# 1. A LabView működése

Mivel a LabView számítógéppel vezérelt mérő és adatfeldolgozó rendszerek programozására szolgál, a gyártó National Instruments a "virtual instrumentation" kifejezést alkalmazza a rendszer jellemzésére. Ennek megfelelően a LabView-ban készült programok a "virtual instrumentation" kifejezésből rövidített \*.vi kiterjesztést kapják. Meg kell azonban jegyezni, hogy a "virtuális" kifejezés némileg megtévesztő abban az esetben, amikor valós műszereket és hardver rendszereket alkalmazunk, mert ilyenkor a LabView-t nem virtuális, hanem nagyon is valóságos berendezések működtetésére alkalmazzuk, igaz, virtuális felületen. Ennek megfelelően nem összekeverendő a virtuális műszer a valóságos, de számítógépről programozható műszerrel, mint ahogyan nem összekeverendő a virtuális laboratórium a távolról vezérelhető, valóságos laboratóriummal. Viszont a LabView valóban lehetőséget kínál virtuális műszerek kezelésére is (azaz nem valóságos műszereket kezelünk, hanem emulált műszereket), ezt a későbbiekben a DAQmx mérésadatgyűjtő rendszer tanulmányozásánál láthatjuk majd. Ezzel a lehetőséggel a szakemberek, egyetemi hallgatók a drága adatgyűjtő kártyák hiányában is, virtuálisan alkalmazhatják a National Instruments adatgyűjtő eszközeit, és sajátíthatják el akár otthon is a mérésadatgyűjtés kezdeti lépéseit...de erről majd később beszélünk.

A *vi* programok speciális könyvtárakba, úgy nevezett "VI Library" -be rendezhetőek. A VI library a fájl kezelőkben "llb" kiterjesztéssel, LabView LLB típusú önálló fájlként jelenik meg, de LabView környezetben egy VI library több *vi* programot is tartalmazhat. A VI programokat nem kell kötelezően LabView Library-ban tárolni, lehet önállóan, különálló fájlként is kezelni, a vi könyvtár egy lehetőség az összetartozó állományok együttkezelésére.

Ha LabView-ban programozunk, akkor a legegyszerűbb esetben 2 ablakot használunk: a felhasználói kezelőfelület ablakát (Front Panel) és a grafikus program ablakát (Block Diagram).

#### Feladat:

Indítsuk el a LabView programot! A kezdő ablakon válasszuk a "Blank VI" menüpontot. Ekkor megjelenik az előző bekezdésben említett két ablak. A fehér felületű ablak a Block Diagram, a szürke felületű ablak a Front Panel.

## 1.1. A felhasználói felület, a Front Panel

Általánosságban tudjuk, hogy amikor egy windows alkalmazást futtatunk, azaz elindítunk egy "exe" kiterjesztésű állományt, akkor megjelenik a képernyőn egy felhasználói ablak, amelyen műveleteket végzünk, majd egy kilépésre szolgáló menüponttal/nyomógombbal, stb. ki tudunk lépni a programból. Ezt a felhasználói felületet tartalmazza LabView-ban a Front Panel. A Front Panelen vannak elhelyezve a programot vezérlő nyomógombok, kapcsolók, (pl. kilépés a programból, mérés indítása, file-ba mentés, visszatöltés, stb.), értékmegadó mezők (pl. mintavételi frekvencia beállítása, csatornák számának megadása, stb.), különböző kijelzők, grafikonok, vagy legördülő típusú menük, vagyis amit a program használója lát. A program fejlesztése közben a LabView fejlesztő környezetéből történő futtatás során is ezen a képernyőn tesztelhető a program.



1. ábra Felhasználói kezelőfelület

## **1.2.** A grafikus programozás felülete, a Block Diagram

A Panelen elhelyezett objektumok névvel ellátva automatikusan megjelenek a grafikus program(Block Diagram) ablakában. Itt tervezzük meg a program futását. Tulajdonképpen olyan ez, mint a szokásos programnyelvekben (Pascal, C) a szöveges formátumú forrásnyelvű program, csak itt nem a szövegesen beírt sorok futnak le sorba egymás után, hanem a grafikus jelekkel meghatározott műveletek, függvények, a köztük lévő vezetékezésnek megfelelően.

A Block Diagram-on az objektumokat kétféleképpen jeleníthetjük meg, nagyobb, vagy kisebb ikonokkal. A rendszer alapbeállításban nagy ikonokkal dolgozik. Az egyes ikonokat az egér jobb billentyűjével előhívható menüben, a "View as icon" menüpont pipájának törlésével átalakíthatjuk kicsire.

Ha a programunkat a fejlesztés elejétől kis ikonokkal kívánjuk megjeleníteni, akkor a LabView beállításinál kell megváltoztatni a megjelenítést az alábbiak szerint:

Tools – Options... - Block Diagram – General – Place front panel terminals as icons melletti jelölőnégyzetből a pipa törlése.



