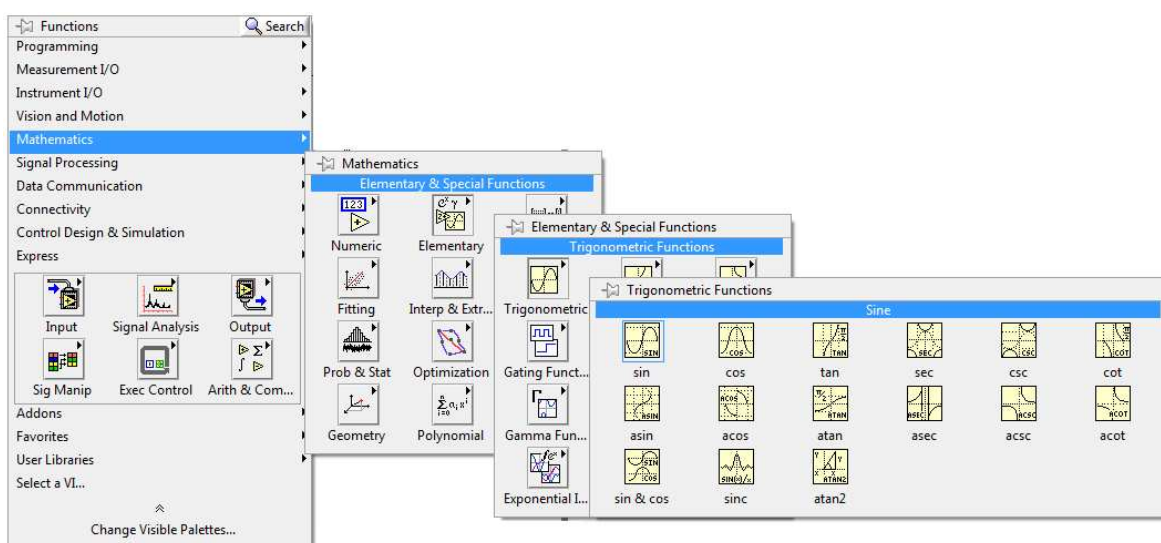


1. Jelgenerálás, megjelenítés, jelfeldolgozás alapfunkciói

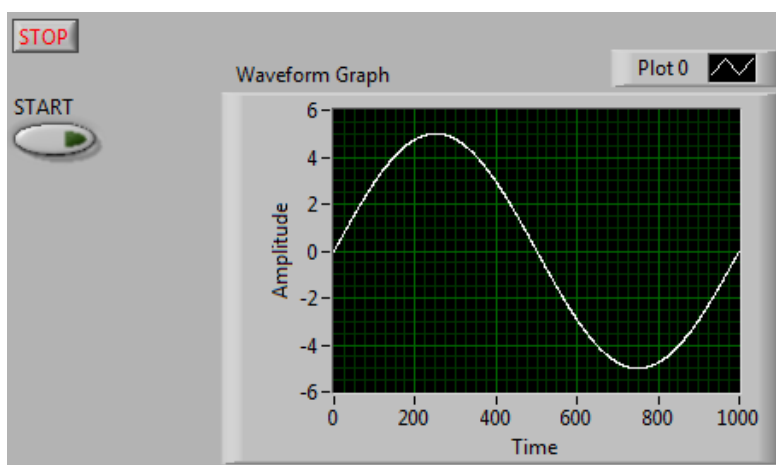
FELADAT

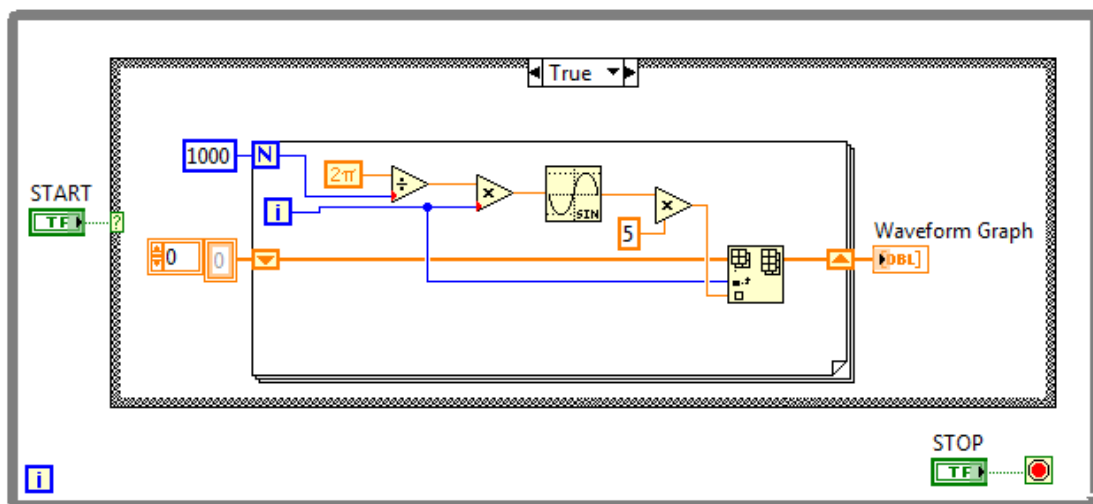
Készítsen egy olyan tömböt, amelynek az elemeit egy START gomb megnyomásakor feltölt a program 1 periódusnyi szinuszosan változó értékekkel. Legyen a tömb elemeinek a száma 1000. A szinusz jel amplitúdója legyen 5,0. Rajzolja ki a kapott szinusz hullámot egy Waveform Graph típusú grafikonra.

