



LABVIEW MUHITI BILAN BIRINCHI TANISHUV

Ibragimov Ulug'bek Muradilloevich

Olimova Muxtarama Dilshod qizi

Buxoro muhandislik-texnologiya instituti

ARTICLE INFO

Received: 02nd November 2023

Accepted: 08th November 2023

Online: 09th November 2023

KEY WORDS

SCADA, ANSI C, "G" tili, for-loop, kross-platforma, portativ, assembler, Virtual Instruments, diagramma bloki, yuzadagi panel, interfeys, paradigma, indikator, Data Flow, komponent, kompilyator.

ABSTRACT

Ushbu maqolada National Instruments (AQSh) kompaniyasining "G" grafik dasturlash tilida asosi bo'lgan LabVIEW tizimida yaratilgan dasturlarni bajarish muhiti va platformasi haqida ma'lumotlar keltirilgan hamda uning asosiy imkoniyatlari ochib berilgan.

LabVIEW - National Instruments kompaniyasining eng yaxshi mahsulotlaridan biri. Avvalo shuni ta'kidlash kerakki, LabVIEW qisqartmasi bo'lib, Laboratory Virtual Instrumentation Engineering Workbench degan ma'noni anglatadi. Allaqachon nomidan laboratoriya tadqiqotlari, o'lchovlar va ma'lumotlar yig'ish yo'nalishini kuzatish mumkin. Darhaqiqat, LabVIEW-da SCADA tizimini yaratish "an'anaviy" ishlab chiqish vositalaridan foydalanishdan ko'ra biroz osonroqdir. Ushbu maqolada men LabVIEW ning mumkin bo'lgan doirasi biroz kengroq ekanligini ko'rsatmoqchiman. Bu mutlaqo boshqacha dasturlash tili yoki agar siz dasturlashning butun "falsafasini" xohlasangiz. Sizni boshqacha fikrlashga majbur qiladigan va ba'zan ishlab chiquvchi uchun mutlaqo ajoyib imkoniyatlarni taqdim etadigan funktsional til. LabVIEW umuman dasturlash tilimi? Bu munozarali masala - bu erda standart yo'q, masalan, ANSI C. Ishlab chiquvchilarning tor doiralarida biz "G" tilida yozamiz, deymiz. Rasmiy ravishda bunday til mavjud emas, lekin bu ishlab chiqish vositasining go'zalligi: versiyadan versiyaga ko'proq va ko'proq yangi konstruktsiyalar tilga kiritiladi. C ning keyingi reenkarnatsiyasida, masalan, for-loop uchun yangi tuzilma paydo bo'lishini tasavvur qilish qiyin. Va LabVIEWda bu juda mumkin. Ammo shuni ta'kidlash kerakki, LabVIEW TIOBE dasturlash tillari reytingiga kiritilgan bo'lib, hozirda o'ttizinchi o'rinni egallab turibdi - Prolog va Fortran oralig'ida.

National Instruments kompaniyasi 1976 yilda AQShning Ostin shahrida (Texas) uchta asoschi - Jeff Kodoski, Jeyms Truchard va Bill Noulin tomonidan tashkil etilgan. Kompaniyaning asosiy ixtisosligi ishlab chiqarishni o'lchash va avtomatlashtirish asboblari edi. LabVIEW ning birinchi versiyasi kompaniya tashkil etilganidan o'n yil o'tgach - 1986 yilda chiqarilgan (bu



Apple Mac uchun versiya edi). NI muhandislari "an'anaviy" dasturlash tillariga qarshi chiqishga qaror qilishdi va to'liq grafik rivojlanish muhitini yaratdilar. Jeff grafik yondashuvning asosiy mafkurasiga aylandi. Yildan yilga yangi versiyalar chiqarildi. Birinchi kross-platforma versiyasi (shu jumladan Windows) 1993 yilda chiqarilgan 3-versiya edi. Joriy versiya o'tgan yili chiqarilgan 8.6. Kompaniyaning bosh qarorgohi bugungi kungacha Ostinda joylashgan. Bugungi kunda kompaniyada deyarli to'rt ming kishi ishlaydi va ofislari deyarli qirq mamlakatda joylashgan.

