

Bezpieczeństwo infrastruktury krytycznej i systemów sterowania przemysłowego IoT - Laboratorium 7 – ICS Security Intro

Forma: praca w samodzielna / małe zespoły (2 - 4 osoby)

Wstęp teoretyczny

ICS

<https://www.paloaltonetworks.ca/cyberpedia/what-is-ics-security>

<https://www.techtarget.com/whatis/definition/industrial-control-system-ICS>

Industrial Control System (ICS) to ogólne określenie na zestaw urządzeń i oprogramowania, które sterują procesami przemysłowymi i infrastrukturą krytyczną: od czujników przy rurze, przez sterowniki i sieć, aż po stanowisko operatora z HMI. To nie jest jeden konkretny system, tylko cała infrastruktura, która zbiera dane z procesu, podejmuje decyzje i wysyła sygnały sterujące do maszyn.

Co dokładnie nazywamy ICS

Według Palo Alto Networks ICS to kombinacja sprzętu i oprogramowania służąca do zarządzania i automatyzacji procesów przemysłowych. Wykorzystuje się je w bardzo różnych branżach, między innymi w produkcji, chemii, przemyśle naftowym, telekomunikacji, sektorze spożywczym, automotive i farmacji. Dodatkowo ICS stanowią główny punkt sterowania infrastrukturą krytyczną, takiej jak sieci energetyczne, transport publiczny, oczyszczalnie ścieków, przepompownie, rurociągi gazowe i naftowe.

Typowy ICS składa się z:

- czujników i urządzeń polowych, które mierzą rzeczywiste parametry procesu (temperatura, ciśnienie, przepływ, poziom, napięcie itd.)
- sterowników i systemów sterowania (PLC, RTU, DCS, SCADA), które przetwarzają dane i podejmują decyzje sterujące
- urządzeń wykonawczych, takich jak zawory, pompy, styczniki, napędy
- sieci komunikacyjnej oraz serwerów, baz danych i stacji operatorskich, które spajają to wszystko w całość

ICS działa w pętli sprzężenia zwrotnego. Dane z czujników trafiają do sterownika, ten przelicza je według zaprogramowanej logiki i steruje aktuatorami, a operator obserwuje całość na HMI i może ręcznie korygować parametry. Celem jest automatyzacja, optymalizacja procesu oraz utrzymanie bezpieczeństwa ludzi, instalacji i środowiska.

Jak zmienia się profil zagrożeń ICS

Kiedyś ICS funkcjonowały jako systemy zamknięte, praktycznie odseparowane od świata zewnętrznego. Dziś są coraz mocniej połączone z sieciami IT i różnymi systemami zewnętrznymi, głównie przez: IIoT, zdalne utrzymanie ruchu, raportowanie produkcji do systemów ERP, integrację z chmurą.

To daje korzyści organizacyjne, ale z punktu widzenia bezpieczeństwa oznacza między innymi:

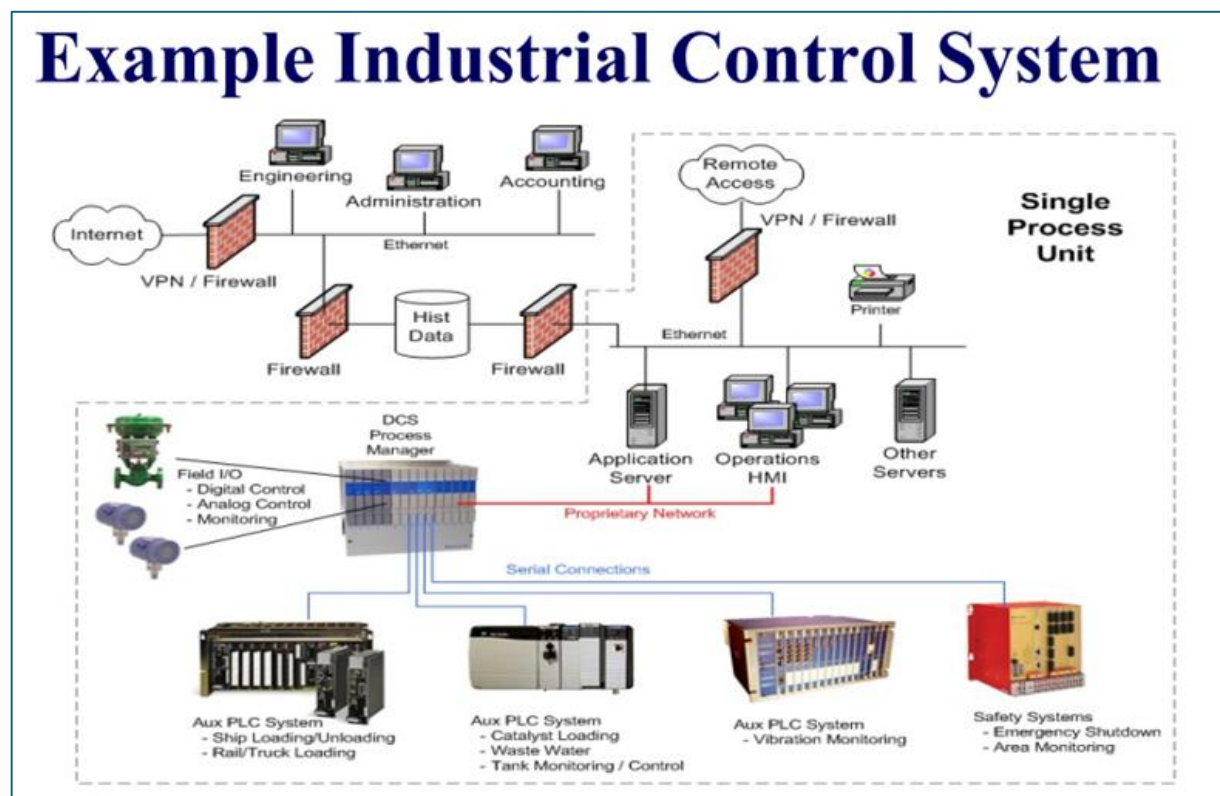
- więcej potencjalnych punktów wejścia dla atakującego (HMI, serwery, bramy komunikacyjne, zdalne VPN)
- przenoszenie się klasycznych ataków IT na środowisko OT (ransomware, malware na Windows)
- większą podatność na błędy konfiguracji, słabe hasła, nieaktualne oprogramowanie

Palo Alto wprost wskazuje, że często punktem wejścia są słabo zabezpieczone połączenia między światem IT a ICS, na przykład brak szyfrowania, niepoprawna kontrola sesji, brak segmentacji.

Typowe ataki na ICS według Palo Alto

- ataki pośrednie, które zaczynają się w sieci IT, a dopiero potem OT (np. NotPetya rozprzestrzeniający się z systemów biurowych do części produkcyjnej)
- ataki bezpośrednie na komponenty ICS, takie jak PLC i RTU, często ukierunkowane na sabotaż procesu lub wyłączenie systemów bezpieczeństwa
- ataki typu DoS i DDoS, które przeciążają sieć lub urządzenia i doprowadzają do opóźnień oraz przerw w działaniu
- manipulacja danymi i komendami, która wygląda na normalną pracę, ale w rzeczywistości zmienia parametry procesu (np. fałszywe wartości ciśnień lub stanów zaworów)

wykorzystywanie podatności w systemach starszej generacji, które trudno zaktualizować i które pierwotnie nie były projektowane z myślą o cyberzagrożeniach



