

# PLC GX Works2 alapjai

Ez a tanfolyam (e-learning) azok számára készült, akik a GX Works2 szoftverrel első alkalommal készítenek szekvencia programot.

A tanfolyam alapvető ismeretet nyújt a GX Works2 szoftverrel való programozásról, a hibakeresésről és a programozható vezérlő (PLC) működésének ellenőrzéséről. A tanfolyam azok számára készült, akik a MELSEC-Q, a MELSEC-L vagy a MELSEC-F sorozat vezérlőihez készítenek szekvencia programokat.

Itt találja a tananyagban foglalt témaköröket.  
Javasoljuk, hogy a tanulást az 1. fejezettel kezdje.

### 1. fejezet – A PLC-rendszer vezérlése

Itt a programozás során használt programnyelvet és a szoftvert mutatjuk be.

### 2. fejezet – Programtervezés

Megtanulja a vezérlőelemek és hardverkonfiguráció alapján történő programtervezést.

### 3. fejezet – Programozás

Megtanulja az erre a célra szolgáló GX Works2 szoftverben való programozás mikéntjét.

### 4. fejezet – Hibakeresés

Megtanulja, hogy a CPU-modulhoz hogyan lehet szekvencia programokat írni, majd ezeken hibakeresést végezni.

### 5. fejezet – Záró teszt

Sikeres eredmény: 60% vagy afölött.

Tovább a következő oldalra		Tovább a következő oldalra.
Vissza az előző oldalra		Vissza az előző oldalra.
Ugrás a kívánt oldalra		Megjelenik a „Tartalomjegyzék”, ahol lehetőség van a kívánt oldal elérésére.
Kilépés a kurzusból		Kilépés a tanfolyamból. A „Tartalom” képernyő és a kurzus egyéb ablakai bezáródnak.

**Biztonság**

A tanulás során valós termékeken kell kipróbálnia a műveleteket, ezért kérjük, mindig gondosan olvassa el a megfelelő használati útmutatót.

**Megjegyzések a tananyag tartalmával kapcsolatban**

- Az Ön által használt szoftververzióban megjelenő képernyők eltérhetnek a tananyagban szereplőktől.

# 1. fejezet A PLC-rendszer vezérlése

Ezt a tanfolyamot azoknak ajánljuk, akik tervezőszoftverekkel dolgoznak. A tananyag a MELSEC-Q, L és F sorozatú rendszerek kezelésének néhány alapvető elvét foglalja magában.

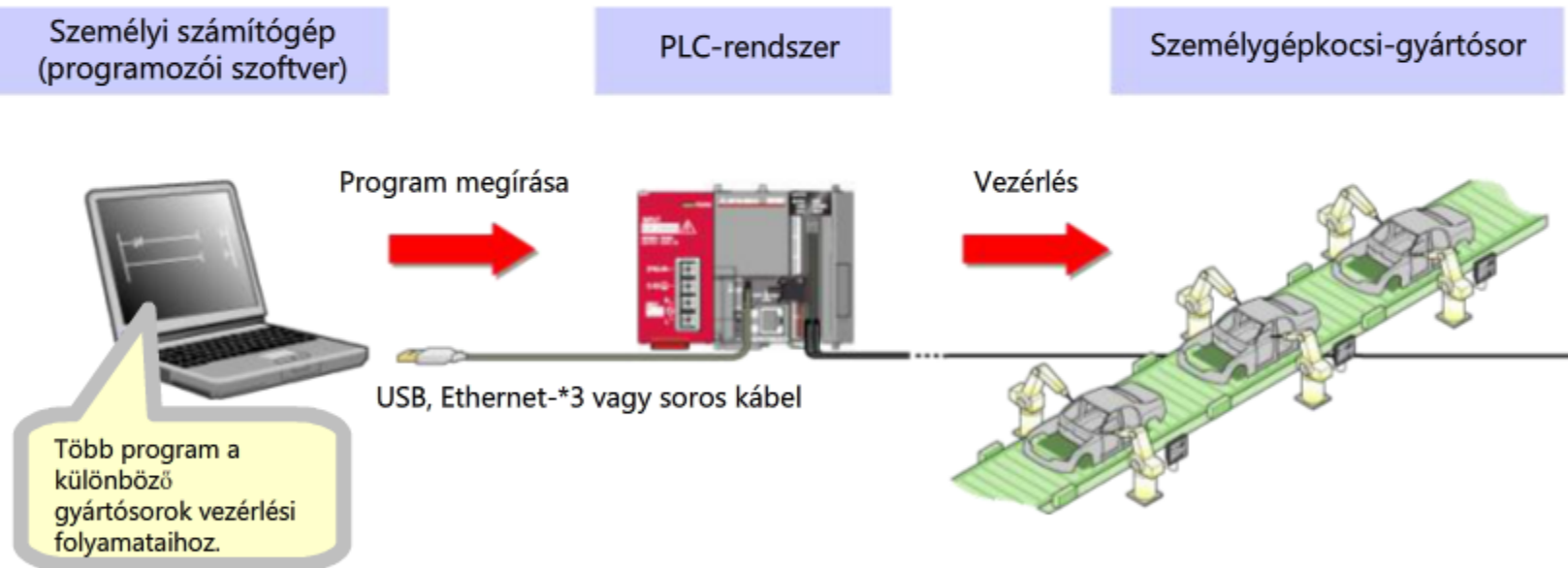
A GX Works 2 (GXW2) nemzetközileg szabványosított programnyelveket használ: szekvenciális funkcióábrát (SFC), utasításlistát (IL)\*1, létradiagramot, funkcióblokk-diagramot (FBD)\*2 és strukturált programnyelvet (ST).

A programok személyi számítógépen, a GX Works2 „tervezőszoftver” segítségével készülnek, melyeket USB-n, Ethernet-kábelen\*3 vagy soros kábelen keresztül rendszerint átírnak a programvezérlő CPU-ba. A CPU-modul tetszőleges alkalommal újraprogramozható, hogy a kívánt vezérlésben bekövetkező bármilyen kért változtatásnak megfeleljen.

\*1 Jövőbeli terv a GX Works2 számára.

\*2 A GX Works2-ben használt jelenlegi elnevezése strukturált létra, IEC (Nemzetközi Elektrotechnikai Bizottság) megfelelés tervezés alatt.

\*3 Az Ethernet a Xerox Corp. regisztrált védjegye.



A tanfolyam példaprogramjában a létradiagramot használjuk, mely az egyik legnépszerűbb PLC programnyelv. Habár a példaprogramban L sorozatú PLC szerepel, a tananyag tartalma a Q sorozatú rendszerekre ugyanúgy érvényes.

Az alapvető vezérlési eljárás a MELSEC-F sorozat esetén is azonos, ám egyes műveletek és funkciók eltérnek.

Ez az e-learning tananyag a programozható vezérlőrendszer alkalmazásához szükséges (zölddel kiemelt) szoftvertervezési lépéseket tárgyalja.

### Hardverkialakítás

(1) Rendszer-kialakítás ..... MELSEC-Q/MELSEC-L kezdő tanfolyam

(2) Termékválasztás ..... MELSEC-Q/MELSEC-L kezdő tanfolyam

(3) Előzetes felkészülés ..... MELSEC-Q/MELSEC-L kezdő tanfolyam

(4) Telepítés és huzalozás ..... MELSEC-Q/MELSEC-L kezdő tanfolyam

(5) Huzalozás ellenőrzése ..... MELSEC-Q/MELSEC-L kezdő tanfolyam

### Szoftvertervezés

(6) Programtervezés ..... 2. fejezet

(7) Programozás ..... 3. fejezet

(8) Hibakeresés ..... 4. fejezet

(9) Üzemeltetés

**A tanfolyam  
témakörei**

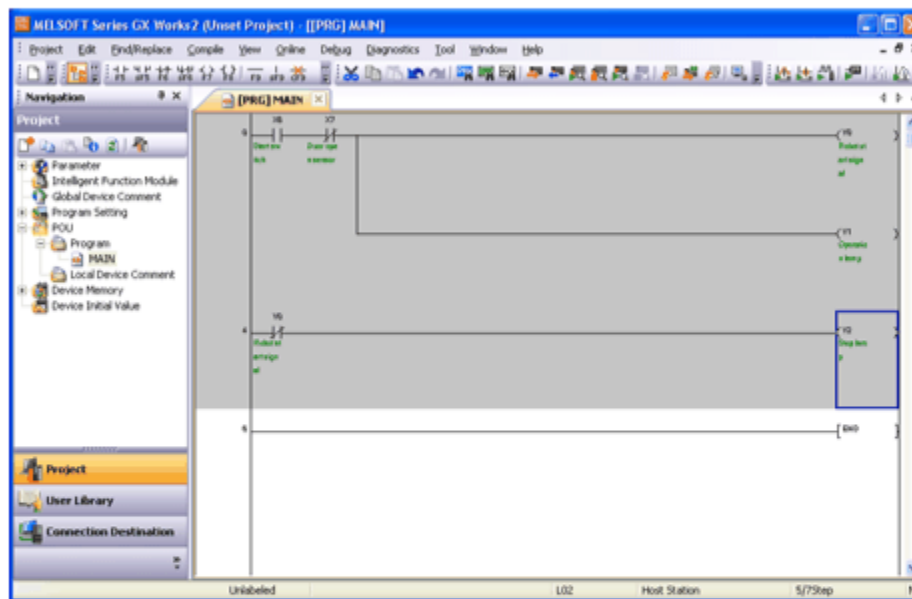
## 1.2

## A programozáshoz szükséges feltételek

A tanfolyam arra koncentrál, hogy a programozható vezérlő GX Works2 tervezőszoftverével hogyan építhető fel a példában szereplő rendszerprogram.

Az alábbi listán a GX Works2 néhány fontosabb funkciója látható.

- Memória- és fájlkezelés
- Programírás programozható vezérlőkhöz
- Programdokumentáció kezelése (megjegyzések stb.)
- Adatok (főként programok) olvasása/írása a CPU-modulból/modulba
- Programműveletek megerősítése
  - PLC-hardverek szoftveres szimulációja
  - I/O (be- és kimenet) kényszerített be- vagy kikapcsolása
  - I/O- és memóriacím állapotának figyelése
- Karbantartási és hibaelhárítási feladatok elvégzése





## 1.3

## A GX Works2 képernyő-konfiguráció

Az alábbiakban a GX Works2 képernyő-konfiguráció látható.

A kurzort egy piros keret fölé mozgatta megjelenik a megfelelő funkció.

