



**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
**Gépészmérnöki Kar**  
Mechatronika, Optika és Műszertechnika Tanszék

---

## **Tudományos Diákköri Dolgozat**

### **PLC vezérlése és felügyelete Interneten keresztül**

Készítette:

**Nagy Zsigmond**  
OPBRT0 (GÉK, V. évf.)

Konzulens:

**Dr. Szabó Tibor**  
Mester tanár

# Tartalomjegyzék

<b>1</b>	<b>Bevezetés, a TDK dolgozat célja</b> .....	<b>3</b>
<b>2</b>	<b>Az Internet</b> .....	<b>4</b>
2.1	Története .....	4
2.2	Felépítése .....	6
2.3	Használata .....	8
<b>3</b>	<b>A TCP/IP hivatkozási modell</b> .....	<b>9</b>
3.1	Az internetréteg.....	9
3.2	A szállítási réteg.....	10
3.3	Az alkalmazási réteg.....	11
3.4	A hoszt és a hálózat közötti réteg .....	11
<b>4</b>	<b>A hálózati réteg és a szállítási réteg kapcsolata</b> .....	<b>11</b>
4.1	Az IP-protokoll .....	12
4.2	IP-címek.....	13
4.3	A TCP-protokoll .....	14
<b>5</b>	<b>A Világháló (World Wide Web)</b> .....	<b>15</b>
<b>6</b>	<b>A HTML dokumentumokról</b> .....	<b>17</b>
<b>7</b>	<b>Programozható vezérlők</b> .....	<b>18</b>
7.1	PLC-k funkcionális felépítése.....	19
7.2	A PLC-k kommunikációs rendszere .....	20
7.3	Hálózati kommunikáció .....	22
<b>8</b>	<b>FEC PLC</b> .....	<b>24</b>
<b>9</b>	<b>EasyVeep</b> .....	<b>26</b>
<b>10</b>	<b>Példafeladat Internetes vezérlésre és felügyeletre</b> .....	<b>28</b>
10.1	PLC-program.....	30
10.2	Web-es felület.....	34
<b>11</b>	<b>Végszó</b> .....	<b>42</b>
<b>12</b>	<b>Irodalomjegyzék</b> .....	<b>43</b>

# 1 Bevezetés, a TDK dolgozat célja

Világszerte gyorsan terjed az on-line kommunikáció és kereskedelem, egyre nagyobb kereslet mutatkozik az olyan megoldások iránt, amelyek lehetővé teszik az internetes kapcsolat megosztást. A katonai indíttatású Internet sikere, és a személyi számítógépek fejlődése mára már legyőzte az ipar idegenkedését e technológiák iránt. Ezzel az ipari folyamatirányítás kommunikációs lehetőségei soha nem látott mértékben bővültek. Az Ethernet-hálózat (ennek vezetékes illetve rádiós változata), és ezen keresztül az Internet, az iparban is egyre nagyobb jelentőséget követel magának, mint kommunikációs fizikai réteg. A hálózat kiépítése kevésbé költségigényes, megbízható, és könnyen karbantartható. Egy tömegméretben, olcsón gyártott, és a mindennapi felhasználó számára is átlátható, kezelhető eszköz.

Nem kivétel ez alól a PLC, Programmable Logic Controller (programozható logikai vezérlő) sem, minden perspektivikus PLC-nek van Ethernet hálózati csatlakozója. A fejlődés során a programozható vezérlők funkciói nagymértékben közeledtek a számítógépek funkcióihoz. Így mára a PLC olyan ipari számítógépnek tekinthető, amely speciális hardveregységei és felhasználói programja révén a technológiai folyamatok tárolt programú vezérlése, szabályozása és intelligens kommunikációs felülete révén hierarchikus és/vagy osztott folyamatirányító rendszerek létrehozására alkalmas.

A rendszer elemei közötti adatcsere a számítógépes hálózatokban leggyakrabban használt, TCP/IP protokoll szerint történhet. Egyrészt ez az Internet nyelve, másrészt ezzel kis és nagy hálózatok is kiszolgálhatóak. Elég rugalmas, és rengeteg eszköz létezik, ingyenes és fizetős is, amivel a TCP/IP-n a legkülönbözőbb szolgáltatások valósíthatóak meg, illetve vehetők igénybe. A beépített TCP/IP kommunikáció részeként a PLC-k beágyazott Web szervert tartalmaznak, melyek megteremtik azt a lehetőséget, hogy minden egyes terepi eszköznek saját közvetlen honlapja legyen. A közvetlenül elérhető honlap egyik fontos alkalmazása, hogy a kezelő számára lehetővé teszi olyan folyamatok ellenőrzését és irányítását, amik elosztottak, fizikailag távol levő (telep)helyeken történnek. Például erőművek, víz-, gáz- és áramszolgáltatás, víz- és szennyvízkezelés, forgalomirányítás, környezetvédelem, mezőgazdaság és különböző ipari folyamatok esetén.

A világháló komplex rendszerében az egymással összeköttetésben lévő egyes elemek működési elvének megismerése után feladatunk volt egy olyan oktatási segédlet összeállítása, melyből a hallgatók könnyen és egyszerűen megértik az ezen területhez kapcsolódó Internet adta lehetőségeket, majd ezt alkalmazni is tudják a programozható vezérlők működtetésében.

## **2 Az Internet**

### **2.1 Története**

A gyökerek a hatvanas évekig nyúlnak vissza, a történet katonai fejlesztések civil szférába való átszivárgásával kezdődött. Abban az időben merült föl ugyanis az USA-ban egy kevésbé sebezhető számítógép-hálózat szükségessége, amelynek egy esetleges atomtámadás után megmaradó részei működőképesek maradnak. Dwight Eisenhower elnök - a szovjetek űrversenybeli sikereit ellensúlyozandó, a Szputnyik 1 fellövésének hírére - elrendelte a Defence Advanced Research Project Agency (DARPA) beindítását, amely a kutatásokat azután finanszírozta. Kidolgoztak egy többközpontú, csomagkapcsolt (ahol az adatok továbbítása kisebb csomagokban történik) hálózati kommunikációs rendszert (az NCP protokollt), mely a mai TCP/IP szabvány őseinek tekinthető. Ezen az elven kezdett működni 1969-ben az ARPANET, és a katonai felhasználásokon kívül a csomagkapcsolt adattovábbítás további kutatásra szolgált, de egyes egyetemek, katonai bázisok és kormányzati laboratóriumok kutatói is használták elektronikus levelezésre, fájlok cseréjére és távoli bejelentkezésre egymás számítógépei között. 1972-ben megszületett az első e-mail program. 1974-ben jelent meg először az "Internet" kifejezés, egy a TCP protokollról szóló tanulmányban. 1983-ban azután, hogy az addig szigorúan ellenőrzött az ARPANET-ből MILNET néven leválasztották a hadászati szegmenst, megszületett a mai fogalmaink szerinti Internet.

A National Science Foundation felismerte hogy a hálózat döntő fontosságú lehet a tudományos kutatásban, ezért igen nagy szerepet vállalt az Internet bővítésében. 1985-86 között építették ki az NSF 6 szuperszámítógép-központját, és az így kialakult hálózatot (mely az NSFNET nevet kapta) összekapcsolták az ARPANET-tel. Az NSFNET több bővítés után (optikai kábeleket fektettek le, újabb vonalakat hoztak létre, melynek eredményeként a hálózat sebessége a kezdeti 56 kbit/s-ról 45 Mbit/s-ra emelkedett) ma is az USA domináns gerinchálózata. Az ARPANET formálisan 1989-ben szűnt meg, hogy helyét átadja a fejlettebb gerinchálózatoknak. Az NSFNET mellett jelentős részben már magáncégek hálózatain folyik

a kommunikáció (AT&T, MCI, UUNET, Sprint, stb.). Az azóta eltelt években több száz különálló hálózaton több tízezer számítógépet kapcsoltak a folyamatosan növekvő Internethez és ez a növekedés napjainkban is tart. Az Internet adatforgalma manapság óriási. Az NSFNET gerinchálózatán 1995 áprilisában mért 13 404 656.5 Mbyte adatforgalmának megoszlását mutatja a jelentősebb szolgáltatások szerint az 1. ábra [2].

Szolgáltatás	Adatforgalom (Mbyte)	%
WWW	3 518 742	26.25
FTP	2 886 742	21.53
NNTP	1 160 496	8.66
SMTP	620 450	4.63
Telnet	339 404	2.53

**1. ábra.** [2] Adatforgalom megoszlás

A 80-as évek végén az NSFNET-hez hasonló elvek alapján számos országban szerveződtek gerinchálózatok. Ezek mindenekelőtt a hatalmas információs és számítástechnikai erőforrásokkal rendelkező NSFNET-hez igyekeztek csatlakozni, de gyakran egymással is kiépítették közvetlen kapcsolataikat. Az utóbbi években a távközlési cégek, kommunikációs vállalatok meglátták az üzleti lehetőséget az Internet technológiájú számítógép-hálózatokban, ill. a hozzájuk kapcsolódó alkalmazásokban (pl. számítógépek, adatbázisok távoli elérése, elektronikus levelezés, adatállományok átvitele, szöveg-, kép-, hanginformációk integrált továbbítása, stb.) így megjelentek az ilyen szolgáltatásokat kínáló üzleti vállalkozások, ill. ezek saját gerinchálózatai. Az Internet talán legfontosabb szervező, összefogó ereje az Internet Society (ISOC). A társaság nyílt, tagja lehet bármely szervezet vagy magánszemély. Célja az Internet technológiával történő információcsere összehangolása, fejlesztése. Az ISOC által felkért, nagy szakmai tekintéllyel rendelkező önkéntesekből áll az Internet Architecture Board (IAB) melynek feladata hogy állást foglaljon alapvető stratégiai kérdésekben, felelős a szabványok elfogadásáért, ill. a szabványosítást igénylő kérdések meghatározásáért és az Internet címzési rendszer karbantartásáért.

Az Interneten nincsen központ, nincs "egy" központi gép. Minden, a hálózatra kötött gép egyszerre fő- és alállomás. Az Internet tehát olyan elméleti szerveződése a számítógépeknek és hálózatoknak, amelynek bármely pontja képes kapcsolatot teremteni bármely másik

pontjával. Az Internet tulajdonképpen nem is hálózat, hanem különféle hálózatok óriási gyűjteménye, amely néhány közös protokollt használnak, és néhány közös szolgáltatást nyújtanak.

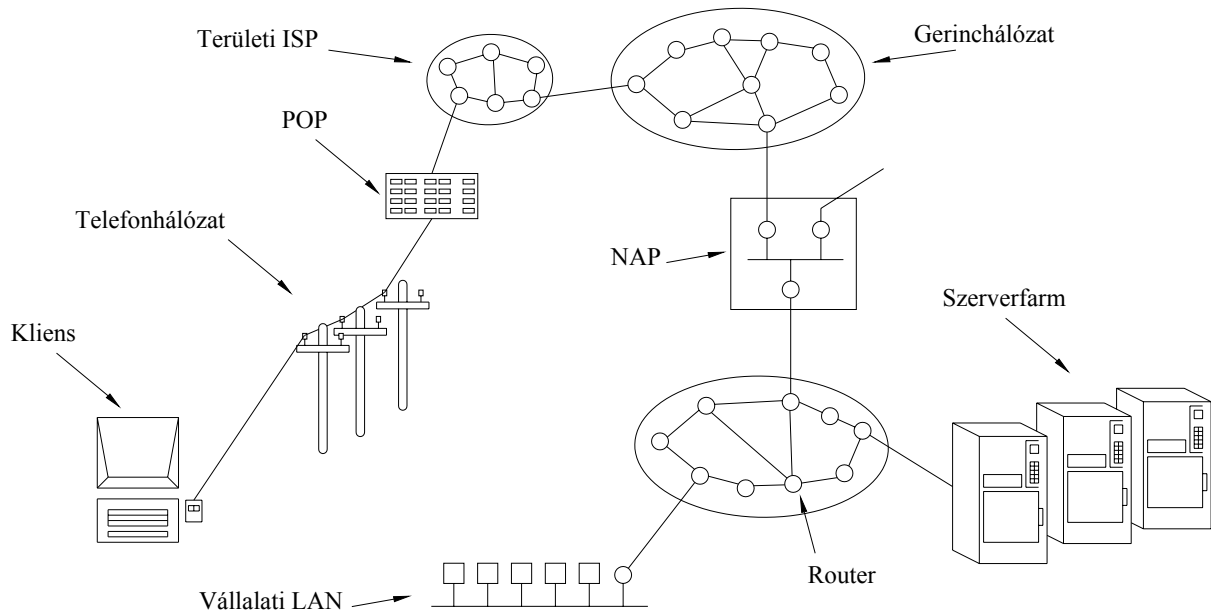
Az eredetileg katonai és szakmai célokra tervezett hálózat gyorsan általános kommunikációs, információtovábbító médiává vált, majd maguktól adódtak az emberi kapcsolatteremtés újabb, sokszor korábban soha nem ismert formái. Az eredetileg elsősorban oktatási intézményekben elérhető hálózatra egyre több intézmény, szervezet, cég kapcsolódott, a szolgáltatást hamarosan a nagyközönségnek is felkínálták. Az elektronikus levelezés (e-mail) forradalmát gyorsan követte a "hírcsoportok" (newsgroup) forradalma, majd az egyre újabb és sokoldalúbb információkereső- és továbbító eszközök elterjedése, amelyek 1992-ben úgy tűnik, egy "természetes" végponthoz, a WWW-hez (World Wide Web) vezettek: innentől a számítógépekhez nem értő laikusok is könnyedén, minden tanulás nélkül navigálhatnak az Internet óceánján. A WWW az európai részecskefizikai kutatóintézet, a CERN szülötte. Tim Berners-Lee és Robert Cailliau elgondolásai alapján olyan rendszer született, ami az elmúlt tíz évben forradalmasította a kommunikációt. 1998-ra mintegy 300 millióra nőtt a web dokumentumok száma, és a Web naponta további másfél millióval bővül.

A gyors elterjedésből és egyszerű kezelhetőségből eredeztethető a korábban a felhasználó szempontjából ingyenes, nonprofit hálózat kommercializálódása is: az óriási és egyre növekvő, többé már nem csak profi számítástechnikusokból álló Internet közösség hatalmas üzleti lehetőségeket rejt, sőt az emberek közötti kommerciális kapcsolatok teljesen új formáit teszi lehetővé. Az Internet jelenlegi felhasználóinak száma ismeretlen, de az biztos, hogy ez a szám az egész világon összesen több százmillió főre rúg, és valószínűleg nemsokára eléri az egymilliárdot.

