

Robert ŁUKASZEWSKI, Piotr BILSKI, Krzysztof MROCZEK

POLITECHNIKA WARSZAWSKA, WYDZIAŁ ELEKTRONIKI I TECHNIK INFORMACYJNYCH, INSTYTUT RADIOELEKTRONIKI,
ul. Nowowiejska 15/19, 00-665 Warszawa

Wykorzystanie w dydaktyce rekonfigurowalnych przyrządów pomiarowo-sterujących i systemów wbudowanych

Dr inż. Robert ŁUKASZEWSKI

Od 1997 roku pracownik pracowni Komputerowej Techniki Pomiarowej, Instytutu Radioelektroniki Politechniki Warszawskiej. Prowadzone prace naukowo-badawcze oraz publikacje ściśle związane są z projektowaniem, modelowaniem i implementacją oprogramowania rozproszonych systemów pomiarowo-sterujących. Szczegółowy obszar zainteresowań obejmuje interfejsy pomiarowe, sieci komputerowe, projektowanie systemów, języki formalne, bezpieczeństwo systemów.

e-mail: rlukaszewski@ire.pw.edu.pl



Dr inż. Piotr BILSKI

Pracuje na stanowisku adiunkta na Wydziale Zastosowań Informatyki i Matematyki SGGW oraz w Instytucie Radioelektroniki Politechniki Warszawskiej. Na tym pierwszym wydziale pełni również funkcję prodziekana ds. studenckich. Jego zainteresowania naukowe obejmują zastosowania sztucznej inteligencji w naukach przyrodniczych i technicznych (np. analiza danych geotechnicznych, diagnostyka systemów analogowych) oraz projektowanie i analizę rozproszonych komputerowych systemów pomiarowych.

e-mail: pbilski@ire.pw.edu.pl



Dr inż. Krzysztof MROCZEK

Krzysztof Mroczek jest pracownikiem pracowni Komputerowej Techniki Pomiarowej w Instytucie Radioelektroniki Politechniki Warszawskiej. Jego działalność naukowa i dydaktyczna dotyczy metod projektowania układów cyfrowych, zastosowania układów FPGA, komputerowej techniki pomiarowej, a także wykorzystania systemów wbudowanych w rozproszonych systemach kontrolno-pomiarowych.

e-mail: K.Mroczek@ire.pw.edu.pl



1. Wstęp

W ostatnich latach rozwój techniki pomiarowej przyniósł trwałą integrację przyrządów pomiarowych z wbudowanymi systemami czasu rzeczywistego, programowo rekonfigurowanymi matrycami bramek logicznych (FPGA – ang. Field Programmable Gate Array) oraz różnymi mediami transmisji danych. Zaowocowało to powstaniem nowej klasy urządzeń zdolnych do programowego definiowania funkcji pomiarowych. Urządzenia te zostały wyposażone w możliwości wymiany informacji z innymi węzłami pomiarowymi oraz zarządzającymi. W wielu przypadkach nowoczesne urządzenia pomiarowo-sterujące pracują pod kontrolą systemu operacyjnego czasu rzeczywistego i są zdolne do spełnienia twardego wymaganie pracy w czasie rzeczywistym.

W odpowiedzi na rosnące potrzeby związane z programowaniem systemów sterowania przemysłowego czołowe firmy z branży automatyki stworzyły nową klasę sterowników przemysłowych – PAC (ang. Programmable Automation Controller). Sterowniki PAC łączą w sobie wytrzymałość i odporność sterownika PLC (ang. Programmable Logic Controller) z funkcjonalnością komputera osobistego w ramach jednej otwartej i elastycznej architektury programowej. Sterowniki te zapewniają szerokie możliwości programowe, takie jak zaawansowane sterowanie, łączność, rejestrowanie danych i przetwarzanie sygnałów.

W głównej części artykułu przedstawiono dwie nowoczesne platformy sprzętowe wykorzystywane w laboratorium studenckim na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej (EiT PW). Obie platformy pozwalają na projektowanie urządzeń z systemami wbudowanymi (również czasu rzeczywistego) jako elementów większych, rozproszonych systemów pomiarowo-sterujących (RSPS).

2. Charakterystyka technologii RIO

Technologia RIO (ang. Reconfigurable Input/Output) [1] firmy National Instruments (NI) umożliwia tworzenie rozproszonych rekonfigurowanych systemów sterowania i akwizycji danych. Dzięki zintegrowanemu układowi FPGA, sterownikowi zarządzanemu przez system czasu rzeczywistego oraz szerokiej gamie urządzeń wejściowych i wyjściowych zapewnia ona wysoki poziom wydajności i elastyczności. Dzięki temu dokładne poznanie technologii RIO gwarantuje nabycie umiejętności koniecznych do projektowania rozproszonych systemów pomiarowo-sterujących z wykorzystaniem większości obecnie dostępnych na rynku nowoczesnych platform sprzętowych. Z drugiej strony technologia RIO umożliwia programowanie graficzne w środowisku LabVIEW (w tym programowanie układu FPGA), co dodatkowo predysponuje ją do wykorzystania w procesie kształcenia studentów.

