

POLITECHNIKA POZNAŃSKA

INSTYTUT ROBOTYKI I INTELIGENCJI MASZYNOWEJ

ZAKŁAD STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ



PROGRAMOWANIE Z WYKORZYSTANIEM
PRZEKAŹNIKÓW LICZNIKOWYCH ORAZ ORGANIZACJA
PAMIĘCI

PROGRAMOWANIE STEROWNIKÓW PLC I REGULATORÓW
PRZEMYSŁOWYCH

MATERIAŁY DO ZAJĘĆ LABORATORYJNYCH

MGR INŻ. PRZEMYSŁAW SIWEK

PRZEMYSŁAW.SIWEK@PUT.POZNAN.PL

I. CEL

Celem zajęć jest zapoznanie się i praktyczne wykorzystanie bloków liczników w sterownikach Simatic S7-1200 wraz z dodatkowymi zagadnieniami związanymi z organizacją pamięci oraz typami zmiennych.

II. PRZYGOTOWANIE DO ZAJĘĆ

a) ZAPOZNANIE Z PRZEPISAMI BHP

Wszystkie informacje dotyczące instrukcji BHP laboratorium są zamieszczone w sali laboratoryjnej oraz u prowadzącego zajęcia. Wszystkie nieścisłości należy wyjaśnić z prowadzącym laboratorium. Wymagane jest zaznajomienie i zastosowanie do regulaminu.

Na zajęcia należy przyjść przygotowanym zgodnie z tematem zajęć. Obowiązuje również materiał ze wszystkich poprzednich zajęć.

b) WPROWADZENIE

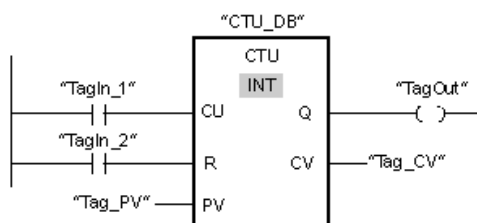
Liczniki w sterownikach PLC

Sterownik S7-1200 wyposażony jest w trzy bloki liczników impulsów. Różnią się one kierunkiem zliczania.

CTU

Tabela 1. Rejestry oraz typy zmiennych wykorzystywane przez CTU.

Parameter	Declaration	Data type	Memory area	Description
CU	Input	BOOL	I, Q, M, D, L or constant	Count input
R	Input	BOOL	I, Q, M, T, C, D, L, P or constant	Reset input
PV	Input	Integers	I, Q, M, D, L, P or constant	Value at which the output Q is set.
Q	Output	BOOL	I, Q, M, D, L	Counter status
CV	Output	Integers, CHAR, DATE	I, Q, M, D, L, P	Current counter value



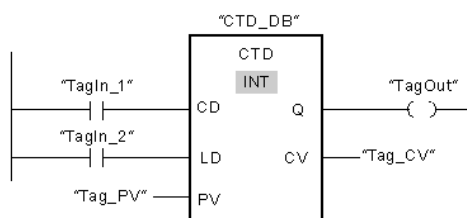
Rys. 1. Wejścia i wyjścia licznika CTU.

Gdy na wejściu *CU* wykryte zostanie zbocze narastające licznik inkrementuje o 1 rejestr *CV*. Wartość przypisana do wejścia *PV* określa moment zmiany stanu wyjścia *Q*. Jeżeli *CV* jest większe lub równe *PV* to na *Q* wystawione zostanie 1. Maksymalna wartość zliczania to 32767 impulsów.

CTD

Tabela 2. Rejestry oraz typy zmiennych wykorzystywane przez CTD.

Parameter	Declaration	Data type	Memory area	Description
CD	Input	BOOL	I, Q, M, D, L or constant	Count input
LD	Input	BOOL	I, Q, M, T, C, D, L, P or constant	Load input
PV	Input	Integers	I, Q, M, D, L, P or constant	Value at which the output Q is set.
Q	Output	BOOL	I, Q, M, D, L	Counter status
CV	Output	Integers, CHAR, DATE	I, Q, M, D, L, P	Current counter value



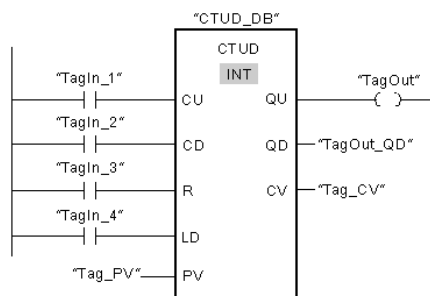
Rys. 2. Wejścia i wyjścia licznika CTD.

