



**KATEDRA AUTOMATYKI, BIOMECHANIKI
I MECHATRONIKI**



Laboratorium Automatyki

TEMAT:

*Kompensacja nieidealności transmisji przekładni
pasowej w układzie z silnikiem krokowym i
enkerem*

Aktualizacja: 30.05.2025

Opracował: dr inż. Krystian Polczyński

wersja: 2.0

Zadanie: Opracować program do sterowania położeniem wału enkodera napędzanego silnikiem krokowym poprzez przekładnię pasową. Program ma mieć możliwość zadania liczby kroków o ile ma się obrócić wał, a także z jaką prędkością. Enkoder wysyła 200 impulsów na obrót, a silnik krokowy wykonuje 200 kroków na obrót. Zadawanie położenia oraz prędkości odbywać się ma przy pomocy klawiatury funkcyjno-numerycznej HKEY. Obrót wału wykonywać ma się po przyciśnięciu przycisku F6, wciśnięcie przycisku F5 ma natychmiast przerwać działanie silnika. Przycisk F4 ma spowodować dokładne, końcowe spozycjonowanie wału enkodera w przypadku rozbieżności pomiędzy kątem obrotu wału silnika a wału enkodera. Kąt obrotu o jaki ma się obrócić wał enkodera zadawany jest przyciskiem F1 po wcześniejszym ustaleniu liczby kroków przyciskami numerycznymi. W razie błędnie wpisanej liczby kroków wartość ma być resetowana przyciskiem F3. Prędkość obrotu zadawana jest przyciskiem F2.

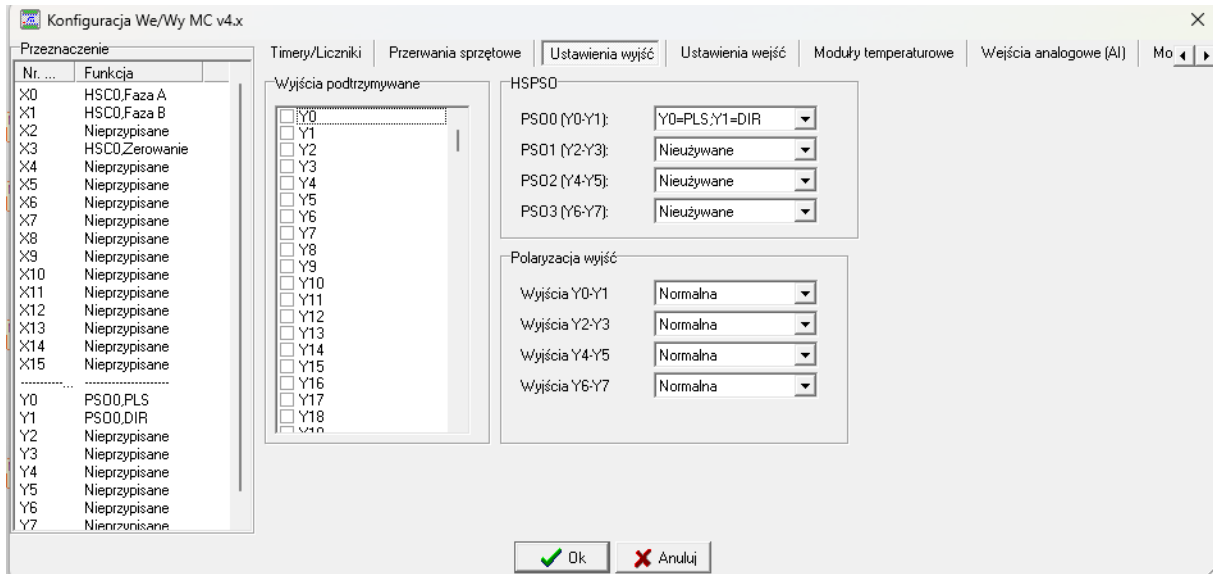
PLAN DZIAŁANIA

- 1) Podłączyć PLC do zasilania oraz połączyć przewodem komunikacyjnym z PC.
- 2) Uruchomić program WinProLadder i utworzyć nowy projekt dla sterownika FBs-24MC
- 3) Dodać sobie zakładkę *Karty Stanu*, w której będziemy przechowywać wartości rejestrów i markerów
- 4) Po lewej stronie w zakładce konfiguracja systemu -> *Konfiguracja we/wy*, wchodzimy w zakładkę *Timery/Liczniki* i wybieramy licznik HSC0. Ustawiamy dla niego tryb pracy – licznik sprzętowy, a tryb zliczania – A/B*4, dzięki temu mamy dokładniejszy odczyt pozycji (z $360/200=1.8$ stopnia robimy $360/800=0.45$ stopnia). Ustawiamy też, że Faza A – X0, Zerowanie – X3.

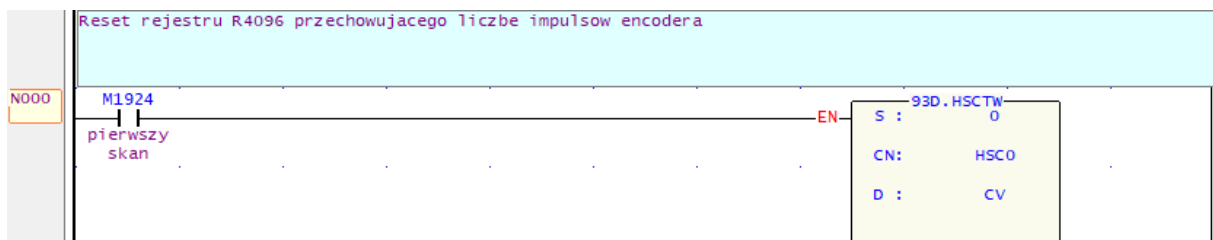
Nr. ...	Funkcja
X0	HSC0,Faza A
X1	HSC0,Faza B
X2	Nieprzypisane
X3	HSC0,Zerowanie
X4	Nieprzypisane
X5	Nieprzypisane
X6	Nieprzypisane
X7	Nieprzypisane
X8	Nieprzypisane
X9	Nieprzypisane
X10	Nieprzypisane
X11	Nieprzypisane
X12	Nieprzypisane
X13	Nieprzypisane
X14	Nieprzypisane
X15	Nieprzypisane
Y0	PS00,PLS
Y1	PS00,DIR
Y2	Nieprzypisane
Y3	Nieprzypisane
Y4	Nieprzypisane
Y5	Nieprzypisane
Y6	Nieprzypisane
Y7	Nieprzypisane