2. ábra Az ikonok megjelenítésének módosítása



3. ábra Grafikus program LabView-ban nagy ikonokkal



4. ábra Grafikus program LabView-ban kis ikonokkal

# 2. A LabView kezelése

Csak a Windows környezetben megszokott funkcióktól eltérő menük kerülnek bemutatásra, a teljesség igénye nélkül, kezdők számára.

A fejlesztő rendszer kétféle üzemmódban működtethető, szerkesztő és futtató módban. Szerkesztő módban felépíthetjük a Front Panelt és felprogramozhatjuk a Block Diagramot, míg futtató módban tesztelhetjük az elkészült programot.

**FONTOS FIGYELMEZTETÉS!** A program futtatásakor mindig vegyük figyelembe és gondoljuk át, hogy a felhasználó a kész alkalmazás futtatását megfelelő módon tudja-e majd kezelni. Például, ha a futtatás előtt be kell valamit állítani a Front Panelen ahhoz, hogy megfelelően működjön a program, akkor az az exe kiterjesztésű alkalmazás futtatásakor nem lesz lehetséges, vagy ha a program leállítása után át kell valamit állítani ahhoz, hogy újra futtatható legyen a program, az exe alkalmazásnál megint csak lehetetlen. Ezért lehetőség szerint kerüljük a folyamatos futtatás (Continuously running 🕑 ) menü vagy a vészleállító (Abort execution ) menü alkalmazását.

# 2.1. A fejlesztő környezet menüjének ismertetése



A program futtatása ezzel az ikonnal történik,



a nyíl futás közben átvált feketére,

ha a nyíl összetöredezett képet mutat, akkor a program nem futtatható, mert hibás. Ilyenkor az egérrel a nyílra kattintva kapunk egy hibalistát. A listán bármely hibára rákattintunk, részletes leírást ad a kiválasztott hibáról, és a *Show Error* gombot megnyomva, a helyét is megmutatja a diagramon.

A program folyamatos futtatása. Ha a program a végére ér, kezdi elölről. Alkalmazását nem javaslom, legfeljebb különleges programtesztelés esetén.

Vészgomb, a program vészleállítására. Üzemszerűen a program a Front Panelon leállítható kell legyen, ezért ezt a gombot csak akkor használjuk, ha a program leállító gombjával ez nem sikerül. (pl. végtelen ciklust hoztunk létre)



pillanatmegállítás a program futása közben

Csak a Block Diagram ablakban találjuk meg, ha futtatás közben a "lámpát" bekapcsoljuk, akkor vizuálisan követhetjük a program futását a grafikus programban. Láthatjuk, mikor melyik változó, milyen értéket vesz fel és továbbít. Ez a függvény kiválóan alkalmazható a hibakereséseknél, a program debuggolásakor.

Hasznos menüpont lehet az *Edit* menü *Remove Broken Wires* pontja, ami a CTRL-B gyorsbillentyűvel is elérhető.

Szintén hasznos lehet a CTRL-T billentyű kombináció, amellyel a Front Panel és a Block Diagram a képernyőn egymás mellé kerül, és mindkettő láthatóvá válik.

## 2.2. Tools Palette

Szerkesztés közben mindig legyen a képernyőn. Megjeleníthető a *View - Tools Palette* menü kiválasztásával. A program kezelését segíti különböző típusú és funkciójú kurzorokkal. A különböző kurzorok segítségével a kezdeti értékek átírhatóak, az objektumok átnevezhetőek, átméretezhetőek, áthelyezhetőek, átszínezhetőek, stb.

A legfontosabb eszközök a kezeléshez:

A kezdeti érték megváltoztatása, nyomógomb átkapcsolása szerkesztő üzem közben.

Általános kurzor az objektumok áthelyezésére, átméretezésére szolgál. Ezzel a kurzorral az objektumok cím mezője az objektumtól függetlenül is mozgatható.



Objektumok megnevezéssel együtt történő áthelyezésére szolgál.

A Bármilyen szöveg átírása. Akár menüsorban, akár címben, akár számérték mezőben ezzel átírható a beállított szöveg vagy érték.

Ezt a kurzort csak a Diagram ablakban használjuk, az objektumok összekapcsolására. Ezzel a "vezetékkel" jelöljük ki a program futásának sorrendjét.

<mark>-</mark> /

Színek megváltoztatása.

A program futása közben megszakítási pontok (breakpoint) elhelyezése. Ezt a kurzort csak a Block Diagram ablakban használjuk.

A beállított értékek csak akkor kerülnek elmentésre, tehát akkor fogja a program defaultként kezelni, ha az *Operate – Make Current Values Default* menüpontra lépünk.

