KNORR-BREMSE

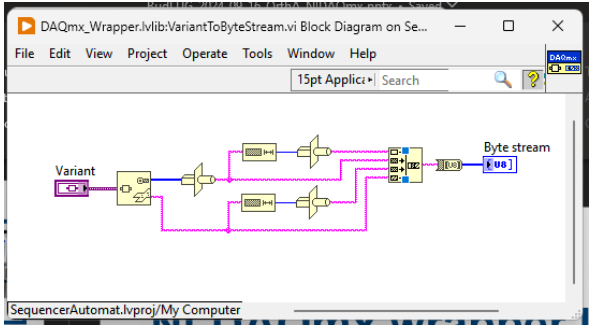
NI-DAQmx driver érdekességei hardware absztrakció során

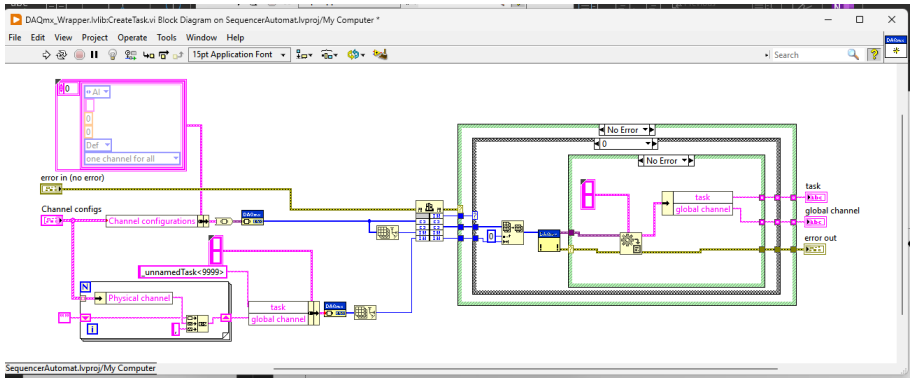
Orth Alán
PR Test Benches – Software Developer Engineer

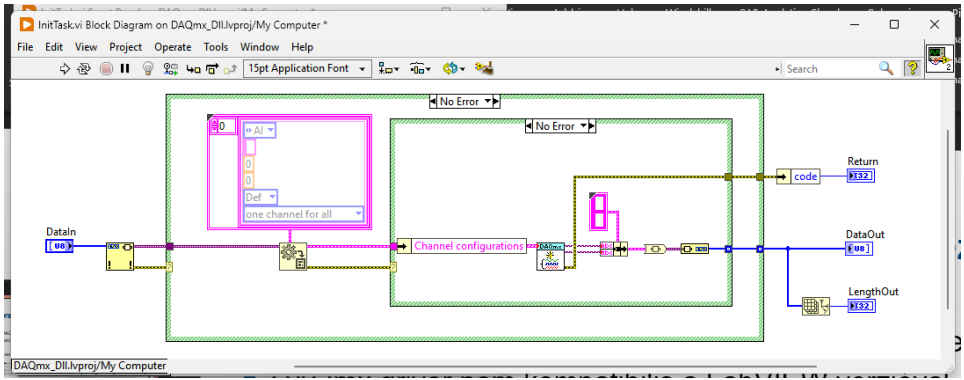
NI-DAQmx wrapper legacy rendszerekhez

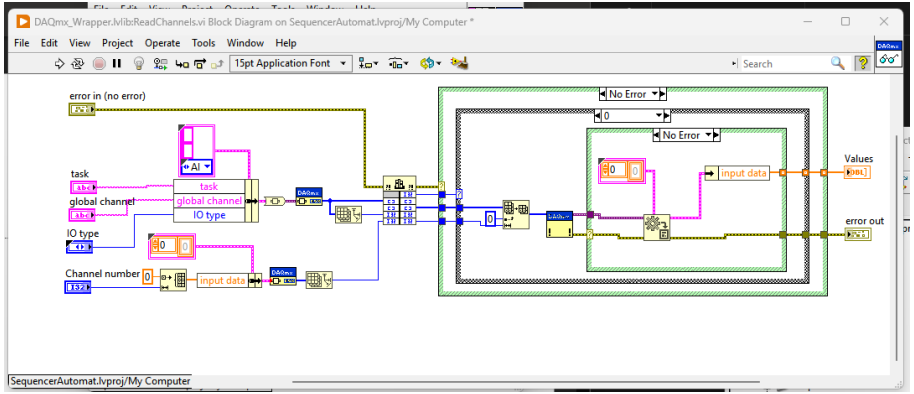
- Régi LV verzióban készült alkalmazás, új NI-DAQmx eszközzel
- DAQmx driver nem kompatibilis a LabVIEW verzióval

