Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej Nr 65

VI Konferencja

e-Technologie w Kształceniu Inżynierów eTEE'2019

Politechnika Gdańska, 19-20 września 2019

doi: 10.32016/1.65.18

LABORATORYJNY SYSTEM CYFROWY PROGRAMOWANY PRZEZ ETHERNET OPARTY NA MAGISTRALI SPI

Krystyna NOGA[1], Dorota RABCZUK[2]

1. Uniwersytet Morski Gdynia, Katedra Automatyki Okrętowej
tel.: 585586458 e-mail: k.noga@we.umg.edu.pl
2. Uniwersytet Morski Gdynia, Katedra Telekomunikacji Morskiej
tel.: 585586552 e-mail: d.rabczuk@we.umg.edu.pl

Streszczenie: W artykule zaprezentowano cyfrowy system laboratoryjny oparty na magistrali SPI z mikrokontrolerem w roli urządzenia Master oraz różnymi urządzeniami Slave podłączonymi do magistrali (cyfrowy termometr, potencjometr, pamięć EEPROM, układ programowalny CPLD). Założeniem projektu jest stworzenie bazy sprzętowo-programistycznej dla rozwojowego systemu zdalnie sterowanego oraz programowanego przez Ethernet. Użytkownik przez stronę internetową ma możliwość wysłania żądania tcp/http dla urządzeń na magistrali SPI, a w odpowiedzi http otrzymuje informacje o stanie urządzeń. System jest przystosowany do zdalnego ładowania nowych wersji programu do pamięci mikrokontrolera po Ethernecie.

Słowa kluczowe: układy cyfrowe, magistrala SPI, systemy wbudowane, układy programowalne.

1. WPROWADZENIE

Układy cyfrowe wykorzystywane w systemach wbudowanych i internetu rzeczy są powszechnie wyposażane w co najmniej jeden interfejs szeregowy, np. SPI (Serial Peripheral Interface), takie rozwiązanie ułatwia przyłączenie urządzeń cyfrowych do systemu. Rynek elektroniczny oferuje szeroką gamę urządzeń sterowanych po SPI, w tym: potencjometry, pamięci, wyświetlacze, ekspandery wejść, różne czujniki a także adapter Ethernet. Algorytm magistrali SPI można również zaimplementować w układzie programowalnym CPLD (Complex Programmable Logic Devices) lub FPGA (Field Programmable Gate Array), który może pracować w jednym systemie z mikrokontrolerem i czujnikami cyfrowymi. Układ CPLD oraz mikrokontroler mogą pełnić różne funkcje w projektowanym systemie, z uwagi na inną funkcjonalność wynikającą z odmiennej logicznej struktury wewnętrznej tych urządzeń. Układ CPLD ze względu na szybkie przetwarzanie równoległe sygnałów powinien wykonywać algorytmy obliczeniowe, podczas gdy mikrokontroler wyposażony standardowo w wiele interfejsów jest przystosowany do kontroli czujników i prowadzenia transmisji danych po magistralach systemowych. W prezentowanej koncepcji laboratoryjnego systemu cyfrowego mikrokontroler pełni na magistrali SPI rolę urządzenia Master, które za pośrednictwem adaptera Ethernetu utrzymuje kontakt z odległym stanowiskiem użytkownika utworzonym na komputerze PC. Pozwala to na wykorzystanie w dydaktyce systemu zdalnie sterowanego i zdalnie programowanego.

W opracowanym i zbudowanym stanowisku aspekt e-learningu polega na wykorzystaniu środowiska programistycznego do zdalnego ładowania programu mikrokontrolera przez sieć Ethernet. Wsparcie takie zapewnia między innymi środowisko Arduino IDE (Integrated Development Environment) dla urządzeń z kontrolerem Ethernet. Do pamięci mikrokontrolera można załadować po Ethernecie nie tylko program wykonywalny, ale również stronę internetową użytkownika systemu napisaną w języku HTML (Hypertext Markup Language), jeśli nie przekracza ona rozmiarów pamięci programu mikrokontrolera. To samo łącze może być wykorzystane do zdalnego debugowania i wyprowadzania wydruków kontrolnych w postaci tekstowej.

Prezentowany projekt bazowy ma charakter rozwojowy, student otrzymuje podstawowe procedury biblioteczne dla wybranych urządzeń sytemu z interfejsem SPI i ma za zadanie wzbogacić architekturę o kolejne urządzenia oraz bibliotekę o kolejne procedury.