Streszczenie

Artykuł przedstawia dwie platformy sprzętowe stosowane w laboratorium studenckim. Pierwszą z nich jest rekonfigurowalny system RIO firmy National Instruments, drugą - płyta rozwojowa przeznaczona do rozwoju oprogramowania wbudowanego do węzłów sieci czujnikowych i komputerowej akwizycji danych. Obie platformy pozwalają na projektowanie urządzeń z systemami wbudowanymi (również czasu rzeczywistego) jako elementów większych, rozproszonych systemów pomiarowo-sterujących (RSPS).

Słowa kluczowe: systemy rekonfigurowalne, systemy wbudowane, systemy czasu rzeczywistego, sieci czujnikowe.

Use of reconfigurable measuring and control devices and embedded systems in education

Abstract

The paper presents two hardware platforms used for didactic purposes in a student laboratory. The first one is a reconfigurable system by the National Instruments CompactRIO. It is an industrial computer aimed at automation and control of processes, for example during diagnostic procedures and in the factory production. As NI solutions often become worldwide standards, it is reasonable to train future engineers in programming their systems. With CompactRIO, the main effort of a designer is to create reliable applications working in the Real-Time mode. The second platform is an evaluation board usually applied as a node in the sensory network or to the computer data acquisition. The main problem here is the FPGA array configuration to perform measuring and controlling tasks. Both solutions can be used to design embedded systems being a part of larger systems. The paper presents the structure and architecture of both devices, including description of programming methodologies. Technical characteristics of both devices are described in detail, revealing their advantages and limitations. Then, applications to the process of engineer education are presented, based on the examples from the Institute of Radioelectronics, Warsaw University of Technology. These include both laboratory exercises and student projects, where they can solve particular engineering tasks using the presented hardware. The paper is supplemented with conclusions and future prospects of the introduced systems in didactics.

Keywords: reconfigurable systems, embedded systems, real time control systems, sensor networks.

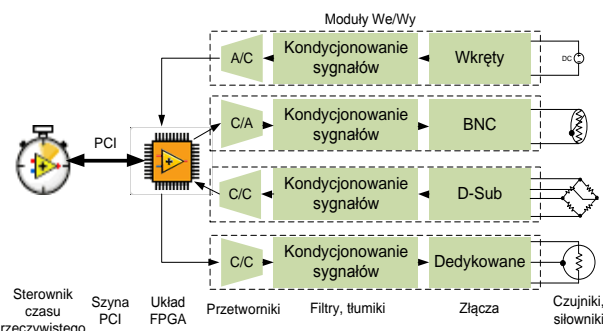
Technologia RIO umożliwia definiowanie własnych sprzętowych obwodów pomiarowych przy wykorzystaniu układów FPGA oraz graficznego środowiska LabVIEW, do np.: szybkiej analizy sygnałów i podejmowania decyzji już na poziomie układu wejściowego/wyjściowego (We/Wy). Wszystkie platformy sprzętowe RIO umożliwiają tworzenie różnych konfiguracji sprzętowych i programowych z wydajnymi układami We/Wy z pełną kontrolą czasu wykonania poszczególnych operacji w systemie. Technologia RIO jest dostępna w postaci kilku platform sprzętowych: kart PCI (ang. Peripheral Component Interconnect), modułów serii R DAQ (ang. Data Acquisition) do systemu kasetowego NI PXI (ang. PCI eXtensions for Instrumentation), autonomicznych płyt rozwojowych NI Single-Board RIO oraz systemów CompactRIO (cRIO). Z wymienionych platform ta ostatnia posiada najszersze zastosowanie z uwagi na największą elastyczność projektowania.

3. Architektura platformy cRIO

Platforma NI cRIO jest zwartym systemem skonstruowanym z myślą o zastosowaniach przemysłowych oraz w systemach akwizycji danych. Platforma ta sprawdza się najlepiej w zastosowaniach, w których niezbędna jest wysoka wydajność, niezawodność, dyscyplina czasowa oraz elastyczność. Przemysłowa obudowa cRIO posiada wbudowany układ FPGA podłączony bezpośrednio do modułów pomiarowych. Pozwala to na otwarty dostęp do zasobów sprzętowych na niskim poziomie (np. do wejść i wyjść przetworników). Dołączony lub zintegrowany kontroler z systemem czasu rzeczywistego zapewnia wykonanie programów w założonych twardych rygorach czasowych. Programowanie aplikacji czasu rzeczywistego oraz układów FPGA odbywa się przy użyciu LabVIEW. Platforma cRIO umożliwia tworzenie systemów rozproszonych opartych o sieć Ethernet.