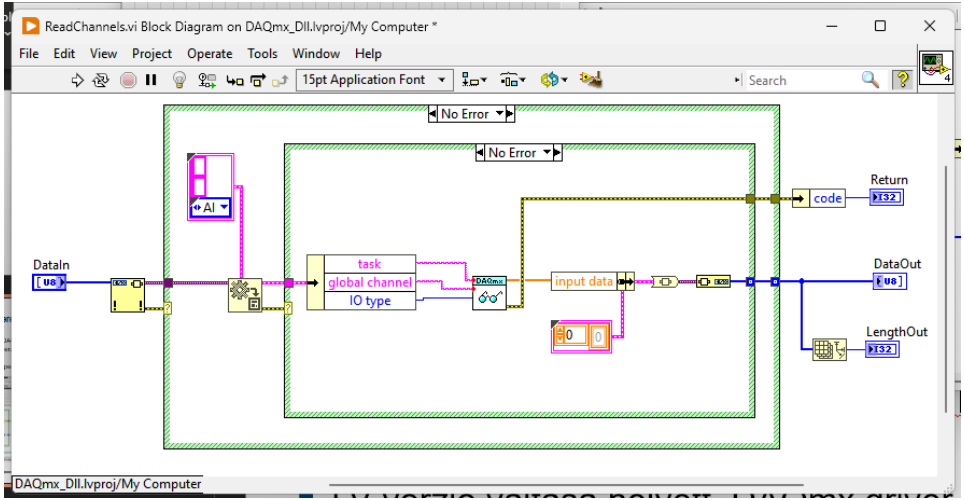
- LV verzió váltása helyett, DAQmx driver wrapper
- Native C interface minden LV specifikus adatot átfordít byte streamre - ez passzolható csak DLL-nek
- Length: Csak statikus memória foglalás van DLL hívásnál, ezért előre le kell foglalni a megfelelően nagy memória területet




DAQmx_Wrapper.lvlib:CreateTask.vi Block Diagram on SequencerAutomat.lvproj/My Computer *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
error in (no error)
Channel configs
Channel configurations
_unnamedTask<9999>
Physical channel
task
global channel
No Error
task
global channel
error out
SequencerAutomat.lvproj/My Computer

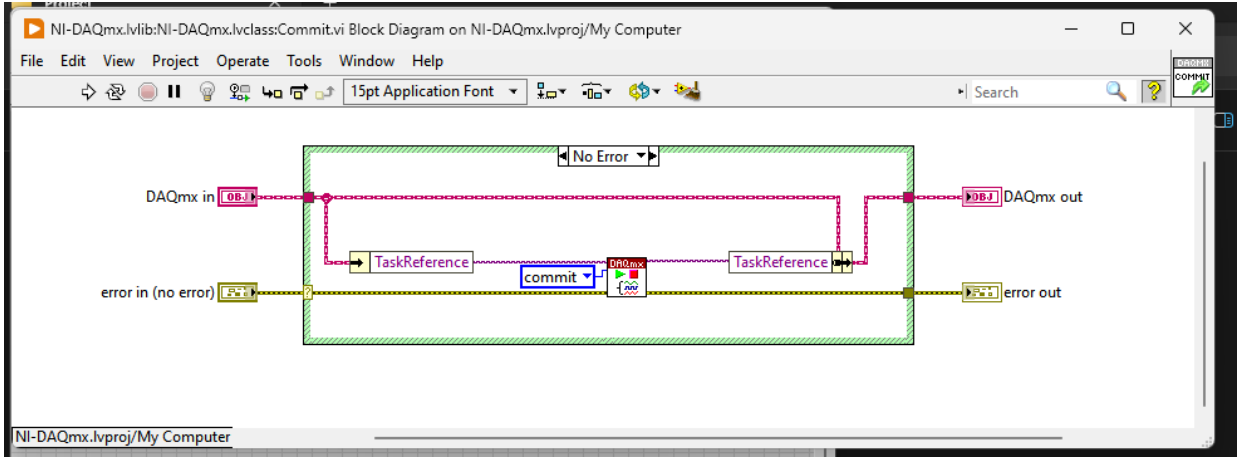

InitTask.vi Block Diagram on DAQmx_Dll.lvproj/My Computer *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
No Error
Def
one channel for all
DataIn
Channel configurations
code
Return
DataOut
LengthOut
DAQmx_Dll.lvproj/My Computer

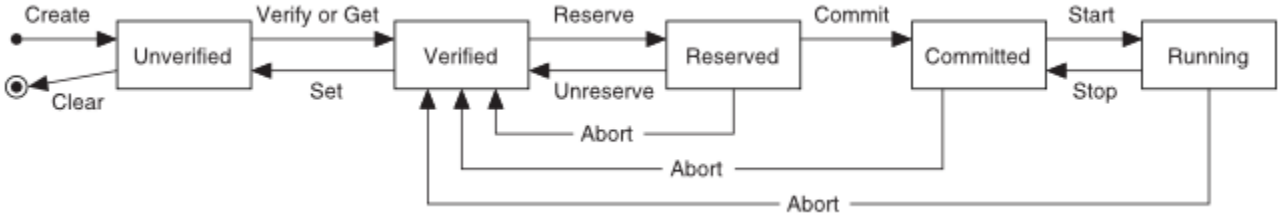

DAQmx_Wrapper.lvlib:ReadChannels.vi Block Diagram on SequencerAutomat.lvproj/My Computer *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
error in (no error)
task
global channel
IO type
Channel number
input data
No Error
Values
error out
SequencerAutomat.lvproj/My Computer


ReadChannels.vi Block Diagram on DAQmx_Dll.lvproj/My Computer *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
No Error
DataIn
task
global channel
IO type
input data
code
Return
DataOut
LengthOut
DAQmx_Dll.lvproj/My Computer

DAQmx Commit

- DAQmx start nem küldi le az eszköznek a cfg-ot, csak lokálisan ellenőrzi a taskot és foglalja a memóriát
- Commit leküldi az eszköznek a meglévő task cfg-ot
- Így nem az első Get-nél kerül az eszköz konfigurálásra
- Gyorsítható a rendszer indítás bizonyos körülmények között
  - Ciklusidő kritikus folyamatoknál a cfg leküldhető előbb így a kritikus folyamat első get-re nem esik ki a ciklusból





- Task start csak a "Reserved" állapotig viszi el a folyamatot
- Ez csak "lokális" - az eszköz nem kerül még foglalásra
- A commit viszont már befoglalja az eszközt is - így csak a trigger-re vár