PLC

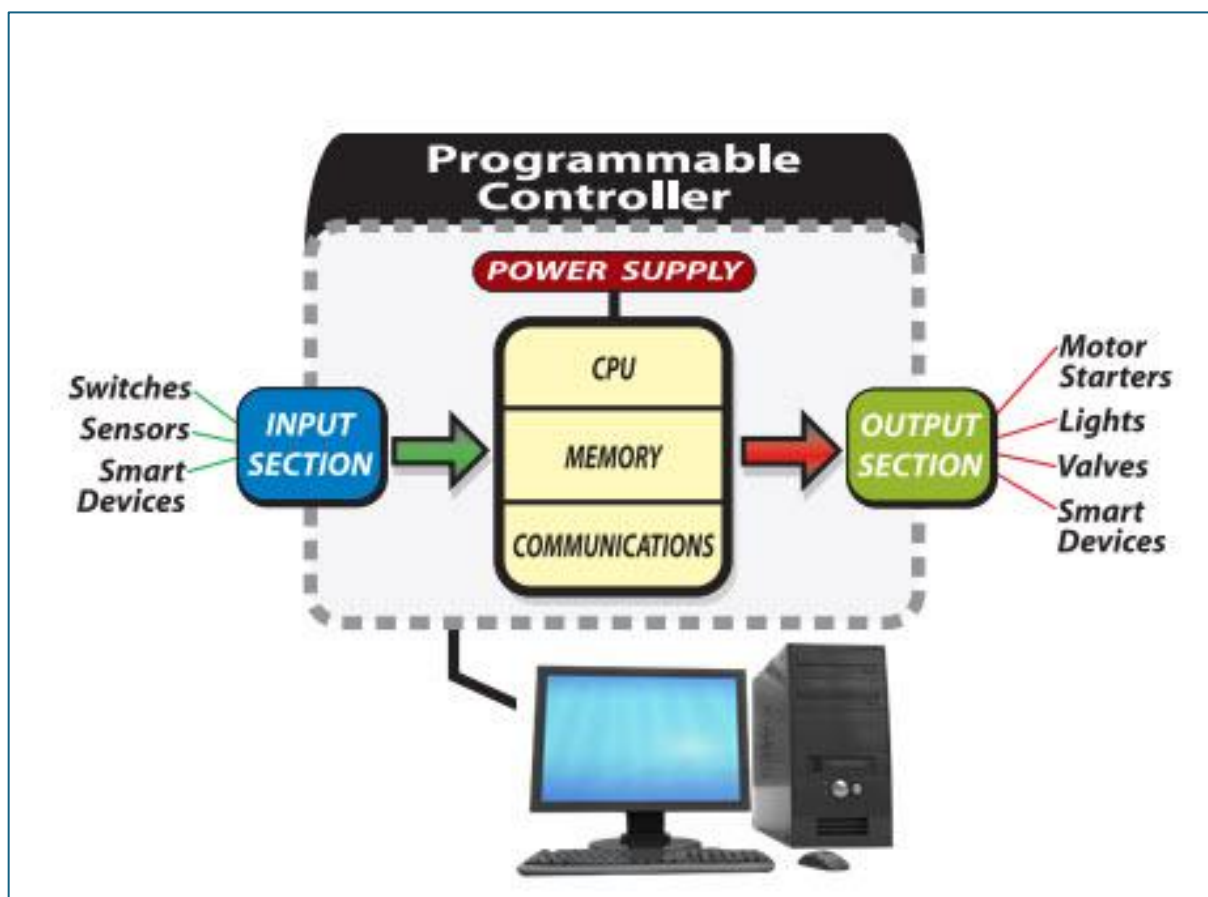
<https://www.astor.com.pl/poradnikautomatyka/podstawa-systemow-automatyki-czym-jest-sterownik-plc/>

Sterownik PLC jest jednym z najważniejszych urządzeń używanych w automatyce przemysłowej. Do jego powstania przyczyniła się ewolucja automatyki przemysłowej. Dzisiaj ciężko wyobrazić sobie zautomatyzowane procesy technologiczne bez sterowników PLC. Czym jednak właściwie one są i do czego służą? Co sprawia, że zastosowanie sterowników PLC w systemie automatyzacji jest korzystne, a często nawet nieodzowne? .

Czym jest sterownik PLC i do czego jest wykorzystywany?

Skrót PLC pochodzi od Programmable Logic Controller, czyli Programowalny Sterownik Logiczny. Jest to urządzenie wykorzystujące układ mikroprocesorowy, które służy do sterowania pracą maszyny lub innego urządzenia stosowanego w przemyśle. Używane może być do sterowania liniami produkcyjnymi, oświetleniem oraz innymi urządzeniami elektrycznymi.

