**ТЕНДЕНЦИИ В РАЗВИТИЕТО НА УПРАВЛЯВАЩИТЕ СИСТЕМИ ЗА АВТОМАТИЗАЦИЯ**

**Емануил Марков**

***Резюме:*** *Целта на работата е да направи обзор в областта на софтуерно-базираните логически контролери (SoftPLC) и програмируемите контролери за автоматизация (PAC) и да ги сравни с традиционните PLC контролери. Тъй като PLC може да се счита за „стара” технология те са разгледани повърхностно и се използват като база за сравнение. SoftPLCs и PAC технологиите трябва да са способни да изпълняват същите задачи като стандартните PLC, както и да предоставят набор от нови възможности (паралелна обработка, обработка на сигнали с интегрирани DSP и FPGA модули).*

***Ключови думи:*** *програмируеми контролери за автоматизация, системи за управление, операционни системи за реално време, многоядрени системи за управление*

**THE FUTURE OF AUTOMATION CONTROL SYSTEMS**

**Emanuil Markov**

***Summary****: This work is meant to give an overview over the field of software based logic controller (SoftPLC) and programmable automation controllers (PAC), and to some extent compare them to traditional PLCs. Since the PLCs can be considered “old” technology not much effort is put into the inner workings of these and they are used more as a reference. SoftPLCs and PACs should be able to perform the same tasks in the same ways as ordinary PLCs and provide a lot of new possibilities (parallel processing, integrated DSP and FPGA).*

***Keywords:*** *Programmable Automation Controllers, control systems, Real-Time Operating Systems, Multicore control systems*

1. **Въведение**

През последните години интеграцията между различните системи за управление става все по-важна. PLC контролерите се свързваха към компютрите с цел събиране на данни през най-различни екзотични интерфейси.

С навлизането на софтуерно базираните PLC работещи върху обикновен компютър, както и на PLC работещи с реално времева операционна система част от проблемите бяха решени чрез използването на стандартни интерфейси между софтуерните системи.

Софтуерно базираните PLC дават голяма гъвкавост при реализация на самостоятелни и по-специално на движещи се машини. Един компютър с вграден екран може да бъде достатъчен за управление и наблюдение на машина или на поточна линия от оператор, диагностика, изчисления навигация, съхранение на данни и т.н.

В това решение обаче лесно се забелязват следните проблеми: Нормалните операционни системи за настолни компютри не са направени за работа в реално време и трябва да се правят допълнения към софтуера. Също така надеждността на персоналните компютри не е достатъчна за работа в агресивни условия или за реализиране на отказо-устойчиви системи.

**2. Определения**

**PLC**: Програмируем логически контролер. Произхождат от програмируеми релета със специализиран микроконтролер. В момента повечето системи имат възможности за комуникация, работа с аналогови входове и изходи, и цифрово управление.

**SoftPLC**: Софтуер реализиращ PLC работещ върху компютър с операционна система с общо предназначение.

**PAC**: Програмируем контролер за автоматизация комбиниращ гъвкавостта и възможностите на компютърно базирана система за управление с надеждността и наличието на входно изходни интерфейси на PLC.

**OPC**: Индустриален стандарт за връзка и обмен на данни между софтуер и хардуер за автоматизация.

**SCADA**: Supervisory Control And Data Acquisition. Система за събиране на данни от системата за автоматизация към мрежата на компанията, даваща възможност за супервайзорски контрол от по-високо ниво.

**MES**: Manufacturing execution system. Система за измерване и контрол на операциите във фабриката. В този контекст тя служи като връзка между системата за автоматизация и системата за планиране на ресурсите на предприятието (ERP).

**3. История**

През 80-те години системите за управление използвани в производствените системи бяха DCS (цифрова система за управление), PLC и различни специфични вградени системи. От тогава постепенно ролята на DCS системите бива иззета от PLC контролери с разширителни карти.

PLC контролерите са базирани на микропроцесор изпълняващ специализирана програма съхранена в постоянната му памет. Програмния цикъл е точно определен и съответно времето за отговор на изходите на системата е точно предсказуемо.

