

Projekt i implementacja laboratoryjnego modułowego manipulatora o pięciu stopniach swobody (5R) z wbudowanym systemem sterowania

Jacek Augustyn*, Łukasz Cichocki**, Łukasz Duda**

*AGH Akademia Górniczo-Hutnicza w Krakowie, Wydział EAIIB, Katedra Automatyki i Inżynierii Biomedycznej

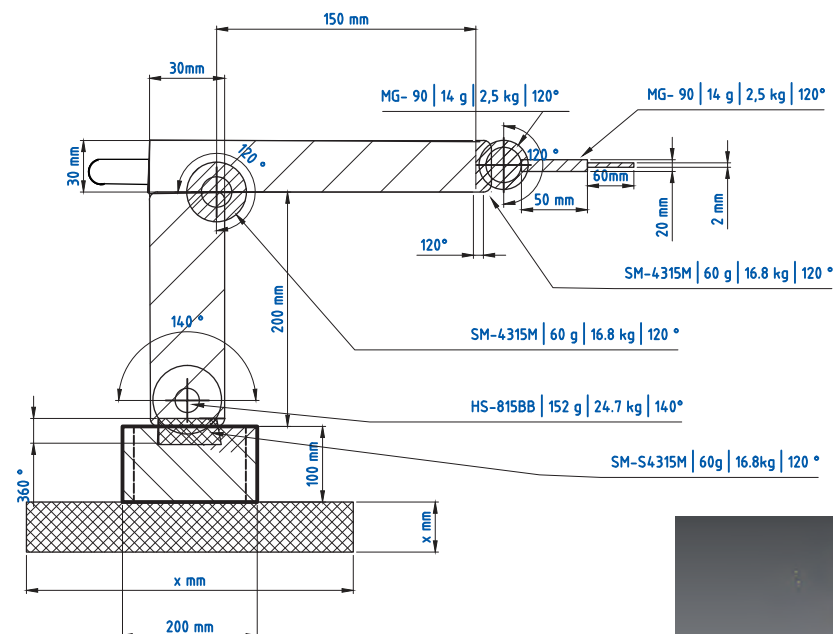
**Zakład Informatyki, Państwowa Wyższa Szkoła Zawodowa w Tarnowie

Streszczenie: W artykule przedstawiono projekt oraz praktyczną realizację modułowego manipulatora o pięciu stopniach swobody. Omówiono założenia projektowe dotyczące części mechanicznej. Zaprezentowano strukturę wieloprocesorowego, hybrydowego systemu sterowania. Manipulator przeznaczony jest do zastosowań laboratoryjno-dydaktycznych.

Słowa kluczowe: manipulator, system mechatroniczny, system wbudowany, system sterowania, system czasu rzeczywistego

Projektowanie i realizacja manipulatorów jest złożonym zagadnieniem mechatronicznym. Łączy ze sobą wybrane działy mechaniki, elektroniki, systemów mikroprocesorowych oraz informatyki z zakresu systemów wbudowanych, informatyki aplikacyjnej, jak również systemów sterowania.

W artykule przedstawiono projekt oraz praktyczną realizację manipulatora dydaktyczno-eksperymentalnego o pięciu stopniach swobody. Została omówiona część mechaniczna oraz architektura wieloprocesorowego, hybrydowego systemu sterowania czasu rzeczywistego.