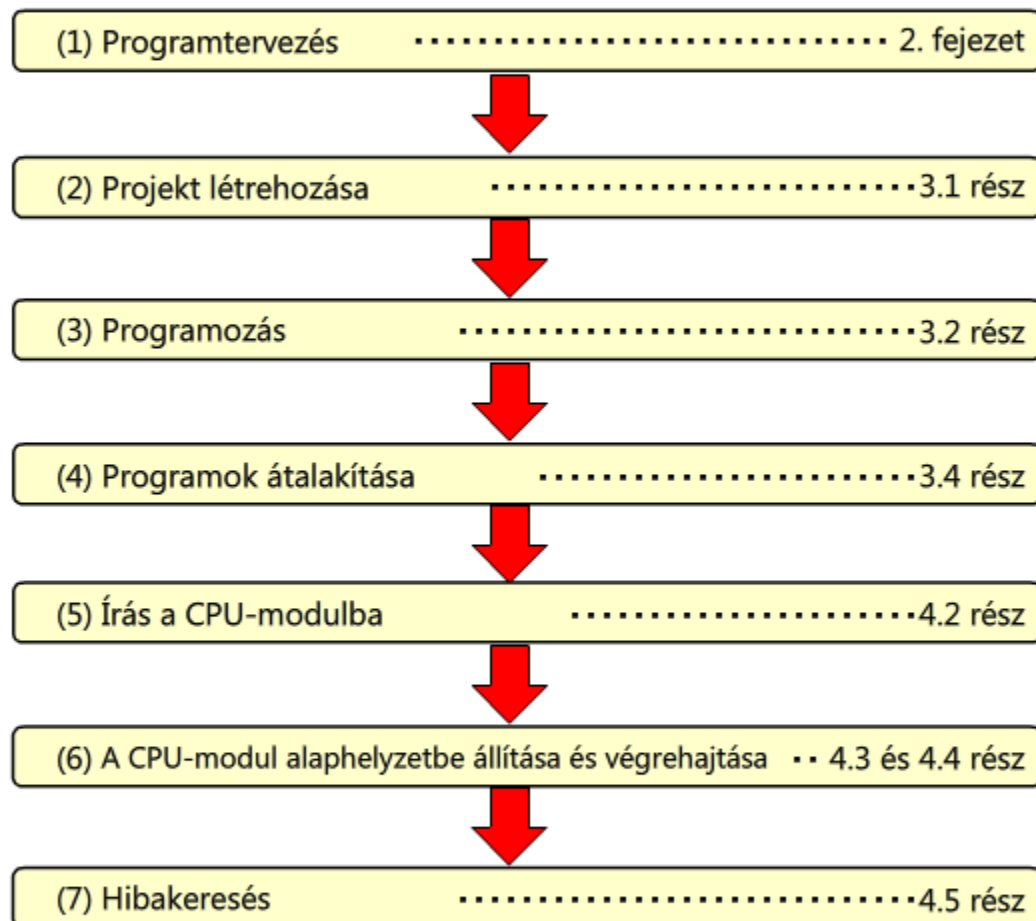
The screenshot displays the MELSOFT Series GX Works2 software interface. The main window shows a ladder logic diagram for a program named [PRG] SUB. The diagram includes the following components:

- Inputs:** X6 (Start Sw. lch) and X7 (Door open sensor).
- Outputs:** Y0 (Robot start signal), Y1 (Operation lamp), and Y2 (Stop lamp).
- Logic:** A normally open contact for X6 and a normally closed contact for X7 are connected in series to output Y0. A normally open contact for Y0 is connected to output Y2.

A red rectangular box highlights the [PRG] MAIN program in the project tree on the left and the corresponding [PRG] MAIN window in the center. The status bar at the bottom indicates the current step is 5/7 Step.

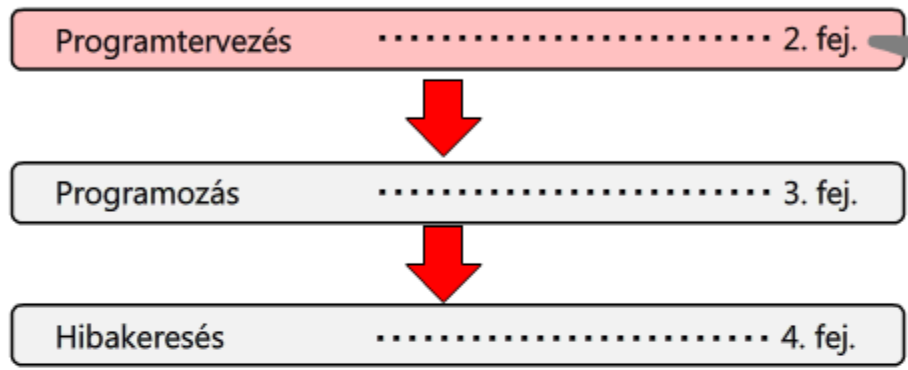
**1.4****Szekvencia program létrehozása**

Az alábbi lépéseket követve hozzon létre egy szekvencia programot.



# 2. fejezet Programtervezés

A 2. fejezetben a programtervezéssel együtt megtanulja a vezérlés részeinek definiálását és ezek programmá konvertálását is.



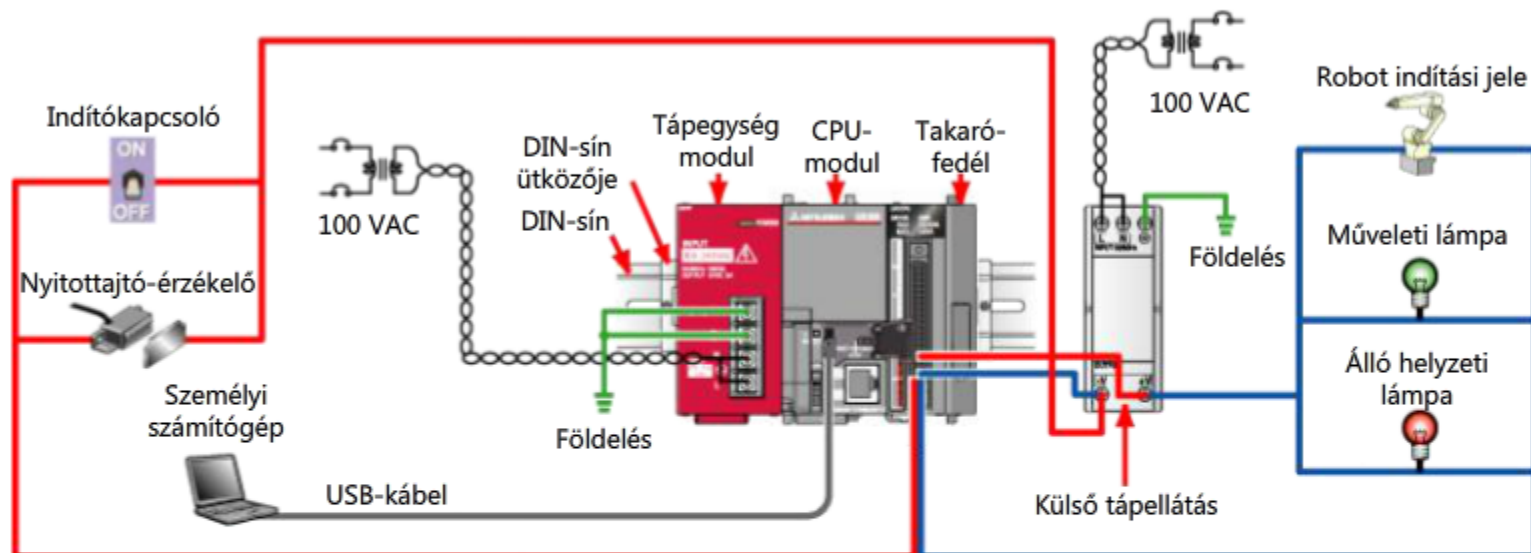
**A 2. fejezet tanulási lépései**

- 2.1 A tanuláshoz használt példarendszer hardverkonfigurációja
- 2.2. Vezérlőelemek definiálása
- 2.3 Az I/O-eszközök és számozásuk megfeleltetési táblázatának létrehozása
- 2.4 Egy program megtervezése

## 2.1 A tanuláshoz használt példarendszer hardverkonfigurációja

Ebben az anyagban létre fog hozni egy PLC-rendszert (a továbbiakban példarendszer), amely a robotot az eljárásnak megfelelően indítja.

Az alábbi ábra a példarendszer hardverkonfigurációját mutatja be a hardverelemek listájával kiegészítve.



Elem	Komponens	Modell	Leírás
PLC-rendszer	Tápegység modul	L61P	Árammal látja el a modulokat, beleértve a CPU- és az I/O-modulokat is.
	CPU-modul	L02CPU	Vezérli a PLC-rendszert.
	Takarófedél	L6EC	A rendszerblokk jobb oldalához van rögzítve.
	USB-kábel	MR-J3USBCBL3M	A GX Works2 szoftverrel rendelkező személyi számítógépet csatlakoztatja a CPU-modulhoz.
	Személyi számítógép	-	Telepített GX Works2 szoftverrel fut.
Külső tápellátás	-	-	Árammal látja el a külső I/O-eszközöket.
Külső I/O-berendezés	Kapcsoló	-	Kapcsolja be a vezérlés indításához.
	Szenzor	-	Észleli, hogy az ajtó nyitva vagy zárva van-e.
	Robot	-	A vezérlőjeleknek megfelelően működik.
	Két lámpa	-	A művelet állapotának megfelelően világítanak.

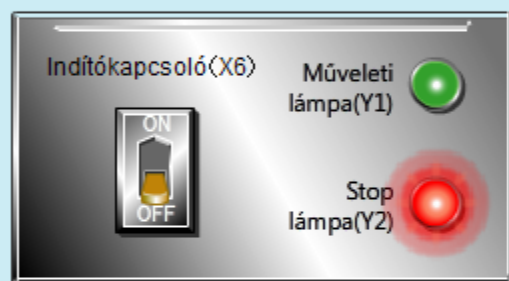
Egy program megtervezésének első lépése a vezérlendő eszközök és a kívánt vezérléshez szükséges I/O-eszközök definiálása. A példarendszerben egy robot működésének elindítási és leállítási vezérlése történik. A robot nem tud elindulni, ha a biztonsági kerítés ajtaja nyitva van, illetve leáll, ha az ajtó művelet közben kinyílik.

A példarendszer működésének jobb megértése érdekében tekintse meg a lenti animációt!

### A példarendszer működése

 Kattintson a piros körbe

Robot vezérlőpultja



Robot biztonsági kerítés mögött



Ha kikapcsolja az **indításkapcsolót (X6)**, a **robot indítási jele (Y0)** kikapcsol, és a robot abbahagyja a működést. Ezzel párhuzamosan a vezérlőpulton a **műveleti lámpa (Y1)** kialszik, a **stop lámpa (Y2)** pedig világítani kezd.

Újra



Előző

## 2.3 Az I/O-eszközök és számozásuk megfeleltetési táblázatának létrehozása

Célszerű minden létrehozott programhoz egy táblázatot készíteni, mely tartalmazza az összes I/O-eszközt és a PLC-ben használt regisztert, valamint a rájuk vonatkozó információkat. Ez csökkenti a tervezési és programozási hibák elkövetésének esélyét, és a programozás hatékonyságának növelését szolgálja. Ha a rendszerhez már van megfeleltetési táblázat (pl. a hardvert konfiguráló személy már létrehozott egyet), akkor érdemes azt használni.

Az alábbi táblázat a tanfolyamon használt példarendszer megfeleltetési táblázata.

