

NI PCI-6251 multifunkciós mérésadatgyűjtő kártya specifikációja

ANALÓG BEMENET

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

Bemeneti feszültség tartományok	Felbontás (5%-os over range)
-10 V ... 10 V	305 μ V (320 μ V)
-5 V ... 5 V	153 μ V (160 μ V)
-2 V ... 2 V	61 μ V (64 μ V)
-1 V ... 1 V	30,52 μ V (32 μ V)
-500 mV ... 500V	15,3 μ V (16 μ V)
-200 mV ... 200 mV	6,1 μ V (6,4 μ V)
-100 mV ... 100 mV	3,05 μ V (3,2 μ V)

Bementi impedancia	10 G Ω
Bemeneti áram	100 pA
Bementi FIFO mérete	4095 minta
Adat transzfer mód	DMA Megszakítás Programozott

Analóg trigger

Trigger jelek száma	1
Trigger csatornák	AI<0...15>, AIPF0

Digitális trigger

Trigger csatornák	PFI<0...15>
-------------------	-------------

ANALÓG KIMENET

Csatornaszám	2
DA felbontása	16 bit
Maximális frissítési frekvencia	2 MHz
Jelgenerálás esetén a beállási idő	2 μ sec/LSB
Feszültség tartomány	± 10 V; ± 5 V
Áramterhelhetőség	± 5 mA
Adat transzfer mód	DMA Megszakítás Programozott

DIGITÁLIS I/O

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

SORKAPOCS

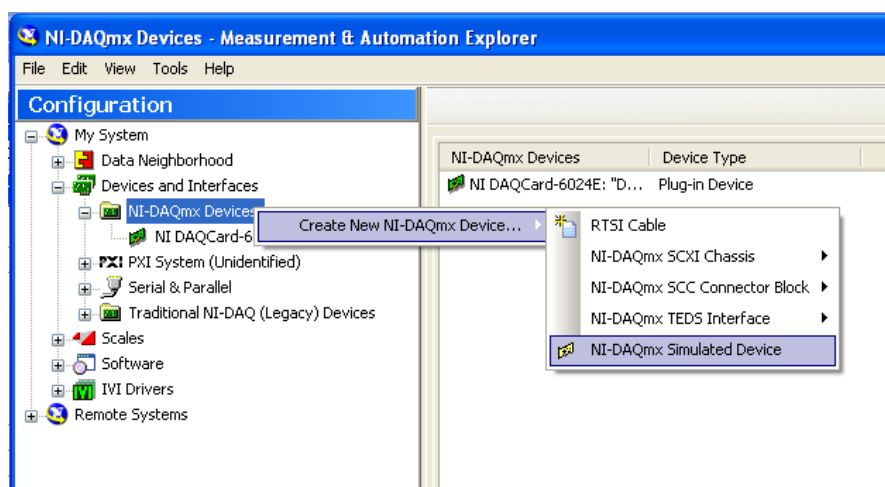
AI 0	68	34	AI 8
AI GND	67	33	AI 1
AI 9	66	32	AI GND
AI 2	65	31	AI 10
AI GND	64	30	AI 3
AI 11	63	29	AI GND
AI SENSE	62	28	AI 4
AI 12	61	27	AI GND
AI 5	60	26	AI 13
AI GND	59	25	AI 6
AI 14	58	24	AI GND
AI 7	57	23	AI 15
AI GND	56	22	AO 0
AO GND	55	21	AO 1
AO GND	54	20	APFI 0
D GND	53	19	P0.4
P0.0	52	18	D GND
P0.5	51	17	P0.1
D GND	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
D GND	44	10	PFI 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
PFI 7/P1.7	38	4	D GND
PFI 8/P2.0	37	3	PFI 9/P2.1
D GND	36	2	PFI 12/P2.4
D GND	35	1	PFI 14/P2.6

Szimulált hardver alkalmazása

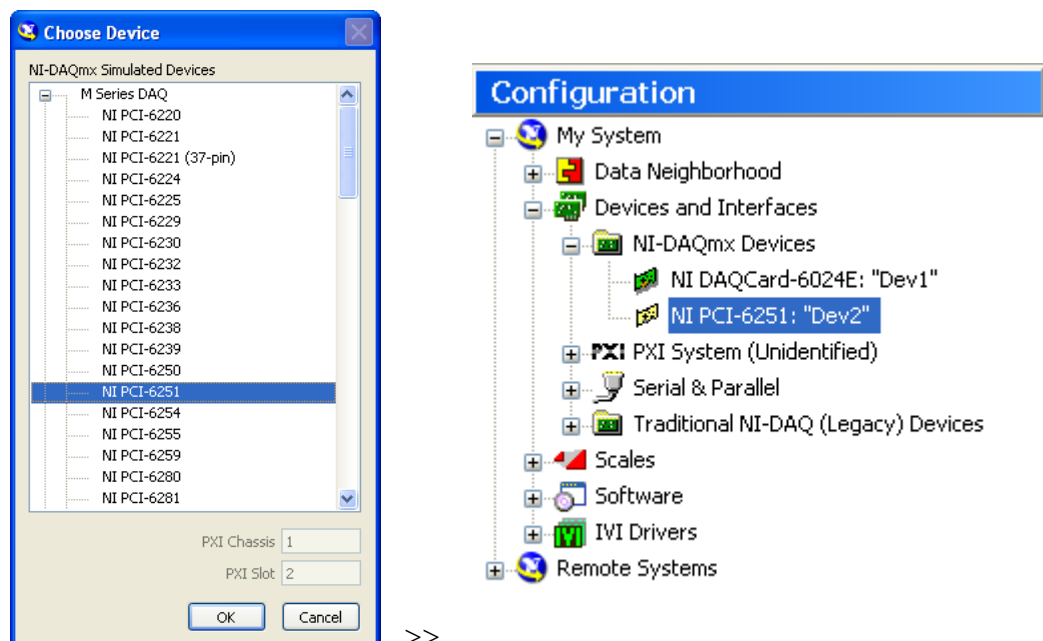
DAQmx rendszerben használhatunk szimulált mérésadatgyűjtő kártyát, ami nagyon hasznos lehet a mérésekre történő otthoni felkészülésnél.

