

Beágyazott rendszerek fejlesztése laboratórium
VIMIMA12

Virtuális műszerek méréssorozat

LV1: LabVIEW cRIO ALAPISMERETEK

Mérési útmutató
V0.1



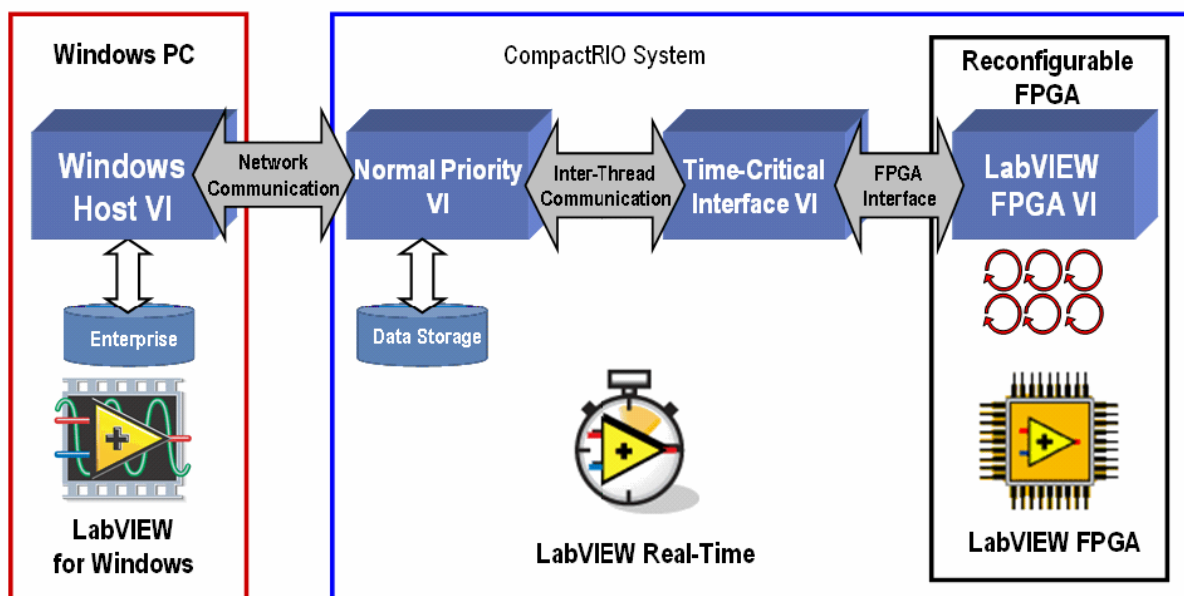
BME MIT 2018.

1. ISMERKEDÉS A COMPACT-RIO RENDSZERREL	3
1.1 A COMPACTRIO FEJLESZTŐKÖRNYEZET	3
1.2 FEJLESZTÉS A COMPACTRIO EGYSÉGRE	5
1.2.1 <i>Kapcsolatfelvétel.....</i>	<i>5</i>
1.2.2 <i>Az első projekt - Projektkörnyezet létrehozása</i>	<i>6</i>
1.2.3 <i>LED-villogtatás.....</i>	<i>11</i>

1. ISMERKEDÉS A COMPACT-RIO RENDSZERREL

1.1 A CompactRIO fejlesztőkörnyezet

CompactRIO A CompactRIO segítségével a különböző mérési beavatkozási funkciókat a megfelelő modulok alapteretbe való behelyezésével lehet végrehajtani. A rendszer érdekessége nem is a funkciójában, hanem konfigurációjának, programozásának újszerű módjában rejlik. Ugyanis az eszközhöz kötődően három külön helyre lehet programot fejleszteni, de mindegyik programot ugyanabban az integrált LabVIEW alapú környezetben (még ha az eszközkészlet picit változik is). Az első hardver közeli rész ahova a programot írhatjuk a mérőmodulokat összefogó alapteretben elhelyezkedő FPGA. Erre az eszközre a hardver közeli feldolgozásokat lehet a LabVIEW FPGA modul segítségével leprogramozni (például digitális szűrés, időbélyegezés, pulzusszámolás stb.). A következő szint ahova lehetőségünk van saját programot helyezni az a szintén a CompactRIO hardverben elhelyezkedő Real-Time ipari PC. Erre a modulra a LabVIEW Real-Time modul segítségével tudunk konfigurációt készíteni, tipikusan az ide kerülő konfigurációk tartalmazzák a magasabb szintű vezérlést, például egy PID szabályozást, adattárolás stb.. Az utolsó absztrakciós szint, amit lehetőségünk van kezelni az egy távoli host gépen futó felügyeleti alkalmazás. Erre a helyre a normál LabVIEW-ban készült programok formájában helyezhetjük el az alkalmazásainkat, tipikusan ez a grafikus felhasználói interfészt helye.



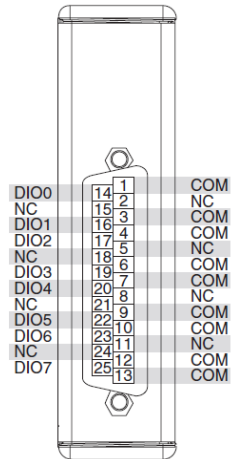
A CompactRIO egység programozási hierarchiája

Bár ez a hármastagság elsőre bonyolultnak tűnhet, a gyakorlatban igen egyszerűen használható egyetlen integrált fejlesztőrendszerből.

cRIO-9074 A mérések során a cRIO-9074-es CompactRIO keretet fogjuk használni. Ez 8 mérő kártya fogadására képes, a hátlapba egy 2M ekvivalens kapu komplexitású Xilinx FPGA foglal helyet. A helyi real-time feldolgozást pedig egy 128MiB DRAM-al és 256MiB háttértárral rendelkező 400MHz-es vezérlő biztosítja. A keretbe beépítve két 10/100BASE-Tx Ethernet csatlakozás található, amelyek közül az 1. sorszámún kapcsolódhatunk a modulhoz fejlesztés közben. Valamint rendelkezik egy RS232 csatlakozási ponttal is.



NI-9401 Az első mérési gyakorlatainkhoz a z NI-9401 8 csatornás TTL szintű, nagysebességű 100ns-es digitális I/O modult fogjuk használni.



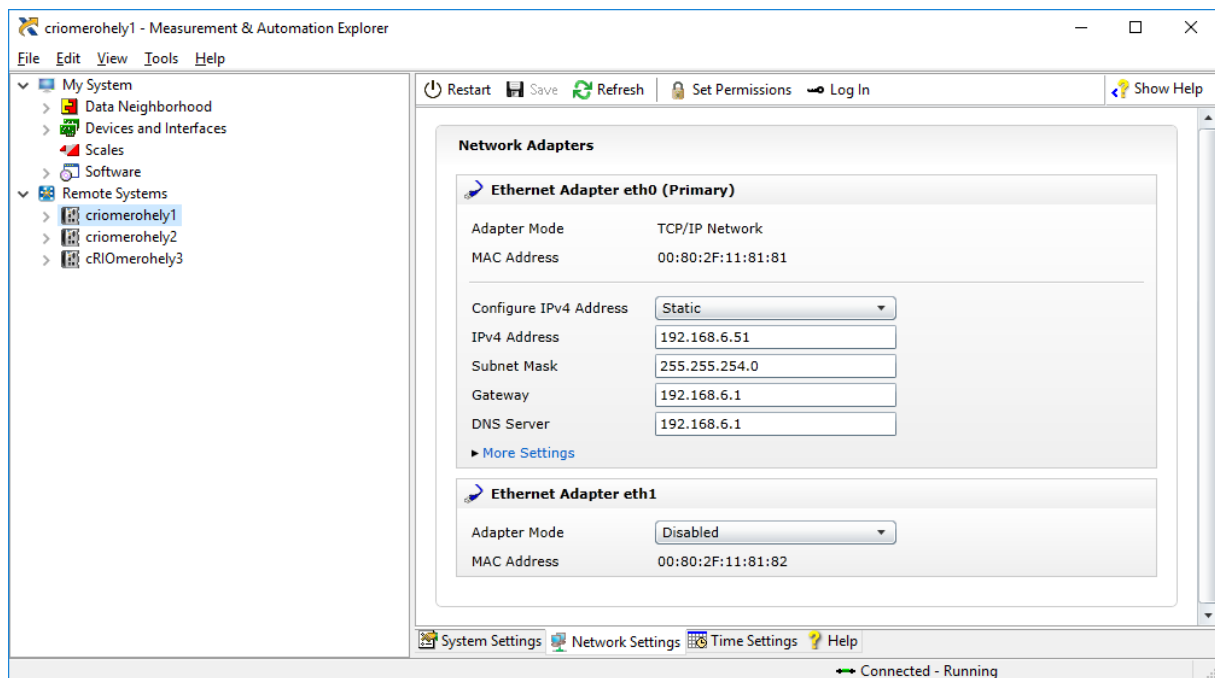
1.2 Fejlesztés a CompactRIO egységre

1.2.1 Kapcsolatfelvétel



Measurement
& Automation

Első lépésként a CompactRIO modul használatához ellenőrizni kell a TCP/IP hálózati beállításokat, amelyek segítségével kapcsolatba lehet lépni az eszközzel. Erre az *NI Measurement & Automation Explorer* programját használjuk. Ez a programcsomag arra szolgál, hogy áttekinthessük, és szükség esetén módosítsuk a PC-khez hozzárendelt mérési adatgyűjtők és beavatkozók beállításait. Itt kerülnek felsorolásra az egyes PCI alapú mérő kártyák, GPIB buszon csatlakozó külső műszerek, valamint a távoli eszközök (*Remote Systems*) kategóriában az egyes CompactRIO eszközök. Ha szerencsénk van, akkor itt már a méréshez tartozó cRIO-9074-es modul szerepel a listában.



Ha nem, akkor a *Remote Systems/Create New* menüpont segítségével adhatjuk hozzá a listához (automatikusan keressük meg, vagy adjuk meg a modul IP címét direktben). Amennyiben nem tudjuk a modul IP címét akkor lehetőségünk van azt kideríteni úgy, hogy a soros porton keresztül hozzátcsoljuk az eszközt a PC-hez, bekapcsoljuk az cRIO-9074 előlapján lévő *Console Out* switch-et és a soros portra 9600 baud sebességgel elküldött üzenetből megállapítjuk a két Ethernet csatlakozáshoz tartozó TCP/IP beállítást.

```
MAX system identification name: NI-cRIO9074-013EF9F4
Initializing network...
Device 1 - MAC address: 00:80:2F:11:81:81 - 192.168.6.51 (primary)
Device 2 - MAC address: 00:80:2F:11:81:82 - 192.168.9.2 (secondary)
```

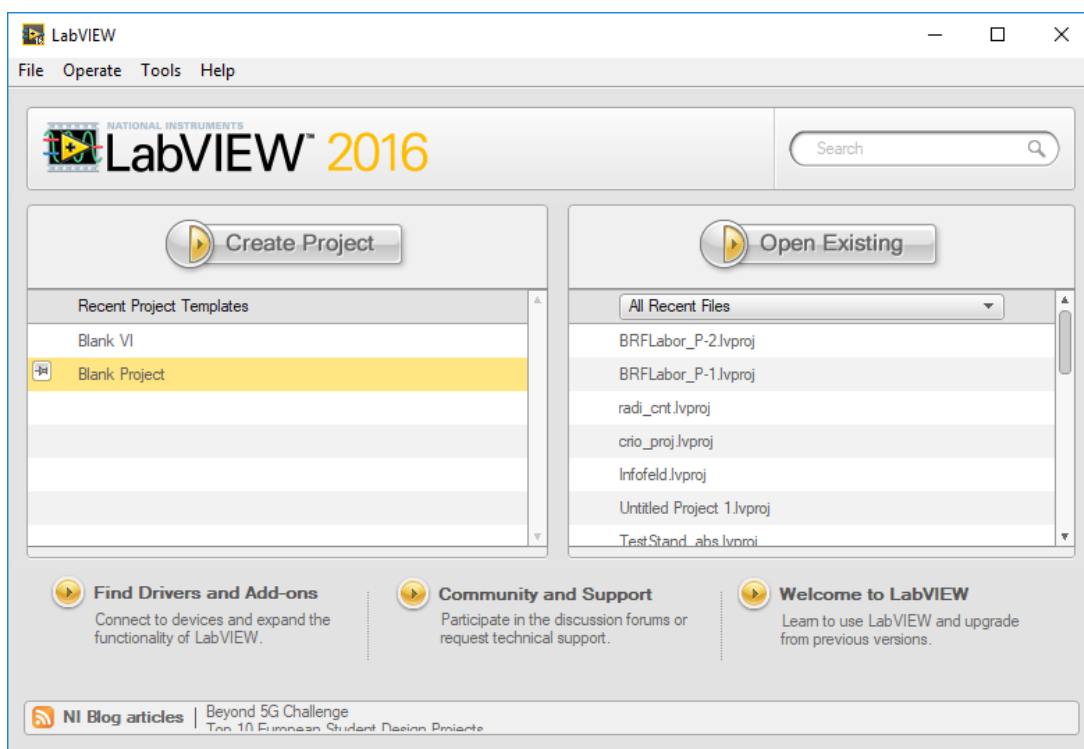
NI Measurement & Automation Explorer segítségével átírhatjuk az eszköz hálózati beállításait. (Azon ne lepődjünk meg, hogy itt nem jelennek meg a sínbe illesztett modulok és azok tulajdonságai, ezeket csak a LabVIEW projektekből lehet majd látni.)

1.2.2 Az első projekt - Projektkörnyezet létrehozása

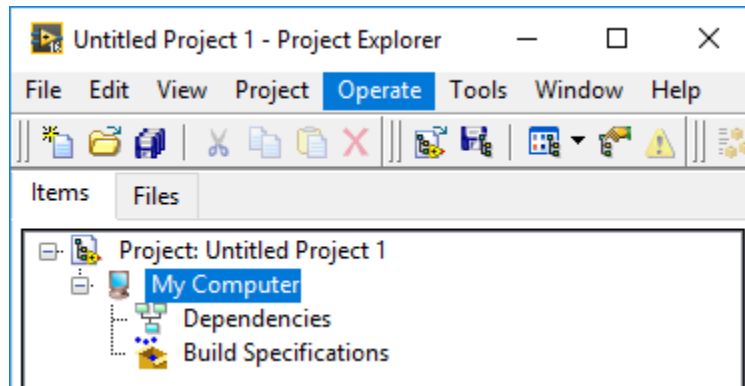


Mint arról a bevezetőben már szóltunk, a CompactRIO modulnál három helyre is lehet programot írni. Az első mintaprojektünkben egy olyan környezetet fogunk varázsló nélkül létrehozni, ahol mindhárom szintre tudunk programot készíteni. Itt érdemes megjegyezni, hogy a beépített varázslók segítségével is lehet projekteket létrehozni, de így nulláról indulva jobban áttekinthető a folyamat.

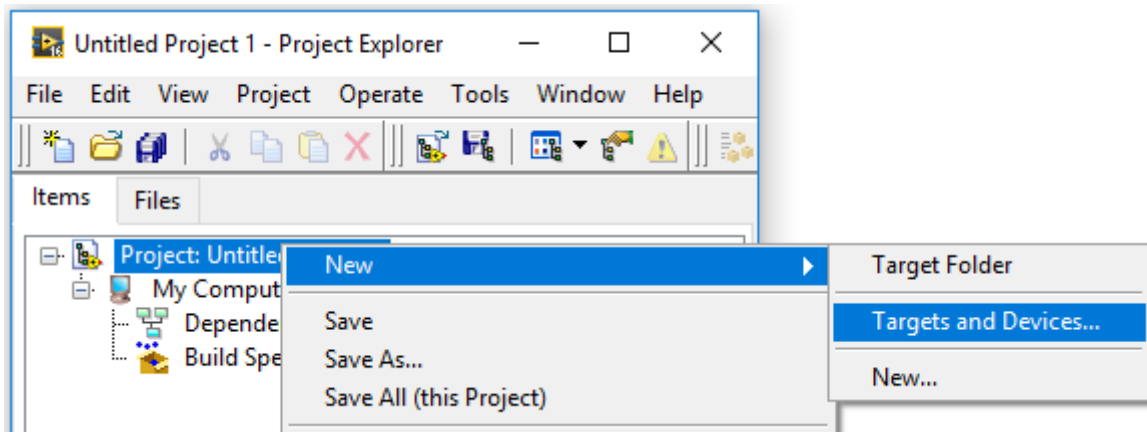
Indítsuk el a *National Instruments LabVIEW*-t, és válasszuk a *Blank Project* opciót!



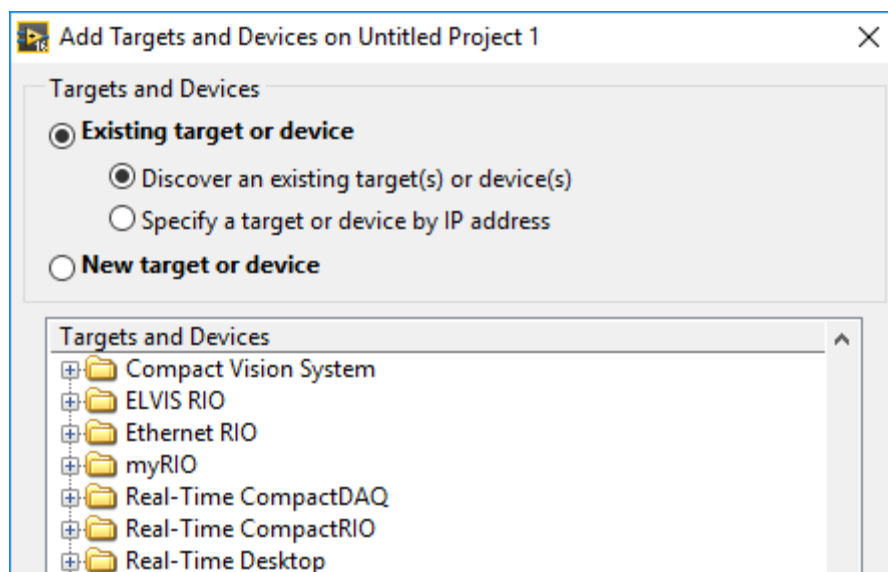
Ennek a hatására megjelenik egy új ablakban a *Project Explorer*, amely felsorolja a használt eszközöket (egyelőre ez még hiányzik).



