

Příloha 1 - Příklady programů

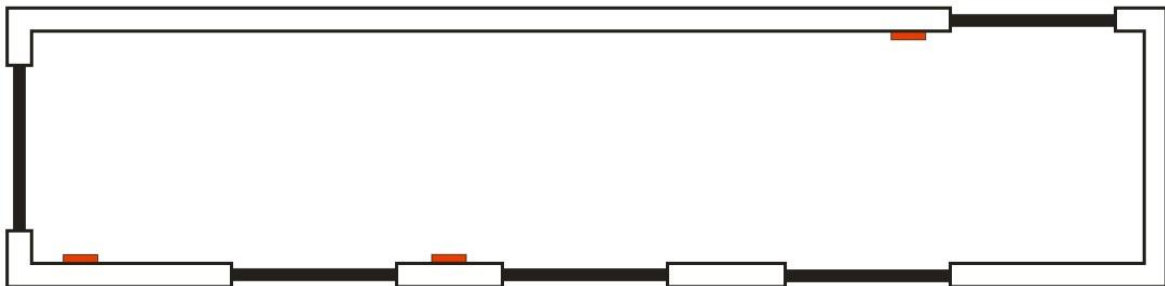
Tomáš Gromus

OBSAH

1	PŘÍKLADY PROGRAMU SCHODIŠŤOVÉHO VYPÍNAČE DLE NORMY IEC 61131-3.....	3
2	SOFTWAREVÉ A HARDWAROVÉ PROPOJENÍ PLC S MODELY.....	5
3	MODEL AUTOMATICKÉ PRAČKY.....	6
4	MODEL KŘIŽOVATKY.....	9
5	MODEL HYDRAULICKÉ POSUVNÉ JEDNOTKY.....	12
6	MODEL MÍŠÍCÍ JEDNOTKY.....	17
7	VIZUALIZACE MODELŮ POMOCÍ WEBMAKER.....	19
8	DALŠÍ MOŽNOSTI VYUŽITÍ KNIHOVNY COMLIB.....	21

1 PŘÍKLADY PROGRAMU SCHODIŠŤOVÉHO VYPÍNAČE DLE NORMY IEC 61131-3

Zadání: Uvažujme chodbu, ve které je pět dveří. V této chodbě jsou nainstalovány tři vypínače. Vytvořme pro toto zadání program tak, aby se vypínače chovali jako schodišťový vypínač. To znamená, jakýmkoliv vypínačem z kteréhokoliv místa můžeme světla rozsvítit nebo zhasnout.



Vypracování: Nejprve potřebujeme získat logickou funkci pro ovládání těchto vypínačů. Podle ní pak napíšeme program. Uděláme si tedy logickou tabulku.

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

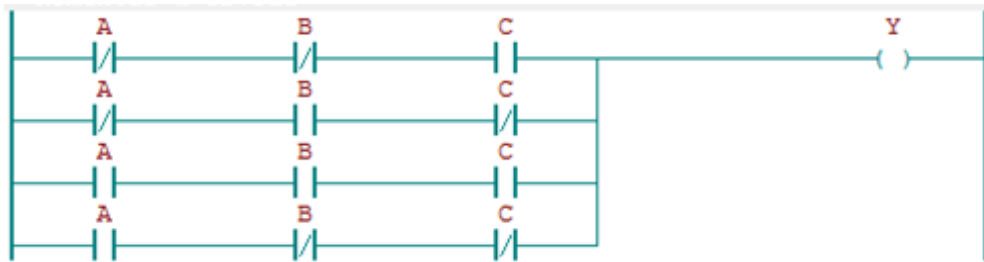
Z tabulky získáme logickou funkci: $Y = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C}$

Nyní už můžeme začít psát programy v jednotlivých jazycích. Pro znázornění zde uvedu všechny typy programování podle normy.

IEC ST:

$Y := ((\text{not } A) \text{ and } (\text{not } B) \text{ and } C) \text{ or } ((\text{not } A) \text{ and } B \text{ and } (\text{not } C)) \text{ or } (A \text{ and } B \text{ and } C) \text{ or } (A \text{ and } (\text{not } B) \text{ and } (\text{not } C));$

IEC LD:

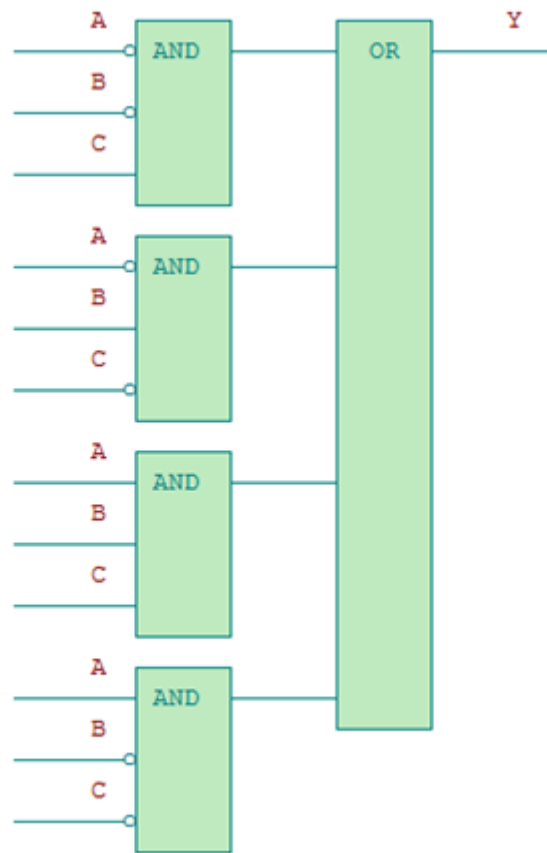


IEC IL:

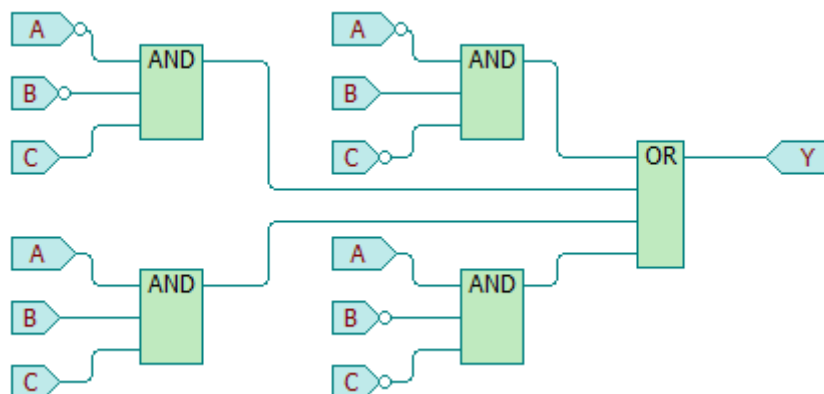
```

ldn    A
andn   B
and    C
or(    A
and    B
andn   C
)
or(    A
and    B
and    C
)
or(    A
andn   B
andn   C
)
st     Y
    
```

IEC FB:

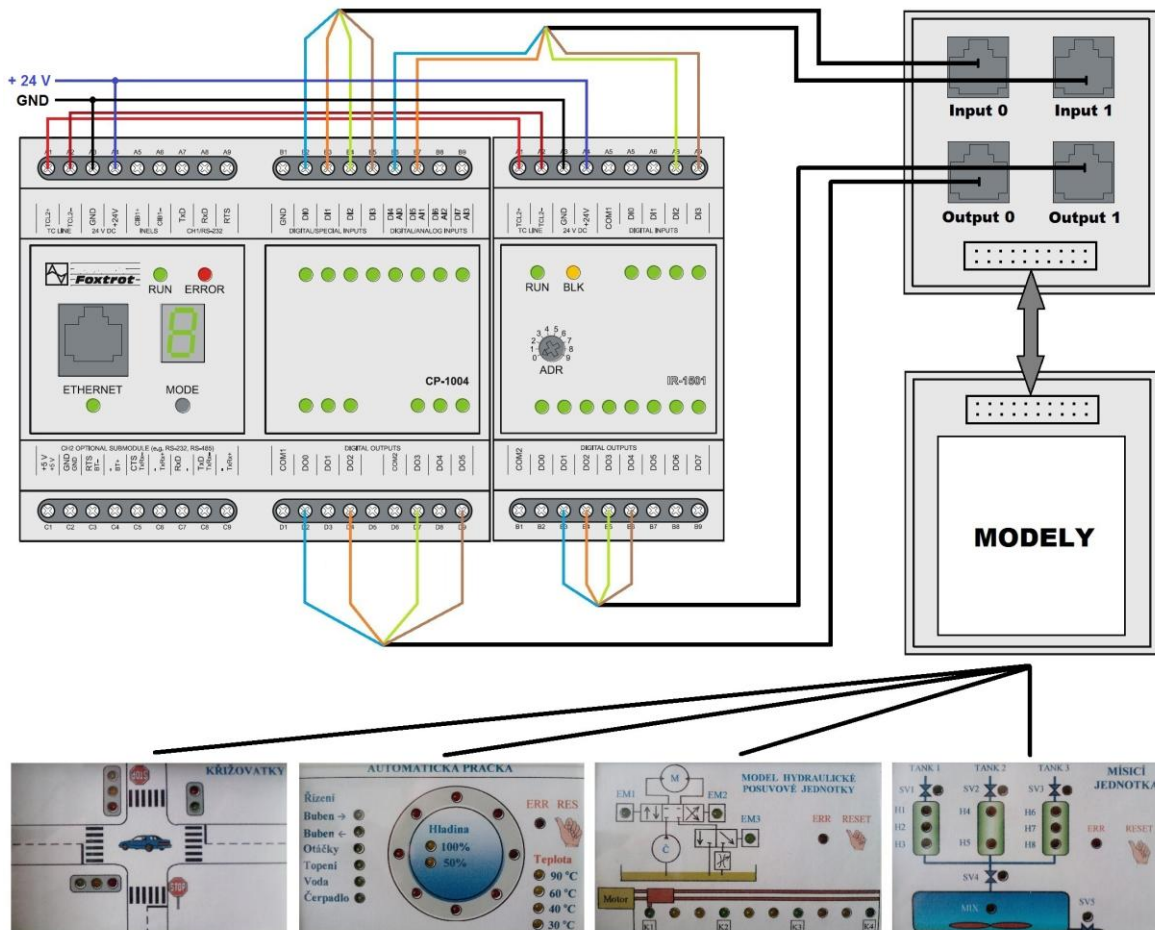


IEC CFC:



2 SOFTWAREVÉ A HARDWAROVÉ PROPOJENÍ PLC S MODELY

V laboratoři jsou používány PLC Tecomat Foxtrot propojené s EDU-mod modely automatické pračky, křižovatky, hydraulické posuvné jednotky a mísící jednotky. Schéma zapojení vypadá následovně:



Nastavení softwarového propojení provedeme v *Manažer projektu*->*Typ sítě*. Jelikož jsou PLC v laboratoři propojeny ethernetem, zvolíme typ připojení *Ethernet*. Vložíme IP adresu, kterou najdeme nalepenou na boku PLC.



Dále nastavíme hardware v *Manažer projektu*->*Hw*->*Konfigurace HW*. Zde v pravém dolním rohu klikneme na tlačítko *Načíst z PLC*. Načtou se nám všechny připojené moduly.

3 MODEL AUTOMATICKÉ PRAČKY

Zadání: Máme vytvořit v prostředí Mosaic program k modelu automatické pračky. Pračka se bude ovládat pomocí potenciometru a tlačítka (např. potenciometr A_A a tlačítko B_B). Vytvoříme také vizualizaci ve WebMakeru. Potenciometrem nastavíte váhu prádla, které nám určí napuštění vody, a potvrdíme tlačítkem. Následně opět potenciometrem vybereme režim praní a taktéž potvrdíme tlačítkem. V tuto chvíli by měla pračka začít prát. Po skončení praní bude možno opět pračku nastavit.