## 2.3. A Front Panel könyvtárai

A Front Panelre a vezérlő palettáról (Controls Palette) választhatjuk ki a programunkhoz szükséges vezérlő és kijelző objektumokat. A vezérlő paletta megjeleníthető a jobb egérgombbal, vagy a *View – Controls Palette* menüpont kiválasztásával. A továbbiakban a Control Palette "Modern" könyvtárának objektumait fogjuk alkalmazni, ezért ez kerül bemutatásra.



5. ábra Controls Palette

## 2.3.1. Vezérlők és kijelzők

A fenti objektumok mindegyike lehet vezérlő vagy kijelző. Minden objektumot átkonfigurálhatunk, a kijelzőből vezérlőt csinálhatunk és viszont, ha az objektum fölött az egér jobb gombjával előhívjuk az editáló ablakot, és kiválasztjuk a *Change to Control/Indicator* menüpontot. A vezérlő és kijelző típusú objektumok a Diagramon különböző képpel jelennek meg. A vezérlők kerete vastag vonal, a kijelzők kerete vékony vonal. Az adatfolyam irányának megfelelően a vezérlőkből az ikon jobb oldalán kifelé mutat nyíl, a kijelzők ikonján pedig a bal oldalon befelé mutat a nyíl, jelezve, hogy a vezérlő adatot küld, a kijelző pedig adatot fogad.



## 2.3.2. A vezérlő palettáról választható legfontosabb objektumok:

A paletta felső sora:

	Modern	
	Image: Sector of the sector	
Numeric	Boolean 🖾	String & Path
☆ Search % Customize*	☆ Q Search & Customize▼	☆ Q Search & Customize▼
123 123 1200 1200	000	abc abc v
Numeric Co Numeric Indi Time Stamp Time Stamp	Push Button Rocker Vert Rocker	String Control String Indica Combo Box
10- 10- 5- 5-	• • 6	Path Path
ومن معنى معنى معنى معنى معنى معنى معنى مع	Round LED Horizontal T Vertical Tog	File Path Co File Path Indi
· · · · · · · · · · · · · · · · · · ·	<b>— — —</b>	
Horizontal Fi Horizontal P Horizontal Pr Horizontal G	Square LED Slide Switch Vertical Slide	
5 5 V	OK CANCEL STOP	
Knob Dial Meter Gauge	OK Button Cancel Button Stop Button	
100-0 5-000-0 50-00000000000000000000000	0	
Tank Thermometer Horizontal S Vertical Scrol	Radio Buttons	
8		
Framed Colo		
számértékek kezelése	kétállapotú egységek	szöveg kezelő mezők

6. ábra A Controls Palette felső sorából választható objektumok

Amennyiben a Boolean típusú egység vezérlő típusú, akkor meghatározhatjuk a kapcsolás módját:



7. ábra Kétállású vezérlő kapcsolók kapcsolási módjai

Funkciók:

- Megnyomásra kapcsol, és bekapcsolva marad (Switch When Pressed)
- Elengedésre kapcsol, és bekapcsolva marad (Switch When Released)
- Addig tart bekapcsolva, amíg a gombot lenyomva tartjuk (Switch Until Released)
- Megnyomásra bekapcsol és a következő órajelre kikapcsol (Latch When Pressed)

- Megnyomás után, elengedésre bekapcsol és a következő órajelre kikapcsol (Latch When Released)

- Megnyomásra bekapcsol, és a gomb elengedését követő órajelig bekapcsolva marad (Latch Until Released)

A paletta második sora:

		Graph			8
Modern	8	순 Q Search	& Customize <b>*</b>		
1 Search & Customize*					
	abc Path	Waveform C V	Vaveform G	XY Graph	Ex XY Graph
Numeric Boolean Strin	g & Path	Intensity Chart In	ntensity Gra Di	gital Wave	Mixed Signal
Array, Matrix List, Table & G	iraph		r tr		
		Compass Plot	Error Bar Plot F	eather Plot	XY Plot Matrix
		Controls		3D Picture	3D Graph
Array Matrix & Cluster					
	Contraction of the search of t	ustomize			8
		ABC			
Array Cluster	Listbox Multico	III IIIIII Iumn Table	Tree	Ex Tabl	e
RealMatrix.ctl ComplexMat					
Error In 3D.ctl Error Out 3D					
······································		c.	"1 1 (AD		

tömbök megadása

táblázatok/listák

grafikonok (2D és 3D)

8. ábra A Controls Palette második sorából választható objektumok

A grafikonok alkalmazása igen széles lehetőségeket biztosít, ezért a grafikonok könyvtárát részletesebben is bemutatjuk.

### Grafikonos megjelenítés lehetőségei LabView-ban

A felsorolás a teljesség igénye nélkül készül, csupán azokra a lehetőségekre mutat rá, amelyek a LabView bevezető kurzusain előfordulnak.

