

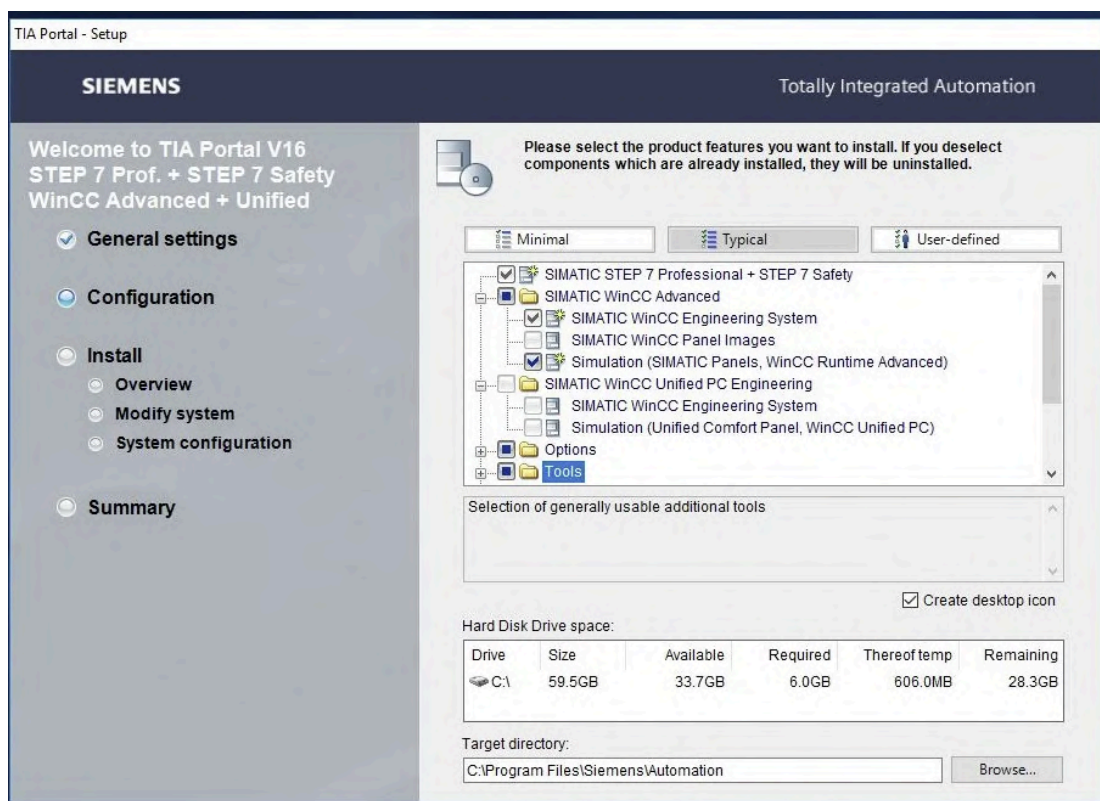
TIA V19



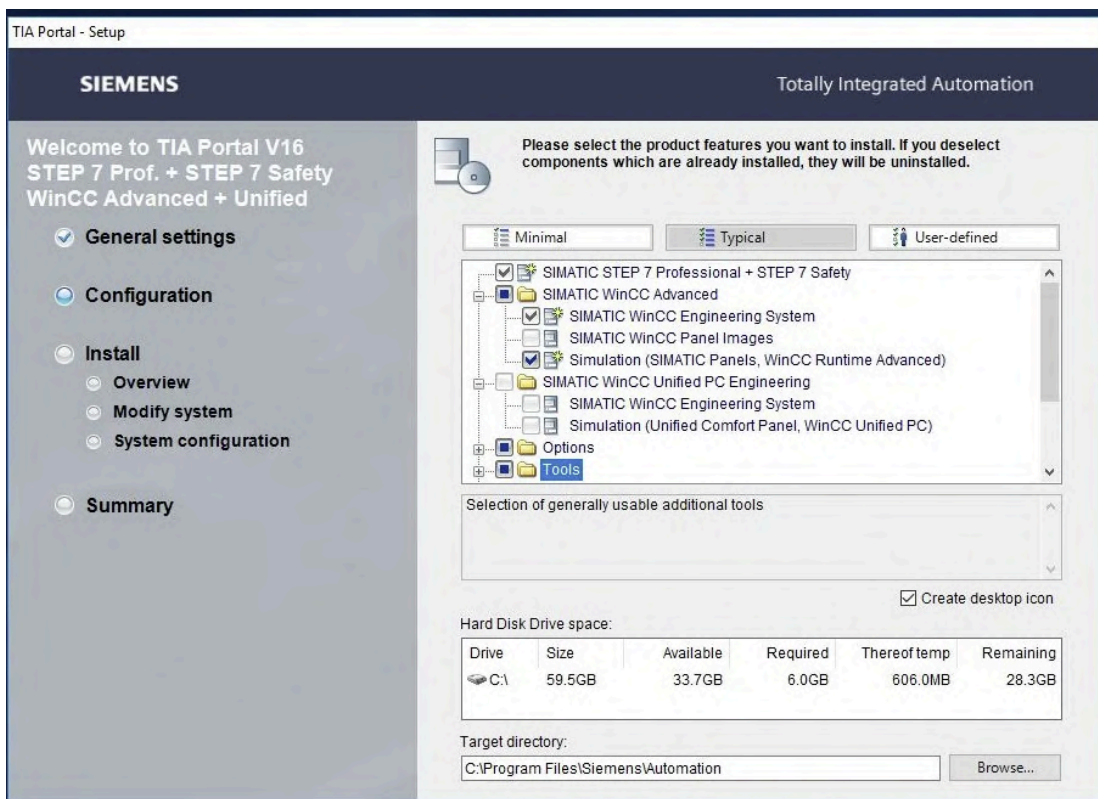
PROGRAMOWANIE PLC

Poradnik przygotowany
przez ControlByte

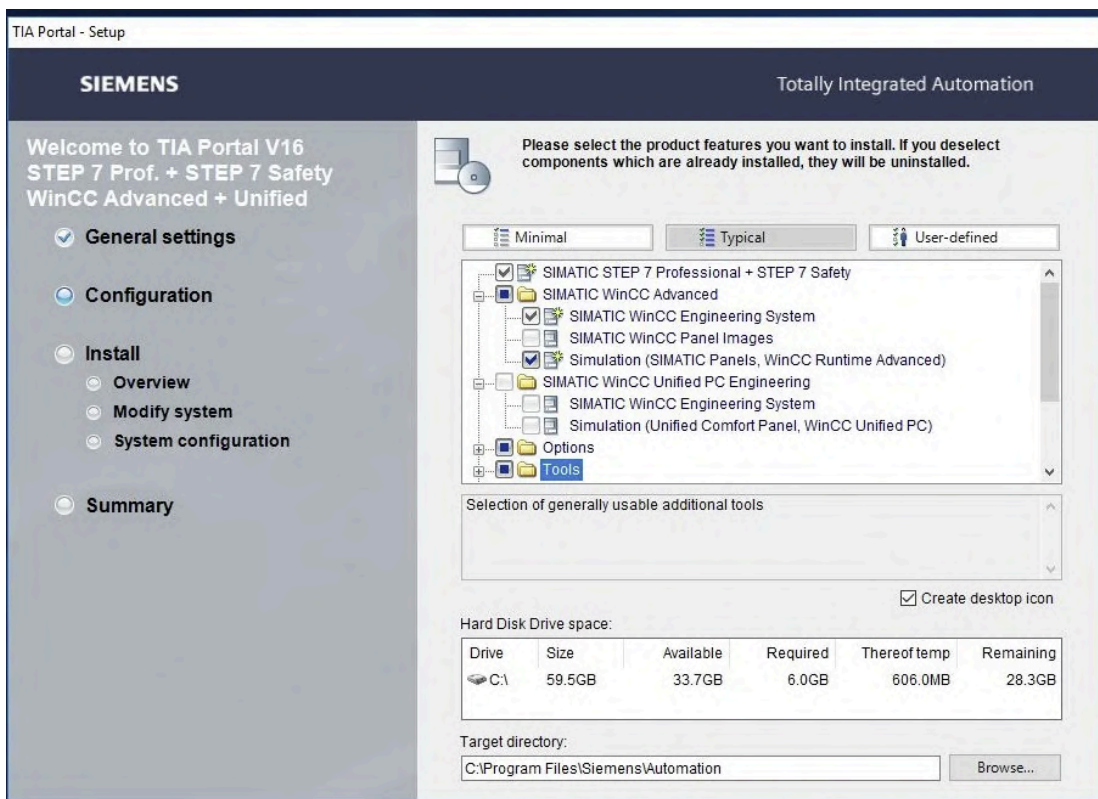




ROZDZIAŁ 1 WPROWADZENIE.....	3
1.1 Dlaczego wybrać tego e-booka?.....	3
1.2 Dla kogo jest ten e-book?.....	4
1.3 Jak wynieść z tej książki jak najwięcej?.....	5
1.4 Dlaczego programowalne sterowniki logiczne (PLC)?.....	6
1.5 Różne ścieżki do zostania programistą PLC.....	8
ROZDZIAŁ 2 PIERWSZE KROKI Z TIA PORTAL I PROGRAMOWANIEM PLC SIEMENS.....	11
2.1 Wprowadzenie do TIA Portal.....	11
2.2 Jaką wersję TIA Portal zainstalować?.....	11
2.3 Zarządzanie wersją TIA Portal w projektach automatyki.....	11
2.4 Darmowa licencja TIA Portal.....	13
2.5 Od jakiej wersji TIA Portal zacząć naukę programowania PLC?.....	13
2.6 Darmowy symulator sterownika PLC Siemens - PLCSIM.....	13
2.6 Instalacja TIA Portal.....	14
ROZDZIAŁ 3 JĘZYKI PROGRAMOWANIA DLA PLC.....	20
3.1 Jaki jest najpopularniejszy język programowania sterowników PLC?.....	21
3.2 Ladder (LAD) - język drabinkowy.....	22
3.3 Structured Text (ST) - język tekstowy.....	24
3.4 Function Block Diagrams (FBD) - język blokowy.....	25
3.5 Sequential Function Charts (SFC) - język blokowo-sekwencyjny.....	26
3.6 Instruction Lists (IL) - język w postaci jednoliniowych instrukcji.....	27
ROZDZIAŁ 4 PODSTAWY JĘZYKA LADDER (LAD).....	29
4.1 Jak zrozumieć język drabinkowy?.....	29
4.2 Struktura w języku LAD.....	32



4.3 Podstawowe symbole w języku LAD.....	35
4.4 Zmienne i typy zmiennych.....	38
Podstawowe typy zmiennych w TIA Portal.....	39
ROZDZIAŁ 5 Projekty i pytania dotyczące rozmów kwalifikacyjnych.....	41
5.1 Jak zaprogramować obwód sterujący silnikiem start/stop przy użyciu Ladder?..	41
5.2 Jak zaprogramować siłownik pneumatyczny z 2 czujnikami i zaworem 5/2?.....	42
5.3 Jak zaprogramować system sterowania zbiornikiem wody z 2 przełącznikami poziomymi i pompą?.....	43



ROZDZIAŁ 1 | WPROWADZENIE

1.1 Dlaczego warto wybrać tę książkę?

W ostatnich latach wspieraliśmy tysiące inżynierów, techników i elektryków w całej Europie w nauce programowania i zdobywaniu pracy w branży automatyki. Wielu z nich znalazło zatrudnienie w czołowych europejskich firmach, takich jak Siemens, ABB czy Schneider Electric, jak również u innych wiodących światowych producentów.

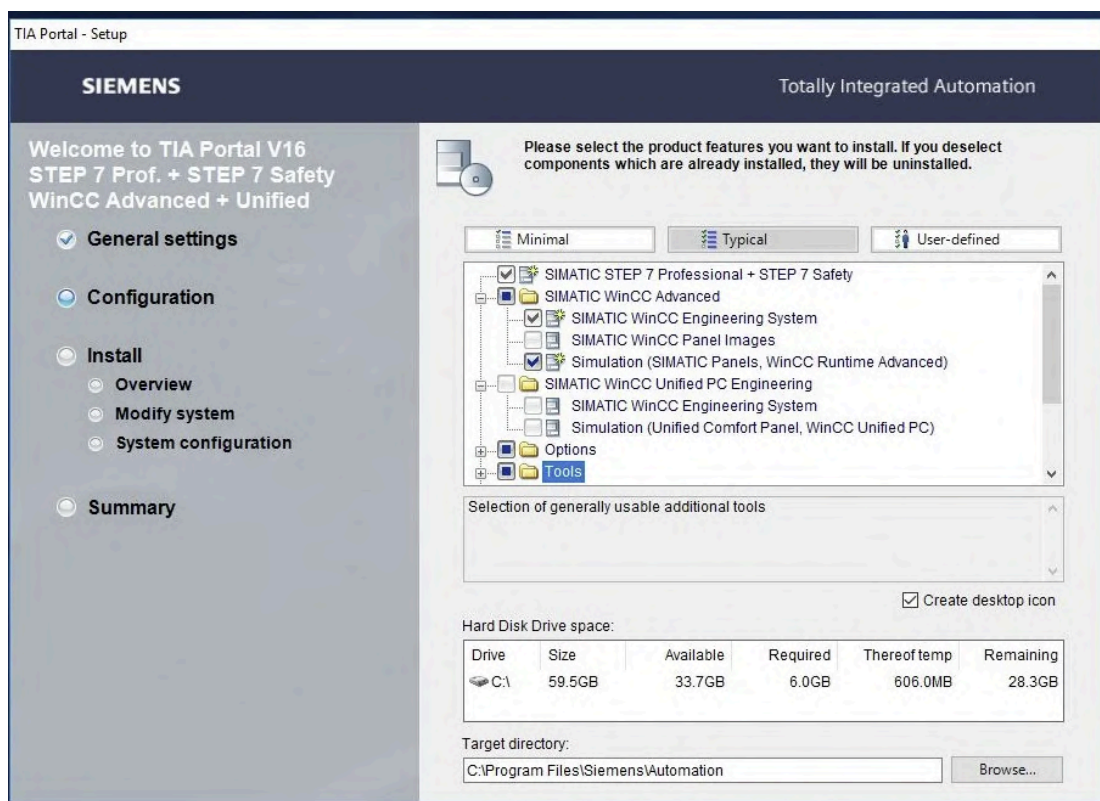
Jednym z powszechnych błędnych przekonań, które zauważyliśmy, jest to, że wiele osób uważa, iż muszą opanować skomplikowane zagadnienia, takie jak pętle PID, skalowanie wejść analogowych czy zaawansowana obsługa pętli FOR, aby kwalifikować się do pracy w programowaniu automatyki. Często prowadzi to do zamieszania, ponieważ uczniowie starają się przyswoić zbyt dużo, zbyt szybko, co ostatecznie powoduje utratę motywacji.

Ta książka została zaprojektowana tak, aby krok po kroku prowadzić Cię przez kluczowe zagadnienia związane z programowaniem automatyki, pomagając Ci zbudować pewność siebie potrzebną do wejścia na rynek pracy. Opanowanie przedstawionych tu podstawowych tematów pozwoli Ci wyprzedzić 90% kandydatów na rynku pracy. Wielu programistów ma trudności z tymi podstawami, które są jednocześnie umiejętnościami najbardziej pożądanymi przez pracodawców.

Dlaczego wejście do świata automatyki i programowania jest wyzwaniem?

Wejście do świata automatyki jest trudne z kilku powodów:

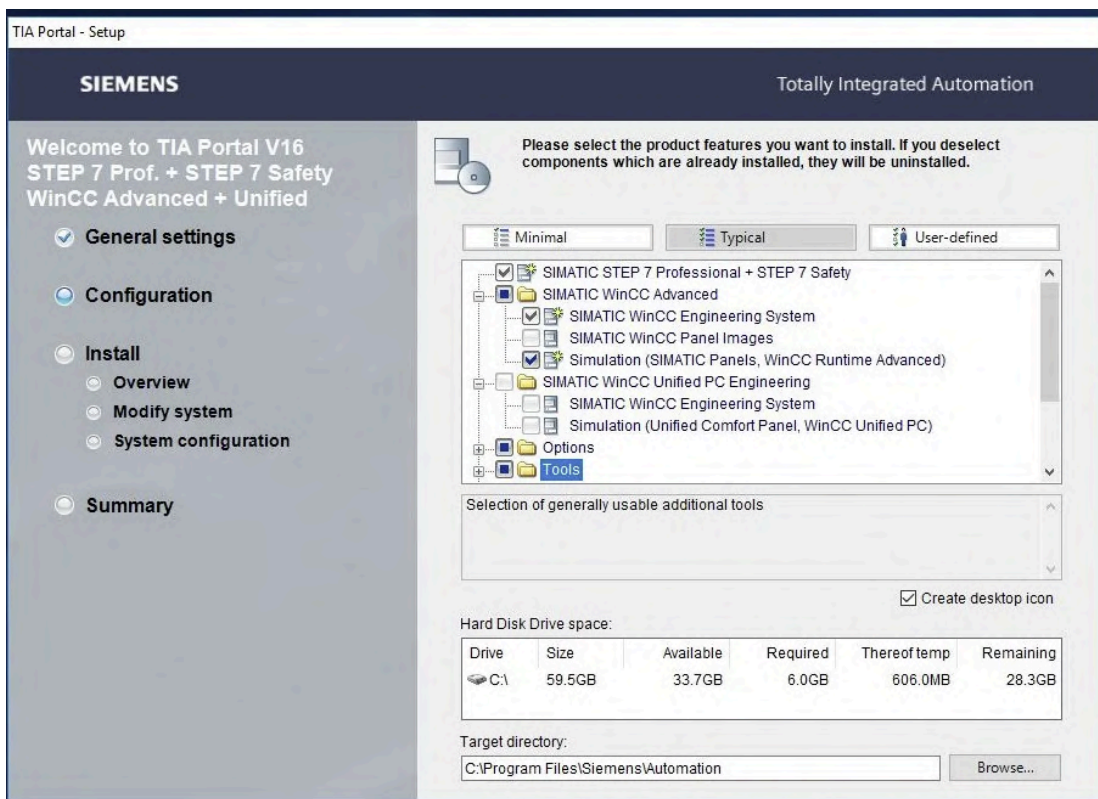
- Wiedza związana z tworzeniem oprogramowania jest chroniona przez firmy produkujące maszyny i systemy automatyki.
- Przykładowe programy nie są łatwo dostępne w Internecie.
- Oprogramowanie do automatyki (np. Siemens TIA Portal) jest drogie.



- Szkolenie z podstaw programowania jest drogie (od 1 500 do 5 000 EUR).
- Sprzęt jest drogi – podstawowy zestaw startowy do automatyki z panelem operatorskim kosztuje około 1 000 EUR.

Nazywam się Kamil Biedulski i jestem autorem tej książki. Pamiętam swoje początki (ponad 10 lat temu), kiedy byłem stażystą i chciałem zacząć programować automaty. Szkolenie z programowania było dla mnie nieosiągalne – kosztowało więcej niż dwumiesięczna pensja! Doświadczony inżynier automatyki podzielił się ze mną swoimi notatkami ze szkolenia, na które wysłała go firma – tak zaczęła się moja przygoda. Pamiętam, jak kopiarka pracowała na pełnych obrotach, kiedy kopiowałem ponad 1 000 stron materiałów szkoleniowych. Pracując w tej firmie, miałem dostęp do programów maszyn, gdzie mogłem uczyć się profesjonalnych programów.

Tak wyglądały moje wczesne, niezwykle mozolne początki. Później znalazłem pracę w firmie integratorskiej i pracowałem nad różnymi projektami – testowałem maszyny, systemy technologiczne i stacje zrobotyzowane. Zdobyte doświadczenie przerodziłem w platformę szkoleniową ControlByte oraz własną firmę, z zespołem inżynierów i programistów, dostarczającą maszyny na zamówienie.

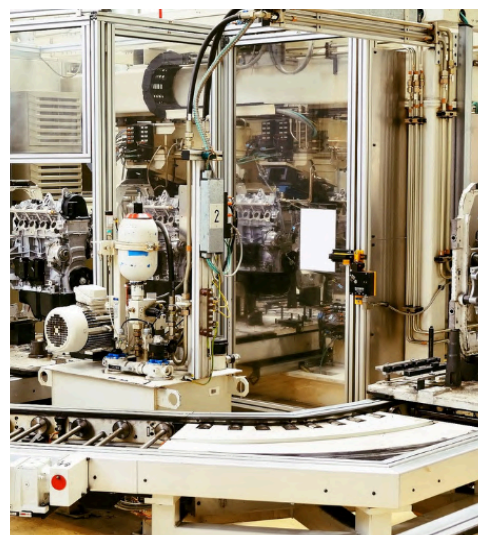


1.2 Dla kogo jest ta książka?

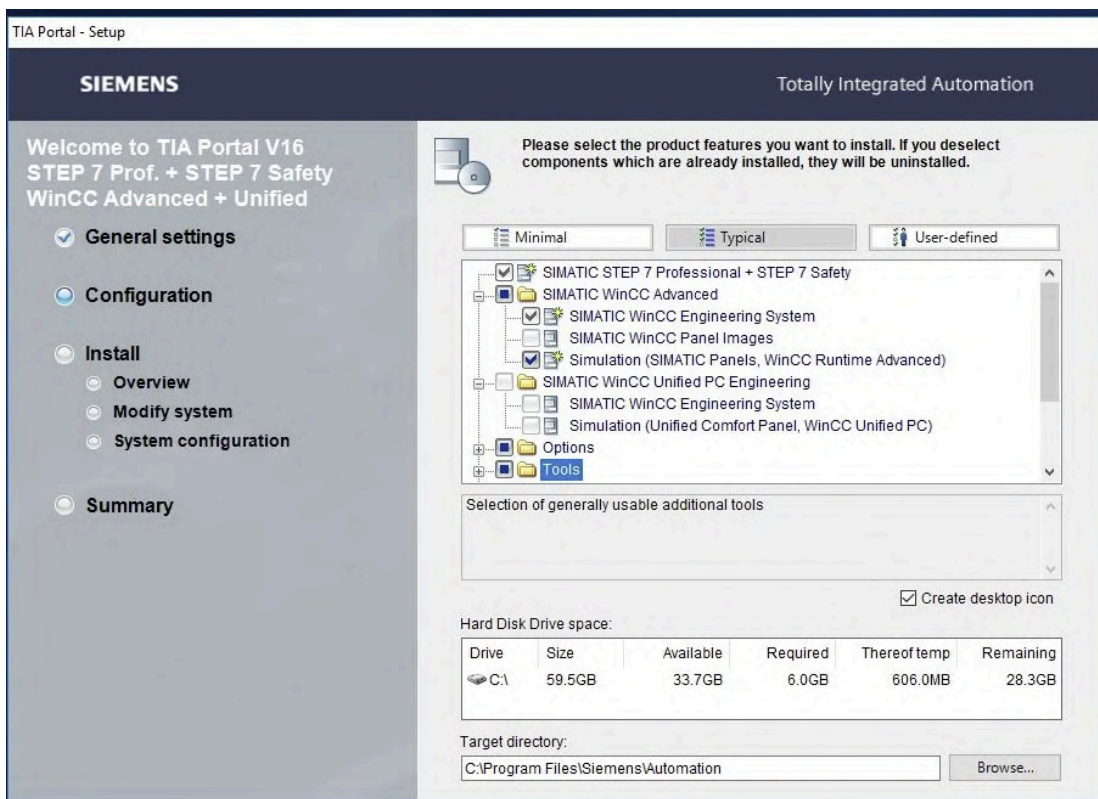
Chociaż byłoby świetnie napisać książkę, która trafi do każdego, zdecydowaliśmy się skupić konkretnie na studentach i profesjonalistach, którzy dopiero zaczynają swoją przygodę z programowaniem automatyki. Naszymi docelowymi czytelnikami są osoby, które chcą zbudować solidne podstawy umiejętności, które pomogą im zdobyć stanowisko inżyniera ds. systemów sterowania lub awansować w życiowej karierze.

Programowanie automatyki może otworzyć drogę do aplikowania o wymarzoną pracę, taką jak:

- Programista automatyki
- Inżynier ds. systemów sterowania
- Inżynier automatyki
- Inżynier elektryk (ze specjalizacją w systemach automatyki)
- Inżynier utrzymania ruchu (ze specjalizacją w automatyce)
- Technik automatyki przemysłowej
- Inżynier ds. kontroli procesów
- Programista SCADA
- Technik ds. instrumentacji i systemów sterowania
- Inżynier integracji systemów



Możliwość nauki programowania automatyki otworzyła wielu osobom drogę do lepiej płatnych stanowisk, stabilnych warunków pracy oraz doskonałych perspektyw rozwoju zawodowego.



Programmable Logic Controllers (PLC) / Automation Jobs by Salary

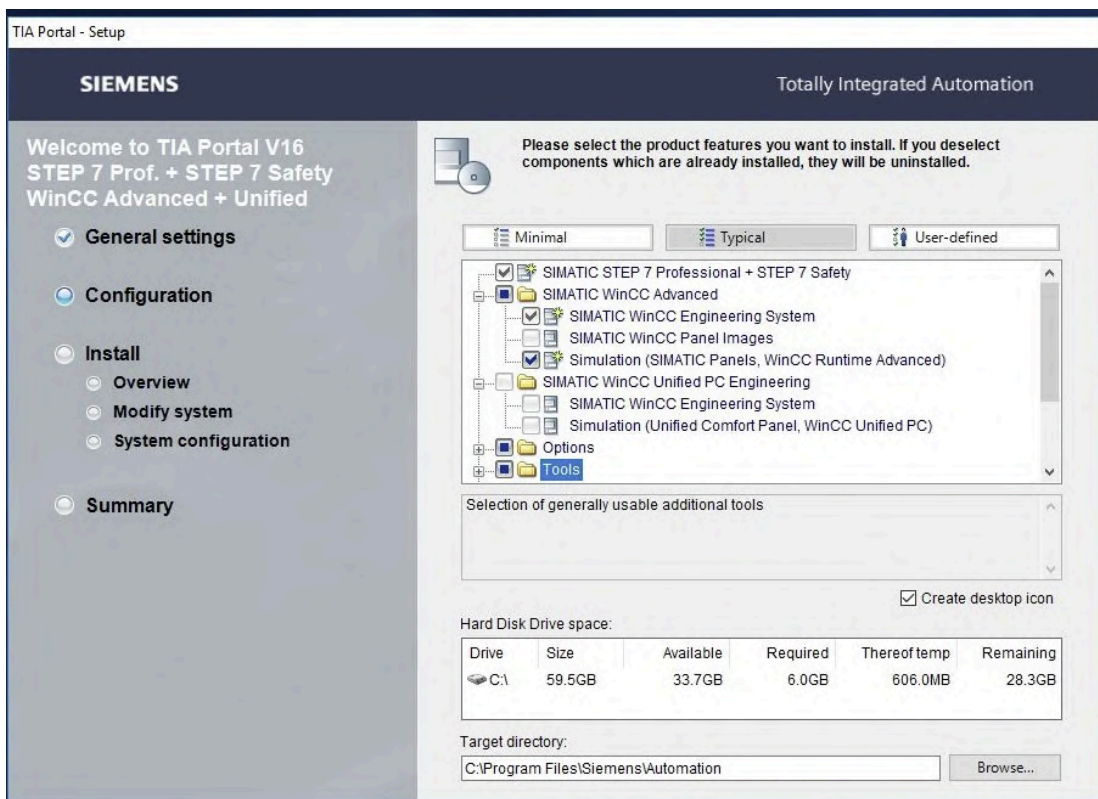
Job Title	Range	Average
Automation Engineer	€42k - €72k	€54,223
PLC Programmer	€45k - €68k	€52,425
Electrical Engineer	€37k - €81k	€49,761
Controls Engineer	€44k - €86k	€61,373
Robotics Engineer	€51k - €59k	€53,412
Control Systems Engineer	€51k - €66k	€62,288

Salaries, źródło: <https://www.payscale.com/>

Celowo pominęliśmy złożone szczegóły teoretyczne, aby skupić się na praktycznej wiedzy i umiejętnościach gotowych do wykorzystania w pracy. Naszym celem jest dostarczenie narzędzi, które pozwolą Ci skutecznie działać w środowisku zawodowym, gdzie najważniejsze jest sprawne wykonywanie zadań.

Ta książka zawiera również materiały do nauki praktycznej, takie jak:

- pytania z rozmów kwalifikacyjnych do przećwiczenia,



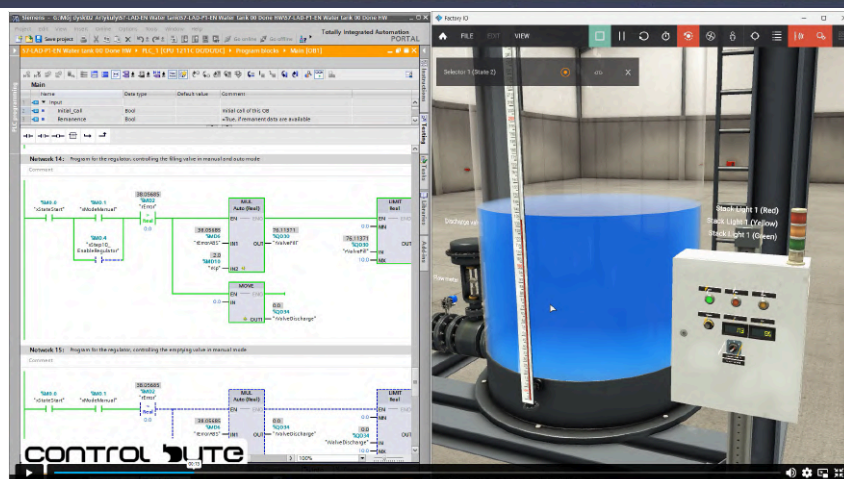
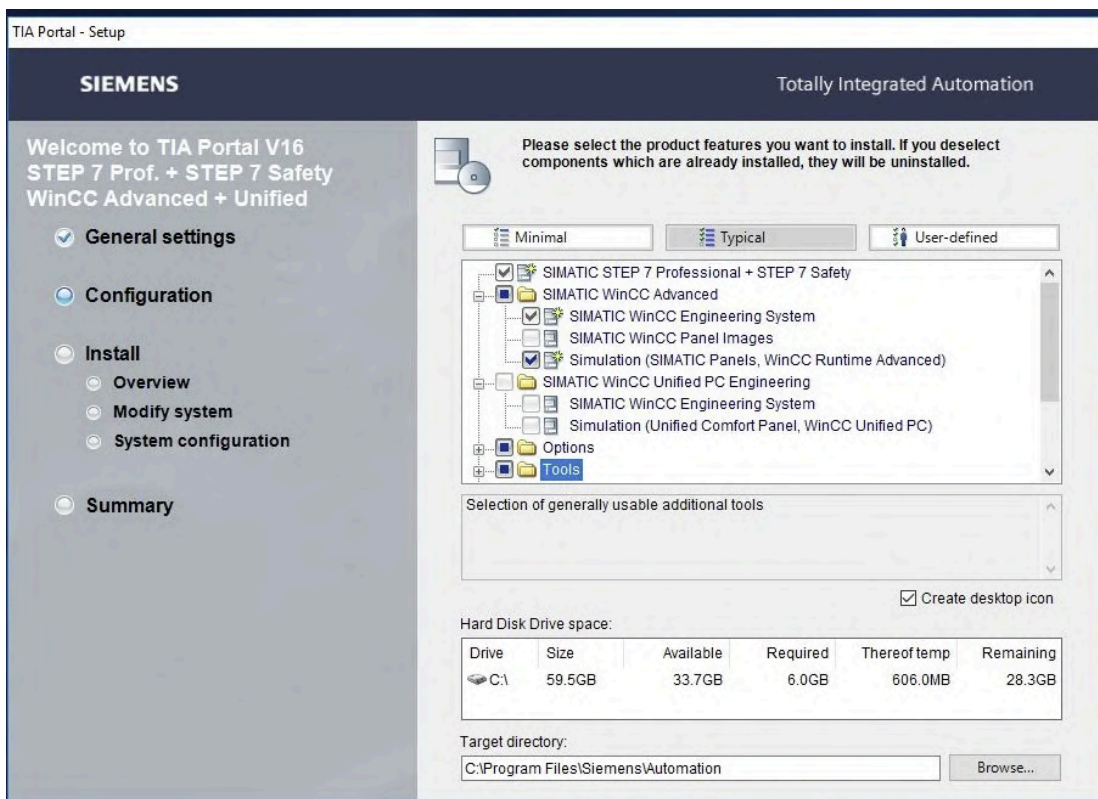
- mini projekty,
- oraz linki do samouczków wideo, które ułatwią dalszą naukę i wizualizację kluczowych pojęć.

Mamy nadzieję, że po przeczytaniu tej książki poczujesz się gotowy, aby rozpocząć swoją drogę do zostania pewnym siebie i kompetentnym programistą automatyki, z solidną znajomością podstaw i motywacją do dalszego rozwoju.

1.3 Jak najlepiej wykorzystać tę książkę?

W miarę postępów w tej książce niektóre tematy będą łatwe do zrozumienia, podczas gdy inne mogą wymagać nieco więcej wysiłku—i to jest całkowicie normalne! Aby zapewnić Ci zasoby potrzebne do zdobycia pracy w programowaniu automatyki, wyróżniliśmy kilka narzędzi i opcji wsparcia dostępnych na **ControlByte.pl**, które mogą Ci pomóc w trakcie nauki:

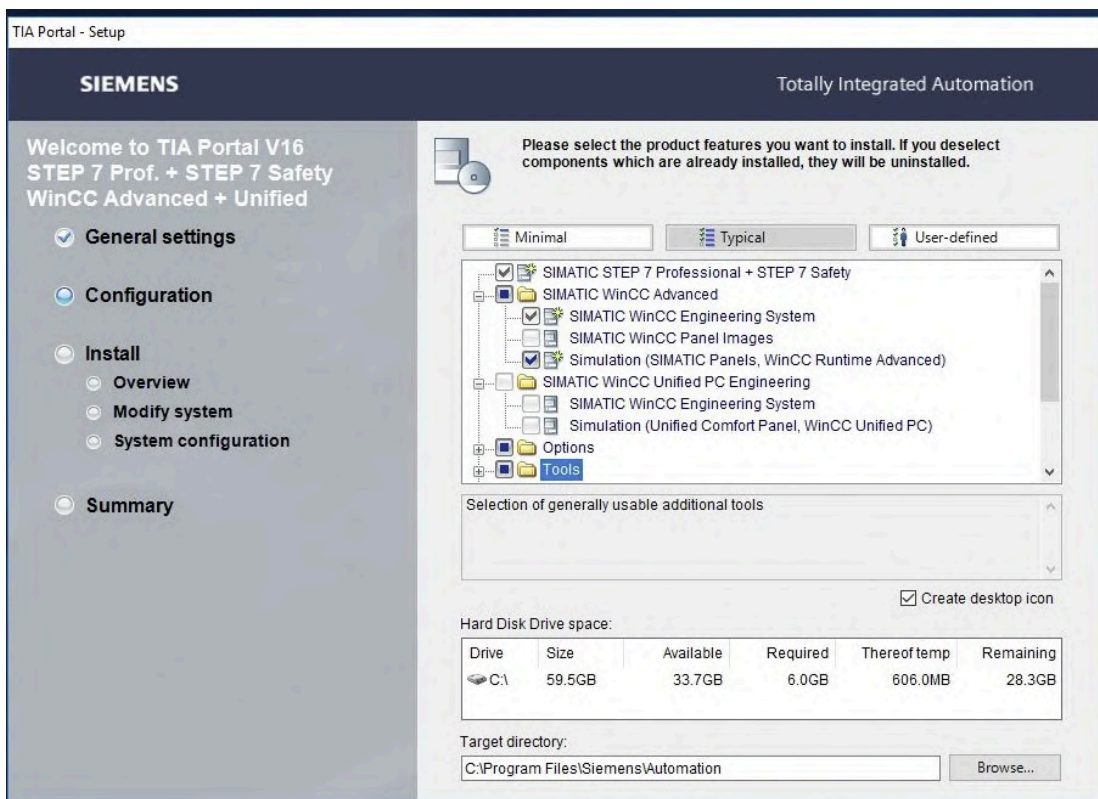
1. **Darmowe tutoriale z programowania automatyki** – W miarę odkrywania nowych tematów, możesz zgłębiać wiedzę dostępną w naszych tutorialach. Stworzyliśmy ponad 100 szczegółowych, technicznych poradników na szeroką gamę tematów. Te tutoriale są dostępne bezpłatnie, więc możesz kontynuować naukę we własnym tempie.
<https://www.controlbyte.pl/blog/programowanie-plc-w-jezyku-scl-na-przykladach-podstawy-czesc-1/>
2. **Webinary na żywo z ControlByte** – Bezpłatne spotkania online na żywo
<https://youtube.com/live/GTuDgLUOTtU>



1.4 Dlaczego sterowniki logiczne (PLC)?

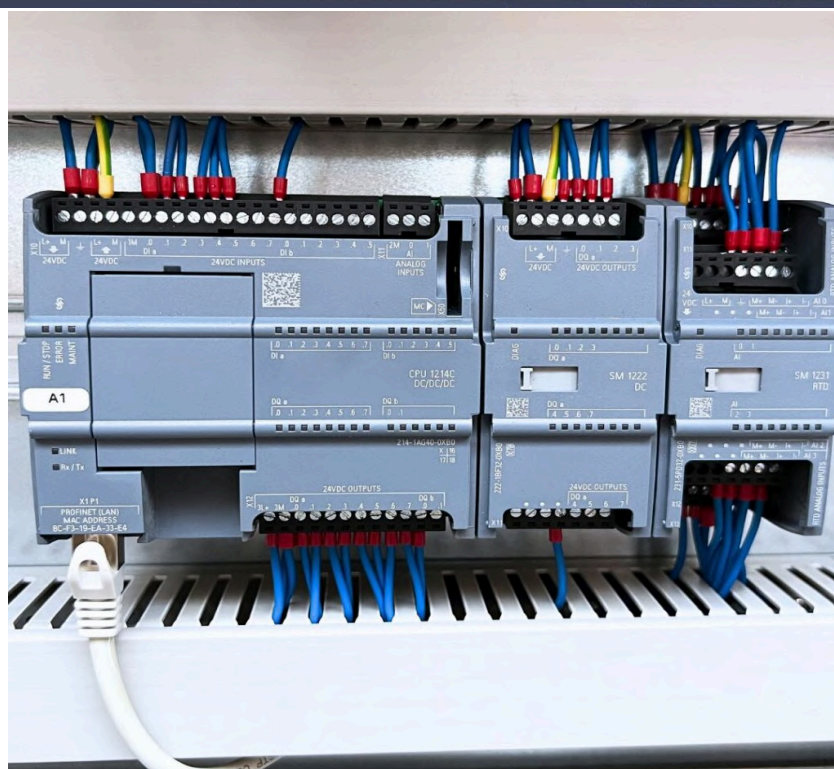
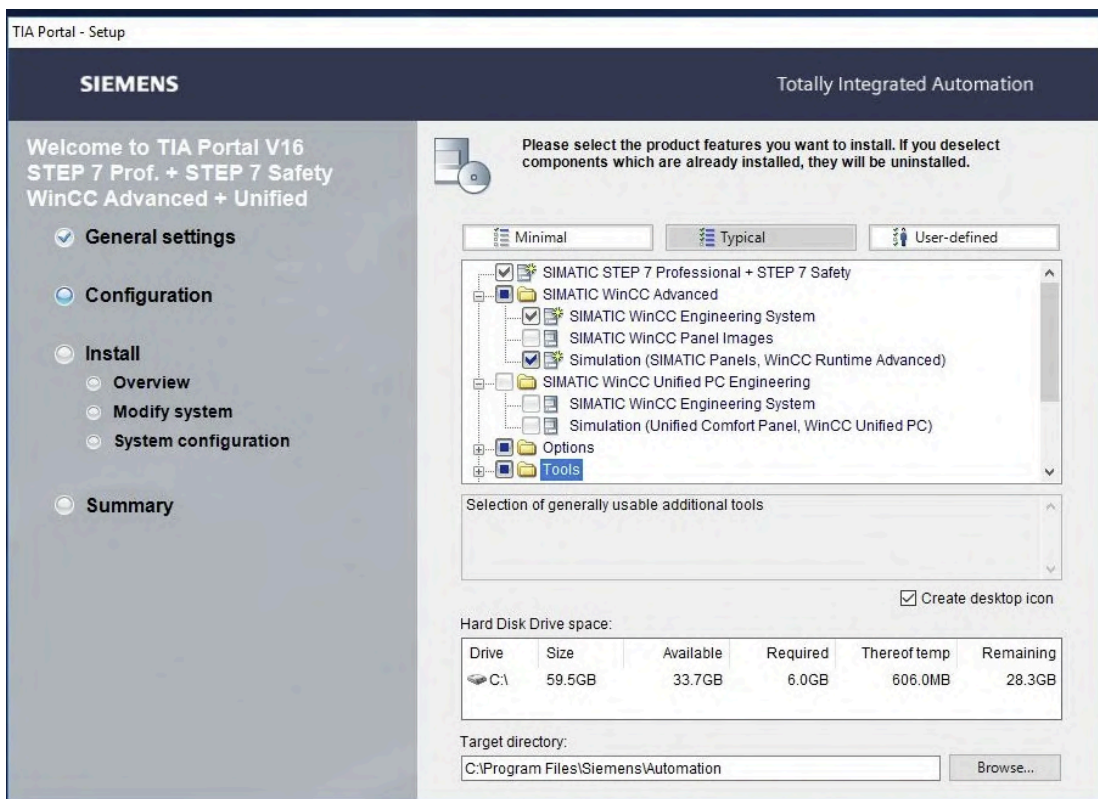
Produkcja większości produktów konsumenckich, od żywności po elektronikę, odbywa się w zautomatyzowanych zakładach. Towary te są następnie dystrybuowane i dostarczane za pomocą zautomatyzowanych systemów, co zmniejsza zapotrzebowanie na udział człowieka. Automatyzacja w produkcji nie jest nowym pomysłem, ponieważ maszyny stale przewyższają ludzi pod względem wydajności, precyzji i powtarzalności w produkcji na dużą skalę. W związku z tym trwają nieustanne starania, aby zautomatyzować każdy etap procesu produkcyjnego.

