

doi:10.15199/48.2021.02.22

## Protokół komunikacyjny Modbus RTU z wykorzystaniem sterownika PLC i układu FPGA – analiza porównawcza

**Streszczenie.** W artykule przedstawiono analizę porównawczą stanowiska badawczego złożonego z układu FPGA oraz sterownika z rodziny S7-1200, które współpracują ze sobą wymieniając informacje zgodnie z protokołem komunikacyjnym Modbus RTU. W obu stanowiskach obiektem sterowania jest silnik trójfazowy sterowany przy wykorzystaniu przemiennika częstotliwości SINAMICS V20.

**Abstract.** This paper shows a comparison analysis of two lab stations designed to control a three phase electric motor. In both stations the motor is driven by a SINAMICS V20 frequency controller which communicates using Modbus RTU protocol with its control system. The control system in the lab station it is an FPGA circuit while in the other lab station it is a S7-1200 series programmable logic controller. (**Communication protocol using a PLC and the FPGA system - comparative analysis**).

**Słowa kluczowe:** układy FPGA, platforma uruchomieniowa DE2-115, sterownik PLC S7-1200, protokół komunikacyjny Modbus RTU.

**Keywords:** FPGA programmable circuits, DE2-115 run platform, programmable logical control S7-1200, Modbus RTU communication protocol.

### Wstęp

Układy cyfrowe są stosowane w wielu dziedzinach życia oraz gospodarki. Układy programowalne PLD (ang. Programmable Logic Devices) serii CPLD (ang. Complex Programmable Logic Devices) i FPGA (ang. Field Programmable Gate Array), których działanie określa zaimplementowane oprogramowanie, umożliwiają pracę jako dowolny układ cyfrowy. Za ich pomocą możliwe jest zastąpienie nawet kilkudziesięciu układów logicznych wykonanych w technologii CMOS lub „dinozaurów” TTL. Układy FPGA są bardziej złożone, dzięki czemu można stworzyć własne procesory i bardzo zaawansowane układy. W systemach automatyki podstawowym elementem wykonawczym są sterowniki PLC (ang. Programmable Logic Controller). Dzięki swojej uniwersalności są one wykorzystywane do obsługi małych, średnich i dużych systemów wymagających niezawodnego i szybkiego sterowania. Sterowniki obsługują także układy wejść /wyjść, współpracują z panelami operatorskimi oraz odpowiadają za komunikację z urządzeniami wykonawczymi. W artykule przedstawiono analizę porównawczą stanowiska badawczego złożonego z układu FPGA oraz sterownika z rodziny S7-1200, które współpracują ze sobą wymieniając informacje zgodnie z protokołem komunikacyjnym Modbus RTU. W obu stanowiskach obiektem sterowania jest silnik trójfazowy sterowany przy wykorzystaniu przemiennika częstotliwości SINAMICS V20.

### Charakterystyka protokołu komunikacyjnego MODBUS RTU

Protokół komunikacyjny Modbus został stworzony przez firmę Modicon w 1979 roku (obecnie firma Schneider Electric) [1]. Pomimo, że jest to jeden z najstarszych protokołów komunikacyjnych dedykowanych do zastosowań w automatyce, wciąż jest popularny i szeroko wykorzystywany. Zaletami sprawiającymi, że protokół ten nie wyszedł z użytku, jest jego prostota, uniwersalność oraz łatwość w implementacji, jak i eksploatacji. Sieć przemysłową opartą na protokole Modbus używa się przy m.in. zdalnej akwizycji danych, kontroli procesów przemysłowych, systemach nadzoru i ochrony (w tym również monitoringu), zarządzania energią oraz w laboratoriach automatyki. Wykorzystywane media transmisyjne takie jak skrętka, standard RS232, RS485 czy światłowód umożliwiają wprowadzenie standardu Modbus zarówno w nowych jak i starych aplikacjach. Istnieją trzy główne standardy protokołu komunikacyjnego Modbus: ASCII, TCP/IP oraz RTU. W artykule do analizy

wykorzystano protokół Modbus RTU. Informacje zakodowane są binarnie, na jeden bajt informacji przypada jeden bajt komunikatu. Rozwiązanie to związane jest z transmisją szeregową RS232 lub RS485. Możliwe szybkości transmisji wynoszą od 1200 bit/s do 115 kbit/s, jednak najczęściej stosowanymi są szybkości 9600 bit/s oraz 19200 bit/s [2].