Gdy na wejściu *CD* wykryte zostanie zbocze narastające licznik dekrementuje o 1 rejestr *CV*. Wartość przypisana do wejścia *PV* określa moment zmiany stanu wyjścia *Q*. Jeżeli *CV* jest mniejsze lub równe *PV* to na *Q* wystawione zostanie 1. Minimalna wartość zliczania to -32768.

CTUD

Tabela 3. Rejestry oraz typy zmiennych wykorzystywane przez CTUD.

Parameter	Declaration	Data type	Memory area	Description
CU	Input	BOOL	I, Q, M, D, L or constant	Count up input
CD	Input	BOOL	I, Q, M, D, L or constant	Count down input
R	Input	BOOL	I, Q, M, T, C, D, L, P or constant	Reset input
LD	Input	BOOL	I, Q, M, T, C, D, L, P or constant	Load input
PV	Input	Integers	I, Q, M, D, L, P or constant	Value at which the output QU is set.
QU	Output	BOOL	I, Q, M, D, L	Status of the counter up
QD	Output	BOOL	I, Q, M, D, L	Status of the down-counter
CV	Output	Integers, CHAR, DATE	I, Q, M, D, L, P	Current counter value



Rys. 3. Wejścia i wyjścia licznika CTUD.

Licznik CTUD łączy funkcjonalności CTU i CTD. Rejestr *CV* zlicza zbocza narastające na wejściu *CU* (wartość rejestru *CV* rośnie) oraz na wejściu *CD* (wartość rejestru *CV* maleje). *QU* przyjmuje 1, gdy *CV* jest większa lub równa *PV*, natomiast *QD*, gdy *CV* jest mniejsze lub równe *PV*. Licznik może zliczać od -32768 do 32767.

Bloki organizacyjne (OB)

W sterowniku S7-1200 program użytkownika wykonywany jest w blokach organizacyjnych OB. Każdy blok musi mieć unikalny numer. Część numerów poniżej 123 jest już domyślnie zarezerwowana przez sterownik. OB wywoływane są przez określone zdarzenia w CPU takie jak przerwanie diagnostyczne, start programu lub interwał czasowy. Należy pamiętać o tym, że bloki organizacyjne nie mogą być wywoływane przez inne bloki lub funkcje. CPU obsługuje OB zgodnie z ich priorytetem – 1 priorytet najniższy, 28 – priorytet najwyższy (przerwania diagnostyczne).

Typy bloków:

- 1) OB cyklu programu - domyślny blok programu głównego.
- 2) OB startowy – wykonywany jednorazowo po zmianie trybu CPU z STOP na RUN. Domyślny numer bloku startowego to 100, przy utworzeniu większej ich ilości należy deklarować bloki o numerze 123 lub większym
- 3) OB opóźnienia – wykonywany jednorazowo po określonym czasie od startu programu. Interwał ten definiuje wartość wpisana do *SRT_DINT*. OB opóźnienia przerywa program główny. Domyślne numery bloków to 20, 21, 22, 23
- 4) OB cyklicznego przerwania – wywoływany jest cyklicznie co ustalony przez użytkownika czas przerywając działanie programu głównego. Domyślne numery to 30 do 38.

Zmienne i pamięć

Pamięć sterownika S7-1200 podzielona jest na pięć obszarów, które przedstawiono w tabeli 4. Poza wejściami, wyjściami, pamięcią nieulotną *M* i blokami danych definiowanych przez użytkownika znajduje się jeszcze pamięć chwilowa temp. Każdy blok i funkcja programu posiadają osobną pamięć chwilową do której inne bloki lub funkcje nie mają dostępu.

Podstawową jednostką pamięci sterownika jest bajt. Przy definiowaniu zmiennej podaje się numer pierwszego bajtu od którego, w zależności od rozmiaru zmiennej, alokowana jest pamięć. Źle utworzone zmienne mogą się nakładać powodując błędy (np. *MW0* i *MW1*). Zmienne można wyświetlać na różne sposoby (np. *UINT* jako *INT*). Nie należy tego mylić ze zamianą typu. Wszystkie typy zmiennych przedstawiono w tabeli 5.

Sterownik posiada obszary pamięci odnoszące się do wejść i wyjść. Działają one jak bufor zabezpieczając sterownik przed stanami nieustalonymi. Należy tutaj pamiętać, że wyjścia zapisywane są tylko raz w jednym cyklu programu. Istnieje jednak możliwość bezpośredniego odczytu i zapisu danych na wejścia i wyjścia PLC. Dokonuje się tego stosując przyrostek :P do adresu. Rodzaje adresowania bezwzględnego przedstawiono w tabeli 6.

Tabela 4. Obszary pamięci.

Obszar pamięci	Opis	Wymuszony	Trwały
I obraz procesu – wejście I_P (fizyczne wejście)	Skopiowany na początku cyklu programu stan wejść fizycznych	Tak	Nie
	Bezpośredni odczyt wejściowych punktów fizycznych CPU, SB, SM	Nie	Nie
Q obraz procesu – wyjście Q_P (fizyczne wyjście)	Stan skopiowany na początku cyklu programu do wyjść fizycznych	Tak	Nie
	Bezpośredni zapis do wyjściowych punktów fizycznych CPU, SB, SM	Nie	Nie
M pamięć bitowa	Pamięć sterująca i danych	Nie	Tak
L pamięć chwilowa	Chwilowe dane dla bloku, lokalne dla tego bloku	Nie	Nie
DB blok danych	Pamięć danych oraz parametrów dla bloku funkcji FB	Nie	Tak

Tabela 5. Typy danych.

Typ danej	Przykład adresu	Rozmiar (w bitach)	Zakres	Przykłady wprowadzania stałych
<i>Bool</i>	M1.1	1	Od 0 do 1	TRUE, FALSE, 0, 1
<i>Byte</i>	MB2	8	Od 16#00 do 16#FF	16#12, 16#AB
<i>Word</i>	MW3	16	Od 16#0000 do 16#FFFF	16#ABCD, 16#0001
<i>DWord</i>	MD5	32	Od 16#00000000 do 16#FFFFFFFF	16#02468ACE
<i>Char</i>	MB9	8	Od 16#00 do 16#FF	'A', 't', '@'
<i>Sint</i>	MB10	8	Od -128 do 127	123, -123
<i>Int</i>	MW11	16	Od -32 768 do 32 767	123, -123
<i>Dint</i>	MD13	32	Od -2 147 483 648 do 2 147 483 647	123, -123
<i>USInt</i>	MB17	8	Od 0 do 255	123
<i>UInt</i>	MW18	16	Od 0 do 65,535	123
<i>UDInt</i>	MD20	32	Od 0 do 4 294 967 295	123
<i>Real</i> np. MD5	MD24	32; 8 bitów eksponenty	Od $\pm 1,18 \cdot 10^{-38}$ do $\pm 3,40 \cdot 10^{38}$	12345.6, -3.4, -1.2E+12, 3.4E-3
<i>LReal</i> np. Q2.0	M28.0	64; 11 bitów eksponenty	Od $\pm 1,79 \cdot 10^{-308}$ do $\pm 2,22 \cdot 10^{308}$	12345.6, -3.4, -1.2E+12, 3.4E-3
<i>Time</i>	MD36	32	Od T#24d_20h_31m_23s_648ms do T#24d_20h_31m_23s_647ms pamiętanych jako -2 147 483 648 ms do +2 147 483 647 ms	T#5m_30s, T#-2d T#1d_2h_15m_30s_45ms
<i>String</i>	MB40	Zmienny	Od 0 do 254 znaków po jednym bajcie	'ABC'

Tabela 6. Adresowanie bezwzględne.

Wejścia	bit	I[adres bajtu].[adres bitu]	I0.1
	bajt, słowo lub podwójne słowo	I[rozmiar][adres startowego bajtu]	IB4, IW5 lub ID12
Wejścia fizyczne	bit	I[adres bajtu].[adres bitu]:P	I0.1:P
	bajt, słowo lub podwójne słowo	I[rozmiar][adres startowego bajtu]:P	IB4:P, IW5:P lub ID12:P
Wyjścia	bit	Q[adres bajtu].[adres bitu]	Q1.1
	bajt, słowo lub podwójne słowo	Q[rozmiar][adres startowego bajtu]	QB5, QW10, QD40
Wyjścia fizyczne	bit	Q[adres bajtu].[adres bitu]:P	Q1.1:P
	bajt, słowo lub podwójne słowo	Q[rozmiar][adres startowego bajtu]:P	QB5:P, QW10:P lub QD40:P
Pamięć, znaczniki	bit	M[adres bajtu].[adres bitu]	M26.7
	bajt, słowo lub podwójne słowo	M[rozmiar][adres startowego bajtu]	MB20, MW30, MD50
Bloki danych	bit	DB[numer bloku danych].DBX[adres bajtu].[adres bitu]	DB1.DBX2.3
	bajt, słowo lub podwójne słowo	DB[numer bloku danych].DB[rozmiar][adres startowego bajtu]	DB1.DBB4, DB10.DBW2,DB20.DBD8

Tablice

Czasem program wymaga tabelaryzowania danych w pamięci. W przeciwieństwie do starych sterowników S7-200 w sterownikach S7-1200 można tworzyć tablice danych. Definiuje się je poprzez edytor interfejsu bloków OB, FC, FB i DB (definicja lokalna) lub w bloku danych (*Data block* – definicja globalna). Należy tu zaznaczyć, że nie można tego zrobić poprzez edytor tagów. W celu utworzenia tablicy należy wybrać dane typu *Array[low..high] of type*, a następnie określić początek tablicy *low* i koniec tablicy *high*, które określają wielkość tablicy i zakresy jej indeksowania. Indeksy mogą przyjmować wartości ujemne np. [-10..10]. Do tablicy odwołujemy się wpisując indeks w nawiasie kwadratowym za znakiem # (*tablica[#k]*). Indeks powinien być zdefiniowany jako zmienna pomocnicza typu *INT*.

III. SCENARIUSZ DO ZAJĘĆ

a) ŚRODKI DYDAKTYCZNE

Sprzętowe:

- Komputer PC
- Stanowisko dydaktyczne ze sterownikiem SIMATIC S7-1200

Programowe:

- Środowisko programistyczne: TIA Portal v.12

b) PRZEBIEG ZAJĘĆ

1. Utwórz i wykorzystaj blok startowy (*Startup*) do wprowadzania danych do programu. Zdefiniuj pięć zmiennych różnych typów (*WORD*, *REAL*, *INT*, *UINT* i *DINT*) umieszczonych obok siebie i przypisz im kolejne wartości całkowite (1, 2, ..., 5) komendami *MOVE*. Następnie sprawdź wartości utworzonych zmiennych w zakładce *Watch table* oraz stan pamięci sterownika w zakładce *Watch table* -> *Program info* -> *Assignment list*. Zmodyfikuj adresy w *Tag table* tak, aby dwie ostatnie zmienne (*UINT* i *DINT*) nakładały się pojedynczym bajtem. Ponownie sprawdź wartość zmiennych oraz stan pamięci sterownika.

2. Napisz funkcję tworzącą dzielnik częstotliwości 4:1. W bloku *main* prześlij do funkcji sygnał prostokątny, o częstotliwości 2 Hz, generowany przez dwa timery.
3. Napisz program obsługujący proces pakowania detali do paczek. Proces posiada następujące czujniki i akulatory:
 - Napęd linii przesuwającej paczki – wyjście Q0.0
 - Czujnik położenia paczki pod podajnikami – wejście I0.0
 - Zwolnienie blokady podajnika czerwonych elementów – wyjście Q0.1
 - Zwolnienie blokady podajnika niebieskich elementów – wyjście Q0.2
 - Czujnik zliczający czerwone elementy (fotokomórka) – zmienna M0.0
 - Czujnik zliczający niebieskie elementy (fotokomórka) – zmienna M0.1

Program powinien rozpoczynać pracę od załączenia napędu linii przesuwającej paczki i oczekiwania na pojawienie się stanu wysokiego na I0.0. Symulacja działania czujnika położenia paczek powinna być wykonywana ręcznie przełącznikami stanowiska: stan wysoki – paczka przy czujniku, stan niski – brak paczki. Po pojawieniu się pudełka program powinien załadować 5 czerwonych oraz 20 niebieskich detali. Przedmioty powinny spadać co 0,5 sekundy, gdy załączone są odpowiednie wyjścia (Q0.1 oraz Q0.2). Symulację pojawiających się elementów wykonaj za pomocą generatora sygnału prostokątnego złożonego z dwóch timerów. Po odliczeniu odpowiedniej ilości detali program powinien zamknąć podajniki oraz włączyć napęd linii przesuwającej paczki.

4. Wykorzystując tablicę bitów (*Data block* -> *Array*) napisz funkcję realizującą rejestr przesuwny o dziesięciu jednobajtowych komórkach pamięci. Dane do rejestru będące wartością wejść IB0 powinny być zapisywane w jego pierwszej komórce co 0,5 sekundy. Program powinien przepisywać wartości ostatniej komórki tablicy na wyjścia QB0. Generator częstotliwości zrealizuj za pomocą dwóch timerów, a indeksy tablicy wykonaj na dwóch licznikach.