

Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Trefort Ágoston Mérnökpedagógiai Központ

DIPLOMAMUNKA

OE – TMPK

2022.

Berecz Norbert

T007716/FI12904/K



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Trefort Ágoston Mérnökpedagógiai Központ

DIPLOMAMUNKA FELADATLAP

Hallgató neve: Berecz Lajos Norbert
Diplomamunka száma: SZD22033010291198
Törzskönyvi száma: T007716/F112904/K

Neptun kódja: XXXXXXXXXX

Szak: tanári [4 félév [mérnök-tanár [elektrotechnika-elektronika]]]
Specializáció:

A dolgozat címe: PLC oktatás a szakképzésben

A dolgozat címe angolul: PLC education in vocational education and training

A feladat részletezése:

1. A PLC oktatás helye a programtervben
2. A tantervhez illeszthető oktatóanyag tervezése
3. A megtervezett oktatóanyag alkalmazása tantermi környezetben
4. Tapasztalatok összegzése

Intézményi konzulens neve: Dr. Holik Ildikó Katalin

A kiadott téma elévülési határideje: 2023. 05. 15.

Beadási határidő: 2022. 05. 15.

A diplomamunka: Nem titkos.

Kiadva: Budapest, 2022. 03. 31.



Intézetigazgató

A dolgozatot beadásra alkalmasnak találom:

belső konzulens

.....
külső konzulens



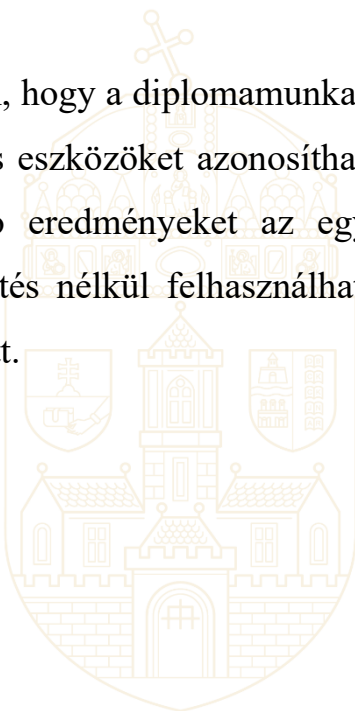
ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
TREFORT ÁGOSTON MÉRNÖKPEDAGÓGIAI KÖZPONT

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült diplomamunkában található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Budapest, 2022. május 22.



.....
hallgató aláírása

Tartalomjegyzék

Bevezetés	4
1. PLC oktatás helye a programtantervben	9
1.1. A magyar szakképzési rendszer	9
1.1.1. Története	9
1.1.2. A szakképzési rendszert szabályozó dokumentumok	11
1.1.3. Az Országos Képzési Jegyzék	13
1.1.4. Modulrendszerű vizsgarendszer	14
1.1.5. Komplex vizsgarendszer	15
1.1.6. Technikumi rendszer	16
2. A tantervhez illeszthető oktatóanyag tervezése	19
2.1. Oktatási formák	19
2.2. A virtuális környezet: szimuláció és emuláció	21
2.2.1. Szimuláció	21
2.2.2. Emuláció	23
2.3. Felhasznált programok bemutatása	25
2.3.1. Siemens TIA Portal	29
2.3.2. Factory I/O	33
2.3.3. Festo FluidSIM	33
2.4. A tervezett oktatóanyag	34
2.4.1. Feladatléírás	34
2.4.2. Feladat elkészítése	37
3. A tervezett oktatóanyag alkalmazása tanteremi környezetben	48

3.1. Fizikai oktatókörnyezet.....	48
3.1.1. Szaktantermek.....	48
3.2. Az új kihívás – online oktatókörnyezet.....	51
3.3. Oktatóanyag alkalmazása.....	54
4. Tapasztalatok összegzése.....	56
Összefoglalás	57
Summary.....	59
Irodalomjegyzék	61
Ábrajegyzék	62
Táblázatok.....	65
Mellékletek listája.....	66

„A tudományt nem újonnan teremti meg minden új nemzedék, a tudomány olyan, mint a folyton égő fáklya, amely nemzedékről nemzedékre száll. Világos, mi a kötelességünk azokkal szemben, akik átveszik tőlünk, s tovább viszik ezt a fáklyát: segíteniünk kell őket, de ne diktáljunk nekik, mert a dogmatizmus ártalmas, gátolja a tudomány fejlődését és még inkább a tudósok nevelését.”

/Nyikolaj Nyikolajevics Szemjonov/

Bevezetés

A programozható logika vezérlők a mára az ipar egyik létfontosságú fundamentális egységévé fejlődte ki magát, pár évtizedes története alapján. A munkaerőpiac egyik legértékesebb tagja az, aki ezen eszközök programozásához, üzemeltetéséhez ért. Még fontosabb, ha valaki a meglévő tudását, a jövő szakembereinek is tovább tudja adni.

Saját magam személyesen a PLC-k világával még Elektronikai technikus OKJ-s tanulmányaim során találkoztam, ahol a képzési programunk része volt, igaz csekély óraszámmal, de ez is elég volt ahhoz, hogy egy életre elköteleződjek mellette.

A diplomamunka célja egy alternatív módszertani lehetőség bemutatása, amivel a technikumi oktatás alappilléreinek mondható TEA – tanulás élmény alapú- oktatást tudjuk biztosítani.

Programozható logikai vezérlők

A programozható logikai vezérlő (PLC – Programmable Logic Controller), egy ipari számítógép, amit analóg vagy digitális jelekkel történő folyamatvezérlésre fejlesztettek ki, főképp üzemi és gyári környezetben való használatra. Eredeti célja az elektromechanikus vezérlések kiváltása volt az autóiparban. Napjainkban a PLC már gyakorlatilag minden iparágban megtalálható. (Juhász, é.n.)

A programozható logikai vezérlők története

A programozható logikai vezérlők története egészen az 1960 évekig nyúlik vissza. A programozható logikai vezérlők kifejlesztésének célja az volt, hogy egyszerűsítsük és kiváltsuk az akkoriban használt elektromechanikus vezérléseket.

1968 – A PLC - koncepció kidolgozása a General Motors felhívására

1969 – Az első Modicon PLC megjelenése huzalozott CPU-val 1K memóriával és 128 I/O-val

1971 – A PLC első alkalmazása az autóiparban

1973 – Az első intelligens (smart) PLC megjelenése aritmetikai funkcióval, nyomtatóvezérléssel, mátrixműveletekkel, képernyő-kijelzéssel

1974 – Az első többprocesszoros PLC gyártása időzítő- és számlálófunkcióval, 12K memóriával és 1024 I/O-val

1975 – Az első PID algoritmussal ellátott PLC kibocsátása

1976 – A távoli modulkezelés kidolgozása és a hierarchikus konfiguráció bevezetése az integrált gyártórendszerben

1977 – A mikroprocesszor bázisú PLC bevezetése

1980 – Intelligens kommunikációs modulok kifejlesztése, valamint a nagysebességű, nagy pontosságú pozicionáló interfész kifejlesztése

1981 – 16 bites mikroprocesszor bázisú PLC színes monitorral

1983 – Olcsó „mini” PLC-k megjelenése

1985 – PLC hálózatok kifejlesztése

(Pintér, 2010, old.: 5)

A programozható logikai vezérlő előnyei és hátrányai

Előnyei:

- a vezérlő kis mérete és helyigénye
- a vezérlőszekrény többi eleméhez formailag is illeszkedik
- lényegesen kevesebb a huzalozási munka
- programozása egyszerű, áttekinthető, a berendezés működése programmódosítással esetenként szerelés nélkül is változtatható
- könnyű üzembe helyezés és hibakeresés
- programírás és futtatás szakaszosan is végezhető, a program és a berendezés működése szimulációval tesztelhető
- rendkívül megbízható, hosszú élettartamú
- széles felhasználási terület

Hátránya:

- A vezérlő és a programozó szoftver viszonylag drága



1. ábra: Siemens S7-1500-as termékcsalád

Forrás: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500.html>

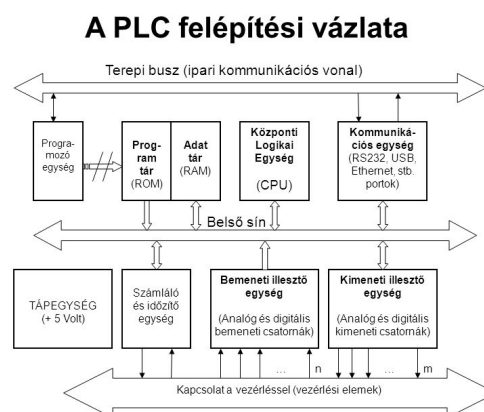
Programozható logikai vezérlők felépítése

A programozható logikai vezérlők az alábbi elemekből épülnek fel:

- Tápegység: Biztosítja a programozható logikai vezérlő energiaellátását. A tápfeszültség lehet AC (váltakozó áram) vagy DC (egyenáram).
- Központi vezérlő egység – Processzor – CPU: A programozható logikai vezérlő lelke, amely a folyamatok irányításáért felel. Kezeli a kommunikációt (buszrendszer), az I/O felületekkel való kapcsolatot, futtatja a rendszer és vezérlőprogramot, kapcsolatot létesít a különböző memóriaegységekkel.

Utasításkészletét tekintve két fajtája lehetséges:

- RISC (Reduced Instruction Set Computer) – csökkentett utasításkészletű processzor, ahol a vezérlést direkt áramköri elemek látják el.
- CISC (Complex Instruction Set Computer) – ebben az esetben a vezérlést mikroprogramok végzik a processzorban.
- I/O (input/output) modulok/portok: Ezek a modulok kezelik a bejövő és kimenő információkat, amelyek szükségesek a program végrehajtásához.
- Kommunikációs modulok: ezek segítik az információtovábbítást PLC-PLC vagy PLC-PC között, illetve átfordítanak, ha a kommunikációs protokollban váltás történik (RS232, RS485, Ipari Ethernet, ProfiBUS, Profinet).



2. ábra: PLC-k belső felépítése

Forrás: <https://slideplayer.hu/slide/2026056/> 3. dia

Kialakításukat tekintve a programozható logikai vezérlőket két nagy csoportba tudjuk foglalni:

- Kompakt PLC-k: Az összes szükséges modul, összetevő egy tokozásban van, amely tovább nem bővíthető.

Ilyenek például a Festo által gyártott PLC-k

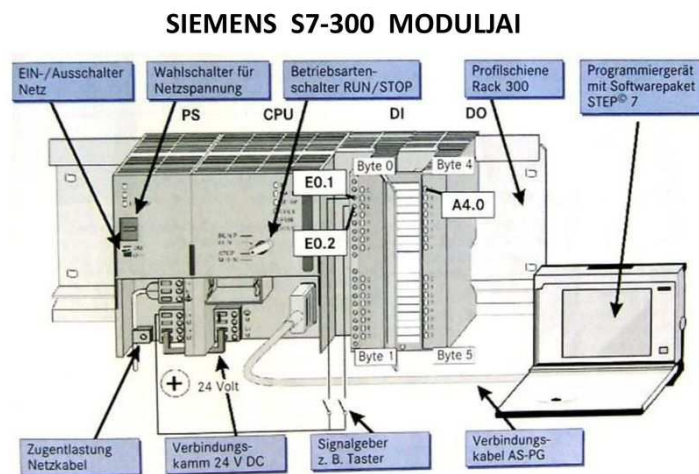


3. ábra: Festo CECC-LK kompakt PLC

Forrás: <http://www.festo.com>

- Moduláris PLC-k: Szabadon építhető, egyénileg kiválasztható modulokból áll, amelyeket bármikor tudunk bővíteni.

Ilyenek például a Siemens egyes PLC családjai (S7-300 és S7-1500as)



4. ábra: Siemens S7-300 moduláris PLC

Forrás: <https://slideplayer.hu/slide/13464799/> 9. dia

1. PLC oktatás helye a programtervben

1.1. A magyar szakképzési rendszer

A szakképzés felsőfokú szakképzettséget nem igénylő munkakör betöltéséhez vagy tevékenység végzéséhez szükséges:

- a) szakmára felkészítő szakmai oktatás és
- b) szakképesítésre felkészítő szakmai képzés. (Innovációs és Technológiai Minisztérium, 2019.)

1.1.1. Története

„A kőműves, az ács és a villanyszerelő azon vitatkoznak, hogy melyikük munkája volt előbb a földön.

– Mi építettük fel a gízai piramisokat – mondja a kőműves -, tehát először mi léteztünk.

– Dehogy! – tiltakozik fejét rázva az ács. – Mi már jóval korábban elkészítettük Noé bárkáját.

Erre a villanyszerelő kuncogni kezd.

– Mi olyan vicces? – kérdi az ács.

– A teremtés első napján mikor Isten azt mondta: “Legyen világosság!” – magyarázza a villanyszerelő – Na, addigra mi már lefektettük a kábeleket!”

Az előbbieken idézett humordarabka alapján látható, hogy a szakmák művelése több évezredes múltra visszatekintő, melyeknek látványos eredményei (pl. különböző építmények – piramisok, viaduktok stb.) illetve kevésbé látványos, de fennmaradt emlékek a kézművesipar és technológia már akkor is magas színvonalát őrizték meg.

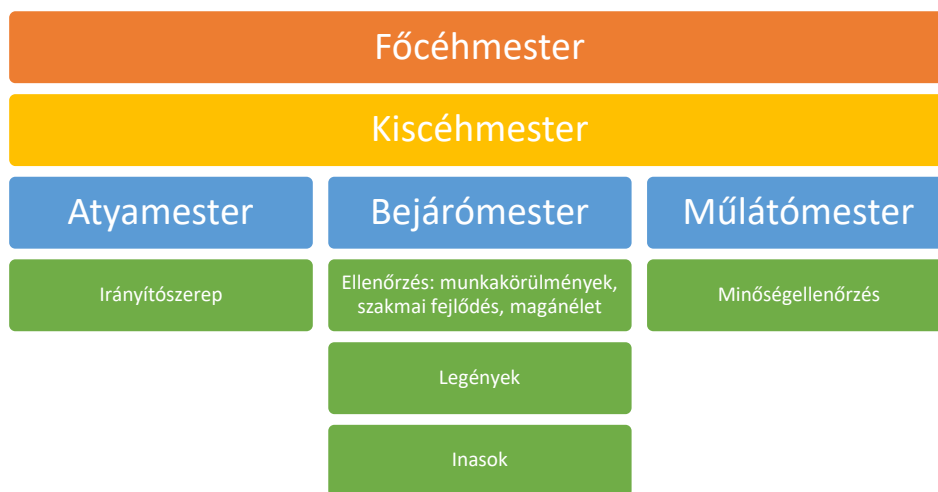
A szakképzés oktatása a magyar nép történetében még jóval a honfoglalást megelőző időkre nyúlik vissza. A mindennapokban használatos tárgyakat a „mesterek” készítették, és a tudás átadása – írásos emlékek híján – valószínűsíthetően a közösségeken belül már fiatalok óta elleték az utódok az adott „szakma fortélyait”.

A honfoglalás követő időkben – melyekről már rendelkezünk írásos emlékekkel – leginkább az egyházi intézmények (kolostorok) voltak az oktatás legfőbb központjai, amelyről az előző alfejezetben már beszámoltam.

A szakma művelését és tudásátadást, avagy a szakképzést az ezredfordulót követő évszázadokban a nyugati államokban a céhek végezték, amelynek mintáját a XIV. században Magyarország is átvette.

A céhek az azonos szakmát művelő emberek szervezete volt, amelynek feladat volt a tagok érdekvédelme, a munka és termelés szabályozása és az utánpótlásképzés, avagy a szakképzés alapjait biztosították.

Az iparosok nevelése a céhekben egy kifejezetten jól átlátható és szabályozott rendszerben történt. A „hierachia”, avagy a szakmai rangsor legalsó fokán az inasok helyezkedtek el. Az inasok több hasonló újonc társaságában dolgoztak. Pozíciójukat tekintve tanítványok voltak, nem alkalmazottak. Mesterük mellett a választott iparáguknak összes részletét megtanulták, és emellett engedelmességgel tartoztak feljebbvalóinak (segéd, mester és a mester háza népének).