2. CHARAKTERYSTYKA MAGISTRALI SPI

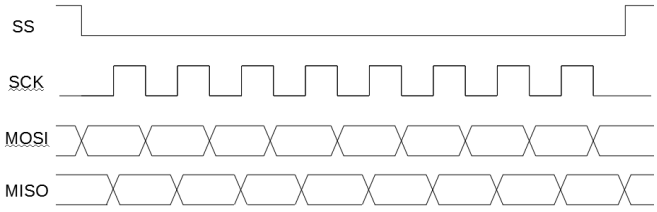
SPI jest magistralą dupleksową o szerokości czterech linii, na której może znajdować się jedno urządzenie Master i wiele urządzeń Slave. Funkcję urządzenia Master może pełnić wybrany mikrokontroler lub układ PLD (Programmable Logic Devices), natomiast pozostałe mikrokontrolery i PLD na magistrali muszą być urządzeniami Slave. Master generuje sygnały na trzech liniach:

- *SCK* – sygnał zegarowy synchronizujący transmisję,
- *MOSI* (Master Output Slave Input) – linia danych od Master do Slave,
- *SS* (Slave Select) – linia wyboru indywidualnego Slave'a.

Slave generuje sygnały na linii danych *MISO* (Master Input Slave Output). Każdy Slave ma niezależną linię *SS*, ale dzieli linie *MOSI*, *MISO* i *SCK* z pozostałymi Slave'ami. Urządzenie Slave ignoruje sygnały na liniach *MOSI* i *SCK* wówczas, gdy jego linia *SS* jest w stanie IDLE (wysokim). Magistrala SPI ma cztery tryby operacyjne będące kombinacją dwóch parametrów: polaryzacji zegara i fazowania zegara. Master jako urządzenie generujące zegar magistrali SPI ma możliwość komunikacji w każdym trybie, ale musi wybrać tryb akceptowany przez urządzenie Slave. Komunikacja odbywa w układzie Master-Slave. Master rozpoczyna transmisję

danych sygnalizując wybór Slave'a przez podanie stanu niskiego na jego linię *SS*.

Podczas każdego taktu zegara magistrali SPI zachodzi transmisja dupleksowa: Master nadaje jeden bit na linii *MOSI*, Slave odczytuje ten bit i w tym samym takcie zegara SPI, nadaje jeden bit po linii *MISO* odczytywany przez Master. Urządzenia Master i Slave działają w tej transmisji jak rejestry przesuwne – w ośmiu taktach zegara SPI bajt danych początkowo znajdujący się w rejestrze danych Master zostanie przesłany do Slave, a bajt z rejestru danych urządzenia Slave znajdzie się w Master. Dwie linie danych magistrali *MOSI* i *MISO* – każda dla jednego kierunku transmisji, tworzą magistralę dookólną, na której w każdym takcie zegara dochodzi do nadania jednego bitu i odebrania jednego bitu rysunek 1.



Rys. 1. Przebiegi na magistrali SPI między mikrokontrolerem a układem CPLD

Układ Slave nie ma możliwości inicjowania połączeń na magistrali SPI. W przypadku konieczności sygnalizowania stanów alarmowych inteligentny Slave, np. układ CPLD może wykorzystać linię przerwania zewnętrznego mikrokontrolera Master. Master mikrokontroler może wykorzystać sieć Ethernet do powiadomienia użytkownika przez wizualizację stanu alarmowego na stronie webowej użytkownika.