DAQmx API - ok a fejlesztésre

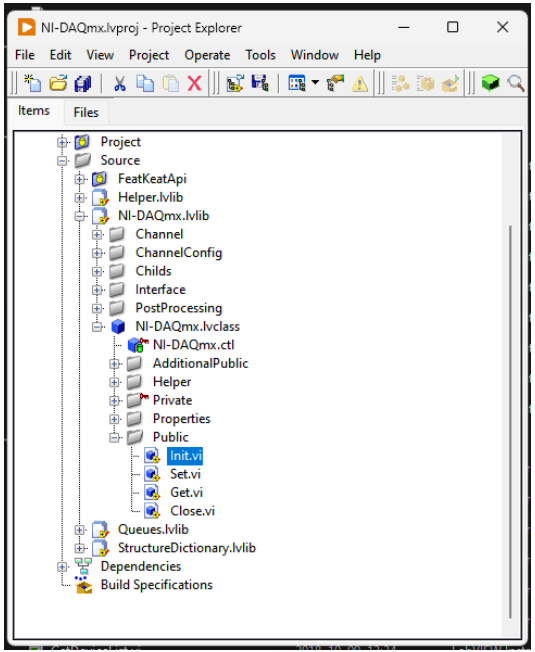
- Illesztés a software architektúrához - KEAT-X
- Kívülről konfigurálható API - nem égetett HW, nem tudom előre, hogy milyen VI-okat kellene "letenni" és konfigurálni
- Általános API a lehető legtöbb lehetőség támogatásával eszköz típustól függetlenül
- Driver alapvetően minden eszközt támogat - de sok API készül eszköz specifikusan. Pl csak cDAQ / PXI stb

Architektúra röviden

- Init
- Set
- Get
- Close

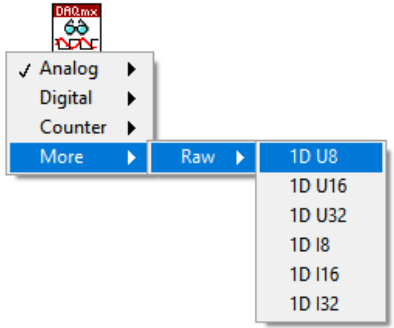
- KEAT-X Framework kompatibilitás - később minden adat a StateVector-ba kerül -> adat típusoknak egyezni kell
- Nincs bufferelés, csak aktuális adatok

- DAQmx driver "nagyon" általános - cél ezt megtartani és absztraktálni
- Ha kívülről akarod konfigurálni és a visszatérő adatokat akarod értelmezni rengeteg elfajulást kell kezelned
- Kihívás: Magas szinten értelmezni, hogy egy vissza kapott adat pontosan mit tartalmaz.

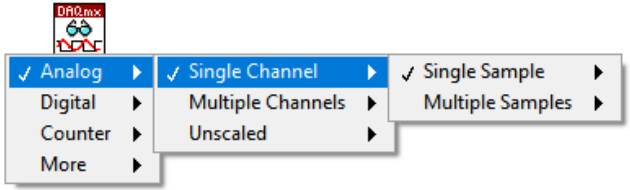


Gyerek osztályok és interface

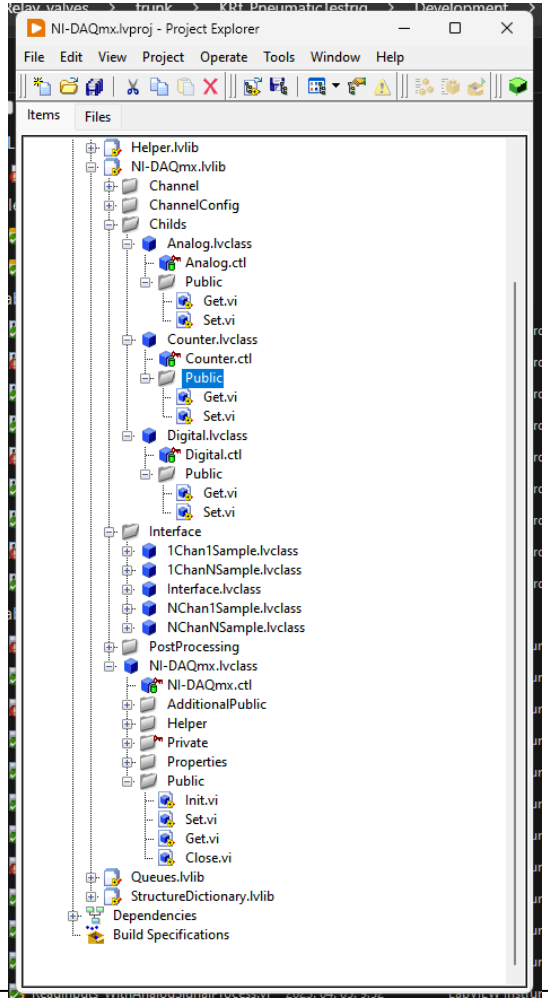
- Analog
- Digital
- Counter



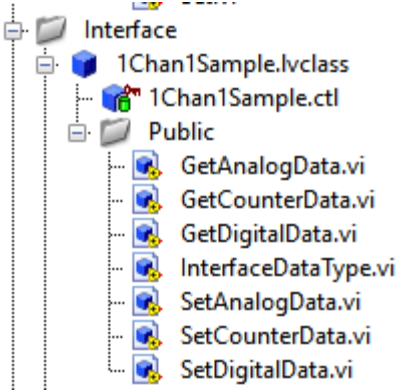
- Interface osztály
  - 1 Channel 1 Sample
  - 1 Channel N Sample
  - N Channel 1 Sample
  - N Channel N Sample