- 5) Teraz podłączamy przewodami wyjścia naszego enkodera do wejść PLC. Wyjście A enkodera łączymy z wejściem X0, a wyjście B z wejściem X1.
- 6) Podłączmy też przewody do wyjść silnika krokowego. Wyjście DIR połączmy z wyjściem Y1 sterownika, natomiast wyjście CK z wyjściem Y0 sterownika.

- 7) Podobnie jak wcześniej, należy teraz skonfigurować wyjścia sterownika Y0 i Y1, jako „szybkie” wyjścia mogące wysyłać sygnały o wysokich częstotliwościach. Robimy to w zakładce *Konfiguracja We/Wy* -> *Ustawienia wyjść*



- 8) W pierwszej linijce programu wstawiamy funkcję FUN93 , w której zadeklarujemy jakie wartości mają mieć rejestry przechowujące dane z enkodera.



Symbol drabinkowy	Argument
	S : Dane źródłowe do zapisu CN : Numer szybkiego licznika sprzętowego do zapisu 0: HSC0 lub HST1 1: HSC1 lub HST2 2: HSC2 lub HST3 3: HSC3 lub HST4 4: HSTA D : Cel zapisu (gdzie: 0 oznacza wartość bieżącą (CV), 1 oznacza wartość zadaną (PV))

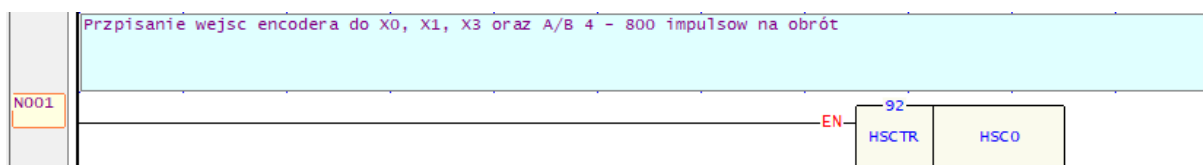
- 9) Ponieważ chcemy, żeby rejestry przechowujące dane enkodera zaczynały zliczanie od wartości 0 po włączeniu programu (uruchomieniu PLC) w N000 dodamy marker M1924, który daje impuls tylko w momencie pierwszego uruchomienia PLC.

3. Sygnały impulsowe		
☐ M1920 ☐ M1921 ☐ M1922 ☐ M1923	Impuls zegarowy 0.01s Impuls zegarowy 0.1s Impuls zegarowy 1s Impuls zegarowy 60s	①
☐ M1924 ☐ M1925 ☐ M1926	Początkowy impuls (pierwszy skan) ② Impulsy zegara skanowania ③ =0, PLC pracuje w trybie STOP =1, PLC pracuje w trybie URUCHOM	
☐ M1927	Status wejścia CTS portu komunikacyjnego 1	• 0 : CTS Prawda (WŁ) • 1 : CTS Fałsz (WYŁ) • Kiedy do połączenia z drukarką lub modem wykorzystywany jest port komunikacyjny 1, do określenia tego, czy drukarka lub modem jest gotowy może posłużyć ten sygnał oraz timer.

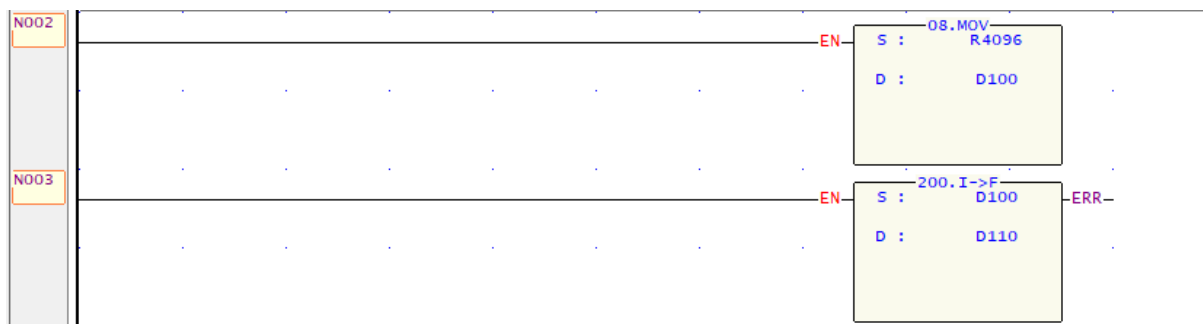
10) Wpiszmy do *Karty Stanu* rejestr R4096 wyświetlający aktualna wartość enkodera, w celu monitorowania jego wartości (czy mierzy kąt obrotu oraz czy został zresetowany).

