

Jan WEREWKA

## KONSTRUKCJA REAKTYWNYCH ROZPROSZONYCH SYSTEMÓW OBŚLUGI ZDARZEŃ BAZUJĄCYCH NA MAGISTRALI CAN NA PRZYKŁADZIE NADRZĘDNEGO SYSTEMU KONTROLNO- DIAGNOSTYCZNEGO

**Streszczenie:** W opracowaniu przedstawiono propozycję rozwiązania dla systemów reaktywnych potraktowanych jako systemy zdarzeń. Systemy reaktywne stanowią szeroką klasę systemów, które związane są z pewnym środowiskiem w ten sposób, że muszą cały czas reagować na zmiany w nim zachodzące. Ponieważ zwykle systemy zdarzeń rozlokowane są na pewnym obszarze, to przedstawione rozwiązania będą związane z tworzeniem architektur systemów rozproszonych.

W opracowaniu przedstawiono metodę konstrukcji nadrzędnego systemu kontrolno-diagnostycznego. Przedyskutowano zagadnienia związane ze spełnieniem ograniczeń czasu rzeczywistego.

Opisano możliwości wykorzystania różnych mechanizmów komunikacyjnych sieci CAN przydatnych przy budowie systemów czasu rzeczywistego.

Prezentowano przykładową implementację takiego systemu kontrolno-diagnostycznego (NSKD) bazującego na magistrali CAN przeznaczonego dla nadzoru poprawności działania systemu sterującego akceleratorem.

**Słowa kluczowe:** Systemy czasu rzeczywistego, systemy reaktywne, systemy rozproszone, magistrale miejscowe, analiza czasowa, CAN, akcelerator.

### 1. WSTĘP

Wiele systemów rzeczywistych można potraktować jako systemy zdarzeń. Systemy takie charakteryzują się zmianą stanu (wystąpieniem zdarzenia), która zazwyczaj dokonywana jest w różnych, trudnych do przewidzenia chwilach czasu. Dla tych systemów opracowywane są różne modele i systemy informatyczne wspomagające obsługę zdarzeń.

W wielu istniejących rzeczywistych systemach powstają zdarzenia, które wygodnie jest obsłużyć poprzez zastosowanie rozproszonych systemów komputerowych. W celu obsługi zdarzeń proponowane są różne architektury systemów komputerowych obsługujące zdarzenia. Ponieważ systemy zdarzeń zwykle rozlokowane są na pewnym obszarze, który może być bardzo obszerny, to przedstawione rozwiązania będą związane z tworzeniem architektur systemów rozproszonych. Dodatkowo wiele takich systemów komputerowych musi w odpowiedni sposób reagować na zdarzenia zachodzące w systemie, dlatego mówi się o systemach reaktywnych. Uwzględnienie dodatkowych wymagań dostarczenia odpowiedzi w zadanym okresie czasu powoduje, że rozpatrywany system jest systemem czasu rzeczywistego. W pracy podjęto próbę dochodzenia do rozwiązania z uwzględnieniem metodyk proponowanych w literaturze.

W wielu rozproszonych systemach komputerowych czasu rzeczywistego stosowane są z powodzeniem magistrale miejscowe [12]. Dla analizy i optymalizacji takich systemów mogą być użyte różne metody teoretyczne [14]. Do badania charakterystyk czasowych magistral miejscowych mogą być stosowane metody analizy czasowej [13]. Dla magistrali CAN, ze względu na jej własności, mogą być stosowane metody analizy typowe dla systemów scentralizowanych (np. systemów operacyjnych).

Interesującym zagadnieniem jest sprawdzenie proponowanej metodyki dla systemu rzeczywistego. Jako system rzeczywisty wybrano nadrzędny system kontrolno-diagnostyczny przeznaczony do nadzoru komputerów sterujących akceleratorem.

## 2. SYSTEMY OBSŁUGI ZDARZEŃ

Systemy bazujące na zdarzeniach muszą mieć możliwość wykrywania zdarzeń i powiadamiania o nich. Tradycyjne systemy zdarzeń bazują na źródłach zdarzeń i odbiornikach zdarzeń (odbiorcach informacji o zdarzeniach).

Środowisko obsługujące zdarzenia (ang. event framework) realizuje dwie podstawowe funkcje:

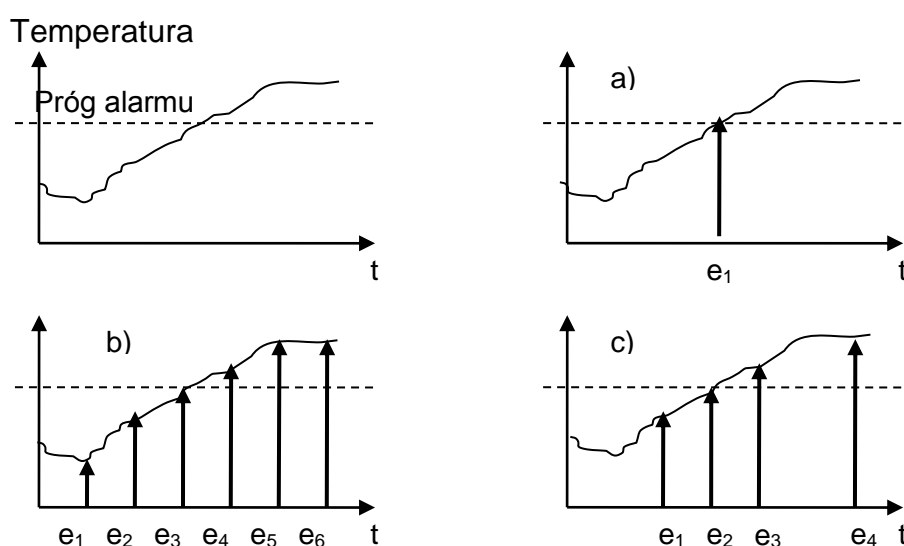
- wykrywanie zdarzenia, które przejawia się w wykryciu zdarzenia będącego przedmiotem zainteresowania.
- powiadomienie o zdarzeniu wszystkich, którzy są zainteresowani informacją o wystąpieniu zdarzenia

Zdarzenie jest reprezentowane poprzez dyskretną zmianę stanu. Zmiana ta może być wywołana przyjęciem przez system pewnego stanu ze zbioru stanów dyskretnych lub przekroczeniem pewnych wartości dla systemów ciągłych.