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.



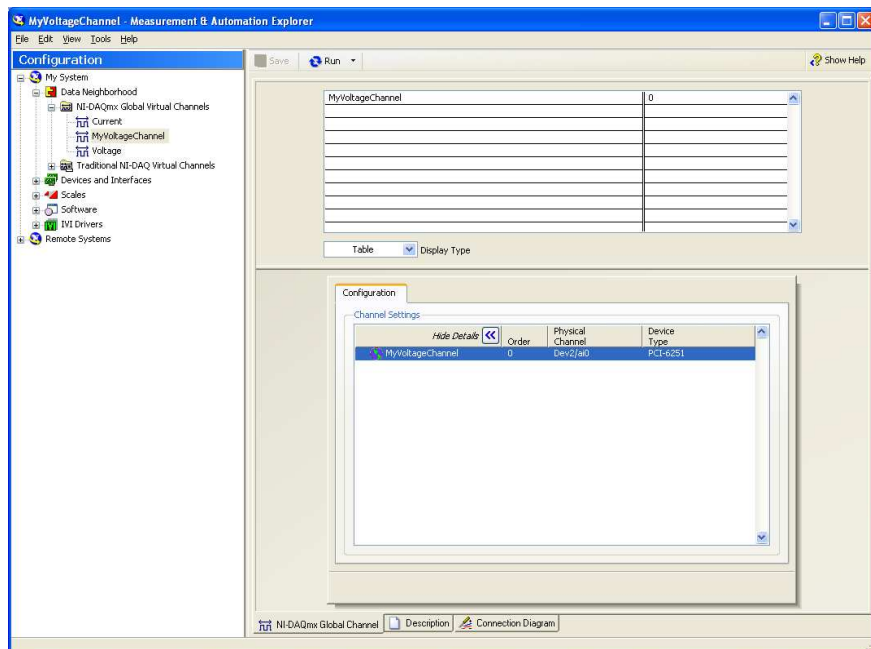
A megjelenő listából válasszuk ki azt az adatgyűjtőt, amit a laborban is használni fogunk, azaz az M sorozatú kártyák közül a 6251 típust.



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:



Az így létrehozott csatornákat használhatjuk a LabView programunkban.

A virtuális task létrehozása a fentiekhez teljesen hasonlóan történik, azzal a különbséggel, hogy itt be kell 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 főlegesen újra elkészíteni azokat a programrészeket, ami egyszer már elkészült és a MAX-ban megtalálható.

2. Ha a MAX-ban nincs a kívánt feladatunknak vagy csatornának 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.

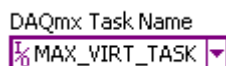
I. TASK létrehozása, kezelése

Minden mérési feladatot ún. „**Task**” keretében végzünk el. **Task** –ot hozhatunk létre virtuálisan a MAX-ban, a DAQ Assistant segítségével, vagy közvetlenül a LabView vi-ban.

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:

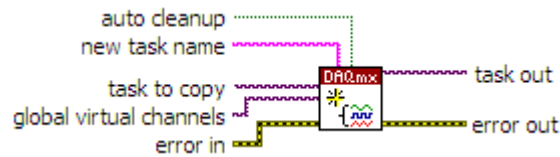


Ugyancsak elérhetők a már létrehozott virtuális csatornák is:



A **Task** -kal az alábbi műveletek végezhetőek:

- Létrehozás



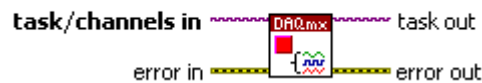
- Futtatás indítása

DAQmx Start Task.vi



- Futtatás leállítása

DAQmx Stop Task.vi



- Törlés

DAQmx Clear Task.vi



- Ellenőrzés, várakozás

DAQmx Wait Until Done.vi



DAQmx Is Task Done.vi



Kiegészítő lehetőségek:

- 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.

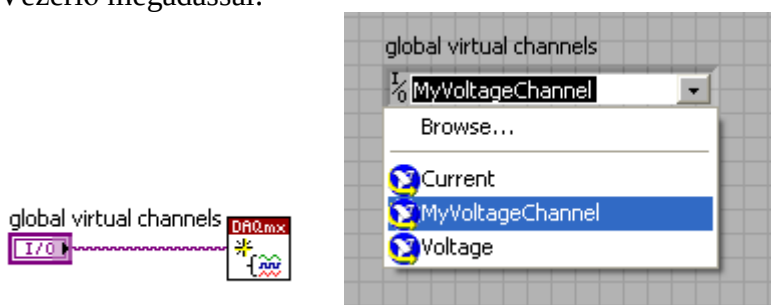
Kétféle megoldás lehetséges a csatornák megadására.

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

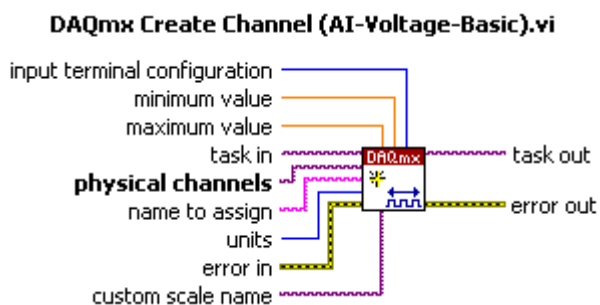
Konstans megadással:



Vezérlő megadással:



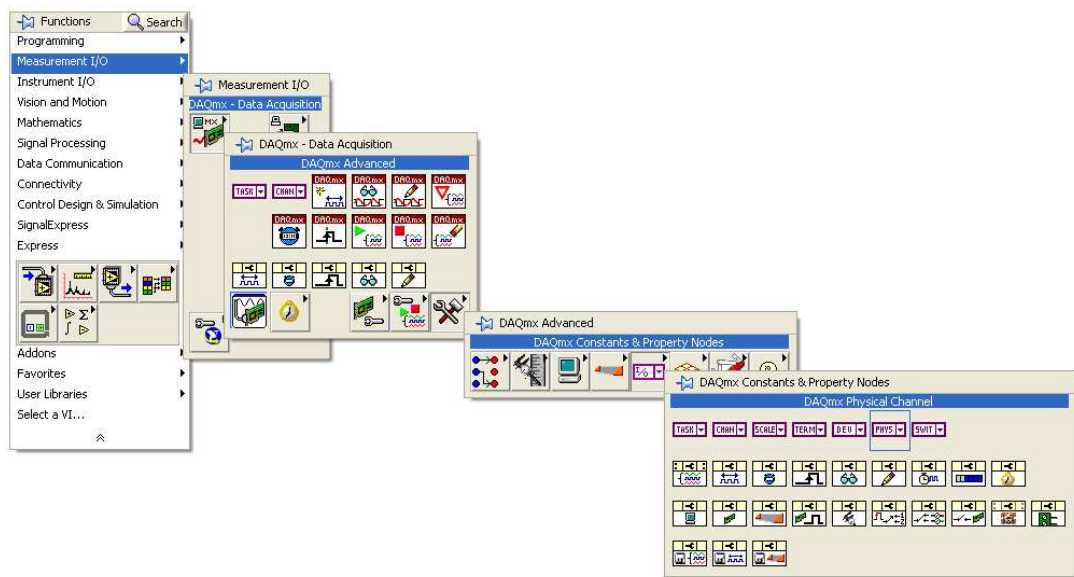
Ha nincs a MAX-ban megfelelő virtuális csatorna definiálva, vagy nem olyan, amire szükségünk van, akkor a **Task** létrehozása után a LabView programban tudunk virtuális csatornát definiálni.



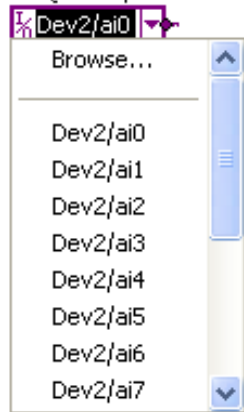
Input terminal configuration: itt döntjük el, hogy közös földpontú mérést, vagy differenciál kapcsolású mérést végzünk. RSE (referenced single ended) módban a jeleket az AIGND ponthoz képest mérjük az AI <0...15> bemeneteken. DIFF (differencial) módban a jeleket az AI0-AI8; AI1-AI9; AI2-AI10....stb. páronkénti pontok között mérjük.

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

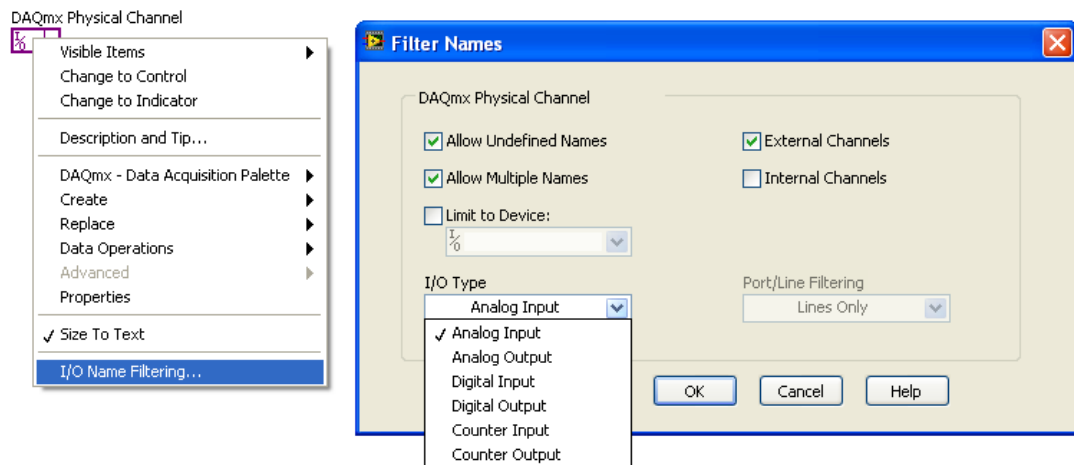
Fizikai (valós) csatornák kiválasztásához használhatjuk az alábbi menüpont ikonjait:



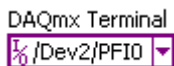
DAQmx Physical Channel



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



Trigger csatornákra a fentiekhez hasonlóan járhatunk el az alábbi ikonnal:

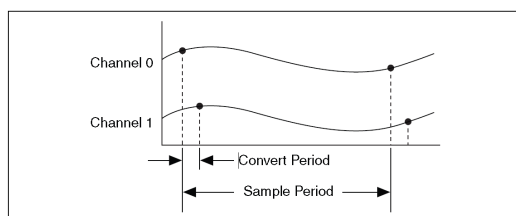


FONTOS MEGJEGYZÉS! Aki szimulált hardverrel dolgozik, annak a MAX-ban virtuális csatornákat kell létrehozni, és azokat lehet behívni.