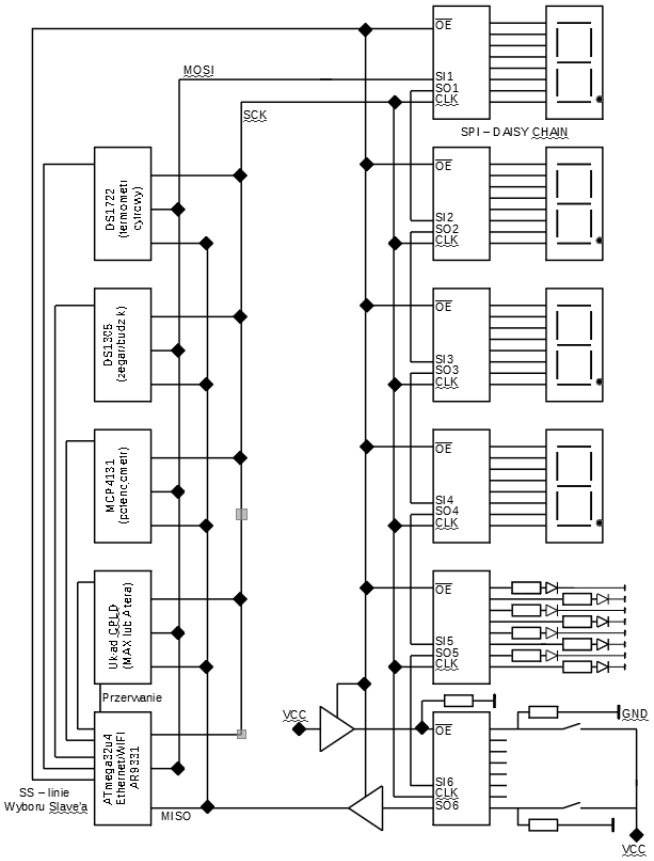
3. TOPOLOGIA LABORATORYJNEGO SYSTEMU CYFROWEGO

Do projektu wzorcowego wybrano mikrokontroler AT-Mega32u4 w układzie Arduino YUN z adapterem Ethernet Atheros9331 i układ CPLD z rodziny MAX7000S typu EPM128SLC84-15 firmy Altera (obecnie Intel), który został umieszczony na platformie uruchomieniowej zaprojektowanej i zbudowanej w Katedrze Automatyki Okrętowej (KAO) Uniwersytetu Morskiego w Gdyni [1].

Rolę urządzenia Master pełni mikrokontroler, ze względu na większą, niż w układach CPLD, liczbę dostępnych interfejsów i łatwiejszy kontakt ze światem zewnętrznym. Z bogatej oferty urządzeń z interfejsem SPI wybrano potencjometr cyfrowy MCP4131, pamięć EEPROM 25C040, termometr DS1722, wyświetlacze 7-segmentowe w układzie „daisy chain". Suwak potencjometru MCP4131 został połączony z wejściem przetwornika ADC wbudowanego w mikrokontroler umożliwiając zwrotny odczyt napięcia. W prezentowanym rozwiązaniu moduł Ethernet mikrokontrolera jest również podłączony do systemu po SPI. Topologię opracowanego systemu laboratoryjnego przedstawiono na rysunku 2.

W typowej architekturze SPI linie wyboru *SS* są prowadzone od Master do każdego Slave'a oddzielnie, co przy dużej liczbie urządzeń Slave wymaga wielu wyprowadzeń GPIO. Redukcję zapotrzebowania na liczbę wyprowadzeń mikrokontrolera można uzyskać łącząc Slave'y w szereg, tj. wyjście pierwszego z wejściem kolejnego, itd. Połączenie to zwane „daisy chain" wymaga tylko jednej wspólnej linii wyboru dla wszystkich urządzeń Slave.

W łańcuchu „daisy chain" dane są wprowadzane do kolejnych rejestrów szeregowo, a wyprowadzane szeregowo oraz równolegle. Do równoległych linii wyjściowych zostały podłączone wyświetlacze 7-segmentowe, diody oraz przełączniki (rys.2), natomiast wyjście szeregowe każdego rejestru w łańcuchu zostało połączone z wejściem kolejnego rejestru.



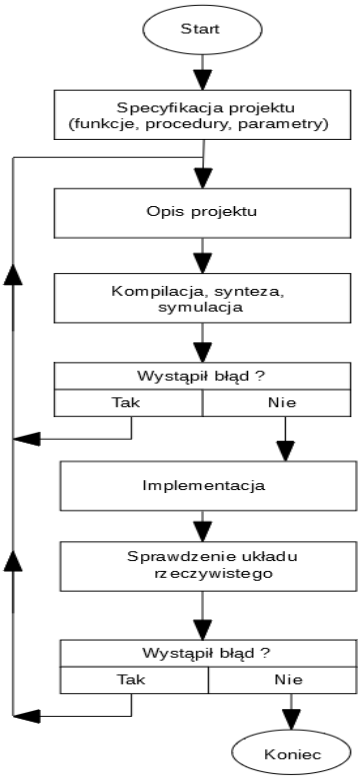
Rys. 2. Topologia laboratoryjnego systemu cyfrowego

4. PROJEKTOWANIE ALGORYTMÓW DLA UKŁADÓW PROGRAMOWALNYCH

Algorytm działania systemu musi uwzględnić transmisję danych między mikrokontrolerem a układem programowalnym CPLD, kod dla każdego z tych układów jest tworzony w innym języku i na innej platformie.

