NI PCI-6251 multifunkciós mérésadatgyűjtő kártya adatai

ANALÓG BEMENET

Csatornaszám	16 SE (8 DIFF)
AD felbontása	16 bit
Max konverziós frekvencia	1,25 MHz (egycsatornás) 1 MHz (többcsatornás)

Bemeneti feszültség tartománvok	Felbontás (5%-os over range)
-10 V 10 V	305 μV (320 μV)
-5 V 5 V	153 μV (160 μV)
-2 V 2 V	61 µV (64 µV)
-1 V 1 V	30,52 µV (32 µV)
-500 mV 500V	15,3 μV (16 μV)
-200 mV 200 mV	6,1 μς (6.4 μV)
-100 mV 100 mV	3,05 μV (3.2 μV)
Bementi impedancia	$10 \ \mathrm{G}\Omega$
Bemeneti áram	100 pA
Bementi FIFO mérete	4095 minta
Adat transzfer mód	DMA
	Megszakítás
	Programozott
<u>Analóg trigger</u>	
Trigger jelek szómo	1
Trigger contornák	$1 = 1 = 0 = 15 \times AIDE0$
ingger csatornak	AI<0132, All 10
<u>Digitális trigger</u>	
Trigger csatornák	PFI<015>
ANALÓG KIMENET	
Csatornaszám	2
DA felbontása	16 bit
Maximális frissítési frekvencia	2 MHz
Jelgenerálás esetén a beállási i	dő 2 µsec/LSB
Feszültség tartomány	$\pm 10 \text{ V}; \pm 5 \text{ V}$
Áramterhelhetőség	±5 mA
Adat transzfer mód	DMA
	Megszakítás
	Programozott
DIGITÁLIS I/O	

Csatornaszám 24 össesen: 8 P0.<0...7> 16 PFI<0...15>

SORKAPOCS

	_	\sim	
(
AI 0	68	34	AI 8
AI GND	67	33	AL1
AI 9	66	32	ALGND
AI 2	65	31	AI 10
ALGND	64	30	AI 3
AI 11	63	29	ALGND
AI SENSE	62	28	AL4
AI 12	61	27	AI GND
AI 5	60	26	AI 13
ALGND	59	25	AI 6
AI 14	58	24	ALGND
AL7	57	23	AI 15
ALGND	56	22	AO 0
AO GND	55	21	AO 1
AO GND	54	20	APFI 0
DGND	53	19	P0.4
P0.0	52	18	D GND
P0.5	51	17	P0.1
DGND	50	16	P0.6
P0.2	49	15	D GND
P0.7	48	14	+5 V
P0.3	47	13	D GND
PFI 11/P2.3	46	12	D GND
PFI 10/P2.2	45	11	PFI 0/P1.0
DGND	44	10	PEL 1/P1.1
PFI 2/P1.2	43	9	D GND
PFI 3/P1.3	42	8	+5 V
PFI 4/P1.4	41	7	D GND
PFI 13/P2.5	40	6	PFI 5/P1.5
PFI 15/P2.7	39	5	PFI 6/P1.6
PEL7/P1.7	38	4	D GND
PFI 8/P2.0	37	3	PFI 9/P2.1
DGND	36	2	PFI 12/P2.4
DGND	35	1	PFI 14/P2.6
(J

Szimulált hardver alkalmazása

DAQmx rendszerben használhatunk szimulált mérésadatgyűjtő kártyát, ami nagyon hasznos lehet a mérésekre történő otthoni felkészülésnél.

A szimulált hardver alkalmazásához a LabWindows CVI –hoz fel kell installálni a DAQmx device driver szoftvert és a Measurement and Automation Explorer szoftvert. Ez utóbbi a DAQmx installálásakor általában automatikusan telepítésre kerül.

Ha az mx driver –t installáltuk, akkor a MAX (Measurement and Automation Explorer) programban a "Device and Interfaces >> NI-DAQmx Devices" menüpontra az egér jobb billentyűjével kattintva megjelenik a "Create New NI-DAQmx Device", amivel a NI-DAQmx Simulated Device-t választhatjuk ki.



A megjelenő listából válasszuk ki azt az adatgyűjtőt, amit a lborban is használni fogunk, azaz az M sorozatú kártyák közül a 6251 típust.



A sikeresen létrehozott szimulált kártya sárga ikon jelzéssel jelenik meg az eszközlistában. A fenti jobb oldali monitorképen látható listában zöld jelzéssel látjuk a gépben található valós 6024E típusú adatgyűjtőt, és sárgával a szimulált 6251 típusú kártyát.

A szimulált kártyához létrehozhatunk virtuális feladatokat (task) és/vagy csatornákat is. Megjegyzés: Nem föltétlenül kell taskokat vagy virtuális csatornákat a MAX-ban létrehozni, ezek a LabView programban is létrehozhatók, de olyan esetekben, ha program futás közben nem akarunk a mérés paraméterein változtatni, talán így egyszerűbb az alkalmazás.

