

AUTOMA

časopis pro automatizační techniku



www.automa.cz
Cena 52 Kč

11

LISTOPAD
2012



**Programový systém
Control Web**

**Systém
strojového vidění
VisionLab**

**Jednotky
průmyslových
vstupů a výstupů
DataLab**

**Průmyslové
počítače
DataLab PC**

**Digitální
kamery
DataCam**

**Osvětlovací
jednotky
DataLight**

ZJEDNODUŠTE SI SVOU PRÁCI



Co bude k vidění na veletrhu SPS/IPC/Drives 2012 v Norimberku
Moduly pro řízení pohonů
Trendy v oblasti PID regulace
Imerzní virtuální realita pro simulaci a výcvik operátorů
Využití grafických procesorů v systémech strojového vidění



9 771210 959006

11

Plánování úloh v systémech RT - II: neperiodické úlohy

Druhý díl seriálu nepřímě navazuje na [6], kde byly představeny základní mechanismy přiřazování statických (pevných) priorit periodickým úlohám tzv. reálného času (*real-time*, RT, úlohy RT), a na [5], kde bylo zmíněno, že vedle periodických úloh RT mohou existovat i úlohy neperiodické. Mezi neperiodickými úlohami RT jsou obvykle rozlišovány úlohy *aperiodické* (intervaly příchodů úloh tohoto typu jsou předem neznámé) a úlohy *sporadické* (je u nich známa nejkratší doba mezi příchody). V tomto článku jsou představeny základní mechanismy *společného* (tzv. *hybridního*) plánování množin periodických a neperiodických úloh (*hybrid/joint scheduling*) [1], [2]. Výklad předpokládá u čtenáře znalost modelu úlohy RT a jeho symboliky podle [5].

Vliv aperiodických úloh na chod systému

Ještě před započatím popisu mechanismů je nutné zdůraznit, že zejména aperiodické úlohy a s velkou prioritou představují problém z pohledu nekontrolovatelného zvýšení zátěže procesoru, na kterém, vedle aperiodických úloh τ_1, \dots, τ_k (obvykle s odezvami typu *soft*), mohou běžet periodické či sporadické úlohy $\tau_{k+1}, \dots, \tau_{k+m}$ s odezvami typu *firm* či *hard*. Celkové využití procesoru U společnou množinou úloh $\Gamma = \{\tau_1, \dots, \tau_k, \tau_{k+1}, \dots, \tau_{k+m}\}$ je dáno součtem u_i využití procesoru dílčími úlohami z Γ [5], tedy

$$U = \sum_{\tau_i \in \Gamma} u_i = \sum_{\tau_i \in \Gamma} \frac{C_i}{T_i} \quad (1)$$

Pro $U = 1$ je součet výpočetních požadavků kladených na procesor roven výpočetním možnostem procesoru (tj. procesor je zcela vytížen), a jakýkoliv další nárůst U tudíž povede k přetížení procesoru, následnému nedodržení časových omezení některých úloh, či dokonce k náhlému kolapsu systému. Z podstaty aperiodických úloh plyne, že podněty vedoucí k jejich spuštění mohou přicházet v nekonečně krátkých intervalech, tj. $T_i \rightarrow 0$, což vede na $U \rightarrow \infty$. Každý dobře navržený systém RT tedy musí být připraven vhodně reagovat na situaci, kdy počet podnětů v čase t nebude ohraničen shora; přitom musí zajistit, aby hodnota U klesla nejen pod 1, ale i pod hodnotu $U^*(M)$, garantující plánovatelnost úloh při použití daného mechanismu přiřazování priorit (M).

Následující text bude zaměřen na popis vybraných mechanismů společného plánování periodických a aperiodických úloh. Důvody, proč nebudou představeny mechanismy zaměřené čistě na plánování aperiodických či sporadických úloh, jsou tyto:

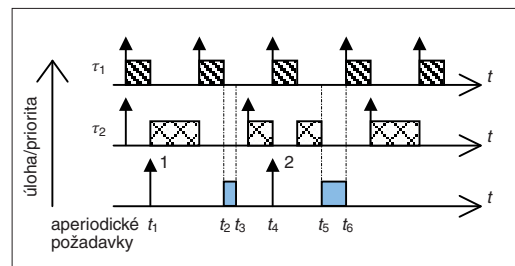
- sporadické úlohy vykazují, vzhledem k omezenosti dob jejich příchodů v rámci daného intervalu, podobné vlastnosti jako periodické úlohy,
- je-li mechanismus schopen společně plánovat množiny periodických a aperiodických úloh, je schopen plánovat tyto množiny

ny i tehdy, je-li podmnožina periodických úloh prázdná.

