

# Programovatelné automaty KOYO<sup>TM</sup> CLICK



**Vydala firma TECON spol. s.r.o.**

MM©2015

## **Programovatelné automaty KOYO™ CLICK**

Autor: Ing. Michal Martin

Copyright © TECON spol. s.r.o., Vrchlabí, Česká republika

Tato publikace prošla jen částečnou jazykovou korekturou. Tato publikace vznikla na základě informací dostupných v době tisku. Veškeré změny jsou vyhrazeny. Všechny ochranné známky uvedené v této knize jsou majetkem jejich vlastníků.

**Žádná část této publikace nesmí být publikována a šířena žádným způsobem a v žádné podobě bez výslovného svolení autora a firmy TECON spol. s.r.o.**

### **TECON spol. s.r.o.**

Komenského 63

Pivovar-historická budova

543 01 Vrchlabí

Tel.: 499 429 100, 499 429 117

Fax: 499 422 508

e-mail: [info@TECON.cz](mailto:info@TECON.cz)

Web: [www.TECON.cz](http://www.TECON.cz)

## Obsah

Začínáme .....	4
CPU moduly .....	5
CPU moduly s Ethernetem .....	5
CPU moduly bez Ethernetu .....	7
CPU moduly CLICK PLUS .....	8
Rozšiřující moduly .....	9
Instalace a připojování.....	10
Instalace programu CLICK Software .....	11
Připojení programovatelného automatu .....	12
Zapojení vstupů a výstupů.....	12
Připojení napájení .....	13
Připojení k PC.....	14
Programování .....	15
Propojka.....	19
Zpoždovací relé.....	24
Čítač sepnutí .....	25
Simulace skladu.....	27
Podprogramy a přerušení.....	33
Podprogram .....	33
Přerušení .....	35
Nastavení rychlých vstupů .....	38
Rozměry a montáž systému CLICK.....	40
Přídavné moduly systému CLICK.....	42
Binární vstupy a výstupy .....	42
Analogové vstupy a výstupy .....	43
Poznámky .....	44

## **Začínáme**

Firma TECON s.r.o je distributorem průmyslových automatů CLICK vyráběných firmou Koyo Electronics. Systémy této firmy jsou jedny z nejrychlejších a nejmenších programovatelných automatů PLC. Přitom jsou tyto systémy jedny z nejekonomičtějších na českém trhu. Firma Koyo je japonská firma a českou distribucí PLC CLICK se zabývá naše firma TECON s.r.o.

Programovatelné automaty řady CLICK jsou kompaktní PLC s možností rozšíření zásuvnými moduly, které jsou určeny pro řízení technologií v nejrůznějších oblastech průmyslu (vodárny, výměňkové stanice, kotelny, jednoúčelové stroje, ...), zemědělství apod.

Tato uživatelská příručka vám ukáže, jakým způsobem lze instalovat, zapojovat, programovat a udržovat automaty řady CLICK. Příručka obsahuje důležité informace pro ty, kteří budou automaty instalovat a pro tvůrce programového vybavení.

**Poznámka:** Příručka v žádném případě nenahrazuje originální uživatelský manuál.

## CPU moduly

CPU moduly lze rozdělit do dvou skupin a to s Ethernetem a bez něj. CPU s Ethernetem je provedeno v řadě Basic a Standard. CPU bez Ethernetu je provedeno ve třech řadách: Basic, Standard a Analog. Dohromady dostáváme, až 15 různých CPU modulů.

### CPU moduly s Ethernetem

CPU moduly s Ethernetem umožňují přehrávat program v režimu **RUN (RUN TIME EDIT)**. Dále disponují reálným časem, který lze měnit. CPU moduly s Ethernet jsou 3-10x rychlejší než CPU bez Ethernetu.

#### Ethernet Basic CPU



CLICK Basic CPU Moduly		
Označení	Vstup (8x)	Výstup (6x)
C0-10DD1E-D	DC(24 VDC, sink/source) (4 vstupy do 100 kHz)	DC(0.1 A, 5-24 VDC, Sink)
C0-10DD2E-D		DC(0.1 A, 24 VDC, Source)
C0-10DRE-D		Relay(1 A, 6-27VDC/6-240VAC)

#### Vlastnosti:

- 8x digitální vstup (4x do 100kHz)
- 6x digitální výstup
- 1x Ethernet 10/100 Mbps
- 1x RS-232 komunikační port
- Reálný čas

#### Ethernet Standard CPU



CLICK Standard CPU Moduly		
Označení	Vstup (8x)	Výstup (6x)
C0-11DD1E-D	DC(24 VDC, sink/source) (8 vstupů do 100 kHz)	DC(0.1 A, 5-24 VDC, Sink)
C0-11DD2E-D		DC(0.1 A, 24 VDC, Source)
C0-11DRE-D		Relay(1 A, 6-27VDC/6-240VAC)

#### Vlastnosti:

- 8x digitální vstup (8x do 100kHz)
- 6x digitální výstup
- 1x Ethernet 10/100 Mbps
- 1x RS-232 komunikační port
- 1x RS-485 komunikační port
- Reálný čas

## Ethernet Analog CPU



CLICK Analog CPU Moduly			
Označení	Vstup (4x)	Výstup (4x)	Analog I/O
C0-12DD1E-D	DC(24 VDC, sink/source) (4 vstupy do 100 kHz)	DC(0.1 A, 5-24 VDC, Sink)	2 kanály In/ 2 kanály Out(0-5 VDC) a (4-20 mA)
C0-12DD2E-D		DC(0.1 A, 24 VDC, Source)	
C0-12DRE-D		Relay(1 A, 6-27VDC/6-240VAC)	
C0-12DD1E-1-D		DC(0.1 A, 5-27 VDC, Sink)	4 kanály In (0-20mA)/ 2 kanály Out(4-20mA)
C0-12DD2E-1-D		DC(0.1 A, 24 VDC, Source)	
C0-12DRE-1-D		Relay(1 A, 6-27VDC/6-240VAC)	
C0-12DD1E-2-D		DC(0.1 A, 5-27 VDC, Sink)	4 kanály In (0-10V)/ 2 kanály Out(0-10V)
C0-12DD2E-2-D		DC(0.1 A, 24 VDC, Source)	
C0-12DRE-2-D		Relay(1 A, 6-27VDC/6-240VAC)	

### Vlastnosti:

- 4x digitální vstup (4x do 100kHz)
- 4x digitální výstup
- 2x/4x analogový vstup
- 2x analogový výstup
- 1x Ethernet 10/100 Mbps
- 1x RS-232 komunikační port
- 1x RS-485 komunikační port
- Záložní baterie (prodávána zvlášť)
- Reálný čas

## CPU moduly bez Ethernetu

### Basic CPU



CLICK Basic CPU Moduly		
Označení	Vstup (8x)	Výstup (6x)
C0-00DD1-D	DC(24 VDC, sink/source)	DC(0.1 A, 5-24 VDC, Sink)
C0-00DD2-D		DC(0.1 A, 24 VDC, Source)
C0-00DR-D		Relay(1 A, 6-27VDC/6-240VAC)

#### Vlastnosti:

- 8x digitální vstup
- 6x digitální výstup
- 2x RS-232 komunikační port

### Standard CPU



CLICK Standard CPU Moduly		
Označení	Vstup (8x)	Výstup (6x)
C0-01DD1-D	DC(24 VDC, sink/source)	DC(0.1 A, 5-24 VDC, Sink)
C0-01DD2-D		DC(0.1 A, 24 VDC, Source)
C0-01DR-D		Relay(1 A, 6-27VDC/6-240VAC)

#### Vlastnosti:

- 8x digitální vstup
- 6x digitální výstup
- 2x RS-232 komunikační port
- 1x RS-485 komunikační port
- Záložní baterie (prodávána zvlášť)
- Reálný čas

### Analog CPU



CLICK Analog CPU Moduly			
Označení	Vstup (4x)	Výstup (4x)	Analog I/O
C0-02DD1-D	DC(24 VDC, sink/source)	DC(0.1 A, 5-24 VDC, Sink)	2 kanály In/ 2 kanály Out(0-5 VDC) a (4-20 mA)
C0-02DD2-D		DC(0.1 A, 24 VDC, Source)	
C0-02DR-D		Relay(1 A, 6-27VDC/6-240VAC)	

#### Vlastnosti:

- 4x digitální vstup
- 4x digitální výstup
- 2x analogový vstup
- 2x analogový výstup
- 2x RS-232 komunikační port
- 1x RS-485 komunikační port
- Záložní baterie (prodávána zvlášť)
- Reálný čas

## CPU moduly CLICK PLUS

CPU moduly CLICK PLUS umožňují přehrávat program v režimu **RUN (RUN TIME EDIT)**. Prostřednictvím Ethernet kabel nebo Wi-Fi. Dále disponují reálným časem, který lze měnit. CPU moduly CLICK PLUS jsou 3-10x rychlejší než CPU bez Ethernetu.

**C2-01CPU**



**C2-02CPU**



**C2-03CPU**



CLICK Basic CPU Moduly		
Označení	Komunikace	Vlastnosti
<b>C2-01CPU</b>	Ethernet / RS232 / USB micro	Realný čas
<b>C2-02CPU</b>	WiFi / USB micro / Bluetooth	Realný čas
<b>C2-03CPU</b>	Ethernet / RS232 / RS485 / USB micro / WiFi / Bluetooth	Realný čas

Jedná se o CPU moduly s **rozšiřujícím slotem**. Lze si tak vytvořit CPU požadovaných vlastností I/O. Tyto CPU lze rozšířit o následující moduly.

CLICK Analog CPU Moduly			
Označení	Binární Vstupy	Binární Výstupy	Analog I/O
<b>C2-14D1</b>	8x DC(24VDC, sink/source) 100kHz	6x (0.1 A, 5-27 VDC, Sink)	-
<b>C2-14D2</b>		6x (0.1 A, 5-27 VDC, Source)	
<b>C2-14DR</b>		4x Relay(1 A, 6-27VDC/6-240VAC)	
<b>C2-08D1-4VC</b>	4x DC(24VDC, sink/source) 100kHz	4x (0.1 A, 5-27 VDC, Sink)	2 kanály In (0-5VDC/4-20mA) 2 kanály Out (0-5VDC/4-20 mA)
<b>C2-08D2-4VC</b>		4x (0.1 A, 5-27 VDC, Source)	2 kanály In (0-5VDC/4-20mA) 2 kanály Out (0-5VDC/4-20 mA)
<b>C2-08DR-4VC</b>		4x Relay(1 A, 6-27VDC/ 6-240VAC)	2 kanály In (0-5VDC/4-20mA) 2 kanály Out (0-5VDC/4-20 mA)
<b>C2-08D1-6C</b>		4x (0.1 A, 5-27 VDC, Sink)	4 kanály In (0-20mA)/ 2 kanály Out(4-20mA)
<b>C2-08D2-6C</b>		4x (0.1 A, 5-27 VDC, Source)	4 kanály In (0-20mA)/ 2 kanály Out(4-20mA)
<b>C2-08DR-6C</b>		4x Relay(1 A, 6-27VDC/ 6-240VAC)	4 kanály In (0-20mA)/ 2 kanály Out(4-20mA)
<b>C2-08D1-6V</b>		4x (0.1 A, 5-27 VDC, Sink)	4 kanály In (0-10V)/ 2 kanály Out(0-10V)
<b>C2-08D2-6V</b>		4x (0.1 A, 5-27 VDC, Source)	4 kanály In (0-10V)/ 2 kanály Out(0-10V)
<b>C2-08DR-6V</b>		4x Relay(1 A, 6-27VDC/ 6-240VAC)	4 kanály In (0-10V)/ 2 kanály Out(0-10V)



## **Rozšiřující moduly**

CPU moduly je možné rozšířit pomocí přídavných modulů (**až osmi**). K dispozici jsou přídavné moduly pro digitální a analogové vstupy a výstupy nebo teplotní moduly. Úplný přehled přídavných modulů je na konci příručky.

## Instalace a připojování

### První experimenty

S pomocí této příručky se můžete pustit do prvních praktických pokusů s programovatelným automatem CLICK. Jestliže budete postupovat dle návodu krok za krokem, velice rychle si osvojíte základní principy zapojování a programování automatů CLICK.

### Základní potřeby

K tomu, abyste mohli začít experimentovat, je potřeba mít k ruce několik věcí:

- Programovatelný automat CLICK
- Odpovídající napájecí zdroj
- Napájecí kabel
- Čtyři přepínače pro simulaci vstupů
- Propojovací vodiče
- Malý křížový šroubovák
- Tuto příručku
- Programovací software *CLICK Software*, který je zdarma
- Programovací kabel na propojení PLC a PC

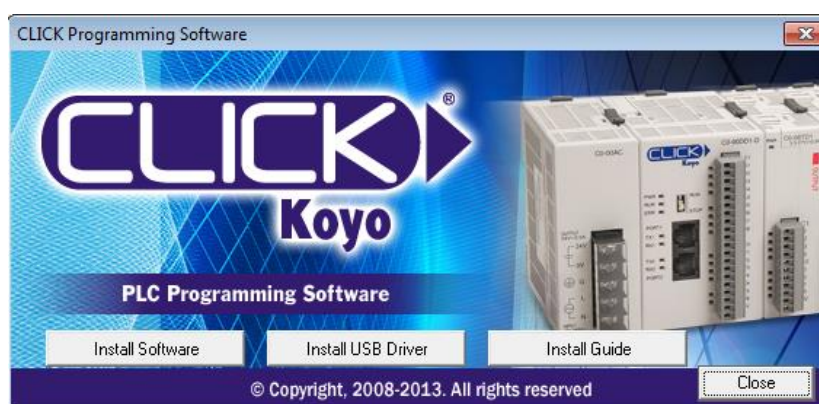
## Instalace programu CLICK Software

### Spusťte Windows

Jestliže jste tak ještě neučinili, spusťte operační systém Microsoft Windows. Jestliže jste doposud nepracovali se systémem Windows, poraďte se s manuálem Windows jak nakonfigurovat systém na optimální výkon a jak spouštět programy. Programovací software CLICK je plně funkční pod systémem Windows XP/Vista/Windows7.

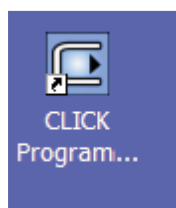
### Stáhněte si program CLICK

Programovací software CLICK je volně šiřitelný a je tak volně ke stažení. Lze ho stáhnout na našich stránkách [www.tecon.cz](http://www.tecon.cz) . Stažený soubor bude ve formátu .ZIP. Rozbalte tento soubor a spusťte install.exe . Po spuštění se zobrazí následující okno.



Pro instalaci softwaru klikněte na „Install Software“, pro bližší informace o instalaci klikněte na „Install Guide“.

Po kliknutí na „Install Software“ se spustí instalace softwaru. Instalace je totožná s jinými programy, tedy zvolíte složku pro umístění softwaru a instalaci dokončíte. Poté nainstalujte USB driver kliknutím na „Install USB drive“. Opět stačí klikat na „NEXT“.



Na ploše se vám vytvořila ikona programovacího prostředí CLICK. Dvojím kliknutím program spustíte.

Dříve než tak učiníte, si ukážeme, jak programovatelný automat CLICK připojíme k napájení a konfiguračním kabelem k osobnímu počítači PC s konfiguračním softwarem CLICK.