Graph			×
🕆 🔍 Search	🖉 🔦 Customiz	e▼	
$\langle \ \rangle$	Wavefor	m Chart	
Waveform C.	Waveform G	XY Graph	Ex XY Graph
2 1 2 2 1 2 2 1 2 2 2 2 1 2 1 2 1 2 1 2		2 <b>31212</b> 1 <b>31217</b> 0 500 1 10	
Intensity Chart	Intensity Gra	Digital Wave	Mixed Signal
	[ <sup>1</sup> 1 <sup>1</sup> 1]	<u>الجر</u>	
Compass Plot	Error Bar Plot	Feather Plot	XY Plot Matrix
		20	
Controls		3D Picture	3D Graph

9. ábra Waveform Chart kiválasztása

A *Waveform Chart* egy folyamatosan "futó" grafikus megjelenítő, amelyen akár pontonként jeleníthetők meg az értékek, és mint egy regisztráló működik.

Graph			E
🕆 🔍 Search	🔦 Customiz	e▼	
	Wayefor	m Graph	
2 1 - Age- 8 60 - 10			
Waveform C	Waveform 6	XY Graph	Ex XY Graph
2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		2 <b>31212</b> 1 <b>31212</b> 0 <b>3121</b> 0 <b>312</b>	
Intensity Chart	Intensity Gra	Digital Wave	Mixed Signal
÷	T T T	<b>K</b>	
Compass Plot	Error Bar Plot	Feather Plot	XY Plot Matrix
		20	
Controls		3D Picture	3D Graph

10. ábra Waveform Graph kiválasztása

A Waveform Graph a "hagyományos megjelenítője a hullámformáknak. Egy tetszőleges típusú tömb értékeit jeleníti meg pontonként úgy, hogy a vízszintes tengelyen a pontok sorszáma látható.

Graph			B
🔒 🔍 Search	n 🔌 Customiz	e▼	
	XY G	iraph	
2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		(團)	
Waveform C	Waveform G	XY Graph	Ex XY Graph
2 1 9 000 110		2 <b>31217</b> 1 <b>1111</b> 0 000 1 100	
Intensity Chart	Intensity Gra	Digital Wave	Mixed Signal
÷	Trate	<u>k</u>	
Compass Plot	Error Bar Plot	Feather Plot	XY Plot Matrix
		20	
Controls		3D Picture	3D Graph

XY Graph kiválasztása

Az XY Graph a Waveform Graph-hoz hasonló, de a vízszintes tengelyhez is hozzárendeljük egy tömb értékeit, így valós értékeket tudunk megjeleníteni mindkét tengelyen, azaz, ha a vízszintes tengely idő tengely, akkor az X tengely tömbjébe időpillanatokat tárolunk, ha mondjuk frekvencia, akkor a megfelelő frekvenciákat, stb.



11. ábra 3D-s grafikonok megjelenítése

A 3D-s grafikonok megjelenítésének, amint azt a 11. ábra mutatja, igen sok lehetőség van. A LabView-nak kiemelkedően jól alkalmazható ez a funkciója, nagyon egyszerűen és ehhez képest sok funkcióval, rugalmasan lehet 3D-s grafikonokat készíteni.

A LabView bevezető kurzusain a paletta harmadik sorából csak az első alkönyvtárat használjuk, ami a Ring & Enum cím alatt található. Ebben az alkönyvtárban a legördülő menük közül válogathatunk. Működésük hasonló, kis különbséggel. Javasoljuk kezdetekben az Enum típusú legördülő alkalmazását, mert a Block Diagramon ezt a legegyszerűbb kezelni.



12. ábra Legördülő menük

# 2.4. A Block Diagram függvényei

# 2.4.1. Érték típusok

LabView programozásban a változó típusok a C (Pascal) programnyelvekhez hasonlóan alkalmazhatóak, de itt a különböző típusok, különböző színekkel jelennek meg:

integer
double (real)
Boolean
string (char)
a Measurement

a Measurement and Automation – ban meghatározott virtuális csatorna.

Integer és double változók típusa megváltoztatható a *Representation* menüpontban (egér jobb gomb)

Nur	meric					
EDR	Visible Items	•				
	Find Indicator					
	Hide Indicator		I			
	Change to Control		I			
	Change to Constant		I			
	Description and Tip					
	Numeric Palette	►				
	Create	•	I			
	Data Operations	•	I			
	Advanced	×				
	View As Icon					
	Adapt To Source	,				
	Representation	×		W	ord	
	Properties		EXT	DBL 18	SGL E	FXP
			<b>164</b> 63 0	<b>I 32</b> 31 0 	I16	18 <sup>70</sup>
			<b>U64</b> 63 0	<b>U32</b> ≟	U16	U8
					CSG	

13. ábra Az érték típusának megváltoztatása

Az utolsó sorban található CXT, CDB és CSG típusok komplex érték típusok, mint azt az alábbiakban láthatjuk:

Numeric Numeric

## 2.4.2. Function Palette

A programban alkalmazható függvényeket a függvény palettáról (Function Palette) hívhatjuk be.