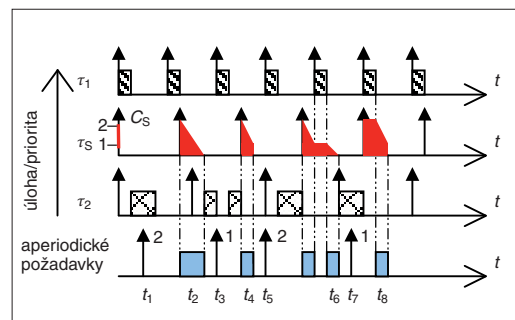
Pro jednotný popis mechanismů si označme množinu periodických (aperiodických) úloh symbolem Γ_P (Γ_A) a činitel využití procesoru úlohami z Γ_P symbolem U_P .

Plánování v pozadí (BS)

Jako první z mechanismů společného plánování bude představen mechanismus *plánování v pozadí* (*Background Scheduling* – BS).



Obr. 1. Ilustrace k plánování v pozadí (BS)



Obr. 2. Ilustrace k vyzývacímu serveru (PS)

Tento mechanismus je založen na následujícím principu. Úlohám z Γ_A je přidělen procesor pouze v intervalech, kdy jej nevyžadují úlohy z Γ_P . Úlohy z Γ_P (Γ_A) pak běží v tzv. popředí (pozadí). K realizaci BS jsou obvykle třeba dvě fronty úloh připravených k běhu (jedna pro úlohy z Γ_P , druhá pro ty z Γ_A), úlohy každé z front mohou být plánovány různými mechanismy. Pro správnou činnost

BS je nutné přidělit úlohám priority tak, aby priority úloh z Γ_P byly významnější než priority úloh z Γ_A , tj. aby pro všechny úlohy τ_i z Γ_P a τ_j z Γ_A platilo $P_i > P_j$ kde P_i, P_j jsou priority úloh i, j .

P_j . Z jednoduchosti realizace BS plyne jeho hlavní nedostatek: doba odezvy aperiodických úloh R_A roste s hodnotou U_P blížící se k 1 do nekonečna ($R_A \rightarrow \infty$ pro $U_P \rightarrow 1$). Ale k přetížení procesoru v důsledku nadměrného počtu aperiodických úloh jistě nedojde.

Ilustrace k BS je na obr. 1 – je zřejmé, že od doby vzniku aperiodických požadavků, tj. požadavku č. 1 (2) příchozího v t_1 (t_4) a vyžadujícího 1 (2) výpočetní jednotku(y) na procesoru, jsou prováděny až v době, kdy procesor není využíván úlohami $\tau_1, \tau_2 \in \Gamma_P$.

Servery úloh

V dalším textu se z existujících mechanismů společného plánování zaměříme na tzv. *servery úloh*. Serverem úloh bude chápána periodická úloha τ_S (s periodou T_S) zařazená do systému speciálně za účelem obsluhy aperiodických podnětů. Doba, kterou má server vyhrazeno pro obsluhu aperiodických podnětů, nazvěme *kapacita serveru*; bude označována symbolem C_S . Server bude značen symbolem $\tau_S(T_S, C_S)$, využití procesoru serverem $U_S = C_S/T_S$.

Vyzývací server (PS)

