

Krzysztof PODLEJSKI*, Jarosław SZAFRAN*

MODELOWANIE STANDARDU PROFIBUS W ŚRODOWISKU LABVIEW

Środowisko LabVIEW dostarcza podstawowych narzędzi do zaprojektowania modelowania standardu PROFIBUS takich jak PROFIBUS Master i Slave służących do wykonania polecenia zapisu i odczytu rejestrów wejściowych, i wyjściowych, obliczania sumy kontrolnej CRC czy zapisu pojedynczego rejestru. Rodzina PROFIBUS składa się z trzech wersji DP, PA i FMS różniących się zastosowaniem i wymaganiami dotyczącymi transmisji i bezpieczeństwa także typami wykorzystywanych ramek protokołu komunikacyjnego. W Profibus wykorzystuje się dwa odmienne podejścia do organizacji ruchu w sieci. Pierwszym jest protokół odpytywania, w którym kontrola dostępu do magistrali jest scentralizowana Metoda druga oparta jest o zasadę pierścienia logicznego (*token passing*).

1. WPROWADZENIE

Profibus to standard przemysłowy przeznaczony do pomiarów, kontroli, sterowania i przesyłania danych w rozproszonych systemach automatyki określony normami DIN 19245 i EN50170. W skład standardu wchodzi trzy wersje: Profibus DP optymalizowany według kryterium dużej szybkości i efektywności transmisji przy możliwie niskich kosztach, Profibus PA stosowany w automatyzacji procesów produkcyjnych w środowiskach zagrożonych wybuchami (wersja iskrobezpieczna) i Profibus FMS wykorzystywany do komunikacji pomiędzy urządzeniami „inteligentnymi” (sterowniki PLC, komputery przemysłowe). Profibus FMS jest wersją uniwersalną wykorzystywaną najczęściej na poziomie wydziału systemu przemysłowego (poziom gniazd). Warstwa fizyczna dla wersji DP i FMS to RS485 i światłowód z wykorzystaniem powtarzaczy a dla wersji PA stosowany jest standard IEEE 1158-2 zapewniający wymagania iskrobezpieczeństwa. Dostępne jest tunelowanie protokołów poprzez sieć Ethernet i protokół TCP/IP. Organizacja komunikacji w sieci opiera się na dwóch

* Politechnika Wroclawska, Instytut Maszyn, Napędów i Pomiarów Elektrycznych, 50-370 Wrocław, ul. Smoluchowskiego 19, krzysztof.podlejski@pwr.wroc.pl

różnych rozwiązaniach: metodzie odpytywania (master-slave) i zasadzie pierścienia logicznego (token passing). W Profibus wykorzystuje się pięć typów ramek (komunikatów) i cztery rodzaje usług [2]–[4], [7], [8], [11].

Środowisko LabVIEW oferuje szerokie możliwości diagnostyczne standardu Profibus. Przykładowo pakiet „Comsoft” zawiera Profibus VIS, dostęp do urządzenia Slave–DP i danych diagnostycznych. Narzędzie PROFIBUS-DP-Master Express VI Standard umożliwia prosty dostęp do urządzenia Master []. Blok konfiguracyjny Express VI integruje środowisko LabVIEW w sieciach PROFIBUS, bez konieczności tworzenia dodatkowego oprogramowania []. Aplikacje napisane w LabVIEW mogą pracować pod kontrolą systemu MS Windows, Vista, Unix, Linux, Mac OS, itp., także w systemach czasu rzeczywistego. Zagadnienia modelowania elementów standardu PROFIBUS lub całej sieci omawiane są w przykładowych pracach [6], [9]–[13], [15]–[18].

