

Marcin Podsiadły

Automatyka przemysłowa dla początkujących



Poznaj sterowniki PLC oraz panele HMI
i naucz się je programować

Helion 

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Małgorzata Kulik

Projekt okładki: Studio Gravite/Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki

Grafika na okładce została wykorzystana za zgodą AdobeStock.com.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl
WWW: helion.pl (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
helion.pl/user/opinie/wpaupr
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-289-1457-5

Copyright © Helion S.A. 2025

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Od autora	7
Wprowadzenie	9
ROZDZIAŁ 1. Automatyka jako rozległa dziedzina	13
1.1. Technologia procesu	15
1.2. Mechanika	16
1.3. Elektryka	20
1.4. Programowanie PLC	27
1.5. Wizualizacja procesu — panel operatorski HMI	33
1.6. Wizualizacja procesu — systemy SCADA	34
1.7. Integracja układów automatyki — urządzenia przemysłowe	36
1.8. Integracja układów automatyki — systemy informatyczne	42
1.9. Programowanie w automatyce a programowanie w IT	44
1.10. Podsumowanie	45
ROZDZIAŁ 2. Na czym polega programowanie PLC?	48
2.1. Przegląd rodziny sterowników PLC Siemens SIMATIC	50
2.2. Budowa sterowników PLC	55
2.2.1. S7-1200 G1	56
2.2.2. S7-1200 G2	59
2.2.3. S7-1500	60
2.2.4. Generacje urządzeń	63
2.3. Zasada działania sterownika PLC	66
2.3.1. Dostępne obszary pamięci sterowników SIMATIC	68
2.4. Pozostałe możliwości sterowników PLC	73
2.4.1. Regulacja PID	74
2.4.2. Sterowanie osiami	76
2.4.3. Sterowanie PWM	77
2.4.4. Szybkie wejścia	78
2.4.5. Specjalistyczne bloki programowe do obsługi zdarzeń cyklicznych i sprzętowych	79
2.4.6. Zaawansowana diagnostyka programowa	81
ROZDZIAŁ 3. Samodzielna nauka programowania PLC	83
3.1. Oprogramowanie TIA Portal	83
3.2. STEP7	84
3.3. WinCC	85
3.4. Wymagane licencje, wersja próbna	86
3.4.1. Licencje do programowania PLC	86
3.4.2. Licencje do programowania paneli HMI oraz systemów SCADA	87
3.4.3. Zakup licencji	89
3.5. Czy do nauki potrzebuję fizycznego sterownika?	89

3.6. Instalacja oprogramowania	90
3.6.1. Platforma — instalacja TIA Portal z symulatorem PLCSIM	90
3.6.2. Platforma — instalacja Factory I/O	90
3.7. Czy wersja oprogramowania ma znaczenie dla początkującego programisty?	90
ROZDZIAŁ 4. TIA Portal — konfiguracja sterownika PLC	92
4.1. Adres IP i maska podsieci	92
4.2. Nazwa PROFINET	95
4.3. Bity systemowe	96
4.4. Platforma — konfiguracja sprzętowa	98
ROZDZIAŁ 5. Przegląd języków programowania PLC	99
5.1. Język drabinkowy LAD	101
5.2. Język FBD	102
5.3. LAD czy FBD?	103
5.4. Język GRAPH	103
5.5. Język SCL	104
5.6. Język STL	105
5.7. Pozostałe języki programowania PLC	106
5.8. Od którego języka zacząć naukę programowania?	108
ROZDZIAŁ 6. Programowanie PLC — podstawowe operacje logiczne	109
6.1. Deklarowanie zmiennych — tagi	109
6.2. Dobre praktyki — nazewnictwo zmiennych	111
6.3. Styk NO, styk NC i cewka wyjściowa	113
6.4. Platforma — tabela tagów i pierwszy program	116
6.5. Funkcje logiczne AND, OR oraz XOR	117
6.6. Platforma — funkcje logiczne AND & OR	119
6.7. Organizacja kodu — funkcje FC i bloki funkcyjne FB	119
6.7.1. Tworzenie uniwersalnych programów	120
6.7.2. Organizowanie kodu z wykorzystaniem FC/FB	122
6.7.3. Utworzenie i wywołanie FC/FB	123
6.8. Platforma — wykorzystanie funkcji FC	125
6.9. Ustawianie i kasowanie pojedynczej zmiennej — cewki Set oraz Reset	125
6.10. Przerzutniki SR/RS	127
6.11. Detekcja zbocza	129
6.12. Bloki danych DB	133
6.13. Platforma — przerzutniki SR/RS, bloki danych DB, narzędzie Cross-references	137
ROZDZIAŁ 7. Typ zmiennej a jej rozmiar	138
7.1. Bloki danych DB czy pamięć systemowa M — porównanie	141
7.2. Tabela Watch table — narzędzie do monitorowania i modyfikowania zmiennych	146
7.3. Przekazywanie wartości parametrów — funkcja MOVE	147
7.4. Platforma — obsługa typów danych z wykorzystaniem funkcji MOVE	149
ROZDZIAŁ 8. Forsowanie zmiennych	151

ROZDZIAŁ 9. Dobre praktyki przy tworzeniu oprogramowania PLC	154
9.1. Wybrane koncepty projektowe	155
9.1.1. KISS	155
9.1.2. YAGNI	155
9.1.3. DRY	156
9.1.4. Oczekiwania vs rzeczywistość	157
9.2. Organizacja w środowisku TIA Portal	157
9.2.1. Struktury w bloku danych DB — typ Struct	157
9.2.2. Zmienne użytkownika — struktury PLC data types	160
9.2.3. Foldery — organizowanie bloków i struktur w TIA Portal	163
9.2.4. Platforma — zmienne strukturalne Struct oraz PLC data types	164
ROZDZIAŁ 10. Instrukcje porównania	165
ROZDZIAŁ 11. Funkcje matematyczne	171
11.1. Operacje arytmetyczne	171
11.2. Wyjście ENO	175
11.3. Inkrementacja i dekrementacja	176
11.4. Funkcje zmiennoprzecinkowe	177
11.5. Funkcja CALCULATE	178
11.6. Funkcje MAX, MIN oraz LIMIT	181
11.7. Platforma — operacje porównania i obliczenia matematyczne	183
ROZDZIAŁ 12. Systemowe bloki funkcyjne	184
12.1. Układy zliczające — counters	184
12.2. Platforma — układy licznikowe	188
12.3. Układy czasowe — timery	188
12.3.1. Timer TON	191
12.3.2. Timer TOF	192
12.3.3. Timer TP	193
12.3.4. Resetowanie pracy timerów TOF i TP	194
12.3.5. Timer TONR	194
12.3.6. Pozostałe informacje	195
12.3.7. Tips and tricks — „Timer nie odmierza czasu”	197
12.3.8. Platforma — układy czasowe	198
ROZDZIAŁ 13. Najważniejsze czynności związane z obsługą sterownika PLC	199
13.1. Tworzenie kopii zapasowych	199
13.1.1. Archiwizacja projektu źródłowego	200
13.1.2. Backup online	202
13.2. Pobieranie projektu z urządzenia	204
13.3. Karta pamięci — formatowanie, wgrywanie projektu na kartę	206
13.4. Projekt na komputerze a projekt na maszynie — porównywanie projektów	208
13.5. Ochrona własności intelektualnej — zabezpieczenia funkcji, projektu i urządzenia	211
13.5.1. Zabezpieczenie projektu źródłowego hasłem	212
13.5.2. Dostęp do projektu — Access control	213
13.5.3. Zabezpieczenie danych poufnych	215
13.5.4. Dostęp do urządzenia — wyświetlacz frontowy	215
13.5.5. Dostęp do urządzenia — web server	216
13.5.6. Zabezpieczenia bloków programowych — Know-how protection, Write protection, Copy protection	218
13.5.7. Blokada zapisu na karcie pamięci SMC	220
13.5.8. Mechaniczna blokada wyświetlacza frontowego	220