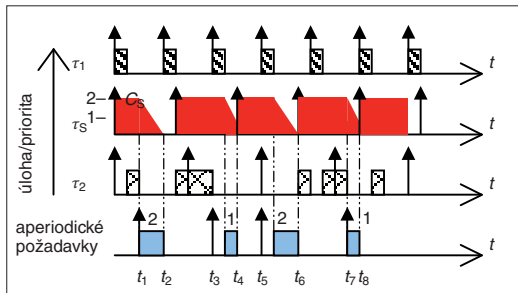
Jako první ze serverů si představme tzv. *vyzývací server* (*Polling Server* – PS). Jeho princip spočívá v tom, že aperiodické úlohy jsou plánovány stejným mechanismem jako periodické. Je-li serveru přidělen procesor v čase t_S a existují-li neobsloužené aperiodické požadavky, server obsluží požadavky až do počtu souhrnně nevyžadujícího k obsluze více než C_S jednotek času na procesoru; obsluhy nad tento limit jsou odsouvány do další T_S . Jestliže v t_S neexistují neobsloužené aperiodické požadavky, server se dobrovolně vzdává přidělení procesoru i nedočerpané kapacity, a to až do následující T_S . Realizace PS spočívá v přidání úlohy $\tau_S(T_S, C_S)$ do existující množiny Γ_P , přičemž je nutné pečlivě zvážit volbu hodnot T_S a C_S a stanovit postup, na jehož základě bude procesor přidělen aperiodickým úlohám v případě, že v čase t_S existuje několik neobsloužených aperiodických poža-

davků. Předností PS (oproti BS) je nastavitelnost T_S a C_S spolu se skutečností, že s příchodem nové T_S je C_S zcela obnovena (tj. bez ohledu na to, zda a popř. jak byla v minulosti konzumována). Nedostatkem je, že přijde-li aperiodický požadavek krátce poté, co se τ_S vzdá procesoru, nebude obsluha takového požadavku zahájena dříve než v následující T_S . Z toho plyne pro stanovování hodnot D_i pro $\tau_i \in \Gamma_A$ plánovaných pomocí PS podmínka $D_i \geq T_S(1 + \lceil C_i/T_S \rceil)$. Množina úloh Γ je plánovatelná s využitím PS a mechanismu M přiřazování priorit, platí-li $U_P + U_S \leq \nu(M)$.

Princip PS je ilustrován na obr. 2. Kratší doby odezvy aperiodických požadavků je dosaženo díky prodloužení doby odezvy úlohy $\tau_2 \in \Gamma_P$. Změna C_S v čase je vyobrazena v rámci části plánu pro τ_S . Klesající hodnota C_S signalizuje, že procesorem zpracovává aperiodické úlohy.

Odkládací server (DS)

Ve snaze zmenšit zpoždění začátku provádění, a tím i průměrnou dobu odezvy aperiodických úloh více, než umožňují PS, lze PS modifikovat na tzv. odkládací server (*Deferable Server* – DS), pracující následovně. S příchodem T_S je (jako u PS) zcela obnovena C_S . Není-li na začátku T_S detekován neobslužený aperiodický požadavek, DS se sice vzdává procesoru, ale ne své nevyčerpané kapacity – tu si až do příchodu další T_S ponechává pro obsluhu případných nově vzniklých aperiodických požadavků. Aperiodické požadavky tedy mohou být pomocí DS obsluženy (za předpokladu, že kapacita DS nebyla



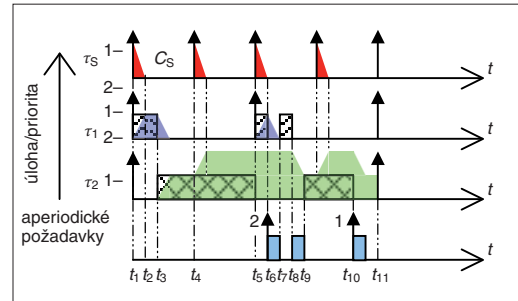
Obr. 3. Ilustrace k odkládacímu serveru (DS)

vyčerpana) i mezi časem, kdy se DS vzdá procesoru, a časem příchodu nové T_S . Oproti PS se tedy zkrátila doba odezvy aperiodických požadavků, avšak na úkor odezvy periodických úloh s prioritou nižší než prioritou DS – pro zajímavost je možné porovnat plány z obr. 2 (plán pro PS) a obr. 3 (plán pro DS).