A virtuális csatorna létrehozásához kattintsunk jobb egér gombbal a "Data Neighborhood" menüpontra, s ott a "Create New"-ra. A megjelenő listából válasszuk a "NI-DAQmx Global Virtual Channel" menüpontot, majd attól függően, hogy bemeneti vagy kimeneti csatornát akarunk szimulálni, válasszuk az "Acquire Signals" vagy a "Generate Signals" lehetőségek egyikét. A további választásokat értelemszerűen a szimulált csatorna igénynek megfelelően kell beállítani. A létrehozandó csatornához hozzá kell rendelni a szimulált kártya egy csatornáját, ugyanúgy, mintha valóságos kártya lenne. A megfelelően létrehozott virtuális csatorna az alábbiak szerint jelenik meg:

🤏 MyVoltageChannel - Measurement & Autom	ation Explorer	
Elle Edit View Iools Help		
Configuration	Save 🚱 Run 🔻	Show Help
Cues Neighborhod Cues Neighborhod Cues Neighborhod Cuerk Cuerk Cuerk Cuerk Cuerk Cuesk Cue	My/oltopoChannel 0 ●	
	Configuration Channel Statings //dd: Petalof C Order Physical Device Channel Type MyVoltageClannel 0 Dev/2/s0 PCI-6051	
	htt NI-DADmx Global Channel 📋 Description 🦧 Connection Diagram	

Az így létrehozott csatornákat használhatjuk a LabView programunkban.

A virtuális task létrehozása a fentiekhez teljesen hasonlóan történik, azzal a különbséggel, hogy itt be kel állítani a mintavételezés paramétereit is (N mintaszám vagy folyamatos; időzítések, triggerelések, stb.)

DAQmx kártya programozása LabView-ban

LabView-ban a DAQmx kártyákat legegyszerűbb módon a DAQ_Assistant függvénnyel programozhatjuk fel. Ezzel a segédeszközzel az egyszerűbb mérések felprogramozása jelentősen egyszerűsödik, sajnos a programozhatóság rugalmassága ezzel egy időben azonban csökken.

Példa:

Ha a program futása közben változtatni kívánjuk a mérendő csatornákat is, akkor az Assistant-ban elkészített mérés funkció kezelése bonyolultabbá válik.

Ezért gyakran nem elég a DAQ Assistant alkalmazása, hanem az alkalmazást az mx függvényekből kell összeállítani. Ezek ismertetését foglalja össze ez az oktatási segédlet.

Azt tudni kell, hogy mx rendszerben mindig két megoldás közül választhatunk:

1. A MAX-ban tárolt virtuális feladatok és/vagy csatornák alkalmazásával állítjuk össze a mérésvezérlő programunkat.

Ezt a módszert akkor érdemes választani, ha a MAX-ban már felkonfigurált feladat, vagy csatorna megfelel a létrehozni kívánt feladatnak vagy csatornának. Ebben az esetben fölösleges újra elkészíteni azokat a programrészeket, ami egyszer már elkészült és a MAX-ban megtalálható.

2. Ha a MAX-ban nincs a kívánt feladatunknak vagy csatornáknak megfelelő virtuális megoldás, akkor vagy létrehozzuk, vagy a LabView programban felprogramozzuk az alább ismertetett függvényekkel. Ha olyan feladatot/csatornát kell létrehoznunk, amit feltehetően a későbbi feladatainkhoz is tudunk majd használni, akkor érdemes azt a MAX-ban letárolni.

I. TASK létrehozása, kezelése

Minden mérési feladatot un. "*Task*" keretében végzünk el. *Task* –ot hozhatunk létre virtuálisan a MAX-ban, a DAQ Assistant segítségével, vagy közvetlenül a LabView viban.

Egy *Task* keretében az alábbi paramétereket definiálhatjuk:

- Csatornák
- Időzítések
- Olvasás/írás
- Triggerelés

Ha a *Task* –ot a MAX-ban már létrehoztuk, akkor csak meg kell nyitni a LabView programunkban az alábbi ikonnal:

DAQmx Task Name %MAX_VIRT_TASK 🔽

Ugyancsak elérhetők a már létrehozott virtuális csatornák is:

DAQmx Global Channel % MyVoltageChannel 💌 A *Task* -kal az alábbi műveletek végezhetőek:



Kiegeszitő lenetőségek:

- Harver ellenőrzése az adott *Task* -ban foglalt feladatokhoz
- Hardver lefoglalása az adott *Task* -hoz
- Hardver felszabadítása

A *Task* paraméterei a *Task* futása közben nem változtathatóak meg. Egy *Task*-ban egy típusú feladat végezhető el. A *Task* –okat lehet szinkronizálni.

II. CSATORNÁK KONFIGURÁLÁSA

Az adatgyűjtő fizikai csatornáit a *Task* létrehozása után kell beállítani. Itt döntjük el, hogy a multifunkcionális kártya melyik egységét fogjuk használni.

Kétféle megoldás lehetséges a csatornák megadására.