ROZDZIAŁ 14. Na czym polega programowanie HMI?	222
14.1. Dlaczego WinCC Unified?	224
ROZDZIAŁ 15. Budowa paneli HMI	226
15.1. Rodzina WinCC	226
15.1.1. WinCC Basic	227
15.1.2. WinCC Comfort	227
15.2. WinCC Unified	228
15.2.1. WinCC Unified Basic	228
15.2.2. WinCC Unified Comfort	229
ROZDZIAŁ 16. Podstawowe operacje — panele HMI WinCC Unified	230
16.1. Platforma — instalacja obrazów paneli HMI oraz oprogramowania PC Unified Runtime	230
16.2. Konfiguracja połączenia panelu HMI z PLC	231
16.3. Zmienne panelu — HMI Tags	232
16.4. Elementy wizualizacji — pole IO field	234
16.5. Elementy wizualizacji — przyciski	234
16.6. Elementy wizualizacji — suwaki, wykresy	237
16.7. Dynamizacja właściwości obiektów w WinCC Unified	238
16.7.1. Dynamizacja obiektu — zarządzanie widocznością	240
16.7.2. Dynamizacja obiektu — animacja koloru obiektu	242
16.7.3. Dynamiczne widżety	243
ROZDZIAŁ 17. Gdzie zdobywać wiedzę?	246
Podsumowanie	251
Bibliografia	253

Wprowadzenie

Od samego początku chciałem, aby moja pierwsza książka była skierowana do każdej osoby potencjalnie zainteresowanej automatyką przemysłową, a w szczególności tematyką programowania sterowników PLC oraz tworzenia wizualizacji na panele operatorские HMI. Dlatego jeśli jesteś:

- początkującym automatykiem lub programistą PLC,
- pracownikiem utrzymania ruchu, który zamierza poszerzyć swoje kompetencje,
- studentem,
- uczniem szkoły średniej,
- osobą planującą zmianę pracy i/lub branży,
- lub po prostu interesuje Cię tematyka programowania PLC, a jedyne, co Cię skłoniło do tego, by wziąć do ręki tę książkę, to czysta ciekawość,

to mam nadzieję, że uzyskasz odpowiedzi na wszystkie pytania, które nasuną Ci się w trakcie lektury.

Nie musisz mieć żadnej specjalistycznej wiedzy, lecz jeżeli masz jakiegokolwiek doświadczenie (np. pracujesz w przemyśle, znasz podstawy dowolnego języka programowania lub miałeś wcześniej styczność ze sterownikami PLC), to prawdopodobnie będzie Ci jeszcze łatwiej wejść w temat. Niemniej osoba bez żadnego doświadczenia także poradzi sobie z zagadnieniami omawianymi w tej książce. Kto wie — być może nawet lepiej.

Jedyne, o co Cię na razie proszę, to otwarty umysł i chęć do przyswajania nowych informacji. Jak widzisz, pozwoliłem sobie na odrobinę kredytu zaufania i od samego początku zwracam się do Ciebie per ty — myślę, że takie podejście ułatwi nam przełamanie pierwszych lodów i pozwoli mi komfortowo przedstawiać poszczególne zagadnienia. Omawiając je, będę konsekwentnie stosował rodzaj męski czasowników. Od razu pragnę Cię uspokoić — nie ma to żadnego związku z moim światopoglądem ani statystyką zatrudnienia na stanowisku programisty czy programistki PLC. Po prostu uważam, że używanie przez kilkaset stron zarówno form męskich, jak i żeńskich mogłoby stać się dla Ciebie uciążliwe.

Niniejsza książka obejmuje tematy sprzętowe, a także programowe, związane z obsługą, konfiguracją i programowaniem sterowników PLC. Ponadto przedstawiam w niej ogólne wprowadzenie do automatyki przemysłowej — ma to na celu pokazać, jak rozległa jest to dziedzina, a także, co bardzo istotne, uzmysłwić Ci, że to nie tylko programowanie sterowników logicznych. W części sprzętowej omawiam kluczowe funkcjonalności i czynności związane z obsługą, poruszam również wybrane najważniejsze aspekty konfiguracji sterownika. Część programowa to oczywiście wprowadzenie do programowania, instalacja oprogramowania i zapoznanie się ze środowiskiem programistycznym, zaczniemy

też omawiać zagadnienia związane z programowaniem PLC i tworzeniem wizualizacji na panele operatorskie HMI. Postaram się również szczegółowo odpowiedzieć na niektóre pytania, które z pewnością Cię nurtują:

- Na czym w praktyce polega automatyzacja?
- Jak programuje się urządzenia automatyki?
- Jak to jest, że urządzenia „wiedzą”, kiedy pracować, a kiedy nie?
- Jak urządzenia komunikują się ze sobą?
- Jak obsługiwane są sterowniki PLC i jak można je wykorzystać?
- Ile muszę wydać, aby zacząć?
- Jak to jest, że maszyna w fabryce pracuje bez ustanku przez 24 godziny na dobę 7 dni w tygodniu?
- Jak tworzy się wizualizacje procesów dla użytkowników i operatorów?

Ostatnie lata, w szczególności okres pandemii, pokazały, że przebranżawianie się jest znacznie prostsze, niż można by sądzić. Nie jest żadną nowością, że ludzie z dnia na dzień porzucają bieżące obowiązki i próbują odnaleźć się w zupełnie nowej dziedzinie — podobnie jest z automatyką przemysłową i programowaniem sterowników PLC. Oliwy do ognia dodaje to, że w wielu regionach naszego kraju niezmiennie brakuje rąk do pracy. Jeszcze do niedawna byliśmy przyzwyczajeni do ofert kierowanych do kandydatów o określonych kompetencjach (np. starszy specjalista ds. automatyki), oczywiście najlepiej z wyższym wykształceniem. A jak sytuacja wygląda dziś? Widniejące oferty pracy zawierają takie zapisy jak „do przyuczenia” czy „brak wymaganego doświadczenia”. Ze względu na brak specjalistów (i to nie tylko na polskim rynku) pracodawcy decydują się obniżyć próg wejścia, byleby pojawił się „ktoś chętny do pracy”. W mojej ocenie to doskonały okres, aby spróbować zainteresować się zagadnieniami automatyki przemysłowej i zacząć działać.

Niemniej jednak już w pierwszych swoich słowach chciałbym podkreślić, że to determinacja w dążeniu do celu i systematyczność będą kluczowe w jego osiągnięciu. Uczestnictwo w specjalistycznych szkoleniach technicznych, które w skompresowanej formie przekazują niezbędną wiedzę, nie jest jedynym warunkiem, który należy spełnić przed rozpoczęciem pracy w charakterze technika, automatyka czy programisty PLC. Nie wolno zapominać o samodzielnej pracy, ćwiczeniach i wnioskach wyciąganych w trakcie tworzenia kolejnych programów. Kiedyś usłyszałem od znajomego stwierdzenie: „zdanie egzaminu na prawo jazdy nie czyni cię dobrym kierowcą” i myślę, że analogię do niego możemy odnaleźć w wielu dziedzinach naszego życia, także w automatyce przemysłowej i programowaniu PLC.

Kiedy jeszcze tworzyłem zarys tej książki, wpadłem na pomysł, aby jej Czytelnik mógł na własną rękę ćwiczyć zagadnienia, które w niej omówię. Dlatego stworzyłem platformę dostępną pod adresem <https://marcinpodsiadly.pl/>, na której przygotowałem poradniki dotyczące instalacji oprogramowania, materiały uzupełniające, a także ćwiczenia pozwalające utrwalić zdobywaną wiedzę. Oznacza to, że masz do wyboru dwie możliwości realizowania materiału. Pierwsza opcja to tradycyjne przeczytanie książki bez dodatkowych ćwiczeń — zawartość oraz ilustracje przygotowałem w taki sposób, aby na bieżąco konfrontować nowe zagadnienia z praktycznymi przykładami. Jeżeli zaś chcesz utrwalić zdobyte informacje, zachęcam Cię — to druga opcja — do wykonywania

dodatkowych ćwiczeń dostępnych na platformie. Znajdziesz tam filmy pokazujące instalację oprogramowania, utworzenie pierwszego projektu czy uruchomienie symulatora do pracy z wirtualnym sterownikiem PLC. Do wykonania ćwiczeń nie potrzebujesz żadnego dodatkowego sprzętu oprócz własnego komputera. Nie musisz zatem kupować sterownika PLC — będziemy w stanie odwzorować pracę sterownika za pomocą odpowiedniego oprogramowania. Więcej na temat samego sprzętu dowiesz się już niebawem.