Protokół Modbus jest protokołem otwartym, a komunikacja odbywa się na zasadzie Master - Slave (Query - Response). W skład urządzeń Master wchodzi komputer lub sterownik programowalny PLC z modulem komunikacyjnym przeznaczonym do pracy w standardzie RS. Do urządzeń Slave należy drugi sterownik PLC lub podrzędne moduły wykonawcze. Przy czym urządzenie nadrzędne (Master) rozpoczyna transmisję danych, wysyła polecenia do pozostałych urządzeń w sieci i nadzoruje ich pracę. Natomiast urządzenia podrzędne (Slaves) sterują procesami automatyki oraz archiwizują dane z urządzeń peryferyjnych, które wysyłają odpowiedź do jednostki nadrzędnej. W tabeli 1 przedstawiono ramkę protokołu Modbus RTU. W omawianym protokole nie ma możliwości zrealizowania bezpośredniej komunikacji pomiędzy Slave'ami, czyli jednocześnie nie może odpowiadać kilka urządzeń podrzędnych (Slaves).

### Charakterystyka sterowników PLC

Programowalny sterownik logiczny PLC to urządzenie stosowane w systemach automatyki. Przeznaczone jest do sterowania pracą maszyny lub urządzenia technologicznego. W sterowniku PLC znajduje się jednostka centralna, która jest układem wieloprocesorowym. O szybkości działania sterownika decyduje typ oraz ilość mikroprocesorów. Pamięć w sterowniku jest wykorzystywana do realizacji programu wykonawczego oraz danych pośrednich. Program przechowywany jest w pamięci typu RAM, EPROM lub EEPROM i jest wykonywany cyklicznie od pierwszej do ostatniej instrukcji. Przed rozpoczęciem pracy programu następuje odczyt stanów wejść oraz ich zapamiętanie, natomiast po każdym z cykli ustawione są wartości wyjść. Jeżeli zadanie projektowe wymaga, aby reakcja na zmianę stanu sygnału wejściowego była możliwie najszybsza, stosuje się przerwania [3]. Ponadto sterowniki, które dzielimy ze względu na budowę na modułowe i kompaktowe, można zastosować zarówno w indywidualnych rozwiązaniach jak i w dużych halach produkcyjnych.

Tabela 1. Ramka protokołu komunikacyjnego Modbus RTU

Znacznik początku	Adres SLAVE	Funkcja	Dane	Kontrola CRC	Znacznik końca
odstęp 3,5 znaku	8 bitów	8 bitów	n x 8 bitów (0 - 252 bajtów)	16 bitów	przerwa 3,5 znaku

### Zasady projektowania algorytmów z wykorzystaniem języka VHDL i układów programowalnych

Projektowanie algorytmów sterowania z wykorzystaniem języka opisu sprzętu VHDL ((ang. Very High Speed Integrated Circuits Hardware Description Language) oraz ich implementacja w strukturach układów programowalnych wymaga realizacji kilku etapów. Na etapie początkowym niezbędne jest sporządzenie specyfikacji czyli zdefiniowanie niezbędnych funkcji, procedur, sygnałów wejściowych i wyjściowych. Istotną rolę odgrywa także opis projektu, czyli zdefiniowanie wszystkich działań. Po przeprowadzeniu kompilacji, po uzyskaniu prawidłowych przebiegów czasowych możliwa jest implementacja algorytmu w strukturze układu programowalnego oraz sprawdzenie układu rzeczywistego. Wśród pakietów oprogramowania narzędziowego dostępnych na rynku najczęściej wykorzystywany jest pakiet Quartus firmy Altera (obecnie Intel), pakiet Foundation ISE, WebPack IDE lub Vivado Design Suite firmy Xilinx [4, 5, 6]. Pakiety te umożliwiają realizację wszystkich etapów cyklu projektowego.

Środowisko Quartus, posiada bardzo intuicyjny interfejs, jego możliwości w pełni wystarczają do pełnego opisu sprzętu. Obecnie dostępne są trzy wersje środowiska Quartus Prime, wykorzystywane w zależności od zaawansowania układów i projektów [4]:

- Pro Edition – dostosowane dla najbardziej wymagających projektów i zaawansowanych układów,
- Standard Edition – zoptymalizowane pod kątem starszych układów programowalnych oraz dodatkowo dla układów firmy Intel z rodziny Cyclone 10,
- Lite Edition – przystosowane do podstawowych projektów i średnio zaawansowanych układów, wersja jest bezpłatna.

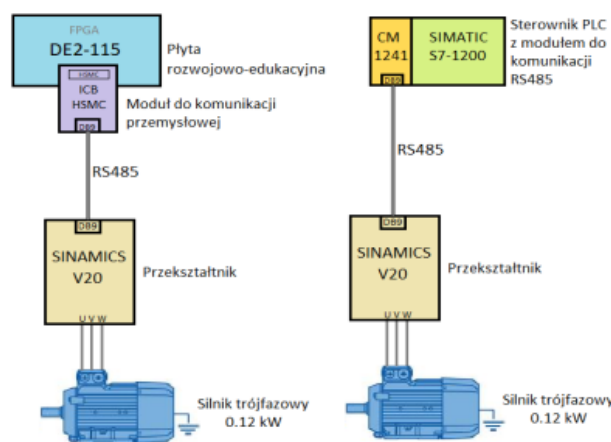
W środowisku Quartus można projektować algorytmy sterowania za pomocą edytora graficznego (wykorzystując bramki, przerzutniki, liczniki, multiplexerów, koderów, kodery, demultiplexery w postaci graficznej) lub edytora tekstowego z wykorzystaniem języków opisu sprzętu, np. VHDL, Verilog czy AHDL (ang. Altera Hardware Description Language), który został opracowany tylko na potrzeby środowiska Quartus i układów Intel/Altera.