Obr. X1 Model automatické pračky



Obr. X2 Potenciometr a tlačítko

Vypravování: Model pračky má 7 vstupů a 6 výstupů.

Vstupy: Buben->, Buben <-, Otáčky, Topení, Voda, Čerpadlo

Výstupy: Teploty 30 °C, 40 °C, 60 °C, 90 °C, Hladina 100%, Hladina 50%

Je zde také tlačítko pro reset, když dojde k nějaké chybě. Tento model určuje dva druhy chyb. Jsou to opravitelná chyba, ta nastává, když dojde současně povel točení bubnu vpravo i vlevo. Začne blikat chybová dioda a buben se nebude otáčet. Po odstranění tohoto kolizního stavu se bude buben normálně otáčet. Pak jsou tu chyby neopravitelné, když přeteče prací vana nebo teplota vody stoupne nad 90 °C. Při této chybě se rozsvítí chybová dioda a musíme použít tlačítko reset.

Tento model simuluje napouštění vody a ohřívání automaticky!

Naše pračka bude mít devět režimů: máchání, odstředování, vyvářka, hedvábí, vlna, nemačkové, sportovní, mix a manuál.

Každý režim bude mít různé teploty a délku máchání. Při odstředování se například nebude máchat. V režimu manuál si nastavíme délku máchání a teplotu.

Pro požítý potenciometr musíme nastavit v *Manažer projektu*->*Hw*-> *Konfigurace HW* ->*I/O nastavení analogový vstup* podle obrázku:

☑ Kanál A10

Napěťový rozsah 0 ÷ 10 V

Předávat hodnotu

Binární hodnota (FS)

Hodnota v inženýrských jednotkách (ENG)

Normalizovaná hodnota (PCT)

Režim filtrace

Časová konstanta 0 [s]

WebMaker může vypadat například takto:



Vypracovaný program je možno nalézt na CD: Programy/EDU-mod/Pracka.zip

V programu byly použity následující jazyky: ST, LD, FBD, IL. Příklady vybraných částí:

Část programu v jazyce ST. Slouží po výběru manuálního režimu k ručnímu nastavení.

```

if NOT nastaveni then //pokud zvolíme manual, tak se nastavuje dodatečně teplota a počet máchání
  if menu then
    if odstredovani then faze3:=true; end_if;
    if manual then
      if NOT nastaveni3 then
        N_prubehu:=real_to_int(potenciometrA);
      end_if;
      if NOT nastaveni4 and nastaveni3 then
        case real_to_int(potenciometrA) of //regulace teploty pomocí relé
          0..2 : teplota:=30; tep:=1;
          2..5 : teplota:=40; tep:=2;
          5..7 : teplota:=60; tep:=3;
          7..9 : teplota:=90; tep:=4;
        end_case;
      end_if;
      if NOT nastaveni3 then
        if t1B then pom3:=true; end_if;
        if NOT t1B and pom3 then nastaveni3:=true; pom3:=false; end_if;
      end_if;

      if NOT nastaveni4 and nastaveni3 then
        if t1B and (hmotnost>0) then pom4:=true; end_if;
        if NOT t1B and pom4 then nastaveni:=true; faze1:=true; menu:=false; pom4:=false; end_if;
      end_if;

    end_if;
  else faze1:=true;nastaveni:=true;
  end_if;
end_if;

```

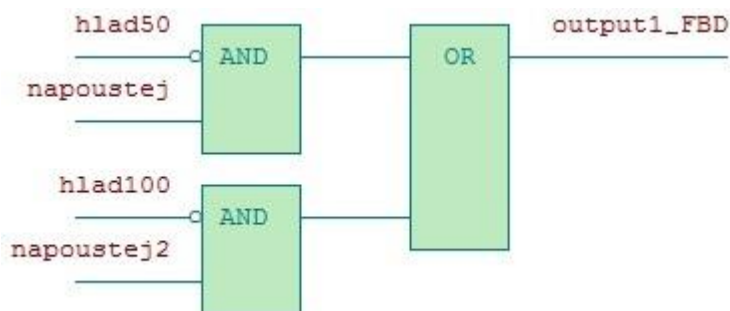
Část programu v jazyce LD. Zjistí nastavenou teplotu a udržuje ji.

Zjistí nastavenou teplotu a pak se stará o udržování této teploty.

0001



Část programu v jazyce FBD. Stará se o napouštění vody.



Část programu v jazyce IL.

Stará se o točení bubnu.

```

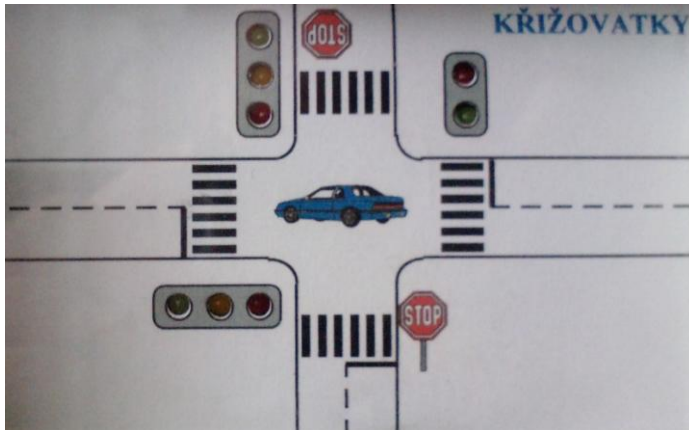
ld   cas
lt   4
st   leva_IL

ld   cas
gt   5
and(
ld   cas
lt   9
)
st   prava_IL

```

4 MODEL KŘIŽOVATKY

Zadání: Máme vytvořit v prostředí Mosaic program k modelu křižovatky. Křižovatka bude pracovat ve dvou režimech. Denní režim od 6:00 do 22:00 a noční režim od 22:00 do 6:00. Na hlavní cestě bude semafor pro chodce s tlačítkem (např. tlačítko B_B), a na vedlejší cestě bude snímač počtu vozidel, který budeme simulovat potenciometrem (např. potenciometr A_A). Použijte operátorská panel ID-07, přes který bude obsluha mít možnost měnit režimy.