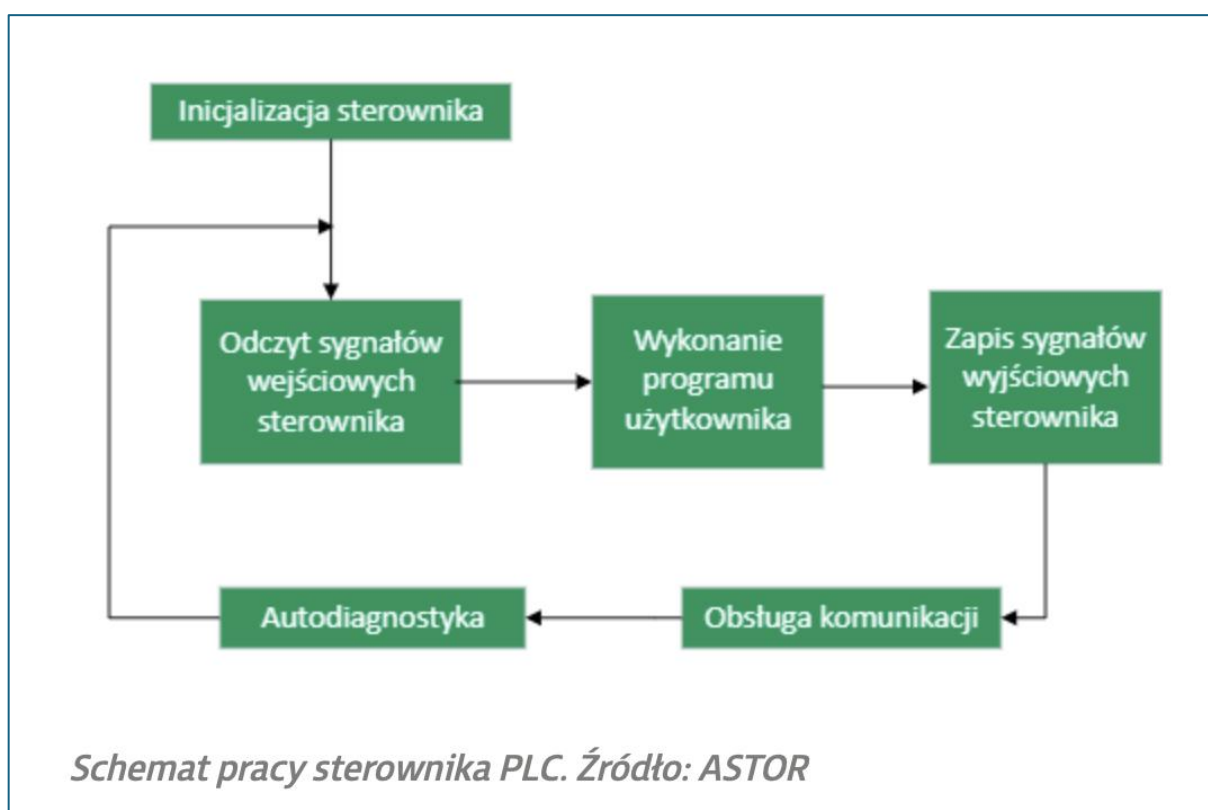
Sterownik PLC został skonstruowany, aby zastąpić układy stycznikowo-przełącznikowe. Pozwala on wyeliminować oraz uprościć skomplikowane okablowanie pomiędzy stycznikami i przetwornikami. Wszystkie działania realizowane przez układ sterowania, takie jak włączanie i wyłączenie urządzeń, zadawanie wartości parametrów, zliczanie czasu, a nawet obsługa receptur, są wykonywane wewnątrz sterowników. Dołącza się zwykle do nich odpowiednią liczbę układów wejściowych oraz wyjściowych. Układy wejść służą do zbierania informacji o stanie sterowanego obiektu, a układy wyjść – do połączeń z elementami sygnalizacyjnymi, wykonawczymi lub do transmisji danych.



Sterowniki PLC są powszechnie używane w praktycznie każdej gałęzi przemysłu – ze względu na uniwersalność tego urządzenia. Sterownik ma też wiele innych zastosowań, nawet w życiu codziennym – na przykład do sterowania sygnalizatorami świetlnymi, ruchomymi schodami, windami czy różnymi systemami, w jakie wyposażone są nowoczesne inteligentne domy.

Jak działa sterownik PLC?