Przykładami zdarzeń mogą być: zakończone sukcesem zarejestrowanie się do systemu, wykrycie sprzętu, przekroczenie lub osiągnięcie poziomów alarmowych. Rys. 1 ilustruje różne sposoby generowania zdarzeń dla przypadku przekroczenia temperatury w systemie ponad zadany stan alarmowy. Na rysunku przedstawiono trzy sposoby wyzwalania zdarzeń.

- Metoda oczekiwania – zdarzenie generowane jest tylko wtedy, gdy nastąpi przekroczenie progu alarmowego.
- Metoda odpytań – zdarzenie generowane jest w równych wystarczająco krótkich odstępach czasu.
- Metoda mieszana - zdarzenie generowane jest wtedy, gdy nastąpi przekroczenie progu alarmowego oraz w równych wystarczająco długich odstępach czasu, wystarczających do stwierdzenia niesprawności systemu.

Wydawać by się mogło, że najlepszą metodą jest metoda pierwsza, gdyż występuje powiadomienie o zdarzeniu tylko wtedy, gdy ono nastąpiło. Wadą tej metody jest to, że podsystem, który melduje o zdarzeniu może nie być sprawny, to znaczy nie wysła żadnych wiadomości. Prosta metodą, ale generującą stosunkowo duży ruch w systemie jest metoda odpytań.



Rys.1. Ilustracja trzech metod uzyskiwania informacji o zdarzeniu (a - oczekiwania, b - odpytań i c - mieszana)

Systemy przeznaczone do realizowania obsługi zdarzeń ogólnie składają się z:

- źródeł zdarzeń (ang. event sources)
- odbiorców zdarzeń (ang. event listener)
- usług powiadamiania (ang. notification services)
- usług filtracji zdarzeń,
- usług magazynowania i buforowania

Systemy obsługujące zdarzenia mogą być budowane na różne sposoby. Jednym z najbardziej rozpowszechnionych wzorców architektonicznych przeznaczonych do obsługi takich systemów jest model publicysta/subskrybent (ang. publisher/subscriber). W systemach zdarzeń występują źródła zdarzeń, które wykrywają zdarzenia i są dostawcami informacji o zdarzeniach. Źródło to w przedstawionym modelu jest publicystą zdarzenia (ang. publisher). Publicysta wykonuje pewną czynność polegającą na opublikowaniu zdarzenia (ang. publish). Po wystąpieniu zdarzenia odbiorca zdarzenia może się dowiedzieć o zdarzeniu poprzez zastosowanie jednego z mechanizmów powiadamiania:

- push – opublikowanie informacji o zdarzeniu poprzez wysłanie wiadomości; w tym przypadku dostawca informacji o zdarzeniu jest stroną aktywną,
- pull – konsument (odbiorca) informacji pobiera od dostawcy wiadomość o zdarzeniu; dostawca jest w tym przypadku stroną bierną, a odbiorca jest stroną aktywną.

Odbiorca zdarzenia jest konsumentem zdarzenia wyposażonym zwykle w pewną aplikację przeznaczoną do obsługi zdarzeń.

### 3. SYSTEMY REAKTYWNE I SYSTEMY CZASU RZECZYWISTEGO

Przedstawione rozważania dotyczą klasy systemów zwanych reaktywnymi. Systemy reaktywne stanowią szeroką klasę systemów, które związane są z pewnym środowiskiem w ten sposób, że muszą cały czas reagować na zmiany w nim zachodzące. Środowisko jest rozumiane jako sam system reaktywny (środowisko wewnętrzne), jak i otoczenie (środowisko zewnętrzne). W systemach reaktywnych decydującą rolę odgrywa środowisko zewnętrzne, czyli otoczenie.

Bazując na opracowaniu [15], można powiedzieć, że systemy reaktywne charakteryzują się:

- wysoką interaktywnością; system reaktywny w sposób nieprzerwany reaguje na zmiany w otoczeniu poprzez interakcje z nim;
- ciągłością działania; procesy realizujące interakcje z otoczeniem są procesami o ciągłym funkcjonowaniu - przerwanie funkcjonowania systemu oznaczałoby jego awarię (ang. failure);
- zorientowaniem na przerwania (ang. interrupt driven); system musi odpowiadać na impulsy (ang. stimuli), gdy są one w otoczeniu i wtedy, gdy one powstają;
- reakcją zależną od stanu systemu (ang. state-dependent response); odpowiedź systemu zależy od stanu systemu i od zdarzeń zewnętrznych; odpowiedź taka może spowodować zmianę stanu systemu;
- reakcjami zorientowanymi na środowisko; reakcje ze środowiskiem polegają na umożliwieniu, wymuszeniu lub zapobieganiu pewnemu zachowaniu lub komunikacji w jego środowisku;
- równoległością działania; procesy w systemach reaktywnych dokonują interakcji z otoczeniem realizowanej w sposób równoległy;

- realizacja aktywności krytycznych ze względu na czas; bardzo często systemy reaktywne działają przy ograniczeniach nakładanych na czas reakcji systemu oraz definiowane są inne wymagania związane z czasem.

Inne ważne pojęcie dotyczy systemów czasu rzeczywistego. Zgodnie ze standardem IEEE/ANSI [6] system czasu rzeczywistego jest to system komputerowy, w którym obliczenia są wykonywane współbieżnie z procesem zewnętrznym (otoczenie) w celu sterowania, nadzorowania lub terminowego reagowania na zdarzenia występujące w tym procesie (otoczeniu).

