

PLC PROGRAMOZÁS 2



SZEGEDI TUDOMÁNYEGYETEM
MÉRNÖKI KAR

KÉSZÍTETTE: CSIKÓS SÁNDOR

SZÉCHENYI 



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

- PLC Programozás 1 alatt elsajátított tudás továbbfejlesztése
- LabVIEW programozási ismeretek továbbfejlesztése
- SCADA rendszerek felépítése, tervezése és kivitelezése
- Felsorolt technológiák ipari alkalmazása

MIT NEVEZÜNK SCADA-NAK

SCADA (Supervisory Control And Data Acquisition) magyarul Felügyelő Szabályozás És Adatgyűjtés egy olyan rendszer ami nagy területeken elterülő folyamatokat figyel/ irányít egy központi helyiségből. Az irányítás nem teljes csak felügyelő jellegű nem alkalmas vészkapcsolók vagy más biztonságtechnikai beavatkozásra. Reakcióidőbeli besorolásában tartalmaz valós idejű és nem valós idejű elemeket. Valós idejűnek nevezünk egy rendszert ami definiál egy maximális időt interrupt request és interrupt request-re reagálás között és garantáltan ezen időn belül reagál. Ez a maximális idő SCADA esetében a feladattól függ.

SCADA RENDSZER FELADATA

- Lehetővé teszi, hogy az operátor egy központi helyről célértékeket változtasson, szelepeket nyisson/zárjon, riasztásokat figyeljen és mérési információkat gyűjtsön egy olyan folyamatról ami területileg nagy (akár több ezer km).
- Miután ez az alap funkciókat ellátjuk a SCADA rendszer segítségével módosíthatunk a szabályozási eljárásainkon és biztonságosabbá tehetjük az üzem működését.

TÖRTÉNELMI FEJLŐDÉS

A SCADA rendszerek kezdetét a telemetria rendszerek jelentették, bár a telemetria szó maga távoli megfigyelést jelent az ipar kibővítette beavatkozási lehetőséggel. Szükség volt kifejlődésére mivel a nagy területen elszórt berendezések felügyelete túl lassú és hatástalan, ha mindhez felügyeleti személyeket küldünk ki. Ezek a telemetria rendszerek főleg vezetékes megvalósításúak voltak majd rádiós megvalósítást is alkalmaztak ahol lehetett. Kb 1960-ig elterjedtek az elektromos művekben és az olajiparban.

TÖRTÉNELMI FEJLŐDÉS

1970-es években már a SCADA kifejezést használták a kétirányú telemetria leírására még a telemetria szó a távoli adatgyűjtésre maradt. A számítógépek fejlődése miatt már nem csak a nagy terepeknél érte meg gazdaságilag SCADA rendszert alkalmazni, így az MTU (Master Terminal Unit) elhelyezése oda került ahol működésileg a legjobb. Mivel a rádió sávok mára már eléggé elfoglaltak nehéz új rádiós megoldást alkalmazni így alkalmazunk manapság telefon, szatelit, ethernet és optikai szálak megoldásokat adatok továbbítására. A jövő főleg a LAN rendszerek felé mutat.

ÉRZÉKELŐTŐL PANELIG SCADA



ÉRZÉKELŐTŐL PANELIG SCADA

- Ebben az esetben az érzékelőktől elvezetjük egy gyűjtő panelig a jeleket és kapcsolókat
- Az ilyen rendszer előnye, hogy olcsó és nem igényel számítógépet
- Hátrányai közül néhány:
- Nem skálázható könnyen, limitált hatótáv, a rendelkezésre álló adatok csak alapadatok nem származtatottak és a nap 24 órájában figyelni kell.

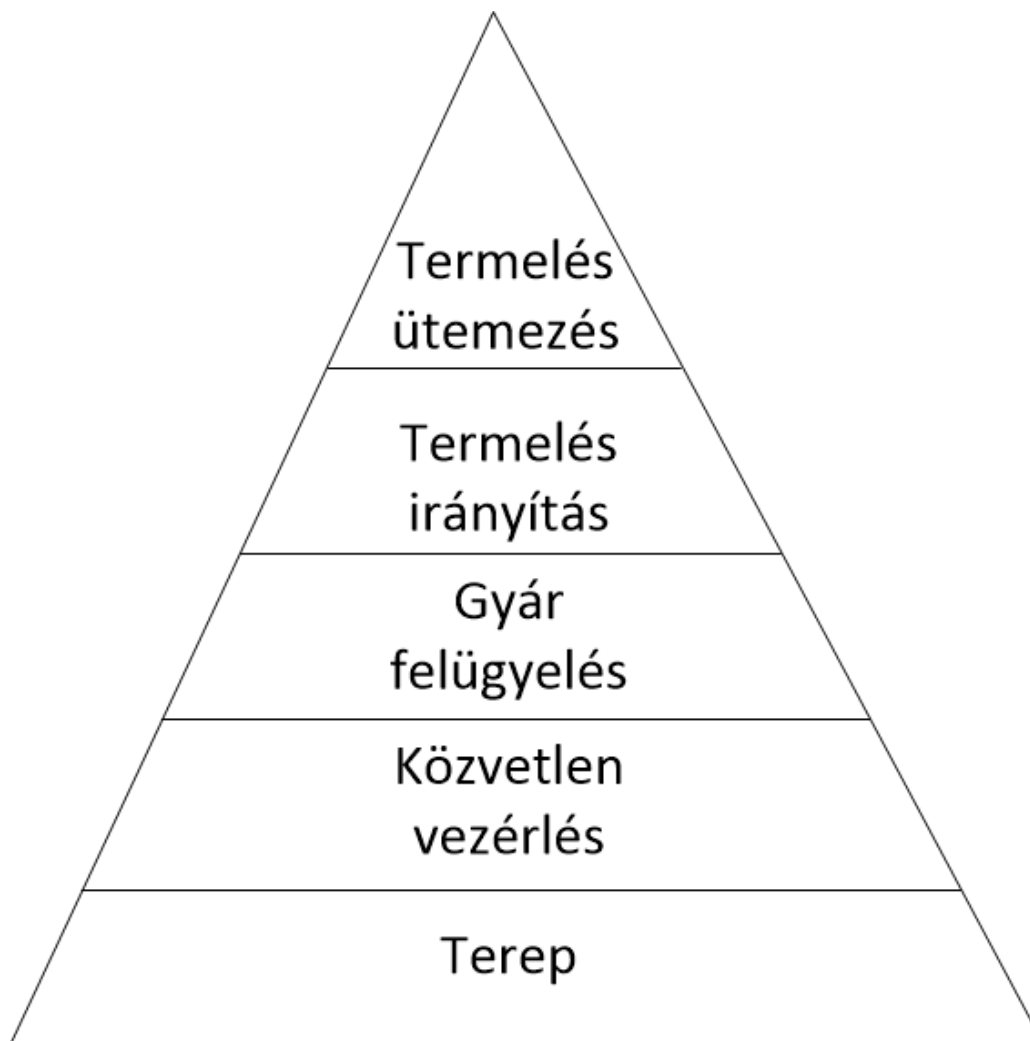
MODERN SCADA



MODERN SCADA

- Itt a távolság nem korlátoz akár a föld másik feléről is irányíthatunk. Nem vagyunk helyhez kötve.
- Az adatokat bárhogy feldolgozhatjuk és ábrázolhatjuk.
- Sokkal nagyobb számú érzékelővel dolgozhatunk.
- Hátrányok: programozás és rendszer technikus képzettség szükséges. Csak az RTU-ig látunk el. Sok a kábel és komplikáltabb mint az előző megoldás.

AUTOMATIZÁLÁSI PIRAMIS



SCADA RENDSZEREK FELÉPÍTÉSE

SCADA rendszereknél a hierarchiát legjobban az automatizálási piramis mutatja be. A legalsó szint a terepi eszközök szintje ezek az érzékelők és aktuátorok amik a rendszer paramétereit mérik illetve a beavatkozást végzik, ezek közvetlen összeköttetésben vannak a vezérlőkkel ami a következő szintje a piramisnak.

SCADA RENDSZEREK FELÉPÍTÉSE

A közvetlen vezérlőkhöz tartoznak a PLC-k, RTU-k (Remote Terminal Unit), IED-k (Intelligent Electronic Device) amik a terepi eszközökhöz közvetlenül kapcsolódnak. Ezeknek rendelkezniük kell kommunikációs lehetőséggel, hogy a következő szinttel összeköttetésben legyenek.

SCADA RENDSZEREK FELÉPÍTÉSE

- A gyár felügyelés szintjén található az MTU (Master Terminal Unit). Ezen az egységen látjuk a SCADA rendszer által összegyűjtött és átdolgozott adatokat.
- Tipikusan ezen a gépen felhasználó kezelés van így szabályozzuk, hogy kik befolyásolhatják a rendszer működését.

SCADA RENDSZEREK FELÉPÍTÉSE

- Termelés irányítás feladata, hogy a megadott termelő berendezéseken meghatározza, hogy mikor mit termelünk úgy, hogy az gazdaságilag kifizetődő legyen.
- A termelés irányítás meghatározza, hogy mely berendezéseken, kik, hány órában dolgoznak, hogy biztosítani tudjuk a termék legyártását időre. Erre a feladatra külön software-eket használunk mint a FlexSim

SCADA RENDSZEREK FELÉPÍTÉSE

- Termelés ütemezés feladata, hogy az összegyűjtött rendeléseket kiossza a termelés irányító egységeknek úgy, hogy logisztikailag kivitelezhető legyen és gazdaságilag kifizetődő a termelés.
- Ide tartozik még a raktárkészletek kezelése is, ha várhatóan valamely termék elkészítéséhez szükség van olyasmire ami nincs raktáron ezt megrendeli, történhet automatikusan illetve manuálisan is.

SCADA RENDSZEREK FELÉPÍTÉSE

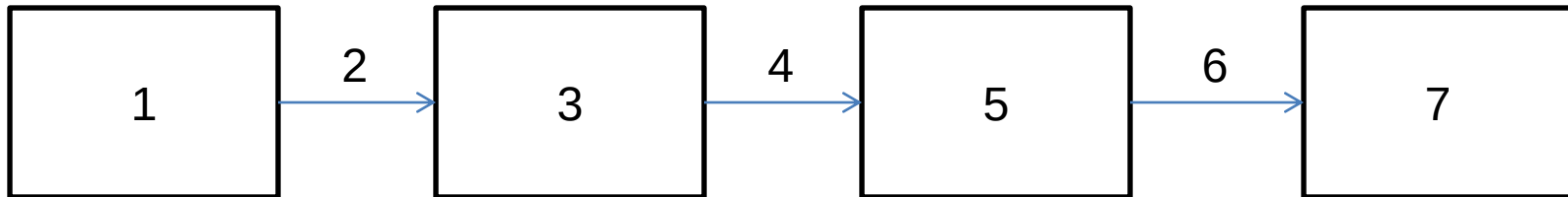
- A felsorolt rétegeket össze kell kötni, hogy egymással kommunikálni tudjanak így ki kell alakítani egy hálózati infrastruktúrát ami összeköti az egyes szinteket az automatizálási piramisban és megakadályozza az illetéktelen hozzáféréseket (pl: termelés ütemező nem kell, hogy módosítson értékeket a gyártási folyamatban)

- SCADA rendszerekhez hasonlóak a DCS (Distributed Control System) rendszerek. Feladatukban hasonlóak néhány fontos eltéréssel:
 - DCS rendszerekben a vezérlés nem felügyelő hanem teljes
 - Decentralizált
 - Az érzékelők és aktuátorok rendelkeznek „intelligenciával” és képesek a hálózati kommunikációra

SCADA VS DCS

- Mindkét rendszernek megvan az előnye és hátránya így nehéz megmondani melyik a jobb, de nem is kell mivel kiépítésük főleg gazdasági döntés. DCS rendszerek drágábbak mivel különleges eszközöket igényelnek még a SCADA rendszerek nyíltabbak.

INFORMÁCIÓ KÓDOLÁS



Az információ átviteli rendszer elemei:

1. Információ forrás
2. Kódolás
3. Adó
4. Átviteli csatorna
5. Vevő
6. Dekódolás
7. Információ rendeltetés

INFORMÁCIÓ KÓDOLÁS

Kódnak nevezzük azokat a rendszereket amik információhalmaz kifejezésére, közlésére, megjelenítésére szolgál. A kód elemei:

1. Szimbólumkészlet (kételemű (pl.:bináris), többelemű (pl.:abc))
2. Kódszavak (adott számú szimbólumból állnak (pl.:byte mindig 8) vagy változó számúból (pl.:magyar szavak))
3. Kódszókészlet (egy rendszeren belül képezhető összes szimbólum kombináció. Ezek egy része megengedett a többi része nem. Ha egy rendszerben a teljes készlet megengedett akkor a rendszer redundancia mentes, egyébként redundáns.

INFORMÁCIÓ KÓDOLÁS

A kódokat tartalmuk szerint két csoportra oszthatunk alfanumerikusra és számkódokra.

Alfanumerikus kód-al betűk, számok, írásjelek és szóközök fejezhetőek ki azaz írásos információk kódolására használható. Könnyen alkothatunk egy ilyen kódot, de célszerűbb a bevezetett rendszereket alkalmazni.

INFORMÁCIÓ KÓDOLÁS

Talán a legrégebbi alfanumerikus kód a morse-kód, ami időtartamban megkülönböztethető (rövid-dot, hosszú-dash), bináris elemekből álló, változó hosszúságú kódszókészlettel 39 alfanumerikus karaktert kódol. A dot egy időegységig tart még a dash 3-ig, két karaktert egy egységnyi csend választ el. A szimbólumok a szavakban használt gyakoriságuk alapján kapták a kódjukat, a leggyakrabban az angol abc-ben az e és a t ezért azok kódja a legrövidebb. Még igen zajos körülmények között is működik.

INFORMÁCIÓ KÓDOLÁS

A ● ■■■
B ■■■ ● ● ●
C ■■■ ● ■■■ ●
D ■■■ ● ●
E ●
F ● ● ■■■ ●
G ■■■ ■■■ ●
H ● ● ● ●
I ● ●
J ● ■■■ ■■■ ■■■
K ■■■ ● ■■■
L ● ■■■ ● ●
M ■■■ ■■■
N ■■■ ●
O ■■■ ■■■ ■■■
P ● ■■■ ■■■ ●
Q ■■■ ■■■ ● ■■■
R ● ■■■ ●
S ● ● ●
T ■■■

U ● ● ■■■
V ● ● ● ■■■
W ● ■■■ ■■■
X ■■■ ● ● ■■■
Y ■■■ ● ■■■ ■■■
Z ■■■ ■■■ ● ●

1 ● ■■■ ■■■ ■■■ ■■■
2 ● ● ■■■ ■■■ ■■■
3 ● ● ● ■■■ ■■■
4 ● ● ● ● ■■■
5 ● ● ● ● ●
6 ■■■ ● ● ● ●
7 ■■■ ■■■ ● ● ●
8 ■■■ ■■■ ■■■ ● ●
9 ■■■ ■■■ ■■■ ■■■ ●
0 ■■■ ■■■ ■■■ ■■■ ■■■

INFORMÁCIÓ KÓDOLÁS

A Baudot vagy telex kód a géptávíró rendszerek szabványos kódja. A rendszer 5 bináris elem és azok két kijelölt kombinációja alkalmazásával 56 karaktert kódol.

INFORMÁCIÓ KÓDOLÁS

[illegible]

INFORMÁCIÓ KÓDOLÁS

ASCII (American Standard Code for Information Interchange) a legelterjedtebb adatátviteli kód. Hét bináris elemmel 128 karaktert kódol ez kiegészíthető egy 8. bittel ellenőrzési célból vagy kiterjesztett ASCII esetén újabb 128 karakterrel bővíthető. Az első két bit 4 kombinációja 32 tagos csoportokra bontja a vezérlő karaktereket, írásjeleket, kis és nagybetűket, speciális karaktereket, számokat és írásjeleket.

INFORMÁCIÓ KÓDOLÁS

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

INFORMÁCIÓ KÓDOLÁS

128	Ç	144	É	160	á	176	☐	192	⌞	208	⌞	224	α	240	≡
129	ü	145	æ	161	í	177	☐	193	⌞	209	⌞	225	β	241	±
130	é	146	Æ	162	ó	178	☐	194	⌞	210	⌞	226	Γ	242	≥
131	â	147	ô	163	ú	179		195	⌞	211	⌞	227	π	243	≤
132	ä	148	ö	164	ñ	180	⌞	196	⌞	212	⌞	228	Σ	244	∫
133	à	149	ò	165	Ñ	181	⌞	197	⌞	213	⌞	229	σ	245	∫
134	â	150	û	166	°	182	⌞	198	⌞	214	⌞	230	μ	246	÷
135	ç	151	ù	167	°	183	⌞	199	⌞	215	⌞	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	⌞	200	⌞	216	⌞	232	Φ	248	°
137	ë	153	Ö	169	⌞	185	⌞	201	⌞	217	⌞	233	⊗	249	·
138	è	154	Ü	170	⌞	186	⌞	202	⌞	218	⌞	234	Ω	250	·
139	ì	155	◊	171	½	187	⌞	203	⌞	219	■	235	δ	251	√
140	î	156	£	172	¼	188	⌞	204	⌞	220	■	236	∞	252	∞
141	ï	157	¥	173	¡	189	⌞	205	=	221	■	237	φ	253	z
142	Ä	158	£	174	«	190	⌞	206	⌞	222	■	238	ε	254	■
143	Å	159	f	175	»	191	⌞	207	⌞	223	■	239	∧	255	

Source: www.LookupTables.com

INFORMÁCIÓ KÓDOLÁS

EBCDIC (Extended Binary Coded Decimal Interchange Code) egy IBM fejlesztés ami 8 biten kódol. 1963-64-ben fejlesztették ki, fő előnye az volt, hogy egyszerűbb lyukkártyákon használni mert a lyukak távolsága miatt nem gyengíti annyira el a lyukkártyát mint pl az ASCII kódolás esetén, ennek ellenére az ASCII-t kényelmesebb használni a rendezettebb felépítése miatt. Sok negatív kritikát kapott a kellemetlen használata miatt ezért és mivel csak lyukkártyákon van előnye kiesett a használatból.

INFORMÁCIÓ KÓDOLÁS

Unicode egy olyan szabvány ami egy kóddal tartalmazza a világ legtöbb írásjelét. 136755 karaktert tartalmaz 139 dialektusból. Több szabványos verziója van UTF-8, UTF-16, UTF-32, ahol a szám az egy karakter kódolására felhasznált biteket jelzi. A leggyakrabban használt az UTF-8 mivel visszafele kompatibilis az ASCII-val ez abból adódik, hogy az első 128 karakter megegyezik.

INFORMÁCIÓ KÓDOLÁS

Szám kódok csak numerikus információt kódolnak, lehetnek végtelen (folyamatos) illetve véges eleműek. Folyamatos kódok ábrázolási tartománya nincs korlátozva, ezek közül a legegyszerűbb a tiszta bináris azaz a kettes számrendszer.

A bináris kód súlyozott mivel tetszőleges n hosszú kódból meghatározhatjuk a képviselt decimális értékét (D) a következő összefüggéssel:

INFORMÁCIÓ KÓDOLÁS

$$D = \sum_{i=0}^{n-1} x_i s_i$$

Ahol x_i a bináris érték $[0,1]$, s_i a súly ami 2^i .
Ugyanakkor létezik úgynevezett egységlépéses kód ami nem súlyozott, azaz nem érvényes rá a fenti képlet. Ezeknél az egymást követő kódszavak csak 1 bitben térnek el, így a kódolást megvalósító technikai megoldás során fellépő zavarok csökkennek. Legismertebb ilyen kód a Gray, Watts, Johnson kódok.

Véges elemű kódoknál az ábrázolható elemek száma és így az ábrázolási tartomány is korlátos. Ezek közül a legismertebb a BCD (Binary Coded Decimal) ami 4 kódolt elemmel a decimális szimbólumkészletet írja le (0-9). Mivel vannak nem felhasznált kombinációk így redundáns, ezért sok változata létezik

INFORMÁCIÓ KÓDOLÁS

Végtelen elemű	Bináris (8-4-2-1)				Gray	Watts	Johnson
0	0000				0000	0000	0000
1	0001				0001	0001	0001
2	0010				0011	0011	0011
3	0011				0010	0010	0111
4	0100				0110	0110	1111
Véges elemű	BCD (8-4-2-1)	(5-4-2-1)	Aiken (4-2-2-1)	(8-4-(-2)-(-1))	Stibitz		
0	0000	0000	0000	0000	0011		
1	0001	0001	0001	0111	0100		
2	0010	0010	0010	0110	0101		
3	0011	0011	0011	0101	0110		
4	0100	0100	0110	0100	0111		
5	0101	1000	1001	1011	1000		
6	0110	1001	1100	1010	1001		
7	0111	1010	1101	1001	1010		
8	1000	1011	1110	1000	1011		
9	1001	1100	1111	1111	1100		

INFORMÁCIÓ KÓDOLÁS

Az ismétlődő vagy kettő hatványával nem egyező súlyozású kódok (pl:5-4-2-1) csak külön megállapodással tehetők egyértelművé pl: úgy, hogy a 4-nél nagyobb számokat képviselő kombinációk egyessel kezdődnek.

INFORMÁCIÓ KÓDOLÁS

Önkomplementáló egy kódrendszer, ha bármely kódszó 1-es komplemente a hozzárendelt decimális érték 9-es kiegészítőjét adja. Pl: a decimális 3 Aiken kódja 0011 ennek az 1-es komplemente (negáltja) 1100 Aiken kódban a decimális 6 ($3+6=9$). Ezeknél a súlyzótényezők összege 9.

BIG ENDIAN - LITTLE ENDIAN

- Big endian byte sorrend– a legnagyobb helyértékű byte-ot írjuk elsőnek.
 - Little endian byte sorrend– a legkisebb helyértékű byte-ot írjuk elsőnek.
 - Big endian bit sorrend– a legnagyobb helyértékű bit-et írjuk elsőnek.
 - Little endian bit sorrend– a legkisebb helyértékű bit-et írjuk elsőnek.
- A bit és byte endian sorrend eltérhet.

INFORMÁCIÓ KÓDOLÁS

Az információ továbbítása során zavarok hatására torzulások léphetnek fel. A mi esetünkben ez annyit jelent felcserélődnek a 0 és 1 értékek. Előny ha az alkalmazott kódrendszer lehetővé teszi az ilyen hibák detektálását vagy javítását, csak olyan kódrendszerek jöhetnek számításba amik rendelkeznek nem használt (redundáns) kombinációkkal. A redundancia mértéke összefüggésben van a hiba felismerési és javítási tulajdonsággal.

ÁTVITELI HIBÁK KEZELÉSE

- Adat átviteli hibák még legjobb körülmények között is előfordulnak ezért van szükségünk a hibák detektálására (hogyan tudjuk mikor kell újraküldeni egy üzenetet) és hiba javító kódokra (hogyan az üzenet olyan módon legyen kódolva, hogy a hiba ellenére is megmaradjon az adattartalma).
- A gyakorlatban a hibák kezelését az OSI modell Adat kapcsolati illetve Szállítási rétege végzi.

ÁTVITELI HIBÁK KEZELÉSE

- Hibának nevezzük azt amikor a fogadott adatcsomag nem azt tartalmazza mint amit elküldtünk. Ezek a hibák számos okból előfordulhatnak (elektromágneses zaj, sérülés a közvetítő közegben...)
- A hibákat felbontjuk:
 - 1 bitnyi hibára (Single bit error)
 - Több bitnyi hibára (Burst error)
- Csak azzal a kódoláson lehet hibát kezelni ami rendelkezik redundáns bitekkel, ha nincs hozzá kell rakni.

HIBÁK DETEKTÁLÁSA: VRC

- Paritásbittel (Vertical Redundancy Check, VRC):
- Lehet páros vagy páratlan (even/odd) megszámoljuk a keretben lévő adatbiteket és kiegészítjük úgy, hogy a kiválasztott paritásnak megfeleljen. Csak páratlan bit meghibásodásának észlelésére alkalmazható, viszont nem tudjuk, hogy melyik bit hibásodott meg. 1 bites hibáknál alkalmazzák.

