

Niestandardowy sposób wymiany danych w hierarchicznym systemie sterowania dla orientowanych ogniw słonecznych

Krzysztof Oprzędkiewicz*, Kamil Szumlański**, Marcin Szejki*

*AGH Akademia Górniczo-Hutnicza

**Przedsiębiorstwo Usługowo-Produkcyjne "SKAMER-ACM" Sp. z o.o., Tarnów

Streszczenie: W pracy zaprezentowano system wymiany danych między aplikacją SCADA i mikrokontrolerowym sterownikiem zespołu orientowanych ogniw słonecznych. Aplikacja SCADA została przygotowana w środowisku Wonderware InTouch, sterownik został zbudowany na mikrokontrolerze STM-32L. Opracowany system wymiany danych zawiera dedykowany driver komunikacyjny oraz bazę danych. Wykonane testy wykazały poprawność zaproponowanych rozwiązań. Zaproponowane rozwiązania mogą znaleźć zastosowanie w każdej sytuacji, gdy korzystne jest połączenie aplikacji SCADA z mikrokontrolerem.

Słowa kluczowe: systemy SCADA, mikrokontrolery, orientowane ogniwa słoneczne

1. Uwagi wstępne

Systemy i aplikacje SCADA są szeroko stosowane w praktyce przemysłowej od wielu lat. Ich przydatność jest bezdyskusyjna, dostępne na rynku środowiska programistyczne oferują coraz większe możliwości funkcjonalne i są niezwykle „przyjazne” dla użytkowników, którymi są zwykle automaty, a nie informatycy.

Typowym obszarem zastosowań systemów SCADA jest monitorowanie i nadzór procesów i urządzeń przemysłowych, które są sterowane przez sterowniki PLC/PAC lub inne urządzenia podobnej klasy. Z tego względu do każdego środowiska programistycznego SCADA producent dostarcza stosunkowo duży zestaw driverów komunikacyjnych, dedykowanych do sterowników PLC/PAC różnych typów.

Jednocześnie, co jest charakterystyczne, żadne z dostępnych na rynku środowisk SCADA nie oferuje gotowych narzędzi komunikacyjnych do wymiany danych między aplikacją SCADA a specjalizowanymi sterownikami wbudowanymi, realizowanymi na mikrokontrolerach. Prawdopodobnie wynika to z faktu, że w przypadku tej klasy sterowników producent zwykle dostarcza proste środowisko do monitorowania i nadzoru pracy systemu. Jednocześnie funkcjonalność takiego systemu jest zwykle ograniczona, a modyfikacja nie jest możliwa ze względu na brak kodów źródłowych oprogramowania.

Omówiona sytuacja ma również miejsce w przypadku systemów sterowania zespołów orientowanych ogniw słonecznych. Sterowanie bezpośrednie jest zwykle realizowane z wykorzystaniem sterowników bazujących na mikro-

kontrolerach. Oprogramowanie nadzorujące i monitorujące dla tej klasy obiektów regulacji jest dostępne przeważnie jako element dedykowany do konkretnego systemu, zwykle dość kosztowny i o ograniczonej funkcjonalności.

Z powyższych względów celowe wydaje się opracowanie systemu SCADA w typowym środowisku programistycznym, jednocześnie umożliwiającemu współpracę z zespołem sterowników bazujących na mikrokontrolerze. Jako przykład rozważono sterownik dla zespołu orientowanych ogniw słonecznych, zrealizowanego na mikrokontrolerze STM-32L, zbudowany przez firmę Sat Control. Zaprezentowane rozwiązania mogą znaleźć zastosowanie w każdym systemie ze sterownikiem mikrokontrolerowym.