A feladat megoldásához szükségünk lesz egy olyan függvényre, amely szinusz értéket számol. Ezt a Mathematics könyvtár, Elementary & Special Functions > Trigonometric Functions alkönyvtárában találjuk.



1. ábra Függvény egy radiánban megadott szög szinuszának kiszámolásához





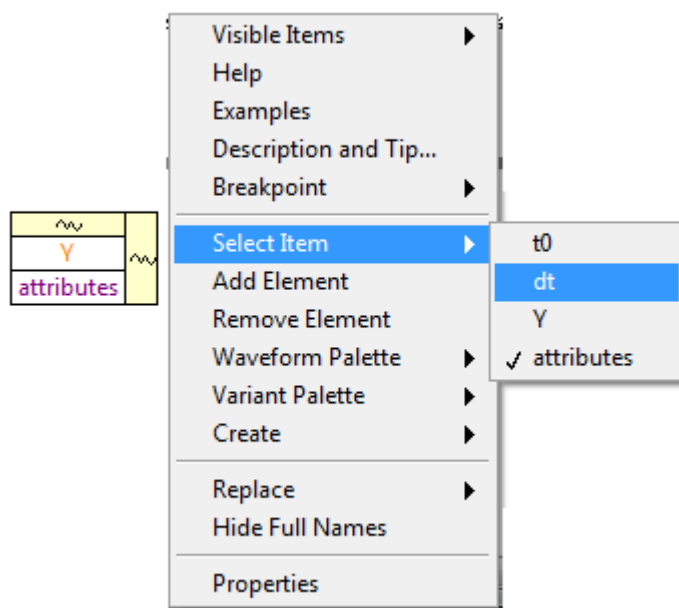
2. ábra Egy periódus szinuszos hullám generálása (LV5_1.vi)