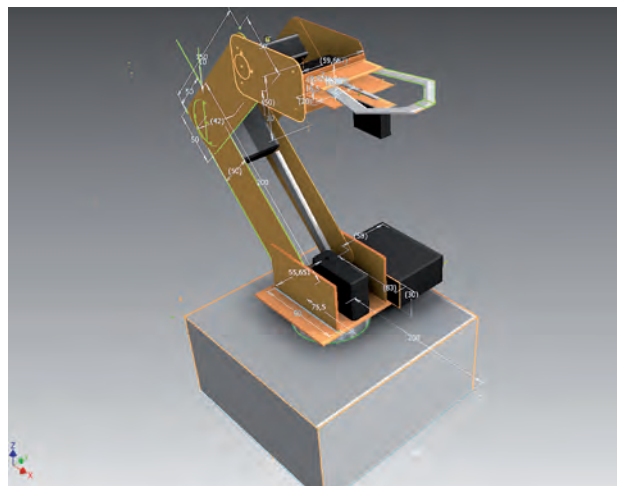
Rys. 1. Projekt mechaniczny

Fig. 1. Mechanical project

Manipulatory są mechaniczną implementacją ludzkiej kończyny górnej, a wyposażenie ich w odpowiedni efektor pozwala na realizację zadań, których stopień trudności manualnych dorównuje możliwościom ludzkiej ręki. Manipulatory o wielu stopniach swobody są coraz powszechniejsze i znajdują liczne zastosowania, zarówno przemysłowe, jak i dydaktyczno-eksperymentalne. Zapewniają bezpieczną i niezawodną pracę w przemyśle motoryzacyjnym oraz elektromaszynowym, gdzie produkowane są różne dobra w skali masowej.

1. Projekt i realizacja części mechanicznej

Do zaprojektowania robota wykorzystano środowisko programowe z rodziny AutoCad, dostępne w wersji edukacyjnej na licencji Autodesk Education [1]. Rysunki poglądowe zostały stworzone w programie AutoCad Mechanical 2011 (rys. 1 i 2). Dzięki komputerowemu wspomaganiu projektowania uzyskano możliwość wstępnej weryfikacji błędów wraz z możliwością podjęcia decyzji zarówno konstrukcyjnych, jak i mecha-

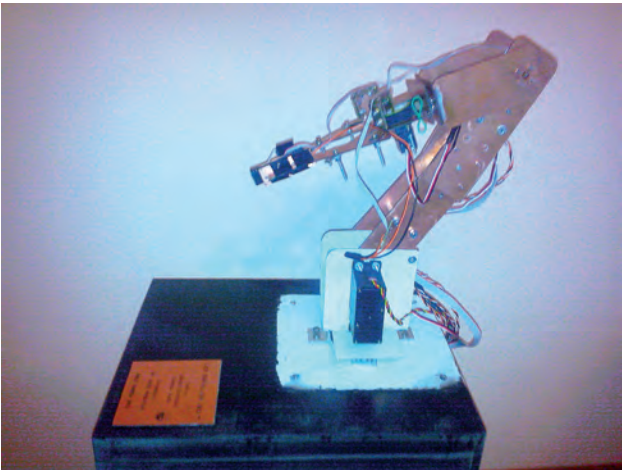


Rys. 2. Trójwymiarowa wizualizacja manipulatora

Fig. 2. Three-dimensional visualization of the manipulator

nicznych. Manipulator został wykonany na podstawie rysunków technicznych wygenerowanych za pomocą programu AutoCad Inventor [1]. Program umożliwia wybór z trójwymiarowej bryły dowolnego elementu aseblacyjnego oraz jego swobodną obróbkę, jak i wygenerowanie rysunku wykonawczego.

Konstrukcja manipulatora wymagała uwzględnienia szeregu czynników, takich jak materiały użyte do wykonania elementów nośnych, a także decyzje co do rozwiązań mechanicznych. Konstrukcję wykonano z laminatu FR4. To wzmacniany włóknem szklanym laminat epoksydowy, charakteryzujący się lekkością i sztywnością. Jego twardość według skali Rockwella to 110 jednostek. Dodatkową zaletą jest łatwa obróbka oraz możliwość względnie prostego mechanicznego montowania dalszych elementów konstrukcyjnych.