14. ábra A teljes Function Palette

A LabView bevezető kurzusaiban szinte kizárólag a "Programming" könyvtár függvényeit alkalmazzuk, ezért a továbbiakban ezt mutatjuk be.



15. ábra A Programming könyvtár eszköztára

#### A Programming könyvtár felső sora:



tömbkezelő műveletek

Cluster kezelő műveletek

ciklusok while, for case, sequence egyenletek

16. ábra A Function Palette felső sorából választható függvények

A függvény paletta második sorában Numeric könyvtár az alapvető matematikai műveleteket tartalmazza (17. ábra), a Boolean könyvtár a Boole algebra alapműveleteit (18. ábra), a Sting könyvtár pedig a karakter és szöveg kezelő műveletekt (19. ábra) :

Numeric						B
ŶQ	Search	n 🔌 Customiz	e*			
🕨	>	Þ	×	÷	÷R 10	) <u>132</u> ) DBL
Add	1	Subtract	Multiply	Divide	Quotient &	Conversion
+	>	∢	$\Sigma$	Þ		
Increm	ent	Decrement	Add Array El	Multiply Arra	Compound	Data Manipu
	>	₽	₽	₽	×2	8+iY
Absolute	Val	Round To N	Round Towa	Round Towa	Scale By Pow	Complex
	>	×2		<b>*</b>		mx+b ∼+^√
Square	Root	Square	Negate	Reciprocal	Sign	Scaling
123	3	⊕ Enum	Ring	r 🗗	EXPR	FXP
Numeric	: Co	Enum Const	Ring Constant	Random Nu	Expression N	Fixed-Point
123	]	+00	-00	E		π ► €
DBL Nur	neri	+Inf	-Inf	Machine Eps		Math Consta

17. ábra A Numeric könyvtárban választható függvények



18. ábra A Boolean könyvtárban választható függvények



19. ábra A String könyvtárban választható függvények

A függvény paletta harmadik sorában a Comparison az összehasonlítások függvényeit tartalmazza (20. ábra), a Timing könyvtár a különböző időzítések, dátum és idő kezelések függvényeit (21. ábra), és a Dialog & User Interface könyvtárban a felhasználóval történő speciális kommunikáció függvényei, üzenet kiíratás, felhasználói művelet végzésére felszólítás, várakozás, stb. (22. ábra).

Comparison					8
🕜 🔍 Searc	h <u> S</u> Customiz	e▼			
	≽	$\geqslant$	$\geq$	$\geqslant$	<b>&gt;</b>
Equal?	Not Equal?	Greater?	Less?	Greater Or E	Less Or Equal?
Þ	<b>‡0</b>	>0	<0	30	<b>\$0</b>
Equal To 0?	Not Equal To	Greater Than	Less Than 0?	Greater Or E	Less Or Equa
	27 w 27 w	<mark>&amp;</mark> ?	<b>&gt;&gt;</b>	2	
Select	Max & Min	In Range and	Not A Numb	Empty Array?	Empty String
92	•			.?>	
Decimal Digit?	Hex Digit?	Octal Digit?	Printable?	White Space?	Lexical Class
_ <mark>_</mark>					
Comparison					Fixed-Point

20. ábra A Comparison könyvtárban választható függvények



21. ábra A Timing könyvtárban választható függvények



22. ábra A Dialog & User Interface könyvtárban választható függvények

A függvény paletta negyedik sorából a fájl kezelő függvényeket alkalmazzuk majd a bevezető kurzuson, ami a 23. ábrán látható, valamint a Waveform könyvtár hullámforma kezelő függvényeit (24. ábra).



23. ábra A File I/O könyvtárban választható függvények

Waveform					8
🕆 🔍 Search 🔍 Cu	ustomize▼				
~~ <u>t0</u> <u>dt</u>	<mark>™ .</mark> ≯	<b>∼</b> .	<u>~                                    </u>	<mark>∿ տ</mark> Կ ∾	
Get Wfm Co Build W	/avef Set Attribute	Get Attribute	Analog to Di	Digital to An	
	<mark>m ∧m</mark> ©©		<mark>∼ m</mark> ↓	<mark>∼ π</mark> ≧	
Idx Wfm Array Copy W	Vfm dt Align Times	Get Wfm Su	Get Final Time	Wfm Duration	
	<u>m</u> ∰ ↓[t]				
Scale Delta t Get XY	Value Get Time Arr				
			Analog Wfm	Digital Wfm	Wfm File I/O

24. ábra A Waveform könyvtárban választható függvények

# 3. Jelgenerálás, megjelenítés, jelfeldolgozás alapfunkciói

### FELADAT

Készítsen egy olyan tömböt, amelynek az elemeit egy START gomb megnyomásakor feltölt a program 1 periódusnyi szinuszosan változó értékekkel. Legyen a tömb elemeinek a száma 1000. A szinusz jel amplitúdója legyen 5,0. Rajzolja ki a kapott szinusz hullámot egy Waveform Graph típusú grafikonra. A feladat megoldásához szükségünk lesz egy olyan függvényre, amely szinusz értéket számol. Ezt a Mathematics könyvtár, Elementary & Special Functions > Trigonometric Functions alkönyvtárában találjuk.