HIBÁK DETEKTÁLÁSA: CHECKSUM

- Checksum (Ellenőrző összeg):
- Ebben az esetben az üzenetet felbontjuk k darab n bitből álló részre. Ezeket összeadjuk 1-es komplementens szabályai szerint majd az eredményt komplementáljuk. Ez az érték lesz a checksum.
- A vevő fogadja az adatot és a checksumot, elvégzi ugyanezeket a műveleteket, ha az új checksum csak 0-ból áll akkor elfogadott a csomag, ha nem akkor hibás.

HIBÁK DETEKTÁLÁSA: CHECKSUM PÉLDA

- Checksum példa:

1. keret	0	1	0	0	1	0	1	1
2. keret	1	1	0	1	0	1	1	0
összeg	0	0	1	0	0	0	0	1
checksum	1	1	0	1	1	1	1	0

1. keret	0	1	0	0	1	0	1	1
2. keret	1	1	0	1	0	1	1	0
3. keret	1	1	0	1	1	1	1	0
összeg	1	1	1	1	1	1	1	1
checksum	0	0	0	0	0	0	0	0

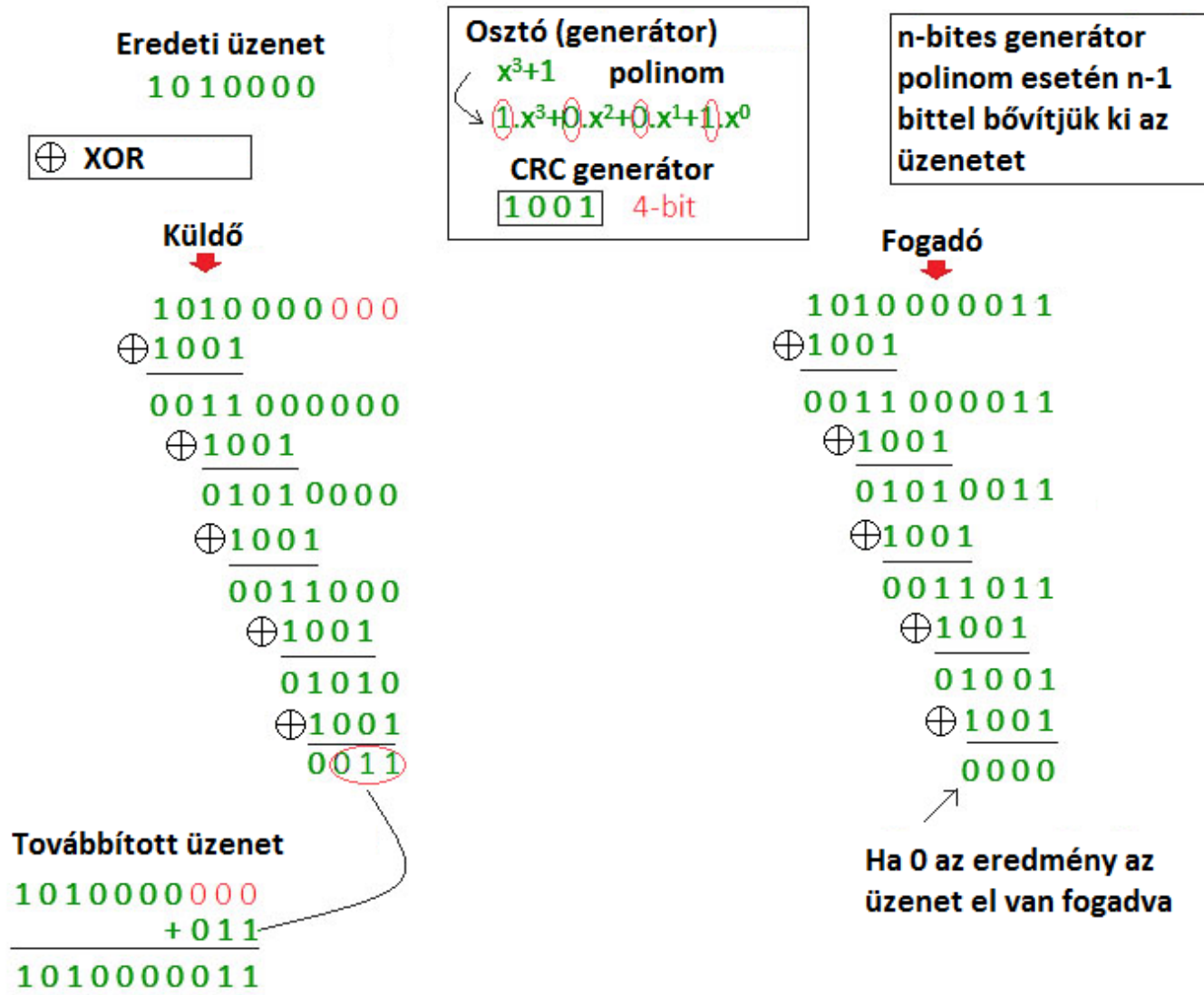
- Azonos oszlopban, ha 2 bit felcserélődik nem veszi észre.

HIBÁK DETEKTÁLÁSA: CRC

- CRC (Cyclic Redundancy Check):
- Az adatcsomagot bináris polinomnak tekintjük és egy előre meghatározott bináris számmal (**generátor**) elosztjuk (polinomok osztása binárisan XOR) és a maradékot hozzáadjuk a csomag végére.
- Az így megkapott csomagot a fogadó ugyanúgy elosztja a ugyanazzal a bináris számmal, ha a maradék 0 akkor elfogadja egyébként hibás.
- A generátor hosszától függ mennyi hibát tudunk detektálni, n bites generátor esetén maximum n bites hibát, ettől hosszabb hibát $\left(1 - \frac{1}{2^n}\right)$ valószínűséggel észlelünk.

HIBÁK DETEKTÁLÁSA: CRC

PÉLDA



HIBÁK DETEKTÁLÁSA: ARÁNYKÓD

- Aránykód:
- n-ből m típusú kódokat nevezünk aránykódnak. Ezt azt jelenti, hogy a kódszavak hossza lehet 1-es.
- A kódszavak száma így $\binom{n}{m}$

	5-ből 2					7-ből 2						
Helyérték	7	4	2	1	0	5	0	4	3	2	1	0
0	1	1	0	0	0	0	1	0	0	0	0	1
1	0	0	0	1	1	0	1	0	0	0	1	0
2	0	0	1	0	1	0	1	0	0	1	0	0
3	0	0	1	1	0	0	1	0	1	0	0	0
4	0	1	0	0	1	0	1	1	0	0	0	0
5	0	1	0	1	0	1	0	0	0	0	0	1
6	0	1	1	0	0	1	0	0	0	0	1	0
7	1	0	0	0	1	1	0	0	0	1	0	0
8	1	0	0	1	0	1	0	0	1	0	0	0
9	1	0	1	0	0	1	0	1	0	0	0	0

HIBÁK JAVÍTÁSA: LRC

- 2D Paritás ellenőrzés (Longitudinal Redundancy Check, LRC):
- Helyezzük az adatkereteket egymás alá. Mindegyik keretnek megvan a saját paritás bitje valamint minden oszlopnak is megvan a saját paritásbitje, így alkotunk egy új keretet ami az oszlopok paritásbitjeiből áll.

1. keret	0	1	0	0	1	0	1	1	0
2. keret	1	1	0	1	0	1	1	0	1
Oszlop paritás	1	0	0	1	1	1	0	1	1

HIBÁK JAVÍTÁSA: PARITÁS PÉLDA

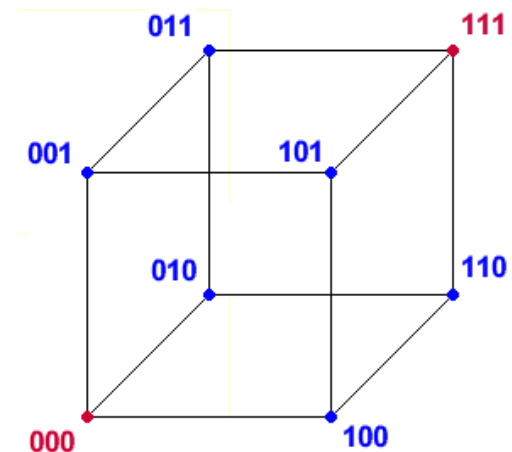
- Nem képes detektálni a következő jellegű hibákat:

1. keret	0	1	0	0	1	0	1	1	0
2. keret	1	1	0	1	0	1	1	0	1
3. keret	0	1	1	0	1	1	0	0	0
Oszlop paritás	1	1	1	0	0	0	0	1	1

- Ha a pirossal jelölt biteket mind az ellenkezőjére váltjuk nem változnak a paritások
- A zölddel jelölt 3 bites hibáknál se jelez, de ezek ritkábbak. Sok ellenőrzés kell.

HIBÁK JAVÍTÁSA: HAMMING TÁVOLSÁG

- Az eddigi kódolásoknál is megadható egy úgynevezett Hamming távolság ami annyit takar minimálisan hány bitet kell megváltoztatnom, hogy egy másik értelmes kódszavat kapjak. Vizuálisan ezen a kockán láthatjuk. Minden él egy bit változtatása.
- Vörös kódok érvényes kódszavak
- Kék kódok esetén hibát észlelünk és javíthatjuk a legközelebbi érvényesre.



HIBÁK JAVÍTÁSA: HAMMING TÁVOLSÁG

- Az példán látható, hogy 000-ból 111-et kapjunk 3 bitnek kell egyszerre meghibásodnia, hogy az üzenet átalakuljon a másikba így a Hamming távolsága 3, de csak 1 bitnyi információt tartalmaz.
- Azok a kódok amik kihasználják ezt a tulajdonságot a **$m+n$** felépítésűek (**m** információ bitet tartalmaz és **n** kiegészítő bitet). Az ilyen kódok akkor tudják a hibás bitet meghatározni ha teljesül a következő feltétel $2^m \geq m + n + 1$

HIBÁK JAVÍTÁSA: HAMMING KÓD

- Az egyik különleges $m+n$ felépítésű kód a Hamming kód ahol az m kiegészítő bit a paritás információját tartalmazza bizonyos elemeknek. BCD kódok esetén a következő a felépítése:

- $m1|n4,n2,n1$
- $m2|n4,n3,n1$
- $m3|n4,n3,n2$

	n4	n3	n2	m3	n1	m2	m1
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	0	1	1	0	0	1
3	0	0	1	1	1	1	0
4	0	1	0	1	0	1	0
5	0	1	0	1	1	0	1
6	0	1	1	0	0	1	1
7	0	1	1	0	1	0	0
8	1	0	0	1	0	1	1
9	1	0	0	1	1	0	0

HIBÁK JAVÍTÁSA: HAMMING KÓD

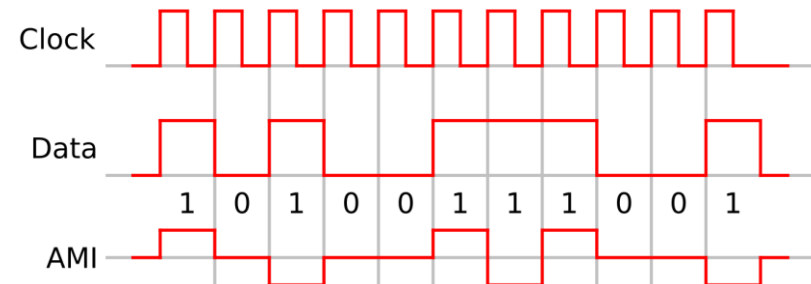
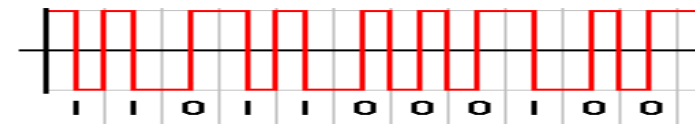
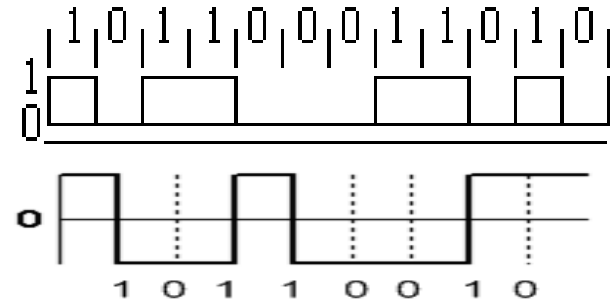
- Az m-ek ebben az esetben jobbról balra haladva a 2^0 ; 2^1 ; 2^2 helyeket foglalják el az adatot hordozó bitek közé vannak beszúrva, ilyen konfigurációban a hibás paritásbitek összege határozza meg a hibás bitet pl:

	n4	n3	n2	m3	n1	m2	m1
0	1	0	0	0	0	0	0

- Ebben az esetben n4 hibás így sérti m1, m2, m3 paritásbiteket ezek összege $1+2+4=7$ azaz a 7. bit hibás ezt invertálva megkapjuk az eredeti üzenetet.
- Hamming kódokat még tökéletes 1 bites hibák javítókódjának nevezik

CSATORNA KÓDOLÁS: KÓDOLÁSOK

- NRZ – Non Return to Zero
- NRZI – NRZ Inverted
- Manchester
- AMI – Alternate Mark Inversion



INFORMÁCIÓ KÓDOLÁS: UART

Ahhoz, hogy tudjunk adatokat átküldeni bármin kell egy meghatározott kódolás amit a küldő és fogadó is ismer. Ezekből több létezik (UART/USART, I2C, SPI), számunkra a legfontosabb az UART (Universal Asynchronous Receiver Transmitter) amit soros adatátvitelnél használunk.



INFORMÁCIÓ KÓDOLÁS: UART

UART esetében az küldött adatcsomag a következő képen épül fel. Elsőként kiküldünk egy start bitet, ezzel jelezzük, hogy megkezdjük a kommunikációt. Mivel a kódolás aszinkron azaz nincs szinkronizáló órajel így ezt a startbitet használjuk arra, hogy jelezzük adatot akarunk küldeni, a vevő álljon készenlétben a fogadásra. Következőnek kiküldjük bitenként az adatot amit szeretnénk. Az adat állhat 7 illetve 8 bitből is és a legkisebb helyértékű bittel kezdjük a küldést.

INFORMÁCIÓ KÓDOLÁS: UART

Az adatbitek után következik a paritás bit. A paritás bit arra szolgál, hogy ellenőrizzük az adatbitek továbbításában történt páratlan/páros számú bit meghibásodás

- A SCADA rendszerek fő eleme az adatátvitel ehhez szükségünk van:
 - Egy **közvetítő közeg** (amin az adat megy valamilyen időben változó tulajdonságként)
 - Egy **kommunikációs szabványra** (Ez meghatározza az alkalmazott fizikai jel szinteket)
 - Egy **kommunikációs protokollra** (hogyan kell értelmezni ezt az időben változó tulajdonságot. Ez meghatározza az alkalmazott fizikai jel szinteket és hogyan kódoljuk az adatot)
 - Legalább egy küldő és fogadó ami ismeri a protokollt

ADATÁTVITEL JELLEMZŐI

- Digitális sávszélesség
- Analóg sávszélesség
- SNR (jel/zaj arány)
- Irány

ADATÁTVITEL JELLEMZŐI: SÁVSZÉLESSÉG

- Alapsáv: diszkrét elektromos jelek továbbítására használják. Itt a sávszélesség bps-ben van meghatározva. Tipikusan csak 1 eszköz jeleinek átvivésére használjuk, de ez kibővíthető időosztásos (Time Division Multiplexing TDM) eljárással.
- Széles sáv: analóg jelek továbbítására használják. Itt az adatjelet egy vivőjelre ültetik rá úgy, hogy a vivőjel 3 jellemzője közül egyet (amplitúdó, frekvencia, fázis) az adatjelnek megfelelően változtatnak, modulálják.

ADATÁTVITEL JELLEMZŐI:

SNR

- $SNR = \frac{P_{jel}}{P_{zaj}}$
- $SNR_{dB} = 10 \log_{10} \left(\frac{P_{jel}}{P_{zaj}} \right)$
- Jel/zaj viszony javítására a környezetet szabályozhatjuk illetve utólag eltávolíthatjuk szűrőkkel
- Csatorna kapacitás $C = B * \log_2(1 + SNR)$
B sáv szélesség

ADATÁTVITEL JELLEMZŐI: IRÁNY

Az adatátvitel irányát tekintve lehet:

- Simplex – csak 1 irányú adatátvitel
- Half duplex – 2 irányú adatátvitel, de egyszerre csak az egyik irányba lehet adatot átvinni
- Duplex vagy full duplex – 2 irányú adatátvitel egyszerre mindkét irányba lehet adatot átvinni

ADATÁTVITEL SCADA RENDSZEREKBEN

SCADA rendszerek esetén elsőknek vezetékes megoldásokat alkalmaztak ami az elején simplex volt majd kifejlődött az igény a visszajelzésre és így lett half duplex illetve duplex. Duplex esetekben több vezetéket kell felhasználni így nagyobb költségekkel jár. Vezetékes megoldások után rádióhullámosat alkalmaztak, majd telefonhálózatokat és szatelliteseit. Mindnek megvannak a saját előnyei és hátrányai így főleg az alkalmazás határozza meg melyiket alkalmazhatjuk.

ADATÁTVITEL JELLEMZŐI: ÁLLOMÁSOK SZÁMA

Az adók és vevők számát tekintve a következő kombinációk lehetségesek:

- Pont-Pont közötti: itt 1 adó és 1 vevő
- Multi-drop: 1 adó több vevő
- Multi-pont: több adó több vevő

ADATÁTVITELI KÖZEGEK:

- Réz vezeték
- Sodrott érpár
- Koaxiális kábel
- Optikai vezeték
- Vezeték nélküli

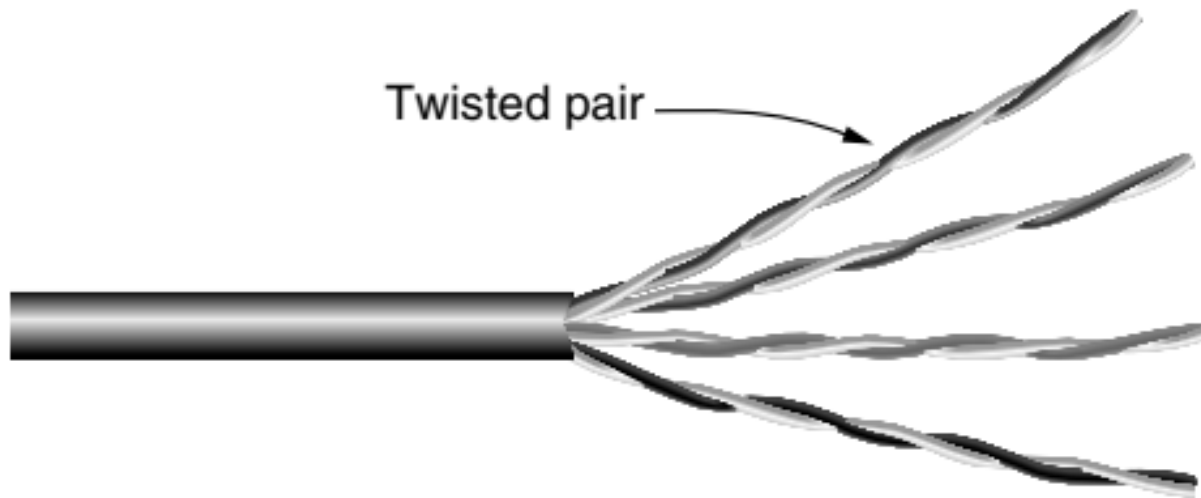
ADATÁTVITELI KÖZEGEK: RÉZ VEZETÉK

- Öreg vicc: hogy lehet 1 vezetékkel jelet átvinni?
- Legalább 2 vezeték kell:
 - Egyiken a jelet visszük át
 - Másikon a referencia feszültséget amihez viszonyítjuk a jelet.
- Csak alacsony átviteli sebesség érhető el vele

ADATÁTVITELI KÖZEGEK: SODROTT ÉRPÁR

- Kb 1mm átmérőjű szigetelt réz huzal spirálosan összetekerve mint a DNS. A sodrás azért kell, hogy az áthallástól jobban védve legyen mivel ami az egyiken indukálódik feszültség azt a másikon indukáló feszültség kioltja. Külső zaj hatása nagyjából egyformán hat a két vezetékre így összehasonlítva nem változik annyit a jel.

ADATÁTVITELI KÖZEGEK: SODROTT ÉRPÁR



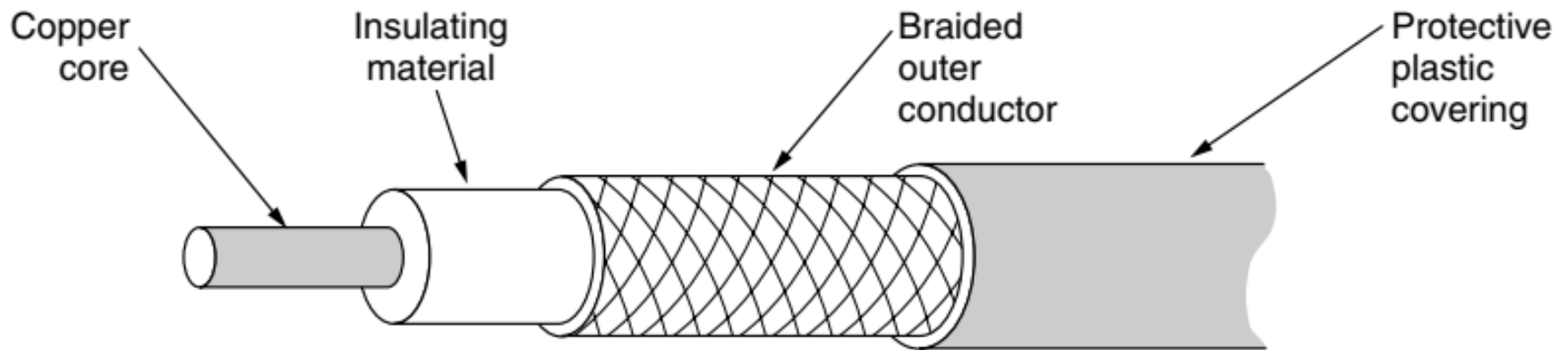
Tipikus felhasználása:

- Telefon hálózatok
- Internet (LAN) itt UTP (Unshielded Twisted Pair)-nek nevezik, mi CAT 5 és CAT 6-okat szoktunk használni (ez határozza meg a maximális átviteli sebességet). CAT 7-esek már árnyékolva vannak

ADATÁTVITELI KÖZEGEK: KOAXIÁLIS KÁBEL

- 2 fajtája elterjedt:
 - Digitális jelek továbbításában 50 Ohm-os
 - Kábel TV-nél a 75 Ohm-os

Nagyobb sáv szélességet enged meg mint



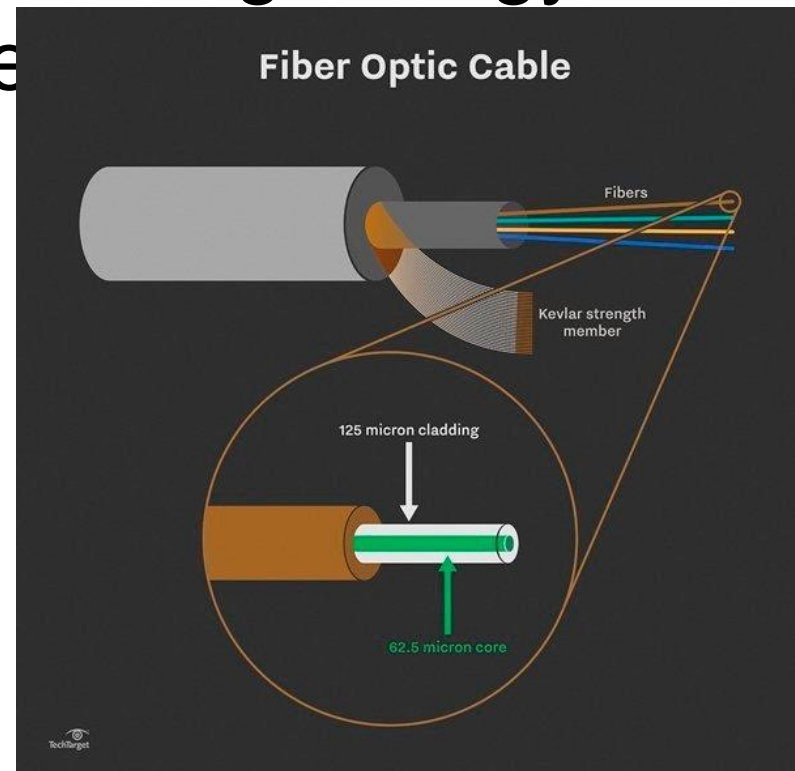
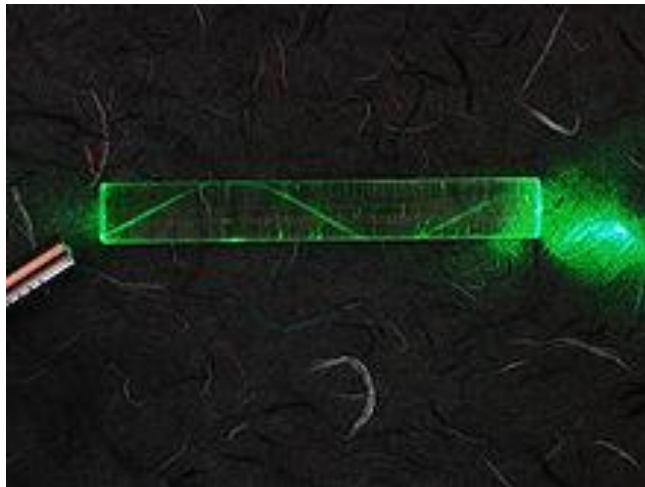
ADATÁTVITELI KÖZEGEK: KOAXIÁLIS KÁBEL