Nr rejestru	Funkcjonalność
R4096	Aktualna wartość młodsze słowa HSC0
R4097	Aktualna wartość starszego słowa HSC0
R4098	Nastawa młodsze słowa HSC0
R4099	Nastawa starszego słowa HSC0

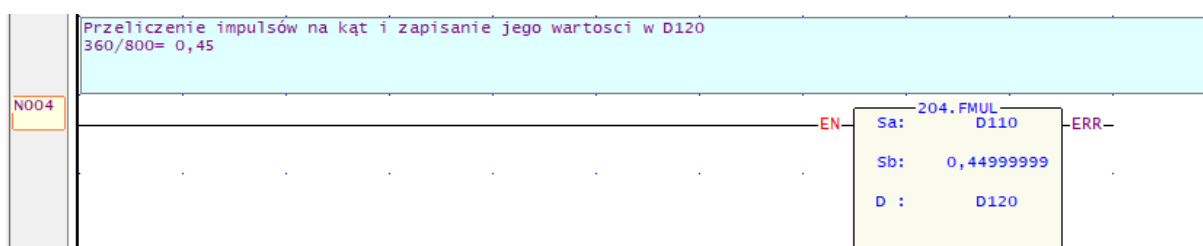
11) Teraz w N002 wprowadzamy funkcję FUN92 odpowiadającą za odczyt danych z enkodera i zapisywanie wartości w rejestrach. Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0→1, funkcja odczyta wartość bieżącą enkodera (CV) z HSC o numerze określonym przez CN i zapisze ją w odpowiadającym temu HSC rejestrze CV (np. wartość bieżąca (CV) w HSC0 zostanie odczytana i zapisana w DR4096.



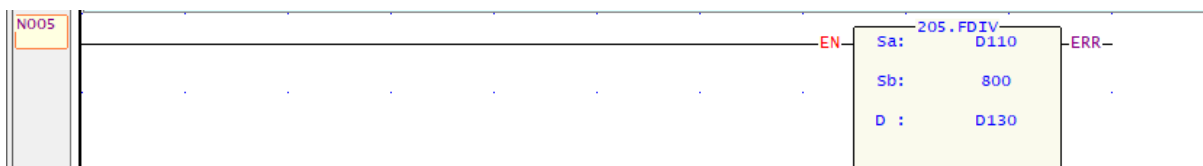
12) Żeby operować na danych enkodera z rejestru R4096 przepisujemy je do rejestru D100 używając funkcji FUN08. Od teraz rejestr D100 przechowuje liczbę całkowitą odpowiadającą liczbie impulsów wysłanych przez enkoder. Następnie, aby móc wykonywać przekształcenia matematyczne tej wartości musimy zamienić zmienną całkowitą (Integer) z rejestru D100 na zmienną przecinkową (Float) i zapiszemy ją w rejestrze D110, co zostało zrealizowane przez funkcję FUN200 w linii N003.



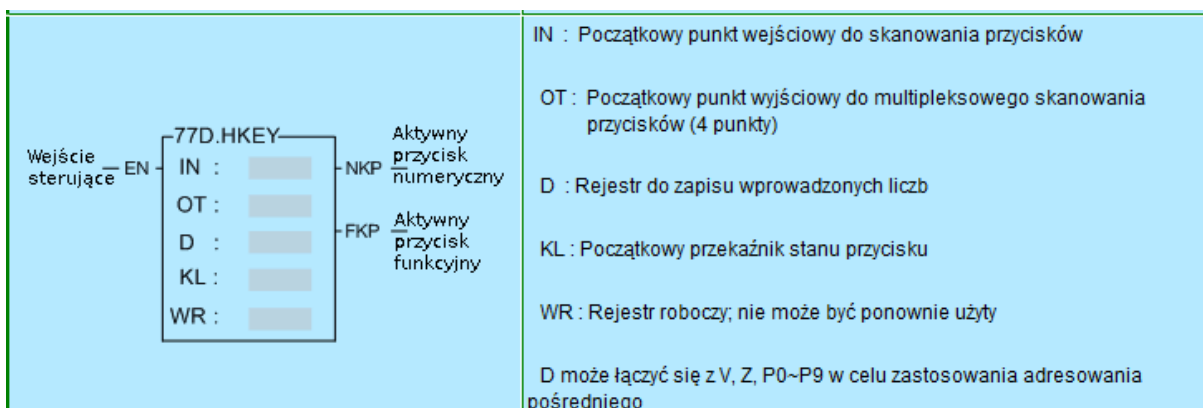
- 13) Przeliczmy więc liczbę impulsów z D110 na kąt obrotu i zapiszmy tę wartość w rejestrze D120. W tym celu skorzystamy z funkcji mnożącej FUN204. Ponieważ 360 stopni dzielimy na 800 impulsów, 1 impuls odpowiada kątowi 0.45 stopnia ($360/800=0.45$). Dlatego wartości D110 mnożymy przez 0.45, co realizuje linia N004.