Rys. 3. Zrealizowany manipulator
Fig. 3. Fully assembled manipulator

Tab. 1. Serwomechanizmy użyte w projekcie
Tab. 1. Servos used in the project

Przegub	Typ	Masa [g]	Moment [kg/cm]	Kąt [°]
Podstawa	SM-4315M	60	16,8	120°
Przegub 1	HS-815BB	152	24,7	140°
Przegub 2	SM-4315M	60	16,8	120°
Przegub 3	S3317S	19	3,5	120°
Przegub 4	SM-S3317M	19	3,0	120°
Chwytnik	SM-S3317M	19	3,0	120°

Główna oś manipulatora została osadzona na łożysku kulkowym o wytrzymałości około jednej tony. W układach napędowych zastosowano zmodyfikowane serwomechanizmy modelarskie. Mechanizmy są zróżnicowane, gdyż zależnie od osi napędowej wymagany jest różny moment obrotowy. Po wykonaniu analiz obciążeń poszczególnych osi, dobrane mechanizmy zestawiono w tab. 1. Przykładowo napęd odpowiedzialny za przegub pierwszy, o największym obciążeniu, ma wartość momentu siły równą 24,7 kg/cm, natomiast napęd wykorzystany w efektorze już tylko 3,0 kg/cm.

Widok zrealizowanego manipulatora pokazany jest na rys. 3. Manipulator zasilany jest ze standardowego zasilacza impulsowego pochodzącego z komputera PC.

2. System sterowania

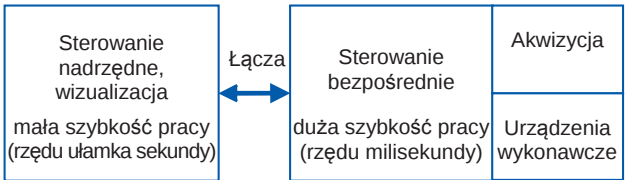
W literaturze przedstawianych jest wiele propozycji architektur systemu sterowania. Autorzy zwykle proponują w systemie sterowania zastosowanie klasycznych mikrokontrolerów 8-bitowych, jak ATmega [2] czy też PIC18 [3], PIC16 [4] w połączeniu z komputerem PC bądź notebookiem. Ze względu na obecność komputera PC, cały system ma duże wymiary oraz pobiera stosunkowo dużą moc. W konstrukcjach zawierających dużą liczbę jednostek napędowych stosuje się rozwiązania modułowe zawierające mikrokontrolery, które współpracują poprzez lokalne łącze. Przykładowe rozwiązania wykorzystujące mikrokontrolery 8051 opisane są w [5], czy też [6]. Najnowsze doniesienia pokazują, że zaczynają być stosowane nowoczesne, 32-bitowe procesory klasy ARM/Cortex, jak np. STM32 [7]. Do komunikacji bezprzewodowej używane są moduły Bluetooth. Należy wspomnieć, że stosuje się także sterowniki złożone z procesorów klasy ARM z systemem Linux [8], wymagają one jednak specjalizowanych bibliotek programistycznych, a uzyskiwana częstotliwość akwizycji danych i uaktualniania wartości sterujących wynosi 100 Hz.

W projekcie zaproponowano model modułowego, zdecentralizowanego, wieloprocesorowego systemu sterowania. Wyodrębniono w nim dwa kluczowe bloki (rys. 4):

- wieloprocesorowy system sterowania bezpośredniego oraz akwizycji sygnałów pomiarowych,
- system sterowania nadrzędnego i wizualizacji.

System sterowania bezpośredniego powinien być szybki, deterministyczny czasowo i bezpośrednio współpracować z układami pomiarowymi (czujniki, przetworniki, wejścia cyfrowe), jak i urządzeniami wykonawczymi (wyjścia, PWM). Wymaga się pracy w reżimie „twardego” czasu rzeczywistego (ang. *hard real-time*) o czasach reakcji rzędu milisekundy lub mniej. Niemal każdy taki system jest ściśle dedykowany i oprogramowywany stosownie do wymagań. Najczęściej nie ma on uniwersalnego systemu operacyjnego, co redukuje opóźnienia (ang. *latency*), zwiększa jego determinizm oraz często ułatwia testowanie i weryfikację rozwiązania. Jest prawie regułą, że nie ma bloków wizualizacji, gdyż mogą one znacząco spowolnić działanie i obniżyć determinizm czasowy sterowania bezpośredniego.