W środowisku przemysłowym zdarza się, że kwestie związane z programowaniem są przedstawiane jako graniczące z wiedzą tajemną. Głęboko wierzę, że uda mi się obalić ten stereotyp i pokazać, że samodzielna praca, chęć i determinacja w zupełności wystarczą, aby samemu tworzyć proste algorytmy sterowania na sterowniki PLC. Naturalnie „im dalej w las, tym więcej drzew” — coraz trudniejsze algorytmy sterowania wymagają stosowania coraz to bardziej skomplikowanych narzędzi i wiedzy technicznej, która pozwoli ocenić, czy rozwiązanie jest osiągalne przy konkretnych założeniach. Nie wolno zapominać także o tym, że automatyka przemysłowa to nie tylko programowanie PLC. Co to oznacza? Nic innego jak fakt, że na działającą nową maszynę czy linię produkcyjną składa się praca całego zespołu ludzi — technologów, projektantów, elektryków, mechaników i — na samym końcu — programistów. Każda branża ma własną specyfikę, własne wymagania, własne zakresy tolerancji. Jakby tego było mało — na polskim rynku oprogramowanie PLC trudno nazwać ustandaryzowanym. Niektórzy piszą po swojemu, niektórzy trzymają się pewnych zasad, chociaż nie do końca wiedzą dlaczego, a jeszcze inni pracują w dużych korporacjach, gdzie minimalne odstępstwa od ogólnie przewidzianego standardu spotykają się z dezaprobatą. W jaki zatem sposób powinno się tworzyć programy? Na co zwracać uwagę już od samego początku? **Dobre praktyki w programowaniu PLC także stanowią nieodłączną część niniejszej książki.**

Teraz będzie odrobinę filozoficznie. Najzwyczajniej w świecie uważam, że umiejętność programowania daje swego rodzaju wolność. Dotyczy to programowania zarówno w branży IT, jak i automatyce przemysłowej. Napisanie programu, który spełnia wszystkie założenia, to jedna sprawa. Pojawiają się jednak pytania. W jaki sposób zostanie on napisany? W jakim języku? Ile czasu zajmie kolejnemu programiście jego przeanalizowanie, gdy wystąpi awaria? Mam nadzieję, że widzisz, co mam na myśli. Jak wspomniałem — panuje pełna swoboda.

W trakcie nauki programowania próbujemy nowych dróg, testujemy pewne konwencje i każdy kolejny program może wyglądać zupełnie inaczej od poprzedniego — jest to w pełni zrozumiałe, ponieważ z każdym kolejnym podejściem wyciągamy nowe wnioski i chcemy, aby nowsza wersja była szybsza, prostsza i lepsza od poprzedniej. A teraz wyobraź sobie korporację, w której wiele firm podwykonawczych jednocześnie uruchamia niezależne fragmenty tej samej linii produkcyjnej. Jedna linia i wielu podwykonawców, a zatem wielu programistów. Skoro jest wielu programistów, to i wiele koncepcji, języków programowania i doświadczeń. Nie brzmi to szczególnie optymistycznie, prawda? I rzeczywistość potwierdza obawy, zdarza się bowiem, że w niektórych polskich firmach taka sytuacja faktycznie ma miejsce. Stanowi to szczególne wyzwanie dla ludzi, którzy na co dzień muszą sobie radzić z bieżącymi problemami, ponieważ to oni muszą obsłużyć finalnie przygotowaną linię. **Pracownicy utrzymania ruchu** — bo o nich tu mowa — to technicy, elektrycy, mechanicy i automatycy. Zdiagnozowanie czegoś względnie prostego, takiego jak uszkodzenie czujnika, może na jednej maszynie zająć pięć

minut, a na innej półtorej godziny. Zapewne się domyślasz, na której z nich lepiej przygotowano program sterownika PLC.

Z powyższej analizy wynika jeszcze jedna ważna lekcja: programów PLC nie tworzy się w pojedynkę, tylko zespołowo. Świadomość tego, że ktoś będzie analizował i modyfikował pisany przez nas program, pozwala znacząco podnieść jakość tworzonego kodu.

W niniejszej książce do nauki programowania sterowników PLC wykorzystamy najpopularniejsze na polskim rynku środowisko programistyczne firmy Siemens, którego pełna nazwa to *Totally Integrated Automation Portal*, a w skrócie — *TIA Portal*. Kompleksowe środowisko pozwala w jednym miejscu programować PLC, tworzyć wizualizacje na panele operatorskie HMI oraz systemy SCADA, konfigurować i programować układy bezpieczeństwa, komunikować ze sobą różne urządzenia automatyki, a także symulować przygotowane oprogramowanie. Wykorzystamy najnowszą — w czasie, gdy ta książka powstawała — wersję, czyli *TIA Portal V20*.

Zapraszam do lektury.

1.7. Integracja układów automatyki — urządzenia przemysłowe

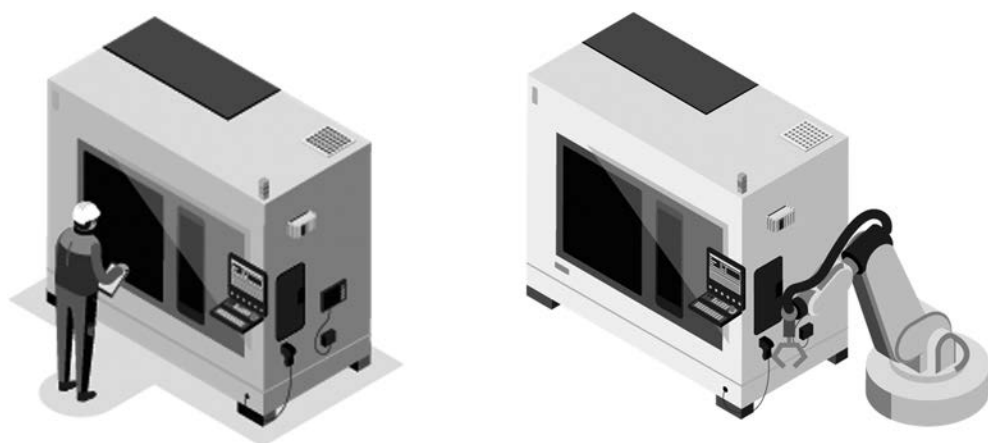
Automatyka przemysłowa to jedna z najprężniej rozwijających się dziedzin. Jeszcze nie tak dawno zarówno producenci, jak i użytkownicy sterowników PLC wychodzili z założenia, że sterownik to urządzenie odpowiedzialne wyłącznie za realizację lokalnego algorytmu sterowania w obrębie pojedynczej maszyny czy zespołu maszyn. PLC miały w pierwszej kolejności zastąpić wspomniane wcześniej przekąźnikowe systemy sterowania — duża elastyczność we wprowadzaniu zmian była jedną z najcenniejszych właściwości tych sterowników. Ponadto zastosowanie urządzeń elektronicznych rozszerzyło zakres funkcjonalności i umożliwiło komunikację pomiędzy urządzeniami wykorzystywanymi w fabryce — innymi sterownikami PLC, przetwornicami częstotliwości (stosowanymi do zasilania silników elektrycznych i regulowania ich pracy), robotami przemysłowymi czy innymi urządzeniami peryferyjnymi szeroko stosowanymi w przemyśle. Mówiąc o komunikacji między urządzeniami, mamy na myśli fizyczną możliwość wymiany danych między nimi. Z pomocą przychodzą tutaj różne **protokoły komunikacyjne**, czyli zestawy konkretnych reguł umożliwiających wymianę danych między dwoma urządzeniami.

Możemy zastosować prostą analogię do życia codziennego — wszyscy porozumiewamy się za pomocą konkretnego języka (polski, angielski czy niemiecki), dzięki czemu możemy swobodnie przekazywać między sobą informacje. W przypadku urządzeń przemysłowych (a także systemów informatycznych) konieczne jest określenie zasad komunikacji tak, aby każde urządzenie umiało przekazać niezbędne dane bez dodatkowego doprecyzowywania. Z pomocą przychodzą ustandaryzowane protokoły komunikacyjne. Dzięki standaryzacji wszyscy dostawcy komponentów automatyki wiedzą, w jaki sposób zaimplementować je w swoich rozwiązaniach tak, aby ich użytkownik potrafił skomunikować ze sobą dwa dowolne urządzenia.

W przemyśle możesz spotkać mnóstwo różnych protokołów komunikacyjnych, których stosowanie ma jeden cel: wymieniać dane pomiędzy urządzeniami. Jakie to będą dane — to zależy od aplikacji i rodzaju zastosowanych urządzeń. Każdy producent implementuje w swoich urządzeniach wybrane protokoły.

Wyobraź sobie dwie maszyny — A oraz B (rysunek 1.13). Żeby rozpocząć pracę, maszyna B musi otrzymać rozkaz startu z maszyny A. W jaki sposób przekazać tak prostą informację? Przecież każda maszyna to zupełnie niezależny układ i każda ma swoją szafę sterowniczą, swój sterownik PLC, swój panel operatorski HMI i swoje układy czujników z urządzeniami wykonawczymi.

