

Dariusz CABAN
Paweł MOROZ

PRZEDSTAWIENIE WYNIKÓW ANALIZY STRUKTURY NAGŁÓWKA RAMKI CAN POD KĄTEM WPLYWU POSZCZEGÓLNYCH PÓL NA LICZBĘ WSTAWEK BITOWYCH

Streszczenie. W artykule przedstawiono wyniki analizy struktury nagłówka ramki sieci CAN, przeprowadzonej dla określenia wpływu zawartości jego poszczególnych pól na liczbę wstawek bitowych w ramce. Analiza ta oraz wykonane eksperymenty wykazały, że możliwe jest ograniczenie liczby wstawek poprzez odpowiedni dobór nagłówka.

Słowa kluczowe: protokół CAN, ramka protokołu, wstawki bitowe

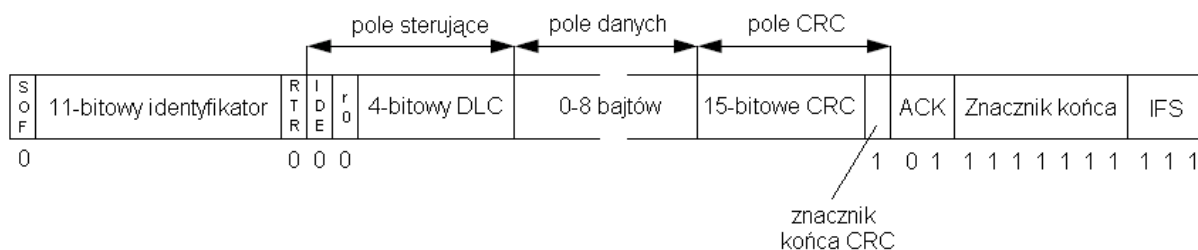
1. WPROWADZENIE

Mechanizm wstawek bitowych (ang. *bit stuffing*) wykorzystywany jest w niektórych protokołach warstwy łącza danych sieci dla zapewnienia synchronizacji węzła nadawczego i węzłów odbiorczych. Jednym z takich protokołów jest protokół sieci CAN (ang. *Controller Area Network*), zaprojektowanej do zastosowań w systemach sterowania pojazdami. Dzięki swoim zaletom sieć ta znalazła też zastosowanie w innych systemach sterowania, m. in. w inteligentnych budynkach, sprzęcie medycznym i systemach wind. Ogólnie docenioną zaletą magistrali CAN jest niska cena interfejsu komunikacyjnego (około 3\$), jednocześnie zapewnia ona wysoką prędkość transmisji (do 1 Mbps).

Przy braku transmisji na magistrali sieci CAN występuje wysoki stan logiczny. Węzeł nadawczy rozpoczyna transmisję ramki od wysłania bitu startu (SOF, ang. *Start of Frame*), o niskim stanie logicznym. Synchronizacja zegarów w węzłach zachodzi po wystąpieniu opadającego zbocza sygnału. W sieci CAN synchronizacja taka musi jednak następować także w toku transmisji pozostałej części ramki. Zbocze opadające nie pojawi się przy nadawaniu dłuższego ciągu jednakowych bitów. Aby do tego nie dopuścić, po pięciu kolejnych jednakowych bitach wysyłany jest bit o przeciwnej wartości, co w konsekwencji wymusi pojawienie się zbocza opadającego najpóźniej po 11 wysłanych bitach. Wstawianie bitów nie powoduje zniekształcenia zawartości ramki, dodatkowe bity są usuwane po stronie odbiorczej.

2. ZAKRES BADAŃ

Skutkiem stosowania wstawek bitowych jest wydłużenie przesyłanych ciągów bitów. Spowodowany tym narzut jest zmienny i zależy od zawartości ramki. Celem prowadzonych badań było określenie jego wielkości oraz ustalenie przyczyn akurat takiej liczby wystąpienia dodatkowych bitów.



Rys. 1. Struktura ramki danych sieci standardu CAN 2.0A

W badaniach, których wyniki przedstawiono w [5], skoncentrowano się na obszarze pola danych ramki. Strukturę takiej ramki przedstawiono na Rys. 1. Użytkownik protokołu (twórca systemu) ma wpływ na wartości przesyłane w polu danych, dzięki czemu można zmniejszyć liczbę wstawek bitowych bez ingerencji w sam protokół. Jednak proponowane rozwiązania ograniczają się do przekodowania informacji, przy czym przyjęto założenie, że w polu danych bardzo często przesyłane są wartości z dużą liczbą jednakowych bitów, zazwyczaj 0. Zastosowanie funkcji mieszającej (XOR bajtu danych i naprzemiennych bitów 0 i 1, np. 01010101) przyniosło pozytywny efekt w postaci zmniejszenia liczby wstawek bitowych. Jednak należy pamiętać o tym, że rozwiązanie to nie jest uniwersalne i każde wdrożenie wymaga ponownej jego weryfikacji.