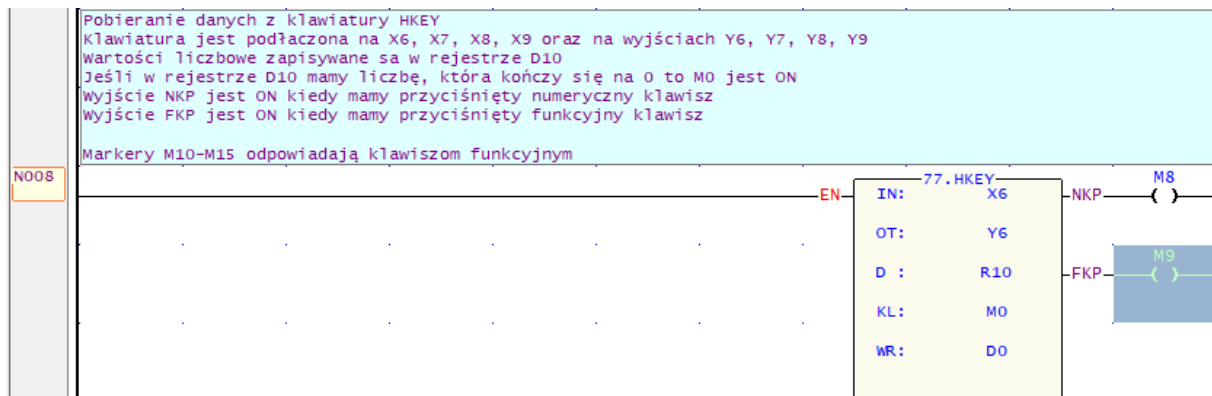
- 14) Teraz możemy obliczyć liczbę obrotów jakie zostały wykonane przez enkoder. Dzieląc liczbę impulsów jakie zliczył enkoder z rejestru D110 przez liczbę 800 impulsów jakie przypadają na jeden pełny obrót (360 stopni) otrzymamy właśnie liczbę obrotów wału enkodera (lub jej ułamek przy niepełnym obrocie). W tym celu skorzystamy z funkcji dzielącej FUN205, gdzie D110 jest dzielone przez 800, a wyniki zapisany jest w D130.



- 15) Teraz zajmiemy się odczytem z klawiatury funkcyjno-numerycznej. W tym celu użyjemy funkcji FUN77. Klawiatura jest podłączona do wejść X6, X7, X8 i X9 oraz wyjścia Y6, Y7, Y8, Y9, dzięki czemu tworzy układ multipleksowany 4x4 obsługujący 16 klawiszy.



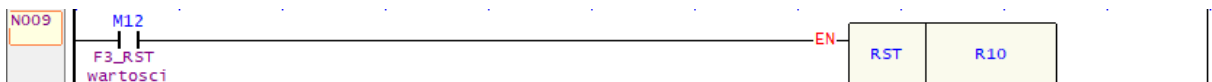
Rejestr do zapisu wprowadzonych liczby ustalamy jako R10. Marker M0 będzie aktywny kiedy wprowadzona liczba będzie miała na końcu cyfrę 0. Rejestr roboczy ustawiamy na D0.



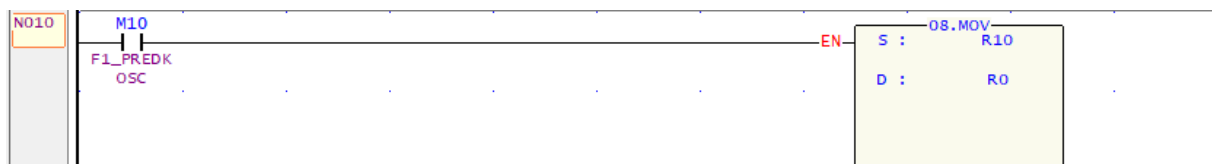
Jeśli któryś z przycisków numerycznych będzie aktywny marker M8 będzie załączony, natomiast jeśli wciśniemy przycisk funkcyjnym, marker M9 stanie się aktywny. Markery M10-M15 odpowiadają klawiszom funkcyjnym.

- M10 – klawisz F1
- M11 – klawisz F2
- M12 – klawisz F3
- M13 – klawisz F4
- M14 – klawisz F5
- M15 – klawisz F6

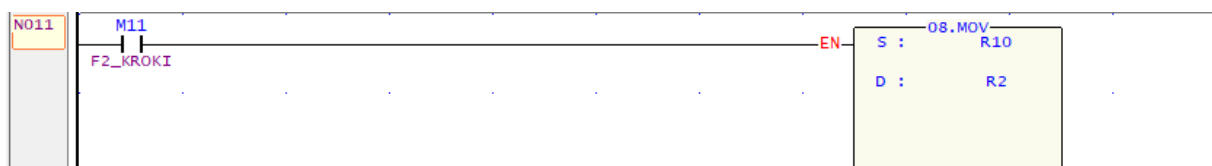
16) Nadajmy przyciskowi F3 funkcję resetowania wartości numerycznej wpisanej w rejestrze R10.



17) Nadajmy klawiszowi F1 funkcję przypisania wartości z rejestru R10 jako wartość prędkości silnika w rejestrze R0.



18) Nadajmy klawiszowi F2 funkcję przypisania wartości z rejestru R10 jako liczbę kroków z rejestru R2, które wykonać ma silnik.



19) W tym miejscu dobrze będzie uzupełnić *Karte Stanu* o istniejące, i o używane w przyszłości zmienne oraz rejestry.