Umuman olganda, grafik va matnga asoslangan dasturlash tillarini solishtirish juda qiyin. Bu, ehtimol, "PC" toifasidan "MAC" yoki "Windows" ni "Linux" ga nisbatan taqqoslashdir - siz xohlaganicha bahslashishingiz mumkin, ammo argument mutlaqo ma'nosiz - har bir tizim mavjud bo'lish huquqiga ega va har birining ham tarafdorlari, ham raqiblari bo'ladi, bundan tashqari, har bir mahsulotning o'ziga xos joyi bor. LabVIEW juda moslashuvchan bo'lsa-da, shunchaki vositadir.

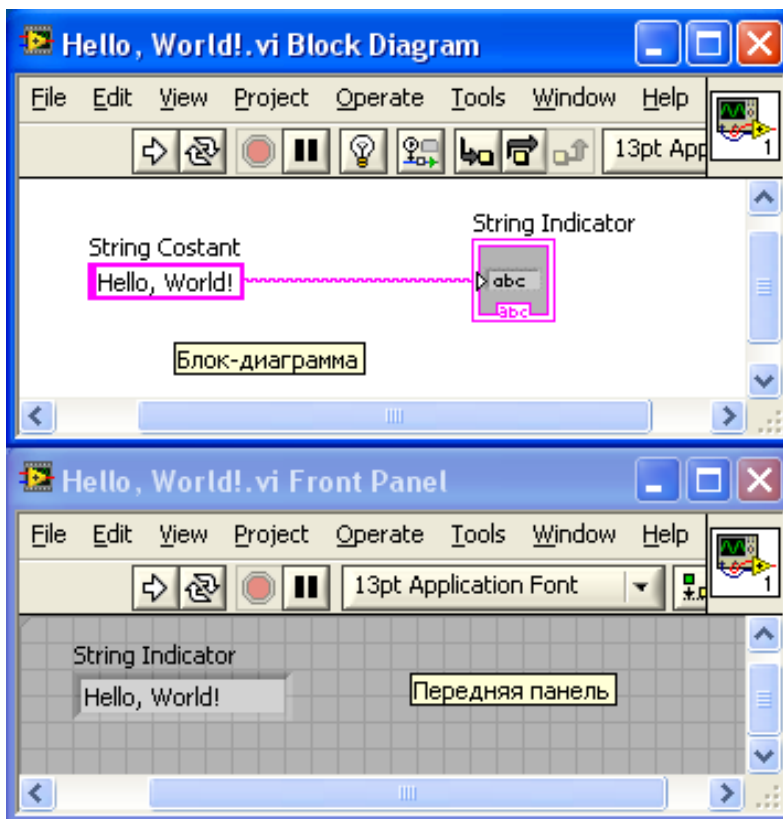
LabVIEW - platformalararo grafik ilovalarni ishlab chiqish muhiti. LabVIEW asosan universal dasturlash tilidir. Garchi bu mahsulot ba'zan National Instruments apparati bilan chambarchas bog'liq bo'lsa-da, u ma'lum bir mashina bilan bog'liq emas. Windows, Linux, MacOS uchun versiyalari mavjud. Manba kodi portativ bo'lib, dasturlar barcha tizimlarda bir xil ko'rinadi. LabVIEW tomonidan yaratilgan kod Windows Mobile yoki PalmOS-da ham ishga tushirilishi mumkin (adolat uchun PalmOS-ni qo'llab-quvvatlash to'xtatilgan, ammo bu erda Palmning o'zi ko'proq aybdor). Ushbu til katta tizimlarni yaratishda, matnlarni, rasmlarni qayta ishlash va ma'lumotlar bazalari bilan ishlashda muvaffaqiyatli ishlatilishi mumkin.