W obudowie zintegrowany jest układ FPGA posiadający od 1 do 3 milionów bramek logicznych, a w jej slotach można umieścić od 4 do 8 modułów We/Wy serii „C”. Konfiguracja ta stanowi kompletny, rekonfigurowany system do skupionych i sieciowych aplikacji akwizycji danych. Można wyróżnić dwie konstrukcje cRIO: zintegrowaną (seria cRIO-907x), gdzie w obudowie zintegrowano zarówno układ FPGA jak i sterownik czasu rzeczywistego oraz modułową, gdzie w obudowie umieszczony jest tylko układ FPGA, natomiast sterownik czasu rzeczywistego dołączany jest za pomocą specjalnego złącza. Druga opcja pozwala na elastyczny dobór sterownika z dostępnej gamy urządzeń typów cRIO-900x i cRIO-901x. Każdy z dostępnych sterowników posiada wbudowany procesor przemysłowy (np. Freescale MPC5200 400 MHz) oraz zainstalowany system czasu rzeczywistego: Wind River VxWorks [2]. System ten pozwala na deterministyczne wykonanie kodu aplikacji do sterowania w czasie rzeczywistym, analizy, zapisu danych i komunikacji. Sterowniki czasu rzeczywistego posiadają wbudowany port Ethernet służący do komunikacji przez sieć z komputerem nadzorcym (ang. host) w czasie osadzenia oprogramowania oraz wykonania zadania pomiarowo-sterującego. Dodatkowo w systemie zostały wbudowane serwery protokołów sieciowych HTTP (ang. Hypertext Transfer Protocol) i FTP (ang. File Transfer Protocol). Wbudowany serwer HTTP umożliwia publikację panelu do zdalnych użytkowników systemu w trybie monitorowania oraz sterowania.

Każdy z modułów We/Wy serii „C” jest bezpośrednio podłączony do układu FPGA co umożliwia niskopoziomą kontrolę czasu akwizycji, dużą precyzję sterowania oraz wydajne przetwarzanie sygnałów. Układ FPGA umożliwia osadzanie algorytmów pomiarowych i sterowania z kontrolą czasową, dokładnym określeniem momentu wyzwania, synchronizacją oraz z przetwarzaniem sygnałów analogowych i cyfrowych. Układ FPGA jest podłączony do sterownika systemu czasu rzeczywistego przez lokalną szynę PCI (rys. 1). Sterownik odpowiada za analizę danych w czasie rzeczywistym, przetwarzanie i zapis danych, a także umożliwia komunikację sieciową z komputerem nadzorcym i ew. innymi węzłami systemu.



Rys. 1. Architektura sprzętowa CompactRIO
Fig. 1. Hardware structure of the CompactRIO platform

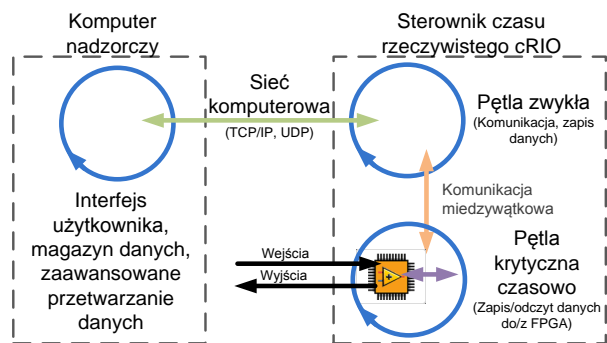
Programowanie układu FPGA oraz aplikacji systemu czasu rzeczywistego odbywa się na komputerze w środowisku LabVIEW z zainstalowanymi dodatkowymi modułami: LabVIEW Real-Time oraz LabVIEW FPGA. Środowisko LabVIEW zawiera wbudowane mechanizmy transferu danych z modułów We/Wy do układu FPGA oraz z układu FPGA do systemu czasu rzeczywistego. Wbudowane funkcje umożliwiają m.in. tworzenie aplikacji wielowątkowych, systemów kontroli w czasie rzeczywistym, analizy i składowania danych oraz komunikacji sieciowej.

Opracowane na etapie projektowania aplikacje osadzone są w układach FPGA przez środowisko LabVIEW. Moduł LabVIEW FPGA kompiluje aplikację LabVIEW w procesie kompilacji. Graficzna postać kodu tłumaczona jest na postać tekstową kodu VHDL (ang. Very high speed integrated circuits Hardware Description Language). Następnie wywoływane są narzędzia kompilacji Xiling ISE [3]. W trakcie procesu kompilacji wykonywana jest optymalizacja FPGA w celu redukcji ilości bramek logicznych i stworzenia optymalnej aplikacji LabVIEW. W dalszej kolejności projekt jest syntetyzowany w złożonych strukturach krzemowych układu FPGA. Efektem końcowym jest strumień bitowy zawierający instrukcje programistyczne LabVIEW, które są pobierane przez FPGA. Po uruchomieniu aplikacji, strumień bitowy jest ładowany do układu FPGA. Strumień ten rekonstruuje tablice logiczne układu FPGA.