## **2.2 Felépítése**

A 2. ábra [5] a „ma Internet” egyszerűsített képét mutatja. Van egy kliens, aki éppen otthon van. Tegyük fel, hogy a kliens felhívja az ISP-jét (Internet Service Provider, internetszolgáltató) egy kapcsolt telefonvonalon a 2. ábrán, látható módon. A modem egy kártya a számítógépben, amely a számítógép által előállított digitális jeleket analóg jelekké alakítja annak érdekében, hogy azok akadály nélkül átmehessenek a telefonhálózaton. Ezeket a jeleket az ISP POP-jéhez (Point of Preference, szolgáltatási pont) továbbítják, ahol ezeket a jeleket eltávolítják a telefonhálózatból, és az ISP területi hálózatára játsszák át! Ettől a ponttól

a rendszer teljesen digitális és csomagkapcsolt. Amennyiben az ISP nem más, mint a helyi telefontársaság, a POP valószínűleg abban a telefonközpontban található, ahová a felhasználó telefonvezetéke befut. Ha az ISP nem a helyi telefontársaság, akkor a POP néhány kapcsolóközponttal arrébb is lehet.



**2. ábra.** [5] Az Internet áttekintése

Az ISP területi hálózata az ISP által kiszolgált különféle városokban elhelyezett, egymással összekötött routerekből áll. Ha a csomag célja egy olyan hoszt, amelyet ugyanaz az ISP szolgál ki, akkor a csomag egyenesen a cél hoszthoz kerül. Egyéb esetekben a csomagot az ISP a saját gerinchálózati szolgáltatójának adja át. A lánc legtetején a nagy gerinchálózati szolgáltatók foglalnak helyet, olyan cégek, mint az AT&T vagy a Sprint. Ezek a cégek üzemeltetik a nagy nemzetközi gerinchálózatokat, több ezer fényvezető szálakkal összekötött router segítségével. A szerverfarmokat (vagyis másodpercenként több ezer weboldal kiszolgálására képes gépeket) üzemeltető nagyvállalatok és társzolgáltatók gyakran közvetlenül a gerinchálózatra kapcsolódnak. A gerinchálózatok üzemeltetői azzal segítik elő az ilyen közvetlen kapcsolatok megvalósítását, hogy úgynevezett szolgáltatói szállodákat (carrier hotels) hoznak létre, ahol helyeket lehet bérelni. Ezek a szállodák lényegében olyan berendezésszekrények, amelyek egy szobában vannak a routerrel, és így teszik lehetővé a rövid és gyors összeköttetést a gerinchálózat és a szerverfarm között.

Ha a gerinchálózat olyan csomagot kap, amelyet egy általa kiszolgált ISP-nek, vagy cégnek címeztek, akkor továbbküldi azt a célhoz legközelebbi routerhez, és ott adja le. Mivel

nagyszámú és változatos méretű gerinchálózat létezik a világon, ezért a csomagoknak esetleg egy versenytárs gerinchálózatba kell átmennie. Minden nagyobb gerinchálózat kapcsolódik a NAP-hoz (Network Access Point, hálózati hozzáférési pont), annak érdekében, hogy a csomagok a gerinchálózatok között át tudjanak járni. Lényegében egy NAP egy olyan szoba, amely tele van routerekkel, minden gerinchálózathoz tartozik legalább egy. A szobán belül egy LAN köt össze minden routert, lehetővé téve azt, hogy csomagot bármely gerinchálózatról bármely másikra továbbítani lehessen. A NAP-okban létrehozott kapcsolatokon kívül a nagyobb gerinchálózatok számos egyéb közvetlen összeköttetést tartanak fenn a routereik között. Ezt a megoldást magántovábbításnak (private peering) nevezik.

## **2.3 Használata**

Mit jelent tulajdonképpen „rajta lenni az Interneten”? A legegyszerűbben megfogalmazhatni, úgy lehetne, hogy egy gép akkor van rajta az Interneten, ha a TCP/IP-protokollkészletet használja, van saját IP-címe, és tud más gépeknek IP-csomagokat küldeni az Interneten át.

Az 1990-es évek kezdetéig az internetet főleg oktatási és kormányzati intézmények, valamint ipari kutatóintézetek használták. Egy új alkalmazás, a Világháló (mellyről bővebben az 5. fejezetben lesz szó) ezt a helyzetet gyökeresen megváltoztatta, és több millió új, az oktatástól független felhasználót vont be a hálózatba. A CERN-es fizikus, Tim Berners-Lee által kifejlesztett alkalmazás nem változtatott semmit a rendelkezésre álló eszközökön, hanem egyszerűen csak könnyebbé tette a használatukat. A NCSA (National Center for Supercomputer Applications) által írt Mosaic nevű böngészőnek köszönhetően lehetőség nyílt arra, hogy olyan szövegeket, képeket, hangokat, sőt akár még mozgóképeket is tartalmazó információs oldalakat hozzunk létre, amelyről más oldalakra is el lehet jutni. Ha rákattintunk egy linkre, pillanatok alatt az általa mutatott oldalon találjuk magunkat. Sok cég honlapja tartalmaz olyan linkeket, amelyek segítségével további oldalakra juthatunk el. Olyanokra, mint például termékmertető, árlisták, vásárlási feltételek, műszaki terméktámogatás, kapcsolattartás az alkalmazottakkal. Rövid időn belül rengeteg új oldal született; térképek, részvénytársi táblázatok, könyvtári katalógusok, rádió és tv adások. Ma már nem is meglepő, ha valakinek személyes oldala (honlapja) van.



### 3 A TCP/IP hivatkozási modell

E modell a számítógép-hálózatok őskének tekintett ARPANET, illetve annak leszármazottja, a világméretű internet hivatkozási modellje. Már a kezdetektől fogva az volt a legfőbb tervezési szempont, hogy lehetővé tegyék tetszőlegesen sok hálózat zökkenőmentes összekapcsolását. Ez az architektúra (3. ábra [5]) – a két legjelentősebb protokollja alapján- TCP/IP hivatkozási modell néven vált ismertté, amelyet elsőként Cerf és Kahn (1974) definiált.

Alkalmazási réteg
Szállítási réteg
Internetréteg
Hoszt és hálózat közötti réteg

3. ábra. [5] A TCP/IP hivatkozási modell

Mivel az amerikai védelmi minisztérium (U.S. Department of Defense, DoD) erősen aggódott amiatt, hogy akármelyik nagy értékű hoszt, router vagy hálózatok közötti átjáró (gateway) egy szempillantás alatt megsemmisülhet, ezért egy másik lényeges tervezési szempont az volt, hogy a hálózat az éppen folyó beszélgetések megszakítása nélkül át tudja vészelné az alhálózat esetleges veszteségeit.

#### 3.1 Az internetréteg

Mindezen az elvárások olyan csomagkapcsolt hálózathoz vezettek, amely egy összeköttetés nélküli internetwork rétegen alapulnak. Ez a réteg az internetréteg, amely az egész architektúrát összefogja. Ennek a rétegnek az a feladata, hogy egy hoszt bármilyen hálózatba csomagokat tudjon küldeni, illetve a csomagokat a célállomástól függetlenül (lehetőleg egy másik hálózatba) képes legyen továbbítani. Az sem gond, ha a csomagok nem az elküldés sorrendjében érkeznek meg, ugyanis, ha erre szükség van, akkor a magasabb rétegek visszarendezik őket a megfelelő sorrendbe. Azt viszont ne felejtjük el, hogy az „internet” szó most általános értelemben használjuk annak ellenére, hogy ez a réteg az internetben is jelen van. Az internetréteg meghatároz egy hivatalos csomagformátumot, illetve egy protokollt, amelyet internetprotokollnak (Internet Protocol, IP) hívnak. A csomagok kézbesítése során azok útvonalának meghatározása, valamint a torlódás elkerülése itt most a legfontosabb feladat. Ezért hasonlít funkciója egy másik modell hálózati rétegére.

### 3.2 A szállítási réteg

A TCP/IP-modellben az internetréteg fölötti réteget általában szállítási rétegnek nevezik. Feladata az, hogy lehetővé tegye a forrás- és a célállomásokban található társentitások közötti párbeszédet.

Az egyik szállítási protokoll az átvitelvezérlő protokoll (Transmission Control Protocol, TCP), amely egy megbízható összeköttetés alapú protokoll. Feladata az, hogy hibamentes bájtos átvitelt biztosítson bármely két gép között az interneten. A beérkező bájtos adatfolyamot diszkrét méretű üzenetekre osztja, majd azokat egyesével továbbítja az internetrétegnek. A célállomás TCP-folyamata összegyűjti a beérkezett üzeneteket, és egyetlen kimeneti adatfolyamként továbbítja őket. A TCP forgalomszabályozást is végez annak érdekében, hogy egy gyors forrásállomás csak annyi üzenetet küldjön egy lassabb célállomásnak, amennyit az fogadni képes.

A másik átviteli protokoll ebben a rétegben a felhasználói datagram protokoll (User Datagram Protocol, UDP) amely egy nem megbízható, összeköttetés nélküli protokoll. Jelentősége akkor van, amikor nem szükséges sem az üzenetek TCP-féle sorbarendezeése, sem a forgalomszabályozás. Elsősorban olyan egylövetű, kliens-szerver típusú kérdés-válasz alkalmazásokban terjedt el, ahol a gyors válasz sokkal fontosabb, mint a pontos. Ilyen például a beszéd- vagy videó átvitel. Az IP, a TCP és az UDP kapcsolatát a 4. ábra [7] szemlélteti.

Szolgáltatások	Hálózatkezelés	Időszinkronizálás	Automatikus FDR újrakonfigurálás		Web-szerver	E-mail	TCP Open	Üzenetkezelés
Alkalmazási	SNMP	NTP	TFTP	FTP	HTTP	SMTP		Modbus
Átvitel	UDP			TCP				
Adatkapcsolati	IP							
Fizikai és adatkapcsolati	Ethernet, LAN							
								Protokollok
								Hálózatok

4. ábra. [7] Protokollok és hálózatok a TCP/IP hivatkozási modellben

### **3.3 Az alkalmazási réteg**

A szállítási réteg fölött az alkalmazási réteg található. Ez tartalmazza az összes magasabb szintű protokollt. Eredetileg csak a virtuális terminál (TELNET), a fájltranszfer (FTP) és az elektronikus levelezés (SMTP) protokolljait tartalmazta, amelyeket a 4. ábrán is láthattunk. A virtuális terminál lehetővé teszi, hogy bejelentkezzünk egy távoli gépre, és azon dolgozzunk. A fájltranszfer protokoll segítségével hatékonyan tudunk adatokat átvinni egy gépről a másikra. Az évek során számos más protokollal bővítették az alkalmazási réteget. Ilyen például a Domain Name Service (DNS), amely a hosztok nevét képezi le a hálózati címükre; a HTTP, amely a World Wide Web oldalak letöltését segíti.

### **3.4 A hoszt és a hálózat közötti réteg**

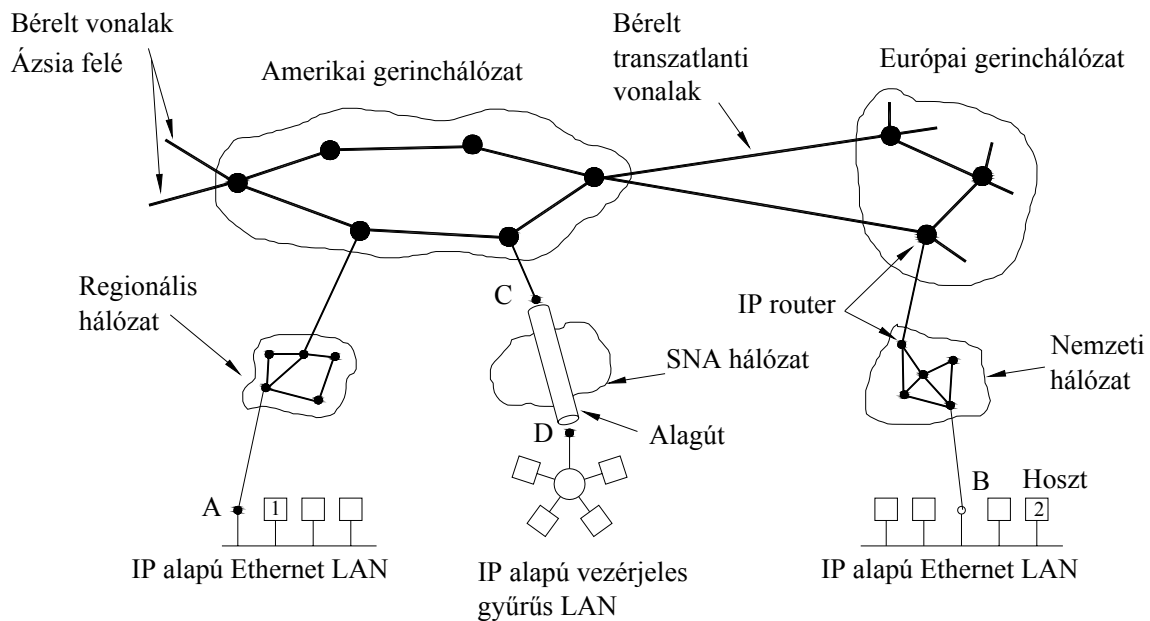
Az internetréteg alatt egy nagy űr tátong. A TCP/IP hivatkozási modell ugyanis nem mondja meg, hogy mi legyen itt, csak annyi megkötést tesz, hogy a hosztnak egy olyan hálózathoz kell csatlakozni, amely az IP-csomagok továbbítására alkalmas protokollal rendelkezik. Ez a protokoll hosztonként, illetve hálózatonként más és más lehet.

## **4 A hálózati réteg és a szállítási réteg kapcsolata**

A hálózati réteg szintjén az Internet autonóm rendszerek (Autonomous System, AS) összekapcsolt együttesének tekinthető. Nincs igazi szerkezete, de létezik számos főbb gerinchálózata, amelyek nagy sáv szélességű vonalakkól és gyors routerekből állnak. A gerinchálózathoz csatlakoznak a területi vagy regionális hálózatok, és ezekhez a hálózatokhoz csatlakoznak az egyetemeken, vállalatoknál és internetszolgáltatóknál lévő LAN-ok. Az 5. ábrán [5] ennek a kvázi-hierarchikus szerveződésnek egy vázlata látható.