LabVIEW - bu juda yuqori darajadagi til. Biroq, LabVIEW dasturlariga "past darajadagi" modullarni kiritishingizga hech narsa to'sqinlik qilmaydi. Assembler qo'shimchalaridan foydalanmoqchi bo'lsangiz ham, bu ham mumkin, siz shunchaki DLL yaratishingiz va kodga qo'ng'iroqlarni kiritishingiz kerak. Boshqa tomondan, yuqori darajadagi til oddiy tilda ko'p satrlarni (agar o'nlab satrlar bo'lmasa) kodni olishi mumkin bo'lgan ma'lumotlar bilan juda ahamiyatsiz operatsiyalarni osongina bajarishga imkon beradi. Biroq, adolat uchun shuni ta'kidlash kerakki, past darajadagi tillarning ba'zi operatsiyalari (masalan, ko'rsatkichlar bilan ishlash) "yuqori darajadagi"ligi sababli LabVIEW-da amalga oshirish unchalik oson emas. Albatta, LabVIEW tili "an'anaviy" tillarda o'xshash bo'lgan asosiy boshqaruv tuzilmalarini o'z ichiga oladi:

- o'zgaruvchilar (mahalliy yoki global)
- shoxlanish (holat tuzilishi)
- For - tugatish tekshiruvi bilan va cheksiz tsikllar.
- while tsikllari
- Guruhlash operatsiyalari.

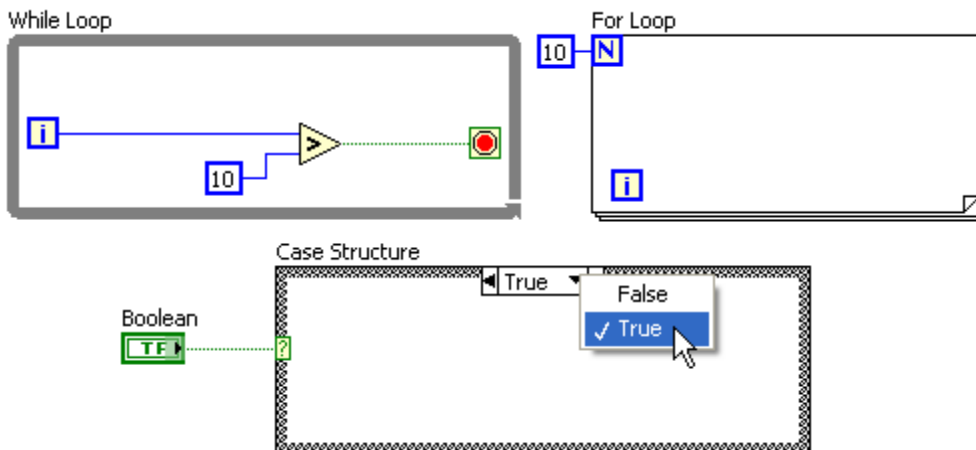
LabVIEW-da ishlab chiqilgan dasturiy modullar "Virtual Instruments" (Virtual Instruments) yoki oddiygina VI deb nomlanadi. Ular *.vi fayllarida saqlanadi. VI - bu LabVIEW dasturini tashkil etuvchi qurilish bloklari. Har bir LabVIEW dasturida kamida bitta VI mavjud. C tili nuqtai nazaridan, siz funktsiya bilan o'xshashlikni ishonchli tarzda chizishingiz mumkin, yagona farq shundaki, LabVIEWda bitta funktsiya bitta faylda joylashgan (siz asboblar kutubxonalarini ham yaratishingiz mumkin). O'z-o'zidan ma'lumki, bitta VI ni boshqa VI dan chaqirish mumkin. Asosan, har bir VI ikkita qismdan iborat - Blok diagrammasi (Blok

diagrammasi) va Old panel (old panel). Blok diagrammasi dastur kodi (aniqrog'i kodning vizual grafik tasviri), Old panel esa interfeysdir. "Salom, dunyo!"ning klassik namunasi quyidagicha ko'rinadi:



LabVIEW markazida ma'lumotlar oqimi paradigmasi joylashgan. Yuqoridagi misolda doimiy va indikator terminali chiziq bilan bog'langan. Ushbu chiziq Wire deb ataladi. Siz uni "sim" deb atashingiz mumkin. Simlar ma'lumotlarni bir elementdan ikkinchisiga o'tkazadi. Bu butun kontseptsiya Data Flow deb ataladi. Blok-sxemaning mohiyati tugunlar (tugunlar) bo'lib, ba'zi tugunlarning chiqishlari boshqa tugunlarning kirishlari bilan bog'langan. Tugun ish uchun zarur bo'lgan barcha ma'lumotlar kelgandan keyingina ishlay boshlaydi. Yuqoridagi diagrammada ikkita tugun mavjud. Ulardan biri doimiydir. Bu tugun o'z-o'zidan - u darhol bajara boshlaydi. Ikkinchi tugun ko'rsatkichdir. U doimiy o'tadigan ma'lumotlarni ko'rsatadi (lekin darhol emas, balki ma'lumotlar doimiydan kelishi bilan oq).

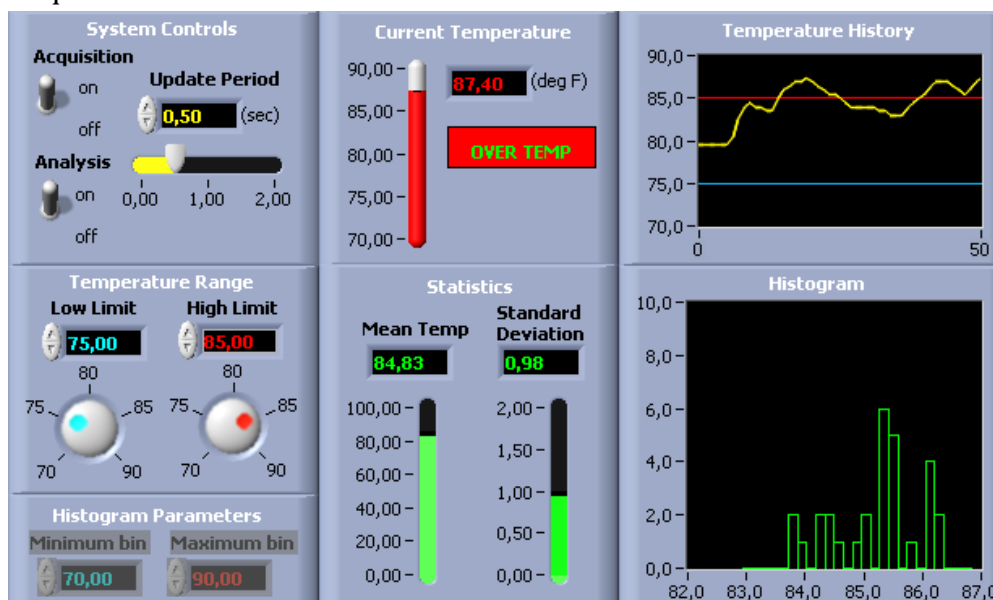
Va bu erda while / for tsikllari va if / then / else tuzilishi qanday ko'rinishga ega:





Yuqorida aytib o'tilganidek, barcha elementlar parallel ravishda bajariladi. Vazifani bir nechta protsessorlarda parallel ravishda ishlashi mumkin bo'lgan bir nechta oqimlarga parallellashtirish haqida o'ylashingiz shart emas. So'nggi versiyalarda siz hatto protsessorlarning qaysi birida bu yoki boshqa while-loop bajarilishi kerakligini aniq belgilashingiz mumkin. Endi ko'p protsessorli tizimlarni qo'llab-quvvatlashga osonlik bilan erishish imkonini beruvchi matn tillari uchun qo'shimchalar mavjud, ammo bu LabVIEW-dagi kabi hech qanday joyda amalga oshirilmagan bo'lishi mumkin. (yaxshi, men hali ham matn tillari bilan solishtirishga o'tdim). Agar biz allaqachon ko'p ish zarralari haqida gapiradigan bo'lsak, shuni ham ta'kidlash kerakki, ishlab chiquvchi iplarni sinxronlashtirish uchun keng imkoniyatlarga ega - semaforlar, navbatlar, uchrashuvlar va boshqalar.