Благодарение на относителната простота на PLC контролерите, в сравнение с персоналните компютри, те са известни с тяхната надеждност.

В последното десетилетие нуждата от свързване на PLC системите с други системи става една от основните насоки за развитие. Това започва с нуждата контролерите да могат да работят с компютърно базирани операторски интерфейси и в последствие се развива до оперативна съвместимост със други системи за управление и отчетност на предприятието. В началото на десетилетието операторските интерфейси бяха специализирани блокове, комуникиращи с PLC по серийна връзка, но развитието на персоналните компютри ги направи много популярно решение за операторски интерфейс. Когато данните достигнат до компютъра те могат да бъдат лесно разпределяни към други системи например по Ethernet мрежа с TCP/IP протокол. Докато PLC контролерите обикновено са затворени системи и могат да бъдат програмирани само с осигурените от производителя инструменти, от страна на персоналните компютри е избрана относително отворена архитектура позволяваща лесното използване на стандартни компоненти и протоколи, като например OLE for Process Control (OPC) за взаимодействие между софтуерните системи [1].

Развитието на полевите мрежи и интелигентните входно-изходни устройства направи възможно персоналния компютър да играе ролята на контролер, изпращайки команди по полевата мрежа директно до сензорите и изпълнителните механизми. Това е особено ефективно в приложения, където има нужда едновременно от управление и от събиране на данни. За целта върху компютър, свързан към полевите устройства през съответната полева мрежа (PROFIbus, CAN и т.н.) се инсталира SoftPLC софтуер. Компютърът играе ролята на централна станция за програмиране, наблюдение и операторско работно място. Възможностите предлагани от полевите мрежи позволяват и разпределението на управляващото приложение върху различни управляващи компютри, както и реализирането на различни смесени решения според нуждите на специфичната задача (фиг. 1).



Фигура 1. Типова архитектура на връзка между различните нива на   
системите за автоматизация [2]

Производителите на устройства за автоматизация отговарят на увеличените изисквания на съвременните индустриални приложения като създават устройства които съчетават предимствата на стриктно определения PLC стил за управление на машини и процеси с гъвкавостта на конфигуриране и възможностите за интеграция на компютърно базираните системи (фиг. 2).



Фигура 2. Приложение на програмируемите контролери за автоматизация [3]

**4. Хардуерна реализация**

Производителите и потребителите на средства за автоматизация поемат по два основни пътя.

Първият е заместване на традиционното PLC с компютър с интерфейсни карти за полевите мрежи използвани в системата за автоматизация. Компютъра контролира процесите изпращайки съобщения до подчинени контролери, интелигентни входно-изходни устройства, сензори и изпълнителни механизми.

Софтуера за компютъра обикновено е компилиран за специализирана реално времева операционна система или за операционна система с общо предназначение, но имаща реално времево ядро или реално-времеви разширения. В последните години голямо внимание се обръща на Windows CE .Net, която притежава реално-времеви възможности и в същото време големи части от Windows библиотеките, което например прави разработването на потребителски интерфейси лесно. Windows CE системите са разработени за вграждане в OEM продукти, което дава възможност изгледа и функционалността им да бъдат конфигурируеми [4].

Други операционни системи с реално-времеви разширения са RTLinux и разширения за Windows XP, като Ardence RTX или HyperKernel от Nematron за Windows 2000.

Компютъра сам по себе си може да бъде обикновен настолен компютър или устойчив индустриален компютър (обикновено предпочитано решение въпреки относително високите цени на подобен клас устройства). Индустриалните компютри често са снабдени с вграден екран (или вградени в екран), а понякога и с клавиатура, което ги прави компактни и удобни за употреба.

При липсата на еднозначно решение за употреба на PLC или компютър за решаване на сложни проблеми, инженерите започват да изискват от производителите на оборудване възможност да комбинират развитите софтуерни възможности на персоналните компютри с надеждността на индустриалните контролери. Това дава възможност за навлизането на програмируемите контролери за автоматизация (PAC). За да се изпълняват сложни функции като бързо събиране на данни, ПИД управление и т.н. типичните PLC базирани системи за управление изискват добавянето на допълнителни и често скъпо струващи разширителни модули. Програмируемите контролери за автоматизация имат тези възможности вградени по подразбиране.