Obr. X3 Model automatické pračky



Obr. X4 Potenciometr a tlačítko



Obr. X5 Operátorský panel ID-07

Vypracování: Křižovatka je pasivní modul. Skládá se pouze z LED diod představujících semaforey. Jsou zde semaforey: hlavní, vedlejší, pro chodce.

Nastavení: Po založení projektu musíme nastavit propojení PLC s operátorským panelem ID-07. Otevřeme si *Manažer projektu*->*Hw*->*Konfigurace HW*. Můžeme použít tlačítko *Načíst s PLC* a následně otevřít *Nastavení CPU*. Zde musíme nastavit jeden sériový kanál na *UNI*.

V programu byly použity následující jazyky: ST, LD, FBD, IL. Příklady vybraných částí:

Část programu v jazyce ST. Stará se o vykonání programu 1.

```

if prog1 then //část automatického řízení programu 1
  if not(K1) then if not(ok) then hlaska:=true; start:=false;end_if; else OK:= true; end_if;
  if OK then
    if faze1 then
      funkce3(cas1:=3000,cas2:=2000,dal1=>faze2);
    end_if;
    if faze2 then
      if not(sekce1) then speed:=false; else speed:=true; end_if;
      faze1:=false;
      funkce2(); //rychlí přesun na K1
    end_if;
  end_if;
end_if; //konec prog1

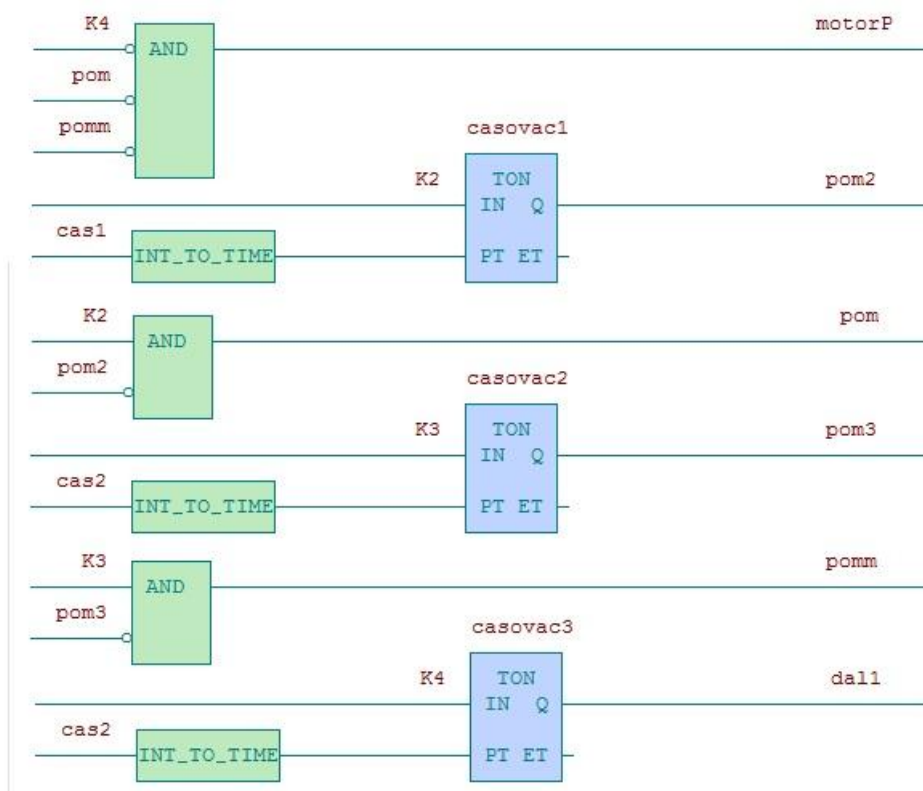
```

Část programu v jazyce LD. Stará se o přesun na čidlo K4. Mění rychlost podle části linky.

Stará se o přesun k čidlu K4 a mění rychlost podle části linky.

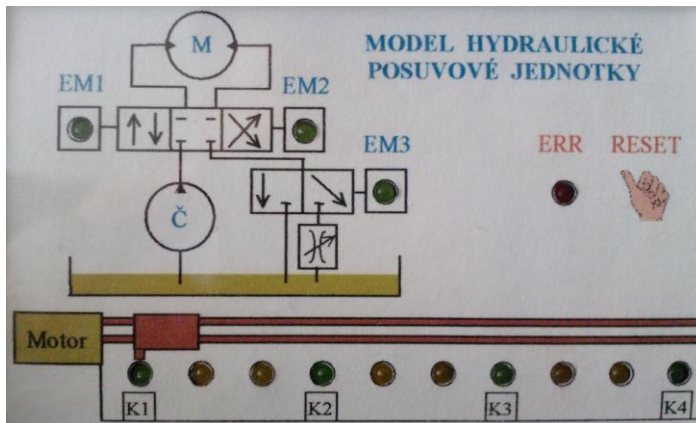


Část programu v jazyce FBD. Stará se o zastavení na jednotlivých čidlech.



