Modern >> Array, Matrix & Cluster** палитрасы ішінен кластер шаблонын таңдайсыз және оны беттік панельге орналастырасыз. Осыдан кейін кластер шаблонын элементтерімен толтыру керек. Кластер өлшемін меңзер көмегімен өзгертуге болады.

Төменде (76-сурет) 3 басқару элементінен тұратын кластердің көрінісі бейнеленген:

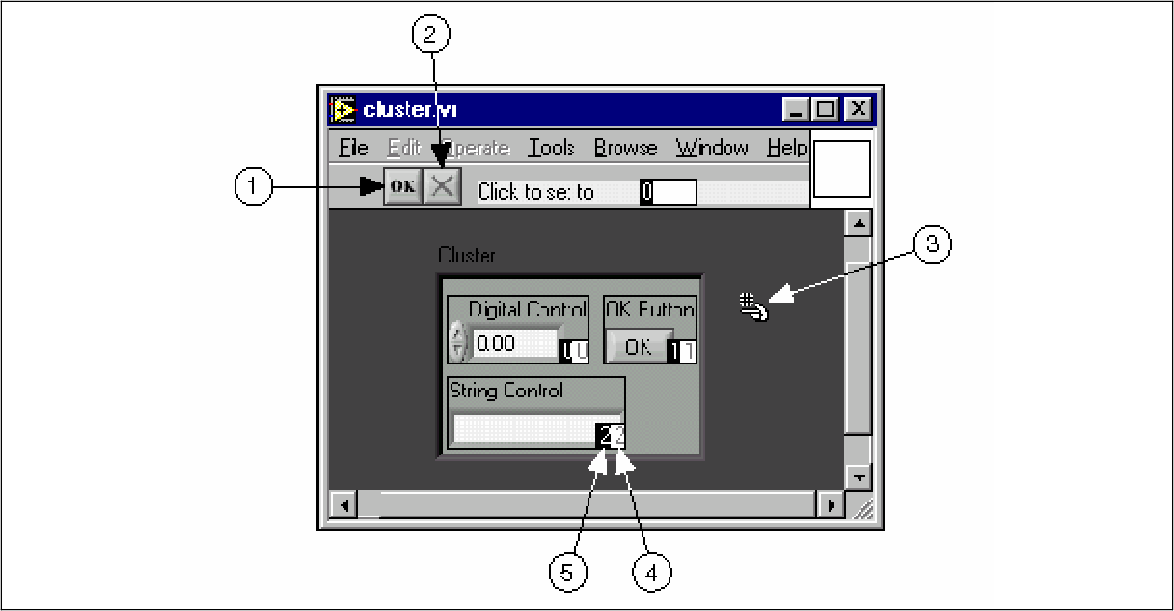


76-сурет. Кластер бейнесі

**Кластердегі элементтер реті**

Шаблондағы орнына байланыссыз кластердегі әрбір элементтің өзінің логикалық реті болады. Кластерге бірінші боп енгізілген элементке автоматты түрде 0 номері беріледі. Сәйкесінше екінші элементке 1 номері және т.с.. Кластердегі элементті жойғанда ондағы элементтер номері автоматты түрде өзгереді.

Кластердегі элементтердің реті кластер элементтерінің **Bundle** (біріктіру) және **Unbundle** (ажырату) функциясы бойынша реттелгендігін көрсетеді. Кластердегі объектілердің реттілік номерін көру және өзгерту үшін тышқан меңзерін кластердің бір шетіне апарып оң жағын басып контексті мәзірден **Reorder Control In cluster**  пунктін таңдау керек. Жабдықтар панелі және кластер төмендегідей көрініске ие болады (77-сурет):



1-Растау батырмасы (Confirm button); 2-Болдырмау батырмасы (Cancel button)

3- Реттілікті анықтау меңзері (Cluster order cursor); 4- Сол моменттегі реттілік номері (Current order); 5- Жаңа реттілік номері (New order)

77-сурет. Кластердегі объектінің реттік нөмерін көру терезесінің көрінісі

Ақ түсті фонда (4) элементтің сол моменттегі номері, ал қара түсті фонда (5)-элементтің жаңа реттік номері көрсетілген. Элементтің реттік номерін орнату үшін **Click to set to** мәтінді енгізу жолына санды енгізу керек және элементті тышқан батырмасымен басу қажет. Элементтің реттік номері өзгереді. Осыған сәйкес кластердегі басқа элементтердің де реттік номері өзгереді. Өзгерістерді жабдықтар панеліндегі **Confirm** батырмасын басу арқылы сақтап отырады. Алдыңғы күйді қайта қалпына келтіру үшін **Сancel** батырмасын басу керек.

Бірдей реттілік номерімен анықталынатын сәйкес келетін элементтер бірдей мәлімет типіне ие болу керек. Мысалы, бірінші кластерде 0 элементі сандық басқару элементі болсын, ал 1 элементі жолдық басқару элементі болсын. Екінші кластерде 0 элементі сандық бейнелеу элементі, 1 элементі-жолдық бейнелеу элементі. Мұндай кластерді біріктіре алмайсыз. Себебі типтері сәйкес келмейді. Біріктіруді жүзеге асыру үшін мысалдағы бірінші және екінші кластер екеуі де сандық басқару немесе сандық бейнелеу элементтерінен тұру керек. Алайда егер екі кластердің біреуінің элементінің реттік номерін өзгертіп жіберсек, онда кластер арасындағы мәлімет өткізгіші ажыратылып қалады.

**6.1.2.Тұрақты кластерді құру.**

Блог-диаграммада **Function >> Programming >> Cluster & Variant** палитрасы ішінен **Cluster Constant** шаблонын таңдап, оның ішіне сандық тұрақтыны немесе қандай да бір мәлімет объектісін, логикалық элемент немесе жолдық элементті енгізу арқылы құруға болады.

Егер де беттік панельде кластер құрылып қойылса, онда блог-диаграммада тұрақты кластерді құру үшін беттік панельдегі кластерді блог-диаграммаға көшіріп немесе кластерге тышқан батырмасын апарып, оң жақ батырмасын басып, контексті мәзірден Create >> Constant пунктін таңдасаңыз жеткілікті.

**6.2.Кластермен жұмыс жасайтын функциялар**

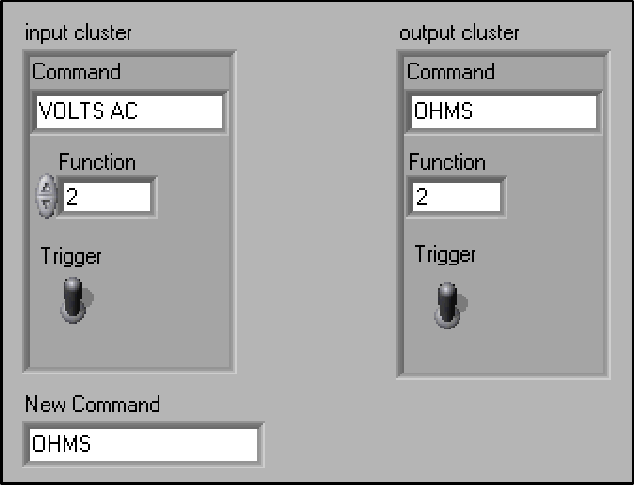
Кластерді құру және басқару үшін **Function >> Programming >> Cluster & Variant** палитрасы ішіндегі **Bundle, Bundle By Name, Unbundle, Unbundle By Name** функциялары қолданылады. **Bundle, Bundle By Name** функциялары кластерді басқару және құрастыру үшін қолданылады. Ал, **Unbundle, Unbundle By Name** кластерді жіктеу үшін қолданылады.

Бұл функцияларды кластер терминалына тышқанды апарып, оң жақ батырмасын басып контексті мәзір ішінен **Cluster Tools** подмәзірін таңдасаңыз болады. **Bundle** және **Unbundle** функциялары автоматты түрде мәліметтерді енгізу/шығару жолы санын көрсетеді. Ал **Bundle By Name** және **Unbundle By Name** функциялары мәліметтерді енгізу/шығару жолында кластердегі бірінші элементтің атын көрсетеді.

**Кластер құрастыру**

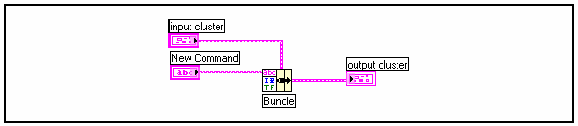
Жеке элементтерді кластерге біріктіру үшін **Bundle** функциясы қолданылады. Осыған дейін құрылып қойылған кластердегі элемент мәліметтерін өзгерту үшін де осы функция қолданылады. Орын алмастыру құралы кластердегі мәліметтерді енгізу жолын арттыру үшін қолданылады. Кластер мәліметтерінің кірісін арттырудың келесі жолы, ол кластердің мәліметтерді енгізу кірісіне тышқан батырмасын апарып, оның оң жағын басып, контексті мәзір ішінен **Add Input** пунктін таңдайсыз. Кластер кірісіне берілетін элементтер саны кластер ішіндегі элемент санына тең болу керек.

Кластердің мәліметтерді енгізу жолына тек бір ғана өзгертетін шаманы беру керек. Мысалы, төменде 3 басқару элементінен тұратын кластер бейнесі көрсетілген (78-сурет):



78-сурет. 3 басқару элементінен тұратын кластер бейнесі

Егер де элементтердің логикалық реттілігі белгілі болса, онда **Command** элементінің мәнін өзгерту үшін элементтерді төменде көрсетілгендей етіп байланыстырып, **Bundle** функциясын қолдануға болады.



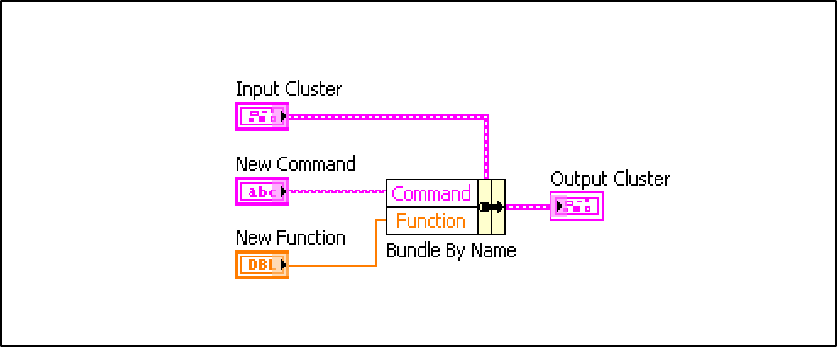
79-сурет. **Bundle**функциясы көмегімен Сommand элементінің мәнін өзгерту көрінісі

**Кластер элементтерін алмастыру және кластер элементтеріне қолжетімділік**

Осыған дейін құрылып қойылған кластердегі элементті өзгерту үшін **Bundle by Name** функциясы қолданылады. **Bundle by Name** функциясы **Bundle** функциясы секілді жұмыс жасайды. Бірақ ол кластер элементтеріне оның реттік номері бойынша емес, аты бойынша сілтеме жасайды. Осыған сәйкес кластердегі элементтерге тек аты бойынша ғана қолжетімділікті қалыптастыруға болады. Кластердің кіріс жолдары кластер ішіндегі элементтердің санына сәйкес келуі міндетті емес.

Басқару элементі көмегімен мәліметтерді енгізу жолы терминалына тышқан батырмасын апарып, оң жағын басып, шыққан мәзір ішінен қажетті кірісті таңдауға болады. Бұл амалды мәліметтерді енгізу жолына тышқанның оң жақ батырмасын басып, контексті мәзір ішінен **Select Item** пунктін таңдау арқылы да орындауға болады.

