

Дәріс 7. If, If-Else және Switch операторларының LabVIEW-да баламасы

Дәріс мақсаты: Студенттерді дәстүрлі бағдарламалау тілдеріндегі (C, Python, Java т.б.) If, If-Else, және Switch операторларының LabVIEW-дегі баламаларымен таныстыру. LabVIEW ортасында Case Structure және Select функциясын пайдалану арқылы шарттық тармақталуды ұйымдастыруды үйрету.

Кіріспе

Көптеген мәтіндік бағдарламалау тілдерінде If, If-Else, және Switch операторлары логикалық шарттарға сәйкес түрлі кодтарды орындауға мүмкіндік береді. Мәтінге негізделген тілдерде if, if-else немесе switch операторларымен таныс болуыңыз мүмкін; LabVIEW балама құрылымдары қарапайым if операторларына арналған Select құрылымы және if-else немесе switch операторы сияқты көбірек енгізу таңдаулары қажет болған жағдайда Case құрылымы болып табылады. LabVIEW-де мұндай логикалық басқару құрылымдары графикалық түрде жүзеге асады. LabVIEW ортасында If және If-Else операторларының баламасы ретінде ең жиі қолданылатын құрал – Case Structure. Бұл құрылым логикалық шарт немесе мәнге байланысты бірнеше код блоктарының ішінен біреуін ғана орындауға мүмкіндік береді. Сонымен қатар, Select (таңдау) функциясы – қарапайым If-Else логикасын іске асыру үшін қолданылады. Ал Switch операторының баламасы ретінде Enum + Case Structure комбинациясы жиі пайдаланылады. Бұл жағдайда әрбір мүмкін мәнге сәйкес жеке Case жазылады. Осы дәрісте біз LabVIEW-дің шарттық басқару құрылымдарын қарастырамыз, олардың мәтіндік тілдердегі ұқсас операторлармен айырмашылығын және практикалық қолданылуын үйренеміз.

If таңдау шарты

Қарапайым шарттар мен әрекеттер үшін Таңдау функциясы if-мәлімдемесінің баламалы жұмысын аяқтайды. Таңдау функциясын «Функциялар» Бағдарламалау >> Салыстыру >> Таңдау (**Functions» Programming >> Comparison >> Select**). бөлімінен таба аласыз. Таңдау

функциясы үш кірісті қабылдайды. Жоғарыдан төменге қарай ретімен тізімделген кірістерді төменде көруге болады:

TRUE жағдайының қажетті шығысы

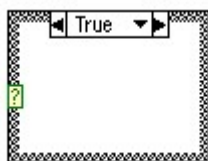
TRUE/FALSE селекторы

FALSE жағдайының қажетті шығысы

Селекторлық кіріс логикалық деректерді қабылдайды. Селектор деректерінің мәніне байланысты Таңдау функциясы ақиқат немесе жалған кіріске жалғанған мәнді қайтарады. Шындық және жалған нәтижелер әртүрлі деректер типтері болуы мүмкін. Жол деректер түрін пайдалану арқылы төмендегі сурет 1-дегі мысалды қараңыз.

