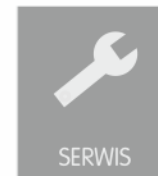


S100 – programowanie sekwencyjne (podstawy)

INNOWACYJNOŚĆ WIEDZA KONSEKWENCJA



ANIRO

Spis treści

1. Podstawowa wiedza

1. Wstęp
2. Programowanie
3. Aktywacja
4. Start i parametry
5. Adresy parametrów
6. Funkcje blokowe
7. 1 pętlowa sekwencja
8. Zasady programowania
9. Zasady wykonywania
10. Typy bloków funkcyjnych
11. Jak używać bloków funkcyjnych?

2. Proste przykłady

1. Dodawanie(1)
2. Dodawanie(2)
3. MPYDIV
4. COMPARE –GT
5. TIMER
6. ANDOR
7. SWITCH
8. BITTEST
9. LIMIT
10. LOWPASSFILTE R
11. UPCOUNT
12. Załączona lista podstawowych adresów



ANIRO

1. Podstawowa wiedza

1. Wstęp

- Sekwencja – uporządkowany ciąg wykonywanych poleceń, wedle zdefiniowanego schematu.
- Programowanie sekwencyjne pozwala na tworzenie prostych kroków sekwencyjnych, wykonywania pewnych instrukcji, wedle ustalonego porządku. Eliminuje to konieczność używania zewnętrznych sterowników logicznych, dla prostych aplikacji.
- Zakres działań, definiowanych instrukcji jest uzależniony jedynie wyobraźnią danego użytkownika.
- Falownik może działać w sposób automatyczny, wedle ustalonej sekwencji, przy użyciu zdefiniowanych stałych i bloków funkcyjnych, połączonych ze sobą adresami.

2. Programowanie

- Sekwencje buduje się za pomocą bloków funkcyjnych i powiązań między nimi (linki).
- Dla każdego bloku funkcyjnego definiujemy jedną lub dwie zmienne wejściowe. Blok funkcyjny po przetworzeniu informacji (wykonaniu funkcji) na wyjściu podaje wartość wynikową (wartość funkcji).
- Wejścia i wyjścia bloków funkcyjnych adresujemy w kodzie hex – adresy parametrów lub adresy wartości stałych.
- Każdemu wyjściu należy przydzielić adres wyjściowy. Wyjścia bloków funkcyjnych można podawać na wejścia kolejnych bloków funkcyjnych.
- Bloki funkcyjne mogą być używane wielokrotnie w jednej pętli programu.



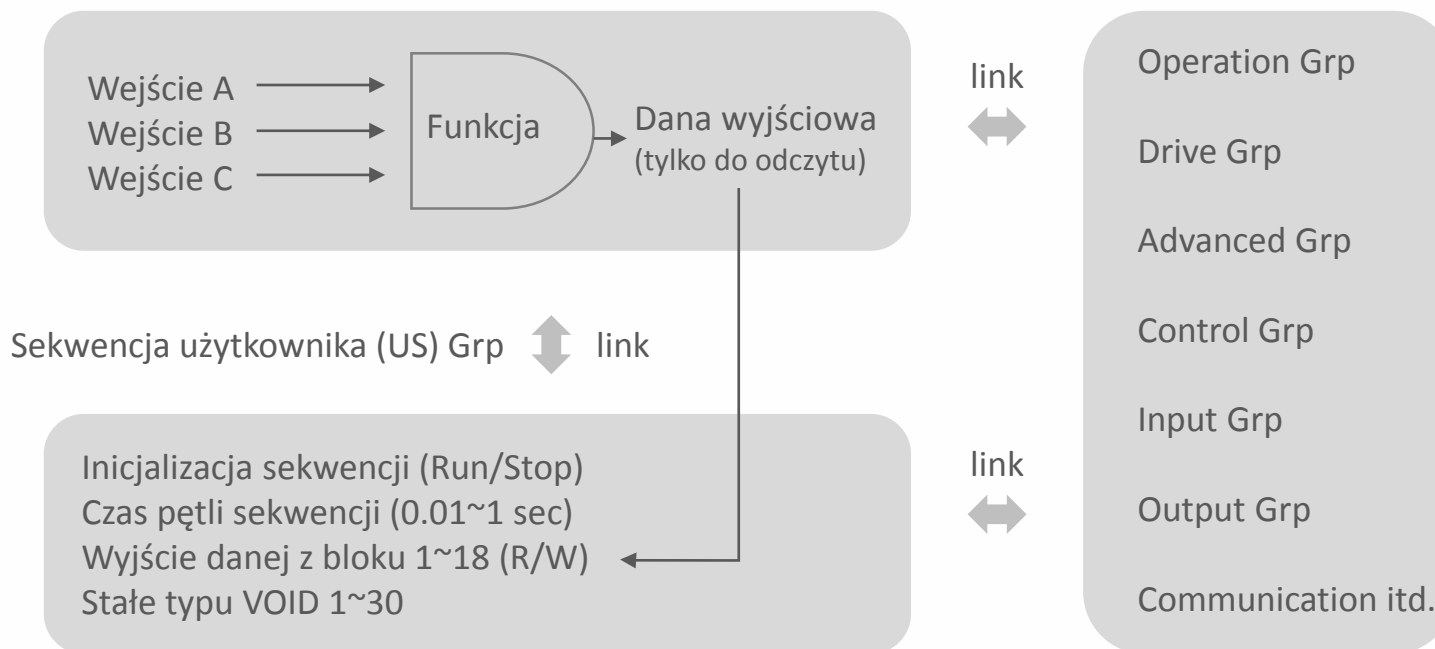
ANIRO

1. Podstawowa wiedza

3. Aktywacja sekwencji

- W celu aktywacji sekwencji należy ustawić: PAR mode → AP Grp → Ustawić wartość 02 na „Yes”
- Po aktywacji pojawią się nowe grupy parametrów: US i UF.

