

I. Treść zadania.

Zrealizować aplikację w postaci pliku wykonywalnego, która wyznacza CRC zgodnie z algorytmem właściwym MODBUS RTU. Aplikacja powinna:

- a) umożliwiać wprowadzenie z klawiatury sekwencji bajtów w notacji heksadecymalnej. Maksymalna długość sekwencji powinna być ograniczona do 256 bajtów.
- b) Przykład: 01 10 0011 0003 06 1AC4 BAD0
- c) umożliwiać wprowadzenie liczby powtórzeń n algorytmu wyznaczania CRC. Liczba ta powinna być zawarta w granicach $1..10^9$,
- d) wyświetlić wartość wyliczonej CRC w zapisie heksadecymalnym,
- e) wyświetlać łączny czas wyświetlania n powtórzeń algorytmu w [s].

II. Algorytm obliczania sumy kontrolnej CRC.

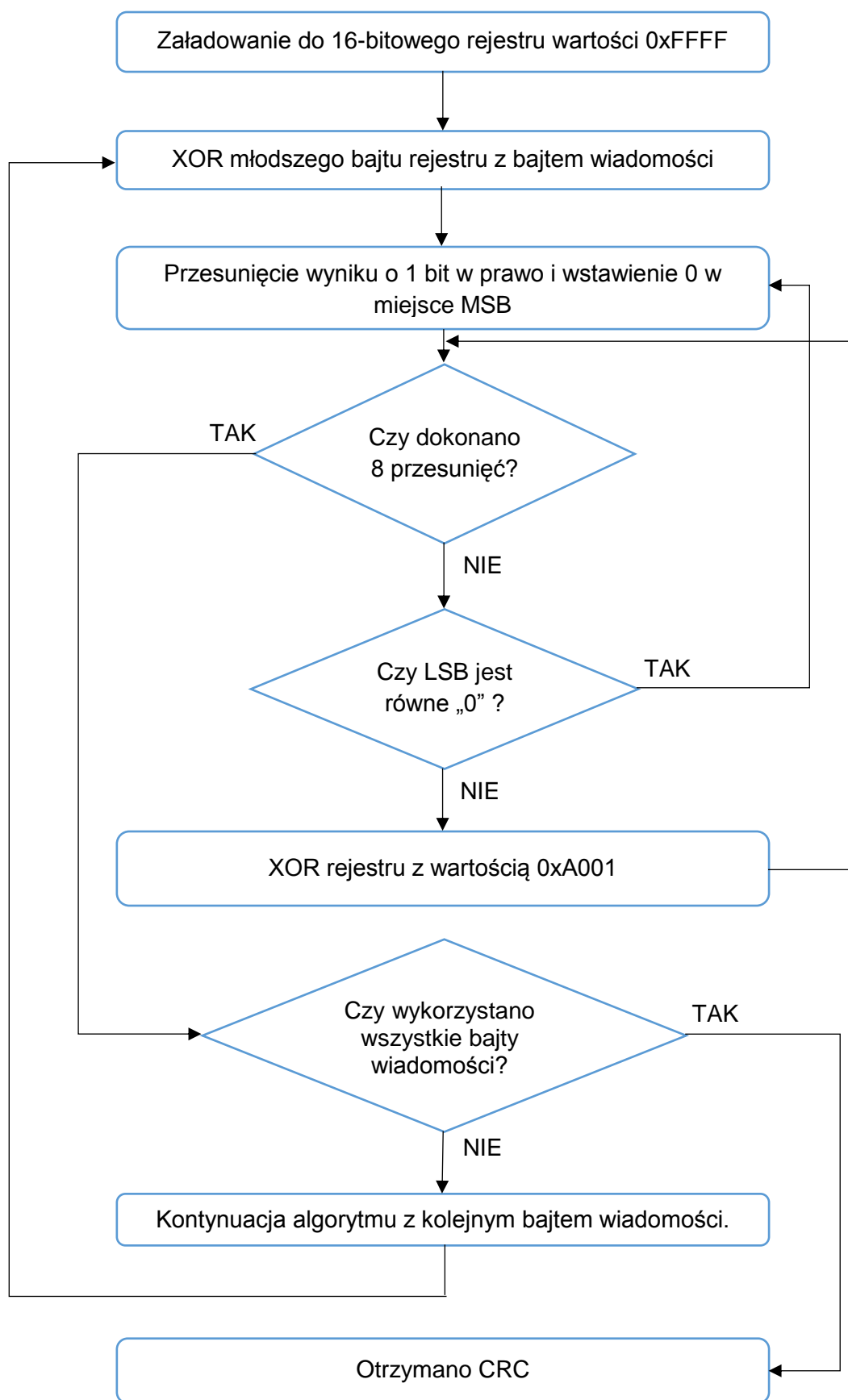
W ogólnym przypadku podstawową jednostką danych w protokole Modbus jest PDU (Protocol Data Unit), która składa się z kodu funkcji i danych do wysłania. Dodatkowe pola w wiadomości to adres i pole kontroli błędów (LRC lub CRC). Zależne są one od typu sieci i razem z PDU stanowią ADU (Application Data Unit). Maksymalny rozmiar ADU wynosi 256 bajtów.