Natomiast firma Xilinx do projektowania udostępnia środowisko ISE (ang. Integrated Synthesis Environment), jak w większości środowisk jest możliwość wykorzystania edytora graficznego oraz tekstowego. Firma ta oferuje dwie opcje środowiska programistycznego. Jedną z nich jest mniej rozbudowana wersja ISE WebPack, która jest wersją darmową (public domain). Obejmuje ona wybrane układy FPGA i wszystkie CPLD. Płatne wersje ISE Design Suite są natomiast pełną wersją środowiska i współpracują ze wszystkimi układami, tj. układami z rodziny Spartan 6, Virtex 6, FPGA oraz CPLD. Ponadto do implementacji najnowszych układów z serii 7 FPGA rekomendowane jest środowisko Vivado Design Suite HL. Środowisko to również jest dostępne w wersji darmowej jako WebPack, natomiast dwie kolejne rozbudowane wersje Vivado HL Design Edition i Vivado HL System Edition są płatne.

Istotną właściwością opisu układów cyfrowych w języku VHDL jest także 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 wpływu 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.

Do opracowania kodu transmisji dla omawianego stanowiska badawczego i laboratoryjnego wykorzystano środowisko Quartus, które jest jednym z bardziej przyjaznych narzędzi CAD (ang. 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 (ang. Electronic Design Automation). Ponadto umożliwia on wprowadzenie i edycję projektu, kompilację, określenie docelowego układu programowalnego, przyporządkowanie wyprowadzeń, symulację czasową i funkcjonalną oraz zaprogramowanie układu. Do budowy omawianego stanowiska wykorzystano edytor tekstowy języka opisu sprzętu VHDL.



Rys. 1. Stanowiska badawcze do analizy protokołu komunikacyjnego Modbus RTU, po lewej z zastosowaniem układu FPGA, po prawej z sterownikiem S7-1214 DC/DC/DC

### Topologia stanowiska badawczego

W artykule przedstawiono dwa układy pracujące w standardzie Modbus RTU (rys. 1). W pierwszym stanowisku nadrzędnym urządzeniem (Master) jest układ FPGA, a w drugim stanowisku funkcję Mastera pełni sterownik S7-1214 DC/DC/DC. W obu stanowiskach obiektem sterowania jest silnik trójfazowyysterowany przy wykorzystaniu przemiennika częstotliwości SINAMICS V20. Parametry protokołu komunikacyjnego Modbus RTU są następujące:

- szybkość transmisji (ang. baud rate) – 9,6 kbit/s,
- kontrola parzystości (ang. parity) – przyjmuje wartość 1, gdy liczba danych w pakiecie jest parzysta,
- bity danych (ang. data bits) – 8 znaków bitowych,
- bit stopu (ang. bit stop) – wartość 1,

- czas oczekiwania na odpowiedź (ang. wait times) – 1 ms, Prezentowane stanowiska (rys. 1) realizują następujące sterowanie silnikiem trójfazowym:
- praca silnika: start/stop,
- zmiana kierunku obrotów: prawo / lewo,
- zmiana częstotliwości na wyjściu przetwornicy: 20/50 Hz.

### Stanowisko badawcze z układem FPGA

Stanowisko badawcze do komunikacji w sieci Modbus RTU zostało stworzone w oparciu o platformę uruchomieniową DE2-115 z dołączonym modułem do komunikacji przemysłowej ICB-HSMC. Platformy te są dostępne w laboratorium Techniki Cyfrowej w Katedrze Automatyki Okrętowej Uniwersytetu Morskiego w Gdyni, zostały także udostępnione przez firmę 4Self z Gdyni, działającej w branży automatyki przemysłowej [7].