W definicji tej podano dwa procesy. Jeden wyznaczony przez obliczenia realizowane w systemie komputerowym oraz drugi zachodzący w otoczeniu, które dostarcza temu pierwszemu informacji wejściowych (zdarzenia) i jest obszarem, gdzie kierowane są wyniki obliczeń (reakcje na zdarzenia). Ogólna definicja systemu czasu rzeczywistego jest uzupełniana przez sformułowanie cech charakterystycznych [11, 14].

1. **Ciągłość działania.** System czasu rzeczywistego winien pracować bez przerw.
2. **Zależność od otoczenia.** System czasu rzeczywistego musi być rozpatrywany w kontekście otoczenia. Oznacza to, że obliczenia tego systemu są uzależnione od zdarzeń i danych generowanych przez proces zewnętrzny.
3. **Współbieżność.** Otoczenie systemu składa się z wielu obiektów (podsystemów), które działają współbieżnie, generując odpowiednie zdarzenia lub dane wymagające obsługi przez system czasu rzeczywistego.
4. **Przewidywalność.** Zdarzenia i dane generowane przez otoczenie pojawiają się zazwyczaj w momentach przypadkowych, a współbieżne wystąpienie zdarzeń może wymagać ich jednoczesnej obsługi, co w sytuacji ograniczonych zasobów prowadzi do współzawodnictwa o uzyskanie dostępu. Współbieżna struktura systemu indukuje niedeterministyczne zachowanie, lecz z drugiej strony wymaga się ścisłego powiązania między zdarzeniami a odpowiadającymi im obsługami, realizowanymi przez system. Stąd system, mimo że złożony jest zazwyczaj z szeregu procesów współbieżnych (struktura wewnętrzna), na zewnątrz musi zachowywać się deterministycznie, tzn. reagować na zdarzenia wg założonych wymagań.
5. **Punktualność.** Odpowiedzi systemu (reakcje na zdarzenia) winny być obliczane zgodnie z zaprojektowanymi algorytmami i dostarczane do otoczenia w odpowiednich momentach czasowych. Natura otoczenia, tzn. brak możliwości zatrzymania procesu zewnętrznego stawia dodatkowe wymagania, by oprócz poprawności uzyskanych wyników, moment ich przekazania do otoczenia spełniał określone wymagania. Wyniki dostarczone zbyt późno (niekiedy również zbyt wcześnie) mogą prowadzić do niepoprawnego działania lub być nawet przyczyną upadku systemu.

Dla systemów czasu rzeczywistego przeprowadza się analizę czasu odpowiedzi systemu. Jedną z bardziej podstawowych metod analizy bazuje na **szeregowaniu zadań ze statycznym przydziałem priorytetów**. Metoda RMS (Rate Monotonic Scheduling) polega na szeregowaniu monotonicznym ze względu na natężenie przybywania zadań do wykonania. Polega ona na przydzieleniu najwyższego priorytetu do zadania o największym natężeniu (o najmniejszym okresie) występowania. Pozostałe priorytety będą nadawane do zadań w sposób monotoniczny do ich natężenia występowania. W literaturze często używany jest skrót RMA (ang. Rate Monotonic Analysis), który dotyczy analizy dokonywanej dla systemów bazujących na szeregowaniu monotonicznym.

Dla metody RMS przyjmuje się następujące założenia:

- każde z zadań ma przydzielony priorytet;
- przydzielenie priorytetów jest zgodne z regułą - im krótszy okres występowania zadania, tym większy priorytet;
- zadanie o wyższym priorytecie może wyłuszczyć zadanie o priorytecie niższym.

W literaturze cytowane jest podstawowe **twierdzenie** dotyczące metody RMS [7].

Przyjmuje się zbiór  $\tau = \{\tau_1, \dots, \tau_n\}$  niezależnych okresowych zadań posiadających czasy wykonania  $c_i$  oraz okresy występowania  $t_i$  takie, że ograniczenie czasowe  $d_i$  równe jest okresowi występowania zadania ( $d_i = t_i$ ). Jeżeli dla zadań z tego zbioru stosowany jest algorytm szeregowania RMS i spełniony jest warunek:

$$\sum_{i=1}^n \frac{c_i}{t_i} \leq n(2^{1/n} - 1)$$

to są spełnione ograniczenia czasowe dla wszystkich zadań z tego zbioru.

#### 4. ANALIZA CZASOWA DLA MAGISTRALI CAN

Przy zastosowaniu magistral w systemach sterowania, jednym z najbardziej istotnych zagadnień analizy czasowej jest zasada dostępu do warstwy łącza danych. Jedną z bardziej popularnych metod dostępu do warstwy fizycznej stanowi metoda master/slave stosowaną w wielu magistralach. W rozważanych zastosowaniach wykorzystano magistralę CAN, w której dostęp do łącza danych wyznaczony jest poprzez metodę dominacji bitowej.

- W metodzie master/slave master wysyła wiadomość z żądaniem podania stanu przez węzeł slave i czeka na odpowiedź od tego węzła. Po otrzymaniu odpowiedzi przechodzi do następnego węzła.
- W metodzie dominacji bitowej węzeł master wysyła jedną wiadomość z żądaniem podania stanu do wszystkich stacji. Na takie żądanie wszystkie stacje odpowiadają, lecz tylko jednej stacji udaje się przekazać wiadomość. Po wysłaniu wiadomości przez tę stację ponownie wszystkie stacje poza stacją, które już wysłały swoje wiadomości, próbują ponownie wysłać wiadomość.

Analiza czasowa dla magistral miejscowych jest rozpatrywana w odniesieniu do szeregowania wiadomości. Najistotniejszą rolę z punktu widzenia szeregowania wiadomości dla magistral miejscowych odgrywa warstwa MAC (ang. Medium Access Control) modelu OSI/ISO. Warstwa ta określa, na jakich zasadach poszczególne węzły systemu uzyskują dostęp do medium transmisyjnego. Ze względu na rodzaj dostępu do medium transmisyjnego sieć CAN może być potraktowana jak scentralizowany system szeregowania zadań (szeregowania wiadomości) [2], dla którego może być stosowane twierdzenie dotyczące metody RMS z uwzględnieniem czasów blokowania.