5 MODEL HYDRAULICKÉ POSUVNÉ JEDNOTKY

Zadání: Máme vytvořit v prostředí Mosaic program k modelu hydraulické posuvné jednotky. Tato posuvná jednotka slouží k vrtání do plátů. U čidla K1 se výrobek vloží, čidlo K2 a K3 slouží pro vrtání dvěma druhy vrtáků. Čidlo K4 je pak posunutí plátu, abychom vrtali v určeném místě. Vytvořte několik tras pro různé vrtání jednotlivých výrobků, mezi nimiž se bude možno přepínat potenciometrem a zvolení potvrdíme tlačítkem (např. potenciometr A_A a tlačítko B_B). Můžeme zde použít propojení PLC tak, že z druhého PLC bude tlačítko sloužit jako centrální stop. Pokud toto tlačítko použijeme, taktéž se pošle přes GSM modem SMS zpráva údržbáři.



Obr. X6 Model automatické pračky



Obr. X7 Potenciometr a tlačítko

Vypracování: Pohyb suportu je simulován deseti diodami, z nichž 4 fungují jako snímače polohy K1 až K4. U tohoto modelu mohou nastat dva druhy funkčních chyb. První chyba je přejetí krajních snímačů K1 a K4. V tomto případě se rozsvítí chybová dioda a všechny polohové snímače. Je nutné resetovat model tlačítkem reset. Druhá chyba nastane při sepnutí posunů na obě strany. Chybová dioda začne blikat, dokud se neodstraní chybový stav. Pak bez restartu může pokračovat.

Zde si každý může libovolně vymyslet různé trasy posunů, čili vrtání na různých místech. Abychom mohli dvě PLC propojit je musíme nastavit v *Manažer projektu*->*Hw*->*Konfigurace HW* kanál ethernetu na UNI. A pro GSM musíme nastavit CH1 na UNI.

Struktura kanálů	rám / pozice	Režim kanálu	Adresa pro komunikaci	Komunikační rychlost	Prodleva odpovědi	Dopravní zpoždění	Detekce CTS	Předávání tokenu	Přenos s paritou
CP-1004	0 / 0								
CH									
CH1		uni <input checked="" type="checkbox"/>							
CH2		OFF							
CH3		OFF							
CH4		OFF							
ETH1			192.168.033.176						
ETH		PC, MDB							
ETH		PLC -off							
ETH		uni <input checked="" type="checkbox"/>							
ETH		BAC -off							

Pokud zapneme více jak dva kanály, musíme vkládat HW klíč. Tento klíč je v počítači u PLC s GSM modemem. Jakmile nastavíme kanál na UNI, je zde ikona pro vstup do nastavení univerzálního kanálu.

Přijímací zóna Délka zóny: <input type="text" value="4"/> Adresa zóny: <input type="checkbox"/> %R0 Přijímací zóna: <input type="text" value="ETH1_UNIQ_IN"/>		Vysílací zóna Délka zóny: <input type="text" value="4"/> Adresa zóny: <input type="checkbox"/> %R0 Vysílací zóna: <input type="text" value="ETH1_UNIQ_OUT"/>	
Typ protokolu <input checked="" type="radio"/> TCP master <input type="radio"/> TCP slave <input type="radio"/> UDP		Vzdálená IP adresa: <input type="text" value="10.5.18.77"/> Vzdálený port: <input type="text" value="61001"/> Místní port: <input type="text" value="61000"/>	

Takto nastavíme kanál na PLC s tlačítkem centrální stop. Protože potřebujeme, aby druhé PLC okamžitě reagovalo na stisknutí tlačítka, musíme nastavit PLC s tlačítkem central stop na master. Druhé PLC nastavíme na slave. U slave se nastaví zóny a místní port na 61001, další věci nepůjdou nastavit. V nastavení vstupů a výstupů si nastavíme alias, aby se nám dobře přistupovalo k datům.

UNI_ETH1_IN0 : TUNI_ETH1_IN0 STAT : TUniStat DSR : BOOL CTS : BOOL TRO : BOOL RCF : BOOL ROV : BOOL TRF : BOOL ARC : BOOL ERR : USINT NUMR : UINT DATA : ARRAY [C {PUBLIC}]	UNI_ETH1_IN0 UNI_ETH1_IN0~STAT UNI_ETH1_IN0~STAT~DSR UNI_ETH1_IN0~STAT~CTS UNI_ETH1_IN0~STAT~TRO UNI_ETH1_IN0~STAT~RCF UNI_ETH1_IN0~STAT~ROV UNI_ETH1_IN0~STAT~TRF UNI_ETH1_IN0~STAT~ARC UNI_ETH1_IN0~ERR UNI_ETH1_IN0~NUMR UNI_ETH1_IN0~DATA	ETH1_UNIQ_IN
UNI_ETH1_OUT0 : TUNI_ETH1_C CONT : TUniCont TRG : BOOL CLR : BOOL ACN : BOOL SIGN : TUniSign DTR : BOOL RTS : BOOL NUMT : UINT DATA : ARRAY [C {PUBLIC}]	UNI_ETH1_OUT0 UNI_ETH1_OUT0~CONT UNI_ETH1_OUT0~CONT~TRG UNI_ETH1_OUT0~CONT~CLR UNI_ETH1_OUT0~CONT~ACN UNI_ETH1_OUT0~SIGN UNI_ETH1_OUT0~SIGN~DTR UNI_ETH1_OUT0~SIGN~RTS UNI_ETH1_OUT0~NUMT UNI_ETH1_OUT0~DATA	ETH1_UNIQ_OUT

V obrázku modře vyznačené jsou přijímané data a červeně vyznačené odesílané data.

Pro odeslání tohoto pole dat musíme nastavit proměnnou TRG:=t1, TRF:=1. Když jsou data k poslání připravena, odesílají se nastavením ACN na hodnotu 1.