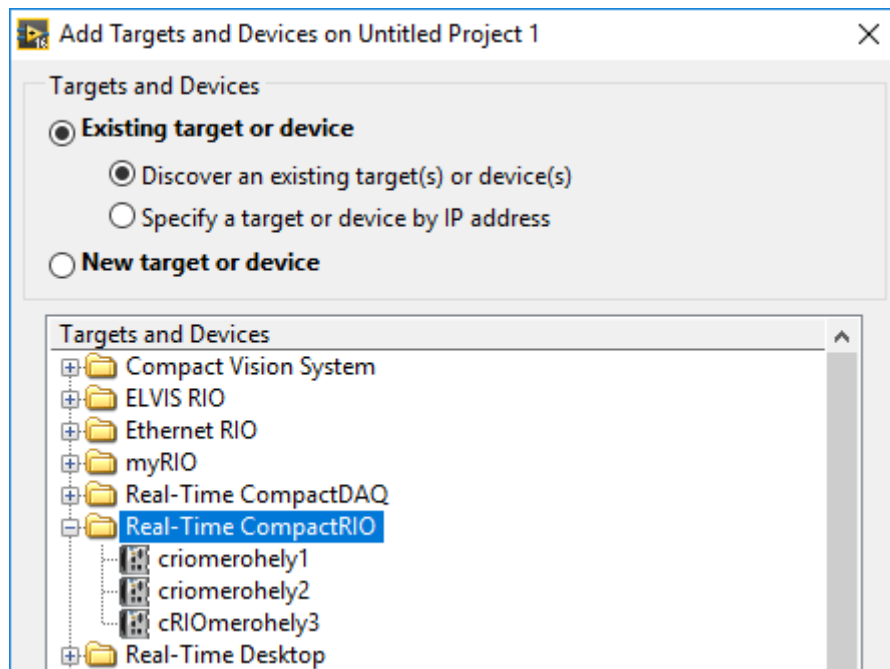
Mivel jelenleg egy üres mérőrendszer nélküli környezetet kaptunk, ezért a projekthez hozzá kell adnunk a CompactRIO-t. Ezt a „*Project: Untitled Project 1*”-en jobb egérgombot nyomva a *New/Target and Devices* menüvel tehetjük meg.



Lehetőségünk van már meglévő eszközt hozzáadni a rendszerhez (ezeket automatikusan detektálja a LabVIEW) vagy egy általunk definiált újat.

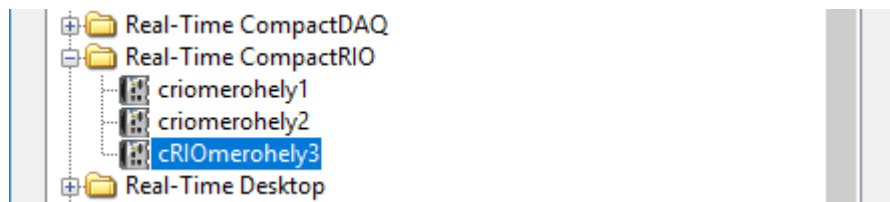


A rendszer felsorolja az automatikusan felismert eszközöket, itt a három cRIO modul.
Válasszuk ki a *Real-Time CopmpactRIO* menüpontot!

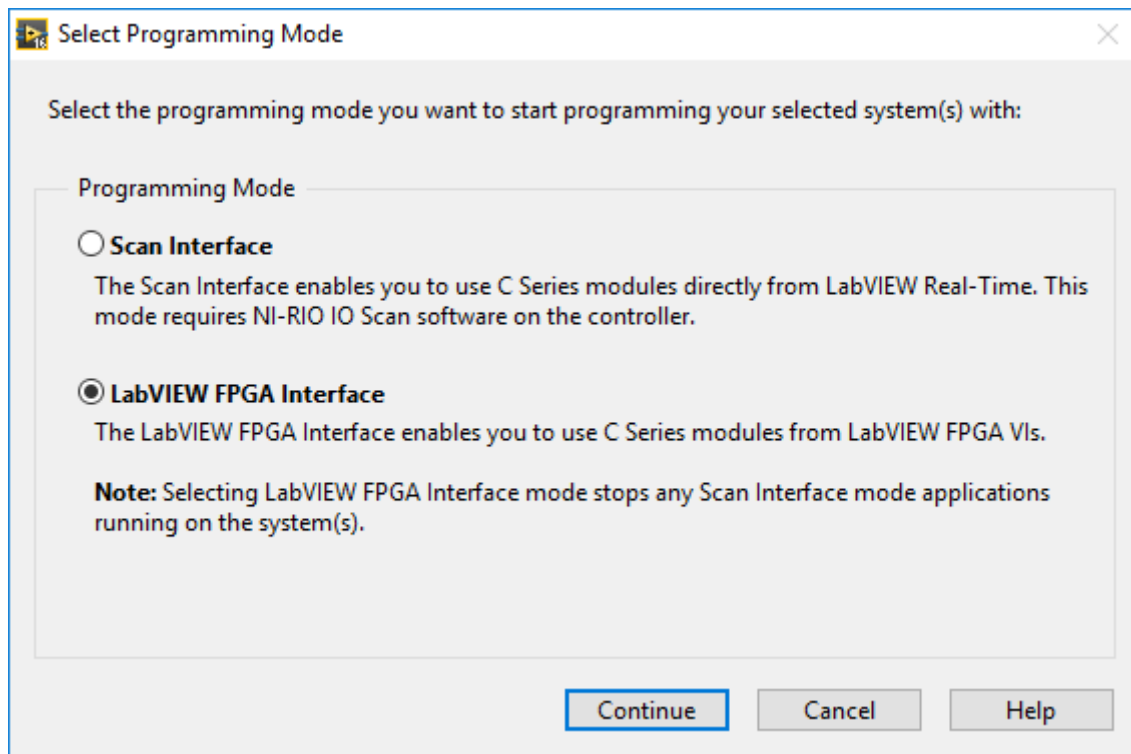


Megjegyzés: Nem mindig detektálja a rendszer a *Measurement&Automation Explorer*-rel megadott eszközöket. Ilyenkor új eszközként kell hozzáadnunk a modult: ki kell választanunk az eszköz típusát (itt cRIO-9074) és megadni az IP címét.

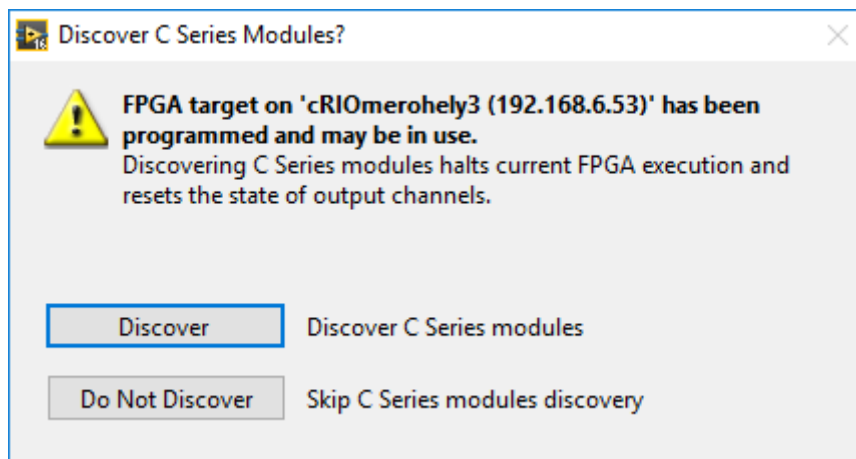
Adjuk hozzá a mérésnél használt cRIO kártyát a rendszerhez (cRIOmehely1, 2 vagy 3)!



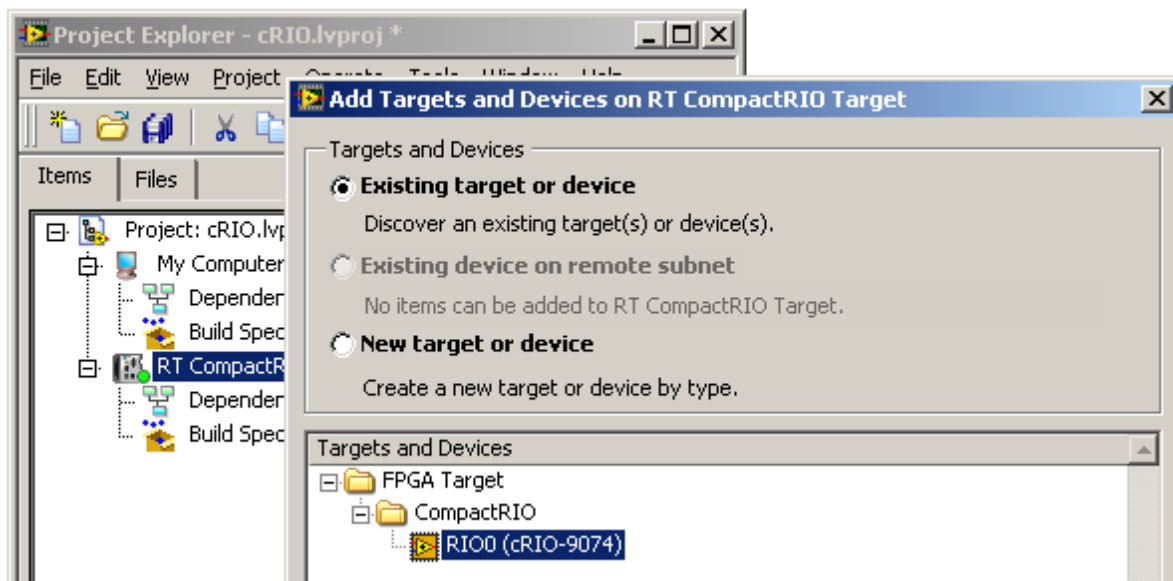
Válasszuk ki a *LabVIEW FPGA Interface* pontot!



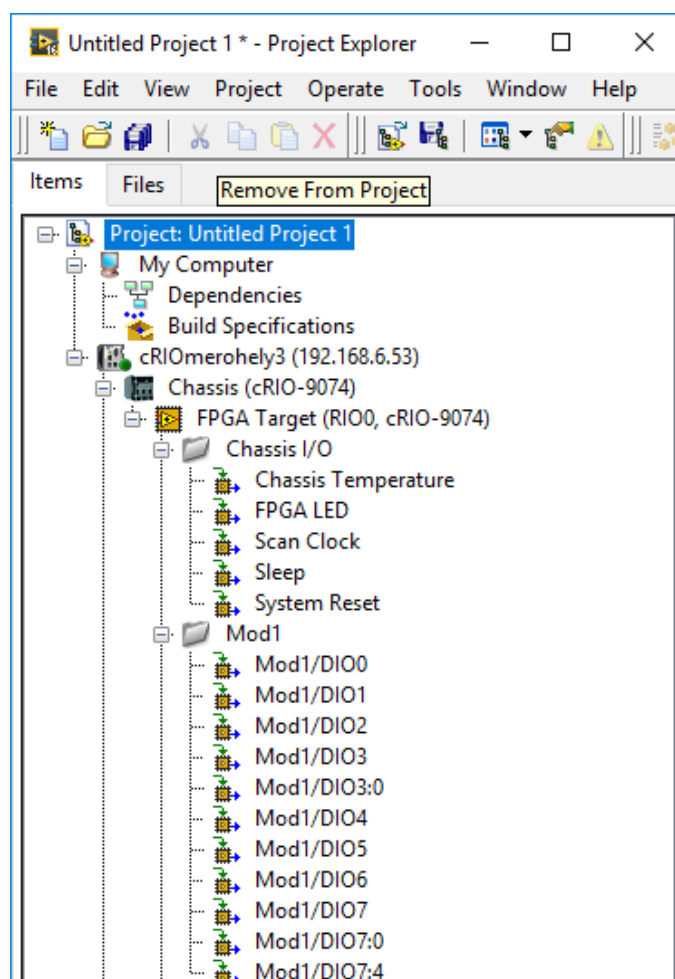
Ha korábban már fel volt programozva a cRIO, és fut rajta program, akkor megjelenhet az alábbi ablak. Kattintsunk a *Discover* gombra, és várjuk meg az eredményt!



Megjegyzés: Amennyiben a rendszer automatikusan nem detektálta a cRIO-9074-es eszközt, akkor szükség lehet arra is, hogy külön hozzáadjuk az FPGA target-et is a projekthez. (Az FPGA huzalozza össze a RealTime PC-vel az egyes mérőkártyákat, ezért ez szükséges ahhoz, hogy lássuk az egyes mérőmoduljainkat.) Ez hasonlóan történik, mint ahogy a projektindításnál bemutattuk azzal a különbséggel, hogy itt a CompactRIO modulhoz adjuk hozzá az új *target*-et. Ha jól adtuk meg a cRIO modul IP címét és végrehajtottuk a *Connect* parancsot, akkor itt már automatikusan fel kell ismernie a rendszernek az FPGA targetet.



Az *FPGA target* felismerése után a rendszer felderíti a modulhoz csatlakoztatott mérőkártyákat. Megjelennek a projektablakban a cRIO-9074-es modulhoz tartozó mérő- és beavatkozó csatornák (*chassis*), valamint az egyes mérőmodulok csatornái. Az ábrán a *Mod1* néven az NI-9401 digitális I/O kártya csatornái láthatóak.

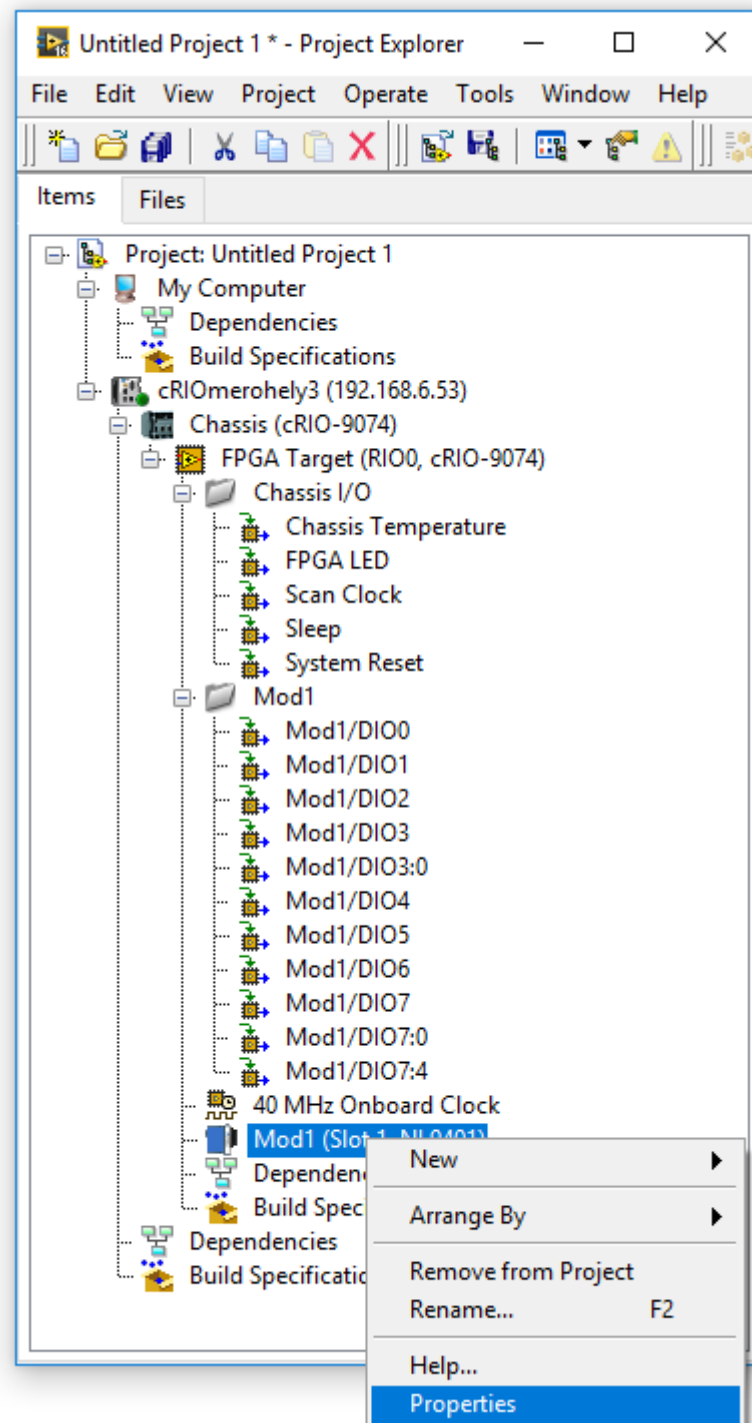


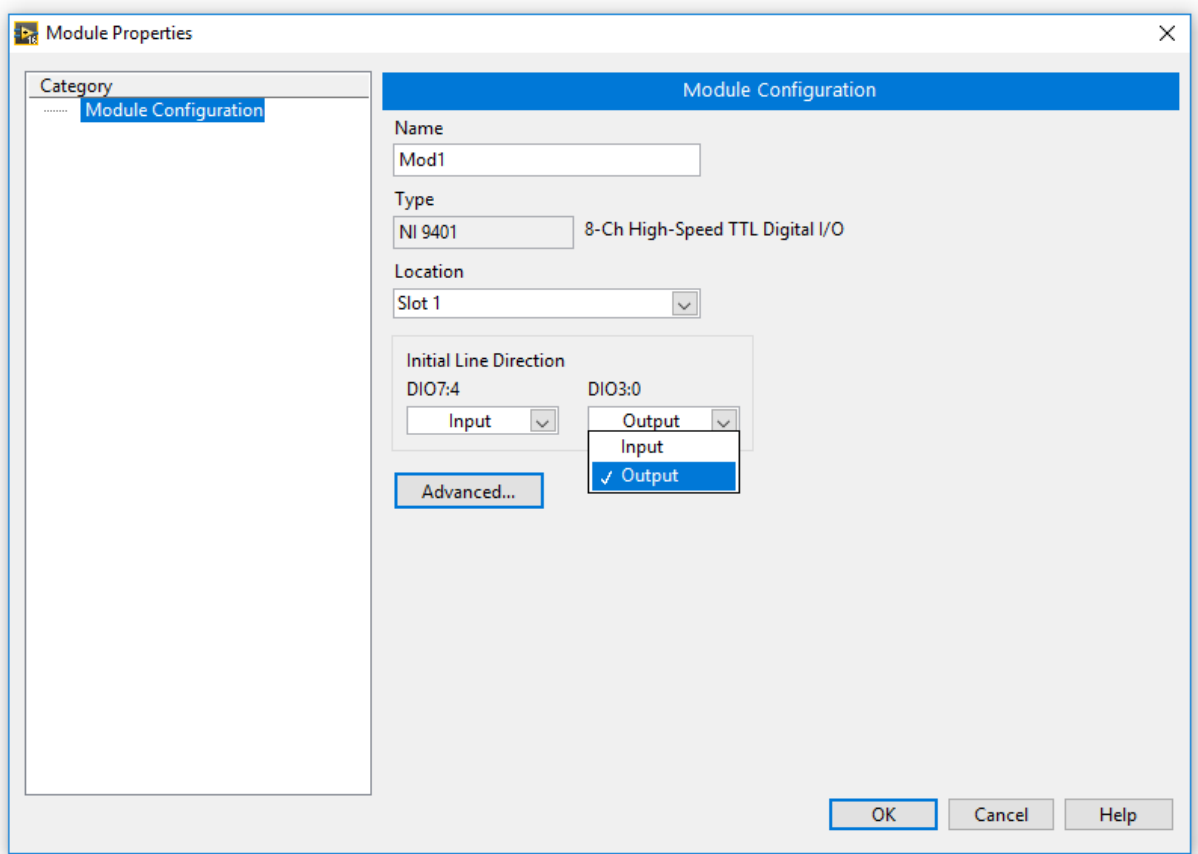
Ezekkel a lépésekkel a projektkörnyezetet előállítottuk.

