Dariusz Caban
Instytut Informatyki
Wydział Automatyki, Elektroniki i Informatyki
Politechnika Śląska

To nie był protokół Modbus RTU

Słowa kluczowe: PLC, protokół komunikacyjny, ramka protokołu

1. Wstęp

W tomie VIII/2015 niniejszych Zeszytów Naukowych ukazał się artykuł na temat implementacji protokołu Modbus RTU dla komunikacji ze sterownikiem PLC firmy IDEC [0]. Przedstawiony w nim protokół komunikacyjny nie jest jednak protokołem Modbus RTU [0, 0], tylko protokołem firmowym [0].

2. Zaznaczenie granic ramki

W obu protokołach wykorzystywana jest ta sama zasada komunikacji – komputer nadrzędny wysyła do sterownika polecenie, nazywane też zapytaniem lub żądaniem, na które otrzymuje odpowiedź (z jednym wyjątkiem). Inaczej jednak są zaznaczane granice ramek, w których polecenia lub odpowiedzi są przesyłane. Na rysunku 2 w [0] przedstawiona jest ramka z poleceniem odczytu N bajtów danych ze sterownika. Jest ona zakończona znakiem o ustalonym kodzie, mianowicie znakiem CR kodu ASCII (0Dh). Pierwszy znak ramki, nazywany znakiem kontrolnym komunikacji, to także znak o ustalonym kodzie. Dozwolone są trzy takie znaki kodu ASCII: ENQ (05h) rozpoczyna ramkę z poleceniem, ACK (06h) lub NAK (15h) ramkę z odpowiedzią [0].

W protokole Modbus RTU (ang. *Remote Terminal Unit*) znacznikami początku i końca ramki jest stan braku nadawania (stan ciszy w łączu) o czasie trwania równym co najmniej 3.5 * czas trwania transmisji jednego znaku. W praktyce znacznik końca jednej ramki Modbus RTU stanowi zarazem znacznik początku następnej.

3. Zapis poleceń i odpowiedzi w ramce

Jeszcze inne elementy ramki protokołu przedstawionego w [0] świadczą o tym, że nie jest to protokół Modbus RTU. Polecenia i odpowiedzi są zapisywane w ramce

w sposób, który wyklucza pojawienie się znaku o kodzie 0Dh przed znacznikiem końca. Na rysunku 5 we wspomnianym artykule widać, że pomiędzy znacznikami granic ramki znajdują się znaki o kodach ASCII tylko cyfr dziesiętnych oraz kilku liter. Kody liter służą w poleceniach do wskazania operacji (zapis/odczyt) oraz rodzaju danej sterownika. Liczby są przekształcane w teksty o stałej długości, zależnej od tego, ile bitów zajmują i jakie mają być ich reprezentacje. Np. wartości 16-bitowych danych odczytywanych ze sterownika lub do niego wysyłanych są zapisywane w kodzie szesnastkowym, zatem liczbie 1000 odpowiada ciąg znaków o kodach: 30h, 33h, 45h, 38h.

W ramce protokołu Modbus RTU mogą natomiast występować znaki o kodach z zakresu 00h÷FFh, gdyż granice ramki nie są zaznaczane znakami o ustalonych kodach. Liczba 1000 będzie w niej przesłana przy użyciu znaków o kodach: 03h, E8h.

4. Suma kontrolna

W obu protokołach są też stosowane odmienne rodzaje sum kontrolnych.

W protokole firmowym IDEC stosowana jest tzw. kontrola parzystości wzdłużnej (ang. *longitudinal redundancy check*). Wartość sumy kontrolnej oblicza się poprzez wykonanie operacji logicznej XOR dla odpowiadających sobie bitów (tj. bitów o takich samych numerach) wszystkich znaków ramki poprzedzających sumę kontrolną. Wykrywane są błędy polegające na przekłamaniu parzystej liczby bitów.

W protokole Modbus RTU stosuje się sumy kontrolne CRC (ang. *cyclic redundancy check*). Programowe obliczanie ich wartości jest nieco bardziej złożone, ale umożliwiają one wykrycie błędów seryjnych [0].

5. Podsumowanie

Protokół przedstawiony w [0] jest protokołem firmowym, jedynym, który umożliwiał komunikację ze starszymi sterownikami PLC firmy IDEC, już nie produkowanymi. Do komunikacji z nowszymi sterownikami można używać zarówno protokołu firmowego jak i Modbus RTU.

Modbus RTU jest właściwie potoczną nazwą protokołu Modbus, używanego w jednym trybów transmisji ramek. W chwili wprowadzenia protokołu Modbus, to jest w 1979 roku, określone były dwa tryby transmisji: RTU oraz ASCII, później wprowadzono jeszcze tryb TCP [0]. Opis protokołu był jawny, a sam protokół prosty, dlatego szybko zdobył popularność, zaczęli go stosować inni producenci sterowników PLC. Korzysta się z niego do komunikacji też z innymi przyrządami kontrolno-pomiarowymi, np. falownikami.

Bibliografia

1. Halsall F., Data communications, computer networks and open systems, Addison-Wesley, 1992.
2. IDEC, FC5A MicroSmart Maintenance Communication Protocol, http://idea.eko-instal.net/wp-content/uploads/2015/05/protokol_komunikacji.pdf.
3. Miziołek M., Algorytm i implementacja protokołu komunikacyjnego MODBUS w środowisku sterownika PLC firmy IDEC oraz języku programowania C#, Zeszyty Naukowe Wydziału Elektroniki i Informatyki Politechniki Koszalińskiej, nr 8/2015.
4. Modicon, Modicon Modbus Protocol Reference Guide. PI-MBUS-300 Rev, J. http://www.modbus.org.
5. Modbus Organization, Modbus over serial line. Specification and Implementation Guide V1.02, http://www.modbus.org.
6. Modbus Organization, Modbus Messaging on TCP/IP Implementation Guide V1.0b, http://www.modbus.org.

Abstract

In one of previous Research Notes of the Faculty Electronics and Computer Science of the Koszalin University of Technology the article about implementation of Modbus protocol for communication with PLC from one manufacturer was published. The structure of protocol frame, with command for the PLC, was depicted on one of figures. The specific characters were delimiters of the frame. It means that this protocol isn't a Modbus protocol, because other delimiters of the frame are used in the Modbus protocol. Also other elements of the frame are pointing to it.

Streszczenie

W jednym z poprzednich numerów Zeszytów Naukowych Wydziału Elektroniki i Informatyki Politechniki Koszalińskiej ukazał się artykuł na temat implementacji protokołu Modbus dla komunikacji ze sterownikiem pewnej firmy. Na jednym z rysunków została przedstawiona budowa ramki tego protokołu, zawierającej polecenie dla sterownika. Znacznikami początku i końca ramki są znaki o ustalonych kodach. To świadczy o tym, że ten protokół nie jest protokołem Modbus, w którym jest stosowana inna zasada zaznaczania granic ramki. Wskazują na to też inne elementy ramki.