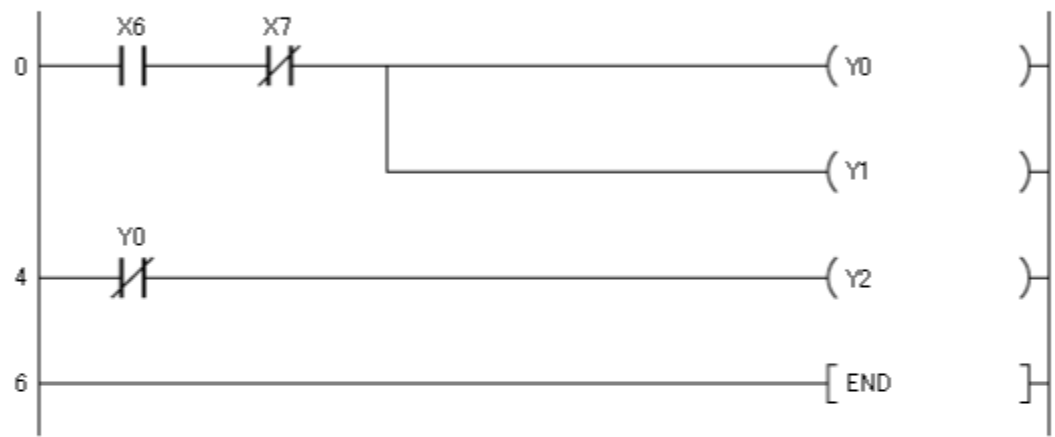
I/O-készüléknev	Készülékszám	I/O típusa	Készülék típusa	Leírás
Indítókapcsoló	X6	Bemenet	Bit	Ez a kapcsoló indítja el vagy állítja le a robot műveletét.
Nyitottajtó- érezékelő	X7	Bemenet	Bit	Ez a szenzor figyel, hogy a robot biztonsági kerítésének ajtaja nyitva van-e. Ha az ajtó kinyílik, a szenzor bekapcsol. Ha az ajtó becsukódik, a szenzor kikapcsol.
Robot indítási jele	Y0	Kimenet	Bit	A jel bekapcsolásával a robot elindítja a műveletet.
Műveleti lámpa	Y1	Kimenet	Bit	A lámpa a robot működése közben világít.
Álló helyzeti lámpa	Y2	Kimenet	Bit	A lámpa a robot álló állapotában világít.

- \* Szavas adat használata esetén adjon meg a táblázatban kezdeti értéket, beállítási tartományt (alsó és felső határ), adattípust (előjeles, valós stb.) és egy megjegyzést.  
Ez az információ hasznos lesz a programok tervezése és módosítása során.

## 2.4 Programok tervezése

A vezérlőelemek és az I/O megfeleltetési táblázat alapján tervezzen meg egy programot a létranyelv használatával. A létra program és a példarendszerhez készített I/O megfeleltetési táblázat lent látható.

### Létra program

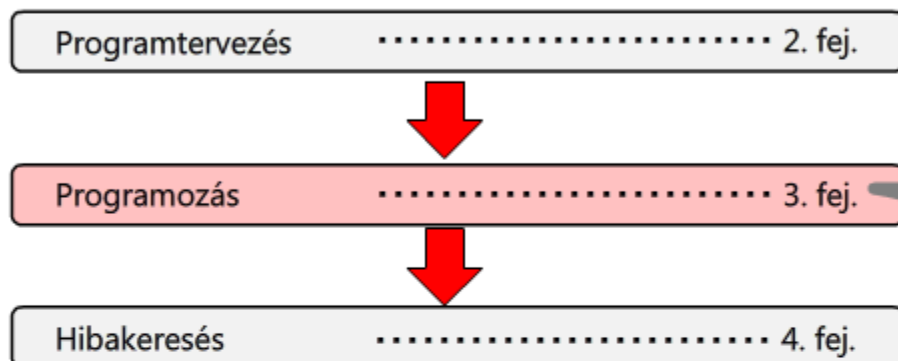


### I/O megfeleltetési táblázat

I/O-készüléknev	Típus	Készülékszám
Indítókapcsoló	Bemenet	X6
Nyitottajtó- érezkelő	Bemenet	X7
Robot indítási jele	Kimenet	Y0
Műveleti lámpa	Kimenet	Y1
Álló helyzeti lámpa	Kimenet	Y2

## 3. fejezet Programozás

A 3. fejezetben megtanulja, hogy a megtervezett programot hogyan programozza le a GX Works2 segítségével.



### A 3. fejezet tanulási lépései

- 3.1 Projektek létrehozása
- 3.2 Programok létrehozása
- 3.3 Átlátható programok készítése
- 3.4 Programok átalakítása végrehajtható formába
- 3.5 Projektek mentése



## 3.1 Projektek létrehozása

A programíráshoz vezető első lépés egy projekt létrehozása.

A projekt olyan adatok gyűjteménye, melyeket a GX Works2 a programok kezeléséhez használ.

Az alábbi lista a projektek főbb komponenseit tartalmazza.

Adattípus	Leírás
Program	A CPU szekvenciális műveleteinek forrás- és lefordított kódja.
Megjegyzés	A programon belül megjelenített egyfajta dokumentáció. Részletek a 3.3 „Átlátható program készítése” című részben.
Paraméter	A rendszer legtöbb vagy összes beállítási és konfigurációs információját tartalmazza.
Transfer setup (Átviteli beállítás)	A GX Works2 szoftvert futtató rendszer és a CPU-modul közötti kommunikáció létrehozásához szükséges kapcsolati útvonal információja.

### Létra program

A GX Works2 az alábbi két projekttypus kiválasztását teszi lehetővé.

A kurzus példaprogramja az „Egyszerű projekt” típust használja.

Projekttypus	Leírás
Egyszerű projekt	Ez a projekttypus kompatibilis a korábbi verziójú GX Developer projektekkel. Az egyszerű projektek később strukturált projekteké alakíthatók, ez visszafelé azonban már nem működik.
Strukturált projekt	Az ilyen projektekben megjelenik egy további programnyelv, a strukturált létra. Emellett a programok kisebb részekre bonthatók, a gyakran használt kódrészek pedig könnyedén modularizálhatók, és felhasználói könyvtár révén újrafelhasználhatók. A címkék a könnyű, ismételt felhasználás érdekében hasonló módon modulárisá tehetők. Ez javíthatja a programozás és a hibakeresés hatékonyságát, különösen nagy projektek esetében.

### Címkék

A címkék a felhasználó által kitalált nevek, az eszközcímkék alternatív nevei. Használhatók globálisan, lokálisan és a MELSOFT Navigator szoftverrel együtt, rendszerszinten is. Egyszerű projekteket címkék használatával és anélkül is létrehozhat. A példaprojektben nem használunk címkéket.

## 3.1 Projektek létrehozása

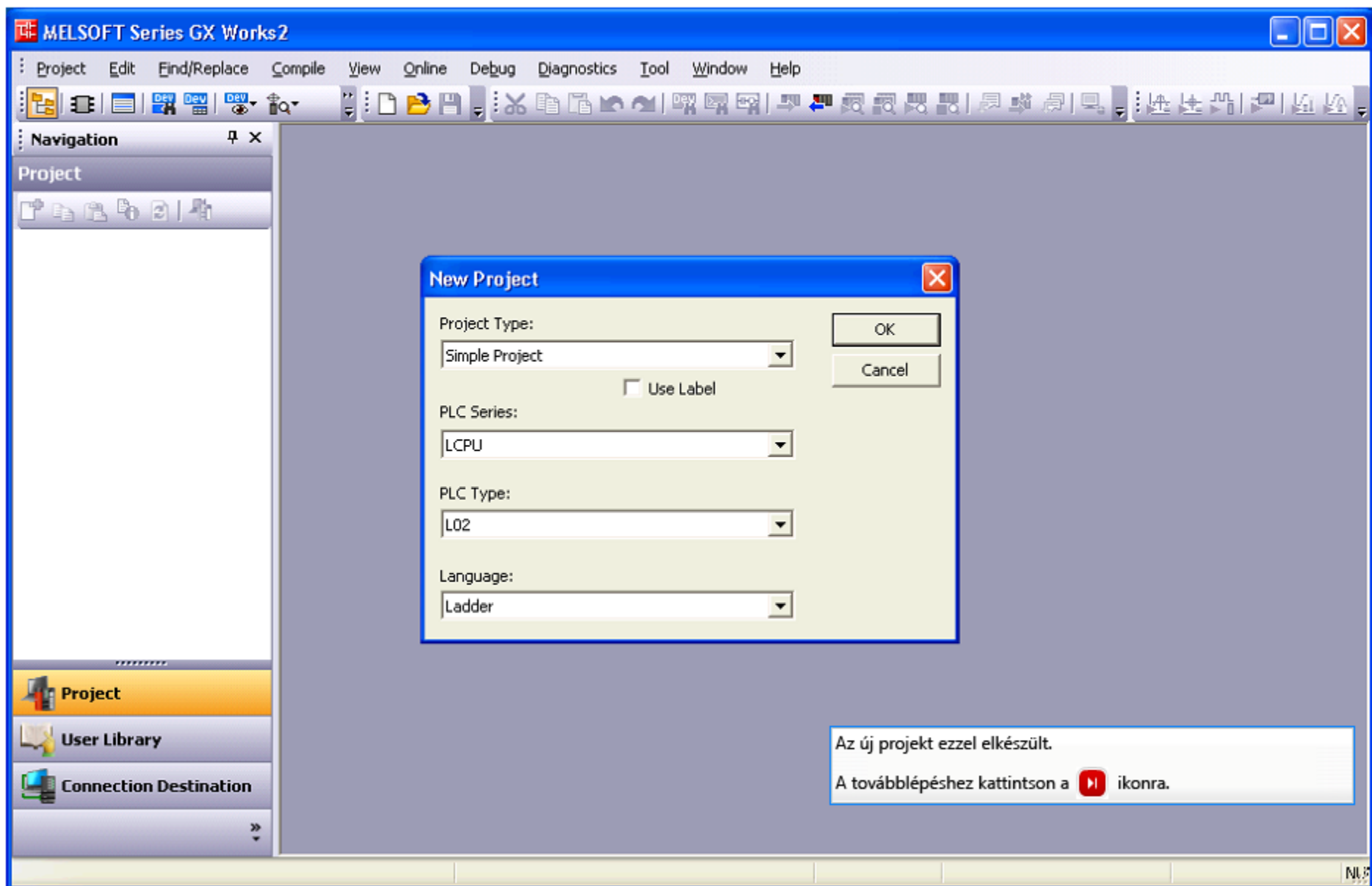
A példaprojekt létrehozásának elkezdéséhez az alábbi beállításokat kell elvégeznie. A projekt létrehozásához ismerni kell a programozható vezérlő sorozat- és típusnevét, valamint a használandó projekt típusát.

Elem	Leírás
Projekttypus	A projekttypus határozza meg, hogy a program írása során milyen lehetőségek állnak rendelkezésre. Ennél a példánál válassza az „egyszerű projekt” opciót.
Címkék használata	Amennyiben szüksége van a programírás során címkék használatára, jelölje be ezt az elemet. A példaprogram nem használ címkéket. Ezért ezt a jelölőnégyzetet hagyja üresen.
PLC-sorozat	A PLC-sorozat határozza meg, hogy a PLC-típusok legördülő listájában milyen típusok lesznek kiválaszthatók. Ennél a példánál válassza az „LCPU” opciót.
PLC típusa	A PLC típusa határozza meg, hogy a fordítóprogram hogyan alakítja át a felhasználói programot gépi kóddá. Válassza ki a programozni kívánt PLC típust, jelen esetben az „L02” lehetőséget.
Programnyelv	A programnyelv határozza meg az első automatikusan létrehozott program (MAIN) programtípusát. További, más nyelveket használó programok később hozzáadhatók. Ennél a példánál válassza a „Létra” opciót.