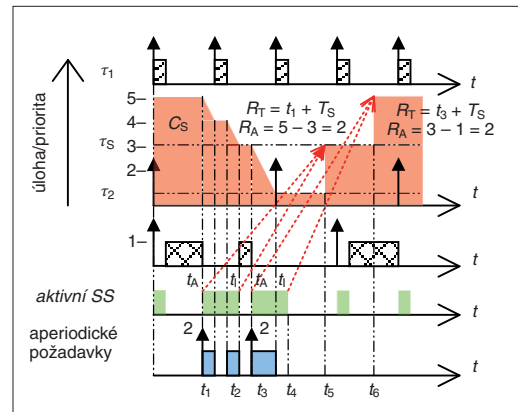
Server s výměnou priorit (PE)

Bude-li DS navíc spravovat informaci o tom, kterým úlohám byl přidělen (z pohledu serveru zapůjčen) procesor poté, co se jej DS z důvodu absence aperiodických požadavků vzdal, získá se *server s výměnou* (směnou, záměnou) *priorit* (*Priority Exchange Server*

– PE). Kapacita serveru se po jeho vzdání se procesoru „neztrácí“, ale zapůjčuje úlohám běžícím v době, kdy je C_S nenulová. Informaci o těchto zápůjčkách udržuje PE pro každou z úloh tak, že za běhu úlohy se její zápůjčka s každou jednotkou času zvyšuje o 1, nejmý-



Obr. 4. Ilustrace k serveru s výměnou priorit (PE)



Obr. 5. Ilustrace ke sporadickému serveru (SS)

še však do celkové výše zápůjček C_S . S příchodem nové periody úlohy se její zápůjčka nuluje. Vznikne-li aperiodický požadavek, PE zjistí úlohu τ s nejvyšší prioritou na množině úloh s nenulovými zápůjčkami. Tuto úlohu přeruší a začne, na její prioritní úrovni, běžet do dokončení obsluhy požadavku, nejdéle však po dobu rovnou výši její zápůjčky. Nebyla-li obsluha požadavku dokončena (tj. zápůjčka byla nedostatečná k dokončení obsluhy), proces hledání zápůjček se opakuje. Jestliže již zápůjčky neexistují, musí PE čekat na doplnění C_S až do příchodu nové T_S . Oproti DS tedy aperiodické úlohy mohou v rámci T_S využít procesor i na dobu delší než C_S jednotek (tj. běžely-li předtím periodické úlohy ze zápůjček od PE), avšak po dobu omezenou součtem zápůjček úloh, které si čas procesoru dříve od PE zapůjčily a dosud jej PE „nevrátily“. Spolu s procesorem jsou mezi úlohami předávány (děděny) i zápůjčky.

Princip PE je ilustrován na obr. 4, kde zápůjčky úloh τ_1, τ_2 jsou značeny modrou, popř. zelenou barvou na pozadí části plánu těchto úloh. V čase t_1 neexistuje aperiodický požadavek, a τ_1 tedy začíná (do t_2) běžet.

Princip PE je ilustrován na obr. 4, kde zápůjčky úloh τ_1, τ_2 jsou značeny modrou, popř. zelenou barvou na pozadí části plánu těchto úloh. V čase t_1 neexistuje aperiodický požadavek, a τ_1 tedy začíná (do t_2) běžet.

Její zápůjčka (zdůrazněna modrým pozadím) je přitom zvýšena na hodnotu 1. V t_3 je procesor předán, spolu se zápůjčkou o hodnotě 1, úloze τ_2 zápůjčka pro τ_2 je zvýrazněna zeleným pozadím), která s touto zápůjčkou běží až do t_4 , kdy si půjčuje od PE další jednotku času, a výše její zápůjčky tak vzrůstá na hodnotu 2. V t_5 přichází τ_1 ; díky absenci aperiodického požadavku si tedy τ_1 od PE zapůjčuje 1 jednotku času na procesoru. V t_6 však přichází aperiodický požadavek a PE si od τ_1 vybírá zápůjčku o velikosti 1, v rámci níž je (na úkor τ_1) obslužena, na prioritní úrovni τ_1 , první polovina aperiodického požadavku. Na provedení jeho druhé části si PE v t_7 , na prioritní úrovni τ_2 , vybírá zápůjčku od τ_2 . Jelikož však $P_2 < P_1$, je nejprve v době od t_7 do t_8 dokončena τ_1 a až poté, v době od t_8 do t_9 , i aperiodická úloha. Následně může být dokončena i τ_2 , a to s další zápůjčkou od PE, kterou si PE vybírá v t_{10} . V t_{11} jsou zápůjčky τ_2 nulovány.