System sterowania nadrzędnego komunikuje się z systemem bezpośrednim ze znacznie mniejszą częstotliwością rzędu kilkunastu razy na sekundę (w ogólnym przypadku losową w dużym zakresie). Pozwala on na modyfikację parametrów roboczych oraz wizualizuje dane układu sterowania. Interakcja z człowiekiem rzadko odbywa się z częstotliwością większą niż raz na sekundę. Jednak z punktu widzenia rejestracji danych i zapisu ich na dysk, wymiana danych może odbywać się znacznie

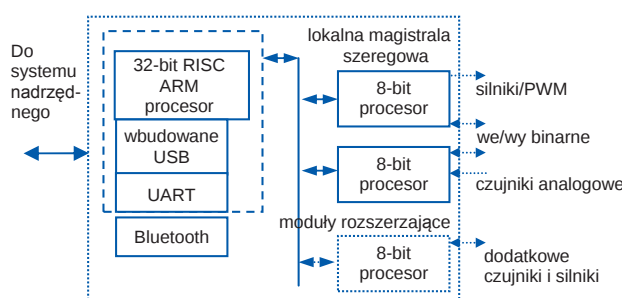


Rys. 4. Dekompozycja głównych bloków systemu sterowania
Fig. 4. Decomposition of main sections of the control system

szybciej zarówno pod względem częstotliwości odpytywania, jak i łącznej ilości transmitowanych danych. Duży strumień danych umożliwia m.in. rejestrację zachowania się manipulatora. Dane mogą posłużyć do identyfikacji obiektu sterowania. System nadrzędny docelowo może także pełnić funkcję generatora trajektorii na podstawie plików tekstowych.

2.1. Podsystem sterowania bezpośredniego i akwizycji

Do realizacji przyjęto architekturę systemu wieloprocesorowego hybrydowego. Składa się on z nadrzędnego 32-bitowego procesora RISC klasy ARM oraz szeregu współpracujących z nim 8-bitowych mikrokontrolerów (rys. 5).



Rys. 5. Architektura systemu sterowania bezpośredniego

Fig. 5. Architecture of direct control system

Procesor nadrzędny koordynuje pracę całości podsystemu. Jego zadaniem jest pozyskiwanie danych pomiarowych, wysyłanie sygnałów wykonawczych do ustawiania serwo mechanizmów oraz pośredniczenie w komunikacji między modułami a aplikacją sterującą. Wykonuje weryfikację danych oraz nadzór nad manipulatorem, gdy główna aplikacja w module nadrzędnym nie jest aktywna. Dodatkowym zadaniem jest reagowanie na wykrycie niebezpiecznych wartości na czujnikach. Zastosowano mikrosystem SoC (ang. *System on Chip*) typu AT91SAM7X512 [9], charakteryzujący się 32-bitową arytmetyką i dużą wydajnością obliczeniową. Jest on wyposażony w szereg szybkich portów komunikacyjnych klasy CAN, Ethernet MAC, UART, SPI, I²C oraz wielokanałowy kontroler DMA, odciażający rdzeń od zadań komunikacyjnych. Procesor zawiera także sprzętowy port USB-UDP pozwalający na uzyskanie rzeczywistego strumienia danych rzędu 600 kB/s [10]. Całość modułu wyposażono w interfejs Bluetooth.

Do realizacji zadań bezpośredniej akwizycji danych oraz bezpośredniej współpracy z jednostkami napędowymi przyjęto architekturę wieloprocesorową, modułową. Procesory 8-bitowe generują przebiegi PWM i mogą być wykorzystywane do sterowania zarówno układami serw, jak i silnikami DC [11]. Wykonują także akwizycję danych z czujników analogowych i cyfrowych, wstępną obróbkę sygnałów pomiarowych – walidacja, filtracja [12] i skalowanie. Procesory pracują niezależnie od procesora nadrzędnego. Współpracują za pomocą lokalnej magistrali szeregowej, niezależnej od łącza komunikacyjnego z systemem nadrzędnym.