1.2.3 LED-villogtatás



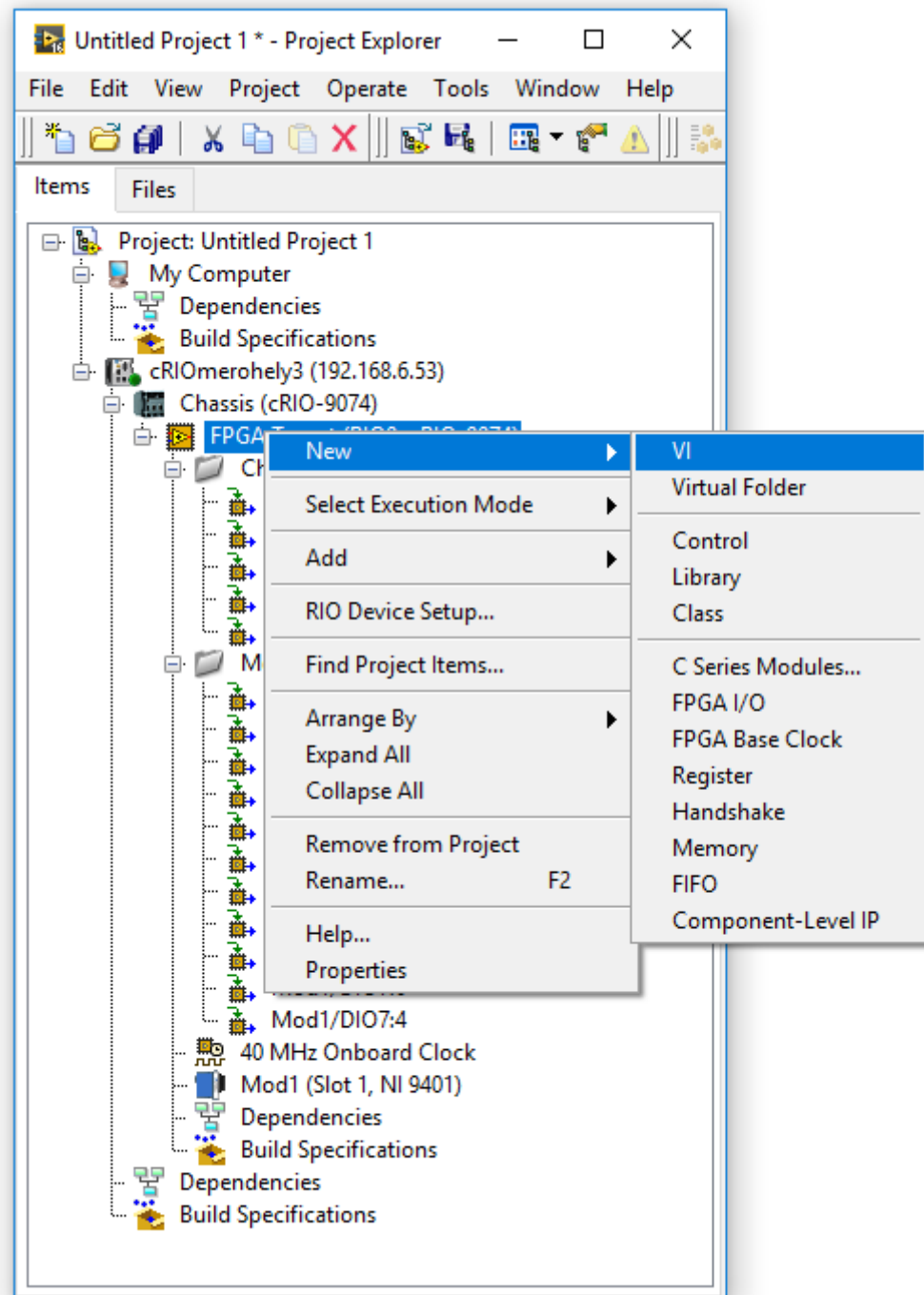
A következőkben megnézzük, hogyan kell működésbe hozni a CompactRIO egyes mérőmoduljait. Első lépésként konfiguráljuk a NI-9401 modul DIO0-DIO3 csatornáját kimenetnek a projektablakban található Mod1 fizikai reprezentációjaként megjelenő ikon *Properties* menüpontjában! (Ne felejtjük el utána elmenteni a projektet!)



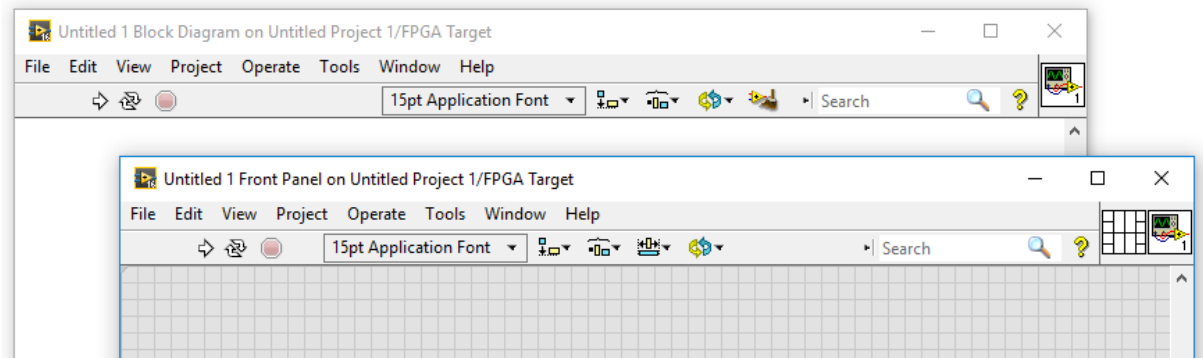


FPGA szint

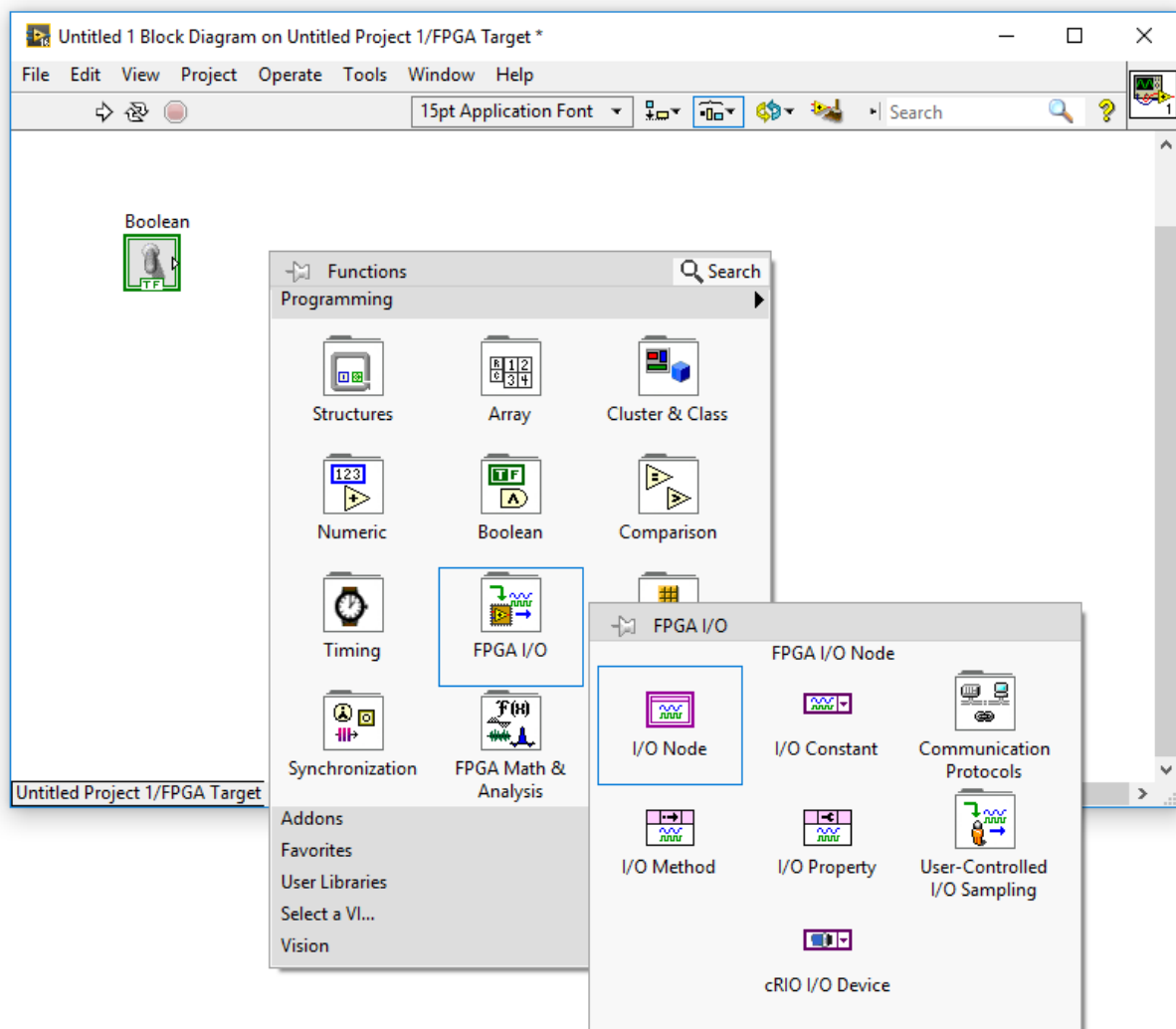
A vezérlés létrehozásához adjunk hozzá a projekthez egy új FPGA VI-t, ez fogja ellátni az egyes I/O lábak vezérlését, hiszen ez a legalsó hardverközeli réteg (az *FPGA Target*-en jobb egérgomb, majd *New/VI*).

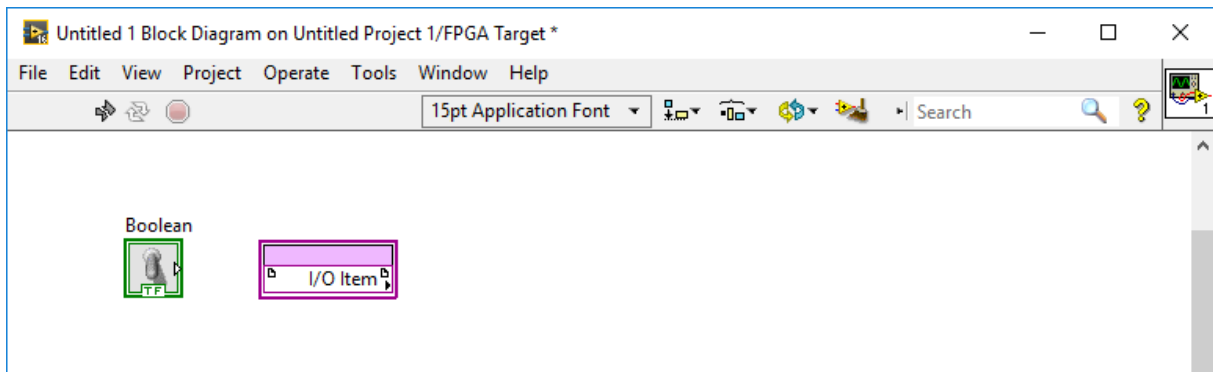


Az új VI rögtön szerkeszthetővé is válik.

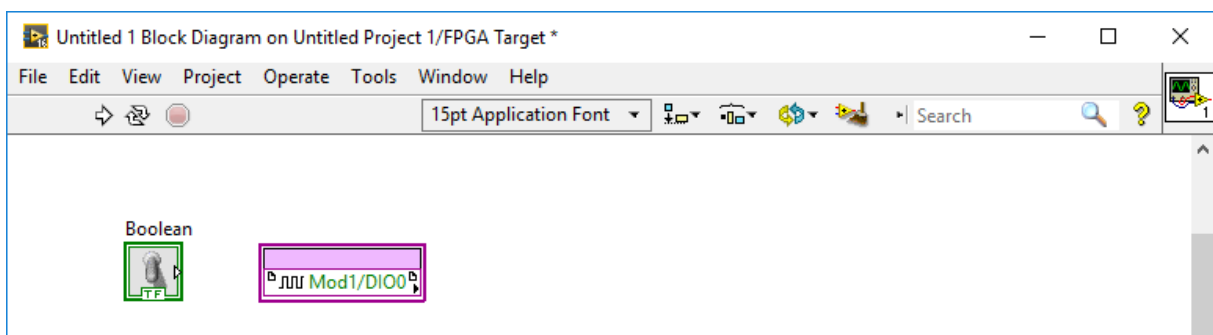
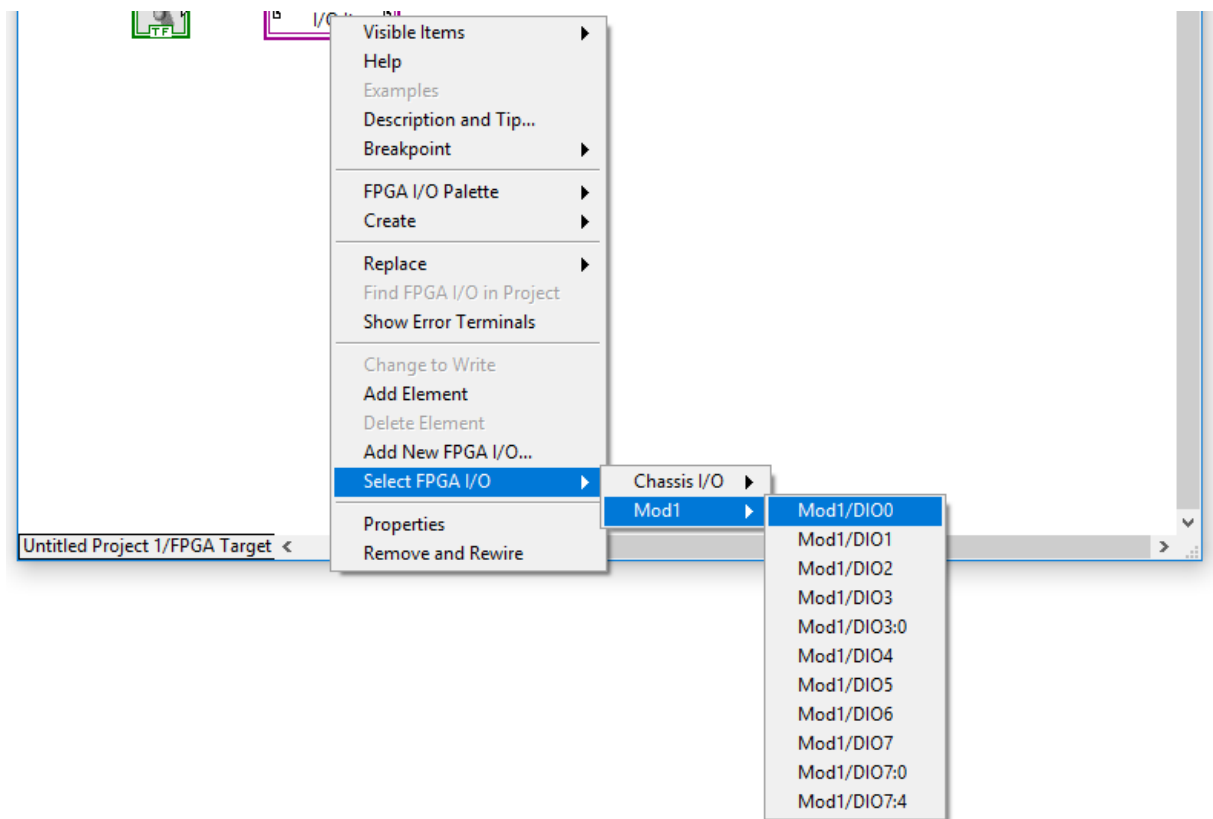


Rakjunk a *Front Panel*-re egy kapcsolót, és a *Block Diagram*-on tegyünk mellé egy *FPGA I/O Node*-ot, amely egy mérőkártya-csatornát fog megisméjelyesíteni.