Projektowanie algorytmów sterowania cyfrowego z wykorzystaniem języka opisu sprzętu VHDL (Very High Speed Integrated Circuits Hadrware Description Language) i ich implementacja w strukturach układów programowalnych wymaga realizacji kilku etapów rozpoczynających się od sporządzenia specyfikacji czyli zdefiniowania niezbędnych funkcji, procedur, sygnałów wejściowych i wyjściowych [1, 2, 3, 4, 5]. Istotną rolę odgrywa także opis projektu, czyli zdefiniowanie wszystkich działań. Po przeprowadzeniu kompilacji możliwa jest implementacja algorytmu w strukturze układu programowalnego oraz sprawdzenie układu rzeczywistego. Wśród pakietów oprogramowania narzędziowego dostępnych obecnie na rynku najczęściej wykorzystywany jest pakiet Quartus (wcześniejsza wersja to Max Plus II Baseline) firmy Altera, pakiet Foundation ISE i WebPack IDE firmy Xilinx [3, 4, 5]. Pakiety te umożliwiają realizacje wszystkich etapów cyklu projektowego (rys. 3). Do napisania kodu transmisji danych po SPI dla omawianego systemu laboratoryjnego wykorzystano środowisko Quartus, które jest jednym z bardziej przyjaznych narzędzi CAD (Computer

Aided Design). W jego skład wchodzi edytor graficzny, edytor tekstowy HDL, kompilator, symulator funkcjonalny i czasowy, bogate biblioteki gotowych bloków, system definiowania stylów kompilacji projektu. Pakiet ten umożliwia projektowanie w obu standardowych językach HDL, tj. w języku VHDL oraz Verilog. Pakiet Quartus umożliwia obsługę układów CPLD i FPGA w jednym systemie, zapewnia elastyczną współpracę z innymi narzędziami EDA (Electronic Design Automation). Pakiet umożliwia wprowadzenie i edycję projektu, kompilację, określenie docelowego układu programowalnego, przyporządkowanie wyprowadzeń, symulację czasową i funkcjonalną oraz zaprogramowanie układu. Do budowy omawianego stanowiska laboratoryjnego wykorzystano edytor tekstowy języka opisu sprzętu VHDL.

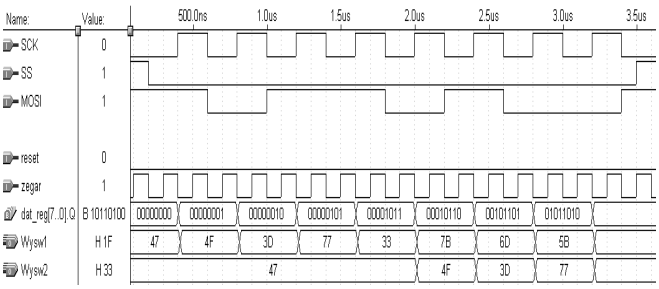


Rys. 3. Etapy projektowania w języku VHDL

Istotną właściwością opisu układów cyfrowych w języku VHDL jest współbieżność. Powoduje to, że podczas symulacji jakakolwiek zmiana sygnału w instrukcjach współbieżnych powoduje ich wykonanie w tej samej chwili czasu. Należy zaznaczyć, że współbieżność nie ma wpływa na sposób kompilacji programu napisanego w języku VHDL, która jest realizowana sekwencyjnie. W środowisku Quartus istnieje także możliwość wykonania instrukcji sekwencyjnych, które definiujemy w tzw. procesie.

Algorytm komunikacji między mikrokontrolerem a układem CPLD po magistrali SPI został zrealizowany w dwóch procesach współbieżnych [6]. Proces odpowiedzialny za transmisję danych jest wykonywany w stanie niskim na linii wyboru *SS*, natomiast proces odpowiedzialny za wyświetlenie odebranego bajtu na diodach, wyświetlaczu 7-mio segmentowym lub LCD (Liquid Crystal Display) jest wykonywany w stanie wysokim na linii *SS*. Układ CPLD jest sterowany dodatkowym zegarem zewnętrznym, ponieważ dla magistrali SPI przebieg taktujący jest generowany przez urządzenie Master, czyli w tym przypadku mikrokontroler. Magistrala w tej komunikacji pracuje w trybie zero, tzn. mikrokontroler zmienia stany na linii *MOSI* na zboczu opadającym zegara, a układ CPLD po rozpoznaniu zbocza narastającego odczytuje stany kolejnych impulsów składających się na przesyłany bajt.