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

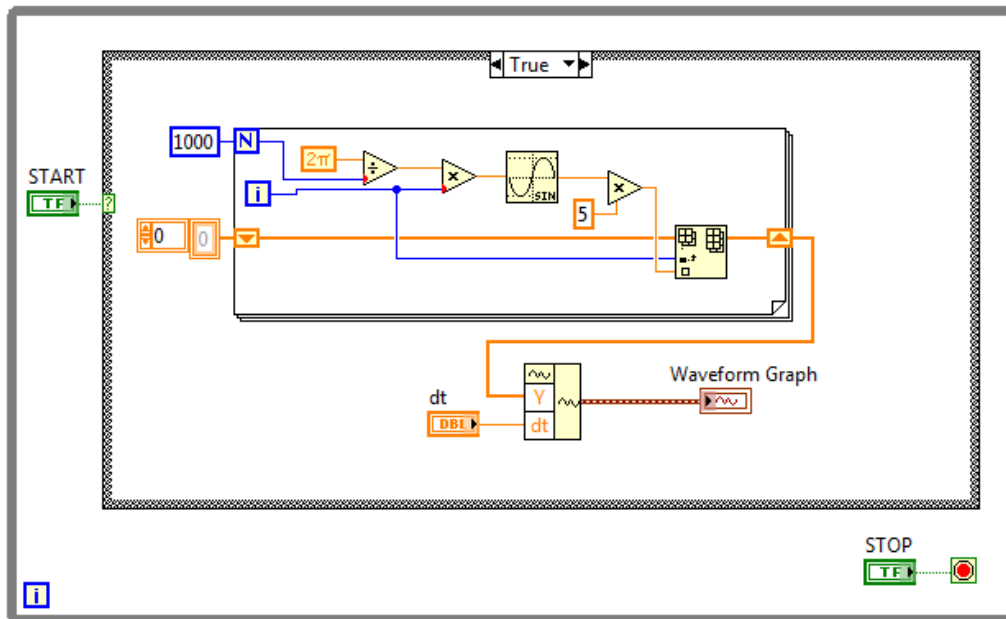
Kérdés: Milyen adatra van szükségünk az időtengely szimulálására? Természetesen arra, hogy két pont között mennyi az eltelt idő, azaz a vízszintes tengely idő felbontására. Hogyan tudjuk ezt egyszerűen létrehozni?

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).



3. ábra Build Waveform alkalmazása

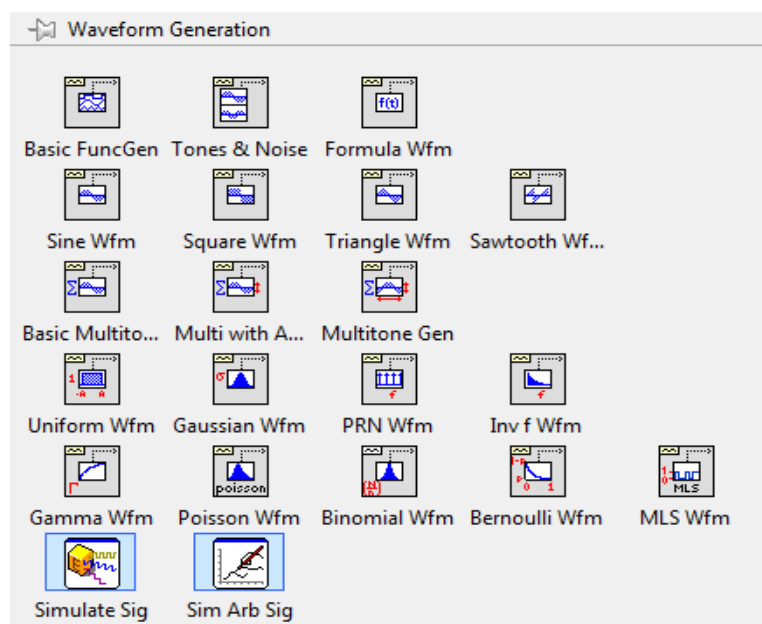


4. ábra Jelalak szimulációja

Nézzük meg, hogyan tudunk a fentiekől 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.

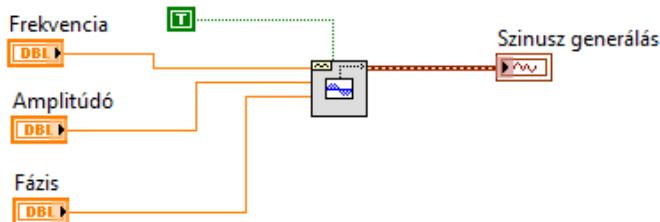
A jelgenerálás függvényeit a Signal Processing könyvtár Waveform Generation alkönyvtárban találjuk. A második sorban a négy alapjel, rendre a szinusz, négyszög, háromszög és fűrész hullámalak generálására alkalmas függvényeket találjuk.