Jedną z ciekawszych funkcjonalności, w jaką został wyposażony manipulator, jest czujnik nacisku umieszczony w efektorze. Pozwala on precyzyjnie określić siłę uchwytu. Dzięki niemu manipulator potrafi dostosować nacisk do delikatnych przedmiotów. Innym czujnikiem zamocowanym w chwytaku jest termistor. Umożliwia on reagowanie na zmianę temperatury i ochronę chwytaka przed uszkodzeniem.

Architektura jest skalowalna i zapewnia łatwość dodawania kolejnych modułów. Jedną z jej zalet jest możliwość stosowania różnych typów procesorów i różnych rozwiązań programowych. Przykładowo, jeśli wymagane będzie zwiększenie możliwości obliczeniowych, można zastosować procesory klasy DSP.

Minimalizację obciążenia procesora nadrzędnego zadaniami komunikacji uzyskano dzięki zastosowaniu odpowiednich kanałów DMA. Uzyskuje się w ten sposób dodatkowy czas potrzebny na wykonanie operacji obliczeniowych przez rdzeń [13].

Implementacja została wykonana w języku C. Do realizacji programowej użyto bezpłatnych wersji środowisk Kail oraz AVR studio.

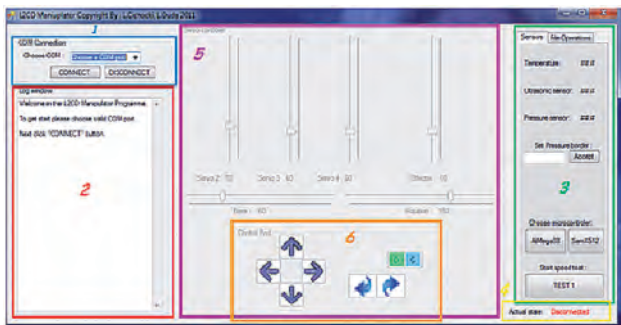
2.2. Podsystem sterowania nadrzędnego

System nadrzędny przeznaczony jest do bezpośredniej interakcji z operatorem. Obsługuje graficzny interfejs użytkownika i wykonuje wizualizację parametrów manipulatora. Ma zaimplementowany szereg usług, począwszy od konfiguracji parametrów pracy i zadawania nastaw serwo mechanizmów, przez akwizycję danych z czujników aż po operacje na strumieniach danych. Umożliwia również rejestrację w czasie rzeczywistym wartości mierzonych przez czujniki wielkości.

Wymagania pracy w reżimie „twardego” czasu rzeczywistego w części nadrzędnej nie są tak istotne, jak w części sterowania bezpośredniego. Pracuje on z częstotliwością rzędu 10 Hz. Zaprojektowano oraz wykonano dwa systemy nadrzędne, wykorzystujące dwa różne systemy operacyjne. Pierwsze z urządzeń to komputer klasy PC/notebook wyposażony w system Windows. Dla środowiska Windows zaprojektowano i zaprogramowano aplikację sterującą-rejestrującą, która umożliwia komunikację z manipulatorem przez łącza komunikacyjne USB, Bluetooth oraz RS-232.

