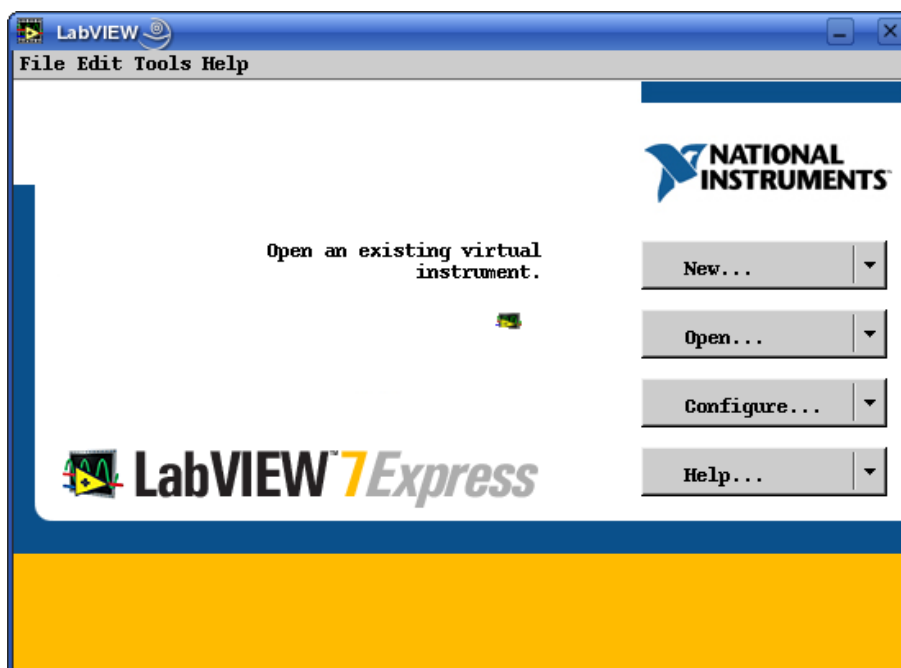


A LabVIEW alapelemeinek valamint alapelveinek megismerése

A LabVIEW program elindítása után az I.1 ábrán látható panel jelenik meg. A rajta található 4 gomb segítségével érhetjük az alapvető funkciókat, úgy mint új program készítése, meglévő program megnyitása, a LabVIEW beállításai és a segítségnyújtó rendszer.

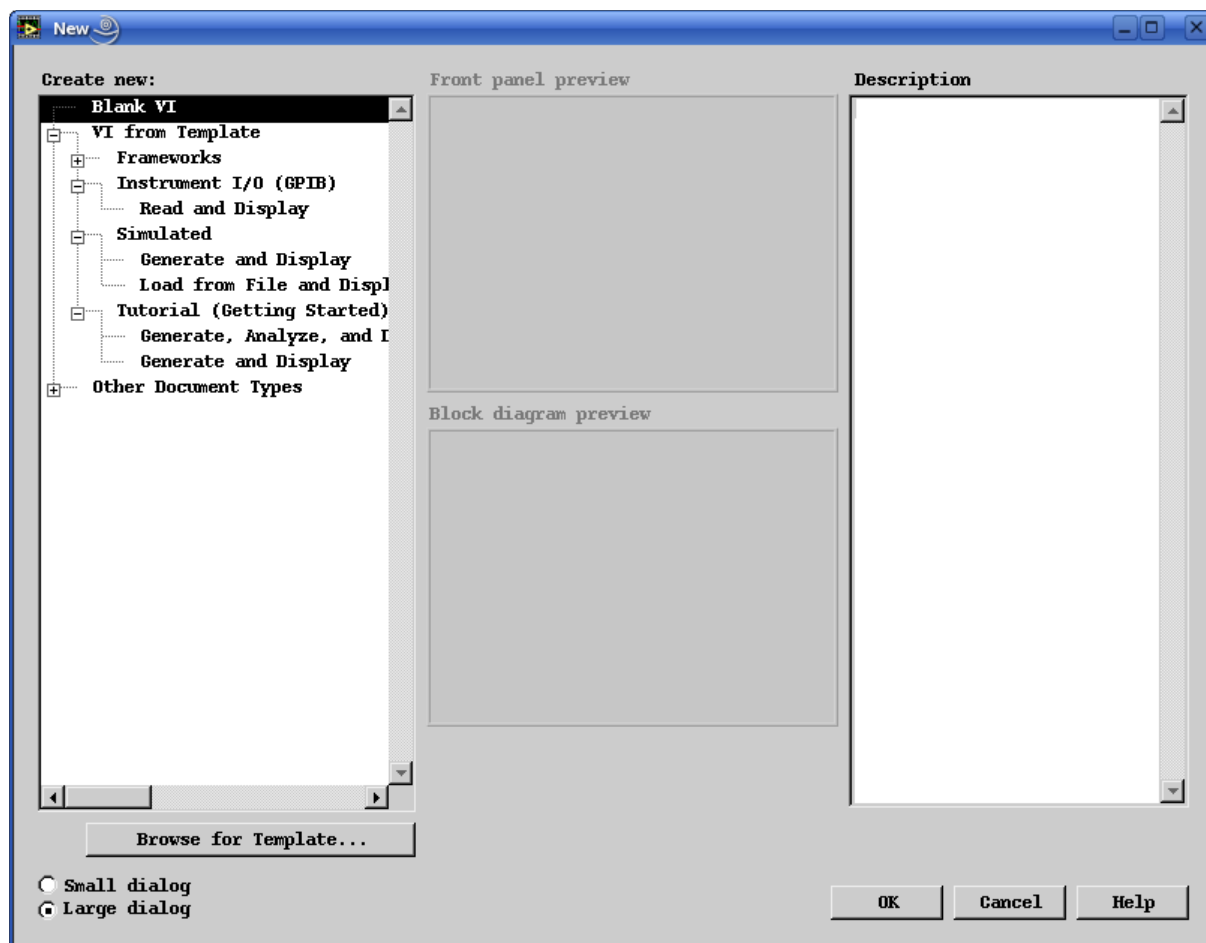


I.1. ábra: A LabVIEW7 kezdőképe

Egy új program létrehozásához két úton kezdhetünk hozzá. Az első esetben a 'New...' gomb feliratozott részére kattintva az I.2-es ábrán látható dialógusablakot nyitja meg a program. Itt választhatunk, hogy milyen típusú programot kívánunk létrehozni. Jelen esetben a 'Blank VI'-t válasszuk ki. (Néhány, a listában megtalálható egyéb típusról a későbbi fejezetekben még lesz szó.)

A 'VI' a 'Virtual Instrument', azaz a Virtuális Műszer szavak rövidítése. A LabVIEW programokat VI-oknak nevezzük. Ennek az az oka, hogy a LabVIEW programozási környezetet kifejlesztő National Instruments, a fejlesztőrendszert alapvetően az általa gyártott mérőkártyák számítógépes interfészének szánta, melyekkel a számítógépen, virtuálisan lehet megalkotni a mérőkártyák kijelzőit valamint kezelőszerveit.

A másik, némileg gyorsabb módszer az I.3-as ábrán látható. Ekkor a 'New...' gomb szélén látható háromszögre kattintva és a bal egérgombot lenyomva tartva kapjuk az I.3-as ábrán látható rövid legördülő menüt. A gombot folyamatosan lenyomva tartjuk, míg az egérmutatót a kívánt menüpont fölé mozgatjuk, és ott felengedjük.



I.2. ábra: Új VI készítése



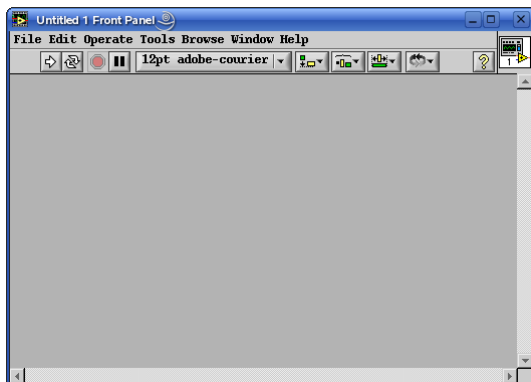
I.3. ábra: Új üres VI megnyitása

Bármelyik módszert is választottunk, az eredmény az I.4-es és I.5-ös ábrán látható két ablak megnyílása lesz. Az I.4-es ábrán az úgynevezett 'Front Panel' látható, mely tulajdonképpen megfelel a valódi műszerek kezelőszerveket és megjelenítőket tartalmazó előlapjának. A bemeneti elemeket beállítóknak (controls), a kimeneti elemeket pedig megjelenítőknak (indicators) nevezzük. A 'Front Panel'-en a bemeneti értékek beállítását valamint a kiszámított vagy mért értékek kiírását, kirajzolását végezhetjük el. Különböző típusú beállító és megjelenítő elemeket (gombokat, kapcsolókat, digitális és analóg típusú beállítókat, diagramokat, grafikonokat stb...) alkalmazhatunk annak érdekében, hogy a kialakított 'Front Panel'-lel a VI használata a felhasználó számára a legegyszerűbb legyen.

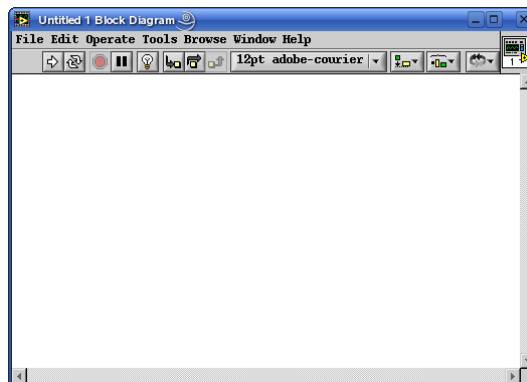
Minden 'Front panel'-nek van saját 'Diagram panel'-je (I.5. ábra), amelyben a VI blokk diagramját hozhatjuk létre. A LabVIEW grafikus programnyelvének segítségével - a G nyelvvel - építhetjük fel a blokk diagramot. Ez a blokk diagram felel meg a szöveges

programnyelvek által leírt utasítássorozatnak, forráskódnak. A programozást az egyes elemek adatvezetékekkel történő összehuzalozásával végezhetjük el.

A LabVIEW-ban alapvetően adatfolyam-orientált programozást végzünk (bár léteznek esemény-orientált programozást lehetővé tevő elemek is!). Ez azt jelenti, hogy programban elsődleges szerepe az adatnak, illetve az adat helyzetének (hollétének) van. Ebből következik, hogy a program megírása tulajdonképpen az adatvezetékek létrehozása a bemeneti, a műveletvégző valamint a kimeneti elemek között.

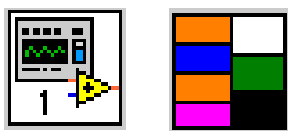


I.4. ábra: Front panel



I.5. ábra: Diagram panel

Itt kell megemlíteni, hogy a LabVIEW programok van még egy igen fontos eleme. Ez az úgynevezett 'Icon connector' (I.6. ábra), mely arra szolgál, hogy egy VI-ból subVI-t hozzunk létre. SubVI-ként a már megírt eljárásokat fel tudjuk használni más programokban, mint szubrutin vagy függvény.



I.6. ábra Icon connector

Az I.6. ábra bal oldalán látható ikon mind a 'Front panel'-en, mind a 'Block diagram'-on látható. Azonban az 'Icon connector' csak a 'Front panel'-en található meg. Ahhoz, hogy az I.6. ábra jobb oldalán levő konnektorokat lássuk, a jobb egérgombbal kattintsunk az 'Icon connector'-ra, majd a legördülő menüből válasszuk ki a 'Show connector' menüpontot. Az 'Icon connector' csatlakozói tulajdonképpen az adott rutin paraméterei, melyek a VI 'Front panel'-jén levő beállító illetve megjelenítő elemeknek felelnek meg. (A későbbiekben részletesen is tárgyalni fogjuk az 'Icon connector' használatát.)

A LabVIEW-beli számítás legnagyobb előnye a VI-ok hierarchikus felépítéséből ered. Miután létrehoztunk egy VI-t, azt azonnal felhasználhatjuk egy másik programban, mint subVI. A hierarchiában nincs korlát a szubrutinok kölcsönös hívásának mélységére vonatkozóan. Ezáltal elérhető, hogy a blokk diagram moduláris felépítésű, könnyen áttekinthető, megérthető és karbantartható legyen.

A továbbiakban a LabVIEW valamint a LabVIEW egyes elemeinek a megismerését példaprogramokon keresztül végezzük el.