**Commond Function** элементтерінің мәнін өзгерту үшін **Bundle by Name**функциясын қалай қолдану керектігі төменде көрсетілген (80-сурет):



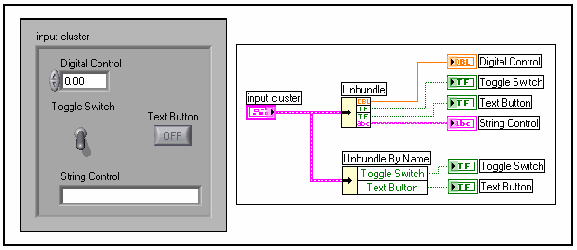
80-сурет.**Commond Function** элементтерінің мәнін өзгертуде **Bundle by Name** функциясын қолдану көрінісі

**Bundle by Name** функциясын жұмыс барысында өзгеруі мүмкін мәліметтер құрылымдарымен жұмыс кезінде қолданған жөн. Кластерге жаңа элемент енгізу үшін немесе элементтің реттік номерін өзгерту үшін **Bundle by Name**функциясын қайта қосудың қажеті жоқ. Себебі элементтің атауы әлі де нақты.

**6.3.Кластерді жіктеу**

Кластерді жеке элементтерге жіктеу үшін **Unbundle** функциясы қолданылады. **Unbundle by Name** функциясы кластердегі элеметті нақты бір аты бойынша белгілеп алады. Мәліметтерді енгізу жолы кластердегі элементтердің санына тәуелді емес.

Басқару элементі көмегімен мәліметтерді енгізу жолында тышқанның оң жағын басып, контексті мәзір ішінен қажетті элементті немесе **Select Item** пунктін таңдауға да болады. Мысалы, **Unbundle** функциясы (төменде көрсетілген кластерді қолданған кезде) кластер ішіндегі 4 элементке қатысты 4 мәліметтерді енгізу жолынан тұрсын. Бұл мысалда элементтер 0-ден бастап жоғарыдан төменге дейін орналасады. Егер **Unbundle by Name** функциясын қолдансақ, онда мәліметтерді енгізу жолдары өздік таңдау бойынша біз таңдап алған санға ие болуы мүмкін. Жеке элементтерге таңдау бойынша сілтіме жасауға болады (81-сурет).



81-сурет. **Unbundle by Name**және **Unbundle**функциялары көмегімен кластерді жіктеу

**6.4. Қателік кластері**

Тіпті мінсіз жұмыс жасап тұрған программаның өзінде қателіктер кездесіп жатады. Сондықтантұтынушыда пайда болуы мүмкін қателіктерді ешқашан да болжай алмайсыз. Қателіктерді тексеру механизмінсіз ВҚ жайында тек оның жұмыс жасамайтындығын айта аласыз. Қателікті тексеру ВҚның қай жерінде байланыстың бұзылғанын, қателіктң қай жерден кеткендігін көрсетеді.

Кез келген операцияны программалау кезінде қателіктердің пайда болу мүмкіндігін ескеру керек. Барлық енгізу/шығару операциялары қателік жөнінде хабарды жібереді. Қателіктерді дұрыс өңдеу үшін ақпаратты жіберу процесстерін, файлдық және тізбектік операцияларды, құралдармен амалдарды, алынған мәліметтерді барлығын мұқият тексеру керек.

ВҚдағы пайда болуы мүмкін қателерді тексеру келесідей мәселелерді шешеуге көмектеседі:

\*Сыртқы құралдармен қате инициализация немесе оған қажет емес ақпаратты жазу.

\*Сыртқы құрылғы қосылмаған немесе жұмыс жасамайды.

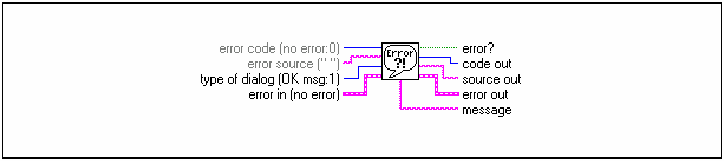
\*Желілік программалық қамтамасыз етуді орнату кезінде немесе қандай да бір себеп әсерінен қажетті файлға жол өзгерген болса .

**Қателікті өңдеу**

LabVIEWде қателікті автоматты түрде өңдеу ұйымдастырылмаған. Бұл, тұтынушыға қателікті өңдеудің өзіндік әдістерін таңдауға мүмкіндік береді. Егер де ВҚ үшін ақпаратты енгізу/шығаруды күту үшін уақыт өтіп кетсе, онда барлық қосымшаның уақытын тоқтамайтындай етуге болады. Тіпті ВҚда мүмкіндікті белгілі бір уақыттан кейін қайта қайталауға мәжбүр етуге де болады. LabVIEWда қателерді өңдеу тек блог-диаграммада жүзеге асады.

ВҚда және функцияларда қателерді қайтарудың екі әдісі қолданылады: қателік кодын көрсететін сан арқылы және кластер қателіктері арқылы. Негізінен функциялар санды қолданады. Ол қатенің коды болып табылады. Ал ВҚ кірісті қабылдап, шығысында қатені кластер түрінде шығарады. Басқа да мәліметтер сияқты қателік жөніндегі ақпараттар ВҚ арқылы өтеді. Қателік жөніндегі ақпаратты ВҚ арқылы жіберу үшін кіріс және шығыс қателік кластерін қолдану керек. Сонымен қоса ВҚның соңына ВҚжұмысы процесінде жүйенің істен шығуы (тоқтауы) болғанын анықтайтын қатені өңдеуші құрылғыны қосу керек.

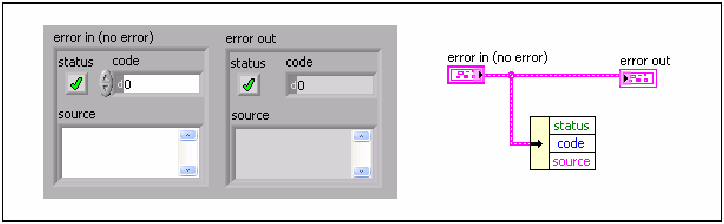
ВҚда пайда болған қатені өңдеу үшін төмендегідей қарапайым қате өңдеуші **Simple Error Handler** қолданылады. **Simple Error Handler** функциясы **Function >> programming >> Dialog & User Interface** палитрасының ішінде орналасқан. Қателік кластерін **Simple Error Handler** функциясының **Error in** кірісіне жалғау керек (82-сурет).



82-сурет.**Simple Error Handler**функциясының бейнесі

**Қателік кластері құраушылары**

83-суретте **Controls >> Modern >> Array,Matrix & Cluster** палитрасында орналасқан қателік кластері компонеттері көрсетілген. Қателік кластері, өткізгіштер, қателік кластерін байланыстырушылар сары түспен белгіленеді.



83-сурет. Қателік кластерінің көрінісі

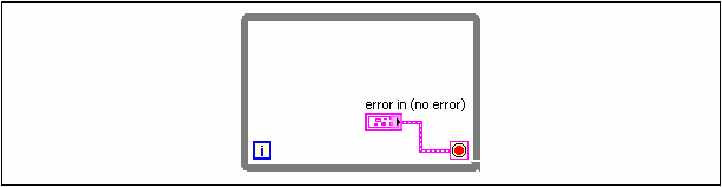
**Status** –логикалық шама. Қате пайда болған кезде True мәнін қабылдайды.

**Code**- қателік санына сәйкес келетін 32 битті бүтін сан. Егер де **Status False** шамасын қабылдаса және **code** 0-ден басқа санға тең болса, онда ВҚда қателік емес тек ескерту ғана бар деген сөз.

**Source** қателіктің орнын анықтайтын жол болып табылады.

**Қателікті өңдеу кезінде While циклін қолдану**

Циклді тоқтату үшін қателік кластері **While** циклінің шарт терминалына байланыстырылуы мүмкін. Қателік кластері шарт терминалына байланысқан кезде терминалға тек statusқа - True немесе False мәндері ғана беріледі. Қателік пайда болған сәтте **While**циклі жұмысын тоқтатады.



84-сурет.Қателікті өңдеу кезінде**While** циклін қолдану

Егер шарт терминалына қателік кластерін жалғасақ, онда контексті мәзірдің **Stop if Тrue** және **Continue if True** пунктері **Stop on Error** және **Continue while Error** пунктеріне ауысады (84-сурет).

**Өтілген материалдардың қысқаша мазмұны, кеңестер, құпиялар**

* Кластерлер әр түрлі типтегі мәліметтерді біріктіреді. Бір мезетте кластер басқару және бейнелеу элементтерін құрай алмайды.
* Егер беттік панель 28-ден артық басқару және бейнелеу элементтерінен тұрса, онда оларды кластерге біріктіру қажет.
* Басқару және бейнелеу элементтерінен тұратын кластерді құру үшін **Controls >> Modern >> Array, Мatrix & Cluster** палитрасы ішінен кластер шаблонын таңдап, оны беттік панельге орналастыру керек. Осыдан кейін кластер шаблонын элементтермен толтыру қажет.
* Кластерді құру және оларды басқару үшін **Function >> Programming >> Cluster & Variant** палитрасында орналасқан функциялар қолданылады.
* Қателік кластері мынадай элементтерді құрайды: **status** (статус),  **code** (код) және **sourc**e (қорек көзі).
* Қателіктерді тексеру ВҚның қай жерінде және не себептен ақау пайда болғанын анықтайды.

**6-лабораториялық жұмыс. Кластерді масштабтау. Радиобатырмалар.**

6.1.Кластерді масштабтау

Кластерде әрбір элемент мәнін масштабтайтын (масштабтау коэффициенттері сәйкес келетін) виртуалды құрылғы құру керек. Біз қысым мәнін, ағын жылдамдығын және температура мәндерін кернеу датчигімен сәйкестендіріп алынған деп алайық. Содан кейін виртуалды құрылғы бұл мәндерді масштабтайды және физикалық шаманың нақты мәнін шығарады.

1.Беттік панельде кластер шаблонын құрыңыз және оны «Кернеу мәндері» деп атаңыз. Осы кластерге 3 сандық басқару элементін енгізіңіз (**Numeric Control**) және оларды В (температура), В (қысым) және В (шығын) деп атаңыз.

2. Бірінші кластер жанына келесі екінші кластерді құрыңыз және оны «Масштабталған мәндер» деп атаңыз. Бұл шаблонға **Numeric** палитрасының ішінен **Vertical Fill Slider**, **Tank** және **Gauge**  элементтерін таңдаңыз. Барлық 3 индикациялық (шығыс) элементері үшін контексті менюден сандық мәндерді көрсету үшін **Visible Items => Digital Display** таңдаңыз. Шығыс элементтерді сәйкесінше Температура (С), Шығын (литр/секунд), Қысым (кН/м^2) деп атаңыз.

3. Блог диаграммадағы «Масштабталған мәндер» кластері үшін контексті менюден **Change To Indicator**-ды таңдаңыз.

4. Беттік панельдегі «Кернеу мәндері» кластерін блог диаграммаға көшіріп, кластер константаны (кластер тұрақты ) құрыңыз. Ұшықтарға мына мәндерді жазыңыз: 5, 1.2 ,0,78 .

5. Көбейту функциясын қолдана отырып (Multiply) «Кернеу мәндері» мен кластер тұрақтыны көбейтіп, шығысына «Масштабталған мәндер» кластерін жалғаңыз.