Příklad odesílání dat:

```
TRF:=1; TRG:=1; ETH1_UNIO_OUT[1]:=1;
```

```
If tlačítko then ACN:=1; else ACN:=0; end_if;
```

Příklad přijímání dat:

```
central_stop:= ETH1_UNIO_IN[1];
```

```
If central_stop=1 then zapnuto:=0; end_if;
```

Nastavení univerzálního kanálu pro komunikaci s GSM modemem:

Délka přijímací a odesílací zóny 360

Komunikační rychlost 9600

Formát dat 8b bez parity

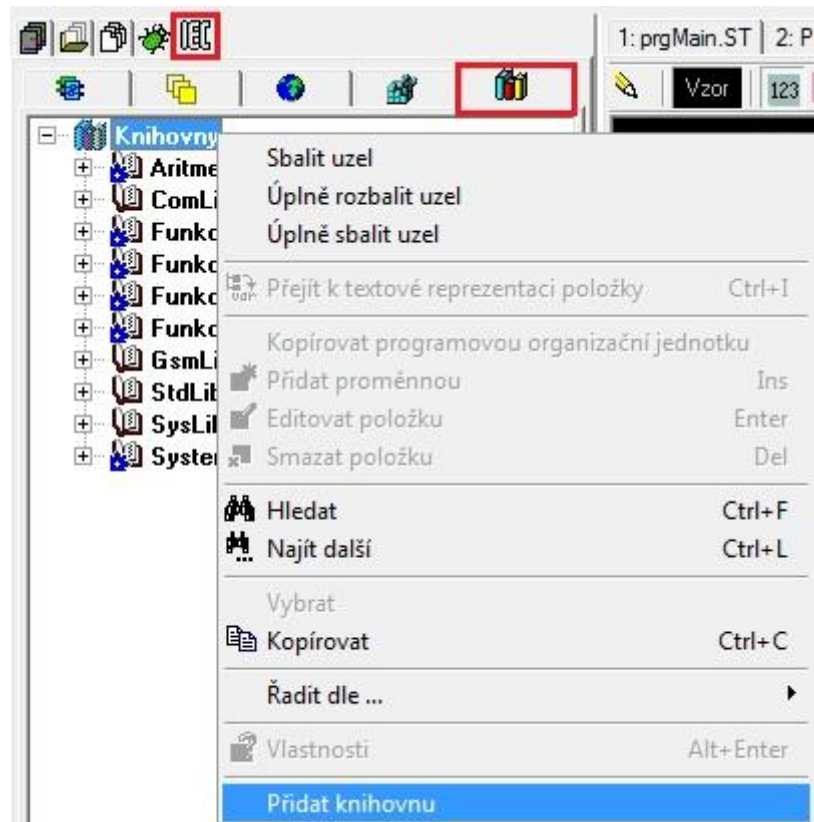
Maximální délka 360

Minimální doba klidu na lince mezi přijímanými zprávami 5

Minimální doba klidu na lince mezi vysílanými zprávami 40

Nyní musíme přidat do našeho projektu knihovnu, která obsahuje funkční blok SMS_Handler. Tato knihovna se jmenuje GsmLib plus ještě knihovny ComLib a SysLib.

Musíme jít do záložek IEC->Knihovny. Klikneme pravým na knihovny a zvolím přidat knihovnu.



Ze seznamu vybereme knihovnu. Nyní už máme k dispozici funkční blok SMS_Handler. Můžeme se podívat na jeho stavbu, když v knihovnách otevřeme GsmLib->funkční bloky.

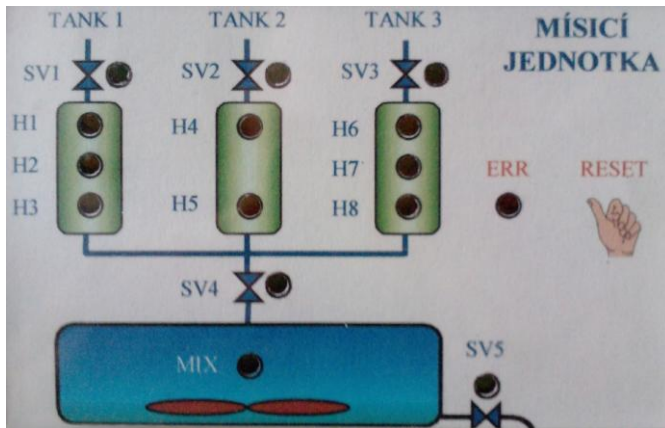


Vypracovaný program je možno nalézt na CD: Programy/ EDU-mod/Linka.zip;

Programy/ EDU-mod/CentralSTOP.zip

6 MODEL MÍSÍCÍ JEDNOTKY

Zadání: Vytvořme v prostředí Mosaic program k modelu míšící jednotky. V továrně míchají každý den jeden typ směsi. Mají stálé zakázky na 5 druhů směsí, které pravidelně vyrábí. Pomocí potenciometru se bude nastavovat druh směsi. Model můžeme propojit s GSM modemem, který bude posílat SMS například při přetečení nádob.



Obr. X8 Model automatické pračky



Obr. X9 Potenciometr a tlačítko

Vypracování: Je to aktivní modul, složený ze tří plnicích nádob a jedné míšící nádoby. Je zde pět ventilů, které slouží pro napouštění nebo vypouštění nádob. I u tohoto modelu může nastat chyba v podobě přetečení kterékoliv z nádob. Rozsvítí se chybová dioda a tento stav se dá vyřešit jedině tlačítkem reset.

Nastavení GSM modemu je popsáno v odstavci 5.

Vypracovaný program je možno nalézt na CD: Programy/ EDU-mod/Mix.zip;

V programu byly použity následující jazyky: ST, LD, FBD, IL. Příklady vybraných částí:

Část programu v jazyce ST. Podle objemu směsi se nastaví délka vypouštění nádrže.

```

if faze3 then //podle objemu směsi se nastaví délky vypouštění
  case objem of
    20 : casovac2 (IN := pom3, PT :=T#2s, Q =>doba2);
    30 : casovac2 (IN := pom3, PT :=T#2s, Q =>doba2);
    62 : casovac2 (IN := pom3, PT :=T#4s, Q =>doba2);
    94 : casovac2 (IN := pom3, PT :=T#6s, Q =>doba2);
    104: casovac2 (IN := pom3, PT :=T#6s, Q =>doba2);
    126: casovac2 (IN := pom3, PT :=T#8s, Q =>doba2);
    136: casovac2 (IN := pom3, PT :=T#9s, Q =>doba2);
    168: casovac2 (IN := pom3, PT :=T#10s, Q =>doba2);
    178: casovac2 (IN := pom3, PT :=T#10s, Q =>doba2);
    210: casovac2 (IN := pom3, PT :=T#12s, Q =>doba2);
    252: casovac2 (IN := pom3, PT :=T#15s, Q =>doba2);
  end_case;
  pom3:=true;
  if doba2 then sv5:=false; faze3:=false; start:=false; end_if;
end_if;

```

Část programu v jazyce LD. Stará se o vypouštění nádob s přísadami, a také o míchání.