Liczbę wstawek bitowych można zmniejszyć też poprzez odpowiedni dobór identyfikatora ramki, który wraz z bitem RTR (ang. *Remote Transmission Request*) oraz polem sterującym stanowi jej nagłówek [4]. Odbiorca ramki na podstawie identyfikatora podejmuje decyzję, czy wiadomość zawarta w ramce ma być dalej przetwarzana. Opracowano dwa standardy sieci CAN różniące się długością identyfikatora. W ramce standardu CAN 2.0A identyfikatory są 11-bitowe, a w ramce standardu CAN 2.0B 29-bitowe. W standardzie CAN 2.0A 16 identyfikatorów jest zarezerwowanych, stąd do dyspozycji projektanta systemu są 2032 identyfikatory [2]. W tabelicy 1 podano, ilu identyfikatorów ramek standardu CAN 2.0A można użyć przy założonej liczbie wstawek bitowych w nagłówku ramki oraz długości pola danych, zakodowanej w obszarze DLC (ang. *Data Length Code*) ramki [4].

Tablica 1. Liczba możliwych do użycia identyfikatorów w funkcji liczby wstawek bitowych w nagłówku ramki oraz liczby bajtów danych

Liczba wstawionych bitów	Długość pola danych [w bajtach]				
	0	1	2-3	4-7	8
0	0	0	0	745	1131
1	1332	1436	1490	1005	765
2	634	560	520	279	145
3	81	51	38	19	7
4	1	1	0	0	0

Tablica 2 zawiera maksymalne liczby wstawek bitowych w ramce w funkcji długości jej pola danych [3]. Szczególnie ważna wydaje się informacja, że w nagłówkach ramek najkrótszych występuje najwięcej wstawek bitowych. Każdy dodatkowy bit w ramce powoduje jej wydłużenie o 1–2%. Wzrost pesymistycznej liczby wstawek bitowych wraz z wydłużaniem się pola danych jest prawie liniowy.

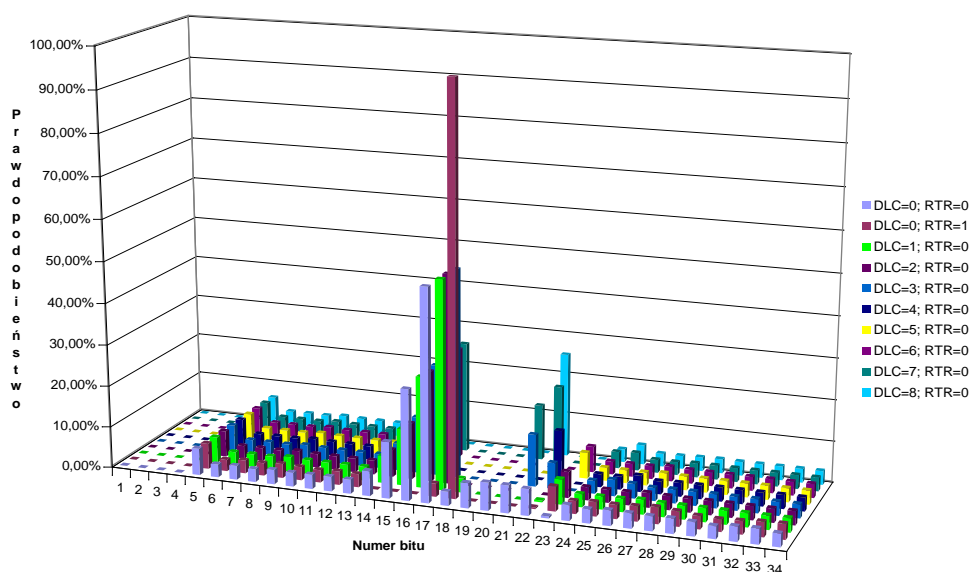
Tablica 2. Maksymalna liczba wstawek bitowych w ramce CAN 2.0A

Liczba bajtów danych	0	1	2	3	4	5	6	7	8
CAN 2.0A [bit]	8	10	11	13	15	17	19	21	23

W ramach badań prowadzonych na potrzeby pracy [3] postanowiono ustalić zależności pomiędzy liczbą wstawek bitowych w nagłówku, a zawartością obszaru DLC, a następnie sprawdzić, jak zmiana układu nagłówka, bądź zawartych w nim wartości, wpłynie na liczbę wstawek bitowych.