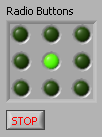
6.Виртуалды құрылғыны іске қосыңыз (85-сурет). Температура, шығын, қысым мәндерін өзгертіңіз. Контексті менюден **Data Range-ді таңдап Use Default Range** үшін жалаушаны алып тастап, индикаторлар үшін мәндер диапазонын өзгертіңіз.



85-сурет.Беттік панель көрінісі

6.2. Радио батырмалар

Зертханалық жұмыстың басты мақсаты: радио батырма жұмысын атақаратын виртуалды құрылғы жасау болып табылады (86-сурет). Кластерде көп кнопкалар ішінен тек бір кнопка басылуы мүмкін.



86-сурет. Беттік панельдегі радиобатырмалардың көрінісі

1.Кластер шаблонын құрыңыз және оны «Radio Buttons» деп атаңыз. Осы кластер ішіне **Round LED** индикация элементін орналастырыңыз. Контексті меню ішінен **Visible Items** менюін таңдап **Label** деген бөліктен жалаушаны алып тастаңыз. Бұл индиктаор атын алып ьастау үшін қажет. Осындай тағы да 8 индикатор құрыңыз. Операцияны қайтадан қайталамас үшін **Сtrl** кнопкасын қолдана отырып, бір индикаторды бірнеше рет көшіріп қоюға болады.

**2.** Виртуалды құрылғыны өшіру үшін **STOP** батырмасын құрыңыз. Ол үшін While Loop құрылымын қолдансаңыз болады.

**3.** Болг диаграммада контексті мәзірден **Create => Local Variable-**ді таңдап«Radio Buttons» кластері үшін локальді айнымалыны құрыңыз. Локальді айнымалы мәнінің өзгерісі элементтің өзінде көрініс табады. Кластердегі мәндерді өзгерту үшін оны басқару элементіне алмастыру керек (**Change to Control**).

4. Тек бір ғана батырма басылу қажетті болу үшін «Radio Buttons» кластерінің мәнінің өзгеру жағдайын қадағалап отыру керек. Ол үшін жағдайларды (оқиғаларды) өңдеу құрылымын құрыңыз. **Edit Events** терезесінен«Radio Buttons» құрылымы үшін (барлық) **Value Changed-**ті таңдаңыз.

**5. Round LED-**тіңкез келген бір элементін өзгерту және жағдайларды (оқиғаларды) өңдеу структурасы үшін мәліметтер терминалы арқылы кластердің мәндеріне **(Old Val)** ескі мәні және **(New Val)** жаңа мәні беріледі. Жаңа мәндегі екі бульдік айнымалылар true мәніне ие болады: оның алдында басылған батырма және содан кейін басылған батырма. Соңғы басылған батырманы қосылған қалпында қалдыру үшін Exclusive Or батырмасын қолдануға болады. Мұны қалай іске қосу жолын ойланып көріңіз.

6.Stop батырмасын басу арқылы виртуалды құрылғыны өшіру үшін тағы да жағдайды (оқиғаны) өңдеу құрылымын құрыңыз.

7.Виртуалды құрылғыны іске қосыңыз және жұмысын тексеріңіз.

6-тапсырма.

7-бөлім.

**Мәліметтерді графикалық бейнелеу**

Бұл бөлімде график көмегімен мәліметтер визуализациясының түрлері үшін қажетті Диаграмма графигі (**Waveform Chart)**, осциллограмма графигі (**Waveform Graph)**, осциллограмманың екі координаталық графигі (**XY graph)** , интенсивтілік графигі (**Intensity Graph)** қарастырылған.

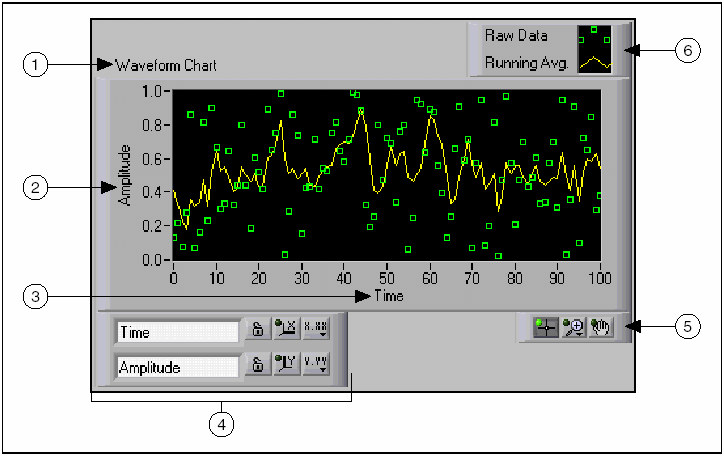
**7.1. Диаграмма графигі**

Диаграмма графигі (Waveform Chart) - бір немесе бірнеше график көмегімен мәліметтерді бейнелеудің арнайы элементтері. Диаграмма графигі **Controls >> Modern >> Graph** палитрасы ішінде орналасқан. 87-суретте тәжірибе және теорияда алынған мәндерді көрсететін екі графиктен тұратын Диаграмма графигі көрсетілген.

Диаграмма графигі мәліметтерді бейнелеудің 3 әр түрлі режимдерін қолданады. Олар **strip chart** , **scope chart** және **sweep chart** (88-сурет).

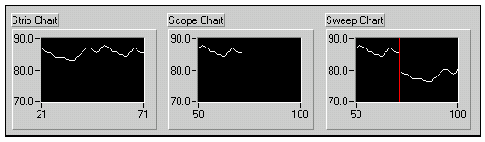
Режимдерді беру үшін диаграммаға тышқан меңзерін апарып, оң жақ батырмасын басып, контексті мәзірден **Advanced >> Update Mode** пунктін таңдау қажет.

**Strip chart** режимі қағаздық лента секілді солға, оңға бұрылатын экранды бейнелейді. **Scope chart** және **sweep chart** режимдері осциллограф экранына ұқсайды. **Strip chart** режимінен айырмашылығы мәліметтерді бейнелеу жылдамдығы өте жоғары. **Sweep chart** режимі **scope chart** режиміне қарағанда график жолын тазаламайды.



1-Атауы (Label); 2-Y шкаласы (Y-scale); 3- X шкаласы (X-scale); 4- шкаланы басқару панелі (Scale legend); 5- графикпен жұмыс жасауға қолданылатын жабдықтар палитрасы (Graph palette) ; 6- графикті басқару панелі (Plot legend )

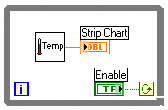
87-сурет. LabVIEWдегі диаграмма графигінің көрінісі



88-сурет. Диаграмма графигіндегі мәліметтерді бейнелеудің үш режимі

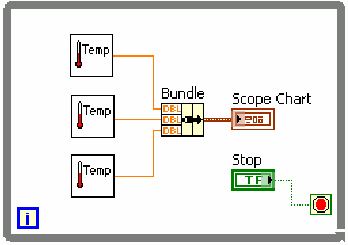
**7.1.1.Графиктерді байланыстыру**

Диаграмма құру үшін **Диаграмма графигінің** мәліметтер терминалына скалярлы шаманың шығыс жолын байланыстыру жеткілікті. Келесі мысалда, **Диаграмма графигі** терминалындағы мәлімет типтері кіріс мәлімет типіне сәйкес келеді (89-сурет).



89-сурет. Графиктерді біріктіру

Диаграмма графигі бірнеше графикті бейнелей алады. Бейнеленетін мәліметтерді біріктіру үшін **Function >> Programming >> Cluster & Variant** палитрасында орналасқан **Bundle** функциясы қолданылады. Төмендегі блог-диаграмма**Bundle** функциясының көмегімен үш ішкі программаның шығыс мәліметтерін Диаграмма графигінде бейнелеу үшін біріктіреді (90-сурет).

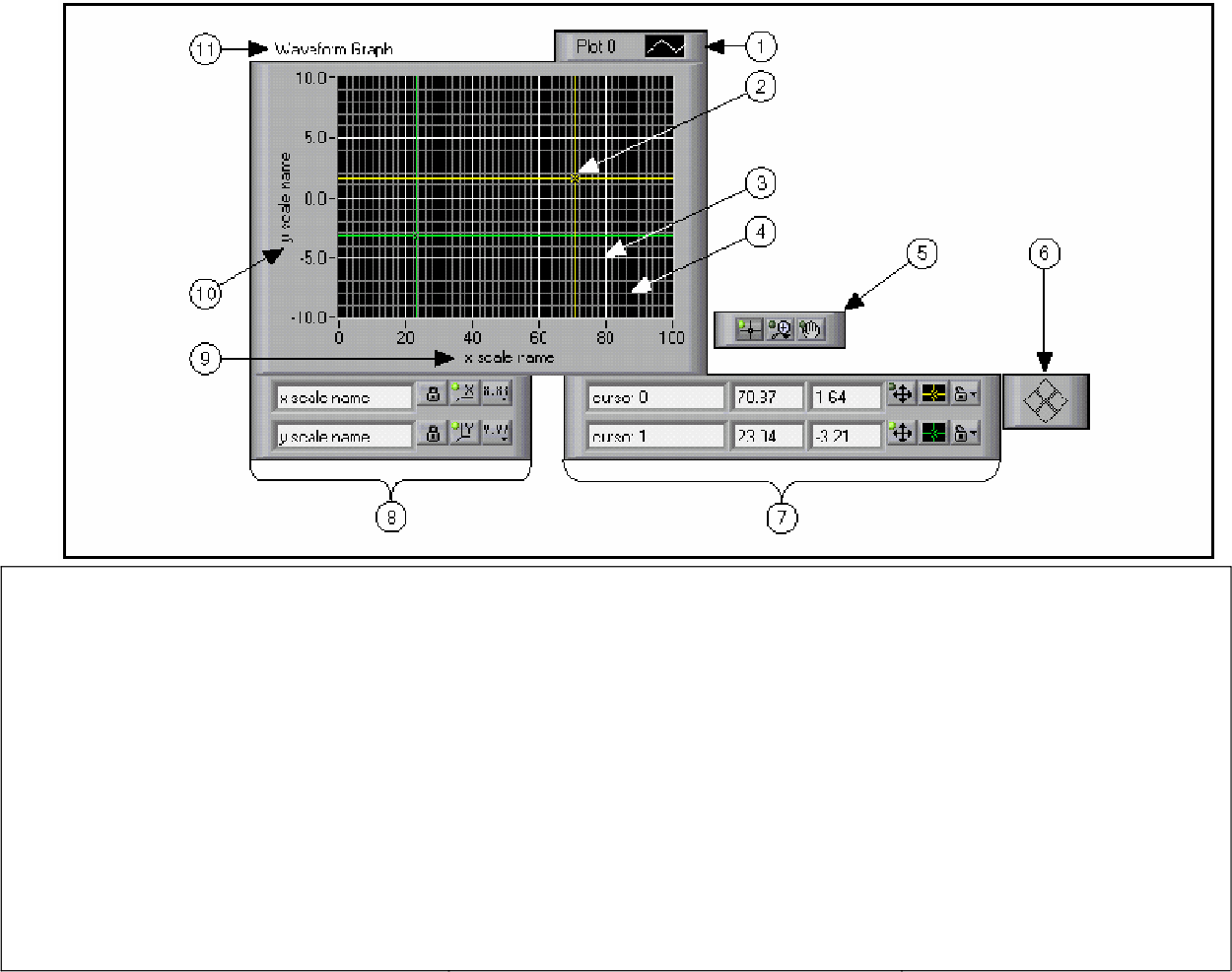


90-сурет. **Bundle** функциясының көмегімен мәліметтерді бейнелеу көрінісі

Диаграмма графигінің мәлімет терминалы **Bundle** функциясының шығыс жолдарына сәйкес келетін кластерлік мәлімет типтерін иеленеді. **Bundle** функциясының кірістерін арттыру үшін орын ауыстыру терминалы көмегімен оның терминалын өзгерту керек.