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)



$1/\text{Sample period} = \text{Sample rate}$ (mintavételezési frekvencia)

$1/\text{Convert period} = \text{Convert rate}$ (Konverziós frekvencia)

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

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

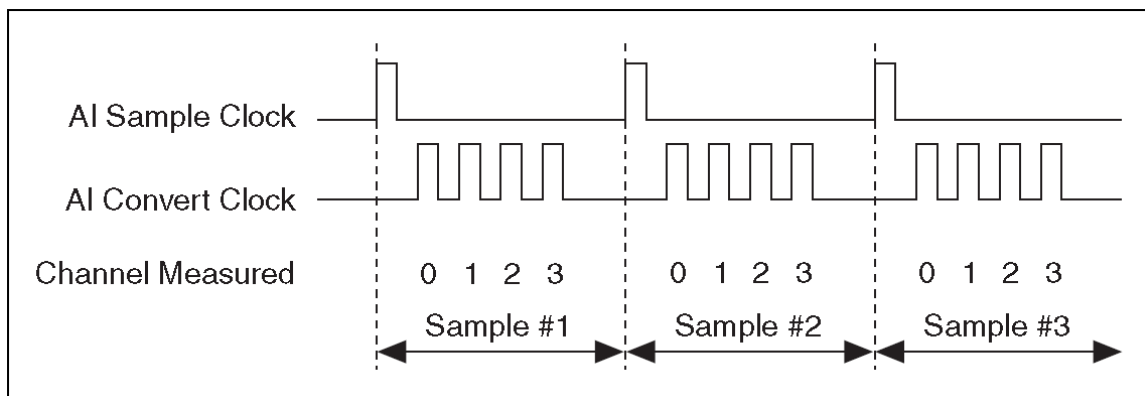
A fenti időalapok oszthatóak le igény szerint a 6251-es kártya AI Sample Clock jelének generálására.

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:

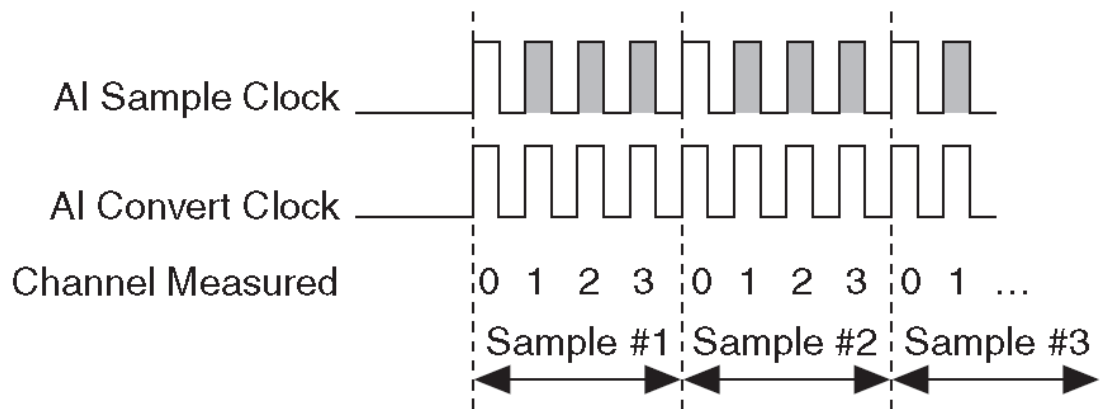
Helyes időzítés:

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.

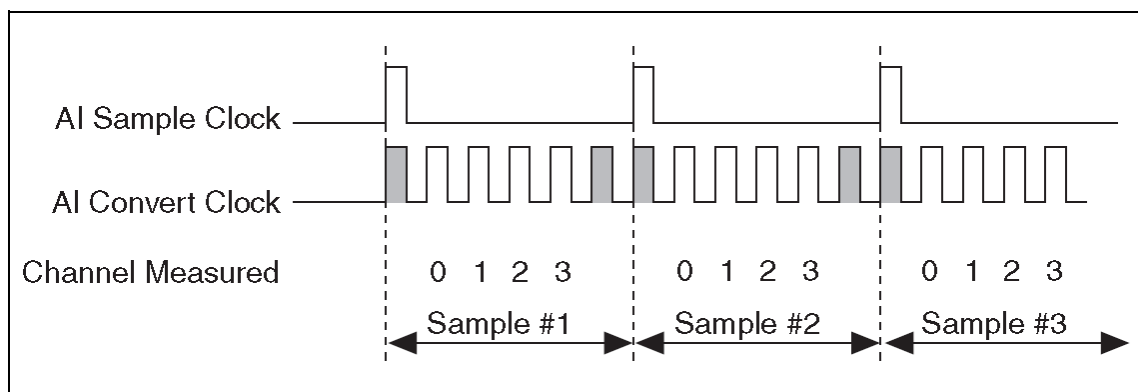


CC*csatornaszám értékétől nagyobb SC intervallumot igénylő mintavételezés

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

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

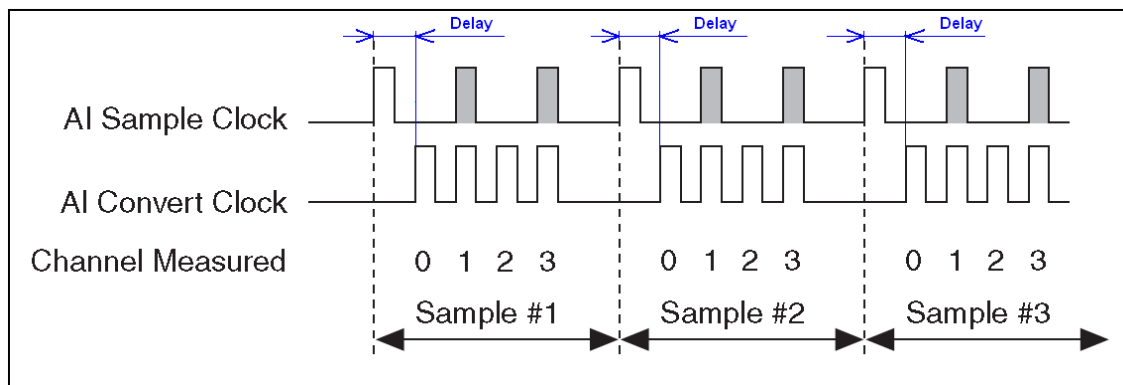
Ebben az esetben a rendszer figyelmen kívül hagyja, és nem mintavételez azokra a CC jelekre, amelyek az utolsó csatorna mintavételezése és a következő SC jel között vannak.