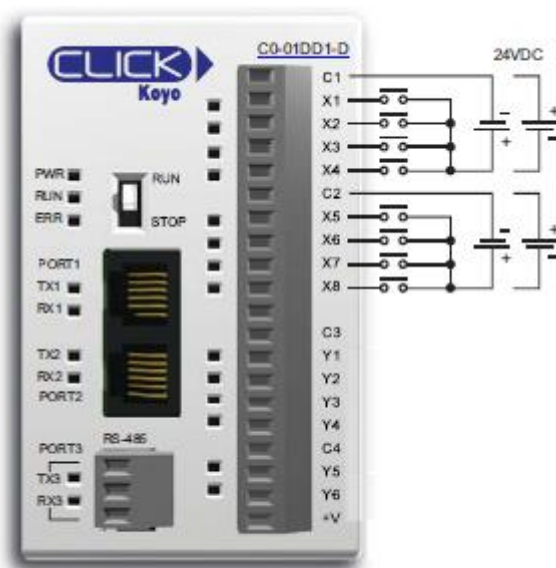
## Připojení programovatelného automatu

### Zapojení PLC

Připravte si programovatelný automat CLICK a ostatní součásti uvedené na začátku této příručky.

### Zapojení vstupů a výstupů

Abychom mohli smysluplně zkusit nějakou jednoduchou aplikaci, potřebujeme, aby náš program reagoval na nějaké podněty zvenčí. K tomu nám nejlépe poslouží simulace binárních vstupů pomocí přepínačů připojených dle následujícího obrázku. Budete tak moci simulovat chování technologie, tj. spínání koncových spínačů, průchod výrobku apod. Kompaktní automat CLICK obsahuje binární výstupy indikované LED diodami. Výstupy nám v dalším postupu budou ukazovat reakci programovatelného automatu, resp. programu v něm na události, které budeme simulovat pomocí binárních vstupů.



## Připojení napájení

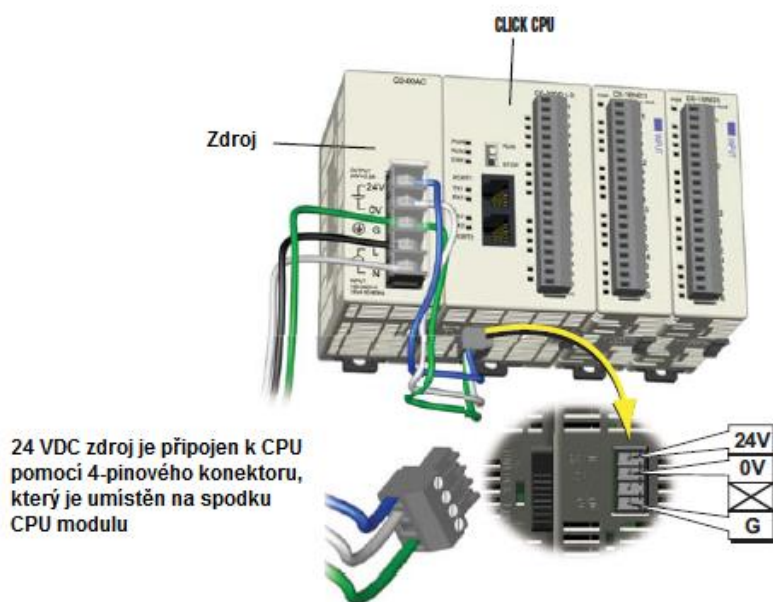
Pro připojení napájení do programovatelného automatu CLICK je zapotřebí použít zdroj s výstupním napětím 24 VDC.

Zdroj C0-01AC



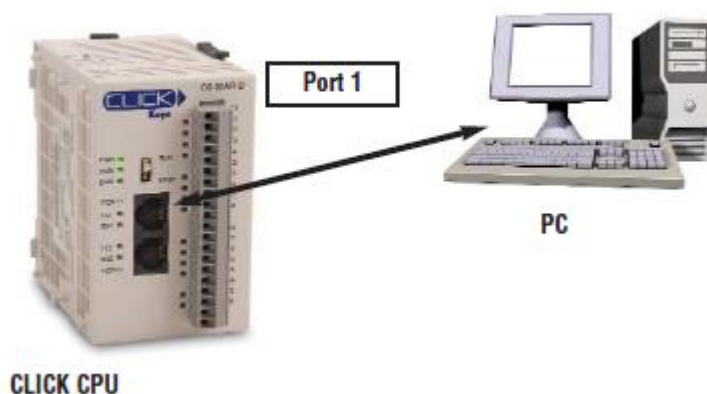
CLICK 24VDC Napěťové zdroje	
Označení	Výstupní proud
C0-00AC	0.5 A
C0-01AC	1.3 A

Připojení k CPU modulu je pak realizováno pomocí 4pinového konektoru, který je umístěn na spodní straně CPU modulu (viz. Obrázek).

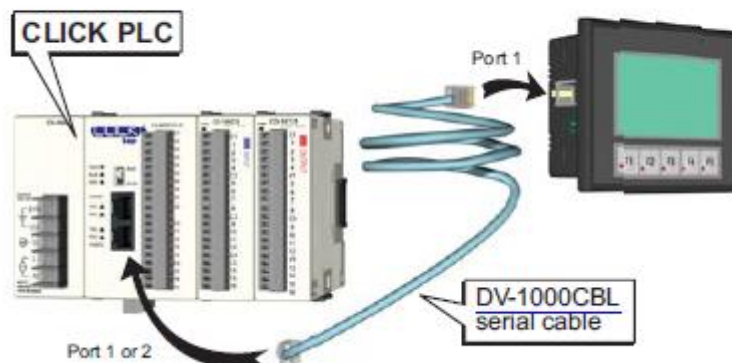


## Připojení k PC

Připojení programovatelného automatu CLICK k PC je realizováno prostřednictvím klasického RS-232 kabelu. Připojení k **Portu 1** je na základě Modbus RTU protokolu pouze ve SLAVE módu. Zařízení CLICK tak pracuje jako SLAVE. Port 2 resp. Port 3 (mimo Basic CPU) pracují s protokolem Modbus RTU nebo ASCII.

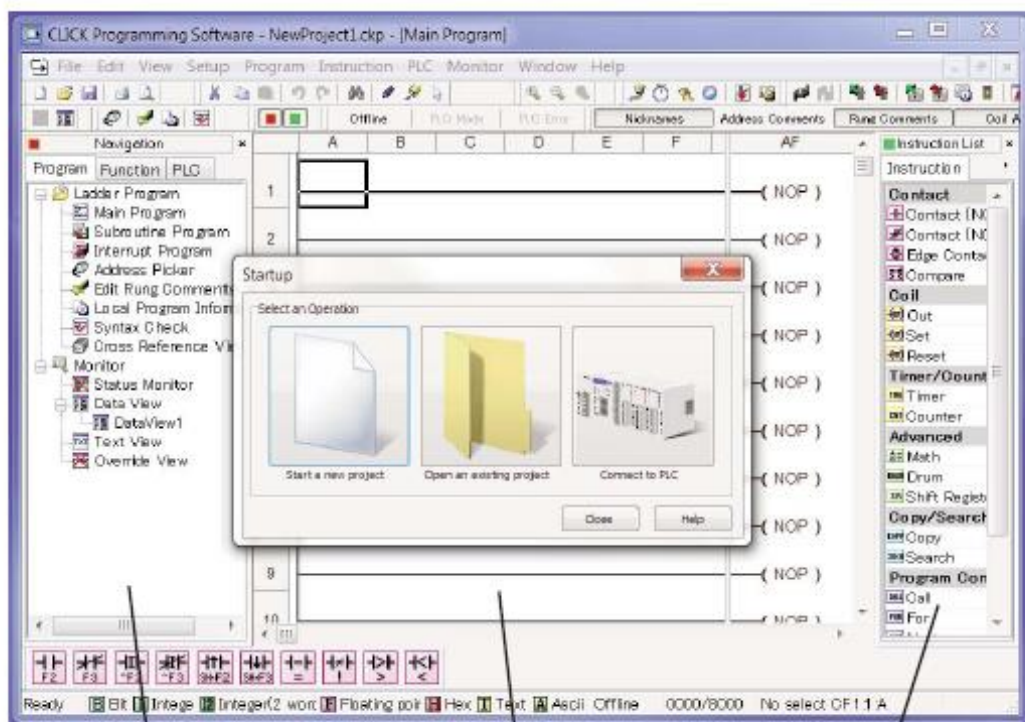


Programovatelný automat CLICK lze dále připojit například k operátorským panelům. Lze použít jak **port 1**, tak **port 2**. Připojení je opět realizováno RS-232 kabelem.



## Programování

Zapojíme-li CPU modul ke zdroji a připojíme-li programovací kabel RS-232, můžeme spustit programovací software CLICK. Dvojným poklepáním na ikonu na ploše spustíte programovací software. Po spuštění programu se objeví hlavní okno programu (viz obrázek).



Navigační okno

Okno liniového programu

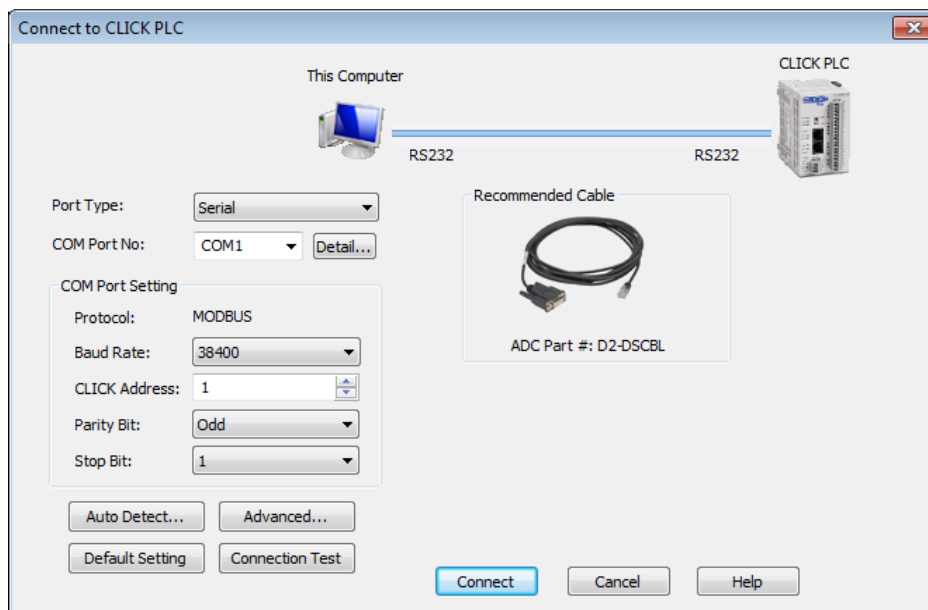
Instrukční okno

Uprostřed obrázku je okno „Startup“. Toto okno se zobrazí po spuštění programovacího softwaru automaticky. V tomto okně lze zvolit ze tří možností. Tedy „**Start a new project**“ pro vytvoření nového projektu. Dále „**Open an existing project**“ pro otevření uložených projektů, nebo poslední volba „**Connect to PLC**“, tato volba se připojí přímo k PLC.

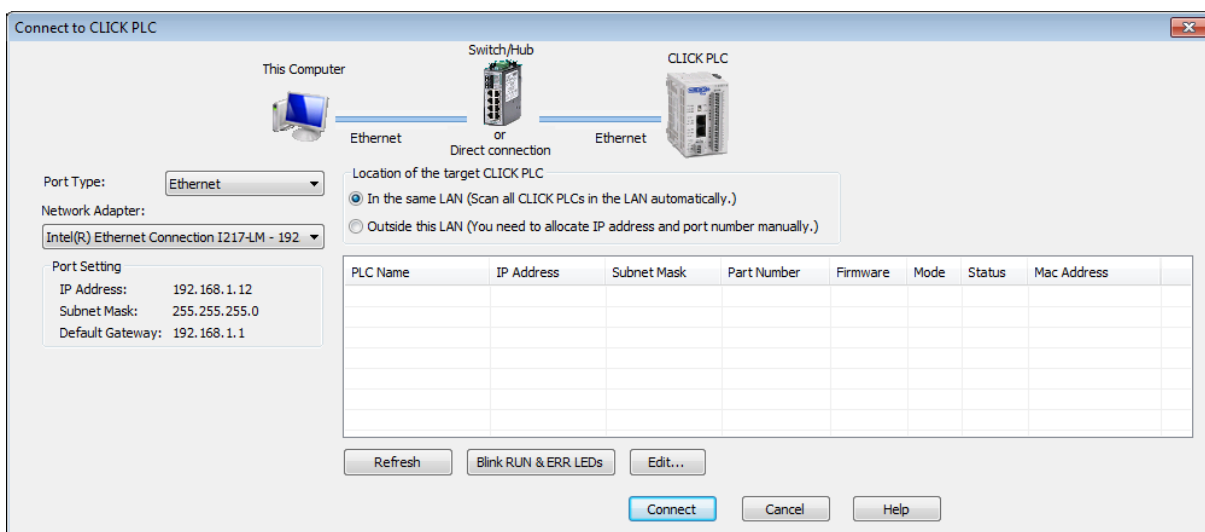
Pokud zvolíme poslední volbu, tedy „**Connect to PLC**“ zobrazí se nové okno, ve kterém se nastavuje připojení k CPU modulu. Prvním nastavením bude typ portu „**Port Type**“.

Nejprve si ukážeme připojení „**Serial**“. Nastavuje se zde PC COM port, dále se nastavují samotné vlastnosti připojení tedy Baud Rate, Address, Parity Bit, Stop Bit. Dále je možné využít volbu Auto Detect. Tato funkce projde veškeré možné kombinace těchto vlastností, a pokud se někde nachází CPU modul, tak ho nalezne a nastaví u těchto vlastností danou kombinaci. Pro programovatelné automaty CLICK je kombinace vlastností připojení dána:

**Baud Rate:** 38400  
**Address:** 1 - adresa CPU modulu, defaultně nastavena 1  
**Parity Bit:** Odd  
**Stop Bit:** 1



Dále je zde možnost vybrat připojení pomocí Ethernet portu (Pouze CPU moduly s Ethernetem). Pro připojení k CPU modulu je možno vybrat ze dvou způsobů. „**In the same LAN**“ nebo „**Outside this LAN**“.



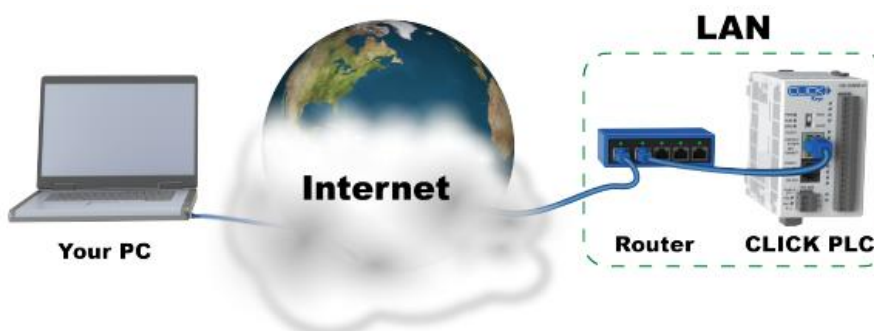
Pro připojení k CPU modulu ve stejné síti není zapotřebí nic nastavovat. V levé části zvolíme „**Network Adapter**“, který je připojen na lokální síť (jedná se o síťovou kartu). V tabulce se zobrazí automaticky všechny CPU moduly na této síti.

Pro připojení k CPU modulu mimo vaši síť existují dva způsoby. Pomocí VPN nebo Port Forwarding na routeru.