Platforma DE2-115 wyposażona jest w procesor EP4CE115 z rodziny Cyclone IV firmy Altera, 2 MB pamięci SRAM, 128 MB SDRAM, 8 MB pamięci Flash oraz czytnik kart SD. Do komunikacji zewnętrznej płytkę posiada port USB oraz port Ethernet, dostępne są również porty do nadawania formatów audio oraz wideo. Zestaw uruchomieniowy posiada również gniazdo 40 pinów wejścia/wyjścia. Do wprowadzania sygnałów wejściowych służą 4 styki membranowe oraz 18 przełączników dwustanowych, a do sygnalizowania wyjść 27 diod LED, 3 wyświetlacze 7 segmentowe oraz wyświetlacz LCD. Do zestawu dołączone jest również specjalne oprogramowanie, tzw. panel sterowania oraz środowisko programistyczne Quartus II. Panel sterowania umożliwia zarządzanie poszczególnymi elementami znajdującymi się na płycie poprzez komputer. Natomiast program Quartus II pozwala na obsługę układów programowalnych. Platforma DE2-115 umożliwia zaimplementowanie skomplikowanych kodów

Inny przykład wykorzystania platformy DE2-115 zaprezentowano w pracy [8], w której omówiono cyfrowy system badawczy i laboratoryjny oparty na magistrali SPI (ang. Serial Peripheral Interface) z mikrokontrolerem jako urządzeniem Master oraz różnymi urządzeniami Slave podłączonymi do magistrali, (np. cyfrowy termometr, potencjometr, pamięć EEPROM, układ programowalny CPLD lub FPGA). System ten stanowi bazę sprzętowo-programistyczną dla rozwojowego systemu sterowanego zdalnie oraz programowanego przez Ethernet. Jest on przystosowany do zdalnego ładowania nowych wersji programu do pamięci mikrokontrolera po Ethernetie. Użytkownik przez stronę internetową ma możliwość wysłania żądania tcp/http dla urządzeń na magistrali SPI, a w odpowiedzi http otrzymuje informację o stanie urządzeń. Ponadto w pracy [11] przedstawiono analizę porównawczą skuteczności działania środowiska Quartus II, jako przemysłowego narzędzia do syntezy programowalnych układów cyfrowych, oraz środowiska Synthagate, które powstało w ośrodku akademickim.

W zbudowanym i omawianym w artykule stanowisku badawczym (rys. 3) komunikacja została oparta o standard transmisji szeregowej RS485 co pozwala na rozwinięcie sieci o 32 moduły pracujące w trybie Slave. Z modułu ICB-HSMC do połączenia z przemiennikiem zostało wykorzystane gniazdo DB9. Układem pośredniczącym pomiędzy układem FPGA, a gniazdem DB9 jest układ ADM2486, który jest układem typu transceiver, czyli służącym do nadawania oraz odbierania danych w trybie half-duplex. Oba składniki mają wspólne obwody i umieszczone są w jednej obudowie. Układ jest oparty o wzmacniacze operacyjne, posiada również izolację galwaniczną.

Część, przygotowanego i przetestowanego na potrzeby prac badawczych, programu odpowiada za komunikację

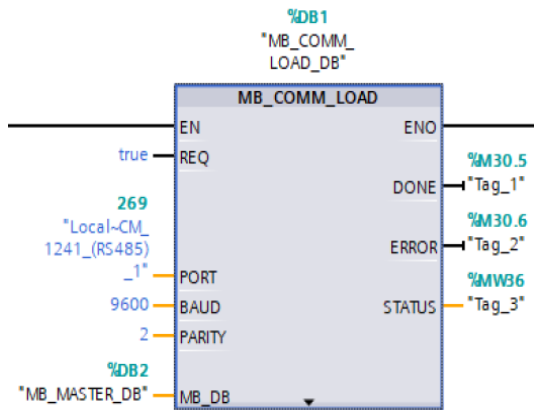
zgodną z protokołem Modbus RTU. Opracowany kod programu umożliwia łatwą zmianę parametrów komunikacji, takich jak prędkość transmisji, bit parzystości oraz bit startu/stopu. Algorytm wysyła oraz odbiera wiadomości w formie ramek, na które składa się adres, funkcja, dane oraz kontrola CRC (tabela 1).

W momencie uruchomienia układu następuje wczytanie parametrów komunikacji oraz załadowanie wartości początkowych i stałych, potrzebnych do działania algorytmu. Układ przechodzi w stan bezczynny do momentu, w którym program wymusi nadawanie wiadomości. Po nadaniu całej zawartości ramki układ oczekuje na odpowiedź przez 1 ms. Gdy układ odbierze bit startu, następuje odbiór reszty wiadomości. Po odebraniu całej ramki następuje jej przetworzenie. Sprawdzane jest, czy odebrana wiadomość jest poprawna, sprawdzany jest bit parzystości oraz kod CRC. Kolejną część programu odpowiada za generowanie zawartości ramek oraz przetwarzanie odebranych wiadomości. Generowane są dwa typy wiadomości: wpisujące wartość do rejestru oraz odczytujące wartość z rejestru. Algorytm sprawdza stan przełączników na płycie DE2-115 i zgodnie z założeniami generuje zawartość ramek. Wiadomości są nadawane cyklicznie, przy czym są one odpowiednio zakolejkowane. Pozostała część programu odpowiada za sterowanie wyświetlaczami 7-segmentowymi. Wyświetlane są na nich informacje dotyczące nastaw przesyłanych do przemiennika oraz odbieranych wartości związanych z prędkością pracy silnika.