Drugie ze stosowanych urządzeń to komputer personalny PDA (ang. *Personal Digital Assistance*) z zainstalowanym systemem operacyjnym Windows Mobile, z graficznym ekranem dotykowym, pozwalającym na interakcję z manipulatorem za pomocą rysika lub palca. Jedną z ważnych zalet tej koncepcji są małe wymiary, mała moc pobierana oraz wbudowane łącza bezprzewodowe (Wi-Fi, Bluetooth) i wbudowana kamera. Programowanie aplikacji dla systemu Mobile wiąże się ze stosowaniem innej metodyki, niż ma to miejsce w zwykłym systemie Windows. Mobile należy do grupy systemów wbudowanych (ang. *embedded system*). Naśladuje on pewne elementy ze „starszych wersji systemu Windows” w odniesieniu do elementów GUI, jednakże znacząco się różni na poziomie wewnętrznej budowy i jądra. Przenośny Mobile jest systemem zaprojektowanym głównie dla urządzeń typu: palmtopy, smartfony i palmfony. Skutkowało to koniecznością realizacji aplikacji z wykorzystaniem innych metod i sposobów w celu osiągnięcia analogicznej funkcjonalności aplikacji jak w komputerze PC.

Przed projektantem systemów i aplikacji wykorzystujących dwa różne systemy operacyjne, powstaje problem doboru takich narzędzi, jak: język programowania, środowisko, wykorzystane biblioteki a także interfejs graficzny do interakcji z potencjalnym użytkownikiem. W pracy założono wykorzystanie interfejsów graficznych w obu aplikacjach sterujących. Po wykonaniu badań literaturowych okazało się, że językiem oferującym najszerszą paletę możli-



Rys. 6. Przykładowe okno aplikacji
Fig. 6. Example of application window

wości interfejsu graficznego, w korelacji z wysoką funkcjonalnością i obiektowością, jest język C#. Jednym z najważniejszych elementów wpływających na wybór tego języka był fakt pełnej synchronizacji z platformą .NET Framework oraz środowiskiem Microsoft Visual Studio 2008 [14, 15]. Platforma .NET umożliwia szybkie projektowanie elementów graficznych okna aplikacji (ang. *rapid prototyping*) dzięki wykorzystaniu zbioru gotowych składników elementarnych okna. Korzyść takiego rozwiązania jest podwójna. Oba systemy zarówno Windows, jak i Windows Mobile mogą wykorzystać platformę .NET Framework do budowy interfejsu graficznego.

Graficzny interfejs użytkownika został zrealizowany na bazie funkcji oferowanych przez środowisko MS Visual Studio 2008 na licencji MSDN AA. Przykładowy wygląd okna przeznaczonego do sterowania ręcznego pokazano na rys. 6.

2.3. Zagadnienia integracji systemów

Zaproponowana architektura podziału na system sterowania bezpośredniego i nadrzędnego wymusiła konieczność zaprojektowania i realizacji modułów przeznaczonych do integracji obu części. Integracja dwóch asynchronicznie pracujących systemów jest zadaniem nietrywialnym i wymaga połączenia wiedzy z zakresu standardów komunikacyjnych, protokołów, sterowników programowych oraz budowy i działania systemów operacyjnych.

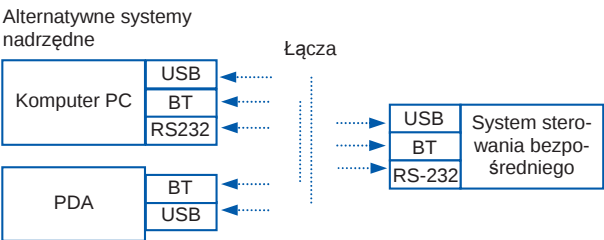
Zaimplementowane drogi integracji (rys. 7) wykorzystują trzy typy interfejsów komunikacyjnych: Bluetooth (BT), bezpośrednie łącze USB oraz interfejs RS-232.

W systemie sterowania bezpośredniego, oprogramowanie części komunikacyjnej zostało zbudowane w całości z kodów źródłowych. Dla łącza USB zdecydowano na implementację klasy CDC (ang. *Communications Device Class*) [16], charakteryzującej się dobrym wykorzystaniem teoretycznego pasma magistrali USB i możliwością uzyskania dużego strumienia danych.