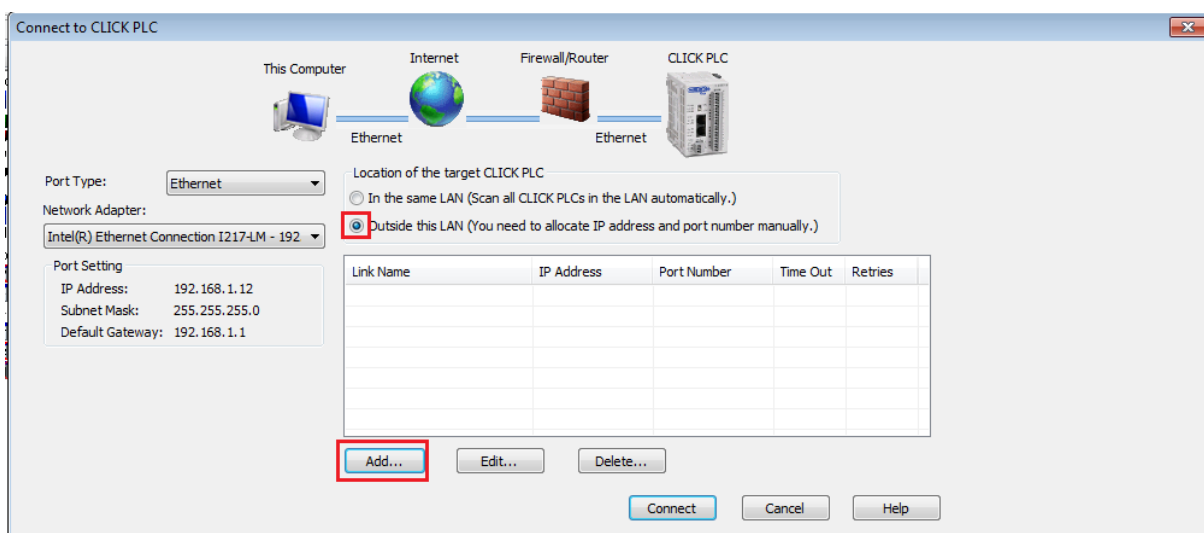


U připojení pomocí VPN (**Doporučeno**), je připojení zprostředkováno routrem nebo jiným zařízením s funkcí VPN.

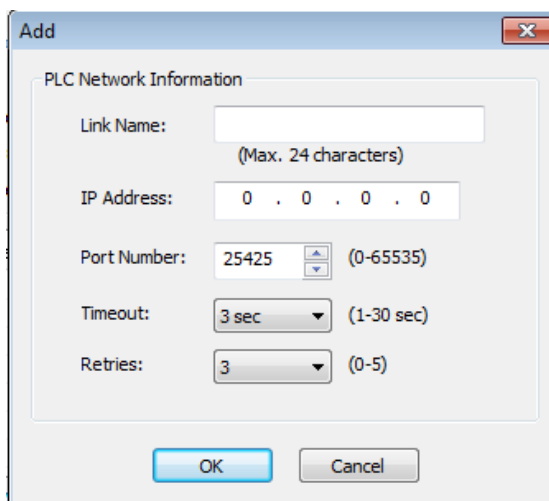


U připojení Port Forwarding je nutné mít router, který podporuje tuto funkci.

Poté se pro oba typy připojení nastaví v okně **Connect to Click PLC** připojení přes Ethernet „**Outside this LAN**“.

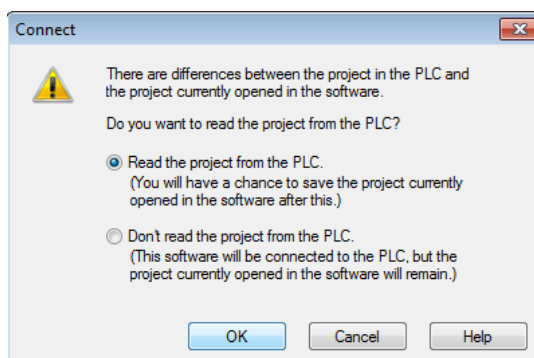


Klikneme na „**Add**“ a vyplníme následující okno.



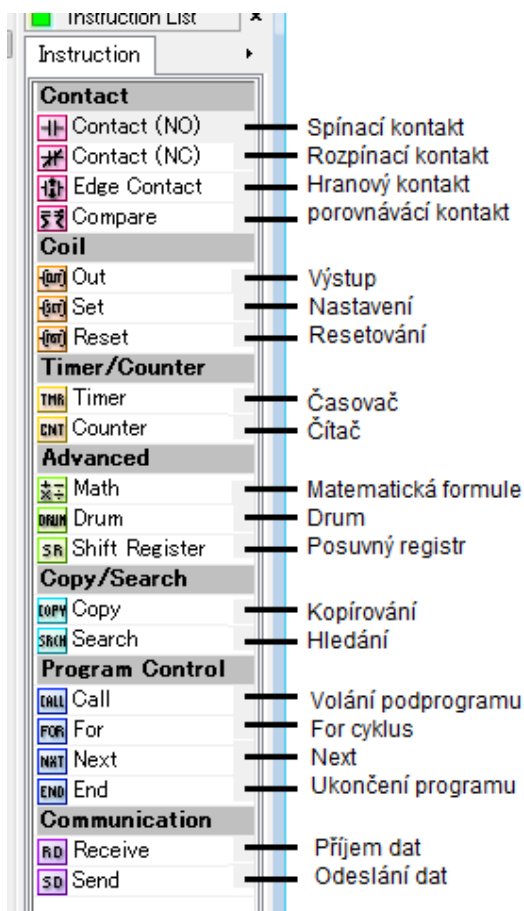
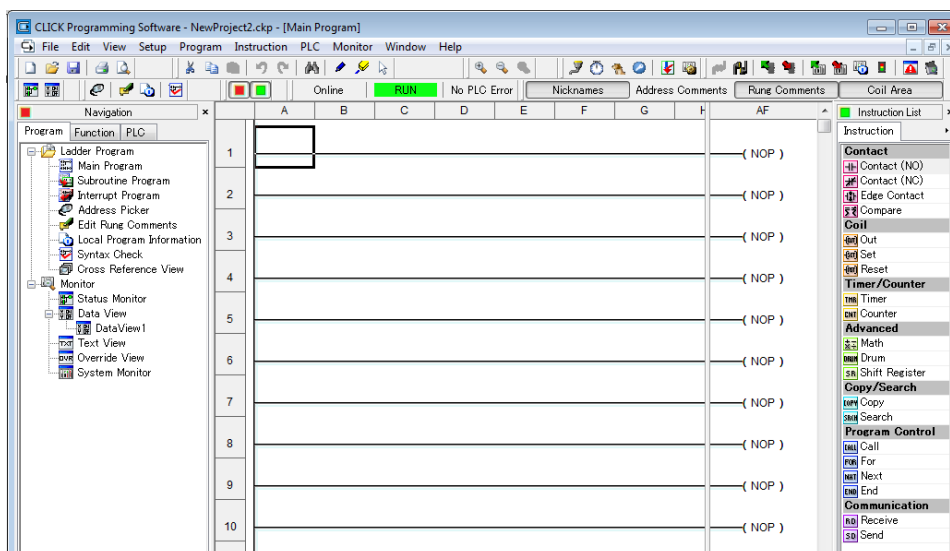
Pokud se připojujete přes VPN zadejte IP adresu CPU modulu ve vzdálené síti. Ujistěte se, že číslo portu je 25425. Pokud se připojujete přes Port Forwarding zadejte IP adresu routeru.

Po nastavení příslušných hodnot nezbývá než kliknout na tlačítko OK. Poté co kliknete na OK se naváže spojení s CPU modulem a zobrazí se následující okno.



V tomto okně je možnost zvolit načtení projektu z CPU modulu nebo nikoli. Při načtení projektu z modulu se načte aktuální projekt, který je v CPU modulu uložen.

Pokud jste nezvolili načtení z CPU modulu, zobrazí se prázdný projekt a hlavní okno tak bude vypadat následovně.

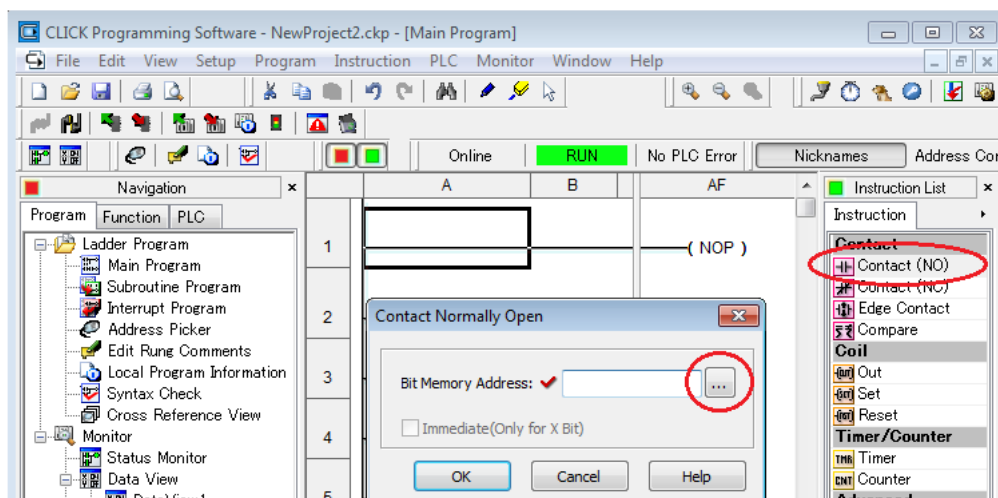


Nyní máte možnost napsat svůj první program. Program se skládá z jednotlivých linií. Každá linie má levou a pravou část. V levé části jsou umístěné instrukce, které tvoří nějaké podmínky a v pravé části jsou pak akce, které se mají podle dané podmínky vykonat. Pro psaní programu resp. podmínek a akcí jsou k dispozici před vytvořené instrukce (viz obrázek vlevo). Tyto instrukce nám umožňují napsat požadovaný program.

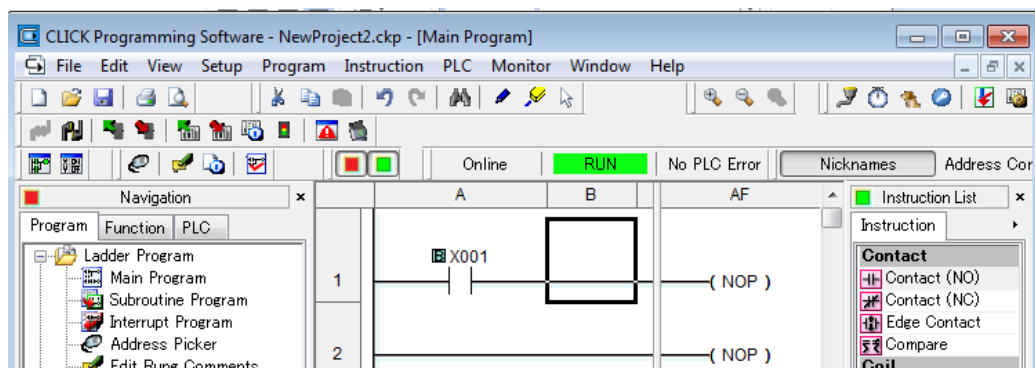
## Propojka

Nejprve si vyzkouším napsat jednoduchý program, který zkontroluje první vstup (X1) a pokud bude sepnut, sepne první výstup (Y1), pokud dojde k rozepnutí prvního vstupu (X1) dojde také k rozepnutí prvního výstupu (Y1). Nejprve vložíme kontakt do programu. Klikneme dvojklikem na „Contact (NO)“ nacházející se v instrukčním okně (viz obrázek). Poté se zobrazí

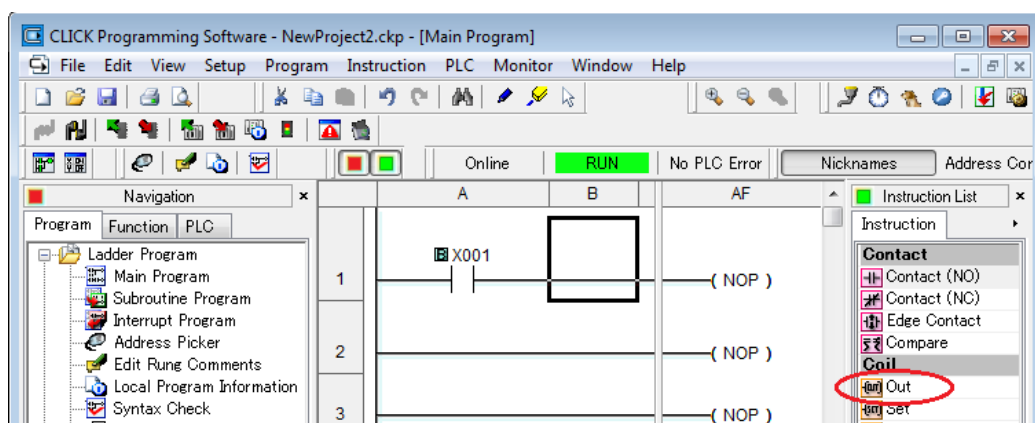
okno „Contact Normally Open“ ve kterém zvolíme příslušný kontakt. V našem případě se bude jednat o kontakt X1.



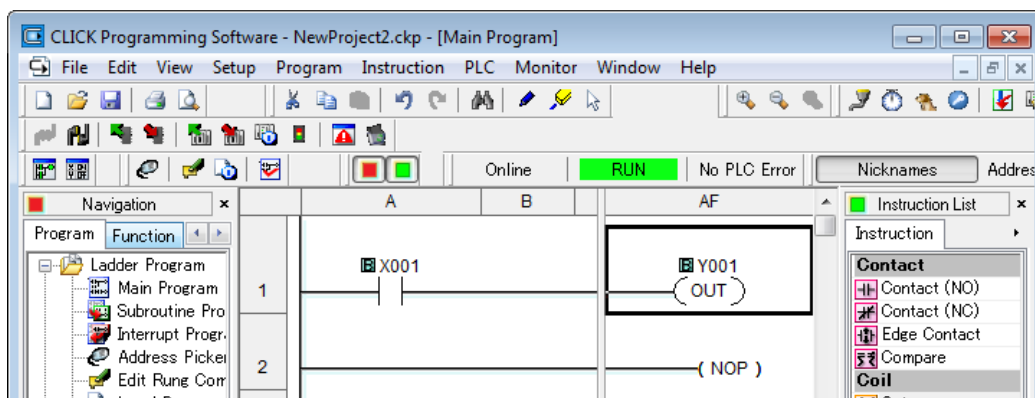
Poté co jsme zadali kontakt X1 a stiskli OK, se na dané linii objevil kontakt X1



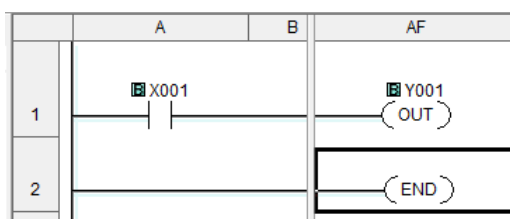
Dále musíme zadat pravou část liniového diagramu tedy akci. Naší akcí je v tomto případě sepnutí výstupu Y1.



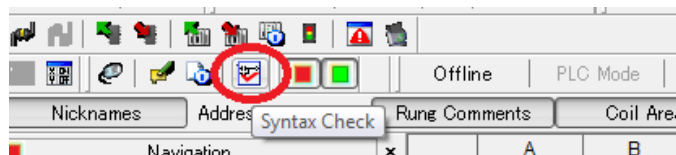
Dvojklikem klikneme na „Out“ a zobrazí se okno, ve kterém zadáme Y1. Poté se v našem programu objeví na pravé straně výstup Y1.



Nyní je potřeba na konec programu vložit ukončení. Jedná se o instrukci END. Klikneme na druhý řádek do pravé části na (NOP). V instrukčním okně dvojným kliknutím na END vložíme danou instrukci. Výsledný program pak vypadá následovně.



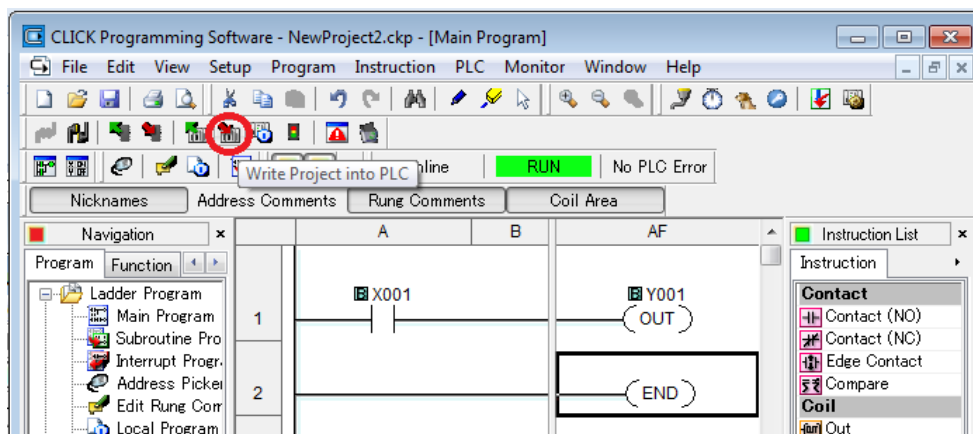
Nyní je program hotov a je potřeba jej nahrát do CPU modulu. Nejprve však ověříme jeho správnost. V horní části pod menu jsou umístěny další funkce. Mezi nimi je i funkce „Syntax Check“ (Klávesa F8). Tato funkce nám ověří správnost našeho programu.



V dolním okně pro výpis, nás bude software informovat o správnosti programu.

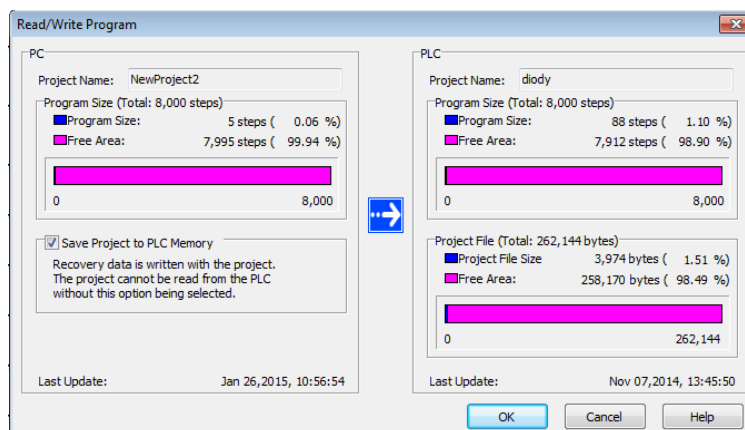


Poté co jsme ověřili správnost programu, nezbývá než nahrát daný projekt do CPU modulu. Opět v horní části softwaru je pod menu umístěna funkce „Write Project into PLC“.



Po kliknutí na tuto funkci se zobrazí následující okno, které popisuje stav paměti a velikosti projektu.

**Pozn.** V případě neuloženého projektu je vyžadáno jeho uložení, před nahráním do CPU modulu.



V tomto okně je demonstrována velikost projektu nového (v levé části), projektu nahraného v CPU modulu (v pravé části) a velikost volné paměti CPU modulu. Po kliknutí na OK se vás zeptá, zda souhlasíte s přepsáním stávajícího projektu a po odsouhlasení, přehraje projekt v CPU modulu.

Poté co jste nahráli projekt do CPU modulu, přepněte modul do RUN módu.

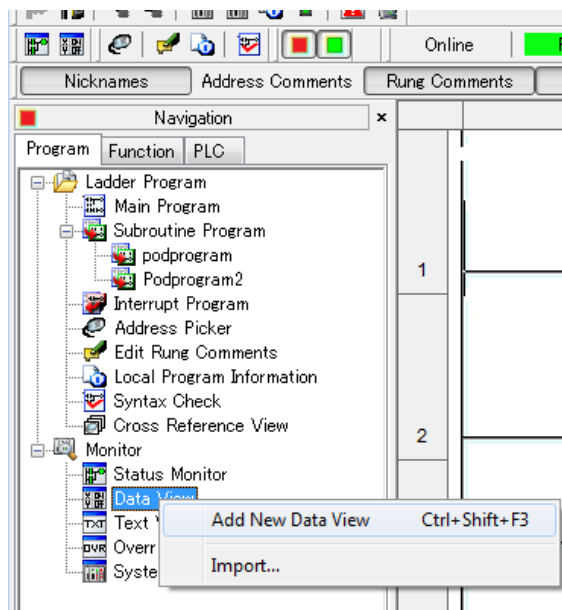
**Pozn.** Pokud je CPU modul v RUN módu je nutné ho přepnout do STOP módu, před nahráním projektu, jinak nelze projekt přehrát.

**Pozn.** Dojde-li při přechodu do režimu běhu programu k chybě, nebo nelze-li programovatelný automat do režimu RUN vůbec přepnout, nahlédněte na konec této příručky do dodatku A: „Řešení problémů“.

Následně můžeme ověřit funkci programu. Přepněte přepínač u vstupu X1 do polohy sepnuto. U vstupu X1 se na čelním panelu programovatelného automatu rozsvítí LED dioda indikující sepnutí daného vstupu. Podle zadání úkolu se má po sepnutí X1 sepnout výstup Y1. Zkontrolujte, zda se inicializoval výstup Y1, tj. že se rozsvítí LED dioda u výstupu Y1 na čelním panelu programovatelného automatu. Rozepněte vstup X1 a zkontrolujte, zda i výstup Y1 se

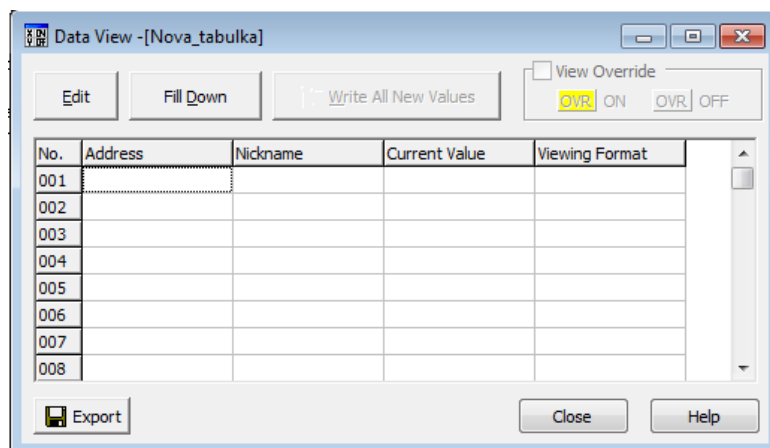
rozepte. Celý tento postup můžete kontrolovat také na obrazovce počítače, kde můžete sledovat sepnutí jednotlivých I/O bodů, pomocí **Data View**.

### Data View



K tomu abychom mohli použít tabulku Data View ji musíme nejprve přidat. Kliknete-li pravým tlačítkem na položku Data View v sekci Monitor. Můžete přidat novou tabulku Data View nebo importovat některou z vytvořených (například z jiných projektů).

Vytvoří se vám následující tabulka. V této tabulce sami určujete, které prvky chcete sledovat. U složitějších programů je to velice dobrá pomůcka. Tuto tabulku, po jejím vytvoření, vyvoláte dvojklikem na její název, který se bude nacházet jako podpoložka Data View.



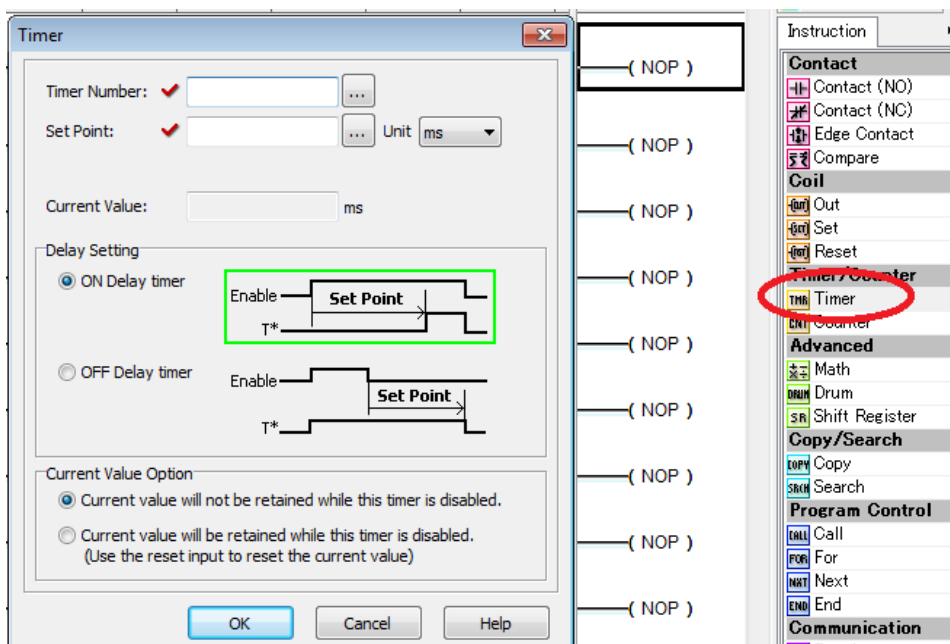
Prvky se do tabulky vkládají následovně. Klikněte v tabulce na prázdný řádek ve sloupci Address a vyberte požadovaný prvek podle adresy. Záznamy v řádku pro zbylé sloupce se vyplní automaticky danými hodnotami. Tedy aktuální hodnotou a jejím datovým formátem.

Výhodou Data View je především zobrazení stavů, ve kterých se jednotlivé prvky nacházejí. Ovšem pomocí Data View jsme schopni tyto stavy měnit a zapisovat do CPU modulu. V horní části se nachází tlačítko „Write All New Values“, pomocí kterého docílíte změny stavu v CPU modulu.

## Zpožd'ovací relé

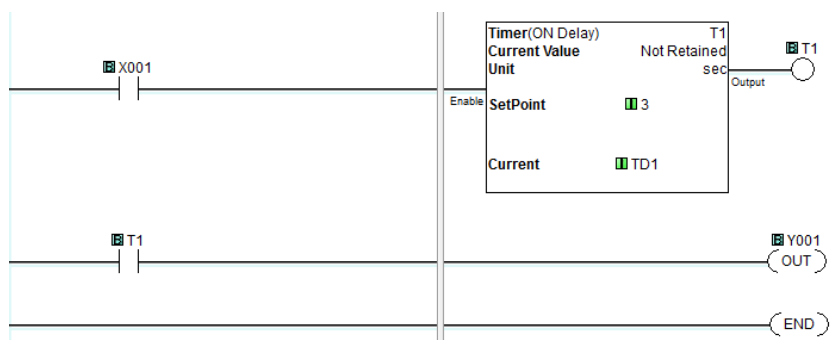
Použijte programovatelný automat CLICK jako zpožd'ovací relé. Výstup Y1 se aktivuje 3 sekundy po sepnutí vstupu X1.

Programovací software CLICK umožňuje vytvořit v liniovém diagramu na straně výstupu tzv. časovač (Timer), který načítá určité časové intervaly do předem určené hodnoty. Časové intervaly jsou buď milisekundy, sekundy, minuty, hodiny, dny. Časovač vložíme do liniového diagramu následovně: Označte NOP, kam chcete vložit časovač a klikněte na TIMER v seznamu instrukcí na pravé části. Poté se zobrazí následující okno.



V daném okně (obrázek výše) lze daný časovač nastavit. V horní části se volí číslo časovač a hodnota, do které má čítat. Tato hodnota může být zapsána jako konstanta, tedy číslo, nebo může být přebírána z nějakého registru. Dále lze vybrat typ časovače ON Delay nebo OFF Delay. V našem případě zvolíme ON Delay. A jako poslední je v nastavení určit jak se bude časovač resetovat. První volba je bez Resetovacího signálu, časovač se resetuje po odpojení signálu Enable. Druhá volba opatří časovač dalším vstupem a to vstupem Reset, který bude nulovat časovač.

K tomu, aby programovatelný automat vykonával tu činnost, kterou jsme vymezili v zadání druhého úkolu, je ještě třeba aktivovat výstup Y1. K tomu poslouží právě spínací kontakt T1 z časovače. Vytvoříme tedy na další přičce tuto akci.





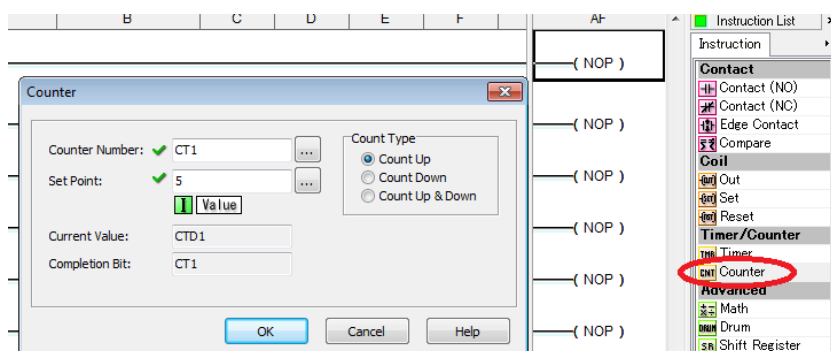
Program nahrajte do programovatelného automatu stejným způsobem jako u první úlohy. Pomocí LED diod na předním panelu a okna Data View ověřte, zda programovatelný automat vykonává tu činnost, kterou jsme mu předepsali v zadání úlohy. Sepněte kontakt X1 a sledujte, jestli se kontakt výstupu Y1 sepne po 3 sekundách. Rozepněte kontakt X1 a sledujte, zda se výstup Y1 okamžitě rozepne. Neplní-li program zcela tyto funkce, překontrolujte, je-li váš líniový diagram shodný s výše uvedeným líniovým diagramem.

## Čítač sepnutí

Vytvořte pomocí líniového diagramu program pro programovatelný automat CLICK tak, aby program načítal počet sepnutí kontaktu vstupu X1. Jestliže počet sepnutí překročí hodnotu 5, program sepne výstup Y1 a jestliže počet sepnutí dosáhne 12, čítač se vynuluje a rozepne se výstup Y1.

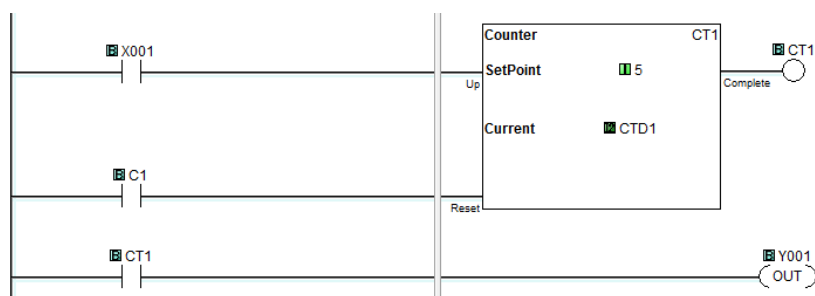
Programovací software CLICK umožňuje vytvořit v líniovém diagramu na straně výstupů tzv. čítač (Counter). Lze vytvořit dvouvstupový čítač, kde jeden vstup čítá impulsy nahoru (nebo dolů) a druhý vstup čítač nuluje. Nebo lze vytvořit třívstupový čítač, kde jeden vstup čítá impulsy nahoru, druhý vstup odečítá a třetí vstup nuluje čítač. Čítač tedy čítá dané impulsy a pokud načítá do nastavené hodnoty sepne svůj výstup. Výstup drží ve stavu sepnuto i po čas kdy je hodnota větší než nastavená.

Čítač vložíme do líniového diagramu následovně: Označte NOP, kam chcete vložit časovač a klikněte na COUNTER v seznamu instrukcí na pravé části. Poté se zobrazí následující okno.

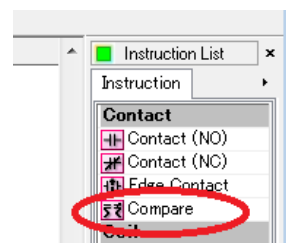
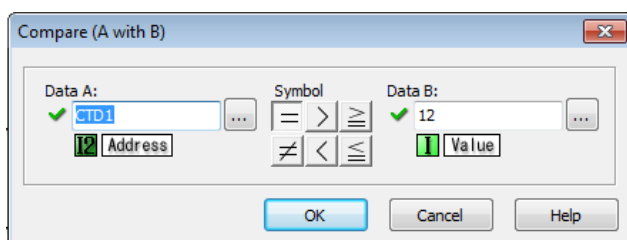


V daném okně (obrázek výše) se tedy vybere příslušné číslo čítače a nastaví se hodnota, pro kterou čítač sepne svůj výstup (v našem případě 5). Dále je možné v pravé části zvolit typ čítače. Tedy zda bude čítat nahoru, dolů nebo bude obsahovat dva vstupy, jeden pro čítání nahoru a druhý pro čítání dolů. Vložíme tedy čítač do líniového diagramu.

Podle zadání úkolu má programovatelný automat přesně po pěti sepnutích vstupu X1 sepnout výstup Y1. Využijeme k tomu kontrolní spínací kontakt daného čítače, tedy jeho výstup. Tento výstup použijeme jako spínací kontakt k výstupu Y1. K vynulování čítače a tedy k rozepnutí výstup Y1 využijeme pomocný kontakt C1, který jsme umístili na nulovací vstup čítače. Pomocný kontakt C1 bude aktivován, jestliže bude splněna podmínka dána porovnávacím kontaktem.



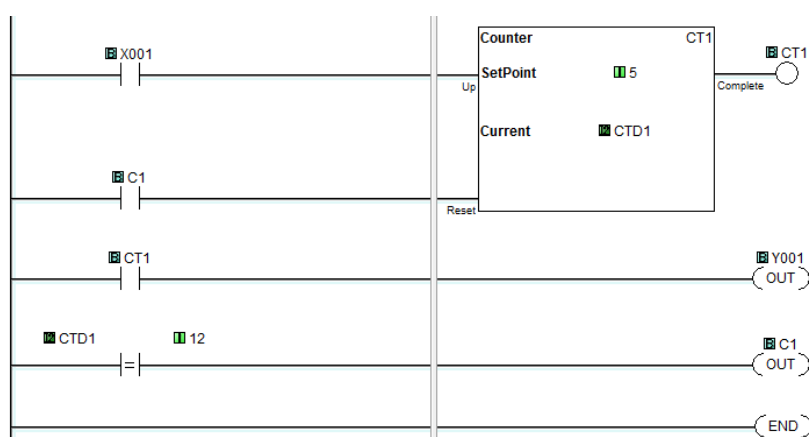
Vložíme tedy na začátek nové příčky prvek COMPARE. Jedná se o porovnávací kontakt, který vyhodnotí dvě hodnoty nebo data uložená v registrech a podle zadaného operátoru určí výstup.



V našem případě zvolíme operátor „=“. Budeme porovnávat aktuální hodnotu z čítače, tedy hodnotu CTD1 a hodnotu 12.



Pomocí porovnávacího kontaktu na straně vstupů programovatelný automat v každém scanu zjistí, zda aktuální hodnota čítaných impulsů není rovna číslu 12 (viz zadání). Jestliže ano pak programovatelný automat sepne pomocný kontakt C1, který vynuluje čítač.

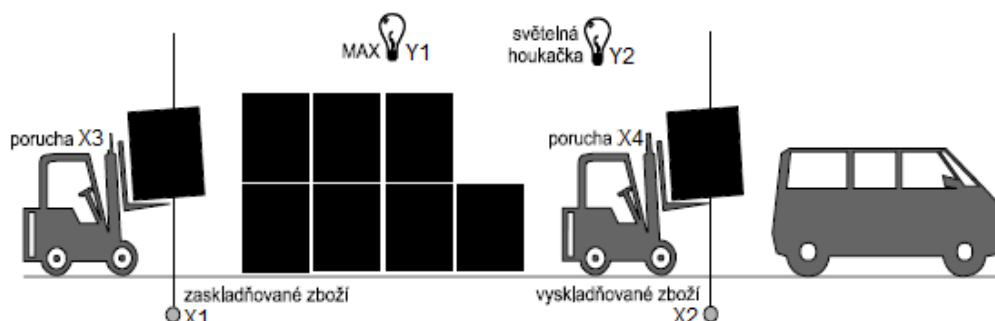


Program nahrajte do programovatelného automatu stejným způsobem jako u první a druhé úlohy. Pomocí LED diod na předním panelu a okna Data View ověřte, zda programovatelný automat vykonává tu činnost, kterou jsme mu předepsali v zadání úlohy. Programovatelný automat bude čítat počet sepnutí vstupu X1. V okamžiku, kdy počet sepnutí dosáhne 5, sepne se výstup čítače CT1, který sepne kontakt výstupu Y1. Automat bude nadále

čítat sepnutí vstupu X1. Kontakt CT1 zůstává sepnutý, protože aktuální hodnota čítače (můžete ji kontrolovat pomocí Data View) je vyšší než přednastavená hodnota. Jestliže čítač dočítá do 12, sepne se porovnávací kontakt CTD1 = 12 a tím i pomocný kontakt C1. V následujícím scanu se vynuluje čítač CT1 kontaktem C1 na nulovacím vstupu čítače. Tím se také rozeprve kontrolní kontakt CT1 (aktuální hodnota čítače je nula, což je menší než přednastavená hodnota), porovnávací kontakt CTD1 = 12 a pomocný kontakt C1. Programovatelná automat je okamžitě připraven dále čítat impulsy od vstupu X1.

## Simulace skladu

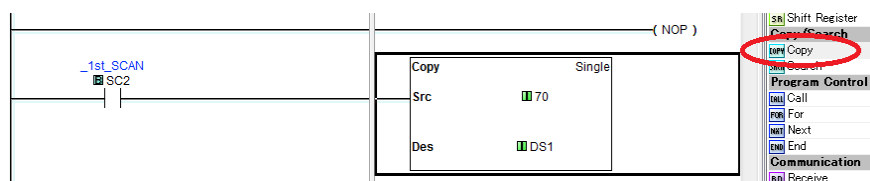
V této poslední úloze vytvoříme pomocí liniových diagramů program pro programovatelný automat CLICK tak, že automat bude hlídat provoz skladu z následujícího obrázku.



- |                       |  |
|-----------------------|--|
| <b>X1</b>             | Bedny přicházející do skladu jsou registrovány čidlem, které simulujeme vstupem X1 na automatu (přepínač připojený podle kapitoly jedna)   |
| <b>X2</b>             | Bedny odcházející ze skladu jsou registrovány čidlem, které simulujeme vstupem X2  |
| <b>X3</b>             | Porucha zaskladňovacího zařízení (porucha = sepnuto)   |
| <b>X4</b>             | Poruchy vyskladňovacího zařízení (porucha = sepnuto)   |
| <b>Y1</b>             | Překročena kritická mez naplnění skladu  |
| <b>Y2</b>             | Světelná houkačka, porucha některého ze zařízení   |
| <b>Počet výrobků</b>  | v bedně může být změněn (standardně 70). Nový počet zadejte do automatu do registru DS1, bude akceptován při vyprázdnění skladu. Aktuální počet výrobků v bedně bude v registru DS2. |
| <b>Kritická mez</b>   | naplnění skladu se odvíjí od velikosti beden, tj. od počtu výrobků v bedně. Do skladu se vejde maximálně 700 výrobků. Kritický počet beden je uložen v automatu v registru DS3.      |
| <b>Aktuální počet</b> | výrobků ve skladu budiž neustále k dispozici v automatu v registr DS4.   |

Protože se jedná opět o nový program, vytvořte nový projekt. Prvním krokem v novém programu bude zajistit, aby programovatelný automat po startu nastavil implicitní hodnotu počtu výrobků v bedně. Podle zadání tedy 70 kusů. Programovací software CLICK umožňuje pomocí speciálních kontaktů SC vykonávání mnoha užitečných funkcí. Speciální kontakt SC2 je sepnut pouze v prvním scanu (v prvním vykonání programu) po zapnutí do režimu RUN, tzn. Jak v případě přepnutí programátorem z prostředí programovacího nástroje, tak i v případě, že dojde k znovuoobnovení napájení po výpadku, jestliže automat byl před výpadkem v režimu RUN. Speciální kontakt SC2 se do liniového diagramu vkládá stejně jako spínací kontakty vstupů v předchozích úlohách.

Při splnění této podmínky (v prvním scanu) se do programovatelného automatu do registru DS1 uloží implicitní hodnota 70. K tomu využijeme instrukci COPY. Pomocí této instrukce vložíme do registru DS1 hodnotu 70.



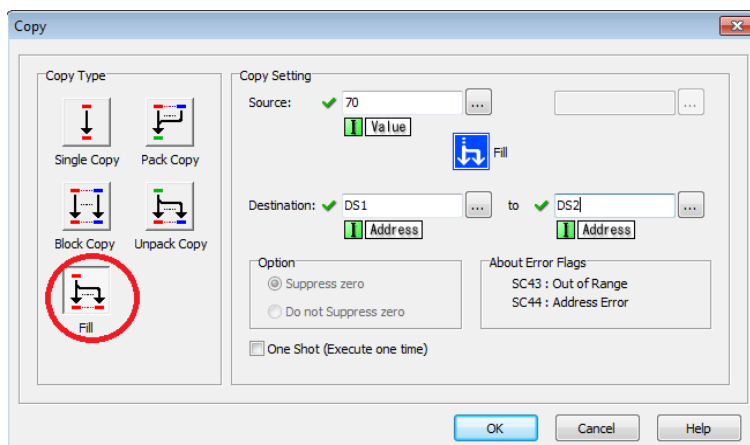
Jelikož bude zapotřebí nahrát tuto hodnotu i do registru DS2 provedeme paralelní instrukci. Navěsíme tedy tuto instrukci na straně výstupu pod instrukci COPY, tedy na stejnou vstupní podmínku na začátku příčky (SC2). Přesuňte kurzor na instrukci COPY, podržte CTRL a stiskněte kurzorovou klávesu šipku dolů, nebo pomocí myši vyberte z hlavní nabídky položku EDIT, v ní nabídku Line a pomocí návodu přidat zmíněnou lajnu.



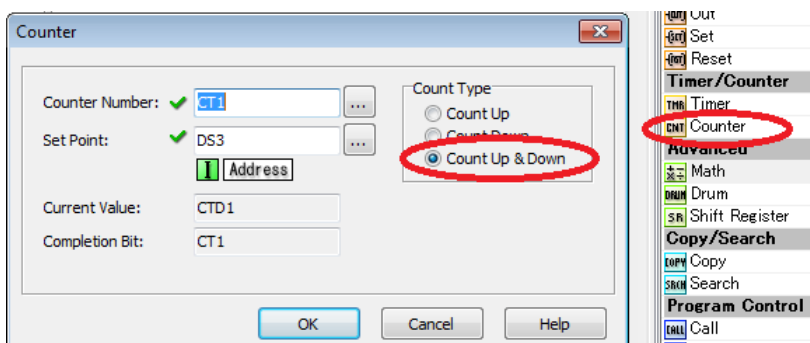
Poté co jsme přidali novou lajnu směrem dolů, můžeme na ni připojit druhou instrukci COPY, která nám naplní registr DS2 rovněž hodnotou 70.



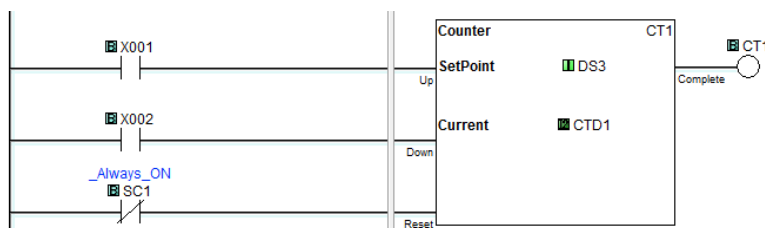
Naplnění obou registrů lze zařídit i pomocí jedné instrukce COPY. V okně COPY zvolíme typ kopírování FILL. A nahrajeme tak hodnotu do obou registrů pomocí jedné instrukce.



V průběhu vykonávání programu můžeme pomocí okna Data View měnit počet kusů v bedně. Podle zadání se však tento počet smí změnit pouze při vyprázdňeném skladu. Protože počet beden ve skladu budeme interpretovat pomocí třívstupého čítače NahoruDolu, kde přičítající vstup bude reprezentovat příchod beden do skladu (X1) a odčítající vstup odchod beden ze skladu (X2) můžeme zjistit vyprázdnění skladu porovnávacím kontaktem, zda se aktuální hodnota čítače rovná 0. Jestliže nastane tento případ, program zkopíruje budoucí hodnotu počtu výrobků v bedně do aktuální hodnoty a provede výpočet kritické meze naplnění skladu. Nejprve tedy vytvoříme čítač pro příchozí a odchozí bedny. Označíme prázdný řádek na pravé straně v části pro vstupy a vložíme čítač.

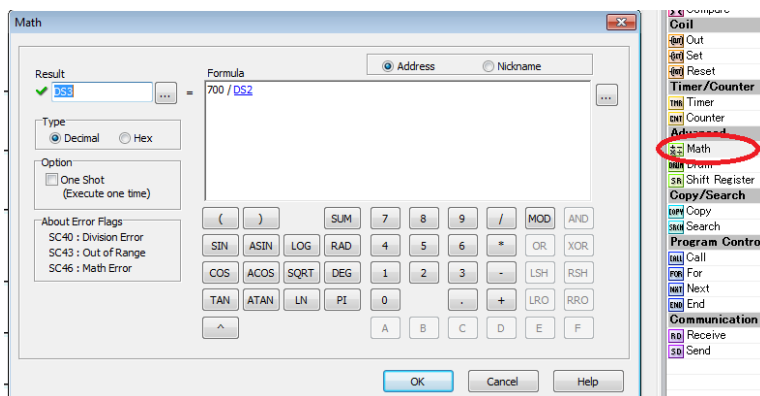


Nesmíme zapomenout zvolit čítač typu NahoruDolu. Jako bod čítání zvolíme registr DS3, ve kterém bude uložena hodnota maximálního počtu beden.

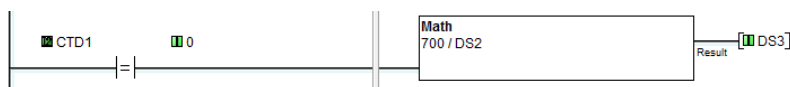


Pro resetovací vstup použijeme speciální kontakt SC1 (Always ON) v režimu No Contact. Dostaneme tak speciální kontakt Always OFF.

Do skladu se vejde maximálně 700 kusů výrobků, tzn., že kritická mez počtu beden ve skladu bude daná jako podíl maximálního počtu beden výrobků (načtená z DS1) a počtem výrobků v bedně. Celočíslné dělení provede automat pomocí instrukce MATH, na pravé straně v seznamu instrukcí. Poté co ji vybereme, se zobrazí následující okno. V daném okně lze zadat matematický vzorec pomocí dostupných operátorů. Dále v levém horním okně vybereme kam ukládat výsledek. V našem případě zvolíme registr DS3. Matematický vzorec je tedy ve formě  $DS3 = 700/DS2$ .



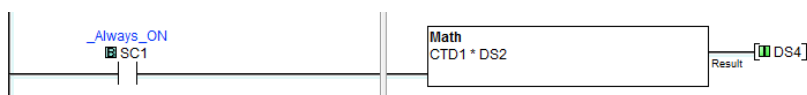
Tato matematická operace se může vykonat pouze, když je počet beden ve skladu roven 0. Čili musí se vykonat pouze, když je aktuální hodnota čítače rovna 0. Vložíme tak před tuto instrukci podmínku v podobě porovnávacího kontaktu, který porovná aktuální hodnotu čítače s nulou.



Kritická mez je tak dána hodnotou v registru DS3. Do této hodnoty čítá náš čítač. Pokud tedy čítač dočítá do této hodnoty nebo ji přesáhne, sepne se jeho kontakt CT1, kterého využijeme k aktivování signalizace překročení meze (sepnutí kontaktu výstupu Y1).



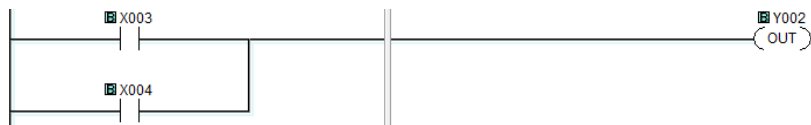
Dle zadání má být v každém okamžiku k dispozici aktuální počet výrobků ve skladu v registru DS4. Počet výrobků automat vypočítá z počtu beden ve skladu (aktuální hodnota čítače) vynásobeného aktuálním počtem výrobků v bedně. K násobení použijeme opět matematickou instrukci MATH. Kde jako vzorec zadáme  $DS4 = CTD1 * DS2$ .



Jako podmínku na levé straně použijeme speciální kontakt SC1, díky kterému se matematická operace provede vždy, tedy v každém novém scanu.

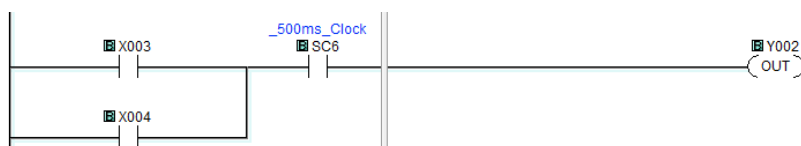
Podle zadání se má světelná houkačka (výstupní kontakt Y2) inicializovat, jestliže dojde k poruše zaskladňovacího nebo vyskladňovacího zařízení. Porucha zařízení je reprezentována spínacím kontaktem X3 respektive X4. Aby byla sepnutá světelná houkačka při poruše

kteréhokoliv z nich, je nutné, aby automat vykonal logickou funkci NEBO (OR) nad vstupy X3 a X4. V liovém diagramu je tato funkce interpretována paralelním spojením kontaktů.



Kontakt X3 umístíte do liového diagramu obvyklým způsobem, přemístíte kurzor po příčce hned za tento kontakt, stisknete a podržte CTRL a stisknete kurzorovou klávesu šipku dolů. Na příčce se vytvoří odbočka, před kterou na nižším řádku obvyklým způsobem umístíte kontakt vstupu X4. Bude-li sepnut alespoň jeden z kontaktů X3 nebo X4 pak se také sepnou kontakt výstupu Y2.

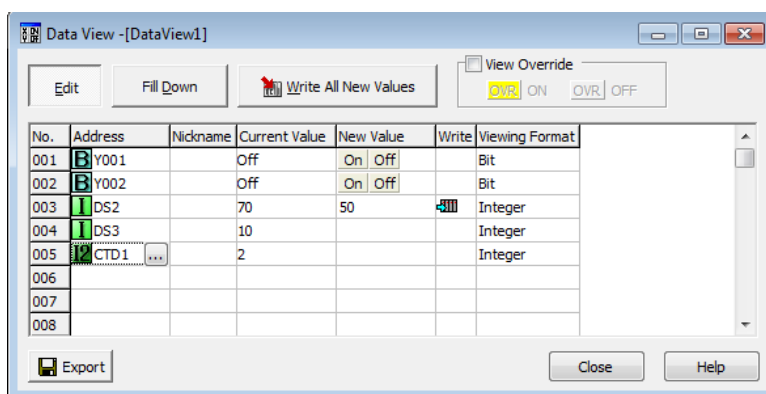
Podle zadání má však světelná houkačka blikat. Aby světelná houkačka blikala, docílíme přerušovaným spínáním kontaktu Y2. Vložíme tedy za vstupy X3 a X4 do série speciální kontakt SC6, který je vždy 0,5 sekundy sepnut a 0,5 sekundy rozepnut.



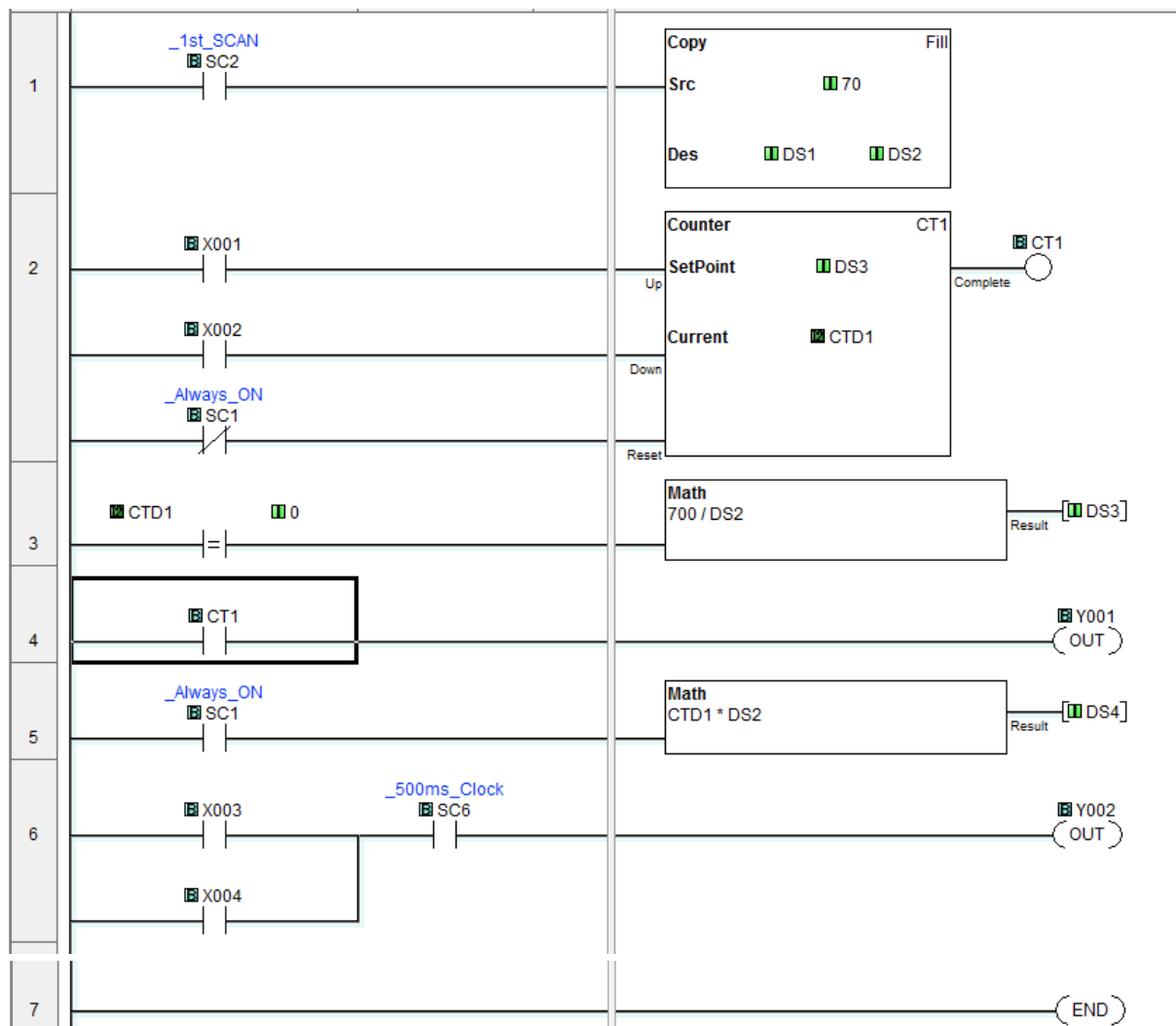
Jestliže tedy bude sepnut alespoň jeden ze vstupů X3 nebo X4, pak každou půl sekundu bude splněna vstupní podmínka příčky (bude sepnut také speciální kontakt SC6) a kontakt výstupu Y2 se sepnou.

Nesmíme zapomenout na ukončení celého programu instrukcí END stejně jako v minulých úlohách. Poté můžeme daný projekt nahrát do programovatelného automatu CLICK a přepnout ho do režimu RUN.

Poté si otevřeme okno Data View a vložíme si do něho registry DS1, DS2, DS3, DS4, CTD1, Y1, Y2. Dále budeme simulovat vstup a výstup beden do skladu pomocí vstupu X1 respektive X2. To tedy znamená, že bedna je zaregistrována při sepnutí a rozepnutí vstupního kontaktu. Sledujte v okně Data View jak se mění hodnota aktuálního počtu beden ve skladu, po sepnutí obou vstupů. Sledujte, zda se sepnou výstupní kontakt Y1 jestliže bude překročena kritická mez naplnění skladu. Vyzkoušejte také všechny kombinace vstupů X3 a X4 (porucha zaskladňovacího a vyskladňovacího zařízení) a kontrolujte, zda výstupní kontakt Y2 spíná při splnění podmínky nebo se sekundovou periodou.



Programovací software CLICK umožňuje za běhu programu změnit hodnotu uloženou na libovolné adrese, tedy v registru. Prostřednictvím okna sledování Data View můžete měnit v jednom okamžiku obsah jedné adresy nebo několika najednou.



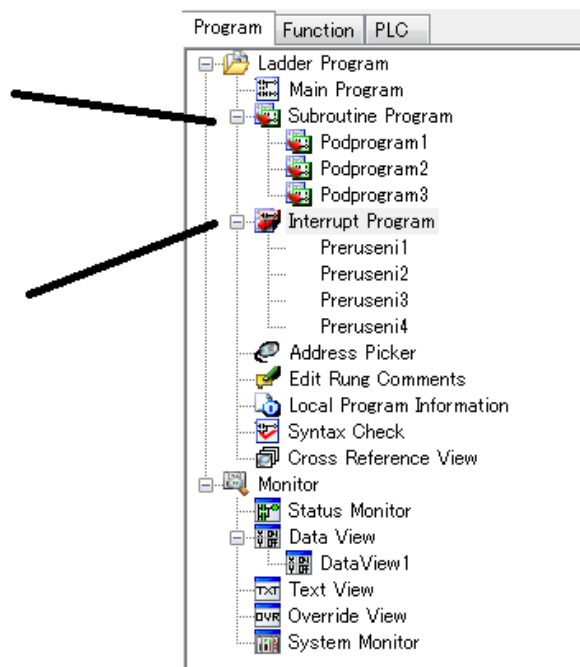


## Podprogramy a přerušení

Programovací software CLICK umožňuje také vytvářet podprogramy (Subroutine) a přerušení (Interrupt).

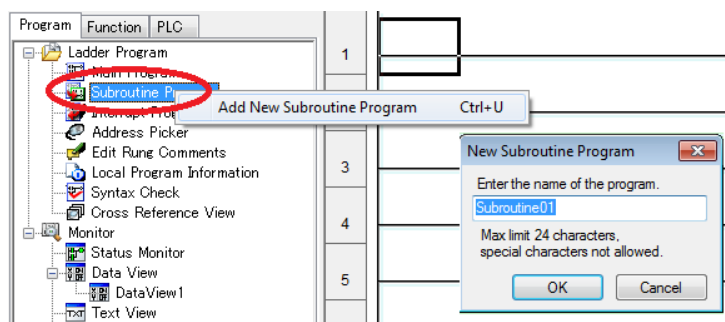
**Podprogramy** mají tu vlastnost, že je lze volat přímo z hlavního programu. Lze si tak dlouhý a nevhledný liniový diagram rozdělit do několika podprogramů tak, že docílíte lepšího programování a především větší přehlednosti v daném liniovém diagramu. Počet podprogramů je omezen na 986.

**Přerušení** je takový podprogram, který se vykoná na jistou událost. Tuto událost můžete vyvolat softwarově nebo externě. Basic a Standart CPU moduly podporují až 12 přerušení. Analogové CPU moduly podporují až 8 přerušení

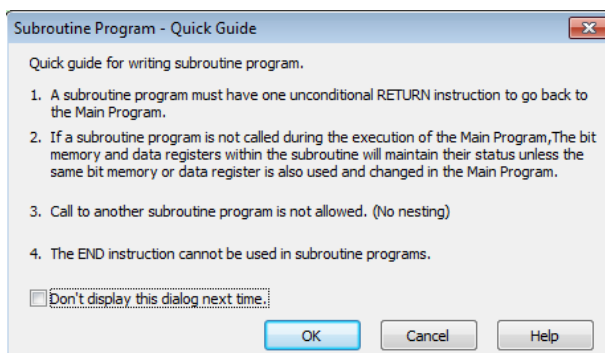


### Podprogram

Pro přidání nového podprogramu, v levém menu klikněte pravým tlačítkem na Subroutine a zvolte „Add new Subroutine Program“. Poté se zobrazí okno kam vložíte název daného podprogramu.



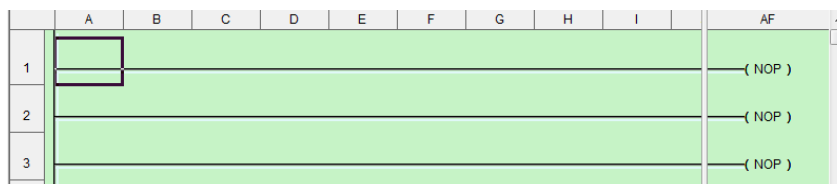
Poté co zadáte název a kliknete na OK, se zobrazí následující okno, které vás bude informovat o nutnostech, které musíte splnit pro správné použití daného podprogramu.



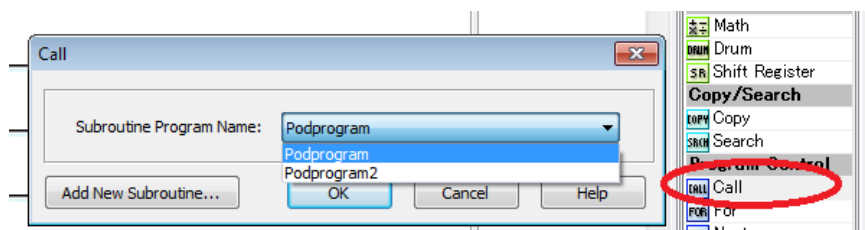
Jedná se především o tyto podmínky:

- Podprogram musí být ukončen instrukcí RETURN (nikoli END jako u hlavního programu) tak, abychom se mohli po vykonání vrátit do hlavního programu
- Pokud nebude podprogram volán z hlavního programu, budou veškeré registry a paměti použité v podprogramu držet svůj statut (hodnotu), pokud nebudou změněny v hlavním programu, potom se změní také
- Zavolat jiný podprogram není dovoleno
- Nepoužívejte END

Poté se otevře nový podprogram, který snadno poznáte podle barvy pozadí, která bude zelená.



Vytvoříme-li tedy nový podprogram a vložíme do něho část programu, řádně ho ukončíme pomocí instrukce RETURN, nebudeme volat jiný podprogram atd. Můžeme do hlavního programu vložit instrukci CALL, která bude daný podprogram volat.

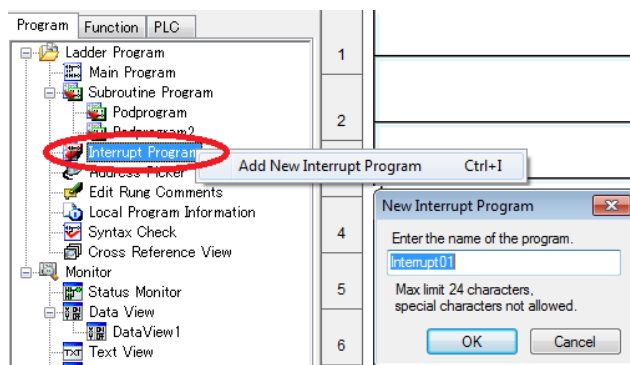


V menu instrukcí vybereme instrukci CALL a poté co se nám zobrazí okno (viz. Výše) vybereme konkrétní podprogram, který chceme volat. A stiskneme OK. V levé části okna se nachází tlačítko „Add New Subroutine...“. Pomocí tohoto tlačítka můžete podprogram vytvořit okamžitě, tedy nemusíte podprogram vytvářet dopředu. Poté co jsme potvrdili podprogram, se na pravé straně přičky vytvoří instrukce CALL, která daný podprogram zavolá.

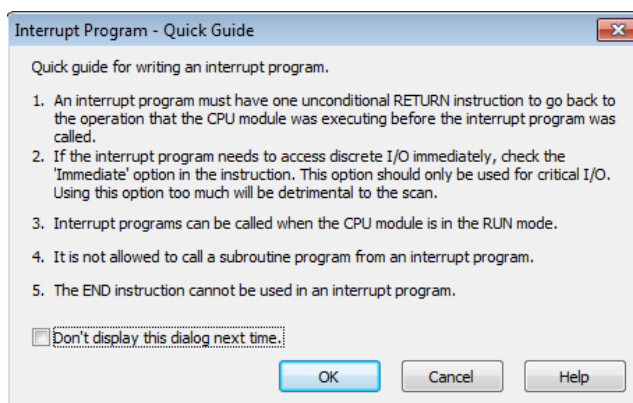


## Přerušení

Pro přidání nového přerušení, v levém menu klikněte pravým tlačítkem na Interrupt Program a zvolte „Add New Interrupt Program“. Poté se zobrazí okno kam vložíte název daného přerušení.



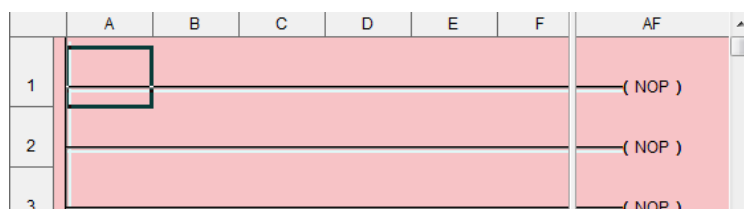
Poté co zadáte název a kliknete na OK, se zobrazí následující okno, které vás bude informovat o nutnostech, které musíte splnit pro správné použití daného přerušení.



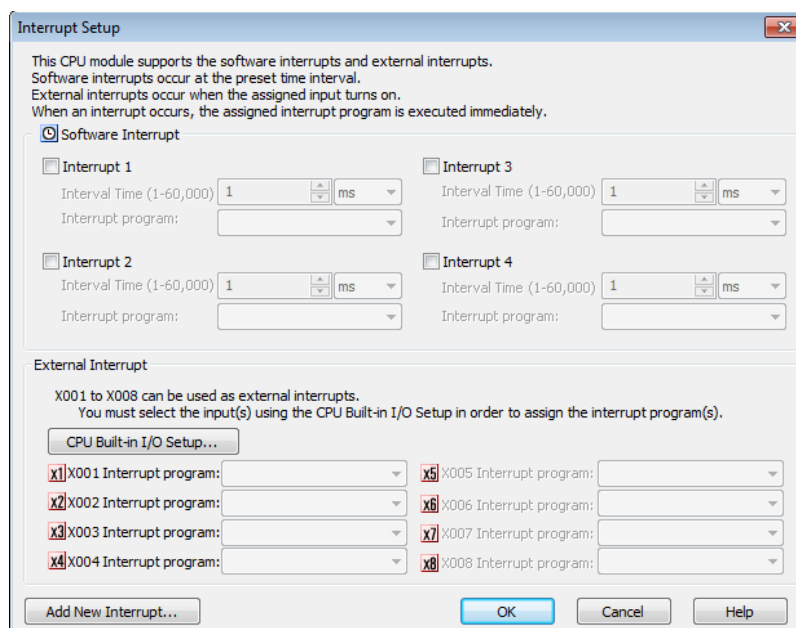
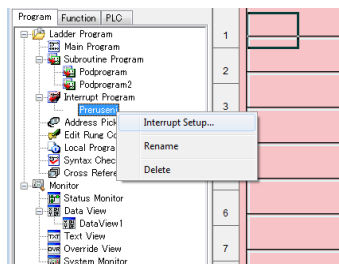
Jedná se především o tyto podmínky:

- Přerušení je vlastně podprogram vyvolán událostí, tudíž musí být ukončen instrukcí RETURN (nikoli END jako u hlavního programu) tak, abychom se mohli po vykonání vrátit do hlavního programu
- Přerušení se zavolá pouze, když je CPU modul v režimu RUN
- Volání podprogramů v přerušení není dovoleno
- Nepoužívejte END

Poté se otevře nový podprogram, který snadno poznáte podle barvy pozadí.

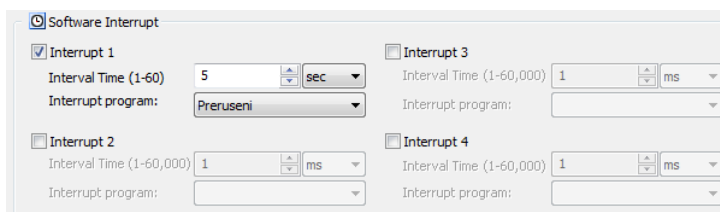


Poté co jsme vytvořili přerušení, je nutné nadefinovat událost na, kterou bude reagovat. V levém menu klikneme pravým tlačítkem myši na naše nové přerušení. Po tom, co tak učiníte, se zobrazí následující okno.

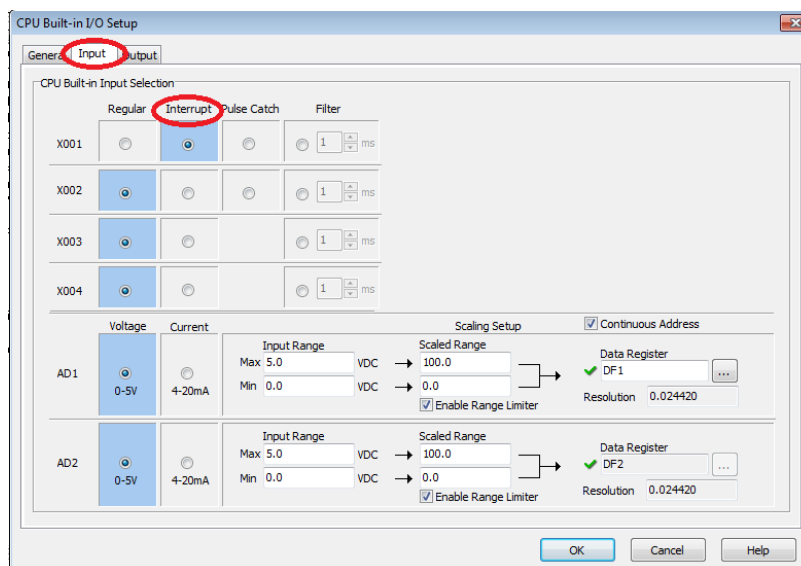


V tomto okně budeme nastavovat, na jakou událost bude přerušení reagovat. Máme dvě základní možnosti a to Softwarové přerušení, které vyvoláme v různých časových intervalech, nebo pomocí externího zařízení připojeného na vstup X1-X4.

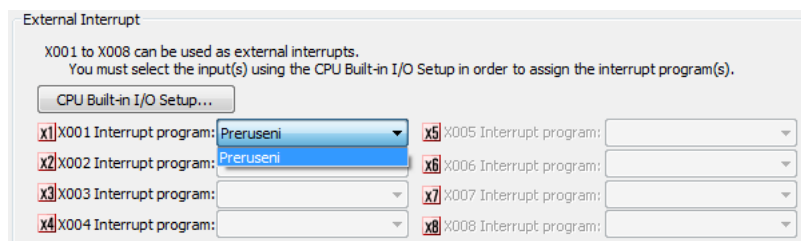
Softwarové přerušení se tedy nastavuje následovně. Zaškrtneme libovolné přerušení. Poté vybereme časový interval, po kterém se dané přerušení bude vykonávat (v obrázku 5 sekund) a poté vybrat program, který se bude vykonávat v našem případě „Preruseni“.



Externí přerušení se nastavuje tak, že musíme nejprve určit, které vstupy budou použity pro jako přerušení. Klikneme-li na tlačítko „CPU Built-in I/O Setup...“ zobrazí se následující okno.



V daném okně nás bude zajímat záložka „Input“, ve které budeme přiřazovat jednotlivým vstupům režim. První sloupec je běžné chování (Vstup), druhý sloupec označuje přerušeni (Vstup jako externí přerušeni) atd. Zvolíme tedy pro první vstup X001 režim přerušeni (Interrupt) a klikneme na OK.

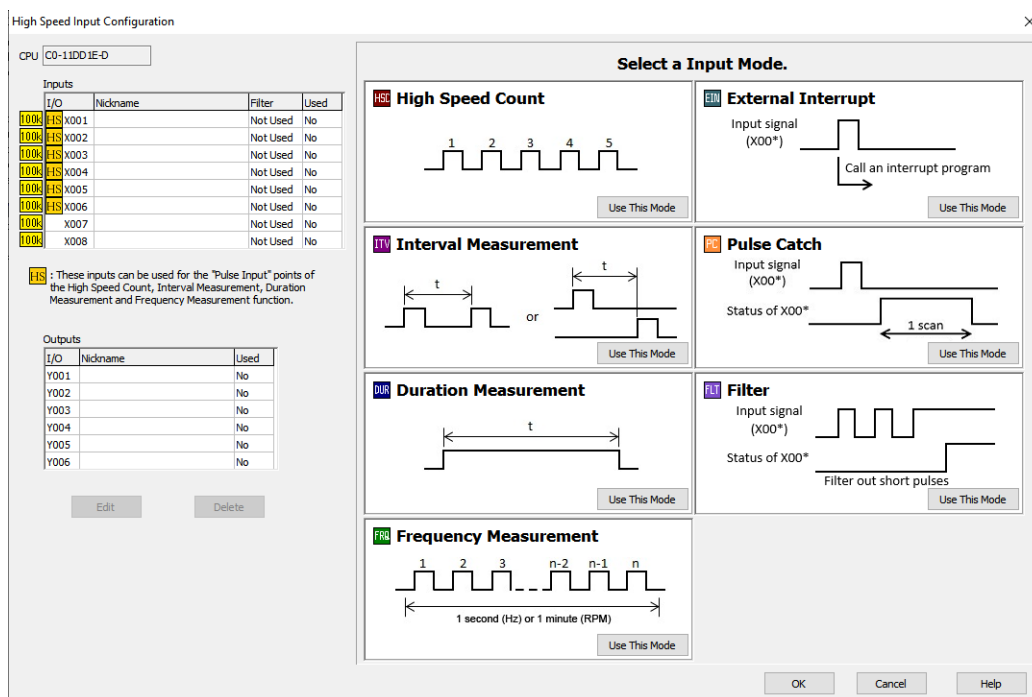


Poté se nám v původním okně pro vstup X001 otevře možnost vybrání programu pro přerušeni. Zvolíme tedy v našem případě „Preruseni“.

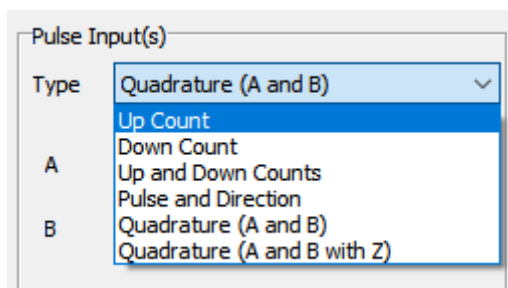
Oba tyto typy přerušeni se vykonávají nezávisle na hlavním programu. Jinými slovy, není nutné je volat v hlavním programu pomocí instrukce CALL. Přerušeni se tak vykoná podle stanovených událostí. V našem případě tedy každých 5 sekund nebo po sepnutí vstupu X001. Čili při příznaku přerušeni se přeruší hlavní program a začne se vykonávat program v přerušeni. Po jeho skončení hlavní program pokračuje v bodě, kde byl přerušeni.

## Nastavení rychlých vstupů

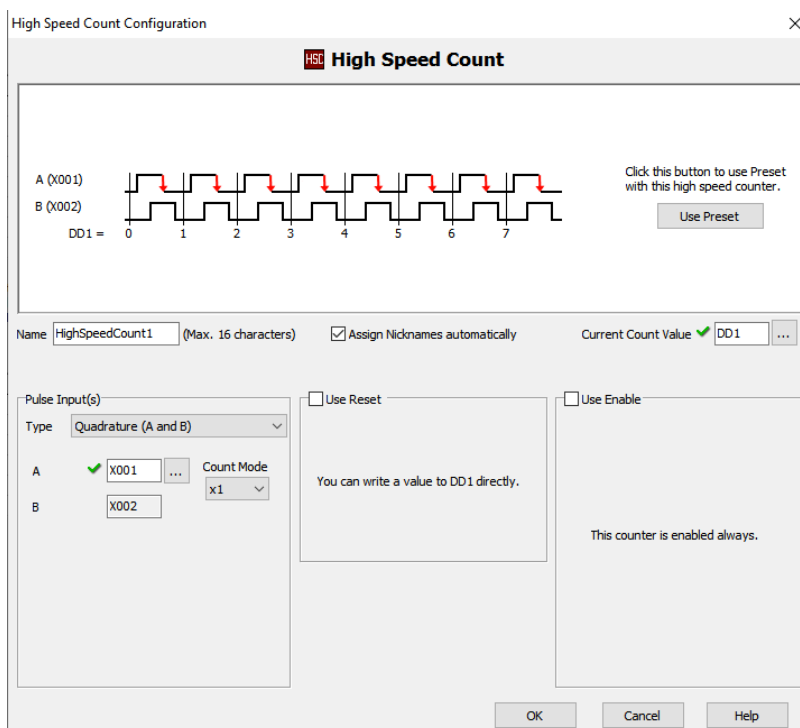
U CPU s Ethernetovým portem lze využít binární vstupy jako rychlé až do 100 kHz. U takto nastavených vstupů, lze použít několik možných konfigurací. Lze je tak využít například pro rychlé čítání, přerušení, měření intervalu, zachycení pulsu atd. Samotné nastavení se nachází v záložce **Setup -> High speed input...**



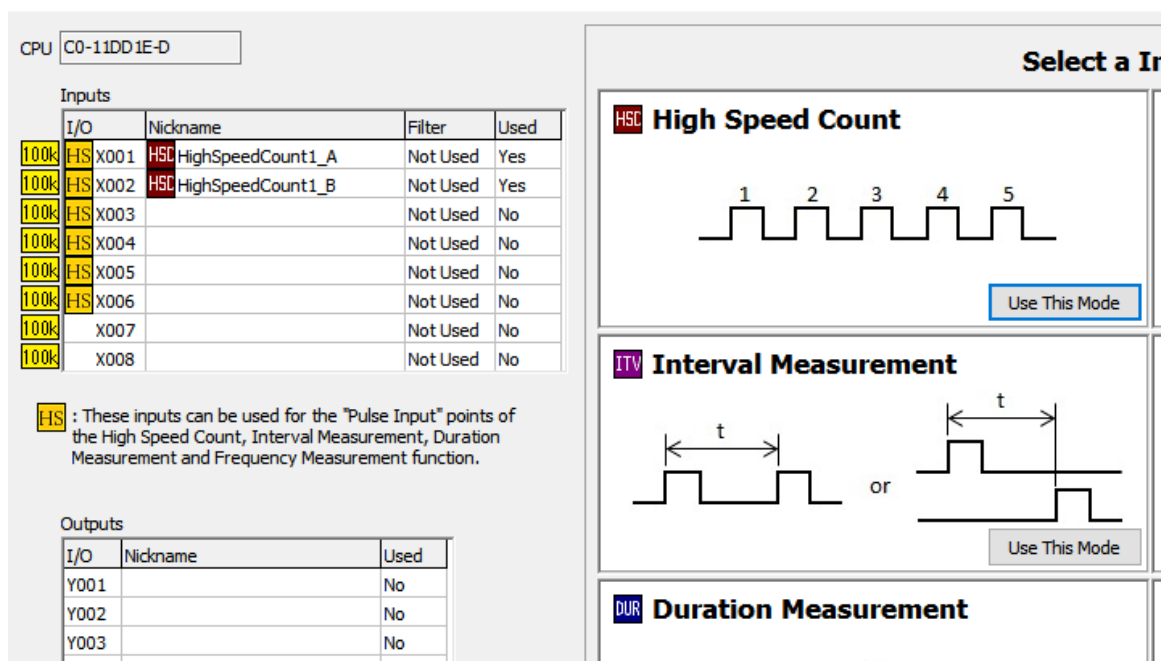
Na obrázku výše lze vidět možnosti konfigurací jednotlivých vstupů. Ukážeme si na příkladu použití vstupu pro enkodér. Použijeme tedy **High Speed Count** a klikneme na tlačítko u této možnosti **Use this Mode**. U tohoto modu si lze vybrat několik typů konfigurace.