Sporadický server (SS)

Z principu DS a PE vychází i tzv. *sporadický server* (*Sporadic Server* – SS). Na rozdíl od DS a PE však SS svoji kapacitu neobnovuje na začátku T_S vždy, ale pouze tehdy, byla-li předtím konzumována aperiodickými úlohami. K popisu činnosti SS je třeba zavést tyto symboly, a pojmy:

- P_{exe} pro prioritu běžící úlohy,
- P_S pro prioritu SS,
- *aktivní SS*, označující predikát pravdivý, platí-li $P_S > P_{exe}$ (tj. SS přerušil úlohu s nižší prioritou),
- t_A pro čas, ve kterém se SS stane aktivním,
- *nečinný SS*, označující predikát pravdivý, platí-li $P_S \leq P_{exe}$ (tj. procesor je vrácen úloze s nižší prioritou),
- t_1 pro čas, ve kterém se SS stane neaktivním,
- R_T pro čas, ve kterém má být doplněna (*replenishment*) kapacita SS,
- R_A pro hodnotu, o kterou se má navýšit kapacita SS.

Pravidla pro doplňování kapacity SS jsou takováto:

- je-li $C_S > 0$, je v t_A nastavena hodnota R_T na $t_A + T_S$,
- byla-li od t_A konzumována kapacita o velikosti K , je v t_1 nastavena hodnota R_A na K .

Ilustrační příklad k SS je na obr. 5, kde v signálu *aktivní SS* jsou zeleně vyznačeny oblasti aktivity SS. Poprvé je SS aktivní při konzumaci C_S v intervalu mezi t_1 a t_2 , během něhož jsou spotřebovány dvě jednotky C_S (tedy poprvé se C_S doplňuje v čase $R_T = t_1 + T_S = t_5$ a je navýšena o hodnotu $R_A = 2$

na cílovou hodnotu 3). Podruhé je SS aktivní při konzumaci C_S v intervalu mezi t_3 a t_4 , během něhož jsou spotřebovány dvě jednotky C_S (tedy podruhé se C_S doplňuje v čase $R_T = t_3 + T_S = t_6$ a je navýšena o hodnotu $R_A = 2$ na cílovou hodnotu 5, tj. maximální hodnotu C_S). Při použití SS lze zřejmě dosáhnout přijatelné reaktivity aperiodických úloh a zamezit přitom přetížení procesoru v důsledku jejich nadměrného počtu. Dále také rozložení časů R_T a hodnot R_A přímo závisí na tom, jak aperiodické podněty přicházejí v minulosti.

Shrnutí

V článku byly představeny základní mechanismy plánování aperiodických úloh spolu s mechanismy společného plánování aperiodických a periodických úloh RT. Při jejich popisu však nebyla probírána např. problematika použití kombinace různých typů serverů či použití více než jednoho serveru daného typu. Je třeba také zmínit, že vedle mechanismů popsanych v článku – jejichž výčet jistě nebyl vyčerpávající a úplný – existuje ještě mnoho dalších, např. (v původním jazyce) *Total Bandwidth Server*, *Constant Bandwidth Server* či *Slack Stealing* [2].

Nicméně lze konstatovat, že neexistuje mechanismus, který by byl schopen garanto-

vat dodržení časových mezí všech periodických i aperiodických úloh. Aby byl, musel by mít dopředu (a priori) informaci o časech příchodů aperiodických podnětů, což je však v přímém rozporu s definicí aperiodických podnětů i úloh [1].

Dokladem důležitosti mechanismů popsanych v článku je, že některé z nich (např. sporadický server) jsou standardní součástí rozhraní služeb operačního systému (jmenujme např. sporadické plánování v QNX Neutrino) či rozšíření obecně používaných norem, mezi které patří např. POSIX 1003.1d pro oblast RT.

Následující díl seriálu bude zaměřen na přehled mechanismů plánování konstruovaných pro situace přetížení systému, tj. schopných zajistit předvídatelné chování systému i tehdy, není-li k dispozici dostatek výpočetního výkonu.

Poděkování

Tento článek byl vypracován v rámci projektu Centrum excelence IT4Innovations, reg. č. CZ.1.05/1.1.00/02.0070, podporovaného Operačním programem Výzkum a vývoj pro inovace, financovaného ze strukturálních fondů EU a ze státního rozpočtu ČR, projektu Výzkum informačních technologií z hlediska bezpečnosti (CEZ MSM 0021630528) a grantu BUT FIT-S-11-1.