### 25. ábra Függvény egy radiánban megadott szög szinuszának kiszámolásához





26. ábra Egy periódus szinusz hullám generálása (LV5\_1.vi)

A fenti programban a grafikon vízszintes tengelyén a pontok száma jelenik meg, azaz az időtengely szimulálása nem történik meg.

Kérdés: Milyen adatra van szükségünk az időtengely szimulálására? Természetesen arra, hogy két pont között mennyi az eltelt idő, azaz a vízszintes tengely idő felbontására. Hogyan tudjuk ezt egyszerűen létrehozni?

Használjuk az X (idő) tengely létrehozásához a "Build Waveform" függvényt a Waveform könyvtárból.

A Build Waveform függvény bemenetén három paraméter állítható be, amelyekből a hullámalakot a függvény felépíti. Ezek az Y (függőleges) tengely értékeinek tömbje, a dt a két pont közötti idő és a t0, ami az első pont időpillanata. A t0 értéke jelen feladatnál nem szükséges, csupán a delta t értékét adjuk meg egy vezérlővel (80. ábra, LV5\_2.vi).



27. ábra Build Waveform alkalmazása



28. ábra Jelalak szimulációja

Nézzük meg, hogyan tudunk a fentiektől egyszerűbben kezelni (generálni, kiértékelni) jelalakokat LabView környezetben.

Mentsük el a fenti feladatban elkészített programot, és kezdjünk egy új programot, készítsük el a while ciklust a leállító gombbal.

A jelgenerálás függvényeit a Signal Processing könyvtár Waveform Generation alkönyvtárban találjuk. A második sorban a négy alapjel, rendre a szinusz, négyszög, háromszög és fűrész hullámalak generálására alkalmas függvényeket találjuk.

-🖾 Waveform	Generation			
		f(t)		
Basic FuncGen	Tones & Noise	Formula Wfm		
Sine Wfm	Square Wfm	Triangle Wfm	Sawtooth Wf	
Σ	Σ <b>e</b>	Σ		
Basic Multito	Multi with A	Multitone Gen		
1 				
Uniform Wfm	Gaussian Wfm	PRN Wfm	Inv f Wfm	
	poisson			
Gamma Wfm	Poisson Wfm	<b>Binomial Wfm</b>	Bernoulli Wfm	MLS Wfm
<b>E</b>	K			
Simulate Sig	Sim Arb Sig			

29. ábra Jelgenerálás könyvtára

Próbáljuk ki a szinusz jel generálását!

Legyen az amplitúdó, frekvencia és fázis a Front Panelről állítható. FIGYELEM! a "Reset signal" bemenetre egy TRUE konstans értéket kell kapcsolni ahhoz, hogy a fázis állítható legyen!



30. ábra Szinuszjel generálása

Az így generált szinuszjel vízszintes tengelye nem szimulálja megfelelően a valós időtengelyt. Ennek az oka, hogy a kirajzolt hullámalak pontjainak számát és a két pont között eltelt időt nem ismerjük, illetve nem tudjuk változtatni. (Alapértelmezésben a jelgenerátor 1000 pontból és 1 ms mintavételezési idővel (dt) generálja a jeleket.) Ha szeretnénk életszerű szimulációt végezni, akkor ezt a két értéket (mintavételezési idő, illetve ennek a reciproka a mintavételi frekvencia és a mintaszám) megadása szükséges. Hozzunk létre egy vezérlőt ehhez a bemenethez is.



31. ábra Szinuszjel generálás a mintavételezési adatok szimulálásával

Egészítsük ki a programunkat egy második grafikonnal, amire egy négyszögjelet rajzolunk ki ugyanazon vezérlők segítségével, mint amiket a szinuszjel generálásnál használunk.



32. ábra Szinusz és négyszögjel generálása ugyanazon vezérlőkkel Tegyünk fel egy vezérlőt a kitöltési tényező változtatásához! Tegyünk fel a kijelzőre egy harmadik grafikont!

Most rajzoljuk ki a generált szinusz és négyszögjelet együtt a harmadik grafikonra! Használjuk a Build Array függvényt a jelek egybekapcsolására.



33. ábra Build Array alkalmazása több függvény egy grfikonra történő kirajzolásához

Húzzuk szét a grafikon jobb felső részén látható, a rajzolás beállításához használható un. Plot Legend mezőt úgy, hogy mindkét kirajzolás jellemzőit állítani tudjuk. Próbáljuk ki a lehetőségeket!



34. ábra A Plot Legend kipróbálása

Tegyünk fel a Front Panelre egy XY grafikont!