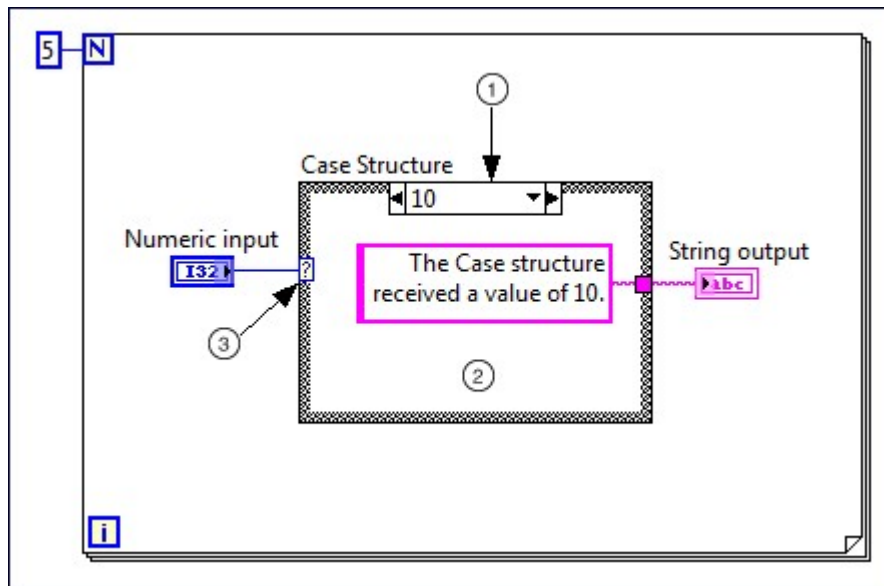
Is (**Case Structure**) құрылымы

Құрылым орындалған кезде дәл біреуі орындалатын бір немесе бірнеше ішкі диаграммаларды немесе жағдайларды қамтиды. Is таңдау құралына қосылған мән қай істі орындау керектігін анықтайды.



Сурет 1 - (**Case Structure**) құрылымын құру

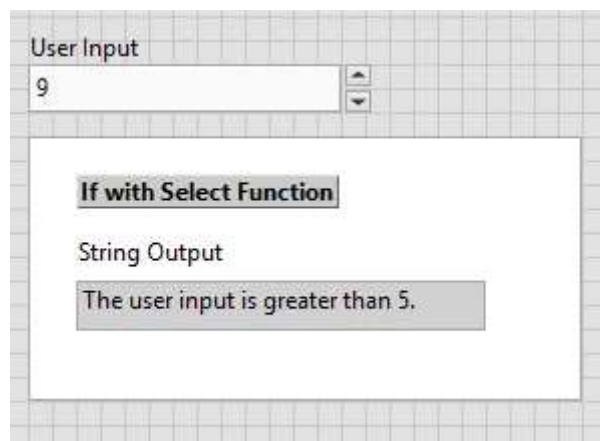
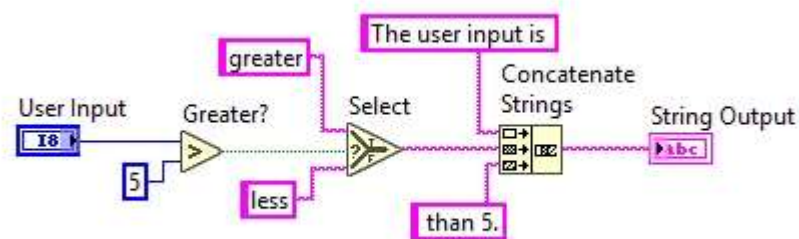
Case құрылымының құрамдас бөліктері 2- суретте келтірілген.



Сурет 2 - Іс құрылымының құрамдас бөліктері

- ① Таңдаушы белгі—байланысты іс орындалатын мән(дер)ді көрсетеді. Жалғыз мәнді немесе мәндер ауқымын көрсетуге болады. Әдепкі жағдайды көрсету үшін селектор белгісін де пайдалануға болады.
- ② Ішкі диаграмма(іс) — регистрді таңдау құралына жалғанған мән селектор жапсырмасында пайда болатын мәнге сәйкес келгенде орындалатын кодты қамтиды. Ішкі диаграммалардың санын немесе ретін өзгерту үшін Case құрылымының жиегін тінтуірдің оң жақ түймешігімен басып, сәйкес опцияны таңдаңыз.
- ③ Іс таңдау құралы — кіріс деректерінің мәніне қарай орындалатын істі таңдайды. Енгізілетін деректер логикалық, жол, бүтін сан, нөмірленген тип немесе қате кластері болуы мүмкін. Іс таңдау құралына жалғайтын деректер түрі селектор белгісіне енгізуге болатын рұқсат етілген жағдайларды анықтайды.

Таңдау функциясы пайдалы, тек кірісіңізге байланысты шығысыңызды өзгерту қажет. Дегенмен, енгізуіңізге байланысты қандай код орындалатынын өзгерту қажет болса, Іс құрылымын пайдалануға болады (Сурет 3).