5. ábra Jelgenerálás könyvtára

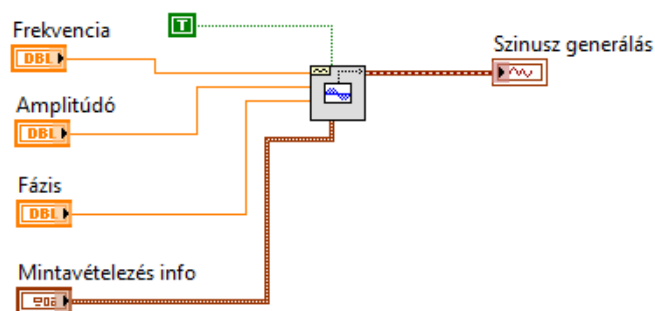
Próbáljuk ki a szinusz jel generálását!

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!



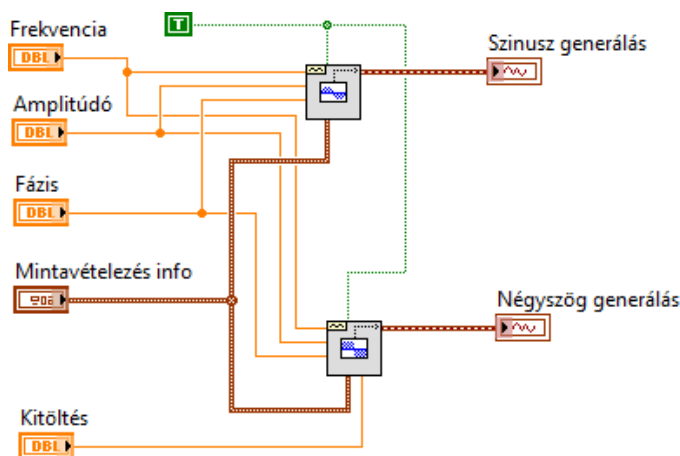
6. ábra Szinuszjel generálása

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 (Δt) 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.



7. 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.

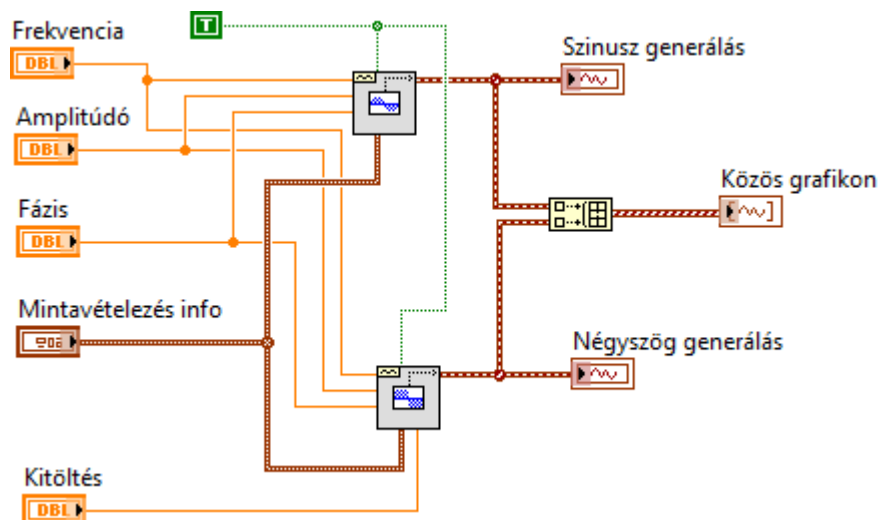


8. ábra Szinusz és négyszögjel generálása ugyanazon vezérlőkkel

Tegyük fel egy vezérlőt a kitöltési tényező változtatásához!