5. ábra: Céhrendszer hierarchiája

Forrás: (Sanda, é.n.)

A szakképzés fejlődésére a Ratio Educationisnak csak közvetett hatása volt, mert alsó- és középfokon nem hozott létre szakiskolákat. Az anyanyelvi iskolákban (alsó fokon) a tananyag korszerűsítés a számtan tanításában ragadható meg, mert a gyakorlati hasznosság szem előtt tartását írta elő. Az egytanítósfalusi iskolákban, a paraszti gazdaságban szükséges számtani ismeretekre kellett felkészíteni, a mezővárosi kéttanítósfalusi iskolákban már a kézműves mesterségek szempontjait is figyelembe kellett venni, a nagyobb városok úgynevezett nemzeti iskoláiban viszont a mértan és a fizika alapjai mellett el kellett érni a kereskedelmi számtan, az adás-vétel szabályainak elsajátítását is. A tanítóképzéssel foglalkozó normaiskolák külön tanítót kaptak az építészeti mértan és rajz oktatására.

A szakképzés fejlődésére a Ratio Educationis csak közvetett hatást gyakorolt, pl. a számtantanításban a gyakorlatias ismeretek előtérbe helyezését írt elő. A nemzeti rajziskolák létesítésének elsődleges célja pedig a céhes keretek között folyó iparosképzés támogatása volt.

Így formálódott a – korabeli Európa integráns részét képező – magyarszakképzés a XVIII. század végig. (Sanda, é.n.)

A történeti áttekintés fontossága az lenne, hogy a jelenleg is aktív jogszabályok több esetben is a mára már történelminek számító korábbi tapasztalatokra építenek, és nagyon sok, jelenleg is meglévő elemet használnak. Erre az egyik legjobb példa a duális képzési rendszer, amely kissé nagyon átalakítva, de hasonlít a múltbéli céh és manufaktúrák „kihelyezett” oktatási sémájára.

1.1.2. A szakképzési rendszert szabályozó dokumentumok

Szakképzési törvény

„Magyarország – a többi közép-kelet európai országgal együtt – nagy optimizmussal és elszántsággal alakította át az ország társadalmi berendezkedését, a gazdaságát és a hozzá szorosan kapcsolódó oktatási rendszerét. A reformok már az 1989-es évet megelőzően is elkezdődtek(ld. pl. az 1985-ös Oktatási törvény, az adaptív szakképzési modell kialakítása stb.) Ezt a korszakot meghatározó fontos események formálták, mint

pl. a Szovjetunió megszűnése, a berlini fal leomlása, Németország újra egyesítése stb. A téma nemzetközi kitekintéseinek az alapját szolgálják ezek az események és a velejáró átalakulások.

Az 1989-es politikai eseményeket követően megalakuló kormányzat fontos feladatának tekintette az oktatási rendszer átfogó átalakítását, mindenekelőtt törvényi szabályozás révén. Több évi előkészület után adták ki 1993-ban a három, oktatással foglalkozó törvényt, köztük a Szakképzési törvényt. Számos dokumentum, –mint pl. az OKJ (jelenleg Szakmajegyzék) – kapcsolódik a törvényhez.

A politikai-társadalmi rendszerváltás hazai gazdasági következményei közül a privatizációt kell kiemelni, amely a főként anyag és energiaigényes ágazatok nagyvállalatainak megszűnését, illetve magántulajdonba kerülését jelentette elsősorban. A nehézipar, a gépi nagyipar radikális visszaszorulása, más szektorok, illetve ágazatok megjelenése, bővülése igen jelentős munkaerő átstrukturálódással járt. Drámaian nőtt a munkanélküliség.” (Lükő, 2019.)

A szakképzési törvény a mindenkor szabályozó alapidokumentuma a magyarországi szakképzésnek.

A jelenleg hatályos szabályozó dokumentum a 2019. évi LXXX. törvény és a szakképzésről szóló törvény végrehajtásáról szóló 12/2020.(II.7.) Kormányrendelet, melynek a főbb részei az alábbiak:

- A szakképzés fogalma és rendszere
- A szakképző intézmény
 - Típusai, tevékenysége és gazdálkodása
 - Szervezési rendelkezések (alapítás, megszüntetés, átszervezés)
 - Fenntartás
 - Működés
 - Alkalmazottak
- A képzésben részt vevő személyek
 - Tanulói jogviszony, jogok és köteleességek

- Szakmai oktatás és a vizsgáztatás rendszere
 - Közismereti oktatás
 - Ágazati alapképzés
 - Szakirányú oktatás
- A szakképzés irányítása, együttműködések és ellenőrzés
- Finanszírozási és információs rendszer

1.1.3. Az Országos Képzési Jegyzék

Az Országos Képzési Jegyzék (röviden OKJ) a megújult szakképzés előtti időben a szakképzési rendszer megújulásának és változásának legfontosabb dokumentuma.

Szerkezetét minden esetben az adott hatályban lévő szakképzési törvény írja elő.

Történetét tekintve, az első változat, a 7/1993 (XII.30.) MüM rendelete alapján készült el, a későbbi módosítások pedig vagy a szakképzési törvény alapján, vagy az éppen aktuális munkaerőpiaci trendek, igények és változások határozták meg.

Az OKJ előzményei az alábbiak voltak:

- Országos szakmunkásképzési Jegyzék (OSZJ)
 - Általános iskolára épült és alapszakmába rendezte a szakmaszerkezetet.
- Középfokú Szakok Jegyzéke (KSZJ – 1978)
 - A nem ipari és nem mezőgazdasági szakmákra terjedt ki
- Technikusi Szakok Jegyzéke (TSZJ)
- Vállalati (ágazati) Szakmák Jegyzéke (VJSZ)
 - A legalacsonyabb szintű, „egyszerű” szakképesítéseket tartalmazta

Az átfogó átalakításhoz 33 bizottság (piacgazdasági és ipari szereplők bevonásával) tett javaslatokat az új rendszerhez. Az új, egységesített rendszerben 955 szakképesítés szerepelt, és ez évente-kétévente módosult. (Kiss, 2010)

1.1.4. Modulrendszerű vizsgarendszer

A modulrendszerű vizsgáztatás esetén a OKJ rendszerben tanuló hallgatónak a kerettanterve szerint minden egyes modul elvégzése után modulzáró vizsgát kellett tennie.

Ezt nappali, iskolarendszerű képzés esetén az évközi számonkérésekkel teljesítették.

Iskolarendszeren kívüli képzés esetén a számonkérés a képzőhely elvárásaitól függően, lezajlottak a modulok befejezése után a modulzáró vizsgák.

A szakmai vizsga mindkettő esetben modulonként történt, tantárgyfelosztástól függően szinte az összes tantárgyból.

Ez egy 545230100000000-es OKJ számú Elektronikai technikus szakmai vizsga esetén:

- 6 db írásbeli dolgozat:
 - Elektronika-elektrotechnikai szakmai ismeretek feladatsor
 - Műszaki alapismeretek feladatsor
 - Analóg elektronikai áramkörök
 - Digitális áramkörök feladatsor
 - Automatikai teszt jellegű feladatsor
 - Digitális irányítástechnikai feladatsor
- 8 gyakorlati vizsgarész:
 - Elektromechanikai méretezési és szerelési gyakorlat
 - Elektronikai áramkörépítési feladat
 - PLC programozási feladat
 - Mikrovezérlő programozási feladat
 - Automatikai gyakorlati feladat
 - Számítógépes szimuláció feladat
 - Általános mérési gyakorlati feladat
 - Hibakeresési feladat

A részvizsgafeladatok eredményei szoftver segítségével lettek súlyozva és a modulok %-os átlaga adta ki a végső jegy %-át.

Látható, hogy ebben a vizsgarendszerben a hallgatók és jövődöbeli technikusok hatalmas terhelésnek voltak kitéve, mivel 1 írásbeli napjuk és 3 gyakorlati vizsganapjuk volt, és ezeket egy 1 napos szóbeli vizsganap zárta.

1.1.5. Komplex vizsgarendszer

A komplex vizsgarendszerben a modulzárásos számonkérési rendszer megmaradt az iskolarendszeren kívül képzésekben.

A szakmai vizsga mindkettő esetben egy egyszerűsített vizsgarendszer alapján zajlott. Ezen felül a OKJ-s számon lerövidültek és egyszerűsödtek.

Ez egy 54 522 01es OKJ számú Elektronikai technikus szakmai vizsga esetén:

- 180 perces írásbeli rész
- 300 perces gyakorlati rész
- 20 perces szóbeli rész

Az új vizsgarendszer nagyban megkönnyítette az értékelés és a feladatkészítés mechanizmusát. Emellett rengeteg terhet vett le a vizsgáztató tanárok válláról az adminisztrációt illetően.

1.1.6. Technikumi rendszer

A technikumi rendszer visszahozatala a szakképzési rendszerbe a 2020.évi szakképzési törvény által lett véglegesítve. Ugyanezen törvény megszüntette a régi Országos Képzési Jegyzéket, és helyette bevezetésre került az egyszerűsített Szakmajegyzék, melyben a korábbi 700+ szakma helyett a tagolás alapszakmákra és a ráépülő részsakképesítési rendszert vezette be.

Emellett rögzítette, hogy az alapszakmákat csak iskolarendszerű képzésben lehet megszerezni, viszont a ráépülő részsakképesítéseket tanfolyami módon is meg lehet tartani.

Ezek mellett a törvény újra átnevezte a különböző szakképzési intézményeket. A szakgimnáziumokból technikumok lettek, a szakközépiskolákból pedig szakképző iskolák.

A technikumi rendszer felépítése

A korábbi szakgimnáziumi rendszerben a képzési 4+1 vagy 4+2 éves (érettségi utáni nem előképzettséges) rendszerben zajlott. A 4. év végén az ágazatok/szakmacsoportok szakmai ismereti érettségit tettek, mely kötelező része volt az érettségijüknek. Ezután tudták elkezdeni a ráépülő +1 évet, ahol a szakmai alapokra építettek a technikus éven, csak szaktárgyakkal, mivel a közismereti tárgyakat elvégezték a 4. év végén.

Technikumi szakképzés esetén kötelezően 5 éves képzés zajlik melynek 2 része van:

- Ágazatai szakmai alapozás: a technikumban 9-10. évfolyamon az ágazati alaptárgyakat tanulják majd 10. év végén ágazati vizsgát tesznek.
 - Elektronikai technikus képzés esetén ezek az ágazati alaptárgyak a Villamos és Gépészeti alapismeretek.
- Szakirányú továbbképzés: A sikeres ágazati vizsga után a hallgatók szakirányt választanak az ágazatukon belül, majd az elkövetkezendő 3 évben a szakirányos tárgyakat tanulják. 12. év végén előrehozott érettségit tehetnek a közismereti tárgyaikból, ezzel is megkönnyítve a technikumi vizsgaévet.

A technikumi vizsgán a hallgatók egy írásbeli interaktív vizsgatevékenységet csinálnak meg, egy központi kérdésbankból. A kérdések lehetnek kifejtősek, igaz-hamis jellegűek, de nagy szerepet kap a hibakeresés és a hibák felismerése az írásbeli vizsgák alatt.

Ezután következik a gyakorlati vizsga, ahol áramkörépítési, szerelési és akár programozási feladatot kell teljesíteniük.

Az 5 éves tanulmányaik alatt a gyakorlaton elvégzett, elkészített munkadarabokat, gyakorlati feladatokat dokumentálniuk kell, mivel a szóbeli vizsgarészben egy portfóliót kell prezentálniuk a teljes 5 éves munkájukról.

A sikeres technikumi vizsga egyenértékű egy emelt szintű szakmai érettségivel.

A technikumi képzés legfőbb célja, hogy a szakmai tudást tanulási élmény alapján adja át a hallgatóknak, emiatt az ágazati alapozó képzéstől kedve a tananyag átadása projektalapú oktatási módszerrel történik. Minden évben legalább 3-4 projektet (ezt az iskola határozza meg, szóval ez intézményspecifikus) szükséges végig vinni, amelyek később alapjául szolgálnak a szakmai vizsgán bemutatásra kerülő portfóliónak.

Képzési és Kimeneti Követelmény (KKK)

A korszerű szakmai kompetenciák elsajátításához fontos mérföldkő a Szakmajegyzék szakmaihoz tartozó Képzési és Kimeneti Követelmények új szemléletű meghatározása, amelyek a szakmájukat kiválóan ismerő szakértők segítségével készültek el. Az anyagokat az ágazatért felelős minisztériumok és az Ágazati Készségtanácsok munkájában részt vevő gazdasági szereplők is véleményezték, így a követelmények szakmailag megalapozottak, korszerűek és a gazdaság igényeit tükrözik.

A Képzési és Kimeneti Követelmények tartalmát a szakképzés rendszerének átalakításához kapcsolódóan az Szkr. 12. §-a határozza meg. A Képzési és Kimeneti Követelmények a Tanulási Eredmény Alapú kimeneti követelményeket jól ismerő szakértők támogatása mellett készültek el.

A Képzési és Kimeneti Követelmények az alábbiak szerint épülnek fel:

- A szakma munkaterületének leírása;
- A szakma képesség deszkriptorainak megfogalmazása;
- Az ágazati alapoktatás munkaterületének leírása;
- Az ágazati alapoktatás tanulási eredményeinek megfogalmazása;
- Az ágazati alapoktatás vizsgakövetelményeinek megfogalmazása;
- Az ágazati alapoktatás, alapvizsga tárgyi feltételeinek definiálása;
- A szakma teljes tanulási eredményeinek kidolgozása
- A szakmairányok teljes tanulási eredményeinek kidolgozása
- A szakmairányok közös követelményeinek meghatározása
- A szakmairányok speciális követelményeinek meghatározása
- A szakma/szakmairány szakmai vizsga követelményeinek meghatározása;
- A szakmai vizsgához szükséges humán és tárgyi feltételek meghatározása;
- A részs szakmák követelményeinek meghatározása.

Programtanterv (PTT)

Az Szkr. 13. § alapján a programtanterveket a Képzési és Kimeneti Követelmények alapján kell kidolgozni.

Az Szkr. 13. § (1) bekezdése alapján:

„a programtantervet a képzési és kimeneti követelmények alapján – a Kormány adott ágazatért felelős tagjának egyetértésével – a szakképzésért felelős miniszter dolgozza ki [...]”

A programtantervek tartalmát az Szkr. 13. § (2) bekezdése határozza meg.

A programtantervek az alábbiak szerint épülnek fel:

- A szakma alapadatai;
- A tanulási területekhez rendelt tantárgyak és témakörök óraszámát évfolyamonként;

- A tanulási területek részletes szakmai tartalmának leírása;
- A részsakmák ajánlott szakmai tartalma. (Innovatív Képzéstámogató Központ, é.n.)

2. A tantervhez illeszthető oktatóanyag tervezése

A szakképző intézmények az évek során különböző ütemben tudtak egymáshoz képest fejlődni, A szakmai vizsga vizsgakövetelményei viszont egységesek maradt, amelyekhez az esetek nagy hányadában nehéz alkalmazkodni.

Emellett a technológia is rohamosan fejlődik, mellyel hasonlóképpen (főleg finanszírozási gondok miatt), nehéz lépést tartani.

Emiatt is új módszerek, módszertanok keresése szükségessé vált, és így módon a virtuális környezetben való oktatás is bekerült a gyakorlati órák palettájába.

2.1.Oktatási formák

Az első a képzési cél kiválasztása és pontos megfogalmazása. Amennyiben a célt megfogalmaztuk, akkor következik a második feladat a célnak és a rendelkezésre álló taneszközöknek a koordinálása valamennyi tényező figyelembevételével. A harmadik feladat annak a módszertani eljárásnak a kiválasztása és megvalósítása, amely lehetővé teszi, hogy a meglévő tényezők figyelembevételével a célt lehetőség szerint optimálisan elérjük.

