

Gabriel **KOST**
Wacław **BANAŚ**
Andrzej **NIERYCHŁOK**

WYKORZYSTANIE cRIO (KOMPUTERA CZASU RZECZYWISTEGO) W STEROWANIU ROZPROSZONYM

Streszczenie. W artykule przedstawiono sterowanie systemem rozproszonym z wykorzystaniem cRIO (komputera czasu rzeczywistego) zawierającego moduł sieci CAN. Systemem rozproszonym jest stanowisko dydaktyczne MPS. Komputer czasu rzeczywistego zawiera moduł FPGA, co umożliwia prawidłową i niezawodną pracę. Powiązane jest to jednak z odpowiednim oprogramowaniem komputera cRIO i układu FPGA. Konfiguracja i kod źródłowy wygenerowane zostały w środowisku NI LabVIEW.

Słowa kluczowe: system rozproszony, cRIO, magistrala CAN.

1. WPROWADZENIE

Komunikacja pomiędzy poszczególnymi elementami automatyki (sensory, akulatory, układy sterowania-US) możliwa jest dzięki przesyłowi sygnałów analogowych, binarnych lub z wykorzystaniem magistrali danych i specjalnie przygotowanych protokołów wykorzystujących tzw. „ramki danych” do przesyłu informacji. Współcześnie dąży się do komunikacji między poszczególnymi jednostkami sterowania wykorzystującymi magistralę danych, co związane jest bezpośrednio ze zmniejszeniem ilości przewodów służących do transmisji danych. Także większość dostępnych na rynku inteligentnych czujników lub aktuatorów wyposażona jest w interfejs sieci przemysłowej, przyczyniający się do szybszej implementacji tych elementów z istniejącą już strukturą sieci przemysłowej. Sieci przemysłowe są sieciami wymiany danych o specyficznych wymaganiach, różniących się w znaczący sposób od powszechnie znanej sieci komputerowej. Sieci przemysłowe muszą spełniać następujące warunki [1], [5], [6], [7], [8]:

- determinizm czasowy,
- niezawodność działania (odporność na czynniki zewnętrzne, np. zakłócenia pochodzące z pól elektromagnetycznych),
- możliwość wykrywania błędów,
- arbitraż oraz bit dominujący.

Powyższe warunki zostały uwzględnione w projektowaniu magistrali CAN, stworzonej na potrzeby rynku motoryzacyjnego przez firmę Bosch, a później wykorzystanej także jako magistrala wymiany danych w sieciach przemysłowych [1], [5], [6], [7], [8].

2. MAGISTRALA CAN

Magistrala CAN jest magistralą szeregową, w której wykorzystuje się metodę rozgłoszeniową. Każdy węzeł sieci może rozpocząć nadawanie komunikatu tylko po wcześniejszej kontroli dostępu do magistrali (czy magistrala jest wolna i można wysłać

komunikaty). W przypadku równoczesnego rozpoczęcia nadawania komunikatów przez dwa lub więcej węzłów sieci, dostęp do magistrali przypisuje się dzięki procesowi arbitrażu. Każdy węzeł sieci posiada przypisany priorytet dostępu do sieci, im priorytet wyższy, tym węzeł sieci jest ważniejszy dla całej magistrali [1], [5], [6], [7], [8].

Specyfikacja magistrali CAN wyróżnia dwie podstawowe warstwy (rys. 1) [1], [5], [6], [7], [8]:

- warstwę transmisji danych:
 - logiczna kontrola transmisji (LLC – Logical Control Link),
 - kontrola dostępu do medium (MAC – Medium Access Control),
- warstwę fizyczną.



ISO 11898-1: protokół CAN

ISO 11898-2: High-Speed CAN (do 1MBit/s)

ISO 11898-3: Low-Speed CAN (do 125kBit/s)

LLC: Logical Link Control

MAC: Medium Access Control

PLS: Physical Signaling

PMA: Physical Medium Attachment

MDI: Medium Dependent Interface

Rys. 1. Model ISO-OSI magistrali CAN [1]

Podwarstwa LLC obejmuje następujące zadania: obsługę transferu danych wraz z zdalnymi zapytaniami, wybór wiadomości akceptowalnych odebranych przez LLC, zarządzanie odzyskiwaniem danych oraz informowanie o obciążeniach.

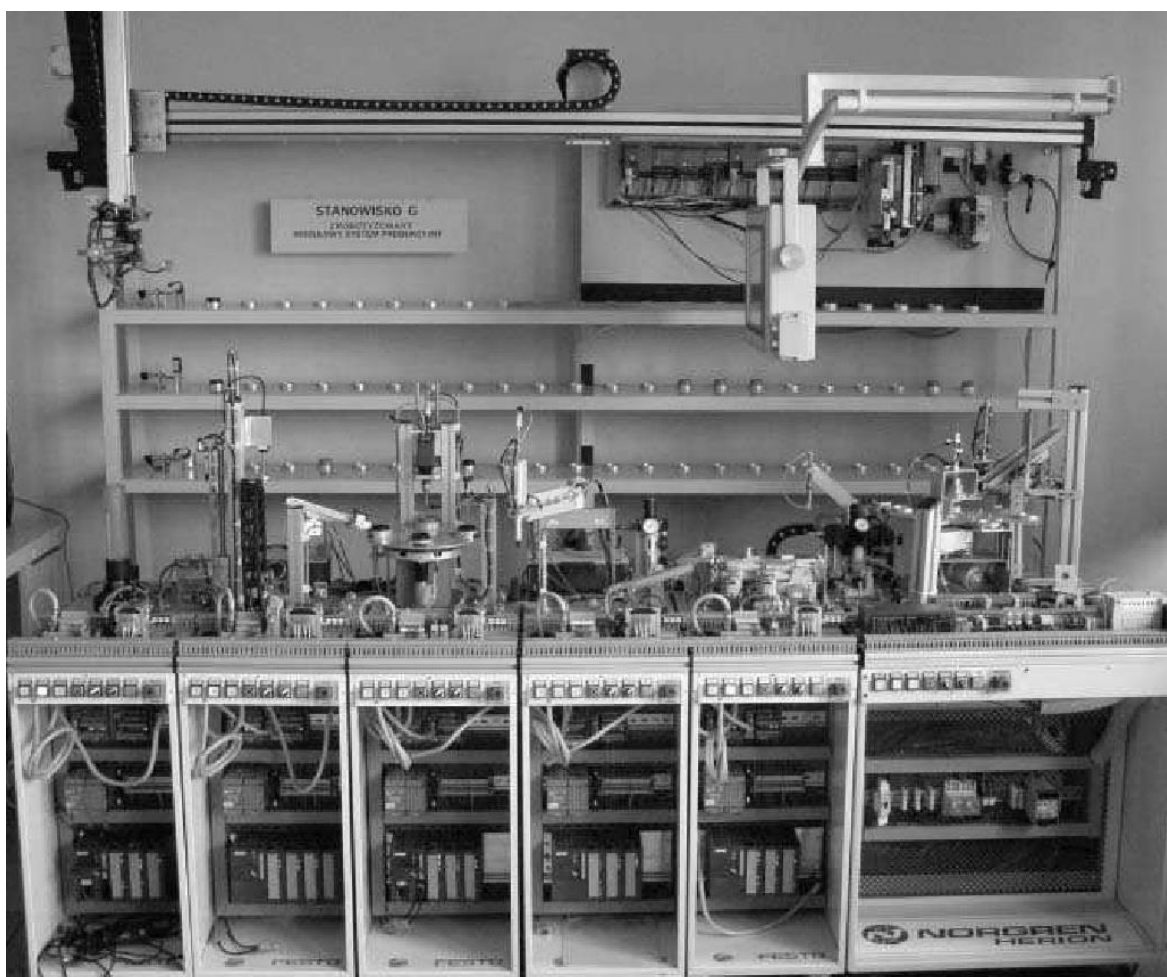
Podwarstwa MAC obejmuje zadania związane z protokołem transmisji: kontrola ramek, arbitraż, kontrola i sygnalizacja błędów, sprawdzenie stanu magistrali (wolna/zajęta), parametry czasowe.

3. ZASTOSOWANIE MAGISTRALI CAN DO STEROWANIA LINIĄ PRODUKCYJNĄ

W przypadku linii produkcyjnej, w której można wyróżnić kilkaset urządzeń automatyki, konieczne jest prawidłowe sterowanie i nadzór nad tymi urządzeniami. W tym celu wykorzystuje się sterowniki programowalne, pełniące rolę urządzeń sterujących oraz coraz częściej urządzeń nadzorujących pracę całej linii produkcyjnej. W Pracowni Automatykacji i Robotyzacji Procesów Technologicznych Instytutu Automatykacji Procesów Technologicznych i Zintegrowanych Systemów Wytwarzania znajduje się stanowisko dydaktyczne linii produkcyjnej MPS (Modular Production System) wraz z magazynem wysokiego składowania (rys. 2). Stanowisko MPS'a realizuje pełny proces technologiczny

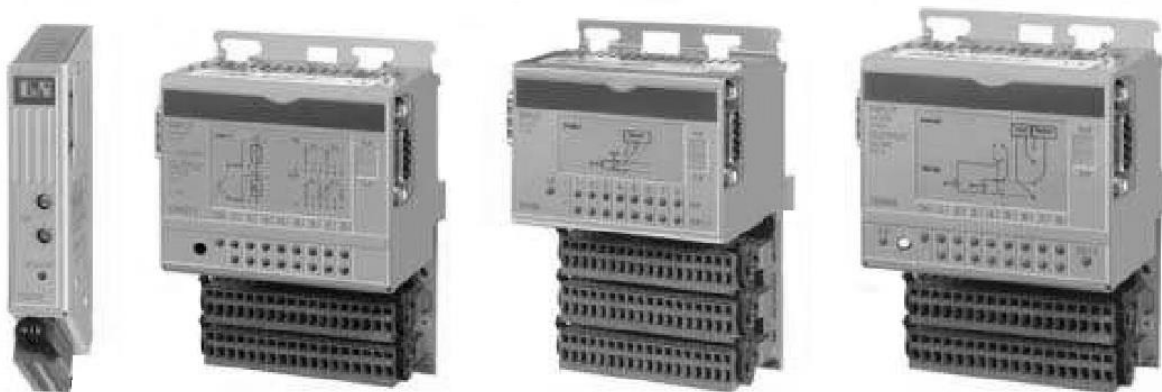
obróbki siłownika. Moduły stanowiska MPS pracują jako jedna zsynchronizowana linia produkcyjna, w której możemy wymienić następujące elementy (moduły linii):

- moduł dystrybucji,
- moduł kontroli,
- moduł obróbki,
- moduł transportu,
- moduł sortowania.



Rys. 2. Stanowisko dydaktyczne linii produkcyjnej MPS wraz z magazynem wysokiego składowania

Każdy z modułów pracuje pod kontrolą sterownika programowalnego firmy B&R wyposażonego w magistralę wymiany danych CAN. Moduły B&R serii System 2003 to najmniejsze sterowniki tej firmy, przeznaczone głównie do niewielkich rozwiązań sterowania procesem, wyposażone w bardzo przydatny interfejs CAN (rys. 3) [2]. Modułem nadrzędnym jest zintegrowane urządzenie Power Panel 4PP220.1505-B5. Urządzenie to wyposażone jest w 15" dotykowy wyświetlacz LCD, pełną gamę złączy I/O (RS-232, USB, Ethernet, itp.) oraz moduł CAN (rys. 4).

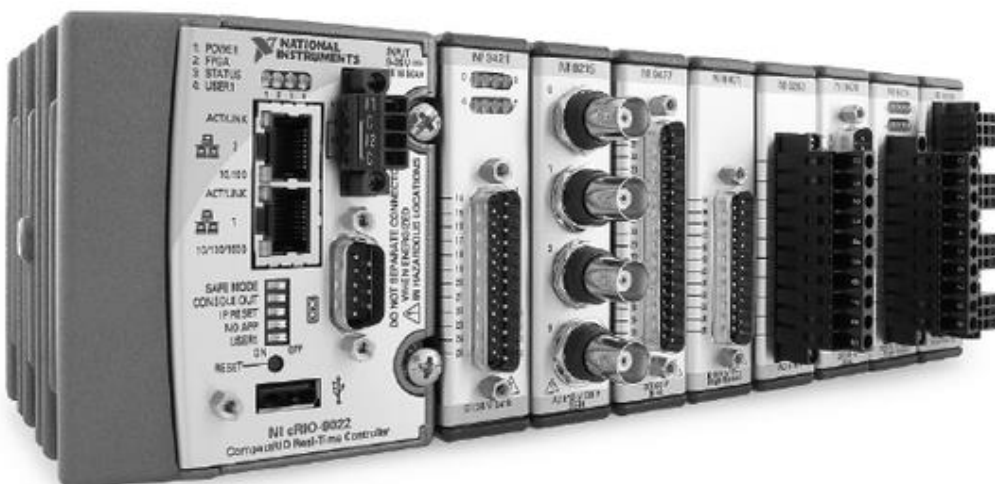


Rys. 3. Moduły B&R serii System 2003; od lewej: kontroler EX270, moduł CM 211, moduł wejść binarnych DI439, moduł wejść/wyjść binarnych DM 465 [3]

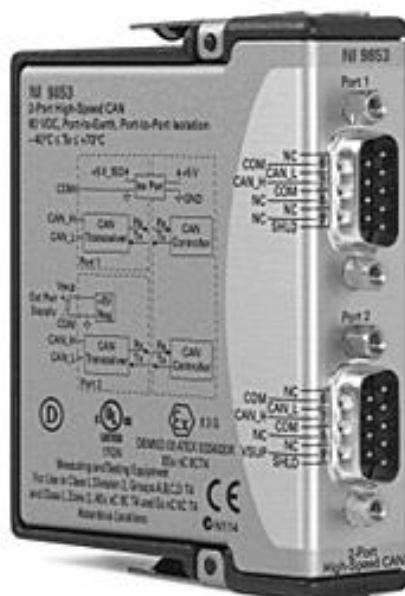


Rys. 4. Moduł nadrzędny Power Panel 4PP220.1505-B5 [3]

Do infrastruktury sieci układu sterowania zbudowanego na podzespołach automatyki B&R System 2003 oraz B&R Power Panel dodano układ kontrolno-sterujący zbudowany z komputera czasu rzeczywistego NI CompactRIO 9022 (rys. 5) oraz modułu CAN NI-9853 (rys. 6).



Rys. 5. Komputer RT CompactRIO 9022 wraz z szyną modułową oraz przykładowymi modułami NI [4]



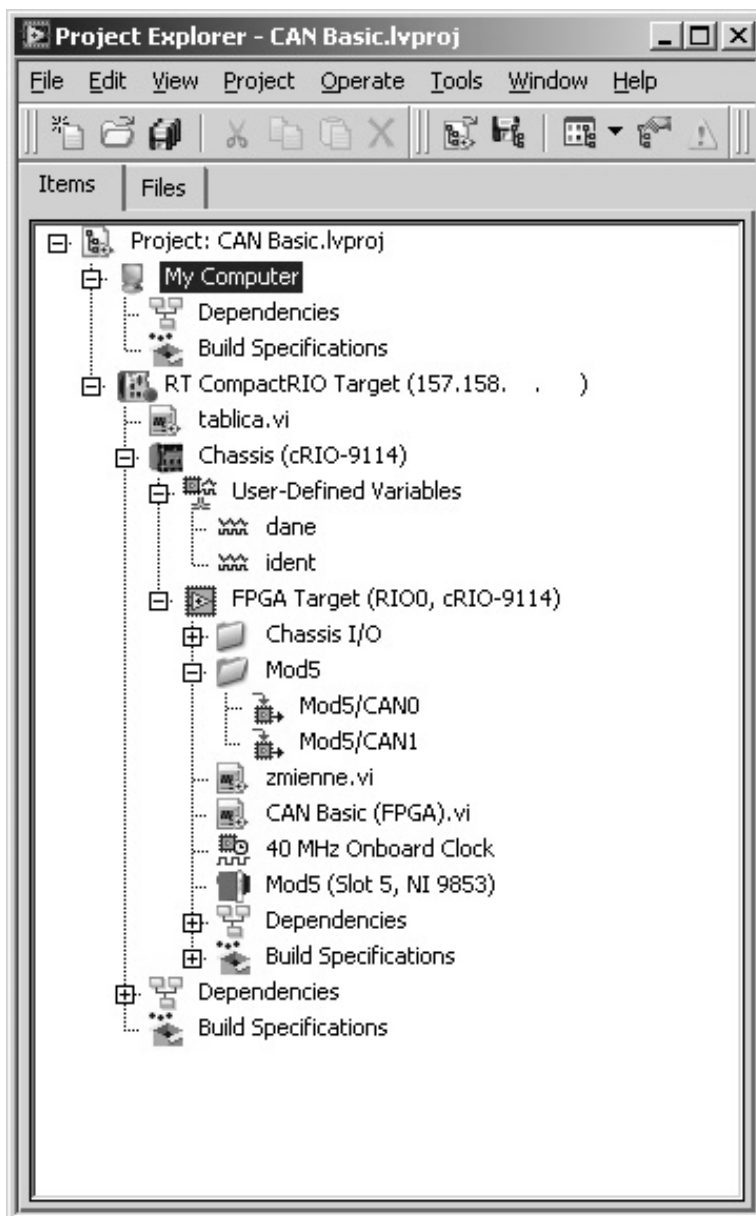
Rys. 6. Moduł CAN NI-9853 [4]

Wprowadzenie dodatkowego układu sterowania (komputera czasu rzeczywistego – CompactRIO) miało na celu usprawnienie procesu sterowania MPS'em poprzez wychwycenie błędnych komunikatów wysyłanych z węzłów sieci, „zapychanie” magistrali komunikatami nadawanymi zbyt często oraz diagnostykę elementów wykonawczych w poszczególnych węzłach sieci.

4. ROZWIĄZANIE PRAKTYCZNE

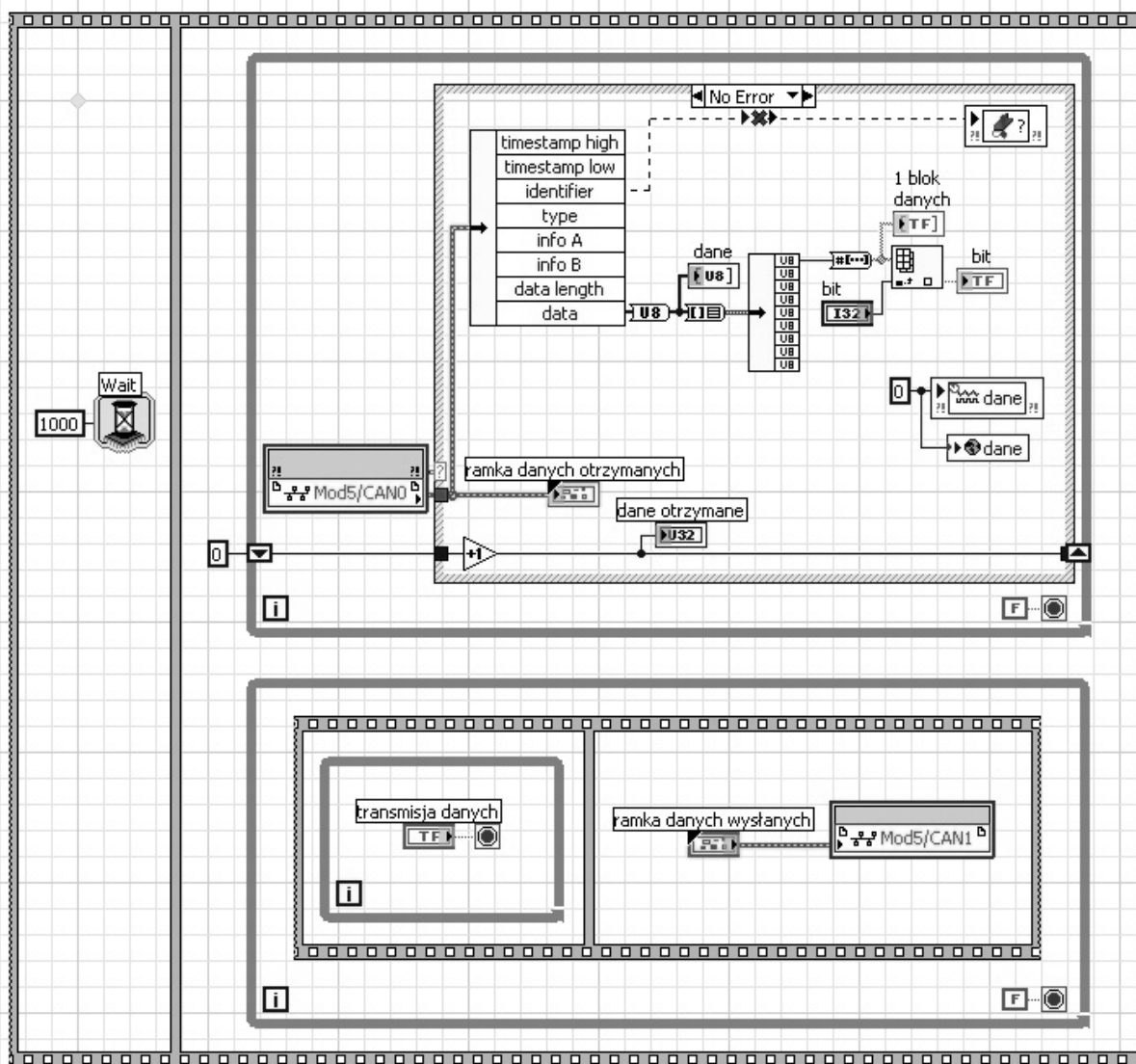
Do postawionych sobie zadań zbudowano kod programu kontrolujący się z komputerem czasu rzeczywistego oraz modułem CAN – NI9853. Moduł CAN pracujący w systemie czasu rzeczywistego wymaga pracy z modułem FPGA, czyli moduł CAN pracuje pod kontrolą komputera czasu rzeczywistego kontrolowanego programowalnym modułem

logicznym. Przedstawione okno projektu (programów zbudowanych w pracy) przedstawiono na rys. 7. Widoczny jest komputer czasu rzeczywistego „RT CompactRIO”, przypisany pod konkretny stały adres IP, szynę modułową „Chassis (cRIO-9114)”, w którą wpina się moduły kompaktowe NI, w tym moduł CAN NI-9853 (moduł osadzony został w slotie 5). Widoczne jest także uaktywnienie modułu logicznego FPGA, który dostępny jest wraz z szyną modułową cRIO-9114.



Rys. 7. Główne okno projektu programu

Zbudowany program (rys. 8 oraz rys. 9) pozwala na zbieranie informacji wysyłanych po sieci CAN i zapis tych informacji w pliku. Informacje tego typu mogą później zostać użyte do diagnozy, kontroli obciążenia magistrali, nadzoru i sterowania urządzeniami wykonawczymi itp.



Rys. 8. Kod programu zapisanego w LabVIEW (block diagram)

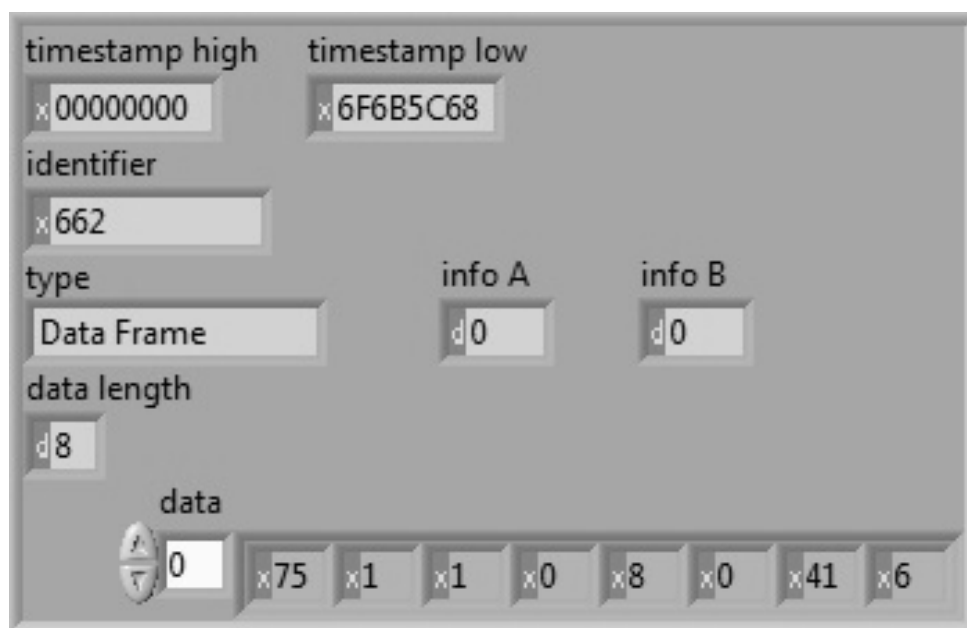


Rys. 9. Okno główne programu (front panel)

Przykładową ramkę danych odebranych informujących o przedmiocie znajdującym się w miejscu pobrania przez manipulator oraz o potwierdzeniu przemieszczenia przedmiotu przez manipulator przedstawiono na rys. 10 oraz rys. 11. Informacje przychodzące są zapisane językiem hexa (opisem szesnastkowych), które wymagają dekodowania.



Rys. 10. Dane odebrane



Rys. 11. Dane odebrane

W chwili obecnej trwają prace nad ulepszeniem interfejsu wraz z dodaniem czytelnych informacji (dekodowanie kodu hex), dodaniem wizualizacji graficznej stanowiska MPS, wraz z symulacją poszczególnych etapów pracy stanowiska.

5. WNIOSKI

W artykule przytoczono aspekty obsługi magistrali CAN, na której bazują sterowniki PLC firmy B&R (realizujące proces obróbki siłownika w modułowym systemie produkcyjnym) oraz z sukcesem nawiązano połączenie z zewnętrznym urządzeniem kontrolno-pomiarowym. CompactRIO firmy NI. Przedstawiono przykładowy program komunikujący się po magistrali CAN z urządzeniami B&R. W napisanym programie uwzględniono możliwość odbioru i nadawania komunikatów. Przyczyniło się to do weryfikacji obciążenia magistrali, wynikającej z niesprawnego czujnika położenia siłownika. W przyszłości planowana jest realizacja systemu nadzorczo-diagnostycznego, zbudowanego na komputerze czasu rzeczywistego CompactRIO oraz wprowadzenie pełnego cyklu pracy automatycznej modułu MPS.

6. LITERATURA

- [1] Lawrenz W.: CAN System Engineering. Springer, Germany 1997 r.
- [2] http://www.br-automation.com/cps/rde/xchg/br-productcatalogue/hs.xsl/cookies_allowed.htm?caller=products_64269_ENG_HTML.htm, 15.05.2012 r.
- [3] <http://www.br-automation.com>, 19.07.2012 r.
- [4] <http://www.ni.com>, 19.07.2012 r.
- [5] Pfeiffer O.: Embedded networking with CAN and CANopen. Copperhill Technologies Corporation. USA 2003 r.
- [6] Etschberger K.: Controller area network: basics, protocols, chips and applications. IXXAT Press, Germany 2001 r.
- [7] Voss W.: A comprehensible guide to controller area network. Copperhill Technologies Corporation. USA 2005 r.
- [8] Di Natale M., Zeng H.: Understanding and using the controllerarea network communication protocol. Springer. USA 2012 r.

THE USE OF cRIO (REAL-TIME COMPUTER) IN THE CONTROL OF A DISTRIBUTED SYSTEM

Abstract. This paper presents a distributed control system using cRIO (real-time computer) containing the CAN module. Distributed system is the didactic MPS. The real-time computer include FPGA module, which allows correct and reliable operation. However, this is associated with the appropriate computer software cRIO and FPGA. Configuration and source code have been generated in the LabVIEW software.

Key words: distributed system, cRIO, CAN bus.