Literatura:

- [1] BUTAZZO, G. C.: *Hard Real-Time Computing Systems, Predictable Scheduling Algorithms And Applications*. Kluwer, 1997, pp. 109–178, ISBN 0-7923-9994-3.
- [2] COTTET, F. – DELACROIX, J. – KAISER, C. – MAMMERI, Z.: *Scheduling in Real-Time Systems*. John Wiley & Sons, 2002, ISBN 0-470-84766-2.
- [3] ČIŽINSKÝ, V.: *Implementace pokročilých mechanismů plánování množin RT úloh běžících pod uC/OS-II*. Diplomová práce, FIT VUT v Brně, 2010.
- [4] *QNX Realtime Operating System* [on-line]. Dostupné z <www.qnx.com>.
- [5] STRNADEL, J.: *Návrh časově kritických systémů II: úlohy reálného času*. Automa, 2010, roč. 16, č. 12, s. 18–19, ISSN 1210-9592.
- [6] STRNADEL, J.: *Návrh časově kritických systémů III: prioritní úlohy*. Automa, 2011, roč. 17, č. 2, s. 50–52, ISSN 1210-9592.
- [7] STRNADEL, J.: *Plánování úloh v systémech RT – I: závislé úlohy*. Automa, 2012, roč. 18, č. 10, s. 42–45, ISSN 1210-9592.

Ing. Josef Strnadel, Ph.D.,
Centrum excelence IT4Innovations,
Fakulta informačních technologií,
Vysoké učení technické v Brně
(strnadel@fit.vutbr.cz)

Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 12)

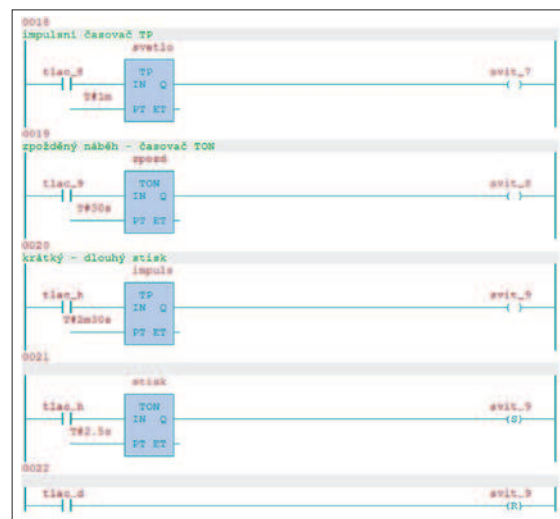
Toto pokračování seriálu je věnováno sekvenčním funkcím, které realizují závislost na čase. Nejprve budou popsány standardní funkční bloky časovačů. Na příkladech pak budou ilustrovány možnosti jejich využití v různých situacích při řízení chytrých domů. V praxi se lze často setkat s případy, kdy je třeba řídit objekt podle časových závislostí, vygenerovat impulzy potřebné délky nebo potřebnou posloupnost signálů v čase. Bývá potřebné měřit čas, dobu trvání události, měřit a nastavovat délky impulzů, zjistit a zaznamenat dobu události nebo řešit složitější úlohy závislé na čase.

Pochybnost o účelnosti tradičního řešení schodišťového vypínače

Vraťme se ještě k různým způsobům ovládání osvětlení schodiště (popř. chodby nebo jiných veřejných prostor). Uvažujme případ jednoduchého schodiště. Bez problémů je situace, kdy je osvětlení zhasnuté, přichází je rozsvítit při vstupu a zhasne při opuštění schodiště. Co když ale chodec vstoupil na již osvětlené schodiště rozsvícené předchozím chodcem, který si nově přichozícího nevšiml, a po chvíli světlo zhasne? Pak musí druhý chodec vyjít zbytek schodiště potmě nebo se po temném schodišti vrátit na začátek a rozsvítit si. Co když na zhasnuté schodiště vstupují

dva chodci, každý z jiné strany a oba se pokoušejí rozsvítit? Nebo oba opouštějí schodiště a současně chtějí zhasnout? Co když obyvatel prvního patra hledá něco ve své skříni nebo se chystá uklízet schodiště a šetrný obyvatel přízemí mu zhasne v domněnku, že někdo zapomněl zhasnout? Opo- menuté zhasnutí je velmi častým problémem společných prostor a je zdrojem zbytečné spotřeby energie.