Sterownik PLC pracuje w zamkniętej pętli programowej, czyli wykonuje program, który zawiera zapętłony ciąg rozkazów. Dzięki temu sterownik może reagować na dostarczane do niego informacje. Program sterownika składa się ze zmiennych oraz rozkazów, wśród których znajdują się operacje logiczne, arytmetyczne i wiele innych, a także polecenia związane z odczytem lub zapisem danych. Aby przygotować program, musisz to zrobić za pomocą komputera lub programatora. Gdy już to zrobisz, program jest przekształcany do postaci zrozumiałej dla układu mikroprocesorowego, przesyłany i zapisywany w pamięci sterownika. Sterowniki otrzymują informację za pośrednictwem swoich wejść, które mogą być cyfrowe (zwane też dyskretnymi lub binarnymi) lub analogowe. Do wejść mogą być podłączone różnego typu czujniki np. zbliżeniowe. Z kolei sterowanie (wysyłanie sygnałów sterujących) odbywa się za pośrednictwem wyjść, które również mogą być binarne lub analogowe. Dodatkowo sterownik może wymieniać dane z otoczeniem za pomocą portów komunikacyjnych pracujących w rozmaitych standardach (szeregowych, Ethernet itp.) oraz różnych protokołach komunikacyjnych. W przypadku nowszych urządzeń praca sterownika może być kontrolowana z poziomu przeglądarki internetowej w komputerze bądź smartfonie.



Powyższy schemat blokowy ilustruje podstawową zasadę działania sterownika PLC. Realizacja programu składa się z następujących etapów:

- Inicjalizacja sterownika – przy każdym uruchomieniu sterownika następuje sprawdzenie poprawności działania
- Odczyt sygnałów wejściowych sterownika – pętla programu w sterowniku analizuje sygnały wejściowe i kopiuje ich wartości do pamięci.
- Wykonanie programu użytkownika – sterownik przetwarza program, realizując kolejno wszystkie rozkazy, wykonując obliczenia i zapisując w pamięci stany sygnałów wyjściowych.
- Zapis sygnałów wejściowych sterownika – wszystkie zapisane w pamięci w trakcie przetwarzania programu stany wyjściowe są ustawiane w postaci odpowiednich sygnałów na wyjściach fizycznych sterownika.
- Obsługa komunikacji – obsługa portów komunikacyjnych, wysyłanie i odbiór informacji do innych urządzeń, jeżeli są połączone.
- Autodiagnostyka – sterownik zbiera informacje o błędach. W przypadku wykrycia błędu, sterownik przerywa działanie.

HMI

<https://automatykaonline.pl/Artykuly/Komputery-i-HMI/Wszystko-o-HMI-Human-Machine-Interface-co-automatyk-powinien-wiedziec>

HMI (ang. Human-Machine Interface) to przemysłowy interfejs między maszyną lub procesem a operatorem, który go obsługuje. Dzięki HMI osoba odpowiedzialna za realizację zadań produkcyjnych może wpływać na przebieg procesu i go kontrolować. Wyobraź sobie, że wchodzisz do otwartej windy. Czego szukasz w pierwszej chwili? Oczywiście popularnych „guzików”, czyli panelu sterowania, który w tym wypadku jest niczym innym jak HMI pomiędzy procesem jazdy windą a jego operatorem, a więc Tobą.

Kiedyś popularne były głównie przyciski, dziś coraz częściej możemy spotkać w tej funkcji np. interaktywne panele operatorskie.



Sterowanie przyciskami w windzie, źródło: ASTOR Technology Park



Panel HMI w windzie, źródło: ASTOR Technology Park

SCADA

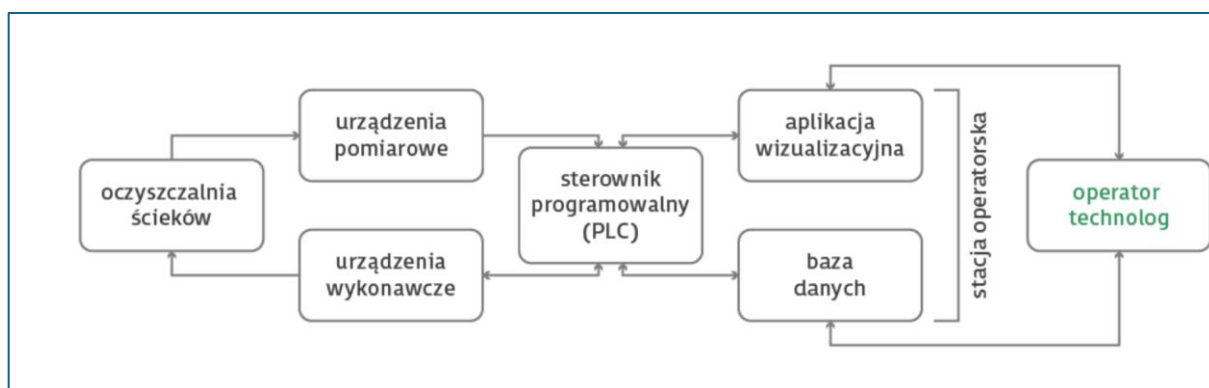
<https://www.astor.com.pl/poradnikautomatyka/co-to-jest-scada/>