1. Példa: Hatványozás logaritmussal (program1_1.vi)

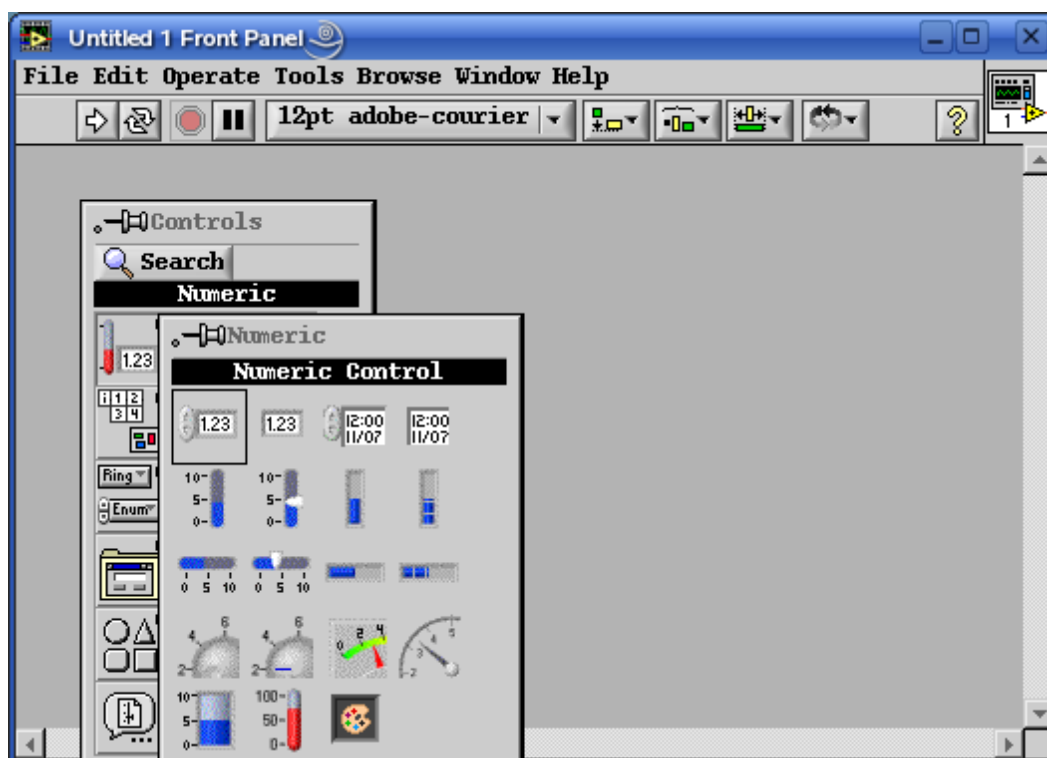
Mielőtt bárki megjedne, nem azért készítjük ezt a programot, mert a LabVIEW-ban nincs a hatványozás elvégzésére más megoldás, hanem azért mert ezen az egyszerű feladaton keresztül kívánjuk megismertetni Önöket az alapvető LabVIEW elemekkel.

A feladat megoldásához a következő matematikai összefüggést kell megfogalmaznunk a LabVIEW által használt G nyelven:

$$\text{eredmény} = e^{\text{kitevő} \cdot \ln(\text{alap})}$$

A programozás elkezdéséhez szükségünk lesz két bemeneti elemre (*kitevő*, *alap*) valamint az *eredményt* megjelenítő kijelzőre. Ehhez két numerikus bemenetet (numeric control) és egy numerikus kimeneti elemet (numeric indicator) kell a 'Front Panel'-en elhelyeznünk. Nagyon fontos, hogy tudatosítsuk magunkban, hogy a LabVIEW-ban külön bemeneti (control) és külön kimeneti (indicator) típusú elemekkel dolgozunk, melyek közül az előbbi adatbevitelre, az utóbbi adatok megjelenítésére szolgál. (Természetesen megoldható, hogy olvassunk egy kijelző típusú elemet (és fordítva), de ezt a gyakorlatot néhány később ismertetésre kerülő oknál fogva lehetőleg kerüljük!)

Az elemek kiválasztásához kattintsunk a jobb egérgombbal a 'Front Panel' felületén. Ekkor megjelenik a 'Controls' paletta, melyen további alpaletták találhatók. Amennyiben valamelyik alpaletta fölé visszük a kurzort, úgy az adott alpaletta neve megjelenik egy fekete sávban a 'Controls' palettán (I.7. ábra), valamint néhány pillanat múlva a kiválasztott alpaletta is láthatóvá válik. Ezeken találhatóak meg az adott csoportba tartozó elemek, esetleg további alpaletták.



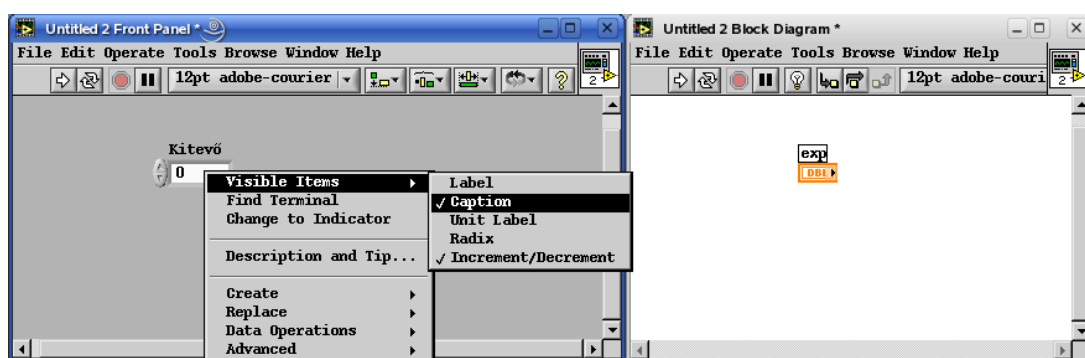
I.7. ábra: A 'Numeric' paletta

Jelen esetben a használni kívánt elemek a 'Numeric' alpalettán található 'Numeric Control' (I.7. ábra) és 'Numeric Indicator'. A kiválasztandó elem fölé érve a kurzorral, annak elnevezése

megjelenik az adott paletta felső részén. A palettán az elemre kattintva, és a 'Front Panel'-en egy tetszőleges pozícióba állítva, egy újabb kattintással tudjuk elhelyezni az egyes elemeket. Látható, hogy a LabVIEW egy nevet ad abban a pillanatban az elemnek, ahogy azt a 'Front Panel'-re tettük, de az is látható, hogy a név kijelölt állapotban van, azaz ilyenkor könnyen átnevezhetjük az általunk célszerűnek ítélt névre.

Fontos azonban tudnunk, hogy ekkor az elem ún. 'label' tulajdonságnak adunk értéket, amely tulajdonképpen az elem, vagyis az elem által képviselt változó neve, és mint ilyen futásidőben nem változtatható meg. Van egy másik 'caption'-nek nevezett tulajdonsága is a LabVIEW objektumoknak, amelyet pedig az objektumhoz tartozó feliratnak nevezhetünk. A 'Front Panel'-en mindkét tulajdonság megjeleníthető, azonban a 'Diagram Panel'-en csak a 'Label', azaz a változó neve látható.

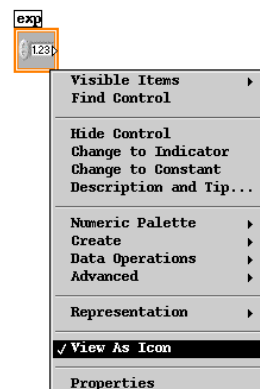
Ha még nem tettük volna meg, helyezünk el egy 'Numeric Control' elemet a 'Front Panel'-re, és nevezzük el 'exp'-nek.



I.8. ábra: A 'label' és a 'caption' tulajdonságok

Az elem területén az egér jobb gombjával kattintva hozhatjuk elő az objektumhoz tartozó legördülő menüt. Ebben a menüben többek között az adott elem megjelenését valamint viselkedését állíthatjuk be (az egyes menüpontokkal a példák során folyamatosan ismerkedünk meg). Az első pont a 'Visible Items' azaz a látható elemek. Itt most a 'Label' illetve az 'Increment/Decrement' pont van kipipálva. Válasszuk ki a 'Caption' pontot. Ekkor a 'Front Panel'-en megjelenik az elem előtt még egy 'exp' felirat. Kapcsoljuk ki a 'Label' tulajdonság megjelenítését, és a felíratra a bal egérgombbal duplát kattintva írjuk azt át 'kitevőre' (I.8. ábra). Ha megnézzük a 'Block Diagram'-ot, látható, hogy azon továbbra is egy 'exp' címkejű objektum látható.

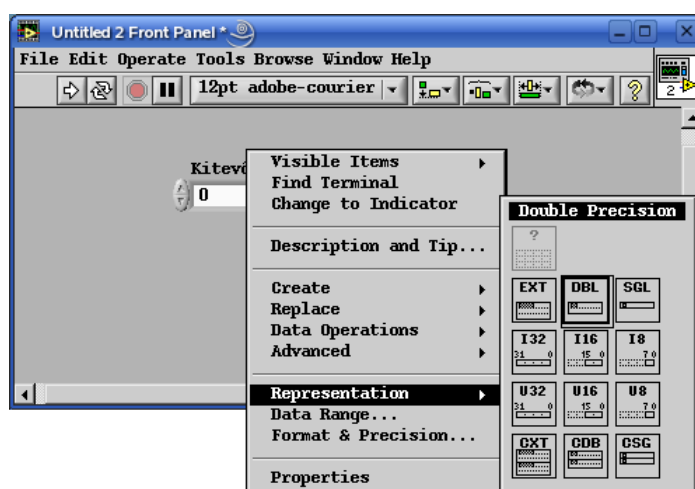
Minden 'Front Panel'-en elhelyezett ki- vagy beviteli elemnek van egy párja a 'Block Diagram'-on. Ez vagy az I.8. ábrán is látható módon néz ki, vagy ikon alakja van (I.9. ábra), a LabVIEW beállításaitól függően. Ha az alapbeállítástól eltérő módon szeretnénk, hogy kinézzen egy elem megjelenési formája, akkor a 'Diagram Panel'-en az adott elemen jobb egérgombbal kattintva a legördülő menüből a 'View as Icon' menüpont segítségével tudjuk megváltoztatni az elem megjelenését (I.9. ábra).



I.9. ábra: Ikonkézet

Amennyiben már jó néhány változónk van a paneleken, és ráadásul különböző címkéket és változóneveket használtunk (mondjuk azért mert a 'Front Panel'-en több nyelven is megszeretnénk jeleníteni a feliratokat), akkor eléggé nehézé válhat az összetartozó objektumok megtalálása. Ezt megkönnyítendő a 'Front panel'-en megjelenő legördülő menüben a 2. helyen található egy 'Find Terminal' pont (I.8. ábra), amely az adott elemhez tartozó 'Block Diagram'-beli terminált villantja fel, illetve a 'Block Diagram'-on megjelenő menüben szintén a 2. helyen van egy 'Find Control' vagy 'Find Indicator' menüpont (I.9. ábra), amely pedig az adott terminálhoz tartozó 'Control/Indicator' elemet villantja fel a 'Front Panel'-en.

A LabVIEW-ban a különböző adattípusok különböző színnel vannak jelölve a 'Block Diagram'-on. Az általunk eddig létrehozott egyetlen változó narancs színnel van jelölve, és az ábrán egy „DBL” feliratot is tartalmaz utalásképpen az adattípusára (ezért javaslom inkább ezt a típusú megjelenítést), azaz a Double típusra. Egy elem adattípusát a hozzá tartozó legördülő menü 'Representation' menüpontján keresztül tudjuk megváltoztatni.




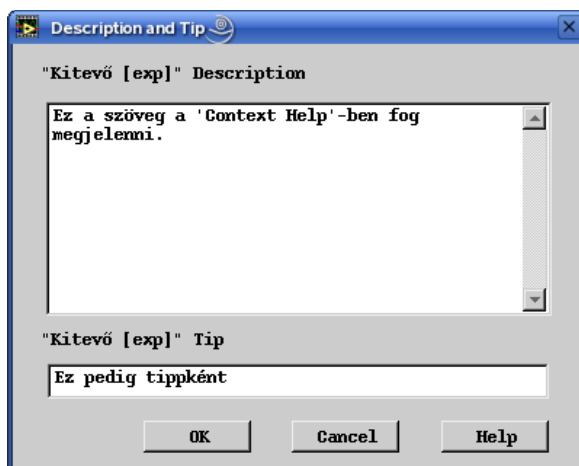
I.10. ábra: Numerikus adattípusok

Ha megnézzük az I.10-es ábrát 12 különböző adattípust látunk, amelyekben az egyes sorokban szereplő típusok között a számbázis pontosság csak a különbség. A két középső sorban egész típusú (előjeles (Integer) és előjel nélküli (Unsigned)) adattípusok vannak, 8, 16 illetve 32 bites pontossággal. Ezt a 6 típust kék színnel jeleníti meg a LabVIEW.