CB Coaxial Cable

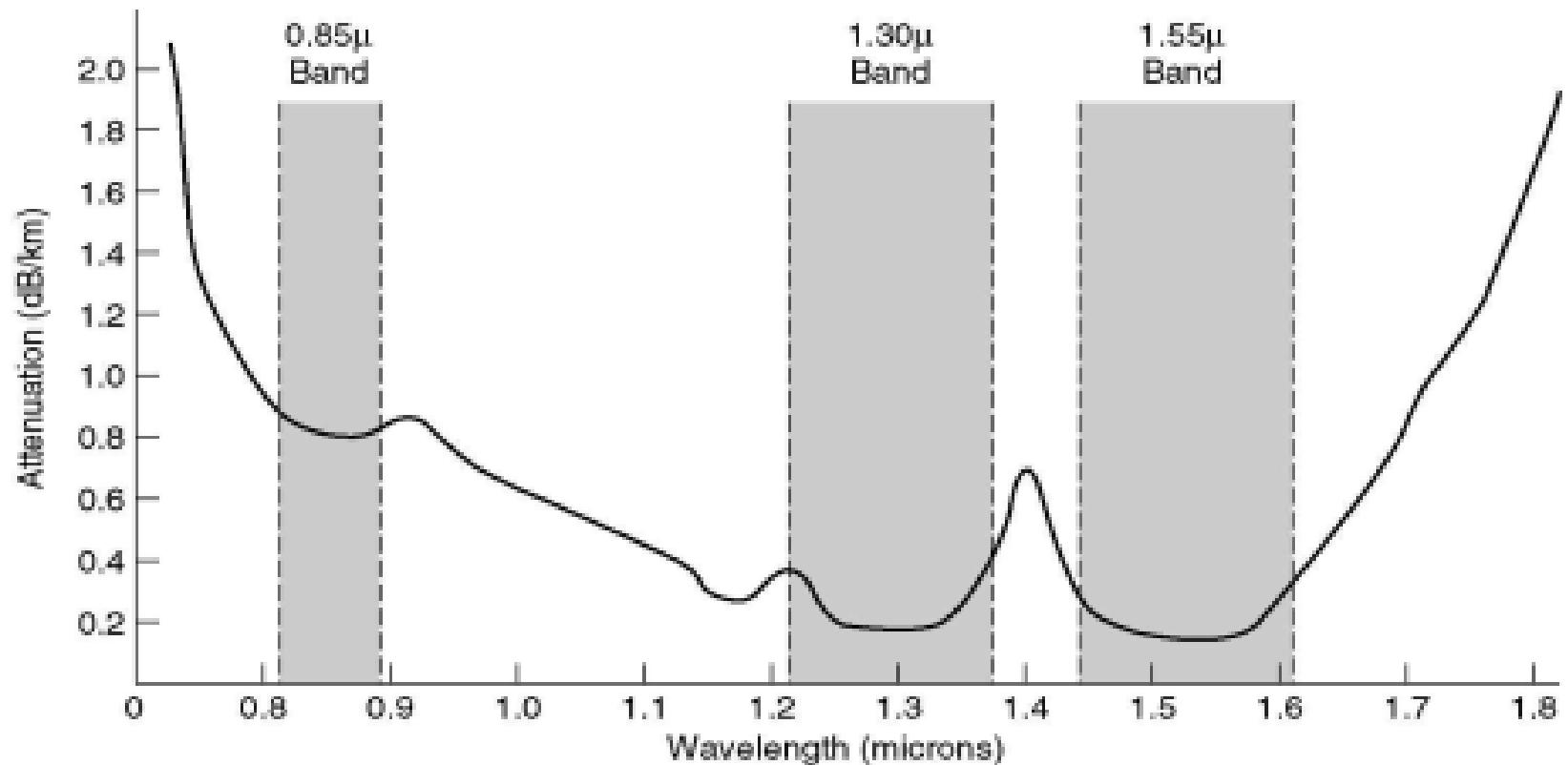


ADATÁTVITELI KÖZEGEK: OPTIKAI VEZETÉK

- Az optikai vezeték a teljes visszaverődés elvén működik, az információt szállító fény a képen látható módon megtörik így végigmenve a kábele



ADATÁTVITELI KÖZEGEK: OPTIKAI VEZETÉK



ADATÁTVITELI KÖZEGEK: VEZETÉK NÉLKÜLI

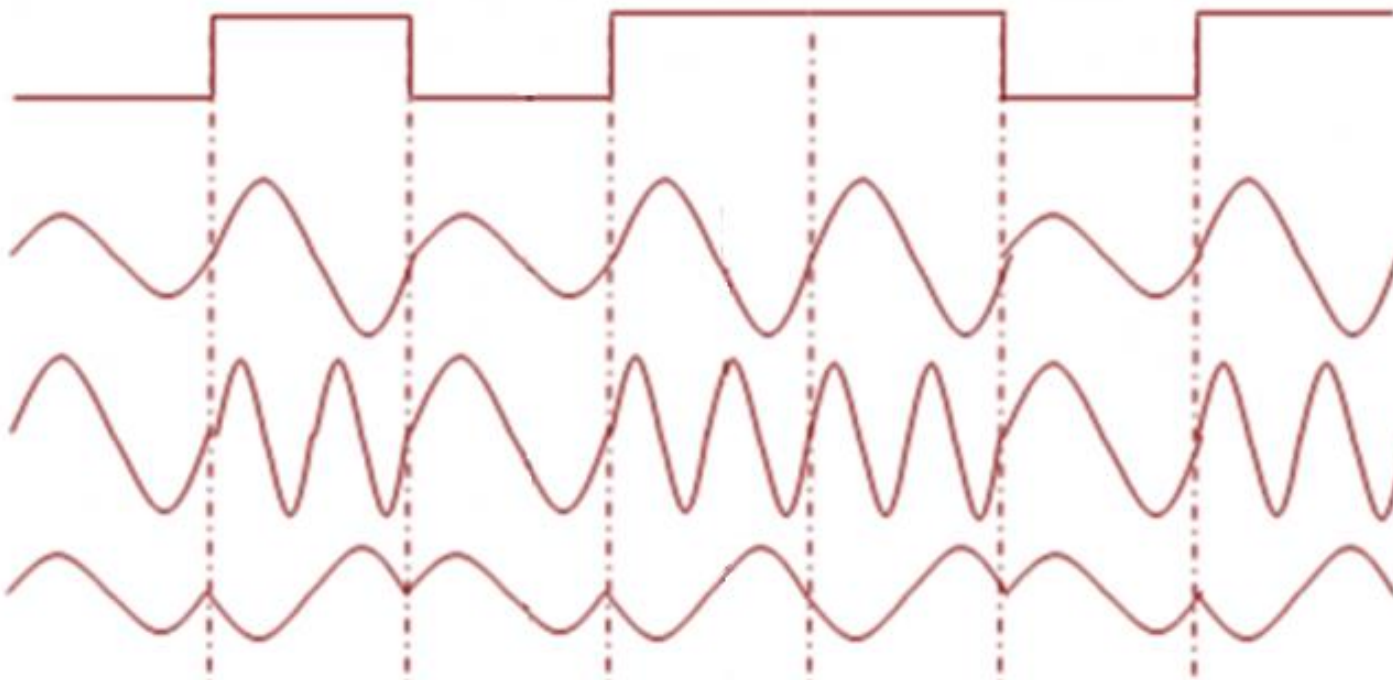
- Vezeték nélküli adatátvitelhez tartozik a laser, rádió, szatellit, wifi és mobilhálózati rendszerek
- Vezeték nélküli hálózatoknál könnyebb rácsatlakozni egy rendszerre ez telepítésnél előnyös, de biztonsági szempontból hátrány

AMPLITÚDÓ, FREKVENCIA ÉS FÁZIS MODULÁCIÓ

- AM (Amplitude Modulation) – az alacsony és magas logikai értékeket a vivő frekvencia amplitudóértéke reprezentálja.
- FM (Frequency Modulation) – az alacsony és magas logikai értékeket 2 eltérő frekvencián kódoljuk.
- PM (Phase Modulation) – az alacsony és magas logikai értékeket 2 fázishelyzettel kódoljuk.

AMPLITÚDÓ, FREKVENCIA ÉS FÁZIS MODULÁCIÓ

Digitális jel



AM

FM

PM

ADATÁTVITELI ESZKÖZÖK: MODEM

- A MODEM a MOdulátor DEModulátor szavakból rakódik össze. Feladata, hogy digitális jeleket alakítson át analóg információvá és vissza (telefonos, ADSL, optikai, rádiós stb)

ADATÁTVITELI ESZKÖZÖK: ERŐSÍTŐ

- Hálózatoknál egy bizonyos távolság után a jel degradálódik és elgyengül, hogy ilyen esetekben a jelek hatótávját kiterjesszük erősítőket kell alkalmaznunk, ezek a beadott jelet továbbítják felerősítve.

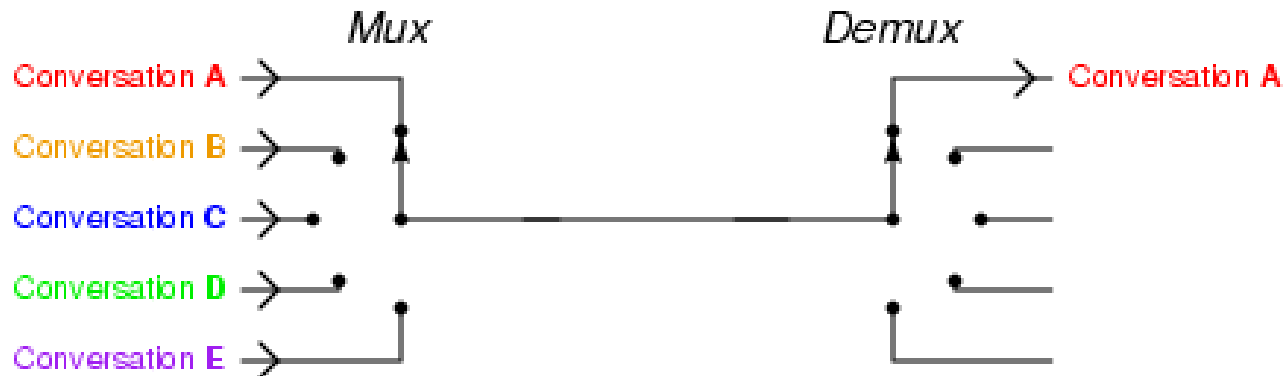
ADATÁTVITELI ESZKÖZÖK:

MULTIPLEXER

- Multiplexelésnek nevezik mikor egy fizikai tulajdonságon több jelet továbbítunk, demultiplexelésnek nevezik azt amikor az egy fizikai tulajdonságon áthozott jelet visszabontjuk eredeti komponenseire. Az eszköz ami ezt a feladatot elvégzi a multiplexer. Ezeket párban szoktuk használni demultiplexerrel.
- Mikor a rádiófrekvenciás jeleket egymás mellé pakolunk frekvencia tartományban azt frekvencia multiplexelésnek nevezik.

ADATÁTVITELI ESZKÖZÖK: MULTIPLEXER

- Mikor idő tartományban csinálják ezt időmultiplexelésnek nevezik.
- Multiplexeléssel jobban ki lehet használni a közvetítő médiumot.
- Szemléltető példa időmultiplexelésre:



ADATÁTVITELI ESZKÖZÖK: SWITCH

- Az **adatátviteli kapcsoló** vagy **switch** egy aktív számítógépes hálózati eszköz, amely a rácsatlakoztatott eszközök között adatáramlást valósít meg.
- Az OSI modellben az Adatkapcsolati rétegben dolgozik.
- Alapvető feladata:
 - csomagokban található MAC címek megállapítása
 - MAC címek és portok összerendelése (kapcsoló-tábla felépítése)
 - a kapcsoló-tábla alapján a címzésnek megfelelő port-port összekapcsolása
 - adatok ütközésének elkerülése, adatok ideiglenes tárolása

ADATÁTVITELI ESZKÖZÖK:

ROUTER

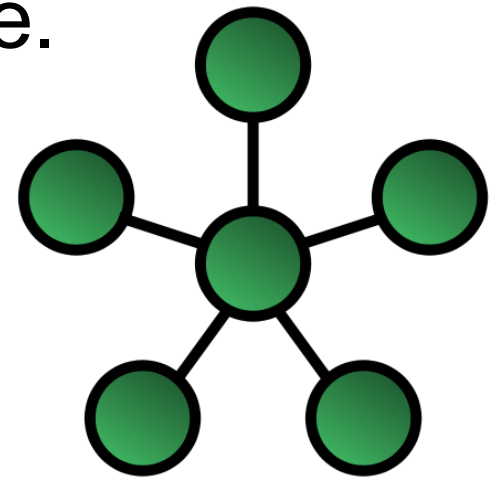
- Az **útválasztó** vagy **router** a számítógép-hálózatokban egy útválasztást végző eszköz, amelynek a feladata a különböző – például egy otthoni vagy irodai hálózat és az internet, vagy egyes országok közötti hálózatok, vagy vállalaton belüli hálózatok – összekapcsolása, az azok közötti adatforgalom irányítása.
- Az OSI modellben a Hálózati szinten dolgozik

LAN hálózatok esetében a következő topológiáról beszélünk:

- Csillag
- Gyűrű
- Lánc
- Mesh
- Fa
- Busz

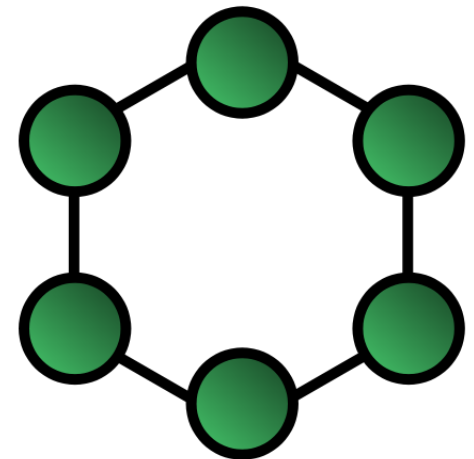
TOPOLÓGIÁK: CSILLAG

- Csillag topológia esetén a hálózat gépei egy központi gépre kapcsolódnak csak és ráhagyatkoznak, hogy adatot tudjanak továbbítani. Ha a központi gép kiesik a hálózat nem képes kommunikációra. Alacsony a kábelezési költsége.



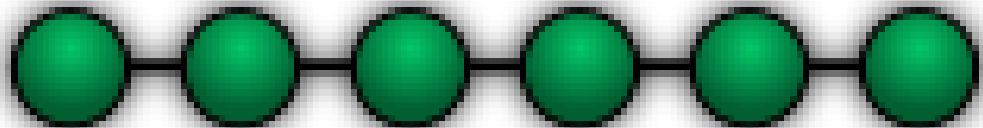
TOPOLÓGIÁK: GYŰRŰ

- Gyűrű topológia esetén a hálózat gépei mindegyikének legalább 2 portja van amin kommunikálni tud és ezeken másik gépekkel van összekötve. Előnye, hogy ha bármelyik gép kiesik létezik egy másik út a célpontig és mehet még a kommunikáció
- Lánc topológiában



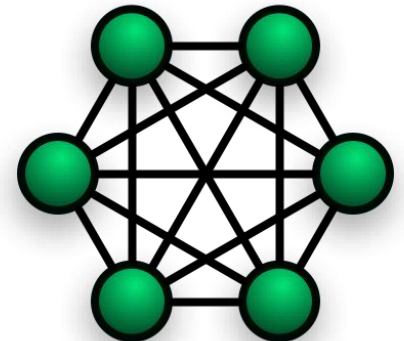
TOPOLÓGIÁK: LÁNC

- Lánc topológia esetén a hálózat gépei mindegyikének legalább 2 portja van amin kommunikálni tud és ezeken másikkal van összekötve. Előnye, hogy kicsi a kábelezési költsége. Hátránya, hogy ha bármelyik gép kiesik a hálózat 2 fele között megszakad a kommunikáció



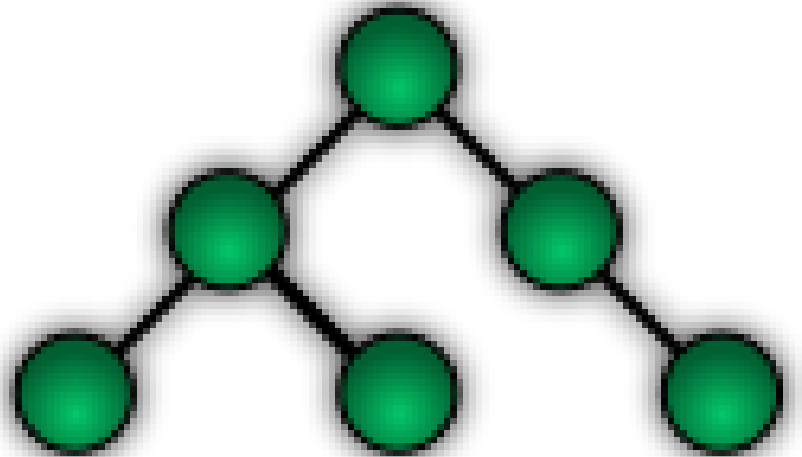
TOPOLÓGIÁK: MESH

- Mesh topológia esetén a hálózat gépei minden géppel közvetlen kapcsolatban állnak, így bármelyik vonal megszakad lesz még amin az adat eljuthat céljába, viszont kábelezési költsége ennek a legnagyobb. Vezeték nélküli megoldásokban gyakrabban alkalmazzák.



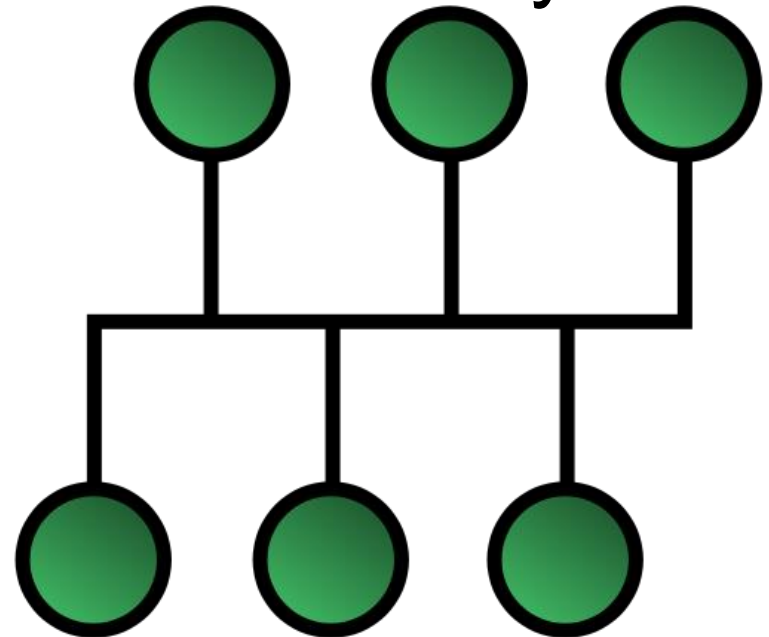
TOPOLÓGIÁK: FA

- Fa topológia esetén a hálózat gépei fa struktúrába vannak rendszerezve így különböző szinteket lehet meghatározni a hálózatban és ezekre külön szabályokat érvényesíteni.



TOPOLÓGIÁK: BUSZ

- Busz topológia esetén egy közös trónkre vannak rákapcsolva az eszközök itt arra kell vigyázni, hogy ne küldjön más üzenetet amikor már valaki küld. Könnyen bővíthető.



ADATÁTVITELI PROTOKOLLOK: OSI MODELL

Az ISO (International Standards Organization) az OSI (Open Systems Interconnection) modellt dolgozta ki, hogy szemléltesse a kommunikáció felépítéséhez szükséges feladatokat és azok egymásra épülését. Ezeket 7 rétegben fogalmazta meg amik közül bizonyos rétegeket elhagyhatunk a feladattól függően.

ADATÁTVITEL: OSI MODELL RÉTEGEK



OSI MODELL: FIZIKAI RÉTEG

A fizikai réteg definiálja a fizikai csatoló mechanikai, elektromos és funkcionális paramétereit mint az átviteli sebesség, feszültség szint, közeg stb. Ezeket szabványok írják le ilyenek az RS 232C, RS 485, USB, IEEE 802.3, IEEE 802.11, DSL stb. Amik vezetékes és vezeték nélküli közegeket is lefednek.

OSI MODELL: ADATKAPCSOLATI RÉTEG

Az adatkapcsolati réteg nem csak az átvitelt biztosítja, hanem meghatározza a telegramszerkezetet, a hozzáférési eljárást, a résztvevők címezését és az adatáramlás vezérlését is. Ez a réteg foglalkozik az adatkeretek hibamentes átvitelével, tehát itt van meghatározva, hogy az előző rétegben definiált közeget melyik eszköz használja adó illetve vevőként. Az ethernetetes hálózatoknál ezen a szinten működnek a switchek. Ez a réteg nem feltétlen szükséges a kommunikáció megvalósításához.

OSI MODELL: HÁLÓZATI RÉTEG

A hálózati réteg kezeli az útválasztást, a hálózati kapcsolatok multiplexelését és az adatfolyam kezelését, valamint felelős a két felhasználó közötti hálózati összeköttetésért, annak fenntartásáért és megszüntetéséért. Az ethernet hálózatokban ezen a szinten működnek a routerek az IP címek alapján.

OSI MODELL: SZÁLLÍTÁSI RÉTEG

A szállítási réteg biztosítja a hibafelismerést és javítást, valamint az üzenetismétlés eljárásait, az adatok tömörítését, valamint felelős a két felhasználó között az előre egyeztetett minőségű adatátvitelért. Az ethernetes hálózatokban itt működik a TCP.

OSI MODELL: EGYÜTTMŰKÖDÉSI RÉTEG

Az együttműködési réteg definiálja a kapcsolat felépítésének és lebontásának eszközeit. Itt történik a párbeszédvezérlés és az adatcsere-irányítás. E réteg elsődleges feladata, hogy váltott adás és vételkapcsolatnak megfelelően vezérelje a felhasználók adásvételét.