Takie podejście umożliwia definiowanie własnych urządzeń We/Wy bez konieczności tworzenia układów sprzętowych oraz bez znajomości specjalizowanych języków programowania tych układów (np. VHDL).

Projektowanie oprogramowania platformy RIO składa się z trzech zadań: oprogramowania układu FPGA, stworzenia aplikacji systemu czasu rzeczywistego oraz aplikacji operatorskiej działającej pod kontrolą systemu Microsoft Windows. Tworzenie wszystkich tych elementów oprogramowania systemu pomiarowo-sterującego odbywa się w środowisku LabVIEW.

Programowanie na platformie cRIO rekonfigurowalnych systemów sterowania i akwizycji danych wymaga stworzenia aplikacji FPGA do sterowania wejściami i wyjściami w modułach We/Wy. Drugim elementem jest aplikacja czasu rzeczywistego z dwoma pętlami głównymi pracującymi w osobnych wątkach (rys. 2) i komunikującymi się ze sobą za pomocą deterministycznych mechanizmów wymiany danych i synchronizacji. Pierwszą z pętli jest deterministyczna pętla (ang. time-critical loop) zawierająca: zmiennoprzecinkowe operacje sterowania systemowego, przetwarzanie sygnałów, analizę danych i podejmowanie decyzji systemowych. Drugą pętlą jest zwykła pętla zawierająca zazwyczaj: zapis danych, zdalny panel w postaci strony WWW, komunikację sieciową, obsługę portu szeregowego kontrolera. Trzecim głównym elementem oprogramowania systemu jest aplikacja operatora, która jest uruchamiana na komputerze PC, laptopie lub komputerze przemysłowym w kasecie PXI. Aplikacja ta zawiera zwykle zdalny graficzny panel użytkownika, zaawansowane funkcje przetwarzania danych w trybie lokalnym (ang. postprocessing), przetwarzanie danych historycznych.



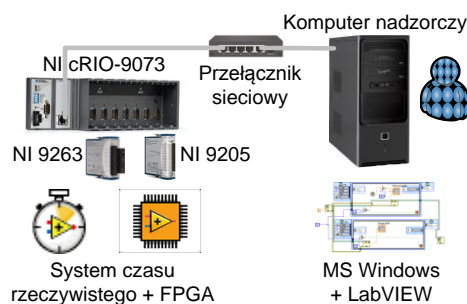
Rys. 2. Architektura oprogramowania na platformie CompactRIO
Fig. 2. Software architecture in the CompactRIO platform

4. Stanowisko laboratoryjne

Stanowisko laboratoryjne (rys. 3) składa się z:

- obudowy cRIO-9073 z wbudowanym układem FPGA Spartan 3 (wyposażonym w 2 miliony bramek) oraz sterownikiem z procesorem przemysłowym (266 MHz) pracującym pod kontrolą systemu czasu rzeczywistego Wind River VxWorks;
- czterokanałowego 16-bitowego modułu wyjść analogowych NI 9263;
- 32-u kanałowego 16-bitowego modułu wejść analogowych NI 9205;
- nadzorczego komputera PC z systemem Windows XP;
- obiektu badanego;

Obudowa cRIO-9073 oraz komputer PC połączone są do sieci lokalnej i komunikują się za pomocą protokołów TCP/IP.