Istniejące mechanizmy sterowania przepływem wiadomości (tj. szeregowaniem wiadomości) dla systemów czasu rzeczywistego można podzielić na następujące kategorie [9]:

- z przydziałem priorytetów (ang. priority-driven) – są to zarówno algorytmy ze statycznym, dynamicznym, jak i mieszanym przydziałem priorytetów.
- z przydziałem czasu (ang. time-driven) np. FTT-CAN (ang. Flexible Time Triggered communication on CAN) [8, 9].
- z przydziałem zasobów (ang. share-driven).

Dla sieci CAN najbardziej naturalnym mechanizmem szeregowania wiadomości jest mechanizm oparty na przydziale priorytetów (tak jak dla systemów scentralizowanych). CANopen [Can00] dla warstwy aplikacji definiuje dwa tryby transmisji danych o zupełnie

różnych charakterystykach, spełniających wymagania systemów automatyki w pełnym zakresie zastosowań. Należą do nich obiekty serwisowe SDO i obiekty danych PDO.

Transfer obiektów SDO jest transferem acyklicznym i o niskim priorytecie, stosowanym głównie do konfiguracji urządzeń sieci CANopen. Dla obiektów SDO stosowany jest zwykle model komunikacji klient-serwer. W takim przypadku sterowanie przepływem wiadomości w sieci inicjowane jest zawsze przez węzeł master, który odgrywa rolę klienta. Natomiast węzeł slave jest dostarczycielem informacji (usługi), a zatem pełni rolę serwera.

Obiekty PDO używane są w celu przesyłania danych krytycznych czasowo. Format obiektu PDO, jak również jego identyfikator może być dynamicznie określony przy użyciu obiektów SDO. Obiekty PDO nie zawierają naddatku (ang. overhead), gdyż jednemu obiektowi PDO odpowiada jedna ramka danych, przez co transfer danych przy ich użyciu jest szybki i elastyczny. Protokół CANopen oferuje cztery tryby wyzwalania transmisji wiadomości z wykorzystaniem obiektów PDO.

- Tryb synchroniczny:
  - wyzwalane przez zdarzenia (ang. event driven) (przy użyciu obiektu SYNC)-model klient-serwer. Węzeł master (klient) wysyła do sieci wiadomość synchronizacji SYNC. Węzły slave (serwery) transmitują do sieci obiekty PDO w odpowiedzi na wiadomość SYNC;
  - wyzwalane przez czas (ang. time driven) (sterowane wewnętrznym zegarem urządzenia) – model producent odbiorca. Węzły slave (producenci wiadomości) wysyłają do sieci w stałych odstępach czasu obiekty PDO. Odbiorcą takich wiadomości jest węzeł master lub/i węzły slave oczekujące na konkretne wiadomości.
- Tryb asynchroniczny:
  - żądane zdalnie (ang. remotely requested) - model klient-serwer. Węzeł master (klient) wysyła do sieci obiekt PDO żądając od konkretnego węzła informację o jego stanie. Węzeł slave (serwer) w odpowiedzi na takie żądanie wysyła do sieci obiekt PDO zawierający właściwe dane.
  - wyzwalane przez zdarzenia (np. temp. zmieniła się o 0.1 °C) - model producent - odbiorca. Węzły slave (producenci wiadomości) wysyłają do sieci wiadomości PDO w wyniku wystąpienia konkretnego zdarzenia, jak np. zmiany monitorowanej temperatury o 0.1 °C.

Rozpatrując magistralę CAN z protokołem CANopen pod kątem transmisji danych z zachowaniem wymagań czasu rzeczywistego należy wziąć pod uwagę przede wszystkim obiekty PDO, gdyż priorytet obiektów PDO może być dynamicznie zmieniany i każdy obiekt PDO jest zarazem ramką danych.

Jak wynika z powyższego, do przesyłu danych w sieci CAN z protokołem CANopen, przy uwzględnieniu wymagań czasu rzeczywistego najlepiej nadają się obiekty PDO. Oferują one bogatszy zakres trybów transmisji danych, jak również elastyczność w doborze priorytetów w stosunku do obiektów SDO.

Kolejnym zagadnieniem w transmisji danych z zastosowaniem protokołu CANopen jest wybór najbardziej optymalnego mechanizmu szeregowania wiadomości. Dla sieci CAN z uwagi na mechanizm dostępu do medium transmisyjnego najbardziej naturalnym jest wykorzystanie algorytmów z przydziałem priorytetów. Uwzględniając protokół CANopen, możemy jedynie zastosować algorytm ze statycznym przydziałem priorytetów. Metoda z dynamicznym, jak również z mieszanym (MTS) przydziałem priorytetów nie może być brana pod uwagę. W metodzie MTS dokonuje się manipulacji bitami identyfikatora w sposób

niemożliwy do zastosowania w protokole CANopen. Początkowe 4 bity identyfikatora w protokole CANopen określają kod funkcji obiektu-wiadomości w sieci.