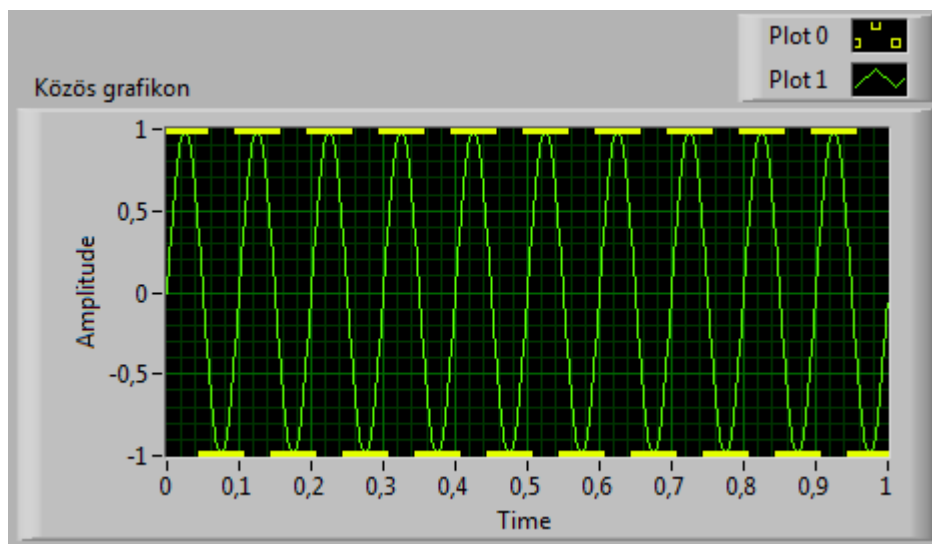
Tegyük fel a kijelzőre egy harmadik grafikont!

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



9. ábra Build Array alkalmazása több függvény egy grafikonra történő kirajolásához

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



10. ábra A Plot Legend kipróbálása

Tegyük 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!

2. Jelfeldolgozás idő és frekvencia tartományban (6. óra)

FELADAT

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

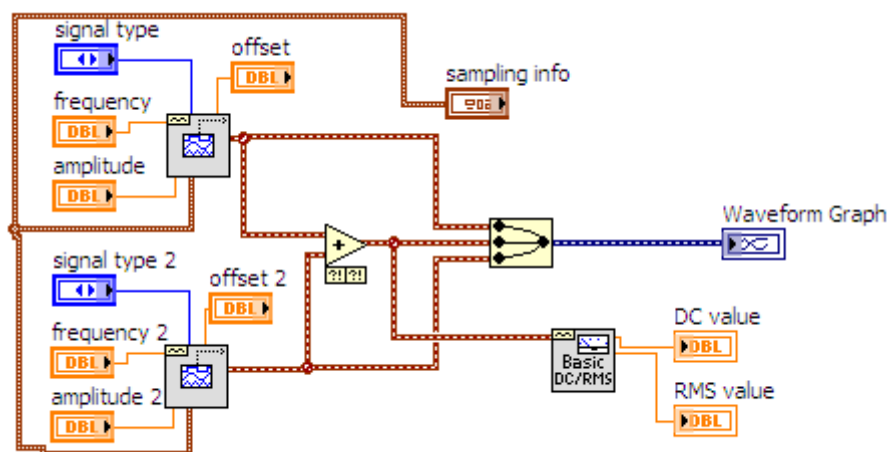
Legyenek közősek a mintavételezési adatok!

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

2.1. Lineáris és négyzetes középérték meghatározása

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

Használjuk ehhez a Signal Processing menüt.



11. ábra LV_6_2.vi

2.2. Frekvencia analízis

Készítsük el az összegzett jel frekvencia spektrumát!