Funkcje sekwencji użytkownika (UF) Grp



ANIRO

1. Podstawowa wiedza

4. Start i parametry

Kod	Opis
AP-02. User Seq. En	Aktywacja sekwencji użytkownika. Pojawienie się nowych grup US i UF.
IN-65~71. P1~P7 Define	Jest możliwość wyzwalania startu i stopu sekwencji za pomocą wejścia cyfrowego. Należy do wejścia przydzielić funkcję nr 50: User Seq.
US-01. User Seq. Con	Wybór wyzwalania sekwencji użytkownika (Start/ Stop). Sterowanie z poziomu klawiatury (0 – start, 1 – stop, 2 – start/stop z wejścia cyfrowego).
US-02. User Loop Time	Ustawianie czasu pętli: 0.01s/0.02s/0.05s/0.1s/0.5s/1s.
US-11~28. Link User Out 1~18	Parametry związane z powiązaniem 18 bloków funkcyjnych. Jeżeli wartość wejścia wynosi 0x0000, skutkuje to brakiem wartości wyjściowej. Dla przykładu, jeżeli wartość wyjścia bloku ma odwoływać się do parametru częstotliwości zadanej, należy ustawić odpowiedni adres parametru (0x1101) w parametrze Link UserOut1.
US-31~60 Void Para 1~30	30 stałych typu void do dowolnego zdefiniowania. Operowanie wartościami stałymi dla bloków funkcyjnych.
UF-01~90	Ustawianie funkcji użytkownika zbudowanej z 18 dostępnych bloków funkcyjnych.

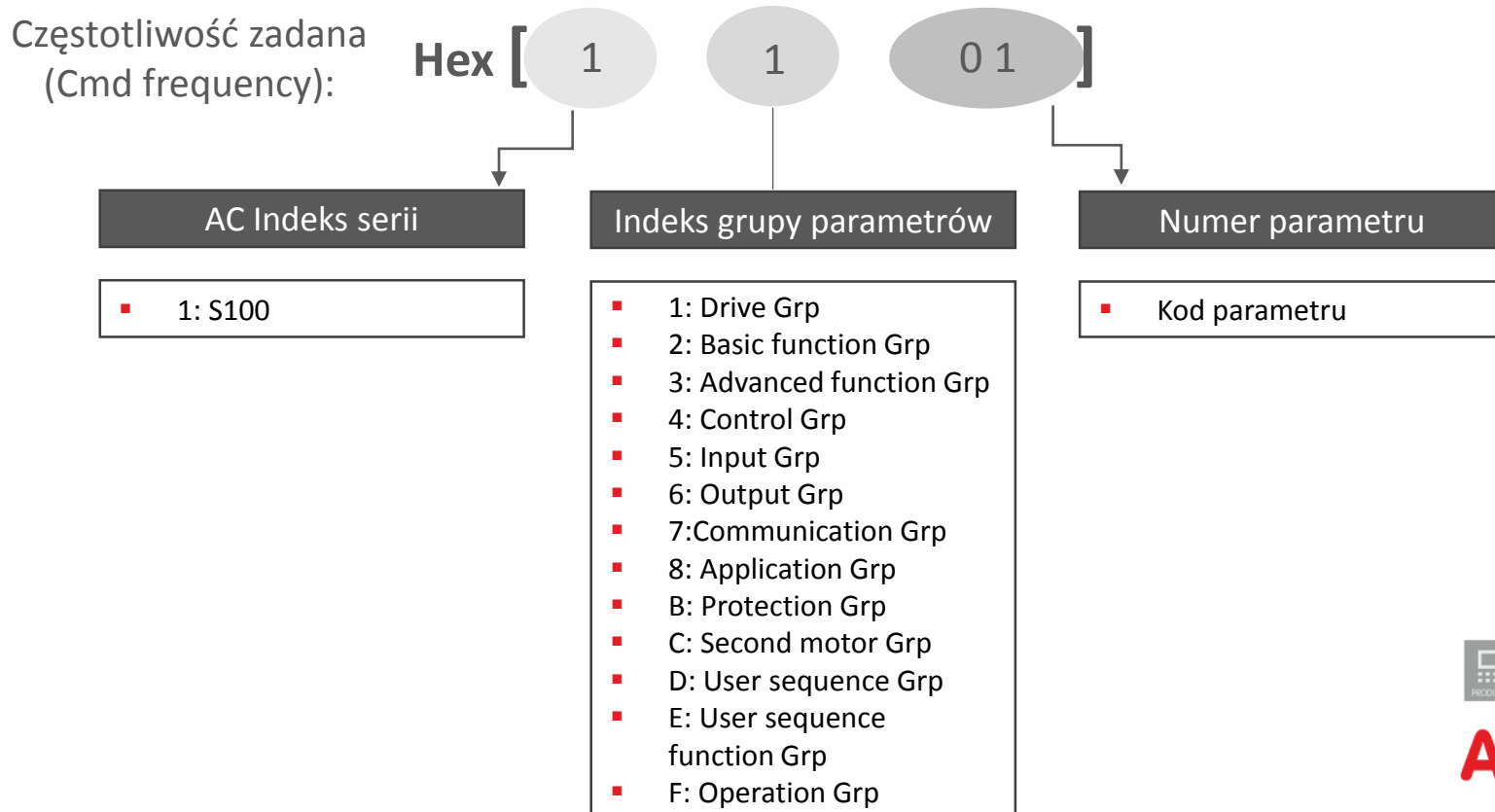
Jak rozpocząć lub zatrzymać sekwencję użytkownika?