Rys. 3. Stanowisko laboratoryjne
Fig. 3. Laboratory test stand

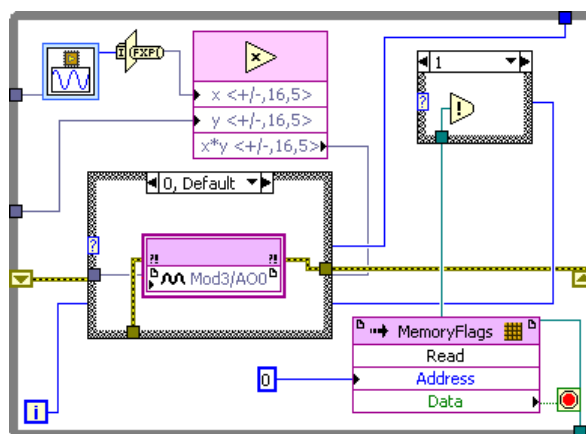
5. Zadania projektowe

Zadania projektowe z zakresu oprogramowania platformy cRIO na przedmiocie RSPK rozdzielone są na trzyosobowe grupy studenckie. Przykładowe zadanie projektowe na przedmiocie RSPK polega na zaprojektowaniu w środowisku LabVIEW 2010 oprogramowania sterownika czasu rzeczywistego służącego do pomiaru charakterystyki amplitudowej badanego obiektu za pomocą systemu złożonego ze sterownika cRIO-9073 oraz dwu modułów serii C: NI 9263 i NI 9205. W tym samym środowisku należy zaprojektować aplikację nadzorczą współpracującą z aplikacją sterownika. Aplikacja nadzorczą ma za zadanie pobrać parametry pomiaru (zakres zmian amplitudy, liczba punktów pomiarowych itp.) z tabeli bazy danych MySQL, pobrać i zapisać w innej tabeli bazy danych wyniki pomiarów oraz wyświetlić je w postaci wykresu na swoim panelu. Dodatkowo aplikacja nadzorczą powinna wyświetlać na bieżąco stopień zaawansowania procesu pomiarowego oraz umożliwiać prezentację na wykresie historycznych danych wcześniej zapisanych w bazie danych. Aplikacja nadzorczą powinna powiadomić użytkownika o wykrytych błędach

wykonania aplikacji pomiarowej i innych nieprawidłowościach działania systemu pomiarowego poprzez wyświetlenie odpowiedniego komunikatu na panelu oraz wysłanie do operatora systemu e-maila na wskazany adres.

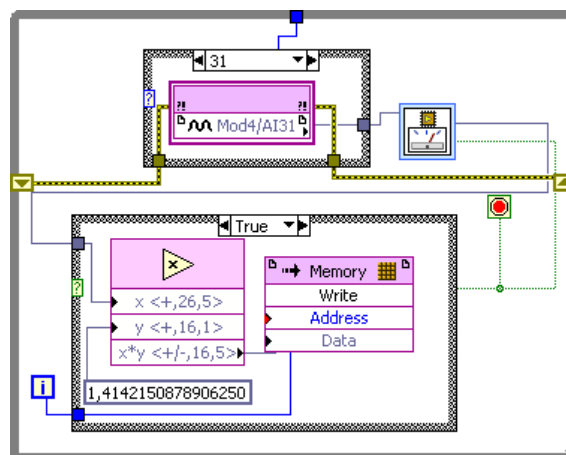
Realizacja przykładowego projektu polega na zaprojektowaniu trzech aplikacji zgodnie z architekturą cRIO. Pierwsza z nich, osadzona na FPGA realizuje funkcje sterowania wejścia obiektu badanego i pomiaru wartości napięcia na jego wyjściu. Sterowanie wejścia obiektu badanego odbywa się za pomocą sygnałów generowanych na wyjściach modułu NI 9263. W diagramie zrealizowanego w ramach projektu programu (rys. 4) funkcja sterowania wejściem obiektu badanego jest realizowana w następujący sposób:

- podprogram (tzw. Express VI) „Sine Wave Generator” tworzy na bieżąco 1024 elementową tablicę wartości fali sinusoidalnej o stałej amplitudzie;
- elementy tablicy wymnażane są z wartością amplitudy sygnału na wejściu obiektu badanego podaną z panelu użytkownika;
- wartości te są podawane na wejście obiektu „FPGA I/O”, który steruje wyjściem „AO0” modułu NI 9263 osadzonego w trzeciej szczelinie (ang. slot) obudowy cRIO.



Rys. 4. Fragment diagramu aplikacji FPGA (obsługa wyjść)
Fig. 4. Fragment of the FPGA application diagram (output handling)

Pomiar wartości skutecznej napięcia na wyjściu obiektu badanego odbywa się przy użyciu modułu NI 9205. W każdym obiegu pętli (rys. 5) odczytywana jest wartość napięcia z wejścia analogowego „AI31”, następnie jest wstępnie przetwarzana i zapisywana do pamięci o adresie określonym przez zmienną sterującą pętlą „i”. Zapis do pamięci realizuje obiekt „Memory” z wykorzystaniem metody „Write”.



Rys. 5. Fragment diagramu aplikacji FPGA (obsługa wejść)
Fig. 5. Fragment of the FPGA application diagram (input handling)

Przedstawiony sposób realizacji projektu w części dotyczącej układu FPGA jest jednym z wielu możliwych. Wykonawcy projektów mogą korzystać z innych mechanizmów dostępnych w paletach LabVIEW takich jak: FPGA I/O, Memoty & FIFO, FPGA Interface.

Druga aplikacja jest osadzona na sterowniku czasu rzeczywistego. Główną rolą tej aplikacji jest odebranie danych z aplikacji FPGA i wstępne ich przetworzenie, przekazanie danych sterujących do tej aplikacji oraz komunikacji z aplikacją użytkownika za pomocą zmiennych sieciowych. W przykładowej realizacji projektu wykorzystano obiekty „Read/Write Control Function” oraz metody „Read” i „Data” do odczytu danych zapisanych do pamięci przez aplikację FPGA.