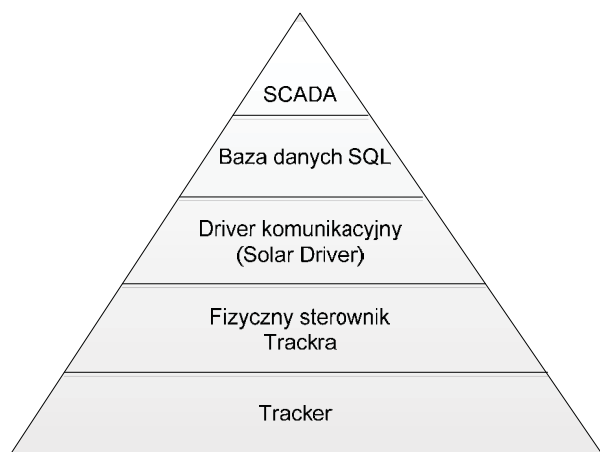
W pracy omówiono następujące zagadnienia:

- struktura systemu sterowania,
- architektura systemu wymiany danych,
- baza danych,
- driver komunikacyjny „Solar Driver”,
- funkcjonalność zbudowanego systemu,
- testy poprawności działania.

2. Struktura systemu sterowania

Rozważany hierarchiczny system sterowania ma nietypową dla tego typu systemów strukturę warstwową (rys. 1). Z punktu widzenia użytkownika na szczycie piramidy obrazującej hierarchię poszczególnych komponentów znajduje się aplikacja SCADA. Stanowi ona podstawowy element odpowiedzialny za komunikację z użytkownikiem. Niżej w hierarchii znajduje się baza danych SQL. Baza ta realizuje dwa zadania.

Pierwszym, nietypowym zadaniem bazy jest utworzenie „pomostu komunikacyjnego” między aplikacją SCADA a driverem komunikacyjnym, który jest następnym elementem systemu. Driver komunikacyjny (aplikacja Solar Driver) został zbudowany od podstaw w ramach realizacji projektu i jego zadaniem jest wymiana danych ze sprzętowym sterownikiem układu ogniw i udostępnienie ich systemowi SCADA. Solar Driver został przygotowany w środowisku .NET3.5. i teoretycznie powinien prowadzić wymianę danych bezpośrednio z systemem SCADA z użyciem mechanizmu DDE lub ActiveX. W praktyce jednakże wystąpiły trudności z współpracą sterownika ze środowiskiem InTouch wersja 10.0, zastosowanym w projekcie. Prawdopodobnie w przypadku użycia wersji 10.1 lub nowszej te kłopoty już nie wystąpią i system będzie mógł być prze-



Rys. 1. Hierarchiczna struktura rozważanego systemu
Fig. 1. The hierarchical structure of the considered system

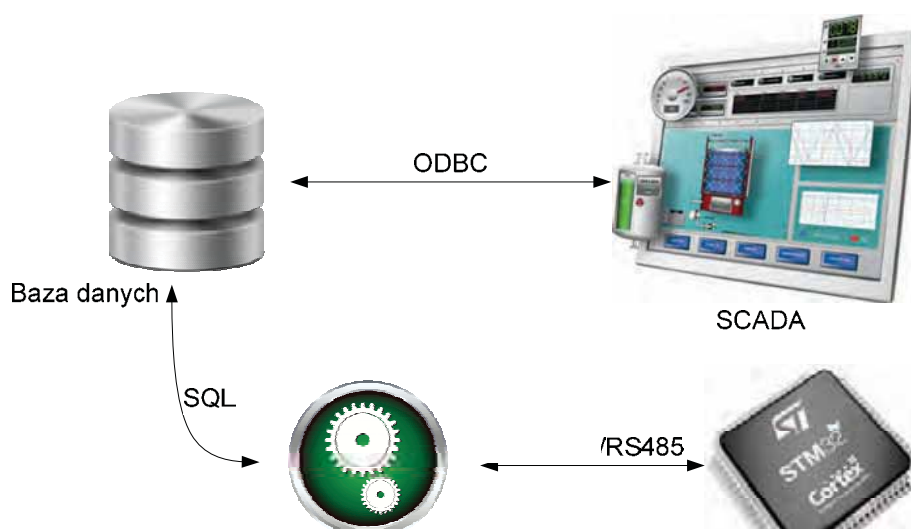
budowany do typowej postaci, w której driver komunikacyjny wymienia dane bezpośrednio z aplikacją SCADA.

Drugie zadanie realizowane przez bazę jest typowe i jest to archiwizacja danych związanych z pracą aplikacji i niezbędnych funkcji związanych z realizacją projektu. Dane te opisują między innymi „własne” zużycie energii przez układ sterowania, aktualne warunki atmosferyczne, zdarzenia występujące podczas pracy systemu itp.

Kolejnym elementem układu jest sterownik obiektu regulacji, którym jest zestaw orientowanych ogniw słonecznych (tzw. tracker). Sam tracker znajduje się w najniższej warstwie fizycznej nadążnego systemu fotowoltaicznego.

3. Architektura systemu wymiany danych

Ogólna architektura zbudowanego systemu wymiany danych pokazana jest na rys. 2. Do wymiany informacji między driverem komunikacyjnym a systemem SCADA



Rys. 2. Ogólna architektura zbudowanego systemu
Fig. 2. The general architecture of built system

wykorzystano interfejs ODBC z zewnętrzną bazą danych. Takie podejście pozwala archiwizować wszystkie parametry i informacje związane z pracą oraz sterowaniem układem. Dodatkowo otwiera ono wiele możliwości związanych np. z wykorzystaniem zdalnego dostępu do bazy danych za pośrednictwem strony WWW i urządzeń mobilnych wyposażonych w dedykowane oprogramowanie, które może zostać zrealizowane w przyszłości.

Praktyczna realizacja proponowanego systemu wymiany danych bazuje na następujących elementach:

- aplikacja SCADA,
- baza danych SQL,
- driver komunikacyjny,
- dedykowany sterownik, oparty na mikrokontrolerze, pełniący rolę serwera.

Aplikacja SCADA została zrealizowana z wykorzystaniem środowiska Wonderware InTouch wersja 10.0 i będzie szczegółowo opisana w następnej części artykułu.

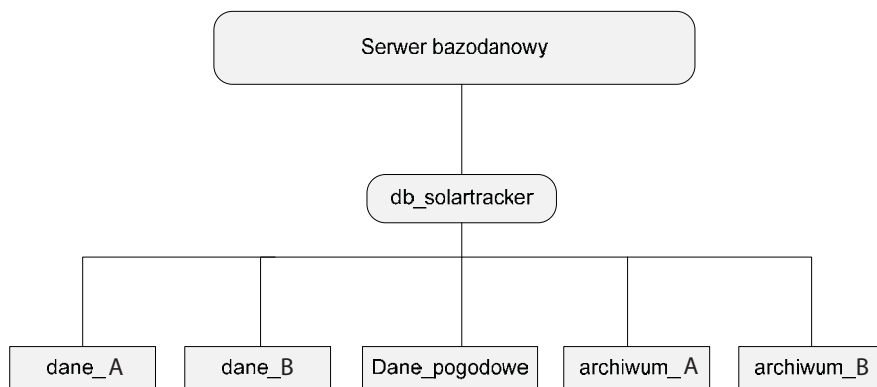
Komunikacja między aplikacją SCADA a sterownikiem odbywa się przez driver komunikacyjny oraz bazę danych, z którą SCADA ma bezpośredni kontakt. Komunikacja programu InTouch z bazą danych MySQL odbywa się za pomocą protokołu wymiany danych ODBC. Konfiguruje się go przez stworzenie listy połączeń, która przypisuje zmiennym w InTouch odpowiednie kolumny bazy danych według ustalonego schematu.

4. Baza danych

W projekcie zastosowano system zarządzania bazą danych MySQL 5.0 zaimplementowany w pakiecie WebServ. Baza danych została założona na serwerze MySQL i nosi nazwę „db_solartracker”. W bazie tej stworzono dwie tabele dla jednego urządzenia i ze względu na to, że baza została przewidziana do obsługi dwóch urządzeń, założono cztery tabele zawierające dane ze sterowników oraz dodatkowo jedną tabelę z danymi pogodowymi, wspólną dla obu urządzeń (rys. 3).

Funkcje tabel (rys. 3) są następujące:

- **dane_A** – zawiera informacje przychodzące ze sterownika urządzenia A oraz dane potrzebne do wymiany informacji między aplikacją SCADA a driverem komunikacyjnym,
- **dane_B** – zawiera informacje przychodzące ze sterownika urządzenia B i ma takie same pola jak tabela „dane_A”,
- **dane_pogodowe** – w tej tabeli są zapisywane dane o pogodzie, jaka panowała podczas zapisu informacji z któregoś urządzenia,



Rys. 3. Schemat bazy danych dla rozważanego systemu

Fig. 3. The scheme of database for the considered system

- **archiwum_A** – kopia zapasowa danych dla tabeli „dane_A”,
- **archiwum_B** – kopia zapasowa danych dla tabeli „dane_B”.

5. Driver komunikacyjny “Solar Driver”

Driver komunikacyjny jest głównym elementem rozważanego systemu. Z punktu widzenia interfejsu użytkownika stanowi element nadrzędny w stosunku do fizycznego sterownika. Do zadań drivera należy:

- nawiązanie komunikacji i wymiana danych z dedykowanym fizycznym sterownikiem systemu fotowoltaicznego za pośrednictwem standardu RS-485,
- kodowanie i dekodowanie odpowiednich komunikatów (rozkazów),
- monitorowanie aktualnych parametrów systemu,
- sterowanie systemem i konfigurowanie parametrów systemu,
- zapis i odczyt odpowiednich danych z/do zewnętrznej bazy danych.

Do realizacji wymienionych zadań wykorzystano metodę programowania wielowątkowego. Takie podejście do implementacji programu pozwoliło znacznie usprawnić działanie programu i wyeliminować problemy związane z obsługą portu szeregowego. Praktycznie realizowany system wymiany danych opiera się na standardzie RS-485. Takie podejście do realizacji komunikacji między komputerem PC i fizycznym sterownikiem pozwala stworzyć jeden uniwersalny driver komunikacyjny, kompatybilny zarówno z oryginalnym fizycznym sterownikiem urządzenia firmy Sat Control (zakupionym wraz z zespołem ogniw), jak również alternatywnym sterownikiem docelowym, bazującym na kontrolerze STM32. Sterownik ten jest aktualnie konstruowany i testowany w ramach projektu w Katedrze Automatyki i Inżynierii Biome-

dycznej Wydziału EAIiB AGH w Krakowie.

Pierwszy wątek odpowiedzialny jest za odczytywanie danych z zewnętrznego sterownika. Jego zadaniem jest nasłuchiwanie portu szeregowego oraz odpowiednia reakcja na odebrane dane. W przypadku odebrania informacji, odebrane dane są dekodowane, odpowiednio interpretowane i zapisywane do pamięci programu. Kolejnym etapem jest aktualizacja kontrolki programu oraz

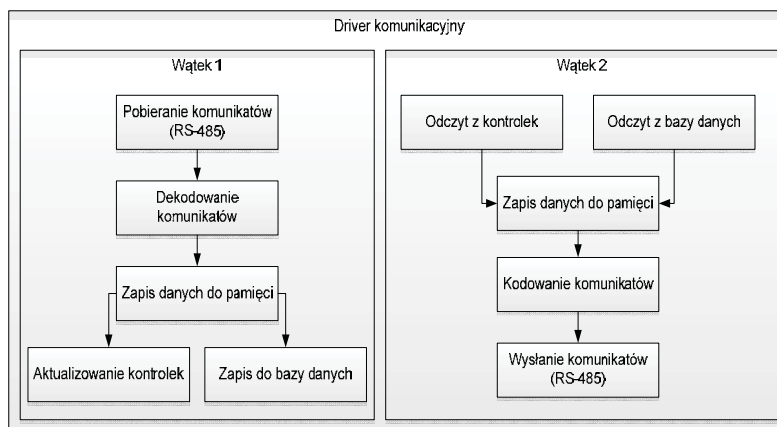
zapis do bazy danych. Cała sekwencja realizowana jest niezależnie od innych operacji wykonywanych w programie.

Drugi wątek pozwala przysyłać informacje w odwrotną stronę. Jego zadaniem jest odczyt odpowiednich danych z bazy danych oraz kontrolki programu, następnie ich zakodowanie i wysłanie do zewnętrznego sterownika za pośrednictwem portu szeregowego (rys. 4).

Fizyczna komunikacja ze sterownikiem może odbywać się z wykorzystaniem portu USB lub łącza RS-485.

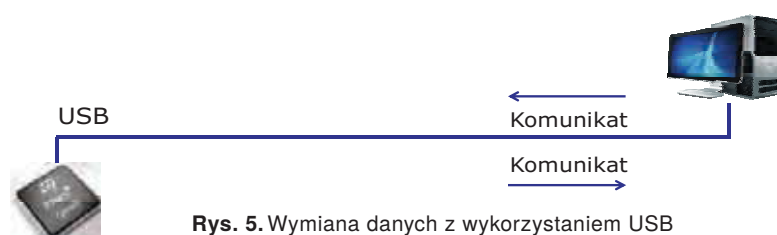
Komunikacja przy pomocy portu USB (rys. 5) umożliwia podłączenie tylko jednego sterownika fizycznego na jednej linii (większa liczba urządzeń wymaga zastosowania koncentratora lub osobnego portu dla każdego urządzenia), komunikaty między komputerem PC z zainstalowanym driverem komunikacyjnym a fizycznym sterownikiem wysyłane są niezależnie od siebie.

Komunikacja z wykorzystaniem portu RS-485 (rys. 6) umożliwia podłączenie wielu sterowników fizycznych. W tym wypadku system działa wg schematu klient-serwer, komputer PC z zainstalowanym driverem komunikacyjnym wysyła zapytanie z konkretnym adresem sterownika i oczekuje na odpowiedź. Program umożliwia jednocze-

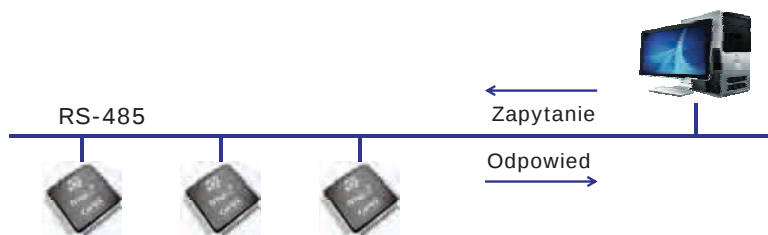


Rys. 4. Działanie wątków programu Solar Driver

Fig. 4. The work of threads in Solar Driver

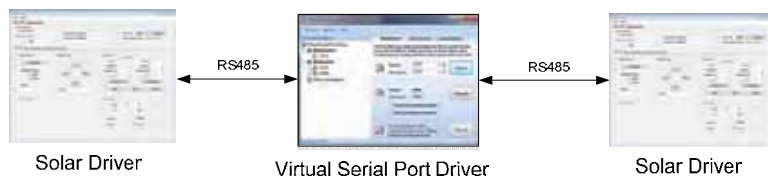


Rys. 5. Wymiana danych z wykorzystaniem USB
Fig. 5. Data exchange with the use of USB



Rys. 6. Wymiana danych z wykorzystaniem RS-485

Fig. 6. Data exchange with the use of RS-485



Rys. 7. Test wymiany danych – część 1

Fig. 7. The data exchange test – part 1

sne połączenie z dwoma urządzeniami, dzięki czemu możliwe jest porównanie ich działania. Urządzenia są naprzemiennie odpytywane, a ich parametry systemowe są zapisywane do bazy danych.

6. Funkcjonalność zbudowanego systemu

Prezentowany system wymiany danych realizuje następujące funkcjonalności:

- import danych ze specjalistycznego sterownika opartego na mikrokontrolerze za pośrednictwem standardu RS-485,
- archiwizowanie aktualnych danych i tworzenie danych historycznych do celów badawczych,
- udostępnianie zebranych danych nadrzędnej aplikacji SCADA.

Dla zbudowanego systemu można wskazać następujących jego użytkowników:

- „zwykły” użytkownik – obsługa systemu,
- aplikacja SCADA – traktowana również jako użytkownik,
- baza danych,
- driver komunikacyjny – traktowany także jako użytkownik.

Wymienionych użytkowników można podzielić na dwie klasy. Pierwszą z nich jest użytkownik fizyczny, czyli operator systemu SCADA, który ma dostęp do danych zebranych w bazie danych oraz może komunikować się z fizycznym sterownikiem zespołu ogni, co pozwala między innymi na definiowanie parametrów algorytmów regulacyjnych, przełączanie trybu pracy zespołu lub realizację pomocniczego sterowania ręcznego.

Należy dodać, że żadne z powyższych zadań nie jest „twardym” zadaniem czasu rzeczywistego i w związku z tym opóźnienia w torze wymiany danych między aplikacją SCADA i fizycznym sterownikiem urządzenia nie są czynnikiem, który może znacząco zakłócić pracę systemu.

Drugą klasą użytkowników są użytkownicy systemowi. Obie te klasy użytkowników mają różne poziomy uprawnień, niezbędnych do realizacji ich zadań. W omawianym systemie rozróżniamy trzech użytkowników fizycznych:

- zwykły użytkownik,
 - administrator,
 - programista.
- oraz dwóch użytkowników systemowych:
- driver komunikacyjny,
 - aplikacja SCADA.

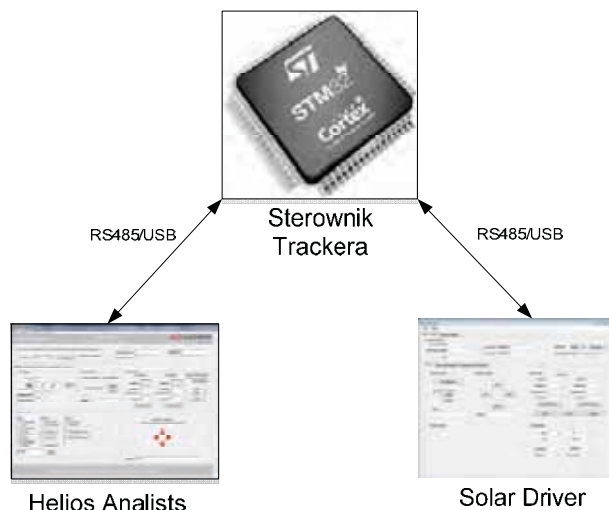
7. Testy poprawności działania systemu

W ramach weryfikacji poprawności działania opracowanego programu Solar Driver wykonano testy sprawdzające następujące funkcjonalności:

- wymiana danych za pośrednictwem standardu RS-485 oraz poprawność dekodowania komunikatów systemowych,
- bezpośrednie sterowanie fizycznego urządzenia z użyciem zbudowanego oprogramowania,
- wymiana danych pomiędzy programem Solar Driver i bazą danych.

Wyniki wszystkich testów potwierdziły poprawność działania zbudowanego oprogramowania. Poniżej jako przykład zostanie omówione sprawdzenie poprawności działania komunikacji za pośrednictwem standardu RS-485 oraz przetestowanie poprawności dekodowania komunikatów systemowych. Test ten składał się z dwóch części.

W pierwszej części testu sprawdzono działanie komunikacji na wirtualnych portach szeregowych utworzonych za pomocą programu Virtual Serial Port Driver. Na komputerze z zainstalowanym systemem operacyjnym Microsoft Windows XP uruchomiono dwie instancje programu Solar Driver. Każda z tych instancji została podłączona do jednego z dwóch skojarzonych ze sobą wirtualnych portów szeregowych (rys. 7). Test wymiany komunikatów odbywał się dla kilku wybranych zestawów konfiguracji portu (szybkość transmisji, parzystość, bit stopu, bity danych) zarówno dla znakowego typu transmisji jak również heksadecymalnego.



Rys. 8. Test i wymiany danych – część 2

Fig. 8. The data exchange test – part 2

Druga część testu polegała na sprawdzeniu poprawności działania wymiany danych między programem Solar Driver z fizycznym sterownikiem trackera (rys. 8) za pośrednictwem USB oraz RS-485. Po ustawieniu odpowiednich parametrów transmisji dokonano porównania wyświetlanych parametrów systemu w programie Solar Driver z danymi wyświetlanymi w oryginalnym programie Helios Analytics firmy Sat Control. W tym wypadku komunikacja również przebiegała prawidłowo.

Podczas realizacji obu części testu stwierdzono, że wymiana danych w wirtualnym systemie odbywa się poprawnie. Wszystkie odbierane komunikaty były zgodne z komunikatami wysłanymi, bez względu na konfigurację i tryb transmisji. Porównanie działania aplikacji Solar Driver z aplikacją Helios Analytics pozwoliło wykazać, iż proces dekodowania wykorzystywanych komunikatów przebiega prawidłowo.

8. Uwagi końcowe

Zaprezentowany układ wymiany danych w istotny sposób zwiększa funkcjonalność typowych aplikacji i środowisk SCADA, umożliwiając im współpracę ze sterownikami mikrokontrolerowymi.

Proponowane rozwiązanie może być także traktowane jako platforma bazowa do budowy systemów sterowania dla zespołów orientowanych ogniw słonecznych wykorzystujących nowe, nietypowe algorytmy, bazujące np. na prognozie pogody.

Podziękowanie

Praca została sfinansowana ze środków NCN, umowa nr 6693/B/T02/2011/40.

Bibliografia

1. Duszeńko A., *Budowa i zastosowanie interfejsów ODBC – JDBC*, mat. do przedmiotu „Bazy Danych”, Politechnika Śląska, 2010, dostępne pod adresem: zti.polsl.pl/bd3wysz/ODBC_JDBC.pdf.
2. Troelsen A., *Język C# 2008 i platforma NET 3.5*, PWN, 2009.
3. Szumlański K., Szejki M., *System wymiany danych pomiędzy aplikacją SCADA i dedykowanym systemem sterowania bazującym na mikrokontrolerze*, praca dyplomowa magisterska, zrealizowana na Wydziale EAIiE AGH w 2012 r. pod kierunkiem K. Oprządkiewicza.
4. [<http://msdn.microsoft.com/netframework>].
5. [<http://wsi.edu.pl/~sismolna/xml.html>].
6. [www.codewalkers.com/c/a/Miscellaneous/Using-SOAP-with-PHP/].
7. [www.controlengineering.pl/].
8. [www.cs.put.poznan.pl/csobaniec/edu/rso/rpc.pdf].
9. [www.ethernetprzemyslowy.pl/].
10. [www.mysql.com/products/connector/].
11. [www.wunderground.com/weather/api/d/documentation.html].
12. [www.w3.org/TR/2000/NOTE-SOAP-20000508].

Non-standard method for data exchange in the hierarchical control system for oriented PV

Abstract: In the paper a data exchange system between SCADA application and microcontroller-based controller for oriented PV system is presented. The SCADA application was built with the use of Wonderware InTouch, the controller was constructed with the use of STM-32L microcontroller. The data exchange system contains dedicated communication driver and a database. After tests it turns out, that all proposed solutions were proper and the system runs correctly. The proposed solutions can be applied in each situation, when a microcontroller-based control unit is applied.

Keywords: SCADA systems, microcontrollers, oriented PV systems

dr hab. inż. Krzysztof Oprządkiewicz

Absolwent Akademii Górniczo-Hutniczej im. S. Staszica w Krakowie z 1988 r. (Wydział Elektrotechniki, Automatyki i Elektroniki, kierunek studiów: Elektronika, specjalność Automatyka). Aktualnie zatrudniony w Katedrze Automatyki i Inżynierii Biomedycznej na Wydziale Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej AGH na stanowisku profesora nadzwyczajnego oraz w Państwowej Wyższej Szkole Zawodowej w Tarnowie na stanowisku profesora nadzwyczajnego. Zainteresowania naukowe: problemy sterowania systemami dynamicznymi o niepewnych parametrach, sterowanie cyfrowe, automatyka przemysłowa.

e-mail: kop@agh.edu.pl



mgr inż. Kamil Szumlański

Absolwent Instytutu Politechnicznego Państwowej Wyższej Szkoły Zawodowej w Tarnowie oraz Wydziału Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie na kierunku Automatyka i Robotyka. specjalność Komputerowe Systemy Sterowania z 2012 r. Obecnie zatrudniony w wydziale projektowym firmy SKAMER-ACM w Tarnowie.

e-mail: szumi88@gmail.com



mgr inż. Marcin Szejki

Absolwent Instytutu Politechnicznego Państwowej Wyższej Szkoły Zawodowej w Tarnowie oraz Wydziału Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie na kierunku Automatyka i Robotyka specjalność Komputerowe Systemy Sterowania z 2012 r. Zainteresowania: projektowanie, konstrukcja, zastosowanie robotów w przemyśle i w życiu codziennym oraz programowanie.

e-mail: mszejki@gmail.com