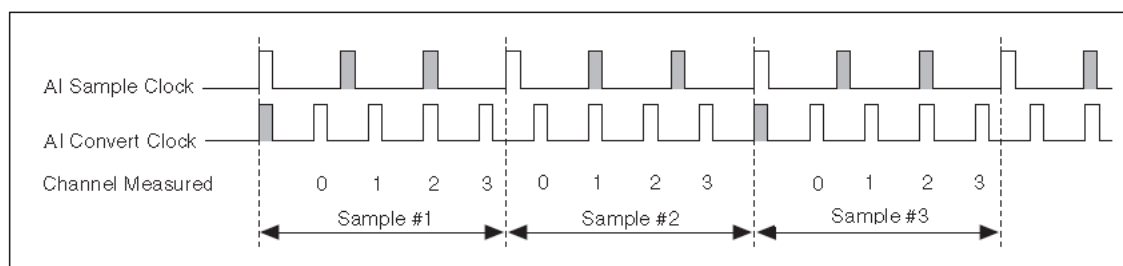
Helytelen időzítések:

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

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

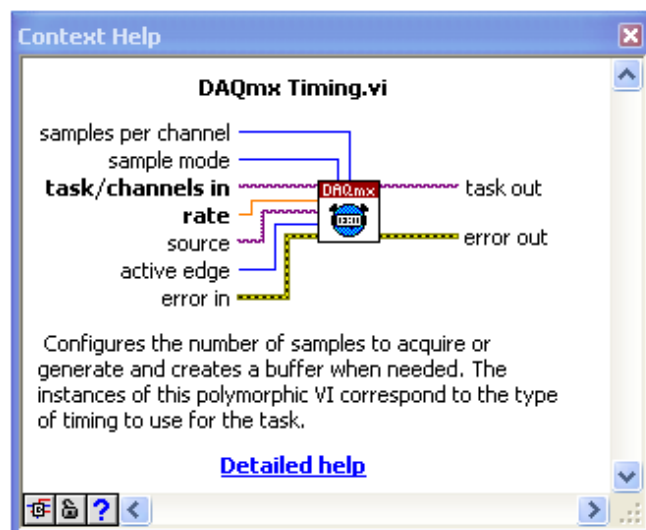
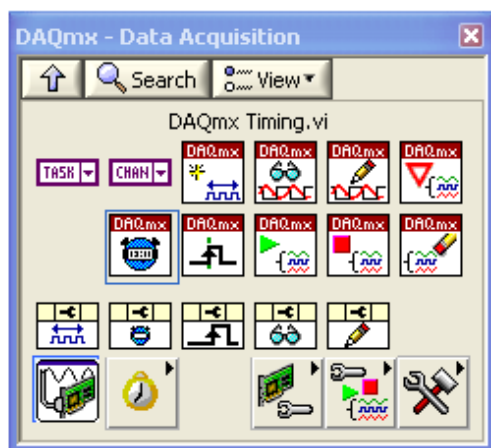


A következő példában azt mutatjuk be, mi a következménye annak, ha a SC periódusa nem egész számú többszöröse a CC órajel periódusának. Ebben az esetben a csatornák mintavételezése a SC jelhez képest mindig más-más időpillanatban kezdődik el.



Az időzítések LabView függvényei:

Mintavételi frekvencia beállítása (Sample Clock Rate):



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 pretriggerelt és posttriggerelt mérést is végezni. Ezeknél a műveleteknél a meghatározott trigger impulzus előtt ill. után mintavételez a rendszer meghatározott számú mintát.

Triggerelési módok:

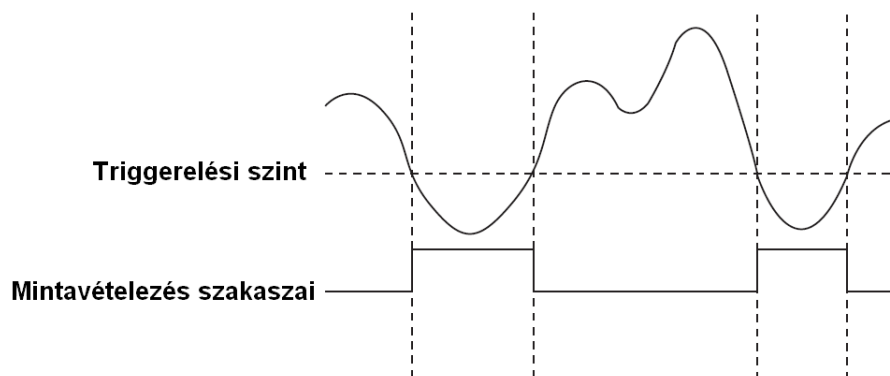
Élvezérelt (Edge triggering)

Élvezérelt hiszterézises

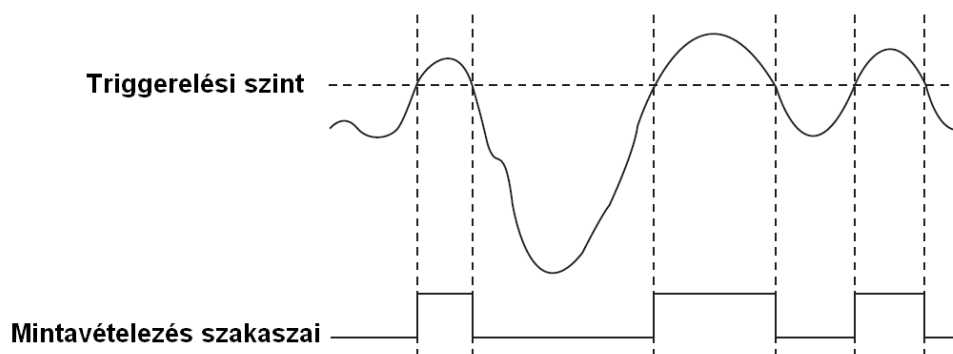
Ablakvezérelt (Window triggering)