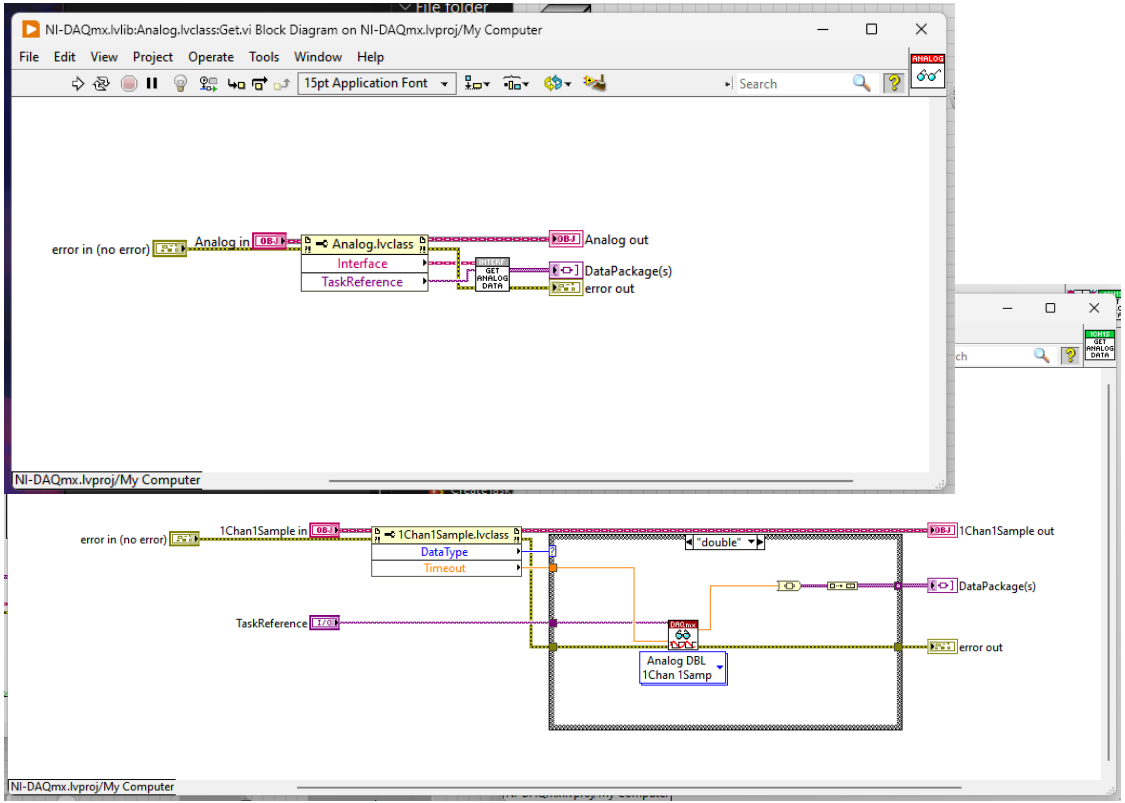


  - Probléma: Get metódus adat értelmezése



Interface osztály gyermeke tartalmazza a Get metódusokat

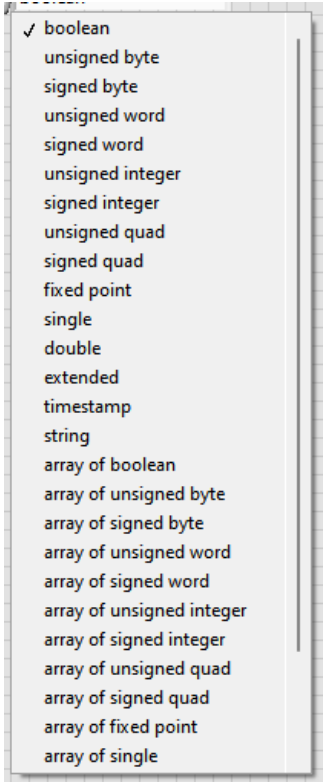




Post processing

- State vector mindig 1 adat pontot fog tárolni - kivéve waveform
- Azaz minden multiple sample típusnál "kezelni kell" az adatokat