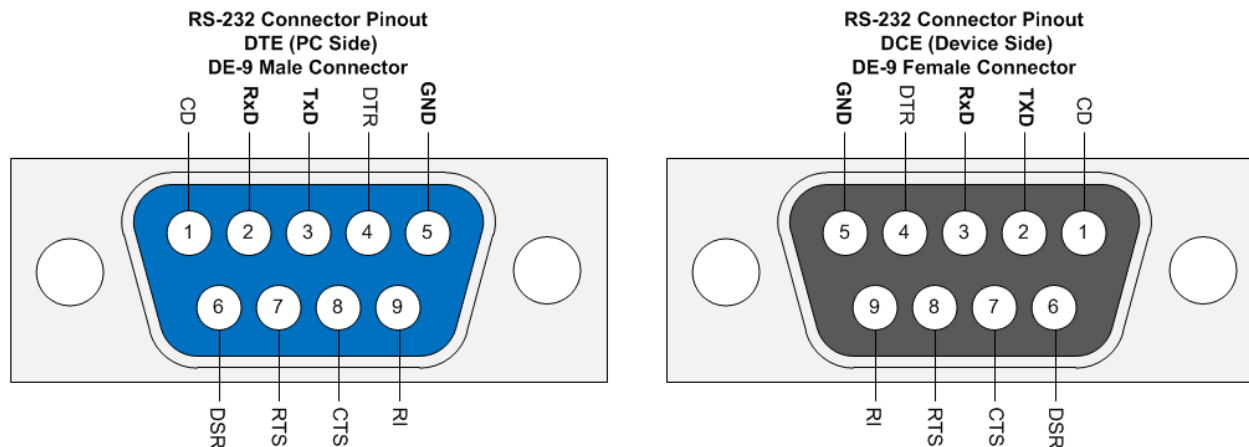
OSI MODELL: MEGJELENÍTÉSI RÉTEG

A megjelenítési réteg definiálja a konvertálási és formátumillesztési szabályokat, amelyek lehetővé teszik az adatok helyes értelmezését. Ez jelenthet karakterkód-fordítást, adat átalakítást, adatszugorítást vagy nyújtást.

OSI MODELL: ALKALMAZÁS RÉTEG

Az alkalmazás réteg csatolófelület a felhasználó és a hálózat között. Ebben definiálják a hálózati szolgáltatásokat, pl: file-átvitel.

Az RS232-nél általában Sub-D (sub-D 9, sub-D 25) csatlakozókat alkalmazunk. Minimális kommunikációhoz elég 3 vezeték: RX, TX, GND. Mivel csak ez a 3 vezeték szükséges, így más csatlakozók is alkalmazhatóak így nem ritka az RJ45 se.

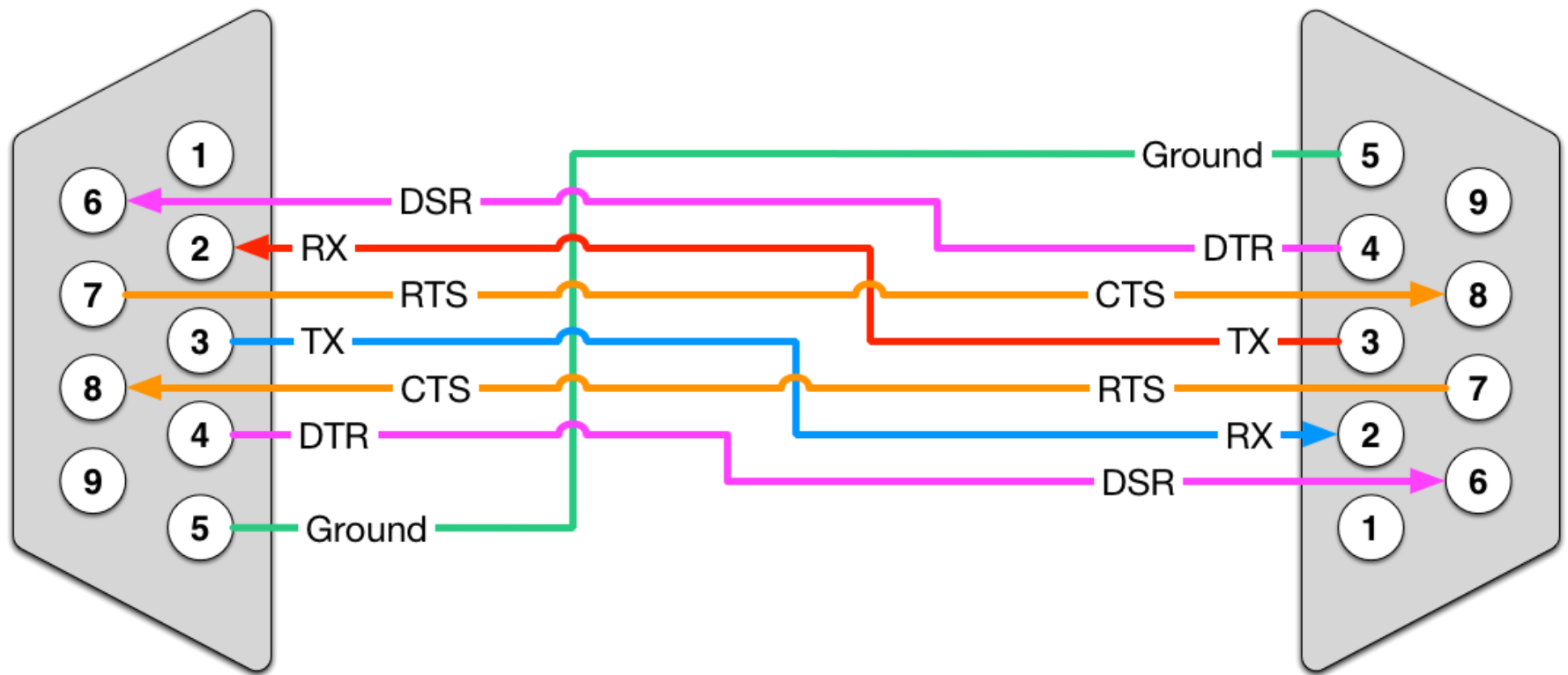


RS232

Sub-D 9	Sub-D 25	Jel neve	Bemenet/Kimenet	Leírás
1	8	DCD	Bemenet	Data Carrier Detect
2	3	RXD	Bemenet	Receive Data
3	2	TXD	Kimenet	Transmit Data
4	20	DTR	Kimenet	Data Terminal Ready
5	7	GND	Kimenet	Ground
6	6	DSR	Bemenet	Data Set Ready
7	4	RTS	Kimenet	Request To Send
8	5	CTS	Bemenet	Clear To Send
9	22	RI	Bemenet	Ring Indicator

A kommunikáció csak 2 pont között létezik. Ahhoz, hogy tudjunk kommunikálni a küldő TX-ét a fogadó RX-ére kell kötnünk. A GND biztosítja, hogy az átküldött feszültségek ugyanahhoz a szinthez legyenek mérve. Az alkalmazott feszültségek RS232-nél 3-15V a magas jelhez és $-(3-15)V$ az alacsony jelhez. Maximálisan 15m hosszú kábelt alkalmazhatunk.

RS232



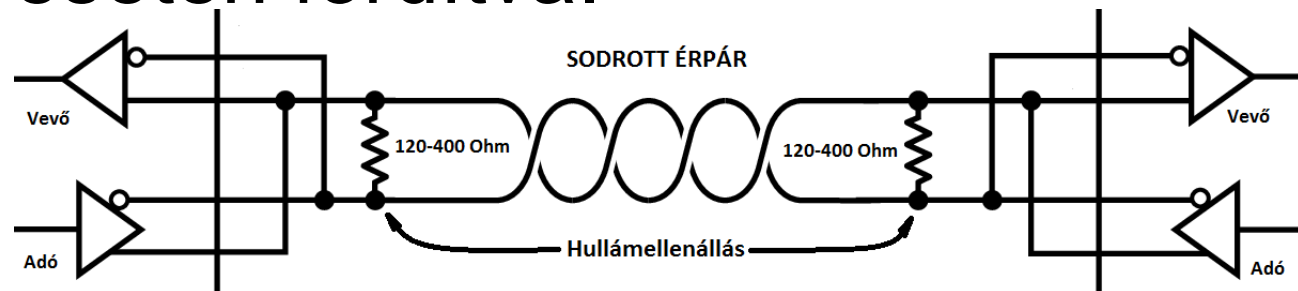
KEY CHARACTERISTICS OF THE RS-232 AND RS-485 SERIAL INTERFACES

Parameter	RS-232	RS-485
Line configuration	Single-ended	Differential
Mode of operation	Simplex or full duplex	Simplex or half duplex
Maximum cable length	50 feet	4000 feet
Maximum data rate *	20 kbits/s	10 Mbits/s
Typical logic levels	± 5 to ± 15 V	± 1.5 to ± 6 V
Minimum receiver input impedance	3 to 7 k Ω	12 k Ω
Receiver sensitivity	± 3 V	± 200 mV

* Maximum rate at maximum cable length

RS485

Az RS485-nek 2 vezeték szükséges (A, B), ezeket sodorni szoktuk, hogy jobban ellenálljanak a külső zavaró jeleknek. Mivel szimmetrikus így a két vezetéken ellentétes előjelű feszültség van, ezt jelzi az ábrán az invertálást jelző karika. A feszültség értéke $\pm 2-5V$, magasnak jel esetén A vezeték feszültsége magasabb mint B feszültsége, alacsony jel esetén fordítva.



Nagy frekvenciás alkalmazásoknál a kábeleket hullámellenállással kell lezárni, hogy megakadályozzuk a vonalon terjedő jel visszaverődését. Ezek a kábel hosszától függően 120-400 Ohm értékek között mozognak.

HART

- Highway Addressable Remote Transducer
- Az adatátvitel HART képes érzékelők és vezérlők között zajlik. Egy 4-20 mA-os kommunikáció ahol a 4-20 mA-os jelre ráteszünk egy 1 mA amplitúdójú szinuszos jelet. Ennél a jelnél 1200 Hz jelenti a logikai „1”-et és 2200 Hz a logikai „0”-t. A maximális sebesség így 1200 bps és csak half-duplex, ebből következik a mester/szolga lellégű kommunikáció

HART

- Egy hálózaton maximum 2 mester lehet aki kérdez és 15 szolgával lehet kommunikálni egy hurkon. Ekkor az eszközök sorba vannak kötve és nem tudják a 4-20 mA-os jelüket továbbítani. Csak a HART protokollt.

Protokoll	Start byte	Cím	Byte count	Status	DATA	Checksum
3-20 byte (FF)	M-S:02 v 82 S-M:06 v 86 BM:01 v 81	1 byte rövid 2 byte hosszú	Status+ data bytok száma	2 byte	Max 25 byte	1 byte

IO-LINK

- Egy 3-5 vezetéken továbbított UART (NRZ) használó protokoll. Mester/szolga jellegű a kommunikáció. Sebessége lehet 4.8 -38.4-230.4 kbps
- A minimális vezetékek amik kellenek 24V, 0V, switching signal/coded switching. A másik kettő tipikusan egy 24V és 0V
- Keret felépítése:
- „0”=24V
- „1”=0V

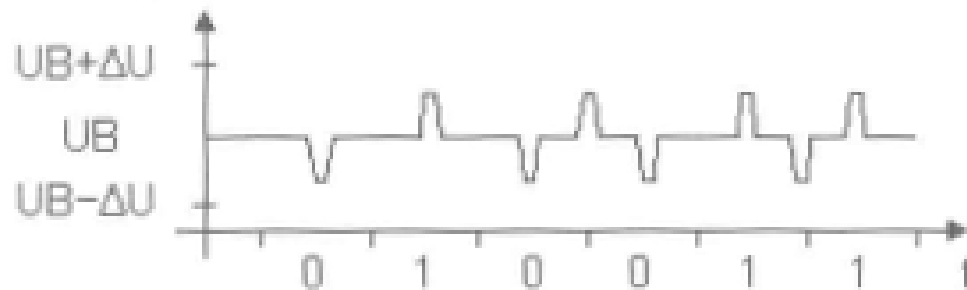
Start bit	B0-B7	Paritás bit	Stop bit
0		páros	1

- Az Actuator Sensor Interface egy olyan protokoll ami csak 2 vezetéket igényel ezek egy különleges AS-I kábelben vannak. Ezen kapja a tápot és ezen kommunikál is. Maximális hossza 100m ismétlők nélkül és maximum 2 ismétlőt lehet használni (ezekkel együtt 300m). Ezen a távolságon maximum 31 slave eszközt lehet hozzákötni és ezeket mind le tudjuk kérdezni 5 ms alatt. A kábelre vámpírcsatlakozókkal kapcsolódunk.

- Az első verziós AS-I kommunikációban a slave 4 bites válaszokat küld vissza a masternek. Minden slavenek maximum 4 bemeneti és 4 kimeneti adata lehet így $31 \cdot 4$ azaz 124 bemenet és kimenetet kezel egy AS-I master mindezt 5ms alatt. Az újabb verziókban megnövelték a slavek számát 62-re 8-ra az I/O-kat és a

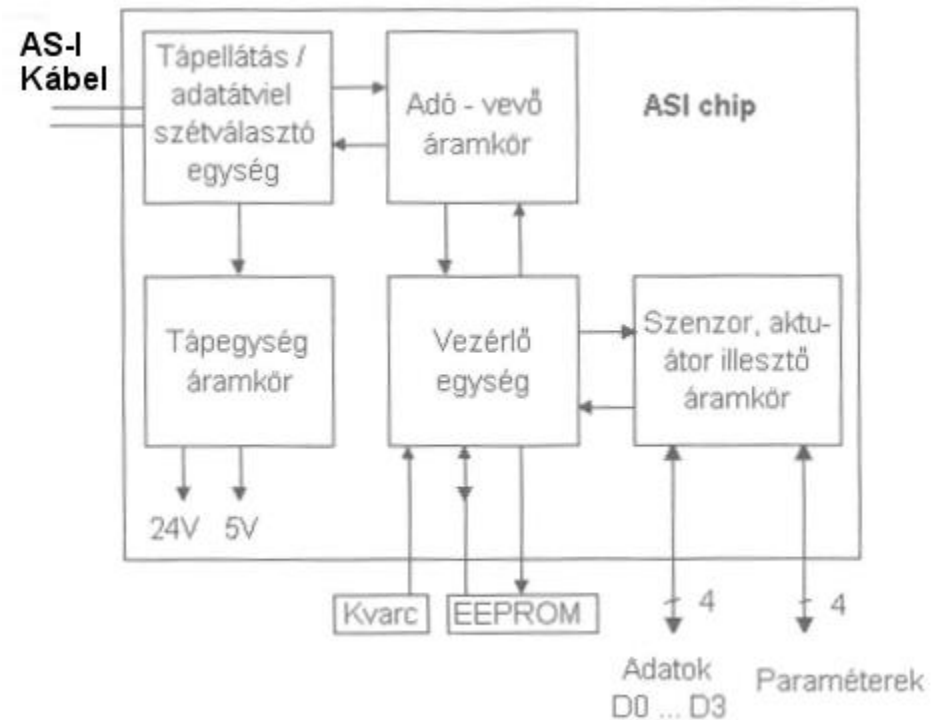
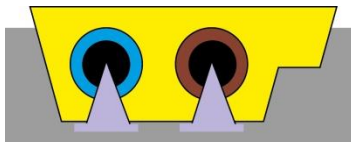
Start bit	SB	A 4	A 3	A 2	A 1	A 0	I 4	I 3	I 2	I 1	I 0	Parity bit	End Bit	Wait	Start bit	I 3	I 2	I 1	I 0	Parity bit	End Bit	
	Írás/ olvasás	Címbitek					Információ bitek							18- 30 us		Informáci ó bitek						
Master üzenet															Slave válasz							
Max 150 us																						

- A kommunikáció létrejöttéhez szükség van egy AS-I tápegységre. Az információ kódolása Manchester még a vezetéken Alternative Pulse Modulation megy.
 $\Delta U = 1V$



AS-I

- A képen egy AS-I kábel, egy slave eszköz és a slave belső felépítése látható



MODBUS

- MODBUS az egyik legrégebben alkalmazott ipari kommunikációs technológia. RS232, RS422 és RS485-ön kommunikál, de léteznek modern verziói is. Az adatcsere master/slave jellegű maximum 247 slave lehet egy masteren. A byte sorrend big endian.
- Az átviteli mód lehet ASCII vagy RTU
- ASCII esetén az üzenetek ASCII formátumban vannak
- RTU esetén az üzenetek binárisan vannak kódolva így nagyobb adatátvitel lehet a kisebb üzenetméretek miatt

MODBUS

- Keret felépítés:

ASCII:

Start	Cím	Utasítás	Adat	LRC	End
:	2 char	2 char	N char	2 char	CR LF
3A	00-F7 (1-247)				0D 0A

RTU:

Start	Cím	Utasítás	Adat	CRC	End
3.5 char idő	1 byte 00-F7 (1-247)	1 byte	N byte	2 byte	3.5 char idő
		Max 253 byte			

MODBUS

Utasítás kód	Megnevezés
01	Read Coil Status
02	Read Input Status
03	Read Holding Register
04	Read Input Register
05	Write Single Coil
06	Write Single Register
15	Write Multiple Coils
16	Write Multiple Registers
20	Read File Record
21	Write File Record
22	Mask Write Register
23	Read/Write Multiple Registers
24	Read FIFO Queue
43	Encapsulated Interface Transport

MODBUS

- ASCII és RTU csomagok összehasonlítása:

Mező neve	RTU (hex)	ASCII (karakterek)
Header	None	: (kettőspont)
Slave cím	04	0 4
Utasítás	01	0 1
Kezdő cím Hi byte	00	0 0
Kezdő cím Lo byte	0A	0 A
Kimenetek száma Hi byte	00	0 0
Kimenetek száma Lo byte	0D	0 D
Error Check Lo	DD	LRC (E 4)
Error Check Hi	98	
Trailer		CR LF
Bytok száma	8	17

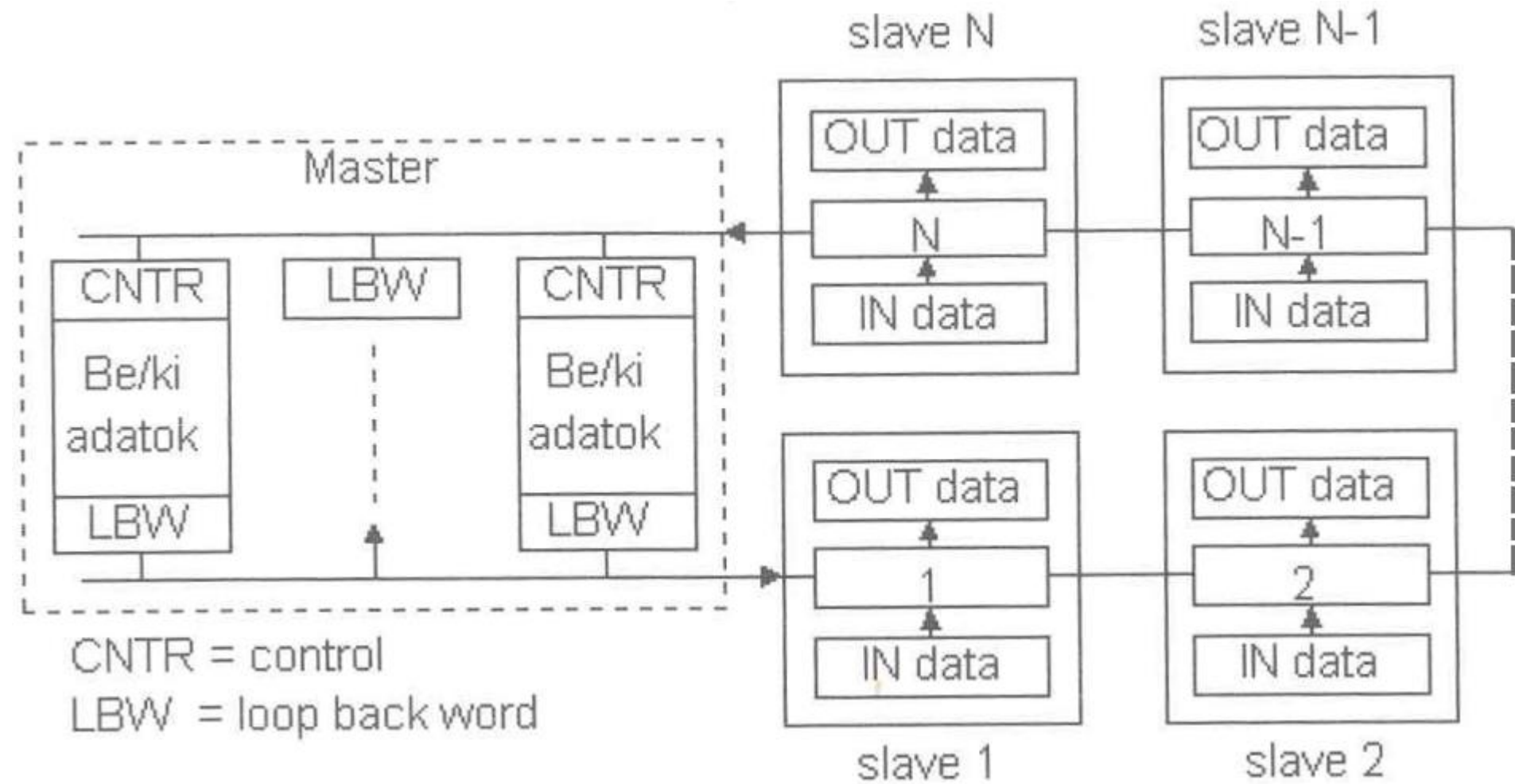
MODBUS TCP

- A MODBUS protokollnak az Ethernetes változata. Az előzőhöz hasonló információt küldünk ki egy TCP csomagban. Ezt azt jelenti, hogy választ is várunk arra a csomagra ami ideális mivel minden MODBUS utasításra választ várunk. Ez a TCP csomag pedig egy IP csomagba van becsomagolva így küldjük ki a hálózatra, hogy eljusson a célponthoz. Néhány eltérés van a MODBUS TCP és a sima MODBUS között, nincs cím mező mert az IP cím helyettesíti és nincs CRC mert a csomag ellenőrzést máshol oldják meg. UID mezővel rendelkezik amit arra használunk, ha egy IP címen több MODBUS eszköz van.

INTERBUS

Az Interbus rendszert a Phoenix Contact cég fejlesztette ki 1993-ban. Egy gyűrű topológiájú hálózat ahol alhálózaton belül algyűrűket is létre lehet hozni terminálmodulokkal. A modulok között az átviteli közeget szabadon lehet választani. Felépítése master/slave, legfeljebb 256 résztvevő lehet egy buszon. Átviteli sebesség 500 kbps és 2 Mbps között van. Az átviteli közeg lehet sodrott érpár és optikai szál.

INTERBUS



A keret felépítése:

LBW	Slave 1 adat	Slave 2 adat	Slave 3 adat	...	Slave n adat	FCS	CNTR
-----	-----------------	-----------------	-----------------	-----	-----------------	-----	------

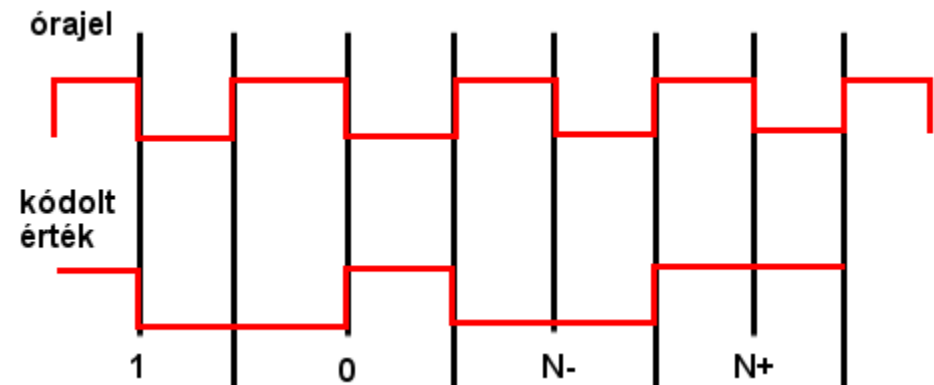
A master folyamatosan lépteti az adatokat, ezzel párhuzamosan a slave-ek beírják adataikat a keretbe, majd a master addig tolja a keretet amíg az LBW-ből nem következtet arra, hogy a teljes telegrammot áttolta.

FOUNDATION FIELDBUS

- Eredetileg a 4-20 mA-os kommunikáció kiváltására terveztek egy digitális protokollt 1984-ben. 1996-ban megalkották a Foundation Fieldbus H1-et és 2000-ben a HSE-t (High Speed Ethernet).
- A kommunikáció H1 esetén kéteres vezetéken még HSE esetében Ethernet kábelon megy. Az elérhető sebesség H1-nél 31.25 kbps még HSE esetén 1Gbps
- A maximális távolság H1 esetén a kábeltől függ STP:1900m még sima 2 vezeték esetén 200m

FOUNDATION FIELDBUS

- Az információ kétfázisú Manchester kódolással van megoldva amit áram modulációval rakunk rá a közvetítő vezetékre.