- Ustaw z poziomu klawiatury w parametrze US-01 wartość RUN (start) lub Stop.
- Jeżeli ustaliłeś US-01 na wartość Digital In Run, przydziel dowolnemu wejściu cyfrowemu funkcję rozpoczęcia sekwencji użytkownika. Po podaniu sygnału na wejście, sekwencja rozpocznie się. Po zdjęciu sygnału, sekwencja zakończy działanie.

1. Podstawowa wiedza

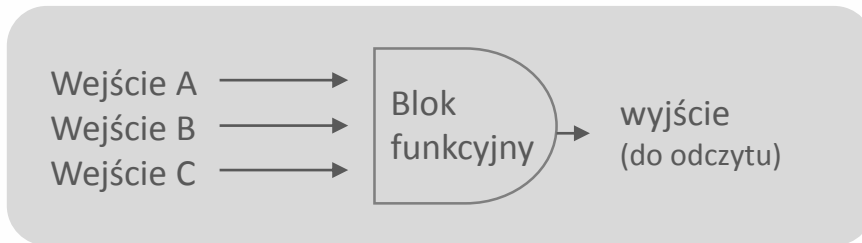
5. Adresy parametrów

- Do powiązania bloków funkcyjnych należy stworzyć odpowiednie łącza adresowe (linki)
- Adres komunikacyjny w kodzie HEX, poniżej formuła ustalania adresów:



1. Podstawowa wiedza

6. Funkcje blokowe



- Każdy blok funkcyjny składa się od 1 do 3 wejść i jednego wyjścia.

Typ	Opis
User Func @* (blok funkcyjny)	Wybrana funkcja do zrealizowania w bloku funkcyjnym.
User Input @-A (wejście A)	Adres komunikacyjny zmiennej wejściowej A
User Input @-B (wejście B)	Adres komunikacyjny zmiennej wejściowej B
User Input @-C (wejście C)	Adres komunikacyjny zmiennej wejściowej C
User Output @ (wyjście)	Wartość wyjściowa po wykonaniu funkcji w bloku funkcyjnym

@ jest numerem kroku (1~18)

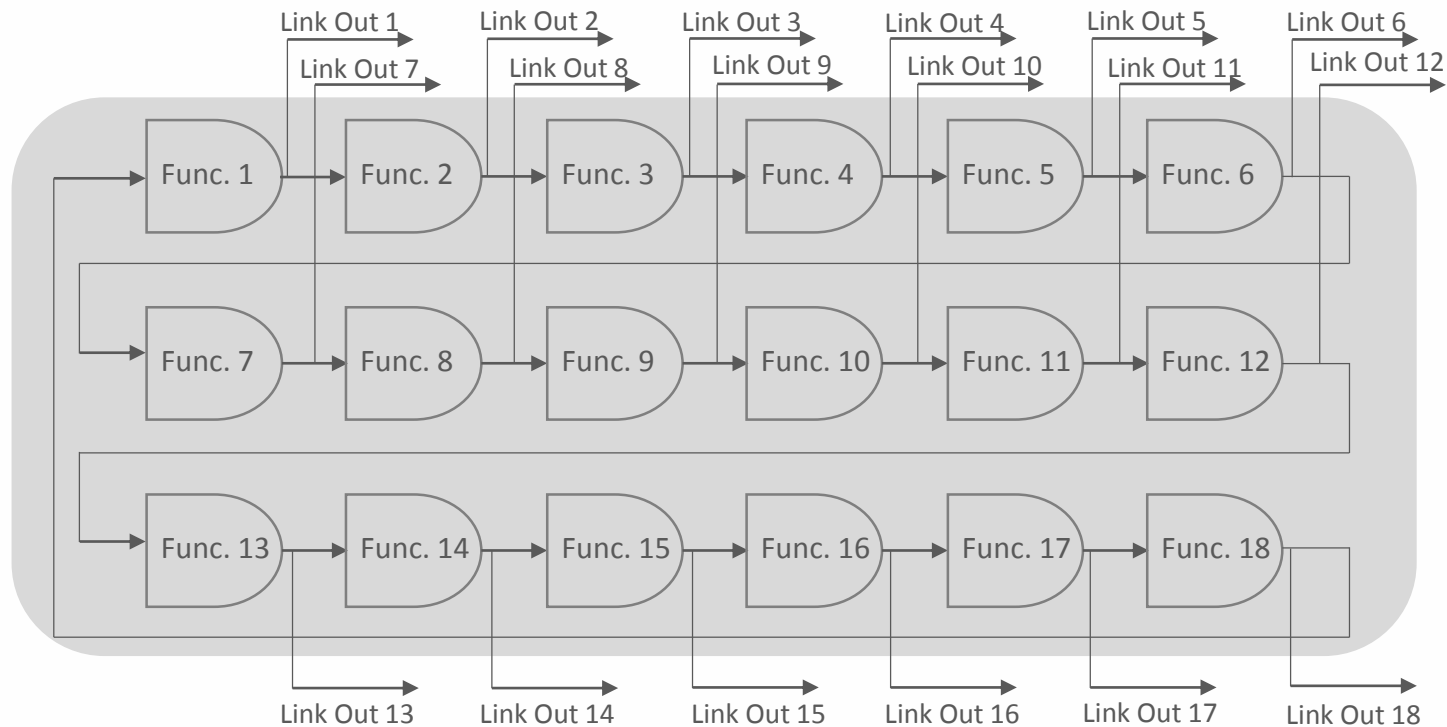


ANIRO

1. Podstawowa wiedza

7. 1-pętlowa sekwencja

- Jedna sekwencja składać się może z 18 kroków, 29 bloków funkcyjnych oraz 30 stałych typu VOID.
- 1 pętla wykonuje się w zdefiniowanym przez użytkownika czasie: 10~1000ms.