W różnych obszarach produkcji wykorzystywane są różnorodne systemy automatyzacji, wszystkie z ukierunkowaniem na osiągnięcie podobnych celów: niezawodności, powtarzalności i łatwości utrzymania. To zapotrzebowanie doprowadziło do powstania sterowników logicznych (PLC) w poprzednim stuleciu.



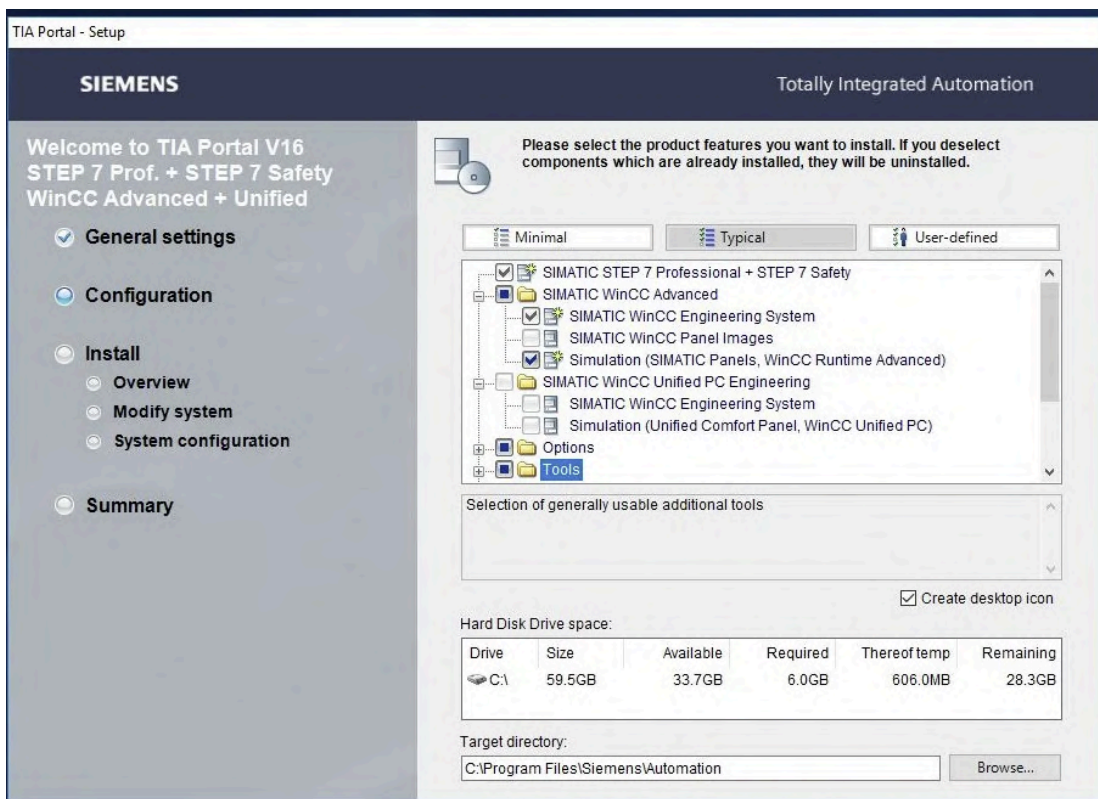
Sterowniki PLC zostały wprowadzone, aby zastąpić skomplikowane systemy oparte na przekaźnikach, oferując prostszy sposób zmiany operacji i zwiększenie niezawodności systemu. Zostały zaprojektowane jako opłacalna alternatywa dla przekaźników, pojawiając się kilka lat po tym, jak tranzystory stały się powszechne. W istocie, sterownik PLC to niezwykle niezawodny komputer, zdolny do ciągłego wykonywania programów 24/7 w wymagającym środowisku.

Początkowo konfiguracja sterowników PLC była prosta, ale ich złożoność wzrosła wraz z ewolucją potrzeb biznesowych i wzrostem złożoności linii produkcyjnych. W związku z tym przedsiębiorstwa potrzebowały osób wykwalifikowanych w programowaniu PLC, aby projektować, optymalizować i dostosowywać programy do wymagań operacyjnych.



Sterownik PLC Siemens Simatic S7-1200 z modułami IO

Sterownik programowalny (PLC) to w istocie sprzęt odpowiedzialny za zarządzanie i kontrolowanie procesów. Fizycznie, PLC przypomina pudełko wypełnione obwodami elektrycznymi, podobne do komputera stacjonarnego. Zazwyczaj jest zamknięty w obudowie ochronnej, aby zabezpieczyć go przed trudnymi warunkami, z jakimi może się spotkać w środowisku przemysłowym. Często, mówiąc o PLC, mamy na myśli cały system sterowania na hali produkcyjnej, ponieważ PLC są podłączone do różnych urządzeń, takich jak czujniki, silniki, przełączniki i zawory.



Od momentu wynalezienia, PLC znacznie się rozwinęły. Choć nadal są to wytrzymałe maszyny zaprojektowane do sterowania procesami, obecnie integrują funkcje powszechnie spotykane w sektorze IT. Na przykład, PLC oferują teraz łączność Ethernet do zbierania danych, monitorowanie czujników za pomocą technologii takich jak IO-Link oraz połączenia serwerowe przy użyciu protokołów MQTT, między innymi. Krótko mówiąc, PLC ewoluowały daleko poza to, czym były zaledwie kilka dekad temu.

Welcome to TIA Portal V16
STEP 7 Prof. + STEP 7 Safety
WinCC Advanced + Unified

☒ General settings

☐ Configuration

☐ Install

☐ Overview

☐ Modify system

☐ System configuration

☐ Summary



Please select the product features you want to install. If you deselect components which are already installed, they will be uninstalled.

Minimal Typical User-defined

- ☒ SIMATIC STEP 7 Professional + STEP 7 Safety
- ☒ SIMATIC WinCC Advanced
- ☒ SIMATIC WinCC Engineering System
- ☒ SIMATIC WinCC Panel Images
- ☒ Simulation (SIMATIC Panels, WinCC Runtime Advanced)
- ☒ SIMATIC WinCC Unified PC Engineering
- ☒ SIMATIC WinCC Engineering System
- ☒ Simulation (Unified Comfort Panel, WinCC Unified PC)
- ☐ Options
- ☐ Tools

Selection of generally usable additional tools

☒ Create desktop icon

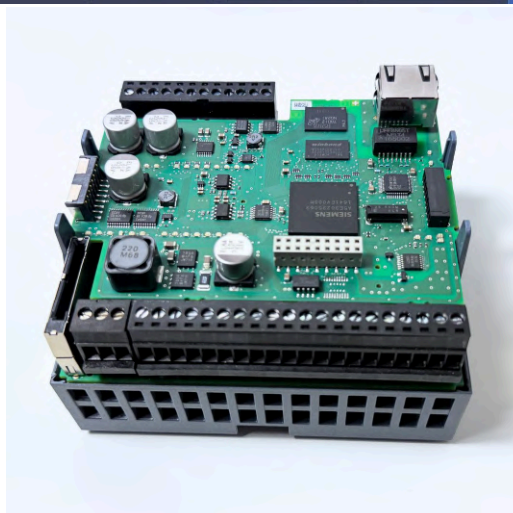
Hard Disk Drive space:

Drive	Size	Available	Required	Thereof temp	Remaining
C:\	59.5GB	33.7GB	6.0GB	606.0MB	28.3GB

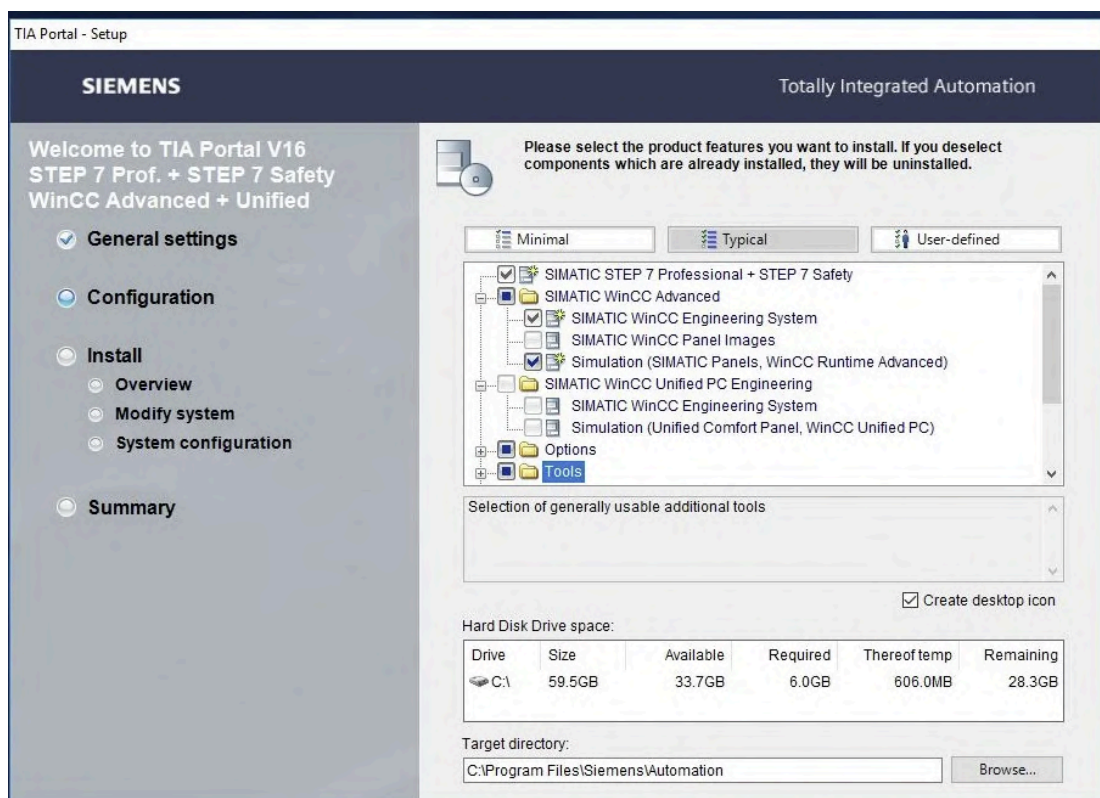
Target directory:

C:\Program Files\Siemens\Automation

Browse...



Demontaż jednostki CPU S7-1200



Ponieważ zakłady produkcyjne na całym świecie w dużym stopniu polegają na sterownikach PLC, a technologia ta wciąż ewoluuje, rośnie zapotrzebowanie na ekspertów zdolnych do opracowywania, wspierania i zarządzania tymi systemami.

1.5 Różne ścieżki do zostania programistą PLC

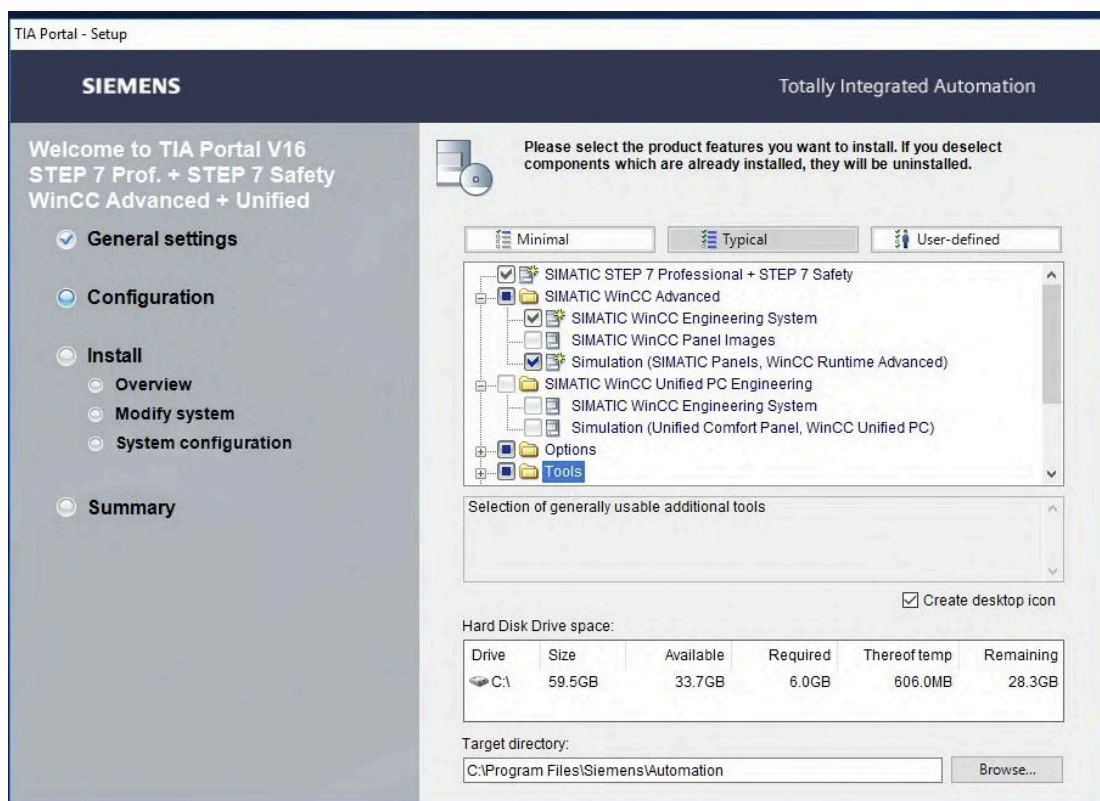
Istnieje wiele sposobów, aby zostać wykwalifikowanym programistą PLC. Możesz zdobyć tradycyjne wykształcenie uniwersyteckie, uczyć się za pośrednictwem samouczków online lub zapisać się na kurs, który po ukończeniu przyznaje certyfikat programowania PLC. Która opcja jest jednak najlepsza, a które certyfikaty mają największą wartość?

Stopnie Inżynieryjne i Technologie Inżynieryjne

Tytuł licencjata w dziedzinie inżynierii elektrycznej, pokrewnej dyscyplinie lub technologii inżynieryjnej jest powszechną drogą do pracy w obszarze automatyzacji przemysłowej. Wiele uczelni oferuje programy, które obejmują tematy takie jak programowanie PLC, rozwój HMI, robotyka i systemy kontrolne. Po ukończeniu programu absolwenci otrzymują dyplom, który często ma większy prestiż niż standardowy certyfikat programowania PLC, szczególnie z uznawanej uczelni.

Chociaż dyplom uniwersytecki może być właściwym wyborem dla niektórych, zazwyczaj wymaga on zaangażowania na okres trzech do czterech lat, wiąże się z znacznymi wydatkami finansowymi i nie zawsze gwarantuje osiągnięcie pożądanych rezultatów zawodowych. Innym minusem jest to, że nie wszystkie kursy mogą koncentrować się na automatyzacji, a studenci mogą znaleźć się w sytuacji, w której muszą uczyć się przedmiotów takich jak rachunek różniczkowy, algebra liniowa, chemia i inne, które nie są bezpośrednio związane z programowaniem PLC.

Oficjalne certyfikaty i kursy programowania PLC OEM



Wielu producentów OEM, takich jak Allen Bradley, Siemens, Wago i Beckhoff, oferuje certyfikaty i szkolenia z programowania PLC, które koncentrują się na rozwijaniu konkretnych umiejętności. Kursy te odbywają się zazwyczaj osobiście, prowadzone przez specjalistów, i mogą trwać od jednego dnia do kilku tygodni.

Chociaż możliwe jest uzyskanie certyfikatu programowania PLC w ramach tych kursów, często nie ma on takiej wartości, jaką studenci oczekują. Chociaż te sesje uczą uczestników, jak programować PLC, często brakuje im praktycznych projektów, zadań czy testów, które zmuszają studentów do krytycznego myślenia o przedstawionym materiale.

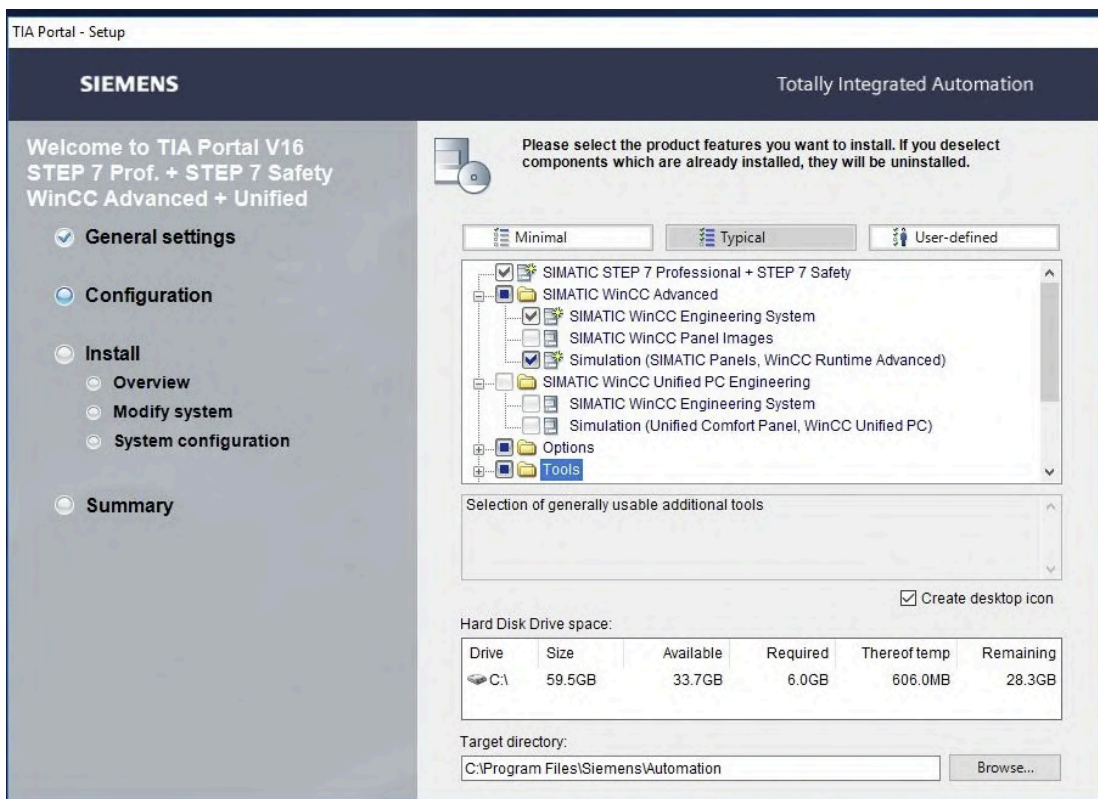
Wielu inżynierów, z którymi rozmawialiśmy, wyraziło rozczarowanie tymi kursami, zarówno pod względem doświadczeń edukacyjnych, jak i tego, jaką wagę miały ich certyfikaty programowania PLC podczas poszukiwania pracy. Wspólne frustracje obejmują:

- Brak ukończonych projektów
- Przeciążenie informacyjne
- Certyfikaty nie cenione przez pracodawców
- Wysoki koszt

Na przykład typowy kurs programowania sterowników PLC firmy Siemens może kosztować od 1500 do 5000 EURO. Chociaż firmy mogą traktować to jako inwestycję w swoich pracowników, często jest to poza zasięgiem osób chcących uczyć się samodzielnie.

Certyfikaty programowania PLC online

Podczas gdy nauka online jest obecna od dłuższego czasu, dziedzina programowania PLC, automatyki przemysłowej i systemów sterowania wciąż się w tym zakresie rozwija. Istnieje szeroki wachlarz jakości w odniesieniu do certyfikatów programowania PLC oferowanych przez różne organizacje.

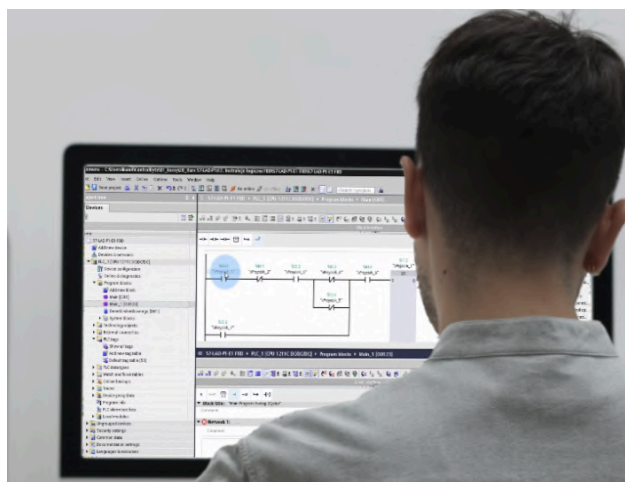


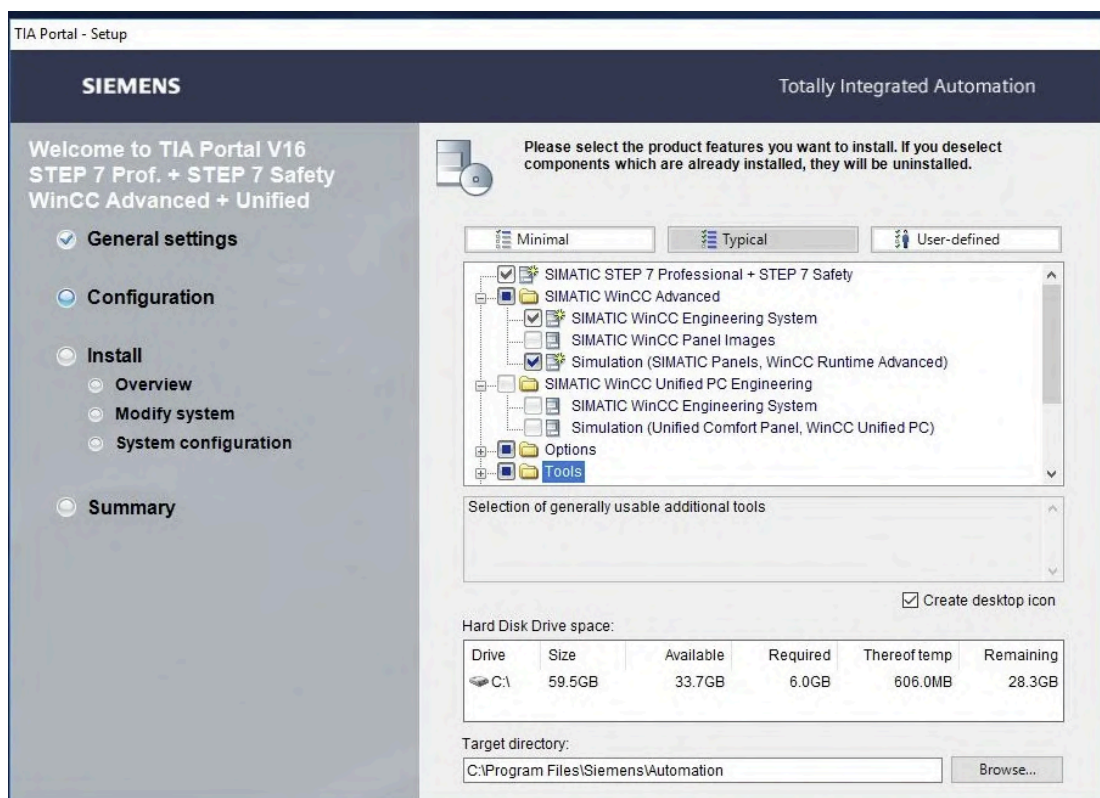
Poniżej znajduje się lista czynników, które warto wziąć pod uwagę przy wyborze instytucji, która zapewni ci certyfikat programowania PLC:

- **Doświadczenie Instruktora** – Czy instruktor ma doświadczenie w branży, czy jest wyłącznie nauczycielem? Choć możliwe jest nauka systemów kontrolnych w sposób akademicki, praktyczne doświadczenie w branży jest bezcenne. Instruktorzy posiadający wiedzę praktyczną będą mogli podkreślić, co jest najważniejsze, podzielić się spostrzeżeniami specyficznymi dla branży oraz pomóc w przygotowaniu się do rzeczywistych wyzwań i rozmów kwalifikacyjnych.
- **Organizacja Wydająca** – Co znajdą potencjalni pracodawcy, jeśli sprawdzą organizację, która wydała certyfikat? Czy dostawca jest uznanym ekspertem w swojej dziedzinie, czy tylko agregatorem programów certyfikacyjnych?
- **Nauka oparta na projektach** – Czy podczas kursu będziesz uczestniczyć w praktycznych projektach? Czy projekty te będą stanowiły wyzwanie podczas przyswajania nowych koncepcji? Te praktyczne zastosowania mają kluczowe znaczenie dla zapewnienia, że możesz zastosować swoją wiedzę w rozwiązywaniu problemów w świecie rzeczywistym, projektowaniu systemów lub tworzeniu od podstaw.
- **Reputacja** – Co inni mówią o programie szkoleniowym? Czy był on wartościowy dla poprzednich uczestników?

Kluczowe korzyści kursy online w zakresie programowania sterowników PLC

Niższy koszt – Kursy online są zwykle znacznie bardziej przystępne cenowo w porównaniu z kursami stacjonarnymi. Zwykle radzimy studentom, aby część zaoszczędzonych pieniędzy zainwestowali w mały zestaw szkoleniowy, który może być przydatny przez wiele lat,



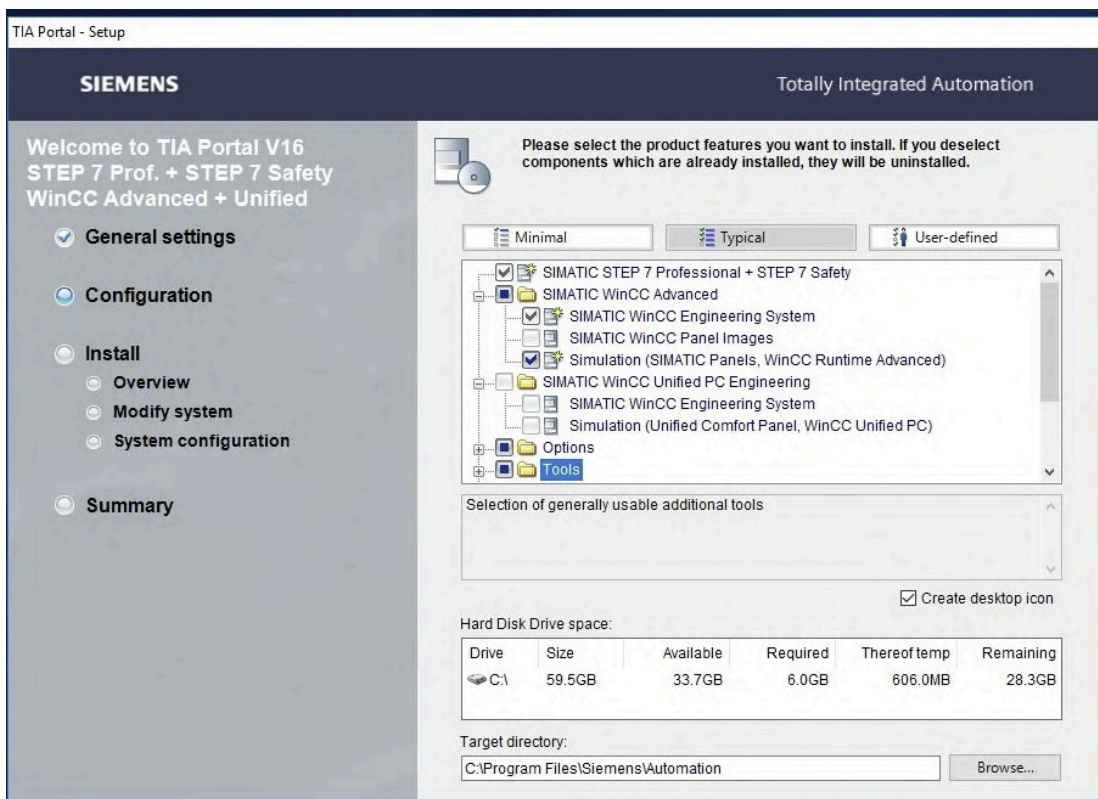


który mogą zaprezentować podczas rozmów kwalifikacyjnych i wykorzystać do dalszego doskonalenia swoich umiejętności po ukończeniu kursu.

Ucz się we własnym tempie – Przez lata wartość elastycznej nauki była niedoceniana. Dzięki materiałom online możesz wstrzymywać i wznowiać lekcje w dowolnym momencie za pomocą kliknięcia przycisku. W przeciwieństwie do tradycyjnych zajęć stacjonarnych, szkolenie online pozwala poświęcić tyle lub mniej czasu na określone tematy, ile potrzeba. Ta elastyczność pomaga opanować koncepcje we własnym tempie i powrócić do wykładów przed rozmowami kwalifikacyjnymi, zapewniając solidne opanowanie materiału.

Projekty, zadania i quizy – Kurs online w zakresie programowania sterowników PLC zapewnia możliwość praktycznej nauki poprzez projekty i zadania, które studenci wykonują samodzielnie. Zadania te są niezbędne do wzmocnienia kluczowych pojęć nauczanych podczas wykładów i zapewnienia głębszego zrozumienia materiału.

Chociaż zachęcamy do zapoznania się z różnymi formami szkoleń w zakresie programowania sterowników PLC, mocno wierzymy, że kompleksowy program online oferuje potrzebną wiedzę oraz certyfikat, który może pomóc w znalezieniu pracy - a wszystko to bez wysokich kosztów związanych z kursami osobistymi.



ROZDZIAŁ 2 | PIERWSZE KROKI Z TIA PORTAL I PROGRAMOWANIEM PLC SIEMENS

2.1 Wprowadzenie do TIA portal

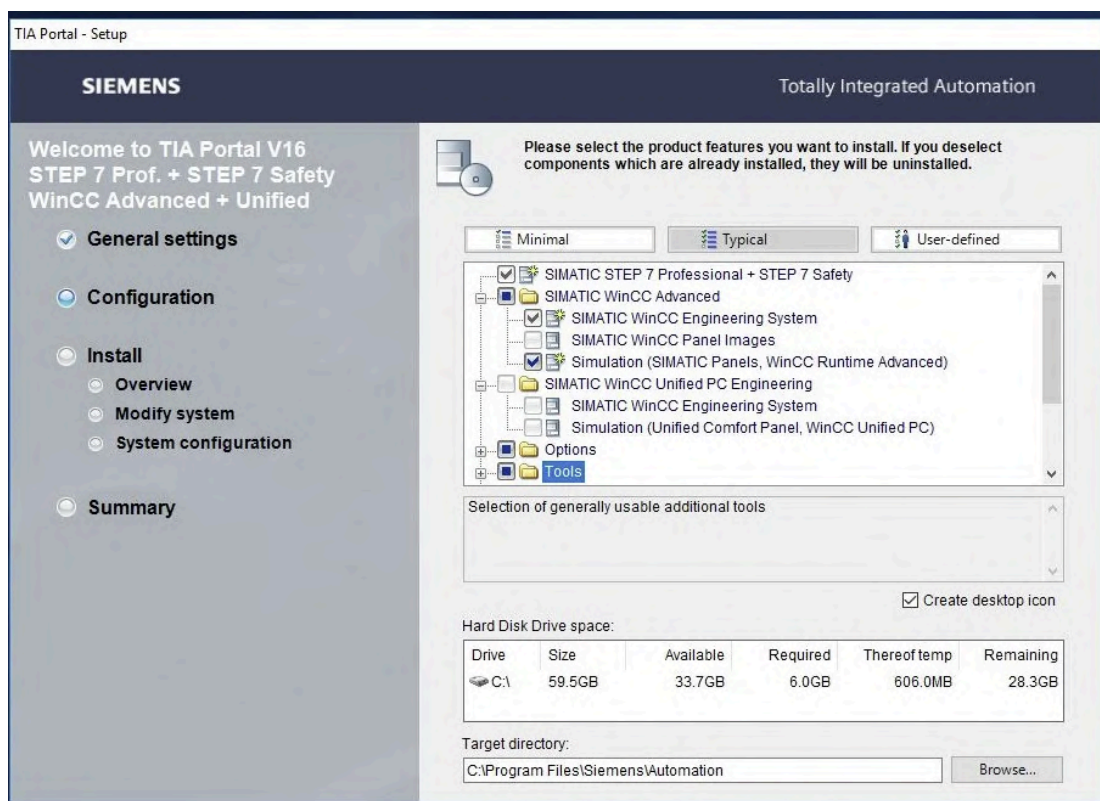
TIA Portal (Totally Integrated Automation Portal) to kompleksowy pakiet oprogramowania firmy Siemens przeznaczony do automatyki przemysłowej, w szczególności do programowania sterowników PLC (Programmable Logic Controllers) firmy Siemens. Integruje programowanie, konfigurowanie i diagnozowanie w jednym środowisku, ułatwiając inżynierom automatyki projektowanie i konserwację systemów sterowania. Niezależnie od tego, czy jesteś początkującym, czy zaawansowanym użytkownikiem, TIA Portal oferuje kompleksowy zestaw narzędzi do tworzenia, symulacji i rozwiązywania problemów w projektach automatyki przemysłowej.

W tym rozdziale zapoznamy Cię z podstawami TIA Portal i przeprowadzimy przez pierwsze kroki programowania sterowników PLC. Pod koniec będziesz zaznajomiony z podstawowymi funkcjami i gotowy do stworzenia swojego pierwszego projektu.

2.2 Którą wersję TIA Portal należy zainstalować?

Od 2015 roku firma Siemens wydała następujące wersje TIA Portal:

1. TIA Portal V13 SP1 (2015) – Dodatek Service Pack dla wersji 13.
2. TIA Portal V14 (2016) – Wprowadzono zwiększoną użyteczność i opcje oparte na chmurze
3. TIA Portal V14 SP1 (2017) – Dalsze ulepszenia i poprawki błędów
4. TIA Portal V15 (2017) – Ważna aktualizacja ze zwiększoną funkcjonalnością sterowników PLC Siemens i integracją SCADA.