A kitöltési tényező vezérlőt bekötő vezetéket töröljük ki, és a négyszögjel generátort cseréljük ki szinuszjel generátorra. A kitöltési vezérlőből csináljunk fázis2 vezérlőt, és kössük be a 2. szinuszjel generátorhoz azért, hogy a két szinuszjelnek különböző fázisértéket tudjunk adni. Kössük be a két szinuszjelet az XY grafikon X és Y tengelyére.

FELADAT:

Oldja meg, hogy a két szinuszjel-generátorra kapcsolt frekvenciák külön állíthatóak legyenek, de mindig tetszőleges egész számú többszörösei lehessenek egymásnak!

# 4. Jelfeldolgozás idő és frekvencia tartományban

### FELADAT

Készítsen LabView programot, amely képes két hullámalakot generálni és azokat összeadni. Legyen külön-külön változtatható a hullámalakok típusa, amplitúdója, offszetje és frekvenciája.

Legyenek közösek a mintavételezési adatok!

Rajzolja ki a generált jeleket és az összegzett jelet egy közös grafikonra!

## 4.1. Lineáris és négyzetes középérték meghatározása

Írassuk ki egy digitális kijelzőre a generált jelek összegének lineáris és négyzetes középértékét.

Használjuk ehhez a Signal Processing menüt.



35. ábra LV\_6\_2.vi

## 4.2. Frekvencia analízis

Készítsük el az összegzett jel frekvencia spektrumát!

Frekvencia spektrumot (FFT) legegyszerűbben az Express menüben lévő SignalAnalysis könyvtárban található Spectral Measurements függvénnyel készíthetünk. A függvény inicializálásakor az alábbi beállításokra figyeljünk:

Selected Measurement:	Magnitude (Peak)
Result:	Linear
Window:	None

Figyeljünk arra, hogy a spektrum megjelenítésekor a spektum vonalak (felharmonikus amplitúdók) függőleges vonalakkal jelenjenek meg és ne összekötött vonallal,hiszen a frekvencia összetevők diszkrét értékek!



36. ábra FFT analízis legegyszerűbb módszere (LV6\_3.vi)

# 4.3. Alul- és felüláteresztő szűrők alkalmazása

Gyakori feladat a jelfeldolgozáskor,hogy a jelbőlbizonyos frekvenciájú összetevőket ki kell szűrni. Erre különböző digitális szűrőket alkalmazhatunk.

Készítsünk aluláteresztő szűrőt az összegzett jelhez! Használjuk ehhez az Express menü Signal Analysis könyvtárának Filter függvényét!

A jel szűrése után ismét analizáljuk a már szűrt jelet FFT-vel, ennek eredményét rajzoltassuk ki egy 4. grafikonra!

A szűrő határfrekvenciájának változtatásához tegyünk fel a front Panelre egy számvezérlőt!



37. ábra Jel szűrése és FFT analizálása (LV6\_4.vi)

## 4.4. Jelek fájlba mentése és visszatöltése

Egészítsük ki a programunkat fájlba mentés és fájlból történő beolvasás lehetőségével! Ehhez tegyünk fel a Front panelre egy MENTÉS és egy BETÖLTÉS FÁJLBÓL feliratú nyomógombot. Készítsünk hozzájuk egy-egy case struktúrát. Figyeljünk a gombok működési funkciójára (Mechanical action), mert ha ezt nem jól állítjuk be, akkor előfordulhat,hogy a while ciklus minden egyes futási ciklusában menteni vagy visszatölteni akar majd a program. A gombok "Latch" típusú funkcióra legyenek állítva!

A fájlbóltörténő betöltés eredményét rajzoljuk ki egy 5. grafikonra!

Az adat fájlok kezelése a hullámalakok esetén a legegyszerűbb. A fájlba írást (Write waveforms to file) és onnan olvasást (Read waveform from file) az alábbi ábrán látható könyvtárban találjuk:



38. ábra Hullámalak mentésére és visszatöltésére szolgáló függvények helye a könyvtárrendszerben

A feladat megoldását az LV6\_5.vi fájlban találjuk.



39. ábra Jelalakfájlba mentése és visszatöltése (LV6\_5.vi)

### FELADAT

Oldja meg, hogy a visszatöltött jel és a generált jel ugyanazon a grafikonon jelenhessen meg, de ne egyszerre, hanem egy kapcsoló állásától függően vagy a generált vagy a legutoljára visszatöltött jelet lássuk a grafikonon! Megoldás: LV6\_6.vi