**7.2. Осциллограмма және екі координатты Осциллограмма графигі**

Виртуалды құрылғы массивке жиналған мәліметтерді осциллограмма түріндегі график көмегімен бейнелейді. Төмендегі иллюстрацияда график элементтері көрсетілген (91-сурет).



1-Осциллограмма қасиеттерін басқару панелі (Plot legend); 2- Меңзер (Cursor); 3-Негізгі өлшемдік тор (Grid mark); 4-Қосымша өлшемдік тор (Mini-grid mark); 5- графикті басқару элементінің палитрасы (Graph palette); 6-меңзердің орын ауыстыру панелі (Cursor mover); 7-меңзер қасиеттерін басқаратын панель (Cursor legend); 8-шкаланы басқару панелі (Scale legend); 9-Х шкаласы (X-scale); 10-У шкаласы (Y-scale); 11-графиктің өзіндік белгісі (Label).

91-сурет. Осциллограмма графигі элементтерінің көрінісі

Осциллограмма графигі (**Waveform graph**) және екі координаталы Осциллограмма графигі (**X-Y Graph**) **Controls >> Modern>>Graph** палитрасының ішінде орналасқан. Осциллограмма кез келген нүкте жиынын бейнелейді. Бірнеше осциллограмма жиынын бейнелеу үшін **Plot legend** панелінің өлшемін міндетті түрде өзгерту керек. Осциллограмма жиынының графигі беттік панельде бос орынды арттыру үшін қолданылады.

**7.2.1.Жеке осцилллограмма графигі**

Бір мәнді Осциллограмма графигі бір өлшемді массивтермен жұмыс жасайды және массивтің мәліметтерін графиктегі нүкте түрінде көрсетеді. График сонымен бірге кластерді де бейнелейді. Ол үшін кластергеx,∆x бастапқы мәні беріледі, ал массив мәліметі *y* шкаласы бойынша орнатылады.

**7.2.2.Осциллограмма жиыны графигі**

Осциллограмма жиыны графигі мәліметтердің екі өлшемді массивімен жұмыс жасайды. Мұнда массивтің әрбір жолы мәліметтердің жеке осциллограммасын көрсетеді және массивтің мәліметтері нүкте түрінде көрсетіледі.

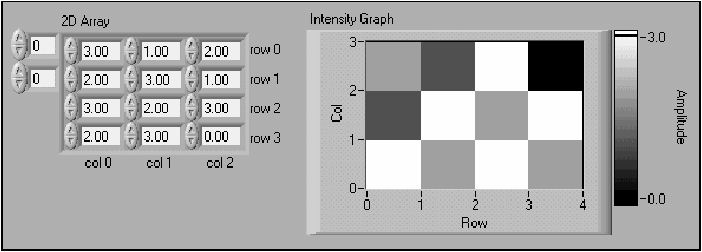
**7.2.3.Жеке екі координаталы Осциллограмма графиктері**

Жеке екі координаталы Осциллограмма графигі x және у массивтерін құрайтын кластермен жұмыс жасайды. Осциллограмманың екі координаталы графигі массивтерді нүкте ретінде қабылдайды.

**7.3. Интенсивтілік графиктері (қосымша)**

Интенсивтілік графиктері және кестелері (**Intensity graphs and charts**) екі өлшемді мәліметтерді бейнелеу үшін ыңғайлы. Мысалы, амплитудасы теңіз деңгейіндегі биіктік болып табылатын аумақтың топографиясын көрсету үшін. Диаграмма графигі мен Осциллограммасы жағдайындағыдай интенсивтілік графигінде де дисплейдің тұрақты өлшемі болады. Интенсивтілік графиктері және кестелері кірісіне екіөлшемді массивті қабылдайды. Массивтің әрбір саны қандай да бір нақты түске сәйкес келеді. Графиктегі түстердің орналасуы массивтегі элементтердің индексімен анықталынады. Интенсивтілік графиктері және кестелері 256 түске дейін бейнелей алу мүмкіндігіне ие.

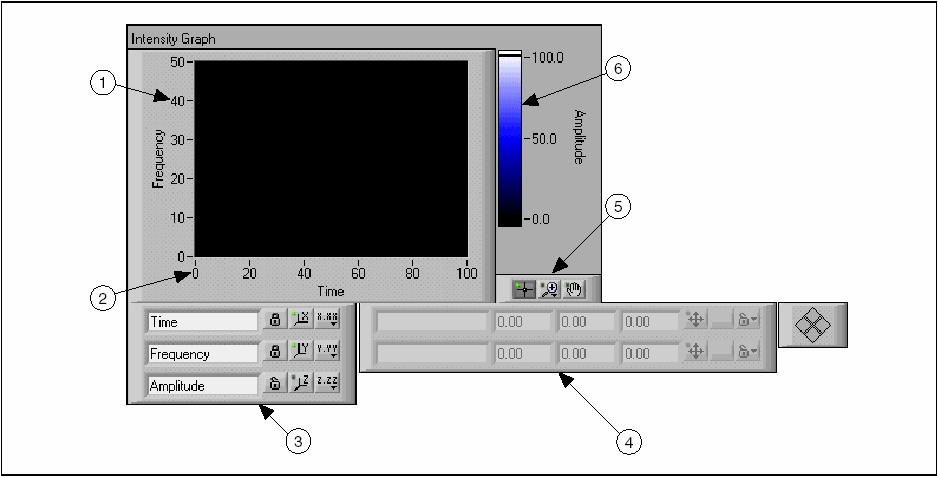
Келесі иллюстрацияда интенсивтілік графигінде визуализацияланған 4х3 өлшемді массив бейнеленген. График транспонирленген массивті бейнелейді (92-сурет).



92-сурет. Интенсивтілік графигінің көрінісі

**7.3.1.Интенсивтілік графигі және кестесін орнату**

Интенсивтілік графигі және кестесі Диаграмма және Осциллограмма графигімен ортақ қасиеттерге ие. Интенсивтілік графигі мен кестесінде үшінші өлшем пайда болатындықтан, қосымша элемент қажет болады. Бұл қосымша элемент мәліметтердің диапазонын және түстік бейнесін көрсететін түстік шкаланы басқару элементі. Төменде интенсивтілік графигін құраушы бөліктер көрсетілген (93-сурет).



Y шкаласы (Y scale); 2.X шкаласы (X scale); 3. Шкаланы басқару панелі (Scale legend); 4. Меңзерді басқару панелі (Scale legend); 5. Графикпен жұмыс жасау үшін қажетті палитра (Graph palette); 6. Z шкаласы (Z scale (color ramp))

93-сурет. Интенсивтілік графигін және кестесін орнату көрінісі

Маркермен ассоциацияланған түсті өзгерту үшін контексті мәзір ішінен **Marker Color** пунктін таңдап, түсті таңдау терезесі ішінен қажетті түсті таңдайсыз. Контексті мәзір Басқару немесе Орын алмастыру элементі көмегімен шақырылады. Түстік шкалаға маркерді енгізу үшін түстік палитраға тышқан меңзерін апарып, оң жақ батырмасын басып контексті мәзірден **Add Marker** пунктін таңдайсыз.

**7.4. Үшөлшемді сахна құру (қосымша)**

Үшөлшемді сахна құру қандай да бір физикалық процессті визуализациялау мақсатында қолданылады. Сіздер үшөлшемді объектілердің формасын, өлшемін, текстурасын, қозғалу әдістерін, сахна ішіндегі басқа объектілермен өзара қатынастарын құруыңызға болады. 3D объектілермен жұмыс жасау үшін **Function >> Programming >> Graphics & Sound >> 3D Picture Controls**  палитрасы ішіндегі виртуалды құрылғыны қолдануыңызға болады. Бұл палитра келесідей енгізілген палитралардан тұрады:

**Object VIs:**



**Create Object VI** –3D объектіге сілтеме орнатады.



**Find Object VI** – 3D объектіге сілтемені табады.

**File Loading VIs()** – осы белгі орналасқан виртуалды құрылғыны 3D сахнада модельдер мен сахналарды құрайтын осыған дейінгі құрылған файлдарды енгізу үшін қолданыңыз.

**Geometries VIs** - бұл палитрада 3D объектіге геометриялық форманы беретін виртуалды құрылғы орналасқан:

**Create Cone VI** – берілген параметрлері негізінде конусты құрады.

**Create Cylinder VI** – берілген параметрлер негізінде цилиндр құрады.

**Create Box VI-** берілген параметр негізінде параллелепипед құрады.

**Create Sphere VI** – берілген параметлер негізінде сфераны құрады.



**Create Height Field VI-** екі өлшемді мәлімет массивін 3 өлшемді объектіге түрлендіреді.

**Create Мesh VI** – 3 өлшемді координаталық торды құрады.

**Transformations VІs** – сахнадағы 3D объектінің орнын беруге мүмкіндік береді.

**Helpers VIs** – сахнада әр түрлі амалдарды орындауға мүмкіндік береді.

**Өтілген материалдарға қысқаша шолу, кеңестер мен құпиялар**

Диаграмма графигі (Waveform Chart) – бір немесе бірнеше диаграммаларды бейнелеу үшін қолданылатын арнайы цифрлық бейнелеу элементі.

Диаграммаграфигі мәліметтерді бейнелейтін 3 режимнен тұрады:

* **Strip chart** оңға және солға айналдырылатын қағаз лентаға ұқсайтын экранды көрсетеді.
* **Scope chart** диаграмманың оң жақ бөлігі тазартылады және диаграмманы толтыру сол жақ бөліктен басталады.
* **Sweep chart** режимінің **scope chart** режимінен айырмашылығы диаграмма терезесі тазартылмайды. Ал жаңа мәліметтер ескі мәліметтерден вертикалды сызық арқылы ажыратылады.

Осциллограмма графигі (Waveform graph) және екі координаталы Осциллограмма графигі (XY graph) массив ішіндегі мәліметтерді бейнелейді.

Диаграмма графигін өңдеу және орнатуын өзгерту үшін диаграмманың оң жақ бөлігіне тышқан меңзерін апарып, оң жақ батырмасын басу керек.

Бір графикте **Build Array** функциясы көмегімен бірнеше осциллограмманы шығаруға болады. Бұл функция **Function >> Programming >> Array**  палитрасы ішінде орналасқан. Екі координаталы осциллограмма графигіне мәліметтер **Function >> Programming >> Cluster & Variant**  палитрасы ішінде орналасқан **Bundle** функциясы көмегімен енгізіледі.

Үш өлшемді мәліметтерді визуализациялау үшін интенсивтілік графигі мен кестелерді қолдануға болады. Интенсивтілік графиктері мен кестелері спектральді талдау есептерінде, температураны визуализациялауда және бейнелерді өңдеуде қолданылады.

Диаграммаға немесе графикке мәліметерді енгізу үшін контексті терезедегі **Context help** анықтамалық терезесін қолдануға болады.

**Қосымша жаттығулар**

1. Екі кез келген сіз таңдап алған сан және sweep chart режимінде Диаграмма графигінде 4 нүктенің ортасында жататын сан арқылы екі график тұрғызатын виртуалды құрылғы құрыңыз. Мына кеңестерді қолданыңыз:

-While циклының орнына For (N=200) циклын қолданыңыз.