Ha már van a MAX-ban virtuális csatorna elmentve, akkor az alábbi ikon alkalmazásával közvetlenül a *Task*-létrehozásakor hozzárendelhetjük a csatornát:

Konstans megadással: DAQmx Global Channel

Vezérlő megadással:



Ha nincs a MAX-ban megfelelő virtuális csatorna definiálva, vagy nem olyan, amire szükségünk van, akkor a *Task* létrehozása után a LabView programban tudunk virtuális csatornát definiálni.





Input terminal configuration: itt döntjük el, hogy közös földpontú mérést, vagy differenciál kapcsolású mérést végzünk. RSE (referenced single ended) módban a jeleket az AIGND ponthoz képest mérjük az AI <0...15> bemeneteken. DIFF (differencial) módban a jeleket az AIO-AI8; AI1-AI9; AI2-AI10....stb. páronkénti pontok között mérjük.

A maximum és minimum értékekhez a mérési tartományt állítjuk be. Lásd a táblázatot az 1. oldalon.

Fizikai (valós) csatornák kiválasztásához használhatjuk az alábbi menüpont ikonjait:





Azt, hogy analóg bemeneti, analóg kimeneti, digitális vagy más típusú csatornát akarunk kiválasztani, azt az inkonra jobb billentyűvel kattintva a menüből az I/O Name Filtering pontot választva tudjuk beállítani:

DAQmx Physical Channel	Filter Names	
Change to Control Change to Indicator	DAQmx Physical Channel	
Description and Tip	Allow Undefined Names	External Channels
DAQmx - Data Acquisition Palette	Allow Multiple Names	Internal Channels
Data Operations		Port/Line Filtering
Properties	Analog Input 💌	Lines Only
✓ Size To Text I/O Name Filtering	Analog Input Analog Output Digital Input Digital Output Coupler Input	OK Cancel Help
	Counter Output	

Trigger csatornákra a fentiekhez hasonlóan járhatunk el az alábbi ikonnal:

DAQmx Terminal 16/Dev2/PFI0 -

FONTOS MEGJEGYZÉS! Aki szimulált hardverrel dolgozik, annak a MAX-ban virtuális csatornákat kell létrehozni, és azokat lehet behívni.

III. MINTAVÉTELEZÉS IDŐZÍTÉSE (TIMING)

A mintavételezésre alkalmazott órajelek:

- AI Sample Clock (egy csatornáról vett minták időzítése)
- AI Convert Clock (két nem azonos csatornáról vett minta közötti időzítés)



1/Sample period = Sample rate (mintavételezési frekvencia)

1/Convert period = Convert rate (Konverziós frekvencia)

A "sample rate" az egy csatornára vonatkoztatott mintavételi frekvencia, a "convert rate" a minták közötti konverziós frekvencia.

$$f_{conv} \ge \text{csatornaszám} * f_{sampl}$$

Az időzítésekhez használható jelek:

• AI Sample Clock Signal

- AI Sample Clock Timebase Signal
- AI Convert Clock Signal
- AI Convert Clock Timebase Signal

AI Sample Clock Signal

Lehet külső vagy belső. Külső forrás a PFI <0...15> lehet. Egy jelre egy mintát vesz az összes beállított csatornáról, ami az adott task-ban definiálva van.

Az *AI Sample Clock Signal* jelet hardveresen is levehetjük a kártyáról a PFI csatornákon.

Él- vagy szintvezérelt üzemmódban minden órajelre a meghatározott PFI csatornán egy impulzus vagy egy négyszögjel jelenik meg.

AI Sample Clock Timebase Signal

- 20 MHz Timebase
- 100 kHz Timebase
- PFI <0..15>
- Analog Comparison Event (an analog trigger)

A fenti időalapok oszthatóak le igény szerint a 6251-es kártya AI Sample Clock jelének generálására.

AI Convert Clock Signal

Lehet külső vagy belső. Külső forrás a PFI <0...15> lehet. Egy jelre egy mintát vesz egy csatornáról.

Az *AI Convert Clock Signal* jelet hardveresen is levehetjük a kártyáról a PFI csatornákon.

Él- vagy szintvezérelt üzemmódban minden órajelre a meghatározott PFI csatornán egy impulzus vagy egy négyszögjel jelenik meg.

AI Convert Clock Timebase Signal

- 20 MHz Timebase
- AI Sample Clock Timebase

A fenti időalapok oszthatóak le igény szerint a 6251-es kártya AI Sample Clock jelének generálására.

A Sample Clock (SC) és a Convert Clock (CC) jelek helyes és helytelen időzítése

Az időzítést 4 csatornás mérésre, csatornánként 3 db mintára mutatjuk be:

Helyes időzítés:

1 SC jel után következik annyi CC jel, ahány csatornán mintavételeztünk, majd az utolsó CC jel után következik ismét a SC jel.



