

PLC Programozás Portfólió

Tartalom

| | |
|---|---|
| Bevezetés..... | 2 |
| Fejlesztői Eszközök és Programozási Nyelvek..... | 2 |
| Egy példa a tanultakra egy sorompó vezérlése PLC-vel..... | 2 |
| Tapasztalatok..... | 3 |
| Borpalackozógép vezérlése PLC-vel..... | 4 |
| Fejlesztési folyamat és kihívások | 5 |
| Önreflexió..... | 7 |

Tantárgy neve: PLC Programozás

Készítette: Fekete Ádám

Projekt címe: Borpalackozógép vezérlése

Osztály: 12.b

Dátum: 2024.09.11.

Bevezetés

A PLC (Programmable Logic Controller) programozás egy rendkívül fontos terület az ipari automatizálásban. A PLC-ket elsősorban olyan rendszerek vezérlésére használják, amelyek automatikusan működnek, és gyakran igénylik a precíz időzítést és reakciókat. A tantárgy során megismerkedtem a PLC programozás alapjaival, különböző fejlesztőeszközökkel, mint az Easy Veep és a CODESYS, és különböző valós alkalmazásokban alkalmaztam tudásomat, mint például a sorompó vezérlés.

Fejlesztői Eszközök és Programozási Nyelvek

A PLC programozás során két fejlesztői eszközt használtam:

1. **Easy Veep:** Az Easy Veep különösen hasznos volt a kezdő projektekhez, mivel könnyen kezelhető felülettel rendelkezik, és gyors visszajelzést ad a program működéséről.
2. **CODESYS:** Ez egy egyszerű, felhasználóbarát PLC programozó szoftver, amely lehetővé teszi a programok gyors megírását és tesztelését. Ez egy komplexebb, ipari alkalmazásokhoz tervezett fejlesztői környezet, amely a PLC programozás széles spektrumát támogatja. A CODESYS segítségével képes voltam fejlettebb rendszerek programozására és hibakeresésére, és teljes mértékben kihasználhattam a különböző ipari szabványokat, mint például a strukturált szöveg (ST), ladder diagram (LD), illetve a funkcionális blokk diagram (FBD).

Egy példa a tanultakra egy sorompó vezérlése PLC-vel

A projekt során egy sorompó vezérlését kellett megvalósítanom PLC segítségével. A sorompó feladata a járművek áthaladásának szabályozása volt egy kijelölt területen. A rendszernek biztosítania kellett, hogy a sorompó megfelelően működjön a járművek érkezésének és távozásának figyelembevételével, és minden körülmények között biztonságos működést kell biztosítania.

A sorompó vezérléséhez az alábbi funkciókat kellett megvalósítanom:

1. Jármű érzékelése: Az érzékelők jelei alapján a PLC-nek meg kellett határozna, hogy jármű érkezik-e vagy sem.
2. Sorompó mozgatása: A PLC-nek vezérelnie kellett a sorompó fel- és leengedését a jármű érkezésének és távozásának megfelelően.
3. Biztonsági mechanizmusok: Biztosítani kellett, hogy a sorompó ne mozduljon, ha a jármű még a közelében van, valamint meg kellett oldani, hogy a sorompó ne mozduljon el hiba esetén.

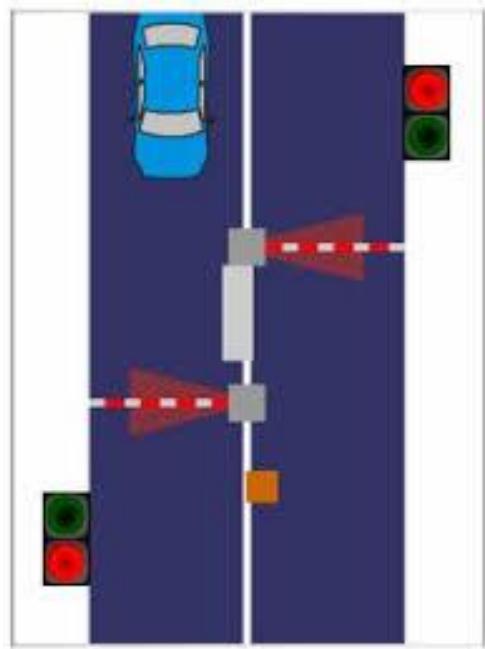
A projekt megvalósításakor a CODESYS fejlesztői környezetet használtam, és létradiagramot (LD) alkalmaztam a program logikai felépítéséhez. Az Easy Veep segítségével a kisebb tesztelési feladatokat végeztem el, mivel az gyors visszajelzést biztosított a rendszer működéséről.