Разширените софтуерни възможности изискват не само софтуер, а също и увеличаване на хардуерните ресурси на контролерите. Производителите в момента ползват индустриални версии на процесори за изчисления с плаваща запетая, DRAM (оперативна памет при която всеки бит се съхранява в отделен кондензатор, масово използвана в съвременните PC), устройства за съхранение на данни като Compact Flash и бързи Ethernet чипове. Това дава възможност на производителите да разработват по-мощен софтуер разполагащ с гъвкавостта и широката приложимост на PC базираните системи за управление, работещи върху реално-времеви операционни системи. В резултат новите програмируеми контролери за автоматизация комбинират най-доброто от възможностите на PLC и PC [5].

Някои производители разширяват възможностите на PAC отвъд интегрирането на комуникационни интерфейси, аналогова и цифрова периферия. Много индустриални приложения изискват събирането на резултати от измервания с висока скорост, например вибрационни системи или системи за управление на електрическата мрежа. Събраните данни се използват за наблюдение, предсказване на потенциални проблеми от износване или претоварване и за корекция на управляващите алгоритми. В PLC системите тези данни се събират и обработват от специализирани инструменти и се предават на управляващата система по комуникационен протокол. PAC системите от своя страна могат да имат вградени сигнални процесори правещи милиони измервания в секунда и предаващи резултатите директно в системите им за управление. Също така се предлагат PAC с вградени процесори за обработка на изображения, премахващи нуждата от интегрирането различни софтуерни и хардуерни платформи [6].

Бъдещето на PAC системите е тясно свързано с развитието на вградените системи. Пример за това е възможността да се интегрира FPGA чип. FPGA са електронни компоненти използвани от производителите за дефиниране на специфична логика на работа на самия чип позволявайки по този начин да се създават интелигентни нови устройства конфигурируеми от потребителя. FPGA могат да бъдат сравнени с компютър, който вътрешно се преструктурира, за да изпълни възложената му задача. Това дава възможност на инженерите да вграждат изключително време-критични функции в хардуера, като сензори за наблюдение на гранични условия, близост, както и за медицински приложения [7].

**5. Софтуерни реализации**

Софтуера е ключовата разлика между PAC и SoftPLC технологиите спрямо PLC.

Традиционните PLC софтуери започват с надеждна и лесна за употреба архитектура и след това добавят към нея допълнителни възможности. PLC софтуера следва общ модел на сканиране на входове, изпълнение на управляващ код, задаване на състояния на изходите. Инженерът трябва да напише само управляващия код, тъй като циклите за сканиране на входовете и опресняване на изходите са скрити.

В развитието на PAC и SoftPLC софтуера могат да се разграничат две основни философии:

***Софтуер базиран на PLC философията***

Благодарение на натрупания през годините опит тази архитектура се счита за по‑лесна и бърза при създаването на системи за управление. Липсата на гъвкавост в тези системи елиминира нуждата инженера да разбира на ниско ниво начина на работа на контролера, за да създава надеждни програми. Повечето PLC производители добавят към съществуващата софтуерна архитектура нови функции, като Ethernet комуникация, управление на задвижвания и други съвременни алгоритми. В същото време те запазват познатото PLC програмиране. В резултат се получава софтуер разработен за да покрие специфични приложения като логическо управление, управление на задвижвания, ПИД, но е по-малко гъвкав за реализиране на специфични задачи като комуникации, събиране на данни или специфични управляващи алгоритми.

***Софтуер базиран на PC философията***

Традиционните доставчици на PC софтуер предлагат много гъвкави програмни езици, които осигуряват достъп на ниско ниво до работата на хардуера. Този софтуер също предлага надеждност, определеност и стандартни архитектури за управление. Софтуера е изключително гъвкав и подходящ за реализирането на сложни приложения, изискващи структури от данни, специфични програмни техники и контрол на системно ниво, но е сложен за реализирането на прости приложения.