Helyes időzítés 1 közös órajellel:

Egyszerre indul a SC és CC jelsorozat, de az SC jeleket mindaddig figyelmen kívül hagyja a rendszer, amíg az utolsó CC jel le nem fut egy-egy mintasorozatban.



```
CC*csatornaszám értékétől nagyobb SC intervallumot igénylő mintavételezés
```

Ez az időzítés nem nevezhető helytelennek, csupán arról van szó, hogy a SC intervallumot nagyobbra választjuk, mint a CC intervallum és a csatornák számának szorzata. Az időzítés helyes mindaddig, amíg a SC intervallum egész számú többszöröse a CC intervallumnak, azaz

 $t_{conv} * N = t_{sampl}$ - ahol N egész szám

Ebben az esetben a rendszer figyelmen kívül hagyja, és nem mintavételez azokra a CC jelekre, amelyek az utolsó csatorna mintavételezése és a következő SC jel között vannak.



Helytelen időzítések:

A SC jel túl gyors a CC jel periódusához képest, azaz

 t_{conv} * csatornaszám + $t_{delay} > t_{sampl}$



A következő példában azt mutatjuk be, mi a következménye annak, ha a SC periódusa nem egész számú többszöröse a CC órajel periódusának. Ebben az esetben a csatornák mintavételezése a SC jelhez képest midig más-más időpillanatban kezdődik el.



Az időzítések LabView függvényei:

Mintavételi frekvencia beállítása (Sample Clock Rate):



X

>

Beállítandó paraméterek:

- mintaszám csatornánként
- mintavételezés módja: folyamatos vagy megadott mintaszámú
- mintavételi frekvencia (Sample Rate)
- időzítéshez alkalmazott időzítő egység vagy csatorna
- élvezérlés típusa

A Sample Clock Signal és/vagy a Convert Clock Signal egyéb beállításainak átállítása vagy lekérdezése a DAQmx Timing Property Node –dal lehetséges:

a konverzió frekvenciájának beállítása

IV. ANALÓG BEMENET TRIGGERELÉSE

Triggerelés alatt azt értjük, hogy a mintavételezést egy analóg vagy digitális csatornára kapcsolt jel indítja a beállított feltételek szerint.

Analóg triggerelés esetén analóg jelet alkalmazunk és erre valamelyik analóg bemeneti csatornát használjuk.

Digitális triggerelésre digitális jeleket és digitális csatorná(ka)t használunk.

Analóg triggerelésre használható csatornák: AI <0...15>;

Digitális triggerelésre használt csatornák: PFI<0...15>

A triggereléshez használható jelek típusai a programozás szerint:

- AI Start Trigger Signal
- AI Reference Trigger Signal
- AI Pause Trigger Signal

AI Start Trigger Signal (STS)

Ezt a jelet a mintavételezés indítására használjuk.

Ha a mintavételezést triggerelés nélkül indítjuk, akkor az a szoftveres utasítás hatására indul.

Ha triggerelést alkalmazunk, akkor a triggerelő jel indítja a mintavételezést. Lehet úgy nevezett pretiggerelt és posttriggerelt mérést is végezni. Ezeknél a műveleteknél a meghatározott trigger impulzus előtt, ill. után mintavételez a rendszer, meghatározott számú mintát.

Triggerelési módok:

Élvezérelt (Edge triggering) Élvezérelt hiszterézises Ablakvezérelt (Wndow triggering)

Analóg triggerelés esetén 5 különböző lehetőségünk van a jel indítására és leállítására. Az alábbiakban ezeket mutatjuk be.

1. Beállított szint alatti értékek mintavételezése



2. Beállított szint feletti értékek mintavételezése



3. Felfutó jelre induló hiszterézises triggerelés



4. Lefutó jelre induló hiszterézises triggerelés



AI Reference Trigger Signal (RTS)

Ezt a jelet a mintavételezés leállítására használjuk.

Ha beállítottuk az RTS jelet, akkor a mintavételezés elindítása után a rendszer folyamatosan mintavételez, de mindig csak annyi mintát tárol és görget, amennyi a "pretrigger" mintaszám. Ha például 26 mintát adtunk meg, akkor mindig az utolsó 26 mintát tárolja a FIFO-ban, a többit eldobja a mintavételezéssel szinkronizáltan. Ha érzékeli a RTS jelet, akkor a megadott pretrigger mintaszám darab mintát mér és tárol le, majd befejezi a mérést.



A fenti példában minden jel felfutó élre van vezérelve. Az összes beállított mintaszám 10, a pretriggerelt mintaszám 3, ami azt jelenti, hogy a RTS után 7 mintát fog lemérni és

elmenteni a rendszer. A STS után következő első órajelre elindul a mintavételezés, és mér a rendszer úgy, hogy mindig az utolsó 3 mintát tárolja. Az RTS jel után következő első órajel előtti 3 mintát és az utána következő 7 mintát menti el a rendszer.

A STS után a pretrigger mintaszámnak a FIFO-ban kell lennie ahhoz, hogy az RTS impulzus fogadható legyen. A fenti példán az első RTS felfutó élének időpillanatában még csak 1 mintát vett a rendszer, ezért ezt figyelmen kívül hagyja.

