

## Дәріс 10 . File I/O: Файлдармен жұмыс істеу

**Дәріс мақсаты:** Студенттерді LabVIEW ортасында файлмен жұмыс істеудің негізгі функцияларымен таныстыру. Файлға мәлімет жазу, файлдан оқу, файл құрылымын басқару және қателерді өңдеу тәсілдерін меңгерту. Практикалық мысалдар арқылы мәліметтерді сақтау, жүктеу және құрылымдық түрде өңдеу дағдыларын дамыту.

### Кіріспе

Көптеген инженерлік және ғылыми қосымшаларда **мәліметтерді сақтау** — міндетті процесс. Сенсорлардан алынған деректер, пайдаланушының енгізген ақпараттары немесе өңделген нәтижелер — бұлардың барлығы **файл арқылы** сақталып, кейін талдауға қолданылады. LabVIEW ортасында файлдармен жұмыс істеуге арналған **File I/O (Input/Output)** палитрасы бар. Ол арқылы біз **.txt**, **.csv**, **.dat** сияқты файл форматтарымен оңай әрекеттесе аламыз.

### Файл енгізу/шығару (File I/O)

Файл енгізу/шығару операциялары деректерді файлдарға және файлдардан береді. Файл енгізу/шығару VI файлдарын және **файлдарды енгізу/шығару** палитрасындағы функцияларды, төмендегілерді қоса, енгізу/шығару файлының барлық аспектілерін өңдеу үшін пайдаланыңыз:

Деректер файлдарын ашу және жабу. Деректерді файлдардан оқу және деректерді жазу. Электрондық кесте пішіміндегі файлдардан оқу және оларға жазу. Файлдар мен каталогтарды жылжыту және атын өзгерту. Файл сипаттамаларын өзгерту. Конфигурация файлын жасау, өзгерту және оқу.

Бір VI немесе функция арқылы файлды ашуға, оқуға немесе жазуға және жабуға болады. Сондай-ақ, процестегі әрбір қадамды басқару үшін функцияны пайдалануға болады.

Әдеттегі файлды енгізу/шығару операциясы келесі процесті қамтиды.

Файлды жасаңыз немесе ашыңыз. Файл ашылғаннан кейін, `refnum` деп аталатын бірегей идентификатор файлды білдіреді.

Файлдан оқу немесе файлға жазу үшін Файл енгізу/шығару палитрасынан VI немесе функцияны пайдаланыңыз. Файлды жабыңыз.

Файл енгізу/шығару VI және кейбір файлды енгізу/шығару функциялары, мысалы, мәтіндік файлдан оқу және мәтіндік файлға жазу функциялары әдеттегі файл енгізу/шығару әрекеттері үшін барлық үш қадамды орындай алады. Бірнеше операцияларға арналған VI және функциялар жеке әрекеттерді орындайтын функциялар сияқты тиімді болмауы мүмкін.

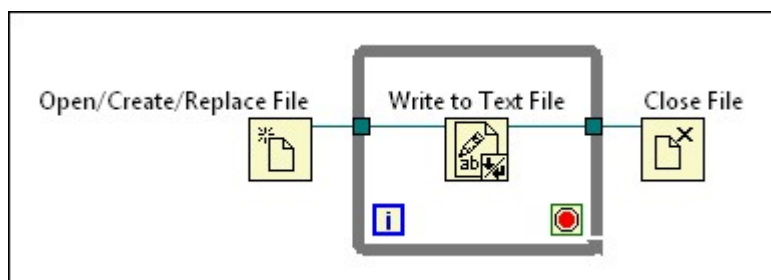
### **Disk Streaming көмегімен жадты сақтау**

Функцияның файлды ашу және жабу үшін амалдық жүйемен әрекеттесу санын азайту арқылы жад ресурстарын сақтау үшін диск ағынын пайдаланыңыз. Диск ағыны цикл ішінде бірнеше оқу немесе жазу әрекеттерін орындаған кезде файлдарды ашық ұстауды білдіреді.