Adat típusok - problémás konverziók

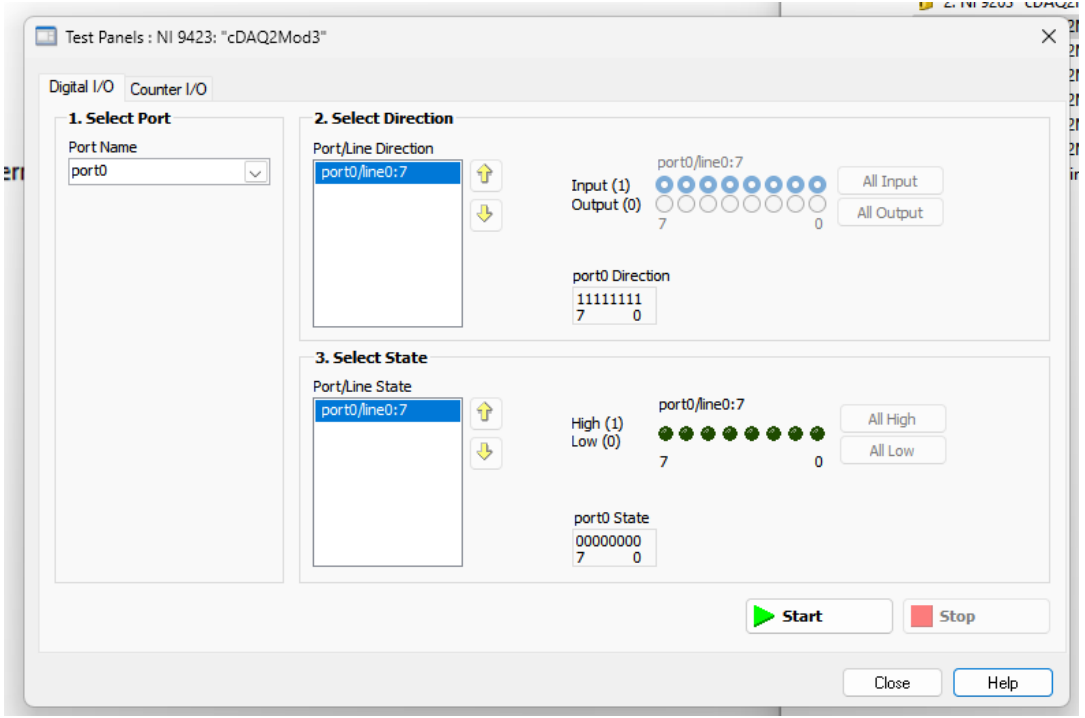


- DAQmx rengeteg adat típust támogat, amik mást és mást jelentenek
- 2D Bool array (Digital 2D Bool NChan 1Samp pl)
- 2D U8 array (Digital 2D U8 NChan NSamp pl)
- 1D Bool array (Digital Bool 1Line 1Point / Digital 1D Bool NChan 1Samp)

- Ugyan az az adat típus több különböző tartalmat tud jelenteni aminek az oldása nem minden esetben triviális

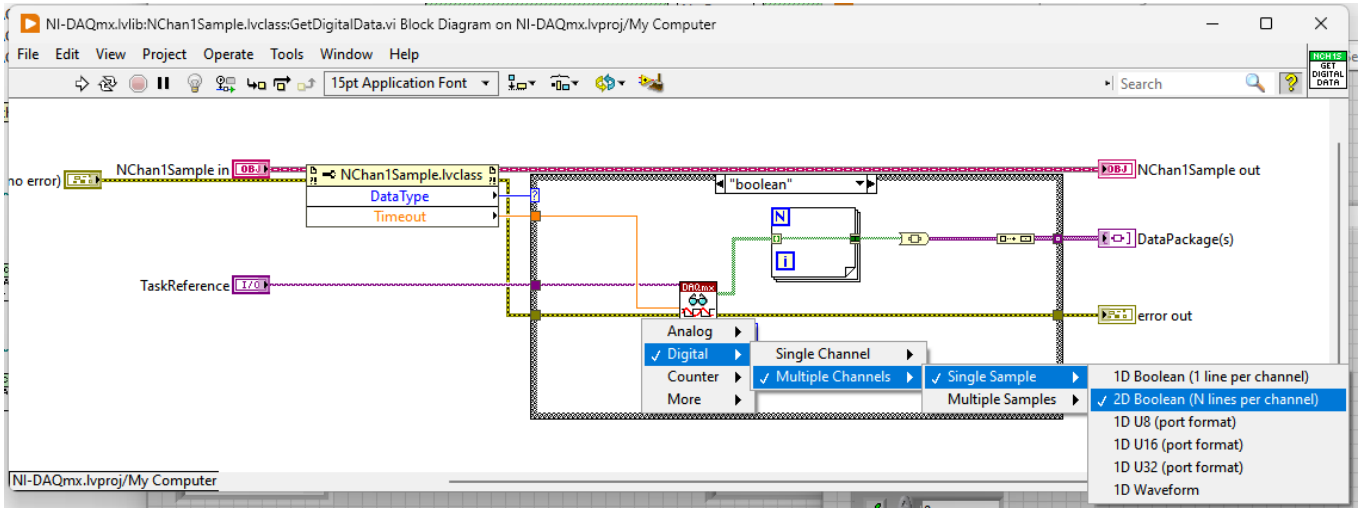
Digital I/O - lines

- Adat struktúra függ a task cfg-tól
- A line-ok lekérdezhetőek külön-külön channel-ben
- N Channel 1 Line lehet 1-1 line channel-enként
- N Channel-nél lehet Mixelt
  - Port0/line4:7 ez 4 bit adat