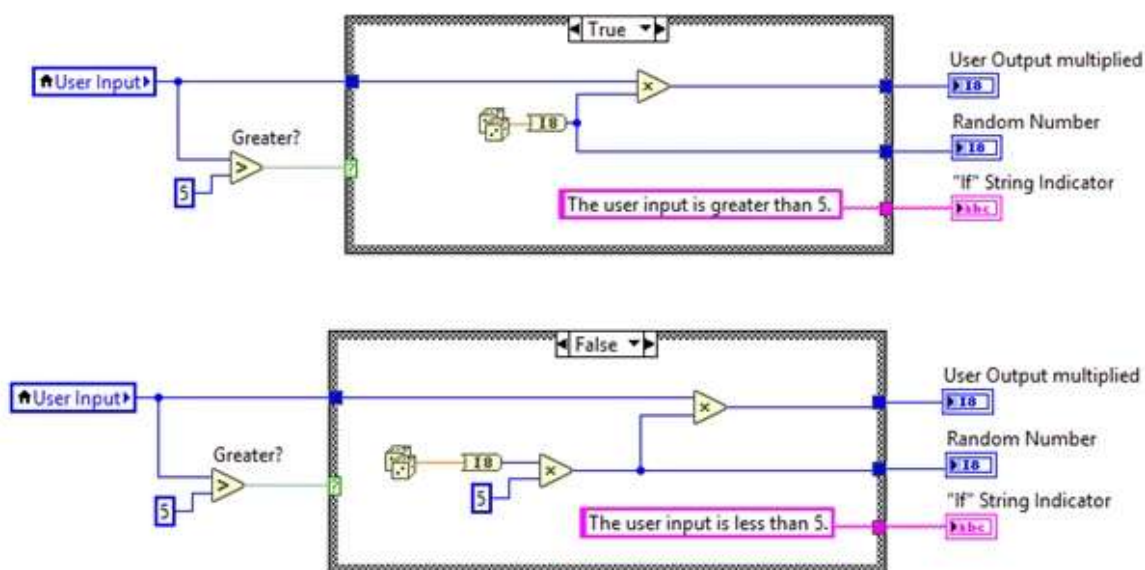
Сурет 3 – IF шартын орындау үшін **Select** функциясын пайдалану мысалы

Іс құрылымын Функциялар палитрасында Бағдарламалау >> Құрылымдар >> Іс құрылымы бөлімінен таба аласыз. Іс құрылымының үш бөлімі бар: Таңдау белгісі; Ішкі диаграммалар немесе жағдайлар; Іс таңдау құралы.

Іс құрылымы әдепкі бойынша бір True және бір False жағдайына сәйкес келеді. Әрбір жағдай үшін істің шарты орындалған жағдайда орындалатын LabVIEW кодын жазуға болады.

Іс құрылымында регистрді таңдау терминалы деп аталатын жасыл сұрақ белгісін енгізу терминалы бар. Әдепкі True/False жағдайларда регистр құрылымын пайдалану үшін осы терминалға логикалық кірісті өткізіп, қай істің орындалуы қажет шарттарды орнатыңыз.

Төмендегі мысалда Таңдау үлгісіне ұқсас код көрсетілген. Негізгі айырмашылық - Іс құрылымымен кодты таңдап орындауға болады. (Таңдау функциясын пайдалансаңыз, төменде сурет 4 - де көрсетілгендей деректеріңізді (яғни кодты орындау) айнымалы түрде басқара алмайсыз.)



Сурет 4 – Case If құрылымын пайдалану мысалы

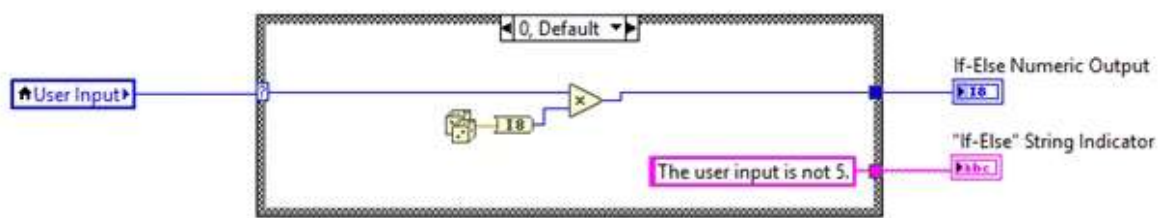
Бұл қарапайым мысалда, егер пайдаланушы енгізуі 5-тен үлкен болса, регистр таңдау құралы шынайы логикалық мәнді оқиды және пайдаланушы енгізуі кездейсоқ санға көбейтілген шынайы жағдайды орындайды. Немесе, егер пайдаланушы енгізуі 5-тен аз болса, регистр таңдау құралы жалған логикалық мәнді оқиды және кіріс үлкен кездейсоқ санды көбейтуге ұшырайтын жалған жағдайды орындайды. Әрбір істің бөлек коды бар және балама жағдайдан басқа нәтижелер береді (Сурет 4).