-соңғы төрт мәннің орта мәнін анықтау үшін 3 қосымша жылжымалы (ығыстырушы) регистрді қолданыңыз.

-Мәліметтерді генерациялау үшін Function>>Programming>>Numeric палитрасында орналсқан Random Number (0-1) функциясын қолданыңыз.

-Сіз таңдап алған сандарды олардың орта мәндерімен біріктіру және оларды бір графикте тұрғызу үшін Function>>Programming>>Cluster&Variant палитрасында орналасқан Bundle функциясын қолданыңыз.

1. Температураны үздіксіз өлшейтін виртуалды құрылғы құрыңыз. Мәліметтерді кідірісі 1 секунд болатын scope chart режимінде Диаграмма графигіне шығарыңыз. Егер температура шекті мәннен асып кетсе, онда сәйкес басқару элементіне енгізілген беттік панельдегі қызыл жарықтық диод жану керек. Диаграмма графигі температураның ағымдағы және шекті мәндерін бейнелеу керек. Бейнелеу элементтері көмегімен температураның шекті мәндерінің өзгеру мүмкіндігін қарастыру қажет.
2. 2-жаттығудағы виртуалды құрылғыны былай өзгертіңіз. Ағымдағы өлшенген температураның тек максималды және минималды мәндерін ғана көрсететіндей етіп.

Кеңес. Function>>Programming>>Comparison палитрасында орналасқан ығыстырушы регистр және екі Max&Min функциясын қолданыңыз.

1. Интенсивтілік графигіне z(x,y)=sin c2 (x2+y2) функциясын құрыңыз. Меңзер көмегімен горизонталды сызықтар түрінде жеке графикте бейнеленетін графиктің қиысуын көрсетіңіз.
2. XY graph және берілген параметрлерге сәйкес келетін график осьтерін масштабтайтын y=ax2+b, y=ax2+bx+c, y=a\*exp(bx) функцияларының графиктерін бейнелейтін виртуалды құрылғы құрыңыз.

Кеңес. Графиктің масштабын беру үшін Property node функциясын қолданыңыз.

**7-лабораториялық жұмыс.**

**LabView ортасында айнымалы ток пен кернеуді өлшеу және модельдеу**

**7.1.Синусоидалық токты және кернеуді модельдеу**

Зертханалық жұмыста уақыттың синусоидалық функциясын көрсететін ток пен кернеу модельденеді

, (1)

Сонымен бірге лездік қуат:

(2)

және активті қуат (Р).

LabView ортасында модельдеуді әр түрлі әдістермен орындауға болады: синусоидалық тербеліс генераторын шақыруға немесе синусоидалық функция өрнегін формулалық түйінде, болмаса синусоидалы функцияны шақыруға болады. Бұл жұмыста синусоидалық заңдылық бойынша өзгеретін ток пен кернеуді берудің 3 әдісі қарастырылған. Кернеу, ток, қуат осциллографта бақыланады.

Айнымалы ток пен кернеу әрекет етуші мәндерімен сипатталады. Бізге белгілі әрекет етуші (ортаквадраттық) мәндер амплитудалық қатынаспен байланысты . Активті қуат (лездік қуаттың орта мәні) теңдеуімен анықталынады.

LabView жүйесінде әрекет етуші және өлшенетін шаманың орта мәнін анықтауда арнайы виртуалды құрылғылар қолданылады. Әрекет етуші мәндер (ортаквадраттық) өлшеу үшін **RMS** (**Root mean square**) құрылғысы қолданылады. Период ішіндегі орташа мән Mean құрылғысымен өлшенеді. Бұл құрылғылардың көрсеткіші цифрлық индикаторлар арқылы көрсетіледі.

**7.2.Синусоидалы сигнал генераторы**.

Синусоидалық тербелістің виртуалды генераторы көмегімен синусоидалық кернеуді модельдеу.

1.Осы мақсатта блог диаграммадан функция палитрасына кіріп **Functions => Analyse => Waveform Generation => Sine Waveform** операцияларын орындаймыз. Кернеудің бастапқы фазасын, жиілігін, амплитудасын беру үшін 3 цифрлық кернеу көздерін құрыңыз **(Controls => Numeric => Digital Control).**

2.Алынған қисықты бақылау беттік панельден шақырылатын виртуалды осциллограф бойынша бақыланады **(Controls => Graph => Waveform Graph).**

3.Генераторды іске қосу үшін генератор иконкасына тышқанды орналастырып, тышқан батырмасының оң жағын басып, оның терминалын іске қосамыз **Visible Items *=>*Terminals ,** содан кейін иконкаға қайта барып тышқанды басамыз және **Helpті** шақырып көздерді (источник) қосу нүктелерін анықтау керек. Блог диаграмманы жинап болған соң кернеудің амплитудасын, фазасын (градуста) және жиілігін (бастапқы мәні 10Гц, тербелістің бір периоды; есепте *f*= 2 қос жиілікті білдіреді, экранда екі период өрнектеледі) беру қажет.

4.Синусоидалық кернеуге әр түрлі амплитуда, фаза және жиілік бере отырып бірнеше осциллограмм тұрғызу керек.

**7.3. Тригонометриялық функцияларды пайдаланып екі синусоидалық шаманы модельдеу**

Бір период үшін қадам 1 градусқа тең, **For Loop циклін,** синусоидалық функцияны қолданып кернеу мен токтың синусоидалық мәндерін модельдеу үшін блог диаграмманы құрыңыз.

1.Блог диаграммада функция палитрасына кіріп, For циклін шақырыңыз. Ол үшін мына амалдарды орындаңыз: **Functions** =>**Structures => For Loop.**

2.Цикл ішіне біреуі кернеуді, ал біреуі токты көрсететін синусоидалық функция иконкасын енгізіңіз. Ол үшін мына амалдарды орындаңыз: **Functions => Numeric => Trigonometric => Sine**. Кіріске бұрыш мәні радианда беріледі, сондықтан градустық өлшемді радианға белгілі қатынасты қолданып өзгерту керек. Ол қатынас мынаған тең: .

3.Бір период ішіндегі есептеу саны біздің жағдайда *N=* 360-ға тең. Олай болса ағымдағы 1 цикл параметрі градус санына сәйкес келеді  және 0-ден 360-ға дейін өзгереді.

4.Бастапқы фазаға синусоиданың сәйкес параметріне қажетті градус саны қосылып беріледі.

5.Синусоидалық функцияның шығыс мәндері ток пен кернеудің амплитудалық мәніне көбейтіледі және виртуалды осциллографта бақыланады.

6.Зертханалық жұмыста бір уақытта екі синусоидалық функцияны виртуалды екі сәулелік осциллограф көмегімен бақылау басты мақсат. Бұл үшін беттік панельге осциллографты шақырамыз **Waveform Graph** және соған екінші график орнатылады. Ол үшін жылжыту құрылғысы көмегімен **Plot** терезесін вертикаль бағытта созуымыз қажет. Содан соң блог схемадағы иконка бетіне тышқан батырмасын алып барып, оң жағын басып массив құрастырғышты шақыру керек **Array Tools => Build Array.** Бұл бізге осциллограф кірісіндегі екі сигналды біріктіру үшін қажет. Пайда болған массив құрастырғышты вертикаль бағытта екі кіріс шыққанға дейін созамыз.

7.Программаны іске қосыңыз. Әр түрлі фазалық ығысу кезіндегі (оң және теріс) ток кернеу қисықтарын тұрғызыңыз.

Ескерту! Екісәулелік осциллограф ортақ координата шкаласына ие, сондықтан оны өлшеніп жатқан шаманың амплитудалық мәні абсолюттік шамадағы мәнімен шамалас болған кезде қолданған дұрыс.

**7.4.Формулалық түйінді қолданып екі синусоидалық шаманы модельдеу**

Формулалық түйінді және **For Loop** циклін қолданып синусоидалық ток тізбегіндегі ток, кернеу, қуатты модельдейтін блог диаграмма құру керек.

1.Ол үшін блог диаграммада **Foor Loop** циклін құрып, оның ішіне формулалық түйінді орналастырыңыз **Functions** =>**Structures => Formula Node**. Формулалық түйін ішіне мына формуланы жазыңыз:  , мұнда  аргументі радианда былай жазылады:  . Мұнда операция есептегіш параметрін ток белгісінен ажырату мақсатында **әрпімен белгілеген.

Ескерту! Итерация есептегішін алмастыру үшін «тексті енгізу» инструментін (курсор А) қолдану жеткілікті. I әріпінің орнына  әрпін енгізесіз.

**параметрі  градус санына сәйкес келеді. Біздің жағдайда есептеуді бір және екі период бойында (аралығында) жүргізген жөн. Сондықтан санау саны N = 360 немесе 720.

2.Формулалық түйін рамкасына тышқан курсорын алып келіп, оң жақты басып қажетті кіріс және шығыс мәндерін енгізіңіз **(Add Input, Add Output).**

3. Есептеліп алынған лездік қуат, кернеу, ток мәндерін осциллографқа енгізу керек .

4. Синусоидалы кернеу, токтың әсер етуші (ортаквадраттық) мәндері **RMS** (**Root Mean Square)** виртуалды құрылғысы көмегімен өлшенеді. Бұл құрылғыны блог диаграммада шақыру мына амалдар көмегімен іске асады: **Functions => Analyze => Mathematics => Probability and Statistics** =>**RMS.** Шығыстарына цифрлық индикаторлар қосылады. Активті қуат болса**Functions => Analyze => Mathematics => Probability and Statistics => Mean** виртуалды құрылғысы көмегімен өлшенеді. Бұған да шығысына индикаторларды жалғайсыздар.

5.Кернеу мен ток үшін әр түрлі фаза ығысуы кезіндегі кернеу, ток, лездік қуат қиысығын тұрғызу керек. Шыққан нәтижені есептеу кезінде алынған нәтижемен (1,2 формула) салыстыруға болады.

***7.5.Резонансты модельдеу***

Көптеген зерттеулерде берілген шаманы ғана өлшеп қоймай сонымен бірге қандай да бір шекте немесе аралықта қандай да бір параметрді өзгерткен кезде оның сипаттамасын тұрғызу міндетті.

Индуктивтілігі L, сыйымдылығы С және кедергісі R болатын жүйені тізбектей жалғау қарастырылады. Тізбек электр қозғаушы күші (ЭҚК)Е, айнымалы жиілігі  болатын қорек көзіне қосылған. Тізбектегі ток, катушка мен конденсатордағы кернеудің тұрақты кернеу  кезіндегі жиіліктен тәуелділігін тұрғызу керек.

1.Ток пен кернеу Ом заңы көмегімен анықталынады:

*; ; ,* (3)

мұнда *.* (4)

2.*U*= 100 В, *L*= 0,025 Гн, *С* = 0,004 Ф, *R*= 1 Ом осындай сандық мәндер берілген болсын.

3.Есепті шешуде **For Loop** циклінде жұмыс жасайтын формулалық түйін қолданылады. Формулалық түйін ішіне (3,4 формула) жазылады.

4.Өзгеріп отыратын жиілік ретінде операция есептегішін *i*  қолдануға болады. 3, 4 формулаға назар аударып қарасақ, жиілік бөлшек бөлімінде орналасқан , бөлімі нөл болмас үшін деп аламыз.

5.Индикаторлар ретінде үшсәулелік осциллограф алуға кеңес береміз. Бұл осциллограф жоғарыда айтып кеткен екісәулелік осциллогра сияқты қосылады. Операция саны ток пен кернеудің резонанстық пиктері байқалатындай таңдалынып алынады. Біздің жағдайда N=200.