Warto wspomnieć, że do zapisu adresów i danych Modbus wykorzystuje system „big-Endian”. Oznacza, że najbardziej znaczący bajt jest wysyłany jako pierwszy (o ile przesyłane dane są dłuższe niż jeden bajt).

W sieciach szeregowych RS-232C i RS-485 wyróżnia się dwa tryby transmisji ASCII (American Standard Code for Information Interchange) i RTU (Remote Terminal Unit), które różnią się zawartością pól poszczególnych przesyłanych wiadomości.

W trybie RTU do kontroli błędów wykorzystywany jest algorytm CRC – Cyclical Redundancy Check, do którego wyznaczenia należało stworzyć aplikację. Pole CRC służy do weryfikacji całej wiadomości. Zawiera ono liczbę binarną 16-bitową, obliczaną i dołączaną do wiadomości wysłanej przez urządzenie nadające. Jednostka odbierająca wiadomość także oblicza CRC na podstawie otrzymanej wiadomości i porównuje swój wynik z polem CRC otrzymanej wiadomości. Jeśli wartości się różnią, zostaje wykryty błąd.

Sposób generacji CRC przedstawiono na poniższym schemacie:



Rysunek 1 – Schemat generowania wartości CRC

Przy dołączaniu CRC do wiadomości należy zamienić miejscami bajty – młodszy bajt CRC jest wysyłany jako pierwszy (jest to ważna informacja, ponieważ zrealizowana aplikacja zwraca CRC w postaci do wysłania). Np. CRC o wartości 0xA2C4 będzie wysłane jako C4A2.

W celu wyznaczenia CRC za pomocą programu można dany algorytm zrealizować bezpośrednio lub skorzystać z funkcji opartej na dwóch tablicach, w których znajdują się wszystkie możliwe starsze i młodsze bajty CRC. Dzięki temu wyznaczenie CRC trwa krócej. Ma to duże znaczenie w przypadku systemów czasu rzeczywistego, gdzie czas na wyznaczenie CRC jest ograniczony.

Dodatkowo w sieciach MODBUS maksymalny dopuszczalny okres czasu pomiędzy kolejnymi bajtami ramki, nie naruszający jej ciągłości, jest równy 1,5-krotnemu czasowi trwania transmisji jednego bajtu. W przeciwnym wypadku zostanie uznana ona za niekompletną.

W celu określenia w ciągu jakiego czasu powinno odbyć się wyznaczenie CRC, można posłużyć się czasem wysłania pojedynczego bajtu. Jeśli wyznaczenie CRC będzie trwało poniżej tej wartości, to czas taki powinien być akceptowalny.

Dla przykładu, przy prędkości transmisji 115200 b/s, nominalny czas trwania transmisji jednego bajtu wynosi 8,68 μ s. Wtedy czas transmisji 1 bajtu pola CRC wynosi około: 69,4 μ s.

III. Opis zrealizowanej aplikacji.

Aplikacja została wykonana w języku C++, w środowisku Code::Blocks ver 17.12. Ma ona postać wiersza poleceń.