A következő oldalon áttekintheti az új projekt létrehozási műveletének szimulációját.

## 3.1

## Projektek létrehozása



MELSOFT Series GX Works2

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

Project

Project Type: Simple Project

Use Label


PLC Series: LCPU

PLC Type: L02

Language: Ladder

OK

Cancel

Az új projekt ezzel elkészült.  
A továbblépéshez kattintson a  ikonra.

## 3.2

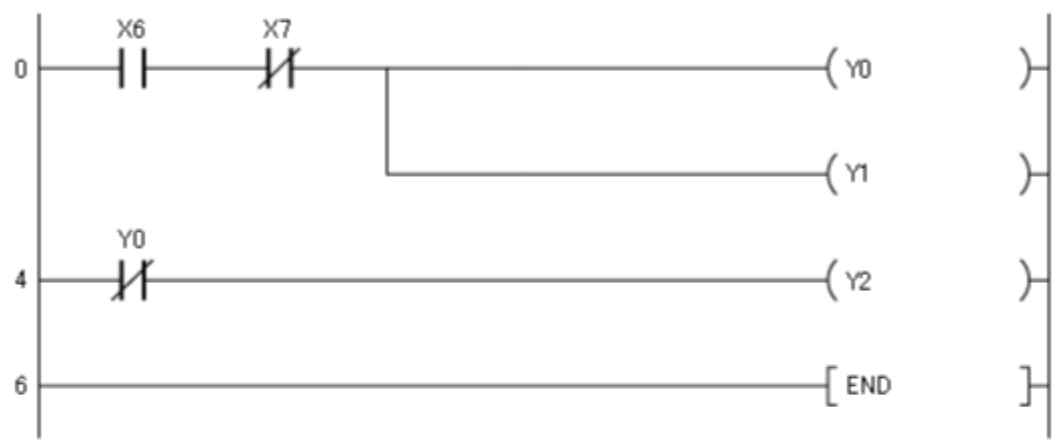
# Programok létrehozása

A projekt létrehozása után készítsen egy programot.

Az alábbi program elkészítése során megtanulhatja az alapvető műveleteket (utasítás megadása, változtatás, törlés, másolás és beillesztés, valamint vonal megadása/törlése).

A 2. fejezet példarendszeréhez tervezett program az alábbiakban látható.

### A példarendszer programja



A következő oldal szimulált ablakában próbálja meg elkészíteni ezt a programot.

# 3.2 Programok létrehozása



MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN


Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
    - Local Device Comment
- Device Memory
- Device Initial Value

Y0 Y1 Y2

0 [END]

Unlabeled L02 Host Station 0/15Step

A létradiagramos program ezzel elkészült.  
A továbblépéshez kattintson a  ikonra.

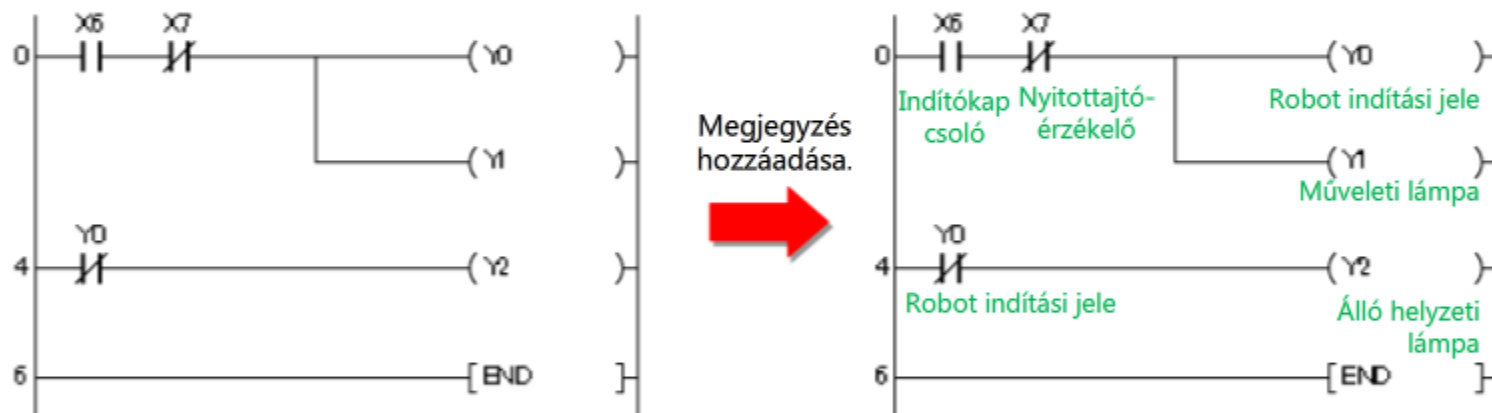
Jelenlegi állapotában a program képi megjelenítése mindössze eszközöket, utasításokat, vonalakat és lépésszámokat tartalmaz. Első ránézésre bonyolult megállapítani egy összetett program esetén, hogy milyen feladatot is lát el.

- Nehéz megtalálni a programozási hibákat, például egy helytelen készülékszámot vagy utasítást.
- Röviden: a műveletek elemzése, a hibakeresés és a program bővítése így bonyolult.
- Ha a program eredeti készítője nem tudja tovább karbantartani a programot, akkor a program működésének megértése egy kívülálló számára ijesztő, sőt akár lehetetlen feladat is lehet.

### Ellenintézkedések

Készítsen **dokumentációt** a programhoz, hogy bárki gyorsan megérthesse, hogyan is működik.

Ami a bevált gyakorlatot illeti, minden programozónak illene megjegyzéseket fűznie a programjához, hogy azt mind önmaga, mind mások számára könnyebben átláthatóvá tegye.



A GX Works2 három különböző típusú megjegyzést használ.

További részletek a GX Works2 Egyszerű projektek bemutató használati útmutatójában található.

Megjegyzés típusa	Megjegyzés helye
Eszközmegjegyzés	Max. 32 karakteres szöveg, amely a kiválasztott eszköz (I/O- vagy más memóriacím) alatt jelenik meg.
Leírás	Leírásonként max. 64 karakteres szöveg, amely a kiválasztott létrablokk tetején (a lépésszám felett) jelenik meg. Az egyes létrablokkokhoz több leírás is hozzárendelhető.
Jegyzet	Max. 32 karakteres szöveg, amely a kiválasztott tekercs vagy más alkalmazási utasítás felett jelenik meg.

A következő oldal a példaprogramhoz való eszközmegjegyzések hozzáadásának műveletét mutatja be.

## 3.3

## Átlátható programok készítése



MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
  - Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled


L02

Host Station

5/75Step

NL

Diagram description: The diagram shows a ladder logic program with three rungs. Rung 0 contains two normally open contacts labeled 'Start switch' (X6) and 'Door open sensor' (X7), connected to two outputs: 'Robot start signal' (Y0) and 'Operation lamp' (Y1). Rung 4 contains a normally closed contact labeled 'Robot start signal' (Y0) connected to an output labeled 'Stop lamp' (Y2). Rung 6 is a simple line connected to an 'END' terminal.

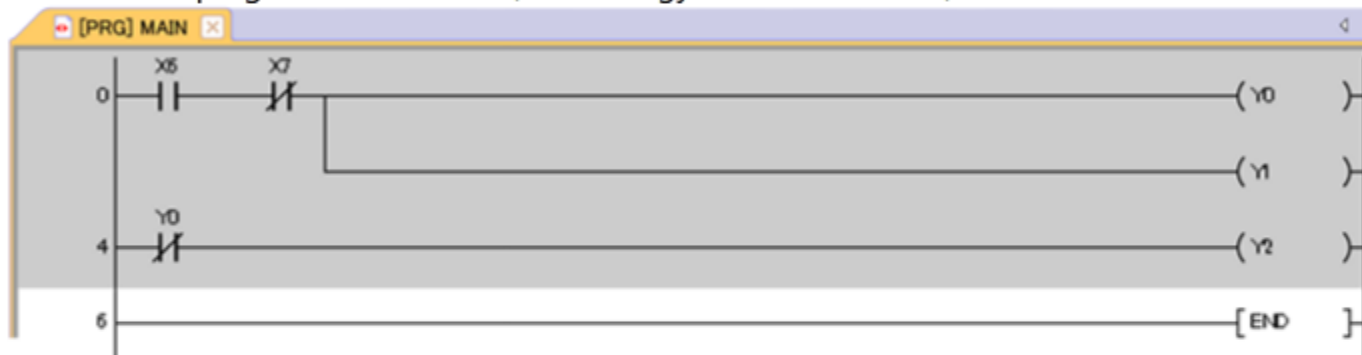
Az eszközmegjegyzések hozzáadása elkészült.  
A továbblépéshez kattintson a  ikonra.

## 3.4

## Programok átalakítása végrehajtható formába

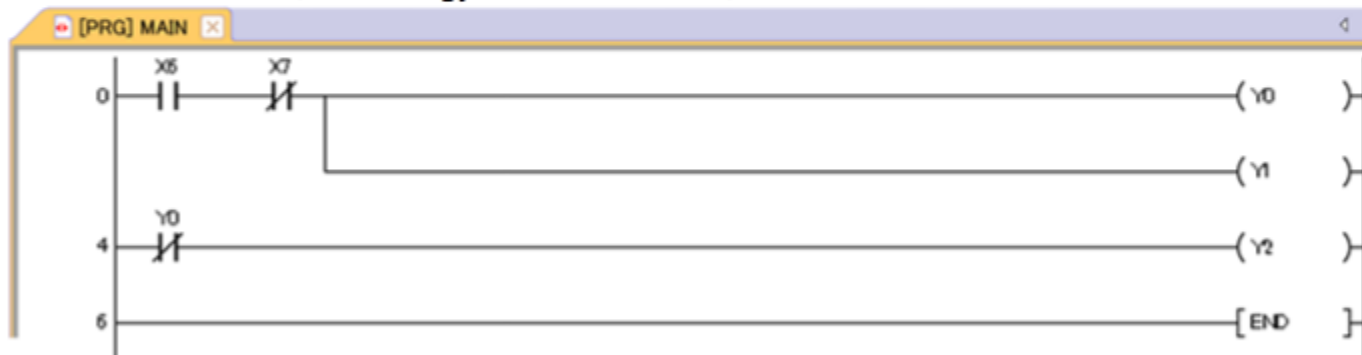
A programot az elkészítése után át kell alakítani olyan formába, amely a CPU-modulban végrehajtható. Az átalakítatlan programokat nem lehet végrehajtani vagy menteni.

Az átalakítatlan programok háttérszíne, mint ahogy az ábrán is látszik, szürke.



Átalakítás

Az átalakítást követően, mint ahogy az ábrán is látszik, a háttérszín fehérre változik.



A következő oldal szimulált ablakában kipróbálhatja, hogy a program hogyan átalakítható át.