Tradiční řešení osvětlení schodišť pochází z doby počátku elektrifikace, kdy byly k elektroinstalaci dostupné jen schodišťové vypí-



Obr. 35. Příklady volání funkčního bloku časovačů v programu LD

nače (přepínače), křížové spínače a podobné elektromechanické prvky. Nyní jsou ale pro „chytrou“ elektroinstalaci k dispozici podstatně pokročilejší nástroje. Především jde o infračervené senzory pohybu (PIR) a možnos-

ADRESÁŘ VYDAVATELSTVÍ

Ředitel: Ing. Emil Širůček

Adresa: Pod Vodárenskou věží 4, 182 08 Praha 8
tel.: 286 583 011-12, 266 052 804, fax: 284 683 022
e-mail: automa@fccgroup.cz, www.automaz

Šéfredaktor: Ing. Petr Bartošik

Zástupce šéfredaktora: Ing. Eva Vaculíková

Redakce: Petr Špůr, Milena Kočišová

Speciální projekty: Ing. Karel Suchý, Ing. Ladislav Šmejkal, CSc.

Odborná spolupráce:

Ing. Karel Bílek, Bernecker + Rainer Industrie Elektronik, Ges. m. b. H.,
Ing. Miroslav Dub, CSc., Sidat, spol. s r. o., **prof. RNDr. Ing. Petr Fiala, CSc., MBA**, katedra ekonomie Vysoké školy ekonomické v Praze, **Ing. Otto Havle, CSc., MBA**, FCC průmyslové systémy, s. r. o.,
doc. Ing. Petr Horáček, CSc., FEL ČVUT v Praze, **Ing. Zdeňek Hurák, Ph.D.**, katedra řídicí techniky FEL ČVUT v Praze, **doc. Ing. Josef Janeček, CSc.**, katedra řídicí techniky Fakulty mechatroniky a mezioborových studií TU Liberec, **doc. Ing. Karel Kadlec, CSc.**, ústav fyziky a měřicí techniky, Fakulta chemicko-inženýrská, Vysoká škola chemicko-technologická v Praze, **Ing. Petr Kašík, Amit, spol. s r. o.**, **doc. Dr. Ing. Vladimír Kebo**, institut ekonomiky a systémů řízení, Hornicko-geologická fakulta, VŠB-TUO Ostrava, **Ing. Marie Martinásková, Ph.D.**, ústav přístrojové techniky, Fakulta strojní ČVUT v Praze, **prof. Ing. Vladimír Mařík, DrSc.**, katedra kybernetiky FEL ČVUT v Praze, **doc. Ing. Pavel Nahodil, CSc.**, katedra kybernetiky FEL ČVUT v Praze, **prof. Ing. Miloš Schlegel, CSc.**, katedra kybernetiky FAV ZČU v Plzni, **prof. Ing. Vilém Srovnal, CSc.**, katedra kybernetiky a biomedicínského inženýrství, Fakulta elektrotechniky a informatiky VŠB-TUO Ostrava, **prof. Ing. Bohumil Šulc, CSc.**, ústav přístrojové techniky, Fakulta strojní ČVUT v Praze, **prof. Ing. Vladimír Vašek, CSc.**, Fakulta aplikované informatiky, Univerzita Tomáše Bati ve Zlíně, **prof. Ing. Petr Vavřín, DrSc.**, ústav automatizace a měření FEKT VUT Brno, **prof. Ing. František Zezulka, CSc.**, ústav automatizace a měřicí techniky FEKT VUT Brno

Inzerce: Ladislava Hošmáňková

Administrace časopisu: Ing. Petra Huňová

Sazba a grafická úprava: Dana Pecháčková, Tomáš Petr

FROM CONTENTS

New products	4
Business	5
News	7, 9, 23, 37, 39, 43, 52, 58
Education for automation	8,10
Fairs and conferences	11, 60
Theory for practice	16
Motion control	20, 24
Controlled electric drives	26, 32, 33, 34
Market survey	28, 30, 31
Standardization and legislation	35, 36
Automation solutions	38
Sensors, transmitters, instrumentation	40, 41, 42
Control technology	44, 46, 49
Communications in industry	50, 51
Components	53, 54, 55, 61
Industrial computing	59
Calendar of events	62
List of abbreviations	63