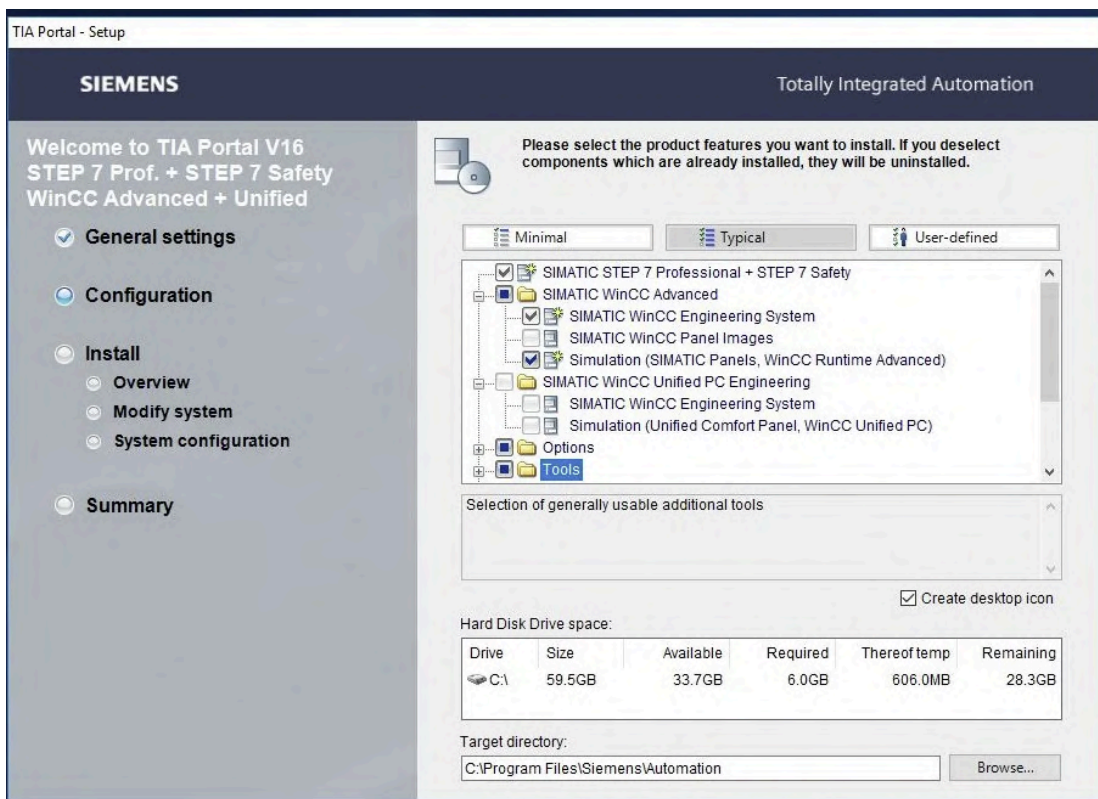
5. TIA Portal V15.1 (2018) – Wydanie w połowie cyklu, poprawiające wydajność i wprowadzające drobne funkcje.
6. TIA Portal V16 (2019) – Znaczące aktualizacje z ulepszonymi bibliotekami, integracją Teamcenter i rozszerzonymi narzędziami symulacyjnymi
7. TIA Portal V17 (2021) – Nowe funkcje, takie jak klient OPC UA, rozszerzone funkcje diagnostyczne i zaawansowane możliwości symulacji.
8. TIA Portal V18 (2022) – Koncentruje się na poprawie łączności w chmurze, narzędziach do współpracy i integracji obliczeń brzegowych.
9. TIA Portal V19 (2023) – Wprowadzono możliwości sztucznej inteligencji i uczenia maszynowego, zaawansowane cyberbezpieczeństwo i rozszerzoną obsługę urządzeń brzegowych.

Tak więc od 2015 roku dostępnych jest 9 wersji TIA Portal (w tym dodatki Service Pack i główne aktualizacje).

2.3 Zarządzanie wersjami TIA Portal w projektach automatyzacji

Jednym z krytycznych aspektów zarządzania projektami automatyzacji za pomocą Siemens TIA Portal jest zapewnienie, że każdy projekt jest utrzymywany przy użyciu tej samej wersji TIA Portal przez cały cykl życia. Projekty automatyzacji są bardzo wrażliwe na zmiany wersji oprogramowania, a niespójne wersjonowanie może prowadzić do poważnych wyzwań w zakresie funkcjonalności, kompatybilności i rozwiązywania problemów.

Kiedy zaczynasz rozwijać lub utrzymywać projekt, ważne jest, aby używać tej samej wersji TIA Portal, która była początkowo używana do tworzenia lub konfigurowania systemu. Oto dlaczego:



- Kompatybilność: Projekty opracowane w jednej wersji TIA Portal mogą nie być w pełni kompatybilne z późniejszymi wersjami. Aktualizacja lub obniżenie wersji projektu może spowodować problemy z konfiguracją sprzętu, wykonywaniem bloków lub diagnostyką systemu.
- Integralność projektu: Nawet niewielkie zmiany w wersji oprogramowania mogą mieć wpływ na wykonywanie logiki, protokoły komunikacyjne lub obsługę danych, co może zakłócić działanie systemu w środowiskach produkcyjnych.

W świecie przemysłowym inżynierowie automatyki często pracują nad wieloma systemami lub maszynami, z których każdy może być opracowany przy użyciu różnych wersji TIA Portal. Aby sobie z tym poradzić, istnieją dwie główne strategie:

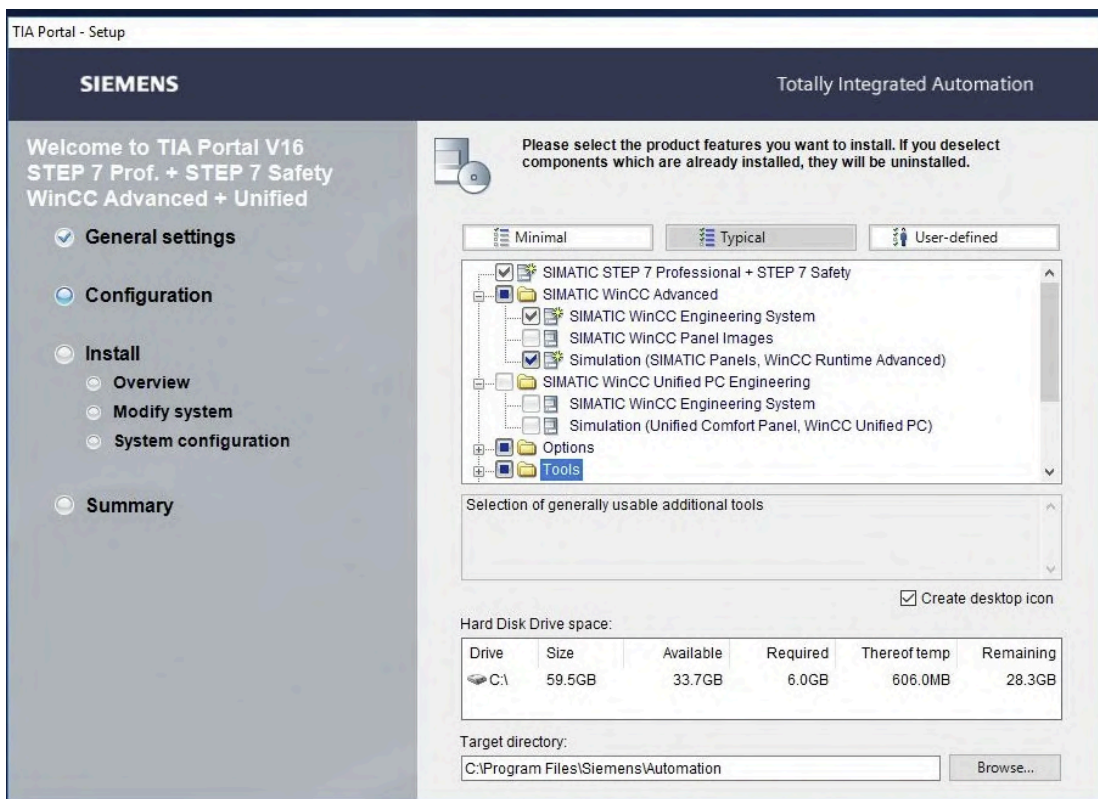
Instalacja wielu wersji portalu TIA

Jednym z praktycznych rozwiązań jest posiadanie wielu wersji TIA Portal zainstalowanych na stacji roboczej lub maszynie inżynierskiej. Siemens umożliwia instalowanie różnych wersji obok siebie, co ułatwia konserwację i rozwiązywanie problemów w różnych projektach bez ich aktualizacji. Takie podejście zapewnia, że każdy projekt pozostaje w swojej oryginalnej wersji, zachowując pełną kompatybilność i stabilność.

Korzyści z wielu wersji:

- Brak ryzyka problemów ze zgodnością: Możesz otwierać i edytować projekty w dokładnie tej wersji, w której zostały opracowane, zmniejszając ryzyko błędów lub problemów spowodowanych niedopasowaniem wersji.
- Znajomość istniejących systemów: Starsze maszyny lub systemy mogą polegać na określonych wersjach TIA Portal, a zachowanie tych wersji umożliwia łatwy dostęp i spójne działanie.

Jak zarządzać wieloma wersjami:



- Instalacja obok siebie: Siemens zezwala na współistnienie wielu wersji TIA Portal w jednym systemie. Na przykład można zainstalować TIA Portal V15, V16 i V19 bez konfliktów.
- Oznaczanie wersji: Starannie oznaczaj pliki i foldery projektu, aby wskazać używaną wersję TIA Portal, dzięki czemu zawsze będziesz wiedzieć, którą wersję otworzyć dla określonych projektów.

Aktualizacja projektów do najnowszej wersji

Innym podejściem jest aktualizacja całych projektów do najnowszej wersji TIA Portal, takiej jak V19. Chociaż może to uprościć środowisko oprogramowania poprzez zmniejszenie liczby zainstalowanych wersji TIA Portal, nie jest to pozbawione ryzyka i wyzwań.

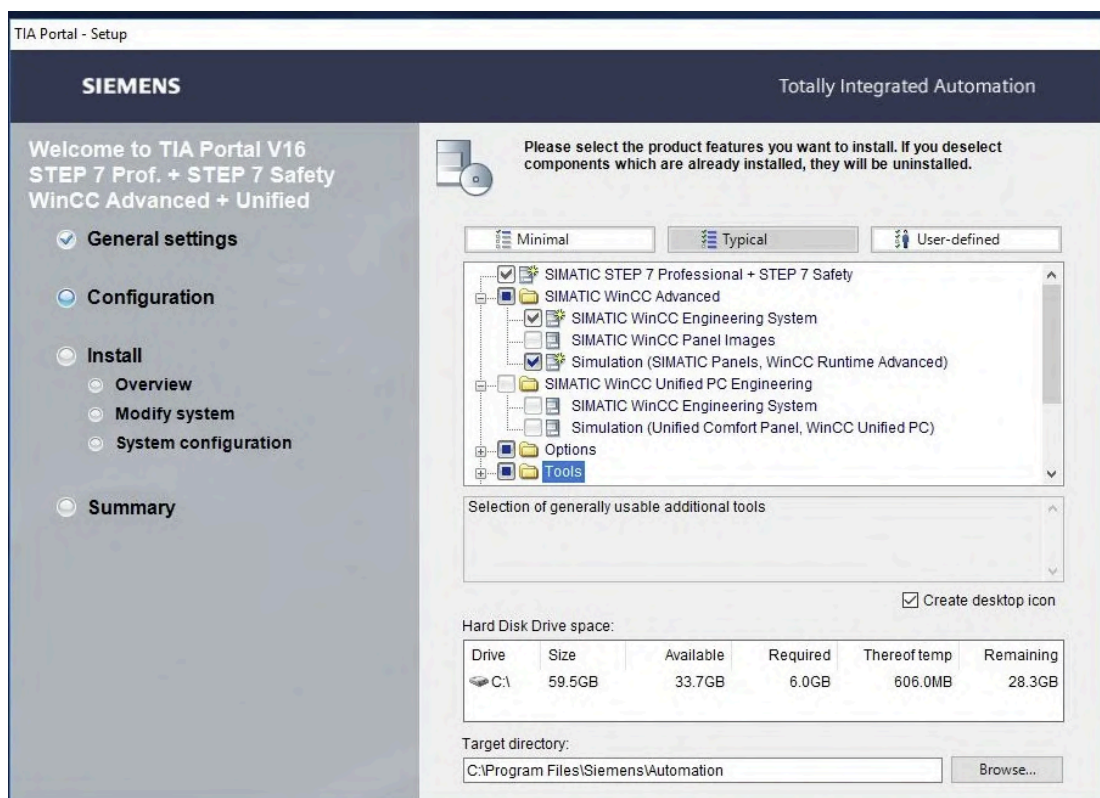
2.4 Bezpłatna licencja na TIA Portal

Siemens oferuje bezpłatną 21-dniową licencję próbną na TIA Portal, umożliwiającą użytkownikom zapoznanie się i przetestowanie oprogramowania bez konieczności dokonywania pełnego zakupu.

Wersja testowa zapewnia dostęp do niemal wszystkich funkcji pełnej wersji, umożliwiając inżynierom tworzenie, symulowanie i testowanie programów PLC, interfejsów HMI i systemów automatyki. Jest to doskonała okazja dla początkujących do zapoznania się z TIA Portal lub dla doświadczonych profesjonalistów do oceny najnowszej wersji przed podjęciem decyzji o zakupie. Wersję próbną można łatwo pobrać ze strony internetowej firmy Siemens, a w okresie próbnym nie jest wymagany sprzętowy klucz sprzętowy.

2.5 Którą wersję TIA Portal wybrać, aby rozpocząć naukę?

Jak widać, masz 21 dni okresu próbnego dla każdej wersji TIA Portal. Obecnie Windows 11 jest najnowszym systemem operacyjnym kompatybilnym z TIA Portal. Jeśli chcesz



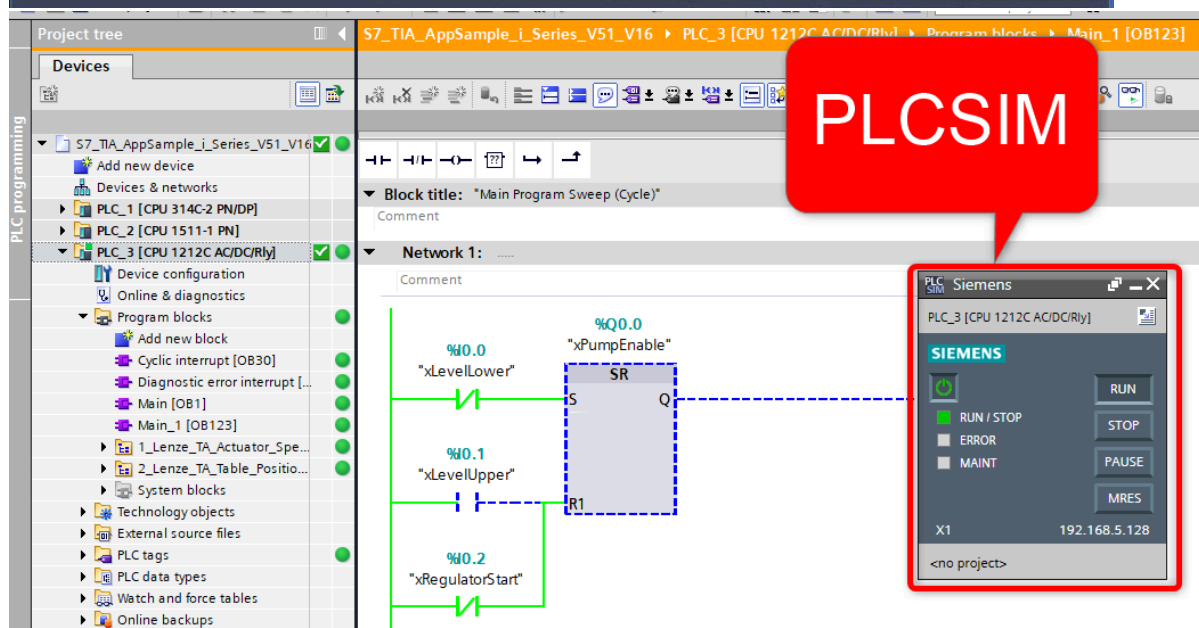
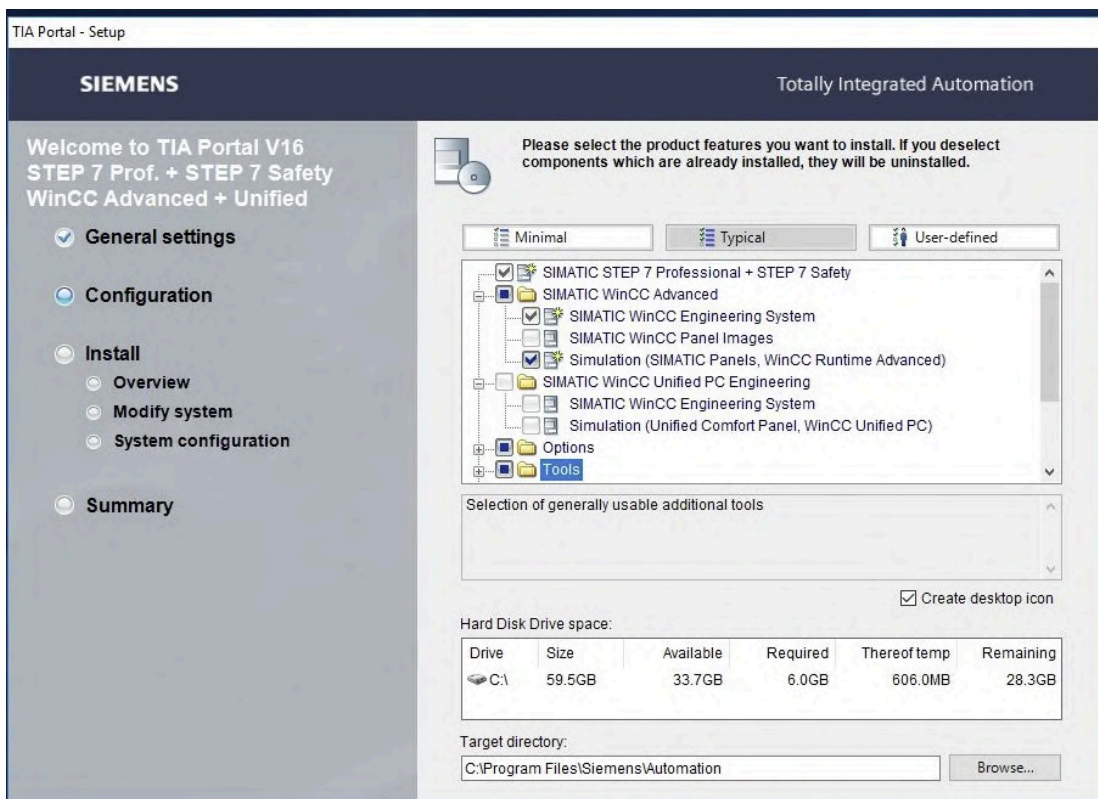
rozpocząć naukę, sugeruję zacząć od wersji V16. Po zakończeniu 21-dniowego okresu próbnego wystarczy pobrać i zainstalować TIA Portal V17. Po zakończeniu okresu próbnego można przejść do wersji TIA V18, a następnie V19.

W ten sposób można wydłużyć okres próbny do ponad 84 dni.

2.6 Darmowy symulator sterownika PLC - PLCSIM firmy Siemens

PLCSIM to darmowy symulator PLC oferowany przez firmę Siemens, który pozwala użytkownikom symulować i testować swoje programy PLC bez konieczności posiadania fizycznego sprzętu.

Jest on zintegrowany ze środowiskiem oprogramowania TIA Portal, ułatwiając symulację systemów automatyki bezpośrednio z interfejsu programowania. Dzięki PLCSIM inżynierowie mogą testować język LADDER, bloki funkcyjne i inne struktury programowania w środowisku wirtualnym. Narzędzie to jest szczególnie przydatne do rozwiązywania problemów i sprawdzania poprawności logiki przed wdrożeniem jej do prawdziwego sterownika PLC.

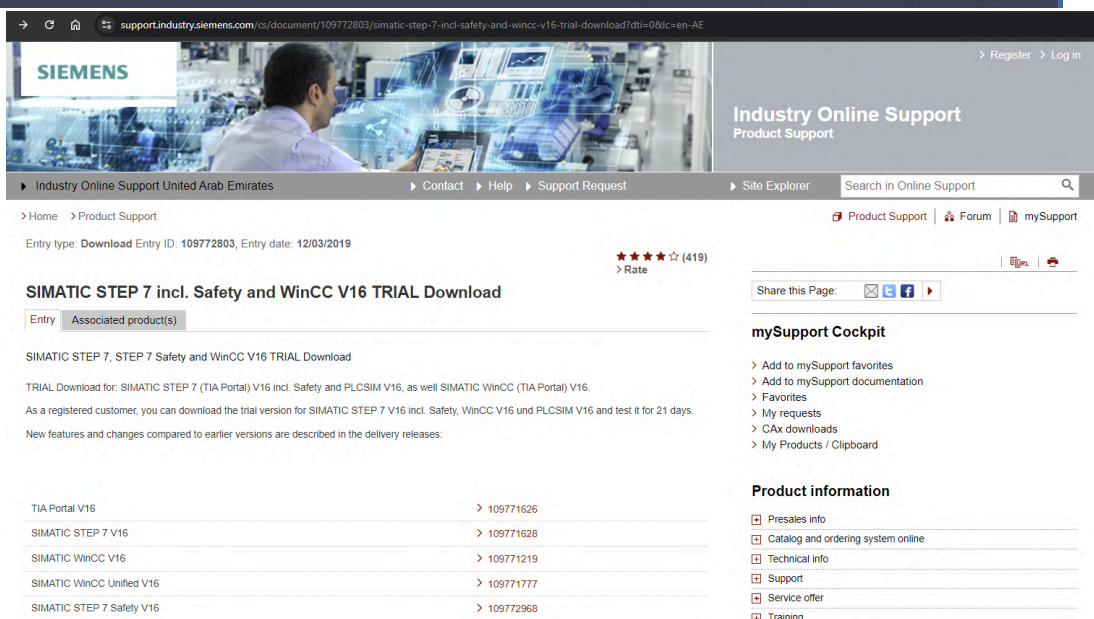
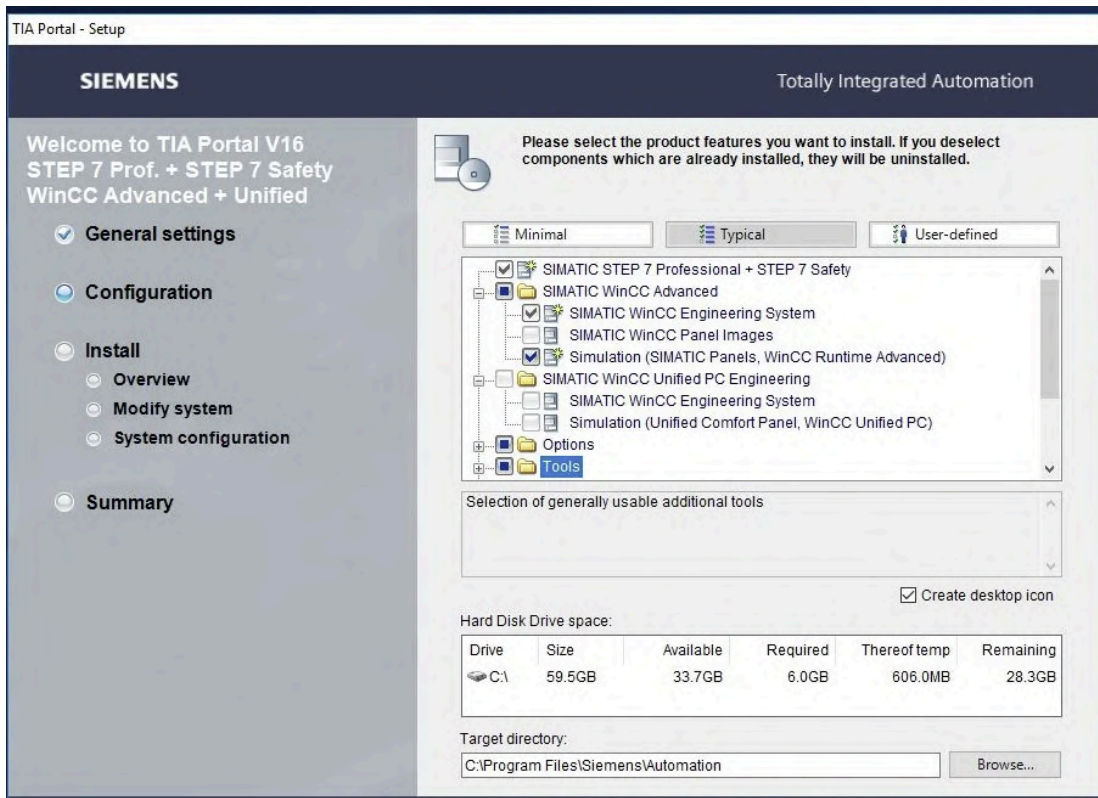


2.6 Instalacja portalu TIA

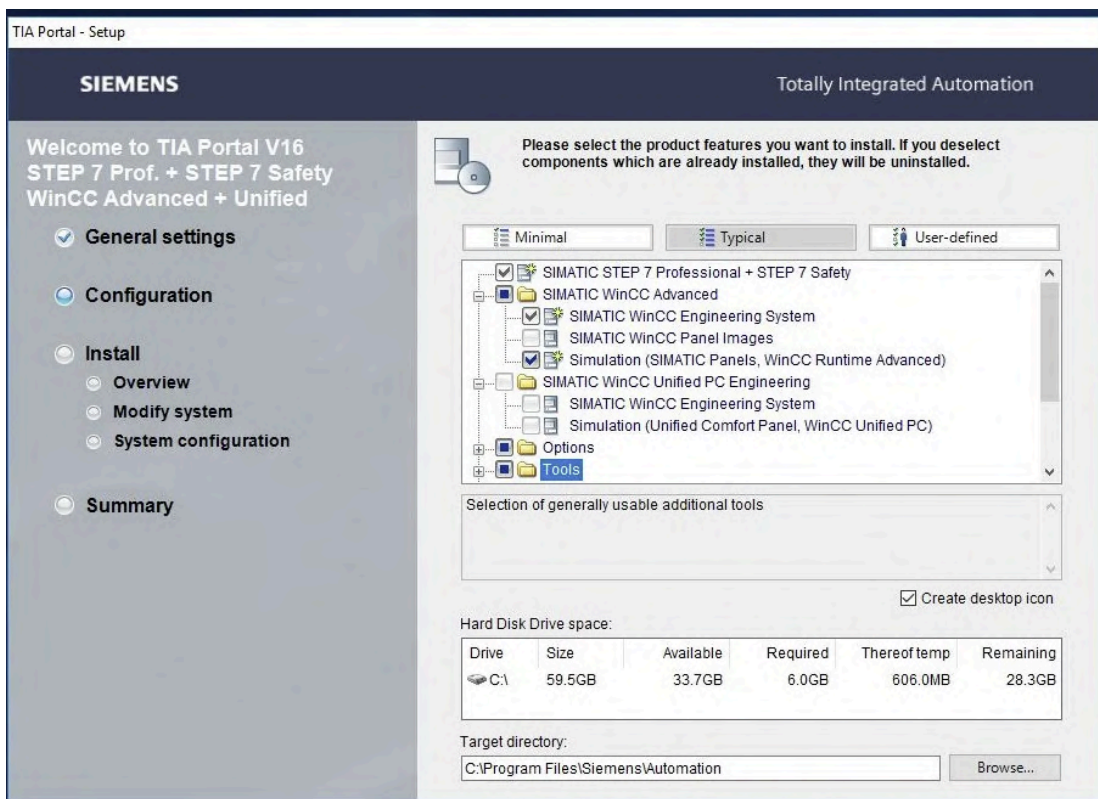
KROK 1

Wejdź na:

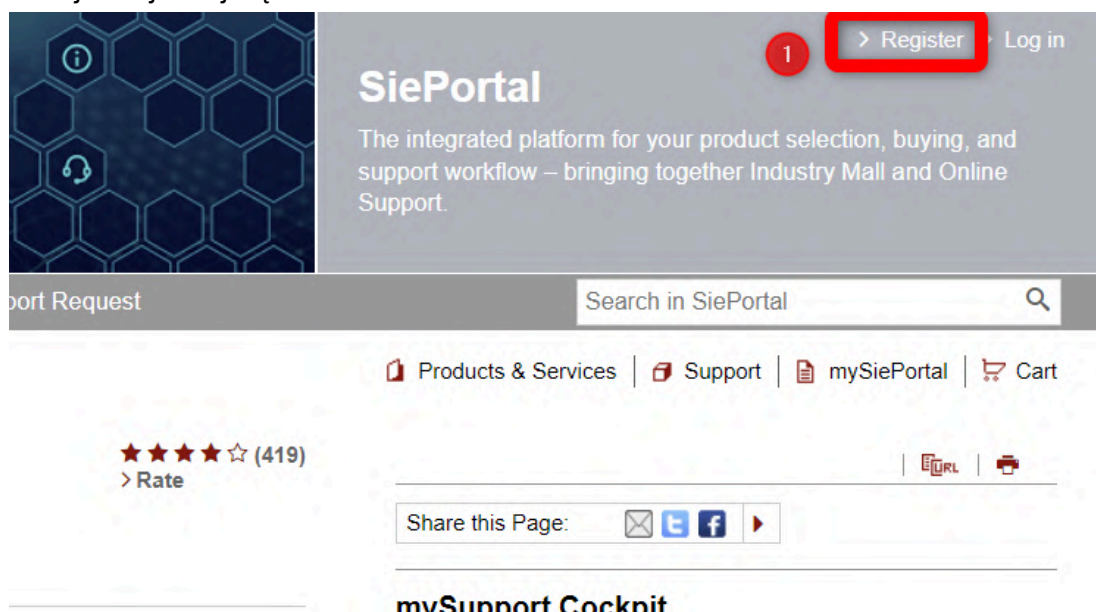
<https://support.industry.siemens.com/cs/document/109772803/simatic-step-7-incl-safety-and-wincc-v16-trial-download?dti=0&lc=en>



KROK 2

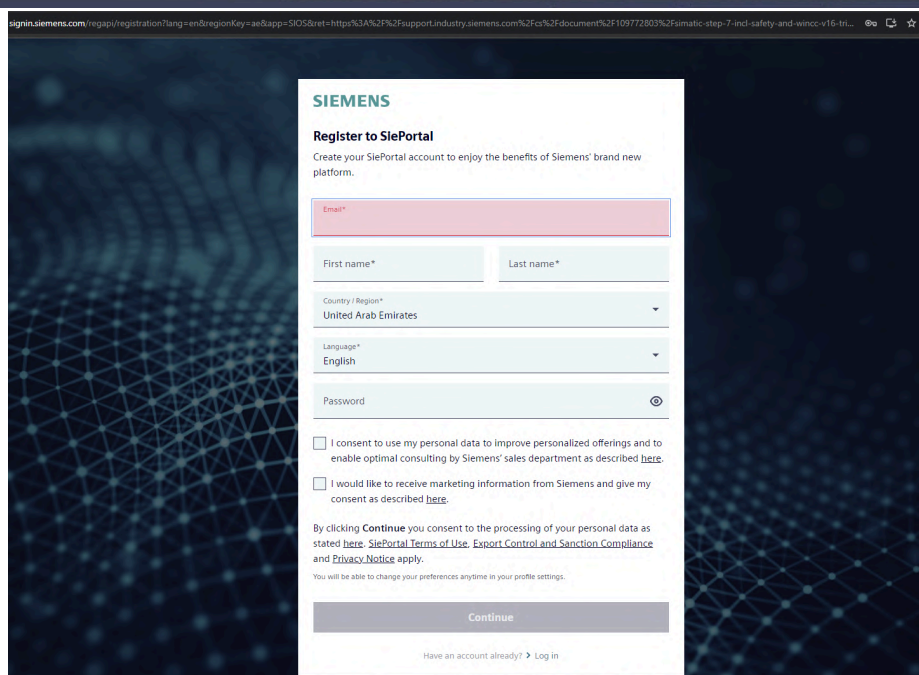
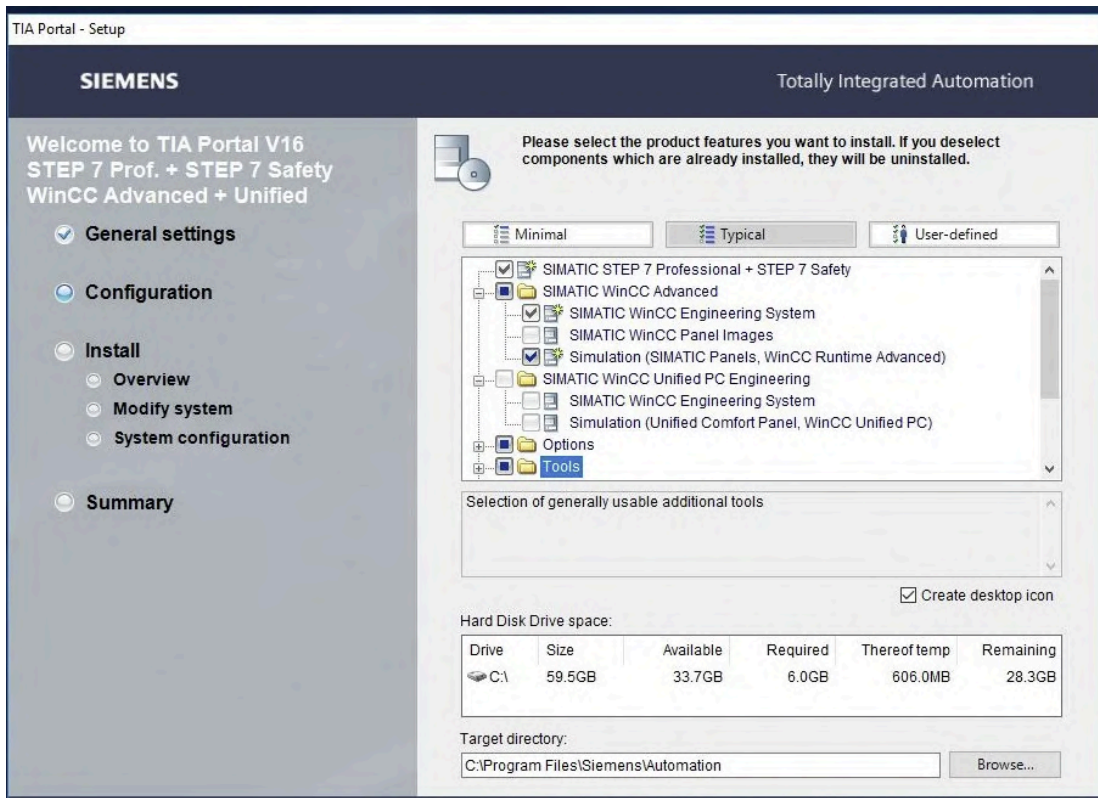


Kliknij Zarejestruj się.

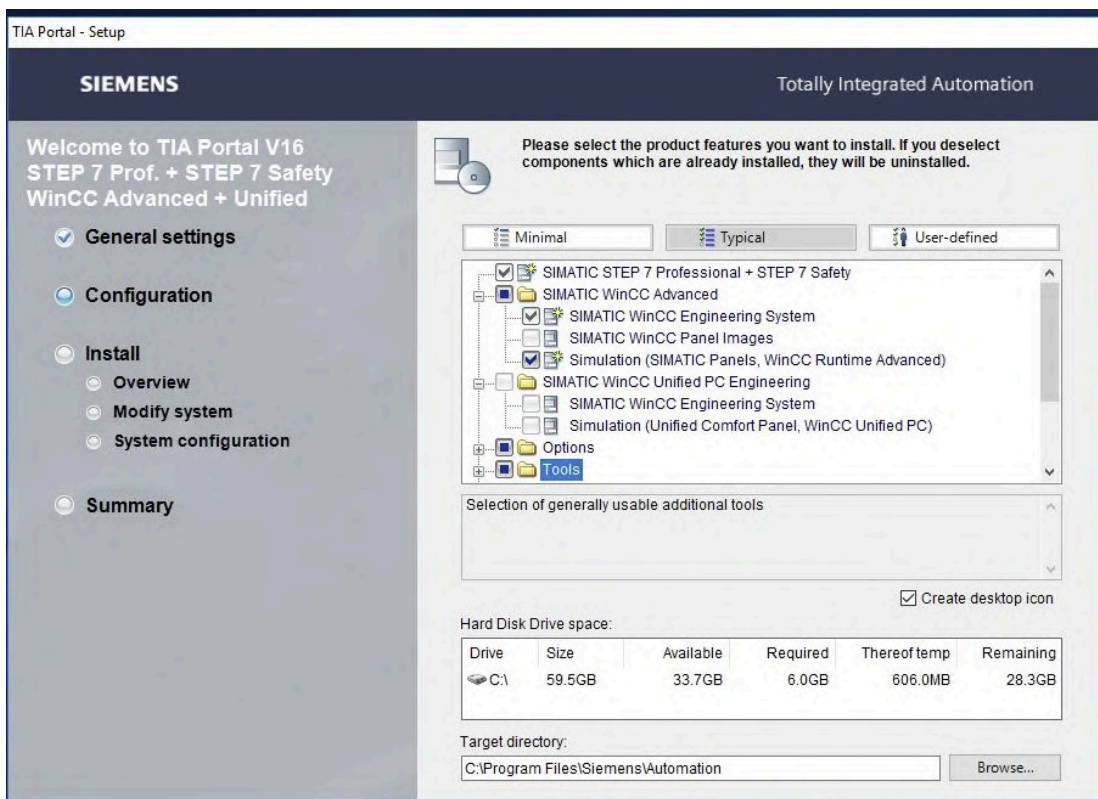


KROK 3

Wypełnij formularz i utwórz konto.



KROK 4



Wróć do:

<https://support.industry.siemens.com/cs/document/109772803/simatic-step-7-incl-safety-and-wincc-v16-trial-download?dti=0&lc=en>

I pobierz pliki zaznaczone na czerwono (1) i (2).