Trzecia aplikacja działająca na komputerze nadzorczym służy do komunikacji z użytkownikiem, zapisuje i odczytuje dane do/z bazy danych oraz wykonuje zaawansowane przetwarzanie danych pomiarowe (ang. postprocessing). Komunikacja z aplikacją czasu rzeczywistego jest zrealizowana za pomocą zmiennych sieciowych. Panel użytkownika aplikacji pozwala na wprowadzenie parametrów pomiaru, określenie numeru wejść i wyjść modułów We/Wy, wyświetlenie w postaci wykresu charakterystyki mierzzonego obiektu, wyświetlenie paska postępu obecnie realizowanej procedury pomiarowej, zapisanie wyników pomiaru do bazy danych, wykonanie pomiaru charakterystyki oraz wprowadzenie innych potrzebnych parametrów konfiguracyjnych systemu.

Opisane oprogramowanie systemu realizuje wiele innych funkcji opisanych w założeniach projektowych, jednakże dokładny ich opis wykracza poza zakres niniejszego artykułu.

Realizacja przedstawionego projektu ma dużą wartość dydaktyczną. Studenci w praktyce stosują różne mechanizmy komunikacji międzyprocesowej i sieciowej, nabywają praktycznej umiejętności projektowania i uruchamiania aplikacji systemów wbudowanych oraz aplikacji systemów czasu rzeczywistego.

6. Płyta edukacyjna z mikrokontrolerem klasy ARM7

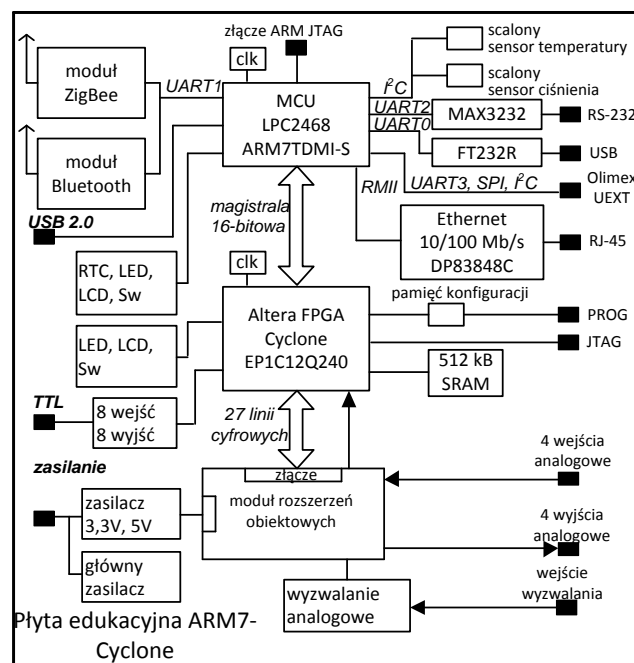
Opracowano własną płytę edukacyjną, przeznaczoną do realizacji mniejszych elementów rozproszonych systemów pomiarowo-sterujących, takich jak węzły czujnikowe lub proste punkty dostępowe. Zdecydowano się na wybór 32-bitowego mikrokontrolera z rdzeniem ARM7. Mikrokontrolery tej klasy dają większe możliwości aplikacyjne i większą moc obliczeniową niż ich 8-bitowi poprzednicy i mogą być brane pod uwagę w tego typu aplikacjach. Większość dostępnych tanich płyt edukacyjnych zawiera układ mikrokontrolera lub FPGA, podsystem pamięci zewnętrznej oraz peryferia komunikacyjne. W opracowanym projekcie zintegrowano na jednej płycie podsystem mikrokontrolera LPC2468 firmy NXP Semiconductors (32-bitowy rdzeń ARM7TDMI-S), średniej wielkości układ FPGA oraz złącze rozszerzeń do przyłączania dodatkowych modułów rozszerzeń. Schemat blokowy płyty przedstawiono na rys. 6.

Wybrano mikrokontroler z rodziny LPC2xxx, która zawiera zarówno tanie małe układy, jak również układy bogato wyposażone w peryferia komunikacyjne. Maksymalna wartość częstotliwości taktowania procesora wynosi 72 MHz. Pojemność wbudowanej pamięci (512 kB FLASH i 98 kB SRAM) jest wystarczająca do implementacji systemów operacyjnych czasu rzeczywistego, np. μ C/OS-2, FreeRTOS. Architektura wewnętrzna jest optymalizowana pod kątem aplikacji transmisji danych, co umożliwia realizację transmisji danych w trybie DMA poprzez wydzielone bufor pamięci SRAM, dostępne dla kontrolerów Ethernet-u i USB[4].