## 3.4

## Programok átalakítása végrehajtható formába



MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
  - Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled


0 X6 X7 Y0 Y1 Y2 [END]

Start switch Door open sensor Robot start signal Operation lamp

4 Y0 Robot start signal Stop lamp

6 [END]

Ha a program átalakítása elkészült, a háttérszín szürkéről fehérre vált.

Program átalakítva.  
A továbblépéshez kattintson a  ikonra.

Unlabeled L02 Host Station 5/75Step

A program átalakítása után mentse el a projektet a programokkal együtt. Ha a GX Works2 a projekt elmentése nélkül bezárul, a társított programok elvesznek, ezért célszerű gyakran menteni a projektet.

Egy új projekt elmentésekor az alábbi típusú projektinformációkat kell megadni. (Felülírásos mentés esetén ez nem szükséges.) Adjon meg olyan információkat, melyek megkönnyítik mások számára a program vezérlési részeinek, a rendszer nevének stb. megértését.

Elem	Szükséges	Leírás
Mentési célhely útvonala	✓	Határozza meg a mappát, amelyhez a munkahelyet hozzá szeretné rendelni.
Munkahely/projektlista		Ha már létezik egy vagy több munkahely a „Mentési célhely útvonala” alatt megadott mappában, a létező munkahelyek kilistázódnak.
Munkahelynév	✓	Adjon meg egy maximum 128 karakterből álló munkahelynevet.
Projektnév	✓	Adjon meg egy maximum 128 karakterből álló projektnevet.
Cím		Adjon meg egy maximum 128 karakterből álló projektcímet. Ez a paraméter akkor hasznos, ha a megadni kívánt név nem fér bele a „Projektnév” keretébe.

A **munkahely** egy több projekt kezelésére szolgáló mappa.

Az alábbi táblázatban egy munkahely felhasználására lát példát. (A projekteket a személygépkocsi-gyártósornál gépjármű típusonként külön kezelik.)

Munkahelynév	Projektnév	Cím
Személygépkocsi-gyártósor	A típus gyártósora	Rendes műveleti program az A típus gyártósorának vezérléséhez
	B típus gyártósora	Rendes műveleti program a B típus gyártósorának vezérléséhez
	C típus gyártósora	Rendes műveleti program a C típus gyártósorának vezérléséhez

### Megjegyzés:

- Egy átalakítatlan programot tartalmazó projekt elmentésekor csak az átalakítatlan program vész el. A projekt elmentése előtt végezze el a program átalakítását a 3.4-es részben tanultak szerint.
- A mentési célhely útvonalát, a munkahelynevet és a projektnevet úgy adja meg, hogy a karakterek teljes száma ne haladja meg a 150-et.

A következő oldal szimulált ablakában kipróbálhatja, hogy a projektet hogyan mentheti el.

# 3.5

# Projekt mentése



MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
  - Local Device Comment
- Device Memory
- Device Initial Value

0 X8 Start Switch Door open sensor X7 Y0 Robot start signal Y1 Operation lamp

4 Y0 Robot start signal Y2 Stop lamp

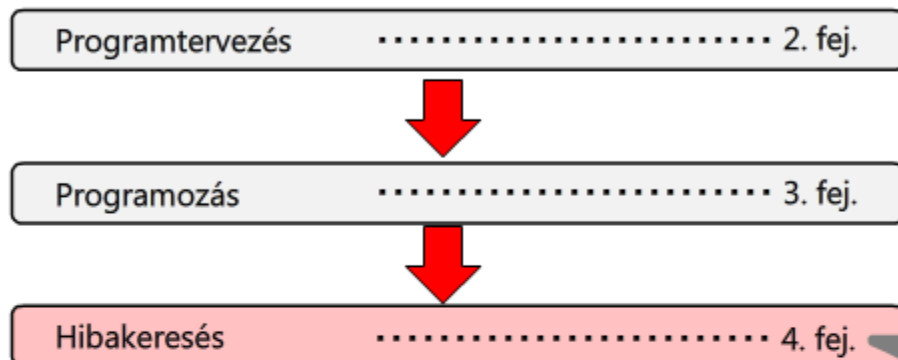
6 END

Project elmentve.  
A továbblépéshez kattintson a ikonra.

Unlabeled L02 Host Station 6/75Step

## 4. fejezet Hibakeresés

A 4. fejezetben megtanulja, hogy a CPU-modulhoz hogyan lehet szekvencia programokat írni, majd ezeken hibakeresést végezni.



### A 4. fejezet tanulási lépései

- 4.1 Hibakeresés (debuggolás)
  - 4.1.1 Hibakeresés a programban a CPU-modul használata nélkül
  - 4.1.2 I/O-eszköz állapotának megváltoztatása
  - 4.1.3 Eszköz állapotának felügyelete
- 4.2 Program írása a CPU-modulba
- 4.3 Áttöltött program engedélyezése
- 4.4 Program futtatása
- 4.5 Hibakeresés a programban
- 4.6 A PLC-rendszer műveletének ellenőrzése
- 4.7 A PLC-rendszer működtetése

Ha megírt egy programot vagy programszegmenst, akkor a kódot tesztelni kell annak megállapítására, hogy az az elvártak szerint működik-e.

A szaknyelvben a szoftverhibákat (amikor a megírt kód nem a célnak megfelelően működik) „bug”, a nem kívánt viselkedés okának megkeresését és kijavítását pedig „debuggolás” névvel illetik.

A tesztelés és a hibakeresés (debuggolás) a programok létrehozásának lényeges lépései.

Kifejezetten igaz ez a programozható vezérlők esetén, ahol ha hiba van a rendszerben, az leálláshoz vezethet, megrongálhatja a berendezést, vagy egyéb balesetet okozhat.

Az alábbi táblázat a GX Works2 néhány olyan funkcióját tartalmazza, amelyek a hibakeresési műveletben segíthetnek.

Funkció neve	Leírás
Simulator (szimulátor)	Ezt a funkciót a program végrehajtásának a CPU-modul nélküli szimulálására tervezték. Ez a funkció használható olyan környezetben való hibakeresésre, amelyben a CPU-modul nem érhető el.
Monitor (felügyelet)	Ez a funkció lehetővé teszi a végrehajtás és az egyes eszközök állapotának felügyeletét a CPU-modul végrehajtása alatt. Az alkalmazástól függően több felügyeleti funkció is elérhető, például létra felügyelete, csak regisztrált eszközök felügyelete, vagy minden eszköz csoportos felügyelete.
Change current value (aktuális érték megváltoztatása)	Ezzel a funkcióval kényszerített módon lehet megváltoztatni az eszköz állapotát (bit: BE ↔ KI, szó: aktuális érték) a CPU-modul végrehajtása közben. Ez a funkció egy szavas eszköz aktuális értékének vagy egy belső relé állapotának a megváltoztatására használható.
Forced input output registration/cancellation (bemenet/kimenet kényszerített regisztrálása/törlése)	Ezzel a funkcióval kényszerített módon lehet megváltoztatni egy regisztrált I/O-eszköz állapotát (BE ↔ KI) a CPU-modul végrehajtása közben. Hibakereséskor vagy csak a CPU-modullal végzett műveleti megerősítéskor ez a funkció egy kapcsoló helyettesítésére szolgál.

Ezeket a funkciókat a hibakeresés műveletével kapcsolatban a fejezet további részében részletesebben tárgyaljuk.

### Megjegyzések a hibakereséssel kapcsolatban

Ne futtasson hibakeresési műveletet, ha a programozható vezérlő egy fizikai I/O-eszközhöz van csatlakoztatva!

A programhibák, kényszerített I/O-eszközök vagy szóérték változások a külső berendezés károsodását vagy súlyosabb gondokat okozhatnak.

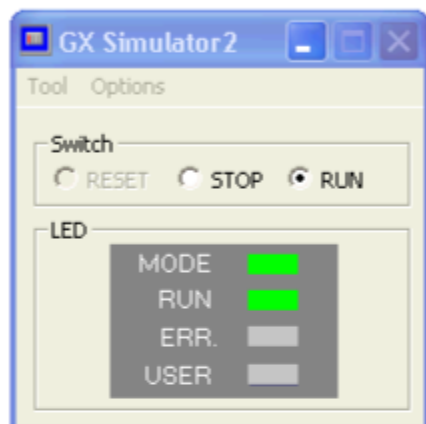
Ha egy leválasztott PLC-rendszer nem érhető el, használja a szimulátor funkciót.

## 4.1.1

## Hibakeresés a programban a CPU-modul használata nélkül

Ha egy CPU-modul nem érhető el a hibakereséshez, használja a **szimulátor funkciót**.

A program egy, a szoftver által biztosított virtuális CPU-modulon, a tényleges CPU-modul használata nélkül is futtatható.



Elem	Állapot	Leírás
Switch (Kapcsoló)	RUN (FUTTATÁS)	Futtatja a virtuális CPU-modult.
	STOP (ÁLLJ)	Leállítja a virtuális CPU-modult.
	RESET (VISSZAÁLLÍTÁS)	Visszaállítja alaphelyzetbe a virtuális CPU-modult. (Csak ÁLLÓ állapotban használható)
LED (LED)	MODE (ÜZEMMÓD)	A virtuális CPU ÜZEMMÓD állapotát jelzi.
	RUN (FUT)	A virtuális CPU futó állapotát jelzi. •Be: RUN (FUT) állapot •Ki: STOP (ÁLLÓ) állapot
	ERR (HIBA)	A virtuális CPU-modul hibaállapotát jelzi. Ha hiba van a rendszerben, a LED bekapcsol vagy villogni kezd.
	USER (FELHASZNÁLÓ)	Jelzi, ha a virtuális CPU-ban felhasználói hiba történt. Bekapcsol vagy villog, ha hiba történt.

### Megjegyzések a szimulátor funkció használatához

- A szimulátor funkció használatával történő hibakeresés nem garantálja, hogy a szekvencia program a javítást követően megfelelően fog működni.
- A szimulátor funkció szimulációs memória használatával I/O-modulok adatainak be-/kimenetét hajtja végre. A funkció bizonyos utasításokat, funkciókat és készülékmemóriát nem támogat. Így a szimulátor funkcióval elvégzett művelet végeredménye eltérhet a tényleges CPU-modullal végzett műveletétől.

A következő oldal szimulált ablakában kipróbálhatja a szimulátor funkció használatát.