ANIRO

1. Podstawowa wiedza

8. Zasady programowania

- Parametry sekwencji użytkownika nie mogą być zmieniane podczas wykonywania sekwencji. W celu zmiany parametrów, należy w pierwszej kolejności występować sekwencję.
- Wszystkie wyjścia bloków funkcyjnych: "User Output@" z grupy UF, są tylko do odczytu. Adres ustala się w: „link@(Link UserOut@)” w grupie US.
- Bloki funkcyjne mogą być używane wielokrotnie w jednej pętli.
- Ustalać adresy „Link UserOut@” w celu połączenia do 18 bloków funkcyjnych. Jeżeli wartość wejściowa wynosi 0x0000, wyjście nie będzie dostępne.
- Stałe typu VOID mogą przyjmować wartości z zakresu: -9999~9999.
- Jeżeli ustawienia bloku funkcyjnego są błędne, wyjście: „User Output@” wynosić będzie -1.

9. Zasady wykonania

- 1 pętla odnosi się do wykonania sekwencji, wedle ustalonego porządku przez użytkownika, składająca się z max. 18 kroków.
- Użytkownik ustala czas wykonania pętli w zakresie: 10~1000ms w parametrze US-02.
- Wartość wyjściową bloku funkcyjnego można ograniczyć, używając funkcji limitu (Limit).
- Jeżeli skala dwóch powiązanych parametrów jest różna, skalowanie nie wykona się automatycznie (należy skalować każdy parametr indywidualnie).

1. Podstawowa wiedza

10. Typy bloków funkcyjnych (1)

Nr	Typ	Opis
0	NOP	Brak operacji.
1	ADD	Operacja dodawania, $(A+B)+C$
2	SUB	Operacja odejmowania, $(A-B)-C$
3	ADDSUB	Operacja złożona dodawanie i odejmowanie, $(A+B)-C$
4	MIN	Wynik w postaci najmniejszej wartości spośród wartości wejściowych, $\text{MIN}(A,B,C)$
5	MAX	Wynik w postaci największej wartości spośród wartości wejściowych, $\text{MAX}(A,B,C)$
6	ABS	Wynik w postaci wartości bezwzględnej parametru A, $ A $
7	NEGATE	Wynik w postaci negacji parametru A, $-(A)$
8	REMAINDER	Operacja uzyskiwania reszty z działania na A oraz B, $A \% B$
9	MPYDIV	Operacja złożona wykorzystująca mnożenie i dzielenie, $(A \times B)/C$. Jeżeli $C=0$ to wynikowo otrzymujemy tylko wynik mnożenia $A \times B$.
10	COMPARE-GT (większe niż)	Operacja porównania: jeżeli $(A>B)$ to wynikiem jest C; jeżeli $(A \leq B)$ to wynikiem jest 0. Jeżeli $C=0$ i warunek spełniony to wynikiem jest 1.
11	COMPARE-GTEQ (większe lub równe niż)	Operacja porównania. Jeżeli $(A \geq B)$ to wynikiem jest C. Jeżeli $(A < B)$ to wynikiem jest 0. Jeżeli $C=0$ i warunek jest spełniony to wynikiem jest 1.
12	COMPARE-EQUAL (równe)	Operacja porównania. Jeżeli $(A==B)$ to wynikiem jest C. Dla wszystkich innych kombinacji wynikiem jest 0. Jeżeli $C=0$ i warunek jest spełniony to wynikiem jest 1.
13	COMPARE-NEQUAL (różne)	Operacja porównania. Jeżeli $(A \neq B)$ to wynikiem jest C. Dla wszystkich innych kombinacji wynikiem jest 0. Jeżeli $C=0$ i warunek jest spełniony to wynikiem jest 1.

1. Podstawowa wiedza

10. Typy bloków funkcyjnych (2)

Nr	Typ	Opis
14	TIMER	<p>Dodaje 1 za każdym razem gdy sekwencja kończy pętlę. A: wartość max pętli, B: Uruchomienie/zatrzymanie układu czasowego, C: Wybór trybu wyjściowego.</p> <p>Jeżeli B wynosi 1, timer stopuje (wyjście 0). Jeżeli B wynosi 0, timer startuje. Jeżeli C wynosi 1, na wyjściu aktualna wartość układu czasowego.</p> <p>Jeżeli C wynosi 0, na wyjściu pojawia się 1 po przekroczeniu A(Max).</p>
15	LIMIT	<p>Ustala wartość graniczną dla A. Jeżeli A mieści się pomiędzy B oraz C, wynikiem jest wartość wejściowa A. Jeżeli A jest większe od B, wynikiem jest B. Jeżeli A jest mniejsze od C, na wyjściu jest C. B musi być \geq od C.</p>
16	AND	Operacja logiczna AND (A & B) & C
17	OR	Operacja logiczna OR, (A B) C
18	XOR	Operacja logiczna XOR, (A^B)^C
19	AND/OR	Złożona operacja logiczna AND/OR, (A and B) C
20	SWITCH	<p>Wynikiem jest wartość po wybraniu jednej z dwóch wartości wejściowych, jeśli (A) to B a w przeciwnym wypadku C.</p> <p>Jeżeli A równe 1, na wyjściu B. Jeżeli A wynosi 0, na wyjściu C.</p>
21	BITTEST	<p>Sprawdzanie bitu B parametru A, BITEST (A,B).</p> <p>Jeśli bit B wartości wejściowej A wynosi 1, to wynikiem jest 1. Jeśli wynosi 0, to wynikiem również jest 0. Wartość wejściowa B musi mieścić się w przedziale 0-16. Jeśli wartość jest większa od 16, to zostanie ona rozpoznana jako 16. Jeśli B wynosi 0, wynikiem będzie 0.</p>