Az a ragasztó, amely az Internetet egybetartja, az IP (hálózatközi protokoll). Az Interneten a kommunikáció a következőképpen működik: a szállítási réteg veszi az adatfolyamokat, és datagramokra tördeli azokat. Elméletben a datagramok egyenként 64 Kbájt hosszúak lehetnek, de a gyakorlatban rendszerint 1500 bájt körüliek. Minden datagram átvitelre kerül az Interneten, esetleg menet közben kisebb egységekre darabolva. Amikor végül minden darab megérkezett a célgéphez, a hálózati réteg összeállítja azokat az eredeti datagrammá. A datagramot azután átadja a szállítási rétegnek, amelyik beilleszti azt a vételi folyamat adatfolyamába. Ahogy az 5. ábrából látszik az 1. hoszttól induló csomagnak hat hálózatot kell

megjárnia, hogy eljusson a 2. hosztig. A gyakorlatban ez rendszerint sokkal több, mint hat hálózat. A hálózati réteg végül is szolgáltatokat nyújt a szállítási rétegnek.

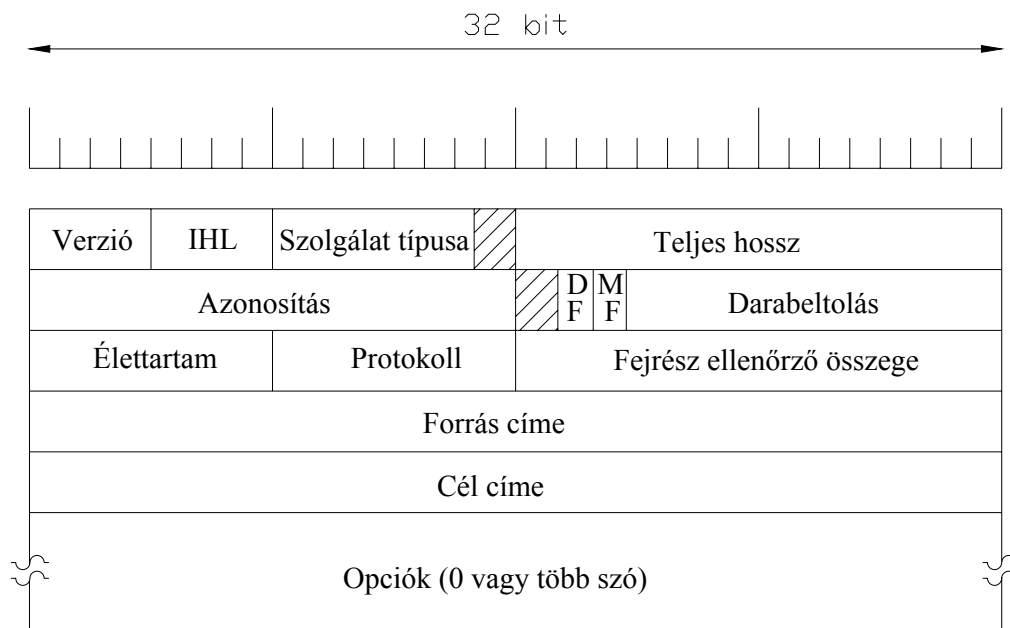


5. ábra. [5] Az Internet sok hálózat egybekapcsolt összessége

#### 4.1 Az IP-protokoll

A TCP az általa feldolgozott datagramokat átadja az IP-nek. Persze ezzel együtt közölnie kell a rendeltetési hely Internet címét is. Az IP-t ezeken kívül nem érdekli más: nem számít, hogy mi található a datagramban vagy, hogy hogyan néz ki a TCP fejléc. Az IP feladata abban áll, hogy a datagram számára megkeresse, a megfelelő útvonalat és azt a másik oldalhoz eljuttassa. Az útközben fellelhető átjárók és egyéb közbülső rendszereken való átjutás megkönnyítésére az IP a datagramhoz hozzáteszi a saját fejlécét. A fejléc fő részei a forrás, és a rendeltetési hely Internet címe (32 bites címek, pl. 128.6.4.94), a protokollszám és egy ellenőrző összeg. A forrás címe a küldő gép címét tartalmazza. (Ez azért szükséges, hogy a vevő oldal tudja honnan érkezett az adat). A rendeltetési hely címe a vevő oldali gép címét jelenti. (Ez, pedig azért szükséges, hogy a közbenső átjárók továbbítani tudják az adatot). A protokollszám kijelöli, hogy a datagram a különböző szállítási folyamatok közül melyikhez tartozik. A TCP egy biztos választási lehetőség. Végül az ellenőrzőösszeg segítségével bizonyosodik meg a vevő oldali IP arról, hogy a fejléc az átvitel során nem sérült-e meg. A TCP és az IP különböző ellenőrzőösszegeket használ. Az IP-nek meg kell tudnia győződni a fejléc sértetlenségéről, különben rossz helyre küldhet el adatot. A TCP és az IP a biztonság és

a hatékonyság növelése miatt tehát külön ellenőrzőösszegeket használ. Az IP fejléc hozzátétele után az eredeti üzenet a 6. ábra [5] szerint néz ki.



**6. ábra. [5]** Az IP-fejrész

A Datagram-eltolás és a DF, MF mezők a datagramok részeinek nyomon követésére használatosak (a bevonalkázott bitet nem használják). Egy datagramot például akkor kell széttördelni, amikor az egy olyan hálózaton halad keresztül, amely számára nagy falatnak mutatkozik. Az Élettartam mezőben lévő szám mindig csökken, amikor a datagram egy rendszeren halad keresztül. Amikor eléri a nullát, a datagram megsemmisül. Ezt az eljárást a rendszerben esetleg felépülő végtelen ciklusok miatt építették a protokollba. Persze ezek felléptének valószínűsége az ideális esetben nulla, de a jól megtervezett hálózatoknak a bekövetkezhetetlen eseményekkel is el kell tudniuk bánni. Amikor a hálózati réteg összerak egy teljes datagramot, tudnia kell, hogy mit tegyen vele. Végül az Azonosítás mező ahhoz kell, hogy a célhoszt meg tudja állapítani, hogy egy újonnan érkezett csomag melyik datagramhoz tartozik. Egy datagram minden egyes darabja ugyanazzal az Azonosítás mező értékkel rendelkezik.

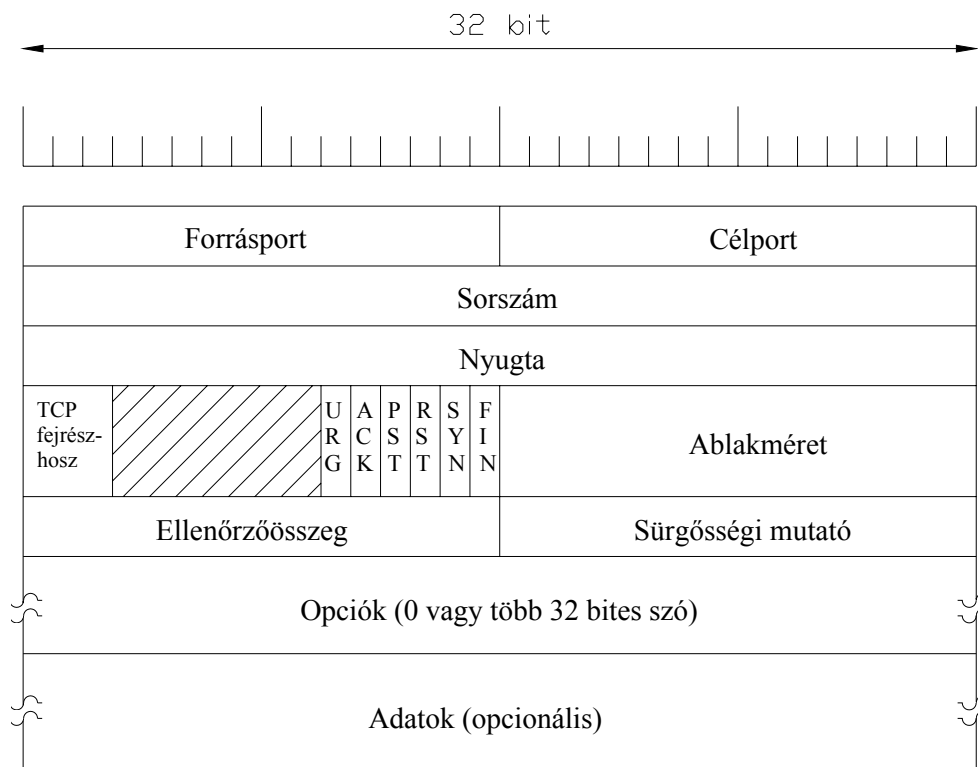
## 4.2 IP-címek

Az Interneten minden hosztnak és routernek van egy IP-címe, amely a hálózat számát és a hoszt számát kódolja. A kombináció egyedi: elméletileg nincs két olyan gép, amelynek ugyanaz az IP-címe. Minden IP-cím 32 bit hosszú és az IP-csomagok Forrás címe és Cél címe

mezőiben hordozzák. A hálózati címeket (amelyek 32 bites számok) rendszerint pontokkal elválasztott decimális jelölésrendszerben írják. Ebben a formátumban minden 4 bájtot tízes számrendszerben írnak ki, 0-tól 255-ig. a legkisebb IP-cím a 0.0.0.0 és a legnagyobb a 255.255.255.255. A 0 és 1 értékeknek speciális jelentéseik vannak. A 0 érték jelentése: ez a hálózat vagy ez a hoszt. Az 1-et adatszóró címként használják, és a jelzett hálózat összes hosztját értik azt. A 0.0.0.0 IP-címet a hosztok az elindulásuk alatt használják. Az olyan IP-címek, amelyeknek a hálózatszáma 0, az aktuális hálózatra vonatkoznak. Ezek a címek lehetővé teszik a gépeknek, hogy a saját hálózatukra hivatkozzanak anélkül, hogy tudnák a számát. A csupa 1-ből álló cím az adatszórást teszi lehetővé a helyi hálózaton, jellemzően egy LAN-on. A helyes hálózattal és csupa 1 hoszt mezővel rendelkező címek lehetővé teszik adatszóró csomagok küldését az interneten bárhol elhelyezkedő távoli LAN-okra.

### 4.3 A TCP-protokoll

Az Internet szállítási protokollja a TCP, melynek működéséről a 3.2 részben már lehetett olvasni. Most viszont az IP-fejrészhez hasonlóan szeretném ismertetni a TCP (7. ábra [5]) felépítését is.



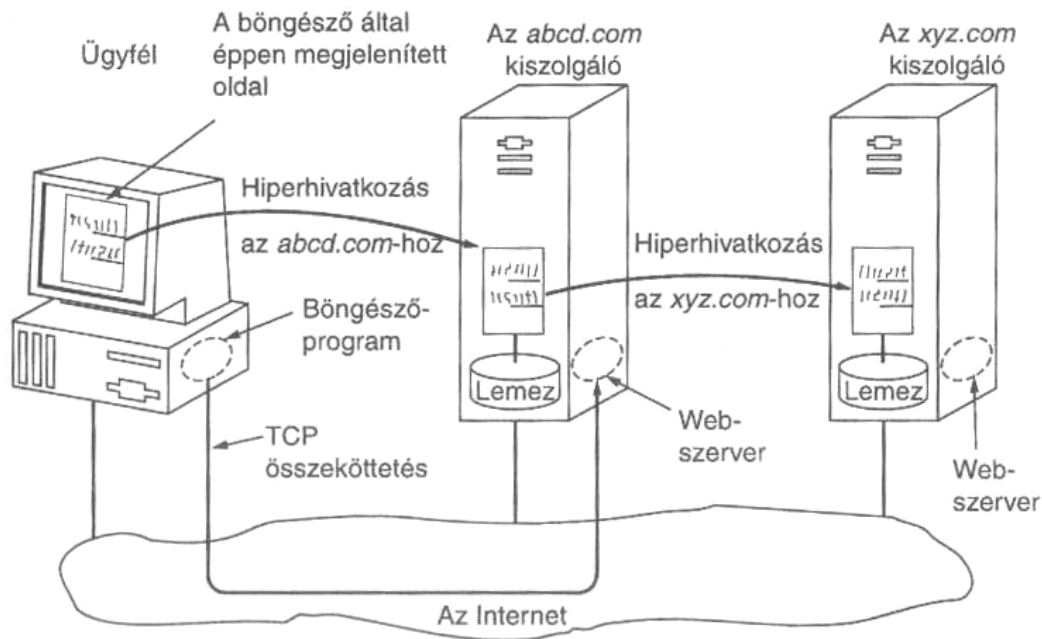
7. ábra. [5] A TCP-fejrész

Minden szegmens egy fix kiosztású 20 bájtos fejrészsel kezdődik, amit fejrészopciók követhetnek. Ezek után legfeljebb  $65535 - 20 - 20 = 65495$  bájt adat állhat. A kivonandók közül az első 20 bájt az IP-, a második 20 bájt a TCP-fejrészt jelenti. A Forráspont és a Célpont mezők azonosítják az összeköttetés helyi végpontjait. A TCP-fejrész hossz mondja meg, hány 32 bites szóból áll a TCP-fejrész URG bit értéke 1, ha sürgősségi mutatót használt. Ez a mutató a sürgős adat bájtban mért helyét jelzi a jelenlegi bájtsorszámhoz viszonyítva. Ez a mechanizmus a megszakítás üzeneteket helyettesíti, lehetőséget adva a küldőnek, hogy jelzést küldjön a vevő felé anélkül, hogy a TCP-nek külön megszakításokkal kelljen foglalkoznia. Az ACK bit 1 értéke jelzi a nyugta mező érvényességét. Ha ez 0 akkor a nyugta mező figyelmen kívül hagyható. A PSH bit jelzi a késedelem nélküli adat továbbítását. Az RST bit egy hoszt összeomlása, vagy más okból összezavart összeköttetés helyreállítására szolgál. Ha valaki RST = 1 értéket viselő szegmenst kap, akkor felkészülhet valamilyen probléma megoldására. SNY bit összeköttetés létesítésére szolgál, míg a FIN bit összeköttetés bontására szolgál. Jelzi, hogy a küldőnek nincs több továbbítandó adata. A TCP forgalom-szabályozása változó méretű csúszóablakkal történik. Az ablakméret mező határozza meg, hogy a nyugtázott bájttal kezdődően hány bájtot lehet elküldeni.