A felső sorban lebegőpontos típusok találhatóak 32, 64 és 128 bites pontossággal. A legalsó sorban levő típusok segítségével komplex mennyiségeket tudunk kezelni, minkét számtengelyen a lebegőpontos adattípusnak megfelelő pontossággal. A komplex mennyiségek a lebegőpontosakhoz hasonlóan narancs színnel vannak megjelölve.

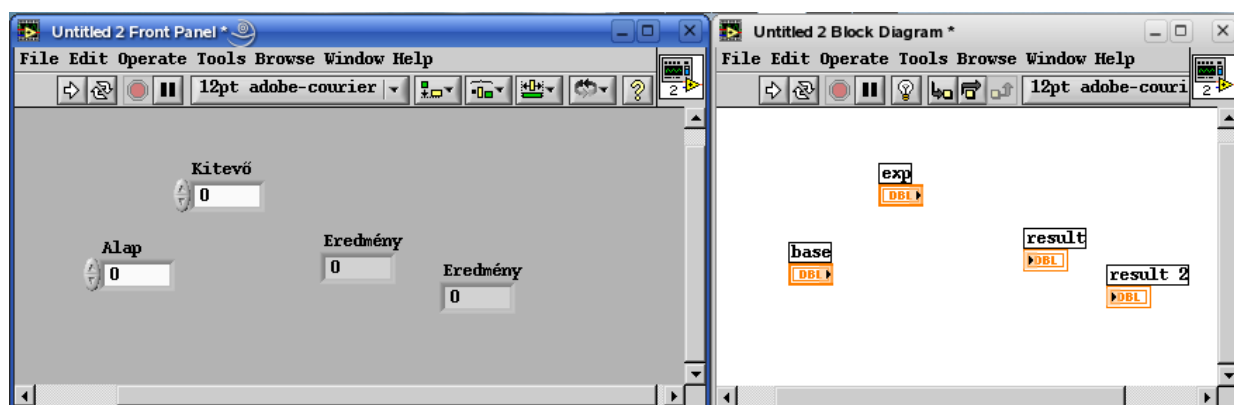
A legördülő menüben található pontok közül, most még eggyel ismerkedjünk meg, nevezetesen a 'Description and Tip' menüponttal. Ide kattintva az I.11. ábrán látható ablak jelenik meg, melyben leírást illetve felbukkanó tippet készíthetünk a 'Front Panel' elemeihez.

A tipp akkor jelenik meg, ha az elem felett tartjuk a kurzort. A leírás pedig a 'Context Help' ablakban fog megjelenni. Ez két módon tudjuk bekapcsolni. Az első lehetőség, hogy a 'Help' menüből kiválasztjuk a 'Show Context Help' menüpontot (természetesen működnek a menüben megtalálható billentyűkombinációk is), a másik mód pedig, hogy a paneleken lévő ikonok mellett található  gombra kattintunk. Ebben a 'Help' ablakban nem csak az általunk írt információk jelennek meg, hanem a LabVIEW által biztosított elemekről is találhatunk egy rövid leírást. Ez főként a kezdeti időszakban nagyon hasznos információforrás, ezért javaslom, hogy legyen mindig bekapcsolva ez az ablak.



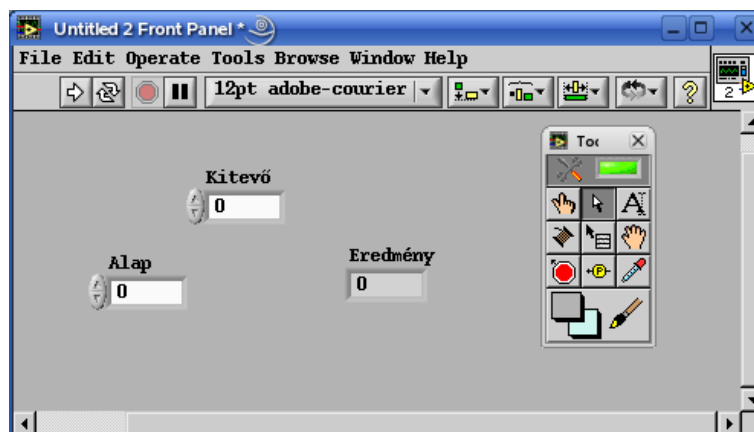
I.11. ábra: Leírás és tipp

Most térjünk vissza a programunk készítéséhez. Vegyünk le a 'Numeric' palettáról egy 'Numeric Indicator' elemet, és az előbbieken bemutatott módon állítsuk be a 'label'-jének értékét „Result”-ra és a 'caption'-jét pedig „Eredményre”. Szükségünk van még egy beviteli elemre. Ezt a már ismert módon levehetnénk a palettáról, de van több más módja is. Az egyik, hogy az 'Edit' menüben található 'Copy'/'Paste' menüpontokat használjuk, vagy a hozzájuk tartozó billentyűkombinációkat (Windows: Ctrl-C/Ctrl-V; Linux: Alt-C/Alt-V). A másik módszer, hogy a Windows: Control / Linux: Control+Alt billentyűket lenyomva tartva megfogjuk a másolni kívánt elemet és odébb húzzuk a 'Front Panel'-en. Ekkor az új elem 'caption'-je megegyező lesz az eredeti elmével, azonban a 'label' értékhez egy szám adódik hozzá intelligens módon, úgyhogy ezeket nevezzük el „Alap”-nak illetve „Base”-nek. A fentiek illusztrálása érdekében a I.12 ábrán az „Eredmény” mezőből is készítettünk egy másolatot, amire azonban nem lesz szükségünk. A 'Front Panel'-re vagy a 'Diagram Panel'-re helyezett felesleges elemeket az elem(ek) kijelölésével és a 'Delete' gomb lenyomásával tudjuk törölni.










I.12. ábra: Elemek másolása

Bizonyára észrevette a kedves Olvasó, hogy a kurzor formája folyamatosan változik, attól függően, hogy a 'Front Panel'-en vagy a 'Diagram Panel'-en egy adott elem mely része felett van éppen. Ahhoz, hogy látható legyen az adott pillanatban használt kurzor típusa, illetve az összes kurzortípusok elérése érdekében, kapcsoljuk be az ún. 'Tools Palette'-t, amit a 'Window' menüben tehetünk meg.



I.13. ábra: Tools palette

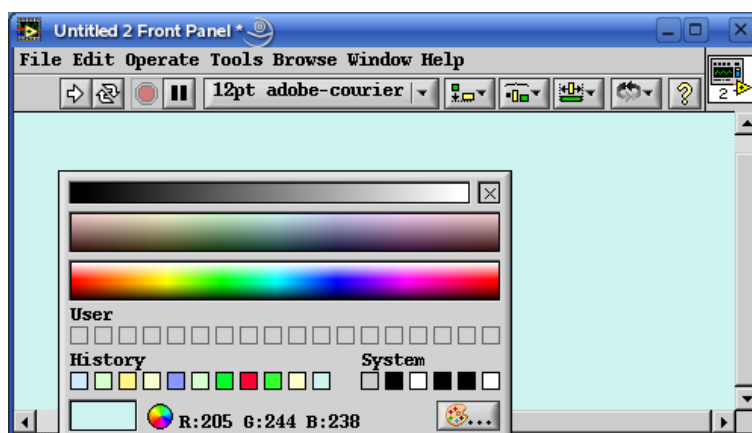
Tekintsük át a legfontosabb eszközöket:

- : automatikus kurzorkiválasztás (bekapcsolva az I.13. ábrán látható). Amennyiben bekapcsoljuk, akkor a következő négy pontban bemutatásra kerülő kurzortípus közül választ a rendszer, attól függően, hogy egy elemnek mely része felett áll éppen a kurzor.
- : működtető kurzor. Ezzel a típussal tudjuk például a 'Front Panel'-en egy kontroll elem értékét megváltoztatni (jelen programban esetében a numerikus kontroll értékét növelni vagy csökkenteni, annak bal szélén található 'Increment/Decrement' eszközével).
- : kijelölő kurzor. Ezt típusú kurzort elem(ek) kijelölésére használjuk. Egyszerre több elem kijelölése kétféle képen történhet, az első hogy az elemeket a 'Shift' gomb lenyomva tartása mellett egyesével jelöljük ki, a másik, hogy ezzel a típusú kurzorral az egér bal gombjának lenyomva tartásával egy kijelölőnégyzetet hozunk létre, amivel minden a négyzetbe akár csak részben belelógó objektumot kijelölhetünk.
- : címkéző kurzor. Ennek a kurzornak a használatával tudunk szöveg jellegű adatokat, feliratokat készíteni illetve megváltoztatni (például a 'caption' értékeket, de magának az „alap”-nak az mennyiségét is közvetlenül megváltoztathatjuk vele). Mivel a legtöbb szöveges mező tartalmazhat 'Enter'-t, ezért egy szöveg bevitelét alapvetően két módon fejezhetjük be. Az egyik, hogy valahol a szövegmezőn kívül kattintunk egyet az egérrel, a másik, hogy szövegbetelkor az adott panel bal felső részén megjelenő pipa gombra kattintunk.
- : huzalozó kurzor. A 'Block Diagram'-on ennek használatával kötjük össze a megfelelő elemeket egymással, vagyis ezzel a típussal írjuk magát a programot.
- : színlopó kurzor. Egy pixel színét „lophatjuk” le vele.
- : színező kurzor. A LabVIEW elemek előtér- illetve háttérszínét állíthatjuk be vele. Nem csak a 'Front Panel'-en, hanem a 'Diagram Panel'-en is használhatjuk, miáltal jelentősen javíthatjuk az elkészített programunk átláthatóságát. Ha ez a kurzor aktív, akkor ha egy elemre a jobb egérgombbal kattintunk, akkor egy színpaletta jelenik meg, mellyel az adott objektum elő- és háttérszínét állíthatjuk be (ekkor a két szín azonos lesz) (I.14. ábra)

A LabVIEW objektumok mind a 'Front Panel'-en, mind a 'Diagram Panel'-en több módon igazítható. Erre szolgál az 'Align' legördülő menü (I.15. ábra). Jelen esetben az „Alap” és az „Eredmény” kijelölt objektumokat vízszintesen egy vonalba úgy, hogy kiválasztjuk őket és az 'Align'

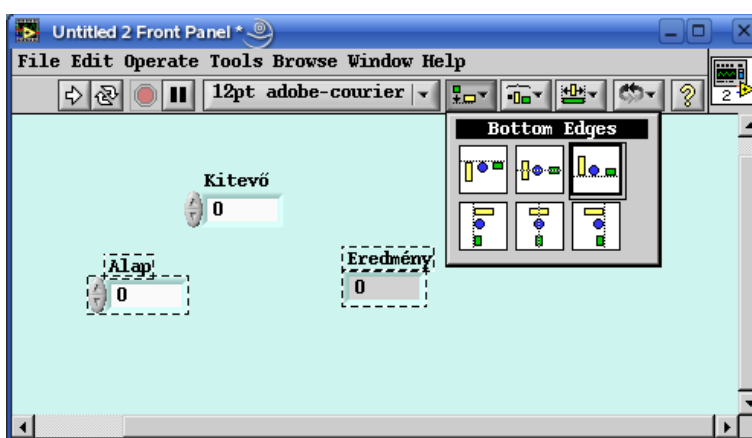
menü első sorából kiválasztjuk valamelyik igazítási típust (mivel egyforma méretű a két elem, ezért mindegy, hogy hová igazítjuk őket, egy vonalban lesznek).

A „Kitevő” elemet pedig helyezük úgy el, mintha egy hatványozás leírásánál szokás (I.15. ábra).