If-Else және Switch операторлары

Әдепкі ақиқат/жалған пішіндегі кейс құрылымдары if операторы ретінде пайдаланылуы мүмкін. Дегенмен, Case Structures пайдаланудың бір артықшылығы мынада: таңдау терминалының деректер түрін өзгертуге және if-else мәлімдемелерін және ауысу мәлімдемелерін жасау үшін жағдайларды қосуға болады.

If құрылымын жасағаннан кейін енгізу селекторының деректер түрін жолға, бүтін санға, тізімге немесе қате кластерінің деректер түріне өзгертуге болады. Қажетті деректер түрінің кірісін регистр таңдау құралына қосу жаңа деректер түріне сәйкес келетін таңдау белгісінің опцияларын автоматты түрде өзгертеді. Мысалы, төмендегі суретте пайдаланушы

кірісінің сандық деректер түрі селектор терминалына жалғанғанын және селектор белгісі енді сандық опцияны көрсетеді (Сурет 5).



Сурет 5 - Case құрылымы

Жасағыңыз келетін басқа салыстырулар (немесе мүмкін күйлер) үшін іс құрылымына ішкі диаграммаларды қосуға болады. Іс құрылымының әдепкі жағдайы if-else операторындағы «else» немесе коммутатор операторының «әдепкі» регистрі ретінде қызмет етеді. Әрбір жағдайда орындалатын код селекторлық терминалға қосылған кіріске негізделген.

Ауыстыру мәлімдемелері ретінде пайдаланылатын Case Structures архитектурасы пайдаланушы енгізе алатын деректер түрлеріне қатысты икемділікті, сондай-ақ болашақта пайдалану үшін өзгертулерді ұсынады. Сандық енгізудің орнына Іс құрылымы сонымен қатар регистрді таңдау кірісі ретінде оңай өзгертілетін нөмірленген енгізуді пайдалана алады.

Тапсырмалар

Тапсырма 1: Көп шартты шешім қабылдау жүйесі (іштелген If-Else логикасы)

Міндет:

Пайдаланушы 3 параметр енгізеді: температура (°C), ылғалдылық (%), және жел жылдамдығы (м/с).

1. **Case Structure** және **Select Function** көмегімен келесі логиканы құрастыр:

- Егер температура > 30 және ылғалдылық $> 70 \rightarrow$ "Өте ыстық және ылғалды" Егер температура > 30 және ылғалдылық $\leq 70 \rightarrow$ "Құрғақ ыстық"

- b. Егер температура ≤ 30 және жел $> 15 \rightarrow$ "Салқын және желді" Басқа жағдайларда $\rightarrow$ "Қалыпты ауа райы"
2. Әрбір жағдайға жеке Case терезе жасап, логиканы ішкі Select функцияларымен құрастыр.

Тапсырма 2: Enum + регистрі бар теңшелетін мәзір (ауыстырғыш мәлімдемесінің баламасы)

Міндет:

1. Enum басқару элементі келесі мәндерден тұрсын: "Шеңбер ауданы", "Тіктөртбұрыш ауданы", "Үшбұрыш ауданы", "Шығу". Case Structure арқылы әрбір геометриялық фигураның ауданын есептейтін жеке Case жаз.
а. Шеңбер үшін: $\pi * R^2$ Тіктөртбұрыш үшін: $a * b$
2. Үшбұрыш үшін: $0.5 * a * h$ "Шығу" мәні таңдалғанда VI-ды тоқтататын логика қос.

Бақылау сұрақтары

1. LabVIEW ортасында шарттық операторлар (If, If-Else) қалай жүзеге асады?
2. Switch Statement-ке балама қандай? сипаттаңыз.
3. Nested Case Structure қандай жағдайларда қолданылады және қандай кемшілігі бар?
4. Case Structure ішінде Error Cluster-ді қалай қолдануға болады?
5. Select Function мен Case Structure айырмашылығы неде?