Tapasztalatok

A PLC programozás tantárgy során megszerzett tudásom alapvetően az automatizált rendszerek tervezésére és programozására összpontosított. A következő területeken szereztem tapasztalatokat:

1. **Hardver és szoftver integráció:** Megtanultam, hogyan integráljam a PLC-t különböző érzékelőkkel, aktuátorokkal és más ipari berendezésekkel. A sorompó vezérlésénél például sikeresen összekapcsoltam a PLC-t a mozgó mechanikával és a fényjelző rendszerekkel.
2. **Logikai tervezés:** A programozás során fontos volt a logikai gondolkodás, hogy a rendszer minden körülmény között biztonságosan működjön. A sorompó vezérlésénél például figyelembe kellett venni a járművek mozgását, valamint az esetleges hibás működést is.
3. **Hibakeresés és optimalizálás:** A különböző szoftveres eszközök segítségével (pl. CODESYS) képes voltam tesztelni és hibákat keresni a programban, és szükség esetén optimalizálni a működést.

1. kép: Sorompó



forrás: Saját munka a CODESYS nevű programban

Borpalackozógép vezérlése PLC-vel

A tantárgy során az egyik legösszetettebb és legérdekesebb projekt egy borpalackozó gép vezérlésének megvalósítása volt. A feladat célja egy olyan rendszer létrehozása volt, amely automatizált módon irányítja a boros palackok töltését, kupakolását és továbbítását a gyártósorron. Ehhez több érzékelőt, működtető egységet és logikai kapcsolatot kellett összehangoltan működtetni.

A rendszer fő részei a következők voltak:

Palackérzékelő szenzorok: Ezek határozták meg, hogy mikor kerül egy új palack a töltőállomás alá.

Töltőszelep vezérlése: A PLC nyitotta és zárta a szelepet, a beállított töltési idő alapján.

Kupakoló egység: A töltés után automatikusan kupak került a palackra.

Szállítószalag mozgatása: A rendszer végén a megtöltött és lezárt palackokat egy motoros szalagrendszer továbbította.

A projekt során CODESYS környezetben dolgoztam, ahol strukturált szöveget (ST) és létradiagramot (LD) kombinálva programoztam a rendszert. Külön figyelmet kellett fordítani a sorrendiségre és az időzítések pontos kezelésére – például a palack töltése nem kezdődhetett el addig, amíg a szenzor nem jelezte a megfelelő pozíciót, és a szállítószalag csak akkor indulhatott újra, ha minden korábbi művelet befejeződött.

Fejlesztési folyamat és kihívások

A borpalackozó gép vezérlésének megtervezése és megvalósítása során az alábbi fő lépéseket végeztem el:

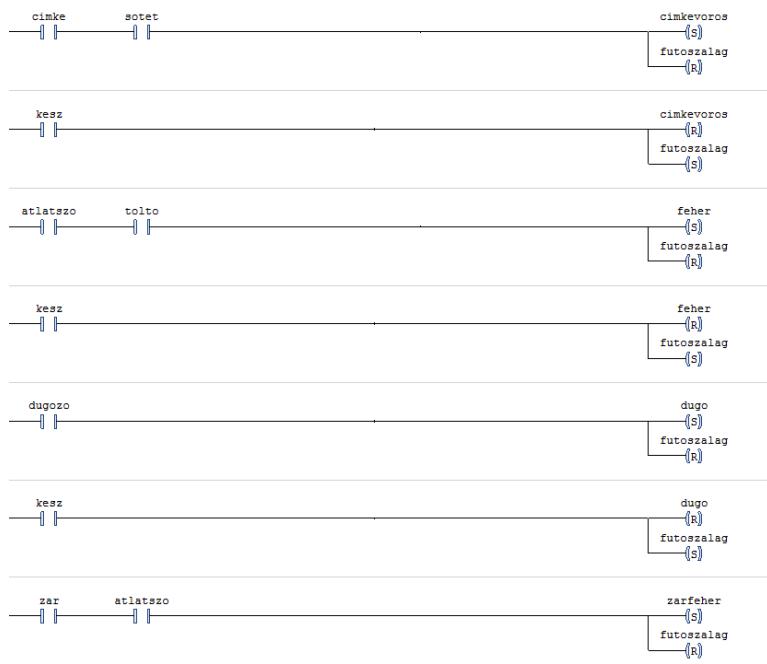
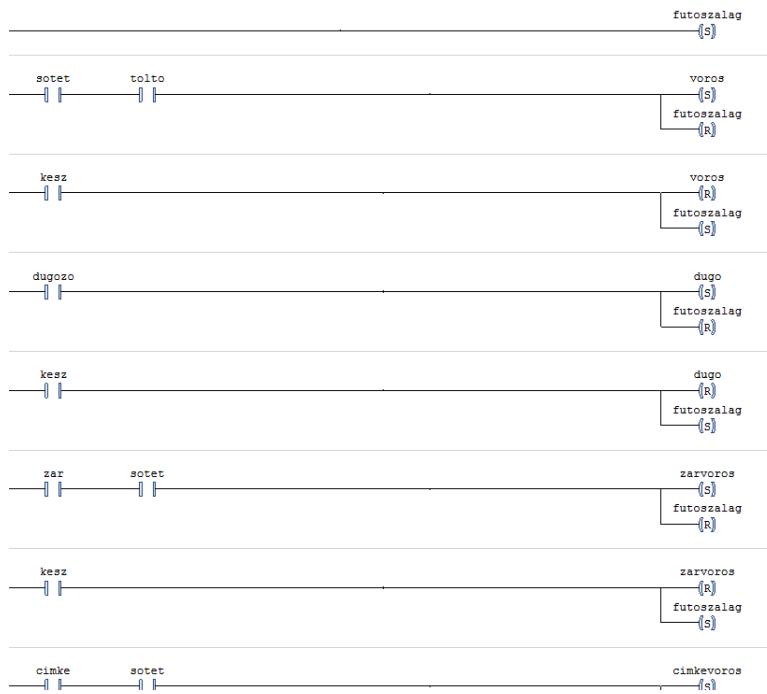
1. Rendszerterv készítése – meg kellett határozni az egyes funkciók logikai sorrendjét.
2. Szimuláció és tesztelés – a CODESYS fejlesztőkörnyezetben szimuláltam a rendszer működését a hibalehetőségek kiszűrése érdekében.
3. Hibakeresés – többször is optimalizálnom kellett a töltési ciklus időzítését, mivel eleinte a palackok nem töltődtek fel teljesen vagy túlcordultak.
4. Végső teszt – a kész programot élesben is kipróbáltam szimulációs környezetben, ahol a rendszer már megbízhatóan működött.

1. kép: Borpalackozó gép kódja

```
VAR_GLOBAL
    tolto AT %IX0.0: BOOL;
    dugozo AT %IX0.1: BOOL;
    zar AT %IX0.2: BOOL;
    cimke AT %IX0.3: BOOL;
    sotet AT %IX0.4: BOOL;
    atlatszo AT %IX0.5: BOOL;
    kesz AT %IX0.6: BOOL;

    voros AT %QX0.0:BOOL;
    feher AT %QX0.1:BOOL;
    dugo AT %QX0.2:BOOL;
    zarvoros AT %QX0.3:BOOL;
    zarfeker AT %QX0.4:BOOL;
    cimkevoros AT %QX0.5:BOOL;
    cimkefeher AT %QX0.6:BOOL;
    futoszalag AT %QX0.7:BOOL;
END_VAR
```

forrás: Saját munka a CODESYS nevű programban



Önreflexió

Ez a projekt lehetőséget adott arra, hogy a korábban megszerzett elméleti tudást egy összetett, gyakorlatban is releváns feladaton keresztül mélyítsem el. A borpalackozó gép vezérlése során nemcsak a PLC programozás technikai aspektusait gyakoroltam, hanem fejlesztettem a rendszertervezési és hibakeresési készségeimet is. A projekt sikeres lezárása megerősített abban, hogy képes vagyok összetett automatizált rendszerek önálló tervezésére és megvalósítására.