SCADA to system komputerowy (najczęściej komputer PC + oprogramowanie), którego celem jest zwiększenie wydajności produkcji. Nie tylko zmienia język maszyn na język ludzi, ale również automatycznie reaguje na sygnały z urządzeń. Dzięki SCADA możliwe jest zbieranie danych z maszyn i urządzeń pomiarowych w czasie rzeczywistym, co umożliwia nadzór nad procesem produkcyjnym. Kolejną, bardzo ważną funkcją systemu SCADA jest wizualizacja aktualnych danych lub danych historycznych.



System SCADA pozwala użytkownikowi na sterowanie procesem produkcyjnym przez zadawanie parametrów za pomocą panelu lub z poziomu aplikacji komputerowej. Umożliwia także wykrywanie sytuacji alarmowych i informuje o nich operatorów, dzięki czemu możliwa jest szybka reakcja na błędy i nieprawidłowości. Ponadto SCADA potrafi archiwizować dane z procesu produkcyjnego – w postaci plików lub w bazie danych.

Zadaniem SCADA jest usprawnienie procesu produkcyjnego, więc jej działanie opiera się na współpracy z elementami automatyki zaimplementowanymi w procesie wcześniej. SCADA odnajduje swoje miejsce pomiędzy urządzeniami sterującymi, pomiarowymi i wykonawczymi (takimi jak sterowniki PLC, moduły I/O, czujniki i liczniki), a operatorem maszyn. Działa w łańcuchu urządzeń, pełniąc nadrzędną rolę. Dodatkowo może integrować wiele sterowników PLC.