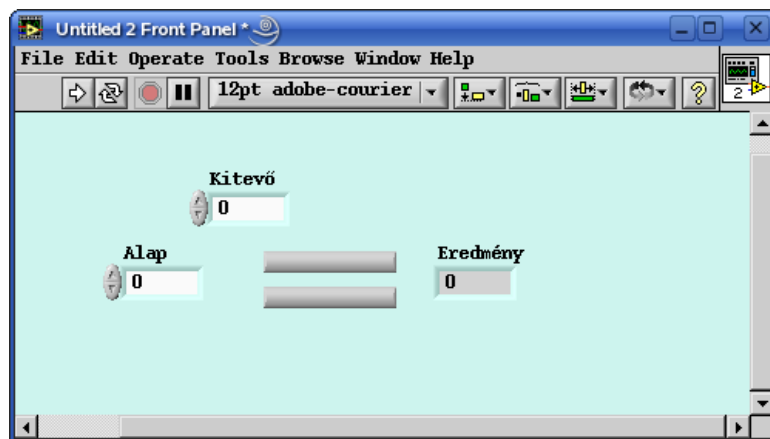
I.14. ábra: A színpaletta használata

A programozás során mindig törekedjünk arra, hogy a 'Front Panel' könnyen átlátható, és ha lehetséges kifejező legyen. Ennek érdekében helyeztük az elemeinket el a I.15. ábrán látható módon. Ahhoz, hogy még kifejezőbb képet tárjunk a majdani felhasználó felé, érdemes a 'Control Palette' 'Decorations' nevű alpalettáját megkeresni, ahonnan különböző dekorációs elemeket helyezhetünk el a 'Front Panelen'.



I.15. ábra: Objektumok igazítása

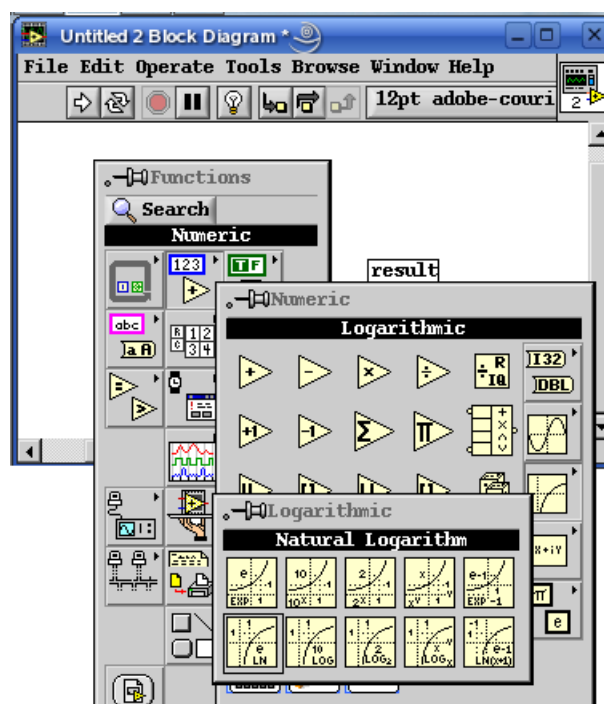
A 'Front Panel'-en kialakítandó egyenlethől még hiányzik az egyenlőségjel. Ennek kialakításához vegyünk le a 'Decorations' alpalettáról egy négyzet jellegű elemet, méretezzük át akkorára, mint egy egyenlőségjel egyik szára. Az átméretezés a 'Front Panel' objektumain akkor végezhető el, amikor a kurzort annak szélénél tartva megjelenik az elem oldalsó, szélső vagy sarokpontjaiban néhány fekete négyzet. Ezek fölé a helyezve a kurzort, annak képe megváltozik (az adott pont jellegétől függően). Ekkor a bal egérgombot lenyomva az elem mérete az adott irányban megváltoztatható. Amennyiben sarokpontnál végezzük az átméretezést, akkor a 'Shift' billentyű folyamatos lenyomva tartása mellett arányosan tudjuk nagyítani vagy kicsinyíteni az adott objektumot. Azért, hogy az egyenlőségjelünk két szára biztosan egyforma nagy legyen, a másikat a már elkészített szár másolásával készítjük el! Ha jól dolgoztunk, akkor a 'Front Panel' az I.16. ábrához hasonlóan kell kinézzen.



I.16. ábra: A végleges 'Front Panel'


Most pedig térjünk rá a tényleges programozási feladatra és ehhez váltsunk át a 'Diagram Panel'-re, melyen jelenleg csak a 'Control' illetve 'Indicator' eleinknek az ún. termináljai találhatóak.

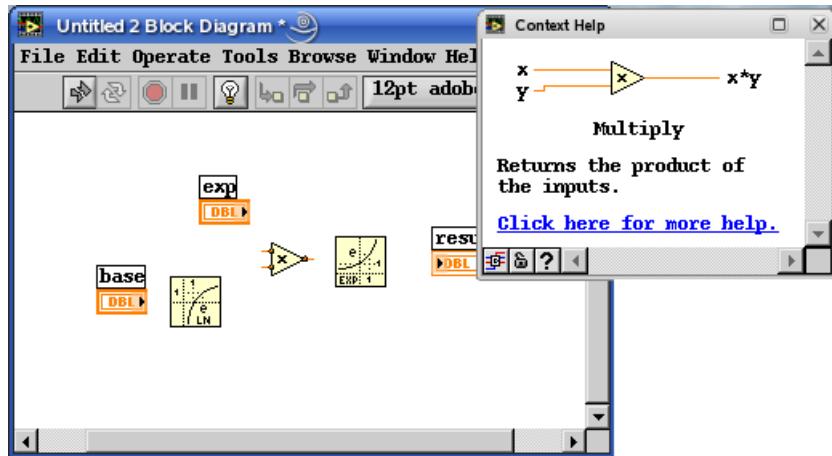
A programozás elvégzéséhez szükségünk lesz egy 10-es alapú logaritmus és egy exponenciális függvényt, valamint egy szorzás műveletet megvalósító elemre.



I.17. ábra: A 'Logarithmic' alpaletta

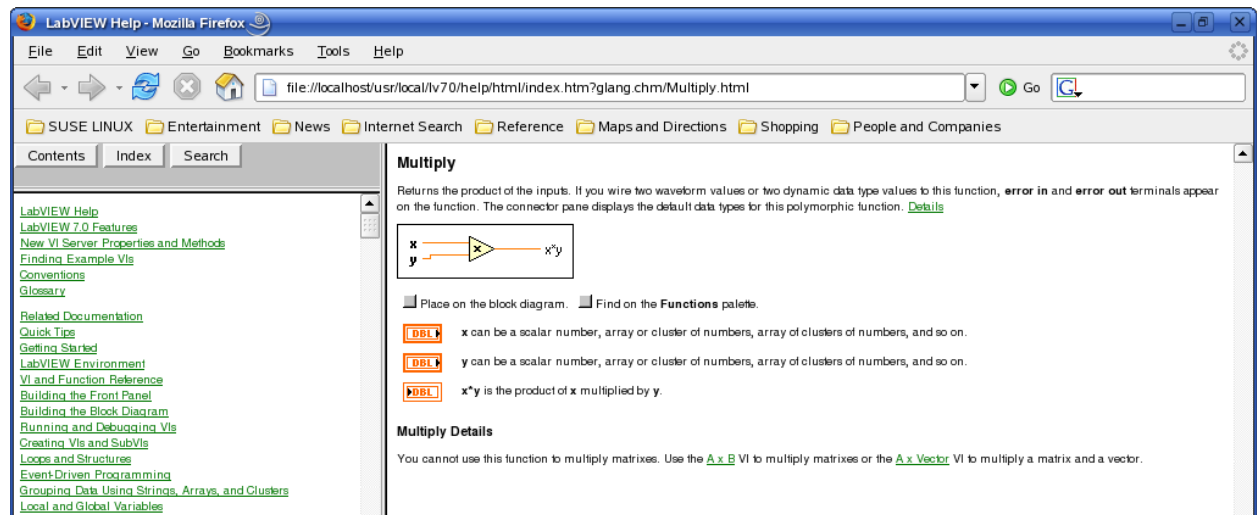
Ezeket az elemeket úgy csalogathatjuk elő, hogy a 'Diagram Panel'-on az egér jobb gombjával kattintunk egyet, mire megjelenik az ún. 'Functions Palette', melyen keresztül a beépített függvényeket érhetjük el, melyek témák szerint csoportosítva helyezkednek el. Az első sor középső, 'Numeric'-nek nevezett palettáján találjuk a számtani műveletekhez használható függvényeket, a szorzást például az első sor harmadik helyén. Az exponenciális típusú függvényeknek külön alpalettája van, az utolsó oszlop harmadik helyéről nyílik meg és ahol az

első oszlopban található az a két függvény amelyre még szükségünk van. A kiválasztandó elem ikonjára kattintva kiválasztjuk, majd a 'Diagram Panel'-en újra kattintva tudjuk a függvényeket a programozási felületre helyezni. Ekkor a 'Block Diagram'-unk az I.18 ábrának megfelelően kell, hogy kinézzen. (Ha a 'Context Help' ablak nincs bekapcsolva, akkor azt a 'Help' menü 'Show Context Help' menüpontjával, vagy pedig a panelek jobb felső részén található  gombbal tehetjük meg.)





I.18. ábra: A 'Block Diagram' és a 'Context Help' ablak

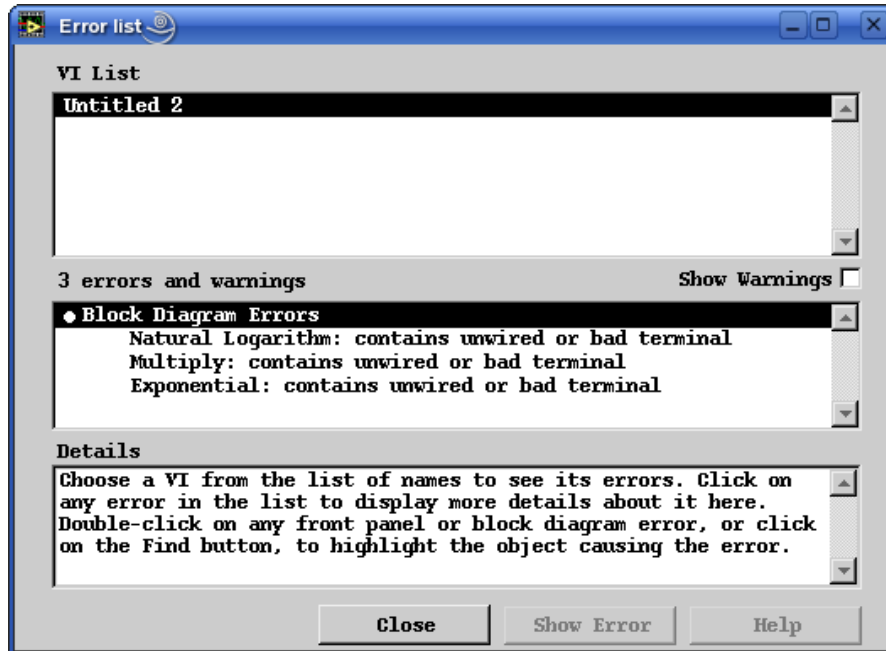
A 'Diagram Panel'-en a kurzort egy elem fölé állítva a 'Context Help' ablakban megjelenik egy rövid leírás, valamint a legtöbb esetben egy link, amelynek a segítségével egy részletesebb ismertetőt kapunk az adott funkciót megvalósító VI-ról. A részletes segítség Windowsban annak a chm-megjelenítőjével történik, Linux rendszerekben pedig valamelyik webböngésző jeleníti meg, ahogy az az I.19 ábrán is látszik.



I.19. ábra: Részletes leírások böngészése

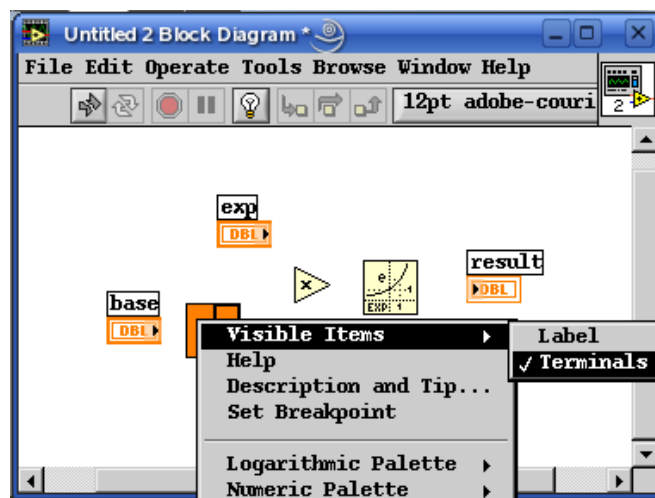
A szemfüles olvasó bizonyára észrevette, hogy a panelek bal felső részén található fehér nyíl  megváltozott és most egy szürke, törött nyíl  látszik helyette. Ez a gomb a 'Run/Error list' gomb. Amikor működőképes a LabVIEW programunk (fehér nyíl), akkor ennek megnyomásával futtathatjuk le, azonban ha valamilyen hibát tartalmaz a program, akkor egyrészt jelzi ezt a

szürkévé és törötte válásával, másrészt ha ekkor nyomjuk meg, akkor egy hibalistát jelenít meg nekünk a LabVIEW (I.20. ábra). A listában egy elemet kijelölve a legalsó mezőben megjelenik az adott hiba rövid leírása. A hibalistában egy hibára duplán kattintva felvillan a hibát kiváltó elem a 'Block Diagram'-on.