Zbudowana w KAO platforma z układem CPLD jest wyposażona w wyświetlacz LCD oraz wyświetlacze 7-mio segmentowe. Przygotowane, w ramach omawianego stanowiska laboratoryjnego, oprogramowanie umożliwia studentom zapoznanie się z zasadami obsługi tych elementów w języku VHDL. W przygotowanym projekcie na LCD jest wyświetlana nazwa uczelni, natomiast na wyświetlaczach 7-segmentowych (Wysw1, Wysw2) pojawia się w zapisie szesnastkowym kod przesyłanego znaku. Przykładowe przebiegi uzyskane w symulacji czasowej zostały przedstawione na rysunku 4.



Rys. 4. Przykładowe przebiegi dla algorytmu opracowanego w języku VHDL

5. KONTROLA CYFROWEGO SYSTEMU LABORATORYJNEGO PRZEZ SIEĆ ETHERNET

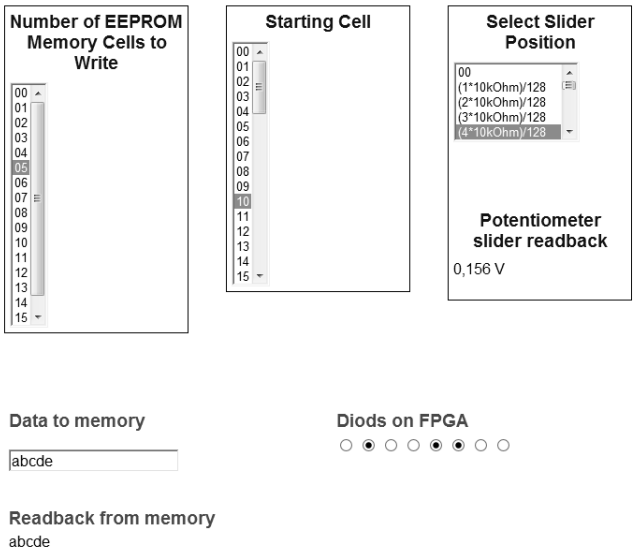
Warunkiem zdalnego dostępu do systemu jest uruchomienie na mikrokontrolerze aplikacji węzła sieci Ethernet. Zgodnie z koncepcją prezentowanego projektu na mikrokontrolerze uruchomiono aplikację serwera tcp/ip obsługującego żądania http. Wówczas przez przeglądarkę i utworzoną w języku HTML stronę internetową, pełniącą rolę klienta sieci, można kontrolować urządzenia na liniach GPIO (General Purpose Input-Output) mikrokontrolera wysyłając żądania http do serwera i odbierając odpowiedzi zawierające stany urządzeń [7, 8, 9].

Podstawowa wersja strony internetowej utworzonej w HTML-u powinna zostać zapisana na karcie mikro SD podłączonej do serwera po SPI ze względu na ograniczoną pojemność pamięci mikrokontrolera. Stany urządzeń mogą być przesyłane dynamicznie z wykorzystaniem technologii AJAX (Asynchronous JavaScript and XML) i wizualizowane na stronie webowej przy użyciu technologii DOM HTML (Document Object Model).

Na rysunku 5 przedstawiono przykład strony HTML do kontroli i sterowania urządzeniami na magistrali SPI z komponentami odpowiadającymi topologii z rysunku 2. Zapytania http do serwera mogą być wysyłane przez przeglądarkę cyklicznie lub na życzenie. W żądaniu http wysyłane są rozkazy dla urządzeń na magistrali SPI, w tym przypadku wykorzystano metodę http GET. Po wykonaniu rozkazów serwer przesyła potwierdzenia w formacie rozpoznawanym przez przeglądarkę dołączając stany urządzeń. W przykładowej aplikacji użytkownik wybiera na stronie internetowej: numer pierwszej komórki i liczbę komórek pamięci EEPROM do zapisu, podaje znaki ASCII do zapisania (rys.5), w odpowiedzi otrzymuje zwrotny odczyt z pamięci potwierdzający udany zapis. W celu zaprogramowania potencjometru cyfrowego użytkownik wprowadza żądaną wartość rezystancji suwak-masa z uwzględnieniem rozdziel-

czości i rezystancji nominalnej potencjometru. W odpowiedzi otrzymuje się napięcie z suwaka części analogowej potencjometru, odczytane przez przetwornik ADC (Analog-Digital Converter) wbudowany w mikrokontroler. Żądanie http może też dotyczyć odczytu temperatury z cyfrowego termometru lub sterowania diodami na liniach układu CPLD. Odpowiedź może zawierać informację o stanie alarmowym odebraną od układu CPLD po linii przerwania zewnętrznego mikrokontrolera dokonanego z pominięciem magistrali SPI.