Munkaformák szempontjából a programozható logikai vezérlők ismeretátadási és ismeretszerzési módszere többféle is lehet, ezeket vizsgáljuk meg különböző példákon keresztül.

1. Frontális osztálymunka:

Tapasztalati tanulási stratégia az alapja, de az osztály felől az aktivitás minimális, ezért ez a módszer inkább az elméleti alapok elsajátításában lehet „hasznunkra”. A foglalkozások menetét, és a célközönség aktivitását különböző IKT eszközökkel tudjuk emelni, mint interaktív prezentációk, játékos számonkérése, ami motivációnövelő hatással is tud lenni a hallgatókra.

2. Probléma-felvetéssel és probléma-megoldással való ismeretfeldolgozás:

Ez az eset leginkább a programozható logikai vezérlők történetének bemutatásakor és a működési előnyök és hátrányok megismerése esetén hasznosítható, hisz a korábban megismert eszközök és ezen eszközök működési sajátosságainak meg tapasztalása alapján a hallgatók ki tudják következtetni az innovációs lépéseket, amik szükségesek voltak a „korszerűsítéshez”.

3. Önálló tanulói megfigyelésre, kutatásra épülő ismeretfeldolgozás:

Ez a módszer leginkább a gyakorlatban, a programhasználat és a szerelési ismeretek megismerése esetén alkalmazható, mivel ez hallgatóként eltérő módon történik. A szakoktató az alapokat megmutatja, a leggyakrabban használt funkciókat, de előfordulhat, hogy a kiadott feladat kíván némi extra utánajárást, és ez esetben a hallgatót tudjuk fejleszteni azon irányba, hogy saját maga találja meg a megoldást, a rendelkezésre álló erőforrások segítségével.

4. Korábbi ismereteket alkalmazó deduktív ismeretszerzés:

A programozható logikai vezérlők gyakorlati oktatásában nagyon sok, a korábbi tanulmányaik alapján megismert eszköz és háttértudás szükséges – gondoljunk itt az előzetes analóg és digitális elektronikai előtanulmányokra – amelyek szükségesek az előrehaladásban.

5. Csoportmunka

Az eszközök mennyiségi korlátoltságai végett egyes esetekben egy feladaton, problémán több hallgató is dolgozik egyszerre. Ezen ismeretszerzési forma segíti fejleszteni a csoportban való kooperációs készségeket, a hallgató képes megismerni, hogy ő milyen szereppel rendelkezik a csapatban, illetve megtapasztalja a csoportdinamika működését.

Ezzel tapasztalatot tud szerezni a jövőbeli munkavállaláshoz is, mivel a műszaki környezetben a projektek leginkább csapatmunkára épülni, ezáltal nem lesz neki újdonság csapatban dolgozni. Hátránya ennek az ismeretszerzési formának, hogy könnyen kialakulhat a „potyautas” effektus és emiatt a csoport egyes tagjai túlvállalhatják magunkat, ami ronthat a motiváción.

2.2.A virtuális környezet: szimuláció és emuláció

2.2.1. Szimuláció

Nem minden esetben szükséges vagy lehetséges – főleg, ha nincs is hozzá megfelelő infrastruktúra és/vagy eszközkészlet – az, hogy az órán tanult ismereteket 100%-os mértékben ki tudjuk próbálni a valóságban. Az Automatika és Elektronikai technikus képzésben tanultaknál számos olyan mintaáramkör, pneumatikus/hidraulikus rendszer, illetve komplex gyártórendszer létezik, amelyek még a legjobban ellátott oktatási helyszínek esetén is kérdőjelesek. Emiatt, hogy a megfelelő tanulási élményt megadjuk a hallgató számára, illetve emellett a jövőben is hasznosítható toolkit-et adjunk számukra, ezért a hiánypótlást megpróbáljuk szimulációs programokon keresztül kieszközölni.

Ez esetben a virtuális környezetet sandboxnak nevezzük.

Mi is az a sandbox?

A sandbox (egyszerű fordításban homokozóláda, vagy homokozós játszótér), egy „izolált”, elszigetelt fejlesztői környezet, amely elősegíti a fejlesztők számára, hogy futtassák a programokat, anélkül, hogy befolyásolná a programozandó technológiát, rendszert, alkalmazást. (Rosencrance, 2021)

A sandbox egyfajta formája a szimulációnak, illetve, ha részlegesen megvan hozzá a fizikai technológia, vagy vezérlőegység, akkor az emulációnak nevezzük.

Szimuláció: Olyan módszer, amely alkalmas a folyamatok valósághű modellezésére, és vele értékelhetőek a folyamat- és rendszerállapotváltozások.



6. ábra.: Szimuláció illusztrálása:

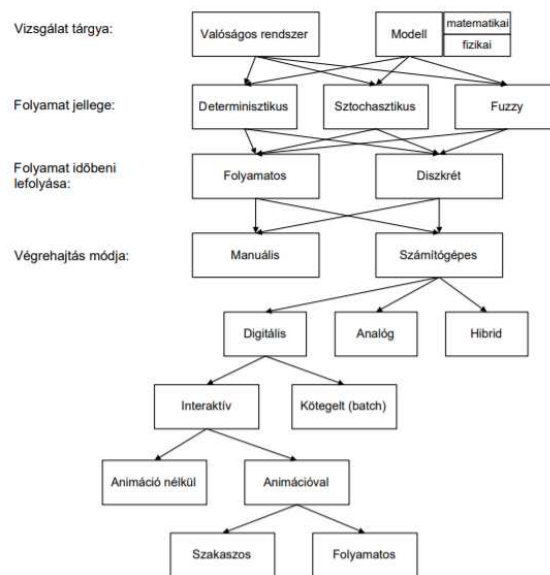
Forrás: <https://www.vstepsimulation.com/assets/uploads/2019/01/VSTEP-Building-Blocks.png>

A szimuláció célja:

- Tervezési hiba elkerülése bonyolult gépeknél, komplex rendszereknél,
- Tervezési változatok összehasonlítása,
- Dinamikus és sztochasztikus folyamatok elemzése,
- Határteljesítmények és állapotok meghatározása,
- Irányítási stratégiaváltozatok összehasonlítása,
- Működési zavarok és azok elhárításának modellezése,
- Minősítő paraméterek, befolyásoló tényezők vizsgálata.

Milyen feladatoknál használható?

- Meglévő gépek, folyamatok, rendszerek működési vizsgálata, értékelése, javítása
- Gépek, folyamatok, rendszerek tervezése,
- Gépek, folyamatok, rendszerek paramétereinek optimalizálása,
- Gépek vezérlése, folyamatok, rendszerek irányítása. (Dr. Cselényi, 2004)



7. ábra: Szimulációs modellek osztályozása:

Forrás: <http://web.alt.uni-miskolc.hu/anyagok/logrend1/szimulacio.pdf>

2.2.2. Emuláció

A számítástechnikában az emuláció egy olyan hardver vagy szoftver, amely segít imitálni egy másik programot vagy eszközt.

Az ipari vezérléstechnikában példaképpen a Festo rendelkezik hasonló emulációs eszközzel, aminek két ismert verziója van. Ezek a Veep és az EasyPort nevű eszközök.



8. ábra: Festo Veep és EasyPort modulok

Forrás: https://manuf.bme.hu/?page_id=262

Forrás: <https://www.festo-didactic.com/>

Ezen eszközök a programozható logikai vezérlőkkel összekapcsolva képesek voltak egy szoftver segítségével a meglévő modulok széleskörű repertoárjában különböző technológiákat biztosítani a hardveres szimulációban.

A Veep és EasyPort modulokhoz tartozó szimulációs szoftver pedig az EasyVeep nevet viseli, melyet a Festo Didactic Kft. fejlesztett. Az EasyVeep képes kommunikálni a Veep és EasyPort modulokkal. Az EasyVeep szoftver több virtuálisan megvalósított technológiát képes szimulálni, valós időben környezetben.

2.3.Felhasznált programok bemutatása

A programozható logikai vezérlők programozásához programok garmadáját tudjuk felhasználni, de gyártóspecifikus az egész, szóval szinte minden vezérlőhöz saját szoftver tartozik, melyek természetesen kiegészítőől függően növekednek árban.

Előrehaladás ehhez képest, hogy a CodeSys nevű cég már egy univerzális programozói felületet létrehozott, amely támogat pár vezérlőt (főleg a FESTO termékeit, illetve DLC-ként – downloadable content („egyéb” letölthető, kiegészítő tartalom) – ezen felül megvásárolhatóak.

A dolgozathoz kapcsolódóan a feladatunkhoz a Siemens által gyártott logikai vezérlőt, illetve programozói és szimulációs szoftvert fogjuk alkalmazni, mivel fizikailag is ezek állnak rendelkezésünkre az iskolai automatizálási laborban.

Az új „világrendben” amely az elmúlt 1,5 év oktatását behálózta, avagy az online oktatás érájában viszont ezen eszközökhöz se volt megfelelő hozzáférése a hallgatóknak – mivel nem látogathatták az intézményt – így a fizikai eszközparkot le kellett cserélni egy majdnem ekvivalens megoldásra, amivel fejlődésre és megértésre voltak képesek az előírt vizsgakövetelményekhez igazodva.

Így a virtuális „eszközparkunk” kiegészült a Factory I/O nevezetű grafikus szimulátorral, amivel valós idejű gyártórendszereket legőszőűen tudtunk ősszeépíteni és műkődtetni.

Az elektropneumatikus oktatáshoz segítségőűl hívtuk a hallgatók által korábban megismert Festo FluidSim nevezetű programot, mellyel a legegyszerűbbtől a legkomplikáltabb kapcsolásokig mindent őssze tudtunk rakni és a megfelelő háttőrszoftverekkel őssze is tudtuk kapcsolni a korábban említett programokkal.

A programozható logikai vezérlők programozása

„Mit csinál a programozó, ha bepánikol? Fejlesztve menekül!”

A programozható logikai vezérlőket a kezdetekben gyártóspecifikus programozási nyelv jellemezte, ezáltal minden egyes programozható logikai vezérlőhöz szükséges volt megtanulni a saját programnyelvét, ami az univerzalitásra való törekvésnek nagymértékben szembement.

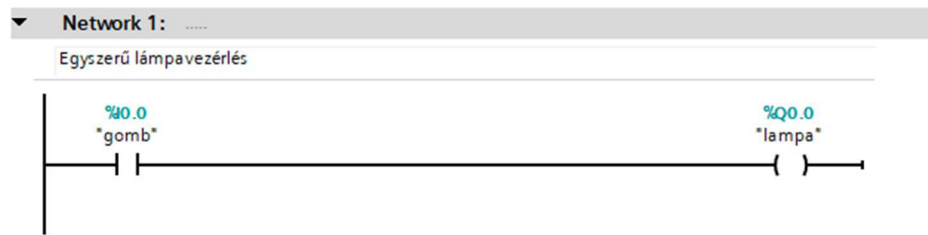
A programozási nyelvek fejlődésének szakaszai:

- „Első generációs programnyelvek”: A programok kódolása nem grafikus formában történt, hanem egyfajta Boole formátum szerint, amely nagyban hasonlított a Boole-algebrai kifejezésekhez.
- „Második generációs programnyelvek”: Megjelent az első grafikus programozási nyelv, amely elsősorban a villamossági szerelő szakembereknek nyújtott segítséget, mivel hasonlóságot mutatott az általuk használt áramutas rajzokkal.
- „Harmadik generációs programnyelvek”: Bővült az „eszköztár” újabb grafikus formátumú programnyelvekkel, illetve a mikroszámítógépek programozásánál is használt ASM jellegű és magasabb szintű szöveges programozási nyelvekkel.

1993-ban az IEC szabványban rögzítette (IEC 61131) a programozható logikai vezérlőkhöz használt programnyelveket.

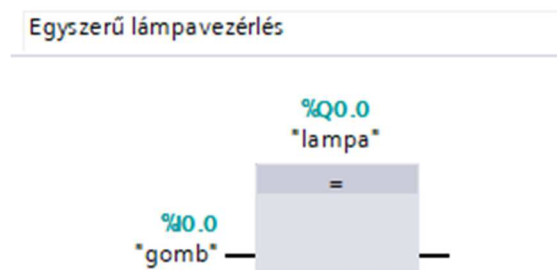
Ezek az alábbiak:

- Létradiagram – **LAD** (Ladder Logic Diagram): grafikus programnyelv, amely nagyban hasonlít az elektromechanikus áramútervekhez. Ez a programnyelv azért hasznos, mert fejlesztésnél és hibakeresésnél is könnyen végig pásztázható a jel, így egyszerűen megtaláljuk a hibát.



9. ábra: Létradiagram programrészlet – saját ábra

- Funkcióblokk-diagram – **FBD** (Function Block Diagram): grafikus programnyelv, amely részben hasonlít a klasszikus logikai áramköröknél megismert kapuáramkörökhöz, de ezek mellett saját magunk létrehozhatunk és konfigurálhatunk új „dobozokat”, amelyekben definiálhatjuk a számunkra szükséges függvényeket.



10. ábra: Funkcióblokk-diagramos programrészlet – saját ábra

- Strukturált szöveg – **ST** (Structured Text): szöveges programnyelv, amely egy magas szintű programozási nyelv, blokkokba vannak szervezve az utasítások és szintaktikája leginkább a Pascalra hasonlít.

```

1 IF #start = 1 THEN
2     //comment
3     "Max_nr" := #Array[0];
4     FOR #i := 1 TO 10 DO
5         // Statement section FOR
6         IF #Array[#i] > "Max_nr" THEN
7             "Max_nr" := #Array[#i];
8         END_IF;
9     END_FOR;
10 END_IF;
11

```

11. ábra: ST program részlet

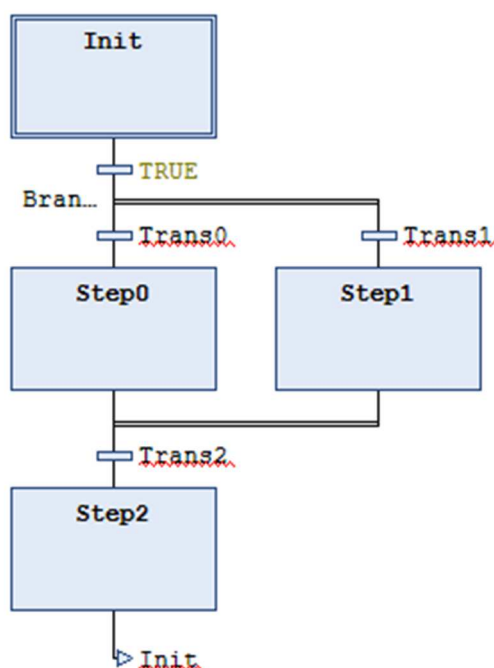
Forrás: https://isd-soft.com/nl/tech_blog/plc-programming-languages-short-overview/

- Utasításlista – **IL** (Instruction List): szöveges programnyelv, amely egy alacsony szintű programozási nyelv, amely leginkább az ASM – assembly – nyelvre hasonlít. az IL-ben írt program minden sora egy-egy újabb utasítást jelent, így lépcsőről lépésre hajtódnak végre az utasítások.

LD	<u>I0.0</u>
OR	<u>Q0.0</u>
AND	<u>I0.1</u>
AND	<u>I0.2</u>
ST	<u>Q0.0</u>

12. ábra: Utasításlista programrészlet – saját ábra

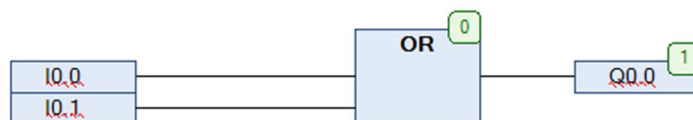
- Sorrendi folyamatábra – **SFC** (Sequential Function Chart): Grafikus programnyelv, amely gráfszerűen vezeti végig a programfolyamatokat, amelyeket lépésekre bont fel.



13. ábra: SFC programrészlet – saját ábra

Az 5 alapnyelven kívül ismerünk és használunk néhány esetben 2 extra programnyelvet, amelyek természetesen gyártóspecifikusak.