Najprostszą formą komunikacji pomiędzy dwoma urządzeniami jest wykorzystanie sygnału elektrycznego. Jest to rozwiązanie popularne ze względu na prostotę realizacji. W tym układzie program sterownika maszyny A będzie wystawiał odpowiednie wyjście cyfrowe (np. sygnał napięciowy 24 VDC), które z kolei będzie sygnałem wejściowym dla sterownika maszyny B. Program sterownika maszyny B będzie oczekiwał na pojawienie się odpowiedniego stanu na wejściu cyfrowym i dopiero wówczas (po spełnieniu wszystkich innych niezbędnych warunków) rozpocznie pracę. Omawiane podejście jest proste, lecz jak zapewne wyczuwasz, jest pewien haczyk, a nawet kilka. Pierwszy z nich to ograniczenie jednego sygnału do jednego przewodu. Musisz wiedzieć, że maszyny wymieniają



RYSUNEK 1.13. Układ dwóch niezależnych maszyn [G1]

ze sobą bardzo wiele informacji, niejednokrotnie w międzyczasie potwierdzając otrzymanie rozkazu od drugiego urządzenia. W praktyce oznaczałoby to konieczność zastosowania dziesiątek czy setek dodatkowych przewodów. Co więcej, wszystkie sygnały elektryczne trzeba gdzieś podłączyć, stąd konieczna byłaby także rozbudowa o dodatkowe moduły sprzętowe w obu szafach sterowniczych (więcej informacji na temat modułów sprzętowych uzyskasz w trakcie omawiania budowy sterownika PLC i obsługiwanych modułów sygnałowych). Dla przykładu dodam, że w jednym z projektów dwa sterowniki PLC niezależnych stacji wymieniały między sobą blisko 200 sygnałów. Nie oznacza to jednak, że zawsze tak musi być. Są też aplikacje, w których sterownik musi wysłać dosłownie jeden sygnał i odebrać również tylko jeden sygnał. Drugie ograniczenie („haczyk”) to rodzaj sygnału — jak przekazać pomiędzy urządzeniami wartości liczbowe, takie jak liczba sztuk do wyprodukowania albo wymiary wyprodukowanej części? Przecież są one tak samo potrzebne algorytmowi sterowania, na przykład żeby robot mógł je odpowiednio spakować w kartonie. Jako ludzie cenimy sobie wygodę, a dalsza analiza wymiany danych na podstawie sygnałów elektrycznych nie należy do wygodnych, opłacalnych, a tym bardziej powszechnie stosowanych w przemyśle. Przejdźmy zatem do rozwiązań sieciowych oferujących implementację wspomnianych wcześniej protokołów komunikacyjnych.

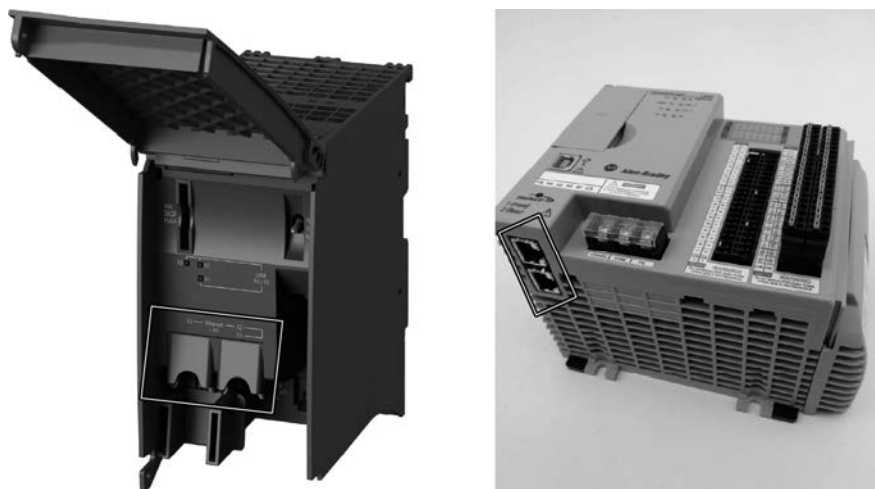
Dzięki temu, że sterownik PLC ma dostęp do wszystkich sygnałów elektrycznych podłączonych do odpowiednich modułów, mamy pełną swobodę nie tylko ich przetwarzania w programie, ale także przechowywania w projekcie urządzenia. Pamięć wewnętrzna sterownika umożliwia gromadzenie danych w postaci struktur, gdzie użytkownik umieszcza wybrane dane w dowolny i wysoce elastyczny sposób, a dzięki gotowym funkcjom programowym może je wysłać do drugiego urządzenia. W porównaniu z poprzednią propozycją to rozwiązanie nie wymaga dodatkowych kosztów związanych z zakupem kolejnych modułów sprzętowych ani prac elektrycznych związanych z okablowaniem. Z drugiej strony do uruchomienia komunikacji pomiędzy urządzeniami konieczna jest wiedza dotycząca dostępnych w danym urządzeniu protokołów komunikacyjnych. Każdy z nich będzie cechował się konkretnymi zasadami, których trzeba przestrzegać, aby sprzęt mógł ukończyć powierzone mu zadanie.



RYСУNEK 1.14. Wtyczki M12 i RJ45 do kabli miedzianych oraz PCF do kabli światłowodowych, wykorzystywanych między innymi w protokole PROFINET [G1]

Pierwszym kryterium, na które należy zwrócić uwagę, jest warstwa fizyczna protokołu komunikacyjnego. Określa ona, w jaki sposób przekazywany będzie sygnał elektryczny pomiędzy urządzeniami (rysunek 1.14). Zaczniemy od czegoś prostego i weźmy pod lupę następujący przykład: **chcesz przenieść kilka świeżo wykonanych zdjęć ze swojego telefonu na laptop**. W pierwszym kroku musiałbyś się zastanowić, czy przesać pliki **z wykorzystaniem kabla** czy **bezprzewodowo**. Jeśli wybierzesz kabel, musisz znać typy gniazd w urządzeniach, które zamierzasz ze sobą połączyć. Twój telefon prawdopodobnie ma złącze USB typu C, a laptop — na przykład zarówno USB typu C, jak i USB typu A. Bardzo podobnie będzie w przypadku urządzeń przemysłowych: mając wiedzę dotyczącą interfejsów dostępnych w urządzeniach, możemy planować zadania komunikacyjne.

Spójrz na rysunek 1.15, ilustrujący dwa sterowniki PLC różnych producentów — firm Siemens oraz Rockwell Automation. Oba sterowniki są wyposażone w złącze RJ45, które umożliwia wpięcie przewodu ethernetowego w postaci skrętki. Niestety zgodność interfejsów w warstwie fizycznej nie wystarczy, żeby zagwarantować możliwość wymiany danych pomiędzy urządzeniami. Dlaczego? Ponieważ konkretne urządzenia przemysłowe obsługują **konkretne protokoły komunikacyjne**. Nawiązując do analogii z językami, dzięki którym ludzie potrafią się ze sobą komunikować — **to, że dwie osoby umieją mówić, nie oznacza, że są w stanie się ze sobą porozumieć**.



RYСУNEK 1.15. Dwa sterowniki PLC różnych producentów z interfejsami Ethernet (ramka) [G1]

W tym miejscu musimy zrobić dodatkowy przystanek. Komunikacja z urządzeniami peryferyjnymi należy do podstawowych zadań sterownika PLC. Jednym z kluczowych parametrów protokołów komunikacyjnych stosowanych w przemyśle jest **deterministyczność**. Jest to cecha protokołu, która gwarantuje dostarczenie danych w określonym czasie, a także precyzyjnie określa, kiedy zostaną one dostarczone. Przykładem deterministycznego protokołu jest **PROFINET**. Pozwala on realizować komunikację w czasie rzeczywistym, dzięki czemu algorytm sterowania w sterowniku PLC ma ciągły dostęp do sygnałów z urządzeń. W praktyce mówimy o **cyklach wymiany danych rzędu milisekund**. Oznacza to, że sterownik PLC wymienia sygnały wejść i wyjść z urządzeniami peryferyjnymi (roboty, wyspy sygnałowe, przetwornice częstotliwości) nawet kilkaset razy na sekundę. Warto dodać, że sam protokół PROFINET stosuje się w dwóch klasach — RT oraz IRT. Klasę RT (ang. *Real Time*, czas rzeczywisty) wykorzystuje się w większości aplikacji i jest to opisana powyżej komunikacja cykliczna z czasami cyklu na poziomie od kilku do kilkudziesięciu milisekund. Istnieją jednak aplikacje, w których synchronizacja sygnałów co kilka milisekund jest niewystarczająca (przykładem takiej aplikacji jest piła latająca)³. Zastosowanie klasy IRT (a także komponentów sprzętowych w sieci obsługujących tę klasę) pozwala uzyskać cykle wymiany danych na poziomie 125 mikrosekund, czyli 0,125 milisekundy. Skrócowa nazwa klasy, IRT, oznacza izochroniczny tryb czasu rzeczywistego, inaczej nazywany stałoczasowym (ang. *Isochronous Real Time*).