# 5. Az gyakorlati óra minta anyaga

#### FELADAT

pl: Készítsen LabView programot, amely segítségével a felhasználó 3 különböző jelet tud generálni, amelyeknek bemenő adatait egyedileg lehet változtatni. Legyen mód a jelgenerátorok be vagy ki kapcsolására. A generált jelek egy közös kijelzőn jelenjenek meg, a felhasználó döntése nyomán. Legyen a jel amplitúdója, frekvenciája, fázistolása, tetszőlegesen változtatható a felhasználó által Az egyes alap jeleket legyen lehetőség a kijelzőről ki- és bekapcsolni. Ekkor az eredmény jel ne változzon, csak akkor, ha a felhasználó a jelgenerálásokat kapcsolja ki. Ha a felhasználó beállította, hogy milyen jelekkel szeretne dolgozni, legyen lehetősége a bekapcsolt jeleket összeadni, kivonni, elosztani és szorozni. Az eredmény jelet jelenítse vastagon kiemelve a generált alapjelekkel közös grafikonban, amelyen legyen kurzor is, aminek az adatai folyamatosan legyenek leolvashatóak egy kijelzőn a grafikon mellett. A program számítsa ki a kiválasztott alapjel vagy az eredmény jel frekvencia spektrumát a felhasználó döntése alapján. Az jelet lehessen lementeni egy adat file-ba, és onnan visszatölteni a jelek grafikonjára.





# 1. Empírikus sűrűség függvény meghatározása

#### FELADAT

Készítsen LabView programot, amellyel a felhasználó egy maximum 100 elemű mérési adat tömb empirikus sűrűségfüggvényét tudja meghatározni. (Az elemek száma 2-100 között legyen megadható). A megadott elemeket rendezze sorba és válassza ki a minimális elemet. Határozza meg az átlag, szórás, terjedelem, átl. absz. eltérés és a valószínű hiba értékét. Vizsgálja meg, hogy a sorozat Gauss eloszlású-e! Készítse el a csoportosított elemek tömbjét a megadott mérési adatok alapján. Készítse el a gyakoriság, rel. gyakoriság és empirikus sűrűség függvény diagramjait. Határozza meg a csoportokba rendezett tömb elemek alapján az átlag, szórás, terjedelem, átl. absz. eltérés és a valószínű hiba értékét. Adja meg a relatív hiba értékeket mind az öt előzőleg számított értékre. Ha a felhasználó nem megfelelő xr1 vagy delta x értéket ad meg a csoportokba rendezés esetén, figyelmeztessen a program, villogó felirat és egy villogó LED segítségével.

N#412 412 111				-		21.25		
Wert enteker		Álag	Sorba rend	dezve	Csoport to	omb	Álag 2	
100,4	Lasil@sr	104,88	100,4		100		104,80	
101,9	Ledinids	Szórás (s)	101,9		102		Szórás (s) 2	
102,2		2,15592	102,2		102		2,21478	
102,8	Elemek szama	Atl. absz. elteres (E)	102,8		103		Atl. absz. elt	érés (E) 2
103	14	Valószínű biba (P)	103	I E	103		1,92	
103,3	Figyelembe vett	3.6	103,3	1 8	103		Valoszínű hi	ba (P) 2
103.3	elemek szama	Terjedelem (R)	103,3	1 -	103		Teriedelem (	(R) 2
103.9	T/20	7,2	103,9	1 -	104	1	8	
103,5	2-100 között adható meg		104,4	1 🗄	104	1 1		
104,4	+15% -15	i%	104,9	1 🗄	105	1 1		
104,9	1,80642 1,3	33518	105,7	1	106			
105,7	s^2/E^2 ara	ny	106,2	1 🗄	106	1 1	Hiba (Atlag)	9/
106,2	1,32634		106,4	1 🗄	106		-0,0762777	/0
106,4	Gauss eloszlású-	e? Nem	106.4		106	1	Hiba (s)	9/
106,4			106.7	1 -	107		2,73019	10
106,7	xr1	delta x	106.8		107	- 1	Hiba (E)	
(106,8	( 100 ( )	1	107		107		2,5641	70
107			107.3	1 8	107		Hiba (P)	
107,3	Csoportok száma	Minimum	107,5		107		11,1111	%
107,4	1 <sup>9</sup>	100,4	107,4		107		Hiba (R)	
107.6	99,9 és 100,9	közé kell esnie	107,6		108		11,1111	%
101.00			0		0			
0	Az xr1 megadása Hely	yes 🔵	10	-	0	*		
1510							a ao ao 1 amin'	_
Gyakoriság	Re	l. gyakoriság (nr/n)		E	mpirikus sū	rűség f	gv (nr/n x dx)	
5-		0,25-			0,25-			
4-	(e	. 0,2-			₩ 0,2-			
sag	a (pr	0,15-			2 0,15-			
akori	orisă,	01-			Des 01-			
\$ <sup>4</sup>	Svak				akor			
1		0,05-	חור		ය <sup>0,05-</sup>			
0-1	2 3 4 5 6 7 8			1	0-,	2	3 4 5 6	8
	Csoportok	Csoport	ok			c	soportok	100
		الكافا كالبلا بكافا كافراك						1 1 1 1 1