Повечето доставчици на PAC се опитват да предложат хибрид между двете философии, предлагайки опростен PLC интерфейс за основните нужди на инженерите по автоматика, както и развити софтуерни библиотеки за реализиране на сложни задачи.

Според [8], PAC трябва да отговаря на следните изисквания:

• Работа върху една платформа с много области включващи логическо управление, управление на задвижвания, управление на процеси.

• Наличие на единна развойна платформа използваща общо именуване на работните променливи и обща база данни за дефиниране на задачите в различните дисциплини.

• Интегриране на софтуера и хардуера на контролера.

• Наличие на софтуерни инструменти за разработка на управляващи програми които поддържат процес който е разпределен между няколко машини или изчислителни възли.

• Работа в отворени модулни архитектури които отразяват индустриалните приложения.

• Наличие на де-факто стандартите за мрежови интерфейси, езици и протоколи, позволяващи обмен на данни между системи от различни производители.

• Осигуряване на ефективно сканиране на входно-изходната периферия.

**6. Многоядрени реализации**

Многоядрените архитектури предлага много възможности в областта на вградените системи. Въпреки това общността остава резервирана по отношение на използването на многоядрените архитектури за реално-времеви задачи. За целта е необходимо да има добре дефинирана среда, за да се осигури детерминизъм на системата. Необходимо е да има ясно разпределение на ресурсите между различните ядра, така че да е невъзможно да стане заключване, както и да се определи кое ядро за кои приложения отговаря или т.нар. “core affinity”.

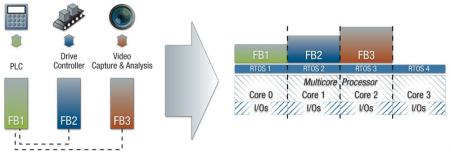
Благодарение на вградените възможности за виртуализация е възможно да се асоциира памет, периферия и прилежащите и прекъсвания към определено процесорно ядро. Също така се осигурява мрежов слой отговарящ за комуникациите между процесите в различните ядра.

С тези технологии разработчиците могат да създадат напълно скалируеми реално-времеви приложения за многоядрени платформи. Такова решение може да направи лесна миграцията от едноядрена към дву- и повече ядрени системи без пълно преработване на приложенията, със запазване на предназначените за тях периферни устройства и ядра за време-критичните задачи.

**7. Изисквания към системите за управление в многоядрени платформи**

Исторически, устройствата работещи с реално-времеви задачи се дефинират като дискретни функционални блокове (ФБ) на системата за управление [12][13]. Всеки функционален блок се реализира на отделен изчислителен възел. Развитието на многоядрените архитектури дава възможност за консолидирането на множество ФБ в една платформа. Това не само намалява цената на системата, но също така осигурява по-ефективно свързване между ФБ и като резултат подобрява производителността и надеждността на системата.

Ако се сметне, че един ФБ се състои от специфични за приложението хардуерни и софтуерни компоненти, както и процесор или контролер за реализиране на функционалността, то многозадачния мениджър трябва да постигне необходимата изолация, за да запази ФБ независими. С използването на техниките за вградена виртуализация всеки ФБ в системата за управление може да бъде назначен към отделно ядро от многоядрената платформа, с заедно със специализираната периферия и управлението на паметта (фиг. 3) [9]. Всяко ядро работи под управлението на индивидуален мениджър (RTOS).



Фигура 3. Функционални блокове пренесени от индивидуални   
платформи към многоядрена платформа.

Следващата стъпка е да се даде възможност на ФБ да комуникират помежду си. Тази функционалност може да се осигури на приложно ниво в операционната система отговаряща за всеки ФБ. Традиционно това ниво се представя като мрежова услуга или като споделена памет. Мрежовата услуга се реализира лесно, но изпълнението и представлява излишен товар. От друга страна използването на споделена памет, макар и оптимално като производителност, крие рискове от конфликти между различните задачи. За целта трябва да се дефинират приоритети на задачите във всяко ядро, както и мениджър на приоритетите следящ че задачите от всяко ядро се съобразяват с приоритетите на задачите от другите ядра с които комуникират.