6. Осциллографтың бір терезесінде үш резонастық графикті  салыңыз.

**7-тапсырма.**

8-бөлім.

**Файлмен жұмыс істеу функциялары. Кестелер**

Жолдар реттілікті біріктіреді және ASCII символдардың массиві болып табылады. ВҚның ішкі бағдарламасы ішкі файлдармен жұмыс жасайды. Файлдан кіріс/шығыс мәліметтерді алу, оны енгізуді қамтамасыз етеді.

**8.1.Жолдар**

Жолдар бұл бейнеленетін және бейнеленбейтін ASCII символдардың реттілігі. Жолдар платформа форматынан тәуелсіз мәліметтер алмасуды қамтамасыз етеді. Кейбір кең таралған жолдық қосымша өзіне төмендегідей амалдарды қосады:

* Қарапайым мәтіндік хабарламаны құру.
* Жолдық символдар түрінде құрылғыларға сандық мәліметтерді жіберу және жолдарды сандық мәліметтерге түрлендіру.
* Сандық мәліметтерді дискке сақтау. Сандық мәліметтерді ASCII файлы түрінде сақтау үшін жазу процесінен бұрын оларды жолдарға айналдыру керек.
* Нұсқаулық және еске түсіру (ойға салу) диалогтік терезесі. Беттік панельде жолдар кесте, кесте енгізу жолы және белгі түрінде пайда болады.

8.1.1**Мәліметтерді басқару және бейнелеу элементтері бойынша жолдар құру**

Мәтін және белгімен жұмыс жасау үшін мәліметтерді басқару және бейнелейтін жолдық элементтер қолданылады. Олар **Controls>> Modern>>String&Path** палитрасында орналасқан. Жолдарда мәтінді құру және оны өңдеу Басқару және Мәтінді енгізу элементі көмегімен жүзеге асады. Жолдық объектінің өлшемін өзгерту үшін беттік панельде Орын алмастыру жабдығы қолданылады. Беттік панельде орынды үнемдеу үшін айналдыру жолағын қолдануға болады. Ол үшін жолдық объектіге меңзерді апарып, тышқан батырмасының оң жағын басып, контексті мәзірден **Visible Items>> Scrollbar** пунктін таңдау қажет.

Жолдық объектінің бейнелену типі өзінің контексті мәзірінде таңдалынып алынады. Жолдарды бейнелеу типтері және мәтінді енгізу жолын қалай толтыру керектігі төмендегі кестеде көрсетілген:

10-кесте. Жолдарды бейнелеу типтері

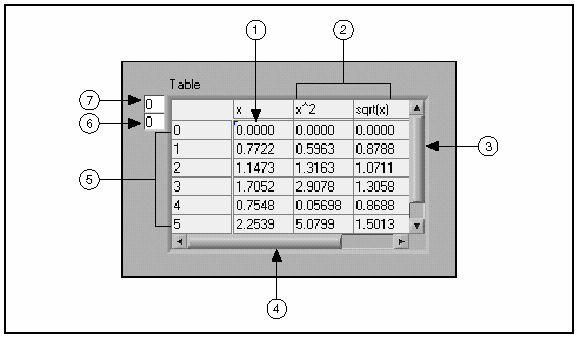
|  |  |  |
| --- | --- | --- |
| **Бейнелену типі** | **Сипаттама** | **Мәтін мысалы** |
| Стандарттық бейнелену режимі (**NormalDisplay**) | Басқару элементі шрифтін қолдана отырып, стандартты ASCII кодтарды бейнелейді. Басып шығару үшін қолданылатын басқару кодтары экранға квадрат түрінде шығады. | There are four display types.  \ is a backslash |
| Басып шығарылмайтын басқару кодтары (**‘\’ CodesDisplay**) бар кері слэшті бейнелену режимі | Барлық басып шығарылмайтын басқару кодтарын (\) шығарады. | There\sare\sfour\s display\stypes.\n\  \\sis\sa\sbackslas h |
| Мәтіндіжасырынбейнелеурежимі (**PasswordDisplay**) | Мәтіндіккеңістіктіңбарлықкодтарыүшін (\*) шығарады. | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| 16-лық ASCII  Кодтарды бейнелеу режимі (**Hex Display**) | Әрбір символ үшін ASCII кодтардың мәндерін шығарады. | 5468 6572 6520  6172 6520 666F  7572 2064 6973  706С 6179 2074  7970 6573 2Е0А  5С20 6973 2061 |

8.2.Кестелер

Кесте **Controls»Modern»List&Table** палитрасында орналасқан басқару элементі. Әрбір ұяшық кестенің жолында және бағанында орналасады. Сондықтан кесте жолдық екіөлшемді массивті бейнелейді. Төменде кесте және оның негізгі бөліктері көрсетілген (94-сурет):

Кесте ұяшықтарына мәндерді жүктеу үшін Басқару немесе Мәтінді енгізу жабдығы қолданылады. Бұл жабдықтар көмегімен белгіленген ұяшықтарға мәтінді енгізе аласыз.

Кесте - бұл жолдық екіөлшемді массив. Сондықтан да мәліметтерді бейнелеу элементі ретінде сандардан құралған массивтерді жолдық екіөлшемді массивке айналдыру керек. Кестенің жолдар және бағандар атауы, символдар кестесіндегідей автоматты түрде жолдар мен бағандар атауы бар түрде бейнеленбейді.



1-Кесте ұяшығы; 2-Бағанның атауы; 3-Вертикальді айналдырудың жолағы; 4-Горизоньтальді айналдыру жолағы;

5-Жолдың атауы; 6-Горизонталь бойынша индекс;

7-вертикаль бойынша индекс

94-сурет. Кестенің LabVieW ортасындағы көрінісі

**8.3. Жолдармен жұмыс жасайтын функциялар**

### Блог-диаграммада жолдарды өңдеу және басқару үшін жолдарды өңдеу функциясын қолданған ыңғайлы. Ол функциялар Functions>>Programming>>String палитрасында орналасқан. Жолдармен жұмыс жасайтын кейбір функциялар төменде қарастырылған:

**String Length** – бос орынды қосқан кездегі жолдағы символдар санын береді. Мысалы, **String Length** функциясы мына мәтін үшін 19 мәнін шығарады:*The quick brown fox.*Айта кету керек бос орын да мәтін ретінде есептелінеді.

**Concatenate String** – жолдарды және бірөлшемді жолдық массивтерді жеке жолдарға біріктіреді. Функцияның мәліметтерді енгізу жолдарын арттыру үшін оның өлшемін өзгерту керек. Мысалы, алдыңғы жолды төмендегі жолдық массивке біріктіріп

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| jumped | over | the | lazy | dog |

**Concatenate Strings** функциясы шығысында мынадай жолды шығарады:The quick brown fox jumped over the lazy dog.

**String Subset** –қандай да бір анықталған **length**ұзындығы бар**offset**(орын ауыстыру) мәнінен басталатын ішкі жолдарды береді. Жолдағы бірінші элементтің орын ауыстыруы 0-ге тең. Мысалы, функцияның мәліметтерді енгізу жолына жоғарыдағы жолды берсек, онда **String Subset** функциясы **offset=4**-ке, **length =5-**ке тең болған кезде *quick* мәнін береді.

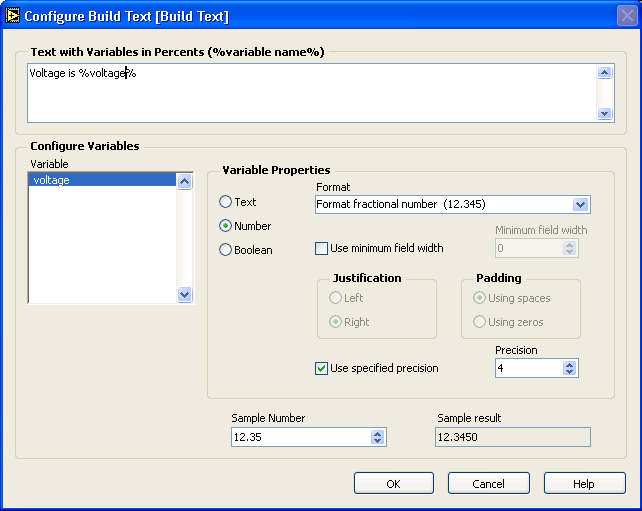
**Match Pattern** – **offset** мәнінен бастап **regular expression** мәліметтерді енгізу жолына берілген қайталанатын реттіліктерді іздейді. Егерде сәйкестікті тауып жатса, онда жолды 3 ішкі жолға бөледі. Егер сәйкестік табылмаса, онда **match substring** мәліметтерді шығару жолы бос болып тұрады. Ал **offset past match** (жолдағы қайталанатын реттіліктердің орын алмасуы) мәліметтерді шығару жолының мәні 1-ге тең болады. Мысалы, **regular expression** (ішкі жол шаблоны) жолына **:** (қос нүкте) мәні беріледі, ал VOLTS DC кірісіндегі жол **VOLTS DC:+1.22863E+1**мәніне тең болады. **Match Pattern** функциясы **before substring** ( ішкі жол алдында) үшін VOLTS DC мәнін, match substring (ішкі жолдар шаблоны) : (қос нүкте) белгісін, after substring (ішкі жолдан кейін) *+1.22863E+1*, ал **offset past match** 9 мәнін шығарады.

**8.3.1.Cандық мәліметтерді жолға айналдыру**

Сандық мәліметтерді жолға түрлендіру үшін **Format Into String** (жолға айналдыру) функциясы және **Build Text Express** виртуалды құрылғысы қолданылады. Бұл екі функцияда кіріс/шығыс қателік кластеріне ие.**Format Into String** функциясын блог-диаграммада орын жетпей қалған кезде қолданған тиімді.

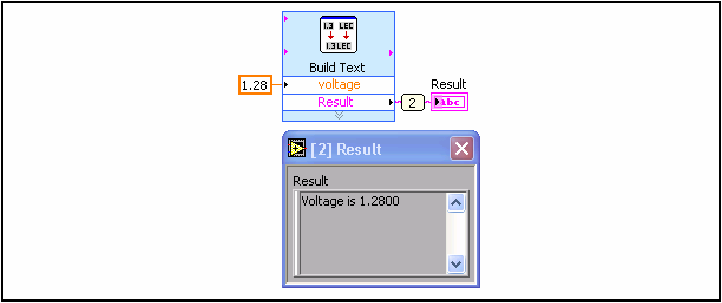
**8.3.2.Build Text Express VIэкспресс виртуалды құрылғысы**

**Build Text** экспресс виртуалды құрылғысы **Function>> Express>>Output** палитрасында орналасқан. Кіріс жолдарды біріктіру қызметін атқарады. Егер кіріс мәндер жолдық емес мәлімет типіне ие болса, онда олар жолдарға айналдырылады.

**Build Text** экспресс виртуалды құрылғысын блог-диаграммада орналастыру кезінде **Configure Build Text** орнатуды жүзеге асыруды орындайтын диалогтік терезесі ашылады. Келесі мысалда кернеу мәні экспресс виртуалды құрылғының кірісіне беріледі және үтірден кейін 4 саны бар жылжымалы үтірлі мәлімет форматына ауысады. Содан кейін бұл мән **Voltage is**(кернеу тең) жолының соңына қосылады (95-сурет).

95-сурет. **Configure Build Text** көрінісі

Мұндай орнату кезінде экспресс виртуалды құрылғы келесідей түрде болады. Шығыс жолды бақылау үшін ретке келтіруші индикатор қолданылады. **Beginning Text** кіріс жолына берілген кез келген мән орнату диалогтік терезесіндегі мәтіннің басына келіп қосылады.