# 4.1.1 Hibakeresés a programban a CPU-modul használata nélkül

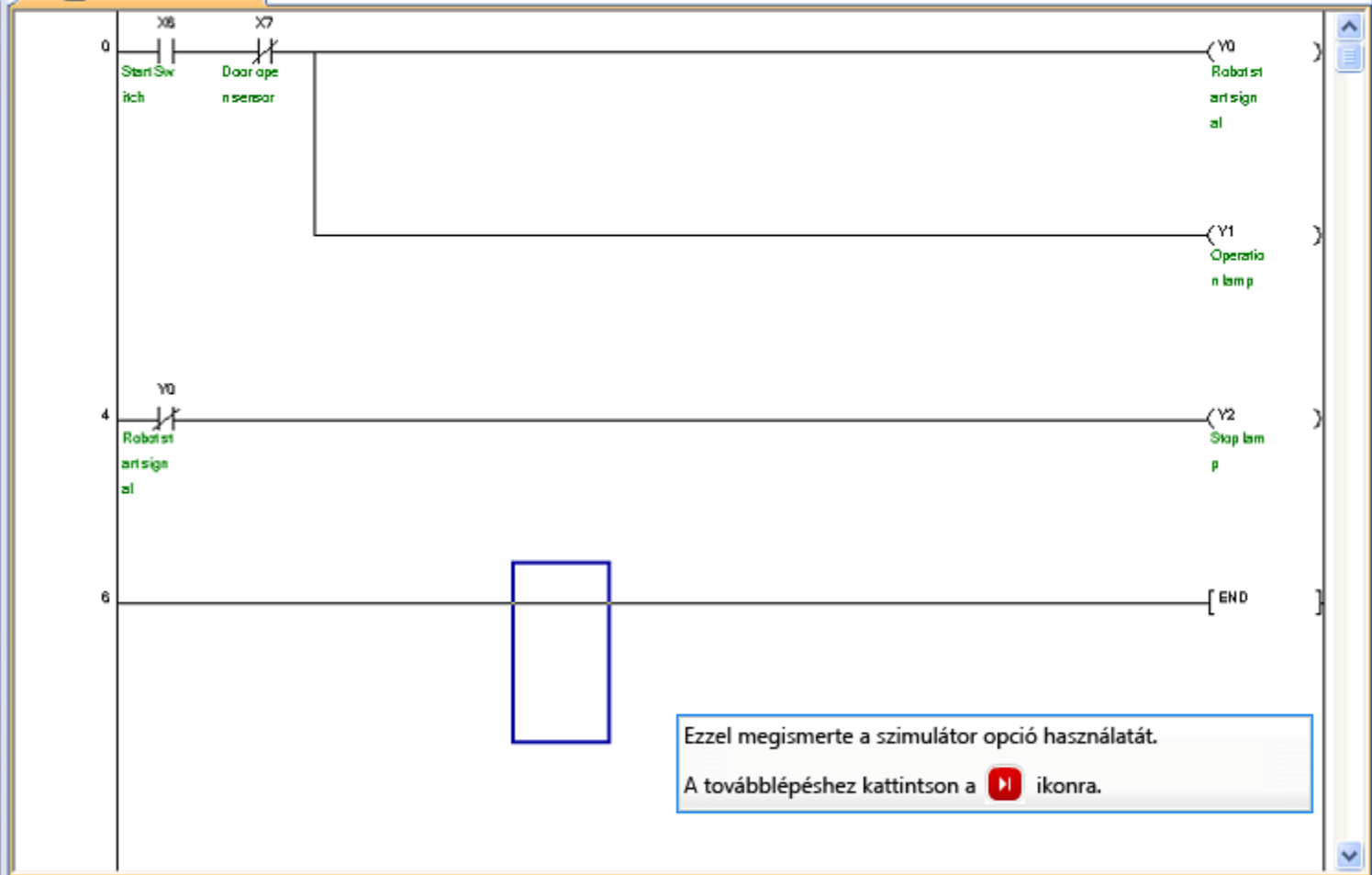
MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]


Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation [PRG] MAIN

- Project
  - Parameter
  - Intelligent Function Module
  - Global Device Comment
  - Program Setting
  - POU
    - Program
      - MAIN
    - Local Device Comment
  - Device Memory
  - Device Initial Value



Ezzel megismerte a szimulátor opció használatát.  
 A továbblépéshez kattintson a  ikonra.

## 4.1.2

## I/O-eszköz állapotának megváltoztatása

Szekvencia program hibakeresésénél ha a CPU-modulhoz nincs I/O-eszköz csatlakoztatva vagy simulator function (szimulátor funkcióban) van, egy I/O-eszköz BE/KI állapotának megváltoztatásához használja a **Forced Input Output Registration/Cancellation** (Bemenet/kimenet kényszerített regisztrálása/törlése) funkciót.

A szoftverrel egy regisztrált I/O eszköz állapota kényszerítve BE vagy KI állapotra változtatható.

(MELSEC-Q és MELSEC-L sorozatok): a „Forced Input Output Registration/Cancellation” (Bemenet/kimenet kényszerített regisztrálása/törlése) képernyőn

(MELSEC-F sorozat): a „Modify Value” (Érték módosítása) képernyőn



Forced Input Output Registration/Cancellation (Bemenet/kimenet kényszerített regisztrálása/törlése) képernyő (MELSEC-Q és MELSEC-L sorozatok)

### Egyéb eszközök állapotának megváltoztatása

Egy szavas eszköz aktuális értékének vagy egy belső relé BE/KI állapotának megváltoztatásához használja az **aktuális érték megváltoztatása** funkciót.

További részletek a használati útmutatóban.



Modify Value (Érték módosítása) képernyő (MELSEC-F sorozat)



## 4.1.3 Eszköz állapotának felügyelete

A szimuláció indításával a felügyelet automatikusan elindul. A felügyeleti mód eléréséhez – egy tényleges programvezérlő CPU-hoz csatlakozva – egyszerűen kattintson az Online (Online), a Monitor (Felügyelet), majd a Start Monitoring (Felügyelet indítása) menüpontokra. Vagy nyomja meg a billentyűzetten az F3 billentyűt.

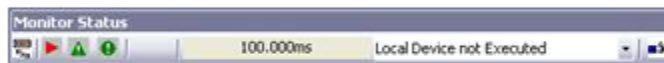
Felügyeleti módban a programban használt minden eszköz értéke és állapota látható lesz a programkód felett. Ez lehetővé teszi a felhasználó számára, hogy lássa az értékek változását, beleértve a „forced input output registration/cancellation (bemenet/kimenet kényszerített regisztrálása/törlése)” funkció használatának hatásait is.

Emellett megjelenik a **Monitor Status** (Felügyeleti állapot) sáv, amely alapvető információkat nyújt a CPU vagy a virtuális CPU állapotának meghatározásához. A **Monitor Status** (Felügyeleti állapot) sáv által nyújtott információkat az alábbi táblázat szemlélteti.

### A CPU-modulhoz csatlakoztatva



### A szimulátor funkció használatakor



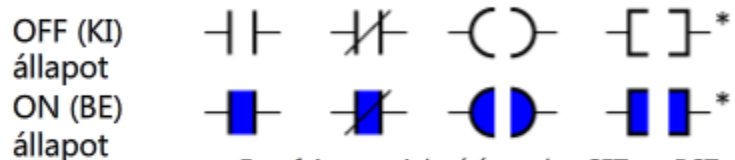
Állapot	Ikon/jel	Leírás
Kapcsolat állapota	CPU-modulhoz csatlakoztatva	Kijelzi a CPU-modullal vagy a szimulátor funkcióval való kapcsolat állapotát.
	A szimulátor funkció használatakor	
FUTTATÁS/ÁLLJ állapot	FUTTATÁS	Kijelzi a CPU futási állapotát (FUTTATÁS vagy ÁLLJ).
	ÁLLJ	
HIBA állapota	FELHASZNÁLÓI ki	Kijelzi a CPU-modul hibaállapotát.
	FELHASZNÁLÓI be	
	↔  HIBA villog	
FELHASZNÁLÓI állapot	FELHASZNÁLÓI ki	Kijelzi a CPU-modul felhasználói hibaállapotát.
	FELHASZNÁLÓI be	
	↔  FELHASZNÁLÓI villog	
Szkennelési idő	0.000ms	Kijelzi a felügyelt CPU-modul maximális szkennelési idejét.
Nem támogatott utasítás jelenléte/hiánya állapot	Nem támogatott utasítás van a rendszerben.	Kijelzi, hogy van-e nem támogatott utasítás a szimulátor funkció végrehajtásakor.
	Nincs nem támogatott utasítás a rendszerben.	Az ikonra való kattintással megnyílik az Unsupported Instruction/Device (Nem támogatott utasítás/Eszköz) ablak.

## 4.1.3 Eszköz állapotának felügyelete

Felügyeleti módban a program minden eszközének aktuális állapota láthatóvá válik.

### Biteszköz állapotának kijelzője (BE/KI)

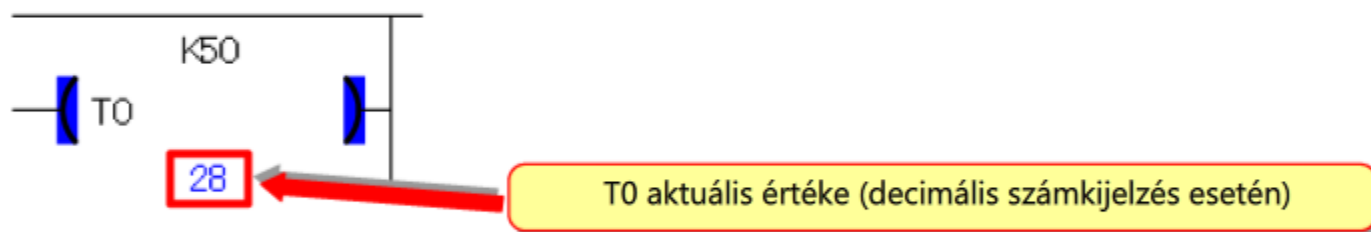
Felügyelet során az ON/OFF (BE/KI) állapot a lenti ábrán látható módon jelenik meg.



\* Ez a fajta megjelenítés csak a SET, az RST, a PLS, a PLF, az SFT, a SFTP, az MC (set, reset, fel- és lefutó élvezérlések, biteltolások, master vezérlés), valamint az érintkező típusú összehasonlító utasításokra vonatkozik. Ügyeljen rá, hogy az RST utasítás esetén csak a BE/KI állapot jelenik meg.

### Szavas eszközök (decimális/hexadecimális számkijelzés) aktuális értékének kijelzője

Felügyelet során az aktuális érték a lenti ábrán látható módon jelenik meg.



### Csak meghatározott eszközök felügyelése

Nagyon nagy vagy igen összetett program felügyeletkor hasznos lehet, ha csak a néhány érintett eszközt felügyeli. Ehhez a GX Works2 szoftver biztosít olyan figyelő ablakokat, melyek a felhasználó számára lehetővé teszik, hogy könnyedén kiválassza az érdekelt eszközt, megfigyelje annak aktuális állapotát, és módosítsa az értékét a felügyelet közben. További részletek a GX Works2 „Operation Manual (Common)” általános üzemeltetési kézikönyvében található.

Watch 1					
Device/Label	Current Value	Data Type	Class	Device	Comment
X7	-	Bit		X7	Door open sensor
Y0	-	Bit		Y0	Robot start signal
Y1	-	Bit		Y1	Operation lamp
Y0	-	Bit		Y0	Robot start signal
Y2	-	Bit		Y2	Stop lamp
Y0	-	Bit		Y0	Robot start signal

Mielőtt egy tényleges CPU-modul használatával hibakeresést végezne, állítsa a CPU-t **STOP (ÁLLÓ) üzemmódba**, ellenőrizze, hogy a CPU-val létrejött a kapcsolat, majd írja át a programokat és a paramétereket a program memóriába.