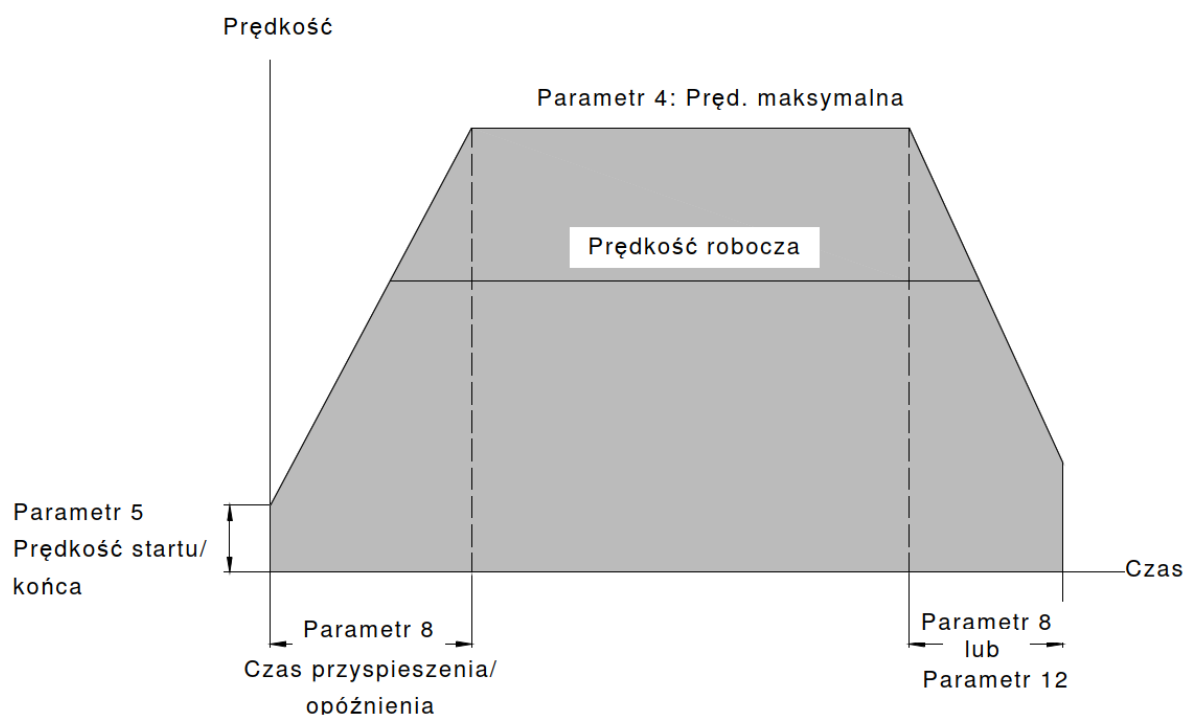
Adres	Status	Wartość	Adres	Status	Wartość
R4096	Całkowity 0				
D110	Całkowity 0				
D120	Całkowity 0				
D130	Całkowity 0		D200	Całkowity 0	
			D210	Całkowity 0	
			D220	Całkowity 0	
			D225	Całkowity 0	
R0	Całkowity 0				
R2	Całkowity 0				
			M100	Zakładz	OFF
			M101	Zakładz	OFF
			M102	Zakładz	OFF
			M102	Zakładz	OFF
			M200	Zakładz	OFF
R10	Całkowity 0		M1941	Zakładz	OFF
M0	Zakładz	OFF			
M10	Zakładz	OFF	R4096	Całkowity 0	
M11	Zakładz	OFF	R4097	Całkowity 0	
M12	Zakładz	OFF	R4098	Całkowity 0	
M13	Zakładz	OFF	R4099	Całkowity 0	
M14	Zakładz	OFF			
M15	Zakładz	OFF			
M100	Zakładz	OFF	D10	Całkowity 0	

20) Teraz zdefiniujemy parametry naszego silnika krokowego. W tym celu po lewej stronie rozwijamy drzewko zakładki *Edycja tabel* -> *Tabele parametrów serw.* Klikamy PPM i tworzymy *Nową tabelę*. Nazwę tabeli nadajemy np. „*Parametry silnika krokowego*”, a adres startowy wpisujemy R1000.

W nowej tabeli w Pozycji 0, ustawiamy jednostki – wybieramy impulsy. W pozycji 1 wybieramy ile impulsów trzeba wysłać do naszego silnika, żeby wykonał jeden obrót, zgodnie ze specyfikacją producenta jest to 200 impulsów/obrót. W pozycji 4, ustawiamy prędkość maksymalną silnika 800 krok/sek – sprawdzone eksperymentalnie. W pozycji 5 ustawiamy prędkość rozpędzania się

silnika i hamowania silnika na 50 krok/sek. W pozycji 8 ustawiamy czas przyspieszania i hamowania na 1000 [ms].

Tabela parametrów serw - [Parametry silnika krokowego]					
Kalkulator(C) Ustaw(S)					
R1000	0.Jednostki:	1:Impulsy	R1013	10.Kompensacja ruchu + :	0 Ps
R1001	1.Imp/obr (16bit):	200	R1014	11.Kompensacja ruchu - :	0 Ps
DR1002	2.Dystans/obr(16bit)	2000	R1015	12.Czas opóźnienia :	0 mS
R1004	3.Dokładność pozycjonowania:	2	R1016	13.Czas interpolacji:	0 mS
DR1005	4.Maksymalna prędkość:	800	DR1017	14.Imp/obr (32bit):	0
DR1007	5.Prędkość początku/końca:	50	R1019_LB	15 0.Wejście bazujące :	Nie używane
R1009	6. Prędkość pełzania:	200	R1019_HB	15 1.Wejście zakresu :	Nie używane
R1010	7.Kompensacja luzu:	20 Ps	R1020_LB	15 2.Wejście PG0 :	Nie używane
R1011	8.Czas przyspieszenia/opóźnienia:	1000 mS	R1020_HB	15 3.Wyjście z bazowania:	Nie używane
R1012_LB	9_0.Kontrola kierunku :	0:Dodatni	DR1021	16.Przyjmij pozycję po zbazowaniu:	0 Ps
R1012_HB	9 1.Kierunek bazowania:	1:Ujemny(w le)	R1023	17.Liczba zliczeń PG0 :	1
Dostępne: 2840 słów(Auto) Użyte: 24 słów Pozycja: R1000-R1023					
Ustaw domyślne			OK Anuluj		