96-сурет. Build Тext функциясын қолданып нәтижені шығару көрінісі

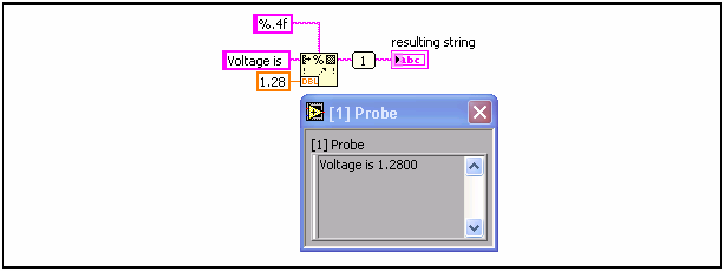
Жолдарды біріктіру құбылысы конкатенация деп аталады.

**8.3.3.Format Into String функциясы**

**Format Into String** функциясы кез келген форматтағы параметрлерді жолдарға түрлендіреді. Параметрлер санын арттыру үшін функцияның өлшемін өзгерту керек.

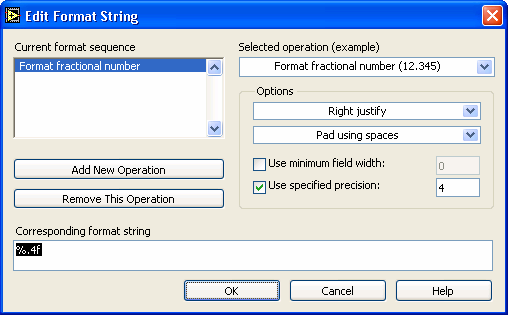
Төмендегі келтірілген мысалда **Format Into String** функциясы индикаторға 97-суретте көрсетілгендей мәндерді format string жолына (жол форматында) %4f мәнін, input string (кіріс жолында) жолына Voltage is (соңында бос орынды ескергенде) мәнін және параметрге 1,28 мәнінбергенде шығарады. Мұндағы

% жол форматында жол форматының басы көрсетіледі, .-нүктеден кейінгі сан санның дәлдігін анықтайды, 4-үтірден кейінгі мәндер санын көрсетеді, ал f- жылжымалы үтірлі мәлімет типін көрсетеді. Жолдар форматын өңдеу және құру үшін функцияға меңзерді апарып, тышқан батырмасының оң жағын басып контексті мәзірден **Edit Format String**пунктін таңдау қажет.



97-сурет. Format Into String функциясының көрінісі

98-сурет алдыңғы мысалдағы **Edit Format String** диалогтік терезесінің көрінісін көрсетеді.



98-сурет. Edit Format String диалогтік терезесінің көрінісі

Форматтар синтаксисі жайында көптеп ақпарат алу үшін анықтамалық орталықты қолдануға болады.

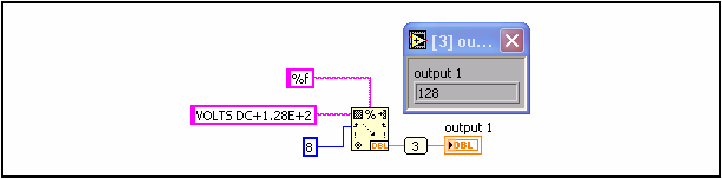
**8.3.4.Жолдарды сандарға айналдыру**

Жолдарды сандарға айналдыру үшін **Scan From String** функциясын қолданған ыңғайлы.

**Жолдарды түрлендіру және көру**

**Scan From String**функциясы мынадай формадағы 0-9, +, -, е, Е сандық символдарды жолдарға түрлендіреді. Функция жолдарды қарауды **initial search location**жолына қойылған **input string** мәліметтерді енгізу жолына енгізілген символ номерінен бастап бастайды. Функция әр түрлі мәлімет типіндегі кіріс жолды жол форматына негіздей отырып қарастырады. Мәліметтерді шығару жолдарының санын арттыру үшін функцияның өлшемін өзгерту керек.

Мысалы, мәліметтер жолына мына мәндерді берген кезде: Format string - %f, initial search location – 8, input string –VOLTS DC+1.28E+2берген кезде функция 99-суретте көрсетілгендей 128,00 мәнін шығарады.



99-сурет. Scan From Stringфункциясының көрінісі

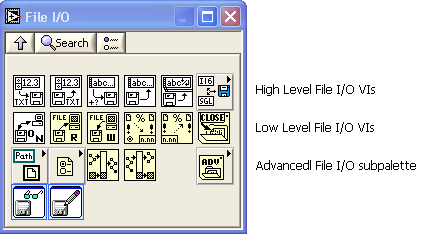
% жол форматы – жол форматының басын, f- жылжымалы үтірлі мәлімет типін көрсетеді.

**8.4. Кіріс/шығыс файлдық функциялар**

Кіріс/шығыс файлдық функциялар мәліметтерді оқу және жазу сияқты амалдарды орындайды. Кіріс/шығыс файлдық функциялар **Functions>>Programming>>File I/O** палитрасында орналасқан. Мынадай амалдарды орындауға арналған:

* Мәліметтер файлдарын ашу және жабу.
* Файлдардан және файлға мәліметтерді жазу және оқу.
* Символдар кестесі түрінде мәліметтерді файлға және файлдан жазу және оқу.
* Каталогтардың және файлдардың атауын өзгерту және орындарын алмастыру.
* Файл сипаттамаларын өзгерту.
* Файлдар конфигурациясын оқу, өзгерту және құру.

**Кіріс/шығыс файлдық функциялар**

Төменде көрсетілгендей, кіріс/шығыс файлдық функциясы 3 бөлікке бөлінген (100-сурет): жоғары деңгейлі функциялар (high level File I/O), төменгі деңгейлі функциялар (Low level File I/O)және кеңейтілген мүмкіндіктегі палитра ішіндегі функция (advanced File I/O) .

100-сурет. Кіріс/шығыс файлдық функциялар бейнесі

**8.4.1.Жоғары деңгейдегі кіріс/шығыс файлдық функциялар**

Жоғары деңгейдегі кіріс/шығыс файлдық функциясы **Functions>>Programming>>File I/O** палитрасының жоғарғы жолында орналасқан. Олар мәліметтерді енгізу/шығару бойынша негізгі амалдарды орындауға арналған. Жоғары деңгейдегі кіріс/шығыс файлдық функцияларын қолдану файлдардан мәліметтерді оқу, жазу амалдарын орындауды қысқартуға арналған. Жоғары деңгейдегі кіріс/шығыс файлдық функциялары файлдарды ашу, жабу сияқты амалдарды да орындайды. Қателік пайда болған кезде жоғары деңгейдегі кіріс/шығыс файлдық функциясы қателіктер сипаттамасын көрсететін диалогтік терезені шығарады. Содан соң программаны жалғастыру қажет пе алде тоқтату керек пе деген сұрақтарды қояды.

**8.4.2.Төменгі деңгейдегі кіріс/шығыс файлдық функциясы**

Төмен деңгейдегі кіріс/шығыс файлдық функциясы **Functions>>Programming>>File I/O** палитрасының орта жолында орналасқан.

Төменгі деңгейдегі кіріс/шығыс файлдық функциясы бұдан бұрын құрылып қойылған файлдарға жүгіну немесе жаңа файл құру амалдарын орындайды.

**Кіріс/шығыс файлдың негізі**

Файлдан және файлға мәліметтерді енгізу/шығаруды жүзеге асыратын стандартты амалдар келесідей реттілікте қызмет етеді:

1. Файлды құру немесе ашу. Файлдың орналасу орнын көрсету немесе жаңа файлды құру жолдарын LabVIEWдағы диалогтік терезе көмегімен көру.
2. Файлдан және файлға мәліметтерді жазу, оқу амалын жүргізу.
3. Файлды жабу.
4. Қателікті өңдеу.

Кіріс/шығыс файлдық негізгі амалдарды орындау үшін келесідей виртуалды құрылғы және функция қолданылады:

**Open/Create/Replace File** – алдында құрылған файлды ашады, қайта оқиды немесе жаңа файл құрады. Егер File path (файлды орналастыру жолы) көрсетілмесе, онда виртуалды құрылғы жаңа файл ашуға болатын немесе бұрын құрылған файлды ашуға болатын жағдайларды қарастыратын диалогтік терезені экранға шығарады.

**Read File**–**refnum** сілтемесі бойынша файлдан мәліметтерді оқиды, data жолына мәліметтерді береді. Ал count жолына оқылған мәліметтердің саны беріледі. Мәліметтерді оқу файл форматына тәуелді.

**Write File**– **refnum** сілтемесі бойынша мәліметтерді файлға жазады.

**Сlose File** - **refnum** сілтемесі бойынша көрсетілген файлды жабады.

**Қателікті өңдеу**

Виртуалды құрылғының ішкі программалары және төменгі деңгейдегі функциялар қателік жөнінде ақпаратқа ие болады. Қателікті өңдеу үшін **Simple Error Handler VI** (қарапайым қателік өңдегіш) қолданылады. Ол **Functions>>Programming>> Dialog & User Interface**палитрасында орналасқан. **Error in** кірісі мен **error out** шығысына қателік жайындағы ақпараттар жазылады. LabVIEW ортасында виртуалды құрылғы жұмыс жасап жатқан уақытта қателіктің болу мүмкіндігін әрбір түйін үшін тексеріп шығады. Егер LabVIEW қателік таппаса, онда түйін дұрыс жұмыс жасайды. Ал қателік анықталынса, онда виртуалды құрылғы жұмыс жасауын тоқтатады. Ал қателік жөніндегі ақпарат келесі түйінге жіберіледі. Келесі түйін де дәл солай жұмыс жасайды.

**Өтілген материалдарға қысқаша шолу, кеңестер, құпиялар**

* Жолдар **ASCII** кодтар тізбектілігі болып табылады. Мәтінді енгізу терезесін және белгішені құру үшін мәліметтерді бейнелеу және басқару элементтерінен тұратын жолдық элементтер қолданылады. Олар **Controls>>Modern>>String&Path** палитрасында орналасқан.
* Жолдық объектілерімен толтырылған кеңістікті айналдырушы жолақ көмегімен минимизациялауға болады. Ол үшін объекті үстінен тышқан батырмасының оң жағын басып, контексті мәзірден **Visible Items>>Scrollbar** пунктін таңдау керек.
* Блог-диаграммада жолдарды өңдеу және басқару үшін **Functions>>Programming>>String** палитрасында орналасқан жолдарды өңдеу функциялары қолданылады.
* Сандық мәліметтерді жолға айналдыру үшін **Scan From String** функциясы қолданылады.
* Жолдар форматын құру және өңдеу үшін **Format Into String**немесе **Scan From String** функцияларына тышқан батырмасын апарып, оң жағын басу арқылы ь контексті мәзірден **Edit Format String** немесе **Edit Scan String** пунктін таңдау қажет.
* Файлдармен, файлдарға амалдарды орындау үшін **Functions>>Programming>>File I/O** палитрасында орналасқан кіріс/шығыс файлдық функциялар қолданылады.
* **Functions>>Programming>>File I/O** палитрасының жоғары жолында орналасқан жоғары деңгейлі кіріс/шығыс файлдық функциялар кіріс/шығыстық файлдық мәліметтерге стандартты әрекеттерді орындауға арналған.
* **Functions>>Programming>>File I/O** палитрасының орта жолында орналасқан төменгі деңгейлі кіріс/шығыс файлдық функциялар кіріс/шығыстық файлдарды басқаруды жеке амалдармен жүргізу үшін қажетті.
* Файлға мәліметтерді жазу келесідей амалдар арқылы жүреді: файлды құру немесе қайта жазу, мәліметтерді жазу және файлды жабу. Файлдан мәліметтерді оқу кезінде құрылып қойылған файлды ашу, мәліметтерді оқу және файлдарды жабу амалдары орындалады.
* Символдар кестесін файлға жазу үшін жолды символдар кестесі жолына форматтау керек. Жолдарды, сандық, логикалық мәліметтерді және жолдарды мәтіндік файлға форматтау үшін Format Into File функциясы қолданылады.