Amint az a lenti képen is látható, a **Write to PLC (Írás a PLC-re)** ablak főbb funkciói lehetővé teszik a felhasználó számára, hogy kiválassza az átírandó fájlokat és azok helyét, valamint hogy ellenőrizze a CPU memóriakapacitását. A fájllista feletti három gomb lehető teszi az átírandó fájl gyors kiválasztását. A legáltalánosabban használt, amelyet a következő szimulációban is használ majd, a „**Parameter+Program**” (Paraméter+program).

Fájllista

Memóriakapacitás

Module Name/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
Robot_Control						
PLC Data					Program Memory(D...)	
Program(Program File)		<input checked="" type="checkbox"/>				
Parameter		<input checked="" type="checkbox"/>		2010/08/05 15:22:24		2256 Bytes
PLC(Network/Remote Password/Switch Setting)		<input checked="" type="checkbox"/>		2010/08/05 15:22:24		2936 Bytes
Global Device Comment		<input type="checkbox"/>	Detail	2010/08/05 15:25:26		
COMMENT		<input type="checkbox"/>	Detail	2010/08/05 15:25:26		
Device Memory		<input type="checkbox"/>	Detail			
MADN		<input type="checkbox"/>		2010/08/05 15:22:25		

Writing Size: 5,192bytes / Free Volume: 76,728 / Use Volume: 5,192bytes

A következő oldal szimulált ablakában kipróbálhatja, hogy a CPU-modulba hogyan írhat.

## 4.2

## Program írása a CPU-modulba

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
    - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

Unlabeled L02 Host Station 6/75Step

0 Start Switch (X6) Door open sensor (X7) Robot start signal (Y0)

4 Robot start signal (Y0) Stop lamp (Y2)

6 END

Ezzel a program a PLC-modulba van írva.  
A további lépéshez kattintson a [play] ikonra.

## 4.3

## Áttöltött program engedélyezése

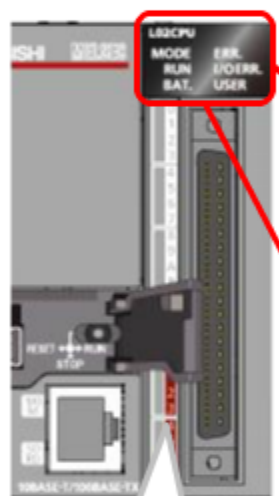
(MELSEC-F sorozat): Az alábbi művelet nem szükséges.

(MELSEC-Q és MELSEC-L sorozatok): Az alábbi művelet szükséges.

Miután átírt egy programot a CPU-modulba, **állítsa vissza** alaphelyzetbe a CPU-modult. Amíg a CPU-modul nincs alaphelyzetbe visszaállítva, az átírt programok nem engedélyezettek.

\* Erre a műveletre nincs szükség, ha hibakereséshez a simulator function (szimulátor funkciót) használja.

A CPU-modult az alábbiak alapján állítsa vissza:



RESET/STOP/RUN  
(VISSZAÁLLÍTÁS/  
ÁLLJ/FUTTATÁS)  
kapcsoló

- (1) Tartsa a CPU-modul előlő paneljén található RESET/STOP/RUN (VISSZAÁLLÍTÁS/ÁLLJ/FUTTATÁS) kapcsolót a RESET (VISSZAÁLLÍTÁS) pozícióban (1 másodpercig vagy tovább).  
[Visszaállítás folyamatban]

L02CPU		MODE (ÜZEMMÓD):
MODE ■	ERR.	Bekapcsolva zöld
RUN	I/OERR.	RUN (FUT): Ki
BAT.	USER	ERR. (HIBA) : Villog

Legalább 1 másodpercig  
tartsa lenyomva.



- (2) Amint a világító MODE (ÜZEMMÓD) LED és a villogó ERR. (HIBA) LED kialszik, engedje el a kapcsolót.  
[Visszaállítás kész]

L02CPU		MODE (ÜZEMMÓD):
MODE ■	ERR.	Bekapcsolva zöld
RUN	I/OERR.	RUN (FUT): Ki
BAT.	USER	ERR. (HIBA) : Ki

- (3) A visszaállítás befejezéseként a kapcsoló visszaáll a STOP (ÁLLJ) pozícióba.

### MELSEC-Q és MELSEC-L sorozatok

Ha a visszaállítás befejeződött, futtassa a programot.

A program futtatásához állítsa a CPU-modult **RUN (FUT) üzemmódba** az alábbiak szerint.

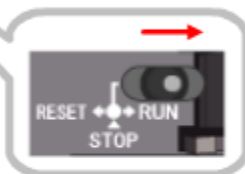
\* Erre a műveletre nincs szükség, ha hibakereséshez a szimulátor funkciót használja.



(1) Állítsa a CPU-modul előlő paneljén található RESET/STOP/RUN (VISSZAÁLLÍTÁS/ÁLLJ/FUTTATÁS) kapcsolót a RUN (FUTTATÁS) pozícióba.

A LED kijelző STOP (ÁLLJ) állapotban

L02CPU	MODE (ÜZEMMÓD): Bekapcsolva zöld
MODE	ERR.
RUN	I/OERR.
BAT.	USER



(2) Ha a RUN (FUT) LED zölden világít, a program rendesen fut.

A LED kijelző RUN (FUT) állapotban

L02CPU	MODE (ÜZEMMÓD): Bekapcsolva zöld
MODE	ERR.
RUN	I/OERR.
BAT.	USER

### MELSEC-F sorozat

A program futtatásához a programnak a főegységbe való átírása után kapcsolja a főegységet RUN (FUT) állapotba az alábbiak szerint. (A visszaállítási művelet nem szükséges.)

(1) Állítsa a főegység előlő paneljén található RUN/STOP (FUTTATÁS/ÁLLJ) kapcsolót a RUN (FUTTATÁS) pozícióba.



A LED kijelző STOP (ÁLLJ) állapotban

(2) Ha a RUN (FUT) LED világít, a program rendesen fut.

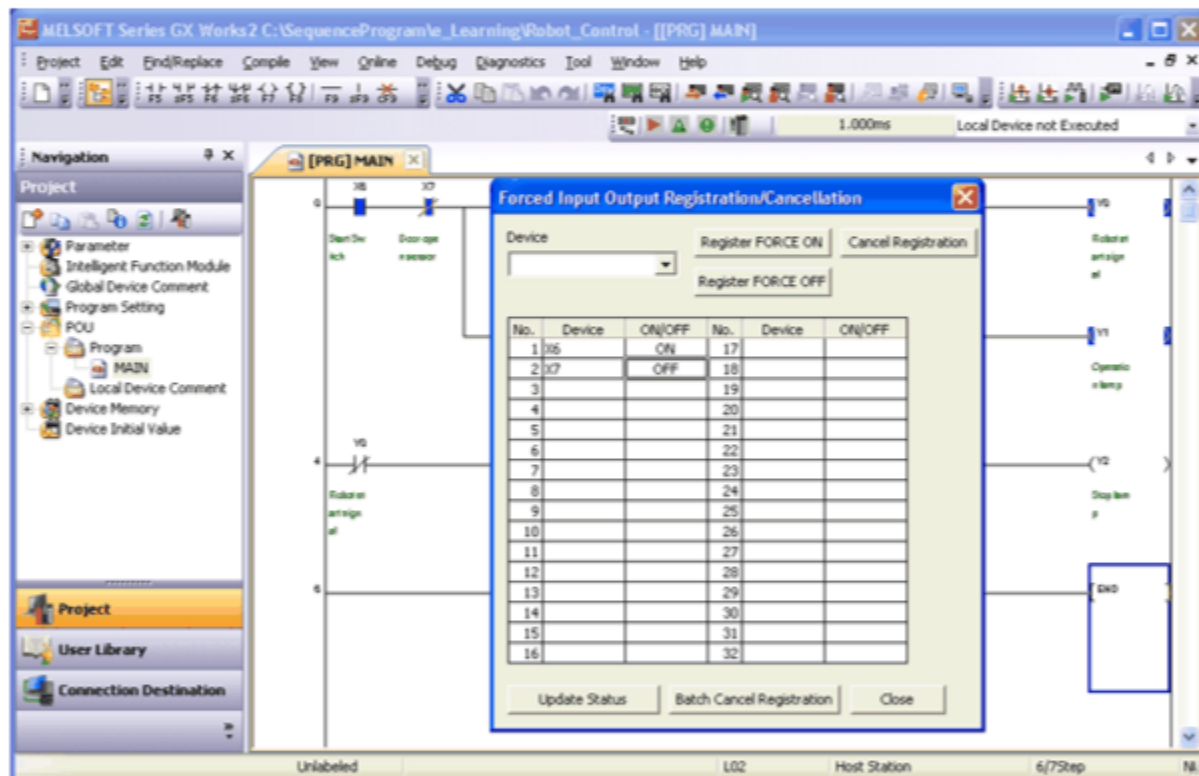


A LED kijelző RUN (FUT) állapotban

## 4.5 Hibakeresés a programban

A CPU futtatása után az egyes eszközök állapotának megváltoztatásához és az eredmény (kimenet) létrán való felügyeletéhez használja a forced input output registration/cancellation (bemenet/kimenet kényszerített regisztrálása/törlése) funkciót.

(Példa MELSEC-Q és MELSEC-L sorozatok képernyőiről)



A következő oldal szimulált ablakában kipróbálhatja, hogy hogyan végezzen hibakeresést a programban.

## 4.5

## Hibakeresés a programban



MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

1.000ms Local Device not Executed

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
  - Local Device Comment
- Device Memory
- Device Initial Value


Project

User Library

Connection Destination

Unlabeled L02 Host Station 6/75Step

Diagram description: The diagram shows a ladder logic program with three rungs. Rung 0 contains two normally open contacts labeled 'Start Switch' (X6) and 'Door open sensor' (X7), connected to two outputs: 'Robot start signal' (Y0) and 'Operation lamp' (Y1). Rung 4 contains a normally open contact labeled 'Robot start signal' (Y0) connected to an output labeled 'Stop lamp' (Y2). Rung 6 contains an 'END' instruction.

A program hibakeresése befejeződött.  
A továbblépéshez kattintson a  ikonra.



## 4.6

## A PLC-rendszer működésének ellenőrzése

A program hibakeresésének elvégzése után a művelet végső ellenőrzéséhez írja át a programot a tényleges PLC-rendszerbe.

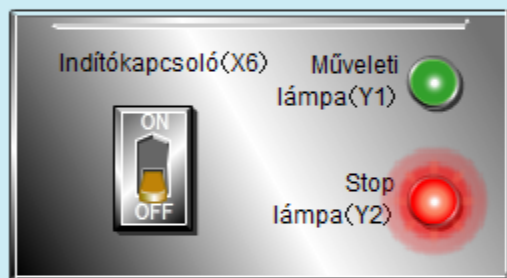
A tervezett működés ellenőrzéséhez működtesse a tényleges I/O-berendezést.

Az egyes eszközök állapota a GX Works2 felügyeleti funkciójával akár az I/O-berendezések működtetése közben is ellenőrizhető.