## **5. NADRZĘDNE SYSTEMY KONTROLNO - DIAGNOSTYCZNE NA BAZIE MAGISTRALI CAN**

Zagadnienie szeregowania wiadomości dla proponowanego systemu NSKD sprowadza się do zaprojektowania procesu akwizycji danych, który byłby w pełni deterministyczny dla systemu jako całości a nie dla poszczególnych jego węzłów. W tym celu wprowadzono pojęcie serwera akwizycji danych, który ma opisywać mechanizmy komunikacyjne. Zadaniem serwera akwizycji danych jest sterowanie przepływem danych w sieci polegającym na wysyłaniu sterowań do węzłów oraz pozyskiwanie informacji o aktualnym stanie mierzonych wielkości fizycznych przez węzły. Cel ten można zrealizować na bazie koncepcji metody szeregowania wiadomości z przydziałem czasu FTT-CAN (ang. Flexible Time Triggered communication on CAN). Biorąc pod uwagę założenia dla systemu NSKD, sprzętową realizację podwarstwy wykonawczej w oparciu o CAN z protokołem CANopen oraz wybór metody szeregowania wiadomości z przydziałem czasu, zostały przeanalizowane następujące modele koncepcyjne serwerów akwizycji danych dla podwarstwy sterowania [Dac05] w postaci serwera: odpytań, oczekującego, mieszanego.

### **5.1. Serwer odpytań**

W przypadku serwera odpytań (ang. polling server) wysyłane są periodycznie lub aperiodycznie zapytania do poszczególnych węzłów sieci CAN. Serwer wysyłając do sieci zapytania otrzymuje odpowiedź od poszczególnych węzłów. Brak odpowiedzi od określonego węzła świadczy o jego niepoprawnym działaniu lub uszkodzeniu sieci. Metoda odpytań charakteryzuje się determinizmem i prostotą. Zastosowanie tej metody powoduje zajętość sporej części pasma transmisyjnego. Do konstrukcji serwera odpytań przy użyciu protokołu CANopen nadają się obiekty PDO w trybie transmisji wyzwalanej synchronicznie. Serwer odpytań polega na periodycznym inicjowaniu przez węzeł master cykli elementarnych, rozpoczynających się od transmisji obiektów synchronizacji SYNC. Węzły slave sieci CAN są tak zaprogramowane, aby pod wpływem obiektu SYNC odpowiedziały przesłaniem na magistralę obiektu PDO zawierającym określone dane. Tryb transmisji synchronicznej przy wykorzystaniu obiektów SYNC (wiadomości rozgłoszeniowych) jest rozwiązaniem rzadko spotykanym w innych rodzajach magistral miejscowych. Zastosowanie tego mechanizmu wydaje się być bardziej efektywne wobec tradycyjnego mechanizmu odpytywania każdego węzła z osobna.

### **5.1. Serwer oczekujący**

W tej metodzie serwer oczekuje na wiadomości wysłane przez węzły. Zajętość pasma, dla tego typu serwera, może być znacznie zredukowana w stosunku do serwera odpytań i ściśle zależy od konfiguracji węzłów sieci. Wiadomości mogą być wysyłane w sposób periodyczny lub sporadyczny. Tylko w przypadku wiadomości periodycznych, serwer może określić, czy wszystkie węzły sieci funkcjonują poprawnie. Kompromisowym rozwiązaniem może być konfiguracja węzłów sieci w taki sposób, aby były wysyłane wiadomości aperiodyczne (np. pod wpływem zmiany mierzonej wielkości fizycznej) w połączeniu z wiadomościami okresowymi w stałych odstępach czasu tzw. ang. heart-beat messages. Serwer oczekujący nie wysyła obiektów synchronizacji jak poprzednio omawiany serwer odpytujący. Jego funkcjonowanie jest oparte na cyklicznym odczytywaniu bufora węzła

master. Cykl ten jednak nie zależy od liczby węzłów slave w systemie, tak jak dla serwera odpytującego. Węzły slave są tak zaprogramowane, aby transmitować obiekty PDO w trybie asynchronicznym pod wpływem zmiany wartości monitorowanej wielkości fizycznej jak np. temperatury. Aby zapewnić jak największą niezawodność systemu, węzły master mogą być tak zaprogramowane, aby bazując na wewnętrznym zegarze wysłać w stałych odstępach czasu obiekty PDO informujące o stanie ich pracy. Mechanizm ten określany mianem „bicia serca” (ang. heart-beat) informuje węzeł master o poprawności pracy węzłów slave. Uszkodzenie lub fizyczne odłączenie od sieci któregokolwiek węzła slave objawia się brakiem oczekiwanej informacji „bicia serca” w węźle master, co odnotowane jest jako błąd w funkcjonowaniu tego węzła.

Do transmisji wiadomości dla tego typu serwera są wykorzystywane obiekty PDO. Obiekty PDO asynchronicznej transmisji danych powinny być obdarzone wyższymi priorytetami niż obiekty transmisji „bicia serca”. Okres pojawiania się obiektów transmisji „bicia serca” powinien być na tyle duży, aby nie wprowadzać do sieci niepotrzebnego obciążenia. Serwer oczekujący, zbudowany w oparciu o powyższe mechanizmy, znacznie lepiej wykorzystuje pasmo transmisyjne niż poprzednio omawiany serwer odpytań. Węzeł master pasywnie oczekując na pojawiające się wiadomości asynchroniczne w sieci nie ma jednak kontroli nad ich częstotliwością występowania. Gdy monitorowana wartość wielkości fizycznej zmienia się zbyt często w wielu węzłach jednocześnie, wówczas może dojść do blokowania dostępu do sieci wiadomości o wyższych priorytetach. Jest to sytuacja bardzo niekorzystna, gdyż węzeł master nie otrzymuje na czas informacji o stanie pracy całego systemu. Rozwiązaniem tego problemu może być takie dobranie parametru MIT (ang. Minimum Interarrival Time) dla wiadomości aperiodycznych, aby pojawiały się one w sieci na tyle rzadko, aby nie blokowane były wiadomości o niższych priorytetach.

Dobór parametru MIT nie jest zagadnieniem trywialnym, gdyż zależy od wielu czynników, jak: liczba węzłów w sieci, długości ramek danych wiadomości aperiodycznych, częstotliwość pojawiania się wiadomości mechanizmu „heart-beat”. Innym podejściem uniknięcia niekorzystnego zjawiska blokowania może być interwencja węzła master „odcinająca” węzeł slave transmitujący zbyt często wiadomości aperiodyczne lub zmniejszająca priorytet wiadomości aperiodycznych wysyłanych do sieci przez ten węzeł.