### Stanowisko badawcze ze sterownikiem S7-1214

Modbus RTU został zaimplementowany w sterowniku SIMATIC S7-1212 DC/DC/DC, z wykorzystaniem modułu do komunikacji w protokole Modbus CM 1241-5 [9]. W środowisku TIA Portal dokonano konfiguracji urządzeń. Zostały nadane odpowiednie wartości w module CM 1241 zgodne z nastawami w przemienniku tak samo jak podano dla układu FPGA. Konfiguracja sterownika i modułu komunikacyjnego związana jest z dodaniem kil-ku bloków funkcyjnych takich jak: MB\_COMM\_LOAD oraz MB\_MASTER. Blok MB\_COMM\_LOAD (rys. 2) znajduje się w bloku organizacyjnym OB100 (ang. Organization Block) i jego instrukcja wykonywana jest jednokrotnie przy włączeniu sterownika. Do wejść bloku wczytywany jest numer portu modułu CM 1241, szybkość transmisji, parzystość (z instrukcji 2 = even) oraz wskazywany jest blok danych DB (ang. Data Block), w którym zostaną zawarte parametry do komunikacji.

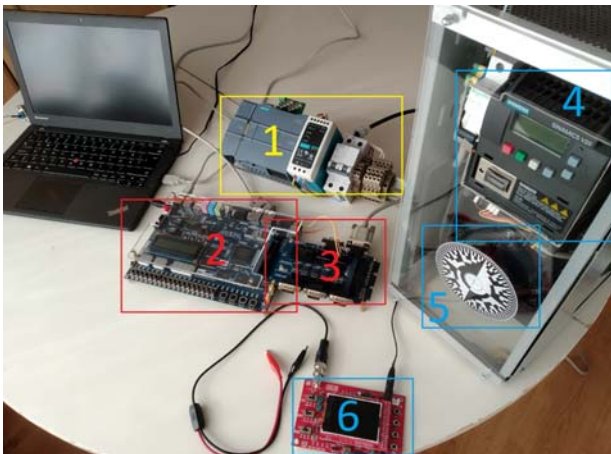
Natomiast instrukcja MB\_MASTER znajduje się w bloku organizacyjnym OB1 i jego stan jest sprawdzany w każdym cyklu wykonywania programu. Wejście REQ (ang. Request) reaguje na zbocze rosnące zegara i odpowiada za wysyłanie zapytań. Zostało wykorzystane wejście zegarowe o częstotliwości 2,5 Hz, co oznacza, że zapytanie (ramka) jest wysyłane co 400 ms. Program oparty o wbudowane biblioteki pozwala uprościć algorytm sterowania. Do komunikacji został wykorzystany jeden blok FBV20\_Modbus, który został umieszczony w bloku organizacyjnym OB1. Zaletą tego rozwiązania jest automatyczne kolejkowanie ramek oraz łatwy dostęp do cyklicznej i acyklicznej komunikacji. Do sterowania podstawowymi parametrami takimi jak start/stop i prawo/lewo wykorzystuje się wejście STW (niem. Steuerwort - słowo kontrolne), natomiast do sterowania prędkością stosuje się wejście HSW (niem. Haupsollwert - wartość zadana prędkości).



Rys. 2. Blok funkcyjny MB\_COMM\_LOAD

### Wyniki badań

Badania zostały wykonane na stanowisku badawczym przedstawionym na rysunku 3. Numerem 4 oznaczony został przemiennik częstotliwości SINAMICS V20, numerem 1 - sterownik z rodziny S7-1214 DC/DC/DC, numerem 2 i 3 odpowiednio układ FPGA oraz moduł ICB-HSMC. Oscyloskop, oznaczony jako 6 został wykorzystany do obserwacji za-wartości ramki transmisji Modbus RTU.



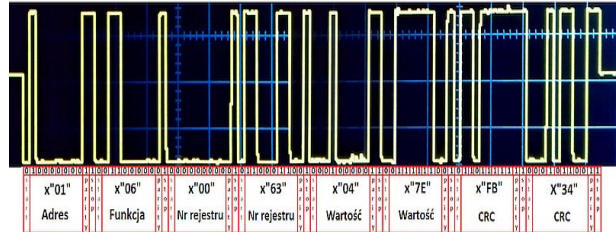
Rys. 3. Stanowiska badawcze. Kolorem żółtym oznaczono stanowisko badawcze nr 2, kolorem czerwonym stanowisko badawcze nr 1, kolorem niebieskim część wspólną obu stanowisk. 1-sterownik S7-1200, 2-płyta DE2-115, 3-moduł ICB-HSMC, 4-przemiennik częstotliwości SINAMICS V20, 5-silnik 3f0.12kW, 6-oscyloskop

Analiza porównawcza zbudowanego stanowiska badawczego dotyczyła następujących czynników:

- stopnia złożoności konfiguracji zestawu badawczego,
- realizacji algorytmów sterowania,
- możliwości dokonywania zmian w istniejącej sieci,
- sterowania,
- monitorowania sieci,
- zastosowania.