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

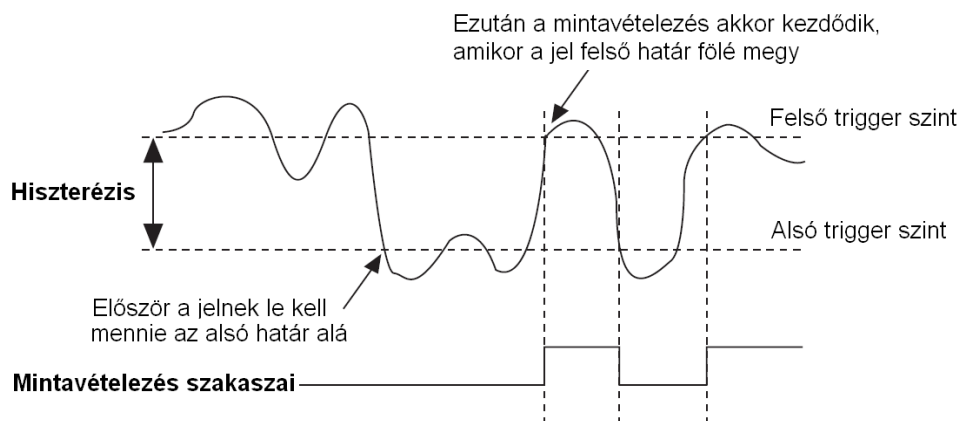
1. Beállított szint alatti értékek mintavételezése



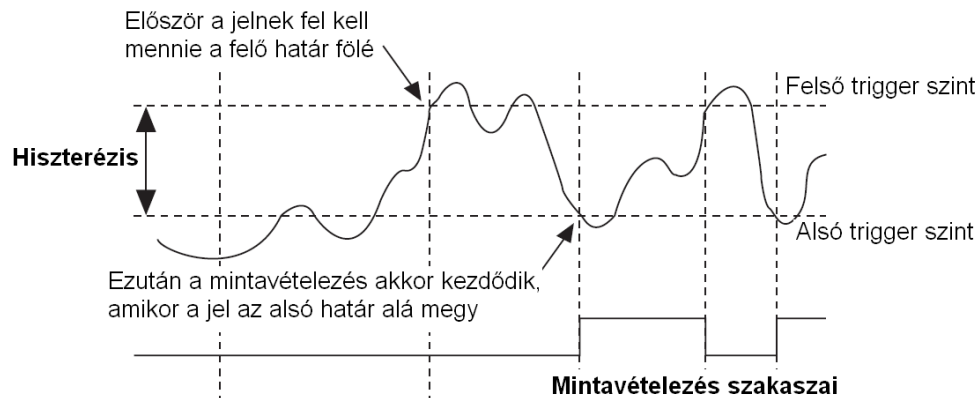
2. Beállított szint feletti értékek mintavételezése



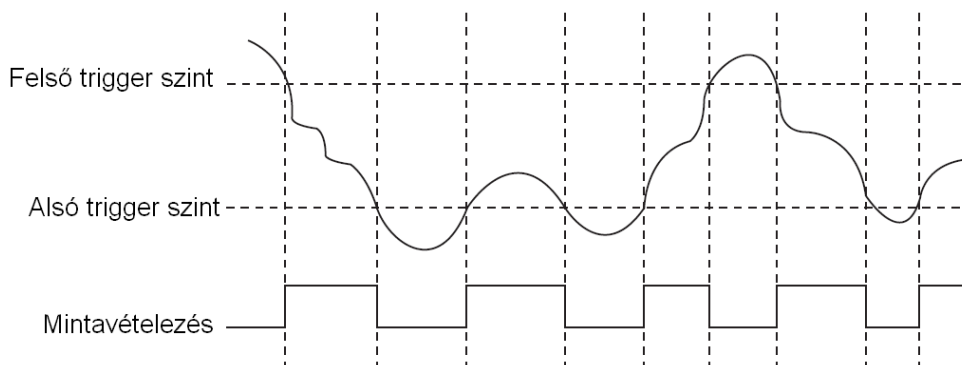
3. Felfutó jelre induló hiszterézises triggerelés



4. Lefutó jelre induló hiszterézises triggerelés



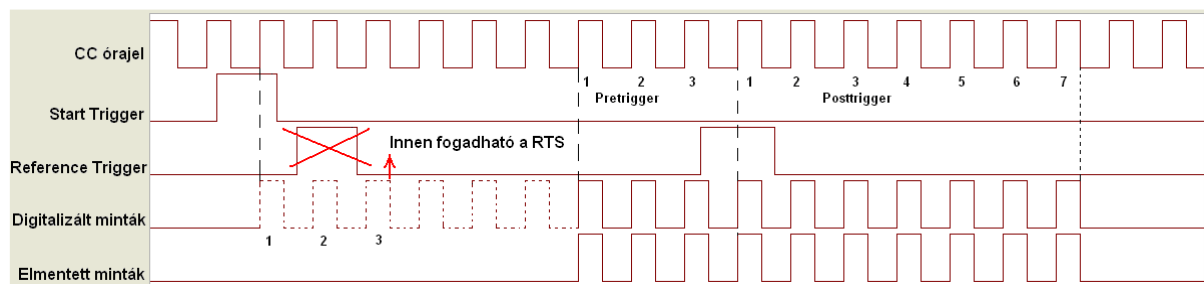
5. Ablak triggerelés



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.