Na płycie umieszczono gniazdo do przyłączenia modułów ZigBee XBee/XBee-PRO firmy Digi oraz zamontowano moduł komunikacyjny Bluetooth ROK101008 firmy Ericsson. Komunikacja z tymi modułami jest realizowana za pośrednictwem 4-liniowego interfejsu szeregowego UART1. Zewnętrzne moduły rozszerzeń mogą być dołączone poprzez gniazdo rozszerzeń UEXT oraz przez port RS-232. Konwerter USB-UART, zrealizowany na układzie FTDI FT232R, realizuje kanał transmisji

szeregowej. Komunikacja z układem FPGA i dodatkowym modulem pamięci SRAM o pojemności 512 kB odbywa się poprzez magistralę pamięci zewnętrznej mikrokontrolera, w trybie transmisji 16-bitowych. Podsystem FPGA zawiera układ Cyclone EP1C12Q240 o pojemności 12060 elementów logicznych i 29952 B pamięci SRAM. Głównym przeznaczeniem układu FPGA jest realizacja sterowników sprzętowych dla modułów rozszerzeń obiektowych oraz akceleracja operacji kryptograficznych i DSP.

Dodatkowe moduły można integrować poprzez złącze rozszerzeń obiektowych. Sterowanie jest realizowane za pośrednictwem układu FPGA, w którym może być zrealizowany odpowiedni sterownik lub układ sprzęgający z mikrokontrolerem LPC2468. Złącze jest przystosowane do realizacji modułów przetworników pomiarowych A/C i C/A.



Rys. 6. Schemat blokowy płyty edukacyjnej ARM7-Cyclone
Fig. 6. Block diagram of the ARM7-Cyclone education board

7. Tematyka projektów dydaktycznych i rozwojowych

Zintegrowanie podsystemów mikrokontrolera i FPGA oraz możliwość dołączania dodatkowych modułów, daje dość duże możliwości obliczeniowe, wykraczające poza poziom typowy dla prostych węzłów sieci czujnikowych. W tego typu aplikacjach wystarczające może być użycie 8-bitowych mikrokontrolerów, o małym poborze mocy. Przykładowe dziedziny aplikacyjne dla projektów dydaktycznych przedstawiono poniżej.

Przetworniki pomiarowe z komunikacją bezprzewodową

Typowy węzeł sieci czujnikowej składa się z modułu komunikacyjnego, mikrokontrolera i właściwego przetwornika pomiarowego lub aktuatora. Pierwszym zadaniem studentów jest skonfigurowanie sprzętu oraz zapoznanie się z programowaniem mikrokontrolera, obsługą portów, peryferiów wewnętrznych, obsługą przerwań. Następnie z algorytmami programowania peryferiów komunikacyjnych oraz modułów zewnętrznych. Kolejnym krokiem jest zapoznanie się z funkcjami stosu komunikacyjnego, konfiguracją, topologią, adresowaniem w sieci. Realizacja docelowej aplikacji wymaga utworzenia sieci, składającej się z węzła nadrzędnego (kontrolera lub koordynatora) oraz kilku węzłów czujnikowych. Można to osiągnąć poprzez wykorzystanie dodat-

kowych modułów komunikacyjnych i uniwersalnych płytek ewaluacyjnych. Oddzielną rzeczą jest przygotowanie oprogramowania dla węzła nadrzędnego, zazwyczaj realizowanego w postaci komputera PC.

Konwertery interfejsów i procesory sieciowe

Zadanie polega na opracowaniu własnego protokołu komunikacyjnego i oprogramowania do komunikacji z przetwornikiem pomiarowym przez sieć TCP/IP, lub realizacja konwertera interfejsów, np. TCP/IP – IEC 625.2. Inna opcja to skorzystanie z opracowanych bibliotek komunikacyjnych i realizacja aplikacji zdalnego dostępu do przyrządów lub przetwornika. W pierwszym przypadku można ukierunkować pracę na zapoznanie się z programowaniem kontrolera Ethernet-u w mikrokontrolerze i w mniejszym stopniu na implementacji TCP/IP. Aplikację służącą do przetestowania programu konwertera stosunkowo szybko można napisać, np. w środowisku LabVIEW.

Bardziej zaawansowanym projektem jest realizacja procesora sieciowego NCAP lub węzła czujnikowego zgodnie z wymaganiami standardu IEEE 1451 [5-6].

Serwery pomiarowe

Istnieją darmowe opracowania stosów TCP/IP na procesory z rdzeniem ARM, np. uIP, NicheLite, które działają bez przerw na mikrokontrolerze LPC2468. Jako przykłady zagadnień problemowych można wymienić: zapoznanie się API stosu, opracowanie protokołu aplikacyjnego, generacja stron www o zmiennej zawartości, realizacja szyfrowania, ocena wpływu szyfrowania na parametry serwera, przyspieszanie operacji szyfrowania w układzie FPGA. Zainteresowani dyplomanci mogą zagłębić się w tematykę realizacji interfejsu LXI.