Dodatkowo z pomocą oscyloskopu sprawdzono, czy występują różnice w warstwie fizycznej sieci, takie jak napięcie, przerwy pomiędzy ramkami, zakłócenia itp. Badano także, czy poprzez konwerter RS485 - USB i dedykowaną aplikację występują zakłócenia oraz przekłamanie podczas transmisji. Na rysunku 4 została przedstawiona przykładowa ramka nadawana z platformy DE2-115 do przemiennika SINAMICS 20. W ramce jest zawarty adres przemiennika, funkcja nadpisania wartości rejestru, numer rejestru oraz wartość, która ma zostać wpisana. Na końcu ramki jest umieszczony kod CRC

wyliczony określonym algorytmem w programie. Rejestr STW odpowiada za uruchomienie silnika oraz kierunek obrotów, w tym przypadku przesłana wiadomość informuje o uruchomieniu silnika w prawo. Fragment kodu VHDL przedstawiony na rysunku 5 odpowiada za zawartość ramki. Natomiast na rysunku 6 został przedstawiony równorzędny fragment programu, opracowany w środowisku TIA Portal, dla sterownika S7-1214 DC/DC/DC. Według opinii autorów artykułu czytelniejszy jest kod napisany w języku programowania sprzętu VHDL.



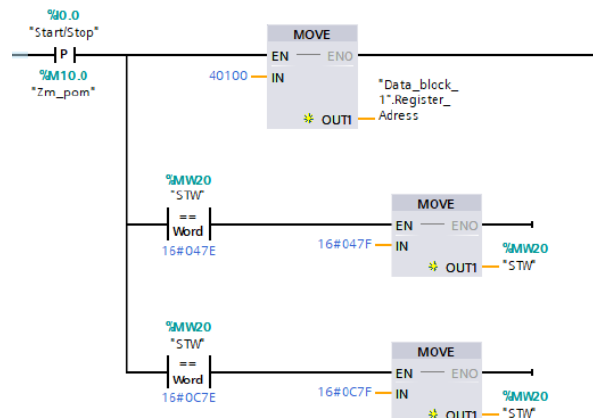
Rys. 4. Ramka protokołu komunikacyjnego Modbus RTU

```

if(rising_edge(CLOCK_50) and SW(0) = '0' and SW(1) = '0' and frame_number = 0) then
    frame_size <= 6;
    TR_slave_ID <= to_stdlogicvector(x"01");
    TR_Function_code <= to_stdlogicvector(x"06");
    TR_Special_data1 <= to_stdlogicvector(x"00"); -- adres 9CA4=40100
    TR_Special_data2 <= to_stdlogicvector(x"63"); -- adres x63= 99
    TR_Special_data3 <= to_stdlogicvector(x"04"); -- wartosc
    TR_Special_data4 <= to_stdlogicvector(x"7E"); -- wartosc
    LEDG(8) <= '0';
    HEX5 <= prawo;

```

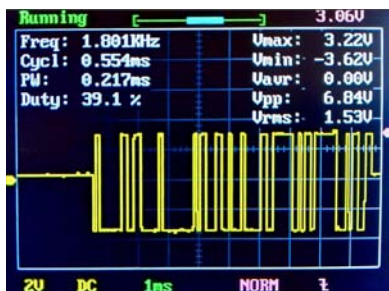
Rys. 5. Fragment programu VHDL odpowiadający za zawartość ramki



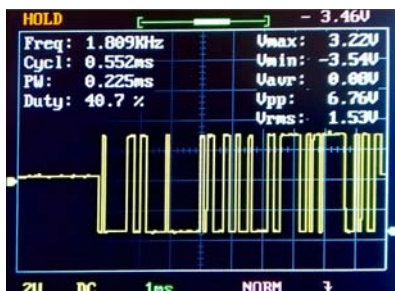
Rys. 6. Fragment programu LAD, który definiuje zawartość ramki, zapisany w środowisku TIA Portal

Na rysunkach 7 i 8 zostały przedstawione wyniki pomiarów napięcia dla platformy DE2-115. Istotny jest zakres napięć dla '0' logicznego oraz dla „1”. Z pomiarów wynika, że informacja będzie przekazywana nawet przy pewnych odkształceniach sygnału. Dla logicznej jedynki różnica potencjałów na przewodach wyniosła 3,06 V, natomiast dla zera – 3,46 V.