- Strukturált Vezérlő Nyelv – **SCL** (Structured Control Language): szöveges programozási nyelv, amely egy magas szintű, C-Lite-ként kategorizált programnyelv, amelyet a Siemens által gyártott eszközknél használhatunk.
- Folytatólagos folyamatábra – **CFC** (Continuous Function Chart): grafikus programozási nyelv, amelyet a CodeSys programozói szoftverben használhatunk. Nagy mértékben hasonlít a funkcióblokk-diagramos programozási nyelvhez.



14. ábra: CFC programrészlet – saját ábra

2.3.1. Siemens TIA Portal

A TIA Portal (TIA - Totally Integrated Automation) a Siemens Simatic S7 szériához tartozó PLC-k programozására alkalmas szoftver legújabb verziója.



15. ábra: Siemens TIA portal

Forrás: <https://traininginplc.com/courses/siemens-tia-portal/>

A TIA Portal önmagában egy komplett szoftvercsomag (ahogy a neve is mutatja), és ezen belül találhatóak külön modulok, amelyek választhatóak a telepítés során.

Ezek közül a legfontosabbak:

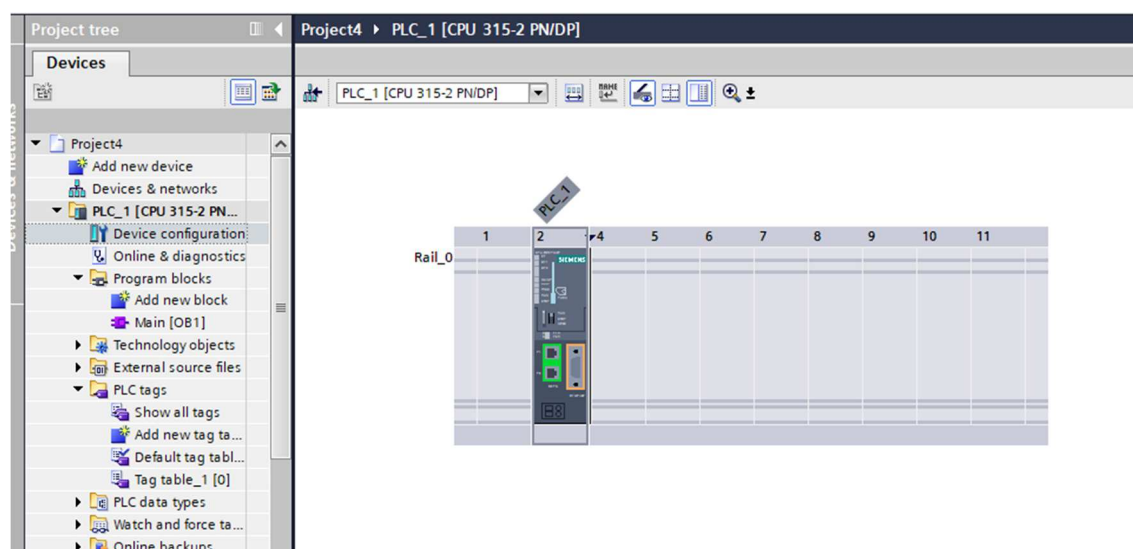
1. STEP 7

A TIA Portal gerincét jelentő szoftver, mellyel a programozható programozása történik. Ez a szoftver tartalmazza a PLC-k adatbázisát, a programozáshoz szükséges nyelveket.

A STEP 7 név egy német mozaikszó, amely a **ST**euerungen **E**infach **P**rogrammieren (vezérlőket egyszerűen programozni) szóösszetételből származik. A 7-es szám pedig az éppen aktuális készülécsaládot jelzi.

A szoftver grafikus kialakítású, és ahogy jöttek hozzá a frissítések, egyre felhasználóbarátabbá vált a program, ezáltal a kezdeti döcögése a TIA Portalnak végül elmúlt (rengeteg optimalizációs hibája volt kezdetben).

A STEP7-ben történik a rendszer konfigurációja (a programozható logikai vezérlő „összeépítése”), a változók hozzárendelése az adott be-kimeneti egységhez, a CPU konfigurálása és a program elkészítése a választott programnyelven.

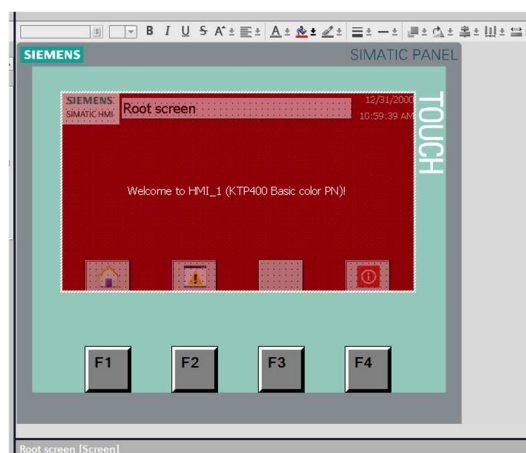


16. ábra: STEP 7 kezelőfelület részlete – saját ábra

2. WinCC

A WinCC modul kéz a kézben jár a STEP7-el. Ezzel a kiegészítéssel tudjuk megtervezni a programozható logikai vezérlő és a kezelője közötti kapcsolatot. Ezt a kapcsolatot HMI-nek (Human-Machine Interface), avagy ember-gép felületnek nevezzük.

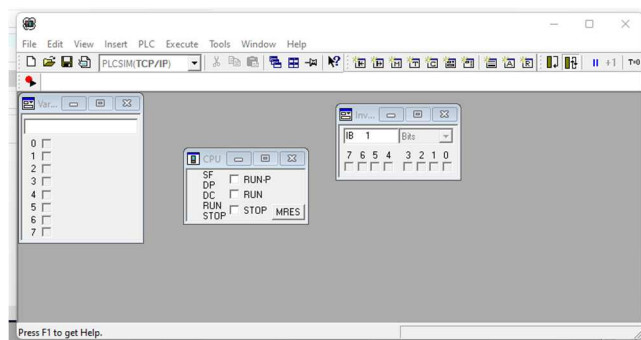
Tulajdonképpen itt tudjuk megtervezni a gépünknek szánt grafikus kezelőfelületet. Ezen a kezelőfelületen elhelyezhetünk nyomógombokat, kapcsolókat, feliratokat, instrukciókat és akár animációkat is.



17. ábra: WinCC - HMI tervezői felület – saját ábra

3. PLCSim

Ha a programunk még nagyon fejlesztői fázisban van, akkor nem szükséges rögtön a fizikai eszközt csatlakoztatnunk a teszteléshez, hanem elég bitszinten működtetni a programunkat, amelyet a PLCSim nevezetű szoftverrel tudunk megtenni.



18. ábra: PLCSim felület, S7-300as PLC-nél – saját ábra

4. PLC Safety

Biztonsági programozható logikai vezérlőkhöz tartalmaz extra kiegészítőket, amelyeket be tudunk tenni a programunkba.

Biztonsági PLC-ket komplex gyártórendszereknél és robbanásbiztos üzemekben használnak legfőképpen.

5. SINAMICS StartDrive

Van, hogy a programozható logikai vezérlőnket valamilyen villamos hajtás működtetésére választottuk ki, és ezzel a kiegészítő programmal tudjuk akár programozni a hajtáshoz tartozó motorvezérlőt/frekvenciaváltót.



19. ábra: Siemens Startdrive illusztráció

Forrás: <https://new.siemens.com/hu/hu/termek/automation/drives/selection-and-engineering-tools/sinamics-startdrive-commissioning-software.html>

2.3.2. Factory I/O

A Factory I/O egy újgenerációs szimulációs program, amivel komplex automatizálási rendszereket tanulhatunk meg összerakni, programozni és tesztelni. Segítségével pillanatok alatt felépíthetünk bármilyen komplexitású gyártósort az iparban használt ekvivalens eszközökkel. Kompatibilitását tekintve a legtöbb programozható logikai vezérlőt támogatja (Siemens, Allen-Bradley, Schneider és a CodeSys fejlesztői környezetet).

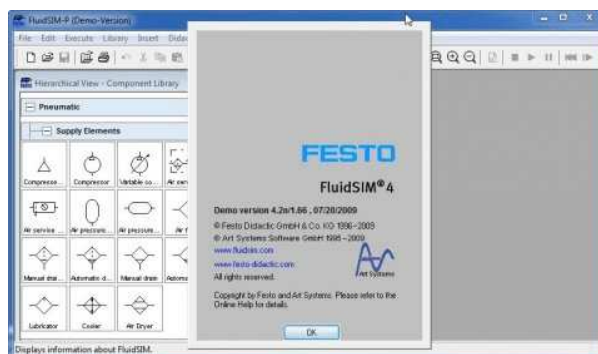


20. ábra: Factory I/O

Forrás: <https://docs.factoryio.com/>

2.3.3. Festo FluidSIM

A Festo több mint 20 éve vezető helyen álló szimulációs szoftvere, amivel elektromechanikus, pneumatikus és hidraulikus kapcsolásokat tudunk szimulálni.



21. ábra: Festo FluidSIM

Forrás: <https://fluidsim.software.informer.com/Let%C3%B6lt%C3%A9s/>

2.4.A tervezett oktatóanyag

2.4.1. Feladatléírás

A feladat egy egyszerű belépő szintű alapprogram megtervezése és kivitelezése a korábban bemutatott programok használatával.

Feladatkiírás:

Tervezzen egy futószalagos vezérlést, amely megmutatja, hogy adott doboz milyen útvonalon halad végig a gyártáslőkészítéstől a logisztikáig.

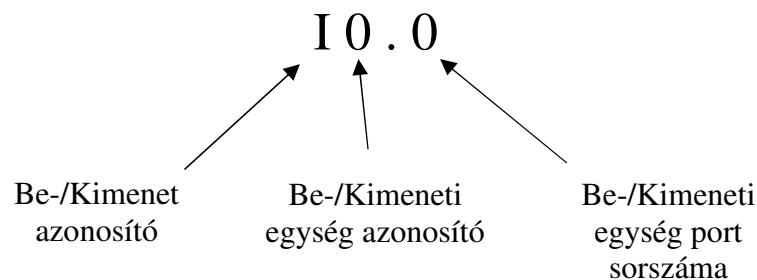
A feladatot a programozható logikai vezérlőt a TIA Portal segítségével programozza fel, programozási nyelvnek létradiagram (LAD) programnyelvet használjon, majd a kész programhoz a tesztkörnyezetet készítse el először Festo FluidSIM, majd pedig a Factory I/O segítségével. Munkájáról egy rövid beszámolót kell készítenie, amit jegyzőkönyvként kell leadnia.

A feladathoz szükséges eszközlista:

Eszköz	Jellemzők	Eszköz	Jellemzők
FluidSIM eszközök		Factory I/O eszközök	
Futószalagmotor 2 db	Indikátorlámpák a könyvtárból (sárga színnel működtetve)	Futószalag	4m hosszú
Érzékelők 5 db	Tapadónyomógomb – záró érintkezős Start-Cél és a köztes érzékelők	Diffuse sensor 5 db	
Pozíciójelző lámpa – 5 db	Egyszerű lámpák a könyvtárból (kék színnel működtetve)	Lámpaoszlop 5 db	
Start-stop gomb 2 db	Visszapattanó nyomógomb – záró érintkezős	Nyomógomb 2 db	
PLC	Siemens Simatic S7-300 CPU 315PN/DP Art. No.: 6ES7 315-2EH14-0AB0 Firmware: V3.2 Tápegység: 5A Be/kimeneti modul: 16 DI/DO	Csomag lerakó	
		Csomag felvevő	

1. táblázat: Eszköz és konfigurációs lista

Alapesetben a programozható logikai vezérlők be- és kimeneteinek szintaktikáját az „I” mint input és „O” mint output-tal (Siemens PLC-k esetében O helyett Q-t szoktunk használni) határozzuk meg. Ezt követi a szintaktikában egy szám (0 – n között lehetséges), amely az adott vezérlő összeállításától függ. Ez alaptól mindig 0, de ez a szám nőhet a konfiguráció függvényében. A következő elemet egy ponttal választjuk el, és az utolsó elem pedig az adott I/O portot jelöli. Ennek számozása 0-7 között lehet, mivel egy egységen maximálisan 8 bitet tudunk kapcsolni.

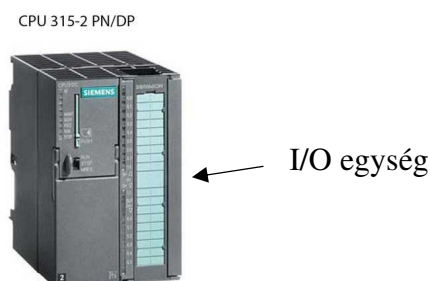


22. ábra: I/O címzés – saját ábra

Példaképpen: I1.2 – Bemenet, az 1-es számú egységen található, és a 2-es port

Programozáskor legegyszerűbb esetben csak hivatkozunk a kódban vagy a grafikus programban a szükséges I/O egységre, de ez nem túl felhasználóbarát egy sok soros, rengeteg szenzort és aktuátort igénybe vevő program esetén. Ilyen esetekben szoktuk alkalmazni az allokációs listát, aminek a segítségével a kiválasztott címzést funkciójához rendelve elnevezzük, így elég a későbbiekben meghívni a port elnevezését.

Legjobb gyakorlat, ha beszélő nevet adunk neki, mint pl.: szenzor1, ajto1, motor2, szalag3, szelep5.



23. ábra: Siemens Simatic S7-300 I/O modul

Forrás: <https://www.indiamart.com/proddetail/siemens-s7-300-cpu-315-2-pn-dp-standard-24301726733.html>

Az allokációs lista a PLC programozásához:

I/O címzés	Hozzárendelt feladat
I0.0	Start
I0.1	Stop
I0.2	1. pozíció szenzorja
I0.3	2. pozíció szenzorja
I0.4	3. pozíció szenzorja
I0.5	4. pozíció szenzorja
I0.6.	5. pozíció szenzorja
Q0.0	1. futószalag
Q0.1	2. futószalag
Q0.2	1. pozíció jelzőlámpa
Q0.3	2. pozíció jelzőlámpa
Q0.4	3. pozíció jelzőlámpa
Q0.5	4. pozíció jelzőlámpa
Q0.6	5. pozíció jelzőlámpa

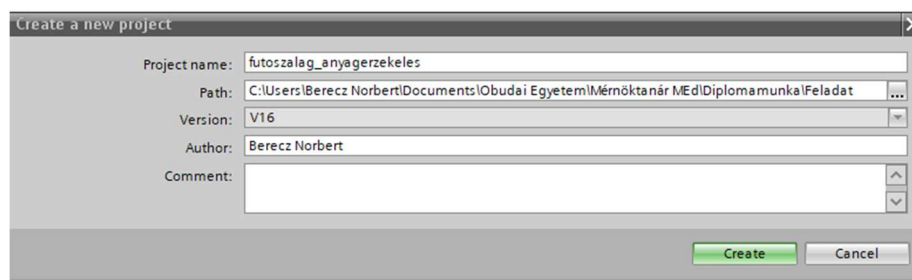
2. táblázat: Allokációs lista

2.4.2. Feladat elkészítése

A feladatmegoldás óraterveit az 2. számú melléklet fogja tartalmazni.