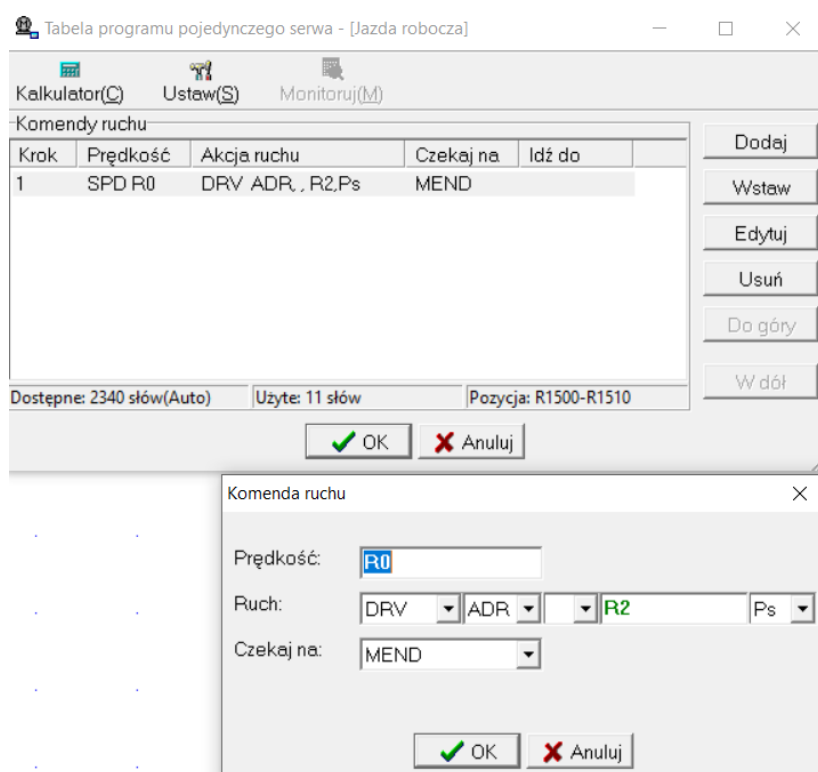
21) Teraz zdefiniujemy sobie dwa rodzaje ruchu silnika krokowego. Po lewo w drzewku znajdujemy zakładkę *Tabele programów serw*, klikamy PPM i wybieramy *Nowa tabela*. Nazwę tabeli wpisujemy „*Jazda robocza*”, a adres startowy tabeli ustawiamy na R1500. W oknie, które się pojawi klikniemy okienko *Dodaj* i tworząc w ten sposób nową komendę ruchu.

Prędkość silnika ma być pobierana z rejestru R0. Mamy do wyboru 3 rodzaje ruchu: **DRV** – silnik startuje i zatrzymuje się z prędkością startu/końca ustawionej w *Tabeli parametrów serwa*. Zastosowanie DRV spowoduje, że kolejne wprowadzone ruchy będą po sobie następować (kolejne kroki programu pozycjonującego) w taki sposób, że pomiędzy nimi będzie dochodziło do rozpędzania/zwalniania z prędkością startu/końca. **DRVC** – ruch ciągły, który różni się od ruchu DRV tym, że zamiast wywoływania kolejnych ruchów jeden po drugim, następuje płynne przejście z jednej zadanej prędkości na kolejną. Można wywołać maksymalnie 8 kolejnych ruchów DRVC, przy czym ostatni ruch musi być ruchem DRV. **DRVZ** – ruch, w którym wybieramy jeden z 3 trybów: MD0: Bazowanie na wejściu DOG (z tabeli parametrów serwa), polegające na najejchaniu na czujnik wejścia z prędkością R0, zwolnieniu do prędkości pełzania (z tabeli parametrów serwa) i kontynuowaniu ruchu w tym samym kierunku aż do „przejechania za czujnik” – stanu opadającego na wejściu. MD1: Bazowanie na wejściu DOG (z tabeli parametrów serwa), polegające na najejchaniu na czujnik wejścia z prędkością R0, zwolnieniu do prędkości pełzania, zmianie kierunku ruchu na przeciwny i „zjeździe z czujnika” – stanu opadającego na wejściu. MD2: Bazowanie polegające na zliczaniu impulsów z wejścia PG0 do wartości Liczba zliczeń PG0 (z tabeli parametrów serwa).

ADR to ruch relatywny, który odpowiada na polecenie „przesuń się od bieżącej pozycji o zadaną wartość”. W ramach tego trybu dostępne są trzy pod-opcje wykonania: „+”: ruch w kierunku dodatnim, „-”: ruch w kierunku ujemnym, [puste pole]: wartość z rejestru zadanego dystansu jest odczytywana ze znakiem, który definiuje kierunek ruchu. **ABS** to ruch absolutny, który odpowiada na polecenie „przejdź do konkretnej pozycji absolutnej”. Ten tryb ruchu jest zazwyczaj wykorzystywany po wcześniejszym bazowaniu osi.

W naszym przypadku ustawiamy DRV ADR [puste pole] R2 Ps, tzn., silnik ma się obrócić o względny kat odpowiadający liczbie impulsów z rejestru R2, rozpędzając się z prędkości startu/końca 50 krok/sek w czasie 1000 ms do prędkości z rejestru R0.

W okienku „Czekaj na” mamy kilka opcji do wyboru: **WAIT TIME** – wartość z pola obok określa czas pauzy po wykonaniu ruchu, zanim przejdzie do kolejnego. Jednostką jest 10 ms, a wartość może być wpisana jako stała lub pobrana z adresu rejestru. **WAIT** – w polu obok wpisuje się adres zmiennej dyskretnej. Po zakończeniu ruchu program będzie czekał na pojawienie się stanu wysokiego na wskazanej zmiennej dyskretnej, zanim przejdzie dalej. **EXT** – w polu obok wpisuje się adres zmiennej dyskretnej. Jeśli podczas wykonywania ruchu pojawi się stan wysoki na wskazanej zmiennej dyskretnej, ruch zostanie przerwany. Jeśli ruch się zakończył, a stan wysoki na tej zmiennej nie pojawił się, program przechodzi w tryb WAIT (musi pojawić się stan wysoki, aby program kontynuował). **MEND** – koniec programu pozycjonującego.