- A keret felépítése:

Bevezető Byte	Start Byte	Adat (PDU)	Keret ellenőrzés	Záró Byte
10101010	1N+N-10N-N+0	1-256 byte	2 byte	1N+N-N+N-101

FOUNDATION FIELDBUS

- A PDU (Protocol Data Unit) felépítése:

PCI	FAS PCI	FMS PCI	Felhasználó Adat
5-15 byte	1 byte	4 byte	0-251 byte
		FMS PDU	
	FAS PDU		

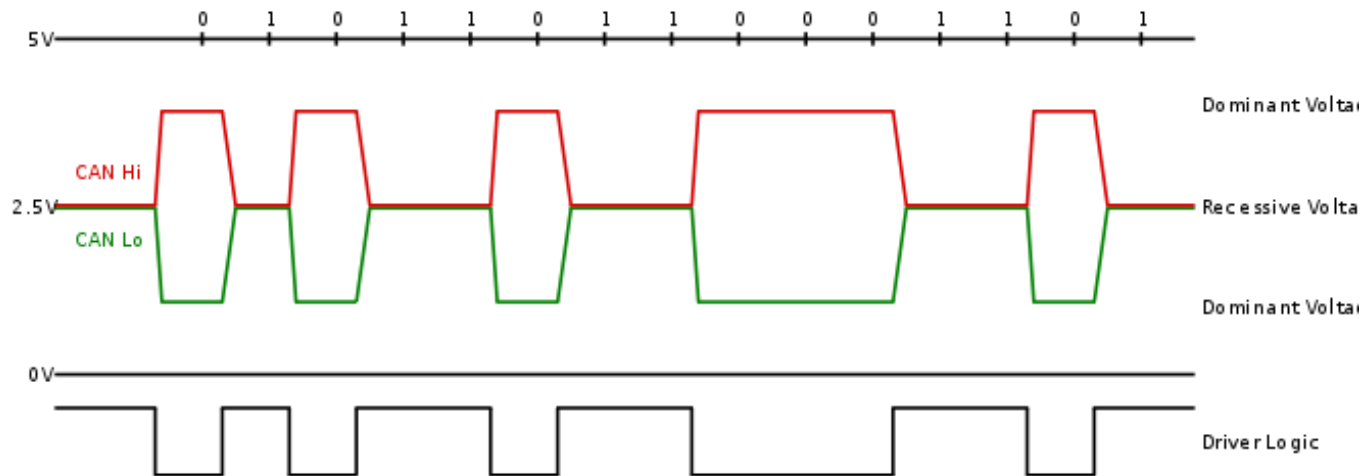
- PCI – Protocol Control Information
- FAS – Fieldbus Access Sublayer
- FMS – Fieldbus Messaging Service

CAN BUS

- A beágyazott rendszerekben főleg az autóiparban alkalmazott protokoll a CAN (Controller Area Network). Az átviteli közeg sodrott érpár amire NRZ-vel kódolt információt viszünk rá. Az átviteli sebesség 5kbps-1Mbps-ig terjed a maximális távolság 40 m.

CAN BUS

- A két vezetéket CAN Hi és CAN Lo-nak nevezik az információt pedig a kettőn lévő ellenfázisú (szimmetrikus) jel hordozza.



CAN BUS

- A keret felépítése:

SOF	Arbitration	Control	Data	CRC	ACK	EOF	Intermission
1bit	12-32bit	6bit	0-8byte	2byte	2bit	8bit	3bit

SOF-Start Of Frame

Arbitration-Üzenetküldési jogosultságot eldöntő mező

Control-üzenetben található adatmennyiséggel kapcsolatos információ

ACK-üzenet nyugtázására a mező

EOF-End Of Frame

Intermission-Szünetidő

PROFIBUS

- PROFIBUS (PROcess Field BUS) a Siemens eszközök protokollja. Több verziója létezik:
 - FMS (Fieldbus Message Specification)
 - PA (Process Automation)
 - DP (Distributed Processing)
- Átvitelhez RS485-öt használnak, de lehet optikai szál is.
- Master/Slave jellegű kommunikáció, egy hálózaton több master lehetséges, de max 126 állomás.
- Sebessége 9.6 kbps-12Mbps ig terjed
- Távolság 1200m -100m (függ a sebességtől)

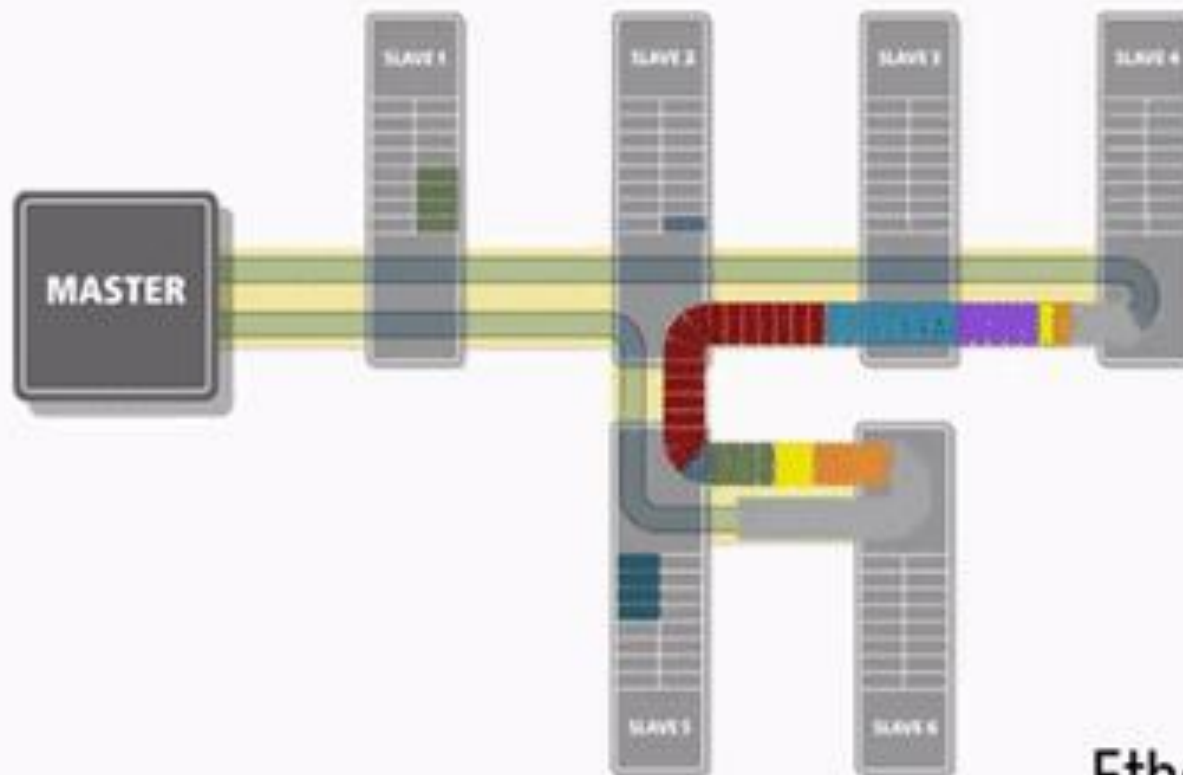
- A gyakorlatokon is használt Siemens PLC-
ket PROFINET (PROcess Field
NETwork)-el kötöttük össze. Ez a protokoll
az ipari Ethernetre (IE) épül maximális
sebessége 1Gbps. A Kábel amit
használunk Ipari Ethernet Kábel ami jobb
árnyékolással rendelkezik mint egy sima
ethernet kábel.

- Az Allen Bradley eszközök által használt protokollok:
 - DeviceNet (CAN továbbfejlesztés)
 - CompoNet
 - ControlNet
 - Ethernet/IP (Ethernet over Industrial Protocol nem IP mint IP cím)
- Mindegyike a CIP (Common Industrial Protocol)-ra épül, de manapság csak az Ethernet/IP ami fennmaradt.

- Ethernet for Control Automation Technology a Beckhoff protokollja. A protokollt úgy képzeljük el mint ha egy vonat vinné az üzenetet. A mastertől indulva minden slave egy állomás ahol a számára megfelelő vagonból olvas vagy vagonba ír. Egy üzenettel több állomást is megjár.

ETHERCAT

EtherCAT Functional Principle



EtherCAT 

A DNP3 (Distributed Network Protocol) egy nyílt kommunikációs protokoll master/slave jellegű. Hálózati felépítésben lehet pont-pont, multidrop, multipont és hierarhikus. A közvetítő médium sodrott érpár (RS232, RS485, TCP/IP).

Egy DNP3 üzenet felépítése egy fejlécből és adat szekcióból áll ahol a fejléc a következő adatokat tartalmazza:

SYNC	Hossz	Link Control	Cél Cím	Forrás Cím	CRC
2 byte	Hány byte jön az adat szekció CRC-ig	1 byte	2 byte	2 byte	2 byte

Adat szekcióval együtt a maximális hossz 292 byte ahol minden 16 bytera egy 2 byte-os CRC jut azaz maximum 250 byte hasznos adatunk lehet.

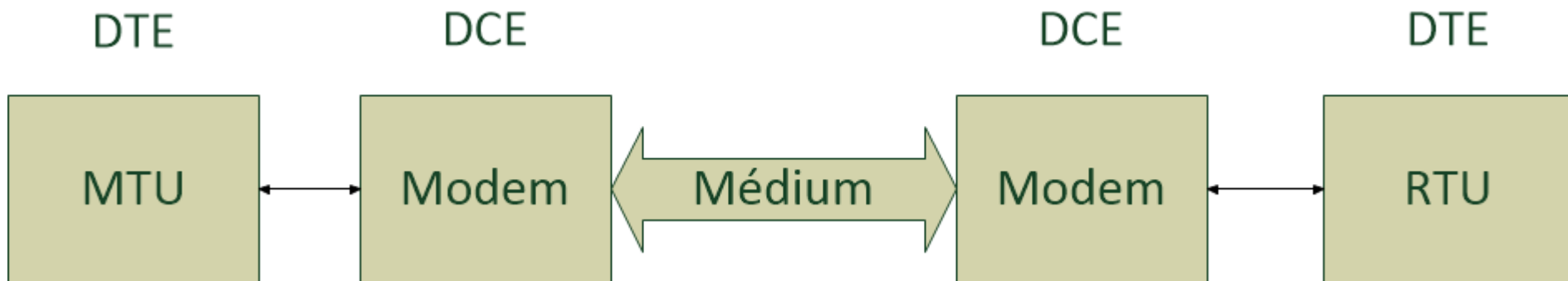
Egy szabványgyűjtemény amit az IEC (International Electrotechnical Consortium) alakított ki, hogy energiaiparban használják. Természetesen más iparágakban is alkalmazható. Használható pont-pont, multidrop, multi-pont konfigurációban. A közvetítési médium sodrott érpár (RS232, RS485, TCP/IP)

A keret felépítése:

Start frame					Adat egység azonosító				Információ objektum		...	Információ objektum n	Stop frame	
S C	L	S C	Crtl	a d d r	T y p e I D	V S Q	C O T	ASDU addr	Inf. Obj addr	Inf. elem			C h k s u m	STC

KOMMUNIKÁCIÓ: ESZKÖZÖK

Az ábrán a kommunikáció felépítéséhez szükséges elemeket láthatjuk. Manapság DTE (Data Terminal Equipment) és DCE (Data Communication Equipment) egy egységként van.



TEREPI ESZKÖZÖK

- Terepi eszközök alatt értjük az érzékelőket, motorokat, munkahengereket és más aktuátorokat amik a gyártósoron vannak. Ezek valamilyen protokollon keresztül kommunikálnak a közvetlen vezérlő egységükkel.

KÖZVETLEN VEZÉRLŐ ESZKÖZÖK

- DAQ (Data AcQuisition) eszközök vagy RTU (Remote Terminal Unit):
 - Intelligens mérőeszközök
 - PLC-k
 - Field I/O
- Feladatuk digitális és analóg jeleket gyűjteni majd továbbítani a szervernek
- Rendelkezhetnek kimenetekkel is valamint programmal
- Rendelkezniük kell kommunikációs lehetőséggel

SCADA és HMI paneleken tipikusan a következő funkciókat szoktuk lekezelni:

- Data Display
- Diagram Display
- Trendek
- Alarmok
- Receptek
- Felhasználók

KONTROLLOK

- Data Display: nagy számú adat ábrázolására szolgál. Itt nem kell, hogy látványos legyen a megjelenés minél jobban ki akarjuk használni a képernyőt.
- Process & Instrumentation Diagram (P&ID): A folyamat egészére ad rálátást és csak a legfontosabb értékeket jeleníti meg, nem szoktunk itt beavatkozni.
- Diagram Display: egy ábra ami a teljes folyamat egy részegységét szemlélteti több információval mint a P&ID. Szelepeket kapcsolhatunk értékeket állíthatunk itt.

DATA DISPLAY

SIMATIC WinCC Runtime Advanced

Finomító A1

Vezér nyomás	0	MPa	Bejövő tömegáram	0	m3/h
Tartály 1 temp	0	C	Kimenő tömegáram	0	m3/h
Tartály 2 temp	0	C	Szivattyú 1 teljesítmény	0	kW
Tartály 3 temp	0	C	Szivattyú 2 teljesítmény	0	kW
Tartály 4 temp	0	C	Szivattyú 3 teljesítmény	0	kW
Alág 1 nyomás	0	MPa	Szivattyú 4 teljesítmény	0	kW
Alág 2 nyomás	0	MPa	Szivattyú 5 teljesítmény	0	kW
Alág 3 nyomás	0	MPa	Szivattyú 6 teljesítmény	0	kW
Alág 4 nyomás	0	MPa	Mérleg 1	0	kg
Alág 5 nyomás	0	MPa	Mérleg 2	0	kg
Alág 6 nyomás	0	MPa	Mérleg 3	0	kg
Alág 7 nyomás	0	MPa	Mérleg 4	0	kg

P&ID

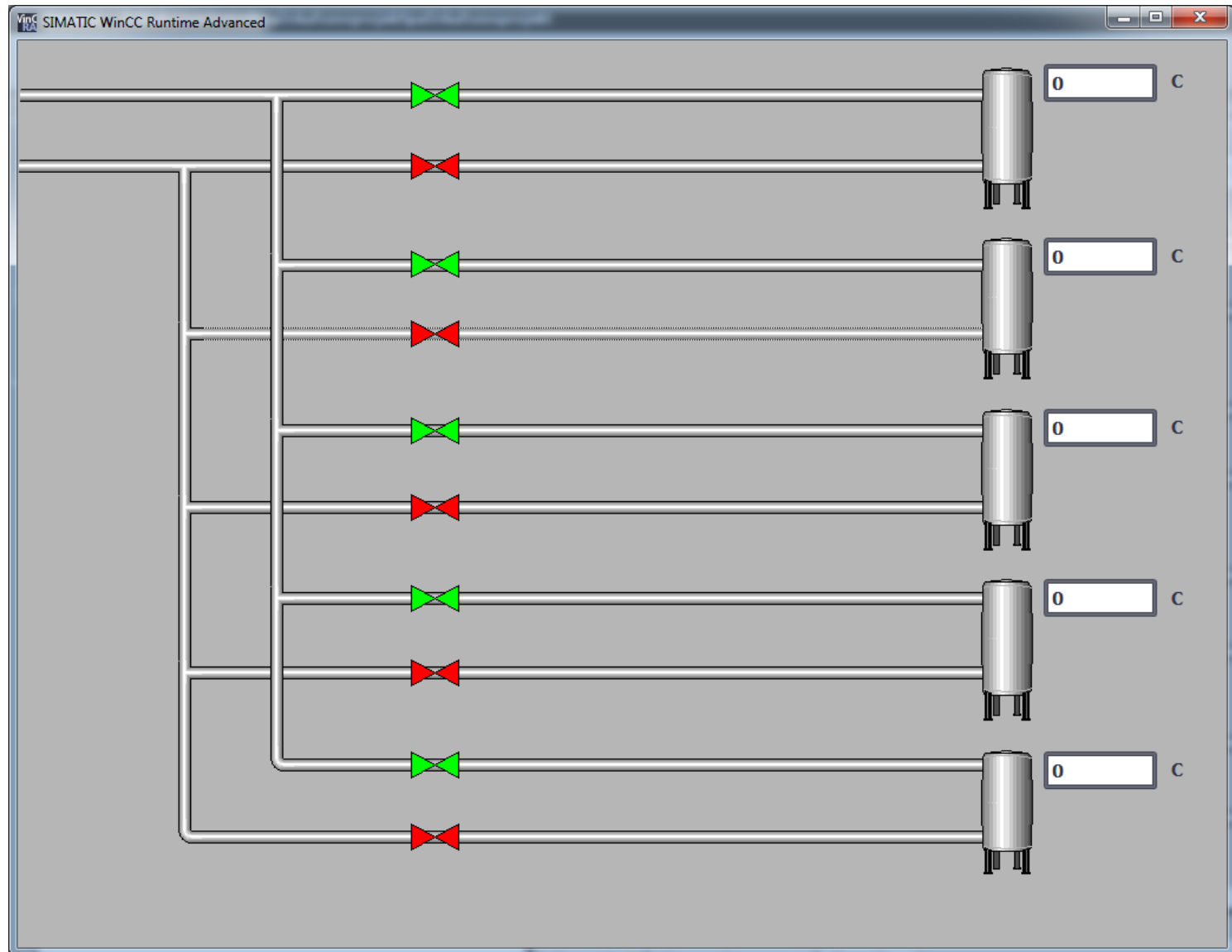
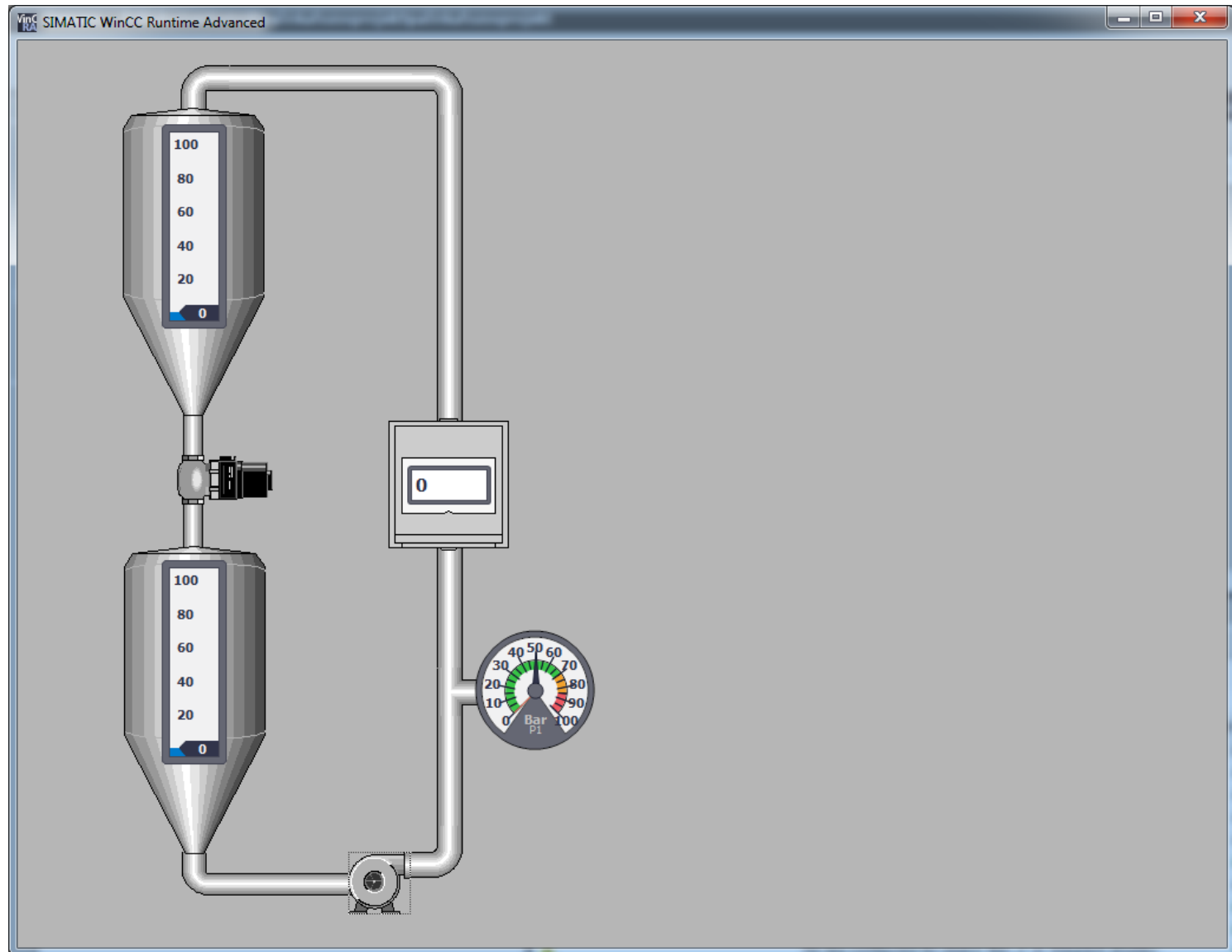


DIAGRAM DISPLAY



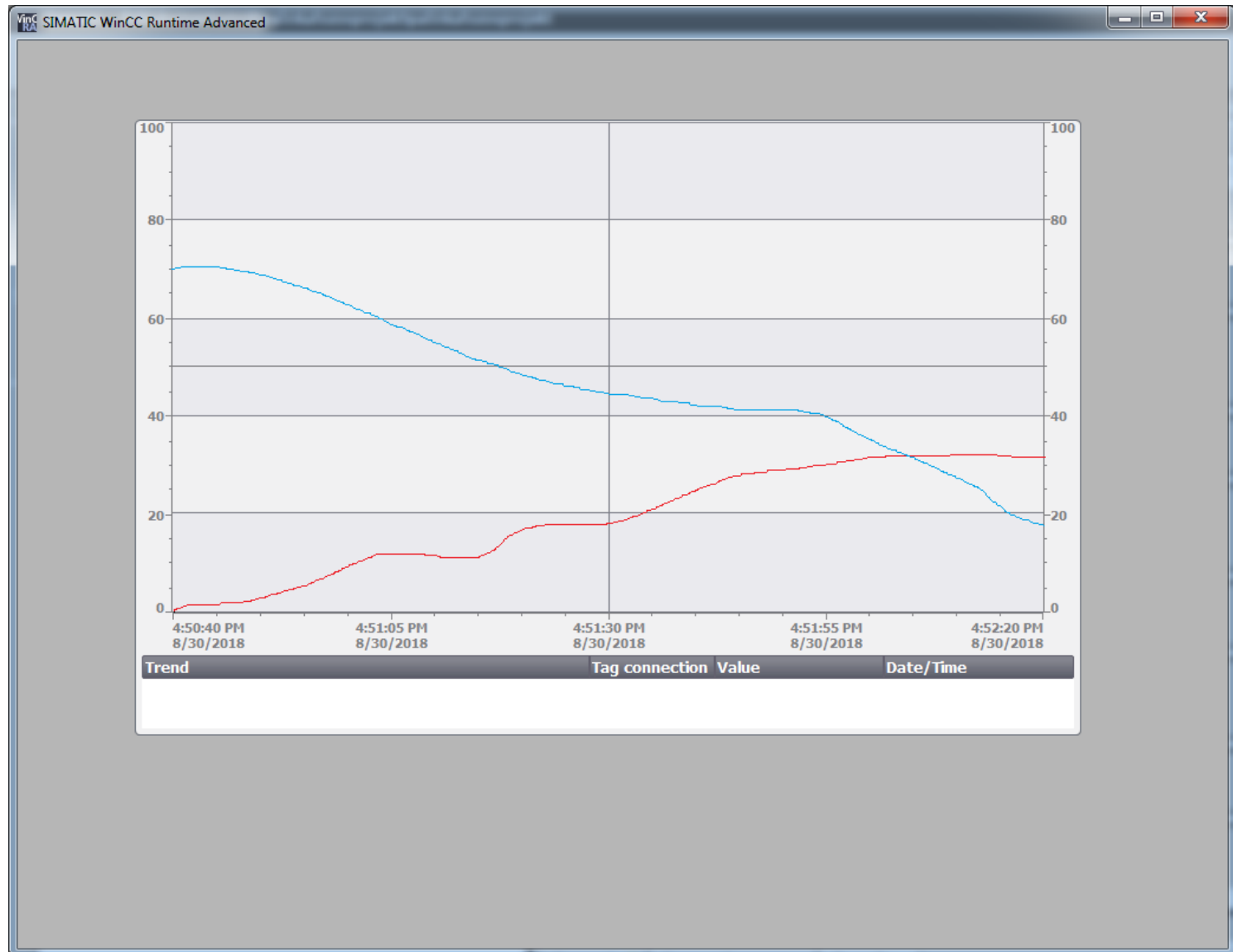
RÖVID TÁVÚ TRENDEK

- Rövid távú trendeknél az aktuális időtől számítva órákban meghatározott időre visszamenőleg láthatjuk valamilyen érték változását nagy rezolúcióban

HOSSZÚ TÁVÚ TRENDEK

- Hosszú távú trendeknél akár évekre visszamenőleg láthatjuk a tulajdonság változását. Ezt úgy tehetjük meg, hogy eltároljuk az értékeket egy Historian adatbázisba, hogy minél kevesebb helyet foglaljon ez az értékek kisebb felbontásban vannak elmentve. (Csak megfelelő mértékű változás esetén ír be újabb értéket a historian-ba)

TRENDEK



ALARMOK

- Prioritások:
 1. Szintű: Hang riasztás+Képernyőn megjelenítés+alarm bejegyzés+alarm loggolás
 2. Szintű: Képernyőn megjelenítés+alarm bejegyzés+alarm loggolás
 3. Szintű: alarm bejegyzés+alarm loggolás

ALARMOK

SIMATIC WinCC Runtime Advanced

No.	Time	Date	Status	Text	Acknowledge group
! 2	5:28:39 PM	8/30/2018	I	T2 HH	0
! 1	5:28:16 PM	8/30/2018	I	Elfogyott a bab	0
! 3	5:28:39 PM	8/30/2018	IA	Víz hőmérséklet túl magas	0

JELENTÉSEK

- Jelentések esetén egy összesített adatlapot értünk amin a legjelentősebb adatokat tüntetjük fel (pl.: éves fogyasztás, legyártott darabok száma, meghibásodások száma...) ezeket auditkor szoktuk használni

RECEPT KEZELÉS

- Recept kezelés esetén a közvetlen vezérlőn egy olyan program fut amit újraprogramozás nélkül átállíthatunk operátor által megadott értékekre (pl.: célhőmérséklet átállítása, idők átállítása, ismétlések számának megváltoztatása...). A recepteket az operátor a HMI-n vagy a SCADA felületen állíthatja be.