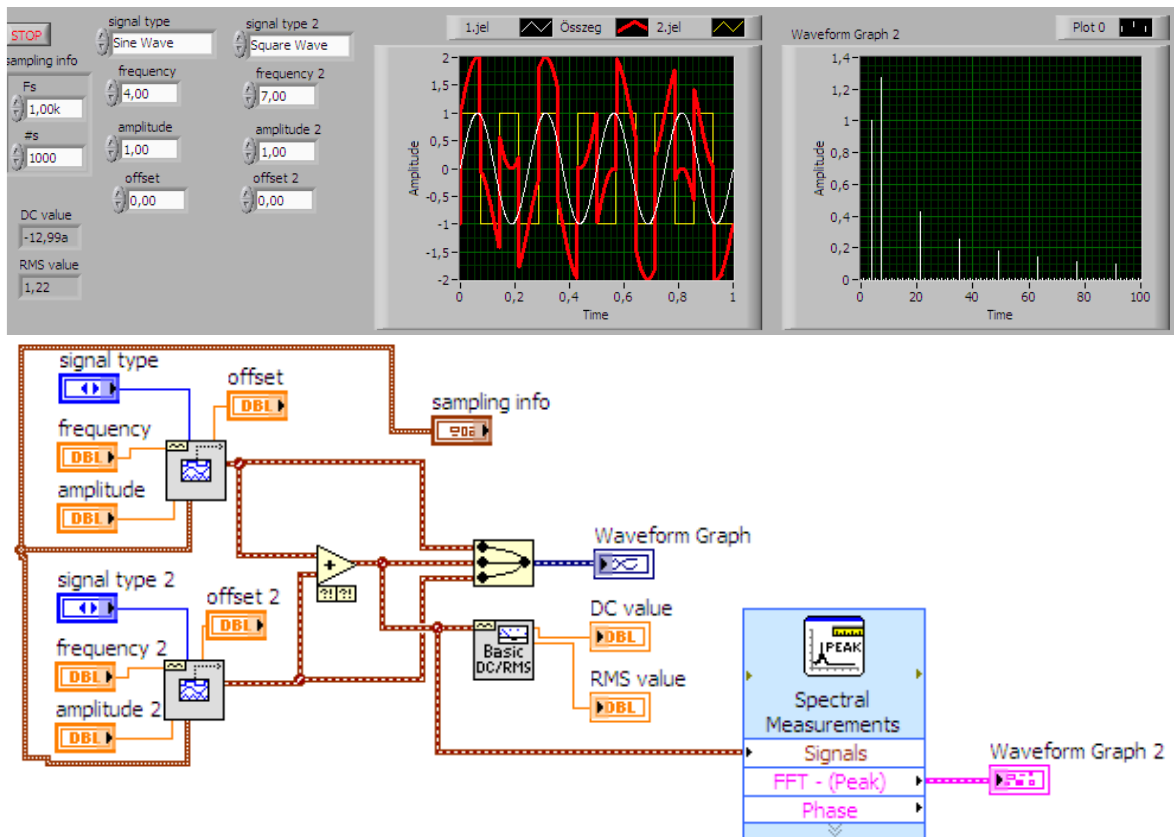
Frekvencia spektrumot (FFT) legegyszerűbben az Express menüben lévő SignalAnalysis könyvtárban található Spectral Measurements függvénnyel készíthetünk. A függvény inicializálásakor az alábbi beállításokra figyeljünk:

Selected Measurement: Magnitude (Peak)

Result: Linear

Window: None

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



12. ábra FFT analízis legegyszerűbb módszere (LV6_3.vi)