SEZNAM INZERENTŮ

ABB s. r. o.	28, 30, obálka 3
ATAS Elektromotory Náchod a. s.	11
B+R automatizace, spol. s r. o.	4, 11, 28, 30, 31
Beckhoff Automation GmbH	11
Berger Positec s. r. o.	29, 31
Cognex Germany	4
Česko-německá obchodní komora	18
ČVUT – Fakulta elektrotechnická	23
Delta Electronics, Inc.	12
Distrelec GmbH	50
Eaton Industries GmbH	12
Elektrotechnický zkušební ústav, s. p.	37
Elvac - IPC s. r. o.	42
Eplan engineering s. r. o.	12
FCC průmyslové systémy s. r. o.	59
Geovap, spol. s r. o.	13
HMS Industrial Networks GmbH	13
Humusoft s. r. o.	23
IMI International - Norgren CZ s. r. o.	4
Invensys Systems spol. s r. o.	obálka 4
Jirka a spol., s. r. o.	13
Kolektor Synatec d. o. o.	30, 31
Kontron AG	13
Krohne Messtechnik GmbH	13

Kwapil & Co GmbH, organizační složka	30, 31
LARM a. s.	14
Micro-Epsilon Czech Republic, spol. s r. o.	41
Mitsubishi Electric Europe B. V. - o. s.	29, obálka 2
Moravské přístroje a. s.	obálka 1
Murrelektronik CZ, spol. s r. o.	39
National Instruments, s. r. o.	5
Omron electronics spol. s r. o.	28
Panasonic Electric Works Czech s. r. o.	4, 14
Phoenix Contact GmbH & Co. KG	14
Profibus CZ	15
Routech s. r. o.	27
RS Components Sp. z o. o.	3
Schneider Electric CZ, s. r. o.	30, 31
Siemens s. r. o.	19, 29, 30
SEW-Eurodrive CZ s. r. o.	28
Terinvest, spol. s r. o.	48
TG Drives, s. r. o.	14
Technická univerzita v Liberci	22
Topinfo, s. r. o.	40
Turck s. r. o.	25
Uzímex Praha spol. s r. o.	43
VŠB – Technická univerzita Ostrava	29
Wago Kontakttechnik GmbH & Co. KG	15

Automa. Vydává firma FCC Public s. r. o. Přetisk je dovolen jen se svolením redakce a s uvedením pramene. Za případné závazky ke třetím stranám ručí autor. Názory autorů nemusejí být shodné se stanoviskem redakce. Vydavatel nezodpovídá za pravdivost údajů uvedených v inzerci a PR příspěvcích. Pro předplatitele v České republice provádí distribuci v zastoupení vydavatele společnost Send Předplatné, Ve Zlíbku 1800/77, 193 00 Praha 9 Horní Počernice; příjem objednávek a reklamace: tel.: 225 985 225, fax: 225 341 425, send@send.cz, www.send.cz. Pro Slovenskou republiku: Magnet Press Slovakia s. r. o., Teslova 12, P. O. Box 169, 830 00 Bratislava, tel.: +421 244 454 559 (předplatné), +421 244 454 628 (sekretariát), predplatne@press.sk, Elez, Zlatovská 27, 911 05 Trenčín, tel.: +421 326 527 672, fax: +421 327 436 536, elez@elez.sk, Mediaprint-Kapa Pressegrasso, a. s., odd. inej formy predaja, Vajnorská 137, P. O. Box 183, 830 00 Bratislava 3, tel.: +421 244 458 821, +421 244 458 816, +421 244 442 773, fax: +421 244 458 819, e-mail: predplatne@abompka-pa.sk a Slovenská pošta, SPT, Nám. slobody 27, 810 05 Bratislava, objednávky prijíma každá pošta a poštový doručovateľ. Objednávky do zahraničí vyřizuje Mediaservis s. r. o., Paceřická 1, 193 00 Praha – Horní Počernice, tel.: 271 199 250, kauerova@mediaservis.cz. Veškeré objednávky přijímá také redakce, která zprostředkuje i případné reklamace. Vychází 12x ročně. Tiskne Kavka Print, a. s., Ke Zdíbsku 620, 250 67 Klecany, tel.: 317 070 745. Do tisku předáno 1. 11. 2012, vyšlo 6. 11. 2012. Cena časopisu: 52 Kč (dvojčíslo 104 Kč).