RECEPTKEZELÉS

SIMATIC WinCC Runtime Advanced

Recipe Name: Pálinkák No.: 1

Data Record Name: Alma No.: 1

Entry Name	Value
Alma	20
Körte	0
Birs	0
Főzési idő	300
Hőmérséklet	65

Ready

FELHASZNÁLÓ KEZELÉS

Felhasználó kezelésre azért van szükségünk mivel nem minden dolgozónak aki hozzáfér a berendezéshez van joga mindent látni vagy változtatni. Helyesen implementált felhasználó kezelés korlátozza a felhasználót, hogy csak azt lássa és csak azt módosíthassa amihez joga is van.

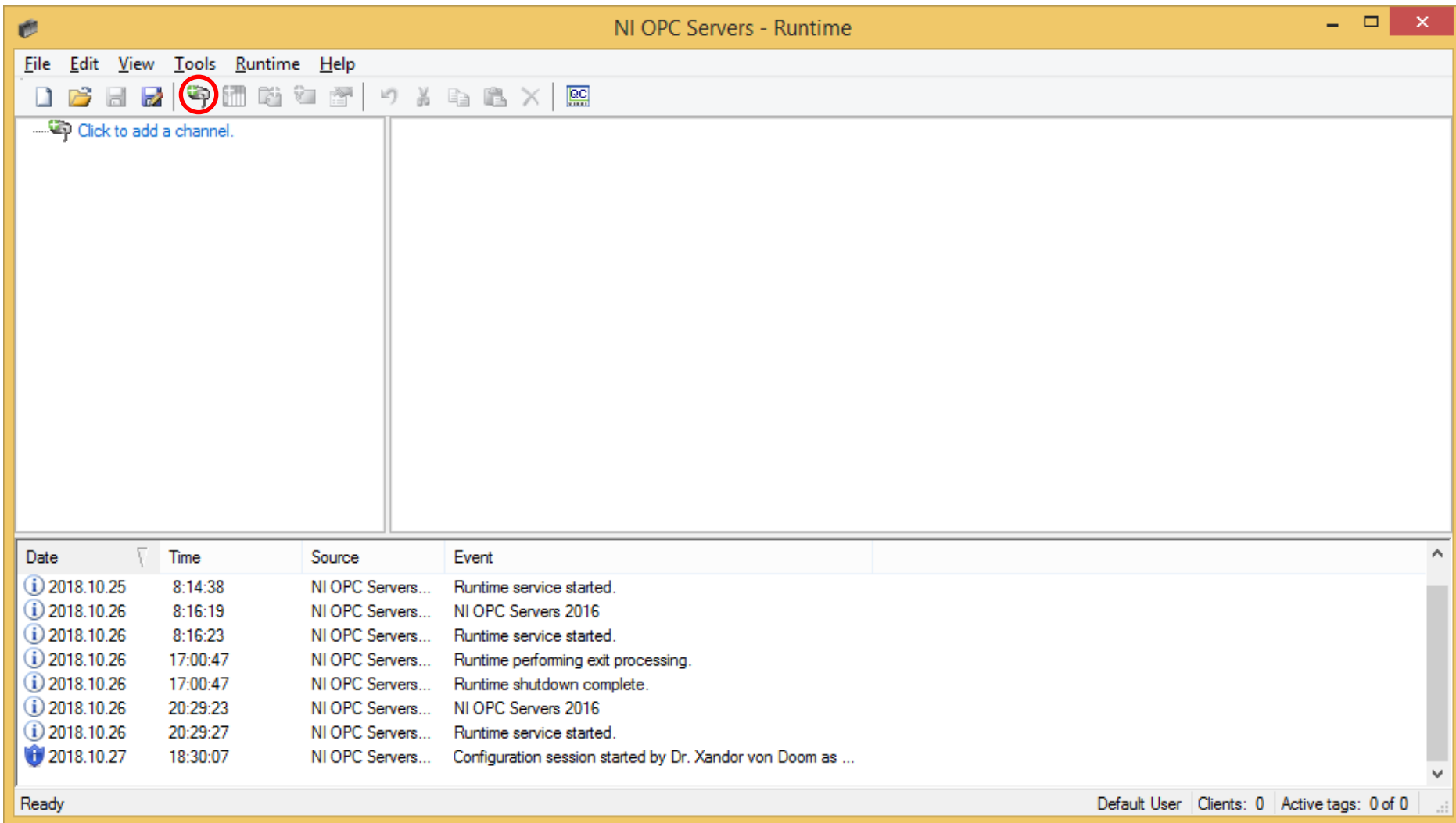
Tipikus felhasználó jogosultságok:

Operátor, Gyakorló Operátor, Adminisztrátor, Mérnök, Menedzser

NYELVI BEÁLLÍTÁSOK

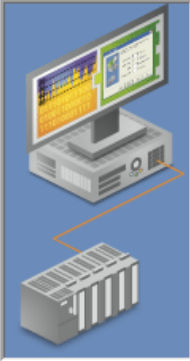
Nem kötelező, de van lehetőség megjelenítési nyelvek változtatását is beépíteni SCADA rendszerünkbe, ez főleg nemzetközi cégeknél várható ahol ugyanazt a keretrendszerrel rendelik meg ahol máshol csak a saját nyelvükön szeretnék olvasni.

SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN



SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN

New Channel - Identification



A channel name can be from 1 to 256 characters in length.
Names can not contain periods, double quotations or start with an underscore.

Channel name:

< Vissza Tovább > Mégse Súgó

New Channel - Device Driver



Select the device driver you want to assign to the channel.
The drop-down list below contains the names of all the drivers that are installed on your system.

Device driver:

☐ Enable diagnostics

< Vissza Tovább > Mégse Súgó

New Channel - Network Interface



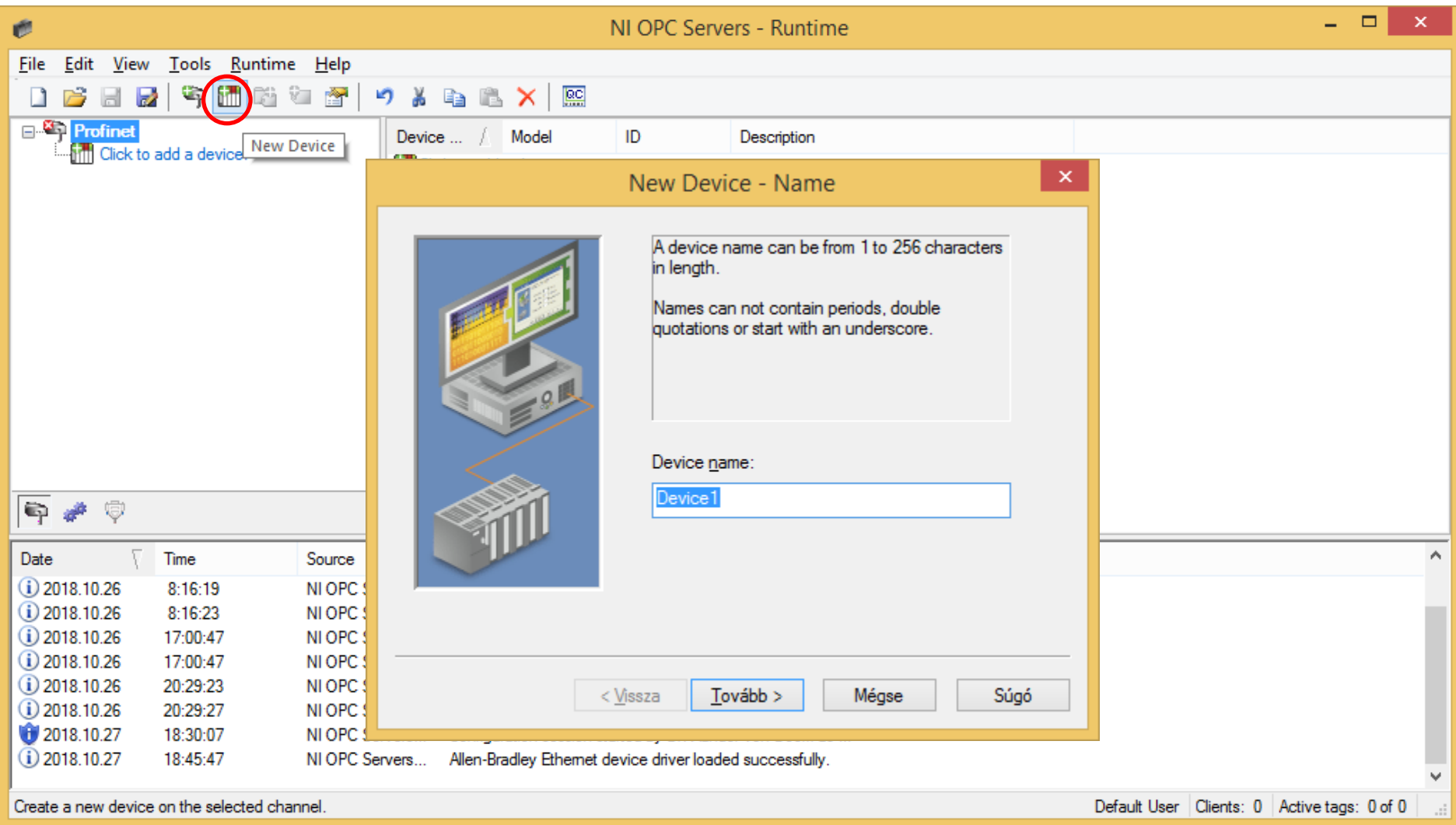
This channel is configured to communicate over a network. You can select the network adapter that the driver should use from the list below.
Select 'Default' if you want the operating system to choose the network adapter for you.

Network Adapter:

Default
Qualcomm Atheros ... [160.114.18.183]
VMware Virtual Et... [192.168.19.1]
VMware Virtual Et... [192.168.75.1]

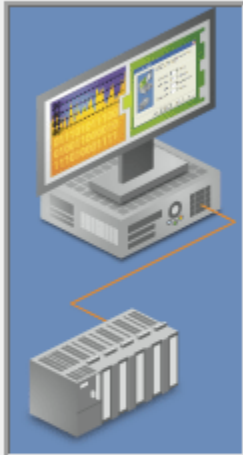
< Vissza Tovább > Mégse Súgó

SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN



SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN

New Device - Model



The device you are defining uses a device driver that supports more than one model. The list below shows all supported models.

Select a model that best describes the device you are defining.

Device model:

- S7-200
- S7-200
- S7-300
- S7-400
- S7-1200
- S7-1500
- NetLink: S7-300
- NetLink: S7-400

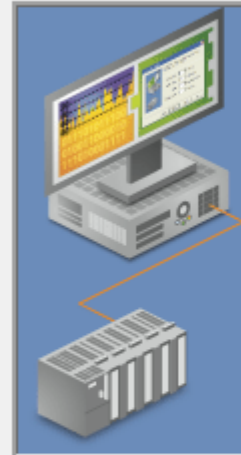
< Vissza

Tovább >

Mégse

Súgó

New Device - ID



The device you are defining may be multidropped as part of a network of devices. In order to communicate with the device, it must be assigned a unique ID.

Your documentation for the device may refer to this as a "Network ID" or "Network Address."

Device ID:

160.114.76.95

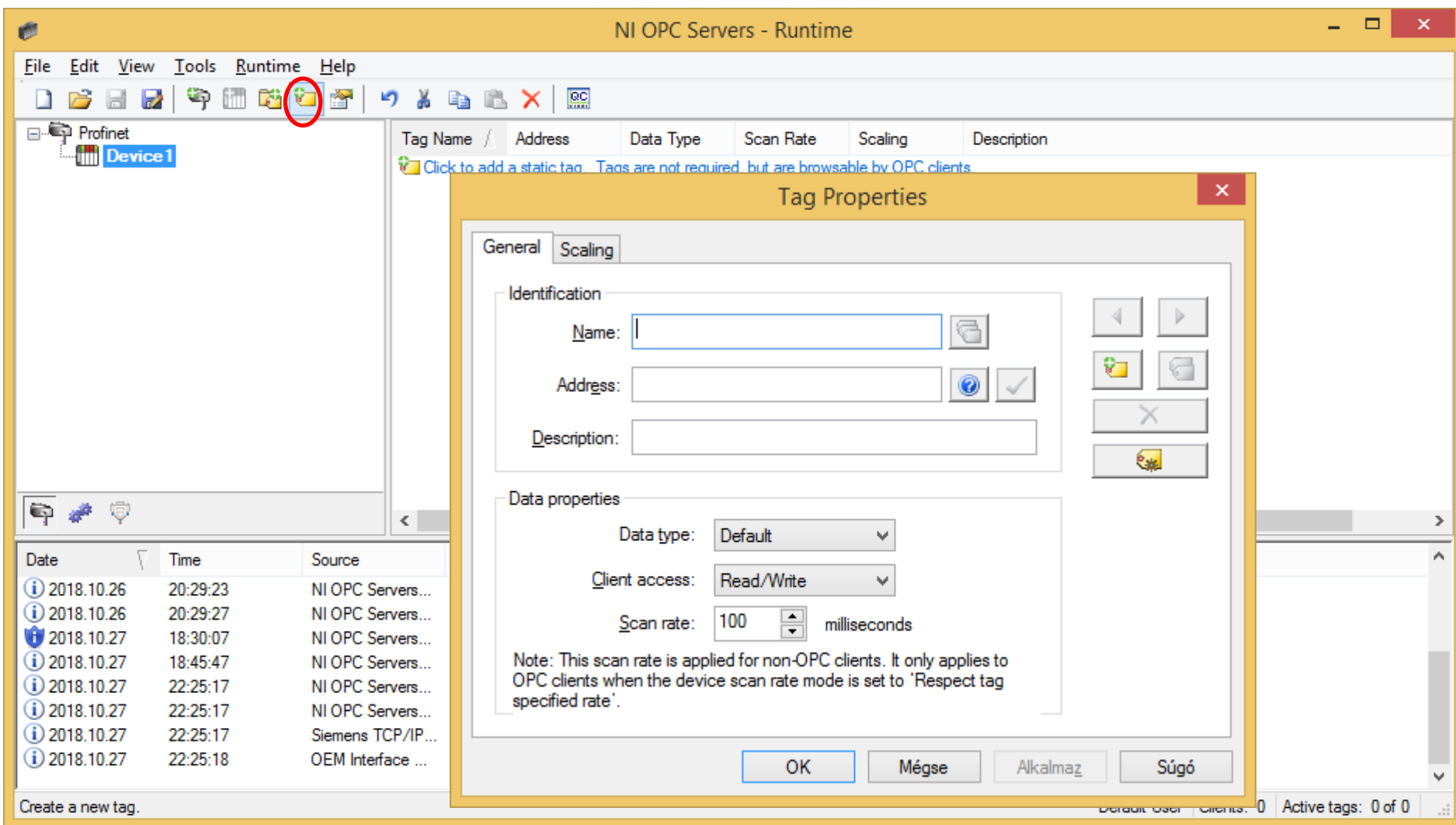
< Vissza

Tovább >

Mégse

Súgó

SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN



SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN

The screenshot displays two windows from the NI OPC Servers software. The top window, 'NI OPC Servers - Runtime', shows a tree view on the left with 'Profinet' expanded to 'Device1'. The main table lists tags for 'szelep' at address 'M0.0' with a Boolean data type, a scan rate of 100, and no scaling. A 'Quick Client' button is highlighted in the toolbar. The bottom window, 'OPC Quick Client - Untitled *', shows a tree view on the left with 'National Instruments.NIOPCServers.V5' expanded to 'Profinet.Device1'. The main table displays three items: 'Profinet.Device1._...' (Byte, Value 0), 'Profinet.Device1._...' (Byte, Value 1), and 'Profinet.Device1.s...' (Boolean, Value Unknown). The bottom status bar shows 'Item Count: 68' and a 'Launch the OPC Quick Client.' button.

NI OPC Servers - Runtime

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
szelep	M0.0	Boolean	100	None	

OPC Quick Client - Untitled *

Item ID	Data Type	Value	Timestamp	Quality	Update Count
Profinet.Device1._...	Byte	0	22:39:10.189	Good	1
Profinet.Device1._...	Byte	1	22:39:10.189	Good	1
Profinet.Device1.s...	Boolean	Unknown	22:39:13.189	Bad	1

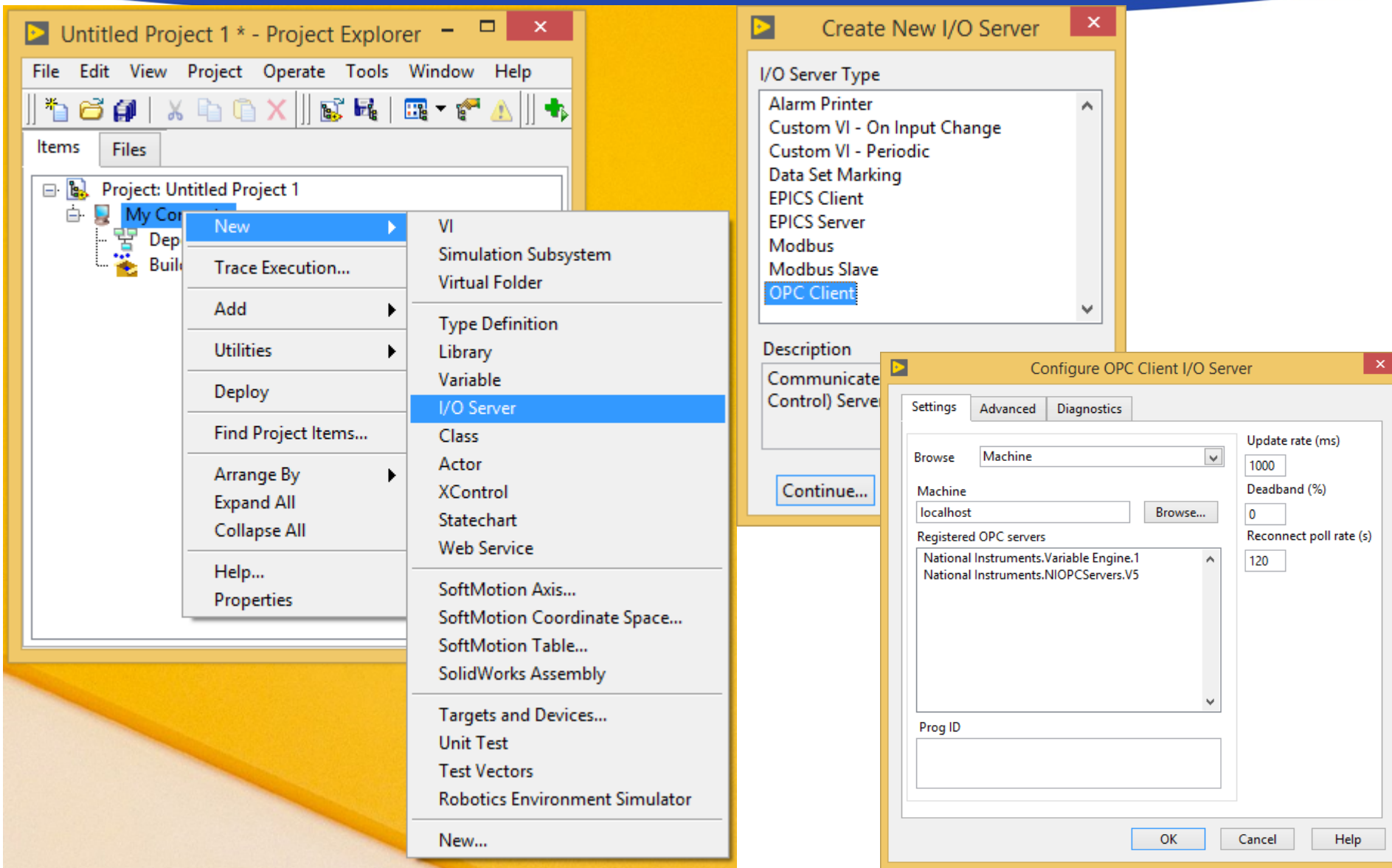
Ready

Item Count: 68

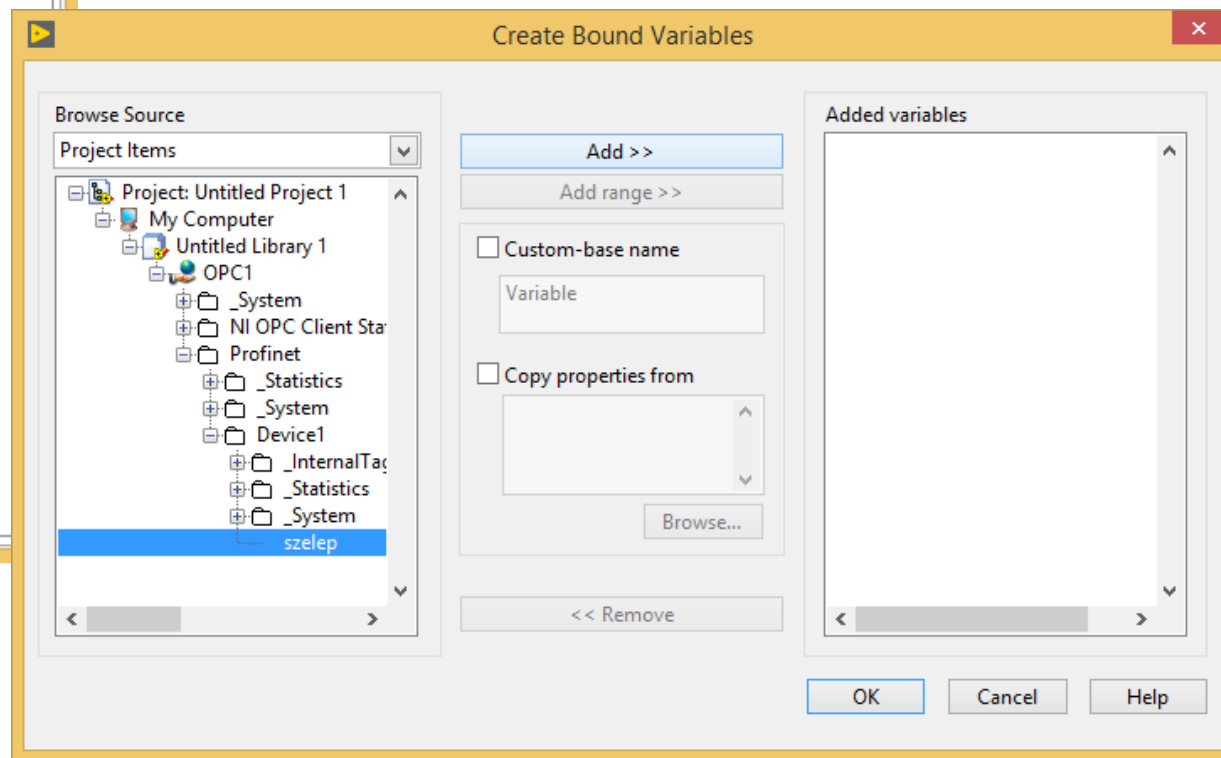
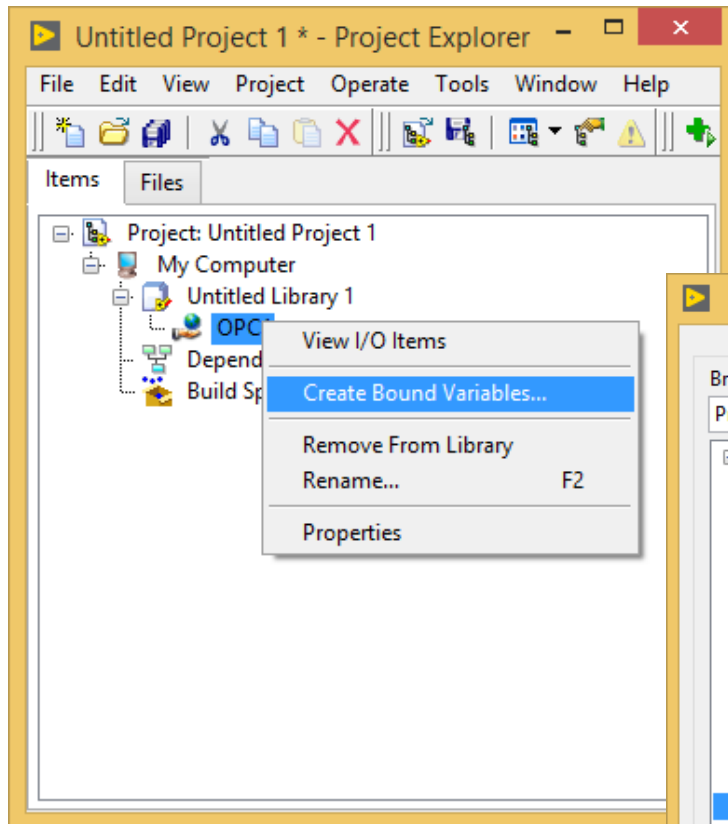
Launch the OPC Quick Client.