Lze si tak zvolit co nejpřesnější typ. Po zvolení daného typu se určí který/které vstup/y budou použity pro tuto konfiguraci.



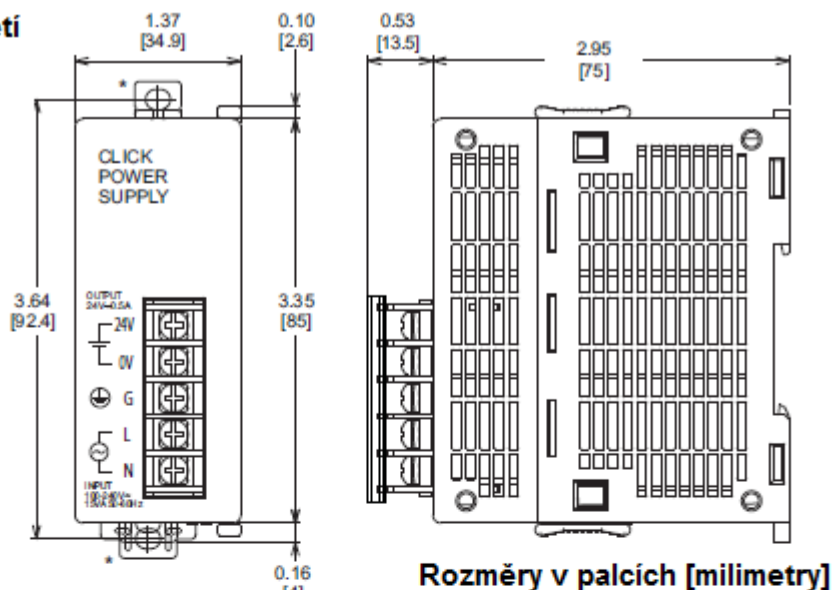
Poté co takto nastavíte vstupy. Vráťte se do úvodního okna, kde již budou použité vstupy znázorněny v levé tabulce.



Použité vstupy již nelze použít pro jinou konfiguraci. Lze však použít ostatní vstupy na odlišné konfigurace. Lze tak použít dva vstupy pro enkodér, dále pak dva vstupy pro zachycení pulsu nebo pro měření intervalu mezi pulsy.

## Rozměry a montáž systému **CLICK**

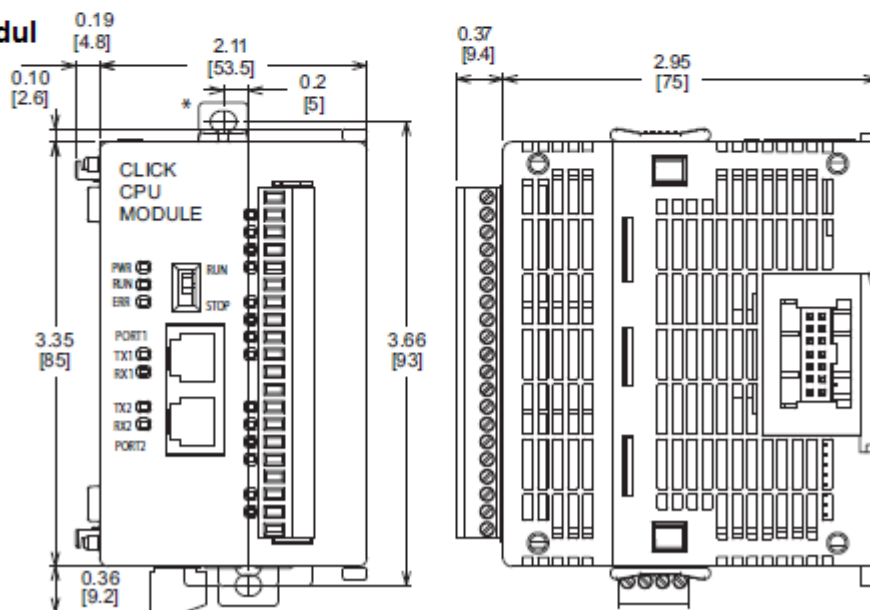
### Zdroje napětí



Rozměry v palcích [milimetry]

\* Použijte šrouby M4 pro uchycení

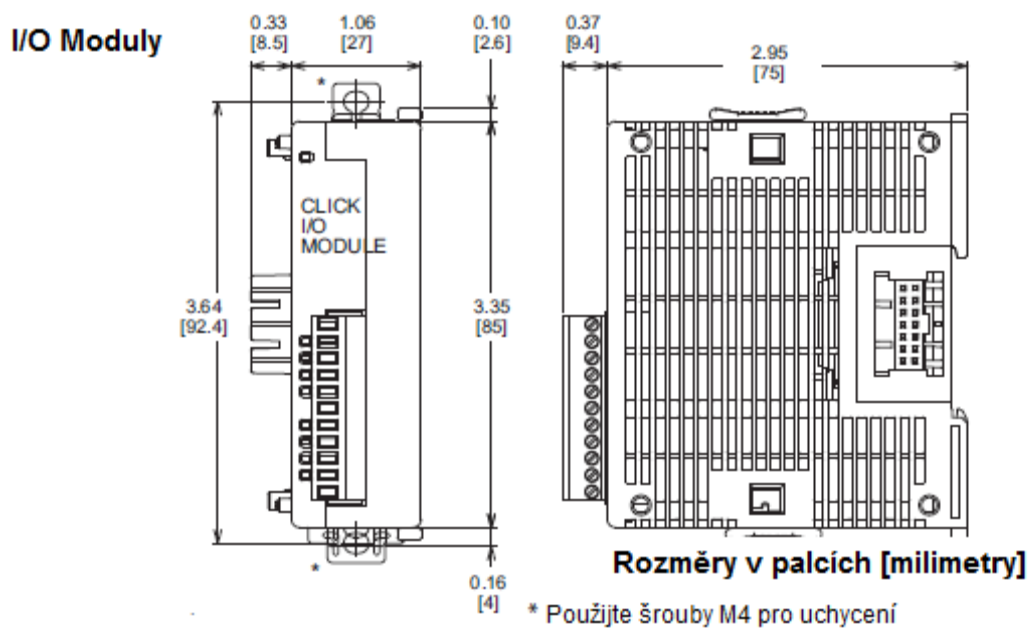
### CPU Modul



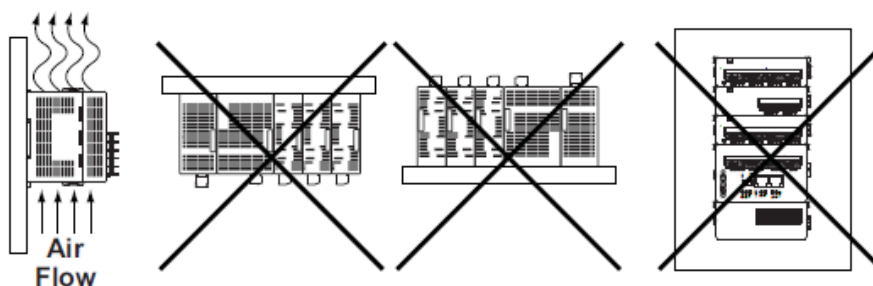
Rozměry v palcích [milimetry]

\* Použijte šrouby M4 pro uchycení

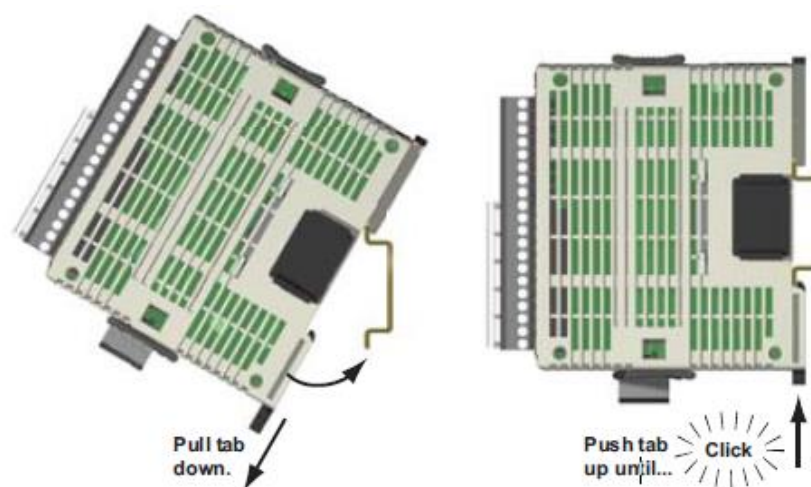




Programovatelný automat umístíme do rozvaděče tak, jak je naznačeno na následujícím obrázku, aby byla zajištěna dostatečná cirkulace vzduchu.



Uchycení na DIN lištu se provádí podle následujícího obrázku.



## Přídavné moduly systému **CLICK**

### Binární vstupy a výstupy

<b>C0-08ND3</b>	8 vstupů 12-24 VDC se společným plus nebo mínus, 2 izolované skupiny (4 body na společný vodič), vyjímatelná svorkovnice
<b>C0-08ND3-1</b>	8 vstupů 3,3-5 VDC se společným plus nebo mínus, 2 izolované skupiny (4 body na společný vodič), vyjímatelná svorkovnice
<b>C0-16ND3</b>	16 vstupů 24 VDC se společným plus nebo mínus, 4 izolované skupiny (4 body na společný vodič), vyjímatelná svorkovnice
<b>C0-08NE3</b>	8 vstupů 24 VDC/VAC se společným plus nebo mínus, 2 izolované skupiny (4 body na společný vodič), vyjímatelná svorkovnice
<b>C0-16NE3</b>	16 vstupů 24 VDC/VAC se společným plus nebo mínus, 4 izolované skupiny (4 body na společný vodič), vyjímatelná svorkovnice
<b>C0-08NA</b>	8 vstupů 100-120 VDC, 2 izolované skupiny (4 body na společný vodič), vyjímatelná svorkovnice
<b>C0-08TD1</b>	8 výstupů 3,3-27 VDC se společným mínus, 2 izolované skupiny (4 body na společný vodič), 0,3A/bod, vyjímatelná svorkovnice
<b>C0-08TD2</b>	8 výstupů 12-24 VDC se společným plus, 1 izolovaná skupina (8 bodů na společný vodič), 0,3A/bod, vyjímatelná svorkovnice
<b>C0-16TD1</b>	16 výstupů 5-27 VDC se společným mínus, 2 izolované skupiny (8 bodů na společný vodič), 0,1A/bod, vyjímatelná svorkovnice
<b>C0-16TD2</b>	16 výstupů 12-24 VDC se společným plus, 2 izolované skupiny (4 body na společný vodič), 0,1A/bod, vyjímatelná svorkovnice
<b>C0-08TA</b>	8 výstupů 17-240 VAC, 2 izolované skupiny (4 body na společný vodič) 0,3A/bod, vyjímatelná svorkovnice
<b>C0-04TRS</b>	4 výstupů 6-27 VDC, 6-240 VAC, relé, 4 izolované skupiny (1 bod na společný vodič), 7A/bod, vyjímatelná svorkovnice
<b>C0-08TR</b>	8 výstupů 6-27 VDC, 6-240 VAC, relé, 2 izolované skupiny (4 body na společný vodič), 1A/bod, vyjímatelná svorkovnice
<b>C0-16CDD1</b>	8 vstupů 24 VDC se společným plus nebo mínus, 1 izolovaný vodič, 8 výstupů 5-27VDC 0,1A/bod, se společným mínus, vyjímatelná svorkovnice
<b>C0-16CDD2</b>	8 vstupů 24 VDC se společným plus nebo mínus, 1 izolovaný vodič, 8 výstupů 12-24 VDC, 0,1A/bod, se společným plus, vyjímatelná svorkovnice
<b>C0-08CDR</b>	4 vstupů 12-24 VDC se společným plus nebo mínus, 4 výstupy 6-24 VDC 6-240 VAC, 1A/bod, vyjímatelná svorkovnice

## **Analogové vstupy a výstupy**

- C0-04AD-1** 4 kanálový vstupní proudový analogový modul, 13 bit rozlišení, rozsah 0-20 mA
- C0-04AD-2** 4 kanálový vstupní napěťový analogový modul, 13 bit rozlišení, rozsah 0-10 V
- C0-04DA-1** 4 kanálový výstupní proudový analogový modul, 12 bit rozlišení, rozsah 4-20 mA, požadováno externí napájení 24 VDC
- C0-04DA-2** 4 kanálový výstupní napěťový analogový modul, 12 bit rozlišení, rozsah 0-10 V, požadováno externí napájení 24 VDC
- C0-4AD2DA-1** 4 kanálový vstupní proudový a 2 kanálový výstupní proudový analogový modul, 13bit rozlišení (vstup), rozsah 0-20 mA, 12 bit rozlišení (výstup), rozsah 4-20 mA, požadováno externí napájení 24 VDC
- C0-4AD2DA-2** 4 kanálový vstupní napěťový a 4 kanálový výstupní napěťový analogový modul, 13bit rozlišení (vstup), rozsah 0-10 V, 12 bit rozlišení (výstup), rozsah 0-10 V, požadováno externí napájení 24 VDC
- C0-04RTD** 4 vstupy pro odporové vysílače, rozsah 0,1 °C/°F, rozsah 0-3125 Ohm, Pt100, Pt1000, JPt100, Cu10, Cu25, Ni120.
- C0-04THM** 4 vstupy pro termočlánky J, K, E, R, S, T, B, N, C, 16 bit rozsah (0,1 °C/°F) nebo napěťový vstup v rozsahu -156,25 mV až 1,25 V

## **Poznámky**