

Tadeusz PIETRASZEK

Politechnika Śląska, Wydział Automatyki, Elektroniki i Informatyki

INTELIĞENTNY CZUJNIK W STRUKTURZE SIECI ROZLEGŁEJ

Streszczenie. W artykule przedstawiono projekt systemu inteligentnych czujników, z możliwością podłączenia do sieci Internet. Projekt obejmuje oprogramowanie inteligentnego czujnika mikroprocesorowego, konstrukcję centrali opartej na komputerze TINI, umożliwiającą podłączenie systemu do sieci Ethernet z wykorzystaniem protokołu TCP/IP oraz oprogramowania dla komputera PC. Komputer PC podłączony przez sieć Ethernet umożliwia gromadzenie i wizualizację wyników, a także diagnostykę i konfigurację sensorów.

SMART SENSOR IN WIDE AREA NETWORK ENVIRONMENT

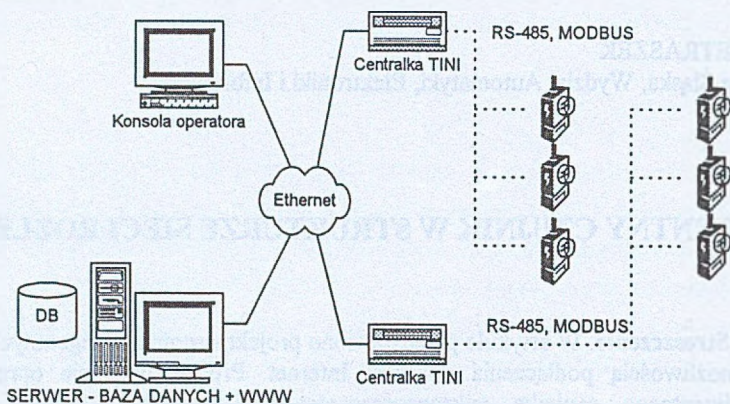
Summary. The paper presents the project of monitoring system based on smart sensors with Internet connection capability. The project covers the following areas: the design and implementation of microprocessor sensors, design of control station based on TINI computer, which allows connecting to Ethernet networks with TCP/IP protocol and software for personal computer. The software, running on personal computer, collects data from TINI stations, processes and stores them in the database. The other program is used for remote sensor monitoring and reconfiguration.

1. Wstęp

Czujniki znajdują zastosowanie praktycznie we wszystkich spotykanych obiektach. Czujnikiem określamy urządzenie, które ma za zadanie przetwarzanie sygnału opisywanego wielkością fizyczną na odpowiadającą mu wielkość analogową lub cyfrową.

Niniejszy artykuł, będący fragmentem pracy dyplomowej autora pod kierunkiem dr. inż. H. Małysiaka, dotyczy projektu systemu inteligentnych czujników z możliwością podłączenia do sieci rozległej za pomocą sieci Ethernet. W pierwszej części zostanie przedstawiony projekt oprogramowania inteligentnego czujnika gazu. Następnie przeanalizowane zostaną możliwości

podłączenia czujnika do centrali opartej na komputerze TINI umożliwiającego przesyłanie danych za pomocą sieci Internet oraz systemu przechowywania i wizualizacji danych z komputerem PC.



Rys. 1. Struktura systemu pomiarowego z inteligentnymi czujnikami
Fig. 1. The structure of measurement system with smart sensors

Na powyższym rysunku przedstawiono strukturę systemu pomiarowego z uwzględnieniem wszystkich elementów omawianych w artykule.

2. Pomiar stężenia gazów w obiektach

2.1. Informacje ogólne

W niektórych obiektach (np. garaże, tunele, kotłownie, stacje diagnostyczne, magazyny,...) ze względu na występujące zagrożenia istnieje konieczność monitorowania stężenia gazów lub innych substancji lotnych w powietrzu. Dokonuje się tego za pomocą czujników wyposażonych w element pomiarowy, który generuje sygnał zależny od stężenia badanej substancji w powietrzu. Ponieważ zależność sygnału od stężenia badanej substancji nie jest z reguły prostą zależnością liniową, istnieje konieczność przetwarzania sygnału.

Przetwarzanie może być realizowane przez układ analogowy lub układ cyfrowy. W przypadku, jeżeli wielkością wyjściową ma być sygnał analogowy, a samo przetwarzanie nie jest zbyt skomplikowane, stosuje się układy analogowe. W pozostałych przypadkach z reguły znajdują wykorzystanie układy mikroprocesorowe.

Wraz z rozwojem technologii i miniaturyzacji na rynku pojawiają się coraz tańsze i bardziej wydajne mikroprocesory, wyposażone w różnorodne układy peryferyjne. Oznacza

to, że mikroprocesor taki przejmuje coraz więcej funkcji, redukując liczbę dodatkowych układów analogowych i cyfrowych.

2.2. Informacje o czujnikach gazu

W artykule wprowadzone zostaje rozróżnienie pomiędzy czujnikami pomiarowymi i detekcyjnymi. Czujnik pomiarowy jest przetwornikiem przetwarzającym mierzoną wielkość fizyczną na sygnał elektryczny (analogowy lub cyfrowy), przy czym w zakresie pomiarowym sygnał wyjściowy odpowiada z określoną dokładnością konkretnej wartości wielkości stężenia gazu lub substancji. Czujnik detekcyjny (detektor) natomiast nie podaje wyniku stężenia, a jedynie wskazuje, czy są przekroczone określone wartości progowe. Dla przykładu, czujnikiem pomiarowym będzie czujnik gazu mierzący stężenie 0-100% DGW¹, natomiast czujnik gazu sygnalizujący jedynie przekroczenie progu alarmowego 150ppm jest czujnikiem detekcyjnym.

W praktyce stosuje się różne czujniki gazów i par lotnych, opierające się na różnych procesach fizycznych i chemicznych. Poszczególne czujniki różnią się parametrami, z których najważniejsze to:

- rodzaj gazu wykrywanego przez czujnik,
- stężenia gazu (lub substancji) poprawnie mierzone przez czujnik,
- największe dopuszczalne stężenie gazu (lub substancji), powyżej którego może nastąpić uszkodzenie czujnika,
- charakterystyka czujnika, rodzaj sygnału wyjściowego,
- zależność charakterystyki od parametrów zewnętrznych (np. wilgotność, temperatura),
- czułość skrośna określająca wpływ innych gazów na wynik pomiaru,
- trwałość, stabilność parametrów.

Typy stosowanych czujników oraz ich podstawowe parametry znajdują się m.in. w [1] i [7]. W projekcie inteligentnego czujnika zastosowane zostały półprzewodnikowe detektory SnO₂, wykorzystywane do wykrywania gazów, takich jak: metan, propan-butan, tlenek węgla.

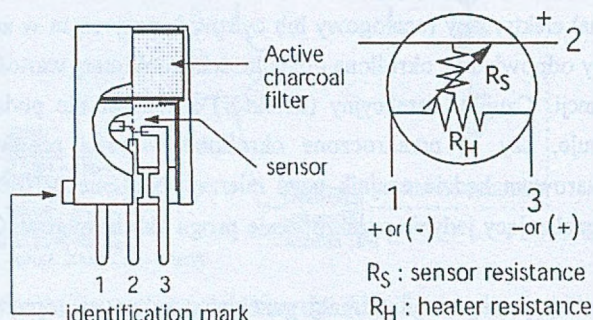
2.3. Informacje o detektorach SnO₂

Jako podstawowy detektor współpracujący z czujnikiem została wybrana rodzina detektorów SnO₂ (na podstawie [6]) i pod jej kątem zostało wykonane oprogramowanie.

¹ DGW – Dolna Granica Wybuchowości. Dla gazów palnych określa najniższe stężenie w powietrzu, powyżej którego możliwe jest samorzutne rozprzestrzenianie się płomienia na niespaloną część mieszaniny.

Jednocześnie starano się uzyskać na tyle elastyczny kod, by po niewielkich przeróbkach umożliwić współpracę z innymi detektorami.

Detektory SnO_2 działają na zasadzie oddziaływania gazu redukującego (takiego jak CO i CH_4) z powierzchnią półprzewodnika wewnątrz detektora. Opór detektora R_S zależy od stężenia badanego gazu i temperatury grzałki. Wygląd oraz model elektryczny sensora przedstawione są na poniższym rysunku:



Rys. 2. Wygląd oraz model elektryczny detektora CO i CH_4 - SB-95 [6]

Fig. 2. The physical outlook and electrical model of CO and CH_4 SB-95 detector [6]

Poniżej przedstawiono przegląd algorytmu sterowania sensora i etapy przetwarzania sygnału.

2.3.1. Sterowanie grzałką

Do poprawnego funkcjonowania sensora niezbędne jest cykliczne kluczkowanie napięcia grzałki. Kluczkowanie powoduje zmianę temperatury grzałki i detektora, a co za tym idzie zmianę parametrów. W zależności od typu detektora i rodzaju wykrywanego gazu w określonym momencie cyklu należy dokonać odczytu napięcia detektora. Dla przykładu, grzałka detektora SB-95 jest kluczkowana z okresem 10 - sekundowym (3 s faza wysoka, 7 s faza niska). Pomiar napięcia elektrody sensora następuje 100ms przed końcem każdej fazy. W fazie wysokiej mierzonym gazem jest metan, natomiast w fazie niskiej mierzone jest stężenie tlenu węgla.

2.3.2. Ekstrakcja

Jak wynika z modelu, wyprowadzenie detektora zachowuje się tak, jakby było rezystorem podłączonym w połowie rezystora grzałki. Do dalszego przetwarzania należy obliczyć wartość oporu sensora na podstawie zmierzonych napięć detektora i grzałki.

2.3.3. Prenormalizacja

W następnym etapie wartość oporu sensora jest prenormalizowana poprzez wymnożenie przez odpowiednią stałą, aby otrzymać zbliżone wartości dla wszystkich sensorów. Konieczność prenormalizacji oporu wynika z rozrzutu parametrów detektorów oraz możliwości wykorzystania różnych oporów pomiarowych w układzie.

2.3.4. Kompensacja środowiskowa i normalizacja

W tym etapie następuje zamiana prenormalizowanego oporu na wartość odpowiadającą rzeczywistemu stężeniu gazu. Należy uwzględnić to, że zależność ta jest silnie zależna od czynników zewnętrznych, takich jak: temperatura i wilgotność powietrza. Czynniki te są mierzone za pomocą czujnika temperatury i czujnika wilgotności, a następnie wyznaczany jest parametr środowiskowy uwzględniany przy normalizacji.

2.3.5. Testowanie poprawności działania sensora i sterowanie wyjść

W ostatnim etapie następuje testowanie poprawności mierzonych wielkości analogowych, ustawianie flag błędów, ostrzeżeń i alarmów oraz odpowiednie wysterowywanie wyjść układu.

3. Projekt inteligentnego sensora

Przedstawiony algorytm idealnie nadaje się do zaimplementowania w układzie mikroprocesorowym. Do implementacji został wybrany procesor firmy Microchip PIC16F873. Podstawowe wymagania dotyczące komunikacji procesora z otoczeniem są następujące:

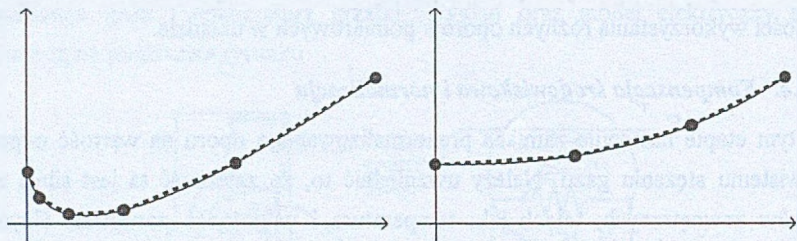
- sterowanie kluczowaniem grzałki i układu pomiarowego sensora,
- odczyt napięć grzałki, detektora, temperatury (KTY) (poprzez wbudowany 10-bitowy przetwornik A/C),
- generowanie sygnału analogowego 4-20mA (poprzez wbudowany generator PŹM),
- sterowanie wyjściami cyfrowymi – sygnalizacja LED,
- obsługa portu szeregowego.

3.1. Kalibracja sensora

Ze względu na duży rozrzut parametrów detektorów SnO₂, a także możliwość podłączenia różnych elementów detekcyjnych każdy inteligentny sensor musi być indywidualnie kalibrowany.

W założeniach kalibracja odbywa się poprzez interfejs szeregowy i polega na wpisaniu do pamięci EEPROM procesora odpowiednich tablic kalibracji. Tablica kalibracji składa się

z pewnej liczby par: *wartość X – wartość Y*. Dla argumentów pomiędzy *wartościami X* z tabeli procesor dokonuje liniowej interpolacji. Dzięki wykorzystaniu wskaźników w pamięci i dynamicznych tablic możliwe jest optymalne dobranie liczby elementów w tablicy do właściwości aproksymowanej funkcji.



Rys. 3. Optymalny dobór liczby odcinków interpolacji dla różnych rodzajów funkcji
Fig. 3. The optimal number of line interpolation segments for different kinds of functions

Dodatkowo możliwe jest przechowywanie wielu kopii tej samej tablicy w pamięci EEPROM procesora, co może odpowiadać kolejnym rekaliczacji czujnika. Zarówno pamięć danych, jak i programu jest chroniona sumą kontrolną – CRC16.

3.2. Protokół komunikacyjny

W trakcie tworzenia oprogramowania inteligentnego czujnika szybko okazało się, że interfejs szeregowy może być wykorzystywany zarówno do celów kalibracji, jak i do wymiany danych w czasie normalnej pracy. W tym celu zaistniała konieczność opracowania odpowiedniego protokołu komunikacyjnego.

Podstawowe wymagania stawiane przed protokołem są następujące:

- możliwość prostej implementacji na mikroprocesorze,
- brak interferencji z istotnymi funkcjami sensora, np. poprzez wymuszanie ścisłych zależności czasowych,
- możliwość dopasowania do specyfiki czujnika (odczyt danych pomiarowych i flag, zapis pamięci EEPROM, możliwość sterowania czujnikiem),
- kompatybilność z istniejącymi rozwiązaniami, możliwość podłączenia do sterowników przemysłowych.

W wyniku analizy zdecydowano się na protokół MODBUS opracowany przez firmę Modicon. Dzięki prostocie implementacji oraz dostępności specyfikacji jest on bardzo popularny w środowiskach sterowników przemysłowych. Protokół występuje w dwóch odmianach:

- MODBUS ASCII,
- MODBUS RTU.

Ze względu na uniezależnienie się od ścisłych zależności czasowych wybrano protokół MODBUS ASCII. Protokół ten charakteryzuje się dwukrotnie mniejszą wydajnością niż odmiana RTU, jednak nie jest ona kluczowa w omawianym rozwiązaniu.

Implementacja protokołu w czujniku zgodna jest z *Modbus class 0*, co oznacza konieczność implementacji funkcji FC03 i FC16. Dodatkowo, dla lepszej współpracy z popularnymi programami została zaimplementowana funkcja FC06. Ze względu na ograniczenie pamięci mikroprocesora rozmiar ramki MODBUS został ograniczony do 42 bajtów. Zapytania przekraczające ten rozmiar powodują wysłanie wyjątku *Exception response* przez czujnik.

3.2.1. Mapa pamięci MODBUS

Protokół MODBUS został zaprojektowany jako protokół wymiany danych pomiędzy sterownikami przemysłowymi firmy Modicon i nie zawsze odpowiada specyfice działania innych urządzeń. Z tego powodu dostęp do czujnika jest możliwy jedynie poprzez odczyt i zapis 16-bitowych rejestrów. Mapa pamięci widziana poprzez protokół MODBUS została opracowana tak, by była ona uniwersalna dla istniejących i przyszłych inteligentnych czujników.

Mapa składa się z następujących bloków pamięci:

- blok identyfikacyjny (wspólny dla wszystkich czujników), stałej długości pozwalający na określenie typu urządzenia i jego numeru seryjnego – tylko do odczytu,
- blok rejestrów wirtualnych charakterystycznych dla danego czujnika (ich interpretacja zależy od typu urządzenia) – tylko do odczytu,
- blok wirtualnych rejestrów sterujących charakterystycznych dla danego czujnika – do odczytu i zapisu,
- blok fizycznych rejestrów RAM czujnika – wykorzystywany w celach diagnostycznych – możliwy do modyfikacji po ustawieniu odpowiedniego rejestru sterującego,
- blok fizycznych rejestrów EEPROM czujnika – wykorzystywany przy kalibracji – możliwy do modyfikacji po ustawieniu odpowiedniego rejestru sterującego.

3.2.2. Wykorzystanie protokołu MODBUS

Zgodnie z założeniami protokół MODBUS wykorzystywany jest zarówno do kalibracji i testowania poprawności działania czujników, jak i w normalnej pracy, np. przy współpracy ze sterownikiem przemysłowym, centralą TINI lub innymi urządzeniami. Z tego też powodu istotna jest warstwa fizyczna sieci wykorzystywana przez inteligentny czujnik.

3.3. Warstwa fizyczna protokołu komunikacyjnego

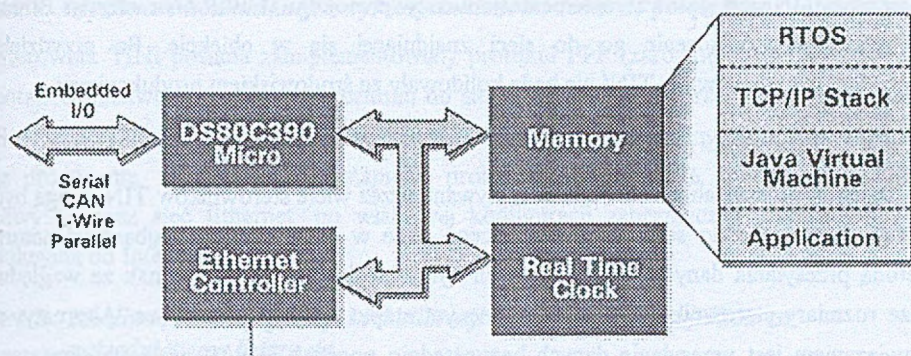
Inteligentne czujniki produkowane są w różnych wersjach, z różnymi interfejsami komunikacyjnymi:

1. Wersja podstawowa (czujnik bez interfejsu cyfrowego) – interfejs RS-232 z poziomami TTL - wykorzystywany jedynie do kalibracji i diagnostyki urządzenia – podłączenie wymaga otwarcia obudowy urządzenia.
2. Wersja Modbus – interfejs RS-485 – Modbus wykorzystywany do odczytu danych, diagnostyki i kalibracji urządzenia – wymaga podłączenia się do magistrali RS-485. Po otwarciu obudowy urządzenia możliwe jest podłączenie do interfejsu RS-232 z poziomem 5V.
3. Wersja Modbus IrDA – interfejs podstawowy RS-485 oraz dodatkowo interfejs IrDA zgodny z IrCOMM, umożliwiający obsługę urządzenia bez konieczności podłączania się do magistrali RS-485. W momencie zestawienia połączenia IrDA – zewnętrzna magistrala RS-485 jest blokowana i czujnik współpracuje z łączem podczerwonym. Moduł IrDA wykorzystuje układ firmy Microchip – MCP 2150.

4. Projekt centrali z komputerem TINI

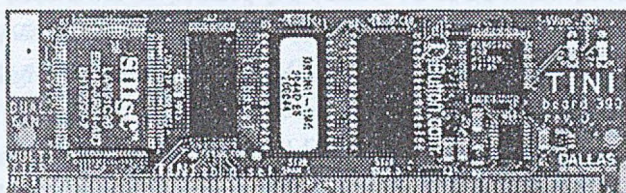
Inteligentny czujnik z interfejsem Modbus i RS-485 umożliwia podłączenie go do istniejących sieci przemysłowych bazujących na sterownikach przemysłowych [2]. Rozwiązanie takie będzie stosowane w przypadkach większych instalacji, w których niezawodność i szybkość działania są najważniejsze.

Alternatywnym rozwiązaniem jest wykorzystanie komputera TINI[5][9] jako centrali sterującej pewną liczbą inteligentnych czujników. TINI (Tiny InterNet Interface) jest platformą opracowaną przez firmę Dallas Inc. specjalnie do uruchamiania i projektowania różnorodnych systemów procesorowych podłączanych do sieci z protokołem TCP/IP. Platforma TINI oparta jest na mikrokontrolerze DS80C390 (zgodny z 8052) i systemie operacyjnym z maszyną wirtualną Javy™ (Java VM). Sterownik zawiera pamięć statyczną RAM, flash EEPROM, sterowniki do portów szeregowych oraz magistrali OW oraz kontroler sieci Ethernet z interfejsem 10Base-T. W aplikacjach interfejsy komunikacyjne obsługuje w sposób zbliżony do innych środowisk Javy. Pełna dokumentacja języka oraz opis API są dostępne w postaci elektronicznej na stronach producenta [10]. Praktyczne informacje dotyczące programowania komputera TINI znajdują się w [4].



Rys. 4. Struktura sterownika TINI [9]

Fig. 4. The structure of TINI microcontroller [9]



Rys. 5. Wygląd kompletnego sterownika TINI

Fig. 5. The outlook of complete TINI board

4.1. Podstawowe funkcje centrali

Do podstawowych funkcji centrali należy odczyt podłączonych do niej czujników oraz sterowanie podłączonymi modułami wyjściowymi. Odczyt podłączonych czujników dokonywany jest za pomocą interfejsu szeregowego i protokołu Modbus. Na podstawie odczytanych danych (oraz innych sygnałów wejściowych) centrala może generować sygnały wyjściowe.

Sygnały wejściowe i wyjściowe mogą pochodzić z niezależnych modułów I/O podłączonych za pomocą interfejsu szeregowego lub z układów podłączonych do magistrali OneWire, np. DS2406 - podwójnego I/O cyfrowego, DS2450 - poczwórnego 8-bitowego przetwornika A/D, DS2890 - potencjometru cyfrowego. Dodatkowo można wprowadzić klucz operatora (układ iButton), umożliwiający np. wyłączenie sygnalizacji alarmu.

4.2. Dodatkowe funkcje centrali

Powyższe podstawowe funkcje centrali mogłyby być realizowane przez dowolny układ mikroprocesorowy, prawdopodobnie znacznie tańszym kosztem. Podstawową przewagą

komputera TINI jest gotowa implementacja warstw protokołu TCP/IP oraz interfejs Ethernet, co umożliwia połączenie go do sieci znajdującej się w obiekcie. Po przydzieleniu odpowiednich adresów IP, TINI nie będą kolidowały ze środowiskiem produkcyjnym.

4.2.1. Przesyłanie danych do serwera

Dzięki wykorzystaniu sieci dane odczytywane przez wiele sterowników TINI mogą być na bieżąco przesyłane do serwera gromadzącego dane w bazie danych. Najbardziej naturalną metodą przesyłania danych do bazy danych byłby mechanizm JDBC, jednak ze względu na duże rozmiary sterowników JDBC na obecnym etapie nie jest to możliwe. Alternatywnym rozwiązaniem jest przesyłanie danych bezpośrednio poprzez TCP/IP lub z wykorzystaniem mechanizmów zdalnego wywoływania procedur. Ze względu na dużą uniwersalność i niezależność od platformy sprzętowej oraz na powszechne stosowanie tego protokołu w przetwarzaniu rozproszonym zdecydowano się na wykorzystanie protokołu XML-RPC. W tej sytuacji sterownik TINI pełni rolę klienta XML-RPC, który wywołuje procedury programu działającego na serwerze PC pod kontrolą systemu Linux lub Windows. Program ten zapisuje dane odebrane z jednego lub więcej sterowników TINI do relacyjnej bazy danych za pomocą interfejsu JDBC. W ogólnym przypadku baza może znajdować się na tym samym lub innym komputerze. Dane z bazy mogą być następnie pobierane przez różne programy w celu obróbki i wizualizacji.

4.2.2. Wizualizacja stanu centrali za pośrednictwem WWW

Na sterowniku TINI można uruchomić usługę serwera HTTP, który może służyć do monitorowania stanu centrali i przeglądania wyników pomiarów. Funkcja taka jest szczególnie użyteczna w momencie uruchamiania systemu lub wystąpienia problemów z serwerem gromadzącym dane lub konsolą operatorską. Ze względu na małe możliwości przetwarzania danych przez sterownik TINI zdecydowano przeniesienie części przetwarzania na stację kliencką. Dane udostępniane są przez WWW w formacie XML, a następnie przetwarzane przez przeglądarkę na podstawie arkuszy stylów XSL. Dzięki temu rozwiązaniu istnieje możliwość różnorodnego przetwarzania tych samych danych bez konieczności modyfikacji programu sterownika.

4.2.3. Zdalne sterowanie wyjściami centrali

Za pośrednictwem Internetu możliwe jest także sterowanie wyjściami centrali ręcznie przez operatora lub automatycznie przez program. W tym celu na sterowniku TINI zaimplementowano obsługę serwera XML-RPC. Sterowanie TINI odbywa się poprzez zdalne wywoływanie udostępnionych procedur.

4.2.4. Możliwość komunikacji z wykorzystaniem modemu do połączenia z Internetem

Sterownik TINI posiada zaimplementowany protokół PPP (zarówno stronę serwera, jak i klienta). Umożliwia to podłączenie centrali do sieci Internet (np. poprzez numer dostępowy TP S.A. 020 2122) i zdalną diagnostykę centrali i wszystkich podłączonych do niej czujników przez producenta, w przypadku wystąpienia problemów. Oczywiście, dostęp również jest możliwy poprzez sieć Ethernet, po właściwej konfiguracji zabezpieczeń chroniących sieć produkcyjną od Internetu.

4.2.5. Możliwość rekonfiguracji i rekaliibracji inteligentnych czujników za pośrednictwem Internetu

Centralka z TINI po przestawieniu w tryb konfiguracji sensorów umożliwia dostęp do wewnętrznej sieci MODBUS za pomocą protokołów XML-RPC lub MODBUS/TCP[3]. Oznacza to możliwość odczytu i modyfikacji dowolnych rejestrów inteligentnych czujników, co umożliwia ich diagnostykę, rekaliibrację i konfigurację.

4.3. Bezpieczeństwo serwera TINI

Możliwości, które wynikają z podłączenia centralki do sieci Internet, otrzymujemy kosztem ryzyka nieautoryzowanego dostępu do centralki. Podstawowym założeniem jest to, że sieć produkcyjna z centralkami z TINI jest wyodrębnioną podsiecią, do której nie mają dostępu osoby niepowołane. Sieć taka może być podłączona do sieci korporacyjnej i Internetu za pośrednictwem serwera typu *firewall* autoryzującego przechodzące połączenia.

Sam dostęp do serwera TINI za pomocą protokołów TELNET i FTP jest chroniony hasłem, jednak z zasady działania tych protokołów wynika możliwość przechwycenia hasła przesyłanego przez sieć.

Protokół MODBUS/TCP nie obsługuje identyfikacji i autoryzacji użytkowników, jedynie można ograniczyć adresy komputerów, z których dokonywane są połączenia. W obecnej implementacji ze względu na ograniczone zasoby nie wykorzystano uwierzytelniania XML-RPC.

4.4. Wydajność serwera TINI

Łatwość programowania i możliwość wykorzystania procedur wysokiego poziomu w języku Java okupiona jest znacznie mniejszą wydajnością sterownika, niż gdyby był on programowany w C czy assemblerze. Problemem również może być różny czas trwania tych samych operacji ze względu na działający proces *garbage collector*. Ze względu jednak na

specyfikę środowiska czujników gazu (odczyty uaktualniane co kilkanaście sekund) czasu przesyłania danych do serwera rzędu kilku sekund nie są poważnym problemem.

5. Projekt oprogramowania dla komputerów PC

Trzecim równie istotnym elementem systemu, oprócz projektu inteligentnego czujnika i centrali opartej na TINi, jest projekt oprogramowania dla komputerów PC umożliwiającego poprawne działanie systemu. Oprogramowanie można podzielić na 2 grupy:

1. Programy serwisowe do diagnostyki i konfiguracji czujników.
2. Programy użytkowe do obsługi działających czujników.

Ze względu na możliwość wykorzystywania części kodu pisanego dla centrali, a także na łatwość korzystania z funkcji TCP/IP i baz danych oraz przenośność kodu zdecydowano się na wybór języka Java do pisania programów serwisowych i użytkowych.

5.1. Programy serwisowe

Celem programu serwisowego jest umożliwienie łatwej diagnostyki i zmiany konfiguracji czujników. Program taki powinien również wspomagać proces kalibracji czujnika, przeprowadzany przez autoryzowany personel. Możliwe jest podłączenie do inteligentnych czujników następującymi metodami:

1. Bezpośrednie podłączenie do czujnika za pomocą protokołu MODBUS poprzez interfejs szeregowy RS-232 w komputerze.
2. Bezpośrednie podłączenie do sieci czujników za pomocą protokołu MODBUS poprzez interfejs szeregowy RS-232 w komputerze (z dodatkowym konwerterem RS-232 ↔ RS-485).
3. Bezpośrednie podłączenie do pojedynczego czujnika za pomocą interfejsu IrDA (z wykorzystaniem driverów IrCOMM emulujących port szeregowy w systemie Windows).
4. Podłączenie do centrali TINi przez sieć TCP/IP i przesyłanie komend do czujników za pośrednictwem protokołu XML-RPC lub MODBUS/TCP.

Program serwisowy umożliwia odczytanie pomiarów i flag inteligentnego czujnika, a także odczyt, interpretację i modyfikację pamięci EEPROM w celu rekalkibracji urządzenia.

5.2. Programy użytkowe

Na programy użytkowe składają się: baza danych i skrypty umożliwiające wizualizację wyników pomiarów na stronach WWW oraz program konsoli operatora.

5.2.1. Baza danych

Podstawowym elementem programów użytkowych jest baza danych działająca pod systemem operacyjnym Linux, zbierająca dane z jednej lub większej liczby central TINI. Wpisywanie danych do bazy realizowane jest poprzez program napisany w Javie, który interpretuje dane odbierane z central. Zestawy skryptów PHP umożliwiają elementarną wizualizację danych z bazy na stronach WWW, statystyki i rysowanie wykresów.

5.2.2. Konsola operatora

Konsola operatora jest programem napisanym w języku Java, korzystającym z bazy danych za pośrednictwem interfejsu JDBC. Umożliwia on monitorowanie stanu czujników i centrali, a także ręczne sterowanie wyjściami centrali za pomocą protokołu TCP/IP.

6. Bezpieczeństwo, niezawodność, szybkość działania

Przy projektowaniu oprogramowania czujników była brana pod uwagę norma europejska EN 50271 [8]. Samo urządzenie i obudowa spełniają wymogi polskich i europejskich norm EX. Czujniki zostały przetestowane pod kątem niezawodności i szybkości działania.

Centrala zbudowana na TINI w chwili pisania artykułu jest na etapie zaawansowanego projektu i kwestie bezpieczeństwa i niezawodności wymagają dalszych badań. Na podstawie przeprowadzonych testów wydajność pojedynczego sterownika TINI wydaje się być wystarczająca do monitorowania 15-30 czujników gazowych, przy średnim czasie obsługi 1 czujnika około 300-500ms.

7. Wnioski

W powyższym artykule przedstawiono oryginalny projekt systemu inteligentnych czujników z możliwością podłączenia do sieci Internet. System składa się z następujących elementów: inteligentnych czujników z interfejsem RS-485, centrali zbudowanej na komputerze TINI z możliwością podłączenia do sieci Ethernet oraz bazy danych i oprogramowania do obróbki i wizualizacji danych.

Inteligentne czujniki zostały zaprojektowane w sposób umożliwiający podłączenie różnych detektorów gazów, a dzięki elastycznemu i sparametryzowanemu procesowi przetwarzania danych możliwe jest wykorzystanie elementów o różnych charakterystykach. Czujniki wyposażone są w klasyczny interfejs analogowy 4-20mA oraz cyfrowy MODBUS RS-485. Dodatkowo istnieje możliwość komunikacji za pomocą łącza podczerwonego w standardzie IrDA. Za pomocą interfejsów cyfrowych możliwy jest odczyt danych, monitorowanie stanu, a także zmiana konfiguracji czujnika.

Centrala zbudowana z wykorzystaniem komputera TINI oferuje wiele funkcji niedostępnych prostym układom mikroprocesorowym. Dzięki wykorzystaniu maszyny wirtualnej Javy z wielozadaniowym systemem operacyjnym i zaimplementowanym protokołem TCP/IP, pisanie skomplikowanych programów wykorzystujących komunikację siecią jest bardzo proste. Jednocześnie możliwe jest pisanie niskopoziomowych procedur w języku asemblera procesora TINI (zgodny z rodziną 8051). Wadą TINI jest niewielka prędkość działania, wystarczająca jednak do obsługi systemu inteligentnych czujników.

System dopełnia baza danych umożliwiająca gromadzenie wyników pomiarów przesyłanych przez TINI oraz programy narzędziowe do diagnostyki i konfiguracji czujników.

Należy zauważyć, iż mimo swojej kompletności system zbudowany wyłącznie na centralach TINI nie zastąpi sterowników przemysłowych, które są stosowane w wysoko wydajnych systemach, wymagających szczególnej niezawodności i determinizmu czasowego.

LITERATURA

1. Świergot F.: Czujniki gazometryczne stosowane w polskim górnictwie. VI Konferencja Naukowa Czujniki optoelektroniczne i elektroniczne. Prace Komisji Metrologii Oddziału PAN w Katowicach, Seria: Konferencje Nr 2, Tom I, Gliwice 2000.
2. Legierski T. i in.: Programowanie sterowników PLC. Wyd. Prac. Komputerowej Jacka Skalmierskiego, Gliwice 1998.
3. Cupek R.: Protokół TCP/IP w systemach wizualizacji procesów przemysłowych. ZN Pol. Śl. Studia Informatica Vol. 22, No 3(45), Gliwice 2001.
4. Don Loomis: "The TINI Specification and Developer's Guide", Addison-Wesley, 2001.
5. Pietraszek T., Zawadiak F.: Zastosowanie sterownika z maszyną wirtualną Javy™ w projekcie „Inteligentny Dom”. ZN Pol. Śl. Studia Informatica Vol. 22, No 3(45), Gliwice 2001.
6. FIS Gas Sensor SB-95 for carbon monoxide and methane detection, 2002, http://www.fisinc.co.jp/PDF/SB10_SB95.pdf.

7. Rodzaje gazów wykrywanych przez czujniki CERGAZ,
<http://www.unima.com.pl/oferta/elzab/cerberus/cergas/gazy.htm>.
8. EN 50271 Electrical apparatus for the detection and measurement of combustible gases, toxic gases or oxygen – Requirements and test for apparatus using software an/or digital technologies.
9. iButton: Introducing TINI. Dallas Semiconductor Corp., 2001,
<http://www.ibutton.com/TINI/index.html>.
10. Java SDK Documentation. Sun Microsystems Inc., 2001,
<http://java.sun.com/j2se/1.3/docs.html>.

Recenzent: Dr inż. Włodzimierz Boroń

Wpłynęło do Redakcji 25 marca 2002 r.

Abstract

The design and implementation of monitoring systems are investigated in this paper. The author presents an idea of microprocessor based gas detector with digital communication capability. The principles of operation of semiconductor gas detectors and the signal extraction algorithms are also explained. The complexity of the algorithm and need of calibration require development of flexible and fully configurable microprocessor software.

The other part of the paper deals with design of control station based on low-cost TINI microcontroller with Java™ Virtual Machine. The TINI platform with built-in TCP/IP stack and a rich subset of Java API seems to be a good choice in non time-critical systems. The programmer can use various protocols such as: telnet, WWW, FTP and WAP using Java libraries as easily as it would be done on other Java platforms. On the other hand, some routines can be written in 8051 assembler and linked with Java code, for faster execution.

In the system described, TINI is used for data acquisition over MODBUS network, basic control and storing data in external database over the Internet.

Finally, the paper describes software used for data visualization, remote sensor monitoring, diagnosis and reconfiguration. Thanks to TINI, these processes can be done remotely via MODBUS/TCP protocol as well as locally, via MODBUS network.

Smart sensors can also be connected to industrial PLC, which would be better solution in industrial environments where high availability, fast responsiveness and integration with already existing systems are of primary concern.