1. Podstawowa wiedza

10. Typy bloków funkcyjnych (3)

Nr	Typ	Opis
22	BITSET	Ustawianie bitu B parametru A, BITSET (A,B). Wynikiem jest zmieniona wartość po ustawieniu bitu B do wartości wejściowej A. Wartość B musi mieścić się w przedziale 0-16. Wartości wyższe rozpoznawane będą jako 16. Jeśli B=0, wynik również 0.
23	BITCLEAR	Kasowanie bitu B parametru A, BITCLEAR (A,B). Wynikiem jest zmieniona wartość po skasowaniu bitu B do wartości wejściowej A. Wartość B musi mieścić się w przedziale 0-16. Wartości wyższe rozpoznawane będą jako 16. Jeśli B=0, wynik również 0.
24	LOWPASSFILTER	Wynikiem jest wartość wejściowa w A gdy filtr B uzyskuje stałą czasową: BxUS-02 (czas pętli US). W powyższym wzorze należy ustawić czas gdy wartość wyjściowa A osiągnie 63,3%. C odnosi się do działania filtru. Jeśli wynosi 0 to operacja zostaje rozpoczęta.
25	PI_CONTROL	Wzmocnienie P,I = A, B parametrów wartości wejściowych, wartość wyjściowa w postaci C. Warunki dla wartości wyjściowej PI_PROCESS output: C=0: Const PI C=1: PI_PROCESS_B>=PI_PROCESS-OUT>=0 C=2: PI_PROCESS-B>=PI_PROCESS-OUT>=-(PI-PROCESS-B).
26	PI_PROCESS	A jest błędem wartości wejściowej, B jest wyjściową wartością graniczną, C jest wartością wyjściową stałej Const PI. Zakres C wynosi 0 - 32 767.
27	UPCOUNT	Zliczanie w górę, następnie podaje wynik w postaci wartości UPCOUNT (A,B,C). Po odebraniu wyzwalającej wartości A, wartości wyjściowe są zliczane w górę zgodnie z warunkami C. Jeśli B wynosi 1, to operacja nie jest wykonywana i wyświetlane jest 0. Jeśli B = 0, to operacja jest wykonywana. Jeśli C=0, zliczanie w górę następuje gdy A zmienia się z 0 na 1. Jeśli C=1, zliczanie w górę następuje po zmianie A z 1 na 0. Jeśli C=2, to zliczanie w górę następuje za każdym razem gdy zmienia się wartość wyjściowa w A.
28	DOWNCOUNT	Zliczanie w dół, następnie daje wartość zliczania DOWNCOUNT (A,B,C). Po odebraniu wartości wyzwalającej A, wartości wyjściowe zliczane są w dół zgodnie z warunkami C. Jeśli B=1 to operacja nie jest wykonywana i wyświetlana jest wartość początkowa C. Jeśli B =0, to operacja jest wykonywana. Zliczanie w dół gdy A zmienia się z 0 na 1.

1. Podstawowa wiedza

11. Jak używać bloków funkcyjnych? (1)

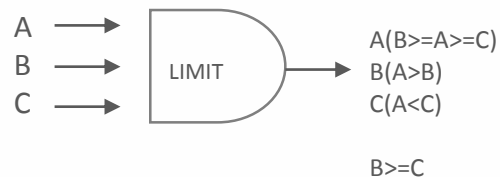
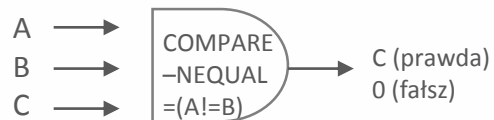
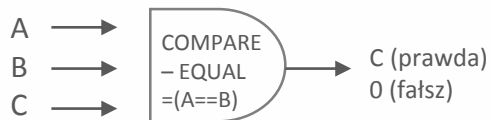
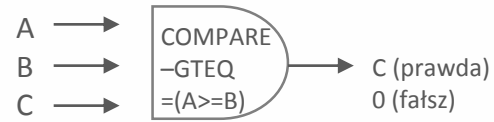
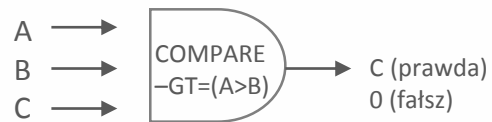
- ADD/SUB/ADDSUB: operacja dodawania/odejmowania. Jeżeli adres parametru c wynosi 0x0000, zostanie rozpoznany jako 0. (wejścia A i B powinny być różne od 0x0000.)
- MIN/MAX: operacja wartości MAX/MIN. Jeśli adres C wynosi 0x0000, operacja jedynie na A,B.
- ABS/NEGATE: Wartość bezwzględna/negacja. Operacja nie wykorzystuje B i C.
- REMAINDER: Reszta z dzielenia. Operacja nie wykorzystuje parametru C.
- MPYDIV: Operacja mnożenia/dzielenia. Jeśli adres C wynosi 0x0000, operacja jedynie na A,B.