### **5.3. Serwer mieszany**

Rozwinięciem koncepcji serwera odpytującego oraz serwera oczekującego jest serwer mieszany bazujący zarówno na wiadomościach synchronicznych, jak i asynchronicznych. W zależności od doboru parametrów odpytywania oraz konfiguracji węzłów sieci można otrzymać lepsze rezultaty niż dla poprzednio omawianych typów serwerów.

W systemie wykorzystującym serwer mieszany każdy z węzłów slave zaprogramowany jest tak, aby transmitować zarówno obiekty PDO synchroniczne jak i asynchroniczne. Wiadomości synchroniczne obdarzone są wyższymi priorytetami niż wiadomości asynchroniczne. Ponadto węzły slave podzielone są na grupy. W każdym cyklu elementarnym węzeł master wysyła wiadomość synchronizacji kierowaną do innej grupy węzłów slave. Długość każdego cyklu jest tak dobrana, aby każdy węzeł danej grupy mógł wysłać wiadomość PDO w trybie synchronicznym oraz aby mogły pojawić się również wiadomości asynchroniczne z dowolnego węzła sieci.

### **5.4. Serwer odpytań spełnienia ograniczeń czasowych**

Spełnienie ograniczeń czasowych dla systemu powstałego w oparciu o magistrale CAN z protokołem CANopen wykorzystującego koncepcję przedstawionych serwerów zależy przede



wszystkim od szybkości transmisji oraz obciążenia transmisyjnego [WeSz01]. Podstawowym problemem w przypadku serwera odpytań oraz serwera mieszanego jest wyznaczenie długości trwania cyklu elementarnego. Cykl ten powinien być wyznaczony w taki sposób, aby nie przekroczyć ograniczenia czasowego dla całego systemu oraz aby zajętość pasma transmisyjnego dla potrzeb transmisji synchronicznej pozwalała na przesłanie dodatkowych wiadomości, takich jak:

- wiadomości asynchroniczne wysyłane przez węzły slave (przypadek serwera mieszanego),
- wiadomości zarządzania siecią oraz sterujące przesyłane z węzła master do węzłów slave.

Sprawdzenia spełnienia ograniczeń czasowych dla potrzeb magistrali CAN dla serwera odpytań można dokonać w sposób analityczny korzystając z zaadaptowanego wzoru dla algorytmu RM (ang. Rate Monotonic):

$$\frac{C_{SYNC} + 3\tau_{bit} + \tau_{bit}}{T_{SYNC}} + \sum_{i=1}^n \left( \frac{C_i + 3\tau_{bit}}{T_i} \right) \leq \frac{R_{sys}}{T_{cycle}} \leq n(2^{1/n} - 1) \Rightarrow T_{cycle} \geq \frac{R_{sys}}{n(2^{1/n} - 1)}$$

gdzie:

- $C_i$  - jest czasem przesyłu ramki wiadomości z węzła  $i$ ,
- $T_i$  - jest okresem pojawiania się wiadomości (pochodzącej z węzła  $i$ ) w systemie; dla serwera odpytań  $T_i = T_{SYNC} = T_{cycle}$ ,
- $T_{SYNC}$  - jest okresem pojawiania się wiadomości synchronizacji SYNC,
- $R_{sys}$  - czas odpowiedzi systemu,
- $T_{cycle}$  - czas trwania cyklu elementarnego,
- $n$  - liczba węzłów w systemie.

Przedstawiony warunek jest warunkiem wystarczającym na spełnienie ograniczeń czasowych dla serwera odpytań.

## 6. NADRZĘDNY SYSTEM KONTROLNO - DIAGNOSTYCZNY DLA AKCELERATORÓW

Zaproponowana metodyka została zastosowana do konstrukcji NSKD dla projektu akceleratorowego SLS [10] (ang. Swiss Light Source) [4]. W celu zapewnienia jak najkrótszych przestojów akceleratora niezbędne jest zastosowanie systemu NSKD [3] umożliwiającego szybkie zdiagnozowanie i zareagowanie na powstałe zaburzenie pracy systemu sterowania akceleratorów.

Obiektem sterowania systemu NSKD w SLS jest 166 komputerów VME służących do monitorowania i sterowania wszystkich urządzeń wykonawczych akceleratora. Komputery VME są obiektami sterowania i w tym celu został wyposażony w interfejs kontrolno-diagnostyczny posiadający następujące rodzaje sygnałów:

- diagnostyczne sygnały cyfrowe (TTL); każdemu sygnałowi przypisany jest jeden bit określający stan „OK” lub „Error”: status pracy wentylatora, status napięć zasilających AC, status napięć zasilających DC, status linii sys-reset, status magistrali VME;
- sygnały sterujące cyfrowe (TTL); każdemu sygnałowi odpowiada jeden bit określający poziom aktywny/nieaktywny: restart systemu (ang. sys-reset), włącz/wyłącz zasilanie (ang. power down/up); pomiar temperatury w przestrzeni roboczej VME sygnał analogowy (0-10V).