Мәтіндік файлға жазу функциясы немесе Екілік файлдан оқу функциясы сияқты файлдан оқитын немесе оған жазатын функцияға немесе VI үшін жолды басқару элементін немесе тұрақты мәнді қосудан аулақ болыңыз, өйткені бұл функция немесе VI орындаған сайын файлды ашу және жабудың үстеме шығындарын қосады. Оның орнына, бұл үстеме шығындарды жою үшін диск ағыны әрекеттерін орындаңыз.

Диск ағыны әрекетінің қолданбаңызға сәйкес келетінін анықтау кезінде келесі ақпаратты қарастырыңыз: Диск ағыны жылдамдық өте маңызды болатын ұзақ деректерді жинау операцияларында өте қолайлы. Сатып алу әлі жүріп жатқанда, деректерді файлға үздіксіз жазу үшін дискілік ағынды пайдаланыңыз.

Дискінің әдеттегі ағындық әрекетін жасау үшін Файлды ашу/Жасау/Ауыстыру функциясын және Файлды жабу функциясын келесі 1 сурттегідей блок-схемадағыдай циклден тыс орналастырыңыз.



Сурет 1 - Дискінің әдеттегі ағындық әрекетін жасау

**Екілік файл енгізу/шығару функциялары үшін файлды буферлеуді өшіру арқылы тиімділікті арттыру.**

Әрбір файлды енгізу/шығару операциясы кезінде LabVIEW операциялық жүйеге (ОЖ) қоңырау шалу және дискідегі файлға және одан деректерді тасымалдауды сұрау үшін бірнеше миллисекундты алады. Бұл миллисекундтар LabVIEW операциялық жүйені шақырған сайын жинақталады. Осындай уақтылы қайталануларды болдырмау үшін ОЖ файлдық жүйелерінің көпшілігі файлға оқуды немесе жазуды күтетін деректердің әрбір бөлігін уақытша сақтау үшін буферді қамтамасыз етеді. Буфер толған кезде ОЖ бір файлды енгізу/шығару операциясын орындайды.

Алдыңғы абзацта сипатталған процесс буферлеу ретінде белгілі және LabVIEW оны әдепкі бойынша қосады. Буферлеу операциялық жүйенің дискіге кіру және өңдеу уақытын жұмсау санын азайтады.

Буферлеуді қашан өшіру керек

Кейбір жағдайларда буферлеуді өшірсеңіз, жылдамырақ деректер ағынының жылдамдығына қол жеткізуге болады. Мысалы, өңдеу жылдамдығын арттыру үшін тәуелсіз дискілердің артық массивін (RAID) пайдалануға болады. RAID – бұл ОЖ бір уақытта қол жеткізе алатын, деректерді оқуға немесе жазуға аз уақыт алатын бір диск ретінде әрекет ететін қатты дискілер жиынтығы. Буферлеу қосылған RAID-ке қол жеткізсеңіз, LabVIEW деректерді ОЖ-ға көшіру үшін ОЖ деректерді дискіге нақты жазуға кететін уақыттан көбірек уақыт алуы мүмкін. Бұл деректер көшірмелерін болдырмау және ОЖ-ны деректерді тікелей дискіге жіберуге мәжбүрлеу үшін Файлды ашу/Жасау/Ауыстыру функциясының буферлеуді өшіру кірісімен буферлеуді өшіруге болады .

Дегенмен, буферлеуді өшірсеңіз, келесі шарттар дұрыс екеніне көз жеткізіңіз:

- Файлдағы деректердің өлшемін файлды қамтитын немесе ең соңында қамтуы мүмкін дискінің байтпен көрсетілген сектор өлшеміне еселі етіп жасау керек. Сектор – деректердің тіркелген

көлемін, әдетте 512 байтты сақтайтын дискілік кеңістіктің бөлімшесі. Дискінің сектор өлшемін анықтау үшін [Get Volume Info функциясының](#) сектор өлшемін (байт) шығысын пайдаланыңыз . LabVIEW деректерін дискіге сақтау үшін деректер бірнеше секторларды қамтуы мүмкін, бірақ әрбір секторды толығымен толтыруы керек. 512 байт сектор 512 байт деректерді қажет етеді. Егер деректер сектор өлшеміне еселік болмаса, LabVIEW файлды қайта оқымас бұрын деректерді толтырғыш деректерімен толтырып, толтырғыш деректерін жою керек.