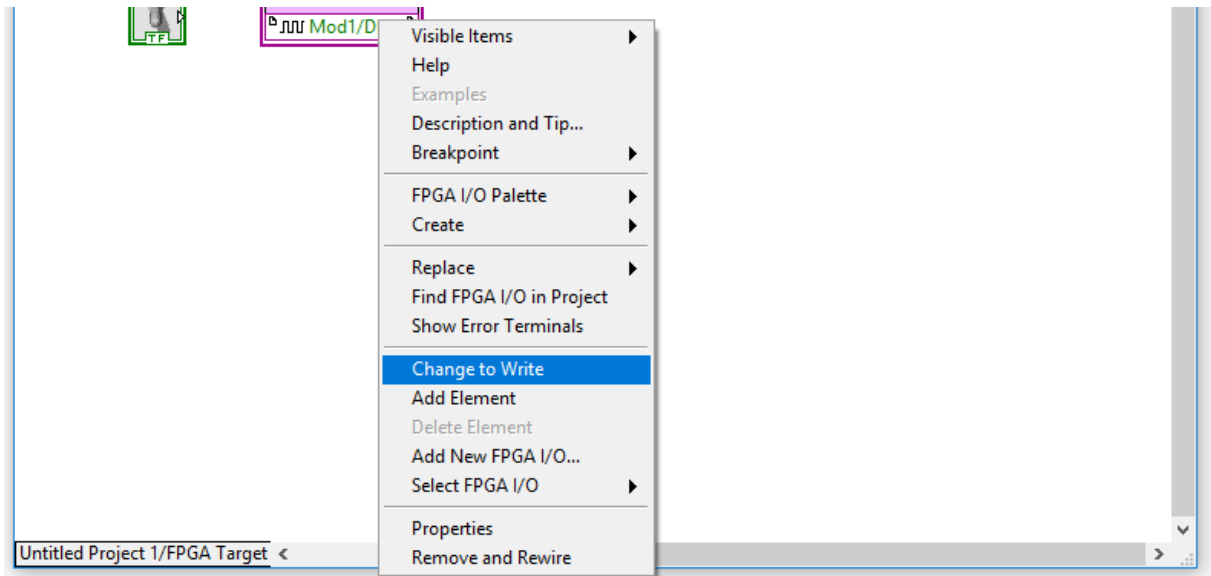




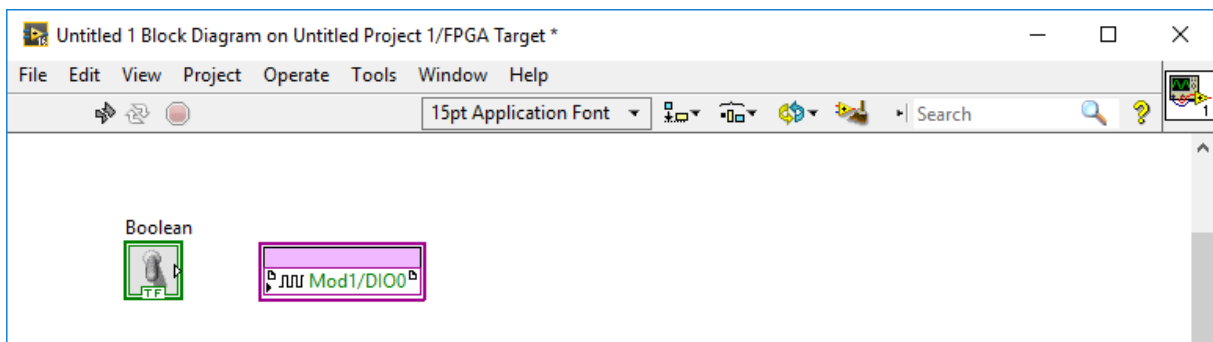
A lehelyezett *FPGA I/O Node* szimbólum *I/O Item* feliratára jobbgombbal kattintva hozzá tudjuk rendelni ezt a grafikus blokkot valamelyik tényleges fizikai csatornához, itt ez legyen NI-9401 modul DIO0 csatornája.



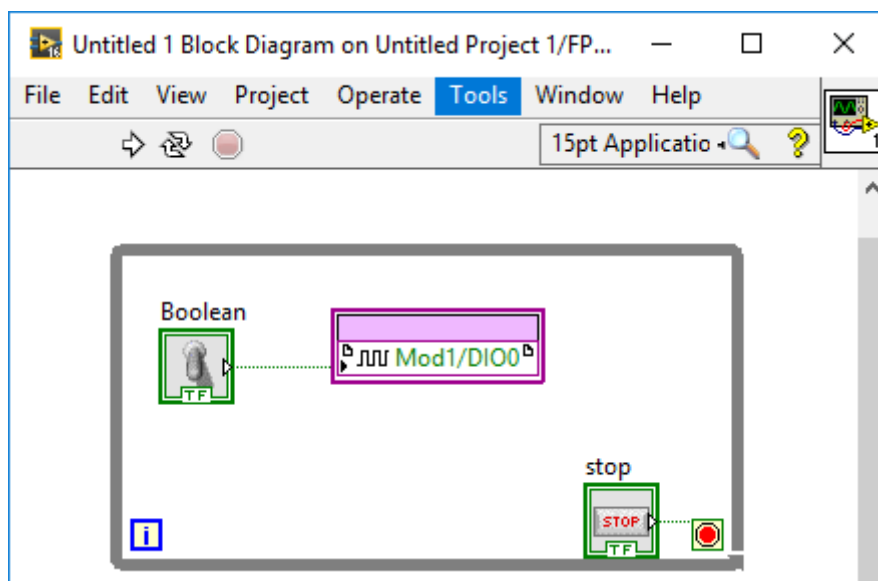
Mivel írni szeretnénk az adott portra, váltsuk át a *Mod1/DIO Node*-ot írhatóvá, a hozzá tartozó jobb egérgombbal elérhető menü *Change to Write* menüpontja segítségével!

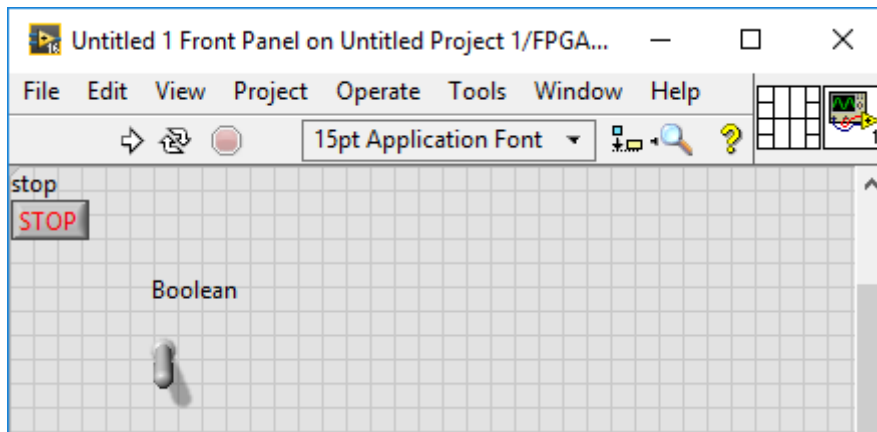


Vegyük észre, hogy a *Node* ikon jobb oldaláról átkerült bal oldalra a kis fekete háromszög, ezzel jelezve, hogy írható a modul. (Emlékezzünk rá, hogy a LabVIEW-ban balról jobbra értelmezik az adatfolyamot, vagyis a „dobozok” bal oldalán vannak a bemenetek és jobb oldalán a kimenetek.)

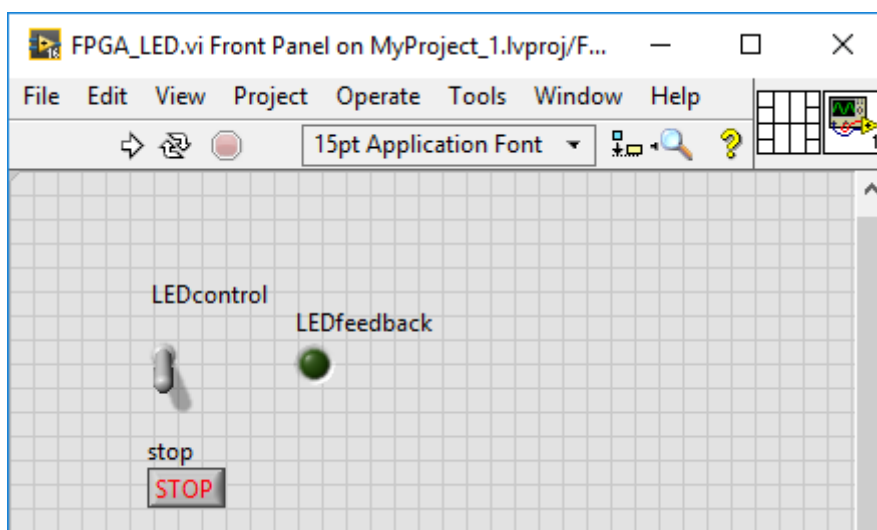
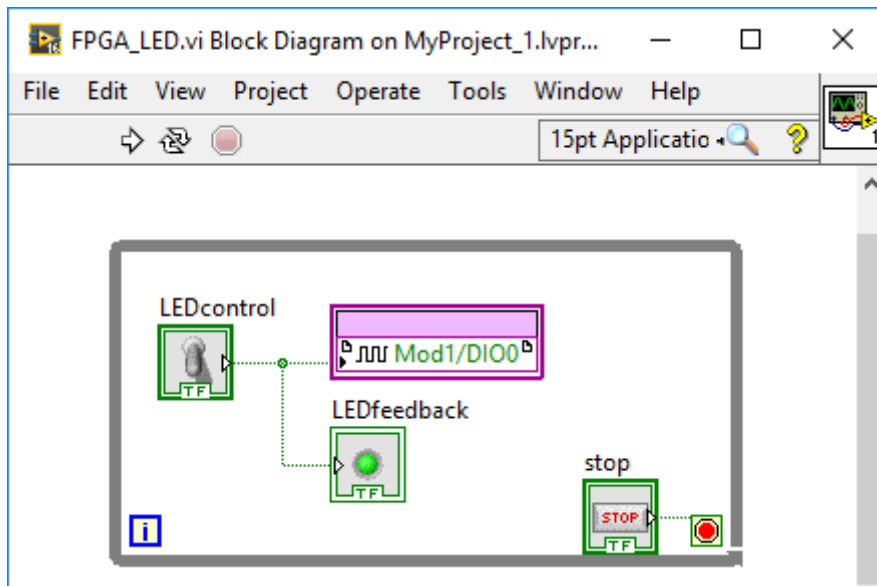


Huzalozzuk össze a blokkdiagramot, majd helyezzük az egészet egy végtelen ciklusba, célszerűen egy *while* ciklusba egy leállító *control* (STOP) gombbal együtt!



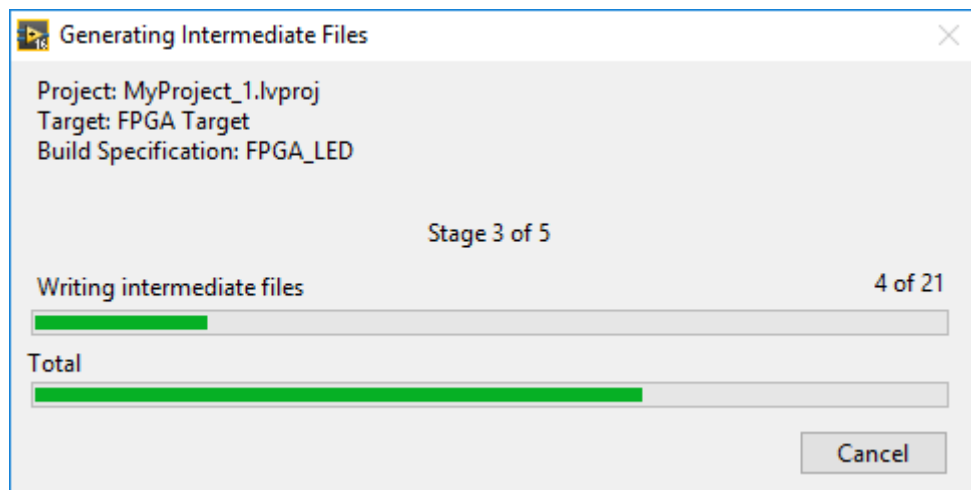
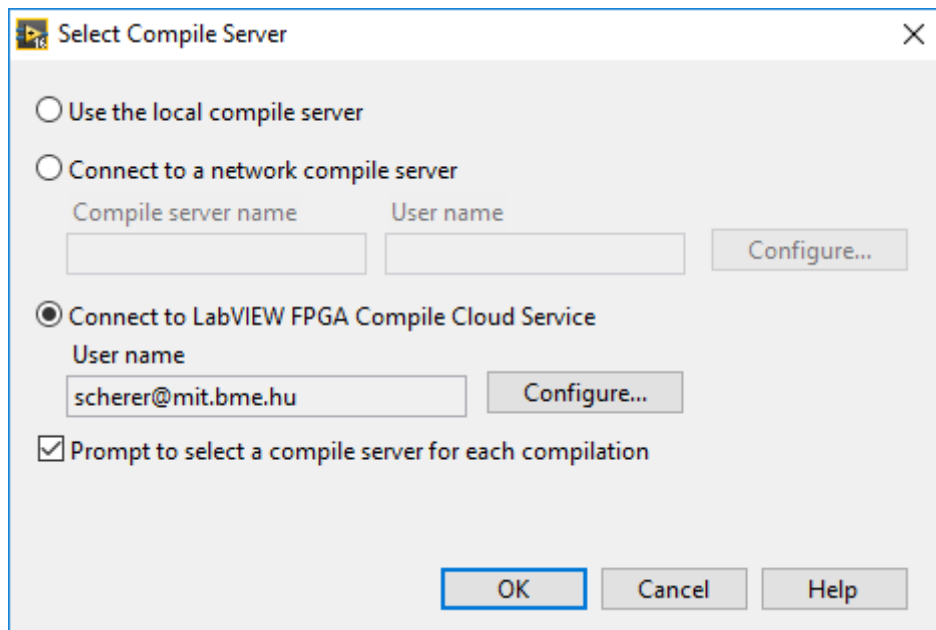


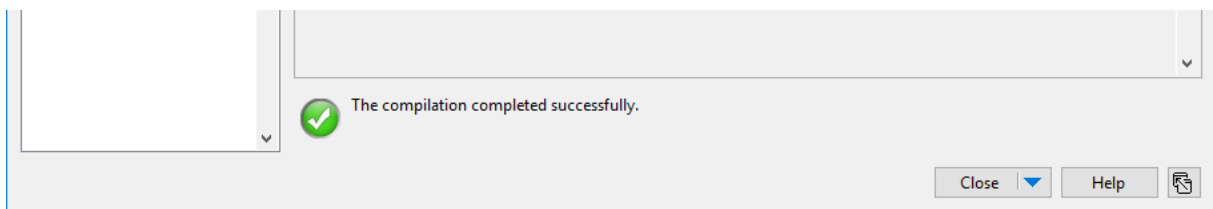
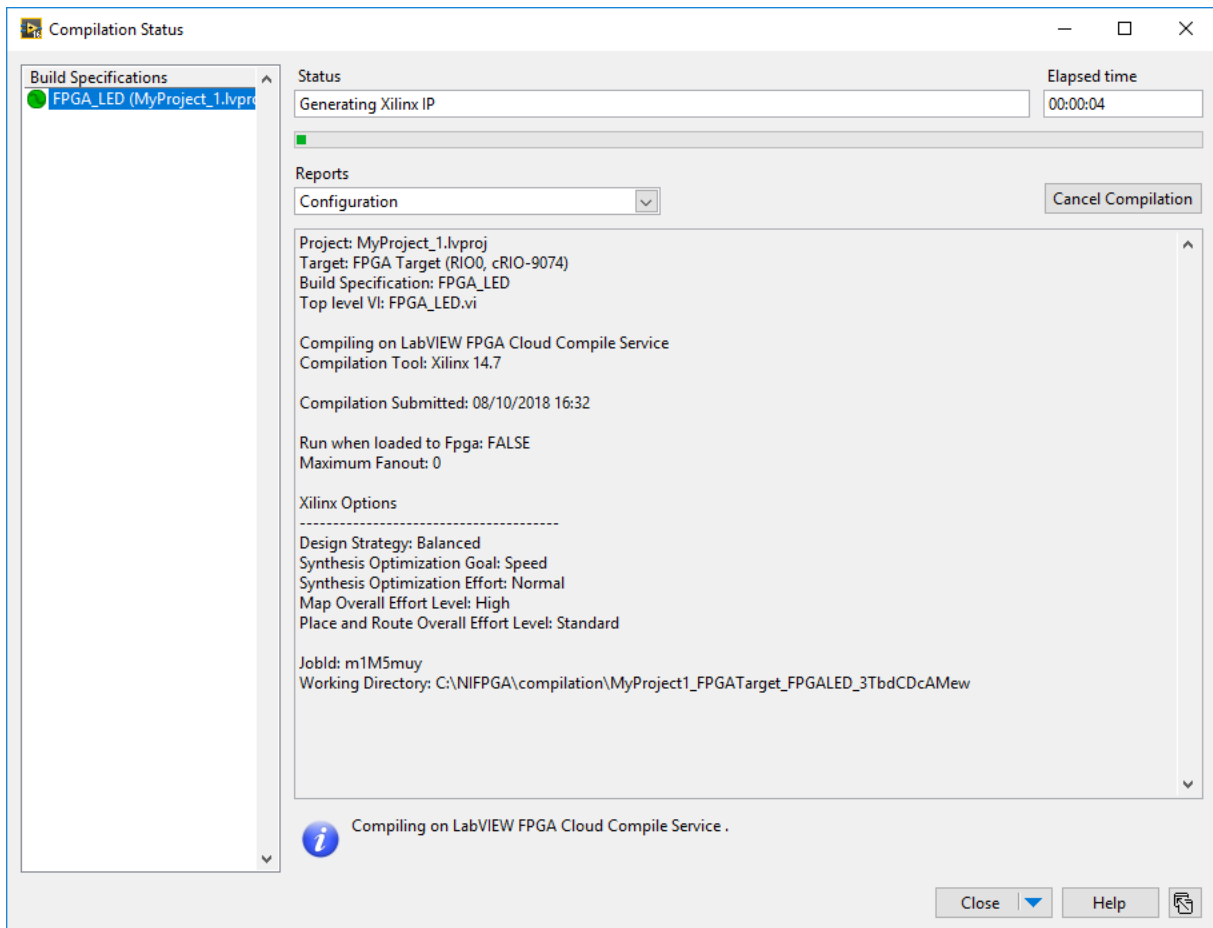
A Boolean kapcsoló nevét cseréljük le valamilyen beszédesebbre (pl. legyen a neve *LEDcontrol*), és helyezünk el a *Front Panel*-en egy ellenőrző LED-et, pl. *LEDfeedback* névvel!