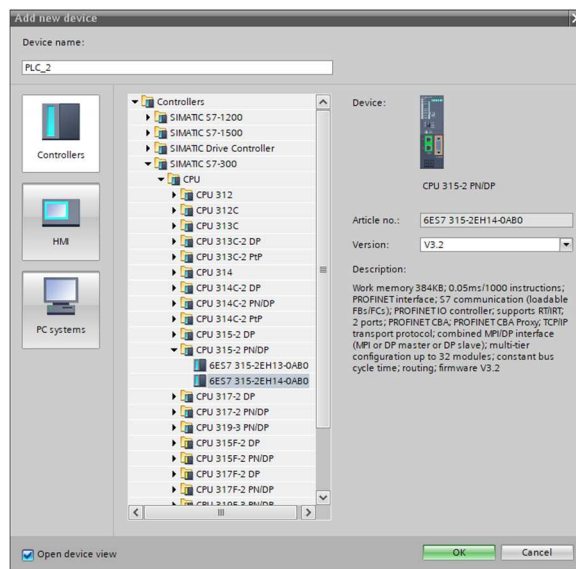
A PLC program elkészítésének lépései:

1. Indítsuk el a TIA Portal alkalmazást.
2. Adjuk meg a projekt nevét, ami az alábbi legyen: *futoszalag_anyagerezkeles*



24. ábra: Új projekt létrehozása – saját ábra

3. Következő lépésben válasszuk ki a feladatleírásban megjelölt PLC típusát a legördíthető könyvtárban.

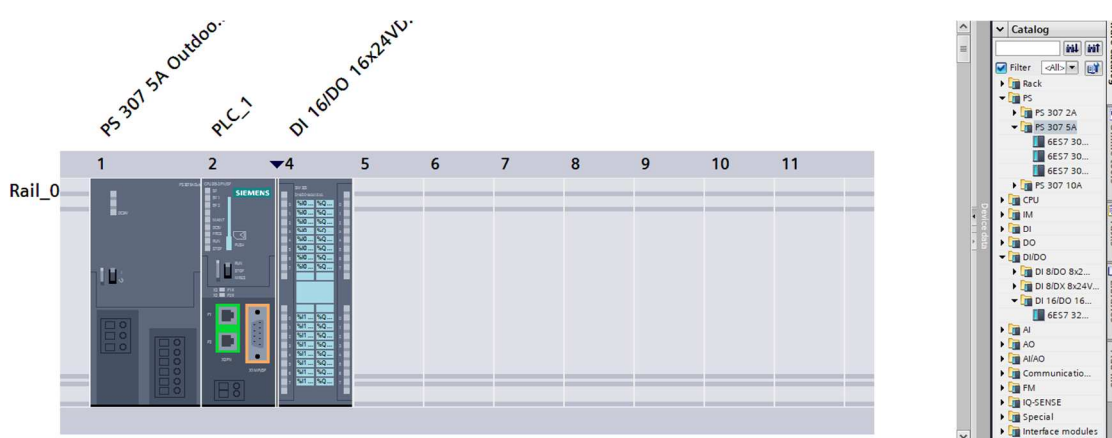


25. ábra: PLC kiválasztása – saját ábra

4. Hardverkonfiguráció összeállítása

Miután sikeres volt a PLC kiválasztása, a program továbbvisz minket a hardverkonfigurációs nézetbe. Az előző lépésben csak a programozható logikai vezérlő központi egységét (CPU) választottuk ki, így szükséges hozzá még beilleszteni a tápegységet, illetve a be-/kimeneti modulokat.

A hardverkonfigurációs nézetben drag and drop jelleggel tudjuk a szerelősínre (rail) elhelyezni az egyéb kiegészítő modulokat.



26. ábra: A kész hardverkonfiguráció – saját ábra

5. Allokációs lista elkészítése

Következő lépésben a korábban bemutatott és feladatleírásban szereplő allokációs listát készítjük el.

A TIA Portal esetében ezt TAG Table-nek nevezik (régebbi STEP7 verziók esetén és még S7-200-as MicroWIN alkalmazásban ezt Symbol Table-nek nevezték).

A projektnézetben a bal oldali menüsoron keressük meg a PLC tags nevű fület, nyissuk le, és kattintsunk duplán a Default tag table-re. A program ekkor előhívja az allokációs listát, amit hasonlóan tudunk kitölteni, mint egy Excel táblát.

A programozható logikai vezérlők a nevükből fakadóan elsősorban BOOL típusú változókkal dolgoznak, mivel kétállapotú digitális jelek érkezhetnek a be-/kimeneti modulokra. A mi hardverkonfigurációnkban is DI/DO modulokat helyeztünk el.

Az általános digitális technikában kétértékű (bináris) jelek esetében 2 fajta logikával (4 féle létezik, mert elkülönítünk pozitív és negatív logikát is) szoktunk dolgozni, amelyek:

- 0-ra hatásos logika – alacsony jelszint
- 1-re hatásos logika – magas jelszint

A magas és alacsony jelszint

1-re hatásos logikánál legfőképpen a bejövő szenzorjeleket értelmezzük, illetve az aktuátorok állapotát is ezzel jellemezzük.

0-ra hatásos logikát a biztonsági protolloknál és blokkoknál szoktunk alkalmazni.

Ezeket a kétállapotú jeleket az alábbi táblázat alapján tudjuk értelmezni:

I/O típus	Érték	Jelszint	Magyarázat
Bemenet	1	magas	Információ érkezett a szenzortól (pl. megnyomtak egy gombot)
Bemenet	0	alacsony	Nem érkezett információ a szenzortól (pl. nem nyomtak meg egy gombot)
Kimenet	1	magas	Bekapcsolunk valamilyen aktuátort (elkezd világítani egy lámpa)
Kimenet	0	alacsony	Nem kapcsolunk be valamilyen aktuátort (nem kezd el világítani egy lámpa)

3. táblázat: Logikai szintek és magyarázatuk

Visszacsatolva ezeket az értékeket szoktunk inverz (negált) formában is felhasználni, mert lehet, hogy a program futását a negált érték fogja előrébb vinni.

Ezen felül természetesen egész típusú változók (INT, FLOAT, DOUBLE...stb.) változókkal is tudunk dolgozni, de ezeket komplexebb vezérlési folyamatok esetén szoktuk használni (számlálók, aritmetikai műveletek...stb.).

	Name	Data type	Address	Retain	Acces...	Visibl...	Cor
1	Start	Bool	%I0.0				
2	Stop	Bool	%I0.1		☒	☒	
3	szenzorpoz1	Bool	%I0.2		☒	☒	
4	szenzorpoz2	Bool	%I0.3		☒	☒	
5	szenzorpoz3	Bool	%I0.4		☒	☒	
6	szenzorpoz4	Bool	%I0.5		☒	☒	
7	szenzorpoz5	Bool	%I0.6		☒	☒	
8	futoszalag1	Bool	%Q0.0		☒	☒	
9	futoszalag2	Bool	%Q0.1		☒	☒	
10	jelzolampa1	Bool	%Q0.2		☒	☒	
11	jelzolampa2	Bool	%Q0.3		☒	☒	
12	jelzolampa3	Bool	%Q0.4		☒	☒	
13	jelzolampa4	Bool	%Q0.5		☒	☒	
14	jelzolampa5	Bool	%Q0.6		☒	☒	
15	<Add new>				☒	☒	

27. ábra: A kész TAG Table – saját ábra

A TAG Table elkészítése során az „I” és „Q” mellett egy harmadik típust is ki tudunk választani, amit „M”-el jelöl a program. Ez merkelnek nevezzük, vagy bizonyos eszközöknél flag-nek. A merkelek segédváltozók, amelyeket nem szeretnénk se a bemenetre, se a kimenetre kiküldeni, viszont néhány ideiglenes állapot eltárolásához hasznos lehet. Ilyen például, amikor elektropneumatikus rendszernél egy munkahenger többször is vált a „+” és „-” állapotok között.

6. Programkészítés

Miután elkészítettük az allokációs listánkat, nekiállhatunk összeállítani a programunkat.

Nem szeretnénk túlbonyolítani a program készítését, így egyszerűen az OB1-es blokkban fogunk dolgozni.

A Siemens PLC-k esetében különböző programblokk típusok léteznek Ezek az alábbiak:

- OB – Organization block (szervezői blokk): Ezek képeznek összeköttetést a programozható logikai vezérlőn található operációs rendszer és a felhasználói program között. Az általunk írt program egésze az OB1-be kerül, ahol aztán ciklikusan meghívódik és lefut. Különböző típusú OB-k léteznek ezen felül:

- OB 100-102: inicializáló blokkok
- Megszakításvezérlési OB-k
- Eseményvezérelt OB-k

Az OB1-be ezek felül meghívhatóak speciális blokkok, amik az alábbiak:

- FC – Function (Függvény): Adott vezérlési feladatot hajt végre, meghatározott bemeneti értékek esetén. A program lefutása során többször meghívható, ezáltal egyszerűsíti a programozás folyamatát.
- FB – Function Block (Függvényblokk): Egy olyan függvény típusú blokk, amelyik rendelkezik egy adatbázissal, ami a ki- és bemeneti adatokat el tudja tárolni.

A programkészítés folyamata

A programunk egy Start-Stop rendszerrel működik, amely Start esetén elindítja a futószalagokat, Stop esetén pedig leállítja.

Kétféleképpen lehet a start-stop rendszert elkészíteni, vagy öntartással, vagy SET-RESET tároló felhasználásával.

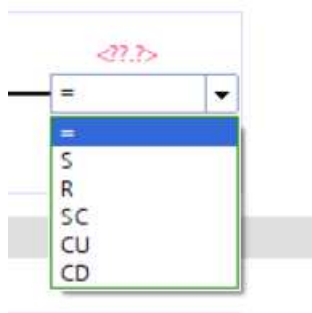
Nekünk jelen esetben egyszerűbb a SET-RESET módszer, mivel egyszerűbb és átláthatóbb a megvalósítása.

Ehhez szükséges nekünk 1-1 záró típusú érintkező, illetve egy S és R típusú coilra. Ezeket a felső menüsoron találjuk meg.



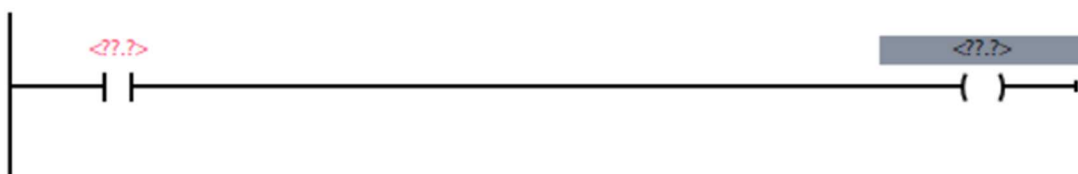
28. ábra: Érintkező eszköztár – saját ábra

Az egyszerű coilra ha duplán rákattintunk, megjelenik egy legördülő menü, amivel hozzá tudjuk rendelni a funkciókat.

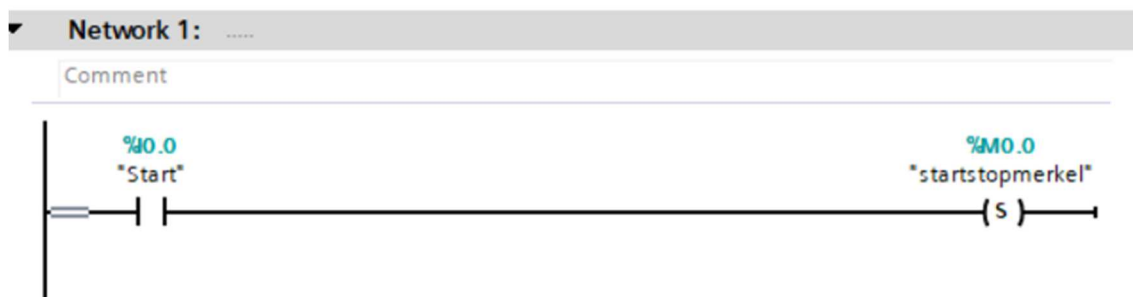


29. ábra: Coil funkció hozzárendelése – saját ábra

Ha ezzel megvagyunk, el kell neveznünk az egyes elemeket. Ezekben segítséget nyújthat az előzőleg elkészített allokációs lista és a így nem szükséges „megjegyeznünk”, hogy melyik ki-/bemenetre hivatkozunk.



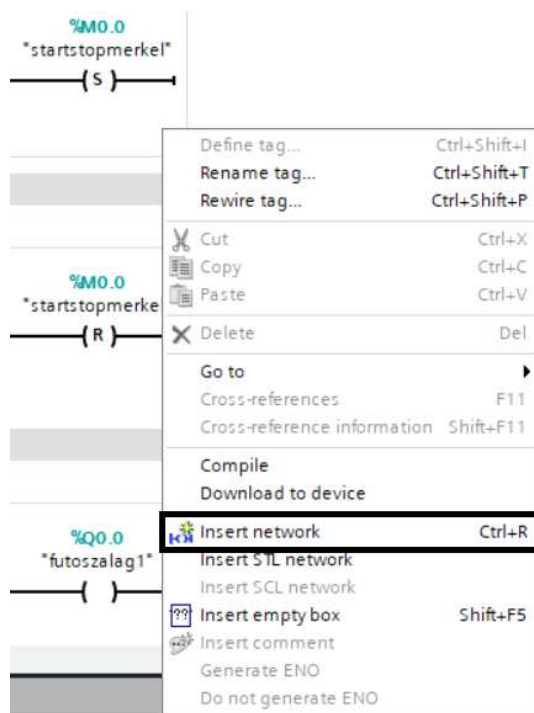
30. ábra: Elnevezés nélküli network - saját ábra



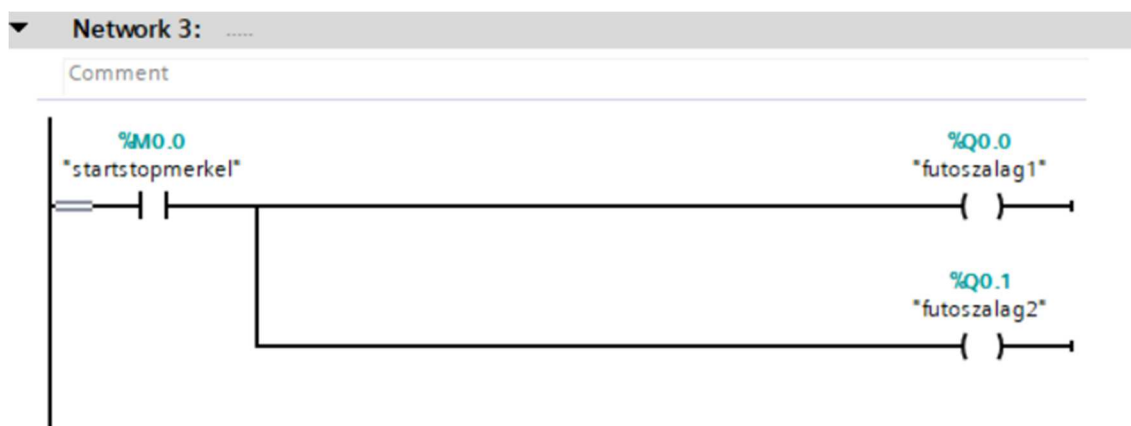
31. ábra: A start rendszer logikai felépítése – saját ábra

Ha a start-stop merkelünk aktív lesz, akkor ennek hatására elkezd működni a két futószalagunk is, amelyhez az alábbi módon készítjük el a logikai kapcsolatot.

Ha új sort szeretnénk beilleszteni, akkor azt a TIA Portal ezt megteszi helyettünk automatikusan, de ha ez mégse tenné, akkor jobb kattintás – Insert new network.



32. ábra: Új network beszúrása



33. ábra: Futószalagok működtetése – saját ábra

Párhuzamos leágazást a korábban már beillesztett eszköztár ábrán a lefelé mutató nyíllal tudunk létrehozni, és fontos, hogy az elágazási pont közelében kattintsunk, máskülönben máshol alakítja ki a csomópontot.

A pozíció érzékelő szenzorok mindegyike ugyanúgy működik, ezért erről csak egy ábra kerül beszúrára.