Przetworniki pomiarowe z interfejsem USB

Mikrokontroler LPC2468 posiada rozbudowany moduł interfejsu USB 2.0 full-speed (12 Mb/s). Można skonfigurować interfejs do obsługi punktów końcowych każdego rodzaju: kontrolnych, masowych, przerwaniowych i izochronicznych. Transmisje danych transferów USB mogą być realizowane w trybie DMA. Nabywanie wiedzy umożliwiającej samodzielne oprogramowanie komunikacji za pomocą USB wiąże się z zapoznaniem się z wieloma zagadnieniami i podjęciem odpowiednich decyzji projektowych. Należy zapoznać się z właściwościami poszczególnych typów transferów USB, które różnią się istotnie pod względem przepustowości transmisji danych a także sposobu reakcji po wykryciu błędów transmisji. Wybór klasy urządzenia USB ma wpływ na oprogramowanie komunikacji przez USB od strony komputera PC, które może być realizowane za pomocą standardowych sterowników programowych, udostępnianych razem z systemem operacyjnym dla niektórych klas urządzeń USB, lub wymagać napisania własnego sterownika. Oprogramowanie komunikacji USB od strony mikrokontrolera jest również złożonym zagadnieniem, które może być istotnie uproszczone poprzez użycie odpowiednich bibliotek programistycznych, zarówno darmowych jak i płatnych.

Programowanie w środowisku systemu operacyjnego

Przejście od modelu sekwencyjnego programowania w natywnym środowisku procesora do aplikacji wielowątkowej można osiągnąć poprzez osadzenie odpowiedniego systemu operacyjnego. Poszczególne części programu mogą być podzielone na odrębne zadania, wykonywane równolegle. W przypadku niektórych

takich systemów operacyjnych, np. Linuksa, dostęp do sprzętu ulega standaryzacji dzięki warstwie sterowników lub dodatkowych modułów. Na opracowanej płycie możliwe jest osadzenie kilku systemów operacyjnych czasu rzeczywistego (RTOSs), w tym FreeRTOS oraz $\mu\text{C}/\text{OS}-2$.

Należy zaznaczyć, że wymienione tematy projektów wymagają dużego nakładu pracy oraz odpowiedniego zasobu wiedzy teoretycznej od studenta. Należy zapoznać się z odpowiednią dokumentacją techniczną, liczącą nawet setki stron tekstu. Niezbędny jest odpowiedni poziom dojrzałości informatycznej, który pozwala na wybór odpowiednich modeli dla realizowanego oprogramowania i samodzielną pracę. Taki poziom dojrzałości informatycznej studenci mogą osiągnąć po kilku latach studiowania lub praktykowania.

8. Podsumowanie

Wykorzystanie platformy cRIO pozwala na projektowanie systemów charakteryzujących się wysoką wydajnością w porównaniu z rozwiązaniami czysto programowymi, porównywalną z rozwiązaniami czysto sprzętowymi. Dzięki otwartej architekturze, oprócz zalet technicznych, platforma cRIO ma bardzo duży potencjał dydaktyczny. Studenci, podczas realizacji projektu, nabywają m.in. praktycznej umiejętności projektowania i uruchamiania aplikacji systemów wbudowanych oraz aplikacji systemów czasu rzeczywistego oraz w praktyce poznają zaawansowane mechanizmy komunikacji międzyprocesowej i sieciowej.

Opracowana płyta edukacyjna pozwala na realizację prototypów elementów sieci czujnikowych. Praktyczne zbadanie możliwości jakie posiada typowy 32-bitowy mikrokontroler z rdzeniem ARM7, oprócz zdobycia konkretnej wiedzy, powinno być przydatne do oceny możliwości wykorzystania tej platformy w takich projektach. Wybór procesora i platformy sprzętowo-programowej do konkretnego projektu będzie łatwiejszy, jeżeli będzie opierał się zarówno na wiedzy jaki i na zdobytym doświadczeniu praktycznym.

Autorzy artykułu pragną podziękować panu A. Prokopowiczowi za jego wkład w opracowanie płyty edukacyjnej.

9. Literatura

- [1] National Instruments, "NI CompactRIO – Reconfigurable Control and Acquisition System": <http://zone.ni.com/devzone/cda/tut/p/id/2856>
- [2] Wind River, Build reliable and optimized embedded systems with the world's most widely adopted RTOS: <http://www.windriver.com/products/vxworks/>
- [3] Denton J. Dailey: Programming Logic Fundamentals Using Xilinx ISE and CPLDs, Prentice Hall, 203 pages. Introduction to PLDs using Xilinx ISE, 2004.
- [4] LPC24XX User manual, Rev. 04 — 26 August 2009, UM10237, NXP Semiconductors.
- [5] IEEE 1451.0-2007, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats.
- [6] IEEE 1451.5-2007 IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.

otrzymano / received: 21.06.2011

przyjęto do druku / accepted: 03.10.2011

artykuł recenzowany