**Қосымша жаттығулар**

8-1. Мәліметтерді символдар кестесіне орналастыратын және кесте мәлісметтерін файлға жазатын кездейсоқ сандардан құралған 100 бағаннан және 3 жолдан тұратын 2D өлшемді массивті генерациялайтын виртуалды құрылғы құрыңыз. Әрбір бағанға атауын беріңіз. Functions>>Programing>> File I/O палитрасының жоғары жолында орналасқан жоғарғы деңгейлі кіріс/шығыс файлдық функциясын қолданыңыз.

**Кеңес.** Файлға баған атауын және мәліметтерді жазу кезінде Write to Text File VI Write To Spreadsheet File VI виртуалды құрылғының ішкі программасын қолданған жөн.

8-2. Табуляция позициясымен шектелген символдар кестесінің жолдарын ажыратқыштар- үтірі бар жолдарға айналдыратын виртуалды құрылғы құрыңыз. Бағандар үтір белгісімен, ао жолдар end of line символымен ажыратылған болу керек. Символдар кестесінің жолдары, шектелген табуляция позициялары және ажыратқыш-үтірлі жолдарды беттік панельде бейнелеп қою міндетті.

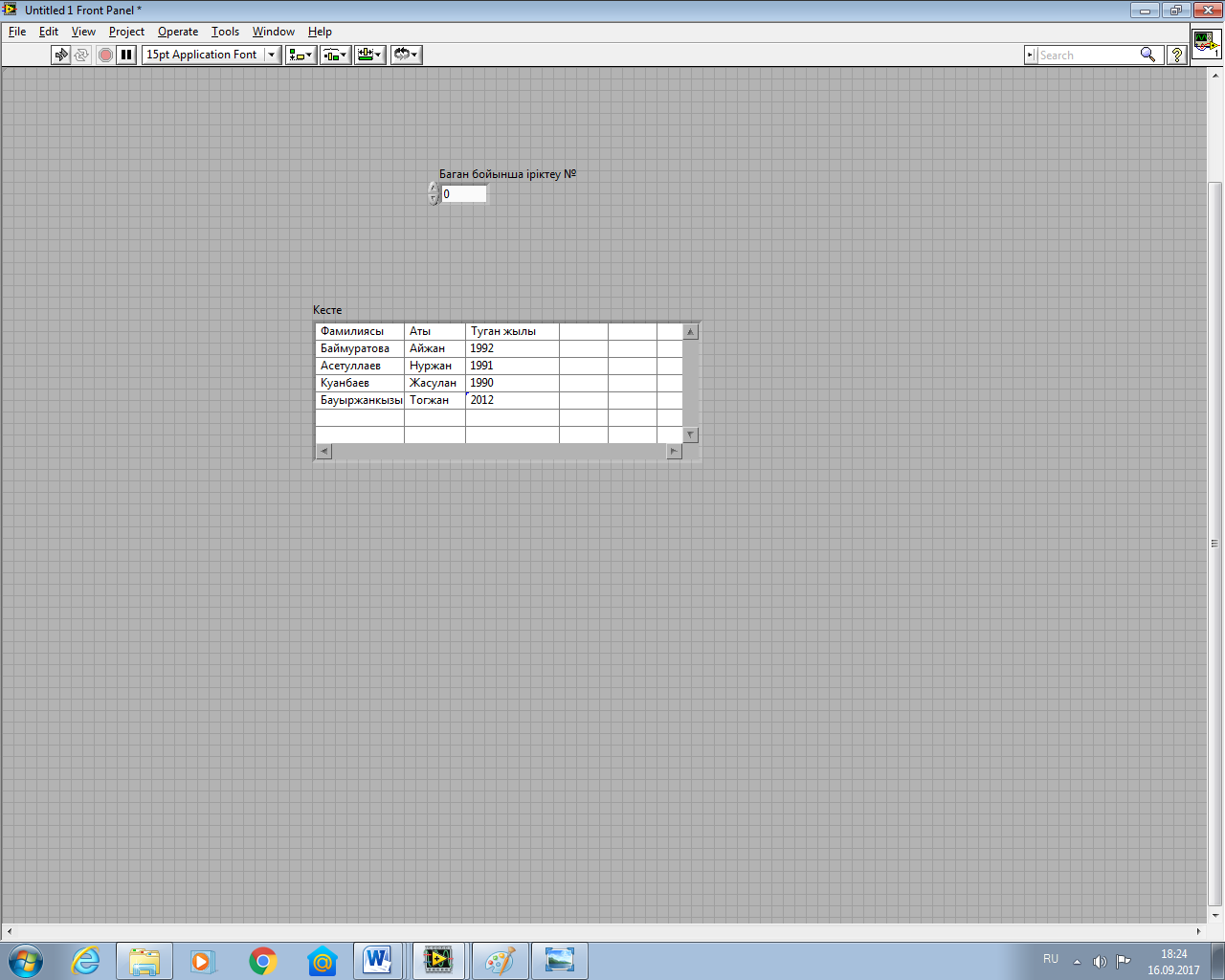
**Кеңес.**Search and Replace string функциясын қолдануға кеңес беріледі.

8-3. Файлдан сандық мәліметтерді әрбір символы бойынша оқитын және оларды Double типіне түрлендіретін (бос орынды есептеу керек) виртуалды құрылғы құрыңыз.

**8-лабораториялық жұмыс. Кестені іріктеу**

Виртуалды құрылғыдағы кестенің әр түрлі бағандарын өсу реті бойынша іріктеуге мүмкіндік болу үшін қажетті кестені өңдеңіз. Іріктеудің өзін **Functions** =>**Array** палитрасында орналасқан **Sort 1D Array** функциясы көмегімен орындаңыз.Бұл элемент бірөлшемді массивтің элементтерін өсу реті бойынша іріктейді. Егер массив кластерлерден тұрса, онда функция массивті кластердің бірінші элементі бойынша іріктейді. Осыған байланысты массивті тұтынушыны қызықтыратын бағанның бірінші тұруы сияқты негізде түрлендіруге болады. Содан кейін алынған массивті кластерге біріктіру керек. Біріктіру процесі іріктеу процесін жүргізуге қажетті.

1. Кестенің өзін және іріктеуге қажетті бағанды көрсетуге қолданылатын бүтін сандық басқару элементін құрыңыз. Кестені қандай да бір ақпаратпен толтырыңыз. 101-суретте көрсетілген мәліметтерді алсаңыз болады.



101-сурет. Кесте

1. Әдепкі қалыпта баған мен жолдың атауы көрінбейді. Бағандар атауы көрініп тұру үшін кестелер қасиеттерінен (қасиеттер деп аталатын диалогтік терезе контексті мәзірден Properties пунктін таңдау арқылы шақырылады) **Show Column Headers** пунктіне жалаушаны қарама-қарсы бағытта қойыңыз. Сонымен қоса атауды өңдеуге мүмкіндік болу үшін кестенің контексті мәзірінен **Editable Headers** опциясын қосыңыз.
2. Іріктеуге қажетті бағанның жылжуы **Functions => Array** палитрасындағы **Rotate 1D Array** пункті көмегімен іске асады. **Rotate 1D Array** функциясының кірісіне берілетін бірөлшемді массив кестенің жолдарын береді. Әрбір жолды **For** циклі көмегімен аласыз. Егер әрбір жолдар үшін элементтердің орындарын жеке-жеке алмастырсақ, онда кестедегі бағандардың барлығы да орындарын алмастырады. Мысалы, 98-суреттегі кестені 3-ші бағаны бойынша іріктейік делік (есептеу 0-ден басталады). Былай айтқанда әкесінің аты жазылған бағанды бірінші орынға орналастыру керек. Егер **n > 0** болса **Rotate 1D Array** функциясы **n** элементтерді басынан аяғына дейін ауыстырады.Ал **n < 0** болса, онда керісінше n элементтерді аяғынан басына дейін ауыстырады. Егер **n** кірісіне **0** мәні берілсе, онда функция жолды өзгертпейді. Кестелер бағандарын іріктеуге дейін **2 бағанға** сол жақ бөліке, іріктеуден кейін **2** бағанға оң жақ бөлікке жылжыту керек. Сондықтан да іріктеу функциясын қолданбай тұрып **Rotate 1D Array** функциясының **n** кірісіне **0**- мәнін**,** содан кейін **+2** мәнін беру қажет.
3. Массивті кластерге және кластерді массивке түрлендіру сәйкесінше **Array To Cluster** және **Cluster To Array** функциялары көмегімен жүзеге асады. Кластерді массивке түрлендіру кезінде ешқандай да қосымша амалдарды орындау талап етілмейді. Ал массивті кластерге түрлендіру кезінде кластердің өлшемін көрсету талап етіледі. Кестенің құрылымы алдын-ала белгілі деп ойлайық. Бұл функцияның контексті мәзірінен **cluster size** пунктін таңдау арқылы жасалынады. Біздің жағдайда кластер өлшемі 4-ке тең.
4. Сонымен, сіздер кестені жеке жолдарға бөлдіңіздер. For циклдағы әрбір жолда бағандар орынын алмастырдыңыздар. Жолдар элементтерінен кластер құрдыңыздар. **For**циклының шығысында кластерден массив құрастырдыңыздар. Кластердегі бірінші элемент бойынша кластерлер массивін іріктедіңіздер. Кластер массивінің әрбір элементінен бірөлшемді массив алу үшін тағы да **For** циклін қолдандыңыздар. Қайтадан бағандар орынын алмастырдыңыздар. **For** циклының шығысында іріктелген жолдарды кестеге жинадыңыздар. Енді сол нәтижені сол кестеге шығару қана қалады. Іріктелген кестенің шығыс кестесінде бейнеленуі міндетті. Мұндай жағдай үшін локальді айнымалыны қолданған дұрыс. Кез келген басқару және бейнелеу элементтері үшін локальді айнымалыларды элементтің контексті мәзіріне кіріп, одан**Create => Local Variable** пунктін таңдау арқылы қолдануға болады. Контексті мәзірдің бұл пункті беттік панель және блог-диаграмма үшін де қолжетімді. Локальді айнымалыларды орналастырып болған соң оның терминалдары басқару және бейнелеу элементтері терминалдары жанынан пайда болады.
5. Локальді айнымалыны блог-диаграмманың кез келген жерінде қолдануға болады. Оны мәліметтерді оқу және жазу үшін де қолдануға болады. Соңғысы ретінде локальді айнымалыларды іріктелген кестемен байланыстырыңыз. ВҚ дайын.
6. Беттік панельге өтіп, программаны сынап көріңіз.
7. Тұтынушы баған нөмерін емес, оның атауын таңдайтын жағдайды ойланып көріңіз.
8. Бағандарды іріктеуді тек өсу реті ғана емес, сонымен қоса кему реті арқылы да іріктейтін мүмкіндікті енгізіңіз. Бұл жағдайда басқару виртуалды құрылғының беттік панелінен орындалуы керек.