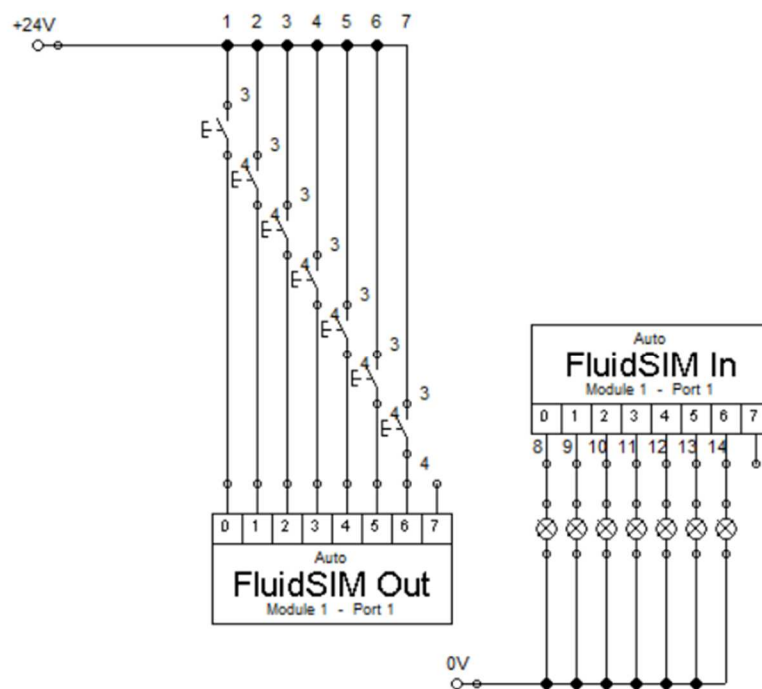


34. ábra: A szenzorok és jelzőlámpák logikai kapcsolata - saját ábra

A program elkészítése után A felső menüszalagon az Online menüpont – Simulation – Start simulation gombra kattintva elő tudjuk hozni a korábban bemutatott szimulátort.

7. Festo FluidSIM szimuláció

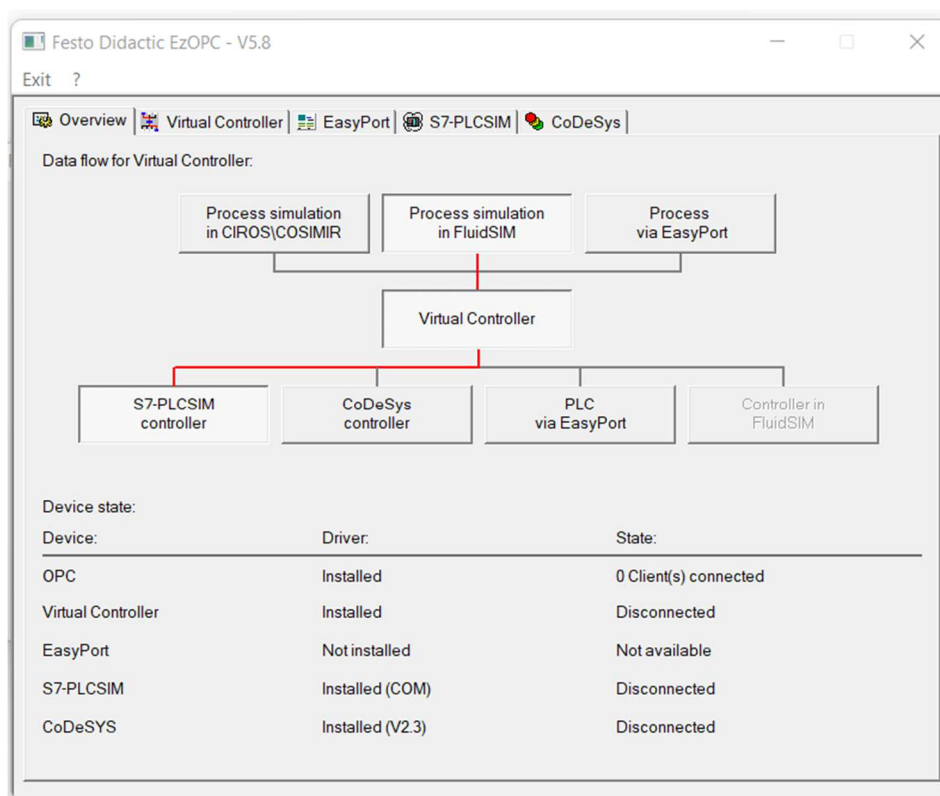
A szimulációhoz először hozzunk létre egy üres rajzot. A bal oldalon lévő eszköztárból helyezzük el a feladatkiírásban szereplő alkatrészeket az alábbiak szerint:



35. ábra: FluidSIM szimuláció kapcsolási rajza – saját ábra

Ha ezzel megvagyunk, kattintsunk duplán a FluidSIM Out majd a FluidSIM in modulokra, és állítsuk be, hogy tudjon a program kapcsolódni a TIA Portal PLCSIM-el.

A két program közötti kapcsolatot az ezOPC nevű programmal tudjuk megvalósítani, amit ha elindítottunk, a kiválasztott útvonalon fognak kommunikálni egymással.



36. ábra: EZopc szoftver beállítása – saját ábra

8. Factory I/O szimuláció

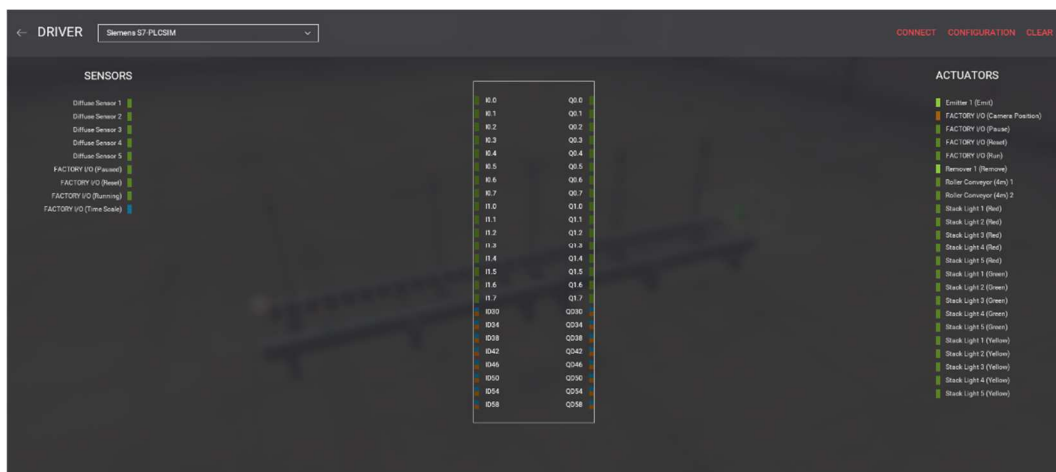


37. ábra: A szimulálandó gyártósor – saját ábra

A szimulált gyártósort előre elkészített jelentként a hallgatók megtalálhatják a programban, Conveyor_belt_basic néven. Ez a modell már előre tartalmazza a jelzőlámpákat, szenzorokat és a működtetett futószalagokat.

Nincs más feladatunk vissza, csak össze kell kötni a megfelelő vezérlővel, amit a File/Drivers menüpontban tudunk megtenni, vagy gyorsgombbal az F4 lenyomásával.

A legördülő menüből kiválasztjuk az S7-PLCSIM opciót, majd összepárosítjuk a szenzorokat az adott ki és bemeneti egységekkel, amiket előzőleg a programban már elkészítettünk az allokációs lista segítségével.



38. ábra: Vezérlő beállítása – saját ábra

3. A tervezett oktatóanyag alkalmazása tanteremi környezetben

A korábbi fejezetben szó esett a tervezett tudásanyag átadásának módszertani sajátosságáról, de emellett, hogy a csoport dinamikáját és a tanulás minőségét az előadási módunk keresztül tudjuk alakítani, a környezetünk kialakítása is hatalmas befolyással bír ebben a folyamatban.

Ebben a fejezetben bemutatásra kerül, hogy milyen módon lehet a legideálisabb oktatási környezetben foglalkozni ezzel a témával.

A tervezett feladat kipróbálására a második nagyobb lezárás idején került sor, így az ismerkedés a programok használatával

3.1. Fizikai oktatókörnyezet

3.1.1. Szaktantermek

A szaktanterem egy olyan helyiség, amely terem vagy laboratórium és ebben a helyiségben megfelelő infrastruktúra áll rendelkezésre egy adott tantárgy vagy tantárgyegyüttes oktatásához,

Alapfelszereltségét illetően a székek, asztalok, tábla, IKT eszközök mellett a tantárgyi jellegéhez szükséges felszereléseket tartalmazza.

A mi esetünkben az elméleti alapozóhoz egy megfelelő felbontású projektor és vetítőfelület elég.

A csoportmunkához vagy egyéni munkához szükséges egy megfelelő tudással rendelkező személyi számítógép vagy laptop, és kiegészítésnek egy extra monitor az osztott képernyőn való munkához.

A munkatér elrendezése a mi esetünkben jelenleg frontális elrendezésben van, de egy pályázatnak hála a jövőben ez meg fog változni. Mint tudjuk a frontális elrendezés csak frontális előadás esetén hasznos, mivel a hallgatók megfelelően látják a vetítőfelületet,

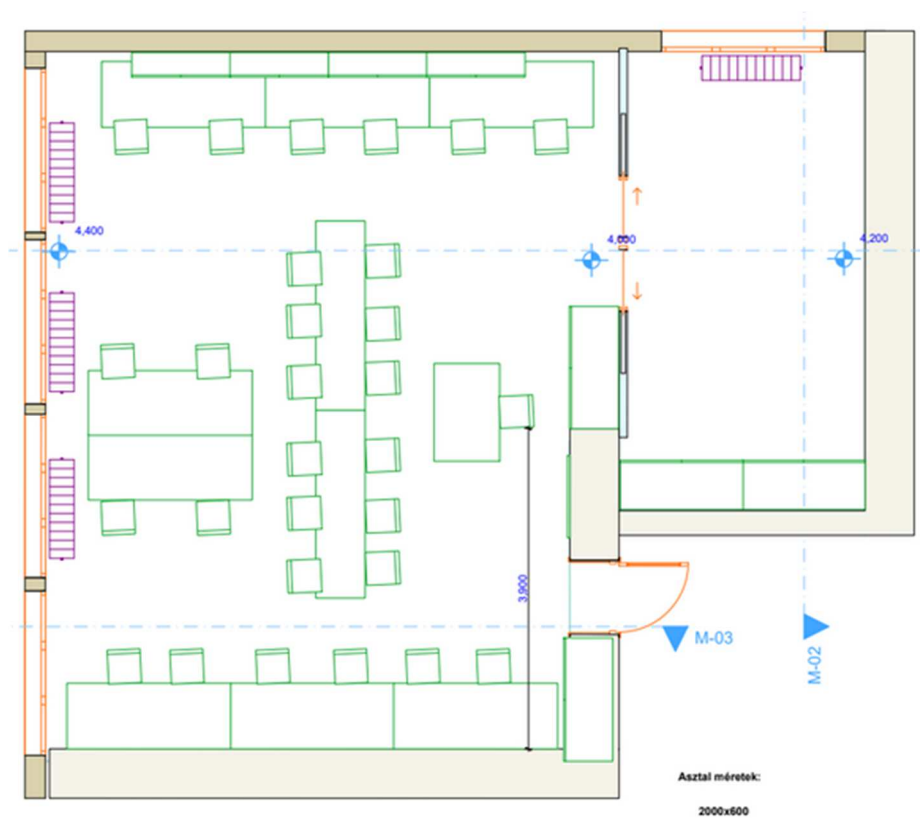
tudnak jegyzetelni. Laborgyakorlatok esetén a terem átjárhatatlan, így a nem lehet elég hatékonyan monitorozni a munkavégzést, emiatt pót megoldásként csak a terem egyik fele lett kialakítva mérőhelyekké. A labor 16 főre van tervezve (8 darab 2 fős asztallal), csoportbontásban vannak a gyakorlatok, így ebben az esetben 4x2 főnek tudjuk biztosítani a mérőhelyeket.

Számítógépes munkavégzés esetén 8 darab laptop áll rendelkezésre, kiegészítve 1-1 monitorral. Felsőbb évfolyamokban (13.-os osztályok), engedélyezett a monitorok használata osztott képernyőn, mert ők már a megbízhatóbb kategóriába tartoznak, viszont az alsóbb évfolyamokon a monitorok a tanári asztal felé néznek, képernyőmásolás üzemmódban, így a tanár figyelemmel tudja kísérni a munkálatokat.

A pályázathoz tartozó fejlesztés egy kombinált tanterem létrehozása, ahol szigetszerű elrendezésben a hallgatók képeseket csoportmunkában dolgozni, illetve emellett kisebb költözéssel átalakítható frontális oktatásra, vagy nagyobb mértékű projektmegbeszélésre. Ez mellett a terem átjárhatósága javul, és a hátsó szertár átalakítása által egy robotcella és villamos gépek mérésére szolgáló kislabor is ki lesz alakítva.



39. ábra: C216-os labor jelenleg a frontális kialakítással és az egyoszlopos mérőállásokkal



40. ábra: Kombinált frontális és csoportmunka elrendezés

Forrás: Saját ábra – saját tervezés

3.2. Az új kihívás – online oktatókörnyezet

Mint tudjuk és tapasztaltuk, az elmúlt 2 év leginkább emlegetett és legnépszerűbb szava az online oktatás volt.

A dolgozatban a saját iskolánk online oktatási módszereinek bemutatására fog sor kerülni.

Több pilléren támaszkodott az iskolai az online térben. Legfontosabb részei az e-Kréta mint adminisztrációs felület, a Google Classroom, mint feladat-adatbázis és legfőbb tárhely, az online kommunikáció pedig a Cisco Webex alkalmazásán keresztül történt.

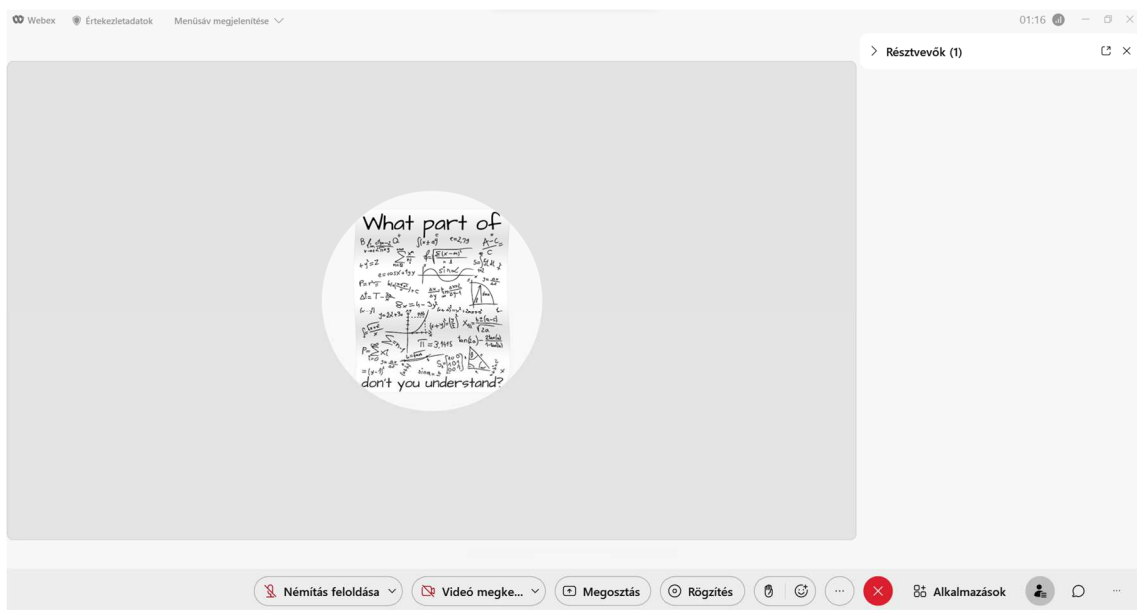
Az e-Krétában a jelenlétek, házi feladatok és az üzenetváltások fő színtere volt, amely egyben a leghivatalosabb felületnek számított.

A Google Classroom-ot feladatfeltöltésre, feladatok kiadására, oktatóanyagok és videók megosztására használtuk, illetve itt adtunk rövid visszajelző értékeléseket a munkákra.