TRIAL Download STEP 7 Basic/Professional, STEP 7 Safety Basic/Advanced and WinCC Basic/Comfort/Advanced and WinCC Unified:

The download is split into multiple files. Please first download all of the parts to the same folder and then execute the file with the extension .exe. Afterwards, the parts are merged and you can execute the setup. The trial version can, however, also be directly ordered on DVD:
6AV2102-0AA06-0AA7

Notes about downloading

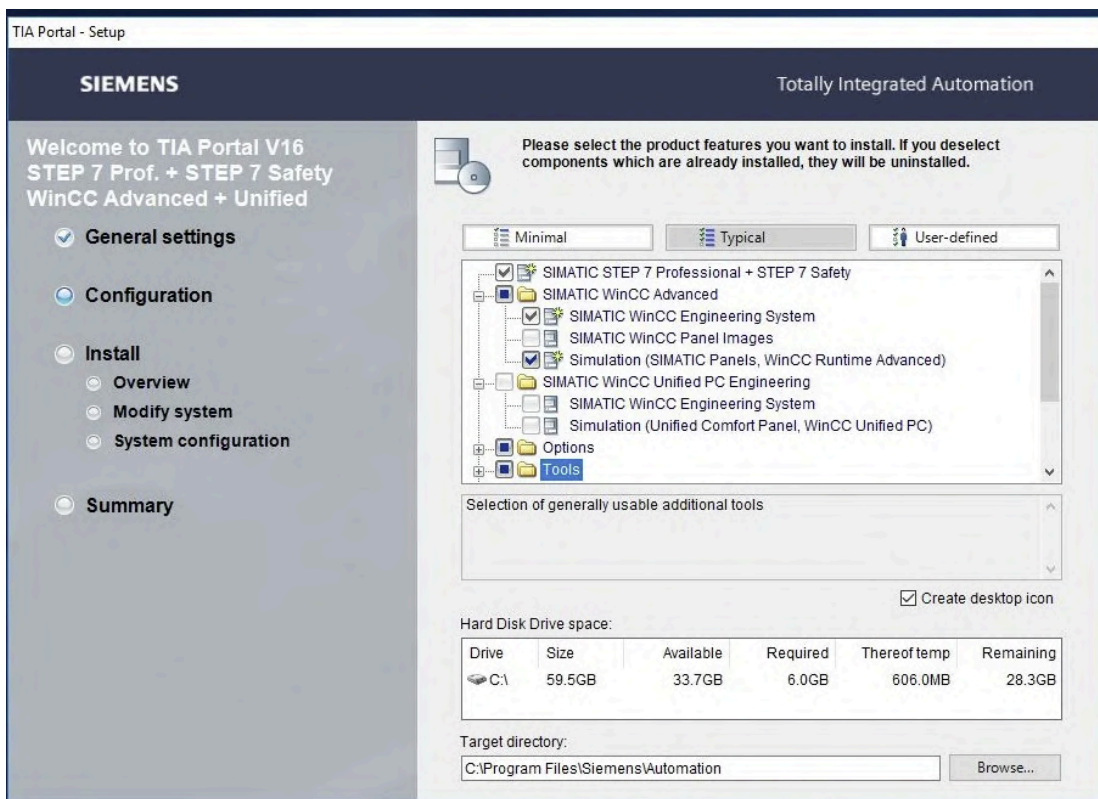
DVD 1 Setup	Files
STEP 7 Basic / Professional STEP 7 Safety Basic / Advanced WinCC Basic / Comfort / Advanced / Unified Engineering	☒ DVD_1.001 (2,0 GB) ☒ DVD_1.002 (2,0 GB) ☒ DVD_1.003 (1,3 GB) ☒ DVD_1.exe (2,8 MB)
DVD 2 Hardware Support Packages, Open Source Software, Tools	☒ DVD_2.001 (2,0 GB) ☒ DVD_2.002 (2,0 GB) ☒ DVD_2.003 (2,0 GB) ☒ DVD_2.004 (1,7 GB) ☒ DVD_2.exe (2,8 MB)
DVD 3 WinCC Panel Images	☒ DVD_3.001 (2,0 GB) ☒ DVD_3.002 (2,0 GB) ☒ DVD_3.003 (2,0 GB) ☒ DVD_3.004 (1,4 GB) ☒ DVD_3.exe (2,8 MB)
SHA-256 checksum	☒ Checksum DVD1&DVD2.txt (1,9 KB) ☒ Checksum DVD3.txt (1 KB)

> Information on SHA-256

TRIAL Download STEP 7 PLCSIM :

DVD 1 Setup STEP 7 PLCSIM	Files
	☒ SIMATIC_S7PLCSIM_V16.exe (1,7 GB)
SHA-256 checksum	☒ SIMATIC_S7PLCSIM_V16.txt (1 KB)

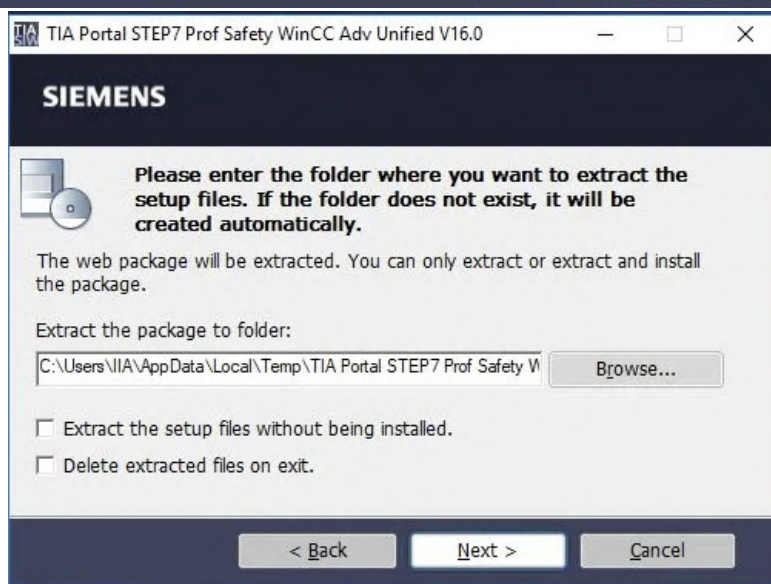
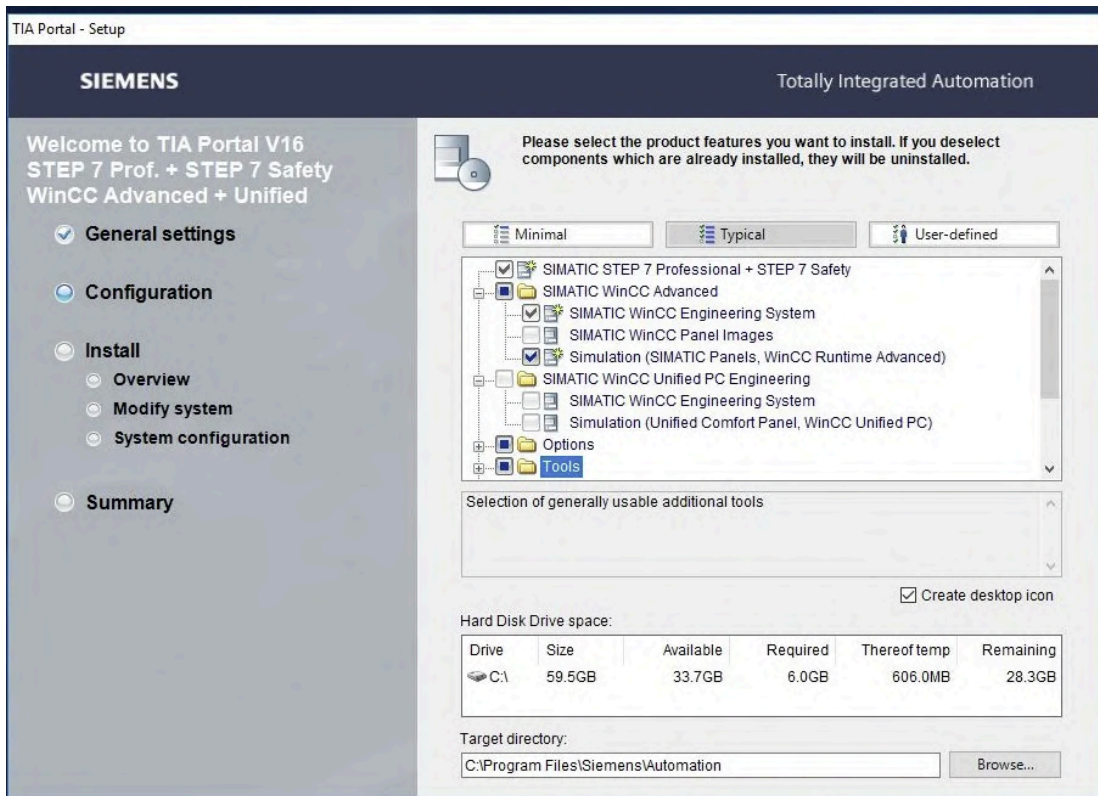
> Information on SHA-256

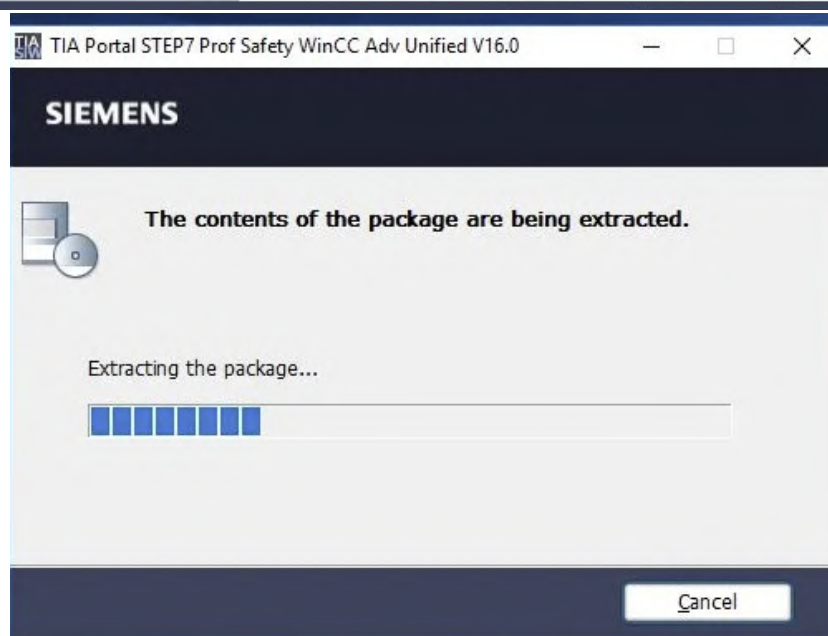
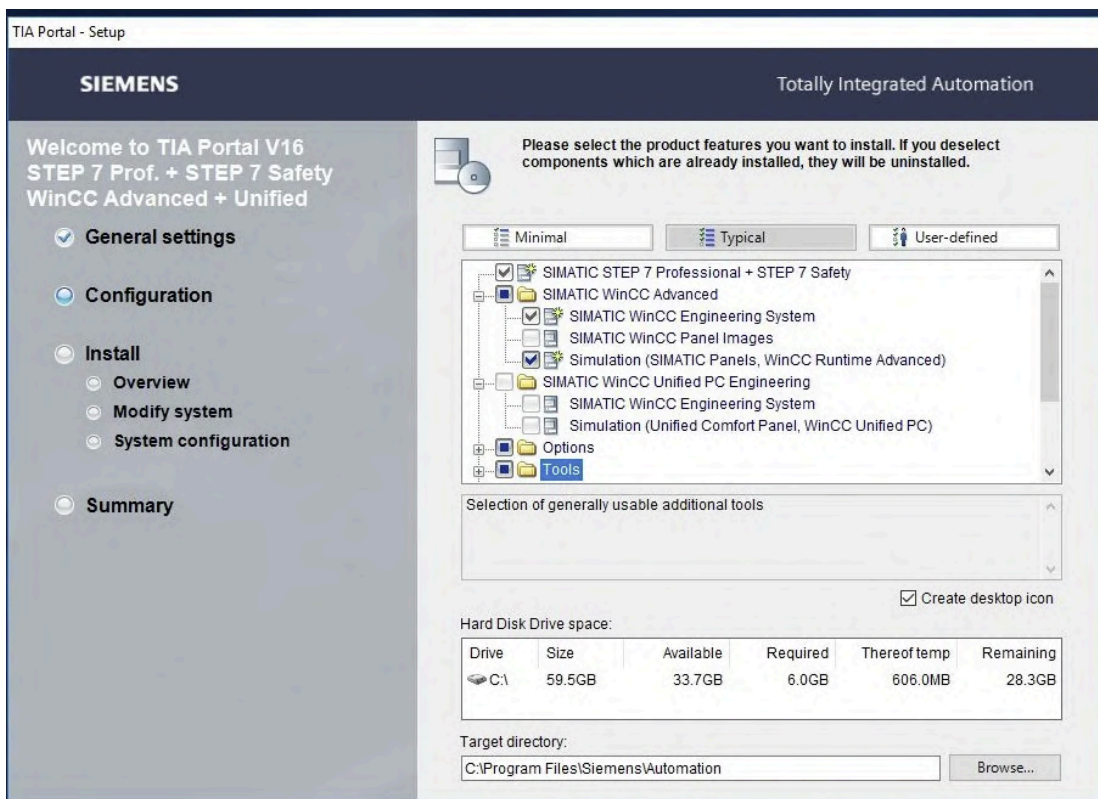


KROK 5

Po pobraniu plików, przenieś je wszystkie do jednego katalogu, a następnie kliknij dwukrotnie na „TIA_Portal_STEP7_Prof_ Safety_WINCC_Adv_Unified_V16.exe”, aby rozpocząć instalację.

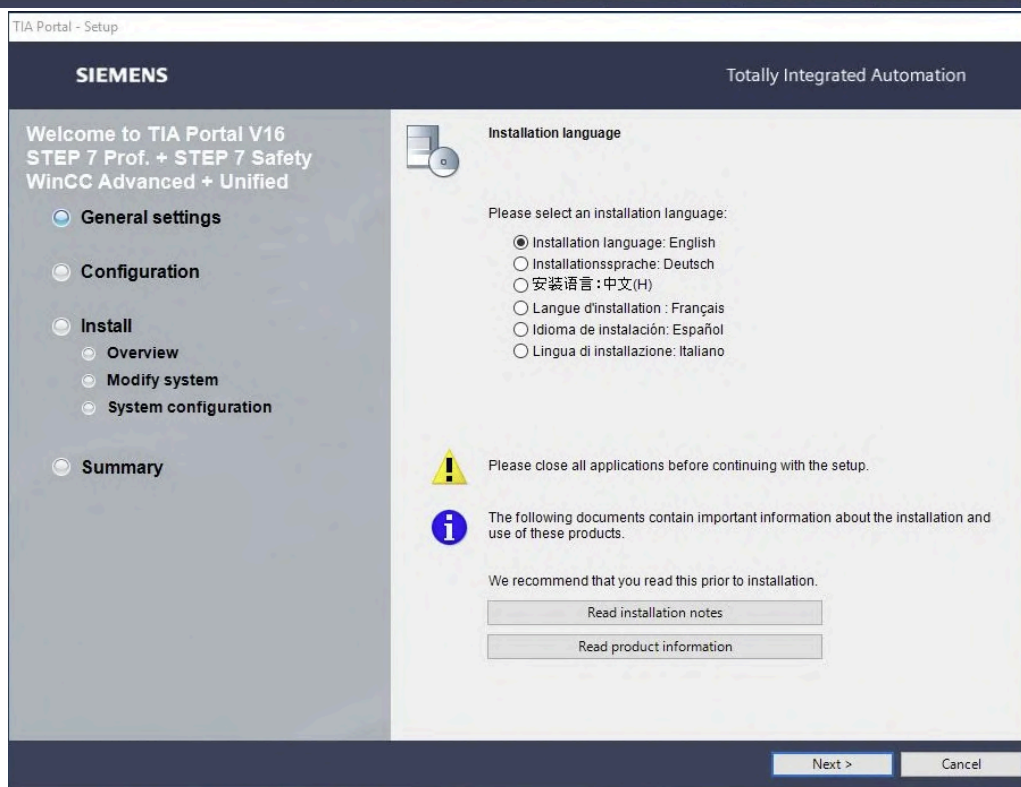
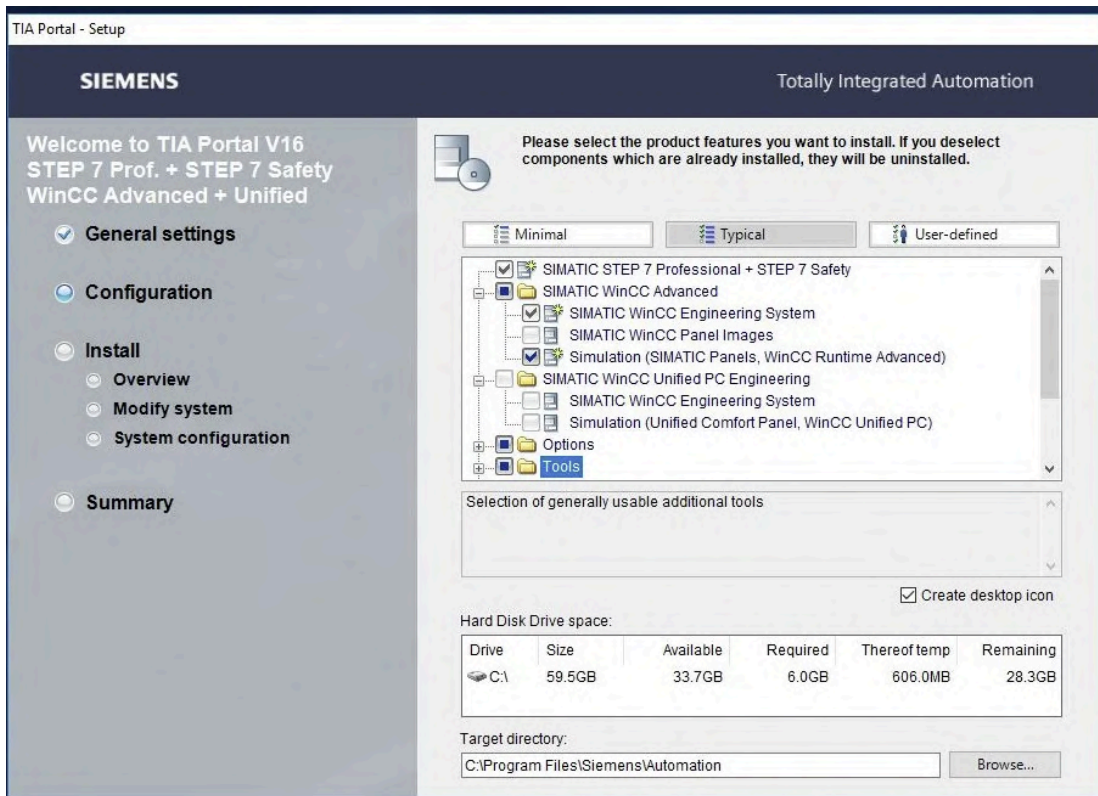


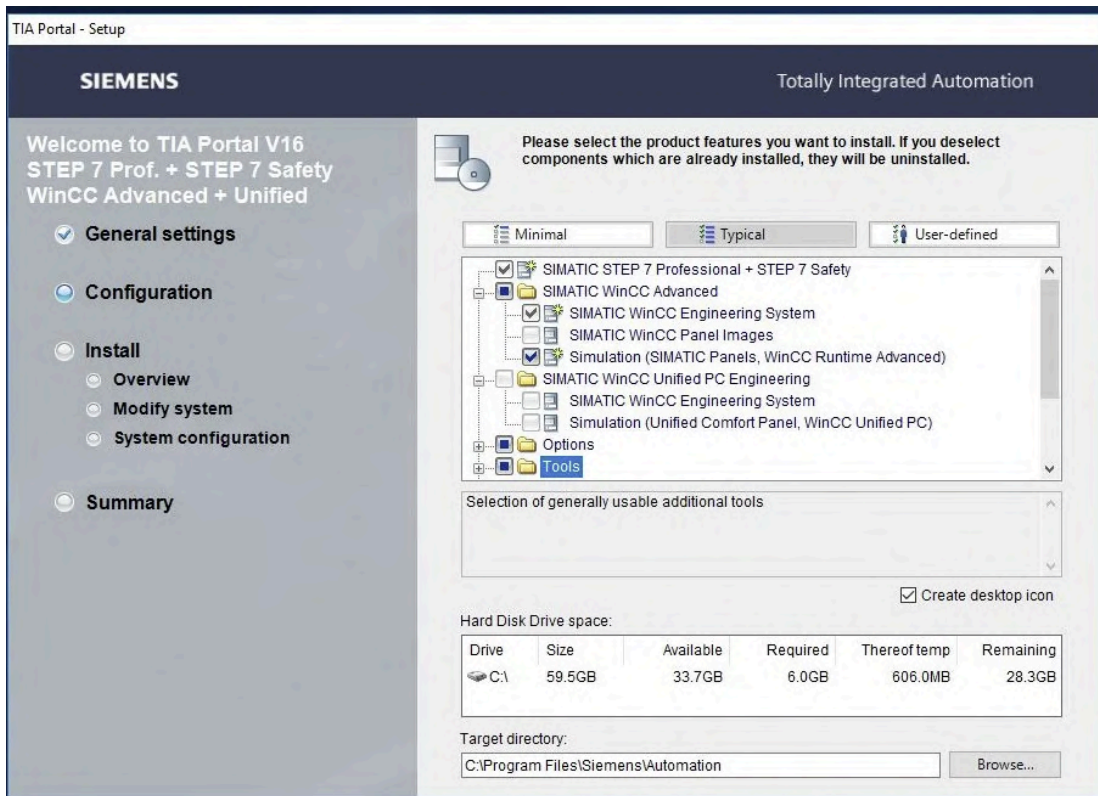


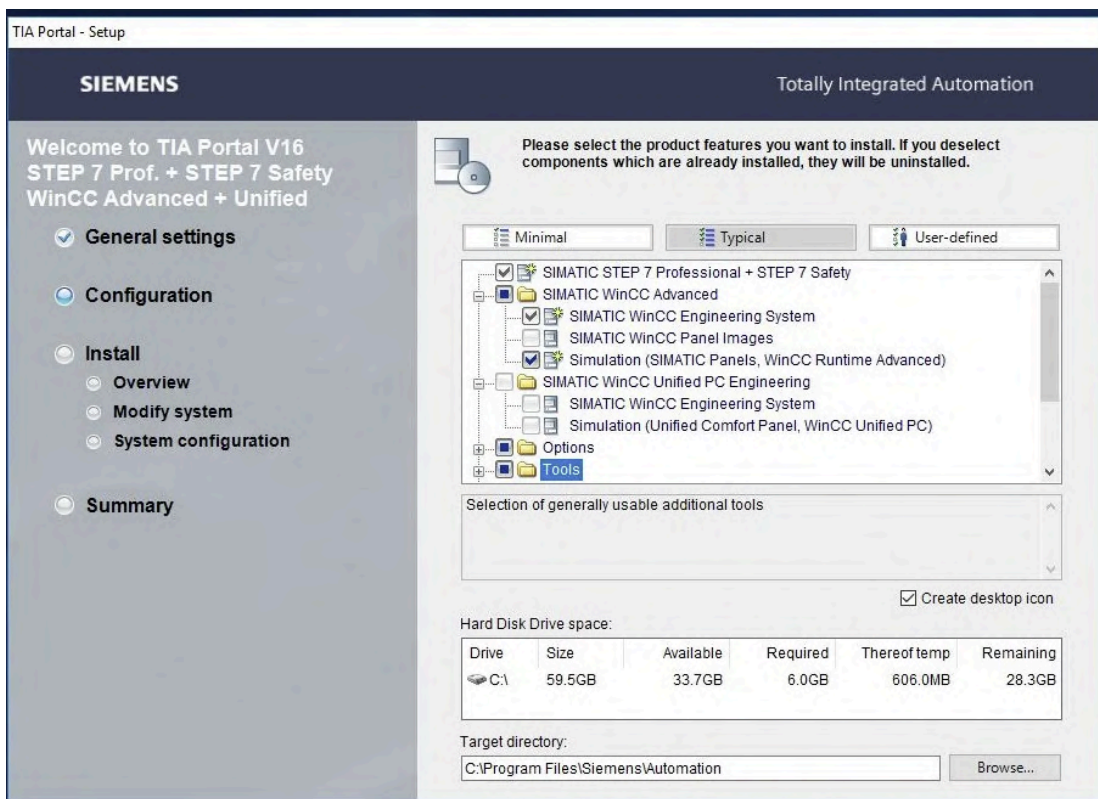


KROK 5

Po rozpakowaniu plików instalacja rozpocznie się automatycznie. Kontynuuj instalację w następujący sposób.







KROK 6 - Zainstaluj PLCSIM

Po zakończeniu instalacji STEP7, kliknij na „SIMATIC_S7PLCSIM_V16.exe” i kontynuuj instalację PLCSIM w ten sam sposób.

KROK 7 - Napisz swój pierwszy program PLC

Świetnie! Masz zainstalowany TIA Portal i jesteś gotowy do rozpoczęcia kodowania!

Jeśli chcesz napisać swój pierwszy program na sterowniku Siemens PLC, sprawdź nasze bezpłatne szkolenie wstępne PLC tutaj:

<https://www.youtube.com/live/MZ63ANIPBs?si=wT1H6B3WAYIN-pO8>



Welcome to TIA Portal V16
STEP 7 Prof. + STEP 7 Safety
WinCC Advanced + Unified

☒ General settings

☐ Configuration

☐ Install

☐ Overview

☐ Modify system

☐ System configuration

☐ Summary



Please select the product features you want to install. If you deselect components which are already installed, they will be uninstalled.

Minimal Typical User-defined

- ☒ SIMATIC STEP 7 Professional + STEP 7 Safety
 - ☒ SIMATIC WinCC Advanced
 - ☒ SIMATIC WinCC Engineering System
 - ☒ SIMATIC WinCC Panel Images
 - ☒ Simulation (SIMATIC Panels, WinCC Runtime Advanced)
 - ☒ SIMATIC WinCC Unified PC Engineering
 - ☒ SIMATIC WinCC Engineering System
 - ☒ Simulation (Unified Comfort Panel, WinCC Unified PC)
- ☐ Options
- ☐ Tools

Selection of generally usable additional tools

☒ Create desktop icon

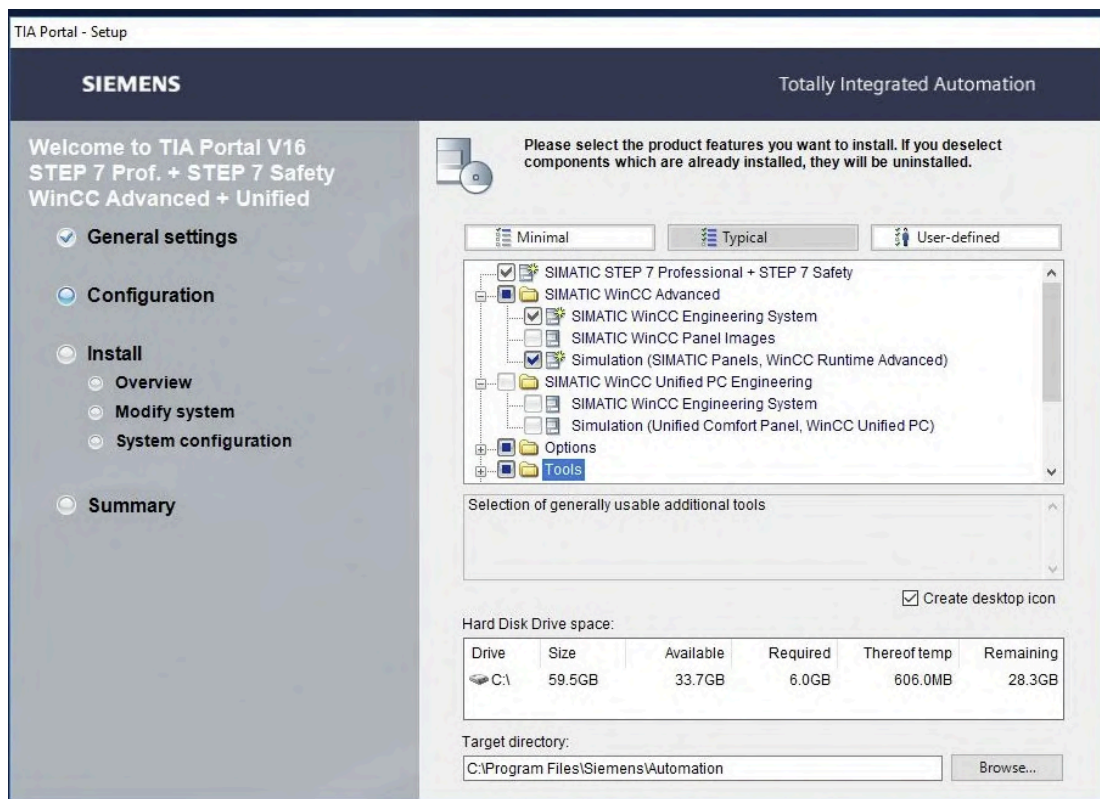
Hard Disk Drive space:

Drive	Size	Available	Required	Thereof temp	Remaining
C:\	59.5GB	33.7GB	6.0GB	606.0MB	28.3GB

Target directory:

C:\Program Files\Siemens\Automation

Browse...

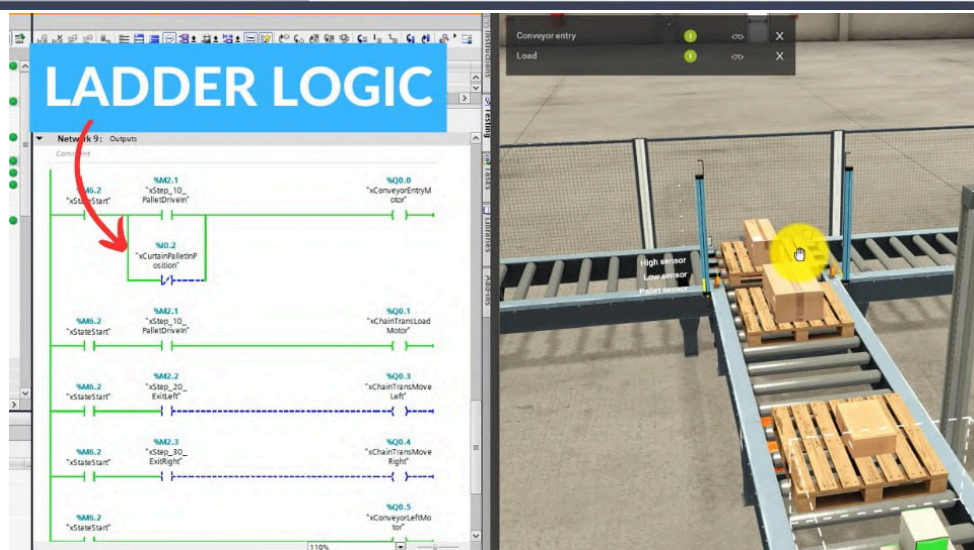
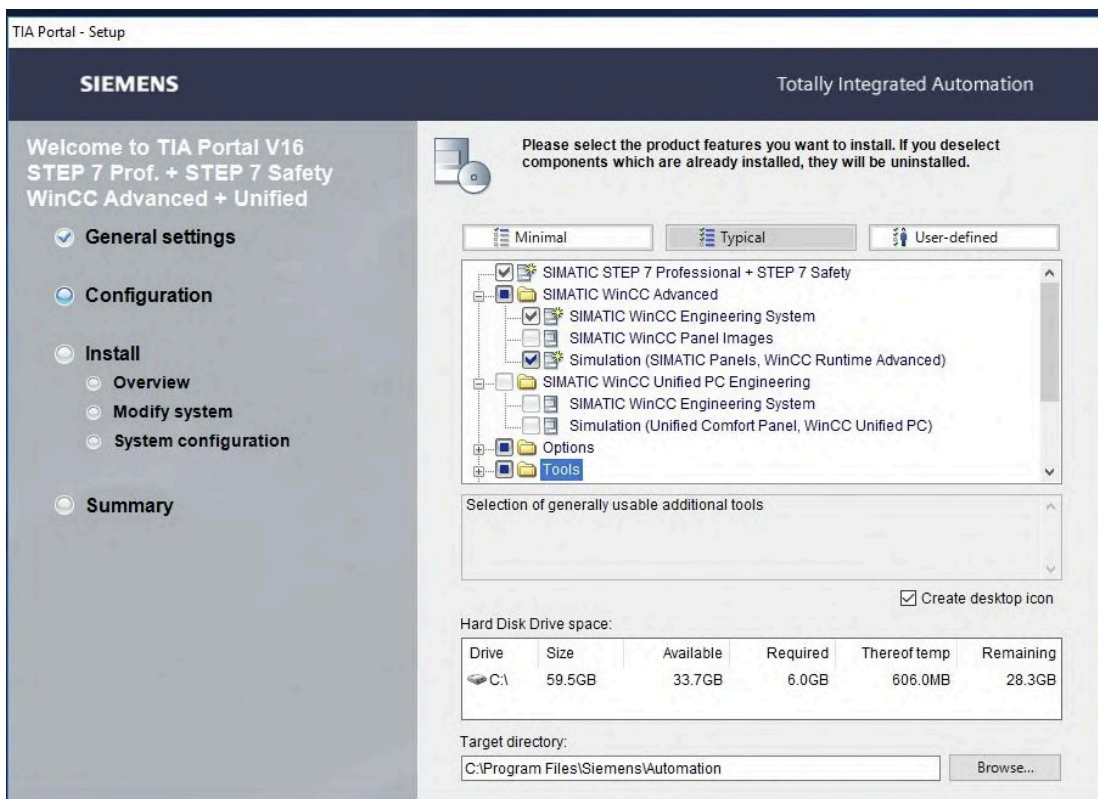


ROZDZIAŁ 3 | JĘZYKI PROGRAMOWANIA DLA PLC

Po wybraniu platformy kolejnym krokiem jest wybór języka programowania. Sterowniki PLC można programować w kilku językach, w tym:

- **Ladder Logic - LAD**
- **Function Block Diagrams - FBD**
- **Structured Text - ST (SCL for Siemens)**
- **Sequential Flow Charts**
- **Instruction List - IL (STL for Siemens)**

Chociaż wybór języka programowania może być kuszący ze względu na jego prostotę lub to, że wydaje się znajomy, zaleca się rozpoczęcie od logiki drabinkowej. Wynika to z faktu, że logika drabinkowa jest najbardziej powszechną i szeroko stosowaną metodą programowania, wywodzącą się z logiki przekaźnikowej. Jest ona znana z łatwości implementacji i debugowania.



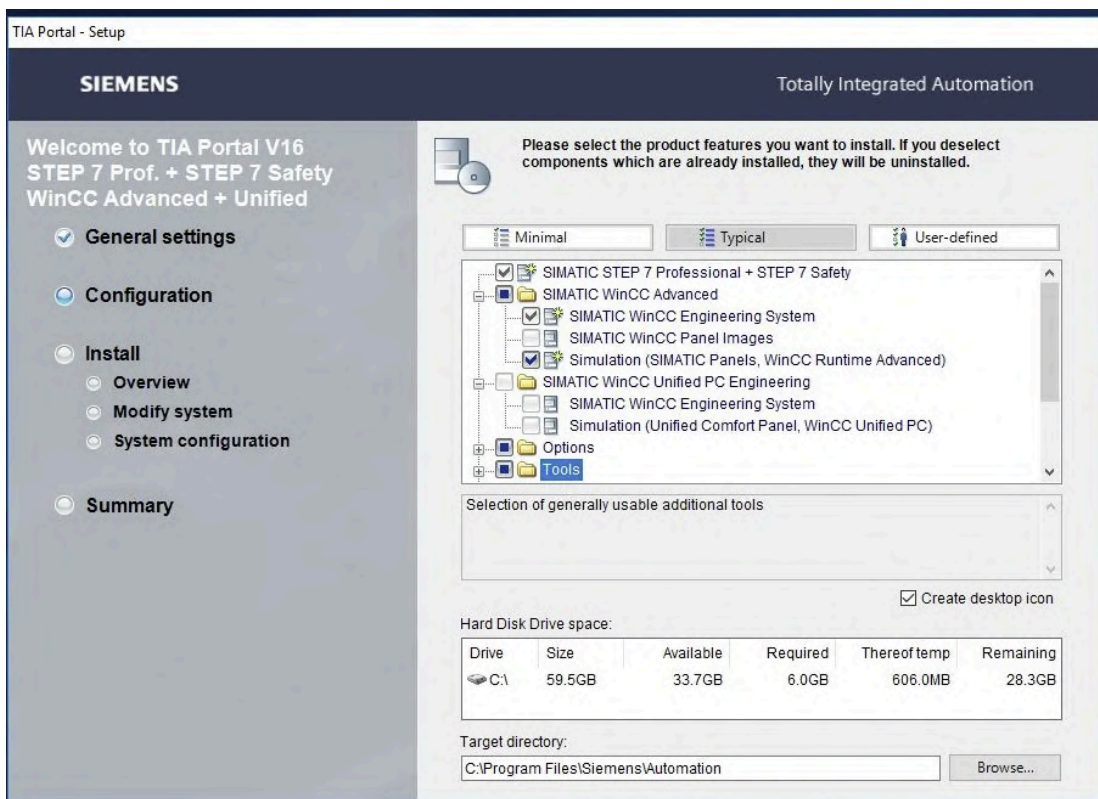
Rdzeń logiki drabinkowej składa się z trzech głównych instrukcji:

- -| | - Styk normalnie otwarty (NO)
- -|/| - Styk normalnie zamknięty (NC)
- -() Cewki

Na tym etapie ważne jest, aby skupić się na opanowaniu struktur szczelbli i rozgałęzień.

Zacznij od zbudowania prostych procedur logicznych przy użyciu powyższych instrukcji. Zwróć uwagę na to, jak wpływają one na stany logiczne, z którymi są połączone i wypróbuj różne konfiguracje szczelbli. Tworząc te różne struktury, zastanów się nad ich interakcją i wpływem na ogólną logikę.

Każdy z tych tematów omówimy bardziej szczegółowo w kolejnych rozdziałach.



Przed zagłębieniem się w język LADDER, kluczowe jest zrozumienie różnych języków programowania sterowników PLC i ich różnic.