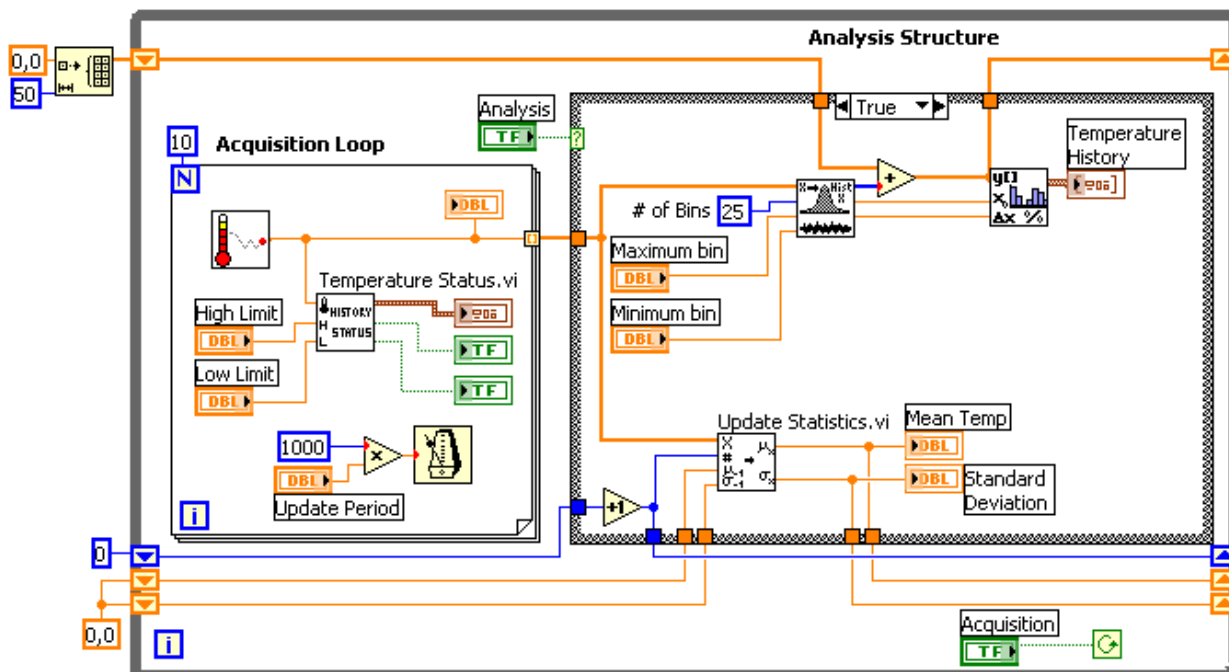
LabVIEW foydalanuvchi interfeyslarini yaratish uchun boy elementlar to'plamini o'z ichiga oladi. Delphi-dagi interfeyslar tezda nimaga "hujum" qilindi va LabVIEW-da bu jarayon yanada tezroq.



LabVIEW-ning standart yetkazib berilishi, shuningdek, ini fayllari bilan ishlash uchun bloklarni, registrni, ikkilik va test fayllari bilan ishlash funksiyalarini, matematik funksiyalarni, chizish uchun kuchli vositalarni (va laboratoriyada ularsiz) va yuqorida aytib o'tilgan imkoniyatlarga qo'shimcha ravishda o'z ichiga oladi. DLL qo'ng'iroqlari, LabVIEW ActiveX komponentlari va .net bilan ishlash imkonini beradi. Sakkizinchi versiyadan boshlab LabVIEW-ga sinflarni qo'llab-quvvatlash qo'shildi - til ob'ektga yo'naltirilgan bo'ldi. Amalga oshirilgan yordamni to'liq deb atash mumkin emas, lekin ob'ektga yo'naltirilgan tillarning asosiy xususiyatlari - meros va polimorfizm mavjud. Shuningdek, tilning funktsionalligi qo'shimcha modullar bilan kengaytirilishi mumkin, masalan, NI Vision Toolkit - tasvirni qayta ishlash va mashinani ko'rish uchun va boshqalar. Va Application Builder moduli yordamida siz bajariladigan exe faylini yaratishingiz mumkin.