Inne przykłady protokołów deterministycznych to PROFIBUS, CC-Link i Ethernet/IP. Każdy z nich to otwarty protokół komunikacyjny, dzięki czemu każdy producent może go zaimplementować. Niemniej jednak w czasie, gdy piszę te słowa, sytuacja rynkowa wygląda tak, że pomimo swojej otwartości protokoły PROFINET oraz PROFIBUS kojarzą się głównie z rozwiązaniami firmy Siemens, CC-Link — z rozwiązaniami Mitsubishi, a Ethernet/IP — z rozwiązaniami Rockwell Automation i jej sterownikami Allen-Bradley. Jako ciekawostkę dodam, że zarówno PROFINET, jak i PROFIBUS są protokołami zaprojektowanymi i rozwijanymi przez organizację PI (PROFIBUS & PROFINET International), a firma Siemens jest aktywnym użytkownikiem tych protokołów.

To już koniec wspomnianego przystanku. Dlaczego w ten sposób chciałem przykuć Twoją uwagę? Ponieważ to nie wszystkie protokoły komunikacyjne dostępne w sterownikach PLC. Nowoczesne urządzenia zarówno oferują obsługę protokołów stosowanych w przemyśle, jak i umożliwiają wymianę danych bezpośrednio z systemami informatycznymi (lub tak naprawdę z dowolnymi platformami). Zasadniczo ich stosowanie — jak wspomniałem na początku podrozdziału — umożliwia wymianę danych pomiędzy dwoma urządzeniami, lecz nie będą to już protokoły **deterministyczne**. W związku z tym nie będziemy już analizować wymiany danych co kilka milisekund, ale raczej będzie to przedział od kilkuset milisekund do kilku czy kilkunastu minut. Okazuje się, że w przemyśle tak „rzadkie” częstotliwości są jak najbardziej do przyjęcia i szeroko stosowane. O jakich protokołach zatem mowa? Zaczniemy od tych, które z reguły wykorzystuje się w **urządzeniach automatyki**. Najpopularniejsze to między innymi OPC UA, S7 i MODBUS TCP. Nie oczekuję od Ciebie żadnej wiedzy, lecz chcę, abyś kojarzył (lub też właśnie poznał) inne możliwości komunikacyjne. Świadomość istnienia danego protokołu może znacząco poszerzyć horyzonty na etapie projektowania czy wprowadzania zmian w istniejących instalacjach.

³ Piła latająca (czasami nóż latający) to wieloosiowa maszyna, której zadaniem jest precyzyjne cięcie materiału podczas ruchu. Do zagwarantowania pożądanej precyzji algorytm sterowania wymaga bardzo wysokich częstotliwości wymiany sygnałów wejść/wyjść.

Protokół S7 opracowała firma Siemens. Najczęściej wykorzystuje się go do wymiany danych pomiędzy panelami operatorskimi HMI a sterownikami PLC. Często stosuje się go także do bezpośredniej komunikacji pomiędzy sterownikami PLC Siemens. Pomimo tego, że S7 oficjalnie nie jest otwartym protokołem, na rynku można znaleźć wiele urządzeń (przede wszystkim paneli operatorskich) zapewniających możliwość integracji właśnie z wykorzystaniem tego protokołu. Przykładami firm, które oferują panele operatorskie z protokołem S7, są Proface i Weintek. Panele operatorskie HMI wspomnianych firm są jednymi z najlepiej sprzedających się na polskim rynku.

OPC UA to najpopularniejszy otwarty, niedeterministyczny protokół komunikacyjny stosowany w nowoczesnych aplikacjach. Opracowała go organizacja non profit OPC Foundation. Implementują go właściwie wszyscy producenci komponentów automatyki, i to zarówno w sterownikach PLC, panelach HMI i systemach SCADA, jak też w niektórych urządzeniach peryferyjnych. Ponadto jego implementacja nie ogranicza się do urządzeń automatyki. Oznacza to, że OPC UA można oprogramować na dowolnej platformie (komputery, systemy mikroprocesorowe, urządzenia mobilne) i w dowolnym systemie operacyjnym. Żyjemy w czasach, gdy pracownicy zakładów przemysłowych chętnie analizowaliby swoje dane produkcyjne oraz zużycie energii elektrycznej czy innych mediów (woda, gaz, sprężone powietrze), by zminimalizować ponoszone koszty. Posiadanie dostępu do tych danych jest jednak pierwszym krokiem do tego, aby realnie móc cokolwiek analizować. Dzięki temu protokół OPC UA — oprócz stricte przemysłowych zastosowań (których ma mnóstwo) — umożliwia także bezpieczne (z punktu widzenia cyberbezpieczeństwa) i komfortowe udostępnianie danych przemysłowych.

MODBUS to otwarty protokół opracowany przez firmę Modicon, występujący w różnych wersjach — TCP (warstwa fizyczna Ethernet), RTU (warstwa fizyczna RS232 lub RS485) czy ASCII (warstwa fizyczna RS232 lub RS485). Pomimo że jest dostępny od 1979 roku, nadal powszechnie implementuje się go w rozwiązaniach przemysłowych. Jeśli mamy dwa sterowniki PLC Siemens najnowszej generacji, możemy skomunikować je za pomocą MODBUS TCP, lecz wygoda implementacji będzie mniejsza, niż zapewniają wcześniej wspomniane protokoły (PROFINET, OPC UA, S7). Kiedy zatem wykorzystuje się MODBUS TCP? Między innymi do odczytu parametrów z przetwornic częstotliwości obsługujących pompy, wentylatory czy przenośniki. Wiele takich urządzeń sterowanych jest za pomocą sygnałów elektrycznych (rozkaz startu, zadana prędkość), a dane dotyczące pracy napędu odczytywane są właśnie za pomocą tego protokołu (bieżąca wartość natężenia prądu, liczba przepracowanych godzin i inne, równie przydatne parametry pracy).

Na koniec tego podrozdziału pozwolę sobie na krótką historię z jednego z moich projektów. Była to stacja zgrzewania progów samochodowych, którą uruchamiałem kilka lat temu. Jak wspomniałem w podrozdziale dotyczącym paneli operatorskich HMI, urządzenia te oferują możliwość zabezpieczenia przed niepowołanym dostępem. Bardzo często stosuje się najprostszą metodę uwierzytelniania użytkownika, w której login i hasło wprowadza się ręcznie za pomocą dotykowej klawiatury. W tym projekcie jednak zależało nam na przyspieszeniu tego procesu przez wykorzystanie pracowniczych kart dostępowych — tych samych, którymi pracownicy „odbijają” się na wejściu i wyjściu z firmy. Pierwotnie planowałem zakup czytnika kart RFID konkretnej firmy, lecz ni stąd, ni zowąd pojawia się kierownik projektu i daje mi do ręki jakieś urządzenie, informując, że kupił znacznie tańsze rozwiązanie, które ma gniazdo RJ45. A skoro i sterownik PLC,

i czytnik RFID mają gniazdo RJ45, to temat jest zamknięty. Tylko trochę kodu dodać, i po sprawie. Niestety, nic bardziej mylnego (myślę, że każdy, kto ma choćby minimalne doświadczenie, mógłby opowiedzieć wiele podobnych historii). Owszem, obecny w warstwie fizycznej przewód Ethernet pasuje do obu urządzeń, ale **to wcale nie oznacza, że możliwe będzie ich skomunikowanie** (tak jak w przypadku pokazanych wcześniej sterowników PLC Siemens i Allen-Bradley). Co zrobić w takiej sytuacji? Pierwsza i tak naprawdę jedyna czynność, którą możemy wykonać, to odnalezienie dokumentacji technicznej urządzenia i sprawdzenie jego możliwości.



RYSUNEK 1.16. Czytnik RFID firmy INVEO [G3]

Po krótkiej analizie udało mi się odnaleźć informację, że urządzenie obsługuje jeden protokół komunikacyjny, który byłbym w stanie obsłużyć, i był to MODBUS TCP (rysunek 1.16). Moja propozycja wymagałaby jedynie instalacji gotowego programu do obsługi użytkowników i haseł. Druga wymagała oprogramowania całej komunikacji pomiędzy sterownikiem PLC a czytnikiem RFID (uruchomienie komunikacji pod MODBUS TCP). Co więcej, sterownik musiał dodatkowo przekazywać dane do panelu operatorskiego HMI (bo w końcu tam się logujemy, a nie na sterowniku). Czy brzmi to bardziej skomplikowanie od mojej propozycji? Tak, bo takie było. Nie zmienia to faktu, że powierzone zadanie tak czy inaczej udało się zrealizować, a wiedza dotycząca **dostępnych w danym urządzeniu** protokołów komunikacyjnych była niezbędna.