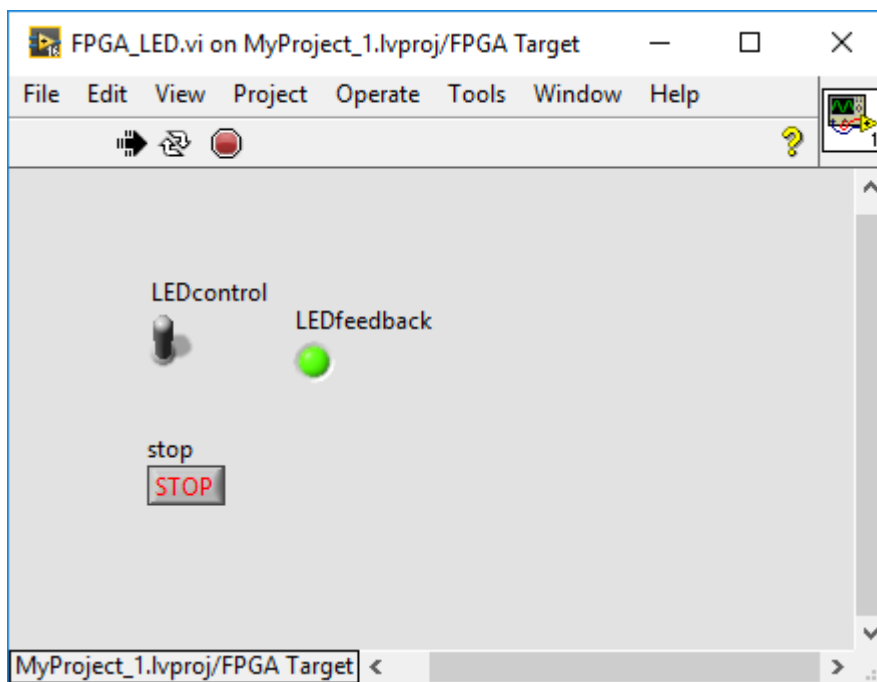
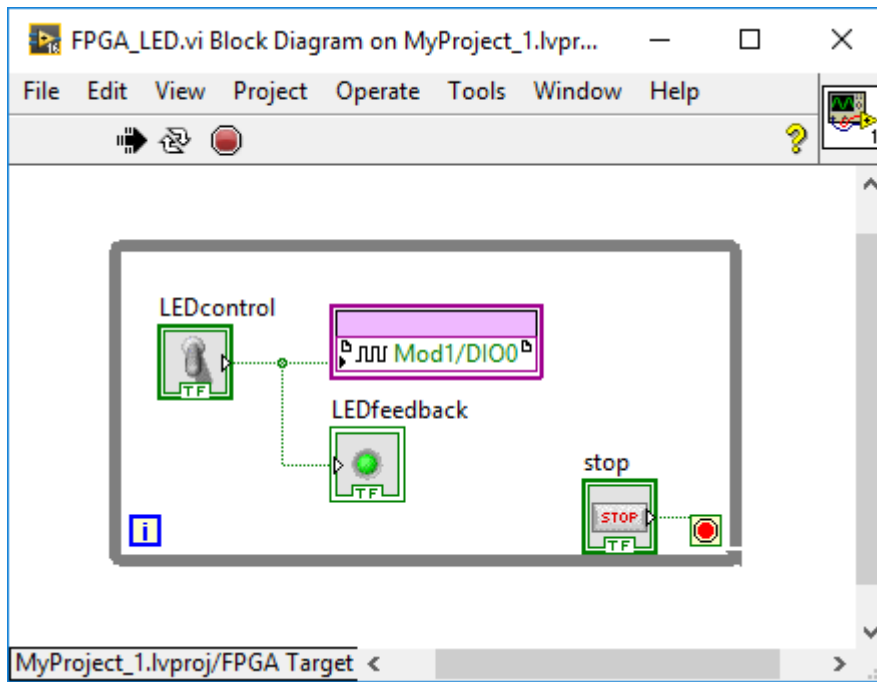
Mielőtt továbbmennénk, mentjük el az egész projektet és a VI-t valamilyen értelmes néven a diszkre!

Indítsuk el a programot (VI-t) a szokásos módon! Ennek hatására generálódik a hátlapban lévő FPGA-hoz tartozó huzalozási bitfile. Ez 4-5 percig is eltarthat. (Közben foglalkozhatunk a dokumentálással.) A programgenerálás végén megjelenik egy státuszriport, ami tájékoztat a fordítás eredményéről.



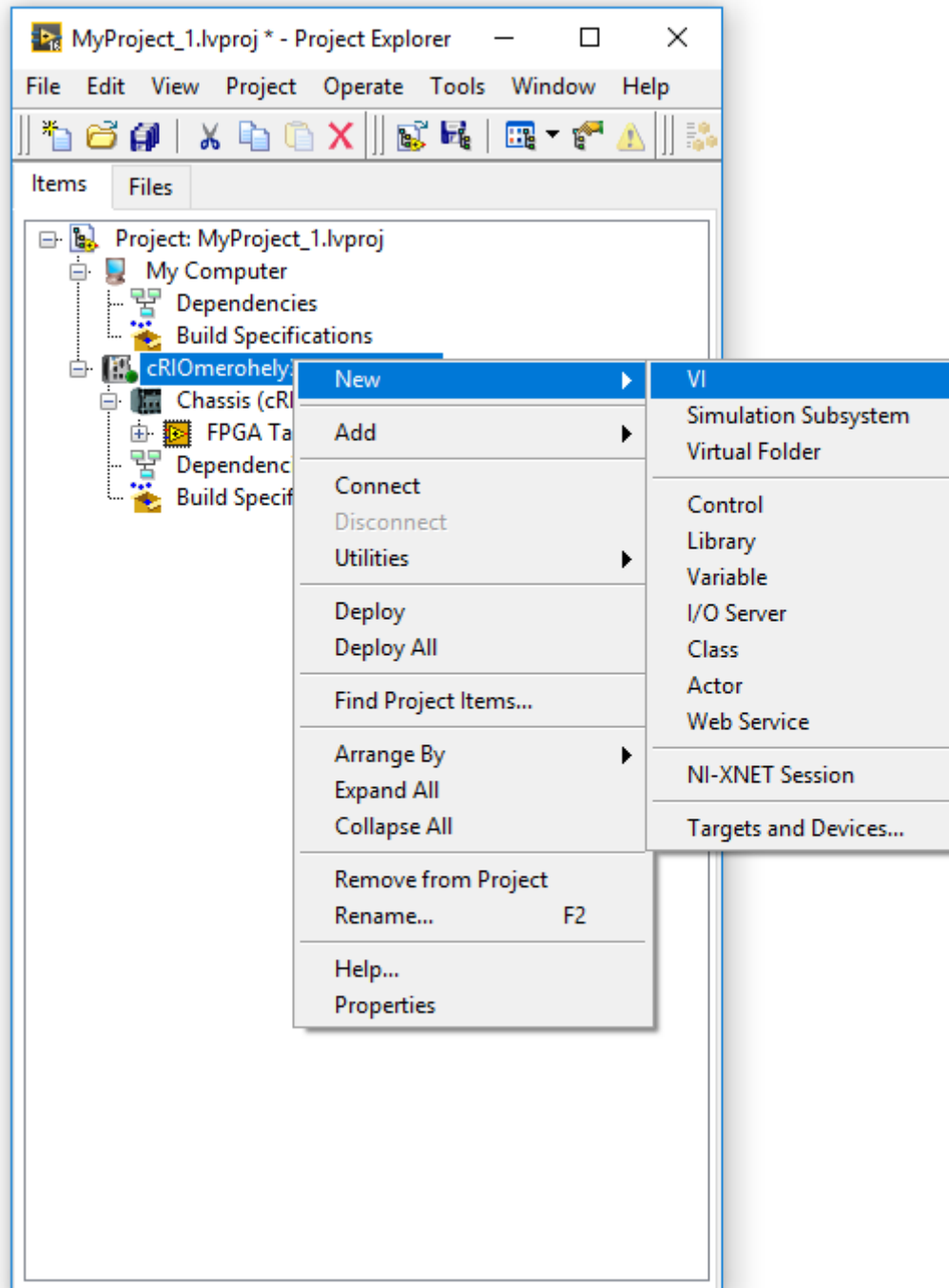


Sikeres fordítás után automatikusan elindul az FPGA VI, és lehetőségünk van kézzel vezérelni az NI-9401 DIO0 kimenetén a LED diódát és vele egyidejűleg a *Front Panel*-en a LEDfeedback jelző LED-et is. Ezzel megvalósítottuk a LED-villogtatás funkciót.

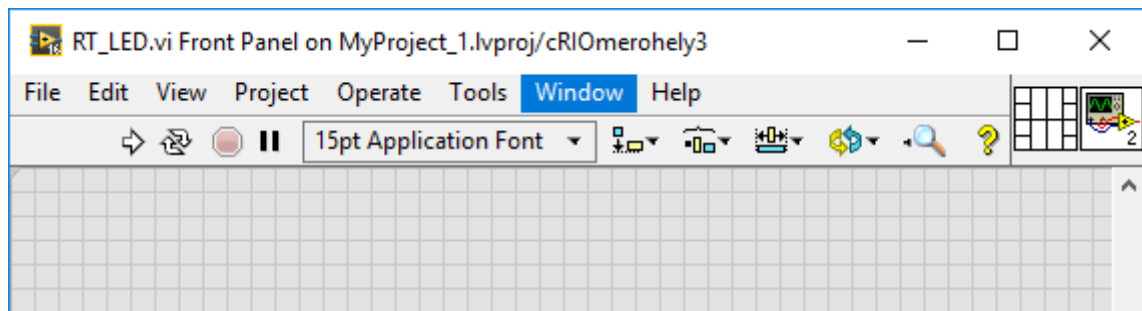
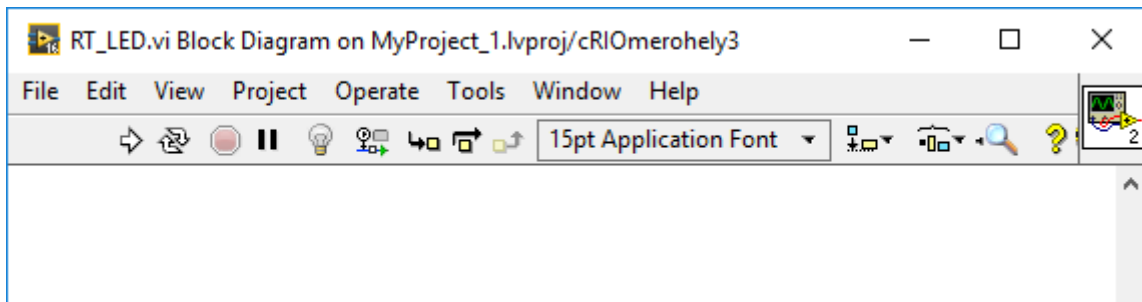


RealTime PC szint

Következő lépésként bővítjük a projektet úgy, hogy készítsünk programot a *CompactRIO*-ba integrált valós idejű vezérlőre is! Adjunk hozzá egy VI-t a *CompactRIO* modul real-time vezérlőjéhez!



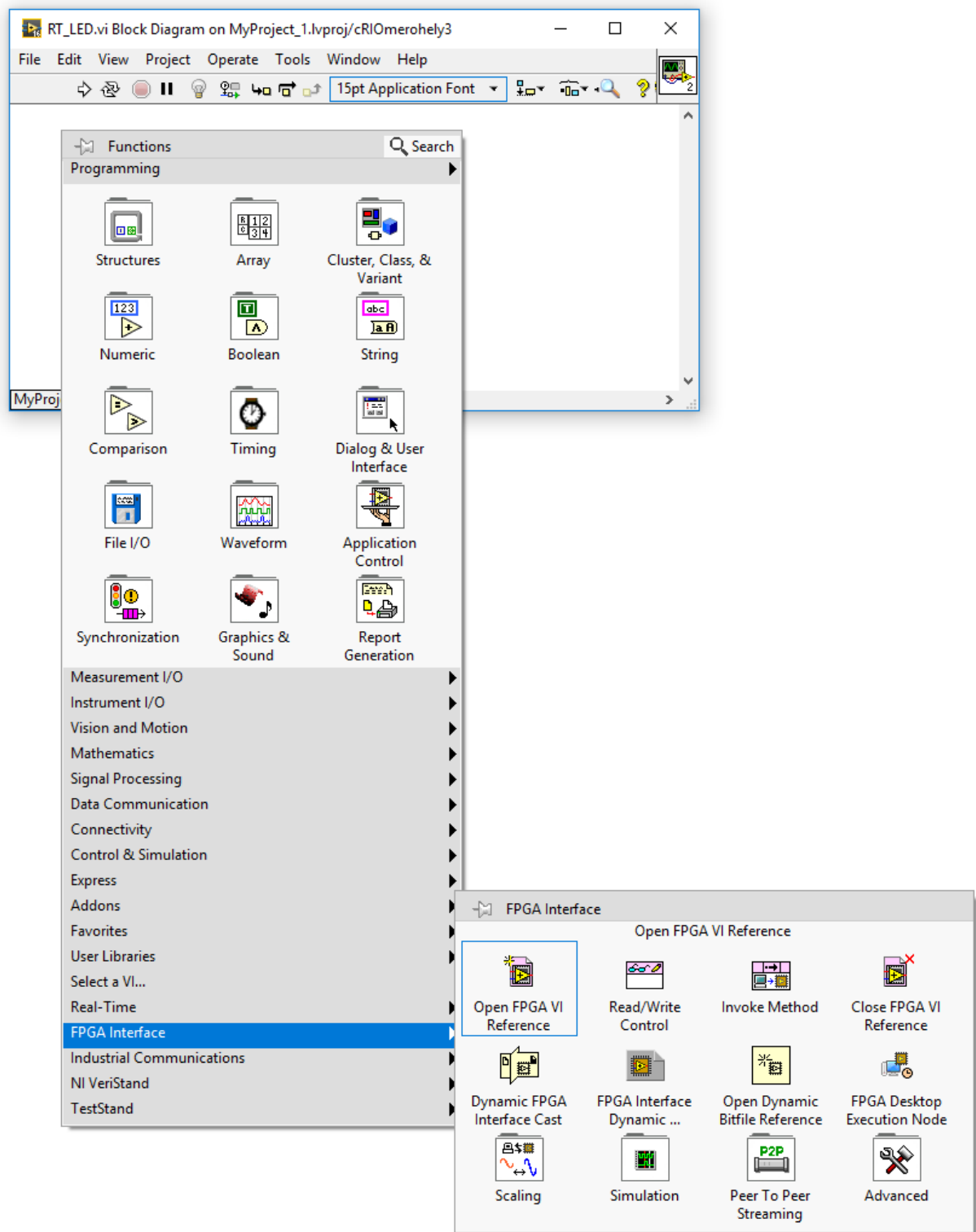
Rögtön mentjük is el valamilyen értelmes fájlnevével a VI-t!