I.20. ábra: A hibalista

Jelen esetben a hibát az okozza, hogy az imént elhelyezett függvényeink nincsenek bekötve. Az egyes elemek közötti adatvonalak létrehozását a huzalozó kurzorral (☞) végezhetjük el, úgy hogy az egy kimeneti elem csatlakozási pontjára kattintunk, majd ezt megismételjük egy bemeneti ponton. A LabVIEW függvények csatlakozási pontjait illetve ezek kiosztását úgy tudjuk megtekinteni, hogy az elemen kattintunk egyet az egér jobb gombjával, és a legördülő menüből kiválasztjuk a 'Visible Items' menüpontból a 'Terminals' pontot (I.21. ábra)



I.21. ábra: Terminálok láthatóvá tétele



Kössük tehát össze a

$$\text{eredmény} = e^{\text{kitevő} \cdot \ln(\text{alap})}$$

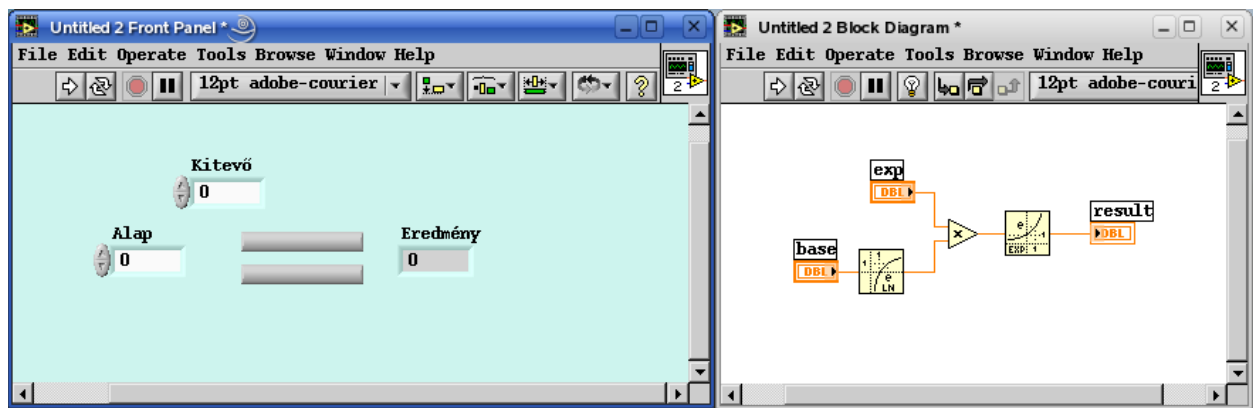
képletnek megfelelően a 'Block Diagram'-on elhelyezett elemeinket. Látható, hogy a huzalok az egyes elemek között narancssárga színűek, azaz lebegőpontos adattípust szállítanak. A LabVIEW-ban az adatvezetékek részint a színükkel, másrészt pedig a mintázatukkal is jelzik, hogy rajtuk milyen típusú információ továbbítódik.

Ha be van állítva, akkor az adathuzalok menetét automatikusan is elkészíti a LabVIEW, de ha magunk szeretnénk meghatározni, akkor lehetőség van arra, hogy két elem között egy bal egérgattintást téve egy új töréspontot hozzunk létre a kábel menetében. Amennyiben ilyenkor rossz irányba törí a program a vezetéket, akkor a 'Space' billentyű lenyomásával meg tudjuk ezt változtatni.



Nagyon fontos, hogy egy bementi ponthoz csak egy forráspontot köthetünk, de természetesen egy adatforrás több függvény bemenetét is kiszolgálhatja. Ha egy vezetéken több forráspontból vezetünk adatot, vagy véletlenül két adatforrást kötnénk össze, akkor a vezetékek szaggatott fekete lesz, valamint egy piros 'X' lesz rajta látható (de ugyanez történik, ha nem kompatibilis adattípust szolgáltató illetve váró pontokat kötünk össze). Ez programhibát jelent, és mint ilyen a hibalistában is megjelenik, valamint a 'Context Help'-ben is olvashatunk róla információt, ha a kurzorral az adatvezeték fölé állunk.

A hibás vezetékeket, csakúgy mint a többi elemet, annak kijelölésével és a 'Delete' gomb lenyomásával törölhetünk. A vonalak szakaszonként (két töréspont között) jelölhetők ki egy szimpla bal kattintással, azonban ha duplát kattintunk, akkor az egész adatvezeték kijelölésre kerül. Ha egyszerre szeretnénk az összes 'Block Diagram'-on levő hibás vezetéket törölni, akkor azt a :Ctrl-B / :Alt-B billentyűkombinációval tehetjük meg.

Ha készen vagyunk az adatvonal létrehozásával, akkor a programunk úgy kell mutasson, mint az az I.22 ábrán is látható.

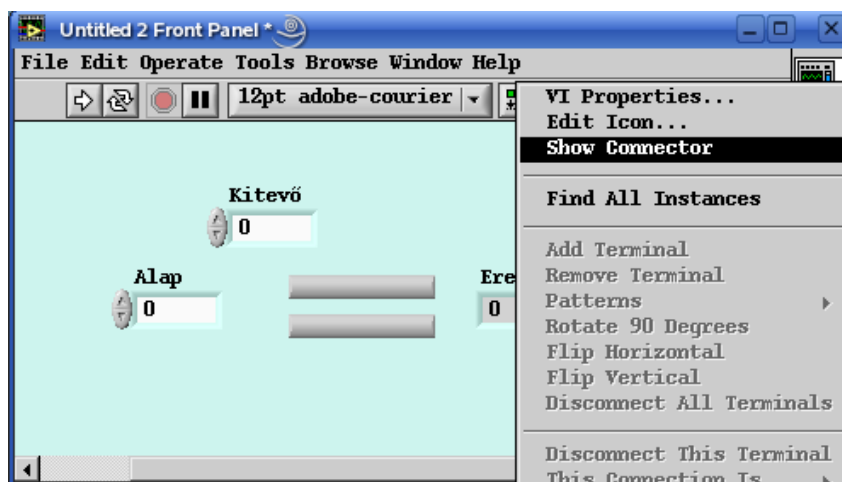


I.22. ábra: A kész hatványozó program

Ezzel sikeresen elkészítettünk első LabVIEW programunkat. A futtatást a  gombbal vagy az 'Operate' menü 'Run' menüpontjával, illetve az itt feltüntetett billentyűkombinációval tudjuk elindítani. Ha nem adtunk az „alap”-nak és a „kitevőnek”-nek valamilyen nullától eltérő értéket a  gombokkal, vagy a címkéző kurzorral a beviteli mezőbe belekattintva és valamilyen számot beírva, akkor egy igen érdekes dolgot láthatunk az eredmény mezőben, még hozzá azt, hogy az eredmény 'NaN'. Mit is jelent ez? Ez a 'Not a Number' azaz a „Nem Szám” rövidítése. Miért kaptuk ezt az eredményt? Gondoljunk bele, egy logaritmikus hatványozóprogramot írtunk! A logaritmikusfüggvény viszont nullára és negatív számokra nem értelmezett!

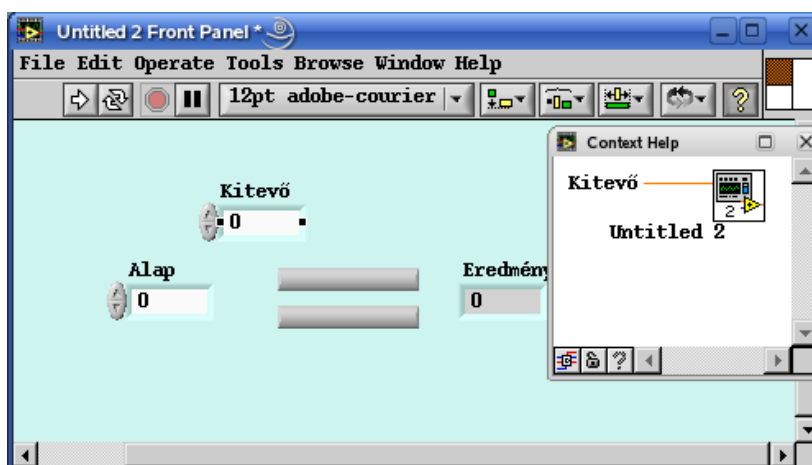
Ezt a programunkat a későbbiekben is fel kívánjuk majd használni egy másik alkalmazásban,

ezért készítsünk most belőle un. 'subVI'-t! Ebben a fejezet elején már említett 'Icon Connector' lesz a segítségünkre. Kattintsunk rá az egér jobb gombjával (Figyelem! Az 'Icon Connector' a 'Front Panel'-en található! (I.23. ábra)). Válasszuk ki a 'Show Connector' menüpontot. Ekkor az ikon helyett egy csatlakozópontokra felosztott négyzetet láthatunk. Ezekhez a pontokhoz rendelhetjük hozzá a 'subVI' be- illetve kimeneteit, melyek tulajdonképpen a programunkban létrehozott 'Control' illetve 'Indicator' elemeink.



I.23. ábra: Az 'Icon Connector' és legördülő menüje

A felkínált csatlakozók és kiosztásuk nem mindig megfelelőek. A legördülő menüben az ilyen problémákra többféle megoldás is található. Külön felhívnam a figyelmet a 'Patterns' pontra, ahol a különböző kiosztás típusok találhatóak. Nekünk most megfelel egy 2+1 konnektort tartalmazó kiosztásminta. Ha fölémegyünk a kurzorral, akkor az húzalozó típusúvá válik. Ilyenkor belekattintva a mezőbe, majd utána a hozzárendelni kívánt elemre, hozhatjuk létre a kapcsolatot a 'Front Panel'-en levő elem és az 'Icon Connector' csatlakozópontja között. Amennyiben ez a művelet sikeres volt, ez a 'Context Help'-ben megjelenik, amint azt az I.24. ábra is mutatja.

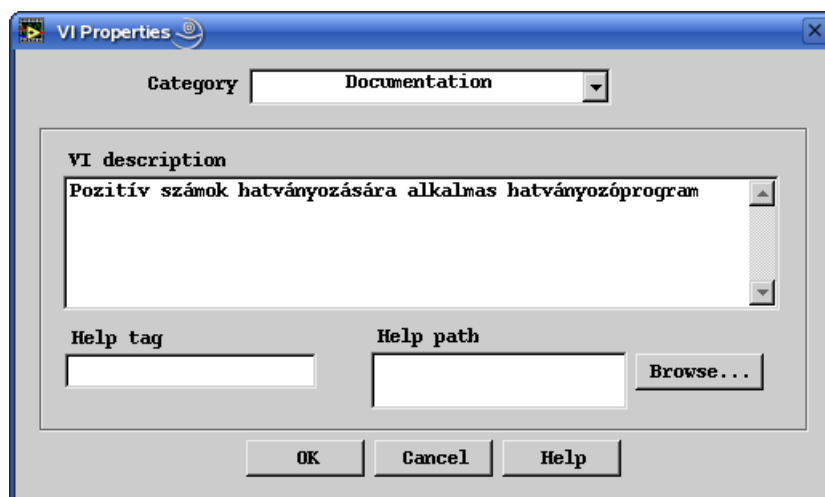


I.24. ábra: Csatlakozópontok definiálása

Kössük be a két bemenetet és a kimenetet. Ezek után még két dolgunk maradt, melyek nem szükségesek ahhoz, hogy működjön illetve 'subVI'-ként meghívható legyen az elkészített az elkészített program, de hozzátartozik a szép programozáshoz.