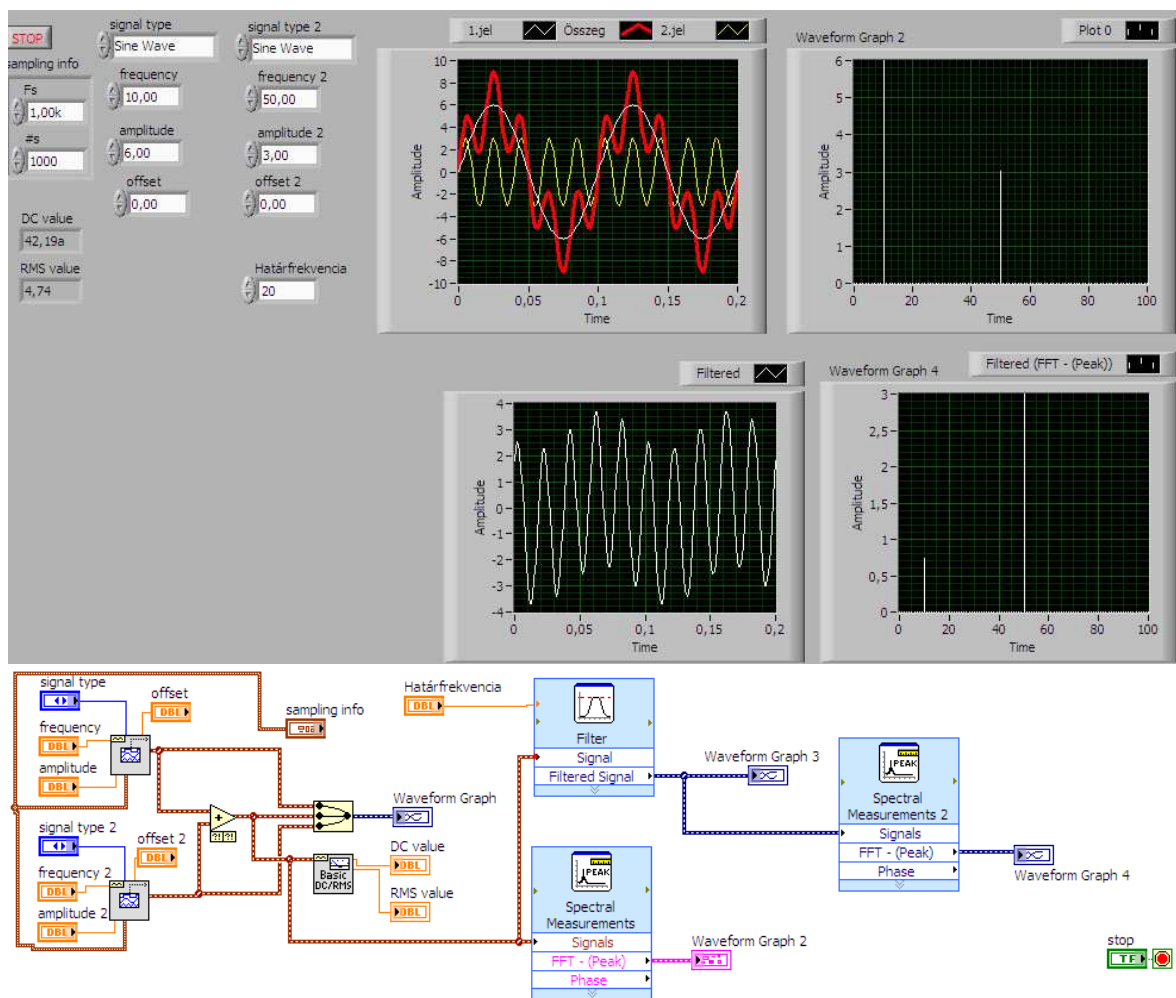
2.3. Alul- és felüláteresztő szűrők alkalmazása

Gyakori feladat a jelfeldolgozáskor, hogy a jeltől bizonyos 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!