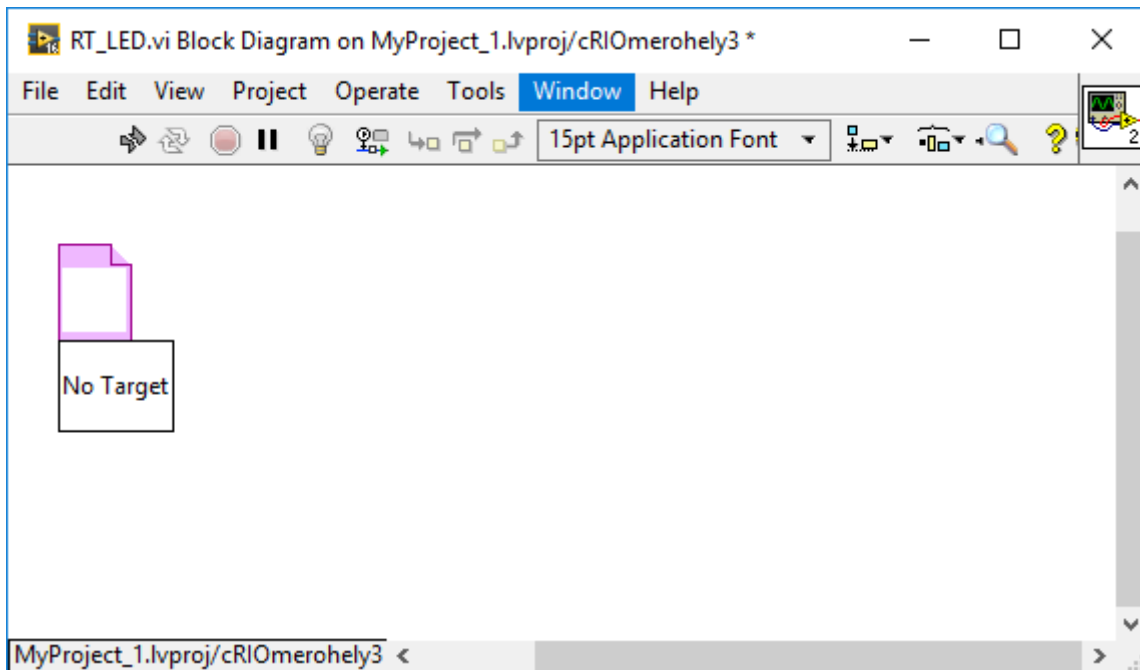


Ahhoz, hogy a CompactRIO real-time vezérlőjéből használni tudjuk az egyes I/O csatornákat, először kapcsolatot kell teremtenünk a FPGA program és a real-time vezérlő programja között. Ez nem a szokásos VI–SubVI kapcsolaton keresztül működik, de annyi hasonlóság van benne, hogy ebben az esetben is az FPGA-hoz tartozó VI *Front Panel*-jére helyezett bemeneteket és kimeneteket tudjuk mintegy függvényhívási felületként kezelni.

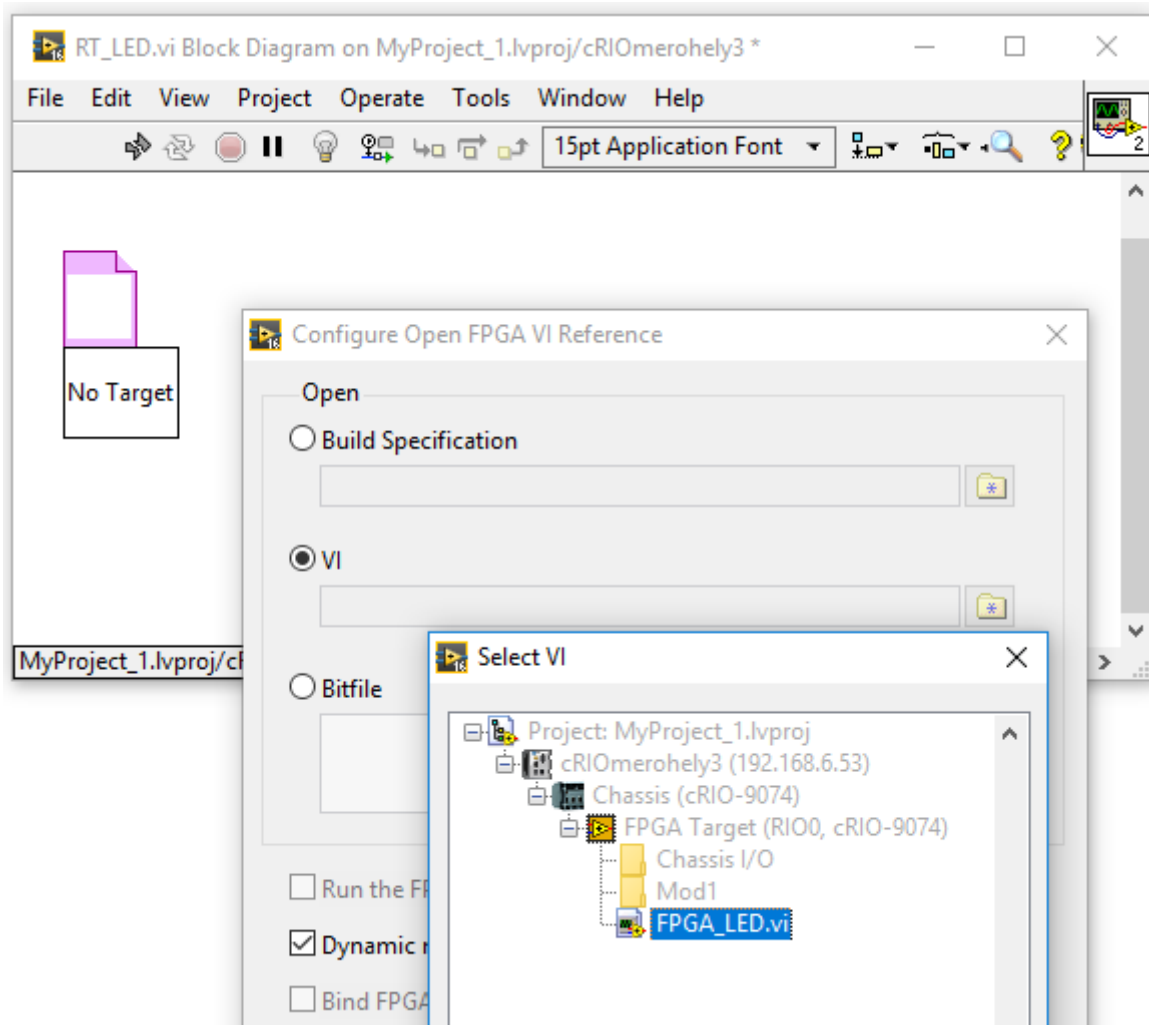
RT és FPGA közötti interface

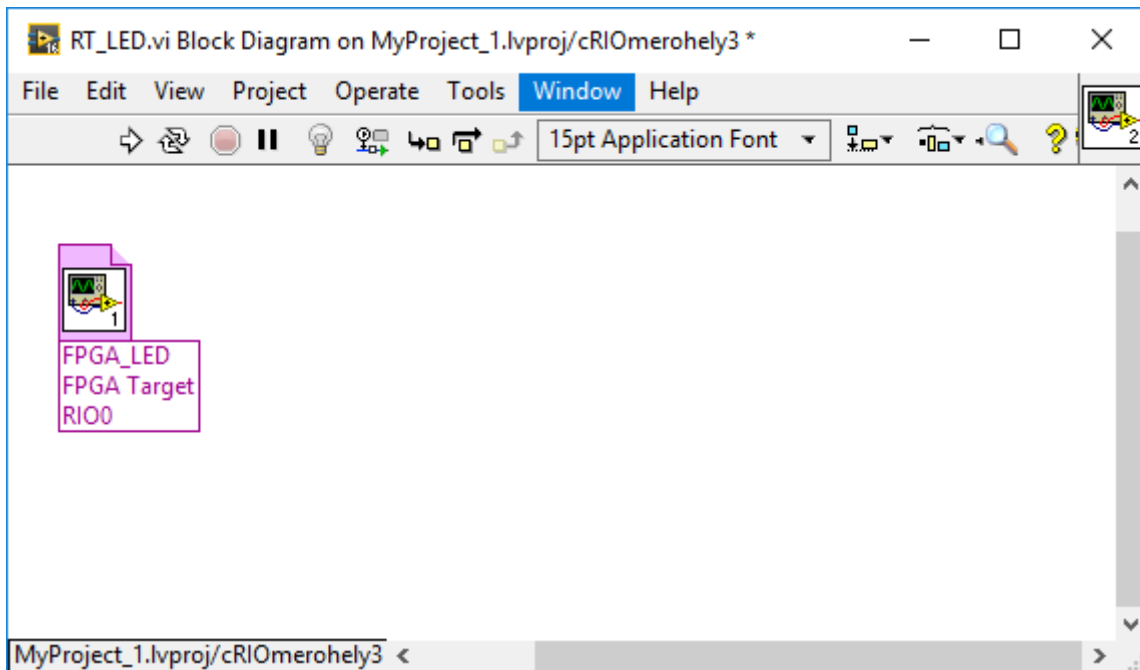
Az FPGA-val való kapcsolat megteremtéséhez először meg kell neveznünk azt az FPGA VI-t, amelyet erről a magasabb absztrakciós szintről vezérelni kívánunk. Ezt a *Block Diagram* palettájában található *FPGA Interface/Open FPGA VI Reference* blokkal tudjuk megtenni.



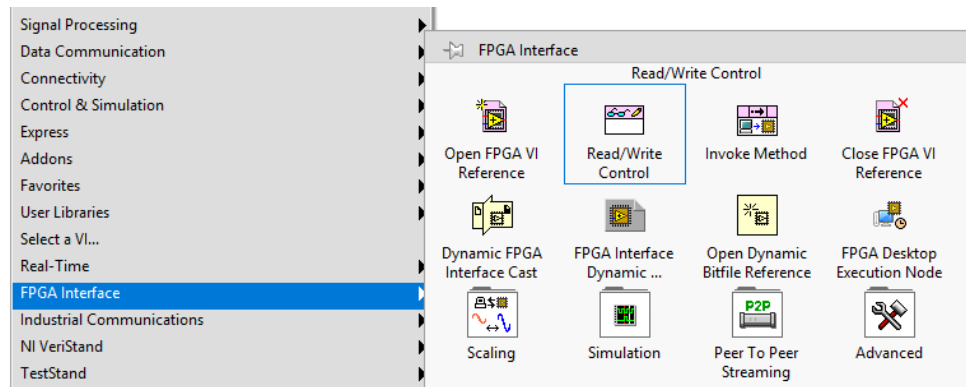


Az *Open FPGA VI Reference* VI helyezése után arra duplán kattintva ki tudjuk választani a projektben használt FPGA VI blokkot.

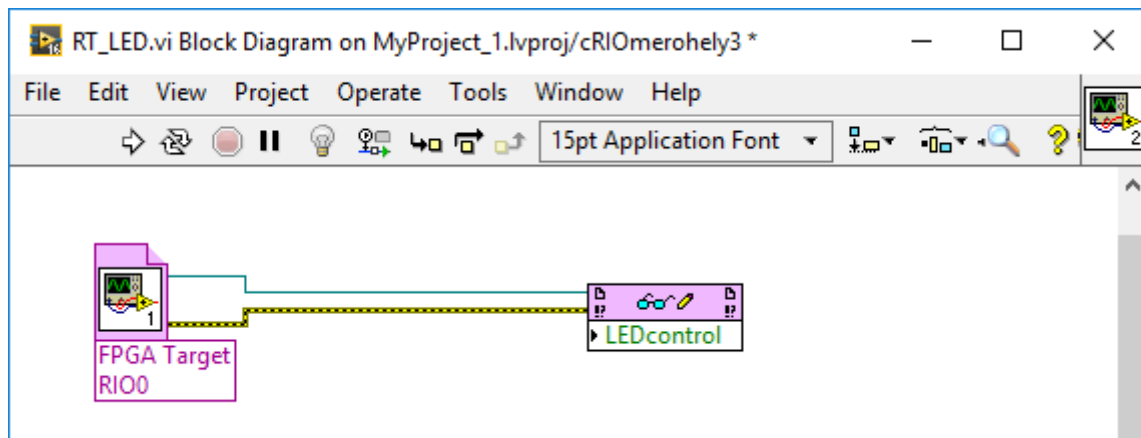
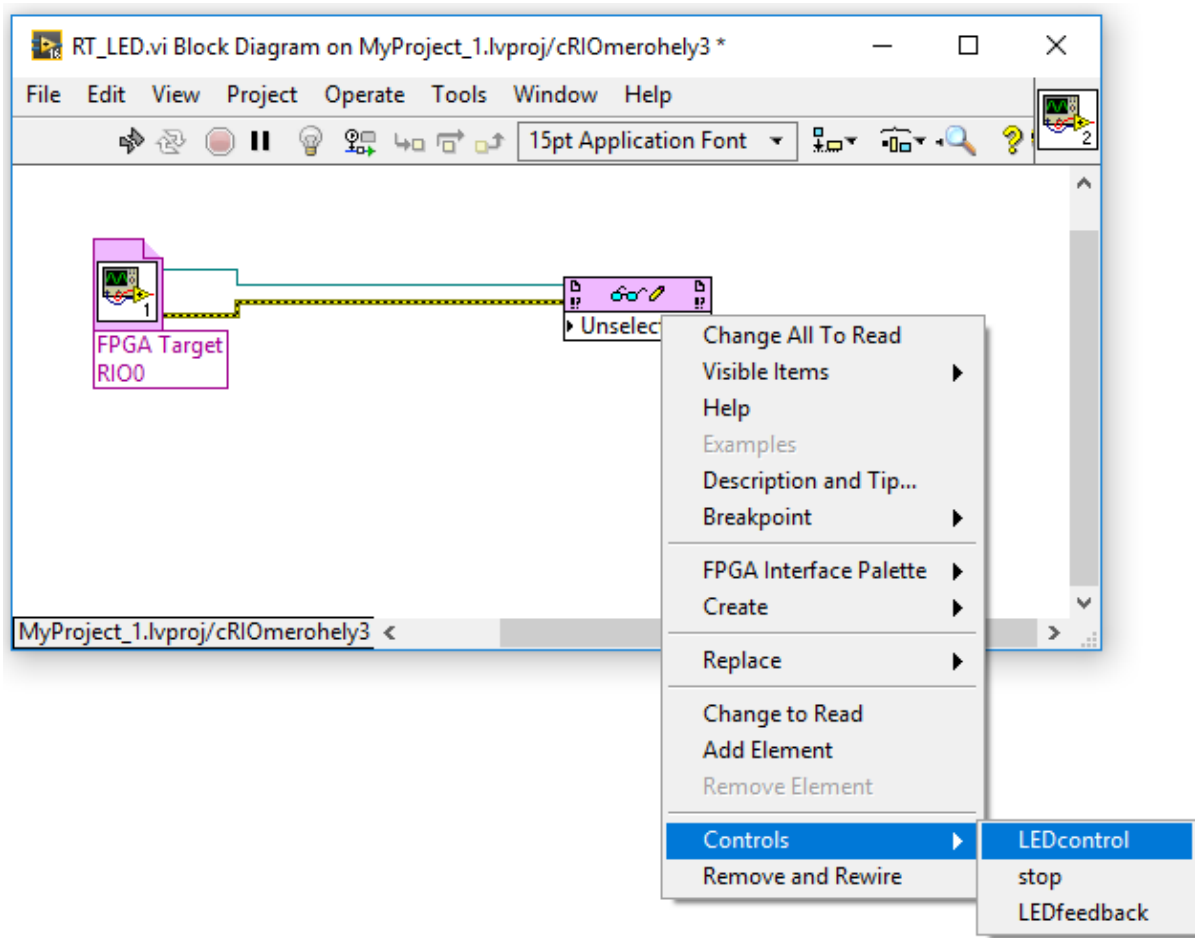




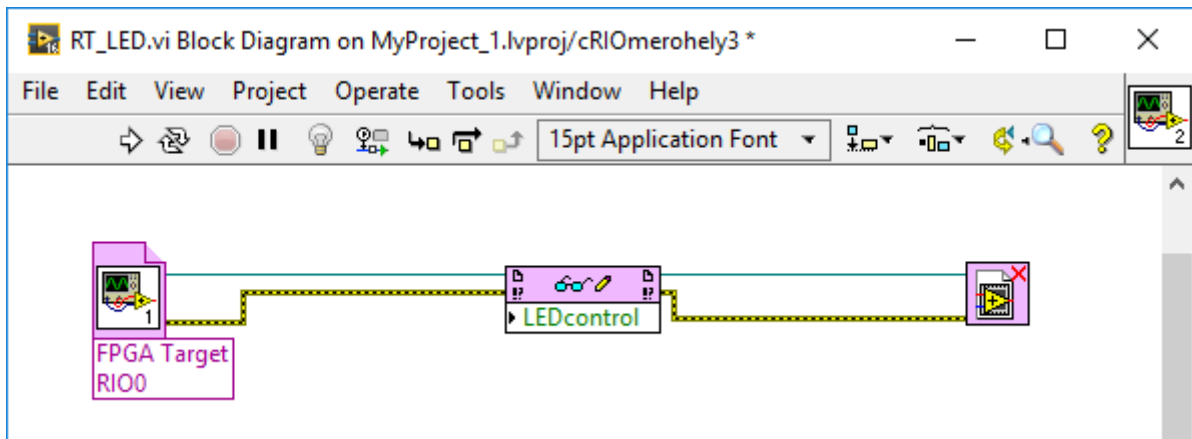
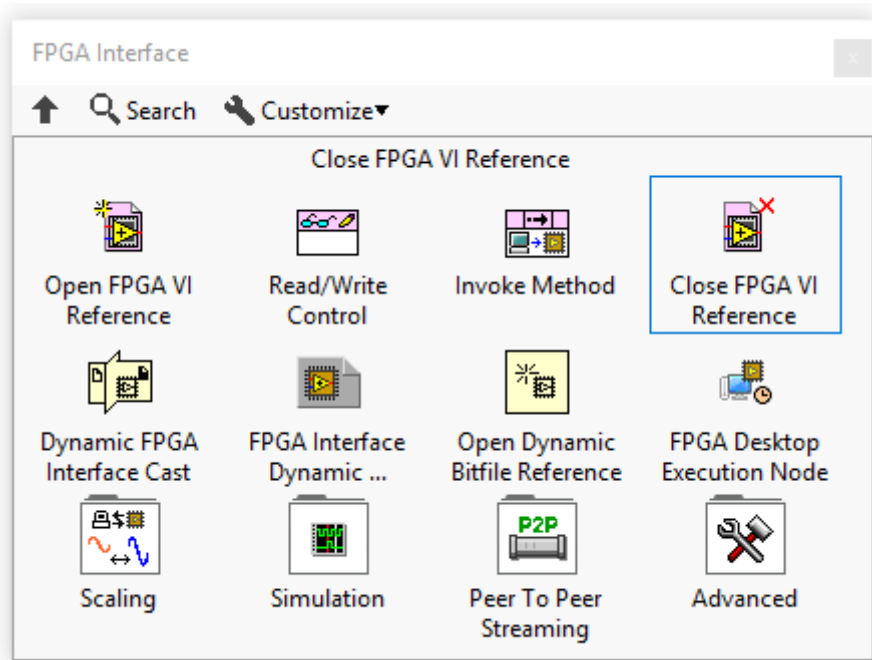
Következő lépésként a szintén az *FPGA Interface* könyvtárban található *Read/Write Control* VI-vel tudjuk az FPGA vezérlő VI *Front Panel*-jére helyezett bemeneteket és kimeneteket kezelni. Ezt lehelyezve, majd a referenciáját összehuzalozva az *Open FPGA VI Reference*-szel, ki tudjuk választani a számunkra szükséges be- és kimenetet.