5 najczęściej używanych języków programowania sterowników PLC w 2020 roku to:

- Structured Text (ST)
- Sequential Function Charts (SFC)
- Ladder Logic Diagram (LD)
- Function Block Diagram (FBD)
- Instruction List (IL)

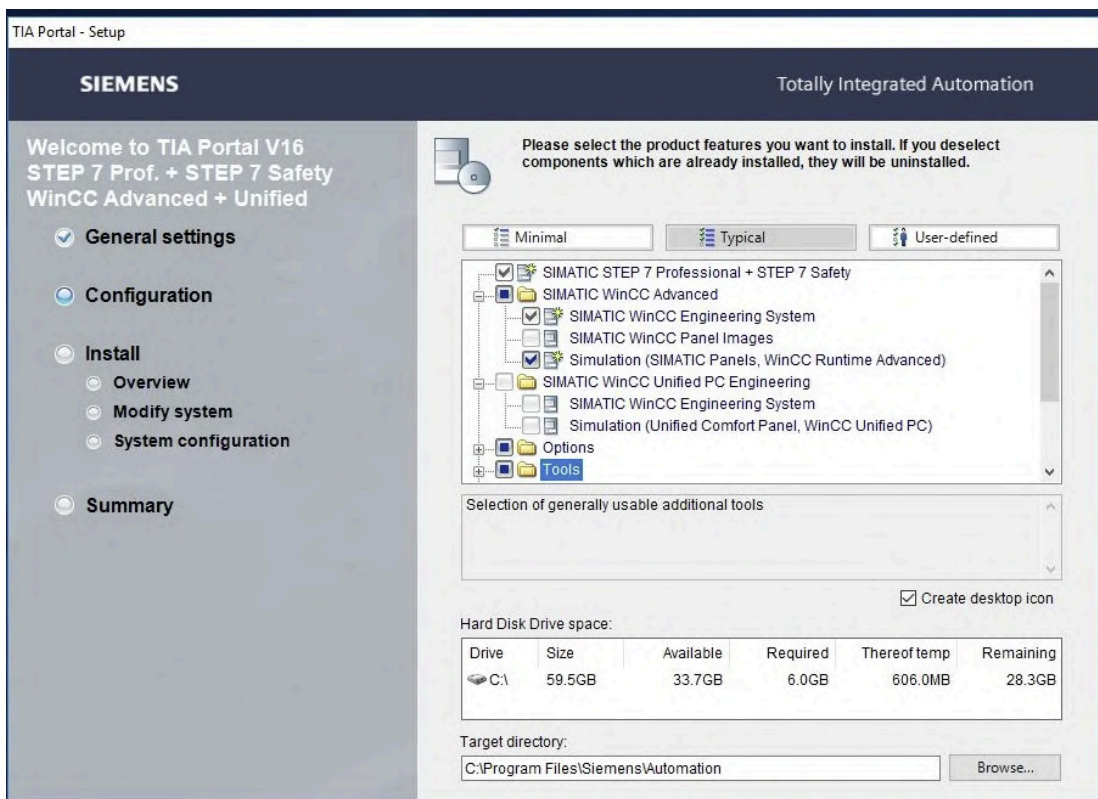
Norma Międzynarodowej Komisji Elektrotechnicznej (IEC) 61131-3 definiuje 5 różnych języków programowania sterowników PLC: logika drabinkowa, tekst strukturalny, schematy bloków funkcyjnych, sekwencyjne schematy funkcyjne i listy instrukcji. Każdy język ma swoje mocne strony, ograniczenia i najbardziej odpowiednie przypadki użycia. Aby być skutecznym programistą PLC, ważne jest, aby znać wszystkie te opcje rozwiązywania problemów, wybierania odpowiedniego narzędzia do pracy i uzyskiwania różnych perspektyw rozwiązywania problemów.

Ponadto dostępność niektórych języków może zależeć od używanej platformy PLC, przy czym niektóre wymagają bardziej zaawansowanej licencji lub w ogóle nie oferują wsparcia. Na przykład tylko pełna wersja TIA Portal i S7-1500 PLC oferuje dostęp do IL i SFC.

W następnych sekcjach przyjrzymy się bliżej każdemu z tych języków, przeglądając ich strukturę, zastosowania i typowe przypadki użycia.

3.1 Jaki jest najpopularniejszy język programowania sterowników PLC?

To pytanie jest często dyskutowane wśród programistów PLC na całym świecie. Panuje ogólna zgoda co do tego, że logika drabinkowa jest najczęściej używanym językiem



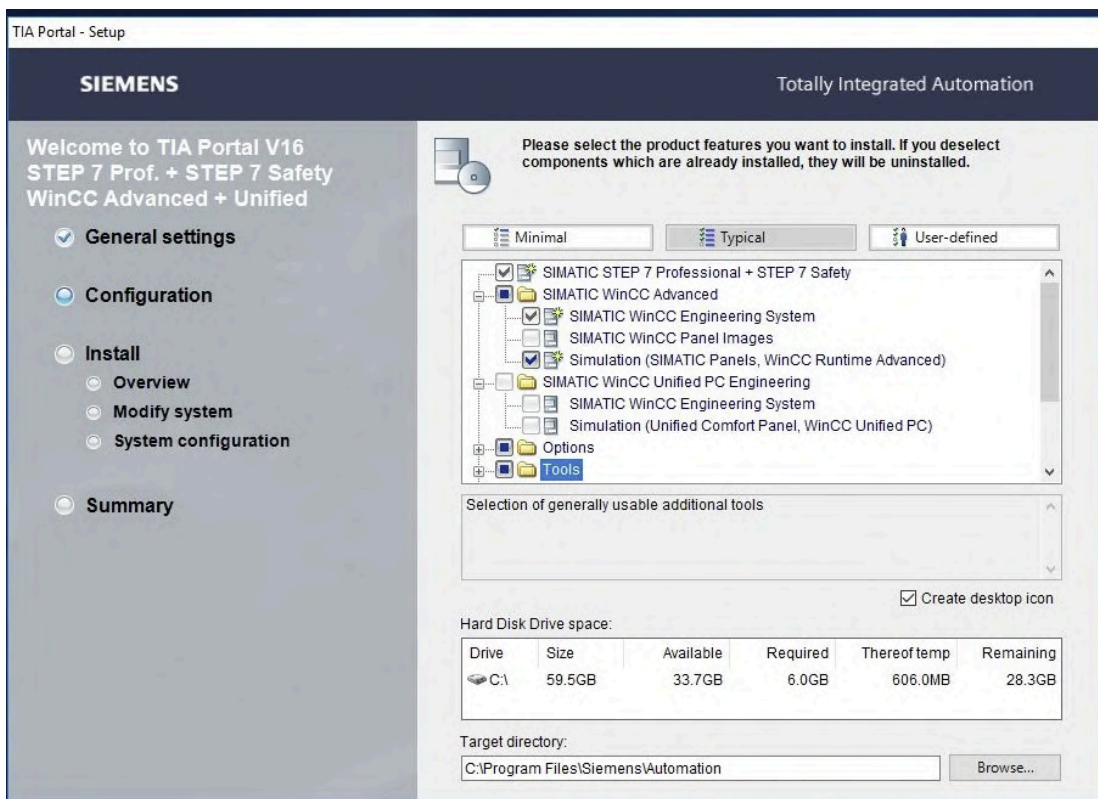
programowania sterowników PLC. Wynika to z faktu, że logika drabinkowa jest bardzo wszechstronna, łatwa do nauczenia i dobrze rozumiana, szczególnie przez elektryków, którzy są zaznajomieni ze schematami odzwierciedlającymi tę architekturę.

Jednak w ostatniej dekadzie młodsze pokolenie wkroczyło do branży produkcyjnej. Ci nowsi inżynierowie i technicy są zazwyczaj szkoleni w nowoczesnych językach, takich jak Java, Python i JavaScript. Ponieważ języki te mają podobieństwa z tekstem strukturalnym (ST), nastąpił wzrost wykorzystania ST w programowaniu PLC.

Ponadto branża, w której używany jest sterownik PLC, również odgrywa znaczącą rolę w określaniu wybranego języka. Na przykład procesy chemiczne są często projektowane przy użyciu schematów rurociągów i oprzyrządowania (P&ID). Schematy te można łatwo przetłumaczyć na programy PLC za pomocą schematów bloków funkcyjnych (FBD).

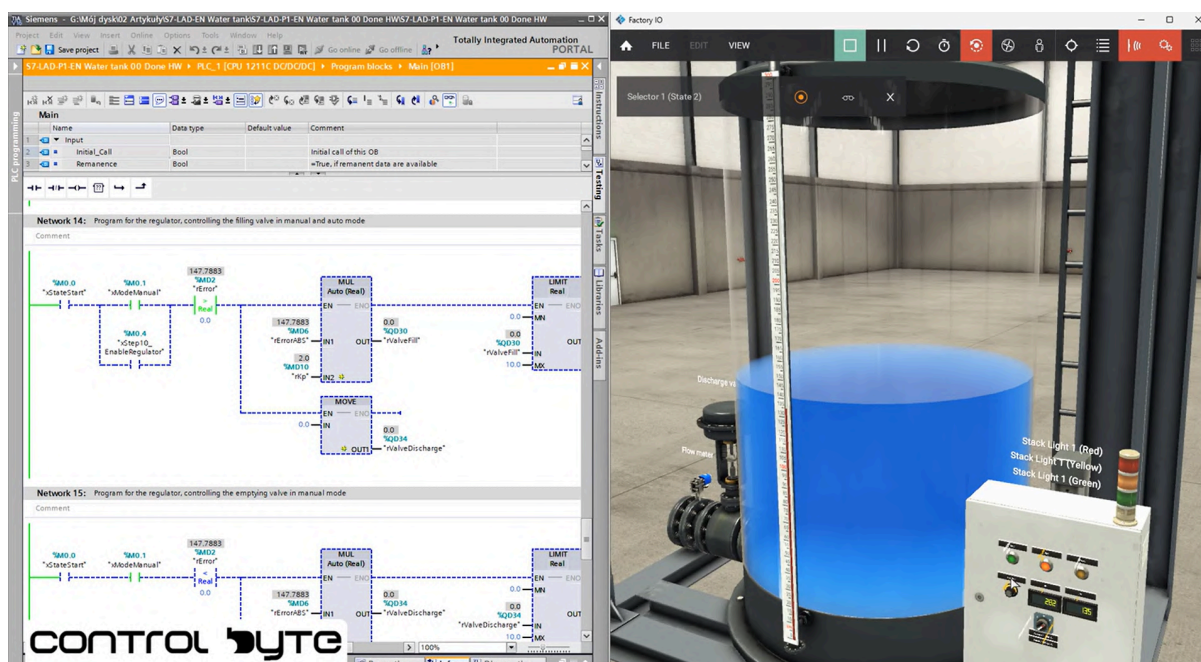
3.2 Ladder Logic

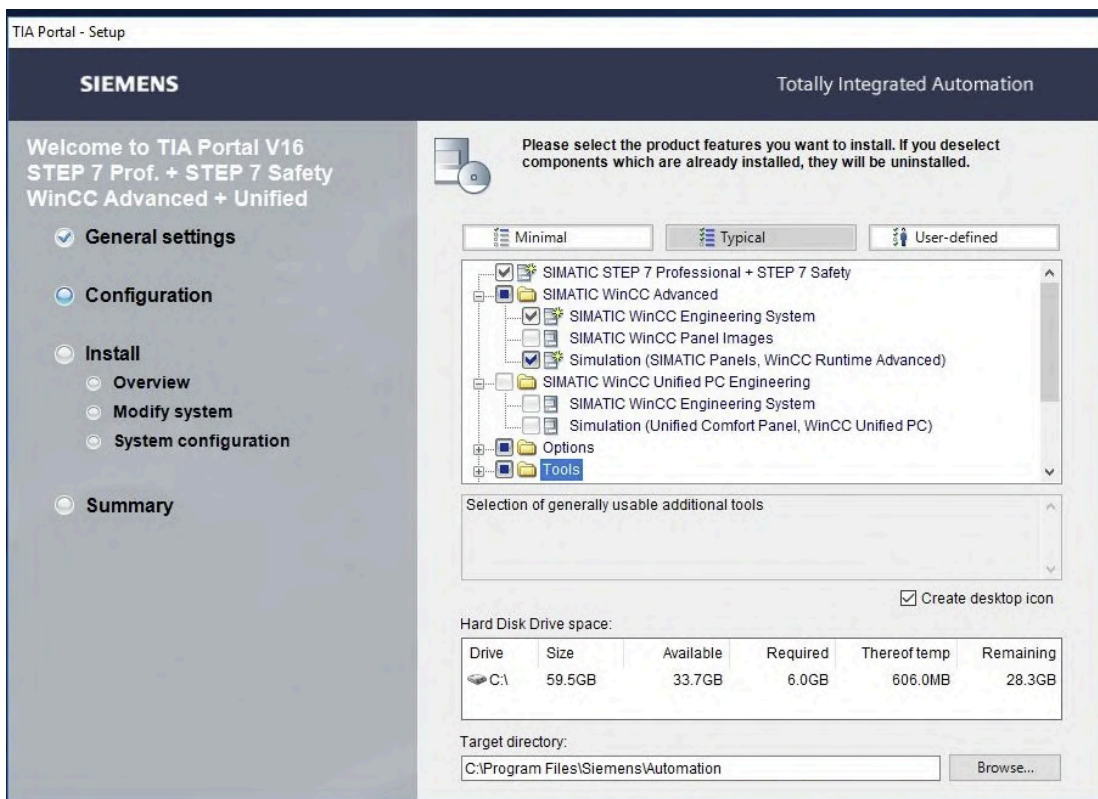
Przed pojawieniem się programowalnych sterowników logicznych (PLC) w większości procesów produkcyjnych dominowały systemy oparte na przekaźnikach. Przekaźniki te sterowały obciążeniami w oparciu o prostą logikę, która została zaimplementowana poprzez



fizyczne okablowanie urządzeń. Schematy elektryczne zostały zaprojektowane do reprezentowania tych obwodów przekaźnikowych w formie przypominającym drabinę. Kiedy wprowadzono sterowniki PLC, opracowano programowanie logiki drabinkowej w celu odtworzenia układu tych obwodów opartych na przekaźnikach. W rezultacie logika drabinkowa stała się jednym z najwcześniejszych języków programowania sterowników PLC i jest nadal szeroko stosowana ze względu na swoją prostotę.

Od czasu jej opracowania, logika drabinkowa przeszła znaczące postępy, ale jej podstawowe zasady działania pozostają w dużej mierze niezmienione. Programowanie logiki drabinkowej ocenia warunki w sposób sekwencyjny, podobnie jak oryginalne systemy przekaźnikowe.



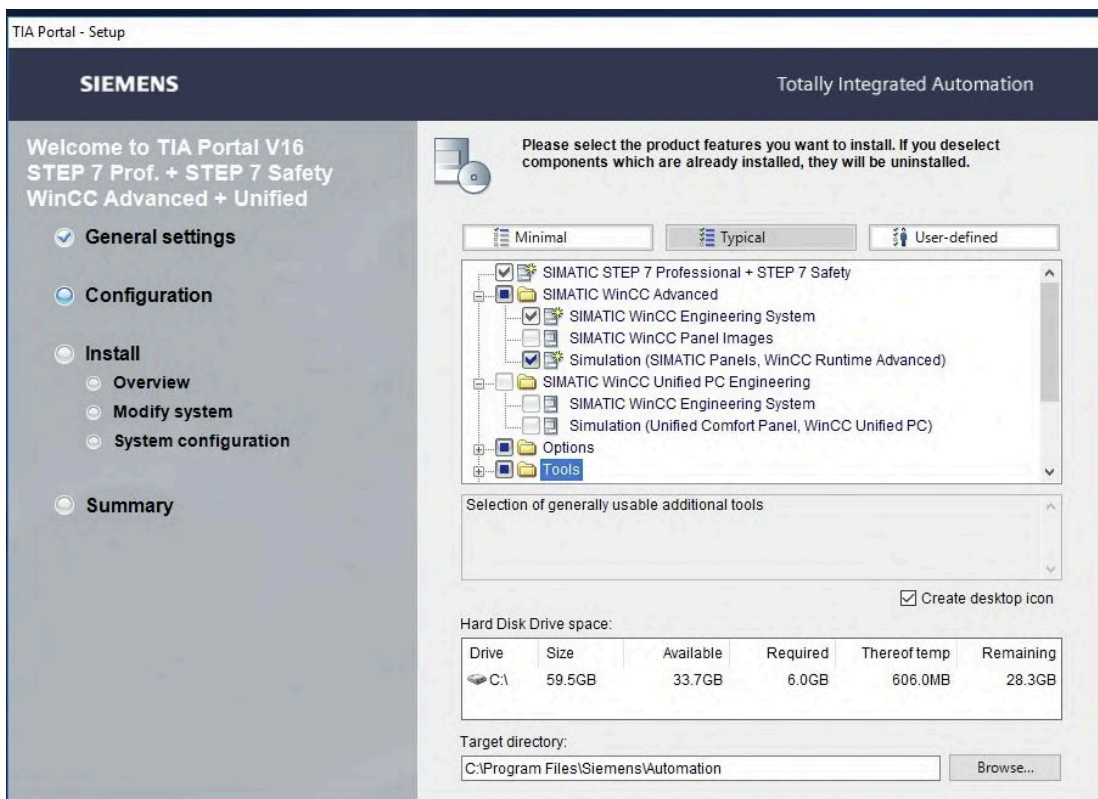


Zalety logiki drabinkowej w programowaniu sterowników PLC

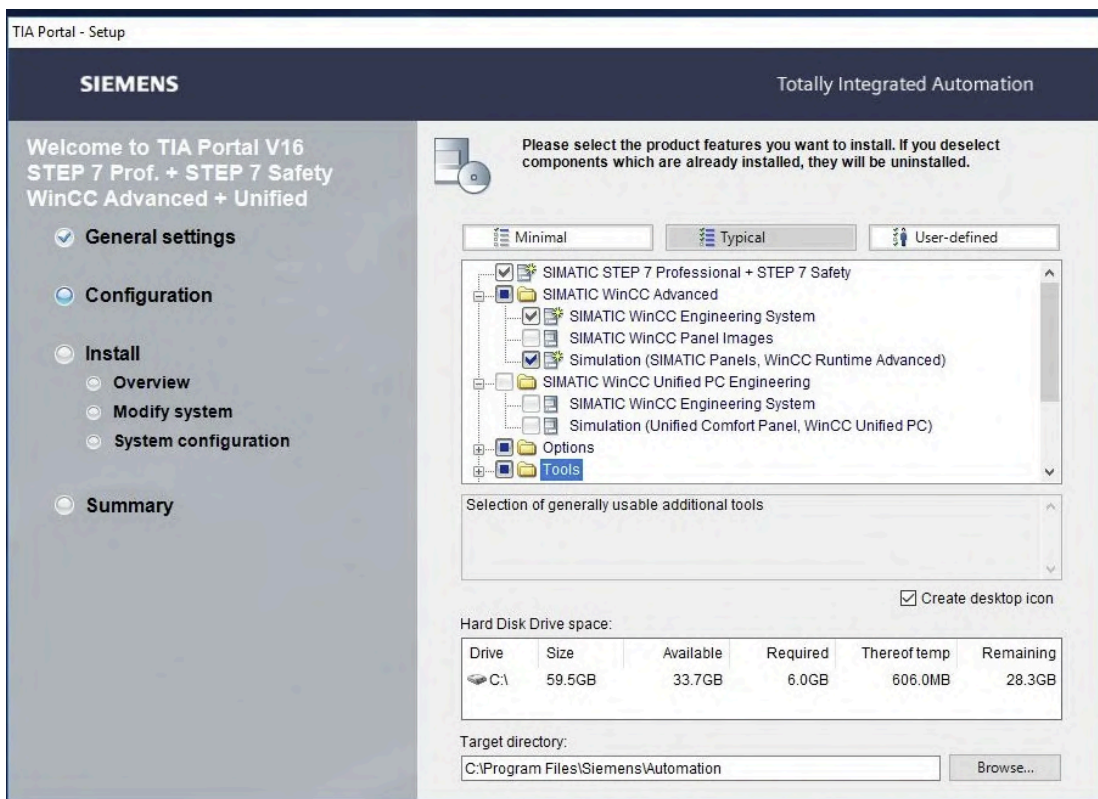
- **Łatwość wdrożenia i rozwiązywania problemów** – Logika drabinkowa to wizualny język programowania, który zapewnia wyraźne potwierdzenie statusu dla większości instrukcji. Ułatwia to użytkownikom, nawet tym z ograniczoną wiedzą na temat procesu, śledzenie przepływu programu i zrozumienie logiki.
- **Modułowa konstrukcja** – Logika drabinkowa jest wysoce modułowa, umożliwiając łatwe modyfikacje poprzez dodawanie lub usuwanie logiki. Każdy szczebel reprezentuje oddzielny warunek, który można modyfikować w razie potrzeby bez wpływu na resztę programu.
- **Odporność i konsekwencja** – Logika drabinkowa pozwala na implementację wielu funkcji przy zachowaniu standardowego formatu. Chociaż nie oferuje pełnej elastyczności, jej standaryzacja zapewnia spójność w różnych implementacjach, dzięki czemu kod jest niezawodny i łatwiejszy w utrzymaniu.

Wady logiki drabinkowej

- **Stroma krzywa procesu nauki** – Chociaż język Ladder jest prostym językiem, może nie być intuicyjny dla osób z doświadczeniem w językach takich jak C, C++, Java lub Python. Jednak osoby z doświadczeniem w inżynierii elektrycznej lub programowaniu montażowym mogą uznać go za łatwiejszy do zrozumienia.
- **Wolniejszy rozwój** – Ze względu na swój wizualny interfejs, rozwój Ladder'a może zająć więcej czasu. Charakter interfejsu „przeciągnij i upuść” często spowalnia proces w porównaniu z innymi nowoczesnymi językami programowania.
- **Nie jest to idealne rozwiązanie dla złożonych aplikacji** – Logika drabinkowa doskonale radzi sobie z sekwencyjną logiką logiczną, ale staje się mniej wydajna, gdy mamy do czynienia z bardziej zaawansowanymi systemami sterowania, takimi jak PID, sterowanie przepływem i pętle sprzężenia zwrotnego. Te bardziej złożone zadania są trudniejsze do wdrożenia i zrozumienia przy użyciu logiki drabinkowej.



Pomimo tych wad, logika drabinkowa pozostaje najczęściej używanym językiem programowania sterowników PLC na świecie. Jest on stosunkowo łatwy w utrzymaniu i obsłudze, zwłaszcza dla użytkowników, którzy nie mają stałego kontaktu z programowaniem PLC.



3.3 Tekst strukturalny

Structured Text (ST) to język programowania sterowników PLC podobny do języka C lub Pascal. Pozwala on użytkownikowi na wprowadzanie linii kodu, które wykonują się w sekwencji, wykonują określone funkcje, oceniają warunki logiczne i aktywują odpowiednie wyjścia sterownika PLC. Dla osób z doświadczeniem w tradycyjnych językach programowania, takich jak C, C++, Java lub Python, Structured Text oferuje płynne przejście do świata sterowników PLC. Ponadto można go łatwo modyfikować w edytorach tekstu, dzięki czemu można go szybko wdrożyć bez konieczności stosowania fizycznego sprzętu.

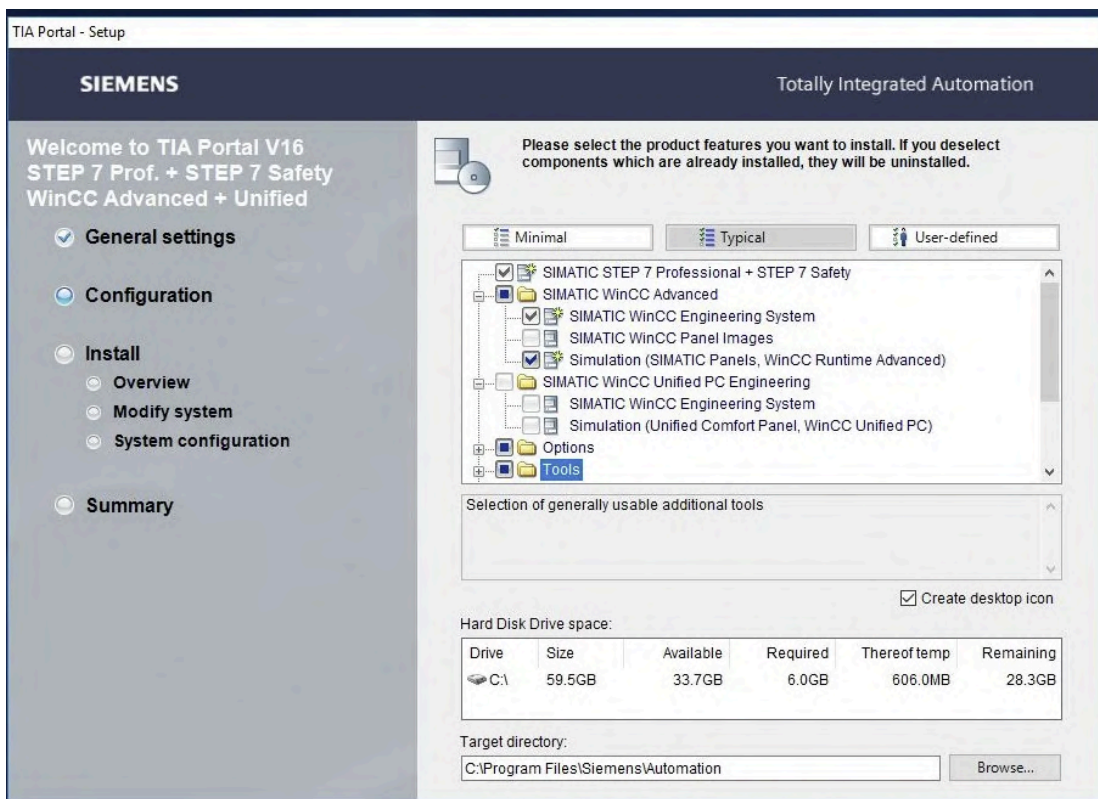
```

Network 3: .....
Comment
1 IF "xLevelUpper" OR NOT "xRegulatorStart" THEN //Level is
2   "xPumpEnable" := FALSE; // If true, turn off the pump
3 ELSIF NOT "xLevelLower" THEN // Check if the liquid level
4   "xPumpEnable" := TRUE; // If true, turn on the pump
5 END_IF;
6

```

Zalety tekstu strukturalnego w programowaniu sterowników PLC

- **Znajomość dla programistów** – Tekst strukturalny jest intuicyjny dla osób z doświadczeniem w tworzeniu oprogramowania, zwłaszcza w językach takich jak C lub Java. Wykorzystuje podobne struktury, zasady programowania i funkcje, ułatwiając programistom przejście do programowania PLC.
- **Wysoka elastyczność** – Structured Text oferuje większą elastyczność w porównaniu do innych języków programowania PLC, pozwalając na implementację bardziej złożonych funkcjonalności. Sprawia to, że jest to potężne narzędzie dla zaawansowanych użytkowników.



- **Kompatybilność międzyplatformowa** – Structured Text jest standaryzowany w większości systemów PLC, co oznacza, że można go łatwo migrować między różnymi platformami. W przeciwieństwie do innych języków, które mogą znacznie różnić się między systemami, Structured Text zapewnia spójność zarówno w środowiskach sprzętowych, jak i programowych.

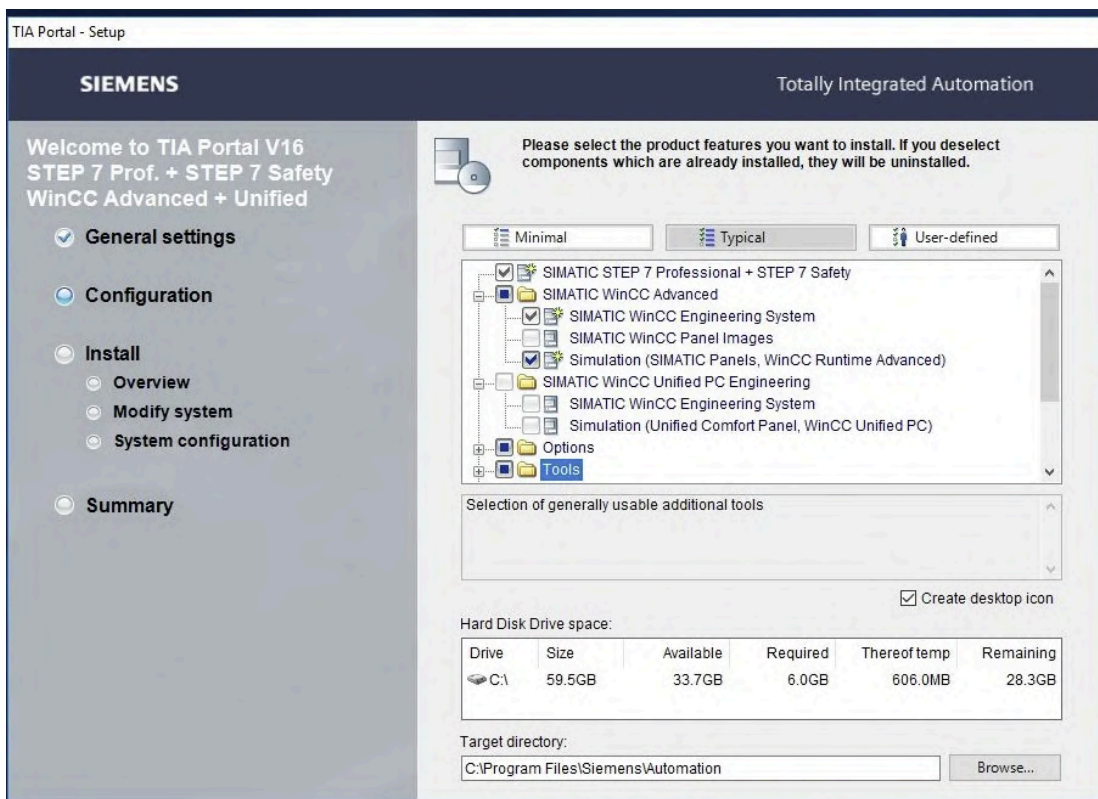
Wady tekstu strukturalnego

- **Trudne do rozwiązania** – W porównaniu z logiką drabinkową, tekst strukturalny może być trudniejszy do rozwiązania. Brakuje w nim wizualnych wskazówek i często wymaga więcej linii kodu, co utrudnia osobom niezaznajomionym z językiem śledzenie i diagnozowanie problemów w procesie.
- **Skłonność do błędów** – Oferując większą elastyczność, Structured Text poświęca część standaryzacji występującej w innych językach programowania PLC. Wymaga to od programistów przestrzegania najlepszych praktyk, takich jak mechanizmy awaryjne i obsługa błędów, aby zapobiec potencjalnym awariom systemu.

Generalnie zalecamy naukę Structured Text dopiero po uzyskaniu solidnego zrozumienia logiki drabinkowej, chyba że masz doświadczenie w innym języku programowania. Ze względu na wyzwania związane z tekstem strukturalnym, nie jest on powszechnie używany w środowiskach produkcyjnych. Jest jednak bardzo skuteczny w zadaniach takich jak manipulacja danymi, implementacja pętli FOR i obsługa innych struktur, które wymagałyby bardziej złożonych kroków w logice drabinkowej.

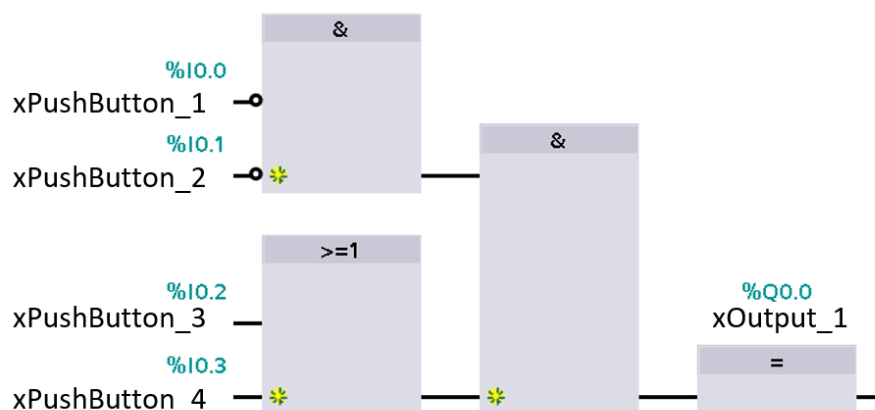
3.4 Schematy bloków funkcyjnych

Function Block Diagram (FBD) to język programowania stworzony z myślą o procesach chemicznych. Umożliwia on użytkownikowi wizualne przedstawienie przebiegu procesu, przy



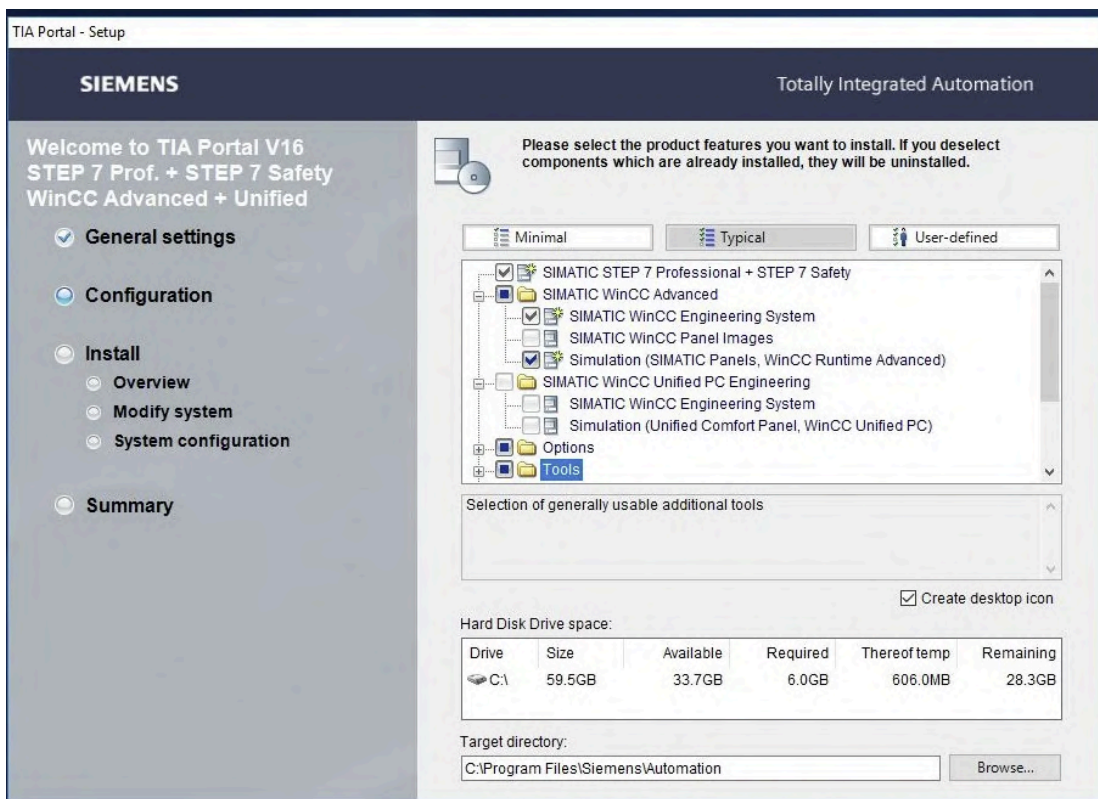
użyciu odpowiednich przejść między instrukcjami. Edytor wizualny jest intuicyjny i przyjazny dla użytkownika, zapewniając naturalne podejście do wdrażania określonych przepływów procesów.

W programowaniu sterowników PLC jednym z najczęstszych zastosowań schematów bloków funkcyjnych jest tworzenie sterowników PID. Wizualny charakter FBD ułatwia wdrażanie, wizualizację, dostrajanie i rozwiązywanie problemów z regulatorami PID bezpośrednio w terenie.



Zalety schematów bloków funkcyjnych w programowaniu sterowników PLC

- **Elastyczny edytor wizualny** – Edytor schematów bloków funkcyjnych jest bardzo przyjazny dla użytkownika i pozwala na proste tworzenie dowolnego układu. Zapewnia łatwy w użyciu interfejs do tworzenia przepływów procesów.
- **Idealny dla złożonych struktur programowania** – W logice drabinkowej do obsługi złożonych zadań może być wymaganych wiele szczebli. W przeciwieństwie do tego,



FBD pozwala na bezpośrednią implementację złożonych instrukcji, takich jak te dla pętli PID, sterowania ruchem i instrukcji dodatkowych (AOI), na jednej stronie.

- **Przyjazny dla użytkownika** – Edytor wizualny w FBD jest intuicyjny dla większości użytkowników. Wykorzystuje on system „przeciągnij i upuść”, ułatwiając odtwarzanie układów procesów bez zbędnego zgadywania.

Wady schematów bloków funkcyjnych

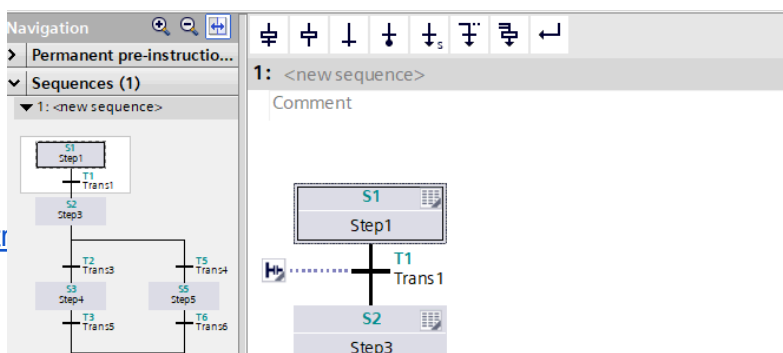
- **Trudne do standaryzacji** – Ze względu na elastyczność układu, standaryzacja programów tworzonych w FBD może być trudna. Każdy programista może opracować inne podejście, co może utrudnić innym śledzenie przepływu informacji.
- **Wyzwania na większą skalę** – FBD działa dobrze w przypadku implementacji na małą skalę. Jednak wraz ze wzrostem złożoności programu łatwo zgubić się w wielu arkuszach instrukcji.