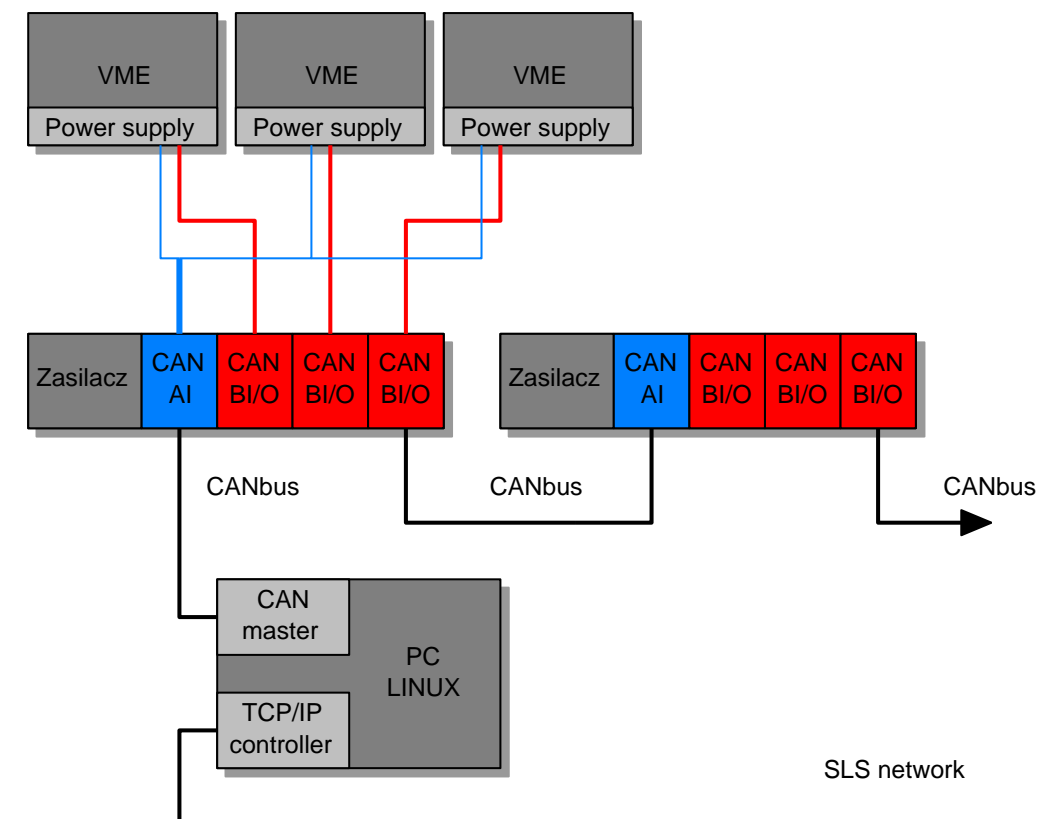
Do komunikacji pomiędzy węzłami systemu NSKD zdecydowano się na magistralę CAN cechującą się prostotą, niezawodnością, elastycznością rozwiązań oraz niską ceną. Moduły CAN zgodnie z przyjętymi założeniami mają do spełnienia następujące funkcje:

- monitorujące: na jeden komputer VME przypada 5 sygnałów cyfrowych oraz 1 sygnał analogowy,
- sterujące: na jeden komputer VME przypadają 2 sygnały cyfrowe.

Kontrolery sieci CAN umieszczone są w komputerze PC. Jeden kontroler sieci CANopen może obsłużyć do 128 modułów CANopen. Ograniczenie to wymusza zastosowanie dwóch kontrolerów CAN (dla dwóch niezależnych sieci CAN) dla zapewnienia możliwości sterowania wszystkich zainstalowanych komputerów VME.

W celu redukcji kosztów zastosowano pojedynczy moduł CANopen uCAN.8.dio [5] dla cyfrowych sygnałów wejściowych i wyjściowych. Zostało to osiągnięte przez odpowiednie zaprogramowanie modułu CANopen, uzyskując w ten sposób 5 linii wejściowych, 2 wyjściowe oraz jedną zapasową (z możliwością użycia jej jako wejściowej lub wyjściowej). Tak zaprogramowany moduł zapewnia sterowanie/diagnostykę jednego komputera VME (Rysunek 2.9). W celu pomiaru temperatury zastosowano moduł uCAN.4.ai [5] posiadający 4 linie wejściowe do pomiaru sygnałów analogowych.

Z punktu widzenia sieci CAN, trzy komputery VME są kontrolowane przez cztery moduły CAN (3 moduły cyfrowe oraz jeden moduł analogowy do pomiaru temperatury). Na podstawie tego można wyznaczyć, że minimalna liczba modułów CAN, jakie należy zastosować wynosi 222.



**Rys.2. Powiązanie modułów CAN względem komputerów VME**

Długość elementarnego cyklu diagnostyczno-sterującego oraz wielkość rezerwy czasowej dla procesu sterującego magistralą CAN, zależy od czasu odpowiedzi dla magistrali CAN. Przy wyznaczaniu czasu odpowiedzi dla segmentu magistrali CAN składającego się ze 108 węzłów należy wziąć pod uwagę 27 modułów do pomiaru wartości analogowych oraz 81 modułów cyfrowych wejścia/wyjścia. W obliczeniach czasu odpowiedzi należy uwzględnić również:

- czas trwania wiadomości synchronizacji SYNC,
- liczbę 8 bajtów danych przypadających na obiekt PDO związany z modułem CAN do pomiaru wartości analogowych,
- liczbę 1 bajtu danych przypadających na obiekt PDO związany z modułem CAN dla sygnałów cyfrowych,
- wystąpienie maksymalnej liczby bitów wtrąconych,
- prędkość transmisji w sieci 50 Kb/s.

Sumaryczny czas odpowiedzi składa się z dwóch elementów a mianowicie: czasu odpowiedzi dla wielkości monitorowanych oraz czasu odpowiedzi dla sygnałów sterujących. Zakładając docelową liczbę modułów wynoszącą 112 na każdy segment magistrali CAN, otrzymujemy czas odpowiedzi systemu równy 299.040 ms. Wyznaczając minimalną wartość cyklu elementarnego dla serwera odpytań na podstawie warunku podanego w [4, 2] uzyskujemy wartość 430.667 ms. Ograniczenia czasowe wynikające z przyjętych wymagań dla systemu NSKD w SLS zakładają pozyskanie danych z magistrali CAN w czasie nieprzekraczającym 500 ms. Na podstawie powyższych wyliczeń oraz założeń przyjęto, iż długość trwania cyklu elementarnego dla serwera odpytań sterującego magistralą CAN (dla każdego z segmentów) wynosić będzie 500 ms przy uwzględnieniu rezerwy czasowej ok. 70 ms. Rezerwa czasowa uwzględnia potencjalną rozbudowę każdego segmentu magistrali do maksymalnej liczby węzłów oraz uwzględnia czas na opracowanie sterowań przy zastosowaniu ewentualnego algorytmu kontrolno-sterującego.