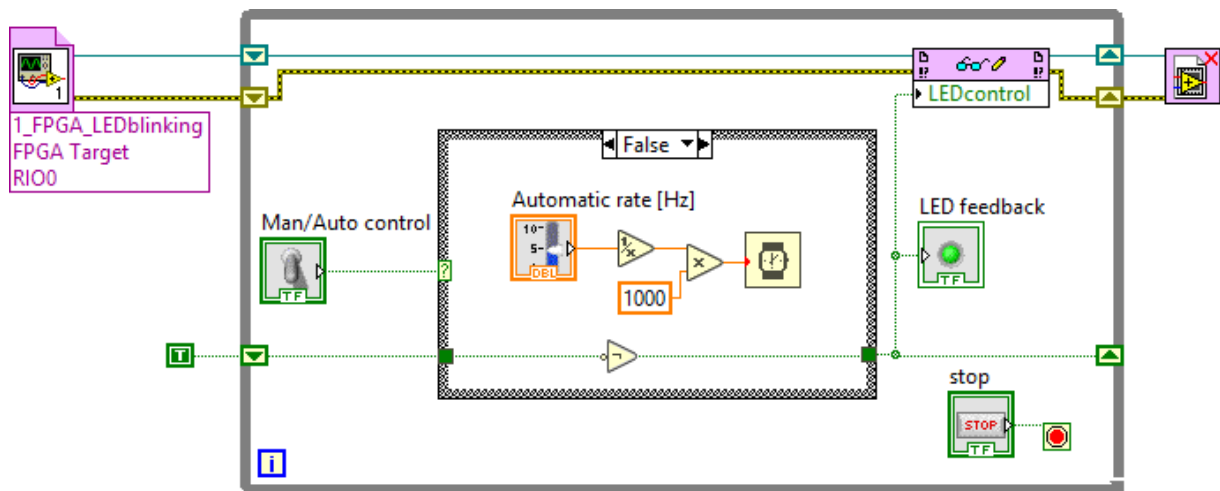
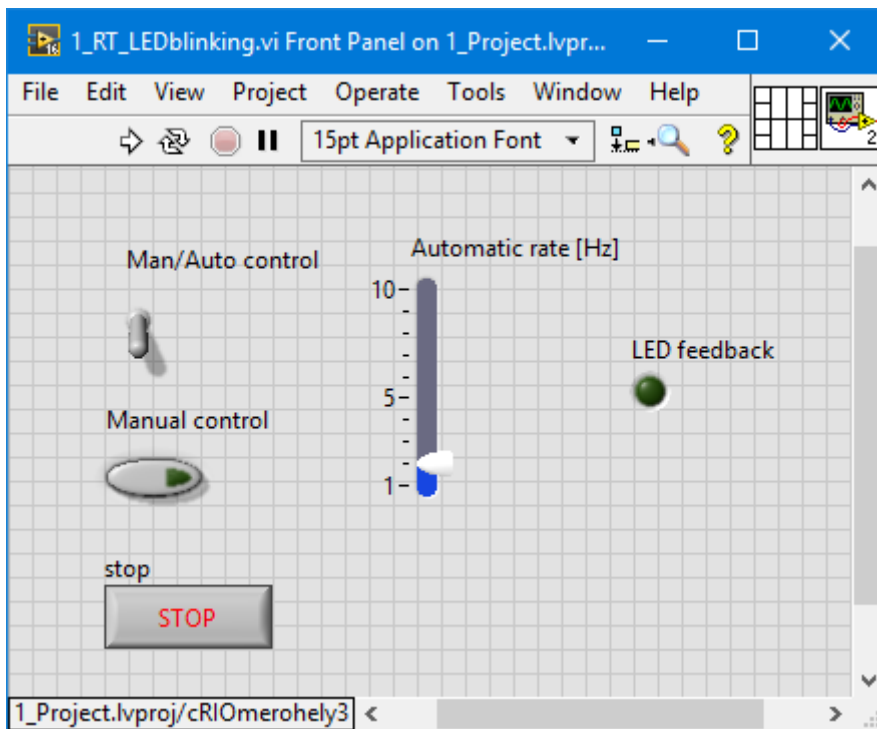
Kössük össze a két modul megfelelő lábait, majd válasszuk ki a *LEDcontrol*-t!



Ezután zárjuk le az FPGA interfészt az *FPGA Interface* könyvtár *Close FPGA VI Reference* blokkjával.

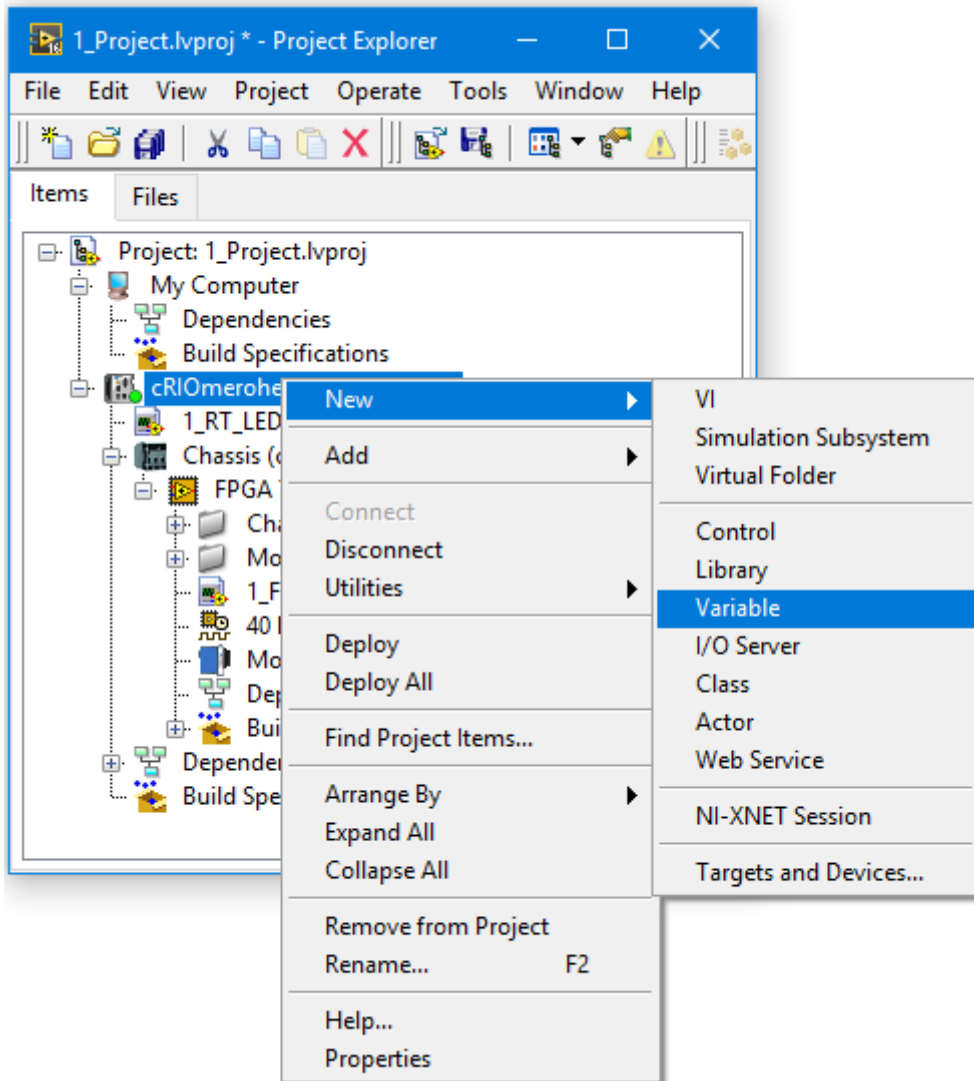


A magasabb szintű vezérlést demonstrálandó, készítsünk egy programot, amely beállítható frekvenciával vagy kézi vezérléssel villogtatja a LED-et.

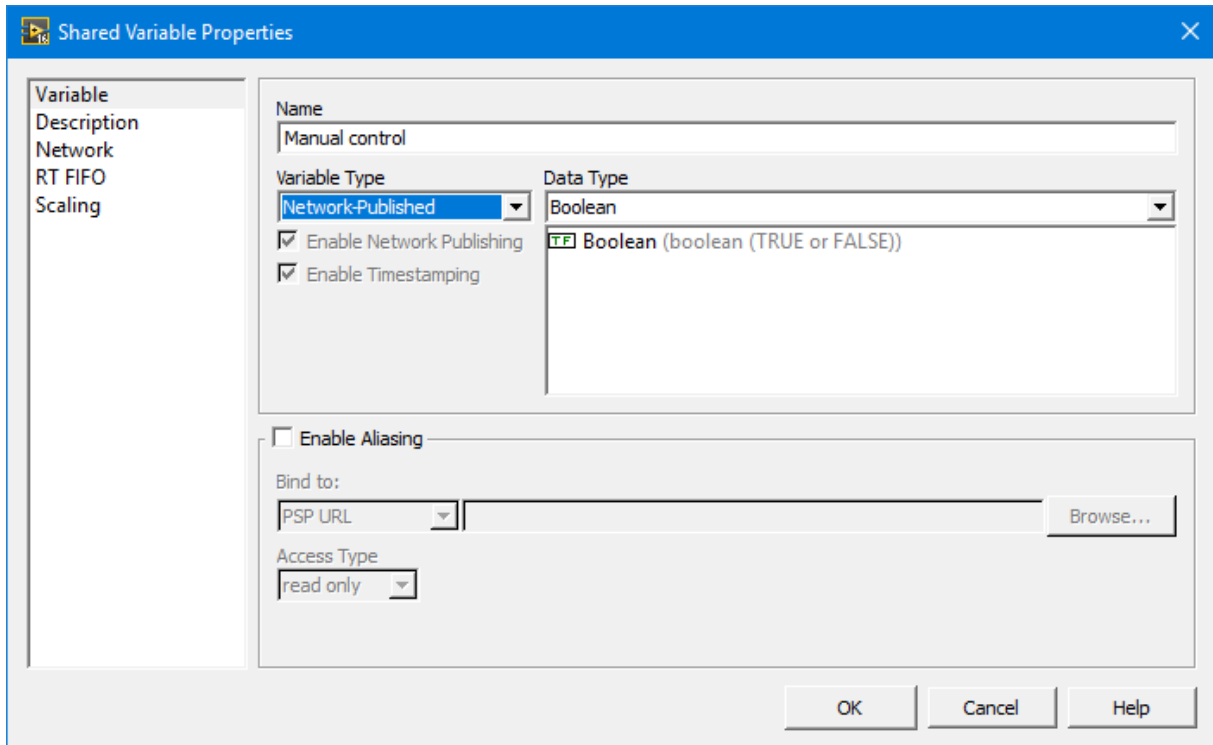


A kész VI-t elindítva (ugyanúgy, mint az FPGA-s esetben) letöltésre kerül a CompactRIO egységre a program, és elindul.

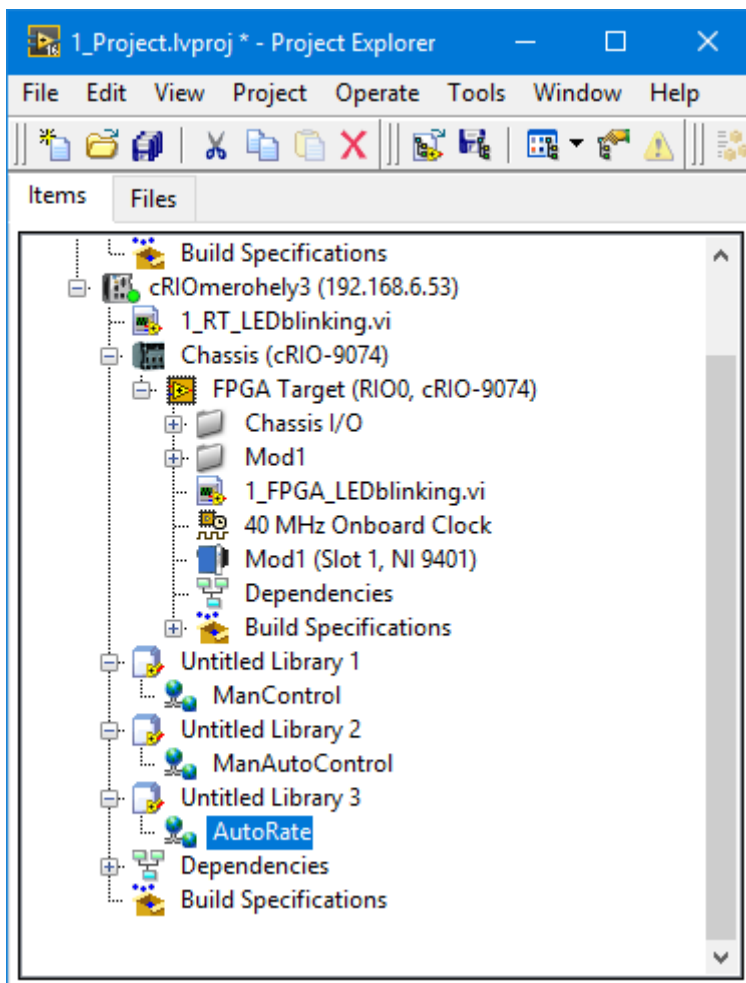
PC-s felhasználói felület Már tudjuk, hogy a terepre kihelyezhető modulra hogyan kell programot készíteni. Most az nézzük meg, hogy egy PC-n futó *Desktop* alkalmazás hogyan tud kapcsolatot tartani a cRIO-val. Erre az ún. *Shared variable* módszer használható a legegyszerűbben: megadjuk a compactRIO real-time PC-je és az asztali PC-n futó alkalmazás számára is, hogy mely változókat használják közösen, és a LabVIEW a két node közötti adatátvitellel automatikusan szinkronizálja a változók értékét.



Shared variable Adjunk hozzá a project CompactRIO részéhez három változót a real-time PC-n futó kezelőszervek (*Man/Auto control* kapcsoló, a *Man control* gomb és az *Automatic rate [Hz]*) kiváltásához (pl. *ManAutoControl*, *ManControl* és *AutoRate* névvel).



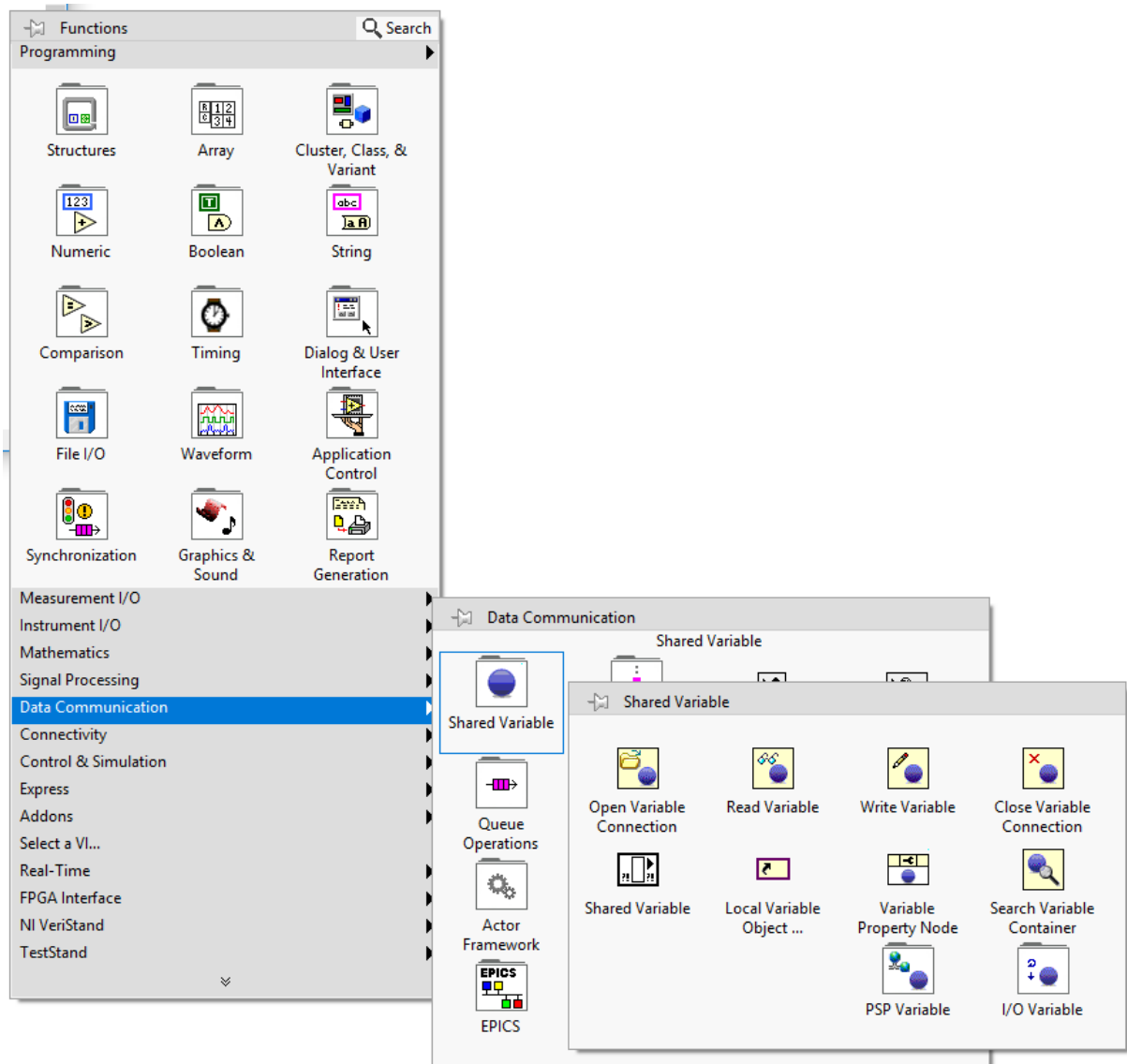
Az eredmény a projektablakban:



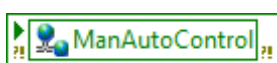
Hozzunk létre egy VI-t a *Desktop* részhez is a felhasználói felület számára.