A Cisco Webex egy multifunkcionális online kommunikációs csomag része, főleg nagyvállalati kommunikációra, de kisebb módosításokkal könnyen alkalmazhatóvá vált az online oktatásra.

A tanárok saját „osztályteremmel” rendelkeztek, amit, ha elindítottak, a hallgatók tudtak csatlakozni hozzá. Ebben a virtuális térben képernyőmegosztással prezentációkat tudtak előadni, illetve, ha adott volt egy rajztábla, akkor táblavázlatok létrehozása is könnyen megoldható volt ezen a felületen. Hátránya – de szinte az összes ilyen programnak – hogy nem lehetett az összes diákot egyszerre monitorozni, viszont egyenként az órák közben be tudták mutatni az órai munkájukat, ha éppen olyan gyakorlati órán alkalmazták. Elakadások esetén kérni lehetett távsegítséget, így az oktató a saját gépéről hozzá tudott nyúlni az adott hallgató munkájához, és tudott segíteni a korrekcióban.

Másik nagy előnye a Webexnek, hogy kis fájl méretben, de jó minőségű felvételt tud készíteni az adott óráról.



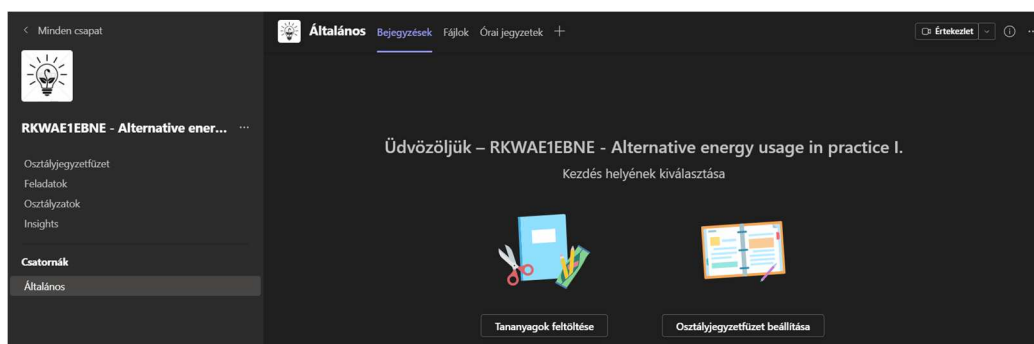
41. ábra: Cisco Webex oktatói nézőpontból – saját ábra

Ennek a „szentháromságnak” az egyik legnagyobb hátránya az volt, hogy mind külön felületen voltak elérhetőek, az integráltság így nem tudott megvalósulni.

Integrált oktatási felületnek a legjobban alkalmazható lenne a Microsoft Teams alkalmazása, de az intézmény nem rendelkezett az ehhez szükséges licencekkel.

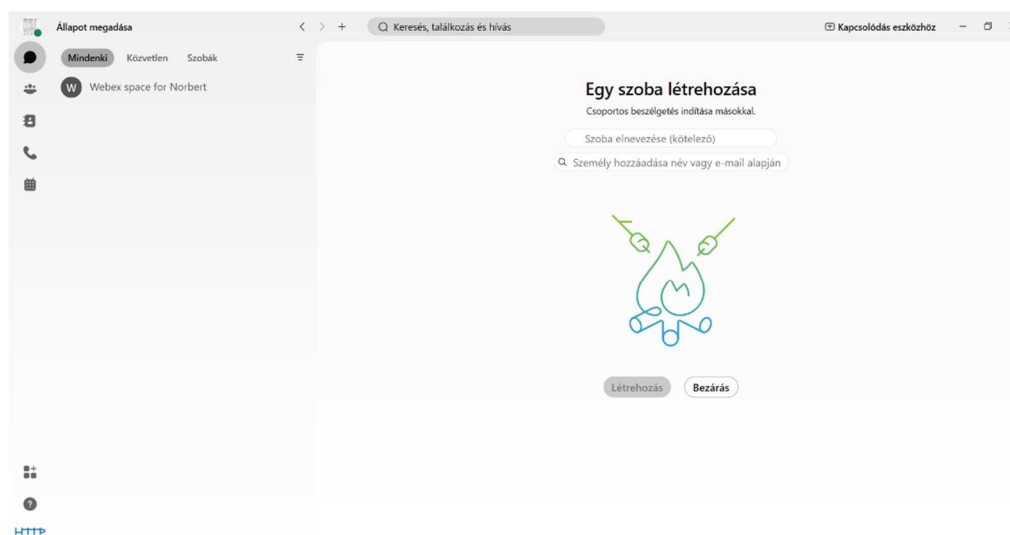
A Teamsben egy helyen képesek vagyunk értekezletet létrehozni, táblavázlatot a beépített OneNote szoftver segítségével és az órai feladatokat fel tudjuk tölteni a belső tárhelyre, amely a csoportban külön fülön van kialakítva.

Ehhez hasonló integráltsággal a későbbieknek a Webex is kísérletezett, először a Webex Teams nevű alkalmazással, a jelenlegi frissítéssel pedig összeolvasztották a Webexet és a Webex Teams-t.



42. ábra: Microsoft Teams részlet - saját ábra

Mint látható a Teams-en baloldali menüsorban lehetséges feladatkiadás, osztályzás, és a felső menüsorban pedig el tudunk helyezni extra kiegészítőket, mint háttértár a tananyagoknak, órai jegyzetek számára OneNote munkafüzeteket, formákat – amiket igazából számonkérésre tudunk használni.



43. ábra: Cisco Webex legfrissebb asztali felülete - saját ábra

Számomra az online oktatás esetén még hasznos volt a TeamViewer nevű applikáció. A TeamViewer segítségével távoli asztal elérésben tudtam segítséget nyújtani a hallgatóknak az elakadásokban, illetve a szakmai vizsgájukat megelőzően, az otthon kialakított házi PLC laborban távoli asztali eléréssel tudták tesztelni az év végi projektmunkájukat.

3.3. Oktatóanyag alkalmazása

A tananyag alkalmazásához választott osztály csoportprofilja

A csoport létszáma 14 fő. A csoport része egy vegyes felépítésű osztálynak, mely 14 Villamosipar és elektronika ágazatos és 16 Sport ágazatos tanulóból áll. Az elektronikai szakmacsoportos része 2 csoportra van felosztva 6-8 fővel. 5. éve vannak egy osztályban, részben helybeliek, részben pedig a környező agglomerációból járnak be.

Képességeiket és motivációjukat figyelembe véve vegyes a társaság. Többen is nehezen tanulnak, vagy kezd elmenni a kedvük a szakmától (a COVID alatt a szakmától, illetve az egész létezésről elment a kedvük, ők voltak az első COVID-érettségizők), így minél több motivációt szükséges beleépítenem a tematikus tervükbe. Sok érdekességgel megfűszerezett órák vannak tervben, amolyan mézesmadzag jelleggel, hiszen a korábbi években inkább a száraz tananyag megtanulása volt a fő feladat. Mivel ez már egy gyakorlatiasabb tantárgy, így megpróbálok minél életszerűbb helyzeteket, példákat felhozni, hogy meghozzam újra a kedvüket a szakmához.

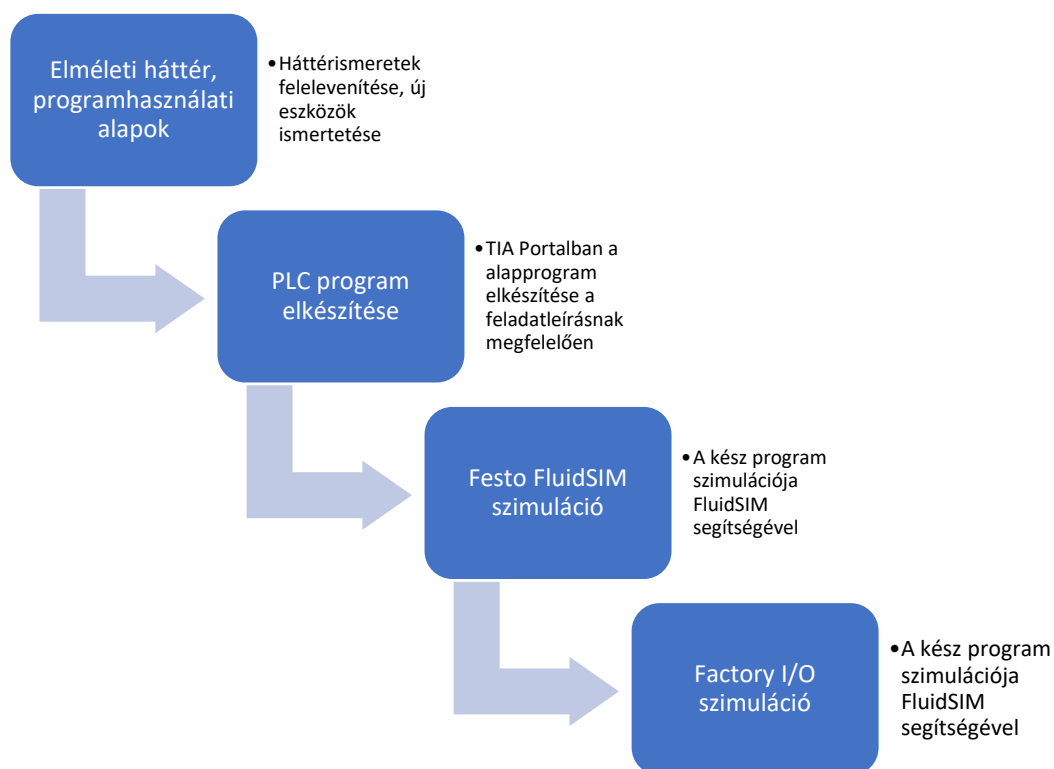
Ezen tantárgyi keretei között mind a kettő csoporttal hetente 3/6 órában találkozom (a tárgy teljes neve PLC és szimuláció gyakorlat, és a szimulációs órát párban tartottuk egy másik kollégával (Ő az analóg szimuláció részét, én pedig a digitális)), A szakmai kerettantervet tekintve ez már 11.-ben tanult tantárgy lehetett volna, de az osztály nem választotta, hogy 12. év végén részsakképesítést szerezzen, mint PLC programozó, így a kerettanterv erre szánt óraszámát szét lett bontva heti 2 óra irányítástechnika gyakorlatra, és 3,5 óra elektronika és elektrotechnika gyakorlatra. mivel így plusz egy évet nyertünk az irányítástechnika oktatására.

A tervezett oktatóanyagot 3 hetes intervallumra bontottam szét, így 3x6 órában tudtak vele foglalkozni, melyet egy tematikus tervben 2. számú mellékletként megtalálható a dolgozat mellékletei között.

Első héten az elméleti háttérrel néztük át, és ismételtük a szenzorok, aktuátorok működését, amelyek a feladathoz szükségesek, és elmélyedtünk a programhasználat ismételtesében - mivel az alkalmazást rég használták.

Második héten megírtuk a teszteléshez szükséges alpprogramot (mely el is vette az órák nagy részét), majd összeillesztettük a FluidSIM-es alpprogrammal és elvégeztük az első szimulációkat.

Harmadik héten, mivel a PLC program kész volt, így az első 3 órában megtanultuk a Factory I/O használatát alapszinten, majd összeállítva a méréshez szükséges gyártósort, leteszteltük a működő programot ebben az alkalmazásban is.



44. ábra: Folyamatábra a tananyag alkalmazásához – saját ábra

4. Tapasztalatok összegzése

Összességében a feladat megtervezése és megvalósítása igen kellemes időtöltés volt, mivel egy teljesen új program – a Factory I/O – működését kellett kiismerni.

A TIA Portal számomra már ismerős program volt, de sajnos kisebb hátulütő volt, hogy nem teljesen rendelkeztek az iskolai gépek a megfelelő erőforrásigénnyel, így egy-egy modul betöltése kicsit időigényesebb volt, és a szobahőmérsékletet is nagyban emelte a gépek túlhajtása.

A FluidSIM régi jó ismerősként mutatkozott be a hallgatónál, a korábbi évek tapasztalataira visszaemlékezve, ez volt az, ami a leginkább megmaradt és a legjobban tudták használni, így túl sok fejtörést nem okozott. Sok új kiegészítő modult megismertek a hallgatók, illetve betekintést nyertek az OPC szerverek működésébe, az adatirányokra és az eszközök közötti kommunikációba.

A Factory I/O az egyszerűsége és a látványossága végett abszolút sikert aratott a hallgatónál, de – megfelelő erőforrású gépek hiányában – nem az elsődlegesen használt szoftver lesz, mivel a bonyolultabb gyártósorok nagyobb erőigényűek is, amely könnyen a gépek összeomlásához vezethet.

A csapat hangulata a feladatfeldolgozás során kissé hullámzó volt, talán az ismétlés része esetén a lejtő aljáról indult motivációban, de amint betekintheztek az újonnan ismerttetett programokba, alig lehetett őket kizavarni a teremből. Több hallgató is emiatt a későbbiekben tervezi, hogy automatikai szakirányon tanul tovább, vagy mechatronikán, mivel sokkal jobban el szeretnék mélyíteni a tudásukat az ilyen komplex rendszerek tervezésében, és megvalósításában.

Számomra mindvégig motiváló volt a csoportdinamika és a gyakorlatok hangulata, mivel pozitív visszacsatolást kaptam a hasznosságról, és hogy a jövőben is tervezzek ilyen oktatóanyagokat.

Összefoglalás

A diplomamunkám készítésénél nagy hangsúlyt fektettem arra, hogy egy általam már nagy mértékben kiismert módszert/módszertant mutassak be. Mint fiatal oktató, a korszerű eszközhasználat implementálása nem volt nagy kihívás. Az online oktatás bevezetése egy tökéletes lehetőséget teremtett meg, hogy ezt élesben is tudjam tesztelni.

A dolgozatban áttekintést adtam a magyarországi oktatási rendszer történetéről, illetve egy részletesebb bemutatót a szakképzési rendszerről. Ismertetésre kerültek a szakképzést szabályozó dokumentumok (szakképzési törvény, Országos Képzés Jegyzék, Szakmai és Vizsgakövetelmény, Szakmajegyzék). Szó esett a moduláris és komplex vizsgáztatási sémáról. Említésre került a 2020-ban bevezetésre kerülő Technikumi szakképzési rendszer, illetve az ezt szabályozó alapidokumentumok (KKK – Képzési és Kimeneti Követelmények, PTT – Programtanterv).

Ezután bemutatásra került a fejlesztéshez használt eszköz, a programozható logikai vezérlő (PLC – Programmable Logical Controller) történeti, hardverfelépítési és programozási szempontból.

Következő lépésként bemutattam, hogy miképp lehet PLC-t oktatni fizikai eszköz vagy technológiai nélkül (szimuláció, emuláció), majd egy szimulációs feladaton keresztül megmutattam, hogy miként is működik ez. A kitalált feladatot először Festo FluidSIM segítségével valósítottam meg, majd Factory I/O-ban, részletesen levezetve. Mindkettő esetben a Siemens TIA Portal-t hívtam segítségül, amelynek a szimulátorával működésbe tudtam hozni a virtuális eszközeimet.

Végül körülírtam, hogy akár jelenléti, akár online oktatás során miképpen is lehetne ezt a fajta módszert implementálni a gyakorlati oktatásban. Jellemeztem az ideális laborkörülményeket, elrendezéseket, oktatási módszereket, amik szükségesen az ilyen fajta gyakorlatnál.

Végül levontam a konklúziót az egész módszerről, és összegeztem a tapasztalataimat, amiket szerencsére sikerült összeszedni az elmúlt félévek során.

Összességében a diplomamunka készítése nem tartogatott nagy kihívást, csak leginkább a szükséges karakter és oldalszám elérése. A korábban tanult fogalmak, módszerek felelevenítése pedig pozitív hatással lesz a záróvizsgára való felkészülésben.

Remélem a jövőben minél jobban elterjedt lesz a virtualizáció, mivel ez sokat tud segíteni az iskolák gyakorlati oktatási színvonalának emelésén.