Chciałbym podsumować ten nieco dłuższy podrozdział. Wiesz już, że odpowiedni przewód komunikacyjny nie oznacza sukcesu w komunikacji pomiędzy urządzeniami. Wiesz także, że różni producenci implementują w swoich rozwiązaniach różne protokoły komunikacyjne. Co bardzo ważne, poruszyliśmy kwestie deterministyczności przemysłowych protokołów, która jest wyjątkowo pożądaną cechą, bo pozwala na realizację systemów sterowania w czasie rzeczywistym. W zależności od wymagań aplikacyjnych możemy zastosować protokoły deterministyczne (między innymi PROFINET, PROFIBUS) czy niedeterministyczne (OPC UA, S7, MODBUS TCP).

A co poza tym? Tak bardziej życiowo? Myślę, że to bardzo dobre miejsce, aby powiedzieć, że praca automatyka czy programisty PLC to w istotnym stopniu analiza dokumentacji technicznych i poszukiwanie (w pierwszej kolejności) dostępnych, a docelowo optymalnych rozwiązań napotkanych problemów. Dotyczy to zarówno pracy w delegacji (kiedy programiści przemierzają się po całym świecie i uruchamiają nowe linie/maszyny), działów konstrukcyjnych projektujących maszyny, jak też działów utrzymania ruchu (w których, ze względu na zróżnicowanie stosowanych technologii, trzeba szukać

„zamienników” w istniejących, nieraz od kilkudziesięciu lat, rozwiązaniach). Każdą przetwornicę częstotliwości można konfigurować w inny, choć zapewne podobny, sposób. Każdego robota przemysłowego programuje się w innym edytorze. I co najważniejsze — ze względu na ogrom dziedziny, jaką jest automatyka — konkretny problem można rozwiązać na wiele, ale to naprawdę wiele sposobów.

Na sam koniec dodam (choć myślę, że to dość oczywiste), że względy ekonomiczne także potrafią mieć kolosalne znaczenie. Nikt nie lubi marnotrawić pieniędzy i niepotrzebnie przepłacać, a w zależności od branży firmy mogą mieć różną elastyczność w stosowaniu zaawansowanych technicznie rozwiązań. Tak czy inaczej, programiście PLC integrującemu różne urządzenia automatyki bardzo często nasuwa się jedno pytanie: **Jak się z tym dogadać?**

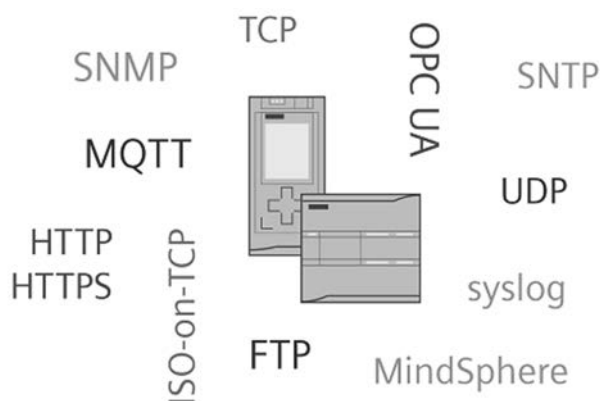
Miniaturyzacja w elektronice sprawia, że urządzenia wykorzystywane w przemyśle przy niezmiennie małych gabarytach są coraz szybsze i mają coraz więcej pamięci wewnętrznej. Obecnie dostępne standardowe czujniki nadal realizują swoje pierwotne funkcje, na przykład detekcję metalowego obiektu, ale poza tym oferują szereg możliwości konfiguracyjnych.

1.8. Integracja układów automatyki — systemy informatyczne

To nie koniec możliwości komunikacyjnych sterowników PLC. Ten sam sterownik PLC, który może z **wykorzystaniem jednej, tej samej karty sieciowej** wymieniać dane w czasie rzeczywistym (PROFINET), a także udostępniać i pobierać dane z innych urządzeń automatyki za pomocą **niedeterministycznych protokołów** (OPC UA, S7, MODBUS TCP), obsługuje dodatkowo **wiele innych protokołów komunikacyjnych** przydatnych w komunikacji z zewnętrznymi usługami informatycznymi. Podobnie jak w poprzednim podrozdziale chcę Ci tylko pokazać, że chociaż program w sterowniku PLC to podstawowa i niezbędna funkcja tych urządzeń, to z roku na rok otrzymują one coraz więcej zaawansowanych funkcjonalności. Co ciekawe, nie wiąże się to w zasadzie z żadnymi dodatkowymi kosztami. O jakich funkcjonalnościach mowa? O możliwości bezpośredniej integracji sterownika PLC z usługami czysto informatycznymi.

W dzisiejszych czasach wiele firm bardzo chętnie analizuje dane pochodzące bezpośrednio z procesu produkcyjnego. Dzięki sterownikom PLC jest to jak najbardziej możliwe i nie wymaga stosowania dodatkowych, pośrednich usług (rysunek 1.17). Chodzi między innymi o:

- logowanie danych i tworzenie zapytań do **baz danych SQL (protokół TDS)**,
- generowanie zapytań za pomocą protokołu **HTTP**,
- wysyłanie danych do usług chmurowych za pomocą protokołu **MQTT**,
- logowanie danych do plików tekstowych na serwerze **FTP**.



RYSUNEK 1.17. Wybrane otwarte protokoły komunikacyjne dostępne poprzez bloki programowe [G1]

Bazy danych SQL to usługi informatyczne powszechnie stosowane do gromadzenia danych. W świecie IT używa się ich do przechowywania niemalże wszystkich możliwych informacji — zarejestrowanych użytkowników, stanów magazynowych, transakcji, zamówień, finansów oraz wielu, wielu innych. Wszystko oczywiście zależy od specyfiki ich wykorzystania. W przemyśle chętnie używa się baz danych do gromadzenia danych historycznych, takich jak alarmy i ostrzeżenia, pomiarów (np. temperatury, ciśnienia), stanów maszyn i pozostałych parametrów procesowych. Zastosowanie baz danych pozwala na analizę danych historycznych w dowolnej chwili. Co więcej, systemy SCADA oraz niektóre panele operatorskie HMI mają wbudowane bazy danych, które w optymalny i wygodny dla użytkownika sposób pozwalają przechowywać dane oraz wyświetlać je na wspomnianych systemach wizualizacji.

Możliwość obsługi zapytań protokołem HTTP otwiera furtkę do komunikacji z niemalże dowolnym systemem informatycznym. W dzisiejszych czasach komunikacja pomiędzy aplikacjami odbywa się w większości przypadków za pomocą API, czyli zbioru metod określających, w jaki sposób można skomunikować się z daną aplikacją. Duże firmy (i nie tylko one) mają systemy informatyczne przeznaczone specjalnie do obsługi zamówień czy płatności i generalnie służące do zarządzania przedsiębiorstwem. Możliwość integracji niezależnych systemów stanowi kluczowy aspekt rozwoju firmy i pozwala rozszerzyć istniejące funkcjonalności. Jako ciekawostkę dodam, że sterowniki PLC najnowszej generacji (Siemens SIMATIC S7-1500) także udostępniają API, dzięki któremu możliwy jest między innymi odczyt i zapis zmiennych w sterowniku PLC, a nawet tworzenie kopii zapasowych z poziomu zewnętrznej aplikacji.

W językach programowania wysokiego poziomu, takich jak Python, JavaScript czy C++, istnieje wiele bibliotek, które umożliwiają swobodny odczyt zmiennych ze sterownika PLC oraz ich zapis w sterowniku. Komunikacja pomiędzy kodem aplikacji a sterownikiem wykorzystuje najczęściej implementacje protokołów S7 czy OPC UA. Tym samym tworzenie własnych aplikacji w jednym z wymienionych języków pozwala nawiązywać połączenie z urządzeniami automatyki i odczytywanie z nich konkretnych danych.