LabView függvény a trigger jelek beállítására:

Digitális start trigger jel:





Digitális reference trigger jel:



Referencia trigger alkalmazásakor nagyon hasznos lehet a grafikonon a referencia trigger jel időpillanatának megjelenítése egy kurzorral. Erre külön függvényt találunk a könyvtárban:

Trigger Cursor.vi

Waveform Graph Reference ——]
Pre-Trigger Scans –	cursor	error out
error in (no error) 🔜		I

This VI is used to place a vertical cursor on a waveform graph at the location of a reference trigger. "CrsrList" is a property of a Waveform Graph and defines the cursors' style, color, position, etc as they appear on the graph. See the LabVIEW Help Index under "Cursor List" for more information.

Note: The Cursor.VI's purpose is purely aesthetic and can be removed from the calling VI without affecting it's functionality.

Analóg start trigger jel:





Analóg ablak trigger jel:



V. OLVASÁS / ÍRÁS FÜGGVÉNYEI

Jelek kiolvasását illetve kiküldését feszültség vagy bináris (kvantum érték) formátumban is elvégezhetjük.

A kiolvasás függvénye alatt található mezőben állíthatjuk be, hogy analóg vagy digitális értékeket olvasunk be, hogy egy- vagy többcsatornás mérés eredményeit kívánjuk kiolvasni, hogy feszültség vagy skálázatlan kvantum értékeket kívánunk kiolvasni.

Ha a csatornánkénti mintaszámhoz -1 értéket írunk, akkor a rendelkezésre álló számú minta kerül kiolvasásra.

Meghatározhatjuk, hogy a olvasáshoz/küldéshez maximálisan mennyi időt engedélyezünk. Ha a timeout paraméter értékéhez -1 értéket írunk, akkor addig vár a rendszer, amíg az összes szükséges mintát ki nem olvassa. Ha 0 értéket írunk, akkor egyszer megkísérli kiolvasni a szükséges adatmennyiséget, ha ez sikerül, akkor 0 status értéket kap a függvény, vagyis sikeres függvényfuttatással tér vissza, ha nem, akkor kiolvas annyi mintát, amennyi rendelkezésre áll, és idő túllépés (timeout) hibakóddal tér vissza a függvény. A paramétert érdemes úgy beállítani, hogy a mintavételezéshez szükséges idő (minta/mintavételi frekvencia) értékéhez néhány másodpercet hozzáadunk. Triggerelés esetén figyelembe kell venni, hogy mennyi időt kívánunk engedni a trigger jel, vagy jelek megjelenésére.



Az értékeket kiolvashatjuk jelalak formátumban, ilyenkor az időtengely adatait is automatikusan megkapjuk a mintavételezés időzítési adataiból, vagy kiolvashatjuk csak a mért feszültség értékeket az időzítési adatok nélkül:

	Context Help	X
	DAQmx Read (Analog 2D DBL NChan NSamp).vi	^
Analog 2D DBL NChan NSamp	task/channels in Channels in channel Channel Channel Channel Channel Channel Channel	
	Reads one or more floating-point samples from a task that contains one or more analog input channels.	
	Detailed help	~
	æ ₿? <	٤

Bináris értékek kiolvasása:

	Context Help	×
	DAQmx Read (Analog 2D I16 NChan NSamp).vi task/channels in """""""""""""""""""""""""""""""""""	^
NChan NSamp	number of samples per channel 60	
	Reads one or more unscaled 16-bit signed integer samples from a task that contains one or more analog input channels.	
	Detailed help	~
	Image: Second secon	

Folyamatos mintavételezés esetén a beolvasás függvényét egy while ciklusba kell helyezni, amelyet a mérés leállítása vezérel.

Analóg jelek küldése kimenetre feszültség értékben (double típus):

A paraméterezés több ponton hasonló a beolvasás paraméterezéséhez. Az "auto start" paramétert akkor érdemes beállítani, ha valamilyen okból a programba nem kívánunk Start Task függvényt tenni.

	Context Help	×
DBO mx	DAQmx Write (Analog Wfm 1Chan NSamp).vi	^
NOC	auto start	
Analog Wfm 🖕	data	
1Chan NSamp	error in error out	
	Writes a waveform to a task that contains a single analog output channel.	
	Detailed help	
		~
	······································	2.3

Egy egyszerű, csak konstansokkal felépített, digitális start és referencia triggereléssel ellátott egycsatornás mintavételezés blokk diagramja:



1. Mintavételezés és adatfeldolgozás idő és frekvencia tartományban

FELADAT

Készítsen LabView programot, amely képes a felhasználó által kijelölt csatornákat mintavételezni. A beállított mintavételi frekvenciával a program leméri a kért mintaszámot. Ezután legyen lehetőség a mért csatornák közül egyet kiválasztani és azt külön megjeleníteni egy zoom-olható grafikonon. Ezen kiválasztott csatornának legyen lehetőség meghatározni az FFT spektrumát. A program számítsa ki külön a jel effektív értékét és az alapharmonikus nagyságát. Legyen lehetőség megadni a jel középértékét és a szórását, ill. a szórás négyzetét a varianciát.

1.1. A program magjának elkészítése

Elsőkén egy mintavételező taszkot kell létrehozni. Ez a mi esetünkben a "meres_feladat" nevet kapta. Ügyeljünk, hogy ékezetes betűk és különleges karakterek, ne kerüljenek a taszk nevébe, mert a program ugyan nem fog fordításkor hibát generálni, de a futás során nehezen megfejthető hibákat fog generálni. A taszk létrehozást követi a csatorna kiválasztás és paraméterezés, ami ugyan csak nagyon lényeges.