1. Podstawowa wiedza

11. Jak używać bloków funkcyjnych? (2)

- COMPARE Operacja porównania. Jeżeli C wynosi 0x0000 i warunek jest spełniony, wynikiem jest C.
- COMPARE-GT/GTEQ: jeżeli A jest większe od B/jeżeli A jest większe równe B, wyjściem jest C, w przeciwnym wypadku 0.
- COMPARE-EQUAL/NEQUAL: Jeżeli A równa się/ nie równa się B, wyjściem jest C, w przeciwnym wypadku 0.
- LIMIT: limituje wartość A/ SWITCH: jeżeli A wynosi 1, wyjściem jest B. Jeżeli A wynosi 0, wyjściem jest C.



1. Podstawowa wiedza

11. Jak używać bloków funkcyjnych? (3)

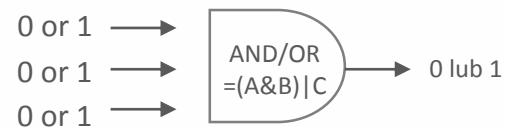
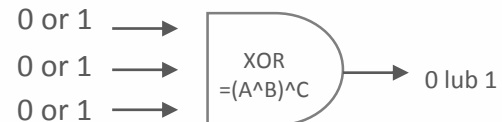
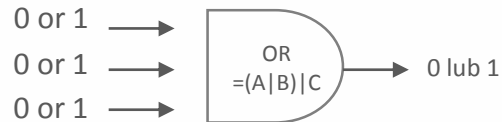
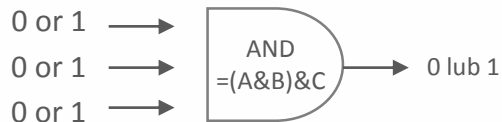


Tabela prawdy

AND		wynik
0	0	0
0	1	0
1	0	0
1	1	1
OR		Wynik
0	0	0
0	1	1
1	0	1
1	1	1
XRO		Wynik
0	0	0
0	1	1
1	0	1
1	1	0

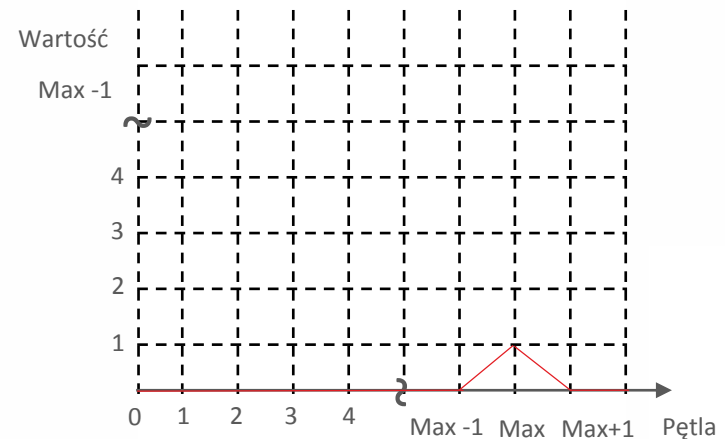
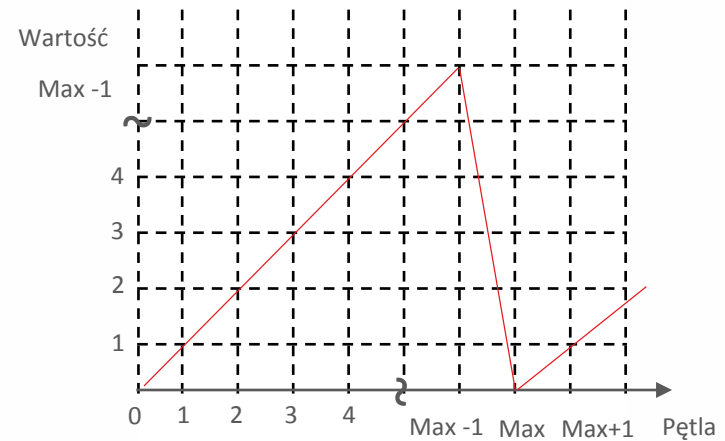
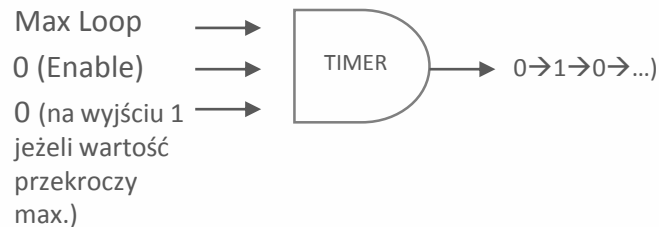
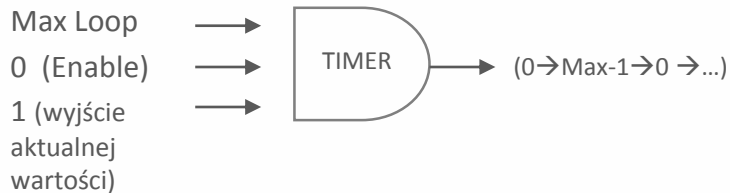
AND/OR			Wynik
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