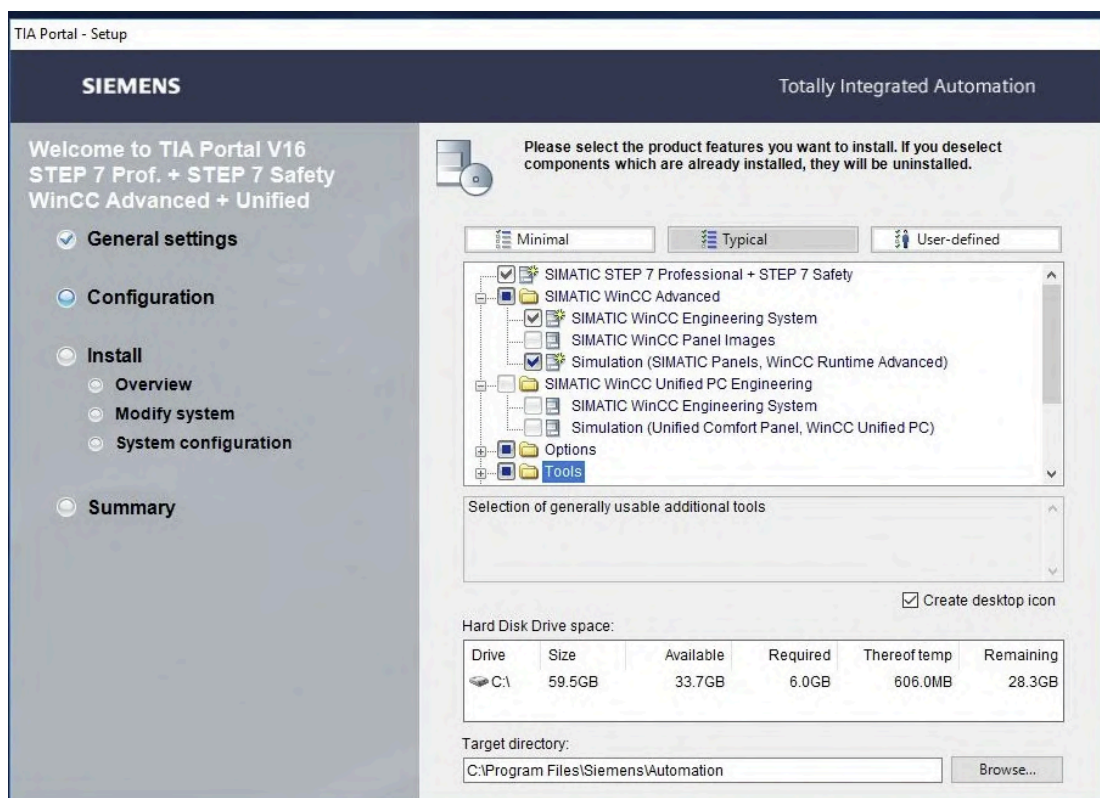
Schematy bloków funkcyjnych są szczególnie przydatne w przypadku skalowania analogowego, pętli PID i sekwencji sterowania ruchem. Po zapoznaniu się z tymi obszarami warto rozpocząć naukę FBD. Zalecamy jednak opanowanie języka Ladder przed przejściem do FBD.

3.5 Schematy funkcji sekwencyjnych

Jak sama nazwa wskazuje, Schematy Funkcji Sekwencyjnych (SFC) są idealne do procesów, które muszą przebiegać w określonej kolejności. Innym przykładem może być zautomatyzowany proces mycia samochodów, który zapewnia, że pojazdy są czyszczone w sposób efektywny i we właściwej sekwencji.

Wyobraźmy sobie w pełni zautomatyzowany system





myjni samochodowej z różnymi etapami, takimi jak mycie wstępne, namydlenie, szczotkowanie, płukanie i suszenie. Gdy pojazd wjeżdża do systemu, proces przebiega zgodnie z serią predefiniowanych kroków. Oto uproszczona wersja tych kroków:

Krok 1 - Wykrycie i przygotowanie pojazdu: System wykrywa pojazd przy wejściu. Sprawdza, czy samochód jest prawidłowo ustawiony na przenośniku, czy obszar mycia jest czysty i czy wszystkie systemy są gotowe. Jeśli warunki są spełnione, samochód jest przenoszony do obszaru mycia i rozpoczyna się proces. Jeśli nie, proces jest opóźniany lub przerywany.

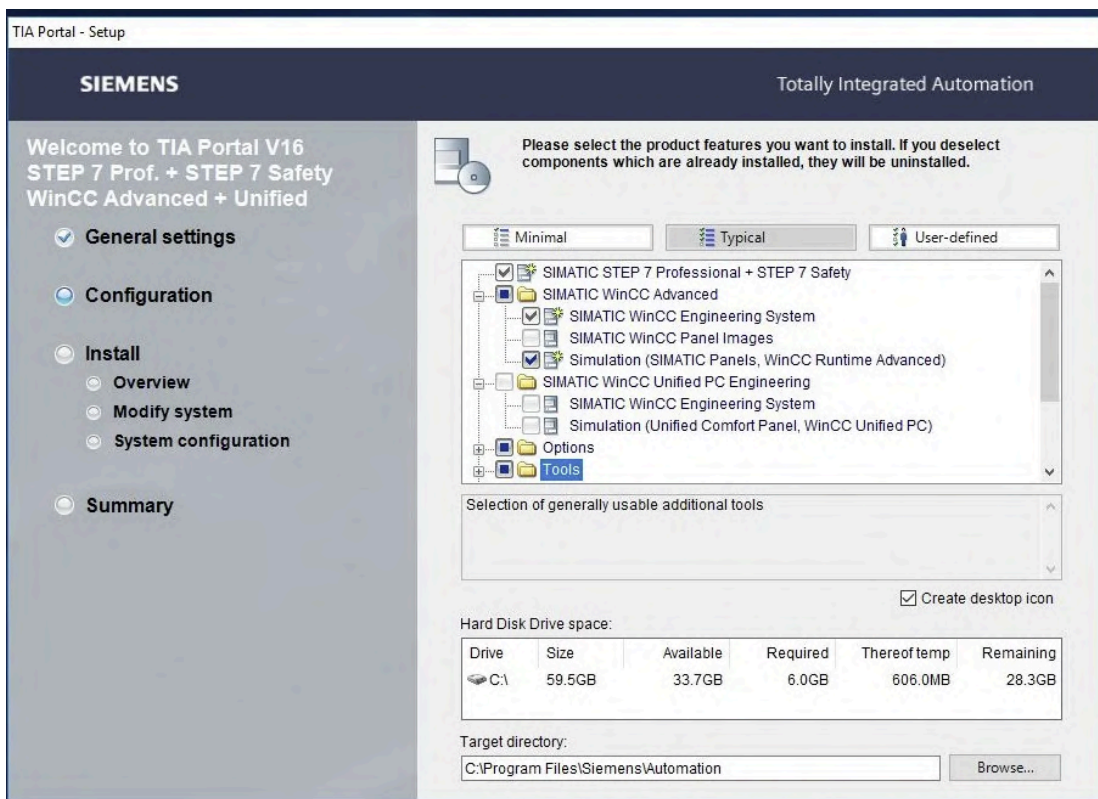
Krok 2 – Cykl mycia wstępnego: Samochód jest spryskiwany wodą w celu usunięcia luźnego brudu. Czujniki upewniają się, że pojazd przeszedł przez cały obszar mycia wstępnego przed przejściem dalej.

Krok 3 – Aplikacja płynu do czyszczenia: System nakłada płyn na pojazd za pomocą rozpylaczy. System zapewnia całkowite pokrycie i przechodzi do następnego etapu dopiero po zakończeniu aplikacji płynu.

Krok 4 – Cykl szczotkowania i szorowania: Obracające się szczotki czyszczą powierzchnię samochodu. System monitoruje prędkość i nacisk szczotek, aby zapewnić optymalne czyszczenie bez uszkodzenia pojazdu. Po zakończeniu szorowania proces jest kontynuowany.

Krok 5 – Cykl płukania: Samochód jest dokładnie spłukiwany czystą wodą. System monitoruje natężenie przepływu i upewnia się, że wszystkie pozostałości mydła zostały usunięte.

Krok 6 – Suszenie: Dmuchawy o dużej mocy osuszają pojazd podczas jego przemieszczania się przez sekcję suszenia. Czujniki potwierdzają zakończenie suszenia i pojazd zostaje zwolniony z systemu.



Jak widać w tym przykładzie, etapy procesu są wykonywane sekwencyjnie, a każdy krok ma zdefiniowane warunki początkowe i przepływ. W logice drabinkowej można to zaimplementować za pomocą instrukcji SET/RESET, ale SFC jest bardziej odpowiednim podejściem do takich zadań.

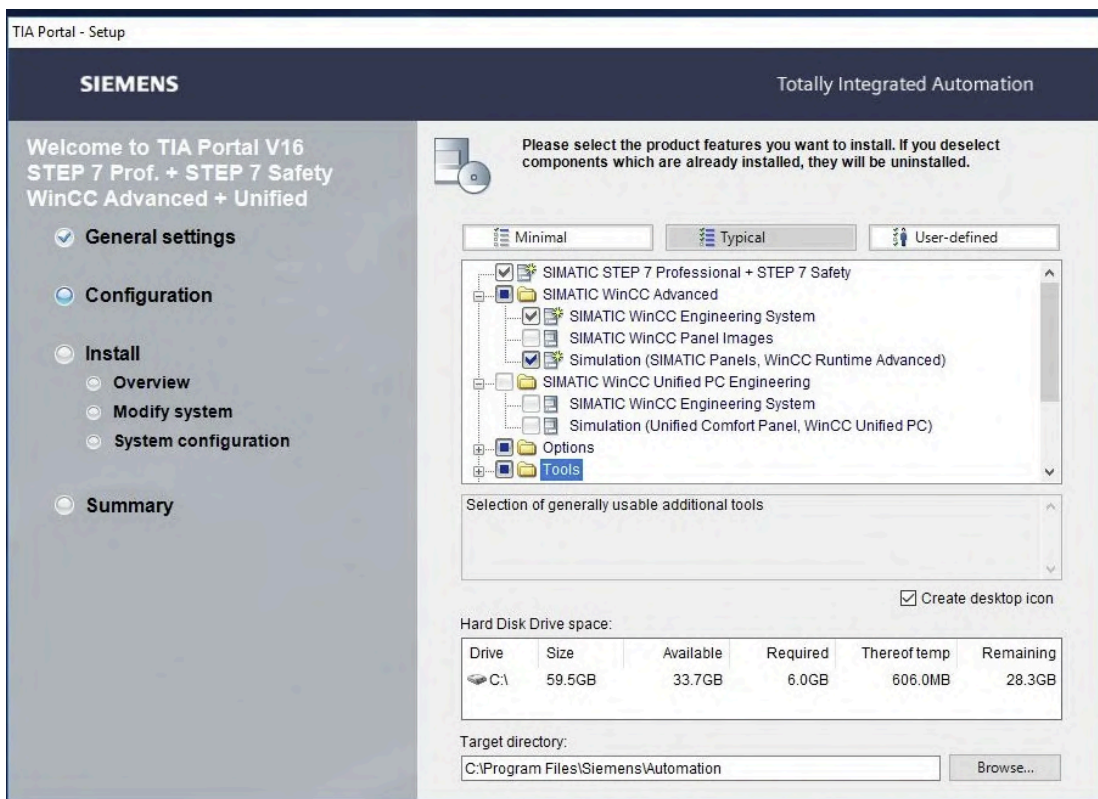
Zalety sekwencyjnych wykresów funkcji w programowaniu sterowników PLC

- **Naśladuje przebieg procesu większości operacji chemicznych** – Sekwencyjne wykresy funkcyjne (SFC) są szczególnie przydatne w procesach chemicznych, takich jak dozowanie. Procesy te wymagają przekształcenia określonej liczby surowych składników w produkt końcowy, a wykresy SFC doskonale radzą sobie z takimi sekwencjami.
- **Integracja z tekstem strukturalnym (ST)** – Wiele edytorów SFC pozwala na łączenie z tekstem strukturalnym, umożliwiając tworzenie bardziej zaawansowanych przepływów logicznych w określonych scenariuszach.

Wady sekwencyjnych wykresów funkcji

- **Ograniczone zastosowanie** – Sekwencyjne wykresy funkcyjne nie są odpowiednie dla procesów, które z natury nie są sekwencyjne, co sprawia, że mają one zastosowanie w stosunkowo niewielkiej liczbie przypadków użycia.
- **Wyzwania związane z przepływami równoległymi** – Podczas gdy SFC może obsługiwać wiele przepływów procesowych, gdy ścieżki rozchodzą się w równoległe przepływy, trudne staje się wdrożenie i rozwiązywanie problemów z oddzielnymi ścieżkami w sposób zapewniający solidne sekwencjonowanie.

SFC są bardzo skuteczne w określonych zastosowaniach, ale mogą być frustrujące, gdy są wymuszane w procesach, które nie są naturalnie sekwencyjne. W miarę zdobywania doświadczenia w środowiskach produkcyjnych niezbędne jest zrozumienie procesu i



przepływu produktu. Zalecamy stworzenie modelu na papierze przed wdrożeniem programowania SFC.

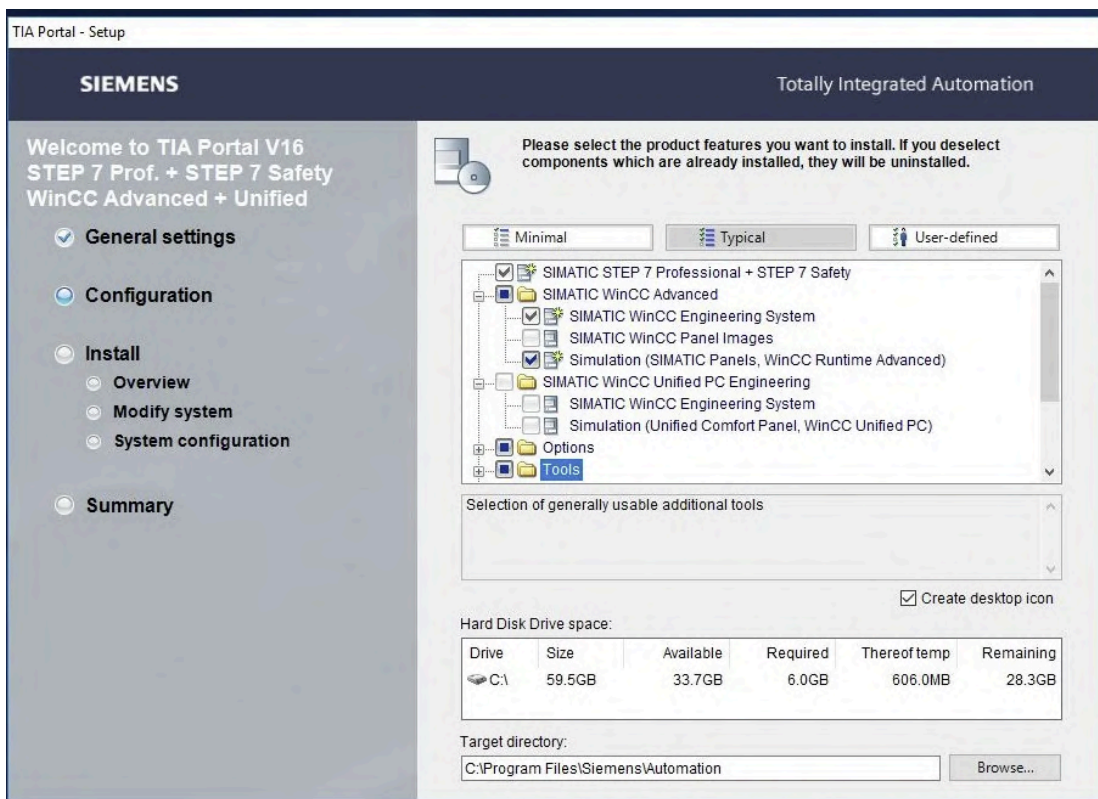
3.6 Listy instrukcji

Listy instrukcji są czasami mylone z tekstem strukturalnym ze względu na podobieństwo ich edytorów. Te dwa języki programowania sterowników PLC występują zazwyczaj na różnych platformach. Na przykład sterowniki Simatic S7-300 i S7-1500 mogą być programowane za pomocą list instrukcji, podczas gdy sterowniki S7-1200 umożliwiają korzystanie wyłącznie z logiki drabinkowej lub SCL (tekst strukturalny).

Jeśli chodzi o przepływ programu, każda linia na liście instrukcji określa instrukcję wraz z warunkami i wynikami jej wykonania.

Network 1:				
Comment				
1	O	"xLevelUpper"	%I0.0	
2	ON	"xRegulatorStart"	%M0.0	
3	R	"xPumpEnable"	%Q0.0	
4				
5	AN	"xLevelLower"	%I0.1	
6	AN	"xLevelUpper"	%I0.0	
7	S	"xPumpEnable"	%Q0.0	
8				

Zalety list instrukcji w programowaniu sterowników PLC



- **Wysoce znormalizowany** – Listy instrukcji przestrzegają ścisłej struktury, wymagając od użytkownika jawnego tworzenia zmiennych, określania warunków i wymieniania każdej instrukcji. Pozostawia to niewiele miejsca na wariacje, co skutkuje kodem, który jest łatwy do naśladowania i zrozumienia.
- **Ukierunkowane na instrukcje** – Jak sama nazwa wskazuje, język ten koncentruje się na instrukcjach, a nie na przepływie danych. Pomaga to zapewnić przejrzystość sposobu przetwarzania danych w programie.

Wady list instrukcji

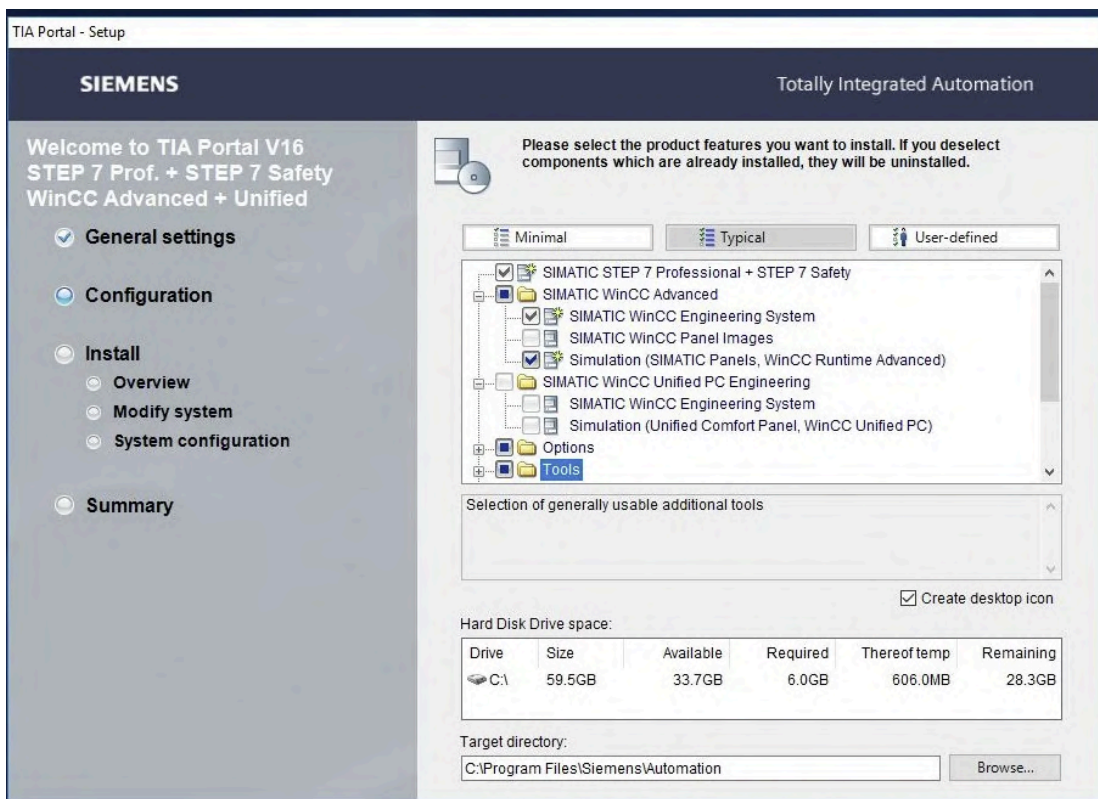
- **Nieobsługiwane na większości platform PLC** – Jak wspomniano wcześniej, listy instrukcji nie są powszechnie używane na platformach PLC. Większość programistów uważa je za nieznane, ponieważ ich styl jest bardziej zbliżony do języków assemblera niż innych nowoczesnych języków programowania dostępnych na rynku.

Tak jak mówiłem, zalecamy, aby każdy programista PLC rozpoczął od opanowania logiki drabinkowej, ponieważ jest ona najczęściej używana w branży. Jednak w miarę zdobywania doświadczenia ważne jest, aby nauczyć się dodatkowych języków, takich jak listy instrukcji, które mogą oferować łatwiejsze sposoby implementacji określonych procesów w zaawansowanych scenariuszach programowania.

ROZDZIAŁ 4 | PODSTAWY JĘZYKA LADDER

4.1 Jak zrozumieć Ladder Logic

Logika drabinkowa jest jednym z 5 najczęściej używanych języków programowania sterowników PLC w przemyśle wytwórczym. Przed zastosowaniem programowalnych sterowników logicznych (PLC) fabryki polegały na systemach opartych na przekaźnikach do



sterowania różnymi obciążeniami w zależności od konfiguracji okablowania. Przekazniki te były drogie, trudne w utrzymaniu i mało elastyczne. Gdy sterowniki PLC stały się standardem, opracowano język LADDER, aby zachować znajomość istniejących systemów, co uczyniło ją pierwszym językiem programowania PLC.

Nazwa „Ladder Logic” (logika drabinkowa) pochodzi od jej wizualnego układu, który przypomina drabinę. Po lewej stronie zdefiniowane są warunki, podczas gdy prawa strona zawiera akcje, które występują, gdy warunki są spełnione. Każdy szczebel drabiny przepływa od lewej do prawej, a sterownik PLC przetwarza go od góry do dołu.

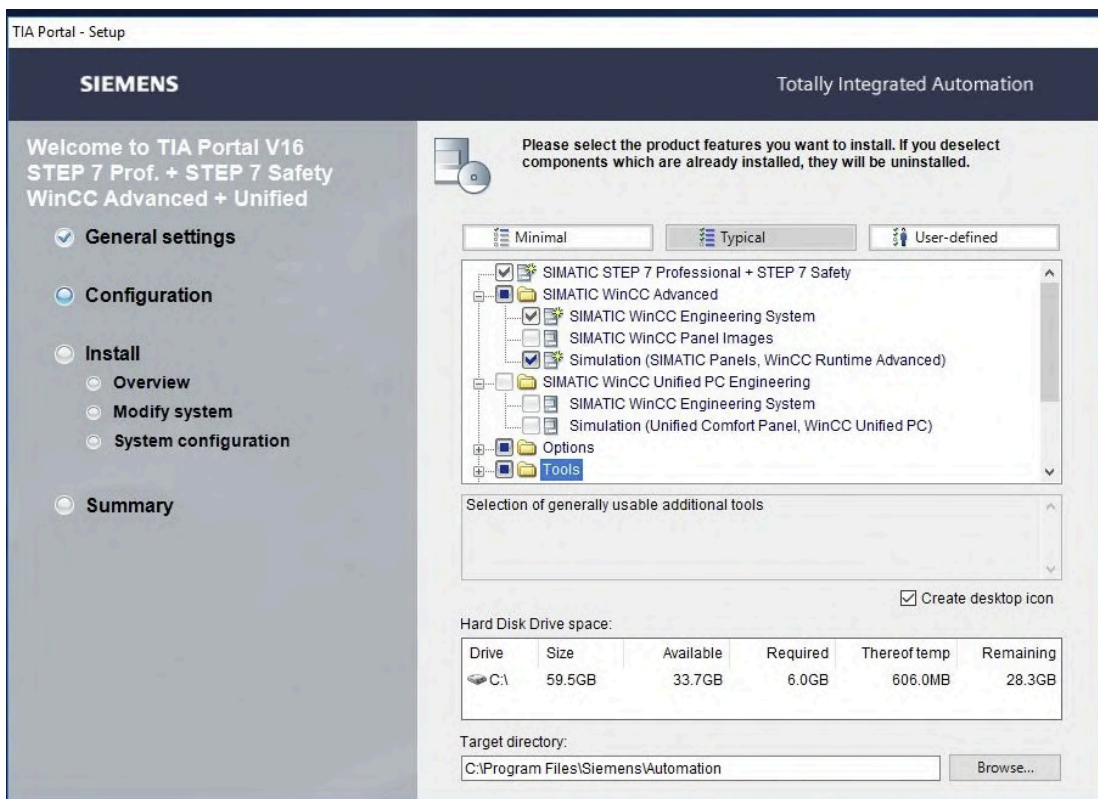
Ze względu na łatwość użycia, podobieństwo do obwodów elektrycznych i prostotę w rozwiązywaniu problemów, logika drabinkowa pozostaje bardzo popularna wśród programistów PLC. Dla wielu osób nauka logiki drabinkowej jest pierwszym krokiem w kierunku kariery w systemach sterowania i programowaniu PLC.

W tej sekcji zbadamy podstawowe komponenty logiki drabinkowej, podstawowe pojęcia i kroki wymagane do jej opanowania.

W standardowej bibliotece znajduje się kilka grup instrukcji logiki drabinkowej:

1. Bitowe operacje logiczne
2. Operacje timera

Name	Description
General	
Insert network	Insert network
Empty box	Empty box [Shift+F5]
Open branch	Open branch [Shift+F8]
Close branch	Close branch [Shift+F9]
Insert input	Insert input
Bit logic operations	
Normally open contact	Normally open contact [Shift+F2]
Normally closed contact	Normally closed contact [Shift+F3]
Invert RLO	Invert RLO
Assignment	Assignment [Shift+F7]
Negate assignment	Negate assignment
Reset output	Reset output
Set output	Set output
Set bit field	Set bit field
Reset bit field	Reset bit field
Set/reset flip-flop	Set/reset flip-flop
Reset/set flip-flop	Reset/set flip-flop
Scan operand for positive signal edge	Scan operand for positive signal edge
Scan operand for negative signal edge	Scan operand for negative signal edge
Set operand on positive signal edge	Set operand on positive signal edge
Set operand on negative signal edge	Set operand on negative signal edge
Scan RLO for positive signal edge	Scan RLO for positive signal edge
Scan RLO for negative signal edge	Scan RLO for negative signal edge
Detect positive signal edge	Detect positive signal edge
Detect negative signal edge	Detect negative signal edge
Timer operations	
Generate pulse	Generate pulse
Generate on-delay	Generate on-delay
Generate off-delay	Generate off-delay
Time accumulator	Time accumulator
Start pulse timer	Start pulse timer
Start on-delay timer	Start on-delay timer



3. Przeciwdziałanie operacjom
4. Operacje komparatora
5. Funkcje matematyczne
6. Operacje przenoszenia
7. Operacje konwersji
8. Operacje sterowania programem
9. Słowne operacje logiczne
10. Przesuwanie i obracanie

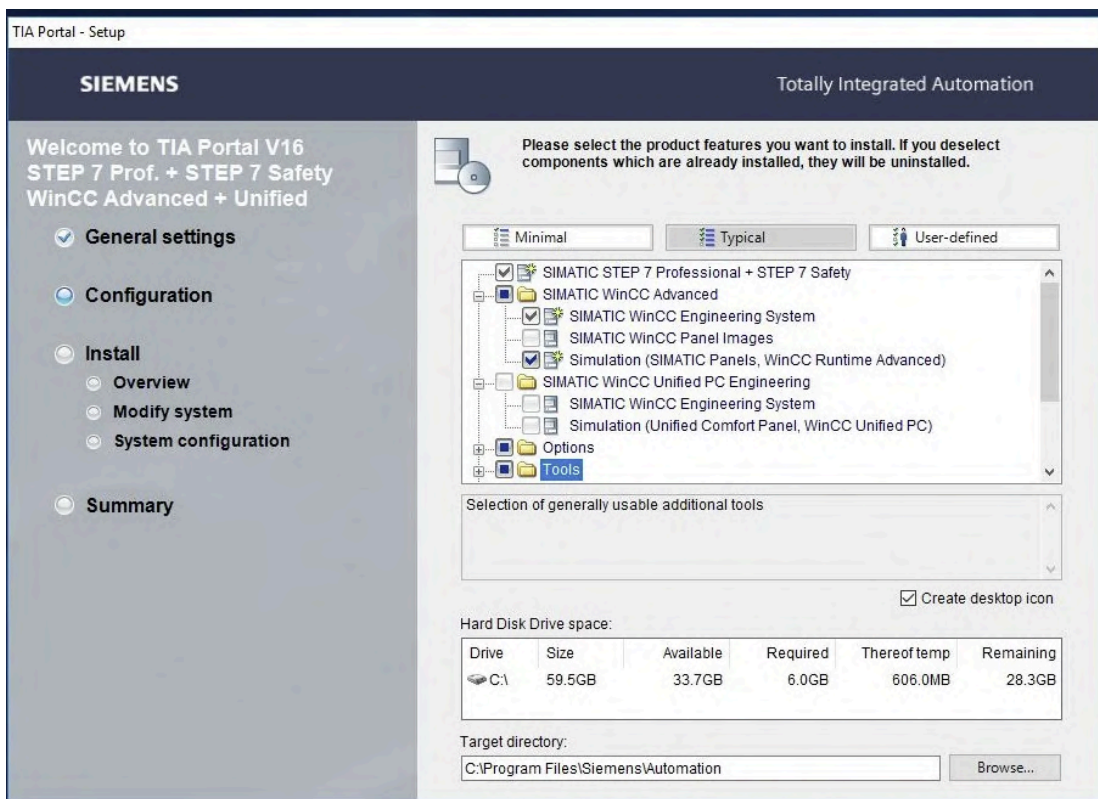
W tym dokumencie znajdziesz wyjaśnienia niektórych instrukcji, które posłużą ci jako pierwszy krok w świat programowania!

Podstawy Języka LADDER

Podobnie jak komputery, sterowniki PLC działają w oparciu o sygnały binarne (stany włączenia lub wyłączenia). W terminologii programistycznej jest to znane jako wartość logiczna, która może mieć tylko dwie możliwe wartości.

Pojedynczy bit w pamięci, który może być ustawiony na 0 lub 1, jest używany w większości podstawowych instrukcji PLC. Sterownik PLC uruchamia program po jednym kroku na raz. Podczas przetwarzania każdego szczebla, sterownik PLC odczytuje instrukcje po lewej stronie i sprawdza, czy logika jest ustawiona na TRUE. Logika ma wartość TRUE, gdy hipotetyczny prąd może przepływać przez instrukcję. Każda instrukcja zawiera warunki, które określają, czy ma ona wartość TRUE, czy FALSE.

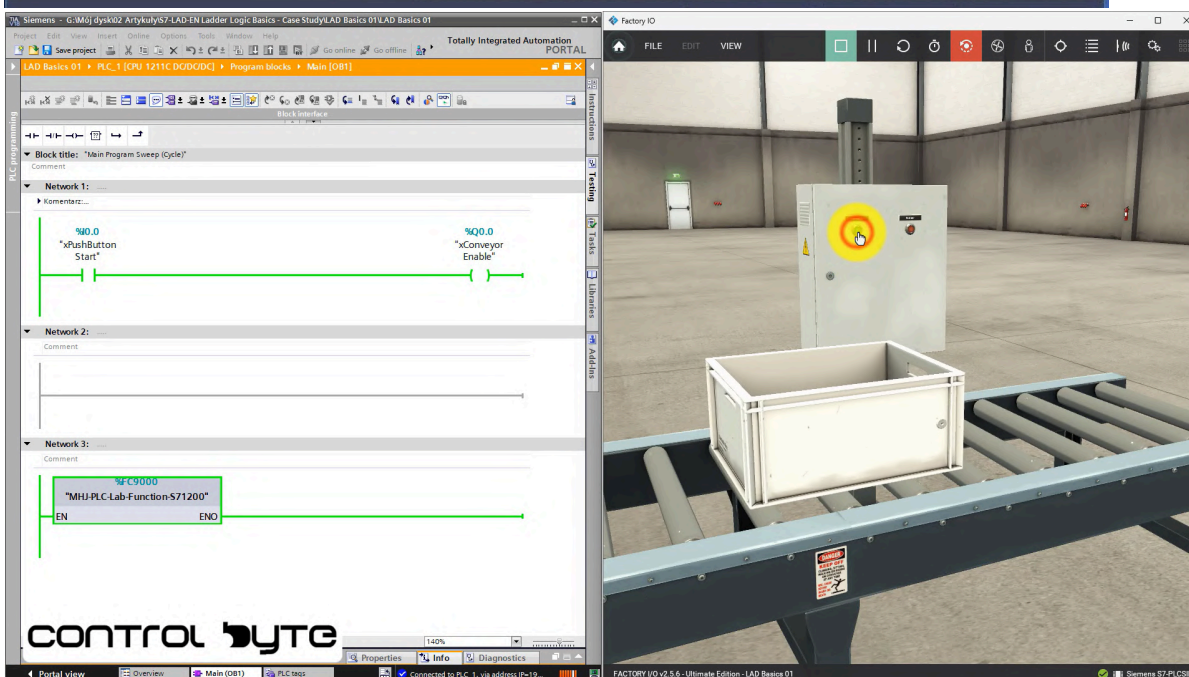
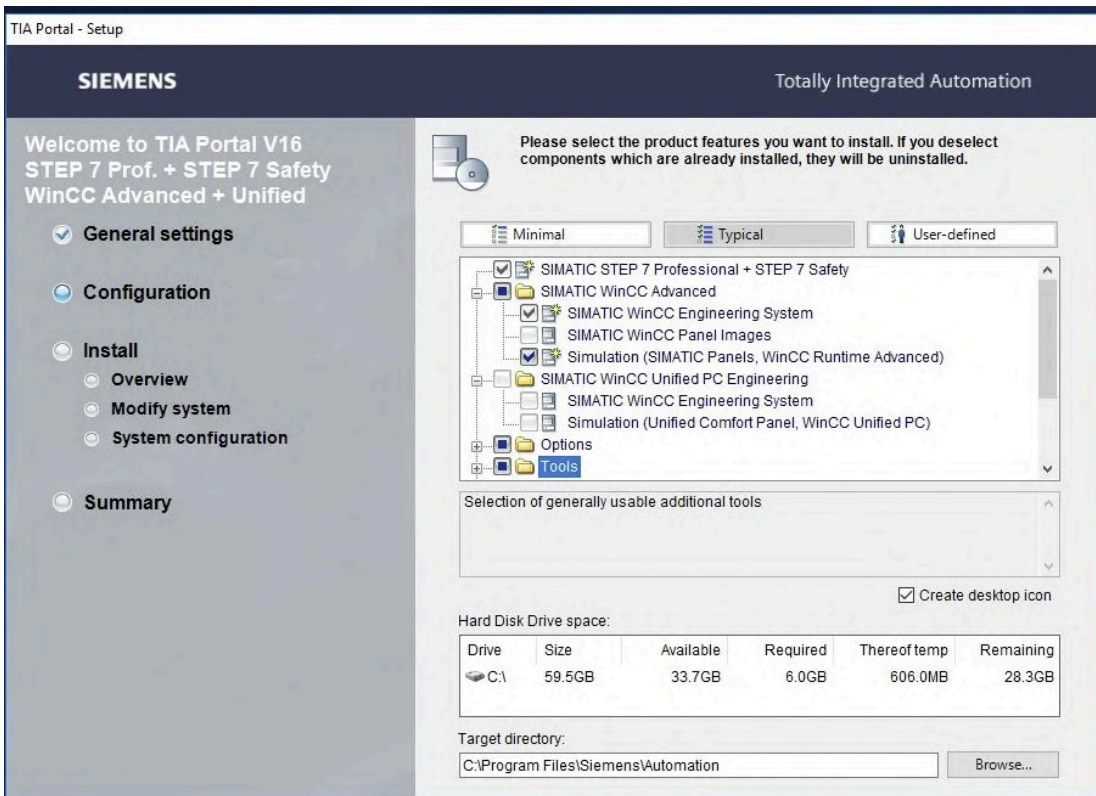
Omówmy teraz dwie najbardziej podstawowe instrukcje języka LADDER używane w TIA Portal: **Styk normalnie otwarty (NO)** i **Styk normalnie zamknięty (NC)**.



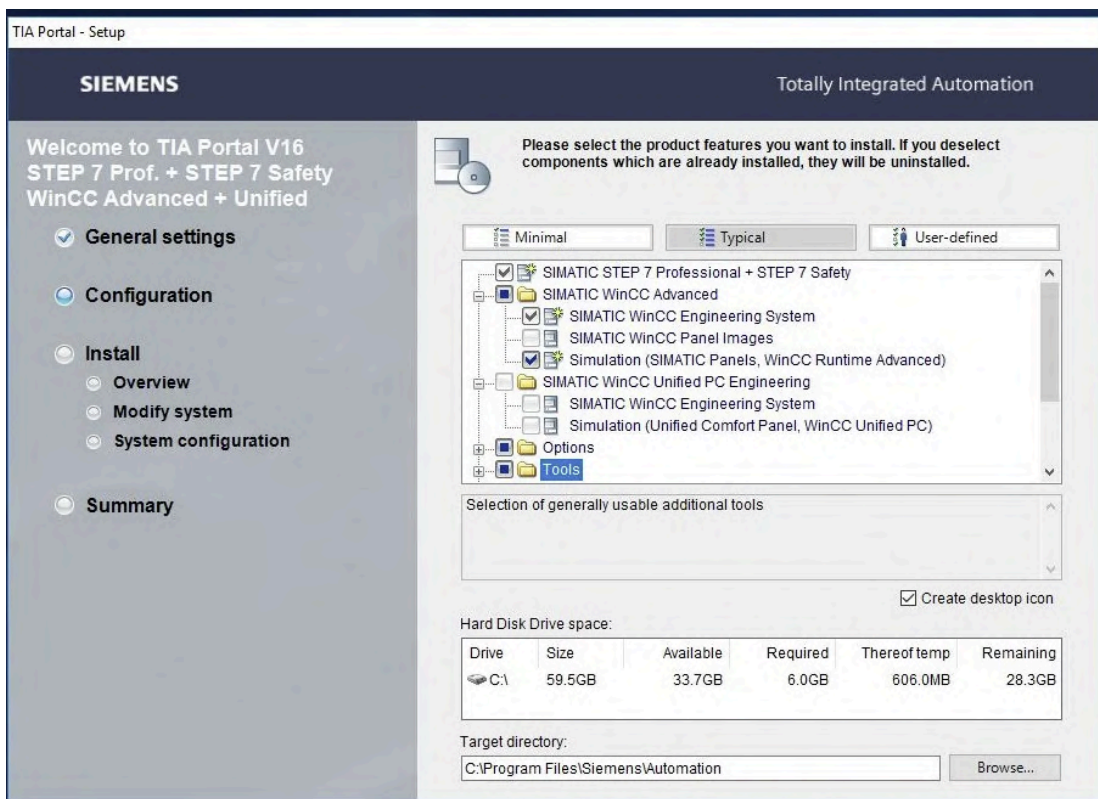
- **Styk normalnie otwarty (NO):** Ta instrukcja wejściowa sprawdza wartość określonego bitu logicznego. Jeśli bit jest ustawiony na 1 (WYSOKI), styk zamyka się, a warunek ma wartość TRUE, umożliwiając przepływ prądu. Jeśli bit jest ustawiony na 0 (NISKI), styk pozostaje otwarty, a warunek ma wartość FALSE.
- **Styk normalnie zamknięty (NC):** Ta instrukcja wejściowa zachowuje się odwrotnie do styku NO. Jeśli bit jest ustawiony na 0 (NISKI), styk zamyka się, a warunek ma wartość TRUE. Jeśli bit jest ustawiony na 1 (WYSOKI), styk otwiera się, a warunek ma wartość FALSE.