1.9. Programowanie w automatyce a programowanie w IT

Skoro wspominałem o językach programowania wysokiego poziomu, to warto tytułem wstępu dodać kilka słów o różnicy pomiędzy programowaniem w automatyce a programowaniem w IT. Jak przekonasz się w następnych rozdziałach, sterowniki PLC programuje się w różnych językach. Na tym etapie istotne jest to, że wyróżniamy dwa rodzaje języków: **graficzne** oraz **tekstowe**. Języki graficzne cechują się prostotą konstrukcji, która ma na celu ułatwić wprowadzanie drobnych zmian pracownikom utrzymania ruchu, a także minimalizować czas poświęcony na diagnostykę problemów — konkretne elementy w programie dosłownie się podświetlają, co ułatwia namierzenie potencjalnego źródła problemu. Z kolei języki tekstowe wykorzystuje się do bardziej zaawansowanych operacji. Ich składnia pozwala chociażby na lepszą obsługę większych rozmiarów danych. Jest to przykład operacji trudnej do zrealizowania w języku graficznym, a nawet jeżeli jest ona osiągalna, to taki program znacząco traci na przejrzystości i zaleta związana z prostotą analizy niestety znika.

Błędne jest założenie, że doświadczony automatyk programujący w językach tekstowych (a wyłącznie takich używa się w IT) jest w stanie z dnia na dzień przebranżowić się do IT. Jest to, naturalnie, osiągalne, a automatykowi do IT jest znacznie „bliżej” niż chociażby pracownikowi biurowemu zatrudnionemu w urzędzie (oczywiście nie ma żadnych przeciwwskazań, aby każdy chętny mógł to zrobić). W świecie IT języki programowania to jedno — ponadto każdy z nich wykorzystuje różne technologie, frameworki (szkielety projektów wraz z zestawami narzędzi) i dodatkowe biblioteki, stosowane w zależności od potrzeb i celów danego oprogramowania. I tak oto, dla przykładu, język Python stosuje się bardzo chętnie w takich dziedzinach jak przetwarzanie obrazów, uczenie maszynowe i wszystko, co związane ze sztuczną inteligencją, AI, ponieważ ma bardzo dobre wsparcie w postaci odpowiednich narzędzi. Inny, równie popularny język programowania, JavaScript, szeroko stosuje się w aplikacjach webowych (przeglądarkowych), których w obecnych czasach tworzy się najwięcej. Czy to oznacza, że języka Python mogą użyć wyłącznie do sztucznej inteligencji, a JavaScript do tworzenia aplikacji webowych? Absolutnie nie. Python ma biblioteki do tworzenia aplikacji webowych, a JavaScript — do uczenia maszynowego. Albo jeszcze inaczej: część odpowiedzialną za algorytmy uczenia maszynowego napiszemy w Pythonie, a obsługę po stronie przeglądarki w JavaScript. Wszystkie chwytły dozwolone.

Ostatnią kwestią, którą pragnę poruszyć, jest cyberbezpieczeństwo, czyli określenie podatności zautomatyzowanych systemów sterowania na niepowołany dostęp czy atak cybernetyczny. Protokół protokołowi nierówny, stąd wybór odpowiedniego rozwiązania do realizacji zadań komunikacyjnych także jest istotny. Wspomniany w poprzednim podrozdziale protokół OPC UA może wykorzystywać szyfrowanie danych i uwierzytelnianie użytkownika, co gwarantuje wysoki poziom zabezpieczeń. Jest to też kolejny powód, dla którego jest tak powszechnie — i coraz chętniej — stosowany. Natomiast protokoły S7 czy MODBUS TCP niestety nie są szyfrowane, dlatego ryzyko przechwycenia danych i przeprowadzenia ataku typu *man-in-the-middle*⁴ jest znacznie wyższe. [5]

⁴ Atak cybernetyczny *man-in-the-middle* (człowiek pośrodku) polega na przechwyceniu danych od nadawcy i przesłaniu do odbiorcy w zmodyfikowanej postaci.

W czasie kiedy piszę tę książkę, rynek IT wydaje się mocno przesycony i bardzo trudno jest zdobyć stanowisko młodszego programisty. Boom w branży rozpoczął się co najmniej 5 lat wcześniej, a kuszące wówczas zarobki nie są już dzisiaj tak atrakcyjne. Jakby tego było mało, coraz częściej zdarza się, że ludzie pracujący w branży IT rozważają przekwalifikowanie się do branży automatyki. Przeciwnie zjawisko miało miejsce od zawsze, ponieważ wraz z rozwojem kompetencji wielu doświadczonych automatyków, ucząc się samodzielnie programowania wysokopoziomowego, przekwalifikowywało się na typowego programistę w IT (darujmy już sobie wymienianie różnych stanowisk, jest ich bowiem cała masa).

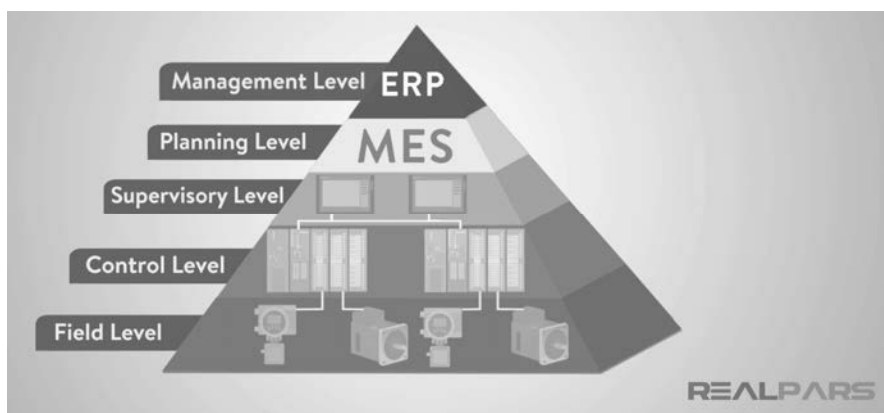
Jak zapewne zauważyłeś, posiadanie wiedzy z zakresu programowania może nie wystarczyć, aby z dnia na dzień zmienić zakres obowiązków i z programisty w IT zostać automatykiem. Praca ta wymaga zrozumienia technologii, wiedzy z zakresu mechaniki, elektryki, pneumatyki czy hydrauliki, obcowania z różnymi, fizycznymi urządzeniami przemysłowymi, a dopiero na końcu programowania PLC i systemów wizualizacji HMI czy SCADA.

Podsumowując — czy to przejście z IT do automatyki, czy odwrotnie nie jest tak oczywiste, jak można by sądzić. W kuluarach często słyszy się opinię, że skoro automatyk potrafi programować, to będzie to dla niego bułka z masłem, bo przecież programować potrafi, a przy okazji wie o wiele więcej (wspomniana wiedza z zakresu elektryki czy mechaniki). Podobnie jest w drugą stronę. Na szczęście świadomość w tej materii stale rośnie, dzięki czemu na przeróżnych grupach w mediach społecznościowych tak płytkich stwierdzeń pada coraz mniej. Skoro jednak istnieje szansa, że czytając tę książkę, nie wiesz, co wybrać, to mam nadzieję, że kilka moich spostrzeżeń ułatwi Ci ten wybór.

Sam uważam, że w niedalekiej przyszłości pojawi się wysokie zapotrzebowanie na stanowisko „IT/OT Engineer”, a więc inżyniera na styku przemysłu (OT — ang. *Operation Technology*, technologie operacyjne, czyli między innymi urządzenia automatyki przemysłowej) i świata IT. Jest to podwójnie wymagające, ponieważ oczekiwałoby się od takiej osoby zarówno doświadczenia w obszarze automatyki przemysłowej, jak i znajomości technologii IT. Niemniej jednak na rynku pracy już teraz jest wielu specjalistów, którzy w swojej codziennej praktyce (najczęściej na stanowisku automatyka) nie tylko programują sterowniki PLC, ale też uruchamiają bazy danych SQL i tworzą skrypty w Pythonie czy JavaScript, których zadaniem jest między innymi analiza danych przemysłowych.

1.10. Podsumowanie

Rozdział ten miał na celu przeprowadzić Cię od dołu do góry pewnej hierarchii. Każdy zakład produkcyjny realizuje procesy, których przebieg jest ściśle określony, a wymagania technologiczne narzucają pożądane parametry produktu końcowego, jego tolerancje, a także określają przebieg wszystkich pośrednich etapów. Znajomość technologii i związanych z nią wymagań danego procesu pozwala odpowiednio zaprojektować instalację (rurociągi, zbiorniki, transportery, roboty), w tym dobrać urządzenia wykonawcze (pompy, silniki, siłowniki, zawory) i aparaturę pomiarową w postaci różnego typu czujników. Jest to **pierwsza warstwa** (ang. *field*, pole, w tym przypadku warstwa polowa) w pięciostopniowej piramidzie narzędzi i technologii automatyki. Piramida ta jest pokazana na rysunku 1.18.