Az első, hogy készítsünk egy rövid leírást a programról, hogy amikor mi egy fél év múlva, vagy

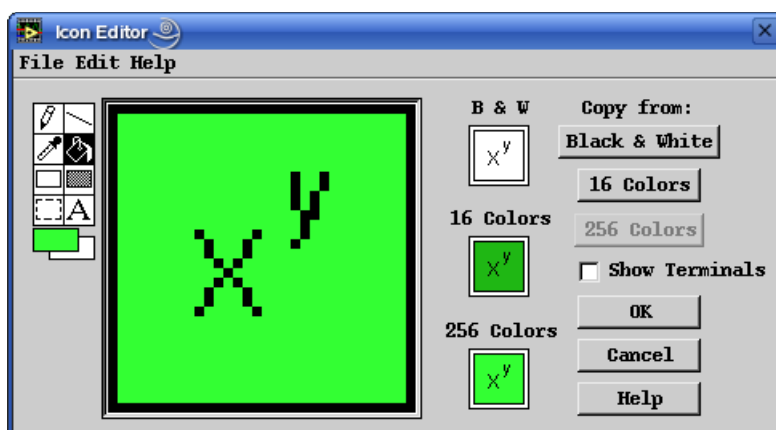
egy másik programozó fel kívánja használni a megírt VI-t, akkor tudjuk hogy mire való és esetleg milyen korlátai vannak az alkalmazásának (merthogy nem tudunk negatív számokat hatványozni vele). Ehhez válasszuk ki az 'Icon Connector' legördülő menüjéből a 'VI properties' pontot.



I.25. ábra: A VI leírása

A felbukkanó panelen a VI futási és egyéb tulajdonságait állíthatjuk be. Most nekünk a 'Category'-ák közül a 'Documentation' pontra van szükségünk, ahol a 'VI description' mezőbe beírhatjuk a program rövid ismertetését (I.25. ábra).

A következő feladat, hogy egy egyéni, a program funkciójára utaló ikon készítsünk a VI-unknak. Ezt az 'Icon Connector' legördülő menüjéből az 'Edit Icon' menüpont kiválasztásának következményeként megnyíló 'Icon Editor'-ral tudjuk megtenni (I.26. ábra).



I.26. ábra: Az 'Icon Editor'

Mint látható, ez egy egyszerű rajzolóprogram, amihez hasonló már mindenki használt, de van néhány sajátossága is. Az ikonokat három színmélységben rajzolhatjuk meg, és ezek között az adott színpalettát jelző ikonra kattintva válthatunk. A 'Copy from' felirat alatt levő gombokkal pedig az adott színelbontáshoz másolható át egy másik tartalma. A 'Show Terminals' mező bejelölésével az ikonon megjelenítésre a program csatlakozókiosztása. Jó még tudni azt is, hogyha az feliratkészítést lehetővé tevő kis ikonra egy dupla bal kattintást elhelyezünk, akkor egy 'Font Dialog' ablak jelenik meg melyben be tudjuk állítani a használni kívánt betűtípust.

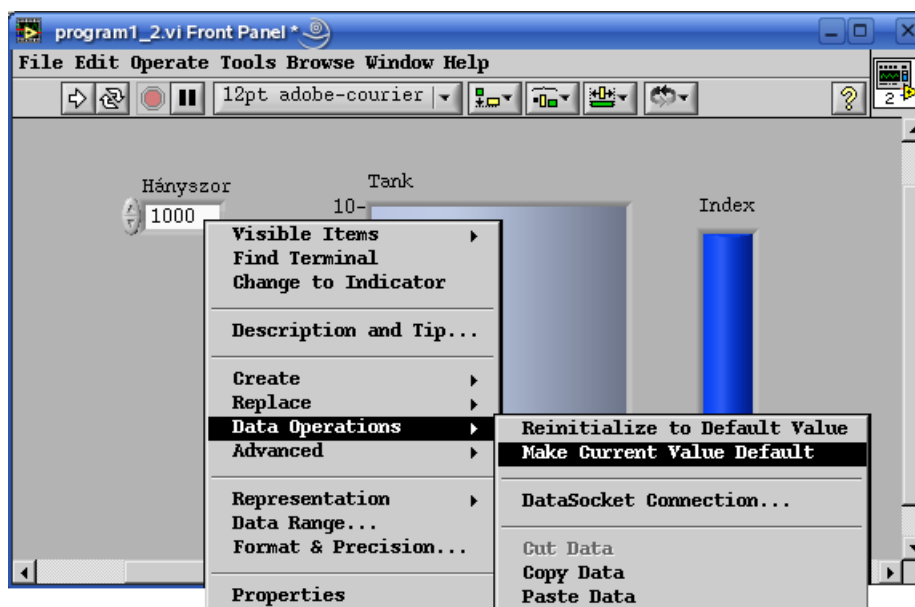
Ha elkészültünk az ikonnal, mentsük el a programot egy olyan helyre, ahonnan a következő fejezet feldolgozása során elő tudjuk majd venni. Ezután csukjuk be a VI 'Front Panel'-jét és nyissunk egy új 'Blank VI'-t!

2.-3. Példa: Tartály vízszintjének változása (program1_2.vi, program1_3.vi)

Mielőtt nekikezdenénk ennek a példának, ejtsünk néhány szót a LabVIEW-ban rendelkezésre álló két fajta ciklusról. Mindkettőt a 'Functions' paletta 'Structures' alpalettáján találjuk a középső sorban (I.28. ábra). A 'For Loop' egy olyan ciklus, amely a bemenetére meghatározott mennyiség szerinti fogja lefuttatni a benne található programkódot. A 'While Loop' (a palettán a 'For Loop' mellett van) egy hátulteszteléses ciklus, ami azt jelenti, hogy legalább egyszer le fogja futtatni a beleírt kódot, és utána megnézi, hogy teljesül-e a leállási feltétel, és ettől függően ismétli tovább az elvégzett műveleteket vagy leáll.

Most pedig térjünk rá a feladatra, melyből két megoldást is készítünk a különböző ciklusok felhasználásával. Tehát a programban változtassuk egy tartályban a benne lévő víz szintjét szinusz függvény szerint.

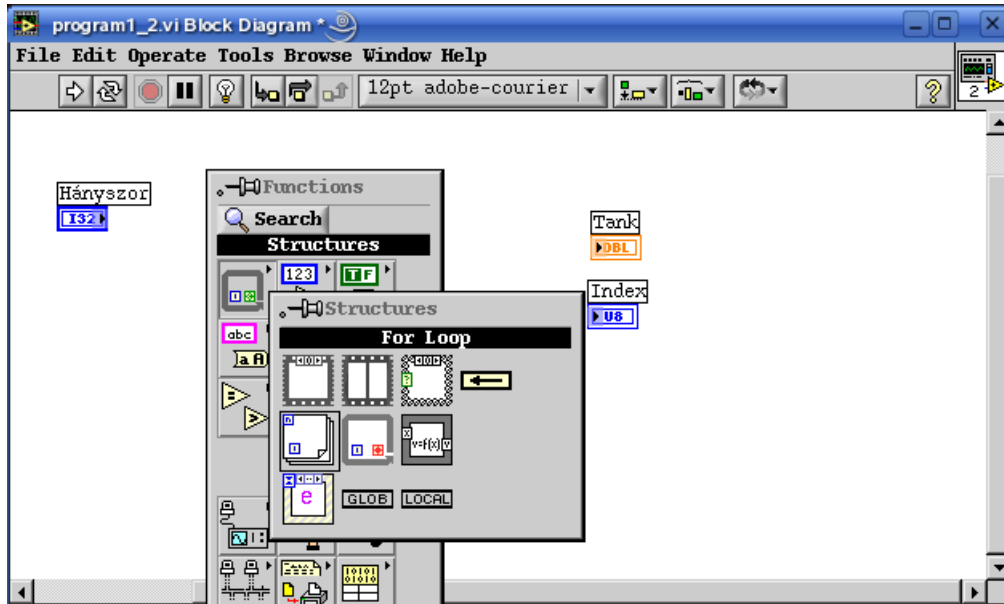
Az első megoldás során 'For' ciklust fogunk használni, úgyhogy meg kell határoznunk, hogy hányszor fusson le a ciklus. Ezenfelül hasznos lenne, ha valamilyen tájékoztatást kapnánk a program futása során, hogy hol tart a folyamat. Ezek alapján helyezzünk el a 'Front Panel'-en egy 'Digital Control'-t (amit nevezzünk el „Hányszor”-nak, lévén ezzel fogjuk beállítani, hogy hányszor fusson le a ciklus, azaz meddig lengessük a vízszintet a tartályban), egy 'Tank' elemet (mely a tartályt lesz hivatott megszemélyesíteni), illetve egy 'Vertical Progress Bar' elemet, mely a folyamatjelzőnk lesz (mindhárom elem a 'Numeric' palettán található).



I.27. ábra: Alapértelmezett érték beállítása

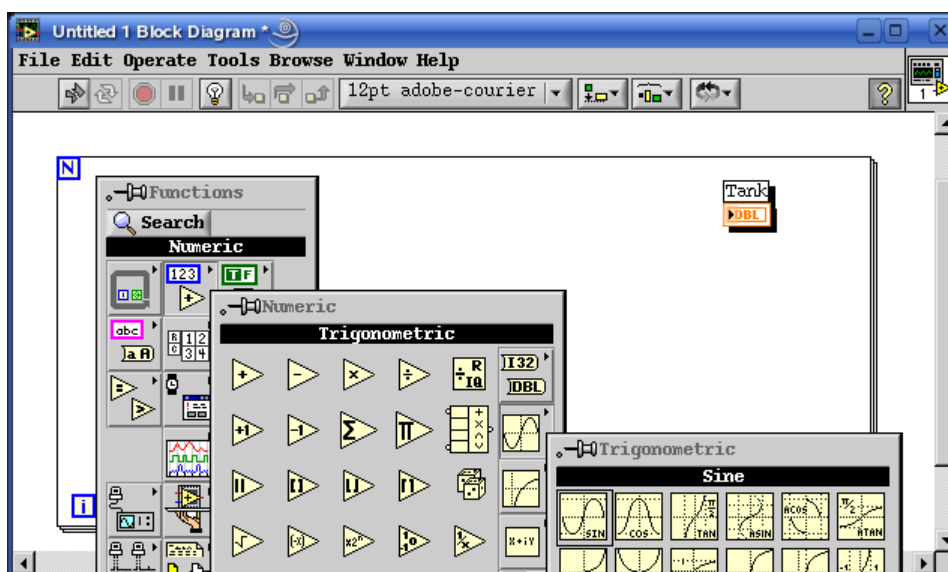
Belátható, hogy igen hasznos lenne, ha a „Hányszor” nevű beviteli mező alapértelmezett értéke nem nulla lenne, hanem mondjuk 1000. Ezt úgy tudjuk elérni, hogy a kontroll értékét 1000-re állítjuk, majd a hozzá tartozó legördülő menüből kikeressük a 'Make Current Value Default' pontot, ahogy azt az I.27. ábra mutatja. Így valahányszor megnyitjuk ezt a programot, egy

értelmes érték lesz ebben a mezőben. Másrészt a 'Reinitialize to Default Value' menüpont segítségével bármikor visszaállíthatjuk ezt az értéket. Ha a fent említett két műveletet az összes a 'Front Panel'-en található elemre akarnánk elvégezni, akkor az 'Operate' menü 'Make Current Values Default' valamint a 'Reinitialize All to Default' pontokat használhatjuk.



I.28. ábra: A ciklusok lelőhelye a 'Function Palette'-n

Térjünk most át a 'Diagram Panel'-re. Először is vegyünk le egy 'For Loop'-ot a palettáról (I.28. ábra). A 'Diagram Panel'-re úgy tudjuk feltenni, hogy egy bal egérgattintással meghatározzuk a bal felső sarkának a helyét, majd a megjelenő szaggatott keretet a kívánt méretre nagyítjuk. Próbáljuk ezt most úgy megtenni, hogy a két kijelzőnk terminálja belekerüljön a ciklusba. Ha nem így sikerült, akkor sincs gond, egyszerűen megfogjuk a kívánt elemet és behúzzuk a ciklus területére. Amennyiben egy terminál vagy más elem a ciklus területén van, de nincs benne a ciklusban, akkor a LabVIEW egy árnyékot rajzol alá, mintha az adott elem lebegne (I.29. ábra).