Te podstawowe instrukcje są niezbędne do kontrolowania przepływu logiki w programowaniu LADDER w TIA Portal.

Przeanalizujemy podstawowy scenariusz z przyciskiem i przenośnikiem.



control byte

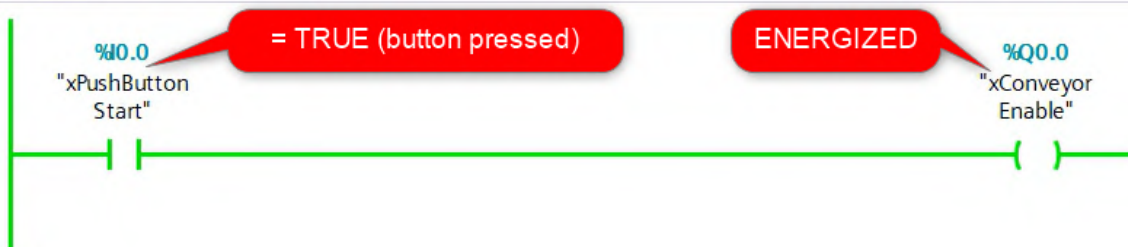
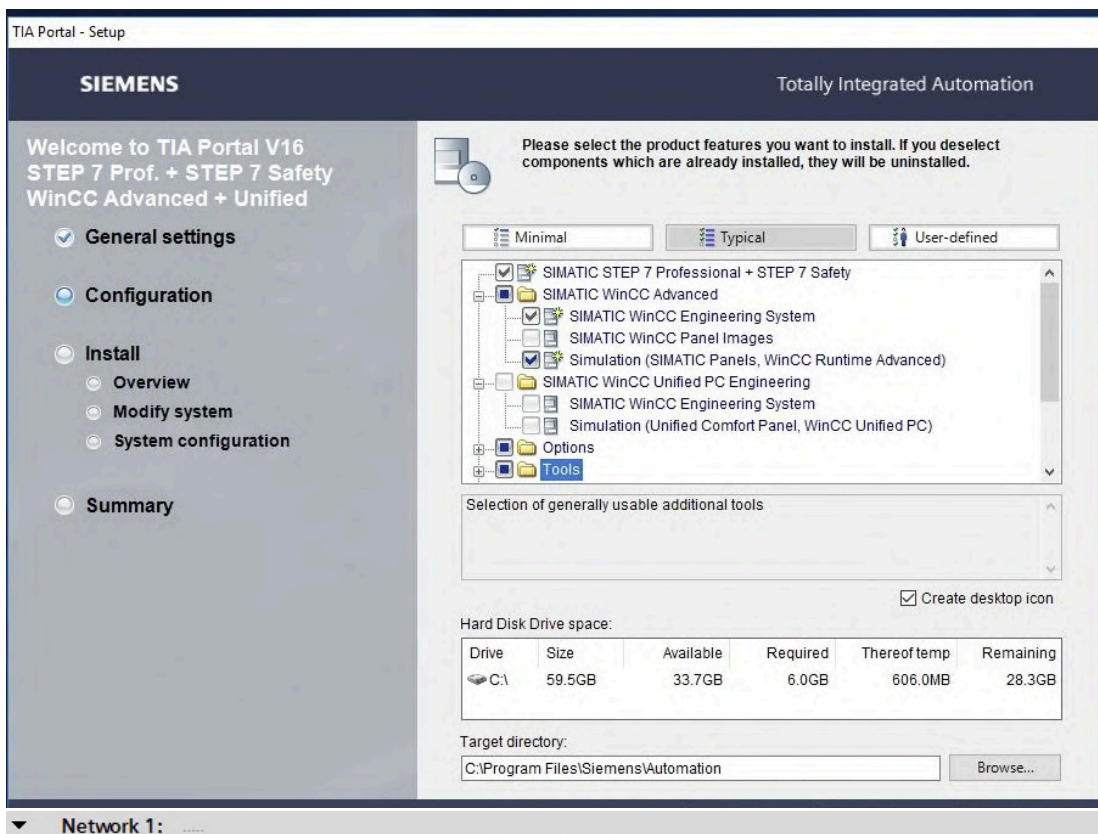


Program w języku Ladder pokazany na obrazku przedstawia zasadę działania systemu przenośnika sterowanego za pomocą wejścia przycisku.



Cykl PLC 1 (przycisk nie naciśnięty - stan NISKI):

- **Stan wejścia:** W górnej sieci przycisk (%I0.0, oznaczony jako "xPushButton Start") jest w stanie niewciśniętym, co oznacza, że wejście ma wartość **FALSE** (0).
- **Logic:** tyk normalnie otwarty (NO) powiązany z przyciskiem nie jest zamknięty, ponieważ wejście ma wartość **FALSE**. W związku z tym przez szczebel nie może przepływać prąd.
- **Stan wyjścia:** Cewka wyjściowa (%Q0.0, oznaczona jako "xConveyor Enable") pozostaje **NIEZASILONA**. Przenośnik nie pracuje w tym cyklu.

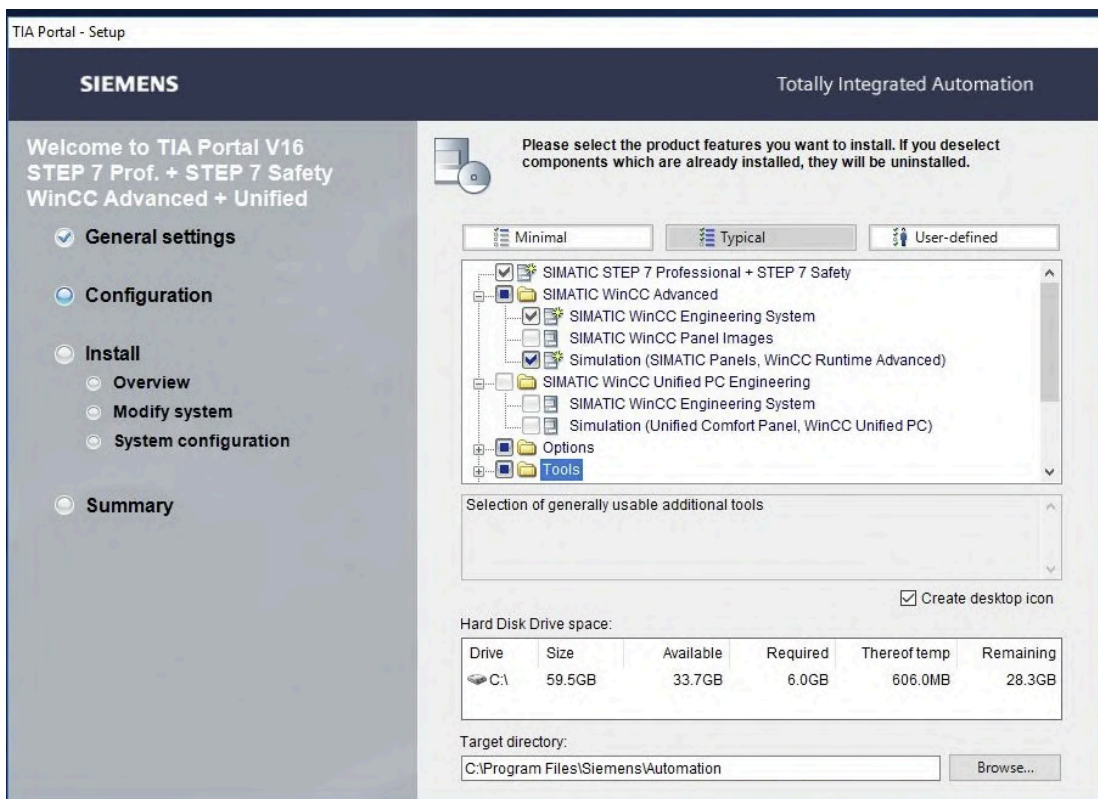


Cykl PLC 2 (naciśnięcie przycisku - stan WYSOKI):

- **Stan wejścia:** W dolnej sieci przycisk (**%I0.0**, "xPushButton Start") jest teraz naciśnięty, co powoduje, że warunek wejściowy jest **TRUE** (1).
- **Logic:** Styk NO zamyka się, ponieważ wejście ma teraz wartość **TRUE**, umożliwiając przepływ prądu przez szczebel.
- **Stan wyjścia:** Cewka wyjściowa (**%Q0.0**, "xConveyor Enable") staje się **ZASILONA**, aktywując przenośnik w tym cyklu.

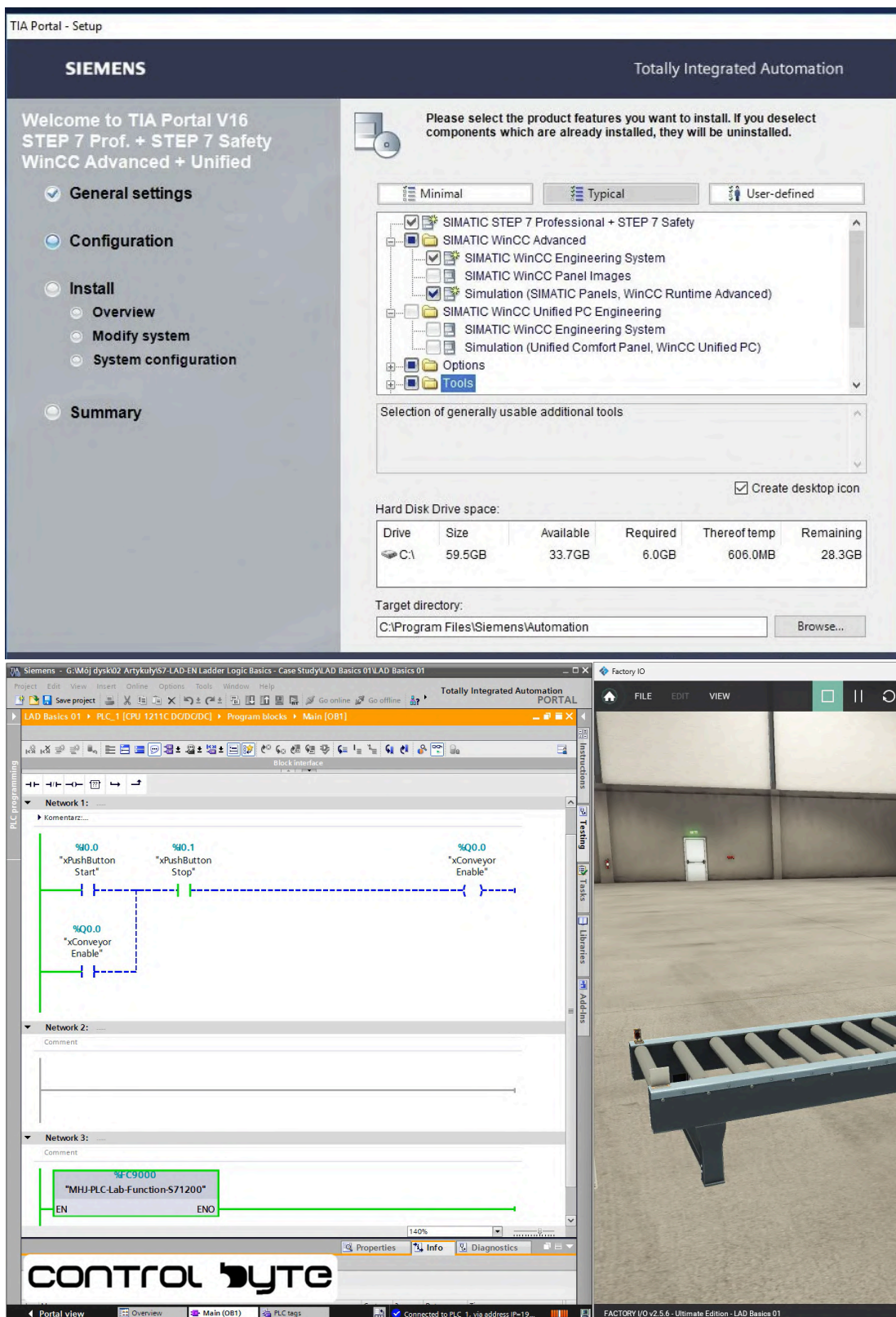
Podsumowanie:

- Gdy przycisk **nie jest wciśnięty** (**%I0.0 = FALSE**), wyjście (**%Q0.0**) pozostaje odłączone od zasilania, a przenośnik nie działa.
- Po **naciśnięciu** przycisku (**%I0.0 = TRUE**), wyjście (**%Q0.0**) jest zasilane i przenośnik jest aktywowany.



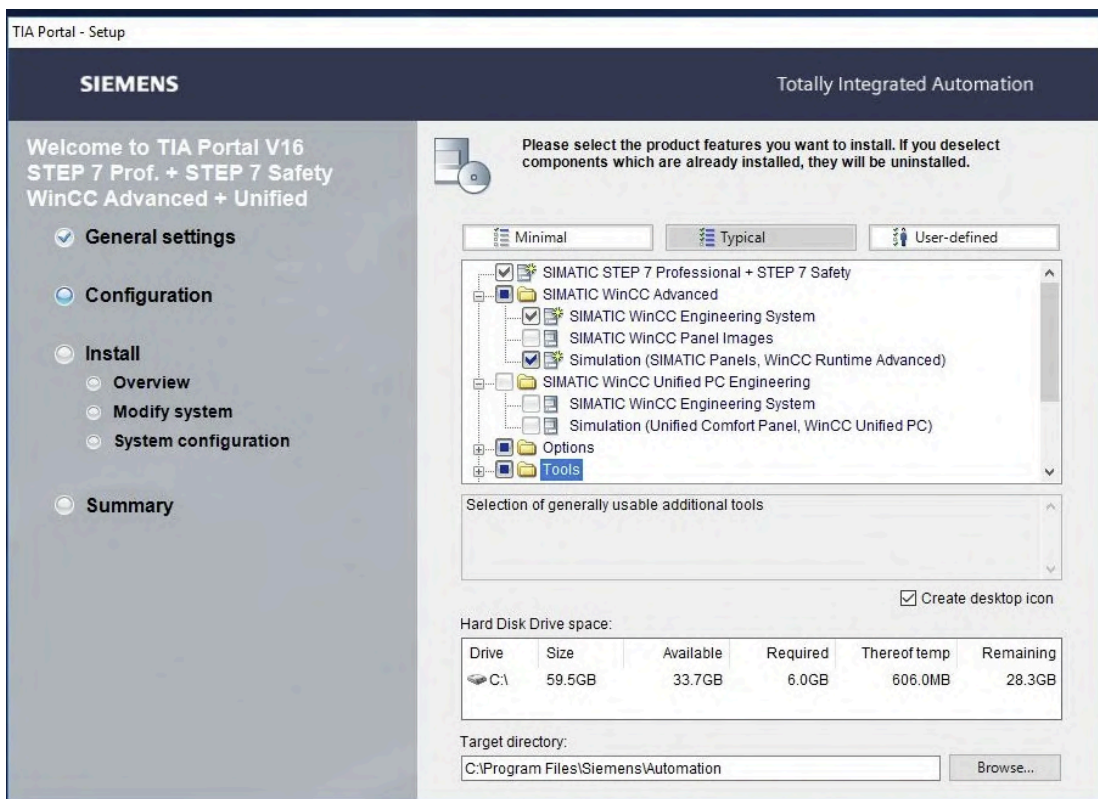
4.2 Struktura LADDER

Teraz, gdy przyjrzeliśmy się podstawowemu przykładowi demonstrującemu działanie pojedynczego szczebla logiki drabinkowej, nadszedł czas, aby porozmawiać o rozgałęzieniach obwodów. Rozgałęzienia obwodów umożliwiają przepływ prądu przez alternatywne ścieżki podczas wykonywania danego szczebla. Podczas gdy instrukcje są nadal przetwarzane w ten sam sposób, musimy wziąć pod uwagę różne trasy, którymi może płynąć prąd.

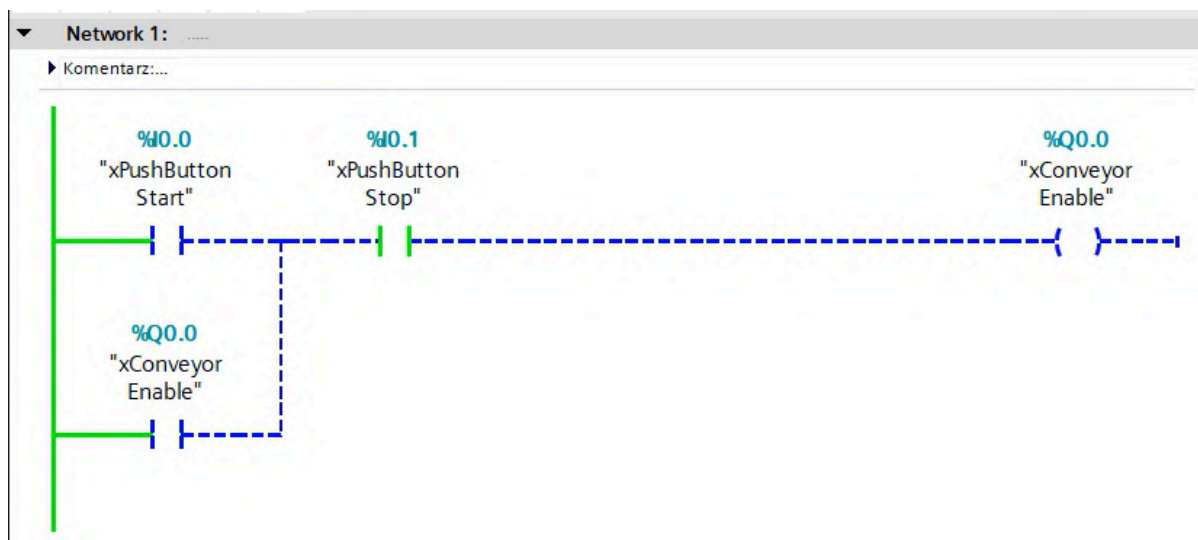


Konstruowanie programu w języku drabinowym Ladder polega na tworzeniu serii warunków i działań przy użyciu gałęzi reprezentujących operacje logiczne, takie jak AND, OR i XOR.

W przykładzie z obrazka można zobaczyć, jak rozgałęzienia są używane do sterowania systemem przenośnika z przyciskami uruchamiania i zatrzymywania. Logika tworzy warunek OR, wymagając zarówno naciśnięcia przycisku „xPushButton Start” (styk normalnie otwarty)



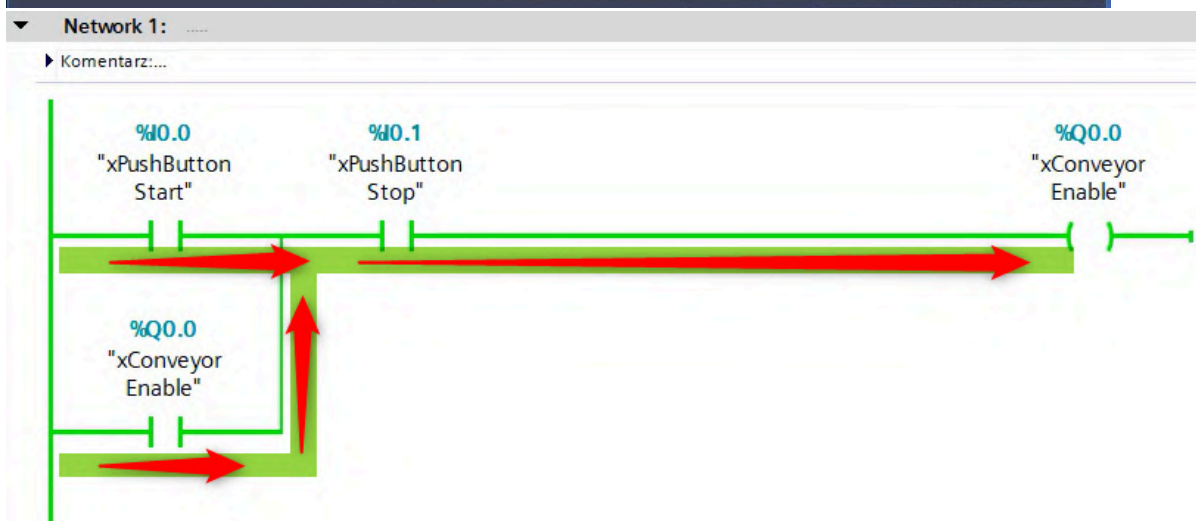
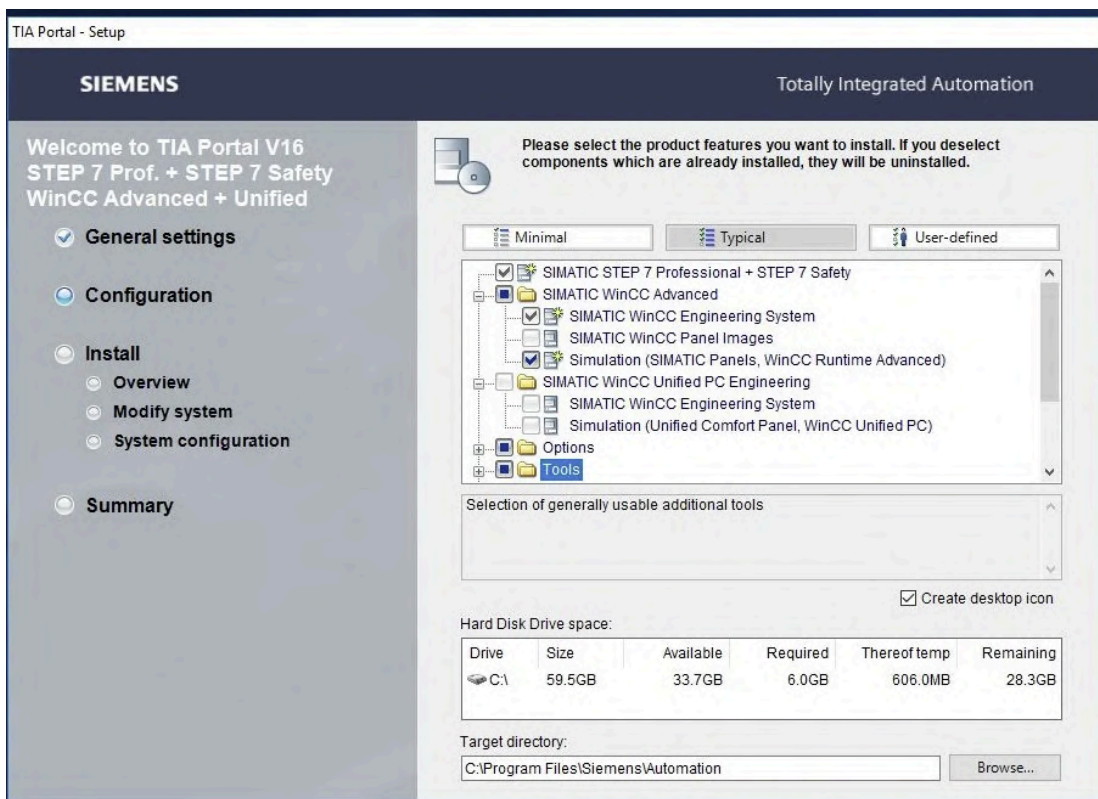
LUB sprzężenia zwrotnego z cewki wyjściowej („xConveyor Enable”), aby przenośnik pozostał włączony.



Jednocześnie fizycznie normalnie zamknięty przycisk zatrzymania („xPushButtonStop”) wprowadza warunek AND, w którym naciśnięcie go przerywa pętlę sprzężenia zwrotnego, zatrzymując przenośnik. Tworząc rozgałęzienia, Ladder pozwala na jednoczesną kontrolę wielu warunków, symulując zachowanie cyfrowych bramek logicznych. Można na przykład zwiększyć złożoność za pomocą dodatkowych wejść, aby utworzyć operację XOR, w której jedno lub drugie wejście, ale nie oba, aktywuje wyjście.

Każda gałąź w programie funkcjonuje jako ścieżka, przez którą musi przepływać prąd (lub sygnał logiczny), a prąd może przechodzić przez jeden lub więcej warunków w zależności od struktury logicznej.

Tutaj mamy przycisk Start zasilający cewkę



Tutaj zmienna xConveyorEnable, przypisana do styku NO równoległe do przycisku Start, zasila cewkę. Jest to tak zwany obwód zatraskowy.

TIA Portal - Setup

SIEMENS Totally Integrated Automation

Welcome to TIA Portal V16
STEP 7 Prof. + STEP 7 Safety
WinCC Advanced + Unified

- ☒ General settings
- ☐ Configuration
- ☐ Install
 - ☐ Overview
 - ☐ Modify system
 - ☐ System configuration
- ☐ Summary

Please select the product features you want to install. If you deselect components which are already installed, they will be uninstalled.

Minimal Typical User-defined

- ☒ SIMATIC STEP 7 Professional + STEP 7 Safety
 - ☒ SIMATIC WinCC Advanced
 - ☒ SIMATIC WinCC Engineering System
 - ☒ SIMATIC WinCC Panel Images
 - ☒ Simulation (SIMATIC Panels, WinCC Runtime Advanced)
 - ☒ SIMATIC WinCC Unified PC Engineering
 - ☒ SIMATIC WinCC Engineering System
 - ☒ Simulation (Unified Comfort Panel, WinCC Unified PC)
- ☐ Options
- ☐ Tools

Selection of generally usable additional tools

☒ Create desktop icon

Hard Disk Drive space:

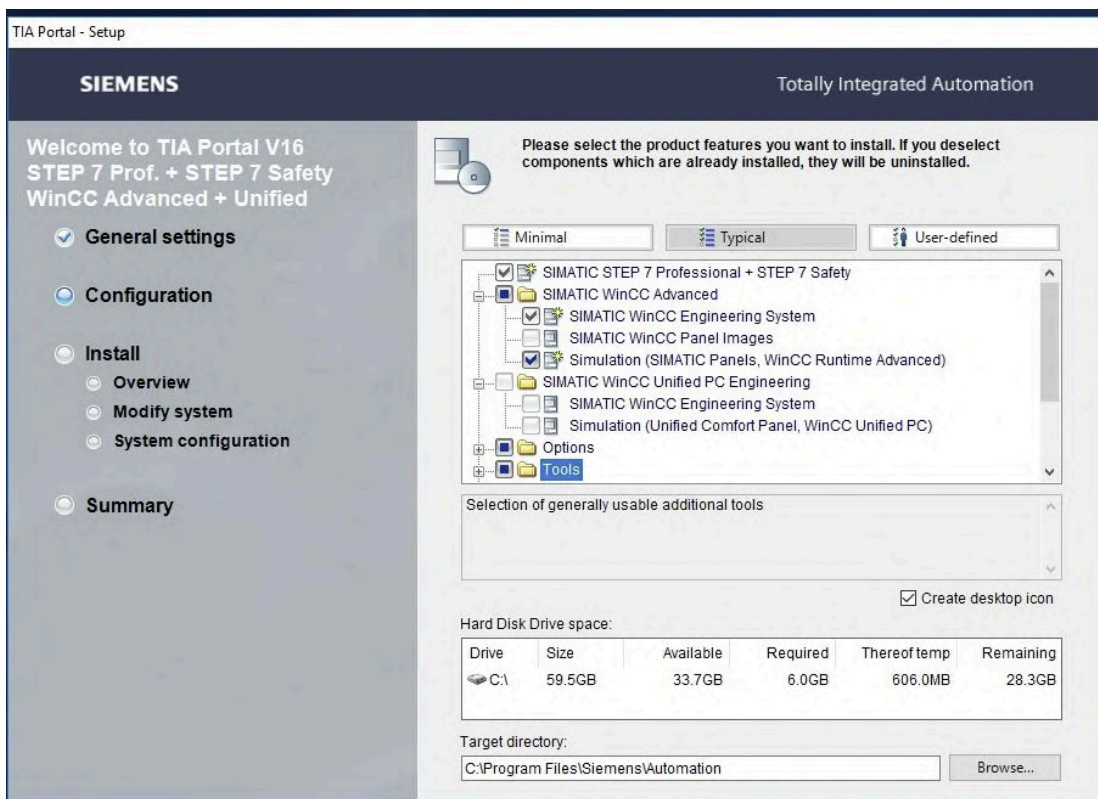
Drive	Size	Available	Required	Thereof temp	Remaining
C:\	59.5GB	33.7GB	6.0GB	606.0MB	28.3GB

Target directory:
C:\Program Files\Siemens\Automation Browse...

Network 1:

Komentarz:...

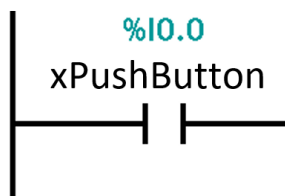
Jeśli chcesz dowiedzieć się więcej, obejrzyj filmy przedstawiające działanie tego systemu:
<https://www.youtube.com/live/R67zQFAGzbM?si=EsW2NLnxmux6JGqX>



4.3 Podstawowe symbole języka Ladder

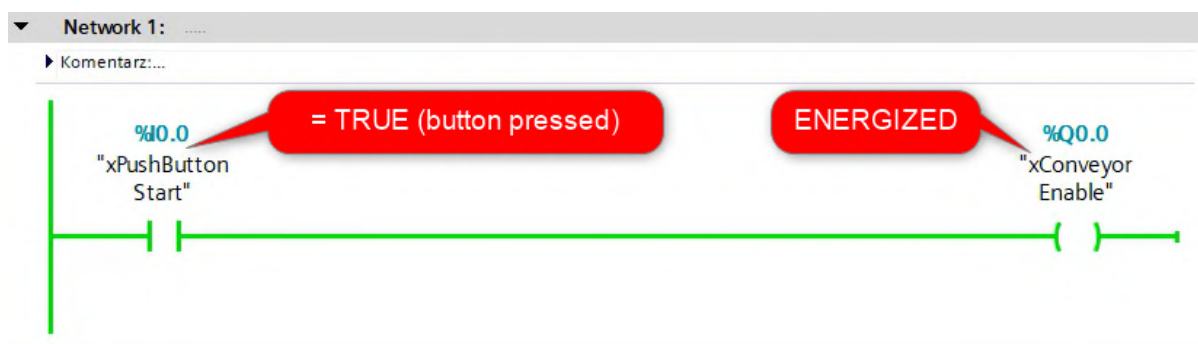
Zestyk NO

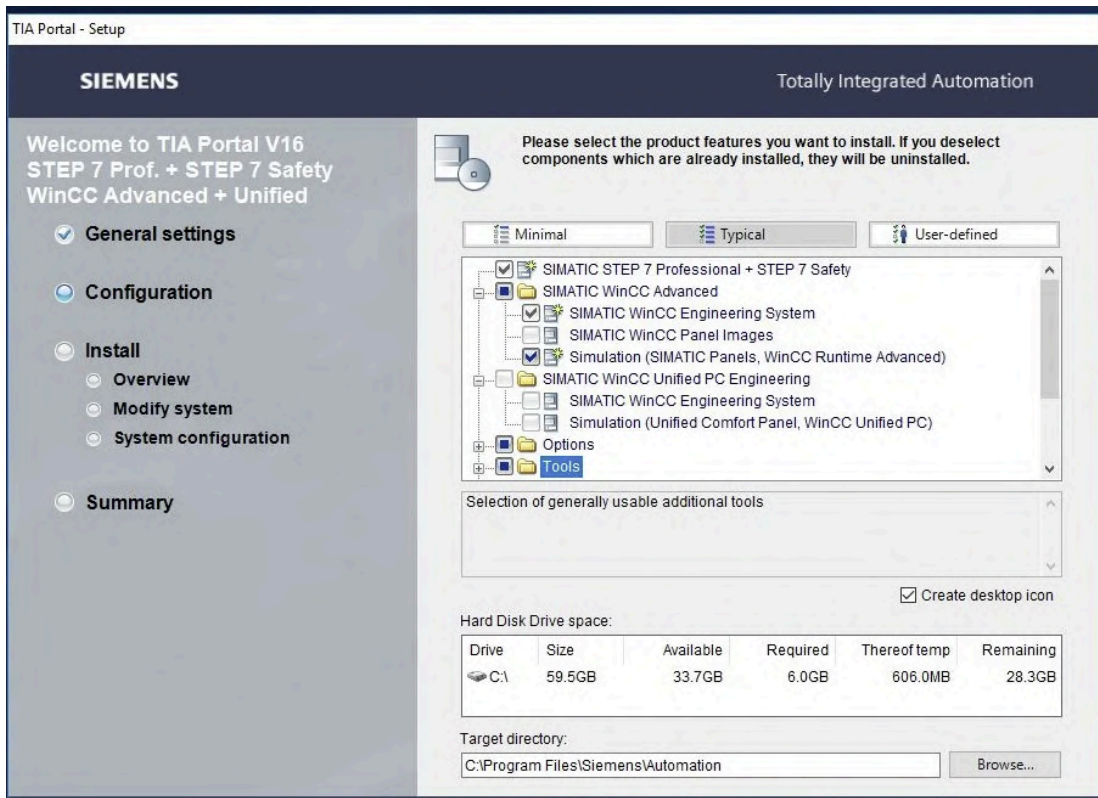
Zestyk NO - normalnie otwarty



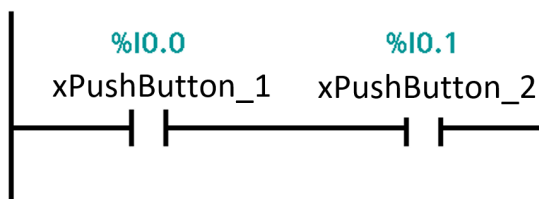
Normalnie Otwarty styk jest **zamknięty** (przewodzi sygnał), gdy operand ma stan sygnału "1" - %I0.0 ma stan sygnału **PRAWDA** / 1.

Normalnie Otwarty styk jest **otwarty** (nie przewodzi sygnału), gdy operand ma stan sygnału "0" - %I0.0 ma stan sygnału **FAŁSZ** / 0.

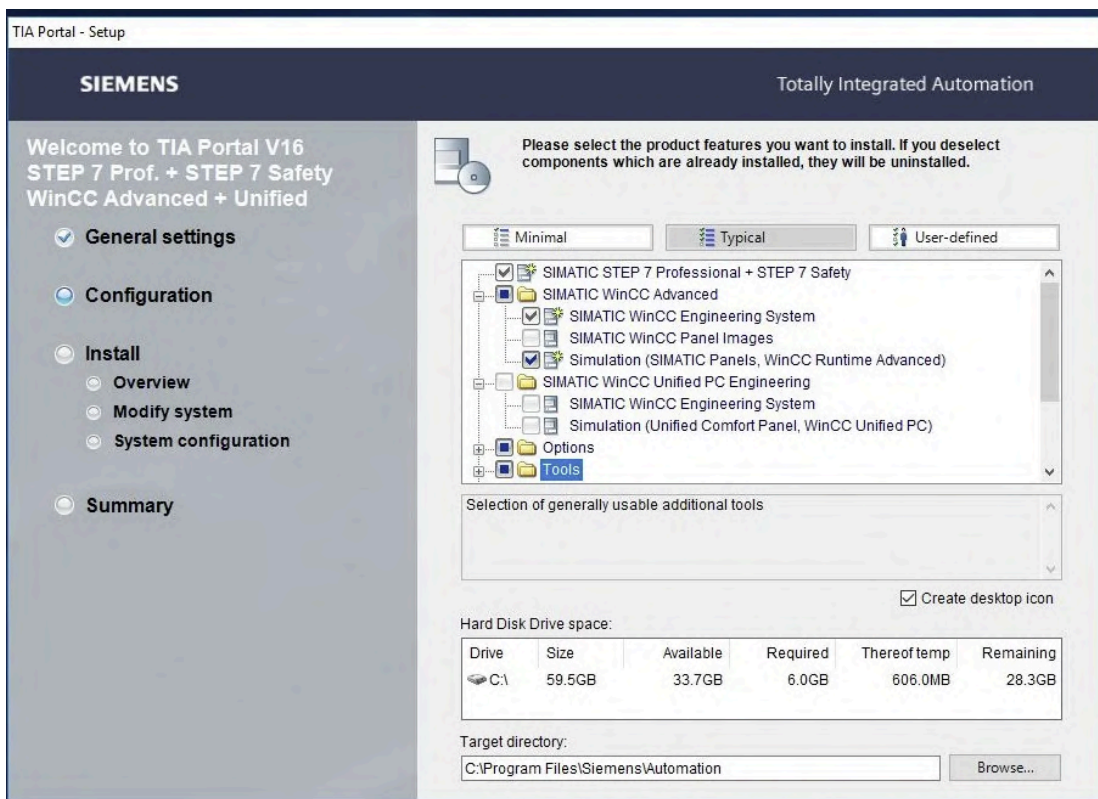




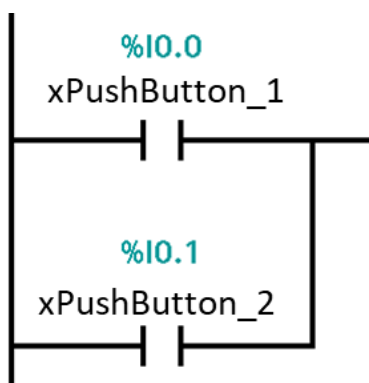
Bramka logiczna AND



I0.0	I0.1	Output
0	0	FALSE
0	1	FALSE
1	0	FALSE
1	1	TRUE



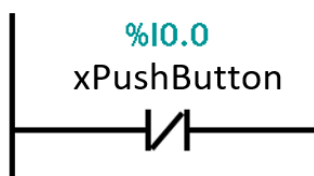
Bramka logiczna OR



IO.0	IO.1	Output
0	0	FALSE
0	1	TRUE
1	0	TRUE
1	1	TRUE

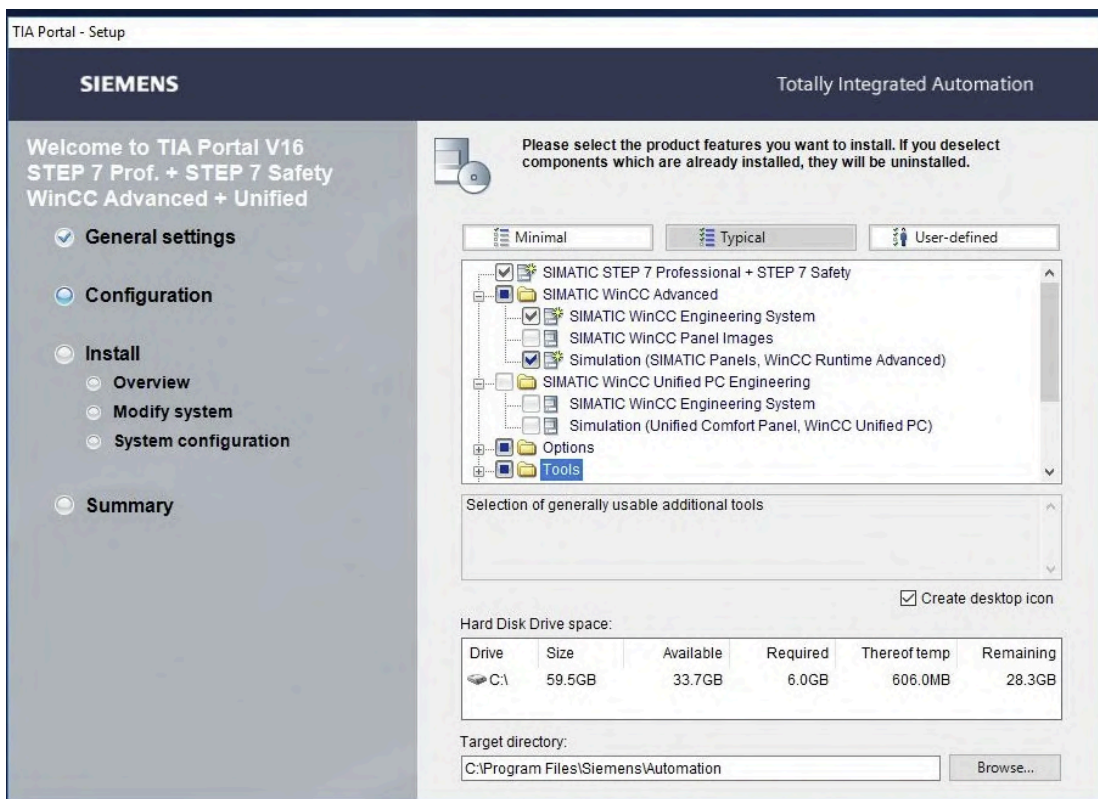
Zestyk NC

Zestyk NC - normalnie zamknięty

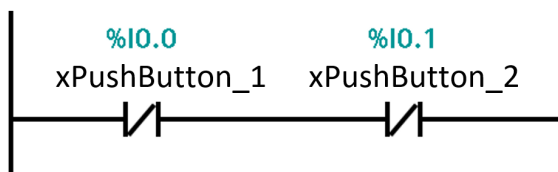


Normalnie Zamknięty styk jest **zamknięty** (przewodzi sygnał), gdy operand ma stan sygnału "0" - %I0.0 ma stan sygnału **FAŁSZ** / 0.