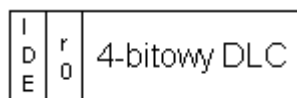
3. OMÓWIENIE WYNIKÓW

Dla ustalenia, który z elementów nagłówka wpływa na liczbę wstawek bitowych wykonano eksperyment, w którym mierzono liczbę wystąpień wstawek bitowych po każdym z bitów identyfikatora. Doświadczenie to wykonano dla wszystkich możliwych nagłówków ramki CAN 2.0A. Uzyskany wynik przedstawiono za pomocą histogramu (Rys. 2).



Rys. 2. Liczba wystąpień wstawek bitowych po każdym z bitów informacyjnych nagłówka

Doświadczenia wykazały, że największe zróżnicowanie w liczbie wstawek bitowych występuje w obszarze DLC. Analizując wyniki, ustalono, że na przedstawioną w poprzednim rozdziale własność nagłówka wpływ ma układ obszarów w polu sterującym ramki. Wartości w tych obszarach nie są związane z przesyłanymi danymi, a jedynie z typem ramki, jaka jest wykorzystywana do transmisji.



Rys. 3. Pole sterujące ramki standardu CAN 2.0A

Ze względu na znaczenie właśnie tych obszarów po analizie struktury ramki CAN zaproponowano następujące zmiany:

- 1) zmiana kodowania w obszarze DLC;
- 2) zmiana wartości domyślnej bitu r0 (bit zarezerwowany, wartość domyślna równa 1).

Wartość bitu IDE (ang. *Identifier Extended*) w ramce sieci CAN 2.0A musi być zawsze równa 0. Przeanalizowano również zależność liczby wstawek bitowych od wartości obszaru DLC.

3.1 Zmiana kodowania długości pola danych

Obszar DLC ma długość 4 bitów, co pozwala na zakodowanie w nim 16 różnych wartości. Dozwolonych jest jednak tylko 9 wartości, gdyż długość pola danych w ramce danych CAN mieści się w zakresie od 0 do 8 bajtów. Sposób kodowania wartości DLC podano w tabelicy 3.

Tablica 3. Metoda kodowania informacji w obszarze DLC [1]

Liczba bajtów danych	Kod długości danych			
	DLC3	DLC2	DLC1	DLC0
0	d	d	d	d
1	d	d	d	r
2	d	d	r	d
3	d	d	r	r
4	d	r	d	d
5	d	r	d	r
6	d	r	r	d
7	d	r	r	r
8	r	d	d	d

d – stan dominujący, logiczne 0

r – stan recesywny, logiczna 1

Przeprowadzono eksperyment dla sprawdzenia, czy dla pozostałych możliwych wartości kodu w obszarze DLC nastąpi zmniejszenie liczby wstawek bitowych w nagłówku. W efekcie uzyskano w większości bardzo dobre wyniki (tablica 4). Pogorszenie nastąpiło tylko przy samych jedynkach w obszarze DLC. Wystąpiło tu podobne zachowanie się ramki jak przy ramkach najkrótszych – było aż 6 nagłówków z 4 wstawkami. Jednak w porównaniu z ramkami o kodzie w obszarze DLC złożonym z samych zer występuje liczna grupa nagłówków bez wstawki bitowej.

Tablica 4. Liczba możliwych do użycia identyfikatorów w funkcji liczby wstawek bitowych w nagłówku ramki dla kodów nie wykorzystywanych w obszarze DLC

Liczba wstawionych bitów	Liczba bajtów danych	
	9-14	15
0	1131	1131
1	765	564
2	145	276
3	7	71
4	0	6

3.2 Zmiana wartości domyślnej bitu r0

W polu sterującym ramki występuje bit r0. Obecnie bit ten nie jest wykorzystywany, jego wartość nie jest weryfikowana przez odbiorcę wiadomości. Dlatego postanowiono przeanalizować, jaki wpływ na liczbę wstawek bitowych będzie miała zmiana wartości domyślnej z 0 na 1. Wyniki analizy zostały zamieszczone w tablicy 5.

Tablica 5. Liczba możliwych do użycia identyfikatorów w funkcji liczby wstawek bitowych w nagłówku ramki po zmianie wartości bitu r0

Liczba wstawionych bitów	Liczba bajtów danych
	0-8
0	1332
1	634
2	81
3	1

3.3 Liczba wstawek bitowych w ramach zdalnego żądania

Węzeł sieci CAN może zgłosić zapotrzebowanie na dane wytwarzane przez inny węzeł poprzez wysłanie ramki zdalnego żądania. Identyfikator tej ramki jest taki sam, jaki będzie miała ramka z żądanymi danymi. W polu sterującym ramki bit RTR ma wartość 1, a w obszarze DLC zakodowana jest liczba oczekiwanych bajtów danych. Pole danych nie występuje. Sprawdzono, jak wartość DLC wpływa na liczbę wstawek bitowych. Okazało się, że najlepiej jest wtedy, gdy mieści się ona w zakresie 2÷8 (tablica 6).

Tablica 6. Liczba możliwych do użycia identyfikatorów w funkcji liczby wstawek bitowych w nagłówku ramki zdalnego żądania oraz wartości DLC

Liczba wstawionych bitów	Wartość DLC	
	0-1	2-8
0	0	1436
1	1436	560
2	560	52
3	52	0

4. PODSUMOWANIE

Analiza działania mechanizmu wstawek bitowych w nagłówku ramki CAN wykazała, że poprzez odpowiedni dobór nagłówka ramki można ograniczyć liczbę wstawek bitowych. Oprócz rozwiązań dostępnych dla projektanta systemu istnieją również możliwości ograniczenia liczby wstawek bitowych poprzez modyfikację wartości występujących w polu sterującym ramki, a mianowicie w obszarach: DLC i r0.

Pierwsze z rozwiązań, polegające na zmianie kodowania obszaru DLC nie zapewnia kompatybilności z obecnie wykorzystywanymi układami, a więc mogłoby być stosowane tylko w nowych rozwiązaniach, gdzie wszystkie sterowniki sieci CAN wspierałyby nowy sposób zapisu informacji o długości pola danych.

Natomiast drugie rozwiązanie – polegające na zmianie domyślnej wartości bitu r0 – jest rozwiązaniem uniwersalnym i zgodnym ze standardem sieci CAN. W dokumentacji wartość bitu r0 określona jest następująco: nadawca powinien ustawić wartość tego bitu na 0, natomiast odbiorca nie sprawdza jego wartości [1]. Dokumentacja nie wspomina o tym, czy wartość tego bitu jest uwzględniana przy wyliczaniu sumy kontrolnej. Przeprowadzono zatem eksperyment, w którym wysłano dwie ramki z bitem r0 ustawionym na 1, różniące się wartościami sumy kontrolnej. W jednym przypadku do wyliczenia sumy kontrolnej (u nadawcy) przyjęto, że bit ten ma wartość 0, w drugim uwzględniono rzeczywistą jego wartość. Układ odbiorczy, w tym przypadku MCP2510, odebrał poprawnie ramkę, w której do wyliczenia sumy kontrolnej uwzględniono rzeczywistą wartość bitu r0.

Natomiast badania dotyczące ramki zdalnego żądania wykazały, że najlepiej jest, kiedy w obszarze DLC zapisane są wartości z zakresu 2÷8. W przeciwnym przypadku w nagłówku zawsze wystąpi przynajmniej jedna wstawka bitowa.

5. LITERATURA

- [1] Bosch: CAN Specification. Robert Bosch GmbH, Stuttgart 1991.
- [2] Lawrenz W.: CAN System Engineering. From Theory to Practical Applications. Springer-Verlag New York, Inc, 1997.
- [3] Moroz P.: Metody ograniczania liczby bitów nadmiarowych w protokołach wykorzystujących mechanizm wstawek bitowych. Praca doktorska wykonana pod kierunkiem prof. dr. hab. inż. A. Kwietnia, Gliwice 2010.

- [4] Nolte T., Hansson H., Norstrom C.: Minimizing CAN Response-Time Jitter by Message Manipulation. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02), San Jose, September 2002.
- [5] Nolte T., Hansson H., Norstrom C., Punnekkat S.: Using bit-stuffing distributions in CAN analysis. IEEE Real-Time Embedded Systems Workshop, Londyn, December 2001.
- [6] Werewka J., Dach M.: Response-time analysis of a CAN network used for supervisory control and diagnostic systems. Control and Cybernetics, vol. 39 (2010) No. 4.

PRESENTATION OF THE RESULTS OF ANALYSIS OF CAN FRAME HEADER STRUCTURE WITH RESPECT TO THE EFFECT OF THE INDIVIDUAL FIELDS ON THE NUMBER OF STUFFED BITS

Abstract. The paper presents the results of the analysis of CAN frame header structure carried out to determine the effect of the contents of its individual fields on the number of stuffed bits in the frame. This analysis along with experiments that were conducted have shown that the number of stuffed bits can be controlled by properly structuring the header.

Keywords: CAN protocol, protocol frame, stuffed bits.