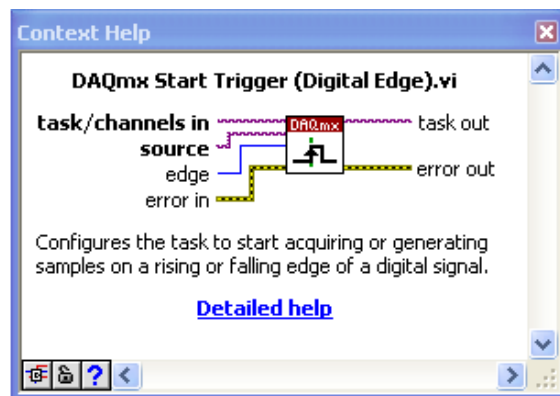
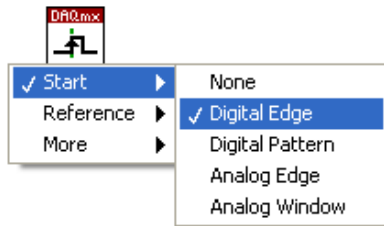
A fenti példában minden jel felfutó élre van vezérelve. Az összes beállított mintaszám 10, a pretriggerelt mintaszám 3, ami azt jelenti, hogy a RTS után 7 mintát fog lemérni és elmenteni a rendszer. A RTS után következő első órajelre elindul a mintavételezés, és mér a rendszer

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.

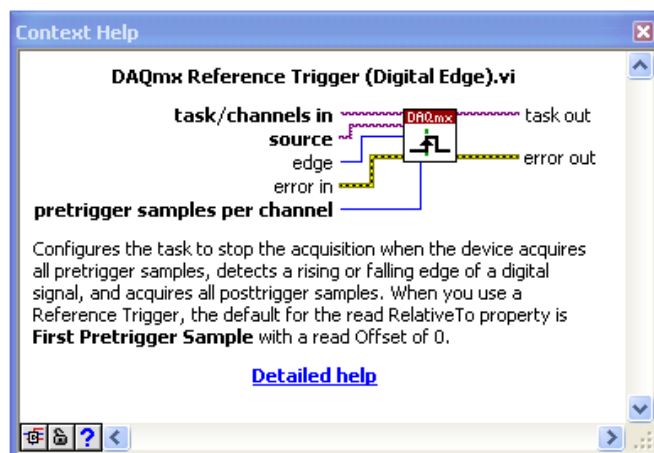
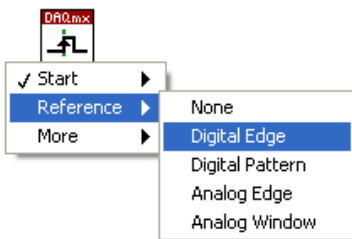
A STS után a pretrigger mintaszámnak a FIFO-ban kell lennie ahhoz, hogy az RTS impulzus fogadható legyen. A fenti példán az első RTS felfutó élének időpillanatában még csak 1 mintát vett a rendszer, ezért ezt figyelmen kívül hagyja.

LabView függvény a trigger jelek beállítására:

Digitális start trigger jel:

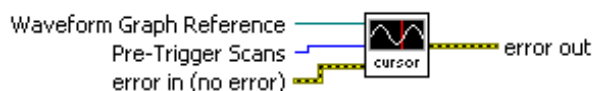


Digitális reference trigger jel:



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

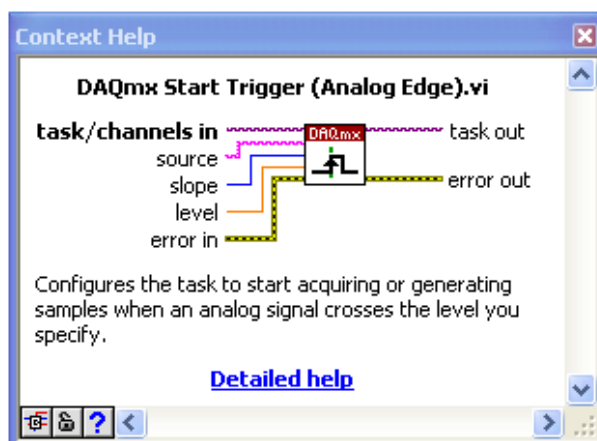
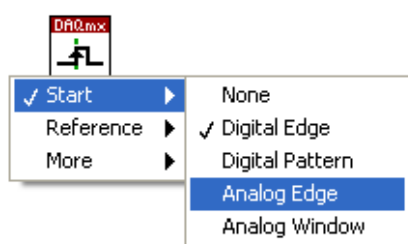
Trigger Cursor.vi



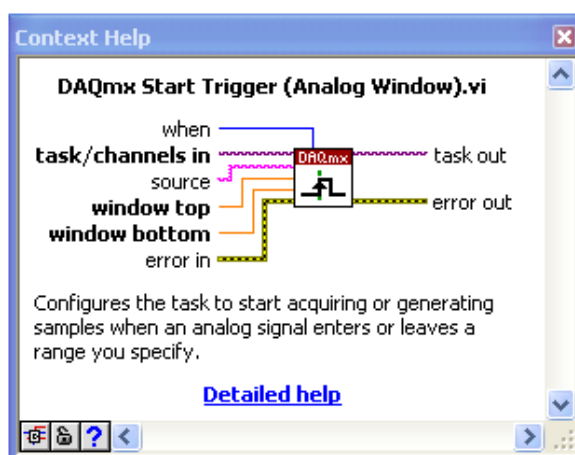
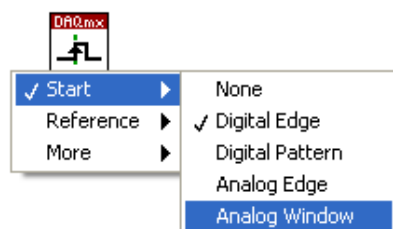
This VI is used to place a vertical cursor on a waveform graph at the location of a reference trigger. "CrsrList" is a property of a Waveform Graph and defines the cursors' style, color, position, etc as they appear on the graph. See the LabVIEW Help Index under "Cursor List" for more information.