A Context Help ablak segítségével láthatjuk azon paramétereket (vastagon szedve), amiket mindenképen meg kell adnunk, mert addig nem lesz a program futóképes. Jelen esetben látható, hogy a fizikai csatornák megadása nélkül nem fog tudni a program elindulni. A program helyes működéséhez az is kell, hogy a max. és min. értéket és a bemenet konfigurálást is megadjuk. Esetünkben ez 5V és -5V ill. a bemenetet pedig közös földpontúra állítsuk be (RSE).

Ezeket az értékeket a kimenetre állított kurzor esetén a jobb gomb és Create / Constant segítségével tudjuk legenerálni.





1. ábra Fix mintás.vi

Ezt követően egy eágazáshoz ér a program amely vizsgálat azt nézi meg, hogy a mintavételi frekvencia kisebb vagy egyenlő mint a mintavételi frekvencia maximális értéke (SampClk.MaxRate). Ez az érték függ attól, hogy hány csatornán mérünk és ezt csak akkor tudjuk ha már inicializáltuk a mérendő fizikai csatornákat.

Ha ez nem igaz akkor a kijelzőn jelenjen meg egy üzenet, hogy "Túl nagy volt a beállított mintavételi frekvencia!". Emellett a max megaható értékkel írja felül a Mintavételi frekvencia értékét. Emellett a grafikon kimenetére használhatjuk a Use Default If Unwired lehetőséget a kimeneti tunelnél.



A mintavételi frekvencia felülírását a property node segítségével tehetjük meg.

Ha helyes a mintavételi frekvencia megadása, akkor a Case ciklus True ága hajtódik végre. Ami beállítja, hogy véges elemszámú mintát szeretnénk gyűjteni és, hogy mennyi legyen konkrétan ez a mintaszám. A mintavételező függvényt több csatorna és több mintára állitva, egy és több csatorna esetén is működni fog a program. Viszont ha csak 1 csatornán mintavételezünk akkor a mérő taszkunk nem tudja teljesen kiaknázni a hardver által nyújtott lehetőségeket!



Ha ebbe az igaz ágba, be tud futni a program akkor a mintavételi frekvencia biztosan kisebb mint a maximálisan megadható, így az esetleg hamarabb kiírt üzenetet, hogy "Túl nagy volt a beállított mintavételi frekvencia!" el kell tünteni.

Ezekután nincs más hátra mint a mintavételező taszkot lezárjuk és töröljük.



Ha a DAQmx Read sub.vi task out kimenete főlé mozgatjuk az egérmutatót, akkor a fenti módon azonnal a keresett palettára kerülünk további kattintások nélkül. Innen a Stop és Clear kitétele lezárja az eddigi taszkunkat.

A 40. ábrán még további két Property Node található amely a progamunk hibás adatbevitel elleni védelmét hivatott megoldani.



A DAQmx Create Channel után lévő task out szálra csatlakozva megkaphatjuk a csatorna számmal és a mintavételi frekvenciával kapcsolatos információkat. Mivel ez a két adat a mért csatornák számától függ, a látható pontban már megkaphatjuk a kívánt adatokat.



Az alábbi Timing Node segítségével adatokat kaphatunk a mintavételezéssel kapcsolatban. Nekünk most a maximálisan megengedhető mintavételi frekvencia értékét kellene megtudnunk. Így megint az alapbeállításra (bal) gombbal kattintva kiválaszthatjuk a SampClk.MaxRate ehetőségét. Amelynek az értékét jobb gombbal kattintva egy indikátort létrehozva ki is tudjuk iratni.

AI Voltage	EL II Clean Up Wire Create Wire Branch Delete Wire Branch		alog 1D Wfm	000000 0000 		one ,	Spe	ctral
rencia	Visible Items	•		idötartan	Measu	gnals An	ip(RMS) Sig	nals FFT
V	Insert	۲		•Visible	Amp	olitude •	FFT - (Peak) •
	DAQmx - Data Acquisition Palette	•			DAQmx Timing	Property Node	Pasteras als	
	Create	۲	THESE	CHEN!+	DR0.mx	DRIMK 63	DROmx	DROMM
	Probe Custom Probe Breakpoint	•	Task Const	Channel Const	Create Chan	Read	Write	Wait
	Description and Tip			Timing	Triggering	Start	Stop	Clear
			Channel Node	Timing Node	Triggering N	Read Node	Write Node	×
			DAQ Assist	Real-Time		Dev Config	Task Config/	Advanced

1.2. Mérés időtartamának vizsgálata

A mérési idő akár hoszú ideig is tarthat. Ha szeretnénk a véletlen helytelen adatbevitelt elkeülni, akkor készíthetünk egy kis rutint amely kiszámolja, hogy mennyi ideig fog tartani a mérés, és ha ez egy általunk megadott limitértéket meghalad akkor a program nem fog mérni.