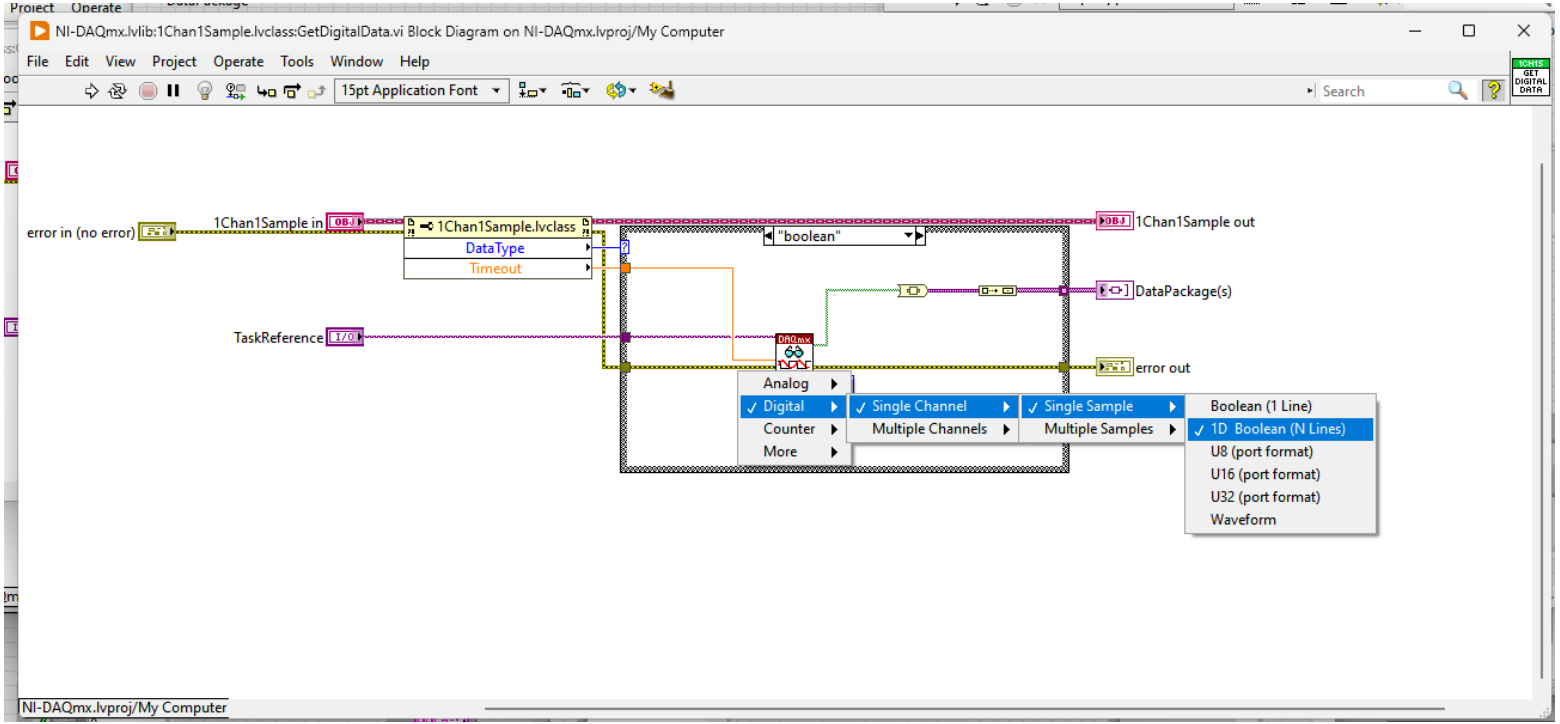


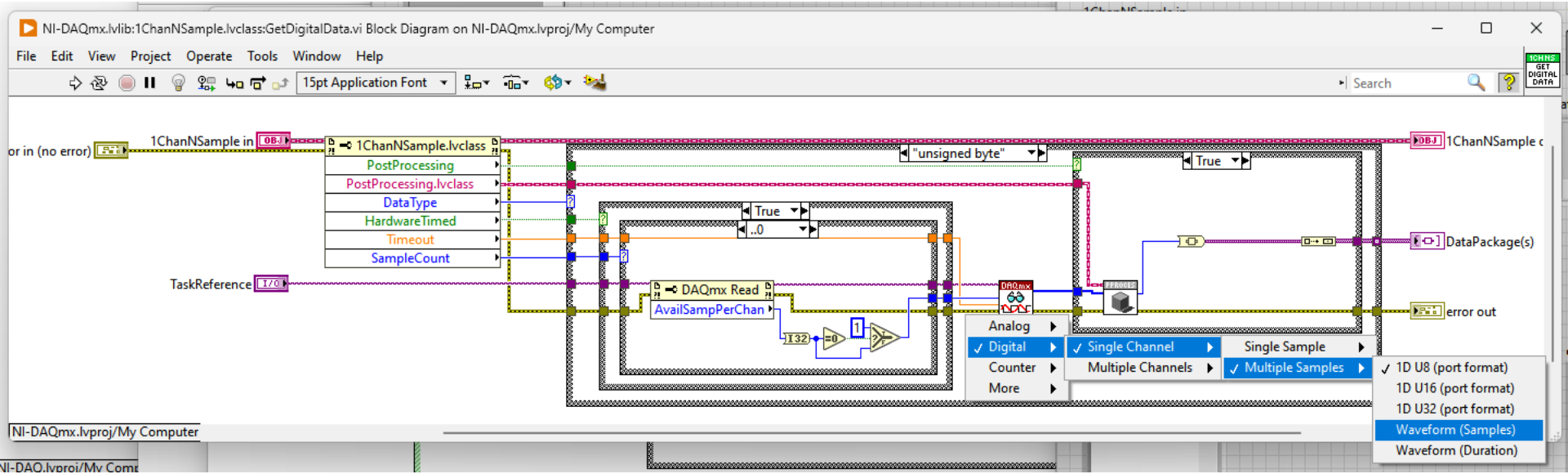
Digital get - N channel 1 Sample

- 2D tömb mindig a legnagyobb line mérethez van méretezve:
  - Pl 2 eszköz - 4 és 8 line -> 2x8 adatot foglal a DAQmx