1. Podstawowa wiedza

11. Jak używać bloków funkcyjnych? (4)

- Jeśli C wynosi 1, dodaje 1 za każdym razem, gdy sekwencja skończy pętlę.
- Jeśli C wynosi 0, na wyjściu 1 gdy timer przekroczy max ilość pętli



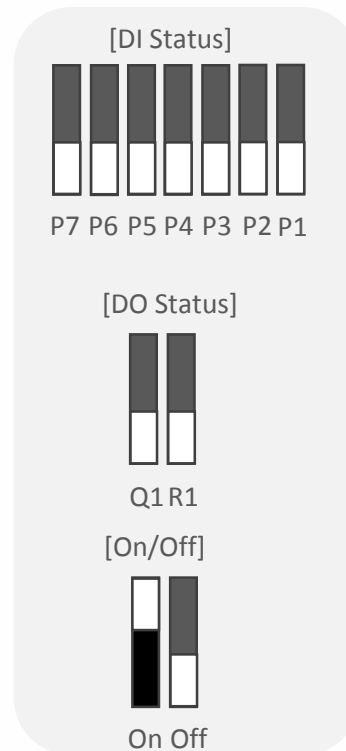
1. Podstawowa wiedza

11. Jak używać bloków funkcyjnych? (5)

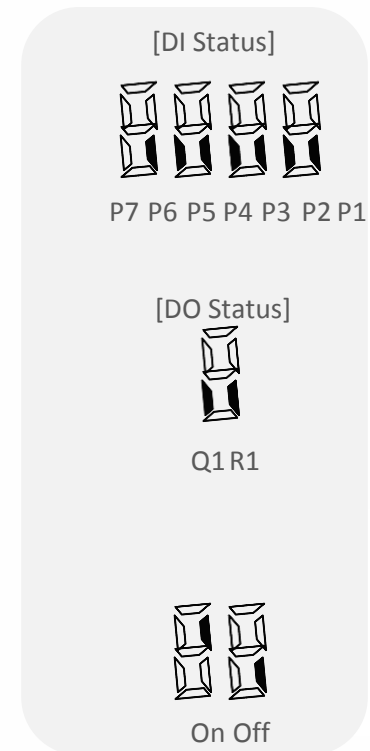
- BITTEST: Jeżeli bit B z A i wynosi 1, na wyjściu 1. Jeżeli wynosi 0, na wyjściu również 0.
- BITSET: Ustawienie bitu B z A, BITCLEAR: resetowanie bitu B z A.



Klawiatura LCD



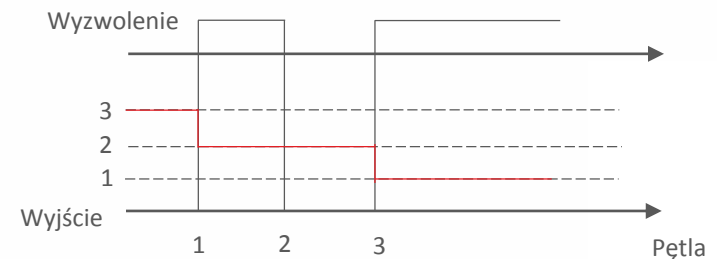
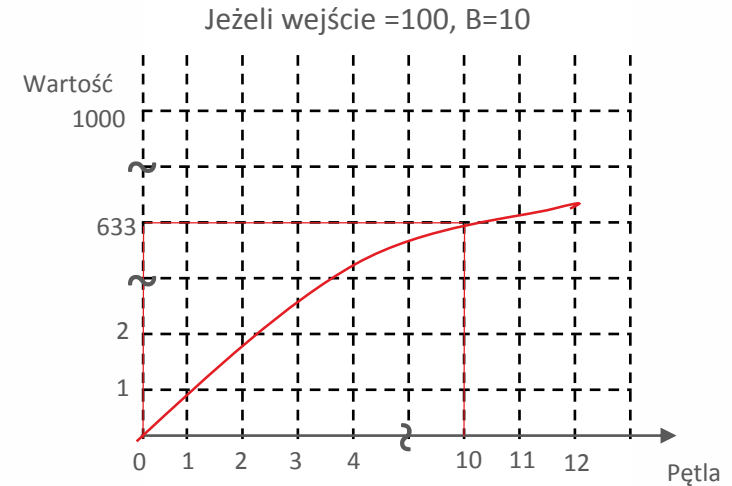
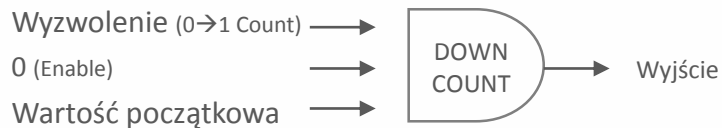
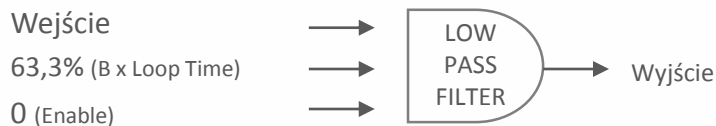
Wyświetlacz 7 segmentowy