Adjuk hozzá az új VI *Front Panel*-jéhez a megfelelő kontrolokat.

Hozzunk létre egy új osztott változót, a *Block diagram Data Communication/Shared Variable/Shared Variable* VI-jával.

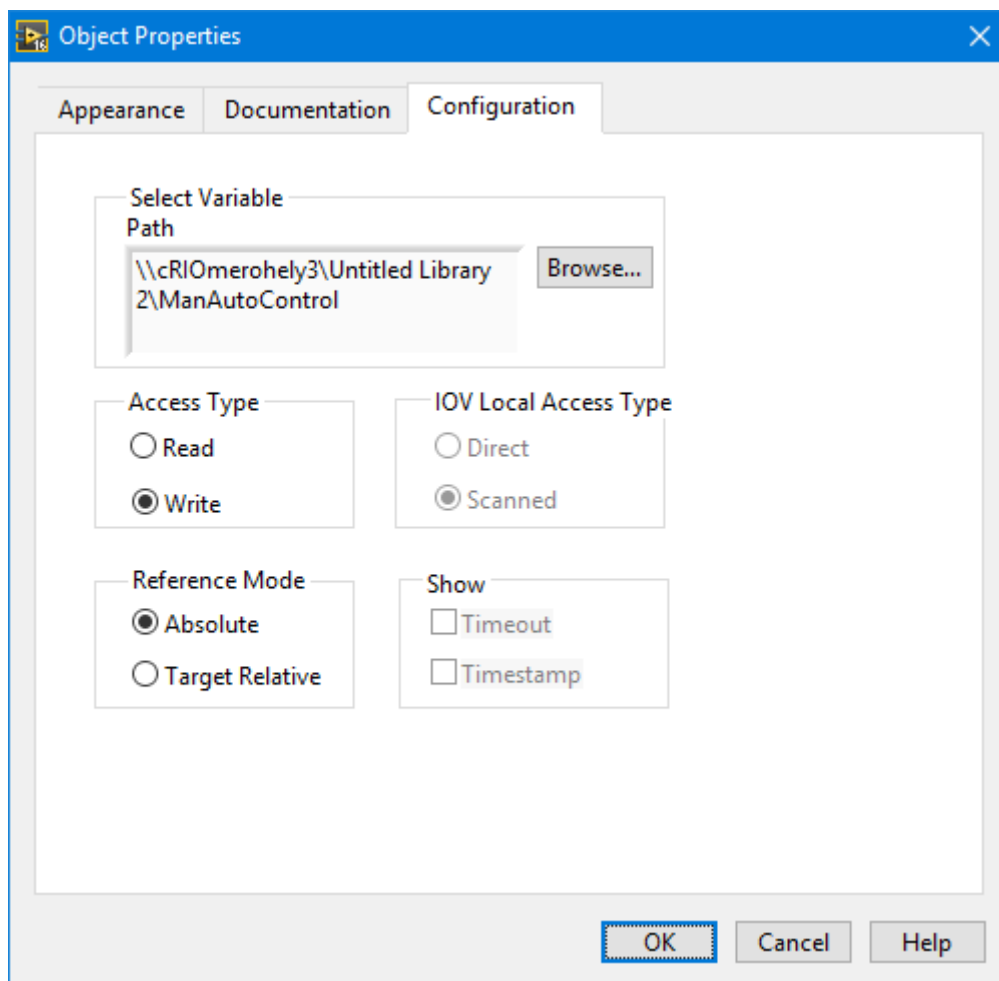


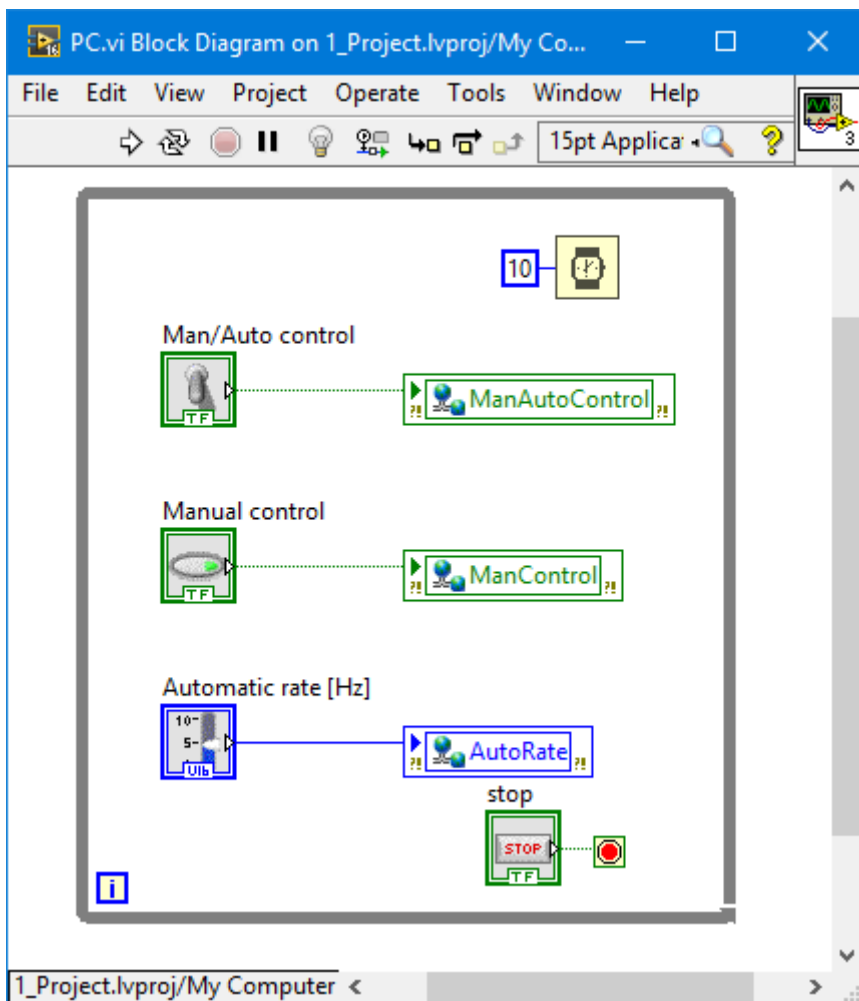
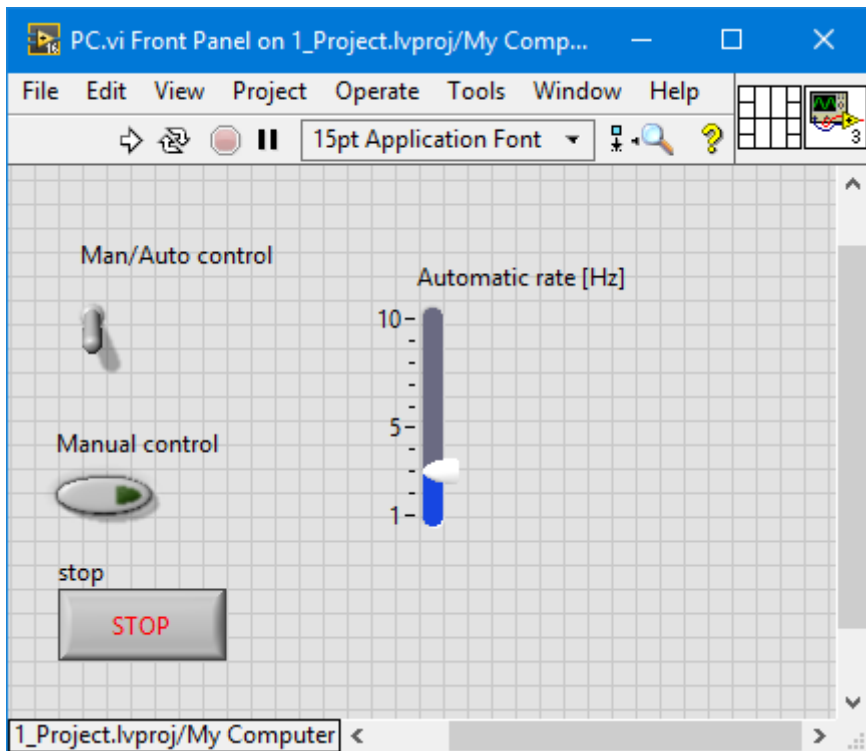
A lerakott ikonra kattintva hozzá tudjuk rendelni a VI-t a projekt egyik osztott változójához (*Select Variable*).

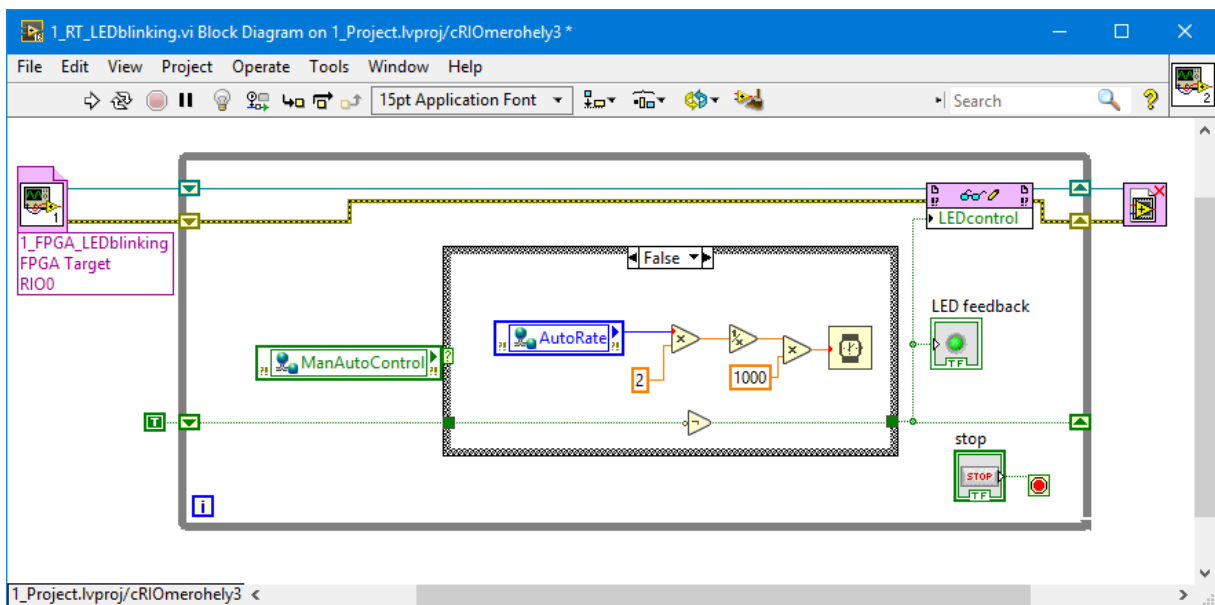
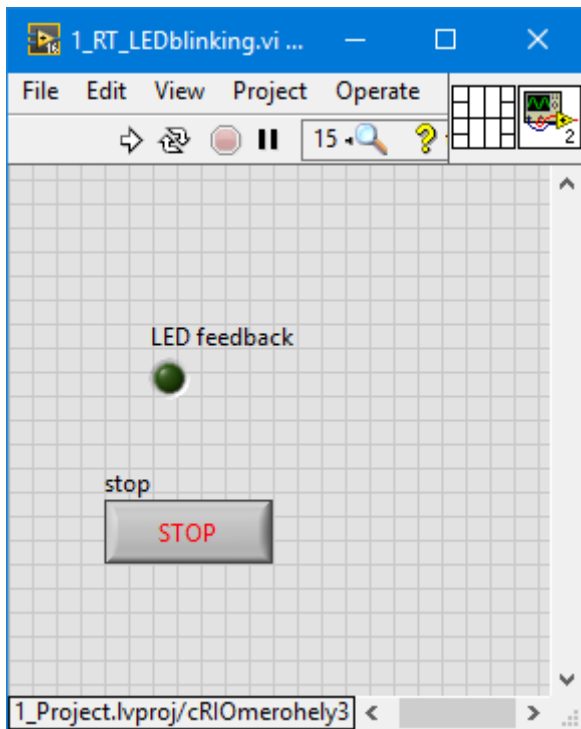


Azt, hogy a változóba írunk vagy olvasunk a jobb egérgombbal elérhető *Properties* menü *Configuration* ablakában tudjuk megadni.

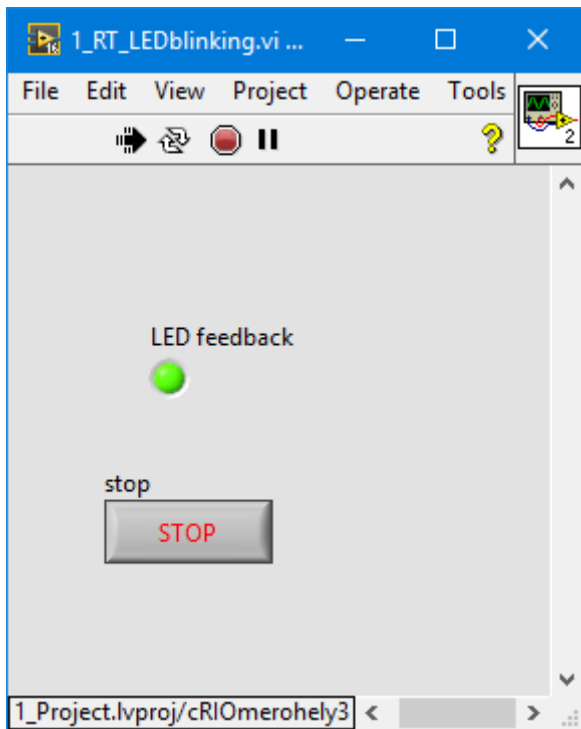
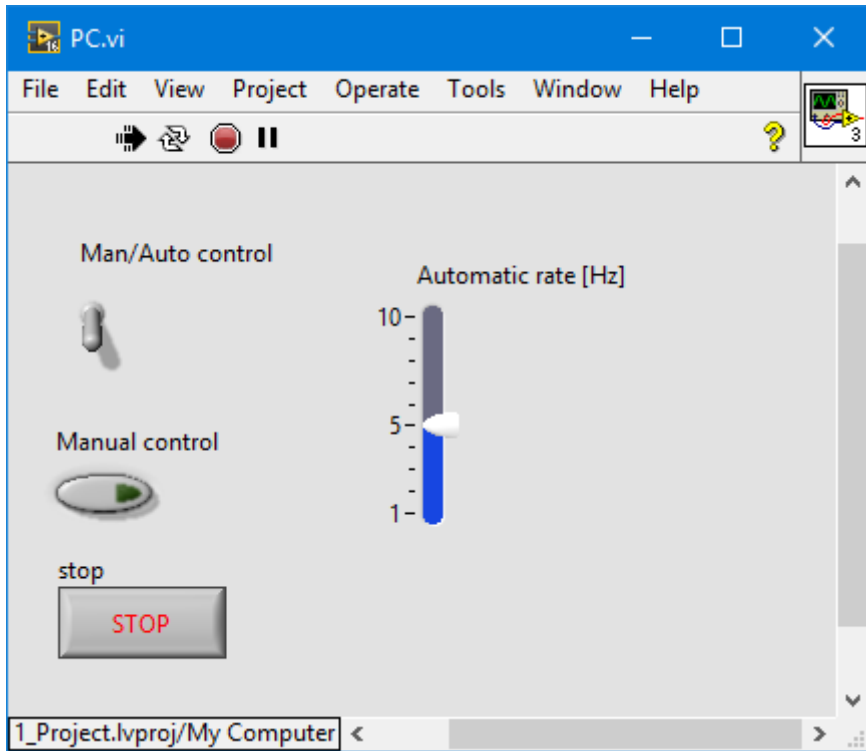
Ismételjük meg ezt a műveletet az összes többi változóra, valamint a CompactRIO VI-ra is.







A rendszer működtetéséhez minkét VI-t el kell indítani.



Tanácsok

Bonyolultabb rendszereknél célszerű hibavédelmet is alkalmazni, illetve a real-time rendszeren lévő VI leállítását megoldani a desktop PC-ről.

Sok esetben célszerű az ún. ütemezett ciklusokat használni (*Timed Loop*) mert ezek gyakorlatilag különálló szállakká képződnek le a végrehajtásnál, ezért garantáltan párhuzamosan működnek, illetve jól időzíthetőek. Jelen esetben például, ha a desktopon futó VI-ból kiszedjük a késleltetést, akkor az akár a rendszer fagyásához is vezethet, mert annyira túltömi az osztott változós kommunikációt.

Szintén érdemes megjegyezni, hogy a fejlesztés során a hierarchia egyes szintjeit akár át is ugorhatjuk, például nem kötelező a CompactRIO real-time vezérlőjéhez programot írni, lehet közvetlenül a desktop PC-ről vezérelni az FPGA VI-t.