W systemie Windows zastosowano następujące biblioteki komponentów:

- komunikacja USB – funkcje strumieni wejścia/wyjścia portu szeregowego – System.IO,
- komunikacja RS-232 – funkcje portu szeregowego – System.IO.Ports,
- komunikacja Bluetooth – biblioteka InTheHand 32.Feet kompatybilna z platformą .NET i językiem C# [17, 18].

W systemie Windows Mobile, do obsługi łącza Bluetooth wykorzystano funkcję wbudowanego stosu. Urządzenie pracowało w klasie SPP [19]. Oprogramowanie zarówno dla komputera PC, jak i urządzenia PDA zostało napisane w języku C#.



Rys. 7. Zaimplementowane możliwości integracji systemów
Fig. 7. Implemented ways of integration

Podczas eksperymentalnych testów urządzenia PDA okazało się, że zastosowanie standardowych komponentów programistycznych dla łącza USB pozwala jedynie na uzyskanie strumienia danych rzędu 64 kB/s. Polepszenie parametrów łącza jest możliwe przez realizację obsługi portu USB w języku C/C++.

3. Podsumowanie

Zaprojektowano i zrealizowano system mechatroniczny, złożony z manipulatora o pięciu stopniach swobody, oraz wieloprotocowy system sterowania, którego centralną częścią jest 32-bitowy mikrosystem SoC. Zintegrowano go z systemem nadrzędnym, zrealizowanym na dwóch różnych platformach. Opracowana i wykonana konstrukcja manipulatora wraz z modułami elektronicznymi i oprogramowaniem wielu systemów umożliwia dalszy rozwój i badania. W szczególności możliwe są badania różnych metod rozpraszania zadań sterowania na poszczególne procesory oraz implementacja zaawansowanych algorytmów sterowania. Od strony mechanicznej wykonana konstrukcja wymaga udoskonalenia w celu poprawy dokładności pozycjonowania.

Zaproponowana koncepcja zastosowania urządzenia klasy PDA pozwala na uzyskanie małych wymiarów, niskiego poboru mocy oraz łatwej rozszerzalności programowej i funkcjonalnej. Przewidywany jest rozwój systemu obejmujący integrację bezprzewodową z siecią Internet. Wbudowane łącze Wi-Fi otwiera drogę do zdalnego kontrolowania pracy manipulatora. Możliwe kierunki badań i realizacji z wykorzystaniem Internetu przedstawione są w [20, 21]. Planowany dalszy rozwój obejmuje także wykorzystanie wbudowanej kamery i implementację algorytmów przetwarzania obrazów. Wybrane algorytmy przeznaczone do zastosowań w robotyce można znaleźć w [4, 22].