- Файлдағы деректер дискіге қажетті теңестірудің бірнеше есесіне туралануы керек. LabVIEW деректерді туралайды және сіз бұл туралауды өзгерте алмайсыз. Деректер туралау талабына сай болмаса, LabVIEW қатені қайтарады және буферлеуді қосып, файлды қайта ашу керек.

### **Қай файл пішімін пайдалану керектігін анықтау**

**Text** — Егер деректеріңізді Microsoft Notepad сияқты басқа қолданбаларға қолжетімді еткіңіз келсе, мәтіндік файлдарды пайдаланыңыз, себебі олар ең кең таралған және ең портативті болып табылады. LabVIEW-те мәтіндік файлдарға келесі файлдар түрлері жатады:

Конфигурация файлдары Excel файлдары LabVIEW өлшем файлдары

Электрондық кесте файлдары XML файлдары

**Екілік** — Кездейсоқ қол жеткізу файлдарын оқу немесе жазуды орындау қажет болса немесе жылдамдық пен ықшам дискілік кеңістік маңызды болса, екілік файлдарды пайдаланыңыз, себебі олар дискілік кеңістіктегі және жылдамдықтағы мәтіндік файлдарға қарағанда тиімдірек. LabVIEW-те файлдардың келесі түрлері екілік файлдарға жатады:

TDM/TDMS файлдары

Толқын пішіндері

Zip файлдары

**Datalog** — LabVIEW бағдарламасында деректердің күрделі жазбаларын немесе әртүрлі деректер түрлерін өңдеуді қаласаңыз, деректер журналы файлдарын пайдаланыңыз, себебі олар деректерге тек LabVIEW арқылы қол жеткізгіңіз келсе және күрделі деректер құрылымдарын сақтау қажет болса, деректерді сақтаудың ең жақсы жолы болып табылады.

(Windows) Жоғарыда аталғандардан басқа файл пішіміндегі деректерге қол жеткізу үшін LabVIEW қолданбасын пайдаланғыңыз келсе, DataPlugins қолданбасын пайдалануға болады. DataPlugins файл пішіміне қарамастан LabVIEW көмегімен деректерге қол жеткізудің бірыңғай әдісін қамтамасыз етеді.

## Файл жолдары

Төменде көрсетілген файл **жолын басқару элементі** дискідегі файлдың орнын анықтайтын LabVIEW деректер түрі болып табылады. 

Файл жолы файлды қамтитын көлемді, файлдық жүйенің жоғарғы деңгейі мен файл арасындағы каталогтарды және файлдың атын сипаттайды. Жолды басқару немесе индикаторы бар берілген платформа үшін стандартты синтаксисті пайдаланып жолды енгізіңіз немесе көрсетіңіз.

Төмендегі кестеде LabVIEW бағдарламасында пайдалануға болатын файл жолдарының әртүрлі түрлері туралы ақпарат бар.

Абсолютті жол файлдық жүйенің жоғарғы деңгейінен бастап файлдың немесе каталогтың орнын сипаттайды. Мысалы, E:\Temp\test.txt .

Салыстырмалы жол файлдық жүйедегі ерікті орынға қатысты файлдың немесе каталогтың орнын сипаттайды. Мысалы, ..\test.txt .