Projektowanie strony internetowej użytkownika w języku HTML jest dobrym punktem wyjścia do nauki technik asynchronicznych AJAX służących do podmiany fragmentów strony webowej wizualizujących stany urządzeń.



Rys. 5. Aplikacja użytkownika – strona internetowa

6. WNIOSKI KOŃCOWE

W artykule zaprezentowano pomysł na rozbudowany projekt z dziedziny układów cyfrowych łączący układy różnego typu i przeznaczenia (mikrokontrolery i PLD) z czujnikami i innymi układami cyfrowymi. W roli głównej magistrali systemu cyfrowego pracuje magistrala SPI. Alternatywną propozycją może być wykorzystanie w tej roli magistrali I2C/TWI (Two Wire Interface).

Projekt powinien być proponowany studentom starszych roczników ze względu na wymaganą do jego realizacji znajomość języków VHDL, HTML, C i wykorzystanie kilku platform uruchomieniowych. Wykonanie projektu daje studentom satysfakcję z potwierdzenia praktycznych umiejętności inżynierskich w dziedzinie współczesnych układów cyfrowych i internetu rzeczy.

7. BIBLIOGRAFIA

1. Noga K. M., Radwański M. - Modern improvements in the digital logic laboratory, Technological Developments in Networking, Education and Automation, USA, Springer 2010, pp 109 - 114, ISBN 978-90-481-9150-5 (HB), ISSN 1425 – 5766.
2. Noga K. M., Radwański M: Projektowanie układów programowalnych w środowisku Quartus II z wykorzystaniem edytora tekstowego, instrukcja laboratoryjna, Akademia Morska, 2008.
3. Kalisz J.: Język VHDL w praktyce, Wydawnictwo Komunikacji i Łączności, Warszawa, 2002.
4. Zwoliński M.: Projektowanie układów cyfrowych z wykorzystaniem języka VHDL, WKiŁ, Warszawa, 2002.
5. Kalisz J.: Kurs Języka VHD, Wojskowa Akademia Techniczna, Warszawa, 2008.
6. Implementation of SPI Protocol in FPGA, Patil V., Dahake V., Verma D., Pinto E.: International Journal Of Computational Engineering Research (online), ISSN 2250-3005, January 2013.
7. Callaghan MJ., Harkin J., McGinnity TM., Maguire LP.: Client-Server Architecture for Remote Experimentation for Embedded Systems, iJOE International Journal of Online Engineering, 2006.
8. Antinic D.: Ethernet Communication in Microcontroller Systems, ReaserchGate Technical Report 2017, DOI: 10.13140/RG.2.2.27380.35206.
9. Pałczyńska B., Rabczuk D.: Low-Cost Embedded Control System for Environmental Monitoring, IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe) 2018, DOI:10.1109/eeeic.2018.8493727.

DIGITAL LABORATORY SYSTEM PROGRAMMED OVER THE ETHERNET BASED ON SPI BUS

The article presents a laboratory system based on SPI bus with a microcontroller as the Master device and various Slave devices connected to the Master over the SPI bus. The variety of digital devices with SPI interface enables the growth of the project. In the exemplary system several digital devices were used: a thermometer, an EEPROM memory, a potentiometer and a PLD structure. The PLD structure is chosen as Slave device on the bus to take advantage of its typical functionality: ability to perform fast arithmetical calculations. In order to inform of an alarm state the CPLD structure must use an external interrupt line to the microcontroller because Slave device cannot start a communication session over SPI bus. The microcontroller is chosen as Master device because it possesses various external interfaces especially Ethernet interface. The aim of the project is creating a remotely controlled system programmed over the Ethernet which can expand through connecting additional devices on the SPI bus and writing libraries for them. The Ethernet interface is used to load the microcontroller program over the Ethernet which gives the student the ability to remotely load and test the microcontroller software.

Keywords: digital systems, Serial Peripheral Interface (SPI), embedded devices, remote control, programmable devices.