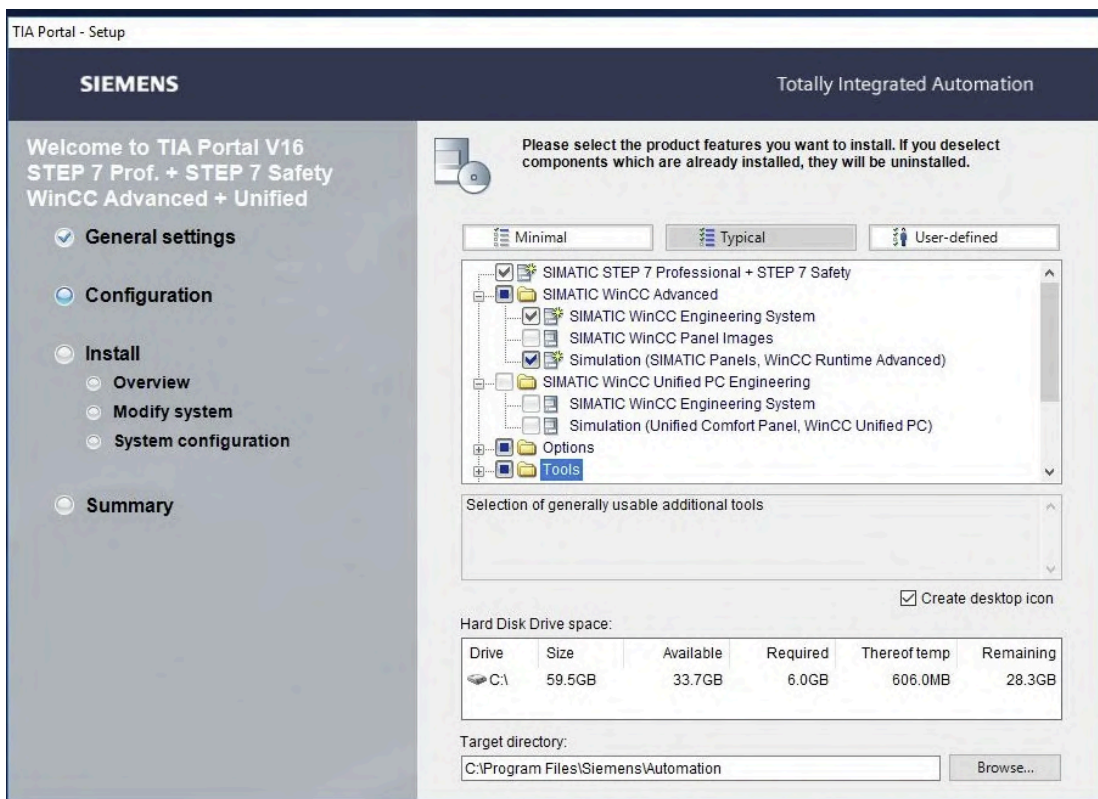
Normalnie Zamknięty styk jest **otwarty** (nie przewodzi sygnału), gdy operand ma stan sygnału "1" - %I0.0 ma stan sygnału **PRAWDA** / 1.



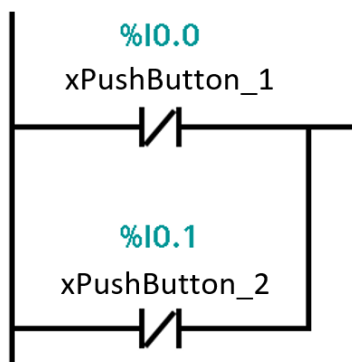
Bramka logiczna NOR



IO.0	IO.1	Output
0	0	TRUE
0	1	FALSE
1	0	FALSE
1	1	FALSE

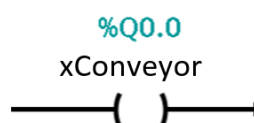


Bramka logiczna NAND



I0.0	I0.1	Output
0	0	TRUE
0	1	TRUE
1	0	TRUE
1	1	FALSE

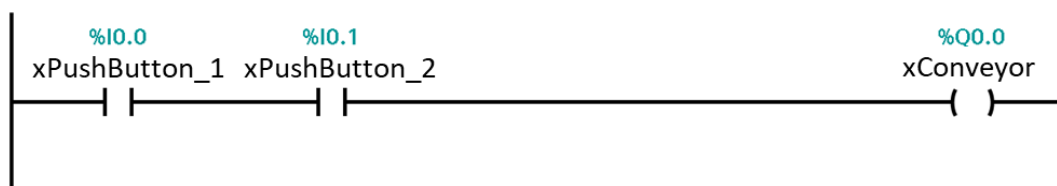
Przypisanie wartości - cewka

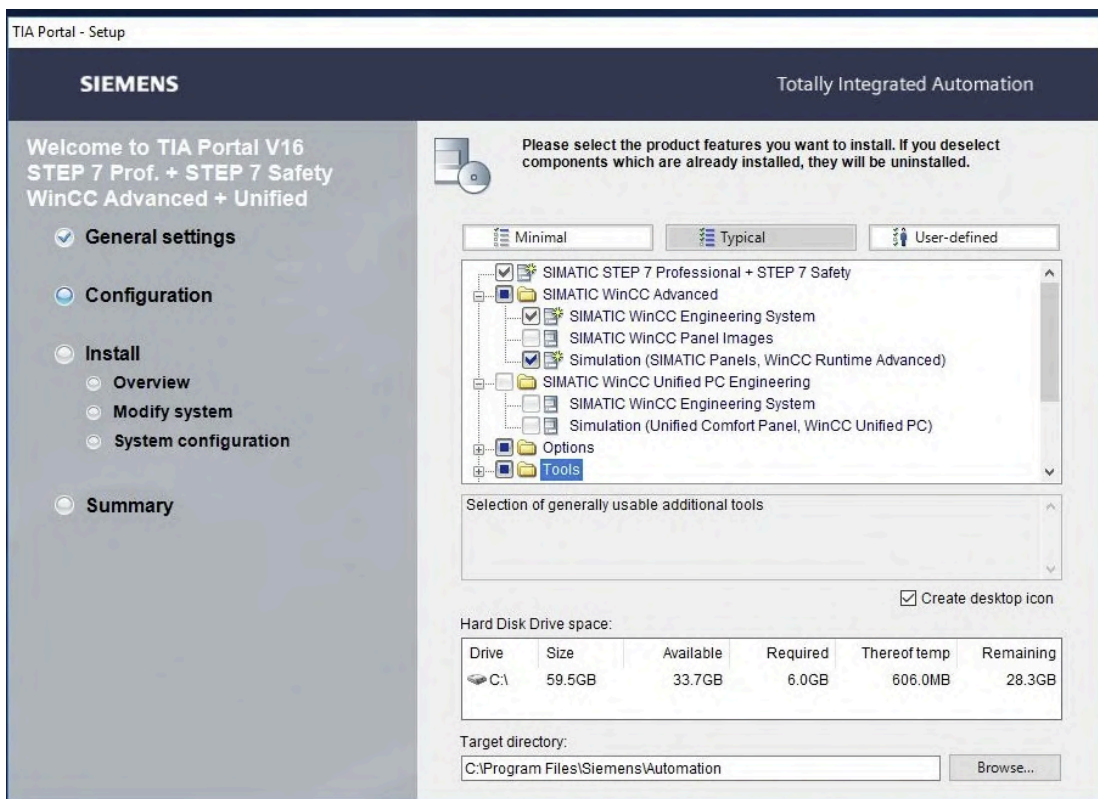


Cewka – instrukcja zapisuje wartość dla określonego operandu.

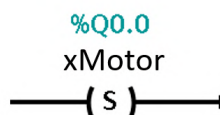
Ustawia **PRAWDA / 1**, gdy stan przed instrukcją w gałęzi (RLO) jest **PRAWDA / 1**.

Ustawia **FAŁSZ / 0**, gdy stan przed instrukcją w gałęzi (RLO) jest **FAŁSZ / 0**.

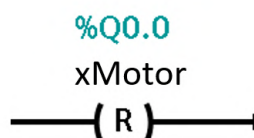




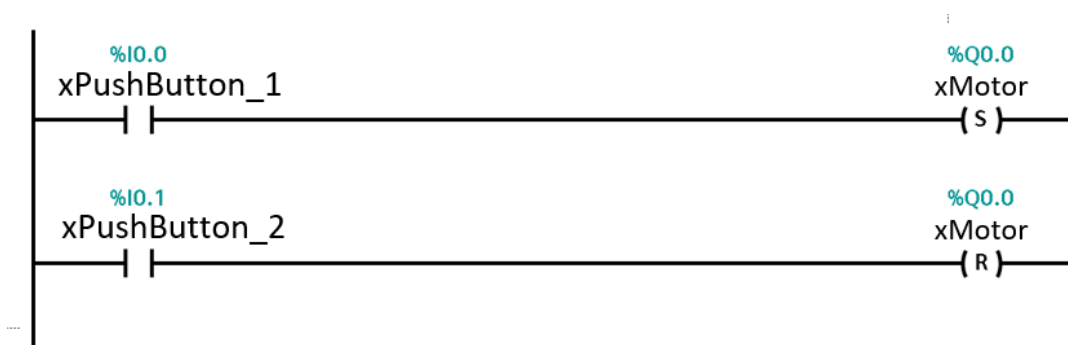
Instrukcje bitowe SET/RESET

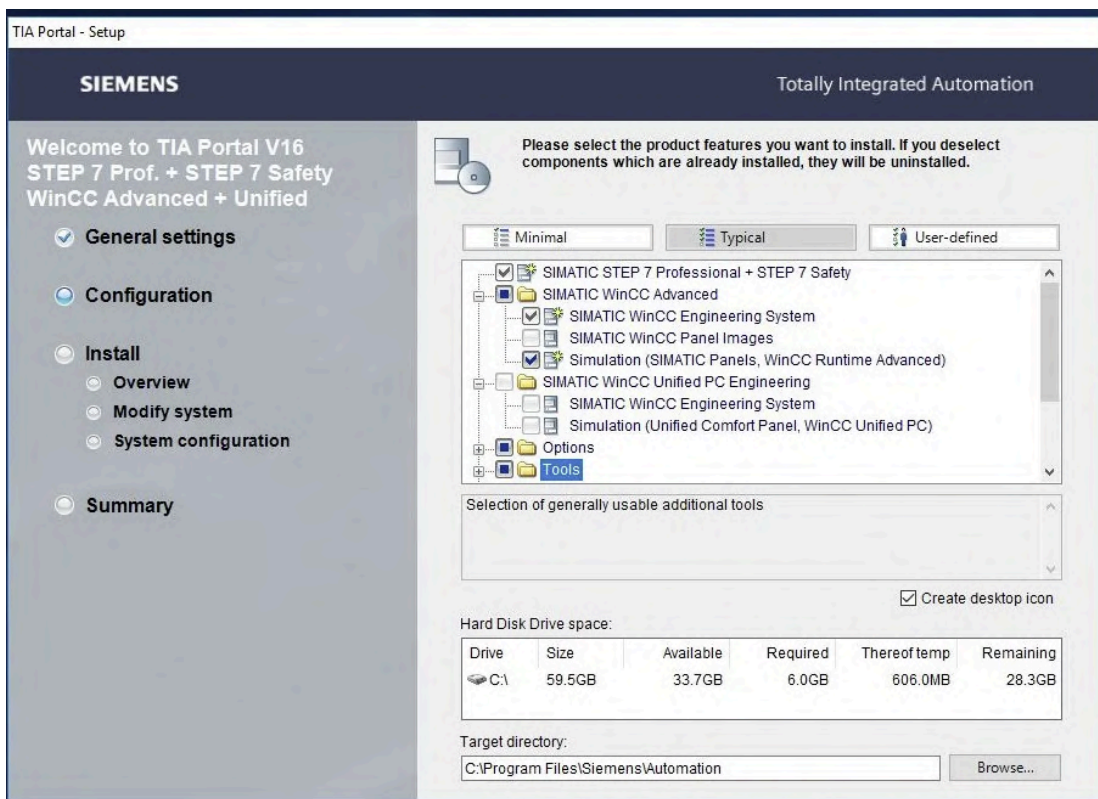


Jeśli cewka **S (SET)** jest aktywna, ustawia stan przypisanej zmiennej na **PRAWDA / 1**.
Jeśli cewka **S (SET)** nie jest aktywna, stan przypisanej zmiennej nie zostaje zmieniony przez tę instrukcję.



Jeśli cewka **R (RESET)** jest aktywna, ustawia stan przypisanej zmiennej na **FALSZ / 0**.
Jeśli cewka **R (RESET)** nie jest aktywna, stan przypisanej zmiennej nie zostaje zmieniony przez tę instrukcję.





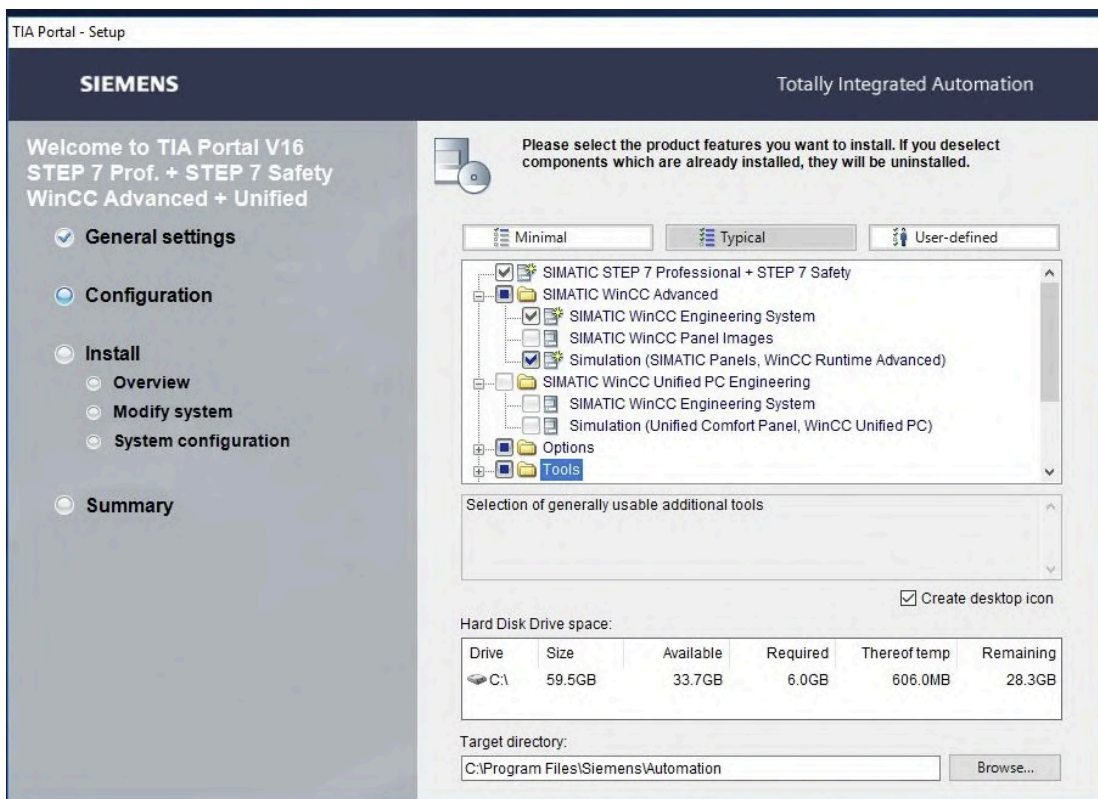
IO.0	IO.1	Q0.0
SET	RESET	OUT
0	0	Previous state
0	1	FALSE
1	0	TRUE
1	1	FALSE

4.4 Zmienne i typy zmiennych

W programowaniu sterowników PLC **zmienne** i **stałe** są niezbędne do zarządzania sygnałami i danymi. **Zmienna** to nazwana lokalizacja w pamięci, której wartość może ulec zmianie podczas wykonywania programu. Pozwala ona systemowi dynamicznie reagować na dane wejściowe, takie jak dane z czujników lub działania użytkownika. Z kolei **stała** również ma adres pamięci i przechowuje dane, ale jej wartość pozostaje stała przez cały czas działania programu, zapewniając spójność krytycznych ustawień, takich jak limity lub wartości zadane.

Kluczowe cechy zmienne/stałe:

- Przechowuje dane
- Ma nazwę
- Ma typ
- Ma adres w pamięci procesora
- Zmienna - zmienia wartość podczas wykonywania programu
- Stała – **nie zmienia** wartości podczas wykonywania programu

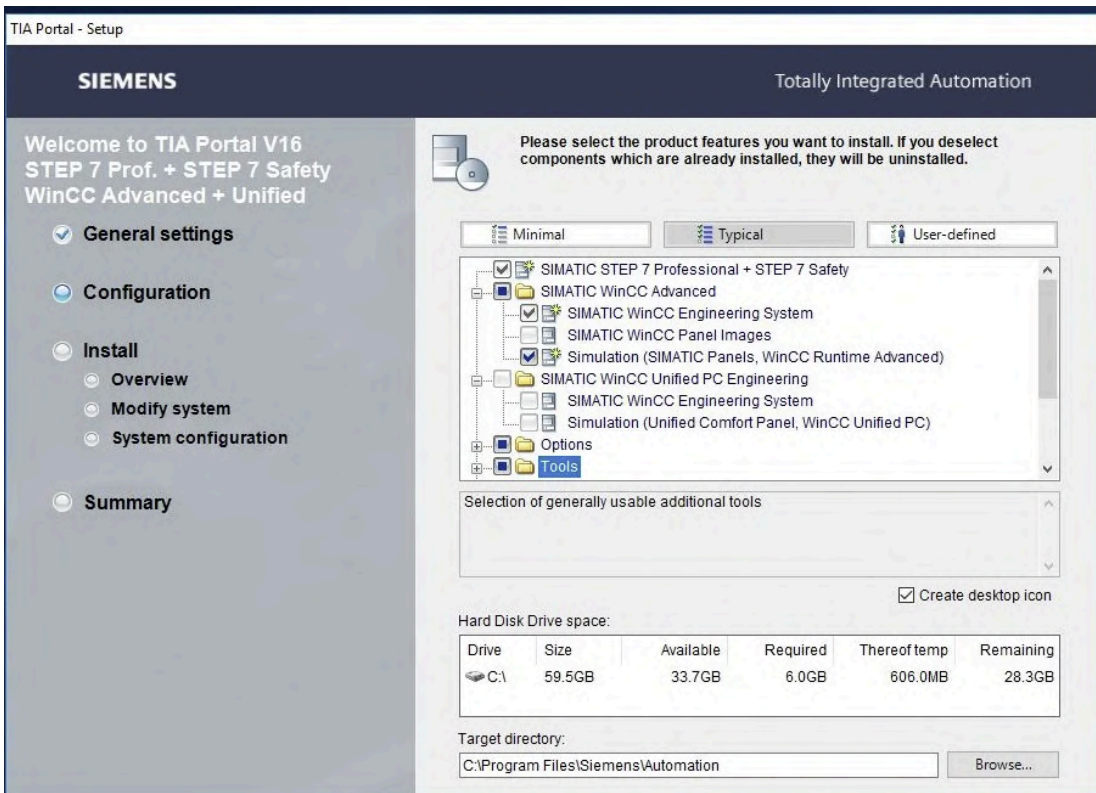


Standard-Variablen-tabelle								
	Name	Data type	Address	Retain	Access...	Write...	Visible...	Comment
1	xConveyor_1	Bool	%Q0.0	☐	☒	☒	☒	
2	xPushButton_1	Bool	%I0.0	☐	☒	☒	☒	
3	xPushButton_2	Bool	%I0.1	☐	☒	☒	☒	
4	<Add new>			☐	☒	☒	☒	

Standard-Variablen-tabelle				
	Name	Data type	Value	Comment
1	iMAX_PRESSURE	Int	250	[bar]

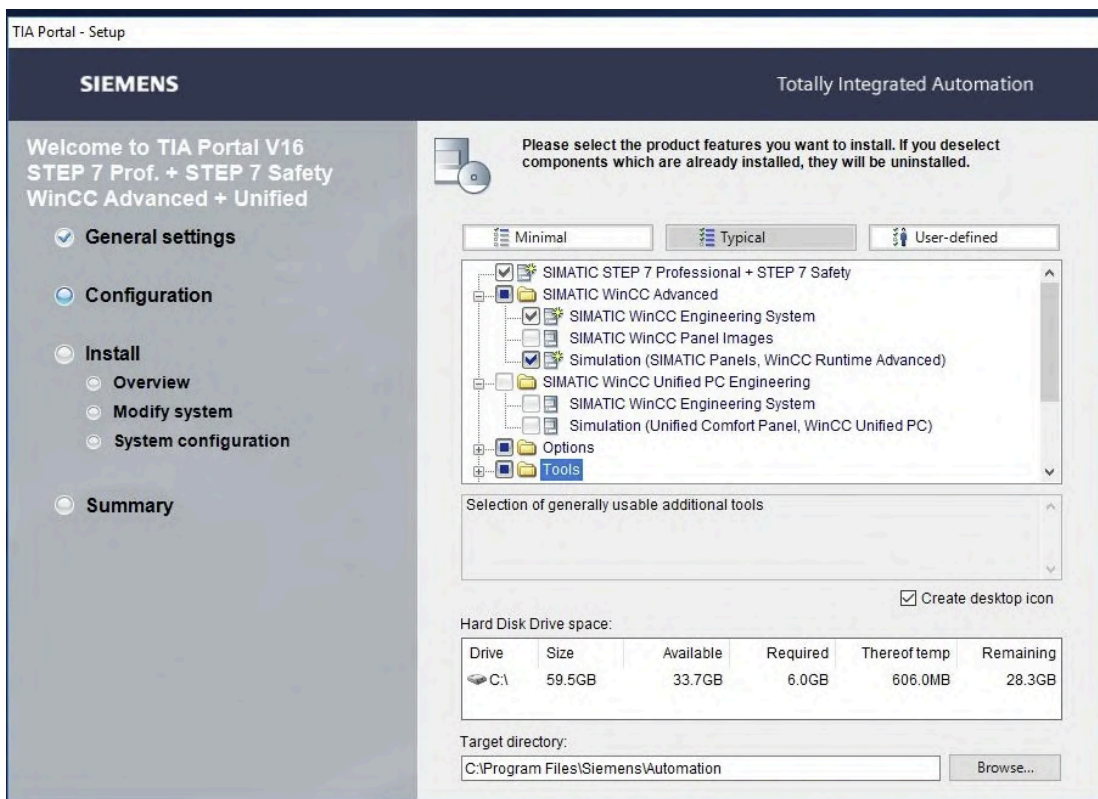
Podstawowe typy zmiennych w TIA Portal

Nazwa	Pre fix	Typ	Rozmiar		Zakres	Przykład
			bits	bytes		
Bool	x	logic	1	–	0, 1	TRUE
Byte	b	Bit string	8	1	-	16#F
Word	w	Bit string	16	2	-	16#F0F0
DWord	dw	Bit string	32	4	-	16#FFFFFFFF
USInt	usi	integer	8	1	0... 255	78



Nazwa	Pre fix	Typ	Rozmiar		Zakres	Przykład
			bits	byt es		
UInt	ui	integer	16	2	0... 65 535	65265
UDInt	udi	integer	32	4	0... 4294967295	24891924
SInt	si	integer	8	1	-128... 127	-50
Int	i	integer	16	2	-32 768... 32 767	23562
DInt	di	integer	32	4	-2 147 483 648...2 147 483,647	-2131754992
Real	r	Floating point	32	4	-3.402823e+38... -1.175 495e-38, ±0, +1.175 495e-38...+3.402823e+38	123.456 -3.4
Char	ch	Character	8	1	ASCII table	'H'
String	s	Character string		256	0...254	'Example string'
Data	date		16	2	D#1990-1-1...D#2168-12-31	D#2014-02-10
Time	tim		32	4	T#-24d_20h_31m_23s_648 ms... T#24d_20h_31m_23s_647m s	T#5m_30s T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630 ms

Nazwa	Pre fix	Typ	Rozmiar		Zakres	Przykład
			bits	byt es		
Time_of _day	tod		32	4	TOD#0:0:0.0 to TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400
DTL	dt		12		Min.: DTL#1970-01-01-00:00:00.0 Max.: DTL#2554-12-31-23:59:59.99 999 999	DTL#2008-12-16-20:30:20.250
Void		empty	0	0		



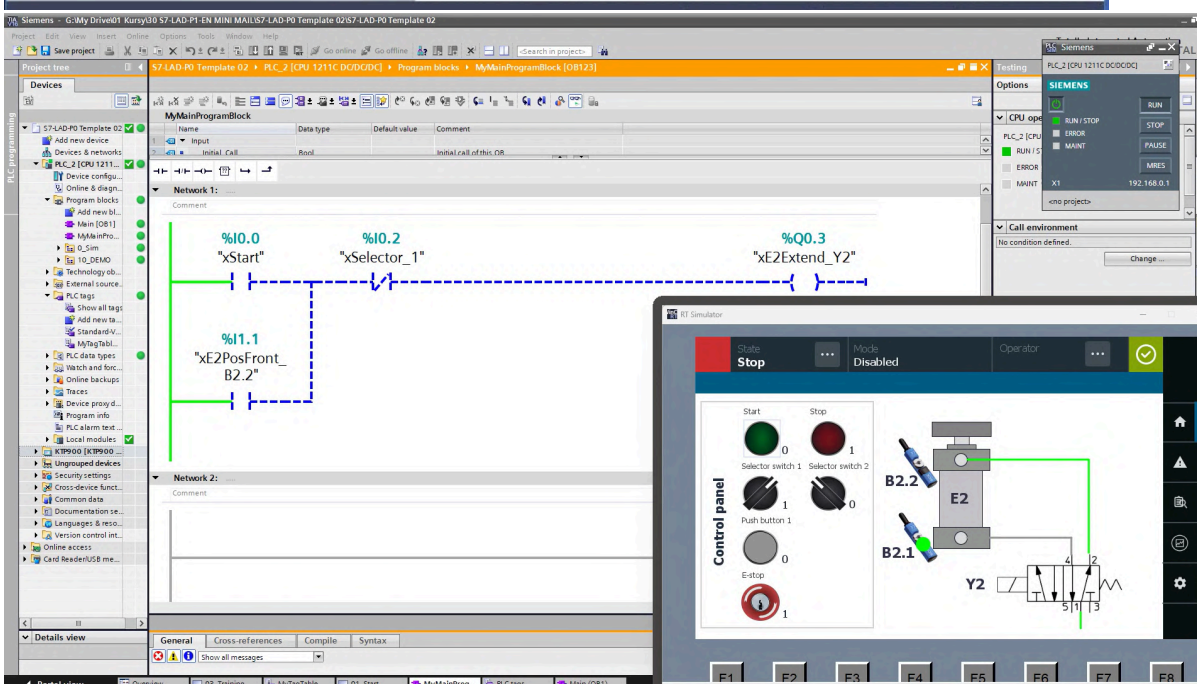
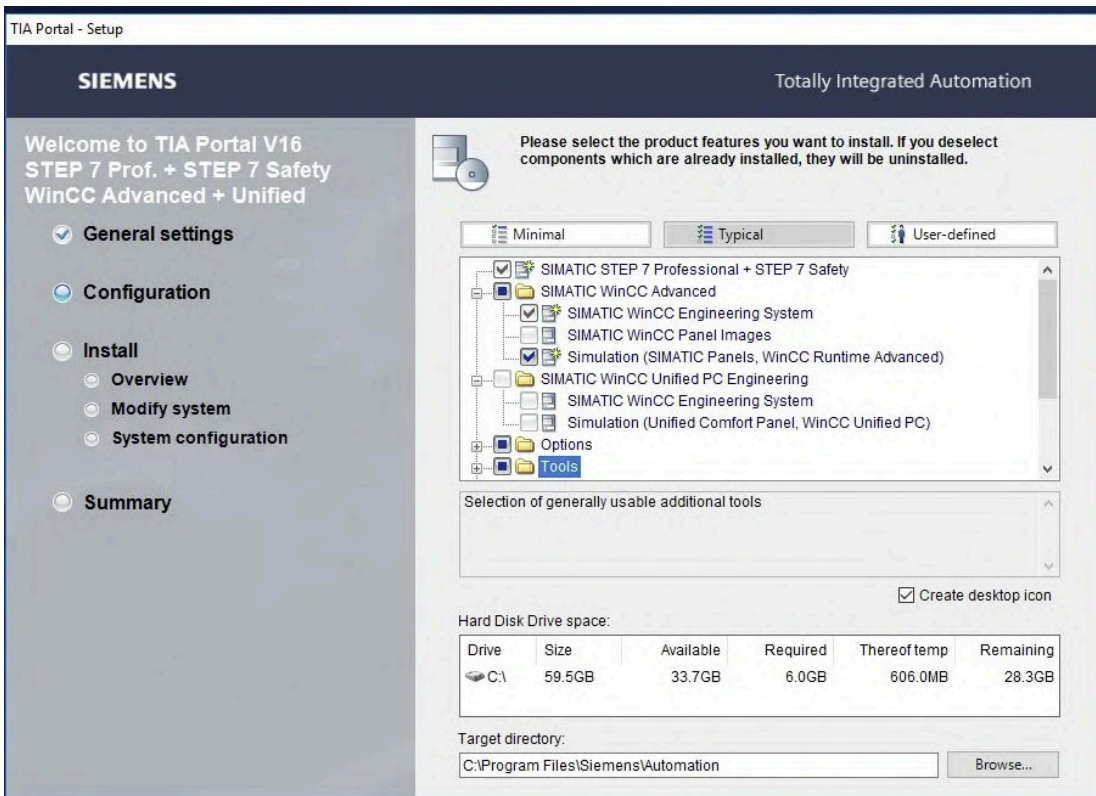
ROZDZIAŁ 5 | PROJEKTY I PYTANIA DOTYCZĄCE ĆWICZEŃ ZWIĄZANYCH Z ROZMOWĄ KWALIFIKACYJNĄ

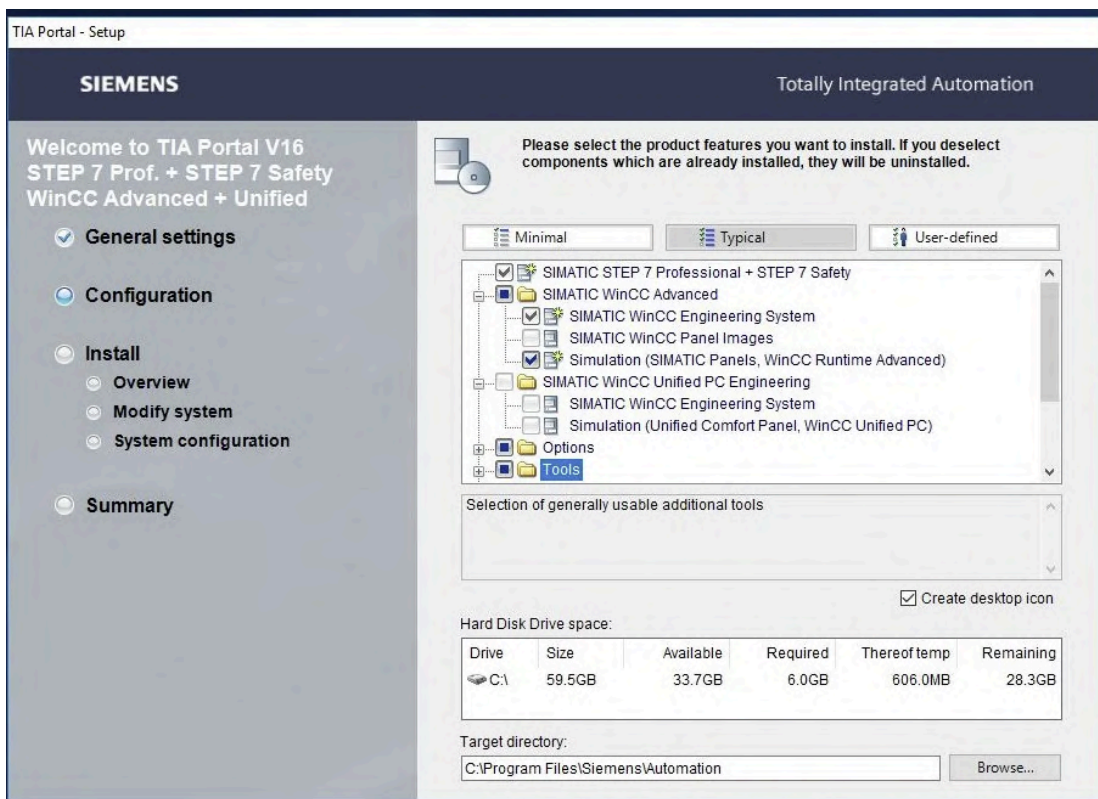
Poniżej znajdziesz najczęściej zadawane pytania na rozmowie kwalifikacyjnej. Pamiętaj, że Twój rozmówca nie ma całego dnia na zaprogramowanie linii produkcyjnej. Będzie to najwyżej 10-minutowe zadanie!

1. Jak zaprogramować obwód sterowania silnikiem start/stop przy użyciu języka LADDER? Przejdź przez kroki programowania podstawowego obwodu sterowania silnikiem przy użyciu LAD.
2. Jak zaprogramować siłownik pneumatyczny z 2 czujnikami i zaworem 5/2? Wyjaśnij kroki sterowania ruchem siłownika pneumatycznego za pomocą czujników i zaworu w LAD lub innym języku programowania PLC.
3. Jak zaprogramować system sterowania zbiornikiem wody z 2 przełącznikami poziomu i pompą? Opisz, w jaki sposób użyłbyś LAD lub innej metody programowania do zarządzania poziomami wody za pomocą przełączników do sterowania pompą.
4. Opis różnych typów zmiennych używanych w TIA Portal. Podaj szczegółowe informacje na temat typów zmiennych dostępnych w Siemens TIA Portal, takich jak zmienne globalne, lokalne i instancje, oraz ich zastosowań.
5. Jakie bloki programowe są dostępne w TIA Portal? Wyjaśnij różne bloki programowe (takie jak OB, FB, FC, DB) używane w Siemens TIA Portal i ich funkcje.

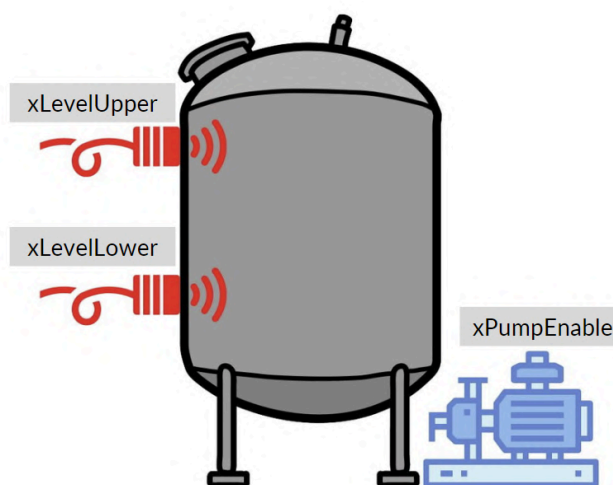
5.1 Jak zaprogramować obwód sterowania silnikiem start/stop przy użyciu LADDER?

Rozwiązanie:





5.3 Jak zaprogramować system sterowania zbiornikiem wody z 2 przełącznikami poziomu i pompą?



W tym kodzie:

- **xLevelLower** to czujnik dolnego poziomu.
- **xLevelUpper** to czujnik górnego poziomu.
- **xPumpEnable** steruje pompą. Jeśli poziom cieczy spadnie poniżej dolnego czujnika, pompa zostanie włączona (**TRUE**). Jeśli poziom cieczy osiągnie górny czujnik, pompa zostanie wyłączona (**FALSE**).