Elsőként a mérés várható idejét kell meghatároznunk. A mintavételi frekvencia reciproka megadja azt az időt amely két minta levétele között eltelik. Ha ezt megszorozzuk a mintaszámmal akkor megkapjuk a mérés várható idejét. Ha ez a fenti minta szerint kisebb mint 5s akkor a mérést végre fogja hajtani. Persze csak akkor ha megnyomjuk a mérés gombot. Tehát ezt a két feltételt AND és kapcsolatba kell hoznunk, amely csak akkor igaz ha mindkettő igaz.

Ha nagyobb tehát nem igaz akkor a mintavételi frekvenciát megszorozza 4,9-el amely biztosan 5-nél kisebb lesz és ezt az értéket írja be a minták számához.

1.3. Az FFT spektrum meghatározása

Gyakori feladat, hogy a mért jel frekvencia komponenseit is szeretnénk ismerni, hogy esetleg milyen felharmónikusokat tartalmaz a mért jel.



Ennek megvalósítására a Spectral Measurement függvényt érdemes alkalmaznunk. Ezen függvényt konfigurálnunk kell használat előtt, annak megfelelően, hogy milyen skálában szeretnénk megkapni az eredményeket, vagy milyen ablakozó függvényt szeretnénk alkalmazni a számítás során. 3 alap beállítástól eltérő értéket kell beállítani Amplitúdó: Magnitude (Peak), Megjelenítés skálája: Linear, Ablakozás: None.



Az eredmény grafikonjának beállításánál figyelni kell arra, hogy a frekvenciák diszkrét értékek nem lehet őket összekötni. Tehát a kirajzolásnál erre a beállításra kell figyelni az alábbiak szerint.

0,22	0 23707 Plot 0	Common Plots		0 22 Plot 0	Common Plots	•
		Color Line Style Line Width Anti-Aliased			Color Line Style Line Width Anti-Aliased	
400	425	Bar Plots Fill Base Line Interpolation Point Style		425	Bar Plots Fill Base Line Interpolation Point Style	
		X Scale			X Scale Y Scale	
		Export •			Export	•

Így a megjelenő grafikon sávos spektrumot fog megjeleníteni, amely jelen esetben elvárt.



Sokszor az alap harmonikus frekvencia értéke is kérdés ekkor a függvény maximális értékét kell kikeresnünk a tömb elemek közül. Erre egy módszert mutatja az alábbi ábra:



A másik módszer egy kicsit egyszerűbb, csak a megfelelő sub.vi-t kell alkalmazni (Tone Measurement):



A két eredményt egymás mellett kiíratva látható, hogy azok csak kismértékben térnek el. Tehát mindkét módszer használható a probléma megoldására.

1.4. A középérték, a szórás és a variancia meghatározása

A Labview-ban a fenti problémák számítására ugyancsak találunk beépített függvényeket. "Std. Deviation And Variance" számító függvény. Ezen sub.vi alkalmazásával megadható mindhárom kérdéses érték a felhasználó számára. Továbbá megadható még a "Measures of Spread" sub.vi segítségével Spread Value a szóródás értéke is.



A kérdéses elemek a fent látható Mathematics könyvtárban találhatóak. Az értékek megjelenítéséhez csak egy-egy indikátort kell a kimeneteken generálnunk és máris látható az eredmény a numerikus kijelzőn.



A program felhasználói felülete (Front Panel)



A teljes program Block Diagram-ja



2. Az analóg kimenet programozása

FELADAT

Készítsen LabView programot, amely képes a felhasználó által generált jelet az adatgyűjtő kártya analóg kimenetén folyamatosan kiadni. A megjelenítendő jel frekvenciáját és jeltípusát tudja a felhasználó állítani. Legyen lehetőség a sampling info állítására is. A kimeneten megjelenő jelet mintavételezze az egyik analóg bemeneti csatornán, és jelenítse meg a kimeneten valóban megjelenő jelalakot egy grafikonon.

2.1. A program két fő taszkból épül fel

A program a Main While cikluson belül tartalmaz egy Case struktúrát amelyet a Generálás gomb vezérel. A gomba mechanikai működése fontos, mert ha az alapbeállításon hagyjuk, akkor az első bekapcsolás után folyamatosan generálni akar majd a program, amiből nem fogunk tudni kilépni. Tehát ezt megelőzendő a Generálás gomb működését kell beállítanunk a mellékelt ábra szerint.

Ezek után egy taszkot kell létrehozni. Amely nevének megadása után a csatorna létrehozása és paramétereinek beállítása következik. Jelen példa esetében a kimenet konfigurálása a cél. Meg kell adni a kimeneti csatornát amelyet, a melléklet módon tudunk beállítani. Ezek után jöhetnek a határok és az "output terminal configuration", ahol megint csak a közös földponthoz képesti (RSE) mérést kell beállítanunk.