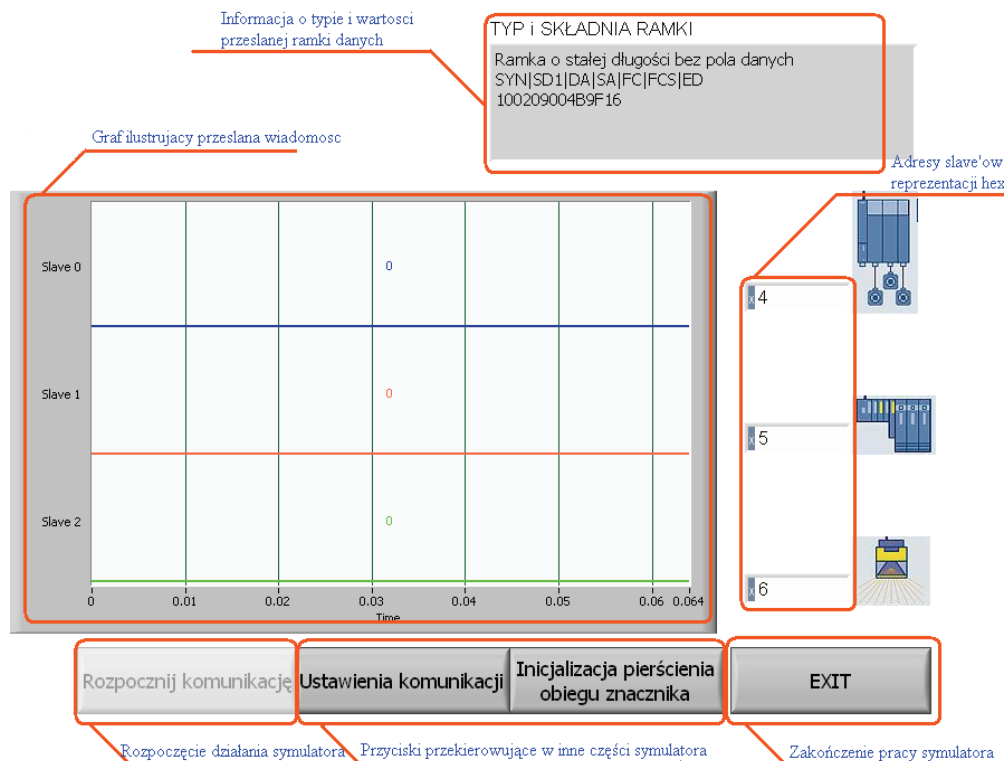
2. PRZYKŁADOWE ELEMENTY APLIKACJI MAINPROFIBUS

2.1. PODSTAWOWE STRUKTURY

Aplikacja modeluje działanie komunikacji pomiędzy urządzeniami Master i Slave w czterech przypadkach: praca poprawna, inicjalizacja komunikacji, utrata znacznika (tokena) w pierścieniu oraz podstawowe ustawienia magistrali (rys. 1). Program napisano w dwóch pętlach używając wielu zagnieżdżeń subVI. Pętla znacznika z częstotnością 1000 ms zmienia położenie znacznika a jej działanie aktywowane jest po wykonaniu ustawień w pętli głównej przechowującej dane o kolejności urządzeń Master w odpowiednich rejestrach, co jest warunkiem przekazywania znacznika zgodnie z priorytetami adresowymi. Pętla główna programu wykorzystuje architekturę maszyny stanów aktywną na zdarzenia wywoływane przez użytkownika. W tej architekturze podstawowymi składnikami są: pętla While zapewniająca ciągłość przejść między stanami, rejestr przesuwany zawierający aktualny stan maszyny oraz struktura Case odpowiadająca poszczególnym stanom. W implementacji pętli głównej aplikacji stany reprezentowane są, jako składowe zmiennej typu Enum [1], [13], [15].

Po uruchomieniu aplikacji wybierany jest stan „rozpoczęcie”, w którym zmiennym przypisane są wartości domyślne i ustawiane właściwości przycisków w panelu użytkownika Stan „bezczynność” jest stanem odniesienia, w którym umieszczona jest struktura „zdarzeniowa” reagująca na zdarzenia wywoływane przez kontrolki z obiektu Front Panel takie jak: ustawienia, inicjalizacja, zmiana adresów urządzeń Slave, i inne.

Realizację stanu bezczynności uzyskano przez wykorzystanie w strukturze zdarzeniowej elementu z aplikacji Timeout z częstotnością domyślną dla tej struktury. Definiowane są zdarzenia zmian wartości następujących kontrolki: inicjalizacja, ustawienia, start, exit, wyslij, zmiana adresu odbiorcy (urządzenia Slave).



Rys. 1. Panel ustawień symulatora

Stan „wyślij” realizuje wysyłanie ramki danych od urządzenia Master do określonego urządzenia Slave wywołując subVI: ramka.vi i subVI: string-to-digital.vi.

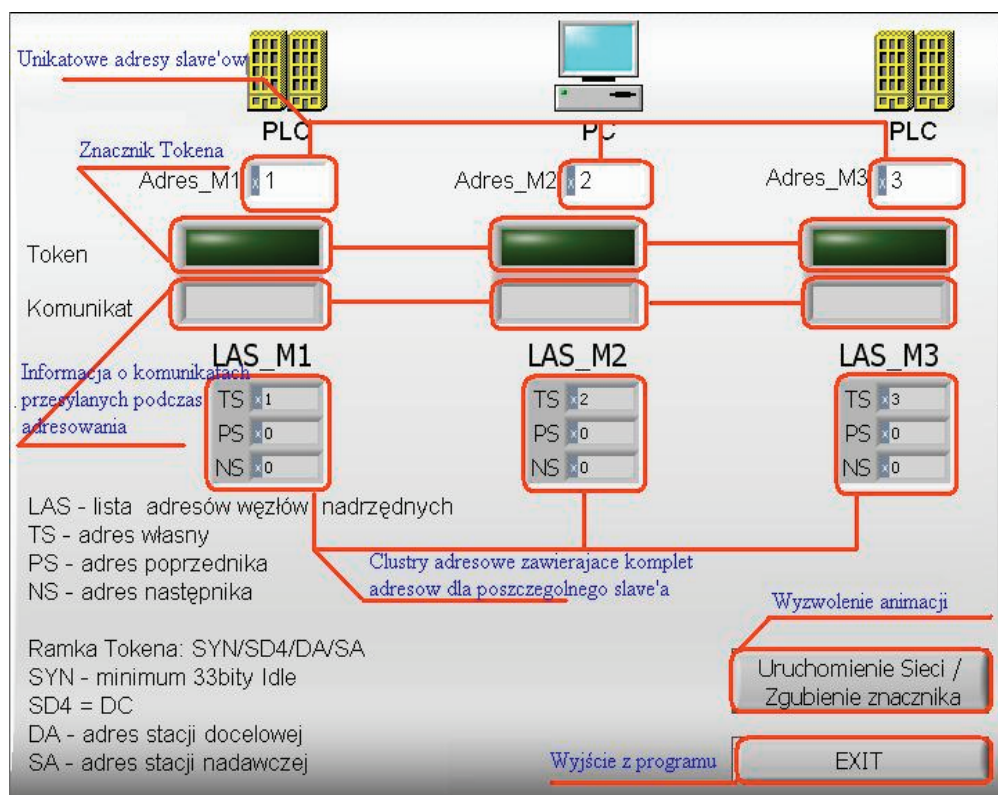
Stan „zakończ” kończy pracę aplikacji po aktywacji zdarzenia EXIT w panelu użytkownika.

2.2. ADRESOWANIE URZĄDZEŃ MASTER

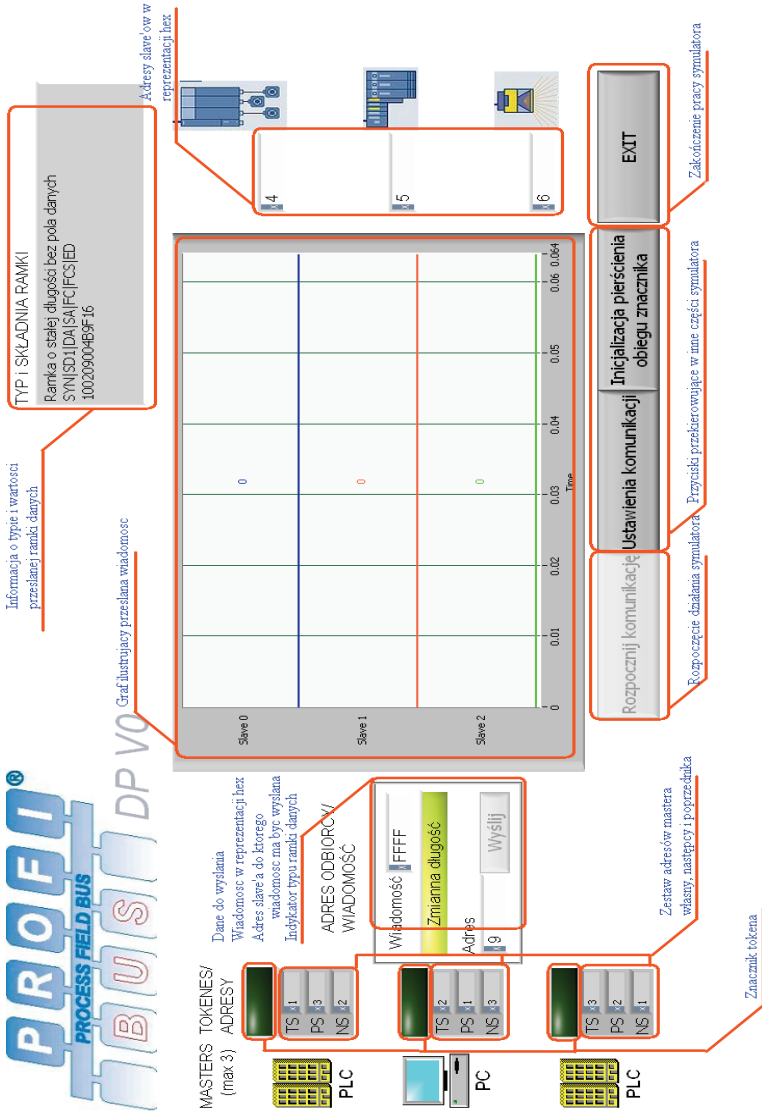
Adresowanie urządzeń Master jest wewnętrzną aplikacją programu głównego odpowiedzialną za inicjalizację obiegu znacznika w pierścieniu logicznym (z aktywnym panelem użytkownika przedstawionym na rys. 2). W poszczególnych stanach realizowane są następujące akcje:

- stan **start** realizuje powtórny inicjalizację zmiennych do wartości domyślnych oraz uruchamia bufor przechowujący adresy urządzeń Master. Dla zmiennych zastosowano referencję do kontrolki znajdujących się na panelu użytkownika programu głównego. Takie rozwiązanie umożliwia zarówno pobieranie aktualnej wartości, jak i ich nadpisywanie,

- stan **beczynność** wykorzystuje struktury zawierające obsługę następujących zdarzeń dotyczących zmiany wartości kontrolek: „exit”, „inicjalizacja” (przejście do stanu „inicjalizacja_ring” maszyny stanów), „A_M1” do „A_M3” (dodanie do bufora adresów urządzeń Master adresu przypisanego do wymienionych zmiennych),
 - stan **inicjalizacja_ring** rozpoczyna obieg znacznika w pierścieniu,
 - stan **stop** wykonuje czynności kończące pracę programu (np. przypisanie wartości zmiennych do zmiennych globalnych, zamknięcie panelu użytkownika).
- Główny panel użytkownika przedstawiono, wraz z opisem, na rys. 3.



Rys. 2. Inicjalizacja obiegu znacznika w pierścieniu logicznym



Rys. 3. Panel główny użytkownika

3. PODSUMOWANIE

Wykorzystując możliwości środowiska LabVIEW zamodelowano działanie standardu PROFIBUS. Po uruchomieniu symulatora (mainPROFIBUS) użytkownik dokonuje adresowania urządzeń typu Master i Slave. Rozpoczęcie komunikacji wymaga

niezbędnych ustawień (np. parzystość lub jej brak, prędkość transmisji). Wstępne czynności poprzedzają inicjalizację obiegu znacznika w pierścieniu logicznym, przy czym urządzenie Master o najniższym adresie ma najwyższy priorytet. Składowe komunikatu do urządzeń typu Slave zawierają typ ramki, jej zawartość i adres. Podstawowym zagadnieniem w realizacji aplikacji w LabVIEW jest typ zmiennej stałoprzecinkowej o długości 64 bitów przy wymaganiach standardu PROFIBUS często przekraczających tą wartość. Problem rozwiązano sklejając dane w odpowiednich etapach tworzenia ramki, co jednak powoduje dodatkowe przekształcenia typów danych, szczególnie widoczne przy obliczaniu sumy CRC.

LITERATURA

- [1] CHRUŚCIEL M., *LabVIEW w praktyce*, BTC, Legionowo 2008.
- [2] DEREŃ A., *Zastosowanie sieci przemysłowych*, Turck, Warszawa 2003.
- [3] MIELCZAREK W., *Interfejsy cyfrowe*, Helion, 2004
- [4] NAWROCKI W., *Rozproszone systemy pomiarowe*, WKŁ, Warszawa 2006.
- [5] [<http://sine.ni.com/nips/cds/view/p/lang/en/nid/201711>]
- [6] <http://www.imne.pwr.wroc.pl/> Krzysztof PODLEJSKI, Krzysztof BRUNICKI, „Zagadnienie komunikacji między sieciami przemysłowymi” PNIMNIPE,
- [7] www.profibus.org.pl
- [8] www.profibus.com
- [9] www.automation.com
- [10] www.prosoft-technology.com
- [11] <http://www.lumel.com.pl>
- [12] www.LabVIEW.pl
- [13] www.eti.pg.gda.pl
- [14] <http://www.ni.com>
- [15] <http://m6.mech.pk.edu.pl>
- [16] <http://www.ee.pw.edu.pl>
- [17] <http://automatykab2b.pl>
- [18] <http://www.anybus.com>

MODELLING STANDARD PROFIBUS IN LABVIEW

Modelled standard PROFIBUS of simulator mainPROFIBUS. Shown communication between master and slave devices. Describes the basic structure of the program, addressing the master and slave devices.