## 5 A Világháló (World Wide Web)

A Világháló egy keretszerkezet, amely az Interneten lévő gépeken elszórva elhelyezkedő, összekapcsolt dokumentumok elérését teszi lehetővé. A felhasználók szemszögéből a Világháló (Web) különféle dokumentumok vagy weboldalak, vagy röviden csak oldalak (pages) hatalmas, világméretű gyűjteményéből áll. Az oldalakat egy böngészőnek (browser) nevezett programmal tekinthetünk meg, mint például a népszerű Internet Explorer vagy a Mozilla Firefox. A böngésző elhozza a kívánt oldalt, értelmezi a szöveget és az abban előforduló formázóparancsokat, majd megfelelően formázva megjeleníti az oldalt a képernyőn. A Web működésének alapvető modelljét a 8. ábra [5] mutatja. A böngésző itt egy weboldalt jelenít meg az ügyfél oldalán. Amikor egy felhasználó egy olyan sorra kattint a szövegben, amely egy, az abcd.com kiszolgálón lévő oldalra mutat, akkor a böngésző követi a hipervivatkozást, és elküld egy üzenetet az abcd.com kiszolgálónak, melyben elkéri az oldalt. Amikor az oldal megérkezik, a böngésző megjeleníti. Ha ezen az oldalon van egy hipervivatkozás az xyz.com kiszolgáló egy oldalára, és a felhasználó arra rákattint, akkor a böngésző elküld egy kérelmet annak a gépnek, és az így megy tovább a végtelenségig. Az

oldalakat URL-ek (Uniform Resource Locator – egységes erőforrás-meghatározó) segítségével nevezik meg. Az URL-nek három fő része van: a protokoll neve (http), annak a gépnek a DNS-neve, ahol az oldal megtalálható (www.bme.hu), és (általában) az oldalt tartalmazó állomány neve (index.html).



8. ábra. [5] A webmodell részei

Tegyük fel, hogy a felhasználó a Weben böngészve talál egy hivatkozást a szikraforgácsoló berendezésekre, mely az EDM Kft. honlapjára mutat (<http://www.edmkft.hu/szakra.html>). Az alábbi felsorolás ezen hivatkozás kiválasztását követő lépéseket írja le:

1. A böngésző meghatározza az URL-t (megnézi, hogy mit választottak ki).
2. A böngésző megkérdezi a DNS-től a [www.edmkft.hu](http://www.edmkft.hu) IP-címét.
3. A DNS válasza: 217.113.62.25
4. A böngésző létrehoz egy TCP-összeköttetést a 217.113.62.25–vel a 80-as porton.
5. Átküld egy kérést, melyben elkéri a /szakra.html állományt.
6. A [www.edmkft.hu](http://www.edmkft.hu) kiszolgáló elküldi a /szakra.html állományt.
7. A TCP-összeköttetést lebontják.
8. A böngésző megjeleníti a /szakra.html –ben található összes szöveget.
9. A böngésző elhozza és megjeleníti az állományhoz tartozó összes képet.

Ahhoz, hogy meg tudja jeleníteni az oldalt, a böngészőnek értenie kell az oldal formátumát. Az oldalakat egy HTML (HyperText Markup Language – hipertext jelölőnyelv) nevű,



weboldalakat leíró, szabványosított nyelven írják, hogy minden böngésző megértse minden weboldalt.

Az ügyfél oldalát megismerve, most had mutassam be a kiszolgáló (szerver) oldalát. A kiszolgáló, akárcsak egy igazi webszerver, egy állomány nevét kapja meg, amit meg kell keresnie és vissza kell adnia, a következő lépéseket hajtja végre a központi ciklusában:

1. Elfogad egy TCP-összeköttetést az ügyféltől (a böngészőtől).
2. Megkapja a kért állomány nevét.
3. Megkeresi az állományt (a háttértáron).
4. Visszaadja az állományt az ügyfélnek.
5. Bontja a TCP-összeköttetést.

A modern webszervernek több funkciója van ennél, de alapjában véve ennyi egy webszerver feladata. A valóságban a kérések feldolgozása egészen bonyolult is lehet. Például a webszerver, ügyfél azonosítást végez, vagy a résztvevőkre vonatkozó hozzáférési jogosultságokat ellenőriz, bejegyzéseket helyez el a kiszolgáló naplójában.

## **6 A HTML dokumentumokról**

A HTML dokumentum- formátumot tekinthetjük az ún. hyper-text egyik megvalósítási formájának is. A HTML dokumentum egy olyan szövegfájl, amely a szövegen kívül tartalmaz ún. "HTML-tag"-eket - formázóutasításokat -, valamint megjelenítendő objektumokra történő hivatkozásokat is. Ezek a HTML formázóutasítások (más szóval: parancsok, elemek) befolyásolják a dokumentum megjelenítését, kapcsolatait. Ezeket az utasításokat a böngészőprogram értelmezi és végrehajtja. Ezen okból a formázóutasítás mindig megelőzi azt a részét a dokumentumnak, amelyre vonatkozik. A dokumentumkészítéshez használható HTML utasítások köre állandóan bővül, a nyelv fejlődik. A szabványosítás csak lassan követi a fejlődést. Ezért nem minden böngészőprogram tudja a HTML utasítások mindegyikét értelmezni. Egy böngésző, ha számára értelmetlen utasítással találkozik, akkor kihagyja, így nem okoznak problémát az újabb keletű - még szabványosítatlan - utasítások a régebbi kiadású WWW-böngészőknek sem. Sajnos a fentiek miatt ugyanazt a dokumentumot két különböző program nem biztos, hogy azonos formában fogja megjeleníteni. Más oka is van ennek. A WWW-n kalandozónál kicsi a valószínűsége annak, hogy rendelkezésére áll ugyanaz a betűtípus, mint a WWW-oldalt fejlesztőnek. Vagy képek esetén semmi garancia nincs arra, hogy minden böngészőprogram ugyanazon felbontásban és színszámmal tudja

megjeleníteni a képet. A HTML-ben mégis az a nagyszerű, hogy nagymértékben megközelíti a platformfüggetlenséget. Egy HTML dokumentum - ha nem is azonos módon - mindenki számára megtekinthető.

## 7 Programozható vezérlők

A programozható vezérlők az 1970-es évektől kezdődően terjedtek el és ma csaknem kizárólagos alkalmazást nyertek el az ipari folyamatok vezérlésében. A programozható vezérlőberendezések, a vezérlési (esetleg szabályozási) funkciókat szoftver útján valósítják meg és beviteli, kiviteli egységeik révén a technológiai folyamatok tárolt programú vezérlésére közvetlenül alkalmasak.

1968-ban a General Motors cég pályázatot hirdetett olyan programozható vezérlőberendezés fejlesztésére, amely ötvözi a relés, a félvezetős és a számítógépes vezérlés előnyeit.

A pályázat specifikációjában az alábbi szempontok szerepeltek:

- Egyszerű, moduláris felépítés, kis méret;
- Mozgó alkatrészt ne tartalmazzon;
- Galvanikusan leválasztott bemeneti/kimeneti fokozatok (24 Vdc-től 240Vac);
- Könnyű programozhatóság és újraprogramozás;
- Valós idejű működés max. 0,1 s válaszüddel;
- Nagy megbízhatóság, minimális karbantartás;
- Versenyképes ár.

A mai PLC-eket (Programmable Logic Controller, programozható logikai vezérlő), kivételük alapján kompakt és moduláris felépítésű csoportba sorolhatjuk.

A kompakt PLC jellemzője, hogy hardverstruktúrája nem módosítható, kizárólag megfelelő védettségű ipari tokozásban készül és kis helyigényű. Felhasználási területei a sorozatban gyártott gépek, berendezések automatikái, illetve a PLC műszaki jellemzői által lefedhető egyedi vezérlések. A kompakt PLC-k speciális típusát jelentik az ún. mikro-PLC-k, amelyek az ember-gép kapcsolat hardver- és szoftverfeltételét is tartalmazzák.

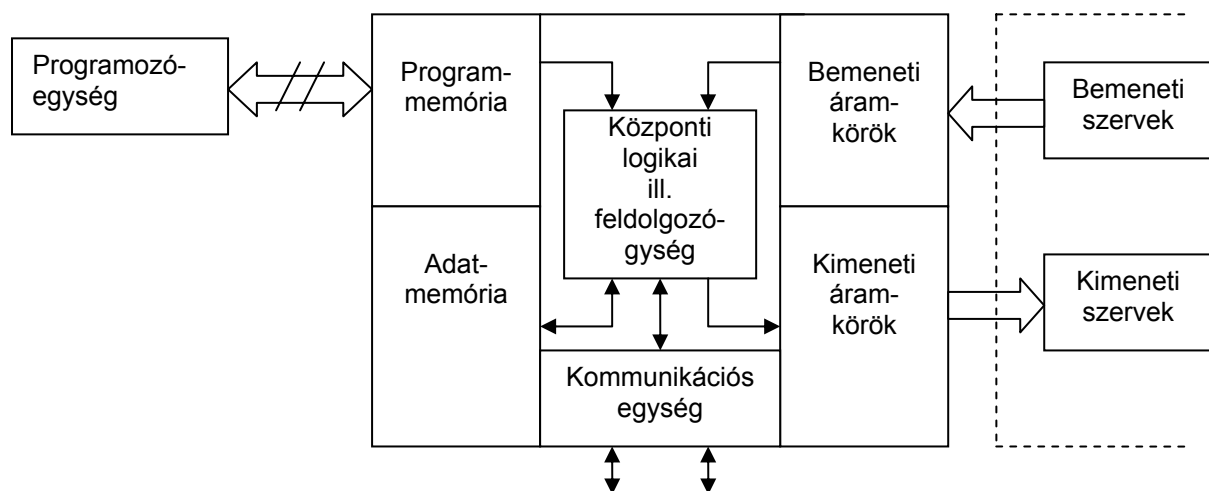
A moduláris felépítésű programozható logikai vezérlők jellemzője, hogy a vezérlőberendezés valamely speciális funkciót önmagában ellátó modulokból épül fel. A modulok fizikai megjelenése rendszerint az áramköri kártya, dugaszolható csatlakozóval. A modulok ún. rack-be (tartó) dugaszolhatók, ezért a rendszer konfigurációja tág határokon belül bővíthető. A rack-ek megfelelő védettségű műszerdobozba vagy műszerszekrénybe szerelhetők. A

moduláris felépítésű PLC-eket közepes, ill. nagyméretű rugalmas gyártórendszerek vagy ipari folyamatok irányítására fejlesztették ki.

## 7.1 PLC-k funkcionális felépítése

A 9. ábra [1] szerint a PLC-k főbb egységei:

- Központi logikai ill. feldolgozóegység (LU, CPU, stb.);
- Programmemória (ROM, EPROM, EEROM);
- Adatmemória (RAM);
- Bemeneti (input) egységek (digitális, ill. analóg);
- Kimeneti (output) egységek (digitális, ill. analóg);
- Kommunikációs egység



9. ábra. [1] A PLC funkcionális felépítése

A programozható vezérlők központi egysége a bemenetek és a kimenetek közötti, többnyire logikai kapcsolatokat időben sorosan és ciklikusan hajtja végre a programmemóriában tárolt program alapján. A soros jellegű adatfeldolgozásból eredően a ciklikus feldolgozást nagy sebességgel kell végrehajtani, hogy a működés kifelé párhuzamosnak (valós idejűnek) tűnjék.

A programozható vezérlő az alábbi funkciók ellátására képes:

- Jel/adat feldolgozási funkció (signal/data processing);
- Technológia interfészfunkció az érzékelők kezelésére, ill. beavatkozók működtetésére;
- Kommunikációs funkciók (PLC-PLC; PLC-számítógép; PLC-hálózat);
- Ember-gép interfészfunkció (man-machine interface, MMI)

- Programozási, tesztelési, dokumentálási funkció;
- Tápellátási funkció.

A programozható vezérlők előnyei: a szabad strukturálhatóság, a gyakorlatilag végtelen kapcsolási szám, a telepi költségek csökkenése, a rendszerbe szervezhetőség lehetősége. A szabad strukturálhatóság felhasználói programozhatóságot jelent, amelynek révén a felhasználó a tárolt, egyedi program révén az univerzális hardvert a feladatra alkalmassá teszi. A PLC-k alkalmazásával a telepítési, beüzemelési, beüzemelési idő nagymértékben lerövidíthető.

A következő részben a PLC-k működési és felépítési jellemzőinek eddigi leírása után szeretném bemutatni, azon funkcióját a programozható vezérlőknek, mely TDK munkám szerves részét képezte.

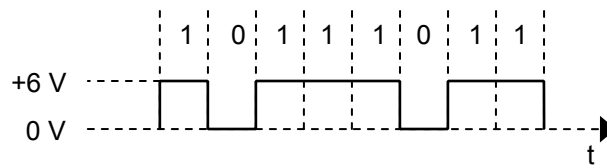
## **7.2 A PLC-k kommunikációs rendszere**

A programozható vezérlők üzemszerűen számos információforrással állnak kapcsolatban. A leggyakrabban a PLC és a technológiai folyamat, PLC és PLC, PLC és számítógép, PLC és kezelő, valamint PLC és periféria közötti kommunikációra van igény. A PLC és a technológiai folyamat közötti kommunikáció általában párhuzamos formában zajlik, kivéve a terepi buszrendszerek által kezelt be- és kimeneteket.

A párhuzamosan kezelt jelek lehetnek:

- Kétállapotú be/kimenetek;
- Analóg be/kimenetek;
- Frekvencia (impulzus) be/kimenetek.

A PLC és PLC, PLC és PC, PLC és kezelő, valamint PLC és periféria közötti kommunikáció rendszerint soros formában történik. Soros adatátvitel esetén az adatok bitenként, a kiegészítő, ellenőrző jelekkel együtt, időben egymás után rendszerint egy vezetéken (érpáron) kerülnek továbbításra. Az információt a feszültség vagy az áram szintje, ill. jelátmenete képviselheti. A 10. ábra [1] szerinti soros adatátvitelnél a 0 szintet 0 V, az 1 szintet 6 V képviseli, amely min. egy bitideig tart. A soros átvitelnek számos jellemzője és szabványa van (RS 232, RS 485, stb.).

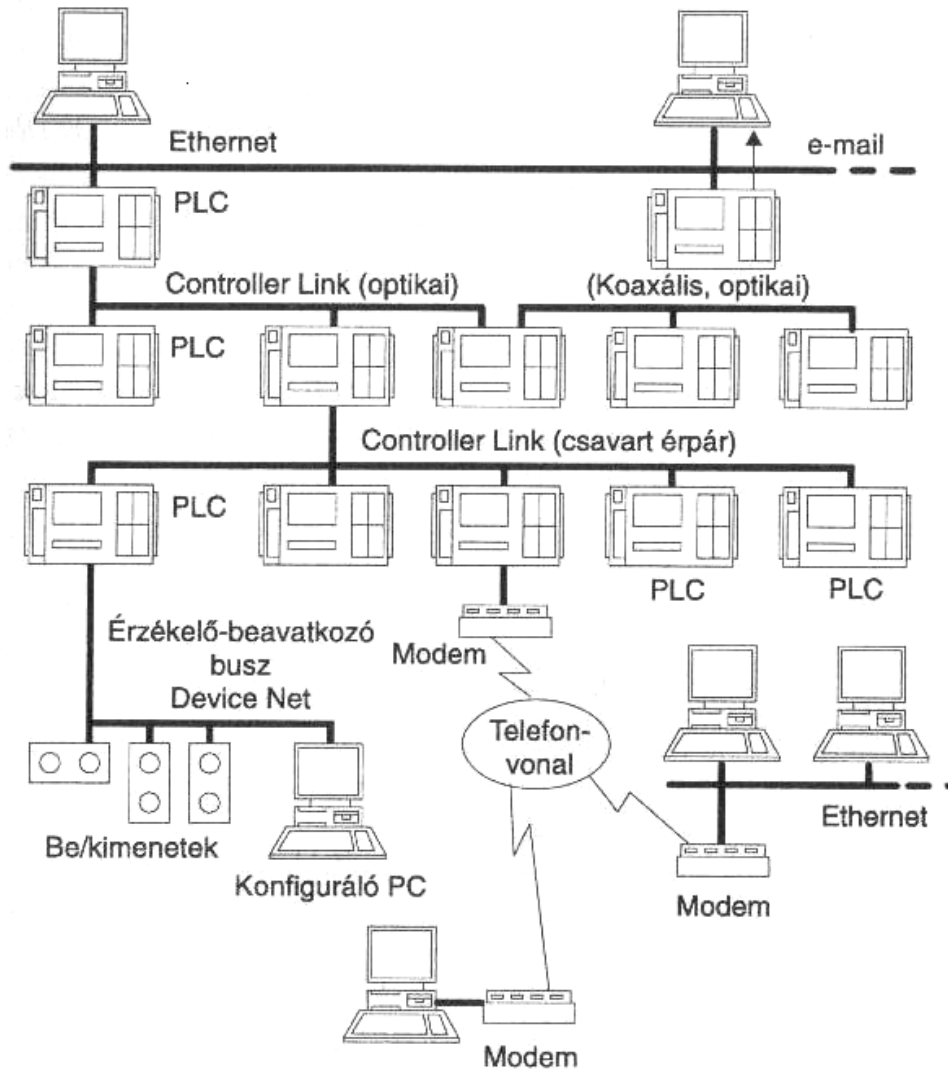


**10. ábra.** [1] Soros adatátvitel formátum

Soros adatátvitel a kommunikációban részt vevő adók és vevők számától függően alapvetően két pont között (pont-pont kommunikáció), ill. több pont között történhet. Pont-pont közötti kommunikáció esetén két egység között folyik az adatátvitel egy vagy mindkét irányban. A több pont közötti (multi-point) kommunikáció napjainkban szinte kizárólag hálózati jellegű, amely jellemzője, hogy kettőnél több egység (adó, ill. vevő) között egy speciális architektúrájú átviteli közegen keresztül történik az adatátvitel. A jelenlegi informatikai hálózatok a hálózat által áthidalta távolságok alapján távolsági hálózatokra, ill. lokális hálózatokra bontható.

A távolsági hálózatok (Wide Area Network, WAN) rendszerint a nyilvános távközlési berendezéseket veszik igénybe és nagy távolságú, földrészek közötti kommunikációt valósítanak meg. A lokális hálózatok (LAN) kis és közepes kiterjedésű földrajzi területen elhelyezkedő, független eszközök közötti egyenrangú kommunikációt valósítanak meg. A hálózati elemeket három csoportba sorolhatjuk: a kommunikációs állomásokra, az állomásokat összekapcsoló kábelrendszerre és a kiegészítő hálózati elemekre. A hálózati állomásoknak megfelelően intelligens szoftverrel kell rendelkezniük ahhoz, hogy kezelni tudják a hálózat számára szükséges kommunikációs és vezérlési funkciókat.

A soros kommunikáció jelentősége napjainkban felértékelődik. A PLC programjának jelentős részét a kommunikációra vonatkozó programrész teszi ki, amely elkészítése a konkrét PLC hardver- és szoftveradottságain túl sokrétű, a kommunikációra vonatkozó általános és specifikus informatikai ismereteket igényel. A 11. ábrán [1] a PLC-k alkalmazása látható többszintű informatikai rendszerben.



11. ábra. [1] PLC-k többszintű informatikai rendszerben

### 7.3 Hálózati kommunikáció

Bár a pont-pont közötti kommunikációnak vannak tipikus alkalmazási területei, napjainkban a több pont közötti kommunikáció igénye növekszik.

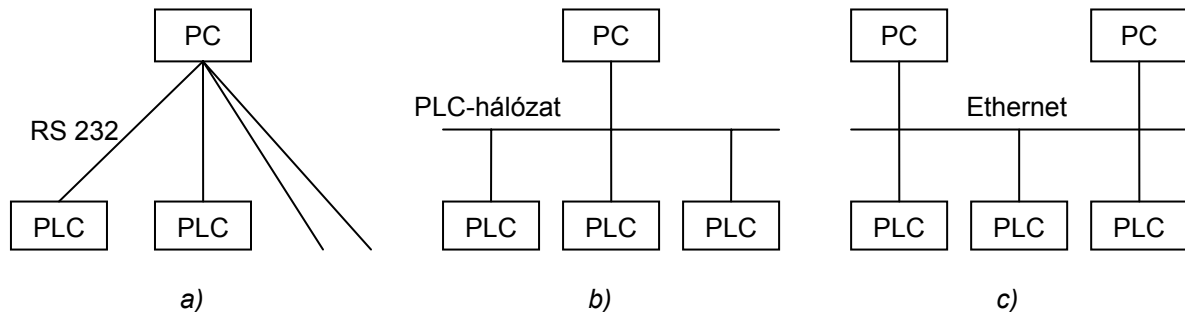
Több PLC közötti kommunikáció módszerei a következők:

- Több soros vonal révén kialakított pont-pont kapcsolat (12.a ábra) [1];
- PLC-hálózat (12.b ábra) [1];
- ETHERNET-hálózat (12.c ábra) [1];
- Adatgyűjtő számítógép

A 12.a ábra szerinti megoldás korlátozott kommunikációra alkalmas. Ezért látják el a PLC-eket egynél több RS 232C szabványú interfésszel. A megoldás hátránya az alacsony átviteli sebesség és a kis távolság.

A 12.b ábrán látható PLC-hálózat a leggyakoribb kommunikációs módszer a több PLC közötti információs kapcsolat kialakításához.

Tekintettel az ETHERNET-hálózat (és ezen keresztül az Internet) világméretű elterjedésére és az ebből eredő előnyökre, különösen a nagy távolságú, ill. nagy adatmennyiségeket igénylő kommunikáció esetén célszerű alkalmazni. Minden perspektívikus PLC-nek van ETHERNET hálózati csatlóója.



**12. ábra. Több [1] PLC közötti kommunikáció lehetőség**

Az irányítástechnika kommunikációs hálózatainak előnyei:

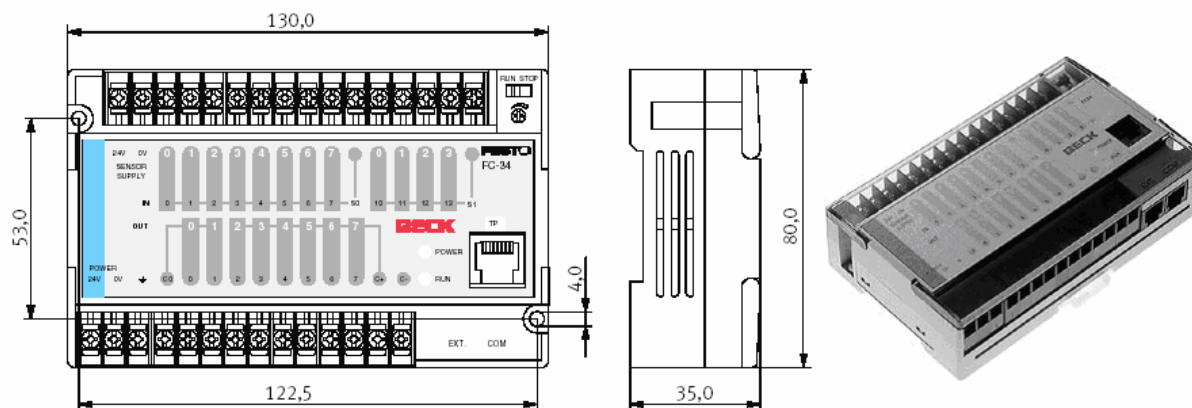
- Jelentősen kisebb kábelköltségek;
- Kisebb méretű kapcsolószekrények;
- Kisebb mennyiségű járulékos, hagyományos technika;
- Kisebb telepítési és ráfordítási költségek;
- Kisebb szervizköltségek;
- Nagyobb üzembiztonság és teljesítmény;
- Rugalmas módosítási lehetőség.

A PLC rendszer ETHERNET-hálózatának alkalmazási példái:

- Távprogramozás és monitorozás;
- Mail-szolgáltatás pl.: a PLC e-mail üzenetet küld hiba detektálásakor;
- PLC-PLC közötti üzenetváltás;
- Fájlvitel valósítható meg a PLC-memória és a PC memóriája között.

## 8 FEC PLC

Munkám során a FESTO, FC34-típusú programozható vezérlőjével dolgoztam (13. ábra), amelyet már egy ETHRET csatlakozóval is elláttak a gyártók. A FESTO PLC-k nagy előnye, hogy oktatásra fejlesztették ki elsősorban, ezért igen jól elsajátíthatók a PLC programozás lépései, módjai. Külön érdekessége hogy egy ilyen berendezésben megtalálható a tranzistoros logikán kívül a régebbi relés kapcsoló megoldás is.



13. ábra FESTO FC34 típusú PLC

A PLC tartalmaz egy FTP szervert (File Transfer Protocol = állománytovábbító protokoll). Fájlok átvitelét lebonyolító protokoll. Szabványosított protokoll, amely az adatátvitel adott környezetre illeszkedő menetrendjét határozza meg. FTP-programokkal azonosító jelszó birtokában távoli adatbázisok tartalmához férhetünk hozzá, onnan, pl. bizonyos állományokat szerezhetünk be, és oda állományokat tölthetünk fel (pl. saját Web-oldalainkat szolgáltatónk gazdagépére). Átviteli idő a modem sebességétől, és a földrajzi helyzettől függ.

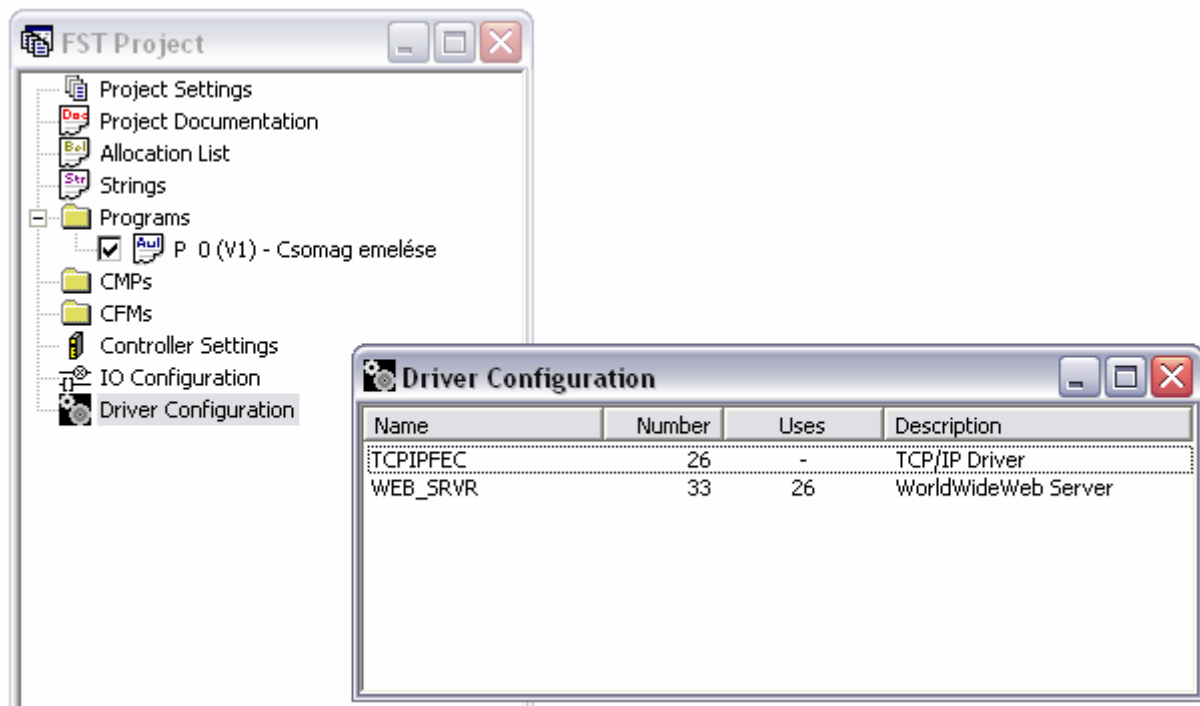
HTTP beágyazott webszerverrel rendelkeznek. A webes szolgáltatások segítségével az automatizálási rendszereken egy egyszerű internetböngésző segítségével diagnosztikai vagy karbantartási folyamatok futtathatók lokálisan vagy távolról:

- PLC rendszer és I/O modul diagnosztika, PLC-hiba megjelenítése (használatra kész „Rack Viewer” oldalak) [7]
- PLC-változók megjelenítése és beállítása (használatra kész „Data Editor” oldalak) [7]

A beágyazott webszerver egy valós idejű PLC adatszerver. Minden adat hagyományos, HTML-formátumú weboldalakon keresztül jeleníthető meg, így minden Java-t támogató internetböngészőben megtekinthető.

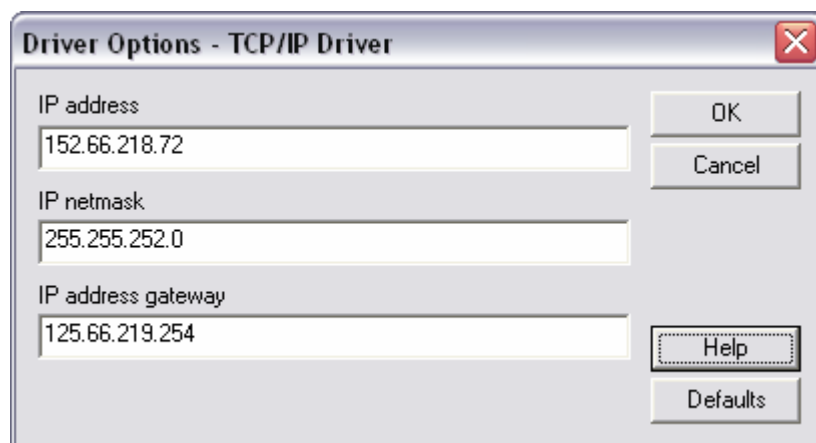


Fontos, hogy ne felejtjük el az FST Project, Driver Configuration opciójában a 14. ábra szerinti Driver-eket hozzáadni.



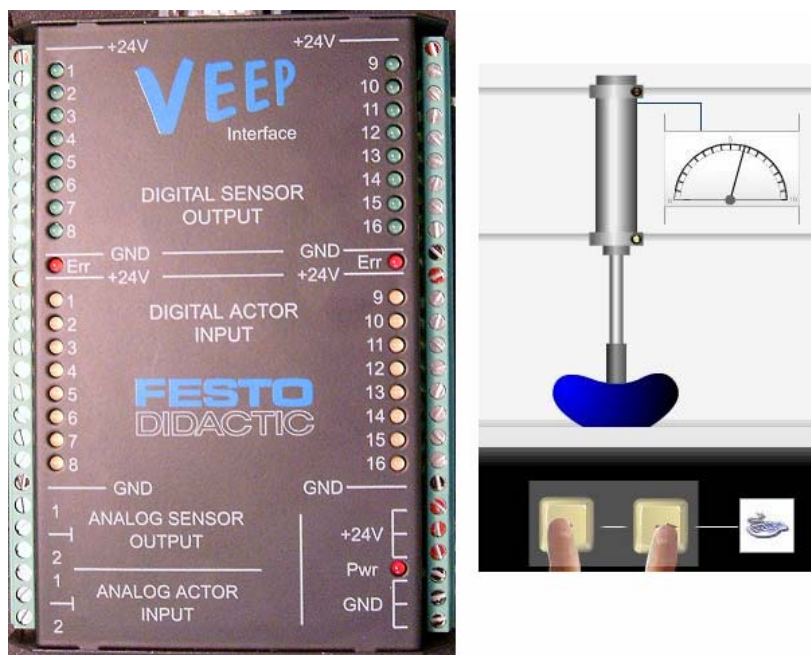
14. ábra Driver-ek

Külön figyelmet kell fordítani még a TCPIPFEC-re, ugyanis meg kell határoznunk a PLC-nk IP-címét (15. ábra), a netmaszk és az átjáró címét is, hogy könnyedén be tudjuk azonosítani és bárhol a világon megtalálni és működtetni azt!



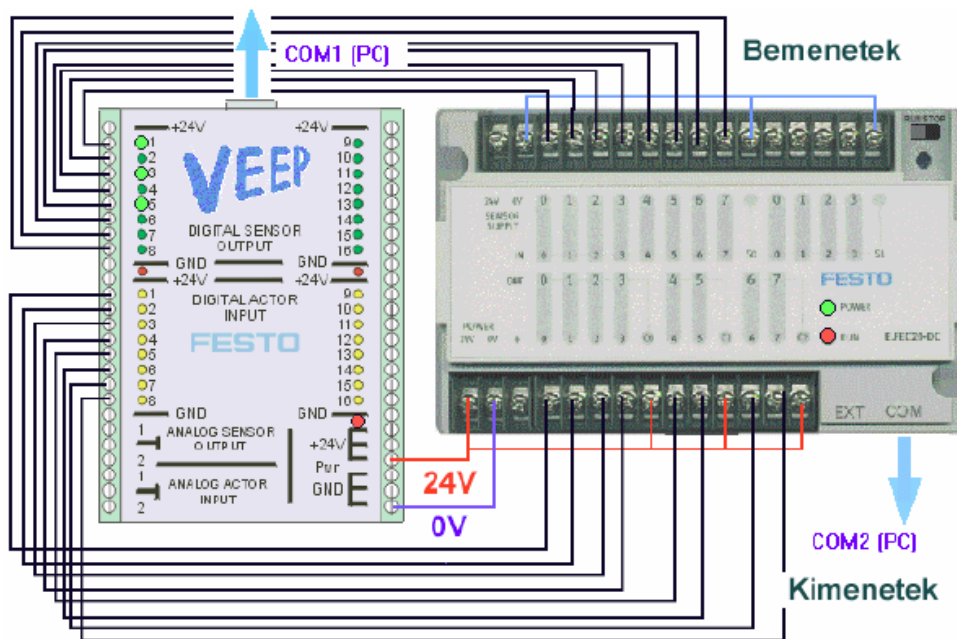
15. ábra TCP/IP beállítások





17. ábra VEEP kimeneti és bemeni sorkapcsa és az egyik emulált berendezés képe

Felhasználáskor elegendő megismerni a kívánt PLC be- és kimeneti pontjait, a VEEP sorkapocs léccre csatlakoztatni (17. ábra) [8], és már vezérelhetők a VEEP segítségével emulált berendezések. A berendezés 24V DC tápfeszültséget igényel, amit célszerű közvetlenül a tápegységről venni. A VEEP vezetékek a sárga LED-es pontokról (a VEEP által ábrázolt berendezés végrehajtói, szelepei, motorai) mindig a FEC kimeneteire csatlakoznak (alsó sorkapocs pontok). A VEEP vezetékek a zöld LED-es pontokról (VEEP által ábrázolt berendezés szenzorai) mindig a FEC bemeneteire csatlakoznak (felső sorkapocs pontok).

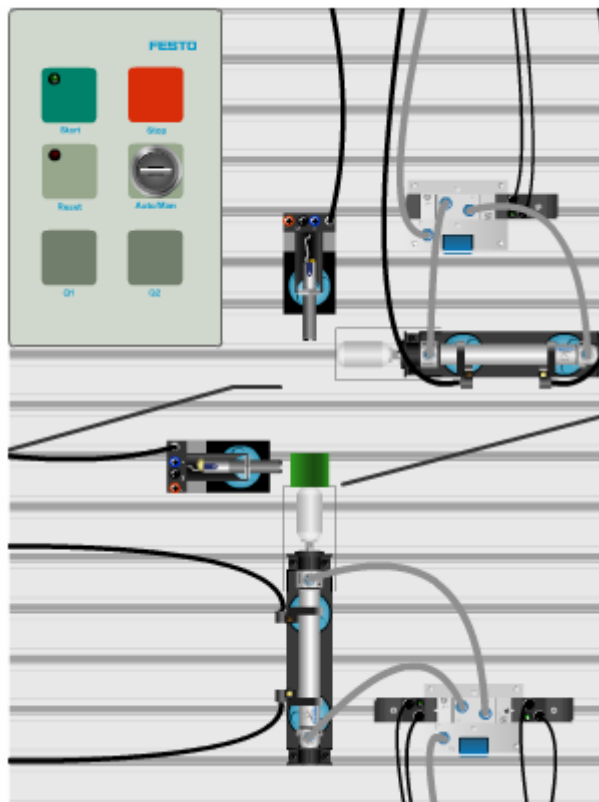


18. ábra [8] A VEEP és FEC elektromos csatlakoztatása

## 10 Példafeladat Internetes vezérlésre és felügyeletre

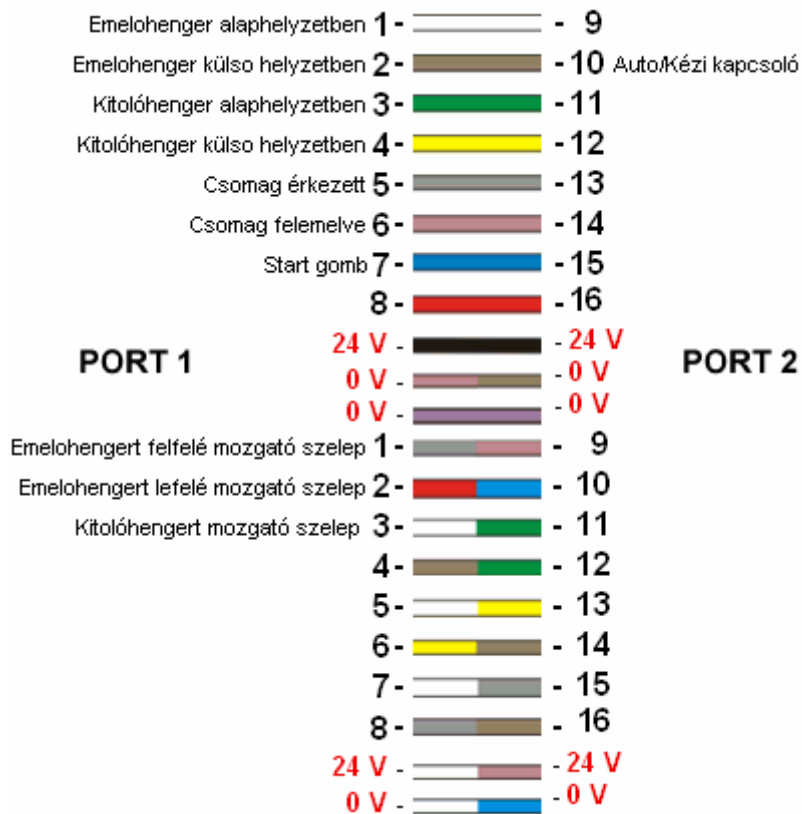
Sokkal könnyebb megérteni az elméletet, ha párosul mellé egy-két példafeladat is! Ezért is volt TDK munkám alap ötlete egy konkrét feladat megoldása, melyen keresztül megvalósítható, megismerhető az Interneten keresztüli működtetés lehetősége programozható vezérlők esetén.

A VEEP emulátoron lévő, adott feladatok közül a választás egy, a csomagok emelését végző berendezésre esett. A 19. ábrán látható elrendezés alapján, egymáshoz csatlakozó csúszdákon csomagot szállítunk. Ha a csomag eléri a csúszda alját, két pneumatikus henger megfelelő vezérlésével elérjük, hogy a csomag a következő csúszda tetejére kerüljön. Az emelőhengert bistabil szelep, míg a kitolóhengert monostabil szelep vezérli.

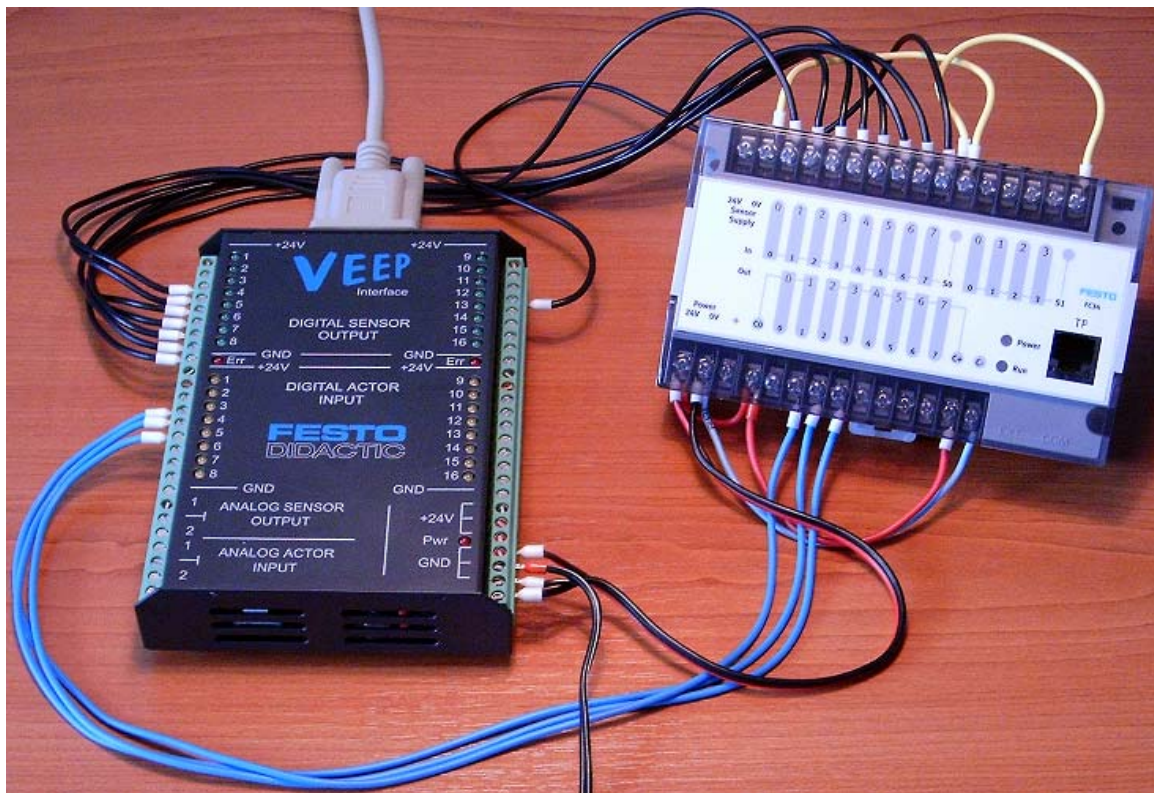


19. ábra Csomagtovábbító berendezés

Össze kell kötnünk a PLC be- és kimeneteit a VEEP megfelelő ki- és bemeneteivel, hisz a rendszerben lévő érzékelőket, szelepeket, munkahengereket, nyomógombot és kapcsolót a VEEP helyettesíti. Célszerű a vezetékek sorrendjét a PLC bekötésénél is megtartani a könnyebb követhetőség érdekében. A VEEP bekötése a 20. ábrán látható.



20. ábra VEEP bekötése



21. ábra A VEEP és a PLC, bekötés után

## **10.1 PLC-program**

A megfelelő bekötés után a berendezés működéséhez szükséges programot kell megírunk, mely most a FESTO AWL utasításlistás nyelv alapján történt az FST 4.1 programot felhasználva. Az utasításlistás nyelv logikai elemeket használ: ÉS, VAGY, NEM, HA, AKKOR, EGYÉBKÉNT.....