**8. Windows Embedded Compact 7**

Последната версия на операционната система Windows CE наричана Windows Compact 7 (за по-кратко - Compact 7) е първата от линията Windows CE, която поддържа многоядрени платформи. Подобренията в Compact 7 започват с пренаписано ядро. Новото ядро на операционната система е с възможност да поддържа до 250 ядра, въпреки че се препоръчва да се ползва с 8 или по-малко ядра. Въпреки това възможността за работа върху много ядра значително подобрява времето за реакция на системата, особено когато системата е натоварена [10].

Compact 7 поддържа ARM, MIPS, както и x86 архитектури. ОС се нуждае от поддържан тип на процесора, RAM, място за съхранение на операционната система, реално-времеви часовник за периодично прекъсване за диспечера на задачите. Всички останали хардуерни решения са оставени на производителите на устройства.

За да се осигури коректната работа на приложения, компилирани за Windows CE 6 и по-стари, всички техни нишки трябва да работят само върху едно ядро. Това предотвратява възможни непроверени ситуации, които могат да възникнат при работа в многопроцесорен режим. Приложения компилирани за Compact 7 могат да разпределят нишките си между всички ядра.

За управлението на множество ядра са осигурени нови библиотеки (API). Приложенията могат да задават принадлежност на нишките си всяка към отделно ядро, както е възможно всички нишки да се насочат към едно определено ядро. При нужда разработчиците могат да изключат ядра, които в определен момент не се ползват с цел пестене на енергия.

Compact 7 има подобрения и в управлението на паметта. Предходните версии на Windows CE поддържаха до 512 Mb. В Compact 7 максималното количество физическа памет е увеличено до 3 Gb. Индивидуалните процеси са ограничени и имат достъп само до виртуалното поле памет достъпно за този процес. Увеличението на достъпната оперативна памет е много полезно за приложения обработващи големи обеми от данни или изображения.

Не на последно място е засегната защитата от изпълнение на зловреден код, като възможности на настолната версия на Windows 7 са пренесени в Compact 7 (Address Space Randomization и Data Execution Prevention) [11].

**9. Сравнение между различните архитектури**

|  |  |  |  |
| --- | --- | --- | --- |
| Парамертър | PLC | SoftPLC | PAC |
| Интеграция | Съществуват интерфейси за комуникация с PC. | Отлична интеграция с други софтуерни системи използвайки стандартни комуникационни протоколи. | Отлична интеграция с други софтуерни системи използвайки стандартни комуникационни протоколи. |
| Надеждност | Изключително надеждни. | Неособено надеждни в сравнение с PLC. Операционната система и хардуера на компютъра имат определяща роля. | Надеждни. Разширените софтуерни възможности все пак дават по-големи възможности за допускане на грешки |
| Програмира-не | IEC 61131-3 (с надстройки)  Необходим е специализиран софтуер за програмиране от производителя. | IEC 61131-3 (с надстройки). Осигурени са интерфейси за връзка със системата за автоматизация посредством стандартни езици като C, C++. | IEC 61131-3 (с надстройки). Осигурени са интерфейси за връзка със системата за автоматизация посредством стандартни езици като C, C++. |
| Цена | От много ниска до висока. Операторски или комуникационни интерфейси оскъпяват системата. | От ниска до висока.  Приложения имащи нужда от компютър или операторски интерфейс спестяват от споделеното му използване | От ниска до висока.  Ценово ефективни за реализиране на по-големи и сложни проекти. |

Таблица 1. Предимства и недостатъци на различните архитектури

**10. Заключения**

В началото на века изглеждаше, че технологията SoftPLC има голямо бъдеще. Възможността да работи с евтини PC части, стандартни програмни езици и оперативната съвместимост със системи за корпоративно управление, системи за съхранение на данни и т.н. е достатъчно привлекателно за да може постепенно тази технология да измества специализираните PLC. Най-големият проблем пред тази технология е в надеждността на използваните компютри. Факта че се използват в тях стандартни компоненти може да бъде и недостатък, тъй като компонентите не са тествани за съвместимост.