Bibliografia

1. www.students.autodesk.com.
2. Lumsden J., Ortega-Sanchez C., *Modular Autonomous Robotics Platform for Educational Use*, TENCON IEEE Region 10 Conference, Fukuoka, 2010, 1577–1582.
3. Alves S., Rosário J.M., Filho H.F., Pegoraro R., *Environment for Teaching and Development of Mobile Robot Systems*, Electronics, Robotics and Automotive Mechanics Conference (CERMA), Morelos, 2010, 302–307.
4. Sagioglu S., Yilmaz N., *Web-based mobile robot platform for real-time exercises*, “Expert Systems with Applications”, 2009, 36, 3153–3166.
5. Lin H-Ch., Chen Ch.H., Huang G-S., Liu Y-Ch., Hsu W-Ch., *Design of Communication Interface and Control System for Intelligent Humanoid Robot*, [in:] *Computer Applications in Engineering Education*, Wiley Periodicals, Inc., 2010.
6. Puiu D., Moldoveanu F., Cernat M., *The Distributed Control of an Articulated Arm Robot*, International Conference on Electrical and Electronics Engineering (ELECO), Bursa, 2009, 247–252.
7. Lin Z., Wang T., Gao Q., Liu Y., *Design of Robot Platform Based on CAN bus*, International Conference on Electrical and Control Engineering (ICECE), Yichang, 2011, 645–648.
8. Miller D.P., Oelke M., Roman M.J., Villatoro J., Winton Ch.N., *The CBC: a LINUX-based Low-Cost Mobile Robot Controller*, IEEE International Conference on Robotics and Automation, Anchorage, 2010, 4633–4638.
9. *AT91SAM ARM-based flash MCU*, Doc. 6120I, Atmel, 2011.
10. Augustyn J., Bień A., *Właściwości czasowe interfejsu USB we wbudowanych systemach pomiarowo-sterujących*, „Przegląd Elektrotechniczny”, 2009, 7, 1–7.
11. Witkowski A., *Mikrokontrolery AVR – programowanie w języku C. Przykłady zastosowań*, Katowice, 2006.
12. Zieliński T., *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*, Warszawa 2007.
13. Augustyn J., *Projektowanie systemów wbudowanych na przykładzie rodziny SAM7S z rdzeniem ARM7TDMI*, Wydawnictwo IGSMiE PAN, Kraków, 302, 2007.
14. Templeman J., Vitter D., *Visual Studio .NET: .NET Framework*. Czarna księga, Gliwice 2003.
15. Jones A., *C# Księga przykładów*, Microsoft, 2006.
16. *Universal Serial Bus Class Definitions for Communications Devices*. Revision 1.2, [www.usb.org], November 16, 2007.
17. [http://32feet.net] – dokumentacja biblioteki In The Hand 32.Feet.
18. Huang A.S., Rudolph L., *Bluetooth essentials for programmers*, Cambridge, 2007.
19. *Specification of the Bluetooth system v1.2.*, Specification, Bluetooth SIG, 2003.
20. López D., Cedazo R., Sánchez F.M., Sebastián J.M., *Ciclope Robot: Web-Based System to Remote Program an Embedded Real-Time System*, “IEEE Transactions on Industrial Electronics”, 2009, vol. 56, nr 12, 4791–4797.
21. Park Y., Cho Y., Yang S.H., Choi E., Min D., *Remote Control System for User Created Robot Supporting Interoperability between Controller and Robot*, International Conference on Control, Automation and Systems, Kintex, 2010, 1371–1374.
22. Coman M., Stan S-D., Manic M., Balan R., *Application of Distance Measuring with MATLAB/Simulink*, 3rd Conference on Human System Interactions (HSI), Rzeszów, 2010, 113–118. ■

Design and implementation of a laboratory modular manipulator with five degrees of freedom with embedded control system

Abstract: This paper presents the design and practical implementation of the modular manipulator with five degrees of freedom. The conceptual design of mechanical parts is depicted. Manipulator platform uses multiprocessor, hybrid control system. This system was designed and developed. The manipulator can be used for laboratory and educational applications.

Keywords: manipulator, mechatronic system, embedded system, real-time system

dr inż. Jacek Augustyn

Pracownik Katedry Automatyki AGH. Zainteresowania naukowe obejmują zagadnienia związane z systemami wbudowanymi czasu rzeczywistego, zmienno-przecinkowymi procesorami sygnałowymi DSP oraz 32-bitowymi systemami SoC. Jest autorem m.in. dwóch monografii. Drugim obszarem zainteresowań jest sterowanie procesami przemysłowymi z użyciem regulacji predykcyjnej oraz projektowanie, implementacja i wdrażanie przemysłowych systemów sterowania.

e-mail: jag@agh.edu.pl



inż. Łukasz Cichocki

Absolwent kierunku Informatyka. Zainteresowania obejmują aplikacje wbudowane w obszarze rozwiązań mobilnych.

e-mail: lukasz.cichocki24@gmail.com



inż. Łukasz Duda

Absolwent kierunku Informatyka. Zainteresowania obejmują projektowanie, realizację oraz programowanie systemów wbudowanych mikroprocesorowych.

e-mail: lukasz.duda.1989@gmail.com

