

Algoritmy riadenia modelu dopravníka

Körösi Ladislav · Elektrotechnika

10.11.2014



Cieľom článku je návrh a overenie jednoduchých algoritmov riadenia pre model dopravníka od spol. Fischertechnik z rodiny priemyselných modelov. Článok nadväzuje na články zaoberajúce sa základnou inštrukčnou sadou softvéru RsLogix 5000, ktorá je potrebná na pochopenie nižšie navrhnutých algoritmov. Článok je vhodný aj pre

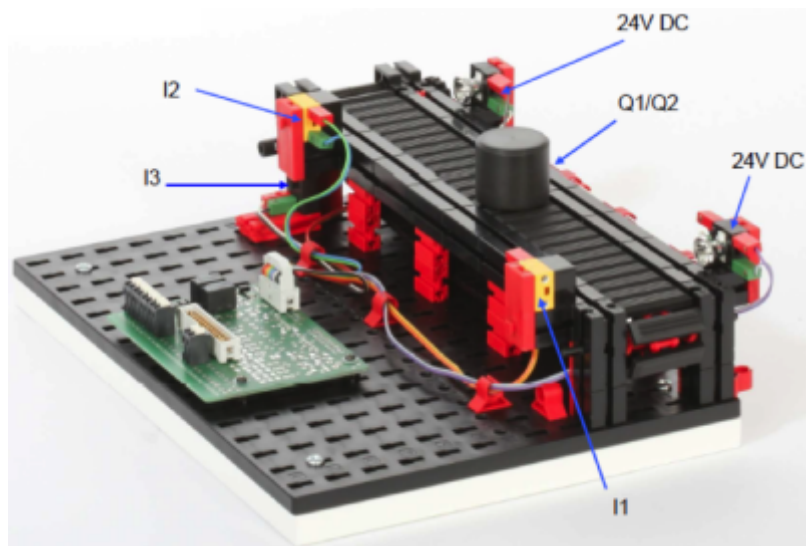
študentov na výučbu základov programovacích techník programovateľných logických automatov (PLC).

Úvod

V dnešnej dobe sa navrhované algoritmy čoraz viac overujú simulačne pomocou softvérových nástrojov na zjednodušených (ale aj zložitých) modeloch. Pri nasadení navrhnutého algoritmu môže vzniknúť množstvo rizík a problémov. Použitie simulácie umožňuje vopred overiť, následne analyzovať správanie sa systému a zároveň nahliadnuť aj na jeho budúce správanie sa a tak v predstihu odladiť prípadné problémy. Požiadavky na simuláciu vznikajú hlavne pri zmene technológie, ale aj pri návrhu nových výrobných postupov. Cieľom príspevku je navrhnúť a overiť algoritmy riadenia pre zjednodušený model dopravníka s použitím reálneho PLC.

Model dopravníka

Model dopravníka (Obr. 1) je produktom spoločnosti Fischertechnik GmbH a slúži na použitie v pedagogickom procese ako aj na výskumné účely v kombinácii s inými modelmi. Jedná sa o model z rady Industry v 24V prevedení, ktorý je pripravený na pripojenie k PLC systémom.



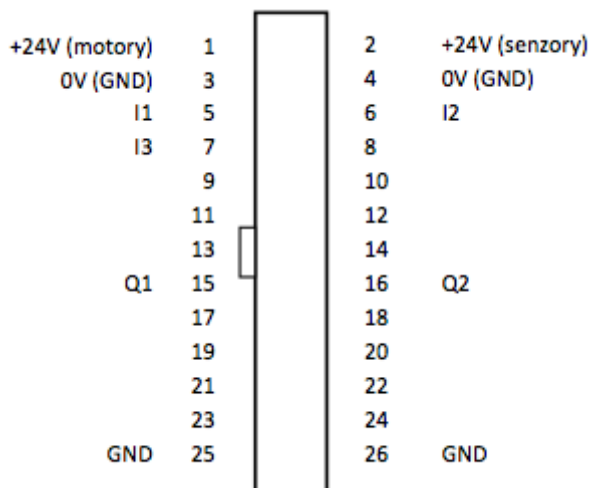
Obr. 1 Model dopravníka [1]

Zoznam všetkých signálov s číslami svoriek je v tab. 1. Snímače I1 a I2 sú optické snímače, ktoré sa skladajú z dvoch častí, z vysielacza a z prijímača. Oba snímače sú rozpínacie, t.j. ak je výrobok na dopravníku medzi vysieláčom a prijímačom, tak sa optický signál preruší a v PLC sa zaznamená signál FALSE (logická 0). Pri neobsadenej pozícii snímače majú stav TRUE. Snímač I3 je kontaktný snímač, ktorý je zopínaný hriadelom motora dopravníka. Počas jednej otáčky sa snímač zopne 4-krát. Snímač môže slúžiť ako inkrementálny snímač (IRC) pre polohovanie dopravníka a aj ako spätná väzba o chode dopravníka. Výstupy Q1 a Q2 ovládajú smer pohybu dopravníka a nemali by byť zopnuté v rovnakom čase. Nastavením Q1 na hodnotu TRUE sa spustí dopravník vpred, t.j. od snímača I1 smerom k I2. Nastavením Q2 na hodnotu TRUE sa spustí dopravník vzad.

Tab. 1 Zoznam signálov modelu

Číslo svorky	Funkcia	Vstup/Výstup
1	Napájanie (+) motory	24V DC
2	Napájanie (+) senzory	24V DC
3	Napájanie (-)	0V
4	Napájanie (-)	0V
5	Fotobunka na začiatku dopravníka	I1
6	Fotobunka na konci dopravníka	I2
7	Impulzný snímač otáčok	I3
15	Motor dopravníka - chod vpred	Q1
16	Motor dopravníka - chod vzad	Q2

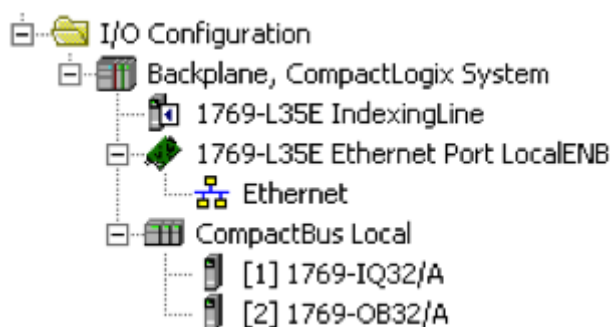
Prepojenie modelu s PLC je možné realizovať aj pomocou 26 pinového konektora (obr. 2). Pri overení navrhnutých algoritmov boli signály pripojené k PLC pomocou tohto konektora. Svorky ako aj konektor možno vidieť na obr. 1 napravo od modelu dopravníka.



Obr. 2 Pripojenie signálov pomocou konektora

Riadiaci systém

Ako riadiaci systém pre návrh a overenie algoritmov bol zvolený PLC CompactLogix od spol. Rockwell Automation, s ktorým sa stretávajú študenti 3. ročníka odboru priemyselná informatika na FEI STU. Riadiaci systém je vhodný pre stredné aplikácie aj keď má v sebe výkon ControlLogix-u v kompaktnom prevedení. Zostava sa skladá z CPU 1769-L35E, diskretného vstupného modulu 1769-IQ32/A a diskretného výstupného modulu 1769-OB32/A. Vstupné signály modulu 1769-IQ32/A môžu byť typu (zapojenia) Sink aj Source a preto vyhovujú pre zvolený model dopravníka. Vytvorená hardvérová konfigurácia PLC v softvéri RsLogix 5000, ktorý slúži na programovanie, konfiguráciu a diagnostiku PLC systémov od spol. Rockwell Automation je na obr. 3.



Obr. 3 Hardvérová konfigurácia PLC

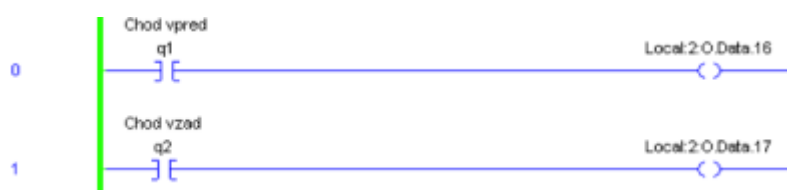
Algoritmy riadenia

Pre návrh a overenie algoritmov boli vytvorené tagy (premenné) s rovnakým názvom ako boli zadané v tab. 1 (I1, I2, I3, Q1 a Q2). Na obr. 4 sú parametrami inštrukcií XIC a XIO fyzické adresy digitálnych vstupov, ku ktorým sú pripojené signály zo snímačov modelu. Pri návrhu boli signály z optických snímačov I1 a I2 zmenené z rozpínacieho na spínacie (obr. 4 - rebríky č. 0 a 1). Dôvodom zmeny je ľahšia práca s priamym signálom zo snímačov ako s rozpínacími stavmi. Snímač I3 bol mapovaný priamo bez zmeny.



Obr. 4 Mapovanie vstupov modelu dopravníka

Na obr. 5 je zobrazená rebríková schéma mapovania výstupných signálov Q1 a Q2 na fyzické adresy, ku ktorým boli pripojené signály ovládania motorov. Dané mapovanie vstupov a výstupov umožňuje použiť vlastných symbolických názvov premenných ako aj možnosť vypnutia mapovania a tak umožniť simuláciu vstupov a výstupov s navrhnutým algoritmom.



Obr. 5 Mapovanie výstupov modelu dopravníka

Algoritmus - Preprava jedného výrobku

Zadanie: Spustiť dopravník vpred, ak snímač I1 detekuje výrobok na dopravníku. Zastaviť dopravník, ak sa výrobok presunul z I1 na I2, t.j. snímač I2 detekuje výrobok. Pre jednoduchosť uvažujme jeden výrobok na dopravníku.

Nadobudnuté stavy dopravníka: spustený vpred, zastavený vpred.

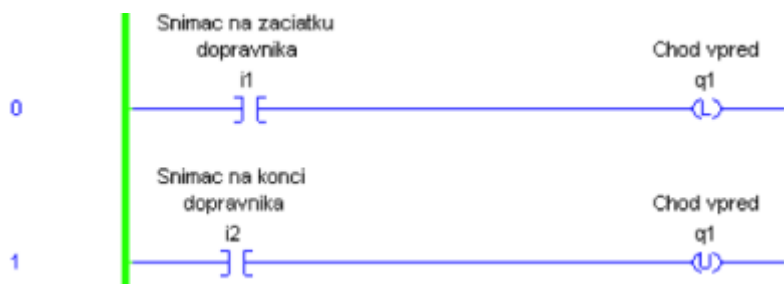
Prechody medzi stavmi: snímač I1 aktivuje prechod do stavu spustený vpred a snímač I2 aktivuje prechod do stavu zastavený vpred.

Existuje veľa možností pre návrh algoritmu riadenia. V článku si ukážeme spustenie dopravníka pomocou inštrukcií OTE a pomocou kombinácie inštrukcií OTL a OTU. Taktiež bude naznačený princíp automatického a manuálneho ovládania dopravníka.

1. Riešenie

Prvé riešenie je na obr. 6. Pomocou inštrukcie OTL sa signálom zo snímača I1 nastaví výstup Q1 na chod vpred. Ak výrobok opustí začiatok dopravníka, tak signál zo snímača bude FALSE, ale výstup bude naďalej zopnutý vďaka inštrukcie OTL. V momente, keď bude prvýkrát zopnutý snímač I2, pohyb dopravníka sa vypne pomocou inštrukcie OTU. Za predpokladu, že je na dopravníku jeden výrobok, tak sa tento výrobok z I1 premiestni na I2. Ak by sme uvažovali viac výrobkov, tak takýto algoritmus by pri zaregistrovaní prvého výrobku snímačom I1 spustil chod vpred a postupným pridávaním výrobkov na začiatok dopravníka by bol dopravník naďalej v chode. Dopravník by zastal opäť v momente zopnutia snímača I2. V prípade, že by boli polohy I1 a I2 obsadené, tak po odstránení výrobku z polohy I2 by sa dopravník automaticky spustil. Ak by bol posledný výrobok pred alebo za snímačom I1, tak by sa

dopravník už nespustil.



Obr. 6 Ovládanie modelu dopravníka s využitím inštrukcií OTL a OTU

2. Riešenie

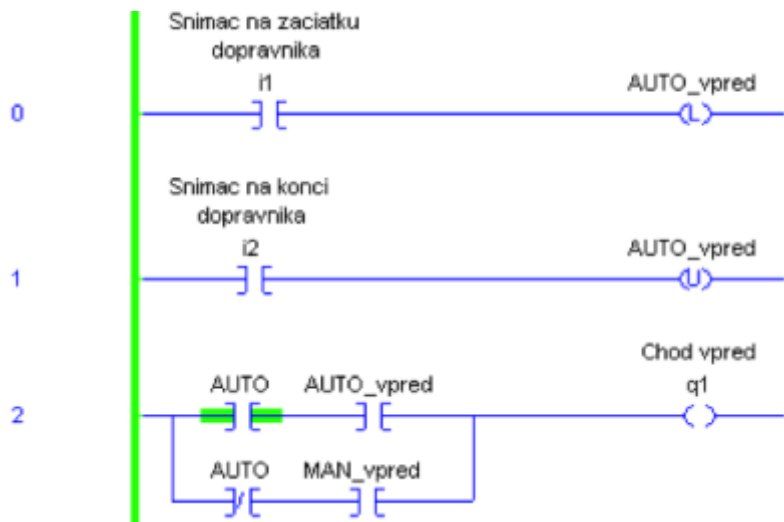
Druhé riešenie je založené na priamom zápise výstupu s využitím inštrukcie OTE (obr. 7). Ak platí podmienka, že $I1 = \text{TRUE}$ a $I2 = \text{FALSE}$, t.j. výrobok sa nachádza na začiatku dopravníka (pred snímačom I1), tak sa zopne výstup Q1. Na začiatku rebríka je podmienka OR signálov I1 a Q1, ktorá slúži ako „samodrž“, keď výrobok opustí polohu I1. Chod dopravníka následne zabezpečí podmienka Q1 AND NOT I2. Dopravník zastane po zaregistrovaní výrobku snímačom I2, t.j. rozopne sa podmienka pred OTE. Algoritmus sa bude správať rovnako pre viacero výrobkov, ako to už bolo uvedené v 1. riešení.



Obr. 7 Ovládanie modelu dopravníka s využitím inštrukcie OTE

3. Riešenie

Poslednou ukážkou je naznačenie realizácie automatického a manuálneho režimu riadenia modelu dopravníka (obr. 8). V praxi sa často využívajú hlavne vyššie spomenuté režimy, ale okrem nich existujú aj ďalšie ako lokálne alebo vzdialené riadenie, núdzový režim atď. Uvedený algoritmus vychádza z 1. riešenia. Prvá modifikácia spočíva v zavedení pomocného tagu `AUTO_vpřed` dátového typu `BOOL`, ktorý bude ovládať chod dopravníka v automatickom režime. Druhá modifikácia je zobrazená v rebríku č. 2. Horná časť podmienky OR realizuje zopnutie výstupu Q1 v automatickom režime (ak platí $\text{AUTO} = \text{TRUE}$ a $\text{AUTO_vpřed} = \text{TRUE}$) a dolná vetva v manuálnom režime (ak platí $\text{AUTO} = \text{FALSE}$ a $\text{MAN_vpřed} = \text{TRUE}$). Prepnutie režimov sa v praxi rieši viacerými spôsobmi, napr. prepnutím prepínača, zatlačením tlačidla na vizualizácii (na operátorskom paneli) príp. vyvolaním stavu, ktorý automaticky prepne režim do manuálneho chodu, atď. Chod vpred v manuálnom režime (`MAN_vpřed`) je najčastejšie realizovaný tlačidlom alebo prepínačom.



Obr. 8 Ovládanie modelu dopravníka v automatickom a manuálnom režime

Algoritmus - Preprava viacerých výrobkov

Zadanie: Spustiť dopravník vpred, ak snímač I1 detekuje výrobok na dopravníku a zastaviť dopravník, ak sa výrobok presunul z I1 na I2, t.j. snímač I2 detekuje výrobok. Uvažujeme ľubovoľný počet výrobkov s cieľom dopraviť ich pred snímač I2.

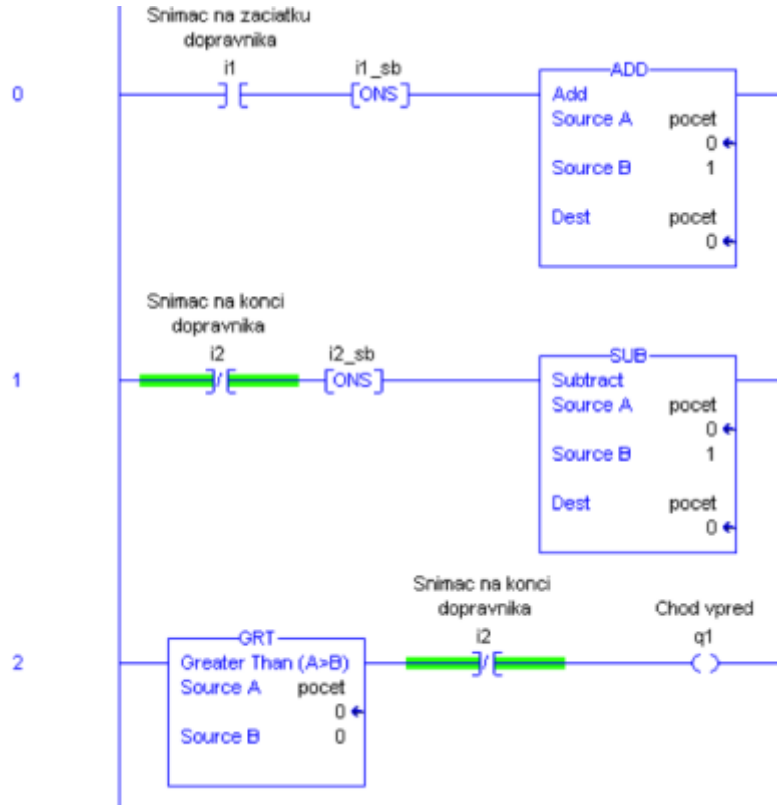
Nadobudnuté stavy dopravníka: Spustený vpred, zastavený vpred.

Prechody medzi stavmi: Snímač I1 zisťuje počet výrobkov na dopravníku. Nenulový počet výrobkov aktivuje prechod do stavu spustený vpred a snímač I2 aktivuje prechod do stavu zastavený vpred.

Opäť je na výber viacero riešení, z ktorých vyberáme nasledovné dve.

Riešenie 1

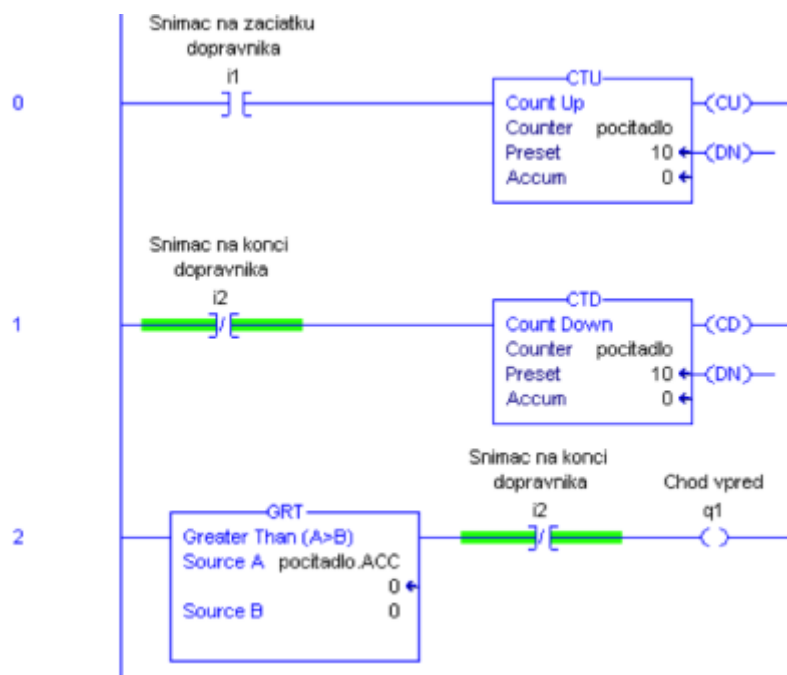
Prvý príklad určovania aktuálneho počtu výrobkov je založený na inštrukcii ADD (obr. 9). Pri každej nábežnej hrane I1 sa zvýši počet výrobkov na dopravníku o 1. Aktuálny počet je uložený v tagu pocet dátového typu DINT. Zníženie počtu výrobkov nastane pri odobratí výrobku zo snímača I2, t.j. pri dobežnej hrane signálu I2. Dobežná hrana je detekovaná inštrukciou ONS pre detekciu nábežnej hrany s negovaným vstupným signálom. Podmienka na chod dopravníka je splnená pri nenulovom počte výrobkov ($pocet > 0$) a neobsadenosti snímača I2 výrobkom. Tagy i1_sb a i2_sb slúžia ako pomocné premenné (operands pre inštrukciu ONS) na detekciu nábežnej resp. dobežnej hrany signálov.



Obr. 9 Riadenie dopravníka od nenulového počtu výrobkov s využitím inštrukcií ADD a SUB

Riešenie 2

V druhom riešení sú inštrukcie ADD a SUB nahradené počítadlami CTU a CTD (obr. 10). Tieto inštrukcie už nepotrebujú inštrukciu ONS na detekovanie nábežnej hrany, lebo ju majú priamo implementovanú vo funkčnom bloku počítadla. Prednastavená hodnota v tomto prípade nehrá žiadnu dôležitú úlohu. Dôležitá je aktuálna hodnota počítadla (pocitadlo.ACC), ktorá podobne ako v riešení 1 je jednou z podmienok pre spustenie dopravníka.



Obr. 10 Riadenie dopravníka od nenulového počtu výrobkov s využitím inštrukcií CTU

a CTD

Ako už bolo spomenuté, diskretný udalostný systém môže nadobúdať rôzne stavy. Tieto stavy môžu, ale nemusia blokovať spustenie určitej sekvencie príkazov. V nasledujúcich dvoch príkladoch si ukážeme dva blokujúce stavy a jeden neblokujúci stav chodu dopravníka.

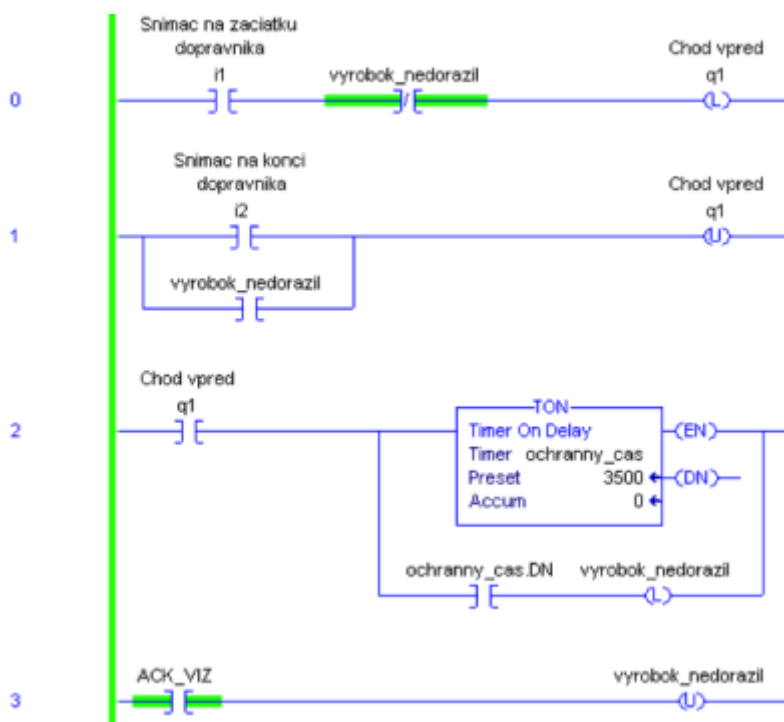
Algoritmus - Zablockovanie chodu dopravníka

Zadanie: Spustiť dopravník vpred, ak snímač I1 detekuje výrobok na dopravníku a zastaviť dopravník, ak sa výrobok presunul z I1 na I2, t.j. snímač I2 detekuje výrobok. Pre jednoduchosť opäť uvažujme len jeden výrobok na dopravníku. Ak výrobok nie je do stanoveného času detekovaný snímačom I2, tak sa chod dopravníka vypne.

Nadobudnuté stavy dopravníka: Spustený vpred, zastavený vpred, blokovaný.

Prechody medzi stavmi: Snímač I1 aktivuje prechod do stavu spustený vpred, snímač I2 aktivuje prechod do stavu zastavený vpred a blokácia aktivuje prechod do stavu zastavený vpred.

Riešenie vychádza z príkladu s jedným výrobkom a použitia inštrukcií OTL a OTU (obr. 6). Na obr. 11 sa v rebríku č. 2 počíta čas presunu výrobku z I1 na I2. Po viacerých meraniach času prechodu výrobku z I1 na I2 bol ochranný čas stanovený na 3,5 sekundy, čo zahŕňa aj rezervu, pre prípad kedy sa výrobok môže dotýkať okraju dopravníka (čím je jeho pohyb spomalený). Po uplynutí ochranného času sa nastaví tag výrobok_nedorazil dátového typu BOOL na TRUE, čo následne spôsobí zastavenie chodu dopravníka (OR vetva v rebríku č. 1) a blokáciu opätovného spustenia dopravníka (podmienka AND v rebríku č. 0). V poslednom rebríku sa realizuje odblokovanie dopravníka signálom z vizualizácie (ACK_VIZ).



Obr. 11 Blokácia chodu dopravníka od ochranného času

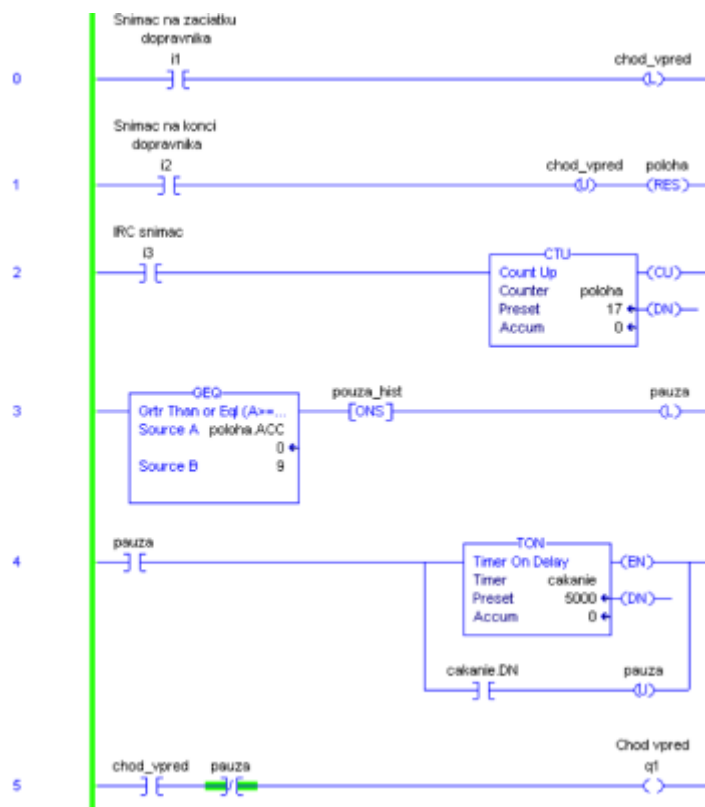
Algoritmus - Pozastavenie chodu dopravníka

Zadanie: Spustiť dopravník vpred, ak snímač I1 detekuje výrobok na dopravníku a zastaviť dopravník, ak sa výrobok presunul z I1 na I2. Bude uvažovaný len jeden výrobok na dopravníku. Ak výrobok prešiel polovicu dráhy na dopravníku, tak zastane a bude sa simulovať opracovanie, ktoré trvá preddefinovaný čas.

Nadobudnuté stavy dopravníka: Spustený vpred, zastavený vpred, pozastavený.

Prechody medzi stavmi: Snímač I1 aktivuje prechod do stavu spustený vpred, snímač I2 aktivuje prechod do stavu zastavený vpred a pozastavenie aktivuje prechod do stavu zastavený vpred.

Riešenie vychádza už z opísaného príkladu pre ovládanie dopravníka pomocou inštrukcií OTL a OTU v automatickom režime. Pri polohovaní sa využíva snímač I3, ktorý pri každej nábežnej hrane signálu zvýši aktuálnu hodnotu počítadla poloha (obr. 12 - rebrík č. 2). Z viacerých meraní bol stanovený maximálny počet impulzov (celková dĺžka trasy) na 17. Po dosiahnutí polovice dráhy sa aktivuje stav pauza (rebrík č. 3), ktorý spúšťa simuláciu opracovania výrobku na 5 sekúnd (rebrík č. 4). Po uplynutí prednastaveného času sa deaktivuje stav pauza. Aktivovaný stav pauza pozastaví spustenie dopravníka, ktorý je ovládaný priamym signálom s využitím inštrukcie OTE (rebrík č. 5). Detekciou výrobku snímačom I2 dopravník zastane a zároveň sa vynuluje aj poloha výrobku (inštrukcia RES v rebríku č. 1). Vynulovanie polohy by bolo možné realizovať aj v rebríku č. 0 pri vložení výrobku. Záleží na tom na aké iné účely sa informácia o polohe výrobku bude ďalej využívať.



Obr. 12 Polohovanie modelu dopravníka

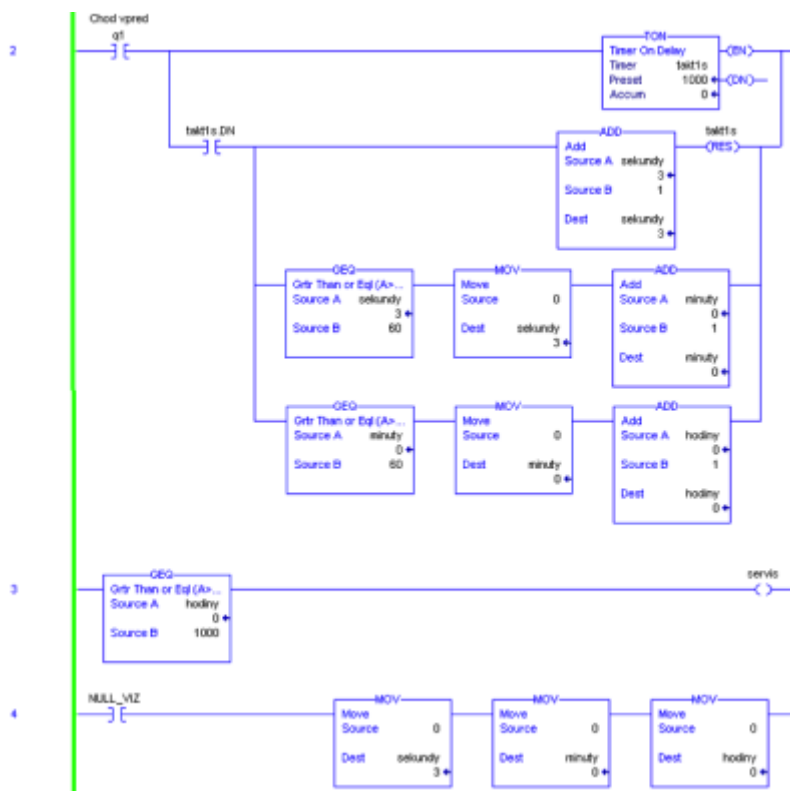
Algoritmus - Prevádzkový čas chodu dopravníka

Zadanie: Spustiť dopravník vpred, ak snímač I1 detekuje výrobok na dopravníku a zastaviť dopravník, ak sa výrobok presunul z I1 na I2, t.j. snímač I2 detekuje výrobok. Pre jednoduchosť opäť uvažujme len jeden výrobok na dopravníku.

Nadobudnuté stavy dopravníka: Spustený vpred, zastavený vpred, servis.

Prechody medzi stavmi: Snímač I1 aktivuje prechod do stavu spustený vpred a snímač I2 aktivuje prechod do stavu zastavený vpred.

V praxi sa veľmi často používa počítanie prevádzkového času zariadení alebo ich častí (napr. motorov, čerpadiel a pod.). Údaj môže byť len informatívny napr. pre upozornenie potreby vykonania potrebného servisu, revízie a pod., ale aj napr. na prepínanie chodu paralelne zapojených čerpadiel z dôvodu dosiahnutia vyrovnaného prevádzkového času. Na obr. 13 je príklad určovania prevádzkového času chodu motora vpred bez samotného spustenia a zastavenia, ktoré už bolo vyššie opísané. Rozšírením by bolo pridanie OR vetvy na chod vzad, ktorý sa v uvedených algoritmoch zatiaľ nevyužíval. Prevádzkový čas sa ukladá do troch premenných: sekundy, minúty a hodiny dátových typov DINT. Po dosiahnutí 1000 hodín sa aktivuje stav servis. V poslednom rebríku sa vykoná vynulovanie motohodín signálom z vizualizácie (NULL_VIZ).



Obr. 13 Signalizácia potreby servisu motora dopravníka

Záver

V článku boli prezentované jednoduché algoritmy riadenia modelu dopravníka, ktoré môžu slúžiť ako vhodný základ na tvorbu komplexnejších riešení riadenia. Zaujímavú časť algoritmov riadenia tvoria diagnostické funkcie, ktoré sú neoddeliteľnou súčasťou každej aplikácie. Spolu so signálmi pre vizualizáciu a ich integrácie do aplikácie tvoria približne 50-75% celkového zdrojového kódu aplikácie. V nasledujúcom článku je

plánovaná ukážka komplexnejších algoritmov na báze detekcie celkov, ktoré majú spoločné vlastnosti a tak umožňujú vytvárať funkčné bloky (objekty).

Podakovanie

Tento príspevok vznikol vďaka podpore Nadácie Tatra banky - E-Talent z Nadácie Tatra banky č. 2013et030.

Literatúra

1. http://www.fischertechnik.de/en/desktopdefault.aspx/tabid-24/41_read-58/usetemplate-2_column_pano/
2. <http://literature.rockwellautomation.com>
3. Interné prednášky z predmetu Softvér riadiacich systémov na FEI STU Bratislava
4. <http://www.fischertechnik.de/home/produkte/industrie.aspx>
5. http://www.christiani.de/index.php/cPath/611_5800_8650

Spoluautormi článku sú Jana Paulusová, Mária Dúbravská