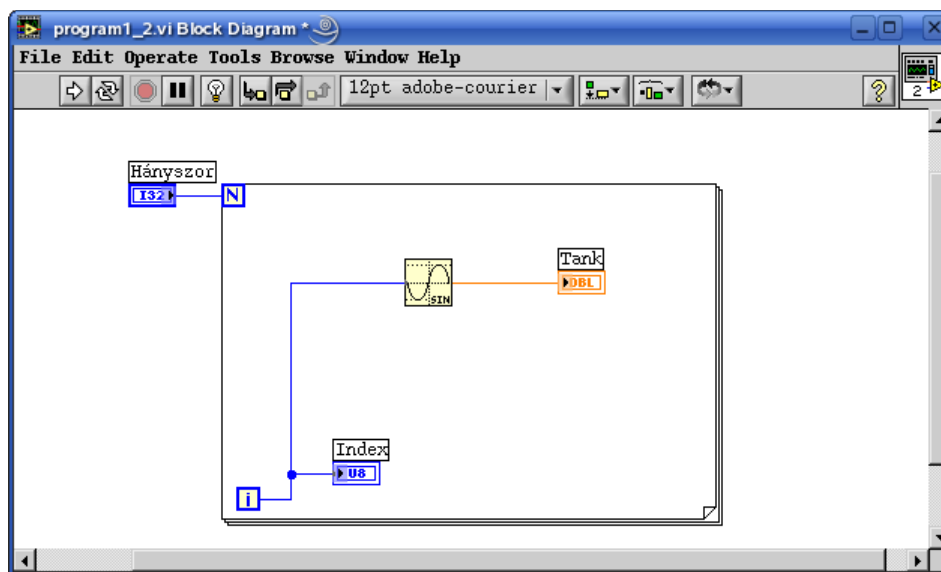


I.29. ábra: A szinusz függvény lelőhelye

A 'For' ciklusnak két igen fontos eleme van. Az egyik a ciklus végérték **N**, amelybe bele kell vezetékelnünk azt a változót vagy konstans, amelyik megmondja, hogy hányszor fusson le a ciklus. A másik a ciklusváltozó **i**, amely azt mutatja, hogy épp hányadik lépésben tartunk. FIGYELEM! A ciklusváltozó értéke 0 és N-1 között változik!

Következő lépésként keressük meg a szinusz függvényt, ami a 'Numeric' paletta 'Trigonometric' alpalettáján található (I.29. ábra) jó néhány más trigonometrikus függvény társaságában. Helyezzük el a szinusz függvényt a ciklusban, és a . ábrának megfelelően végezzük el programozást.

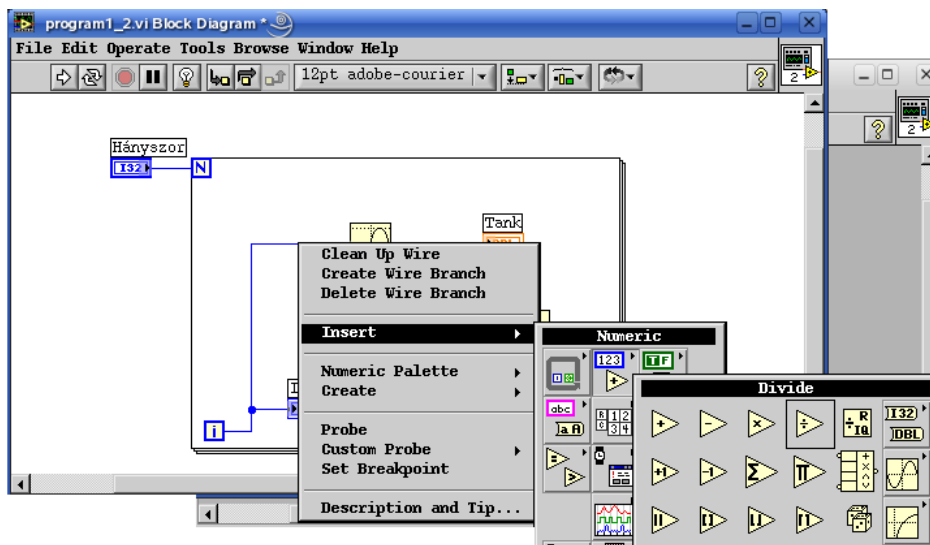
Ha ezzel elkészültünk, a futtatás gombbal  indítsuk el az elkészült programot, és fogalmazzuk meg a véleményünket a látható eredménnyel kapcsolatban!



I.30. ábra: A második program 'Diagram Panel'-je

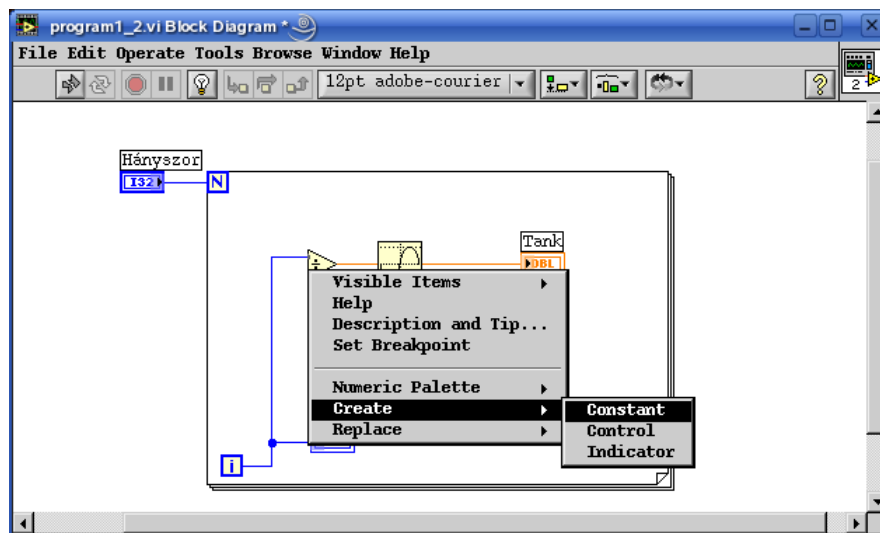
Akkor most tekintsük át, hogy milyen tüneteket észlelhettünk, melyek azok a hibák illetve figyelmetlenségek, amelyeket elkövettünk a programozás során, és hogyan lehet őket kiküszöbölni.

- A tartályban ugrál a vízszint, ami a legkevésbé sem hasonlít egy szinusz függvény szerinti mozgásra.
 - Ennek három oka van. Az első, hogy nem figyeltünk oda arra, hogy milyen adatot vár a bemenetén a szinusz függvény. Azonban, ha megnézzük a 'Context Help'-jében (ugye folyamatosan látható a 'Context Help' ablak a programozás során!?!), rögtön kiderül, hogy radiánban megadott értékeket vár, mi pedig egész értékekkel etettük. A probléma korrigálásaként osszuk el a szinusz függvény bemenetére érkező értéket 100-zal. Ezt a feladatot két módon is elvégezhetjük. Az első, hogy leveszünk a 'Numeric' palettáról egy osztás subVI-t, a jelenlegi adatvezetékét a ciklusváltozó és a szinusz függvény között töröljük (kijelölés, majd 'delete' billentyű), és ezek után bekötjük újra a vezetékeket. A másik mód, hogy a vezeték azon pontján nyomunk az egérrel egy jobb klikket, ahova el szeretnénk helyezni az osztás műveletet és a legördülő menüből az 'Insert' pontot kiválasztva illesztjük be az osztást (I.31. ábra). Ennek az a hátránya lehet, hogy az osztásnál nem mindegy, hogy mit osztunk mivel, és a beillesztés során nem biztos, hogy a jó bemenetbe lesz a ciklusváltozó értéke bekötve.



I.31. ábra: 'SubVI' beillesztése az adatfolyamba

Az osztót is két módon tudjuk a 'Diagram Panel'-re varázsolni. Az egyik út, hogy a 'Numeric' palettáról leveszünk egy 'Numeric Constant' elemet, az értékét 100-ra állítjuk és bekötjük. A másik lehetőség, hogy az osztás művelet ikonján az osztó bemenete felett egy jobb egérgattintással előhívjuk az ide tartozó legördülő menüt, és a 'Create' pontból a 'Constant'-ot kiválasztva (I.32. ábra) egy konstans készítünk az adott bemenethez, amelynek már csak az értékét kell beállítani. (Felhívnam a figyelmet arra, hogy hasonló módon tudunk beviteli/kontroll illetve kijelző/indikátor elemet is készíteni egy be- illetve kimenethez!)

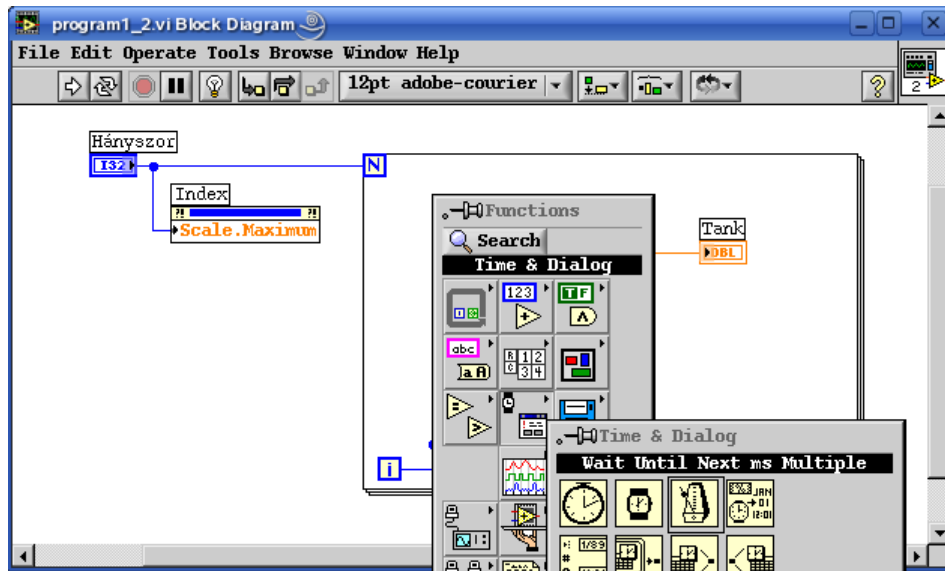


I.32. ábra: Konstans készítése egy bemenethez

- A másik oka a nem megfelelő vízszintábrázolásnak az, hogy a megjelenítés valamint a program futási sebessége különböző. Ezt úgy tudjuk megoldani, hogy a 'Functions Palette' 'Time & Dialog' alpalettájáról a 'Wait Until Next ms Multiple' VI-t elhelyezzük a cikluson belül (I.33. ábra), és egy konstans 10 értéket kötünk a bemenetére a fent említett két módszer valamelyikével, minek hatására a

ciklusunk minden lefutás után 10 milliszekundumot vár mielőtt újra elindulna.

- A harmadik ok pedig az, hogy a szinusz függvény értékkészlete a $[-1,1]$ zárt tartomány, a „Tank” nevezetű tartályunk pedig jelenleg a $[0-10]$ tartományban jelez ki értékeket. Ezt korrigálendő állítsuk át a skálát a tartályon oly módon, hogy a legalsó illetve legfelső értéket írjuk át a megfelelő értékekre.