### **A programban szereplő hivatkozások jelentése:**

O0.1	Emelőhenger felfelé mozgató szelep
O0.2	Emelőhengert lefelé mozgató szelep
O0.3	Kitolóhengert mozgató szelep
I0.0	START
I0.1	Emelőhenger alaphelyzetben
I0.2	Emelőhenger külső helyzetben
I0.3	Kitolóhenger alaphelyzetben
I0.4	Kitolóhenger külső helyzetben
I0.5	Csomag érkezett
I0.6	Csomag felemelve
I0.7	Kézi / Automata (Helyszíni vezérlés / Interneten keresztüli vezérlés)
F1.0	Emelés indul jelzőbit
F1.1	Emelőhenger alaphelyzetben jelzőbit
F1.2	Emelőhenger külső helyzetben jelzőbit
F1.3	Kitolóhenger alaphelyzetben jelzőbit
F1.4	Kitolóhenger külsőhelyzetben jelzőbit
F2.2	Kézi / Automata jelzőbit
FW0	Kimeneti szó
C0	Számláló bit
CP0	Számláló értéke
CW0	Számláló aktuális értéke

## Programtörzs:

```
STEP 1
IF          I0.7          'Kézi / Automat
  AND      F2.2          'Kézi / Automat jelzőbit
THEN JMP TO 2
ELSE JMP TO 3

STEP 2
IF          NOP
THEN LOAD  V500
  TO      CP0          'Számláló értéke
  SET    C0          'Számláló bit
IF          FW0          'Kimeneti szó
  =      V2
THEN SET  O0.1        'Emelőhenger felfelé mozgató szelep
ELSE RESET O0.1      'Emelőhenger felfelé mozgató szelep

IF          FW0          'Kimeneti szó
  =      V3
THEN SET  O0.2        'Emelőhengert lefelé mozgató szelep
ELSE RESET O0.2      'Emelőhengert lefelé mozgató szelep

IF          FW0          'Kimeneti szó
  =      V4
THEN SET  O0.3        'Kitolóhengert mozgató szelep
ELSE RESET O0.3      'Kitolóhengert mozgató szelep

IF          I0.1        'Emelőhenger alaphelyzetben
THEN SET  F1.1        'Emelőhenger alaphelyzetben jelzőbit
ELSE RESET F1.1      'Emelőhenger alaphelyzetben jelzőbit
IF          I0.2        'Emelőhenger külső helyzetben
THEN SET  F1.2        'Emelőhenger külső helyzetben jelzőbit
ELSE RESET F1.2      'Emelőhenger külső helyzetben jelzőbit
IF          I0.3        'Kitolóhenger alaphelyzetben
THEN SET  F1.3        'Kitolóhenger alaphelyzetben jelzőbit
ELSE RESET F1.3      'Kitolóhenger alaphelyzetben jelzőbit
IF          I0.4        'Kitolóhenger külső helyzetben
THEN SET  F1.4        'Kitolóhenger külső helyzetben jelzőbit
ELSE RESET F1.4      'Kitolóhenger külső helyzetben jelzőbit

IF          NOP
THEN JMP TO 1

STEP 3
IF          ( I0.1        'Emelőhenger alaphelyzetben
  AND      I0.3        'Kitolóhenger alaphelyzetben
  AND      I0.5        'Csomag érkezett
  AND      I0.0        ) 'START
  OR      ( F1.0        'Emelés indul jelzőbit
  AND      I0.5        'Csomag érkezett
  AND      I0.7        ) 'Kézi / Automata
THEN JMP TO 4
ELSE JMP TO 1

STEP 4
IF          I0.1        'Emelőhenger alaphelyzetben
THEN RESET O0.2      'Emelőhengert lefelé mozgató szelep
```

```

STEP 5
IF      (      I0.1      'Emelőhenger alaphelyzetben
      AND      I0.3      'Kitolóhenger alaphelyzetben
      AND      I0.5      'Csomag érkezett
      AND      I0.0      )      'START
      OR      (      F1.0      'Emelés indul jelzőbit
      AND      I0.5      'Csomag érkezett
      AND      I0.7      )      'Kézi / Automata
THEN SET      O0.1      'Emelőhenger felfelé mozgó szelep
      INC      CW0      'Számológó aktuális értéke

STEP 6
IF      I0.6      'Csomag felemelve
      AND      I0.2      'Emelőhenger külső helyzetben
THEN SET      O0.3      'Kitolóhengert mozgó szelep
      RESET     O0.1      'Emelőhenger felfelé mozgó szelep

STEP 7
IF      I0.4      'Kitolóhenger külső helyzetben
THEN SET      O0.2      'Emelőhengert lefelé mozgó szelep
      RESET     O0.3      'Kitolóhengert mozgó szelep

STEP 8
IF      I0.1      'Emelőhenger alaphelyzetben
THEN RESET     O0.2      'Emelőhengert lefelé mozgó szelep

STEP 9
IF      NOP
THEN JMP TO   1

```

### A program működése:

A feladat, tehát két munkahenger vezérlése. A program STEP-ekből áll, melynek lényege, hogy a munkaciklust felbontsuk elemi lépésekre. Ha a STEP-ben teljesül az előírt feltétel, csak akkor ugrik a következő lépésre, és így tovább.

A STEP 1-ből látható, hogy a program két fő részre bontható, ezek közötti váltást a Web-es felületről az F2.2 jelzőbit értékének állításával valósíthatunk meg, és engedélyeznünk kell a csomagemelő berendezés mellett található kapcsolóval a távoli vezérlés lehetőségét. Ha a kapcsolót nem fordítjuk el az 1-es állásba, akkor csak a program második fele (STEP3-STEP9) fut, amely a csomag továbbítását hivatott megoldani. A STEP 3-ban a folyamat indulási feltételeit vizsgáljuk. Mindkét hengernek alaphelyzetben kell lennie, a csomag megérkezése után a START gomb megnyomásával indulhat az emelés. A folyamat elindítható még ugyanebben a lépésben, ha engedélyezzük a távoli vezérlést a kapcsoló elfordításával és a PLC WEB TESZT 2 nevű HTML oldalon (25. ábra) lévő Ciklus START gombot megnyomta valaki (F1.0 jelzőbit értéke 1), és a csomag is a helyén van. Mindezen feltételek teljesülése esetén a következő STEP-be ugrik, egyébként, pedig vissza a STEP 1-re, ahol tovább vizsgálja, hogy melyik programrészt akarjuk használni.



Átugorva a STEP 4-re az emelőhengert lefelé mozgató szelepről technológiai megfontolásból levesszük a vezérlő jelet, ha a henger alsó véghelyzetében van. A következő STEP-ben az emelőhenger a leírt feltételek teljesülése esetén a felső véghelyzetébe megy ki. Valamint egy számláló aktuális értékét egyel, növeli, ami a PLC WEB TESZT 2 nevű HTML oldalon (25. ábra) -Csomagok száma- adatként jelenik meg. STEP 6-ban miután felemeltük a csomagot, a kitoló munkahengert működtető monostabil szelep, vezérlő jelet kap. A következő lépésben, miután külső véghelyzetébe ment a kitolóhenger, a monostabil szelep vezérlő jele megszűnik, így a kitolóhenger belső véghelyzetébe megy, az emelőhengert, pedig a bistabil szelepen keresztül, alsó véghelyzetébe indítjuk. Miután leért az emelőhenger, megszüntetjük a szelep vezérlését. Az utolsó STEP-ben pedig ha már az egész folyamat lejátszódott és nem történik semmi, a program futását visszaléptetjük az első STEP-re, ahol a már ismert folyamat játszódik le.

A program első fele a STEP 2 rész, szerepét tekintve ez az Internetes vezérléshez készült. Elemeit a PLC WEB TESZT 1 nevű HTML oldal (26. ábra) használja, jeleníti meg. A programrész lényege, hogy belső regisztereket (flag) használunk. A jelzőbitek értékeinek változásából információt kapunk a PLC be- és kimeneti állapotairól (rendszerfelügyelet), és tudjuk e jelzőbitek értékeit a webes felületen keresztül változtatni, ezzel a PLC működését befolyásolni (vezérlés). Látható a 26. ábrán, hogy a csomagemelő berendezésnél a munkahengerek szelepeit vezérelni tudjuk, pl.: Emelőhenger fel- Kitolóhenger be. Ezt egyszerűen a PLC programban az FW0-s kimeneti szó bizonyos értékre állításával érhetjük el. A STEP 2-ben látható, hogy a munkahengerek véghelyzet állapotait érzékelő szenzorok jeleit, jelző bitekbe íratjuk be. Erre magyarázatot szintén a PLC WEB TESZT 1 nevű HTML oldal ad, ugyanis ezen az oldalon kijelzésre kerül a hengerek helyzetei is (Lent, Fent, Kint, Bent). Ezeket az információkat, pedig pontosan azok a regiszter értékek szolgáltatják, amit mi előzőleg definiáltattunk a szenzorokkal.

És hogy hogyan történik mindezen adatok cseréje a PLC és a világ bármely részén lévő felhasználó között? Úgy, hogy ebben részt vesz mindazon technológia, amiről eddig, igaz csak szűk szavúan a TDK munkámban is szó esett.

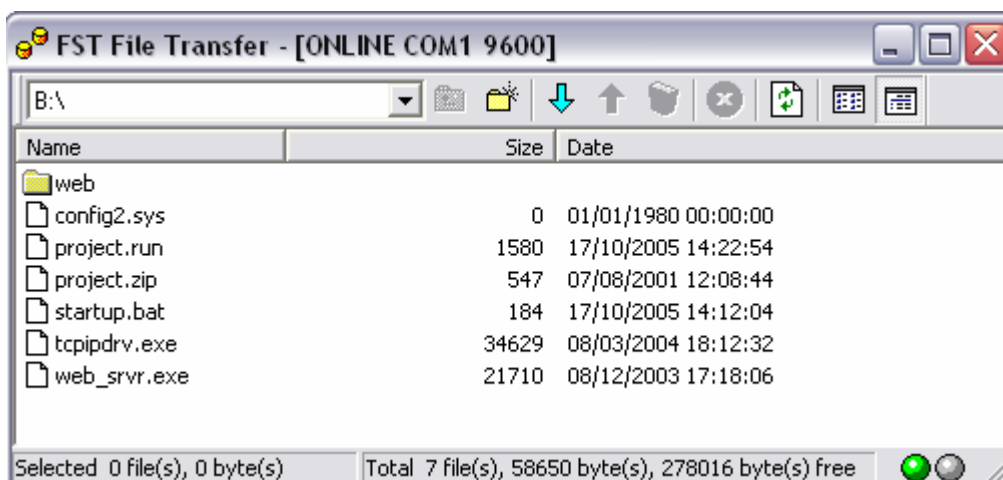
A rendszer szerves részévé válik a programozható vezérlőn kívül a TCP/IP- Internet- Web-technológia is. Hiszen a PLC-n futó program mellett szükség van a TCP/IP protokollokra, az Internet (Ethernet) hálózatra, annak elemeire, és egy Web-es felületre, amelyen keresztül működtethetjük az adott berendezést, vagy információkat gyűjthetünk a berendezés viselkedéséről.

## 10.2 Web-es felület

A Web oldalunkat bármilyen HTML-szerkesztővel elkészíthetjük, akár a hagyományos Microsoft FrontPage vagy a Dreamweaver MX használatával. Aki most lát neki az első weboldalának megszerkesztéséhez, annak az utóbbi programot javaslom. A Dreamweaver MX a Macromedia Dreamweaver, a díjmentes HTML-szerkesztő és webalkalmazás-fejlesztő eszköz legújabb változata. A Dreamweaver használatával rögtön láthatjuk a tervezés alatt álló weboldalt, és bármilyen működésre könnyedén rá is bírhatjuk. A weboldalak HTML nyelven íródnak, de rengeteg dolgot megtehetünk anélkül, hogy egyetlen sornyi HTML-t is látnunk kellene. A Dreamweaver egy „azt kapod, amit láatsz” típusú weboldal-szerkesztő, mely nagyon hatékony, ugyanakkor egyszerűen használható.

Az általános HTML nyelvbe, a FESTO PLC programozási nyelvben használt hivatkozásokat, utasításokat bele kell csempészni valahogyan, hogy a PLC és a weboldal, egymás számára érthető nyelven tudjon kommunikálni. Sajnos nem állt rendelkezésre olyan forrás, amelyben ez részletesen le lett volna írva. Egy létező demo verziójú FESTO weboldal kielemezése, valamint a PLC utasításlistás kódelemeinek megismerése után, sikerült olyan HTML-ben megírt weboldalt szerkeszteni, amellyel különféle folyamatok vezérelhetők, felügyelhetők. A példafeladat weboldalainak bemutatása közben könnyen megérthetők ezen utasítások.

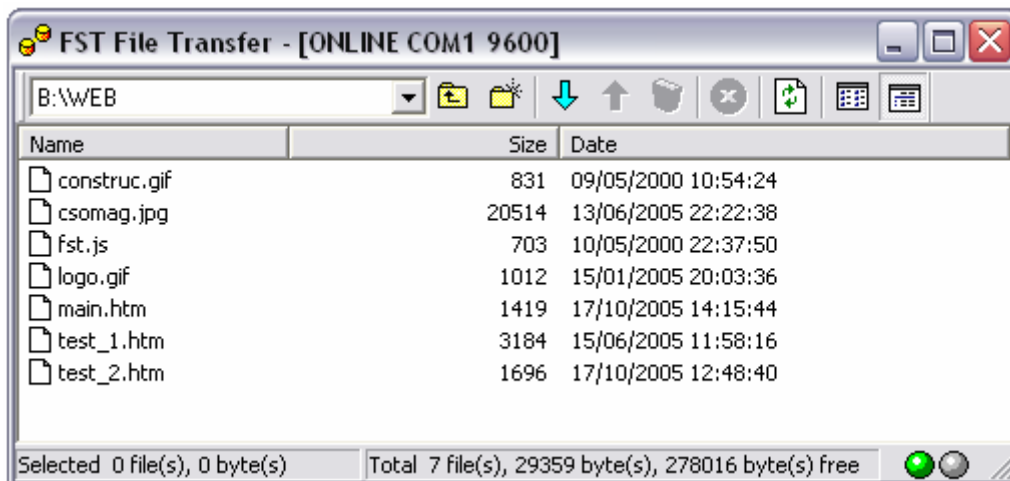
A legfontosabb, hogy a PLC-n lévő FTP szerver B könyvtárba (ahol a PLC működéséért felelős megírt program is tárolódik) egy web elnevezésű alkönyvtárt (22. ábra) hozzunk létre.



22. ábra PLC FTP szerver B könyvtára

A beágyazott webszerver ebből a könyvtárból (23. ábra) fog dolgozni, itt találja meg a felhasználó által kért oldalakat és az oldalhoz tartozó elemeket (képek, animációk, videók...).

Ebbe a könyvtárban kell elhelyezni a JavaScript számára értelmezhető fst.js fájlt is. Oda kell figyelniük még arra is, hogy amit főoldalként nyit meg böngészőnk, amikor elérjük a PLC Web-szerverét, annak main.htm nevűnek kell lennie, különben „A lap nem jeleníthető meg” hibaüzenetet kapunk. A további oldalakat, amikre a főoldalból jutunk, már tetszés szerint nevezhetjük el.