Grafik kodni yomon o'qish mumkin degan fikrni tez-tez eshitishingiz mumkin. Darhaqiqat, odat bo'yicha, piktogramma va o'tkazgichlarning ko'pligi biroz hayratda qoldiradi. Shuningdek, boshlang'ich ishlab chiquvchilar "varaqa" dasturlari va "spagetti" dasturlarini yaratadilar. Biroq, tajribali LabVIEW dasturchisi, hatto dastur yuzlab modullardan iborat bo'lsa ham, ekran o'lchamidan kattaroq diagrammalarni hech qachon yaratmaydi. Yaxshi ishlab

chiqilgan dastur aslida "o'z-o'zini hujjatlashtiruvchi" hisoblanadi, chunki u allaqachon grafik tasvirga asoslangan.



Uzoq vaqt davomida LabVIEW da dasturlash jarayonida men LabVIEW tarjimon ekanligiga va blok diagrammalar yadro tomonidan doimiy ravishda talqin qilinishiga to'liq ishonchim komil edi. NI muhandislari bilan suhbatlashgandan so'ng, bunday emasligi ma'lum bo'ldi. LabVIEW - kompilyator (kod yaratish sifati ko'p narsani talab qiladi). Ammo kompilyatsiya tezda amalga oshiriladi - dastur ishlab chiqishning istalgan vaqtida har doim ishlashga tayyor. Shuningdek, LabVIEW kodi LabVIEW o'rnatilmagan kompyuterda ishga tushirilishi mumkin bo'lgan to'liq bajariladigan faylga kompilyatsiya qilinishi mumkin (garchi u LabVIEW Run-Time talab qilsa ham). Shuningdek, siz o'rnatish paketini o'rnatishingiz mumkin, InstallShield kabi uchinchi tomon yordam dasturlari talab qilinmaydi.

Paket xususiyatlarining batafsil va batafsil tavsifi ushbu maqola doirasidan tashqarida, lekin men uni sinab ko'rishni taklif qilaman (havolalar quyida keltirilgan). Buyuklar aytganidek, "...yangi dasturlash tilini o'rganishning yagona yo'li bu tilda dasturlar yozishdir". Xo'sh, tajribali dasturchilar olingan bilimlarni o'z ehtiyojlari uchun ekstrapolyatsiya qilishlari mumkin.

References:

1. Bress, T. Effective LabVIEW Programming: (*new file uploaded 02/19/15). — NTS Press, 2013
2. Сидеренко.Ф.А «Принципы использования системы NI LabVIEW в физическом практикуме технического вуза»- Уральский государственный педагогический университет, 2008
3. Кривоносов А.И., Прохвятилов В.Ю., Рыбальченко Т.В., Титов Н.Н. «Оперативно технический комплекс АСДУ нового поколения в Северной энергосистеме Украины» ООО "ХАРТЭП" корпорации " МАСТ-ИПРА", г.Харьков



4. Незнанов В.Н, фирма Dekart SRL, г. Кишинев «WEB и SCADA.» «Передовые ТЕХНОЛОГИИ и технические РЕШЕНИЯ» 1-2-2003.
5. Ибрагимов У.М., Нурова Ҳ. Муҳандислик олий таълим муассалари ихтисослик фанларини ўқитишда Modelica дастуридан фойдаланиш аҳамияти. Евразийский журнал математической теории и компьютерных наук, 3(5), 104–106. извлечено от <https://in-academy.uz/index.php/EJMTCS/article/view/15561>
6. Р.А.Гуляев. У.М.Ибрагимов. Ҳ.Б.Исмойилов. Элементы автоматизации как помощники цифровизации агропромышленности. "Science and Education" Scientific Journal / Impact Factor 3.848 March 2023 / Volume 4 Issue 3. 282-287 стр.