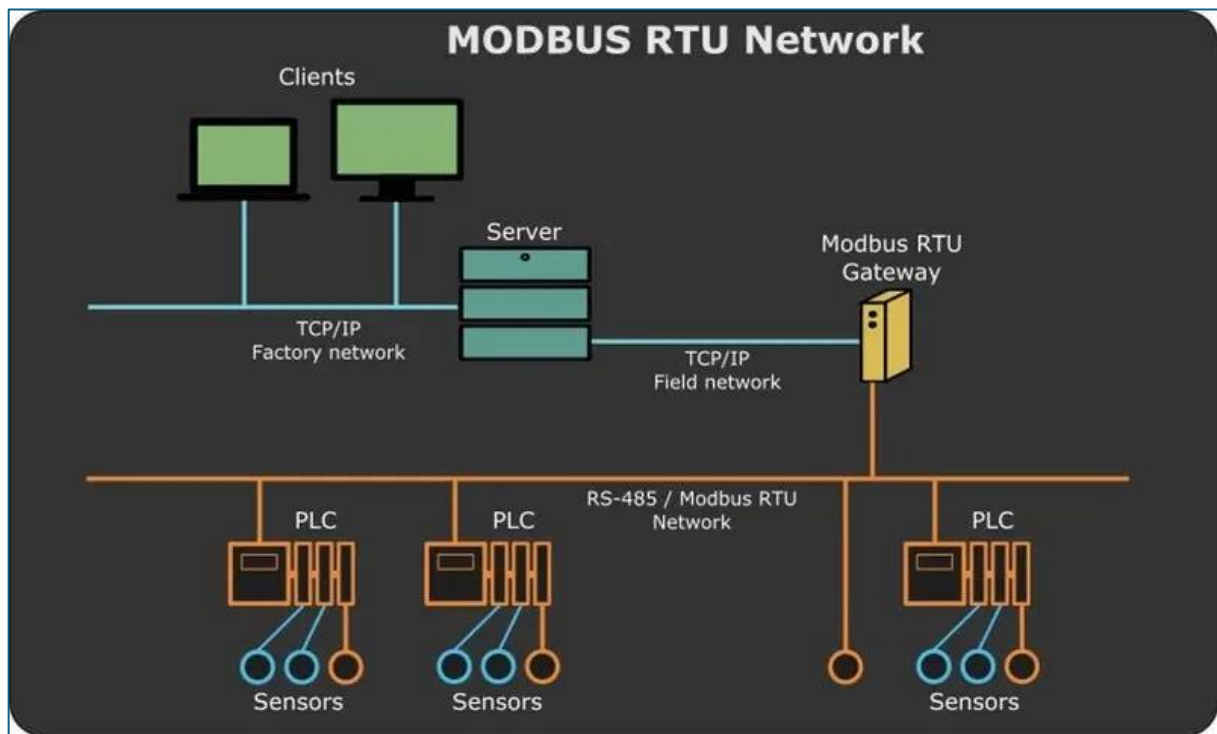


Modbus

https://www.csimn.com/CSI_pages/Modbus101.html

Szczegółowy opis: <https://ntronic.pl/jak-dziala-modbus/>

Protokół Modbus RTU to prosty, stary, ale wciąż bardzo żywy standard komunikacji w automatyce przemysłowej. Służy do wymiany danych pomiędzy jednym urządzeniem nadrzędnym (Master) a wieloma urządzeniami podrzędnymi (Slave) po wspólnej magistrali najczęściej RS-485. W opisywanym tekście autor krok po kroku pokazuje zarówno teorię (ramki, kody funkcji, typy danych), jak i praktykę na przykładzie przetworników TD2 oraz pracy z programem testowym na PC.



Modbus powstał w 1979 roku w firmie Modicon (dziś Schneider Electric) jako prosty, otwarty i darmowy protokół do komunikacji między sterownikami a urządzeniami obiektowymi. Mimo wieku ponad 40 lat nadal jest powszechnie używany, bo:

- jest łatwy w implementacji,
- ma prostą ideę działania,
- jest tani,
- ma mało ograniczeń co do rodzaju danych i urządzeń.

Może pracować na różnych mediach transmisyjnych: skrętce TP, RS-232/RS-485, Ethernet, modemach telefonicznych, sieciach radiowych czy GSM. Dzięki temu przy modernizacji często da się wykorzystać istniejącą instalację (np. starą skrętkę RS-485), a z drugiej strony bez problemu przenieść komunikację w świat TCP/IP. Pierwsze wdrożenia bazowały na RS-232, później protokół przystosowano do RS-485, co pozwoliło zwiększyć zasięg i prędkość transmisji.

Architektura Master-Slave i typowe urządzenia

Modbus RTU opiera się na modelu Master-Slave:

- **Master** – jedno urządzenie w sieci, inicjuje wszystkie zapytania (np. PLC, system DCS, RTU, komputer PC).
- **Slave** – do 247 (czasem mówi się o 255) urządzeń podrzędnych, odpowiadają tylko na zapytania Mastera.

Slave'ami są zazwyczaj przetworniki, moduły I/O, liczniki, mierniki itp. Autor opisuje m.in.:

- **TD2** – przetwornik do odczytu temperatury i wilgotności z czujników w sieci OneWire (do ~400 m), komunikacja po RS-485 i USB, galwaniczna separacja.
- **TD1.01** – przetwornik temperatury z wyświetlaczem LED, obsługujący do 64 czujników DS18B20, odczyt po RS-485 przez Modbus RTU.

Te urządzenia pełnią rolę Slave'ów i są naturalnymi partnerami dla sterowników PLC.

Odmiany protokołu: ASCII, RTU, TCP

- **Modbus ASCII** - Dane są zapisane hexem, ale wysyłane jako znaki ASCII. Każdy bajt danych staje się dwoma bajtami transmisji, co czyni ten wariant najwolniejszym. Plusem jest dobra odporność na opóźnienia i „dziwne” media (modemy, transmisje radiowe), bo ramka jest wyraźnie odseparowana znakami.
- **Modbus RTU** - Dane są wysyłane binarnie. Jeden bajt danych = jeden bajt w ramce. Idealny do RS-232/RS-485, obsługuje typowe prędkości 1200–115200 bit/s (w praktyce często 9600 lub 19200). To właśnie RTU jest najpopularniejszym wariantem w klasycznej automatyce.
- **Modbus TCP** - Ten sam protokół, ale „opakowany” w TCP/IP, z adresowaniem IP zamiast klasycznych adresów sieciowych PLC. Dzięki temu każda sieć Ethernet może stać się nośnikiem komunikatów Modbus.