Az új technikumi rendszerben a duális partnernél megtapasztalják előben azokat a kísérleteket, tervezéseket, amikkel az iskolai napok során csak szimulációval tudtunk bemutatni. És így legalább felkészülve tudjuk a hallgatókat küldeni a duális képzésre, mert tulajdonképpen látták már az eszközöket, készülékeket, csak tapintani nem tapinthatták.

Summary

During the preparation of my thesis, I placed great emphasis on the presentation of a method / methodology that I have already become very familiar with. As a young instructor, implementing state-of-the-art device use was not a big challenge. The introduction of online education has created a perfect opportunity for me to test this in live.

In the thesis I gave an overview of the history of the Hungarian education system and a more detailed presentation of the vocational training system. The documents regulating vocational training (Vocational Training Act, National Training Register, Vocational and Examination Requirements, Vocational Register) were presented. There was comparison of a modular and complex examination scheme. Mention was made of the Technical Vocational Training System to be introduced in 2020 and the basic documents regulating it (KKK - Training and Output Requirements, PTT - Curriculum).

The tool used for the development, the Programmable Logical Controller (PLC), was then presented in terms of history, hardware architecture and programming.

As a next step, I showed how to teach a PLC without a physical device or technology (simulation, emulation), and then I showed through a simulation task how it works. I realized the task first with Festo FluidSIM and then in Factory I / O, derived in detail. In both cases, I called the Siemens TIA Portal for help, with which I was able to activate my virtual devices.

Finally, I outlined how this type of method could be implemented in hands-on teaching, either in attendance or online teaching. I have characterized the ideal laboratory conditions, layouts, teaching methods required for this type of practice.

Finally, I drew a conclusion about the whole method and summarized my experiences, which I fortunately managed to gather over the past semesters.

Overall, the preparation of the thesis was not a big challenge, only the achievement of the required character and number of pages. The revival/revision of previously learned concepts and methods will have a positive effect on the preparation for the final exam.

I hope that virtualization will become more widespread in the future, as it can do a lot to raise the standard of practical education in schools.

In the new technical education system, the dual partner the students will experience the experiments and designs at the VET partner, that we could only present with simulation during the school days. And so, we can at least we can boldly send them to VET partner, because they have actually seen the tools and devices, except they just couldn't touch them.

Irodalomjegyzék

Dr. Cselényi, J. (2004). *Logisztikai rendszerek I.* Forrás: Miskolci Egyetem:
<http://web.alt.uni-miskolc.hu/anyagok/logrend1/szimulacio.pdf>

Innovációs és Technológiai Minisztérium. (2019.). A 2019. évi szakképzési törvény.

Innovatív Képzéstámogató Központ. (é.n.). Forrás: KKK és PPT felépítése:
<https://szakkepzes.ikk.hu/kkk-ptt>

Juhász, R. (é.n.). *Ismerkedés a PLC-vel.* Forrás: <https://docplayer.hu/35747021-Ismerkedes-a-plc-vel.html>

Kiss, G. (2010). Módszertani összefoglaló - 2. Modulrendszerű, kompetencia alapú képzés.

Lükő, I. (2019.). Nemzetközi kitekintés az első szakképzési törvény korszakáról. *Opus et Educatio 6. évfolyam 1. rész*, 3-14.

Pintér, J. (2010). Gépipari automatizálás ea. 5-6. Széchenyi István Egyetem,
http://rs1.sze.hu/~pinter/NGB_AJ005_2%20G%E9pip_autom%20II/KJ_2010_j%F3/G%C9PIPARI%20AUTOMATIZ%C1L%C1S%20II_5-6e_PLC.ppt.

Rosencrance, L. (2021. szeptember). *TechTarget.* Forrás:
<https://www.techtarget.com/searchsecurity/definition/sandbox>

Sanda, I. D. (é.n.). A modern szakképzés rendszerének kialakulása és professzionalizálódása a késő középkortól a XVIII. század végéig.

SiemensAG. (2009). *TIA Portal STEP 7 Basic V10.5.* Siemens AG.

Simonics, I., & Makó, F. (2016). *Az elektrotechnika tanításának módszertana.* Budapest: Typotop Kft.

Ábrajegyzék

1.	ábra:	Siemens	S7-1500-as	termékcsalád	Forrás:	
					https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500.html	6
2.	ábra:	PLC-k belső felépítése	Forrás:	https://slideplayer.hu/slide/2026056/	3. dia	7
3.	ábra:	Festo CECC-LK kompakt PLC	Forrás:	http://www.festo.com		8
4.	ábra:	Siemens S7-300 moduláris PLC	Forrás:	https://slideplayer.hu/slide/13464799/	9. dia	8
5.	ábra:	Céhrendszer hierarchiája	Forrás:	(Sanda, é.n.)		10
6.	ábra::	Szimuláció	illusztrálása:	Forrás:	https://www.vstepsimulation.com/assets/uploads/2019/01/VSTEP-Building-Blocks.png	22
7.	ábra:	Szimulációs modellek osztályozása:	Forrás:	http://web.alt.unimiskolc.hu/anyagok/logrend1/szimulacio.pdf		23
8.	ábra:	Festo Veep és EasyPort modulok	Forrás:	https://manuf.bme.hu/?page_id=262		23
			Forrás:	https://www.festo-didactic.com/		23
9.	ábra:	Létradiagram programrészlet – saját ábra				27
10.	ábra:	Funkcióblokk-diagramos programrészlet – saját ábra				27
11.	ábra:	ST program részlet	Forrás:	https://isd-soft.com/nl/tech_blog/plc-programming-languages-short-overview/		27
12.	ábra:	Utasításlista programrészlet – saját ábra				28
13.	ábra:	SFC programrészlet – saját ábra				28
14.	ábra:	CFC programrészlet – saját ábra				29

15. ábra: Siemens TIA portal	Forrás: https://traininginplc.com/courses/siemens-tia-portal/	29
16. ábra: STEP 7 kezelőfelület részlete – saját ábra		30
17. ábra: WinCC - HMI tervezői felület – saját ábra		31
18. ábra: PLCSim felület, S7-300as PLC-nél – saját ábra		31
19. ábra: Siemens Startdrive illusztráció	Forrás: https://new.siemens.com/hu/hu/termekek/automation/drives/selection-and-engineering-tools/sinamics-startdrive-commissioning-software.html	32
20. ábra: Factory I/O	Forrás: https://docs.factoryio.com/	33
21. ábra: Festo FluidSIM	Forrás: https://fluidsim.software.informer.com/Let%C3%B6lt%C3%A9s/	33
22. ábra: I/O címzés – saját ábra		35
23. ábra: Siemens Simatic S7-300 I/O modul	Forrás: https://www.indiamart.com/proddetail/siemens-s7-300-cpu-315-2-pn-dp-standard-24301726733.html	35
24. ábra: Új projekt létrehozása – saját ábra		37
25. ábra: PLC kiválasztása – saját ábra		37
26. ábra: A kész hardverkonfiguráció – saját ábra		38
27. ábra: A kész TAG Table – saját ábra		40
28. ábra: Érintkező eszköztár – saját ábra		41
29. ábra: Coil funkció hozzárendelése – saját ábra		42
30. ábra: Elnevezés nélküli network - saját ábra		42

31. ábra: A start rendszer logikai felépítése – saját ábra	42
32. ábra: Új network beszúrása.....	43
33. ábra: Futószalagok működtetése – saját ábra.....	43
34. ábra: A szenzorok és jelzőlámpák logikai kapcsolata - saját ábra	44
35. ábra: FluidSIM szimuláció kapcsolási rajza – saját ábra	44
36. ábra: EZopc szoftver beállítása – saját ábra.....	45
37. ábra: A szimulálandó gyártósor – saját ábra	46
38. ábra: Vezérlő beállítása – saját ábra	47
39. ábra: C216-os labor jelenleg a frontális kialakítással és az egyoszlopos mérőállásokkal	49
40. ábra: Kombinált frontális és csoportmunka elrendezés Forrás: Saját ábra – saját tervezés	50
41. ábra: Cisco Webex oktatói nézőpontból – saját ábra.....	52
42. ábra: Microsoft Teams részlet - saját ábra	53
43. ábra: Cisco Webex legfrissebb asztali felülete - saját ábra.....	53
44. ábra: Folyamatábra a tananyag alkalmazásához – saját ábra.....	55

Táblázatok

1. táblázat: Eszköz és konfigurációs lista	34
2. táblázat: Allokációs lista	36
3. táblázat: Logikai szintek és magyarázatuk	39

Mellékletek listája

- I. Tematikus terv
- II. A feladathoz tartozó PLC program
- III.

Tematikus terv

A pedagógus neve: Berecz Lajos Norbert

Szakmai ágazat: XI. Villamosipar és elektronika ágazat

Tantárgy: PLC és szimuláció gyakorlat

A tanulási-tanítás egység témája: Egyszerű gyártósor programozása

A tanulási-tanítási egység cél- és feladatrendszere: Alapvető villamos berendezések ismertetése, működésének bemutatása. Az elméleti alapfogalmak összekapcsolása a gyakorlattal. PLC programozási ismeretek felelevenítése

Tantárgyi kapcsolatok: Elektrotechnika, elektronika

Óra	A téma órákra bontása	Didaktikai feladatok	Fejlesztési területek (attitűdök, készségek, képességek)	Ismeretanyag (fogalmak, szabályok stb.)	Módszerek, munkaformák	Szemléltetés, eszközök
1.	PLC alapok ismétlése	Korábban tanult ismeretek átismétlése	Emlékezés		Frontális, kérdve kifejtős, „vita”	Projektor, PowerPoint prezentáció
2.	Boole algebrai alapismeretek (Változók, Negáció, ÉS és VAGY kapcsolat)	Ismeretszerzés, érthetőség	Ráismerő készség, képzelet	Bináris jelek, igazságtábla, kétértékűség, Boole-algebrai változók	Frontális, magyarázat és szemléltetés, előadás, kiértékelés	Projektor, PowerPoint prezentáció, tábla
3-4.	Szenzorok, aktuátorok ismétlése	Ismeretszerzés, érthetőség	Ráismerő készség, képzelet	Működési alapok, kapacitív, induktív és optikai érzékelők	Frontális, kérdve kifejtős, szemléltetés	Projektor, PowerPoint prezentáció, tábla, YouTube videó
5-6.	Festo FluidSim alapjai	Ismeretszerzés, érthetőség	Ráismerő készség, képzelet, digitális képességek alkalmazása, szimulációs programban való jártasság fejlesztése	Szimulációs környezet, szimuláció, alkatrésztípusok, szimuláció előnyei	Frontális, kooperatív	Projektor, Festo FluidSim

7-10.	PLC program elkészítése	Korábban tanult ismeretek átismétlése	Emlékezet, ráismerő készség, képzelet	Logikai kapcsolatok, allokációs lista, létradiagram	Kooperatív, csoportos	Projektor, TIA Portal
11-12.	FluidSIM szimuláció futtatása	Ismeretek alkalmazása, visszacsatolás	Kapcsolatteremtő készség fejlesztése, szimulációs program alkalmazása	Szimulációs program használata	Kooperatív, csoportos	Projektor, Festo FluidSim
12-14.	Factory I/O alapok	Szimulációs programban való jártasság fejlesztése	Kapcsolatteremtő készség fejlesztése, szimulációs program alkalmazása	Szimulációs program használata	Kooperatív, csoportos	Projektor, Factory I/O
15.	Factory I/O szimuláció futtatása	Ismeretek alkalmazása, visszacsatolás	Rendezés, kommunikáció	Szimulációs program használata	Csoportos, kérdve kifejtős	Tábla, tanulók jegyzetei
14.	Számonkérés	Rendszeresség, tartósság	Rendezés, kommunikáció	Az elméleti órákon elsajátított tananyag rendszerezése	Csoportos, kérdve kifejtős	Tábla, tanulók jegyzetei

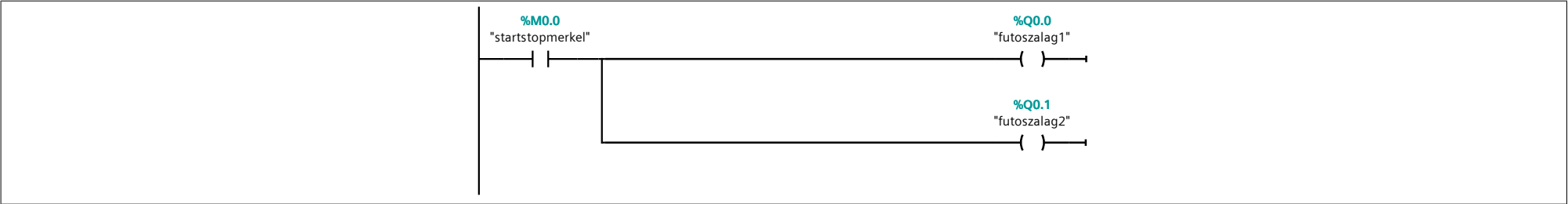
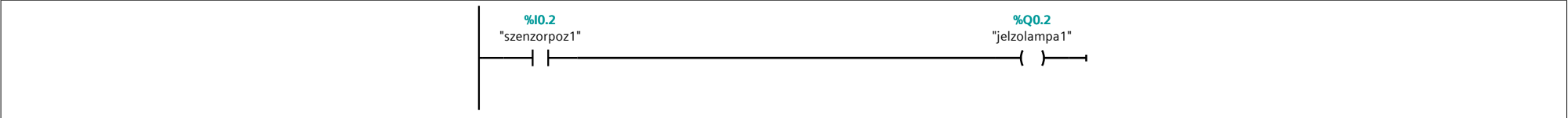
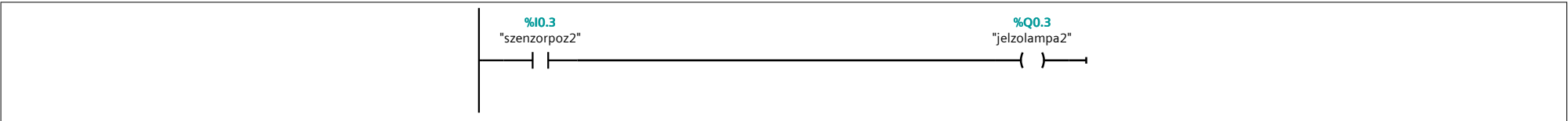
futoszalag_anyagerzekeles / PLC_1 [CPU 315-2 PN/DP] / Program blocks

Main [OB1]

Main Properties					
General					
Name	Main	Number	1	Type	OB
Numbering	Manual			Language	LAD
Information					
Title	"Main Program Sweep (Cycle)"	Author		Comment	
Version	0.1	User-defined ID		Family	
Main					
Name	Data type	Offset	Default value	Comment	
▼ Temp					
OB1_EV_CLASS	Byte	0.0		Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)	
OB1_SCAN_1	Byte	1.0		1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)	
OB1_PRIORITY	Byte	2.0		Priority of OB Execution	
OB1_OB_NUMBR	Byte	3.0		1 (Organization block 1, OB1)	
OB1_RESERVED_1	Byte	4.0		Reserved for system	
OB1_RESERVED_2	Byte	5.0		Reserved for system	
OB1_PREV_CYCLE	Int	6.0		Cycle time of previous OB1 scan (milliseconds)	
OB1_MIN_CYCLE	Int	8.0		Minimum cycle time of OB1 (milliseconds)	
OB1_MAX_CYCLE	Int	10.0		Maximum cycle time of OB1 (milliseconds)	
OB1_DATE_TIME	Date_And_Time	12.0		Date and time OB1 started	
Constant					

Network 1:



Totally Integrated Automation Portal		
Network 2:		
		
Network 3:		
		
Network 4:		
		
Network 5:		
		

Totally Integrated Automation Portal		
Network 6:		
	%I0.4 "szenzorpoz3" %Q0.4 "jelzolampa3" ()	
Network 7:		
	%I0.5 "szenzorpoz4" %Q0.5 "jelzolampa4" ()	
Network 8:		
	%I0.6 "szenzorpoz5" %Q0.6 "jelzolampa5" ()	