Ситуацията при използване на SoftPLC във вградени системи е съвсем различна. Там тази технология може да замести голямо количество специализирана електроника, и да намали разходите за разработка. Движещи се машини (комбайни, трактори …) могат да има само един бордови компютър, който се грижи за всичко, лесен е за поддръжка и е относително евтин. Въпреки че такъв клас системи не са директно съвместими една с друга използването на стандартни програмни езици осигурява относително лесна преносимост ако определен компютър бъде спрян от производство и трябва да бъде заменен с друг с подобни характеристики.

Увеличената интеграция на производствените системи (автоматика-MES-ERP) налага PLC да стават все по-близки до софтуерните решения. В резултат на това се появяват PAC системите. Въпреки това малки самостоятелни системи без операторски интерфейс осигуряващи критични системи ще продължат да използват специализирани PLC решения.

Предвид очертаните тенденции може да се предположи, че с течение на времето PAC системите ще изместят до голяма степен PLC системите от висок и среден клас. Софтуера за тях тепърва ще се развива, а с разширеното внедряване цените все повече ще се доближават до тези на PLC. Дори в момента за реализиране на приложения свързани с по-сложни управления, голям брой входове/изходи, обработка и събиране на данни, много бързо управление машинно зрение и т.н. и особено за системи, съчетаващи няколко от тези изисквания, PAC е най‑доброто възможно решение. На кратко, ако има нужда от система в която участват PLC и PC, е най-добре да се използва PAC.

**ЛИТЕРАТУРА**

[1] Harjula M. (2007) , ***SoftPLC vs. PLC***, Automaation tietotekniikan seminaari, Syksy

[2] Texas Instruments whitepaper (2011), ***Programmable Logic Controller Solutions***, http://www.ti.com/lit/ml/slyb180/slyb180.pdf

[3] Opto 22 whitepaper (2006), ***Understanding Programmable Automation Controllers (PACs) in Industrial Automation***, <http://www.opto22.com/site/documents/doc_drilldown.aspx?aid=2813>

[4]Dicklin R. (2003)*,* ***Windows CE .NET Combines Real-Time Control, HMI and Networking****,* RTC Magazine, July 2003

[5] Bell I. (2005). ***The Future of Control***, IEEEManufacturing Engineer, August/September 2005

[6] Bell I. (2006). ***PACs for Industrial Control, the Future of Control***, http://zone.ni.com/devzone/cda/tut/p/id/3755

[7] Hansen F. (2006)*,* ***FPGA Takes Control, "Design & Elektronik"****,* issue 10/2006.

[8] Crump D. & Hill D.*,* ***Programmable Automation Controllers Bring Together the Best of PLCs and PCs****,* RTC Magazine, January 2010

[9] Grujon C. (2012)*,* ***Harnessing the Potential of Multicore Processors for Real-Time Application****,* RTC Magazine, April 2012

[10] Boling D. (2011)*,* ***CE Goes Multicore: Microsoft Windows Embedded Compact 7****,* RTC Magazine, January 2011

[11] *http://www.microsoft.com/windowsembedded/en-us/campaigns/compact7/*

[12] Angelov C. K., Ivanov I. E., “***Formal Specification of Distributed Computer Control Systems (DCCS). Specification of DCCS Subsystems and Subsystem Interactions***”. Proc. of the International Conference “Automation & Informatics’2001”, May 30 - June 2, 2001, Sofia, Bulgaria, vol. 1, pp. 41-48.

[13] Evgeniev I., Gueorguiev V. E., “***Formal Methods for System Design***”, Information Technologies and Control, 4, 2004

**Автор:** Емануил Марков, докторант ТУ-София, факултет Автоматика, катедра Системи и управление, лаборатория Съвременни системи за управление, *еmail: markov@takt-ltd.com*