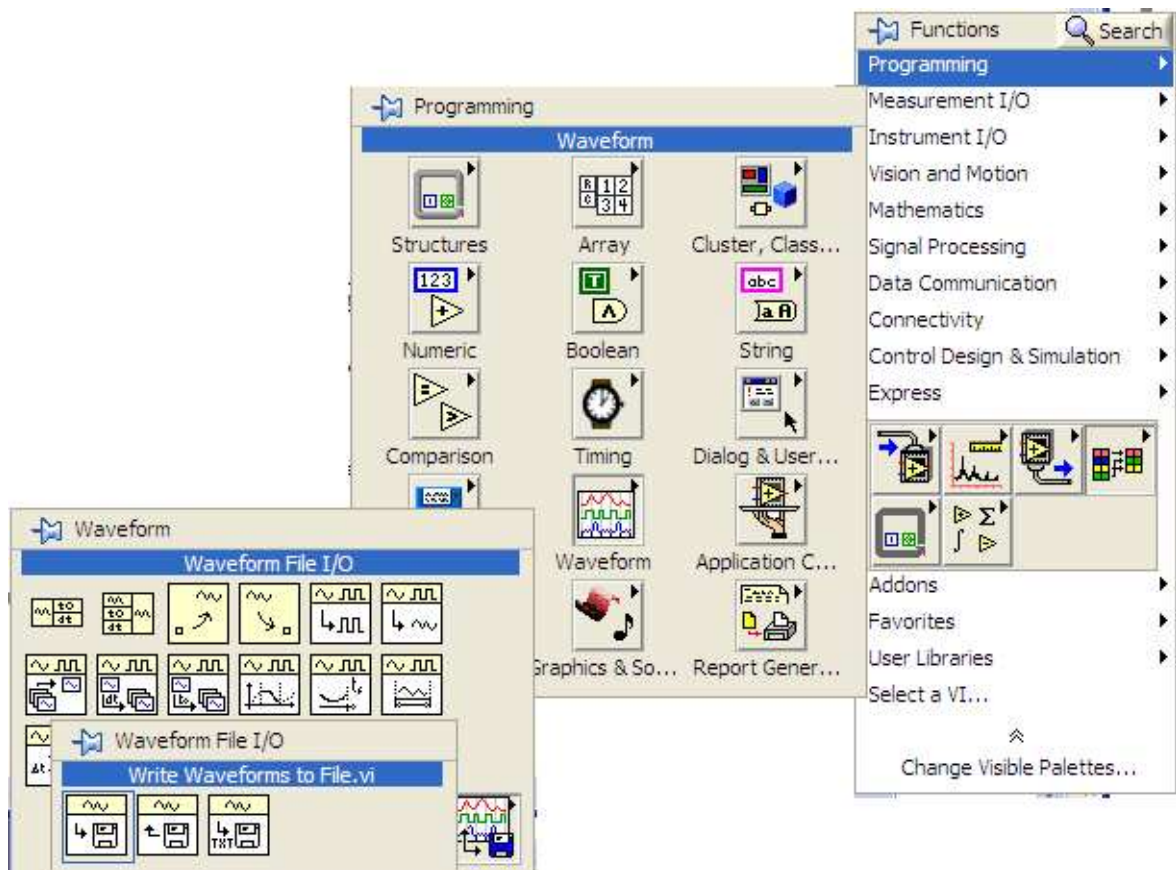
13. ábra Jel szűrése és FFT analízálása (LV6_4.vi)

2.4. 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!

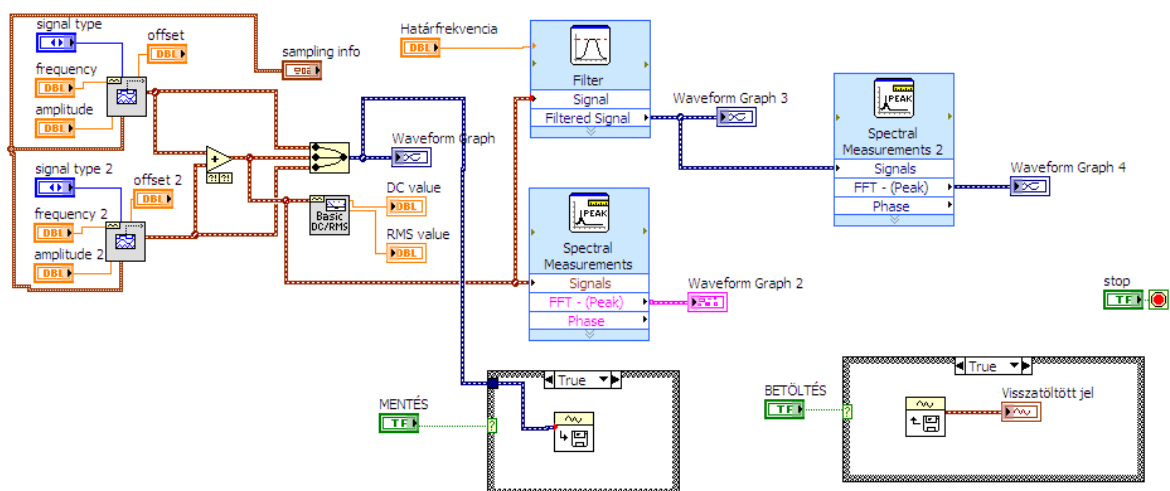
A fájlból történt betöltés eredményét rajzoljuk ki egy 5. grafikonra!

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:



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

A feladat megoldását az LV6_5.vi fájlban találjuk.



15. 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