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

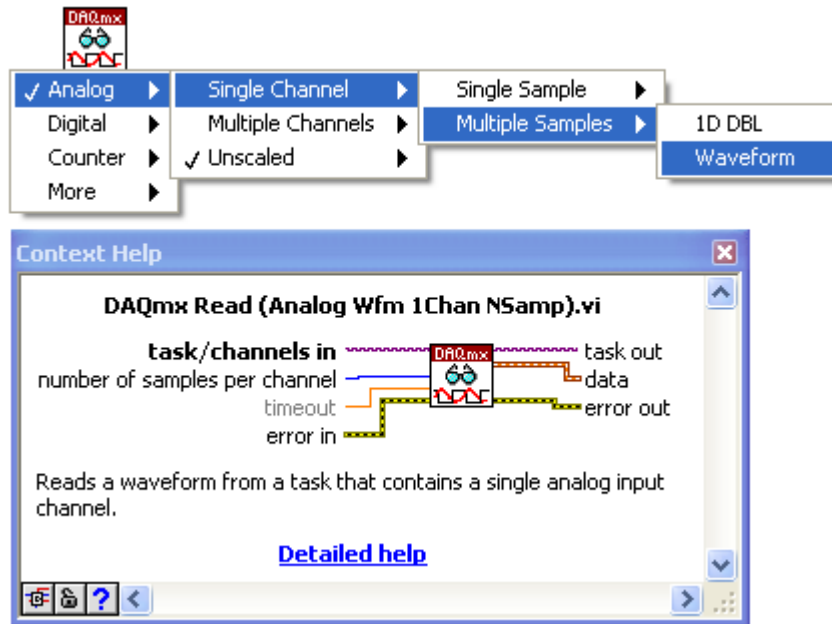
Jelek kiolvasását illetve kiküldését feszültség vagy bináris (kvantum érték) formátumban is elvégezhetjük.

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.

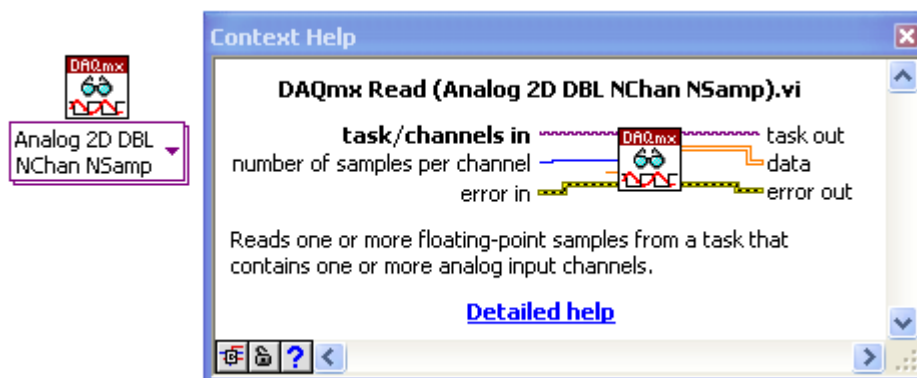
Ha a csatornánkénti mintaszámhoz -1 értéket írunk, akkor a rendelkezésre álló számú minta kerül kiolvasásra.

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

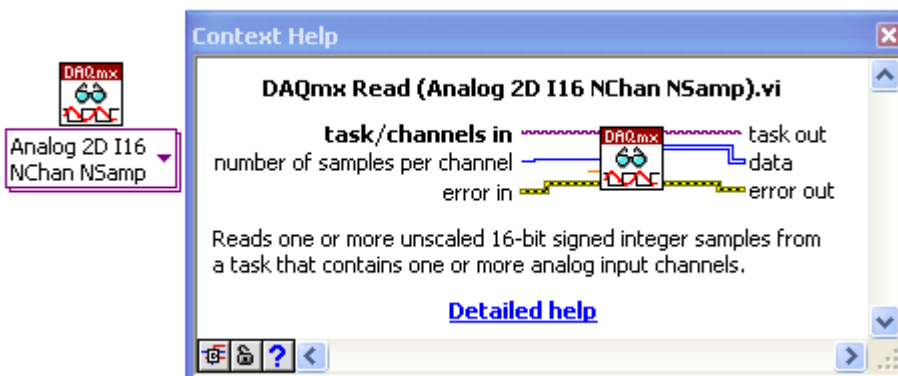
frekvencia) értékéhez néhány másodpercet hozzáadunk. Triggerelés esetén figyelembe kell venni, hogy mennyi időt kívánunk engedni a trigger jel, vagy jelek megjelenésére.



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:



Bináris értékek kiolvasása:

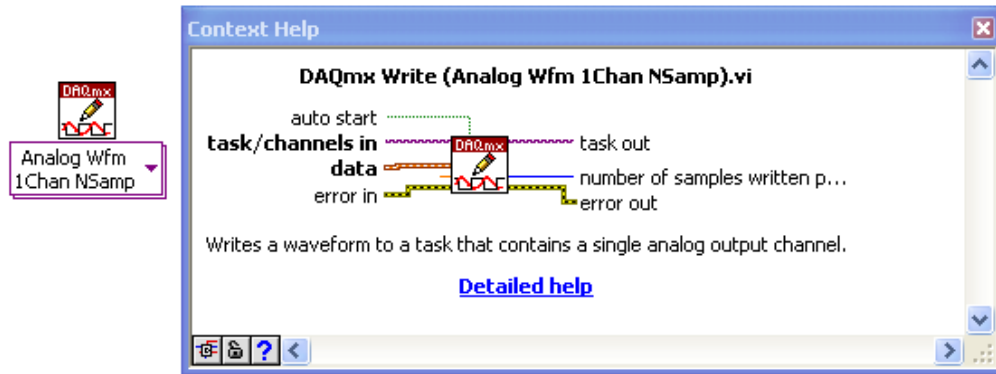


Folyamatos mintavételezés esetén a beolvasás függvényét egy while ciklusba kell helyezni, amelyet a mérés leállítása vezérel.

Analog 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.



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