Až se vypustin nádoby pod spodní čidla zapne čítač který udržuje vypouštění po dobu 2 sekund pro úplné vypuštění. Zároveň se stará o míchání.

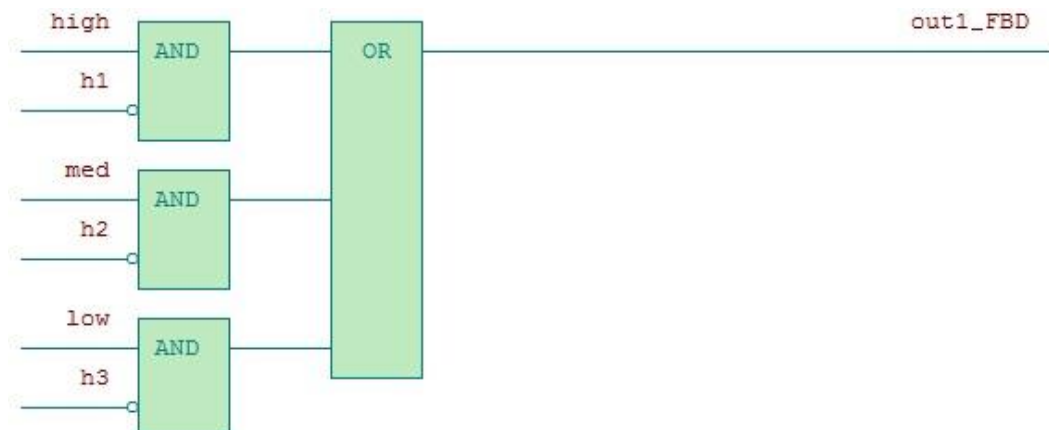
0001



0002



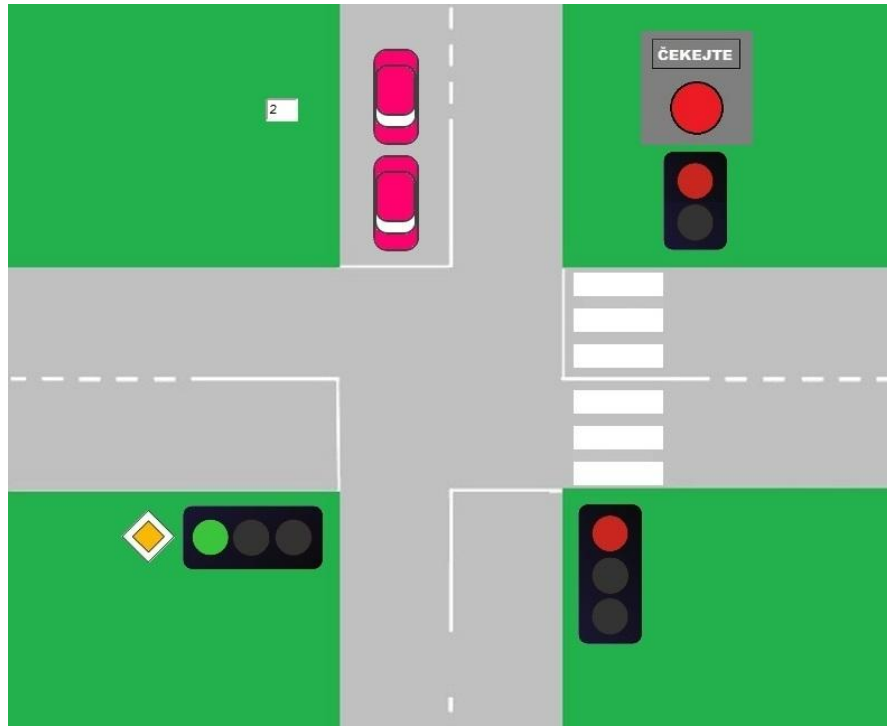
Část programu v jazyce FBD. Zjistí kolik přísady má být napuštěno a tolik i napustí.