## 7. WNIOSKI

Budowa rozproszonych systemów komputerowych stanowi złożone przedsięwzięcie. Przy konstrukcji takich systemów należy stosować podejście strukturalne. W takim podejściu ważnym zagadnieniem jest dokonanie analizy takiego systemu i określenie klasy systemów, do których dany system należy.

W opracowaniu przytoczone zostały pojęcia systemów reaktywnych i systemów czasu rzeczywistego. Systemy reaktywne stanowią szeroką klasę systemów, które związane są z pewnym środowiskiem w ten sposób, że muszą cały czas reagować na zmiany w nim zachodzące. W systemach czasu rzeczywistego wymagane jest spełnienie ograniczeń klasowych.

Dla magistrali CAN ze względu na jej własności mogą być stosowane podstawowe metody analizy czasowej RMA.

W opracowaniu przedstawiono możliwości komunikacyjnych rozwiązań architektonicznych dla magistrali CAN w zastosowaniu do nadrzędnego systemu kontrolno-diagnostycznego. Wykazano możliwość dokonania analizy czasowej dla tych rozwiązań.

Przedstawiono przykład zastosowania takiej metodyki dla systemu bazującego na magistrali CAN przeznaczonego dla nadzoru poprawności działania systemu sterującego akceleratorem.

## 8. LITERATURA

- [1] CANopen Application Layer and Communication Profile. CiA Draft Standard 301, 2000.
- [2] DACH M., WEREWKA J.: Analiza czasów reakcji w systemie kontrolno-diagnostycznym bazującym na magistrali CAN. Materiały Konferencyjne X Konferencji Systemów Czasu Rzeczywistego. Ustron, 15-18 września 2003. Instytut Informatyki Politechniki Śląskiej w Gliwicach. Strony 157-168.
- [3] DACH M., KORHONEN T., PAL T.: Interfacing CANbus to EPICS at the Swiss Light Source (Korea 03, referat na konferencji ICALEPCS).
- [4] DACH M.: Zagadnienia Konstrukcji Nadrzędnych Systemów Kontrolno-Diagnostycznych Czasu Rzeczywistego Bazujących na Magistrali CAN, AGH, Wydz. EAEiI, Kraków 2004, rozprawa doktorska, promotor J. Werewka, pp. 122.
- [5]  $\mu$ CAN, Dokumentacje modułów cyfrowego i analogowego: [www.microcontrol.net](http://www.microcontrol.net/download/doc/mcan8dio_ip20_de.pdf) w [download/doc/mcan8dio\\_ip20\\_de.pdf](http://www.microcontrol.net/download/doc/mcan8dio_ip20_de.pdf) oraz [download/doc/mcan4ti\\_ip65\\_de.pdf](http://www.microcontrol.net/download/doc/mcan4ti_ip65_de.pdf).
- [6] IEEE/ANSI Std 729: Glossary of Software Engineering Terminology. 1983.
- [7] LIU L., LAYLAND J.: Scheduling algorithms for multiprograming in a hard real - time environment, J. Assoc. Comput. Mach. Vol. 20, No. 1, pp. 46 - 61, January 1973.
- [8] NOLTE T., "Reducing Pessimism and Increasing Flexibility in the Controller Area Network", ISBN 91-88834-48-4, 2003.
- [9] NOLTE T., Sjodin M., Hanssen H.: Server-Based Scheduling of the CAN Bus. Malardalen University, Sweden 2003.
- [10] Synchrotron Lichtquelle Schweiz SLS. Villigen PSI. Wydawnictwo wewnętrzne PSI 1999.
- [11] SZMUC T.: Zaawansowane Metody Rozwijania Systemów Czasu Rzeczywistego. Wydawnictwo CCATIE, 15, 1998.
- [12] WEREWKA J.: Systemy rozproszone sterowania i akwizycji danych. Sterowniki programowalne i magistrale miejscowe, Wydawnictwa CCATIE, Kraków, 1998.
- [13] WEREWKA J., DRWAL A., ŻABA S.: Zagadnienie badania charakterystyk czasowych i przepustowości magistral miejscowych, IV Konferencja Systemy Czasu Rzeczywistego, 22-25 wrzesień 1997 Szklarska Poręba, str. 325-336.
- [14] WEREWKA J., SZMUC T. (red.): Analiza i projektowanie systemów komputerowych czasu rzeczywistego o różnym stopniu rozproszenia, Wydawca PTI i KA AGH, Kraków, 2001, pp. 560.
- [15] WIERINGA R.J.: Design Methods for Reactive Systems, Elsevier Science, 2003, pp. 457.

### **DESIGN OF REACTIVE DISTRIBUTED EVENT SERVICE SYSTEMS BASED ON CAN BUS ON EXAMPLE OF A SUPERVISORY CONTROL AND DIAGNOSTIC SYSTEM**

**Abstract:** In the paper a solution proposal of reactive systems treated as the systems of events was introduced. Reactive systems are a wide class of the systems connected with certain environment to react all the time on changes in the environment. Because the systems of events are spread usually over certain area, so the introduced solutions concerns creating architectures for distributed computer systems. In the paper a method of the construction of Supervisory Control and Diagnostic systems (NSKD) basing on CAN bus was introduced. The fulfilment of the limitations of the real time constraints was discussed. The possibilities of utilization of various CAN bus communication mechanisms useful in construction of real time distributed systems were described. An implementation example was introduced of a Supervisory Control and Diagnostic system (NSKD) basing on CAN bus, designed for the supervision of the correctness of the working of the system steering the accelerator.

Recenzent: dr hab. inż. Edward Hrynkiewicz, prof. Politechniki Śląskiej