W celu podania ramki danych należy wpisać jej wartość w postaci heksadecymalnej (bez spacji/odstępów). Aplikacja zwraca wpisaną wartość i jej rozmiar w bajtach. Następnie należy podać liczbę powtórzeń algorytmu z przedziału 1 - 10^9 , a następnie zostaje zrealizowanie wyznaczenie CRC i pomiar czas realizacji algorytmu. Otrzymane wyniki są wyświetlane. Wartość CRC jest przedstawiona w postaci, w jakim należy dołączać ją do wiadomości. Czyli młodszy bajt CRC jest wyświetlony jako pierwszy (np. wyświetlona wartość 0x12B4 to CRC równe 0xB412).

IV. Uzyskane wyniki.

Za pomocą przykładowej ramki danych, o wartości 0x01 10 0011 0003 06 1AC4 BAD0, przeprowadzono pomiar czasu realizacji wyznaczenia CRC.

W przypadku pojedynczego wykonania algorytmu wyznaczenia CRC, wykonana aplikacja nie jest w stanie zmierzyć poprawnie czasu jego realizacji (wynika to z zastosowanej funkcji clock() w języku C++). Dlatego w celu oszacowania czasu obliczenia CRC danej ramki zostanie on wykonany wielokrotnie.

W tym przypadku przyjęto do realizacji pomiaru czasu, liczbę powtórzeń $n = 50\,000\,000$.

Poniżej przedstawiono uzyskane wyniki:

```
Podaj ramke danych w postaci heksadecymalnej (bez odstępów): 011000110003061AC4BAD0
Wpisano: 0x011000110003061AC4BAD0
Ilosc bajtow w wpisanej ramce: 11

Podaj liczbe powtorzen algorytmu wyznaczenia CRC: 50000000

Wartosc CRC wiadomosci < 011000110003061AC4BAD0 > to: 7f67
Czas calkowity: 2449ms
Czas 1 iteracji: 0.000049ms

Aby kontynuować, naciśnij dowolny klawisz . . . █
```

Rysunek 2 – Wynik testowania zrealizowanej aplikacji do wyznaczania CRC

Rzeczywista wartość CRC wynosi 0x677F (aplikacja zwraca CRC w postaci do wysłania.

Jak widać czas zastosowanego algorytmu wyznaczenia pojedynczej sumy kontrolnej CRC dla ramki o rozmiarze 11 bajtów wynosi około 0,049 μ s.

Na podstawie tego wyniku, można oszacować czas realizacji algorytmu dla wiadomości o wielkości 256 bajtów. Wynosi on około: $0,049[\mu\text{s}] / 11 * 256 = 1,14 \mu\text{s}$. Wartość ta jest wystarczająca, zanim zostanie wysłany kolejny bit, więc spełniony jest warunek o wyznaczeniu CRC w czasie rzeczywistym.

V. Wnioski.

- Zrealizowana aplikacja wyznacza CRC w czasie nie przekraczającym czasu wysłania 1 bitu (około 8,7 μ s). Jednak czas wyznaczenia zależy od długości ramki wiadomości i sprzętu który wykonuje obliczenia. W przypadku sieci o mniejszych prędkościach transmisji, czas na wyznaczenie CRC będzie większy.
- W celu poprawnego oszacowania wymaganego czasu na wyznaczenie CRC, należy określić sieć w której będzie wyznaczana. Zależy ona wtedy od właściwości i cech dołączony do sieci urządzeń podporządkowanych. Trzeba też zwrócić uwagę, że CRC będzie wyznaczane zarówno w jednostce nadrzędnej i urządzeniu podporządkowanym.
- CRC można zacząć wyznaczać już od pierwszego otrzymanego bajtu ramki wiadomości. Może to znacznie poprawić czas jego wyznaczenia (jednak nie każdy sprzęt ma taką możliwość).
- W przypadku wykonanej aplikacji zostały sprawdzone dwa algorytmy wyznaczenia CRC: tradycyjny i wykorzystujący dwie tablice. Algorytm z dwiema tablicami okazał się około 4 razy szybszy (w przypadku użytego środowiska i sprzętu).