Ruch *Jazda robocza* jest ruchem podstawowym mający za zadanie wykonać obrót silnika o zadaną w rejestrze R0 liczbę kroków. Jednakże, ze względu na obecność połączenia wału enkodera i silnika gumowym paskiem o przekroju prostokątnym, mogą pojawić się niedokładności pomiędzy kątem obrotu wału silnika a wału enkodera. Niedokładności te są wynikiem kwadratowego przekroju gumowego paska, który podczas obracania się potrafi się skrzywić, co wpływa na rozbieżność pomiędzy kątami obrotu. Te małe rozbieżności będą kompensowane przy pomocy drugiego zdefiniowanego ruchu nazwanego *Dojazd do zadanego kąta*. Stwórzmy więc nową tabelę w zakładce *Tabele programów serw* i nazwijmy ją „*Dojazd do zadanego kąta*”, adres startowy tabeli ustawmy na R1600. Dodajmy ruch, w którym prędkość będzie 30 krok/sek, a liczba kroków pobrana zostanie z rejestru D225.

Edycja tabeli

Właściwości tabeli

Typ tabeli: Tabela programu pojedynczego serwa

Nazwa tabeli: Dojazd do zadanego kąta

Adres startowy tabeli: R1600

Rozpiętość tabeli: ☒ Przydzielana dynamicznie
☐ Sztywna

☐ Zaczynam wartości ze sterownika
☐ Zaczynam wartości z rejestrów ROP

Opis

OK Anuluj

Tabela programu pojedynczego serwa - [Dojazd do zadanego kąta]

Kalkulator(C) Ustaw(S) Monitoruj(M)

Komendy ruchu

Krok	Prędkość	Akcja ruchu	Czekaj na	Idź do
1	SPD 30	DRV ADR+, D225,Ps	MEND	

Dostępne: 2240 słów(Auto) Użyte: 11 słów Pozycja: R1600-R1610

OK Anuluj

Komenda ruchu

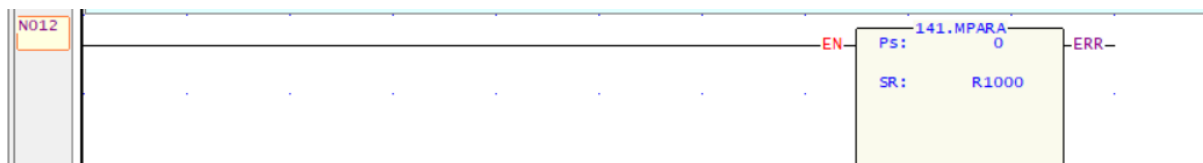
Prędkość: 30

Ruch: DRV ADR+ D225 Ps

Czekaj na: MEND

OK Anuluj

22) Teraz należy wywołać parametry silnika które wprowadziliśmy w programie przy pomocy funkcji FUN141.

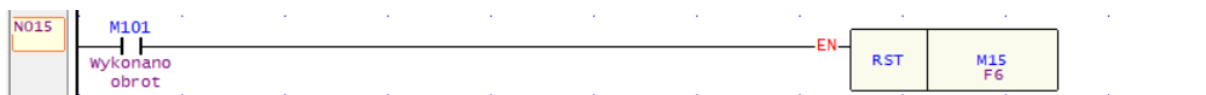


23) W kolejnym kroku wywołamy funkcję, która będzie miała za zadanie obrócić silnikiem zgodnie z ruchem zdefiniowanym przez Nas jako „Jazda robocza”. W tym celu skorzystamy z FUN140. W bločku ustawiamy Ps na wartości 0, która odpowiada

szybkim wyjściem impulsowym Y0 i Y1. Parametr *SR* ustawiamy na wartości rejestru R1500, w którym zdefiniowaliśmy nasz ruch „*Jazda robocza*”. Adres roboczy rejestrów *WR* ustawiamy na np. R3000. Przyciskiem F6 rozpoczniemy ruch, który będzie można przerwać przyciskiem F5. Kiedy silnik wykona ruch zgodnie z definicją „*Jazda robocza*” na wyjściu marker M101 załączy się.



- 24) Kiedy obrót zostanie wykonany zresetujemy wartość markera M15, który przechowuje informację, że przycisk F6 został wciśnięty.



- 25) Teraz wykonamy kilka obliczeń matematycznych. Zaczniemy od zamiany liczby całkowitej przechowywanej w rejestrze R2 na liczbę zmiennoprzecinkową i zapiszemy ją w R20.



Następnie obliczymy na podstawie zadanej liczby kroków R2, ile obrotów ma zostać wykonanych przez silnik, co opisuje równanie $R20/200=D200$, bo 200 to liczba kroków na pełen obrót. Wynik tego dzielenia zapiszemy w rejestrze D200. Funkcja dzielenia to FUN205.



- 26) Obliczymy jaka jest różnica pomiędzy wartością obrotów wykonanych przez silnik a tych wykonanych przez enkoder. W tym celu przy pomocy funkcji odejmowania FUN203 odejmiemy liczbę obrotów enkodera zapisaną w rejestrze D130 od wartości obrotów jakie zostały zadane do wykonania silnikowi w D200. Wynik różnicy zapiszemy w rejestrze D210.