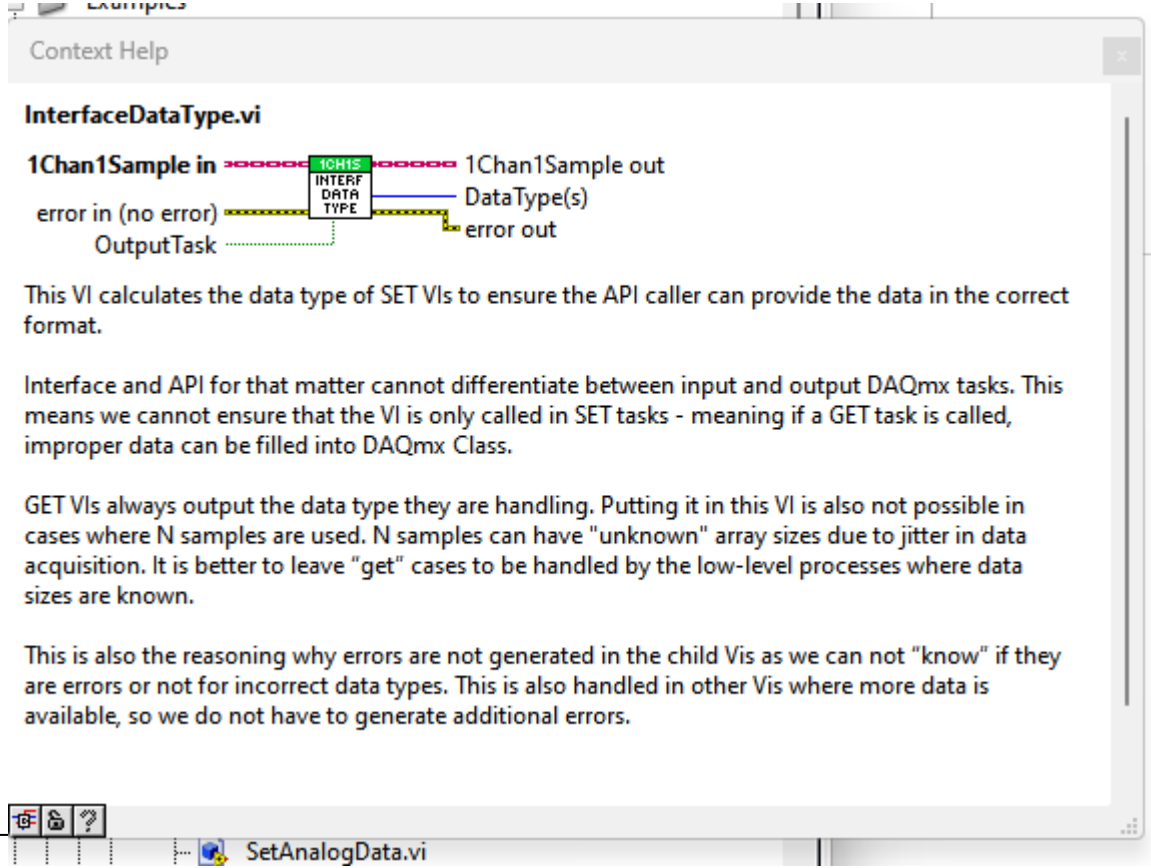


Digital get - 1 channel 1 Sample



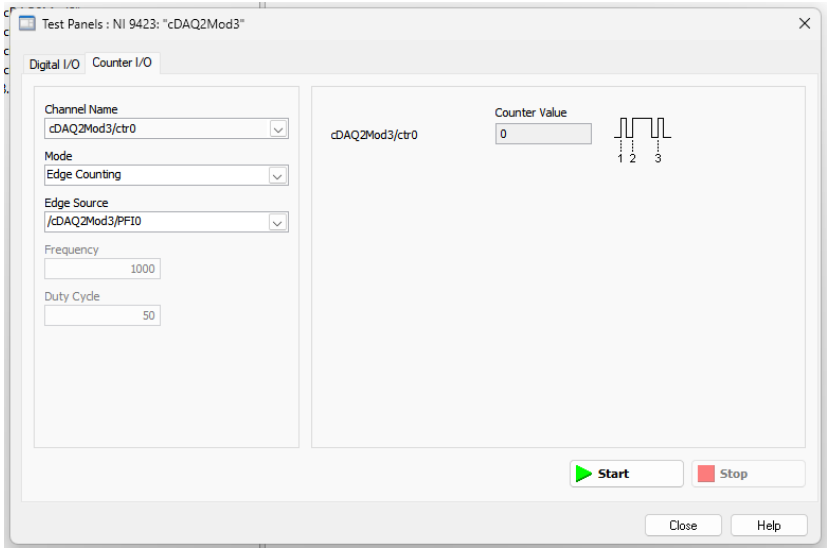

NI-DAQmx.lvlib:1ChanNSample.lvclass:GetDigitalData.vi Block Diagram on NI-DAQmx.lvproj/My Computer
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
1ChanNSample in
1ChanNSample.lvclass
PostProcessing
PostProcessing.lvclass
DataType
HardwareTimed
Timeout
SampleCount
TaskReference
DAQmx Read
AvailSampPerChan
"unsigned byte"
True
..0
DataPackage(s)
error out
Analog
Digital
Counter
More
Single Channel
Multiple Channels
Single Sample
Multiple Samples
1D U8 (port format)
1D U16 (port format)
1D U32 (port format)
Waveform (Samples)
Waveform (Duration)
NI-DAQmx.lvproj/My Computer

Probléma oldása



Tangens - DI kártya mint counter

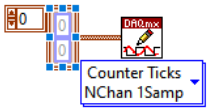
- Channel ref megnevezés nem egyezik a többi standard elnevezéssel
- Kell egy extra " / "
- Ez sehol nem szerepel - de MAX-ban látható az extra " / "