Default User Clients: 0 Active tags: 0 of 0

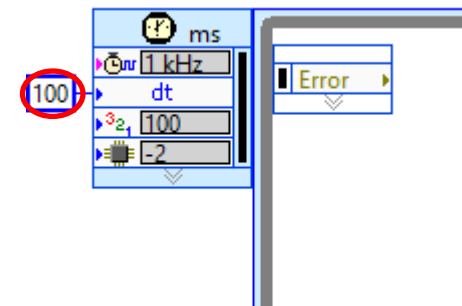
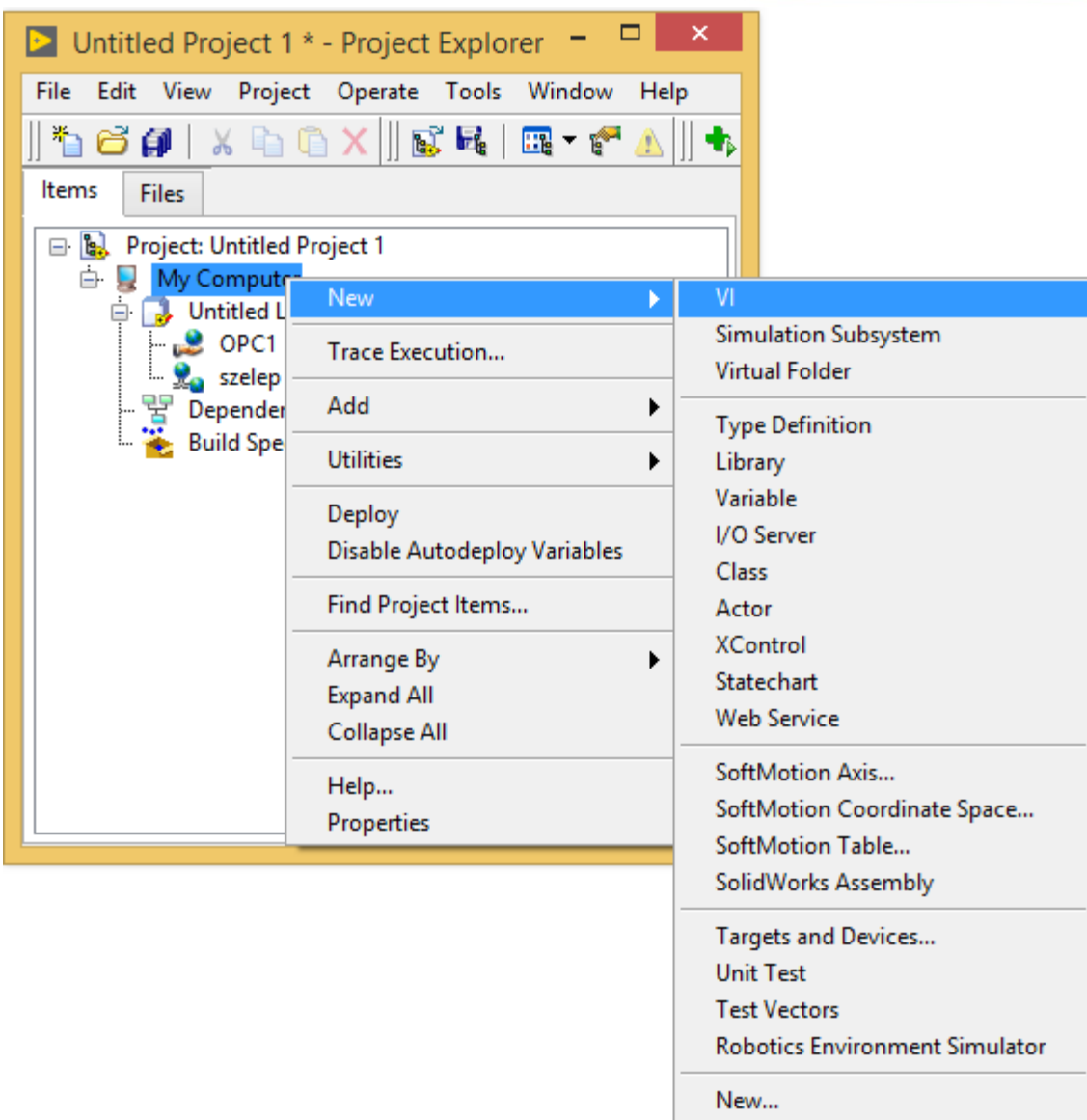
SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN



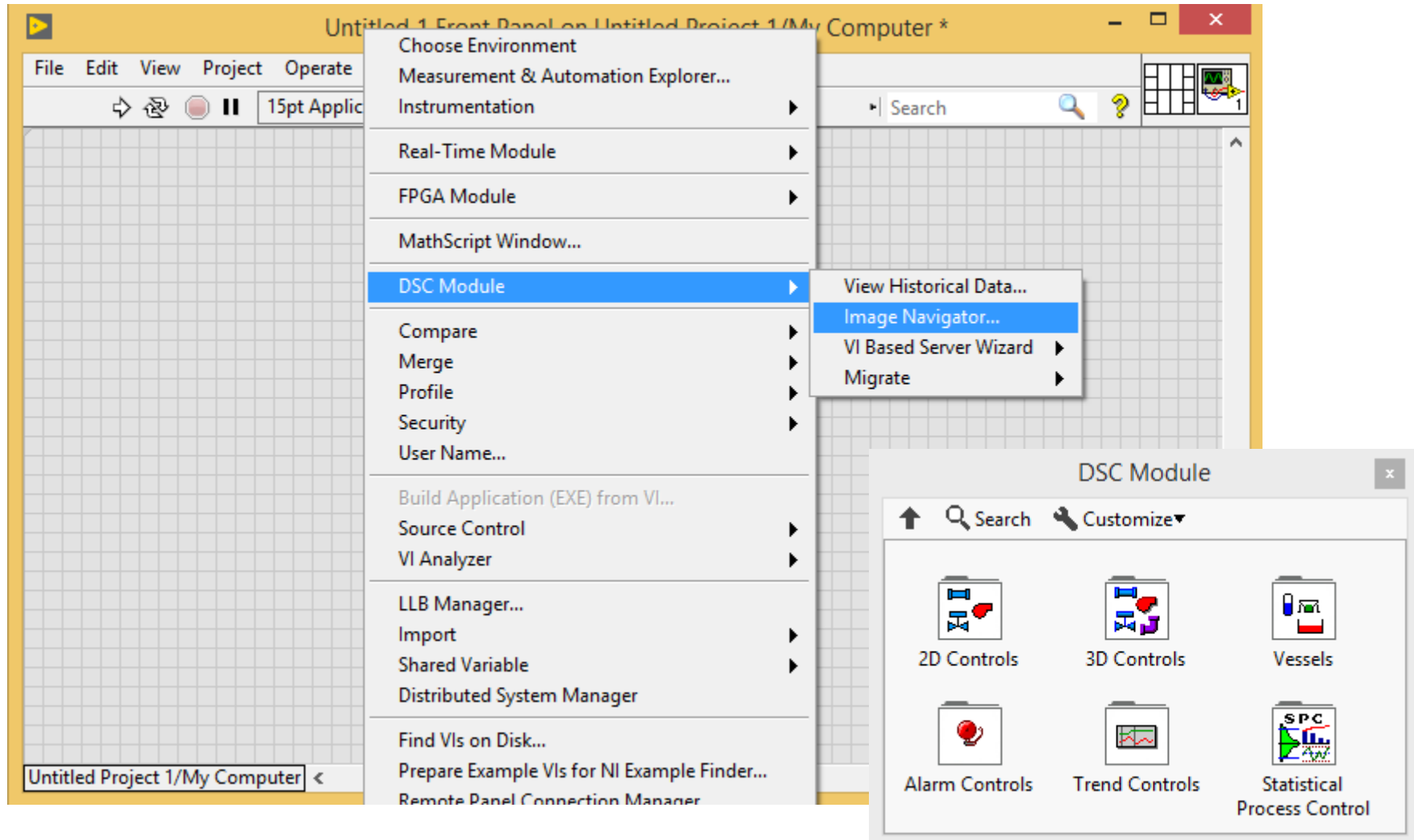
SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN



SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN

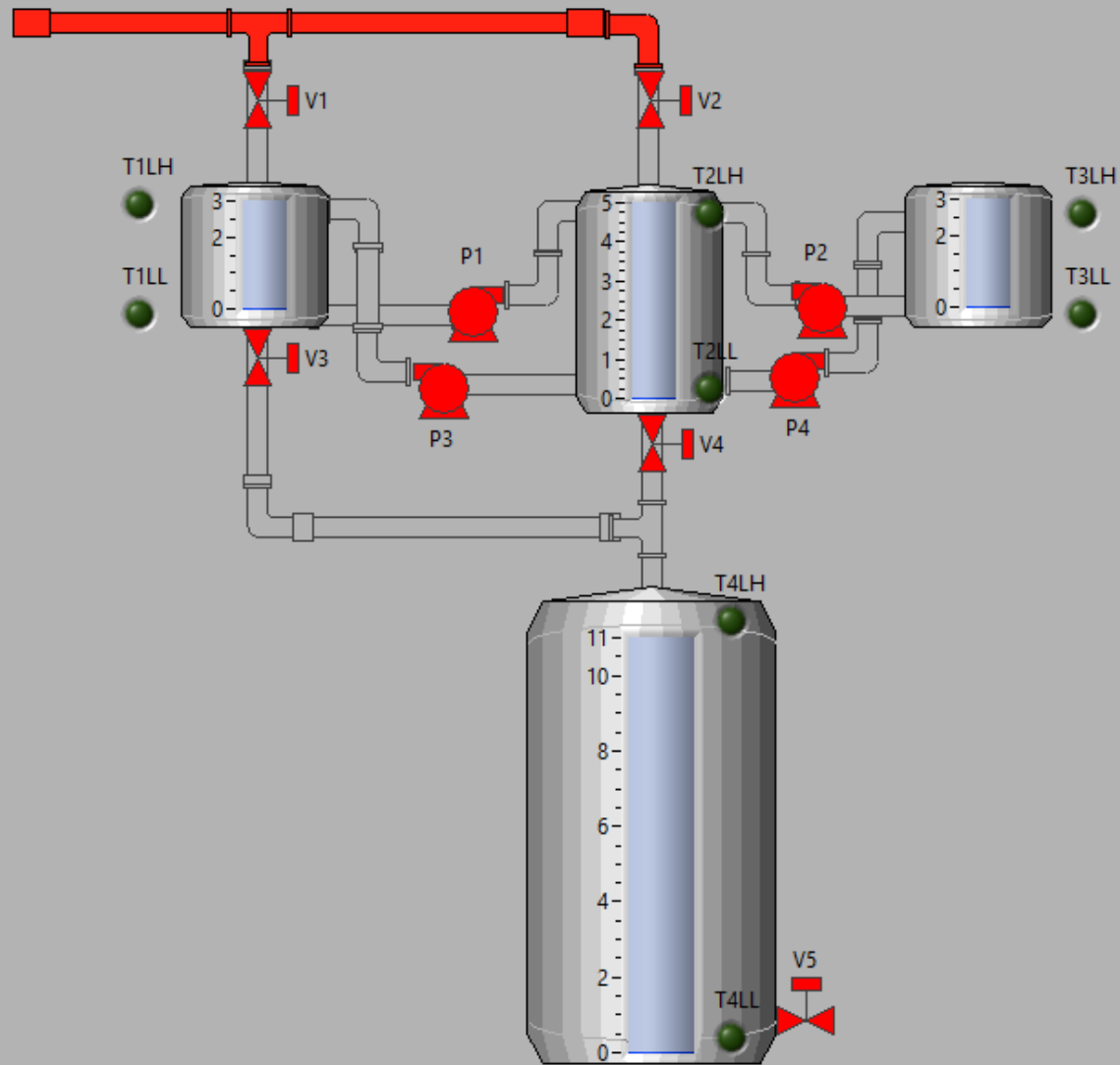


SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN



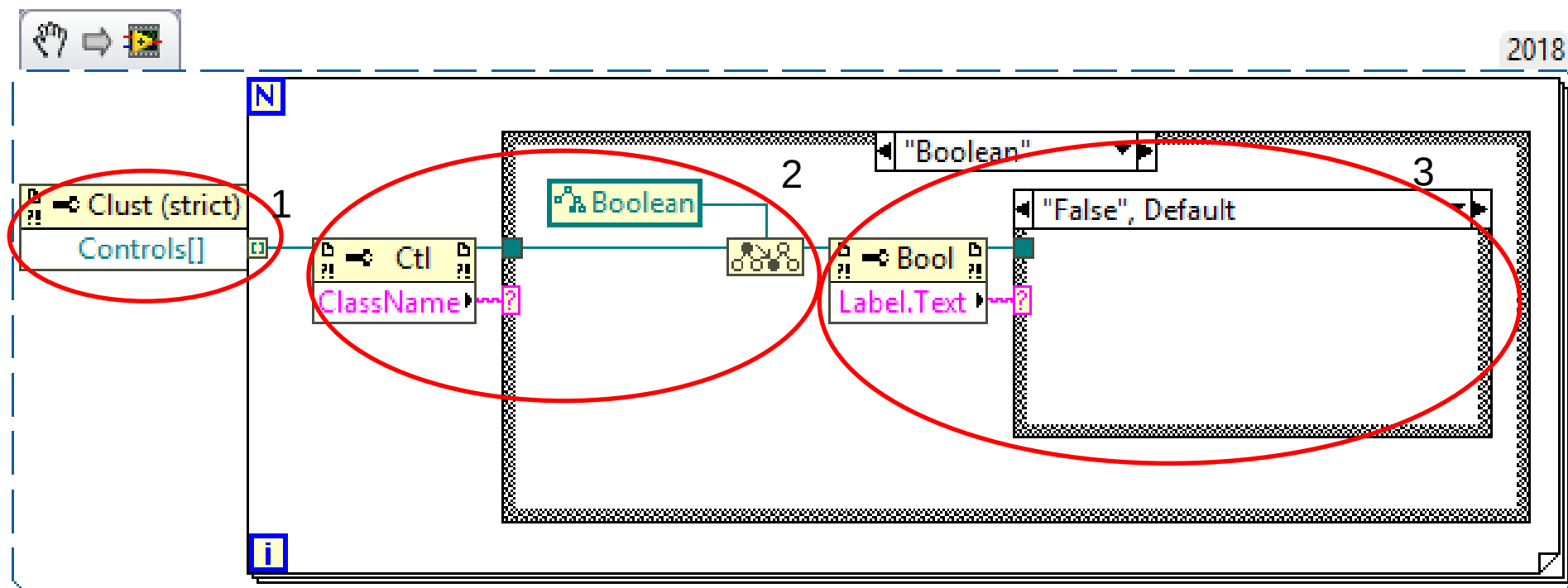
SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN

Cluster

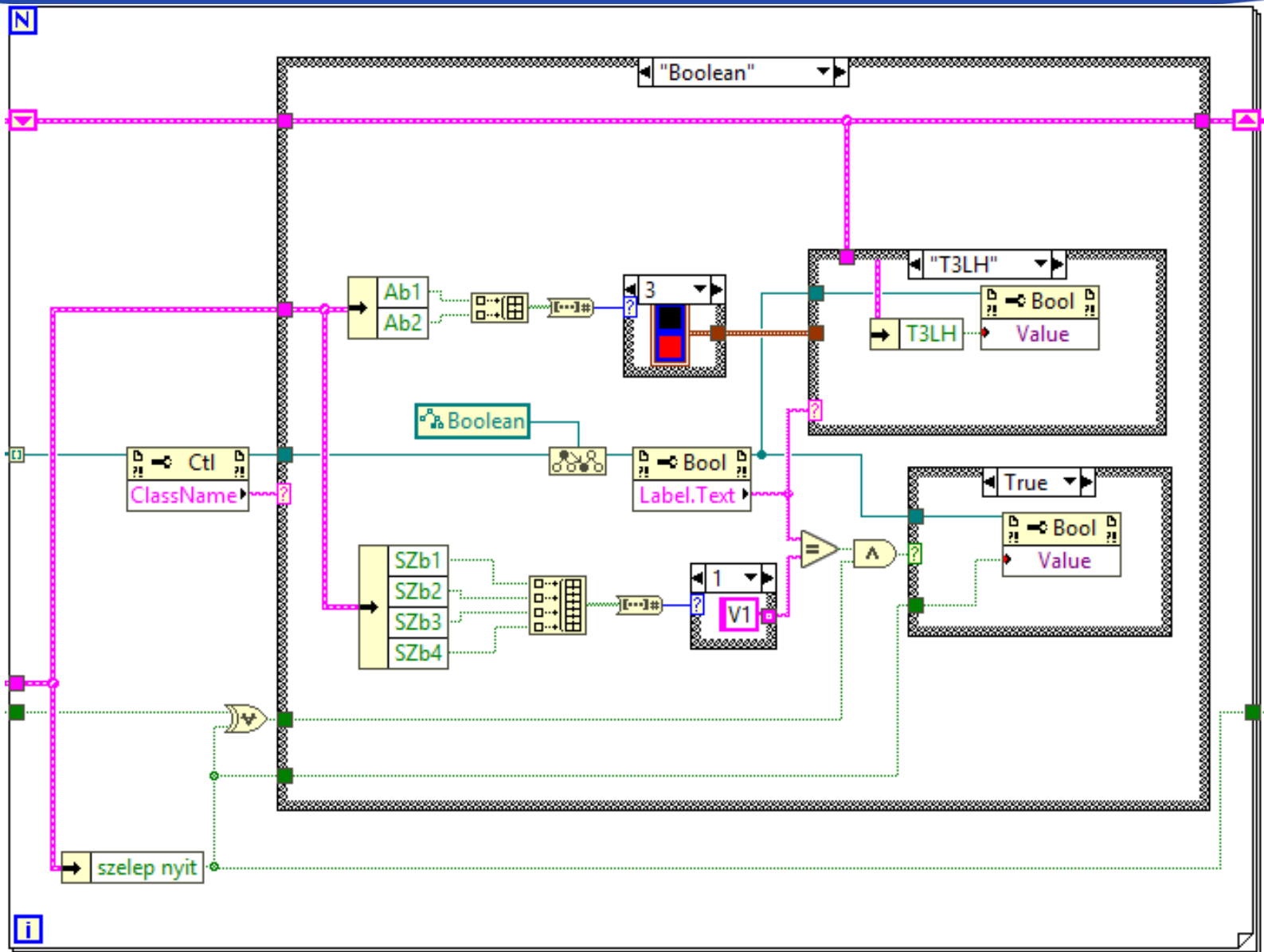


SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN

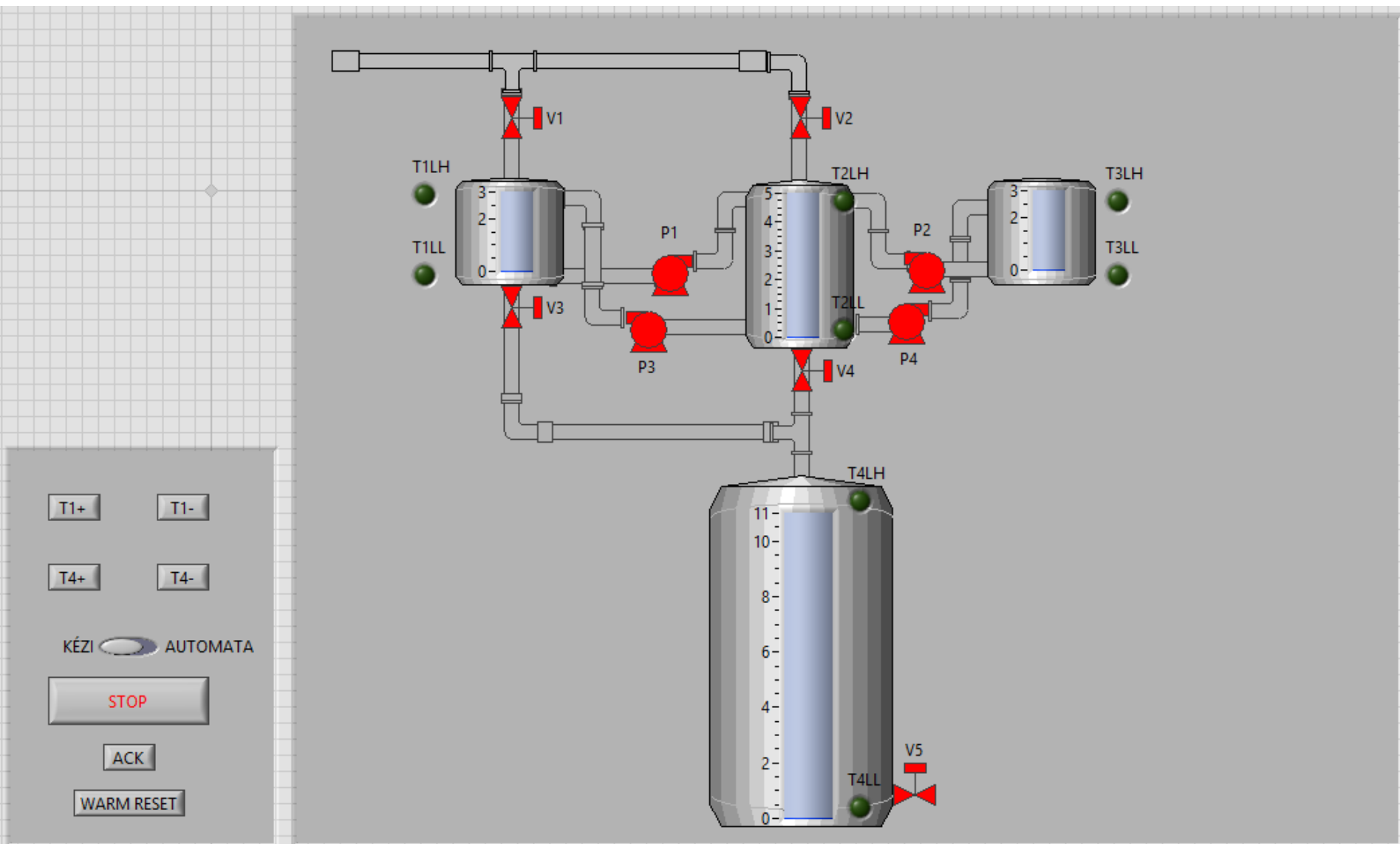
2018



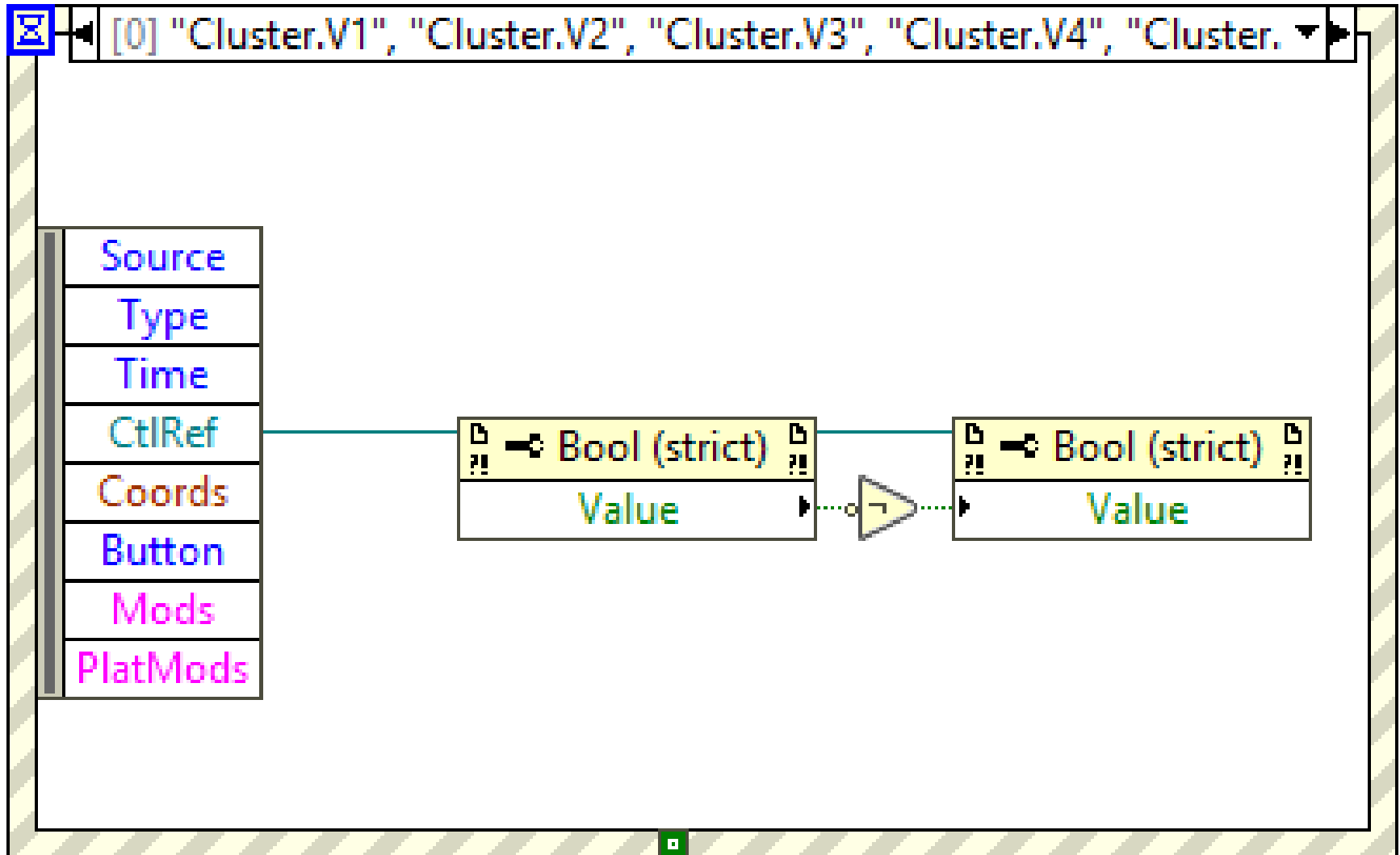
SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN



SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN



SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN



SCADA RENDSZER KÉSZÍTÉSE LABVIEW KÖRNYEZETBEN

A felvázolt keretrendszerrel a változókat időkritikusan lekezeljük, valamint jól karbantartható kódot írtunk. A szükséges ábrák mindegyikét elkészíthetjük a felhasználói adminisztráció kivételével, ez különleges eset. Windows enterprise edition esetében hozzárendelhetjük, hogy melyik felhasználó képes hozzáférni bizonyos adatokhoz, más esetekben nekünk kell leprogramoznunk ezt a hozzáférést ami adatbázis kezelést igényel.

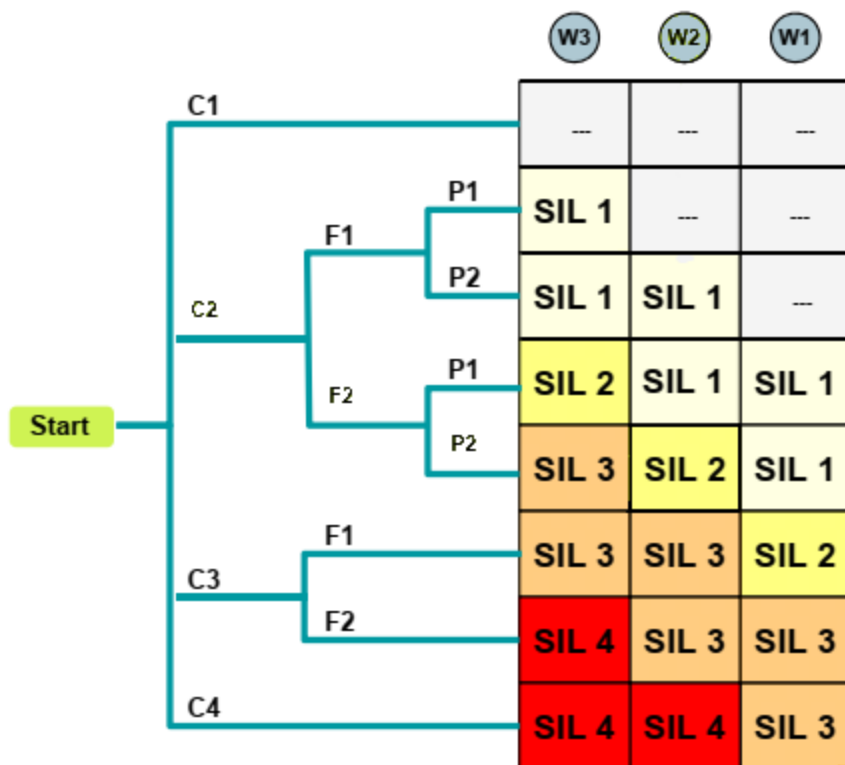
BIZTONSÁG

- Mivel a SCADA rendszerek által vezérelt eszközök veszélyes szituációkhoz vezethetnek, ha szándékból vagy gondatlanságból rossz utasításokat küldenek ki(pl.: STUXnet) a biztonságra kell törekednünk.
 - Fizikai biztonság esetén a SCADA rendszerhez való fizikai hozzáférést korlátozzuk valahogy (pl.: zárt ajtókkal, falakkal, zárankkal, őrkkel...)
 - Informatikai biztonság esetén a SCADA rendszerhez való informatikai hozzáférést korlátozzuk (pl.: jelszavakkal, informatikai infrastruktúra helyes felépítésével, vírusirtókkal, behatolás jelzőkkel, biztonságos protokollokkal...)

SAFETY INTEGRITY LEVEL (SIL)

- SIL egy 1-4-ig tartó skála ami egy berendezés biztonsági besorolását jelöli, minél nagyobb annál szigorúbb előírásoknak kell, hogy a berendezés megfeleljen tervezéstől kivitelezésen át az üzemeltetésig.

SAFETY INTEGRITY LEVEL (SIL)



	Következmény
C1	Enyhébb sérülés
C2	Komoly sérülés 1 vagy több személynek, 1 személy halála
C3	Több személy halála
C4	Nagyon sok személy halála
	Milyen gyakran vagyunk kitéve a veszélynek
F1	Ritkától inkább gyakori
F2	Gyakorítól folyamatos
	Lehetőség veszélyes esemény elkerülésére
P1	Lehetséges bizonyos feltételek mellett
P2	Lehetetlen
	Valószínűsége a veszély bekövetkeztének
W1	Kicsi
W2	Közepes
W3	Magas

MEGBÍZHATÓSÁG NÖVELÉSE

Az üzembiztonság javítására valamilyen szintű redundanciára van szükségünk. Relés vagy diszkrét logikájú rendszerek esetében ez többségi logikára alapszik (pl:3-ból 2 jelezzen). Mikrovezérlős rendszerek bár nagy megbízhatósággal működnek (de azért a WiFi routert is néha újra kell indítani), de az esetleg fellépő zavar katasztrofális is lehet.

MEGBÍZHATÓSÁG NÖVELÉSE

A redundancia lehet:

- Gépi – Berendezést többszörözünk

- Információ – Járulékos információ hozzáadása
pl: paritásbit

- Program – programegységek többszörözésével
növelik a megbízhatóságot pl: utasítás
ismétlése

Redundancia kialakítási szintjei lehetnek:

Elem, Modul, Rendszer

MEGBÍZHATÓSÁG NÖVELÉSE

Párhuzamos redundancia esetén az egységek párhuzamosan működnek, a helyes működéshez elég lenne egyetlen modul is.

Többségi elven működő redundancia esetén az egységek szintén párhuzamosan működnek, de a kimeneti jelük bekerül egy szavazó áramkörbe ahol többségi elv alapján jut a szavazóegység kimenetére

MEGBÍZHATÓSÁG NÖVELÉSE

Átkapcsolható redundáns egységeknél meghibásodáskor egy kapcsoló átkapcsolja a rendszer kimenetét a meghibásodott egységről a tartalék egységre. Ilyen rendszereknek az előnye, hogy hibamentes esetben a tartalék egység járulékos munkákat tud folytatni.

Fontos definiálnunk a **Hiba** és a **Meghibásodás** közti különbséget.

MEGBÍZHATÓSÁG NÖVELÉSE

Hiba esetén egy **rendszer** működése az üzemi feltételek betartása mellett a megkívánt működéstől eltér.

Meghibásodás esetén egy **elem** vagy **modul** paraméterei üzemi feltételek mellett a meghatározott határokon kívül esnek.

Nem minden meghibásodás vezet hibához, ezért fontosak a rendszeres tesztek mivel meghibásodásokat felfedhetjük mielőtt működési hibát okoznának.

MEGBÍZHATÓSÁG NÖVELÉSE

Hogy számszerűsíteni tudjuk PLC-s rendszerek üzemidejét szükségünk van a:

- MTBF – Mean Time Between Failure
- MTTF – Mean Time To Failure
- MTTR – Mean Time To Repair
- MT – Maintenance Time

Ezekkel meghatározható a PLC üzemideje %-ban:

$$\text{Üzemidő}[\%] = \frac{MTBF}{MTBF + MTTR + MT} \cdot 100\%$$

MEGBÍZHATÓSÁG NÖVELÉSE

A hibamentes várható élettartam:

$$R = e^{-\frac{t}{MTBF}}$$

Az irányítási cél elérése szempontjából a lehető legnagyobb rendelkezésre állás lenne a legkedvezőbb, de az igényeinket meghaladó megbízhatóság felesleges többletköltségekkel jár. Fontos meghatároznunk a megbízhatósági követelményeket.

MEGBÍZHATÓSÁG NÖVELÉSE

Megbízhatósági követelmények lehetnek:

- Leállás egyáltalán nem engedhető meg
- Hosszú idejű megállás nem engedhető meg
- Csak az adatbázis védelmét kell biztosítani

SAFETY PLC

- Egy módszere, hogy megfeleljünk a SIL szint előírásának a Safety (biztonsági) PLC-k használata.
- Tipikusan biztonsági feladatoknál párban használunk PLC-eket, ha az egyik meghibásodik a másik biztonságosan leállítja a folyamatot. Ebben a konfigurációban a 2 PLC keresztbe van kötve.
- Safety PLC-k rendelkeznek redundáns mikroprocesszorral és memóriával amit watchdog áramkör figyel.

SAFETY PLC

- Safety PLC bemenetek ellenőrizni tudják a saját működésüket így ellenőrizhetik menet közben, hogy még helyesen működnek.
- Safety PLC kimenetek 2 mikroprocesszorra vannak rákapcsolva, ha nem ugyanazt a kimenetet adják mindketten a kimenet egy biztonsági értéket vehet fel.
- Az ilyen PLC-ket alkalmazhatjuk működésbiztos és veszélybiztos konfigurációban.

SAFETY PLC

- Működésbiztos: (hibatűrő vagy fault-tolerant) ha hiba lép fel a PLC képes lekezelni a hibát és a berendezés tovább működik.
- Veszélybiztos: ha hiba lép fel a PLC képes a berendezést biztonságosan leállítani.

SAFETY PLC

Hibatűrő konfigurációban két PLC vagy egy PLC-n belül 2 microprocesszor szinkronizáltan és párhuzamosan fut, egymással kommunikációs kapcsolatban vannak, közülük az egyik aktív ez irányítja a folyamatot a másik passzív, de a kommunikációs kapcsolat révén bármikor átveheti a folyamat irányítását ezt nevezik melegtartaléknak (hot-standby)

Hibatűrő konfigurációnál a következő feladatokat kell végrehajtania a PLCnek:

- Adatcsere és hibakezelés
- Szinkronizáció
- Önteszt

SAFETY PLC

Ahol a biztonságos működés az első számú követelmény mint gáz és olajszállítással kapcsolatos automatikák, vegyipar, nukleáris erőművek veszélybiztos PLC konfiguráció szükséges. Itt is két PLC vagy egy PLC-n belül 2 mikroprocesszor van kapcsolatban egymással a fő különbség, hogy a két egység folyamatosan összehasonlítja egymás állapotait, eredményeit és megakadályozza a veszélyes válaszok kijutását.

Veszélybiztos konfigurációnál a következő feladatokat kell végrehajtania a PLCnek:

- Adatcsere és válasz a hibára
- Szinkronizáció
- Önteszt

A fő különbség a hibatűrő üzemmódhoz képest a válasz a hibára ami lehet:

- Teljes egység stop állapotba kerül
- Csak a veszélyes komponensek állnak le
- Más a felhasználó által definiált válasz

- Az OPC egy olyan alkalmazás amivel ipari vezérlőkkel lehet kommunikálni
- Elsőnek OLE for Process Control volt ahol az OLE Object Linking and Embedding-et jelentette.
- Ma az Open Platform Communications-nek a rövidítése.
- Az UA (Unified Architecture) 2008-ban lett megalkotva és az aktuálisan legújabb technológia különböző gyártók eszközei közötti kommunikációra

INDUSTRIAL INTERNET OF THINGS (IIOT)

- Az IIoT vagy más néven Ipar 4.0 az aktuális ipari forradalom.
- A többi ipari forradalommal ellentétben ezt előre bejelentették és tudatosan fejlesztenek ennek irányába.
- A cél többszintű:
 - Az automatizálási piramis vertikális szintjei közötti jobb átjárás
 - A több telephely horizontális szintjének összehangolása
 - 1 elemű sorozat gyártása

INDUSTRIAL INTERNET OF THINGS (IIOT)

- Az alkalmazott technológiák:
 - Big Data adatbányászat
 - Felhő és kód alapú infrastruktúra
 - Termékek Digitális iker
 - Blockchain
 - Smart termékek
 - Kiterjesztett valóság

BIG DATA

Big Data adatbányászat alatt olyan mértékű adatok feldolgozását értjük amik több TB méretűek, ezekben az adatokban keresünk olyan összefüggéseket amik nem láthatóak szabad szemmel.

Egy tipikus példa a repülőgépek adatainak elemzése amit repülés közben mértek az érzékelők, ez több TB-nyi adat ami első ránézésre elképzelhetetlen mennyiség csak rezgés adatokból.

BIG DATA

Ezeket az adatokat adatbányászok fejlett módszerekkel és programokkal elemzik, hogy következtetéseket vonjanak le a repülőgép állapotáról és így meghatározzák a hibák hol összpontosulnak, hogy lehet jobban kihasználni a berendezést esetleg más eloszlással az utasok területén vagy más anyagok használatával egy következő gyártmánynál.

Már mind használunk felhő alapú programokat vagy tisztában vagyunk létezésükkel. A Google Drive, Dropbox és egyéb online tárhelyek felhőben tároló alkalmazások, de az ipar 4.0 ban nem erre támaszkodunk csak, hanem felhő alapú infrastruktúrára. Ebben az esetekben nem egy külön meghatározható számítógépeken megy a folyamatunk hanem több számítógép által alkotott „felhőben”

FELHŐ ÉS KÖD

Az ilyen módon elosztott infrastruktúra lehetővé teszi cégek számára, hogy az alkalmazásaikat gyorsabban fejlesszék nagyobb megbízhatóság mellett kevesebb karbantartási igénnyel és ne foglalkozzanak számítógép infrastruktúra kiépítéssel. Egy másik előnye ennek a rendszernek a rugalmasság. Képzeljük el, hogy az előbb felhozott több TB-os adatot fel kell dolgoznunk és szükségünk van 100 számítógépre, hogy végezzünk egy óra alatt viszont a napi működéshez csak 10 is elég.

FELHŐ ÉS KÖD

Ha vennénk 100 számítógépet amiből csak 10-et használnánk folyamatosan 90-et meg csak néha nagy lesz a kihasználatlan gépek száma. Felhő alapú infrastruktúra esetén lehetőség van „annyit fizetni amennyit használsz” ami azt jelenti, hogy ha hirtelen megugrik az igényünk megkapjuk az extra 90 magot majd mikor elvégződött a számítás csak a felhasznált időt számlázzák ki nekünk.

Köd alapú infrastruktúra esetén hasonló koncepcióink vannak mint a felhő esetén a fő különbség az ún. edge eszközökben van. Még a felhő egy vagy több központi datacenterrel biztosítja a szolgáltatásait addig a köd felhasználja az adatszállításban használt eszközöket (edge node) számítások végrehajtására így elosztja a terhelést egyenletesebben és gyorsítja a feldolgozást mivel másolás közben is feldolgoz adatokat.

Tervezéskor egy koncepciót készítünk, gyártás közben ennek a koncepciónak egy példányát feltöltjük a gyártási folyamat közben szerzett értékekkel így lesz egy képünk termék fizikai állapotáról.

Folyamatos frissítéssel ezt az állapotot felügyelhetjük és okos szolgáltatásokkal szólhatunk, hogy szervizelésre vagy cserére van szükség (pl: abronycsoknál)

DIGITÁLIS IKER

Képzeljünk el egy terméket ami egy 6002-es csapágyat amiben vannak érzékelők és egy kis intelligencia. A rajta lévő elektronika képes adatkapcsolatot létesíteni egy edge node-al és rezgési adatokat továbbítani a központi szervernek. Ennek a terméknek és minden másiknak a sorozatában keletkezett egy digitális iker, ennek a csapágyynak az iker fel van frissítve a mérési adataival. Így adatelemezéssel meghatározható az élettartamának melyik szakaszában van.

DIGITÁLIS IKER

Ha összevetjük ennek a terméknek a digitális ikrét a sorozat többi elemével kapunk egy általános képet arról, hogy mekkora élettartammal számíthatunk. Ez egy elképzelt termék volt, de már létezik hasonló szolgáltatás kamionoknál, itt a gumi abroncsokba épített érzékelők gyűjtenek rezgési adatokat amiből következtetnek a gumi kopásának állapotára így csak akkor cserélik ki amikor kell.

BLOCKCHAIN

Ezt a technológiát a Bitcoinnál alkalmazták elsőnek, a feltalálóját nem ismerik csak az álnévét Satoshi Nakamoto. A technológia lehetővé teszi a bizalom nélküli rendszereket és így a biztonságban használható fel. Mivel mindent szeretnénk interneten elérhetővé tenni így eddiginél nagyobb szintű biztonságra van szükségünk, lehet ez lesz a megoldás.

BLOCKCHAIN

A blockchain (magyarul blokklánc) egymáshoz fűződő adatelemekből (blokkokból) áll. Minden block egy hash-el kezdődik amit úgy kapunk, hogy az előtte levő blokkot hasheljük, a következő elem az adat amit a blokk tartalmaz és végül az aktuális block hashét. Az első block-ot kézzel adjuk hozzá és a többi legenerálhatjuk.

BLOCKCHAIN

Bitcoin esetén a block-ok a tranzakció adatait tartalmazzák (kitől, kinek mennyit küldünk), így a blockchain egy könyvelése az eddigi tranzakcióknak. Ha hamisítani akarnánk múltbéli adatokat akkor a módosított block hashét nem elég kiszámolni mivel az összes rákövetkező block hash-e is függ tőle, tehát ez sokáig tartana. Bár a hashelés maga kevés ideig tart 1 block módosítása a Bitcoinnál kb 10perc.

BLOCKCHAIN

Ha mégis sikerülne ezt megcsinálni, még akkor se sikerülne a módosítás ugyanis a blockchain egy elosztott rendszer. Ez azt jelenti, hogy mindenkinek van egy másolata az eredeti blockchainről és többségi szavazat alapján egyeznek meg, hogy melyik az eredeti, így a módosítást az aktuális blockon az összes rákövetkezőn el kell végezni a felhasználók 51%-nál, hogy elfogadott legyen a módosítás.

BLOCKCHAIN

A blockchain technológiát nem csak ilyenfajta könyvelésben lehetne használni. Képzeljük el, hogy a mi termékünk dokumentációja az adat és ezt szeretné valaki meghackelni, hogy gyártáskor selejt termékeket gyártsunk, ilyenkor hasonló eljárásen kell keresztülmennie mint amit az előbb leírtunk.

SMART TERMÉKEK

Smart termékek olyan termékeke amik tartalmazzák a gyártásukhoz szükséges összes információt. A termékek gyártás közben rendelkeznek egyedi azonosítóval ahogy végighaladnak a gyártósoron a gépek ezt érzékelik és ennek megfelelő módon avatkoznak be. Így lehetséges az utolsó pillanatig módosítani egy rendelésen.

KITERJESZTETT VALÓSÁG

Ez talán a legjobban a sci-fi világ szülötte, képzeljünk el egy olyan szemüveget ami rávetít a mi általunk látott képre, így megjeleníthet számunkra hasznos adatokat (Terminátor látás). Ezek már léteznek, ilyen a Microsoft Hololens nevű technológiája. Ilyen eszközök lerövidítenék a betanulási időszakokat, egy munkásnak mit kell tennie szerelés közben, segíthetik a tervezést, mutathatják az utat egy reptéren stb.

FELHASZNÁLT FORRÁSOK

- Automatika mérnököknek, ISBN 9631855791
- SCADA: Supervisory Control and Data Acquisition 4th edition, ISBN 1936007096
- Ipari kommunikációs rendszerek, ISBN 9789630658133
- Programozható irányítóberendezések, hálózatok és rendszerek, ISBN 9631618978
- Designing SCADA application software: a practical approach, ISBN 9780124170001
- Az ábrák a google images-ből származnak

A TANANYAG AZ EFOP-3.5.1- 16-2017-00004 PÁLYÁZAT TÁMOGATÁSÁVAL KÉSZÜLT.

KÖSZÖNÖM A FIGYELMET!

SZÉCHENYI  2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE