

Szymon **KOSTRZEWSKI**  
Piotr **SZYNKARCZYK**

## ZASTOSOWANIE MAGISTRALI CAN W ROBOCIE INSPEKCYJNYM „SMR-100 EXPERT”

**Streszczenie:** W artykule umówiono implementację magistrali CAN w robocie „SMR-100 Expert” produkowanym w Przemysłowym Instytucie Automatyki i Pomiarów w Warszawie. Przedstawiono klasę robotów inspekcyjnych wraz z istotnymi dla zastosowania cechami, wymagania systemowe wewnętrznego rozproszonego systemu komunikacji, wybór magistrali polowej, cechy implementacji na poziomie sprzętu i bardziej szczegółowo z punktu widzenia programu. We wnioskach podsumowano cechy wdrożenia i przedstawiono plany modernizacji.

### 1. WPROWADZENIE

Istnieje wiele sytuacji niebezpiecznych, w których można zmniejszyć zagrożenie zdrowia i życia człowieka. Są to między innymi: zadania saperskie, inspekcja nieznanych i/lub niebezpiecznych obszarów (np. gruzowisk po trzęsieniach ziemi, zawalonych budynków), akcje policji i inne. W ostatnich latach znacznie wzrosło zagrożenie atakami terrorystycznymi i konieczne się stało opracowanie procedur umożliwiających zneutralizowanie ich skutków. Te oraz inne niewymienione przypadki stanowią przykład zastosowań robotów inspekcyjnych. Przed robotami inspekcyjnymi stawia się szereg zadań, jednak ich głównym celem jest zastępowanie człowieka w ogólnie pojętych sytuacjach niebezpiecznych. Wynika to z podstawowego aksjomatu techniki, że życie ludzkie stanowi wartość najwyższą i zawsze lepiej jest poświęcić maszynę niż człowieka. Najczęstsze zastosowania robotów inspekcyjnych to przede wszystkim:

- inspekcja, przenoszenie i neutralizacja ładunków niebezpiecznych,
- zakładanie i odpalanie ładunków niszczących obiekty niebezpieczne,
- wspomaganie operacji antyterrorystycznych,
- badanie i przeszukiwanie terenu,
- praca w warunkach szkodliwych lub niebezpiecznych dla człowieka,
- ochrona i inspekcja obiektów,
- dostarczanie narzędzi (aparaty rentgenowskie, wyrzutniki pocisków, czujniki promieniowania itp.) i wykorzystanie ich do pracy w miejscach niebezpiecznych.

Historia polskich robotów pirotechnicznych rozpoczęła się w roku 1999, kiedy powstał prototyp robota Inspector. Od roku 2000 roboty te są sukcesywnie wprowadzane na wyposażenie oddziałów pirotechnicznych polskiej policji. Sukces robotów klasy Inspector zachęcił konstruktorów PIAP do tego, aby bazując na dotychczasowych doświadczeniach, w ramach nowego projektu celowego KBN podjąć się budowy i wdrożenia nowego robota oznaczonego symbolem „SMR-100 Expert” (por. [1]).

Robot „SMR-100 Expert”, pokazany na rysunku 1 został zaprojektowany tak, aby móc skutecznie zastępować człowieka przy rozpoznawaniu i neutralizacji terrorystycznych ładunków wybuchowych. W szczególności docelowym obszarem działań robota są ciasne pomieszczenia i środki komunikacji, takie jak samoloty, autobusy i wagony kolejowe.

Robot składa się z:

- bazy mobilnej poruszającej się na gąsienicach,
- przedniej gąsienicy ułatwiającej pokonywanie przeszkód terenowych,
- dwóch stabilizatorów bocznych wspomagających pracę przy wysuniętym ramieniu,



**Rys. 1. Robot „SMR-100 Expert” produkcji PIAP**

- manipulatora o sześciu stopniach swobody i zacisku szczęk chwytaka,
- kamery głównej o dwóch stopniach swobody oraz funkcjach powiększania i ogniskowania,
- sprzętu dodatkowego:
- pięciu kamer z oświetlaczami,
- obsługi uzbrojenia,
- czujników temperatury.

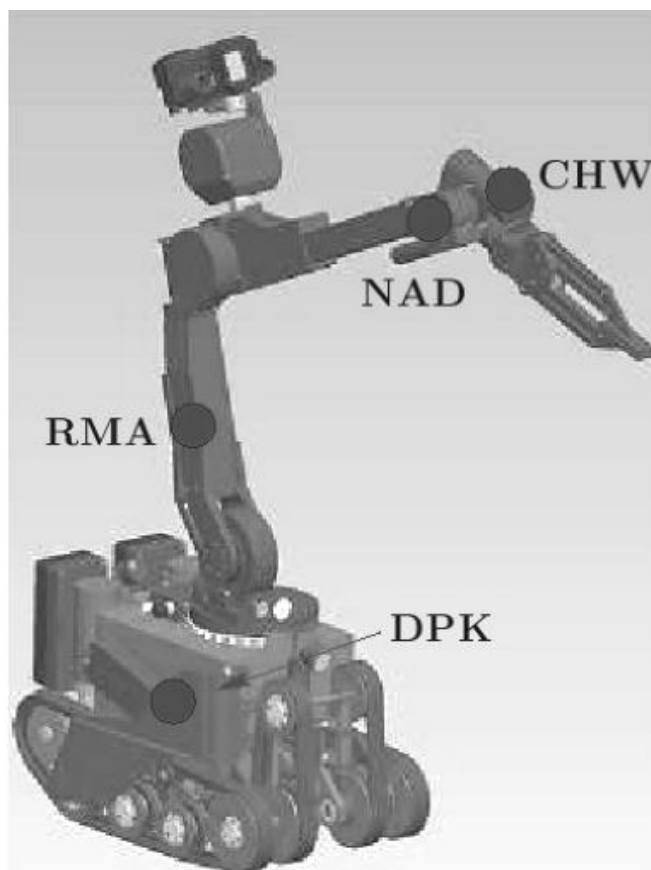
Liczba sterowanego sprzętu stawia duże wymagania przed systemem komunikacji mającym za zadanie działać w czasie rzeczywistym na wysokim poziomie bezpieczeństwa. W dalszej części artykułu zostanie przedstawiony przegląd wymagań konstrukcyjnych i praktyczne implementacje systemu.

## 2. WYMAGANIA SYSTEMOWE I WYBÓR MAGISTRALI POŁOWEJ

Robot Expert jest sterowany w systemie rozproszonym (wg [2]). Na rysunku 2 przedstawiono rozmieszczenie elektronicznych układów sterujących robotem. Poszczególne moduły dzielą się zadaniami sterowania. Moduł komputera pokładowego DPK jest m.in odpowiedzialny za komunikację z pulpitem operatora, publikowanie i odbieranie rozkazów pulpitu dotyczących sterowań: bazy mobilnej, przedniej gąsienicy, stabilizatorów bocznych i obrotu wieży oraz pomiary położeń i krańcówek dla ww. par kinematycznych.

Tab. 1. Oszacowanie transmisji wewnętrznej pomiędzy modułami robota Expert

Nazwa węzła		Ilość bitów	Częstotliwość [Hz]	
DPK	Komputer pokładowy	128		12800
RMA	Moduł ramienia	64		6400
NAD	Moduł nadgarstka	32	100	3200
CHW	Moduł chwytaka	48		4800
GKV	Moduł kamery	48		4800
		Transfer [b/s] 32000		



**Rys. 2. Rozmieszczenie elektronicznych układów sterujących w robocie Expert**

Umieszczony w ramieniu dolnym moduł RMA wykonuje sterowania dla ramienia dolnego, ramienia górnego i wysuwu. Moduł nadgarstka NAD wykonuje sterowania obrotu nadgarstka, a moduł chwytaka CHW steruje obrotem i zaciskiem szczęk chwytaka oraz mierzy siłę zacisku. Na wszystkich płytkach wykonywane są pomiary temperatury. Wyżej wymienione wartości muszą być przesyłane w sieci wewnętrznej łączącej układy sterowania. Pozwala to wykonać oszacowanie transferu danych przedstawione w tabeli 1. Do innych wymagań stawianych systemowi komunikacji należy zaliczyć:

- oszczędność liczby potrzebnych przewodów; ze względów konstrukcyjnych bardzo trudne jest przeprowadzenie przewodów wewnątrz ramienia,
- łatwe rozszerzanie systemu o dodatkowe moduły w przyszłości,
- dużą odporność na szumy elektromagnetyczne,
- dobre zabezpieczenie danych i detekcję błędów transmisji.
- Magistrala CAN spełnia powyższe wymagania. Jej podstawowe parametry to (wg [3]):
- transfer: 1 [Mbit/s] dla przewodów krótszych niż 40 [m],
- maksymalna ilość dołączonych stacji: 64,
- 8-bajtowa ramka danych.

Dodatkową zaletą w tej aplikacji jest to, że magistrala jest w użyciu od wielu lat oraz jest standaryzowana i dobrze udokumentowana. Na rynku jest dostępnych wiele mikroprocesorów z wbudowaną obsługą tej magistrali na poziomie warstwy danych i z wsparciem technicznym. Z tych względów zdecydowano się na implementację protokołu CAN.

### 3. IMPLEMENTACJA MAGISTRALI CAN W ROBOCIE „SMR-100 EXPERT”

#### 3.1 Implementacja sprzętowa

Do sterowania w systemie rozproszonym zastosowano mikrokontrolery Hitachi z rodziny H8S/2600 (por. [4]). Posiadają one architekturę 16-bitową z 12 [kB] pamięci RAM oraz wbudowana pamięć FLASH o pojemności 128 [kB]. Ich istotną zaletą jest fakt, że obsługują protokoły transmisji RS-232, I2C oraz CAN w implementacji Hitachi nazywany (zgodnie z dokumentacją firmy) HCAN.

Fizycznie HCAN jest wyprowadzony na dwa piny mikrokontrolera: HTxD0 (pin 97) i HRxD0 (pin 98). Do konfiguracji transmisji służą rejestry wewnętrzne o następujących funkcjach:

- rejestry przerw: wskazują stan źródeł przerw,
  - rejestry konfiguracyjne:
    - transmisji bitowej CAN i parametrów czasowych,
    - ustalenie funkcji buforów na wysyłanie/odbior,
  - rejestry sterujące
    - głównym interfejsem,
    - na poziomie poszczególnych paczek i buforów,
  - rejestry statusu: sygnalizują stan transmisji,
- Oprócz ww. są niżej wymienione jeszcze dostępne rejestry specjalistyczne:
- wysyłania:
    - oczekiwania na wysłanie,
    - sygnalizujące, że wysłanie zakończono powodzeniem,
  - odbioru:
    - otrzymania wiadomości.

Transmisja odbywa się w następujący sposób. Na początku należy wyprowadzić system komunikacji ze stanu oczekiwania i wyzerować bit odpowiedzialny za transmisję. W kolejnym kroku należy ustalić parametry komunikacji: prędkość transmisji i wybrać bufor. Następnie należy zainicjować bufor transmisji w pamięci RAM i ustalić priorytet. W tym kroku następuje przejście z trybu konfiguracji do trybu transmisji, w którym należy zdefiniować dane do wysłania, które następnie należy umieścić we wcześniej skonfigurowanym buforze. Paczka zostaje wysłana z poziomu sprzętowego. O przebiegu transmisji, jak również o tym, czy została ona zakończona powodzeniem informują odpowiednie rejestry.

Odbiór odbywa się bardzo podobnie do przedstawionego powyżej schematu. Po skonfigurowaniu buforów i przejściu do stanu oczekiwania na wiadomość aktywowany jest filtr wiadomości. Odebrana paczka jest automatycznie mapowana na wewnętrzną pamięć RAM.

Omówiony sposób transmisji jest zaimplementowany na poziomie warstwy pierwszej i drugiej transmisji. Jest to rozwiązanie wygodne i bezpieczne, ponieważ zostało gruntownie przetestowane przez producenta i użytkowników oraz znacząco przyspiesza proces projektowania układów.

#### 3.2 Implementacja programowa

Wszystkie rejestry wymienione w punkcie 3.1 są dostępne z poziomu programu (por. [5]). Aby móc się do nich odwoływać, w kodzie programu należy w kompilatorze zgodnym z ANSI C dla procesorów z rodziny H8S/2600 dołączyć bibliotekę specyficzną dla danego procesora (np. dla H8S/2625 jest ona reprezentowana przez plik nagłówkowy „2625S.h”)

a dostarczaną przez producenta. W odniesieniach do dokumentacji można znaleźć wszystkie nazwy zmiennych, jak również dozwolone wartości konfiguracyjne i sterujące.

Konfiguracja mikrokontrolera o tak dużej ilości funkcji jest trudna, wymaga wysokich kwalifikacji, jest czasochłonna i kłopotliwa w późniejszej modyfikacji. Na rynku są dostępne produkty automatyzujące ten proces i gwarantujące integrację z kodem programu. Z tych powodów w implementacji zastosowano produkt *IAR MakeApp for Hitach H8*. Program ten umożliwia inicjalizację komunikacji, kontrolę run-time, obsługę przerw i SFR (ang. *Special Function Registers*). Komunikacja z użytkownikiem odbywa się za pomocą interfejsu graficznego oraz kreatorów. Po podaniu odpowiednich danych i wyborze funkcji program generuje bibliotekę konfiguracyjną, którą należy dołączyć do programu. Dodatkowo jest również generowany raport i dokumentacja funkcji ułatwiająca korzystanie z biblioteki oraz kontrolę wersji.

Warstwę obiektową magistrali CAN zaimplementowano w programie w postaci dołączanej biblioteki o nazwie *hcan*. Jest ona napisana w języku ANSI C w ten sposób, że może być dołączana do każdego z używanych w PIAP mikrokontrolerów, gdyż w warstwie komunikacyjnej opiera się na kodzie wygenerowanym przez *MakeApp*. W ten sposób przenoszenie kodu programu jest niemal bezpośrednie i wymaga jedynie wprowadzenia nowej konfiguracji do *MakeApp* co skraca proces z tygodni do godzin.

Biblioteka *hcan* dostarcza użytkownikowi zbioru funkcji wysokiego poziomu do zarządzania komunikacją CAN dla mikrokontrolera. Wartości parametrów transmisji są w używanych mikrokontrolerach ze względów praktycznych ujednolicone i zachowane w kodzie wygenerowanym przez *MakeApp*. Gdyby zaszła potrzeba zmiany tych parametrów należy odwołując się do dokumentacji ponownie wygenerować i skompilować kod warstwy transmisji. W pierwszych krokach po uruchomieniu programu powinno się zainicjować transmisję. W tym celu należy wywołać funkcje `InitTxMailbox` i `InitRxMailbox`. Zapisują one wartości zdefiniowane w *MakeApp* do rejestrów, zwracając potwierdzenie prawidłowego działania buforów lub kody błędów. W celu publikowania wartości na CAN należy wywołać funkcję `SendHcan` z parametrami: wskaźnikiem do danych do wysłania, długością struktury i wartością ID paczki. Należy pamiętać, że ze względu na 8-bitowe paczki w standardzie wysyłane dane nie powinny przekraczać tej wartości. Długość struktury jest prosta do obliczenia, natomiast wartość ID odnosi się bezpośrednio do CAN. Ponieważ w definicji tej magistrali nie ma funkcji nadzoru jako pierwsze są transmitowane paczki z wyższym priorytetem. Każdemu typowi przypisywana jest unikatowa wartość ID będąca liczbą 16-bitową zgodnie z zasadą, że im mniejsza wartość, tym wyższy priorytet. Pozwala to na przyporządkowanie wagi przesyłanym strukturom (np. wartości temperatury płytek są zdecydowanie mniej istotne niż obsługa uzbrojenia) oraz umożliwia odbiornikom rozpoznanie przychodzących paczek. Odbiorem paczek zajmuje się funkcja `RM0_InterruptHandler`. Jest ona automatycznie wywoływana za każdym razem, gdy przychodzi przerwanie od odbiornika CAN. Funkcja podczas jednego przebiegu przyporządkowuje otrzymaną paczkę do kategorii na podstawie wartości ID i mapuje otrzymane dane na wewnętrzną strukturę programu (zmienne globalne). W bibliotece znajdują się również rzadziej używane funkcje przydatne do odnajdywania błędów i testów.

Struktura programu dla tak zdefiniowanej biblioteki jest jednorodna na wszystkich płytkach współpracujących ze sobą w sieci. Wymieniane dane są następującej postaci:

- wartości z czujników
  - położenia,
  - temperatury,
  - prądu,
  - siły,
  - wyłączników krańcowych.

- wartości zadane:
  - położenia,
  - prędkości, prądu,
- Przełączniki:
  - wybór kamery,
  - włączenie/wyłączenie oświetlaczy,
- Wartości z interfejsu wysokiego poziomu:
  - sterowanie w kartezjańskim układzie współrzędnych,
  - włączanie/wyłączanie stref bezpieczeństwa,
  - wybór pozycji do automatycznego pozycjonowania
- Wartości serwisowe i kalibracyjne.

Na każdym mikrokontrolerze funkcjonują te same struktury danych przechowujące wyżej wymienione dane w postaci pól bitowych pozwalających na bezpośredni dostęp do danych bez potrzeby maskowania i przesuwania bitowego. W pętli głównej programu jest wywoływana funkcja `SendHcan` publikując dane z pól bitowych na CAN. Częstotliwość wywołania jest definiowana przez przerwanie systemowe od timera i można ją łatwo definiować dla każdego typu danych. Jednocześnie „w tle” (tzn. spoza pętli głównej) po przyjsciu każdej paczki funkcja `RM0_InterruptHandler` automatycznie mapuje przychodzące dane na wewnętrzną strukturę danych zgodnie z wartością `ID`.

W opisany wyżej sposób na każdym węźle sieci uzyskuje się dostęp do wartości struktur z wszystkich innych węzłów. W razie potrzeby każdy węzeł może spełniać funkcję *master* w warstwie aplikacji magistrali. Dodanie nowego węzła do sieci wymaga jedynie dopisania jego struktury i skompilowania kodu dla wszystkich współpracujących węzłów co zwiększa modułowość rozwiązania. Dodatkowo fizycznie przewód CAN jest wyprowadzony na zewnątrz poprzez gniazdo z robota, co pozwala na łatwe testowanie i umożliwia sterowanie za pomocą komputera (z pośrednictwem wbudowanej karty CAN lub np. zestawu testowego zaprogramowanego do działania jako *switch* pomiędzy RS-232 a CAN).

#### 4. PODSUMOWANIE

Magistrala CAN została z powodzeniem zaimplementowana w robocie inspekcyjnym „SMR-100 Expert”. Możliwości fizyczne sieci, takie jak transfer i ilość węzłów są wystarczające dla zastosowania oraz pozwalają na dalsze rozszerzanie protokołu komunikacji. Cechy warstwy fizycznej CAN, takie jak duże odporność na błędy, zabezpieczenia i fakt, że do transmisji potrzebne są tylko dwa przewody znacząco wpływają na jakość i bezawaryjność konstrukcji.

W perspektywie znajdują się następujące ulepszenia działania robota z wykorzystaniem możliwości sieci CAN:

- dodanie transmisji mowy przesyłanej w postaci cyfrowej,
- zwiększenie częstotliwości przesyłania danych pomiarowych i sterowań,
- dodanie możliwości programowania mikrokontrolerów poprzez CAN.

Nowoczesne metody kodowania mowy pozwalają na obniżenie transferu danych do wartości 16 [kbit/s]. W celu zwiększenia dokładności danych z pomiarów oraz zmniejszenia opóźnienia w układach sterowania planuje się znaczne zwiększenie częstotliwości przesyłanych danych. Dokładne wartości transferów nie są jeszcze znane, jednak dotychczasowe rezultaty sugerują, że przy implementacji nie należy się spodziewać przekroczenia wartości dopuszczalnych. Inną bardzo ciekawą możliwością jest programowanie wszystkich

mikrokontrolerów w sieci przez jedno złącze CAN. Jest to możliwe ze względu na fakt, że obsługa protokołu jest zaimplementowana na poziomie sprzętowym oraz wymaga napisania programu typu *kernel* kopiującego skompilowany program do pamięci FLASH. Znacznie ułatwiłoby to proces programowania i przyspieszyło wszelkie zmiany ze względu na fakt, że często dostęp do płyt sterujących jest utrudniony.

Dotychczasowe doświadczenia świadczą o tym, że wybór sieci CAN do połączenia modułów sterowania rozproszonego w robocie „SMR-100 Expert” był właściwy i dobrze prognozują dalsze implementacje w przyszłości.

## 5. LITERATURA

- [1] SZYNKARCZYK P. i in.: Robot antyterrorystyczny – wybrane zagadnienia projektowe, XV Krajowa Konferencja Automatyki, Warszawa 27-30 czerwca 2005.
- [2] KOSTRZEWSKI S.: Wykonanie projektu i wdrożenie nowego systemu sterowania dla robota „SMR-1 00 Expert”, Praca magisterska, Politechnika Warszawska, Warszawa 2006.
- [3] MAHALIK P.N.: Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control, Springer-Verlag, Berlin Heidelberg 2003.
- [4] Renesas H8S Family Application Note, Rev. 1 z 22 lipca 2005.
- [5] Reneses H8S Family Software Manual, Rev. 4.0 z 24 lutego 2006.

## PRACTICAL IMPLEMENTATION OF CANBUS FOR PYROTECHNICAL ROBOT ``SMR-100 EXPERT”

**Abstract:** this article presents the implementation of CANbus in robot „SMR-100 Expert” produced by The Industrial Research Institute for Automation and Measurements in Warsaw, Poland. It describes the class of pyrotechnical robots with focus on the features influencing the practical application of the communication system. System assumptions are presented with fieldbus selection. Main implementation ideas are described on both hardware and software level. Conclusion sums up current works and briefly shows further improvements.

Recenzent: dr hab. inż. Gabriel Kost, prof. Pol. Śl.