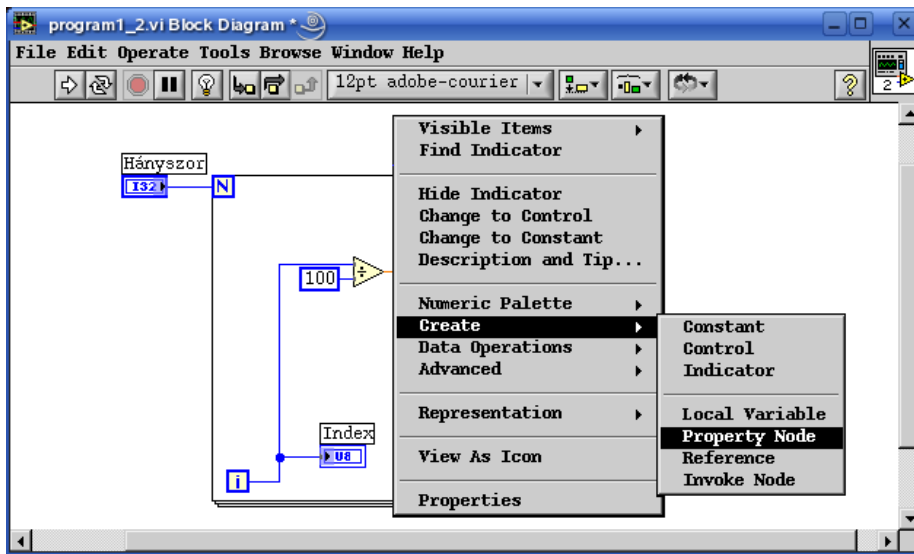


I.33. ábra: Ciklus futásának készítése

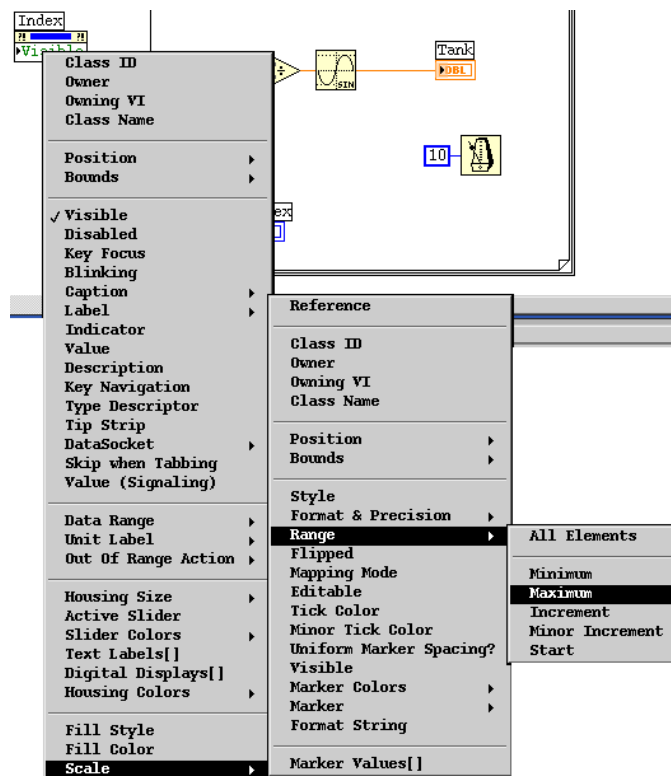
- A folyamatjelző csúszka túl hamar betelik, és aztán időnként nullázza magát.
 - A nullázódás oka az, hogy a csúszka alapértelmezetten U8-as, azaz 8 bites előjel nélküli típus, ami 0-255 – ig tartalmazhat értékeket, márpedig mi 1000 ciklust futtatunk le, ami kétségtelenül nagyobb 255-nél. Úgyhogy állítsuk át a kijelzőnk típusát U32-re (az elemhez tartozó legördülő menü 'Representation' pontjában).
 - Az idő előtti telítődés oka, pedig az elem rossz skálázása (bár a skála alapértelmezésben nem látszik ezen az elemen). Azonban most nem járhatunk el úgy, mint a tartály esetében, mivel a skálának mindig olyan maximális értékűnek kell lennie, ahány ciklust le akarunk futtatni.

Ezt a feladatot az úgynevezett 'Property Node' segítségével tudjuk megoldani. A 'Property Node' egy olyan LabVIEW elem, amely arra való, hogy az egyes kijelzők, beviteli elemek tulajdonságait megváltoztathassuk a program futása során. Egy elem 'Property Node'-ját a hozzá tartozó legördülő menü 'Create' menüpontjában a 'Property Node'-ra való kattintással hozhatjuk létre (I.34. ábra). A 'Property Node'-ok hasonlóan a LabVIEW-ban található elemekhez lehetnek írhatóak vagy olvashatóak (átváltani az egyik típusról a másikra a legördülő menüjükön keresztül lehet).

Hozzunk tehát létre a folyamatjelzőnkhez egy 'Property Node'-ot. Ez olvasható típusú lesz, úgyhogy váltsuk át a legördülő menüjében levő 'Change All to Write' pont segítségével írható típusúra. Ha ez megvan, akkor ki kell választanunk a megfelelő tulajdonságot, amit módosítani szeretnénk. Ez vagy a 'Property Node' legördülő menüjének 'Properties' menüpontjából érhető el, vagy pedig az elemen, amikor a kurzor működtető típusúvá (☞) válik a bal egérgombbal kattintva választhatjuk ki. Minket most a 'Scale' – 'Range' – 'Maximum' pont (I.35. ábra) érdekel. Ide a „Hányszor” nevű változót kell bekötnünk.

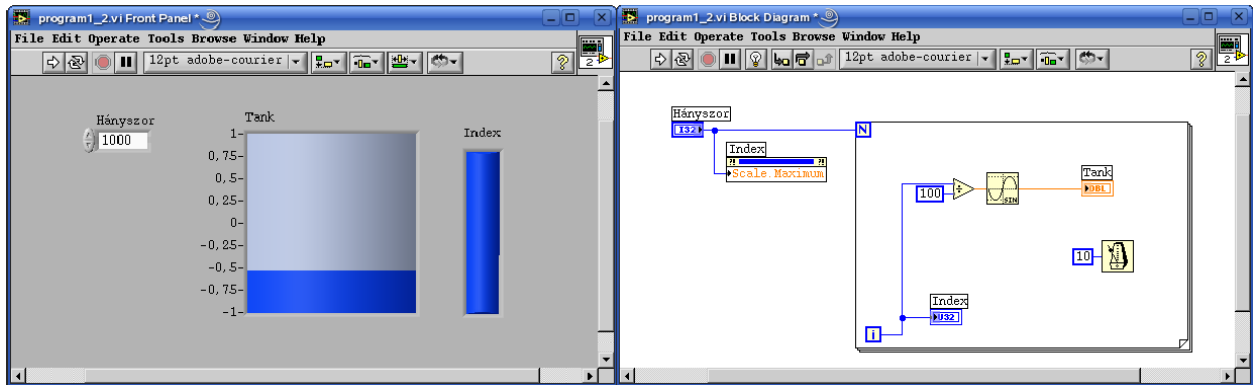


I.34. ábra: 'Property Node' létrehozása



I.35. ábra: A skála maximumának kiválasztása

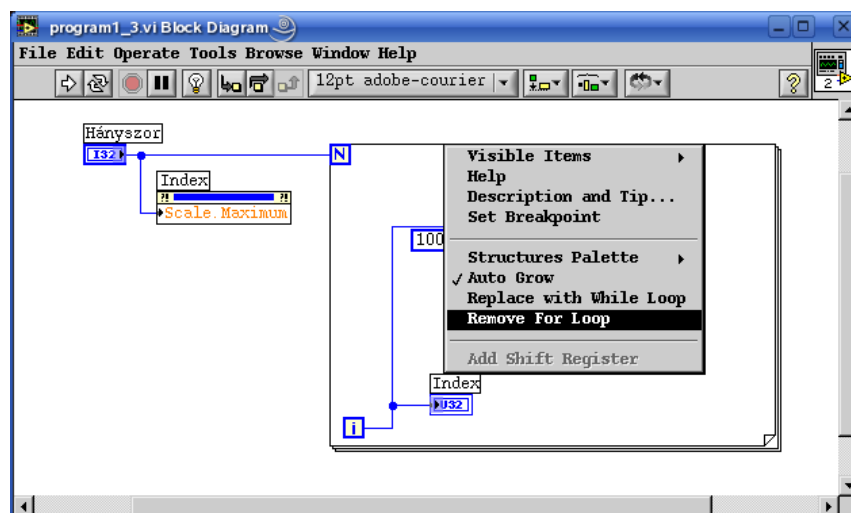
Amennyiben jól dolgoztunk, akkor a programunk most már az alapvető elvárásoknak megfelelően fog működni, és a 'Front Panel'-jének valamint a 'Diagram Panel'-jének hasonlóan kell mutatni, mint az az I.36. ábrán is látható.



I.36. ábra: Az elkészült 2. program

A harmadik példaprogramunk tulajdonképpen az előző program módosítása úgy, hogy 'While' ciklussal működjön. Elkezdhetnénk előlről is a program írását, de ehelyett most inkább mentjük el új néven a már elkészített második programot.

A 'While' ciklussal működő változatban nem lesz szükségünk a „Hánszor” nevű beviteli elemre, hiszen egy más típusú feltételtől függően akarjuk majd leállítani a program futását (habár megtehetnénk, hogy egy megadott ciklusszám elérésekor állítjuk le, de arra a 'For' ciklus használandó). Ez a feltétel egy gomb megnyomása lesz, ezért a folyamatjelzőre sem lesz szükség, hiszen nem tudjuk megmondani egy adott pillanatban, hogy meddig fog még futni a program. Ezt a két elemet tehát töröljük (több elemet a 'shift' billentyű lenyomva tartása mellett tudunk kijelölni). Látható, hogy a program ebben a pillanatban nem futtatható. A hibalista gomb (🔍) megnyomásával előhozható a lista a programban lévő hibákról. Az egyik ilyen hiba a semmiből induló, vagy ott végződő huzalok a 'Diagram Panel'-ben. Ezeket egyenként kijelölve is el lehet távolítani, de van erre egy sokkal gyorsabb módszer, méghozzá a :Ctrl-B / :Alt-B billentyűkombináció alkalmazása. A 'For' ciklust ne töröljük egyszerű kijelöléssel, hiszen akkor a benne lévő tartalom is elveszik (és ez igaz minden struktúrára a LabVIEW-ban!), hanem helyette nézzük meg, hogy milyen lehetőségeink vannak, ha 'For' ciklus szélén elhelyezett jobb egérgombnyomással próbálkozunk.



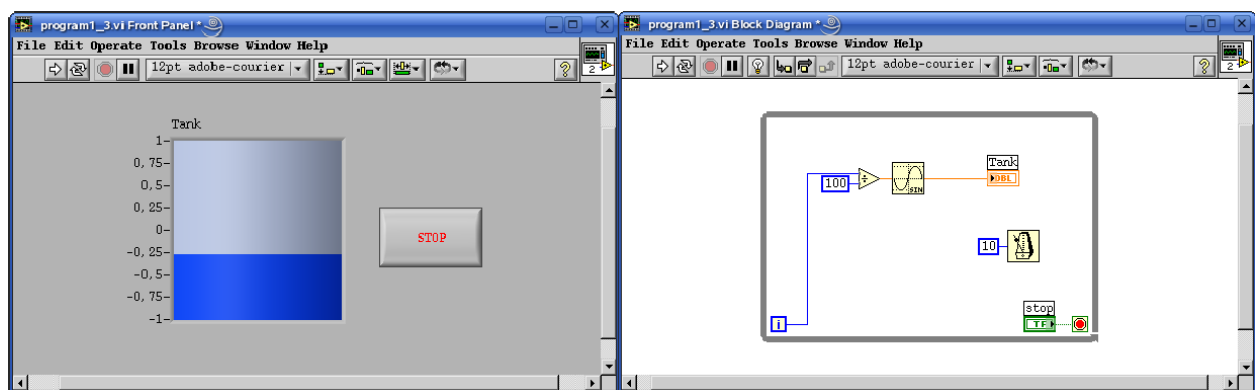
I.37. ábra: A 'For' ciklus eltávolítása

Két megoldás közül választhatunk. Az első, hogy a 'Remove For Loop' menüpont választásával eltávolítjuk a ciklust, és egy új 'While' ciklust helyezünk el a programban. A másik, hogy ez előbb említett menüpont felett levő 'Replace with While Loop' pontot választjuk, miáltal a cikluscsera egy lépésben végrehajtható (I.37. ábra).

A program még mindig nem futtatható, hiszen hiányzik a leállási feltétel, ami 'Boolean' típusú kell legyen, és a leállási feltétel (🛑) termináljába kell csatlakoztatni. Ez a feltétel két típusú lehet: leállítás, ha a feltétel igaz (🛑); illetve leállítás, ha feltétel hamis (🛑). A két típus között a váltás a terminálon végzett jobb kattintásra előbukkanó menün keresztül lehetséges, vagy pedig működtető típusú kurzorral a terminálba kattintással.

Nekünk most az első típusra lesz szükségünk. A terminál felbukkanó menüjéből válasszuk ki a 'Create Control' pontot, mely hatására egy 'Stop' gomb keletkezik a 'Front Panel'-en (ezt a műveletet is elvégezhetnénk úgy is, hogy a 'Front Panel'-en a 'Controls' paletta 'Boolean' alpalettájáról kiválasztjuk a 'Stop' gombot, és a panelen való elhelyezése után a 'Diagram Panel'-en bekötjük a megfelelő helyre). A gombot méretezzük át, és távolítsuk a címkéjét, ha látszik.

Amennyiben ezekkel a feladatokkal elkészültünk, máris rendelkezésre áll és működőképes az új, immár 'While' ciklussal működő program, melynek a 'Front Panel'-je és 'Diagram Panel'-je az I.38. ábrán látható.



I.38. ábra: Az elkészült 3. program