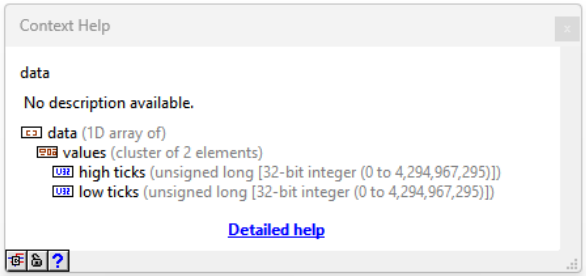



```
    <Channels>
        <Channel channelID="0" channelName="cDAQ1Mod1/ai0"/>
        <Channel channelID="1" channelName="cDAQ1Mod1/ai1"/>
        <Channel channelID="2" channelName="cDAQ1Mod1/ai2"/>
        <Channel channelID="3" channelName="cDAQ1Mod1/ai3"/>
        <Channel channelID="4" channelName="cDAQ1Mod1/ai4"/>
        <Channel channelID="5" channelName="cDAQ1Mod1/ai5"/>
        <Channel channelID="6" channelName="cDAQ1Mod1/ai6"/>
        <Channel channelID="7" channelName="cDAQ1Mod1/ai7"/>
        <Channel channelID="8" channelName="cDAQ1Mod1/ai8"/>
        <Channel channelID="9" channelName="cDAQ1Mod1/ai9"/>
        <Channel channelID="10" channelName="cDAQ1Mod1/ai10"/>
        <Channel channelID="11" channelName="cDAQ1Mod1/ai11"/>
        <Channel channelID="12" channelName="cDAQ1Mod1/ai12"/>
        <Channel channelID="13" channelName="cDAQ1Mod1/ai13"/>
        <Channel channelID="14" channelName="cDAQ1Mod1/ai14"/>
        <Channel channelID="15" channelName="cDAQ1Mod1/ai15"/>
    </Channels>
</Device>
<Device deviceID="1" deviceName="cDAQ1Mod2">
    <Channels>
        <Channel channelID="16" channelName="cDAQ1Mod2/ai0"/>
        <Channel channelID="17" channelName="cDAQ1Mod2/ai1"/>
        <Channel channelID="18" channelName="cDAQ1Mod2/ai2"/>
        <Channel channelID="19" channelName="cDAQ1Mod2/ai3"/>
        <Channel channelID="20" channelName="cDAQ1Mod2/ai4"/>
        <Channel channelID="21" channelName="cDAQ1Mod2/ai5"/>
        <Channel channelID="22" channelName="cDAQ1Mod2/ai6"/>
        <Channel channelID="23" channelName="cDAQ1Mod2/ai7"/>
    </Channels>
</Device>
<Device deviceID="2" deviceName="cDAQ1Mod3">
    <Channels>
        <Channel channelID="24" channelName="cDAQ1Mod3/ao0"/>
        <Channel channelID="25" channelName="cDAQ1Mod3/ao1"/>
        <Channel channelID="26" channelName="cDAQ1Mod3/ao2"/>
        <Channel channelID="27" channelName="cDAQ1Mod3/ao3"/>
    </Channels>
</Device>
<Device deviceID="3" deviceName="cDAQ1Mod4">
    <Channels>
        <Channel channelID="28" channelName="cDAQ1Mod4/ai0"/>
        <Channel channelID="29" channelName="cDAQ1Mod4/ai1"/>
        <Channel channelID="30" channelName="cDAQ1Mod4/ai2"/>
        <Channel channelID="31" channelName="cDAQ1Mod4/ai3"/>
        <Channel channelID="32" channelName="cDAQ1Mod4/ai4"/>
        <Channel channelID="33" channelName="cDAQ1Mod4/ai5"/>
        <Channel channelID="34" channelName="cDAQ1Mod4/ai6"/>
        <Channel channelID="35" channelName="cDAQ1Mod4/ai7"/>
    </Channels>
</Device>
<Device deviceID="4" deviceName="cDAQ1Mod5">
    <Channels>
        <Channel channelID="36" channelName="/cDAQ1Mod5/PFI0"/>
        <Channel channelID="37" channelName="/cDAQ1Mod5/PFI1"/>
        <Channel channelID="38" channelName="/cDAQ1Mod5/PFI2"/>
        <Channel channelID="39" channelName="/cDAQ1Mod5/PFI3"/>
        <Channel channelID="40" channelName="/cDAQ1Mod5/PFI4"/>
        <Channel channelID="41" channelName="/cDAQ1Mod5/PFI5"/>
        <Channel channelID="42" channelName="/cDAQ1Mod5/PFI6"/>
        <Channel channelID="43" channelName="/cDAQ1Mod5/PFI7"/>
    </Channels>
```

Counter outputok nem implementálhatóak

- 2 signal kell a Set-hez (Példa lent)
- Interface az absztrakcióban variant array
  1 elem 1 csatorna
- API szint felett tudnom kellene, hogy 2 adatot össze kell tenni a 1 variantba. Ez azt jelenti, hogy az API "előtt" kellene tudnom olyan információkat, amikre nincs lehetőségem





Waveform implementáció kihagyása

- Rendszer fix adat hosszakat vár el
- HW időzített adatgyűjtésnél ezt nem lehet garantálni - Jitter
- Vicces adat típusok: 1D array of Waveform, amiben 1 sample boolean értékek vannak :)

Kérdések?

Tangens - XOr művelt értelmezése DI kártyán - N Sample esetében

- Gyakorlatban 2 implementáció létezik - mi ezt tartottuk "értelmesnek"
- Ez az implementáció azt adja vissza, hogy melyik line-on "történt" érték változás

XOR gate truth table

Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0