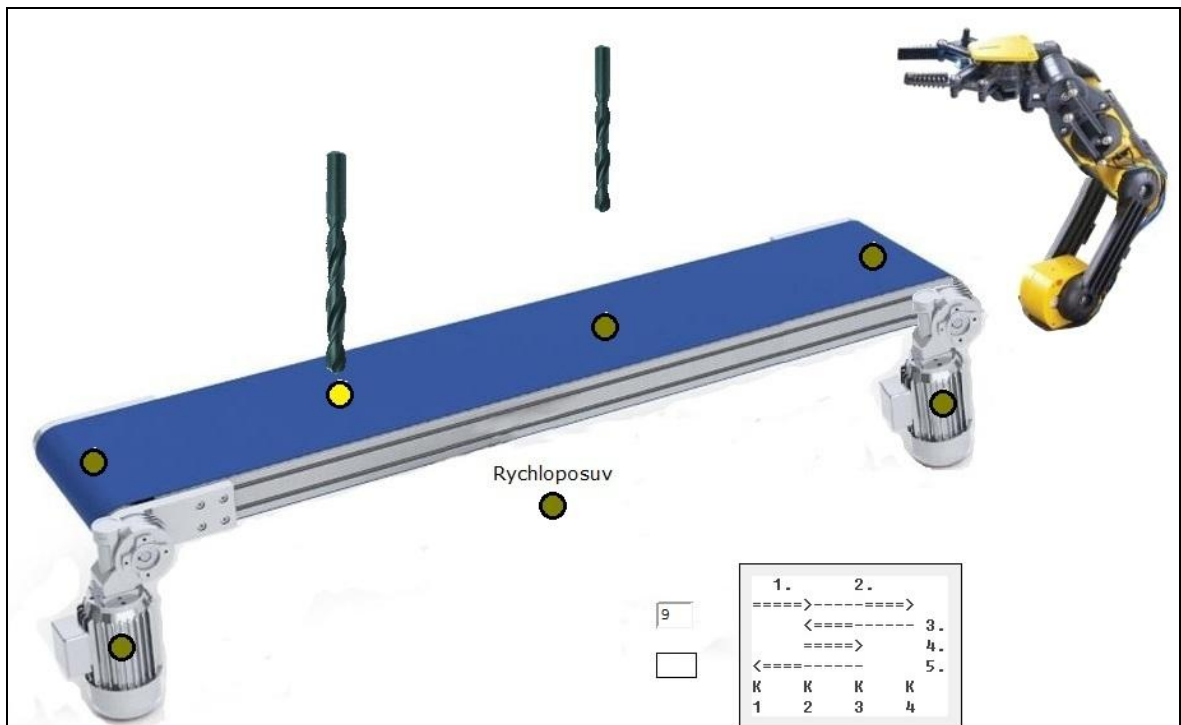
7 VIZUALIZACE MODELŮ POMOCÍ WEBMAKER

Pro každý program můžeme vytvořit vizualizaci za použití nástroje WebMaker. Vizualizace programů mohou vypadat následovně:

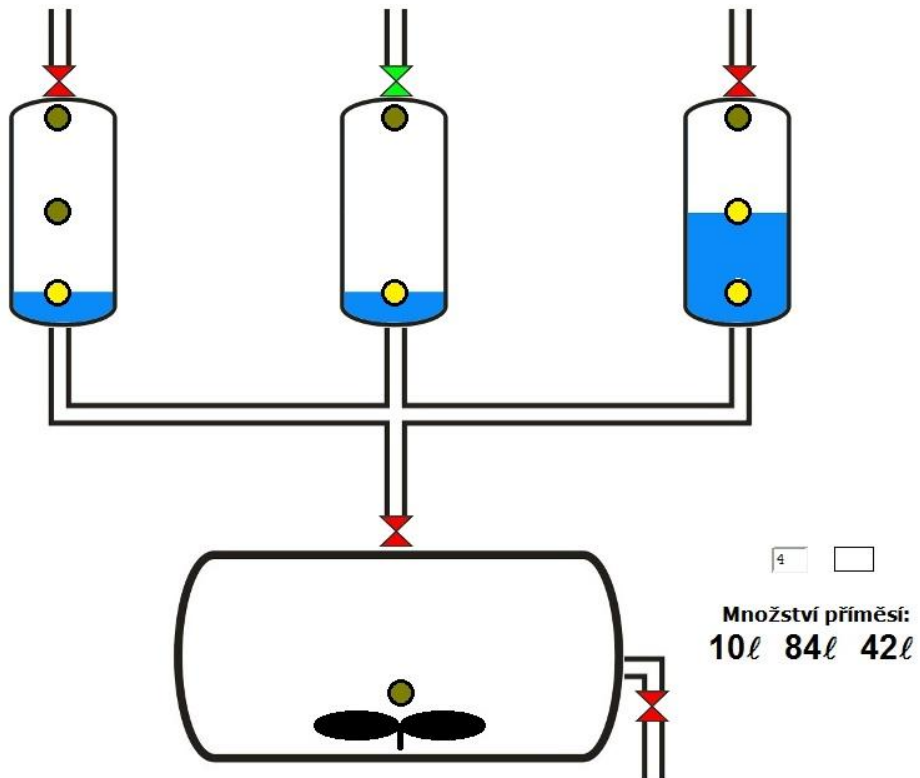
Ukázka vizualizace pomocí nástroje WebMaker pro model křižovatky:



Ukázka vizualizace pomocí nástroje WebMaker pro model hydraulické posuvné jednotky:



Ukázka vizualizace pomocí nástroje WebMaker pro model mísicí jednotky:



Ukázka vizualizace pomocí nástroje WebMaker pro model automatické pračky:



8 DALŠÍ MOŽNOSTI VYUŽITÍ KNIHOVNY COMLIB

Součástí knihovny ComLib jsou funkce, které pracují s IP adresou. Například funkce GetIPAddress vrátí IP adresu, IP masku a adresu brány pro zadaný ethernet kanál. Získané hodnoty jsou uloženy v EthAdr, která má strukturu typu TLocalEthAdr. Tato struktura má tři položky (IP adresu - **IP**, IP masku - **IM**, adresu brány - **GW**). Každá s položek se skládá ze čtyř polí, ve kterých jsou uloženy části adres. Tato funkce nám vrátí informace o místním PLC. Pro zjištění informací o vzdáleném PLC musíme použít funkci GetRemoteIPAddress. Zde se používá struktura TRemoteEthAdr. Tato struktura má také tři položky (IP adresa - **remoteIP**, číslo vzdáleného portu - **remotePort**, číslo místního portu - **localPort**). I zde se IP adresa skládá ze čtyř polí, ale porty jsou typu UINT.

Příklad získání IP adresy vzdáleného PLC:

VAR

eth_adr : TRemoteEthAdr;

tmp : BOOL;

END_VAR

tmp := GetRemoteIPAddress(chanCode := ETH1_uni0, EthAdr := eth_adr);

chanCode – zde se zadává ethernetový kanál pomocí kterého jsou PLC propojena.

Získání poslední části IP adresy: cast4 := eth_adr.remoteIP[3]

Vypracovaný program je možno nalézt na CD:

Programy/ Komunikace/TCP_2na1_slave.zip;