Logika komunikatów jest wspólna różni się tylko sposób kodowania i transportu.

Jak wygląda rozmowa Master-Slave?

Master musi znać adresy wszystkich Slave'ów. Wysyła do magistrali ramkę zawierającą:

1. adres urządzenia,
2. kod funkcji (co ma zrobić Slave),
3. pole danych (adresy rejestrów, ilości, wartości),
4. sumę kontrolną CRC.

Ramka dociera do wszystkich urządzeń, ale odpowiada tylko ten Slave, którego adres zgadza się z polem adresowym.

Adres **0** jest zarezerwowany dla broadcastu wtedy wszystkie Slave'y odbierają polecenie, ale żadne nie odpowiada (Master nie dostaje potwierdzenia).

Jeśli wystąpi zakłócenie:

- brak ciągłości ramki (przerwa dłuższa niż dopuszczalna) → Slave uznaje ramkę za uszkodzoną i milczy,
- zmiana choćby jednego bitu → błędne CRC → ramka jest odrzucana, brak odpowiedzi.

Master po wysłaniu zapytania czeka na odpowiedź przez określony czas (time-out). Jeśli jej nie ma albo CRC się nie zgadza, może wysłać zapytanie ponownie. Czas odpowiedzi trzeba dobrać tak, żeby zdążył odpowiedzieć najwolniejszy Slave w sieci.

Materiały dodatkowe:

- <https://sansorg.egnyte.com/dl/CCyBv7Dpvy4Y>
- <https://sansorg.egnyte.com/dl/Phkkmc9gpxTx>
- <https://sansorg.egnyte.com/dl/twyQtQDccmMC>
- <https://sansorg.egnyte.com/dl/2eHmlaB9Bv>
- <https://sansorg.egnyte.com/dl/4EzgHqHtec>
- <https://sansorg.egnyte.com/dl/3n7ryfl6QA>
- <https://sansorg.egnyte.com/dl/BudUKRmBKT>
- <https://sansorg.egnyte.com/dl/MliqyUJreb>
- <https://sansorg.egnyte.com/dl/TB2lOIInS0>
- <https://sansorg.egnyte.com/dl/qEz7R3zvwH>
- <https://github.com/ITI/ICS-Security-Tools>
- <https://github.com/miguelob/ICS-Hacking>
- <https://github.com/hslatman/awesome-industrial-control-system-security>
- <https://github.com/w3h/icsmaster>
- <https://github.com/vatsalgupta67/All-In-One-CyberSecurity-Resources?tab=readme-ov-file>
- <https://instrumentationtools.com/free-industrial-control-system-ics-cyber-security-training-course/>
- <https://github.com/neutrinoguy/awesome-ics-writeups>
- <https://arxiv.org/pdf/2001.02925>

Zadania praktyczne

Zadanie 1 – Wireshark Investigation

Pobierz plik PCAP i przeanalizuj pakiety modbus za pomocą Wireshark: <https://github.com/ITI/ICS-Security-Tools/tree/master/pcaps/ModbusTCP> lub uruchom cloudshark: <https://www.cloudshark.org/captures/76038eaa4a3b>

Odpowiedz na pytania:

- Which IP address is the master on?
- How many slaves is the master talking to?
- Is the master writing any data to the slaves?
- Does the traffic spike in the middle related to modbus?

Inne pliki PCAP do analizy innych protokołów: <https://github.com/ITI/ICS-Security-Tools/tree/master/pcaps>

Zadanie 2 – Tryhackme Attacking ICS 1/2

Wykonaj następujące zadania na platformie Tryhackme:

- <https://tryhackme.com/room/attackingics1>
- <https://tryhackme.com/room/attackingics2>

Zadanie 3 - Labshock

Zainstaluj i uruchom środowisko labshock następnie zapoznaj się z panelem:

<https://github.com/zakharb/labshock?tab=readme-ov-file>

Install guide: <https://github.com/zakharb/labshock/wiki/Quickstart-Guide>

Example: <https://medium.com/@josegpach/hands-on-ot-security-building-and-breaking-with-labshock-f99f21af2a96>