RYSUNEK 1.18. Pięciostopniowa piramida narzędzi i technologii automatyki [G4]

Druga warstwa dotyczy sterowania (ang. *control*, sterowanie), w której znajdują się urządzenia odpowiedzialne za obsługę sygnałów z warstwy polowej. Będą to między innymi:

- sterowniki PLC, odpowiedzialne za obsługę wejść i wyjść,
- przetwornice częstotliwości, których zadaniem jest płynna regulacja prędkości obrotowej i momentu obrotowego silników elektrycznych,
- przekaźniki bezpieczeństwa, obsługujące sygnały odpowiedzialne za bezpieczną dla człowieka pracę instalacji (między innymi wyłączniki bezpieczeństwa typu grzybek, kurtyny świetlne, skanery), które w przypadku zadziałania odetną dopływ energii elektrycznej,
- inne inteligentne urządzenia przetwarzające sygnały z warstwy polowej.

Urządzenia w warstwie sterowania mogą się ze sobą komunikować za pomocą sygnałów elektrycznych czy konkretnych protokołów komunikacyjnych. Często się zdarza, że sterownik PLC przekazuje odpowiednie sygnały związane ze sterowaniem pracą napędu do przetwornicy częstotliwości, a sama przetwornica przechowuje wewnętrznie kluczowe z punktu widzenia pracy napędu parametry, takie jak napięcie i prąd znamionowy silnika, prędkość znamionowa silnika, maksymalna częstotliwość, czasy rozpędzania i hamowania czy tryb hamowania. Analogicznie niezależne sterowniki PLC mogą wymieniać się danymi w celu lepszego kontrolowania powierzonych im procesów.

W **trzeciej warstwie** znajdują się urządzenia nadzorcze (ang. *supervisory*, nadzorczy), czyli wszystkie elementy systemów sterowania umożliwiające podgląd pracy instalacji, wprowadzanie nastaw, przeglądanie i kasowanie aktywnych alarmów, a także przeglądanie danych historycznych i obsługę logowania użytkowników z odpowiednimi uprawnieniami. Będą to wspomniane wcześniej panele operatorskie HMI oraz systemy SCADA. Warto dodać, że niejednokrotnie urządzenia w warstwie sterowania mają usługę *web serwera*, za pomocą którego można z poziomu przeglądarki podejrzeć parametry pracy chociażby sterownika PLC czy przetwornicy częstotliwości. Nie są to jednak funkcjonalności o ogólnym zastosowaniu, jak HMI czy SCADA, lecz narzędzia wspierające diagnostykę i prace serwisowe. **Warstwa trzecia to ostatnia warstwa obszaru OT.**

Warstwa czwarta dotyczy planowania produkcji (ang. *planning*, planowanie). Będą to systemy informatyczne odpowiedzialne za kontrolę surowców niezbędnych do zachowania

ciągłości procesu produkcyjnego, a także za planowanie zamówień zarówno surowców, jak i gotowego produktu. Systemy informatyczne umożliwiające podgląd procesu na podstawie rzeczywistych danych to systemy MES (ang. *manufacturing execution system*, system zarządzania produkcją). Korzystając z informacji dostarczonych przez oprogramowanie, kierownictwo może podejmować trafne decyzje co do przyszłych działań w obrębie firmy. Mowa tu przede wszystkim o optymalizacji procesu produkcyjnego przez precyzyjne składanie zamówień oraz minimalizowanie czasów przestoju wynikających z braków magazynowych. Dane produkcyjne systemów MES mogą pochodzić zarówno z paneli HMI czy systemów SCADA, jak i bezpośrednio ze sterowników PLC.

Ostatnia, piąta warstwa piramidy to systemy ERP (ang. *Enterprise Resource Planning*, planowanie zasobów przedsiębiorstwa), które w przeciwieństwie do systemów MES nie skupiają się wyłącznie na procesie produkcyjnym, lecz także na całościowym i kompleksowym prowadzeniu przedsiębiorstwa. Pozwalają one na obsługę finansów, działań kadrowych, sprzedaż wyprodukowanych dóbr czy obsługę wypłat pracowników. Integracja systemów ERP z systemami MES jest niezbędna do zachowania pełnej informacji dotyczącej działań wewnątrz firmy.

Mam nadzieję, że powyższy opis chociaż w minimalnym stopniu przybliżył Ci to, jak ogromną dziedziną jest automatyka przemysłowa. Pomimo szczerych chęci wiele tematów musiałem ograniczyć do absolutnego minimum po to, abyś mógł zobaczyć cały „obrazek” i to, gdzie dokładnie w hierarchii znajdują się tytułowe sterowniki PLC. Z drugiej strony uważam, że sama świadomość chociażby istnienia stosowanych protokołów komunikacyjnych pozwoli Ci trochę pewniej wkroczyć w zagadnienia związane z programowaniem PLC.

Jeżeli trafiłem na doświadczonego w przemyśle Czytelnika, to mam nadzieję, że powyższy rozdział co nieco rozjaśnił, pozwolił usystematyzować pewne kwestie i być może wprowadził kilka nowych informacji. Jeżeli natomiast kontakt z automatyką masz po raz pierwszy, to ufam, że nadmiar skrótów i informacji absolutnie nie zniechęcił Cię do dalszej lektury.

Myszę, że na tym możemy zakończyć wstęp teoretyczny i przejść do sedna — najwyższy czas skupić się na programowaniu sterowników PLC, ich budowie, językach programowania i samym tworzeniu programów.

PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Twoje drzwi do świata automatyki przemysłowej

Programowanie PLC to już nie tylko automatyzacja procesu. To dziedzina, która skupia wokół siebie szereg technologii, narzędzi i języków programowania. W nowoczesnych aplikacjach to również bezpośrednia integracja z systemami IT, logowanie danych do informatycznych baz danych — wszystko z uwzględnieniem wytycznych związanych z cyberbezpieczeństwem. Idea stosowania sterowników PLC pozostaje jednak niezmienna: to urządzenia, których zadaniem jest automatyzacja procesu i które ze względu na wysoką elastyczność stosowane są w praktycznie każdej gałęzi przemysłu. Przemysłu, w którym ciągle brakuje specjalistów i osób chętnych do nauki nowych umiejętności. Może to właśnie Ty dołączysz do tego grona?

Jeżeli zastanawiasz się nad wyborem ścieżki zawodowej, rozważasz zmianę branży i myślisz o automatyce, studiujesz, jesteś uczniem szkoły średniej lub po prostu chcesz się dowiedzieć, czym jest automatyka i na czym polega programowanie PLC — koniecznie sięgnij po tę książkę. Dzięki niej zrozumiesz zależności pomiędzy technologią, mechaniką i elektryką, a także poznasz sposób, w jaki tworzy się programy realizujące w pełni automatyczne algorytmy sterowania. Na polskim rynku najpopularniejsze są rozwiązania firmy Siemens, w związku z czym opis konfiguracji sprzętowej i instrukcji programowych jest omawiany w środowisku **Siemens TIA Portal V20**. Poznaj **sterowniki PLC SIMATIC S7–1500 i S7–1200**, a także najnowsze **panele operatorskie HMI WinCC Unified**.

Ta książka:

- pomoże Ci postawić pierwsze kroki w świecie automatyki przemysłowej
- otworzy przed Tobą możliwości samodzielnego rozwijania zdobytych umiejętności
- przedstawi najpopularniejsze sterowniki PLC w polskim przemyśle
- nauczy Cię tworzyć własne algorytmy sterowania

Marcin Podsiadły — magister inżynier automatyki i robotyki, absolwent Politechniki Śląskiej, ekspert w dziedzinie programowania sterowników PLC, systemów wizualizacji HMI/SCADA oraz integracji rozwiązań IT/OT. Przeprowadził ponad 200 szkoleń dla przeszło 1300 uczestników w największym centrum szkoleniowym w Polsce. Prowadzi własną działalność, w ramach której skupia się na tworzeniu kompleksowych rozwiązań oraz standardów dla firm produkcyjnych. Wykładowca na studiach podyplomowych, autor unikalnych materiałów eksperckich z zakresu automatyki przemysłowej, zwycięzca pierwszej edycji Mistrzostw Polski Programistów PLC (2018).

Patronat:

Helion 	KOD KORZYŚCI Sięgnij po więcej! ▶ 
 helion.pl	ISBN 978-83-289-1457-5
 HELION S.A. ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	 9 788328 914575
Cena: 69,00 zł	

SIEMENS

CENTRUM SZKOLEŃ INŻYNIERSKICH