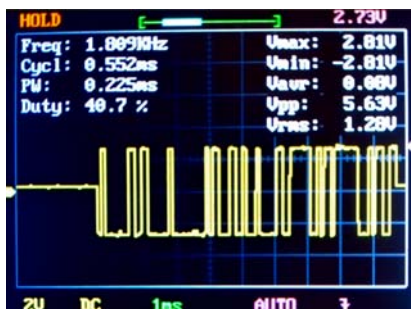
Dodatkowo na rysunkach 9 i 10 przedstawiono wyniki pomiarów napięcia pomiędzy przewodami AB dla sterownika S7-1200. Dla logicznej jedynki różnica potencjałów na przewodach wyniosła 2,73 V, natomiast dla zera – 2,65 V.



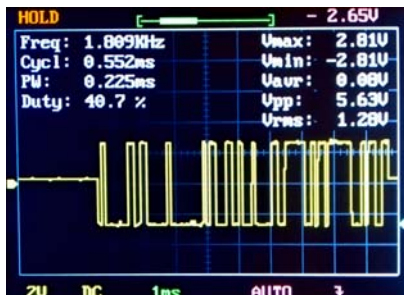
Rys. 7. Pomiar napięcia dla jedynek logicznej dla DE2-115



Rys. 8. Pomiar napięcia dla zera logicznego dla DE2-115



Rys. 9. Pomiar napięcia dla jedynek logicznej dla S7-1214C



Rys. 10. Pomiar napięcia dla zera logicznego dla S7-1214C

Z przeprowadzonych pomiarów wynika, że napięcia dla platformy uruchomieniowej DE2-115 są wyższe. Dla logicznej jedynek o 0,33 V, natomiast dla logicznego zera o 0,81 V. Według standardu RS485 nadawca powinien mieć wyjście różnicowe na poziomie napięcia minimum 1,5 V, podczas gdy odbiorca powinien odbierać sygnały różnicowe o wartości przynajmniej 200 mV. Dla obu rozwiązań spełnione są więc powyższe wymagania.

W tabeli 2 przedstawiono podstawowe wnioski dotyczące oceny działania, konfiguracji i uruchomienia omawianego stanowiska badawczego.

Tabela 2. Podsumowanie analizy dla obu stanowisk badawczych.

	S7-1214C	DE2-115
Trudność w konfiguracji, algorytm sterowania	Średnia trudność. Wymagana znajomość języka LAD oraz podstawowa wiedza z zakresu protokołu Modbus RTU. Konfiguracja może sprawić trudności w tworzeniu średnio zaawansowanych algorytmów.	Wysoka trudność. Wymaga znajomości języka VHDL, elementów elektronicznych oraz dokładnego zaznajomienia się z funkcjonowaniem protokołu Modbus RTU.
Dokonywanie zmian w istniejącej sieci	Zmiana w jednej części programu może negatywnie wpłynąć na działanie całego algorytmu. Zmiana parametrów sieci nie sprawia trudności.	Edycja odpowiedniego fragmentu kodu, bez zagłębiania się w pozostałą część programu. Zmiana parametrów sieci nie sprawia trudności.
Sterowanie	Dostępne wejścia cyfrowe, możliwe podłączenie modułu z przełącznikami.	Dostępne przełączniki, przyciski oraz wejścia cyfrowe.
Monitorowanie sieci bezpośrednio na urządzeniu	Monitorowanie sieci ograniczone do diod wskazujących nadawanie oraz odbiór poszczególnych wiadomości.	Szeroki zakres dostępnych elementów do monitorowania sieci. Możliwość wyświetlania dowolnych parametrów poprzez skonfigurowanie algorytmu.
Zastosowanie	Przeznaczony do pracy w przemyśle, spełnia normy oraz posiada certyfikaty.	Pomocny przy projektowaniu prototypów.
Dostęp do danych	Dostęp do danych jest ograniczony.	Nieograniczony dostęp do danych podłączonych urządzeń i danych dotyczących transmisji.
Napięcie dla 1 logicznej	2,73 V	3,06 V
Napięcie dla 0 logicznego	-2,65 V	-3,46 V

## WNIOSKI KOŃCOWE

W ramach prac badawczych zbadano dwa układy sterujące przemiennikiem częstotliwości SINAMICS V20 poprzez protokół komunikacyjny Modbus RTU. Z przeprowadzonych badań wynikają następujące wnioski:

1. konfiguracja protokołu z wykorzystaniem platformy uruchomieniowej DE2-115 jest zdecydowanie trudniejsza. Należy poznać zasady programowania w języku VHDL oraz elementy elektroniczne, umożliwiające wdrożenie komunikacji. Dodatkowo, aby bezbłędnie skonfigurować połączenia komunikacyjne w standardzie Modbus, należy wnikliwie zapoznać się z dokumentacją opisującą protokół,
2. w języku VHDL budowanie średnio zaawansowanych algorytmów jest łatwiejsze. Realizacja ramek, kolejność nadawania i podział wiadomości na cykliczne oraz acykliczne jest bardziej przystępna dla użytkownika. Wykorzystanie i operacje na odebranych danych jest także mniej złożone,
3. wprowadzenie zmian w opracowanej sieci w języku LAD wymaga zaznajomienia się z całym algorytmem sterowania. Aby edytować kod w języku VHDL wystarczy

znaleźć odpowiednią sekcję programu, bez zbędnego wgłębiania się w całość. Zmiana parametrów takich jak prędkość transmisji, bit parzystości i bit stopu w obu rozwiązaniach nie sprawia żadnych trudności. Czas reakcji na zmiany parametrów jest zadowalający, jest on silnie zależny od prędkości transmisji,