Majd következhet az időzítés beállítása. Itt viszont az első kötelező

megadandó érték "rate" a mintavételi frekvencia. Viszont ez az érték függ attól, hogy milyen jelet állítunk elő. Tehát a jel generálása alapján létrehozható "sampling info" két értékét kell ennek megadásához felhasználnunk. A két értéket szét kell választanunk az "Unbundle By Name" funkció segítségével. Az Fs a mintavételi frekvenciát jelenti, #s pedig a minta számot jelenti, mivel ez utóbbi csak pozitív egész lehet, ezért található egy típus konverzió is (Numeric / Conversion / U32) az adatfolyamban.

Közvetlen mellette található a Sample Mode amely lehet Finite Samples vagy Continous Sample. Jelen esetben, ha nem a Continous Sample esetet választjuk akkor a kimeneten csak egy pillanatra fog megjelenni a jel és utána azonnal eltűnik onnan.









i.

60

folyamatos stop

TF

Analog Wfm

1Chan NSamp

A mért jelet egy "Strip Chart" grafikonon kell megjelenítenünk. Amely olyan elemek megjelenítésére alkalmas aminek az elemszámát előre nem ismerjük. Így a grafikon méretének megfelelő számú (alapbeállításban 1024db) utolsó adatot látjuk csak a képernyőn, és ezek a mérés során folyamatosan frissülnek.

A program felhasználói felülete:

*

.....

AI Voltage 🔻 Sample Clock 🔻

100

Generálás		Programból kilépés	
	0. csatorna j	jele 🕂 😰 🕅 Dev	3/ai0
sampling info Fs 1,00k #s 1000 frequency 100,00	5- 4- 3- 2- oppn1- 0- uw -1- -2- -3- -4- 1:00:00,000 1904.01.01.	Īdő	1:00:02,041 1904.01.01.
Kimeneti csatorna ¹ / ₀ Dev3/ao0 Késleltetés 1 · · · · · · · · · · · · · · · · · · ·	T	A program az AO0-ás kimeneten generál egy 100Hz-es szinusz jelet. Ami ha vissza van kötve AI0-ra akkor azt folyamatosan mintavételezi és megjeleníti.	

A program belső while ciklusában található még egy várakozás, amely arra szolgál, hogy a processzornak biztosítson egy kis szabad időt. Így a jelek kirajzolása nem lesz olyan "rohanó", ezzel lehetséges a kirajzolás lelassítása.

Man Inchastor	T Tunnanananananananananananananananananan
Kesleltetes	
0321	L

A program futtatása során hamar észrevehetjük (főleg ha bekapcsoljuk a Highlight Executin" debug lehetőséget), hogy a program a generálás során a paraméterezés után eljut a belső While ciklushoz és mindaddig csak ott forog amíg a felhasználó úgy nem dönt, hogy kilép a generálásból. Tehát ez esetben a proram például a külső ciklus "Programból kilépés" gombját nem veszi figyelembe, tehát azt hiába nyomkodjuk. Ugyanez elmondható a "folyamatos stop" generálásból való kilépés gombról is, mivel ha nem

generálunk hiába nyomkodjuk a fent említett gombot nem történik semmi. Ennek a megoldására a már korábban megismert "Visible" / "Invisible" lehetőséget használjuk. A Generálás gombbal is hasonló módon járhatunk el. A generáláson kívül a jobb oldali beállításokat használhatjuk, míg a generálás ciklusában ellentétes konstans értékekkel kell megadnunk ugyanezeket.

Generálás Folyamatos stop

Programból kilépés

A következő felmerülő probléma, hogy a frekvency és a sampling info értékeit is tudjuk módosítani mivel azok is "control" tipusúak. Viszont he a generálást bekapcsoljuk akkor ezt a változatást nem veszi

figyelembe a program futó szálja. Ezért ezeket is el kéne tüntetni, ha generál a kimeneten a felhasználó. Az eltüntetés jelen esetben nem az ideális megoldás mivel a generálás közben is esetleg szeretnénk rápillantani a generált jel frekvenciájára vagy a minta adási frekvenciára. Tehát ezeket a kontrollokat csak "Disabled" álapotba kellene tenni és elszürkíteni. Erre az alábbi lehetőség használható. A sampling info és a frequency kontrol valamelyikére kattintunk majd Property Node és Disabled. Ezekután Change To Write, hogy mi tudjunk neki értéket adni. Majd következhet jobb gomb és Create Constant. A generálás ciklusán kívülre "Enabled", belülre pedig "Disabled And Grayed Out" konstanst kell megadnunk.





Végül a program egy hibás értékmegadás folyamán leállhat. Ez akkor áll elő amikor a felhasználó úgy adja meg az adatokat hogy a Shannon törvényt sem tartja be. Ezt nagyon egyszerű ellenőrizni még a generálás indítása előtt is akár.



Az fenti programrész tehát kiolvassa a sampling info "Fs" (mintavételi frekvencia) értékét és összehasonlítja a generálandó jel frekvenciájának 2,3 szeresével. Ami a minimális 2 szeres plusz 1 minta gyakorlati megfogalamzása szokott lenni. A programrészletből jól látható, hogy ha a feltétel igaz akkor minta előállítási frekvencia értékét módosítja a 2,3 szeres szorzat értékére.

A teljes program Block Diagram-ja:

