POLITECHNIKA ŚWIĘTOKRZYSKA

Wydział Elektrotechniki, Automatyki i Informatyki

Katedra Elektrotechniki Przemysłowej i Automatyki

Zakład Urządzeń i Systemów Automatyki

Komputerowe Sieci Przemysłowe

Wprowadzenie do sieci przemysłowych. Protokół Modbus.

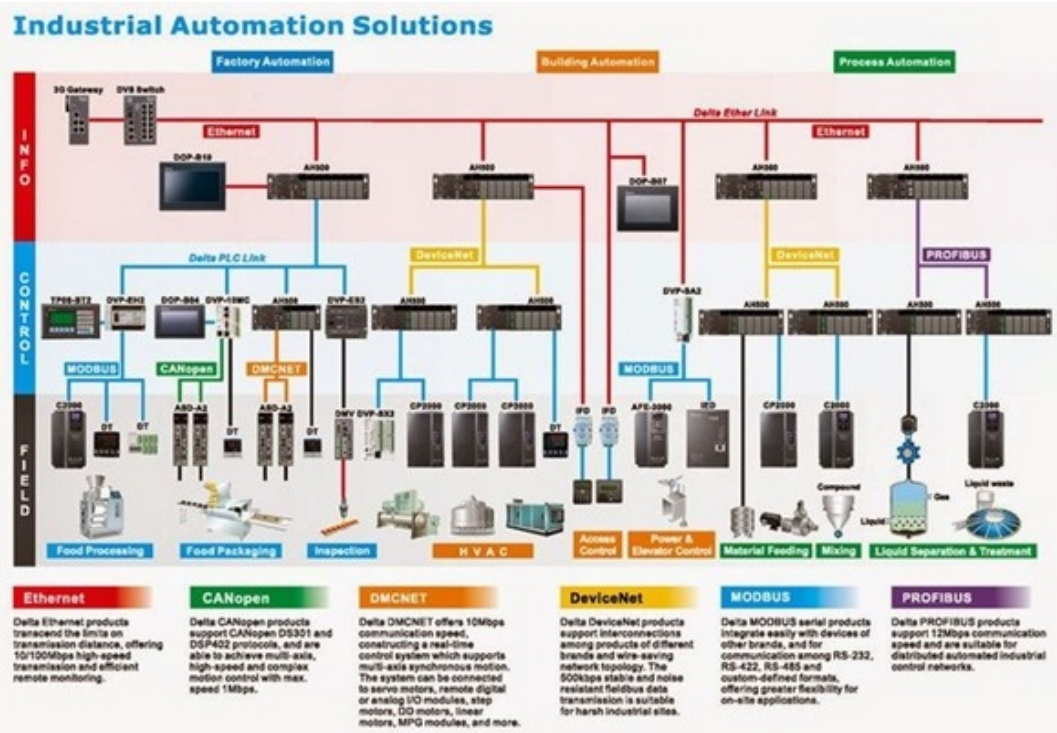
Instrukcja laboratoryjna

Paweł Strączyński

2018

1 Protokoły komunikacyjne wykorzystywane w automatyce przemysłowej

Obecnie w środowisku przemysłowym wykorzystywanych jest wiele protokołów określających zasady wymiany informacji miedzy urządzeniami. Konieczność stosowania skutecznych systemów wymiany informacji jest następstwem zdecentralizowanych systemów sterowania. Aktualnie powszechnie używane w przemyśle urządzenia takie jak sterowniki PLC, panele HMI, falowniki, sterowniki serwomechanizmów itp. wymieniają między sobą informacje w oparciu o komunikację określoną wedle danego protokołu komunikacji. Specyficzne wymagania systemów kontrolno-pomiarowych w przemyśle (niezawodność, szybka, częstą wymiana stosunkowo małych porcji informacji) spowodowały, że ze środowiska tego wyrosły różnego rodzaju protokoły które urosły do rangi przemysłowych standardów. Wśród dużej ilości



Rysunek 1.1: Protokoły komunikacyjne stosowane w przemyśle

protokołów komunikacyjnych wyróżnić można te oparte na Ethernecie jak i wykorzystujące komunikację szeregową w oparciu o standardy RS-422 oraz RS-485. W oparciu o sieć Ethernet pracują m.in. bardzo popularne w przemyśle protokoły takie jak Profinet, Modbus TCP, EtherCAT, Powerlink, DeviceNET, SERCOS III. Do najbardziej popularnych proto-

kołów przemysłowych opartych o komunikację szeregową należą Profibus DP, Modbus RTU, CanOpen, MPI.

2 Protokół Modbus

Protokół Modbus stworzony został w 1979 roku przez firmę Modicon i pierwotnie dedykowany był do komunikacji między sterownikami PLC tego producenta. Jego niezawodność i prostota sprawiły że jest on obecnie w automatyce uznanym standardem i jednym z najchętniej stosowanych protokołów wymiany informacji. Uznanie zyskał ze względu na prostą zasadę wymiany informacji (Master-Slave), wbudowany system potwierdzania komunikatów (Zapytanie-Odpowiedź) oraz kontrole poprawności przesyłanych danych. Wyróżnia się wersje protokołu oparte o łącze szeregowe (Modbus RTU oraz Modbus ASCII) oraz wersję protokołu opartą na Ethernecie - Modbus TCP. Wszystkie odmiany tego protokołu mają wspólne funkcję i rodzaj organizacji pamięci. Różnią się nieznacznie budową ramek.

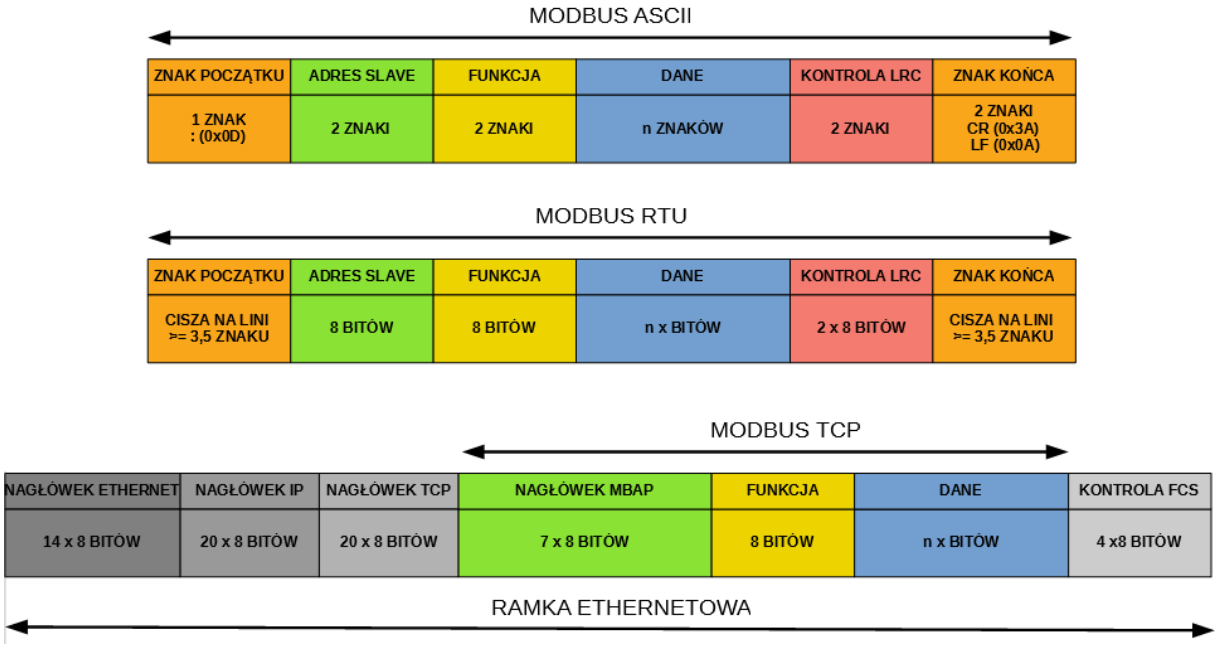
2.1 Budowa ramek

Struktura ramek komunikacyjnych obu szeregowych odmian protokołu (Modbus RTU oraz Modbus ASCII) jest taka sama. Składa się z pola adresowego, pola danych, kontroli błędów oraz znaczników początku i końca ramki. Różnice widoczne są w samym sposobie przesyłania tych informacji. Modbus ASCII to rodzaj protokołu w którym informacje przesyłane są w postaci znaków ASCII - dokładnie każdy bajt informacyjny to dwa znaki ASCII. Protokół dopuszcza stosowanie dużego odstępu czasowego pomiędzy znakami - nawet do 1 sekundy. Ramka komunikacyjna zawiera jasno zdefiniowane znaczniki początku i końca:

- ramka zaczyna się znakiem dwukropka : (0x3A),
- ramka kończy się znakami CR (Carriage Return - 0x0D) oraz LF (Line Feed - 0x0A).

Poprawność przesyłania ramki Modbus ASCII badana jest z wykorzstaniem kodu LRC *(Longitudinal Redundancy Check)*. Wadą transmisji w trybie ASCII jest nadmiarowość danych koniecznych do przesłania tej samej informacji względem trybu RTU. W protokole Modbus RTU informacje przesyłane są w postaci binarnej. Ramki rozpoczyna się zwłoką czasową co najmniej 3,5 razy dłuższą od czasu trwania pojedynczego znaku. Przerwy pomiędzy nadawaniem kolejnych znaków nie mogą być dłuższe niż 1,5 czasu trwania każdego z nich znaku. Do weryfikacji poprawności przesyłanych informacji wykorzystywana jest suma kontrolna CRC

(Cyclic Redundancy Check) - 16-bitowe słowo, przesyłane w postaci dwóch bajtów na końcu ramki. W przypadku obu odmian protokołu słowo kontrolne jest wyliczane przez urządzenie nadawcze i odbiorcze oraz porównywane. Ewentualne rozbieżności świadczą o występujących przekłamaniach w transmitowanych danych. W przypadku protokołu Modbus TCP ramka protokołu jest osadzona w standardowej ramce ethernetowej. Budowę ramek komunikacyjnych przedstawiono na rysunku poniżej.



Rysunek 2.1: Budowa ramek protokołu Modbus

2.2 Rodzaje transakcji

Komunikacja w ramach protokołu Modbus zawsze inicjowana jest przez urządzenie typu Master. Transakcje można podzielić na:

- transakcja typu zapytanie - odpowiedź (*Unicast*),
- transakcja typu rozgłoszenie (*Broadcast*).

W transakcji typu zapytanie-odpowiedź urządzenie typu Master łączy się z konkretnym urządzeniem typu Slave. W przypadku poleceń dotyczących odczytu stanu bitów lub rejestrów w odpowiedzi Slave odsyła odpowiednie informacje. W przypadku wysłania przez urządzenie

Master polecenia zapisu do rejestrów, Slave po wykonaniu operacji jako potwierdzenie odsyła ramkę do urządzenia typu Master. Przy zapisie większej liczby informacji jest to ramka skrócona zawierająca ilość zapisywanych rejstrów oraz adres pierwszego z nich. Drugi typ transakcji, a więc transakcja typu rozgłoszeniowego adresowana jest do wszystkich urządzeń na linii (zarezerwowanym adresem rozgłoszeniowym jest 0). Slave'y po wykonaniu poleceń otrzymanych przez Master'a nie wysyłają ramki zwrotnej.

2.3 Funkcje i typy obszarów pamięci

Wymiana danych w ramach protokołu odbywa się według ściśle zdefiniowanych funkcji. Te najczęściej stosowane przedstawiono w tabeli poniżej. Odnoszą się one bezpośrednio do

Tabela 2.1: Wybrane funkcje protokołu Modbus

Kod funkcji	Rodzaj operacji
1	Odczyt wyjść binarnych *Read Coils*
2	Odczyt wejść binarnych *Read Discrete Inputs*
3	Odczyt n rejestrów wyjściowych - *Read Holding Register*
4	Odczyt n rejestrów wejściowych - *Read Input Register*
5	Zapis pojedynczego wyjścia binarnego *Write Single Coil*
6	Zapis pojedynczego rejestru wyjściowego - *Write Single Register*
7	Odczyt statusu - *Read Exception Status*
8	Test diagnostyczny - *Diagnostics*
15	Zapis n bitów - *Write Multiple Coils*
16	Zapis n rejestrów wyjściowych - *Write Multiple Register*

odczytu i zapisu jednego lub wielu obszarów pamięci danego typu. Więcej informacji na temat

Tabela 2.2: Typy zmiennych protokołu Modbus

Typ zmiennej	Rodzaj	Dostęp	Zastosowanie
Discretes Inputs	1-bit	Odczyt	Dwustanowe wejście
Coils	1-bit	Odczyt/Zapis	Dwustanowe wyjście
Input Register	16-bitowe słowo	Odczyt	Rejestr wejściowy
Holding Register	16-bitowe słowo	Odczyt/Zapis	Rejestr wyjściowy

Modbus'a można znaleźć w dokumentacji protokołu.

3 Zadania do wykonania

Zadanie 1 Korzystając z protokołu Modbus TCP nawiązać połączenie pomiędzy dwoma sterownikami PLC Siemens S7-1200. Utworzyć Slave'a protokołu Modbus TCP (*Modbus Server*) na sterowniku jednym sterowniku PLC oraz Master'a protokołu (*Modbus Client*) na drugim zgodnie z podręcznikiem.

Zadanie 2 Wykorzystując program qModMaster nawiązać komunikację ze sterownikiem PLC. Przetestować działanie funkcji 3,6,10 protokołu Modbus. Zarejestrować ramki komunikacyjne z wykorzystaniem programu Wireshark.

Zadanie 3 W oparciu o protokół Modbus TCP nawiazac połaczenie pomiedzy sterownikiem PLC Siemens S7-1200, a aplikacja Promotic SCADA. Utworzyc Slave'a protokołu Modbus TCP (Modbus Server) na sterowniku PLC . W aplikacji SCADA uzyc obiektu PmModbusMr. Zarejestrowac ramki komunikacyjne z wykorzystaniem programu Wireshark.

Zadanie 4 W środowisku Promotic SCADA/LabView utworzyć aplikację sterującą robotem mobilnym. Do komunikacji z układem napędowym robota wykorzystać protokół Modbus RTU. Poniżej przedstawiono parametry transmisji oraz organizację pamięci MB robota.

Adres urządzenia	*1*
Prędkość transmisji	*9600*
Ilość bitów danych	*8*
Bit parzystości	*nie*
Liczba bitów stopu	*1*

Nr bitu/rejestru	Adres	Typ pamięci	Przeznaczenie
10001	0	Coil	*IN1* - silnik lewy
10002	1	Coil	*IN2* - silnik lewy
10003	2	Coil	*IN1* - silnik prawy
10004	3	Coil	*IN2* - silnik prawy
40001	0	Holding Register	*PWM* - silnik lewy
40002	1	Holding Register	*PWM* - silnik prawy

Zadanie 5 Wykorzystując analizator stanów logicznych przeanalizować transakcje nawiązywane pomiędzy programem qModMaster, a robotem mobilnym. Zbadać urządzenie typu Slave (robot mobilny) pod kątem obsługi funkcji 1,3,5,6,15,16 protokołu Modbus. Zaobserwować nawiązywane transakcję typu rozgłoszeniowego.