4. konfiguracja dla sieci opartej o FPGA okazała się trudniejsza, natomiast dokonywanie zmian w istniejącej już sieci okazało się łatwiejsze. Środowisko pracy podczas tworzenia nowych rozwiązań wymaga przeprowadzenia wielu testów, porównywania wyników oraz poprawianie błędów. Dla takich omawianych w artykule zastosowań zdecydowanie zaleca się korzystanie z układu FPGA,

5. projekt zrealizowany na bazie sterownika można wykorzystać do prostych aplikacji w systemach automatyki.

Dodatkowa analiza listy rozszerzeń dostępnych na platformie DE2-115 pozwala na jej różnorodne zastosowania, zarówno do prac badawczych oraz do dydaktyki. Ponadto jest ona bardzo łatwa do rozbudowania. Przy użyciu kilku elementów można uzyskać bardziej zaawansowane układy. Instrukcja producenta [10] proponuje kilka z nich, między innymi odtwarzacz muzyki z kart pamięci SD, co odbywa się za pomocą wbudowanego czytnika kart i wyjścia audio. Po rozszerzeniu zestawu o mikrofon i zewnętrzne głośniki, przy użyciu kodeka audio uzyskamy sprzęt do karaoke. Ponadto dostępne na DE2-115 złącza pozwalają na podłączenie odtwarzacza DVD, po ustawieniu odpowiednich parametrów na odtwarzaczu obraz będzie przekazywany na wyjście VGA oraz na wyjścia liniowe audio, do których może zostać podłączony monitor i głośniki. W interesujący sposób można użyć także podczerwieni. Po wykorzystaniu karty SD z wgranymi plikami muzycznymi, odpowiedni kodek odszyfruje pliki, następnie za pomocą procesora Nios II sygnał zostanie przeniesiony na wyjście liniowe audio. Obsługa odtwarzacza jest możliwa dzięki dołączonemu do DE2-115 pilotowi.

**Podziękowanie:** Artykuł został sfinansowany na Wydziale Elektrycznym Uniwersytetu Morskiego w Gdyni ze środków projektu badawczego o numerze WE/2020/PZ/03.

**Autorzy:**

dr inż. Krystyna Maria Noga, Uniwersytet Morski w Gdyni, Katedra Automatyki Okrętowej, ul. Morska 81-87, 81-225 Gdynia, E-mail: k.noga@we.umg.edu.pl

dr inż. Monika Rybczak, Uniwersytet Morski w Gdyni, Katedra Automatyki Okrętowej, ul. Morska 81-87, 81-225 Gdynia, E-mail: m.rybczak@we.umg.edu.pl

LITERATURA

- [1] <https://www.se.com/us/en/product-range-presentation/574-modbus/>, (dostęp luty 2020)
- [2] <http://www.modbus.org/> (dostęp marzec 2020)
- [3] Kwaśniewski J., Sterowniki PLC w praktyce inżynierskiej. *Wydawnictwo BTC*, 2008
- [4] <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/br/br-quartus-prime-software.pdf> (dostęp styczeń 2020)
- [5] <https://www.xilinx.com/products/design-tools/ise-design-suite.html> (dostęp styczeń 2020)
- [6] <https://www.xilinx.com/products/design-tools/vivado.html> (dostęp styczeń 2020)
- [7] [www.4Self.pl](http://www.4Self.pl)
- [8] Noga K.M., Rabczuk D., Laboratoryjny system cyfrowy programowany przez Ethernet oparty na magistrali SPI, VI Konferencja e-Technologie w Kształceniu Inżynierów eTEE'2019, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, nr. 65, 111–114, 2019 r., ISSN 2353-1290, doi: 10.32016/1.65.18
- [9] <http://support.automation.siemens.com/WW/view/en/47755811>, (dostęp maj 2020)
- [10] [www.ee.ryerson.ca/~courses/ee8205/Data-Sheets/DE2-Board/Altera-DE2-115-User\\_Manual.pdf](http://www.ee.ryerson.ca/~courses/ee8205/Data-Sheets/DE2-Board/Altera-DE2-115-User_Manual.pdf), (dostęp luty 2020)
- [11] Węgrzyn M., Karatkiewicz A. Analiza porównawcza narzędzi syntezy Altera Quartus II i Synthagate. *Przegląd Elektrotechniczny*, 89 (12), 2013, 380-383