23. ábra WEB könyvtár

Most pedig nézzük meg a főoldal (21. ábra), azaz a main.htm oldal HTML szerkezetét:

1	<html>
2	<head>
3	<title>PLC Honlap</title>
4	<script language="JavaScript" src="fst.js">
5	</script>
6	</head>
7	<body>
8	
9	<p align="center"><span style="text-decoration: underline"><u>
10	<font color="#FF0000" size="5">FESTO&nbsp; PLC&nbsp; WEB&nbsp; TESZT- OLDALAK</font></u></span><font color="#008000" size="4"> 
11	Mechatronikai, Optikai és
12	Műszertechnikai Tanszék</font><font color="#008000"> 
13	<font size="4">Irányítástechnikai Laboratórium,&nbsp; V2 épület 007</font></font><hr color="#FF0000">

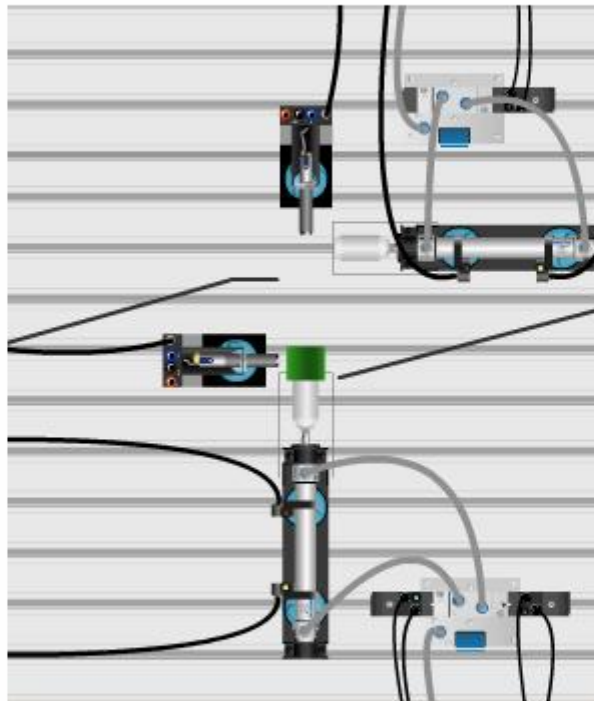


- 15. Kép megadása
- 17. Második határoló vonal meghatározása
- 19-21. Más oldalakra való egyszerű hivatkozások megadása
- 22-26. TESZT 1 nyomógomb meghatározása
- 28-32. TESZT 2 nyomógomb meghatározása.

## FESTO PLC WEB TESZT-OLDALAK

Mechatronikai, Optikai és Műszertechnikai Tanszék  
Irányítástechnikai Laboratórium, V2 épület 007

---



---

[TESZT 1](#)   [TESZT 2](#)   [FESTO](#)   [BME](#)

**24. ábra** main.htm oldal

Számunkra a main.htm oldal szerkezetéből a nyomógombok funkciója az érdelemes momentum. A TESZT 1 gomb esetén az, hogy megvalósít elsődlegesen egy másik lapra való váltást. A 24. sorban (1. Táblázat): `<a href="test_1.htm">` látható, azaz a test\_1.html-re ugrik böngészőnk, valamint a nyomógomb funkcióját kihasználva: `onClick="CISend('MW2', '4')` olvasható, azaz rákattintás hatására a PLC belső regiszterét felszólítja, hogy a 2-es kimeneti szó értéke 4-re íródjon át. A PLC programozásban a szavak kezelésére az FW megjelölés

használatos, a hivatkozásban mégis MW jelölést láthatunk. Az eltérés abból adódhat, hogy a PLC Flag angol kifejezést használja, a Web-es alkalmazás, pedig a Merker német kifejezést. TESZT 2 nyomógombnál szintén egy másik oldalra való ugrást biztosít a 30. sor (1. Táblázat): `<a href="test_2.htm">`, és a gombfunkció: `onClick="CISend('MW2', '0')`, azaz a 2-es kimeneti szó 0 értékűre íródik át.

Azzal, hogy ezt az értéket 0 vagy 4-re állítjuk, tudjuk a 2-es kimeneti szó egyes bitjeit változtatni, ezzel információt közölni a PLC-vel. Jelen esetben, a program STEP 1 részében az F2.2, azaz a 2-es kimeneti szó 2. bitjét figyelve döntjük el, hogy melyik programrészt akarjuk indítani. TESZT 1 esetén a munkahengerek külön-külön működtethetősége, TESZT 2 esetén, pedig egy ciklus elindítása és csomagok számolása megoldott.

A következő megszerkesztett oldal (25. ábra) a test\_2.html oldal, mely oldalra a főoldalról az imént látott módon juthatunk el, vagy a test\_1.html oldalról is ide átjuthatunk. A teljes HTML felépítés helyett most már inkább a vezérlés és visszajelzés számára lényeges dolgokat emelném ki.

A PLC működését tekintve, ha valaki engedélyezte a külső vezérlést és a honlapot felkeresve a főoldalról a TESZT 2 oldalra kattintott, akkor a PLC program a STEP 3 lépésnél áll és várja az indító jelet a csomagok emelésének megkezdéséhez. A használt jelzőbit az F1.0, a TESZT 2 weboldalon pedig a Ciklus START nyomógomb funkciója: `onClick="CISend('MW1', '1')`, vagyis az 1-es kimeneti szó 1 értéket vesz fel, jelen esetben az 1-es szó 0-s bitje 1 értéket vesz fel, azaz a csomagemelő berendezés működése elindul, ha az egyéb feltételek is teljesültek. A Ciklus STOP nyomógomb, e működés leállítását valósítja meg, a csomagtovábbító ciklus végeztével, úgy hogy az F1.0 értékét felülírja: `onClick="CISend('MW1', '0')`.

Látható még az oldalon (25. ábra) egy piros felirat, miszerint a „Csomag szállítása folyamatban.” Ez a funkció egy visszajelzést szolgál, információt ad, hogy a berendezés áll, vagy működik-e egyáltalán. A kiolvasás már egy kicsit bonyolultabb, de nem olyan vészes, ha rendelkezésünkre áll egy FESTO-s oldalakat kezelő JavaScript, ami egy HTML fájlba tervezett objektum-orientált nyelv. Alapjai a C és C++ nyelvek, így ezekre sokban hasonlít. Tulajdonképpen a JavaScript a weboldal motorja. Először is be kell olvastatni a JavaScript által felhasznált fst.js script-et. Ezt az eljárást már a weboldal fejlécében meg kell valósítanunk:

```
<script language="JavaScript" src="fst.js">  
</script>
```

A visszajelzés HTML-kódja:

```
<script language="JavaScript">
  switch ( <FSTCI dnm1> )
  {
    case 0: document.write("A berendezés nem üzemel!"); break;
    case 1: document.write("Csomag szállítása folyamatban."); break;
    default: document.write("Kapcsolódás"); break;
  }
</script>
```

azaz a FESTO FEC PLC belső regiszteréből kiolvassa az 1-es szó aktuális értékét. 0-ra „A berendezés nem üzemel!”, 1-re pedig „Csomag szállítása folyamatban.” feliratot írja ki, amit mi definiáltunk előzőleg. Köztes állapot esetén, amikor nem érkezik válasz a PLC felől (pl: a hálózat megterhelt, lassú) „Kapcsolódás” felirat olvasható.

A folyamatos adatkiolvasást az oldal állandó frissítésével érjük el. A HTML szerkezetében közvetlenül a fejrészbe tehetjük frissítő parancsunkat, meghatározva a frissítés gyakoriságát [sec]:

```
<meta http-equiv="Refresh" content="3;url=test_2.htm">
```

## FESTO PLC WEB TESZT 2

Mechatronikai, Optikai és Műszertechnikai Tanszék  
Irányítástechnikai Laboratórium, V2 épület 007

---

### Csomagok emelése

Ciklus START

Ciklus STOP

Csomag szállítása folyamatban.

Csomagok száma: 19

---

TESZT 1

[PLC HONLAP](#)

25. ábra test\_2.html oldal

Az eddig szállított csomagok számát (25. ábra) is kiolvashatjuk a PLC-ből, hisz e célból definiáltunk egy számlálót a STEP 2-ben, melynek értéke (CW0) minden egyes csomag emelése után eggyel nő. A HTML parancs: `<FSTCI dzw0>`, szintén szintaktikai eltérés mutatkozik a PLC programozásában ismert kód és a HTML-es kód között a számláló használatára. A PLC Counter angol kifejezést használja, a Web-es alkalmazás, pedig a Zähler német kifejezést.

Az aláhúzott PLC HONLAP feliratra kattintva, pedig visszajuthatunk a főoldalra.

TESZT 1 gomb megnyomására a böngésző betölti a test\_1.htm oldalt (26. ábra), és az F2.2 jelzőbit értékét felülírja: `onClick="CISend('MW2', '4')`, hogy a PLC váltson az első programrészre.

**FESTO PLC WEB TESZT 1**  
Mechatronikai, Optikai és Műszertechnikai Tanszék  
Irányítástechnikai Laboratórium, V2 épület 007

---

**Csomagok emelése**

<b>Emelőhenger</b>	<b>Kitolóhenger</b>
<input type="button" value="FEL"/> <input type="button" value="LE"/>	<input type="button" value="KI"/> <input type="button" value="BE"/>

---

<b>Emelőhenger helyzete</b>	<b>Kitolóhenger helyzete</b>
<b>Lent</b>	<b>Kint</b>

---

[PLC HONLAP](#)

26. ábra test\_1.htm oldal



A STEP 2-ben látható hogy a munkahengerek helyzeteit az FW0 kimeneti szó értékeivel határozhatjuk meg.

- Ha az FW0 értéke 2, az emelőhengert felfelé mozdító szelep vezérelt.
- Ha az FW0 értéke 3, az emelőhengert lefelé mozdító szelep vezérelt.
- Ha az FW0 értéke 4, a kitolóhengert működtető szelep vezérelt.

A fenti feltételekből következően adódik a nyomógombok funkciója:

FEL: `<input name="button2" type="button" onClick="CISend('MW0', '2')" value=" FEL">`

LE: `<input name="button3" type="button" onClick="CISend('MW0', '3')" value=" LE">`

KI: `<input name="button4" type="button" onClick="CISend('MW0', '4')" value=" KI">`

BE: `<input name="button" type="button" onClick="CISend('MW0', '0')" value=" BE">`

A hengerek véghelyzeteit érzékelő szenzorok jeleit a STEP 2 részben, jelzőbitek értékeinek feltöltésére használunk az alábbiak szerint:

Emelő henger helyzete	Kitoló	Memória bit értéke				Decimális értéke
		1.4	1.3	1.2	1.1	
Lent	Bent	0	1	0	1	10
Lent	Kint	1	0	0	1	18
Fent	Bent	0	1	1	0	12
Fent	Kint	1	0	1	0	20

**2. Táblázat**

Ezen kombinációk alapján, a már ismert memória kiolvasás alapján az emelőhenger helyzetét (23. ábra) leíró HTML-kód:

```
<script language="JavaScript">
  switch ( <FSTCI dmw1> )
  {
    case 10: document.write("Lent"); break;
    case 18: document.write("Lent"); break;
    case 12: document.write("Fent"); break;
    case 20: document.write("Fent"); break;
    default: document.write("Kapcsolódás"); break; }
</script>
```

A kitolóhenger helyzetét kiolvasó HTML-kód:

```
<script language="JavaScript">
  switch ( <FSTCI dmw1> )
  {
    case 10: document.write("Bent"); break;
    case 18: document.write("Kint"); break;
```

```
case 12: document.write("Bent"); break;
case 20: document.write("Kint"); break;
default: document.write("Kapcsolódás"); break; }
```

```
</script>
```

Látható a két leírásban, hogy ugyanazokat a decimális értékeket használjuk, csak mást íratunk ki a weboldalra, ennek csak annyi érdekessége van, hogy a legelőször meghatározott leíró sorrendet (jelen esetben: 10, 18, 12, 20), a következő meghatározásban is tartani kell. Ellenkező esetben nem kerül adat, kijelzésre. A test\_1.htm oldal HTML szerkezetében is megtalálható a már ismert oldalfrissítési eljárás:

```
<meta http-equiv="Refresh" content="3;url=test_1.htm">
```

Valamint a JavaScript működéséhez szükséges fst.js beolvasása is.

## 11 Végző

A webtechnológia optimalizálja az ember-gép kapcsolatot. Rendszerünket könnyen elérjük PC-nk vagy PDA-nk egyszerű internetböngészőjéről, a PLC-be integrált webszerveren keresztül. A felhasználók, a karbantartó csapatok könnyen elérhető megoldáshoz, valós idejű információkhoz jutnak. Nincs licence a „kliens” állomásokon és nincs szükség a felhasználók képzésére, a webes felület kezelése egyszerű. Jogosultsági szinteket állíthatunk fel, felhasználónév és belépési jelszó kérésével megszabhatjuk, hogy ki-milyen folyamatokat irányíthat.

Sok minden változott 1968, a PLC-konceptió kidolgozása óta. A ma technológiájában már megvalósított, hogy a PLC-k e-mailek, sms-ek küldésével adnak jelentést a változókról, eseményekről és riasztásokról.

Azzal viszont, hogy megnyitottuk a PLC-nket a világ felé, sajnos számolnunk kell, a világháló nyújtotta veszélyforrásokkal is. Vírusok; kémprogramok; férgek próbálják meg tönkretenni rendszereinket. A PLC-ben nincs megoldva ezek elleni védelem, ezért külön gondoskodnunk kell tűzfalokról, mely a saját hálózatot valóságos erőddé alakítja át a sebezhető kommunikációs portokat célzó hacker-ekkel és vírustámadásokkal szemben.

Munkám során megpróbáltam a hangsúlyt a kezelő felületben rejlő egyszerűség, és azt megvalósító technológia bemutatására helyezni. Maradt még számos nyitott kérdés, melyek megoldásával diplomamunkám során szeretnék foglalkozni.

## 12 Irodalomjegyzék

- [1.] Dr. Ajtonyi István: Programozható irányítóberendezések, hálózatok és rendszerek. Műszaki Könyvkiadó, Budapest 2002.
- [2.] Makk Attila: Hálózatépítés otthon. Kossuth Kiadó 2003.
- [3.] Stephen A. Thomas: IP kapcsolás és útválasztás. Kiskapu Kiadó 2002.
- [4.] Charles L. Hedrick: Introduction to the Internet Protocols. Computer Science Facilities Group 1987. Rutgers, New Jersey State University
- [5.] Andrew S. Tanenbaum: Számítógép-hálózatok. Panem Könyvkiadó Kft. Budapest 2004.
- [6.] Betsy Bruce: Tanuljunk meg a Macromedia Dreamweaver MX használatát 24 óra alatt. Kiskapu Kiadó 2002.
- [7.] Schneider Electric: Ethernet TCP/IP és a web. Határok nélküli barangolás egy univerzális hálózaton. Katalógus 2005.
- [8.] FESTO: A VEEP és FEC PLC gyakorlólhely kézikönyve.