- 27) Mając różnicę obrotu zadanego silnikowi a zmierzonego przez enkoder, należy przeliczyć ją na liczbę impulsów jakie należy wystać silnikowi, aby wał enkodera osiągnął wartość obrotów, które zostały przez nas zadane. W tym celu należy pomnożyć wartość D210

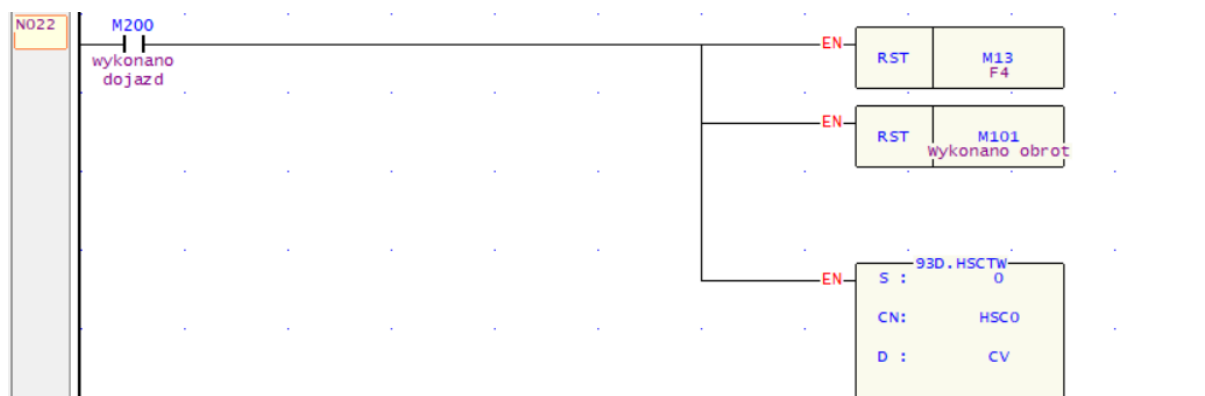
przez liczbę 200 kroków na obrót, korzystając z funkcji FUN204. Następnie należy tę wartość zamienić z wartości zmiennoprzecinkowej na całkowitą.

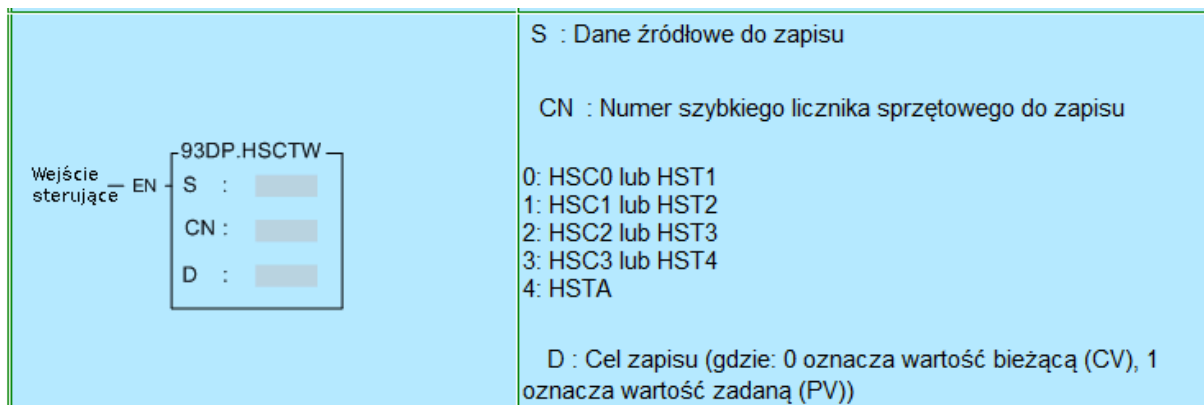


28) W kolejnym kroku nałożymy, żeby silnik obrócił się o brakującą liczbę obrotów, tak aby enkoder wykonał zadaną liczbę obrotów. W tym celu wywołamy funkcją FUN140 ruch zdefiniowany przez Nas jako „Dojazd do zadanego kąta”. Dojazd ten ma nastąpić po wciśnięciu przycisku F4, ale tylko po wykonaniu się *Jazdy roboczej*, czyli marker M101 jest aktywny. Parametr *Ps* ustawiamy na 0, a *SR* na rejestr R1600 odpowiadający ruchowi „Dojazd do zadanego kąta”. Kiedy brakująca liczba kroków zostanie wykonana, marker M200 załączy się.



29) Kiedy marker M200 jest aktywny ma on zresetować działanie przycisku F4, zresetować marker M101 przechowujący informację o wykonaniu ruchu „Jazda robocza”, a także przy pomocy funkcji FUN93, ustawić ma obecną wartość położenia zmierzoną przez enkoder na wartość 0 (zresetować go). W tym celu ustawiamy *S* na 0, *CN* na HSC0 i *D* na CV.





- 30) Ponadto po czasie np. 0.1 sekundy od zadziałania markera M200 ma nastąpić zasetowanie markera M14 odpowiadającego przyciskowi F5 mającego zatrzymać każdy ruch silnika. W tym celu należy skorzystać z czasówki T1 oraz pomocniczego marker M201, którego wartość na koniec również należy zresetować.



Podczas opracowywania tego ćwiczenia bardzo przydatne były informacje dotyczące programowania sterowników FATEK zawarte na stronie www.multiprojekt.pl.

SPRAWOZDANIE

W sprawozdaniu proszę zawrzeć zdjęcie układu oraz screen program PLC wraz z komentarzami, co realizowane jest w poszczególnych liniach kodu oraz parametry jakie zostały ustawione. Na koniec należy napisać swoje obserwacje, spostrzeżenia i wnioski z tego jak działa układ oraz opracowany program.