## A példarendszer működése

 Kattintson a piros körbe

Robot vezérlőpultja



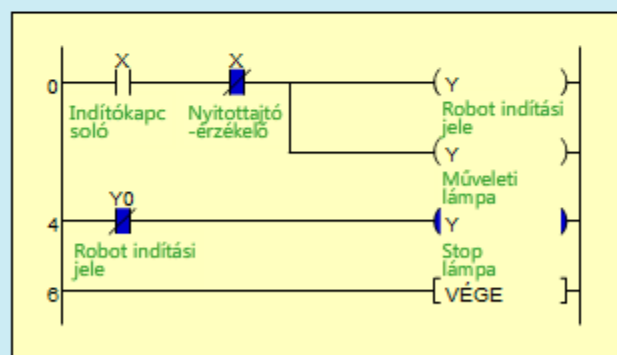
Robot biztonsági kerítés mögött



Ha kikapcsolja az **indítókapszolót (X6)**, a **robot indítási jele (Y0)** kikapcsol, és a robot abbahagyja a működést. Ezzel párhuzamosan a vezérlőpulton a **műveleti lámpa (Y1)** kialszik, a **stop lámpa (Y2)** pedig világítani kezd.

Újra

 Előző



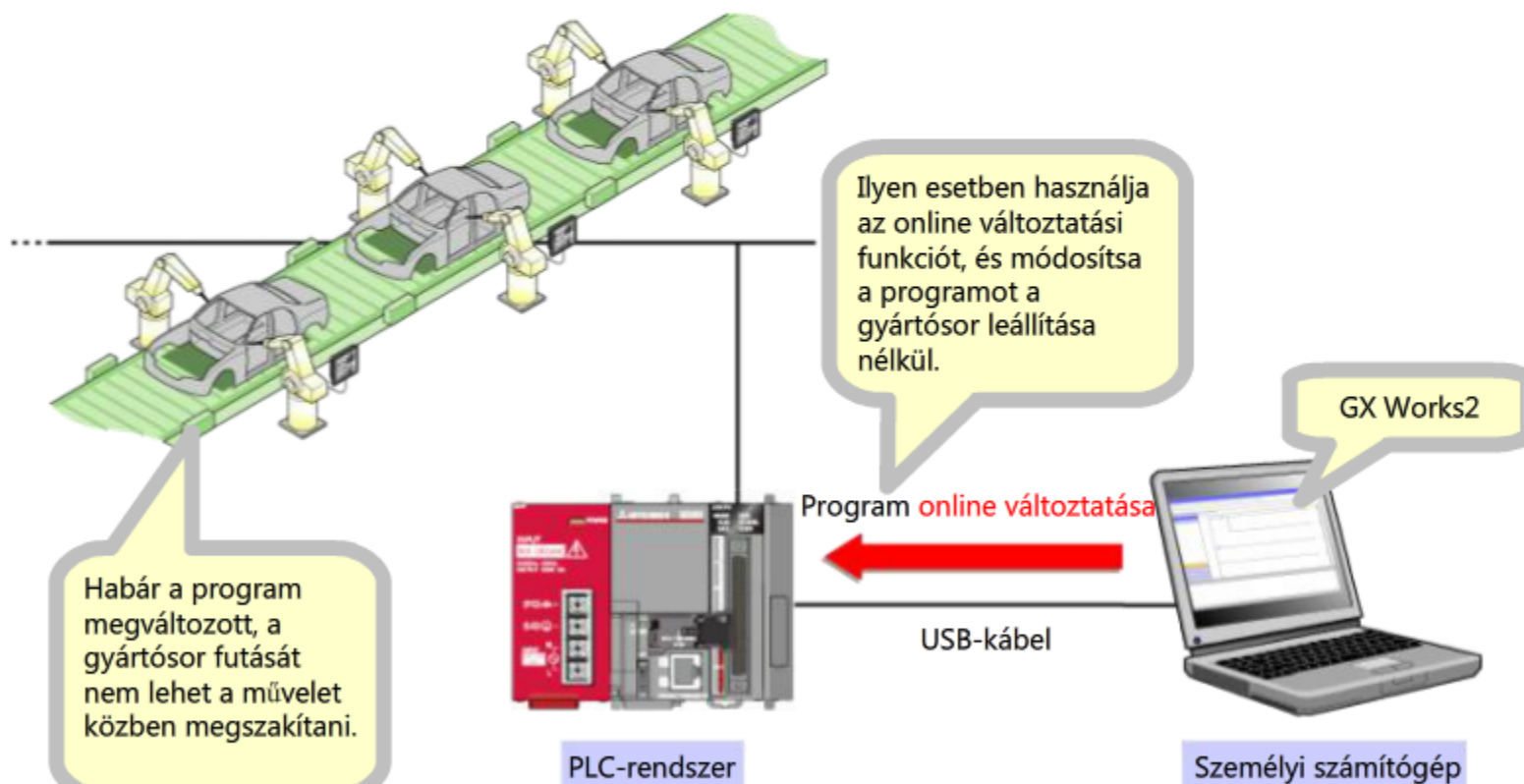
## 4.7 A PLC-rendszer működtetése

Ha a műveleti megerősítés elkészült, a művelet elindításához futtassa a PLC-rendszert.

### Ha a programot a rendszer futása közben kell módosítani

A rendszerművelet elindítása után is lehet szükség a program módosítására, például hibakeresés vagy rendszerbővítés esetén. Normál esetben a módosított program átírásához a rendszert (CPU-modult) le kell állítani, de ez nem mindig kivitelezhető. Ezen probléma megoldására a GX Works egy online változtatási funkciót biztosít, amellyel a program átírása a futó CPU-modul megállítása nélkül is lehetséges.

### Példa: 24 órás működésű személygépkocsi-gyártósor



A következő oldal szimulált ablakában kipróbálhatja az online változtatási funkciót.

## 4.7

## A PLC-rendszer működtetése



MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

1.000ms Local Device not Executed

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
  - Local Device Comment
- Device Memory
- Device Initial Value


Project

User Library

Connection Destination

Unlabeled L02 Host Station 2/75Step

Diagram description: The diagram shows a ladder logic program with three rungs. Rung 0 starts with a normally open contact labeled 'Start Switch' (X0) and a normally closed contact labeled 'Robot start signal' (X0). This rung is connected to two outputs: 'Robot start signal' (Y0) and 'Operation lamp' (Y1). Rung 4 starts with a normally open contact labeled 'Robot start signal' (Y0) and is connected to a normally open contact labeled 'Stop lamp' (Y2). Rung 6 is labeled '[END]'.

A módosított program online változtatása elkészült.  
A továbblépéshez kattintson a  ikonra.

## 4.8

## Összefoglalás



Ezzel a programozható vezérlők szoftvertervezését tárgyaló alapvető ismeretek végéhez értünk.

A kurzus alatt megtanulta:

- a PLC-rendszer programozásához szükséges elemeket,
- a programtervezés néhány alapvetőbb elvét, beleértve a megjegyzések használatát,
- a GX Works2 szoftver használatát a PC programozás alapvető feladatainak elvégzéséhez,
- a PLC programok hibakeresésének néhány megoldását.

Most, hogy elvégezte **A PLC GX Works2 alapjai** kurzust, készen áll a zárótesztre. Ha valami nem világos a témával kapcsolatban, használja ki a lehetőséget az ilyen témák áttekintésére.

**Ebben a zárótesztben összesen 5 kérdés (15 elem) található.**

A záró tesztet annyiszor végezheti el, ahányszor csak akarja.

### A teszt pontozása

A válasz kiválasztása után feltétlenül kattintson az **Válasz** gombra. A választ a rendszer nem rögzíti, ha az Válasz gombra való kattintás nélkül lép tovább. (A kérdés megválaszolatlanként lesz rögzítve.)

### Pontozási eredmények

A pontszám oldalon a helyes válaszok száma, a kérdések száma, a helyes válaszok százalékaránya és a teszt sikeres/sikertelen eredménye jelenik meg.

Helyes válaszok: 4

Összes kérdés: 4

Százalék: 100%

Továbblépés

Áttekintés

A teszt teljesítéséhez a válaszok  
**60%**-ának kell helyesnek lennie.

- Kattintson a **Továbblépés** gombra a tesztből való kilépéshez.
- A teszt áttekintéséhez kattintson a **Áttekintés** gombra. (Helyes válaszok megtekintése)
- Kattintson a **Újra** gombra a teszt újbóli megpróbálásához.

Az Önre bízott programot egy másik személy vette át, akinek gondjai akadtak a program vezérlőelemeinek megértésével. Mi a megfelelő intézkedés az ilyen probléma megelőzésére?

- Megfelelő cím és magyarázat hozzáfűzése a programhoz a GX Works2 megjegyzés funkciójának
- Szóban elmagyarázni az új személynek a vezérlőelemeket.
- Az összetett, nagy programok átvállalásának elkerülése.
- I/O-eszközök és számozásuk megfeleltetési táblázatának átadása a programmal együtt.

Egészítse ki a helyes programozási eljárást!

1. lépés Programtervezés

2. lépés ( K1  )

3. lépés ( K2  )

4. lépés Program átalakítása

5. lépés Projekt mentése

6. lépés ( K3  )

7. lépés ( K4  )

8. lépés CPU modul futtatása (FUTTATÁS)

9. lépés ( K5  )

10. lépés A PLC-rendszer működésének ellenőrzése

Válasz

Vissza

Töltse ki az üresen maradt részeket, hogy a program elkészítése utáni teendőkről szóló magyarázat teljes legyen.

Miután megírtak egy programot, le kell tesztelni, annak ellenőrzésére, hogy az elvártak szerint működik-e.

A (  ) (amikor a megírt kód nem a tervezettnak megfelelően működik) szaknyelvi megnevezése

(  ), a művelet pedig, amely során megkeressük az okot és kijavítjuk azt,

a (  ).

Ez a folyamat a programok létrehozásának egy lényeges lépése.



Társítsa az egyes GX Works funkciókhoz a megfelelő alkalmazást!

Funkció	Alkalmazás
Simulation (szimuláció)	--Select--
Forced input output registration/cancellation (bemenet/kimenet kényszerített regisztrálása/törlése)	--Select--
Change current value (aktuális érték megváltoztatása)	--Select--
Ladder monitor (létraprogram figyelése)	--Select--
Watch (figyelő)	--Select--

Válassza ki az online változtatási funkció megfelelő definícióját!

- A funkció automatikusan leállítja a CPU-t, átírja rá a programot, majd automatikusan elindítja a CPU-t.
- A funkció összehasonlítja a futó CPU-modulban lévő programot a GX Works2 által megnyitott programmal.
- A funkcióval a futó CPU-modul biztonságos leállítása után lehetőség van egy programot a CPU-modulba átírni.
- A funkcióval lehetséges egy program átírása a CPU-modulba annak leállítása nélkül.

Válasz

Vissza

**Teszt****Tesztpontszám**

Befejezte a záró tesztet. Az eredményei a következők.  
A záró teszt befejezéséhez lépjen a következő oldalra.

Helyes válaszok: **5**

Összes kérdés: **5**

Százalék: **100%**

[Továbblépés](#)[Áttekintés](#)

**Gratulálunk! A teszt sikerült.**

Ezzel elvégezte **A PLC GX Works2 alapjai** kurzust.

Köszönjük, hogy részt vett kurzuson.

Reméljük, élvezte a tananyagot, és a kurzuson szerzett információk  
hasznosak lesznek az Ön számára a jövőben.

A kurzust annyiszor tekintheti meg, ahányszor csak akarja.

**Áttekintés**

**Bezárás**