Символдық жол немесе псевдопат - бұл LabVIEW бағдарламасында пайдалануға болатын жолдың алдын ала анықталған қысқаша түрі. Мысалы, <vilib> символдық жолы labview каталогындағы <vi.lib> каталогын көрсетеді . LabVIEW іске қосу уақытында <vilib> символдық жолын абсолютті жол ретінде түсіндіреді, мысалы,

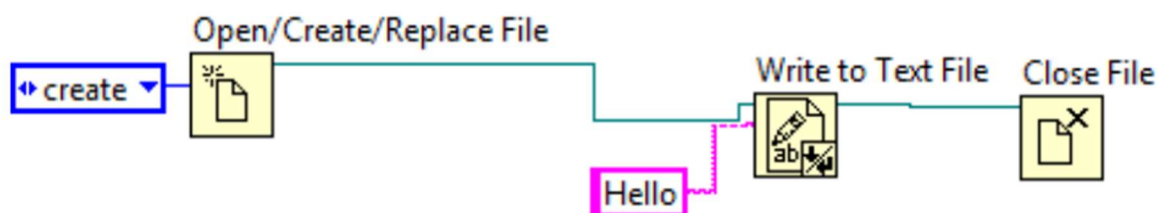
C:\Program Files (x86)\National Instruments\LabVIEW 2009\vi.lib  
немесе D:\Program Files\National Instruments\LabVIEW 2009\vi.lib ,  
LabVIEW орнатылған жерге байланысты.

**Деректерді мәтіндік файлға жазыңыз: келесі функцияларды пайдалана аласыз**

файлға деректерді жазу (Функциялар>>Бағдарламалау>>Файл  
енгізу/шығару):

- Файлды ашу/Жасау/Ауыстыру
- Мәтіндік файлға жазу
- Файлды жабу (жадыны босату)

Келесі кодтар файлға «Hello» жолын жазады



Сонымен қорындылай келе жалпылама функциялар түрлерін келтірейік:

## 1. Файл түрлері және олармен жұмыс принциптері

**Text File (.txt, .csv)** – адам оқи алатын құрылым

**Binary File** – тиімді сақтау, бірақ оқуға қиын

**Spreadsheet File** – кестелік мәліметтер үшін

## 2. Файлмен жұмыс істеудің негізгі қадамдары

1. Open/Create/Replace File – файлды ашу немесе жасау
2. Write to File / Read from File – мәлімет жазу немесе оқу
3. Close File – файлды жабу
4. Error cluster арқылы қателерді бақылау

Практикалық тапсырмалар:

Тапсырма 1: Журнал жүргізу

Сенсордан келетін деректерді (температура, қысым, ылғалдылық) уақыт белгісімен бірге мәтіндік файлға жазатын VI құрыңыз. Формат мысалы: 14:52:01 – Temperature: 23.6°C – Humidity 45% – Pressure: 1.01 bar

Тапсырма 2: Мәліметті қайта оқу және график салу

1-тапсырмада жазылған файлдан мәліметтерді оқып, тек температура мәндерін шығарып, Waveform Graph арқылы график құрыңыз.

Тапсырма 3: CSV форматындағы деректерді сақтау және жүктеу 2 өлшемді массивті (мысалы, 10x3: температура, қысым, уақыт) CSV файлға жазыңыз

Тапсырма 4: Қате өңдеу

Егер файл жоқ болса немесе тек оқуға арналған күйде болса – қате өңдейтін механизм қосыңыз. Қолданушыға хабарлама шығару және қате кодын экранға шығару.

Тапсырма 5: Файлды автоматты түрде сақтау

Қолданушы енгізген мәтінді әр 10 секунд сайын автоматты түрде жаңа жолмен файлға жазып отыратын жүйе жасаңыз.

## Бақылау сұрақтары

1. Файлға жазу және оқу операцияларының орындалу ретін түсіндіріңіз.
2. Format Into String және Write to Text File функциялары қалай әрекеттеседі?
3. Spreadsheet файлы дегеніміз не? LabVIEW-де онымен қалай жұмыс істейді?
4. Error Cluster қандай мақсатта қолданылады және оны қалай бақылауға болады?

5. LabVIEW-де мәтінді массивке, массивті мәтінге қалай түрлендіруге болады?