1. Podstawowa wiedza

11. Jak używać bloków funkcyjnych? (6)

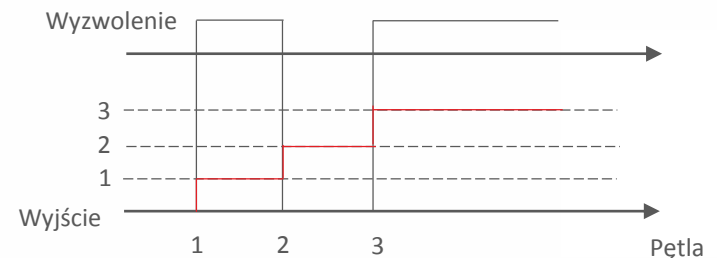
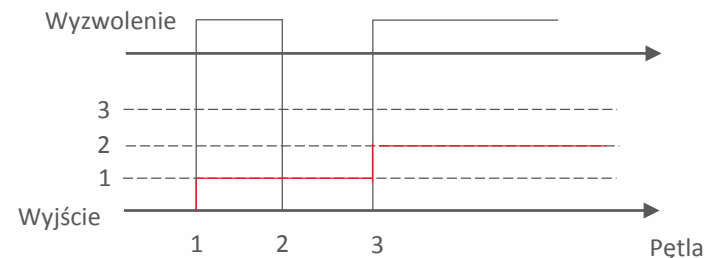
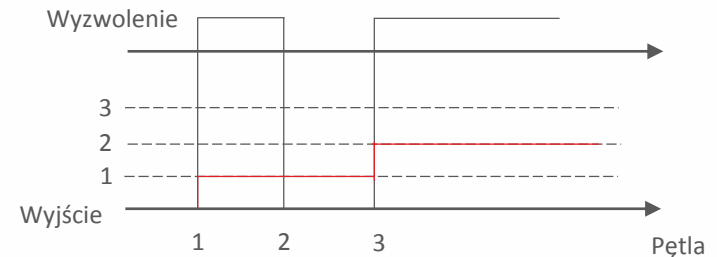
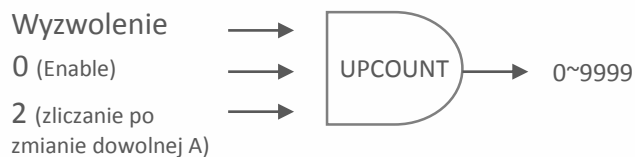
- LOWPASSFILTER: Wyjście w postaci A, gdy filtr B uzyskuje stałą czasową $B \times US-02$
- po otrzymaniu wyzwolenia w postaci(A), wyjście zliczane w dół zgodnie z warunkiem C. Zliczanie gdy A zmienia się z 0 na 1



1. Podstawowa wiedza

11. Jak używać bloków funkcyjnych? (7)

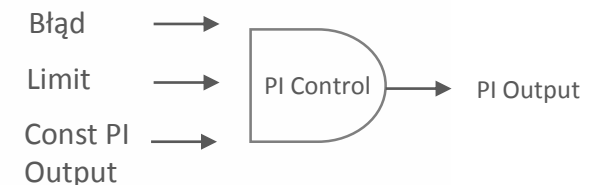
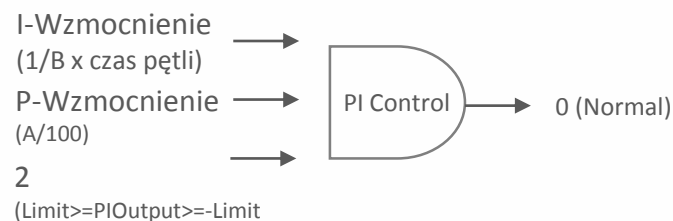
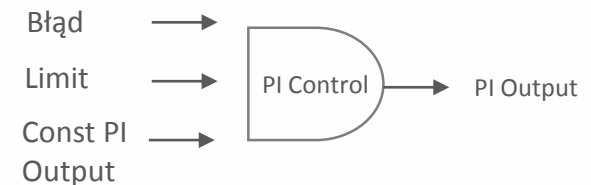
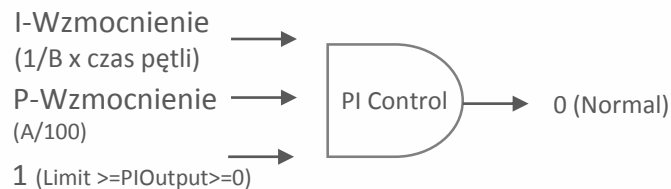
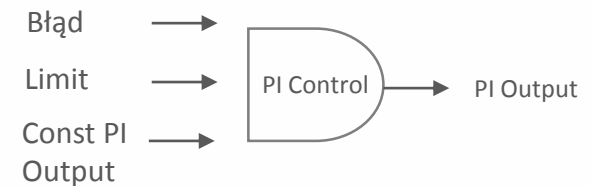
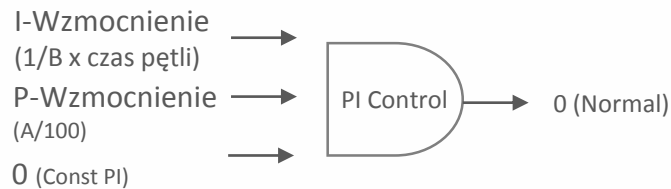
- UPCOUNT: Po otrzymaniu wyzwolenia (A), wyjście zliczane w przód zgodnie z warunkiem C. Jeżeli C=0, zliczanie gdy A zmienia się z 0 na 1. Jeżeli C=1 zliczanie przy zmianie A z 1 na 0. Jeżeli C=2 zliczanie po dowolnej zmianie A.



1. Podstawowa wiedza

11. Jak używać bloków funkcyjnych? (8)

- PI CONTROL+PI PROCESS: P,I wzmacnienie=A,B parametry wejściowe, wyjście jako C.
- PI Output = Błąd x P-Wzmacnienie (A/100) + [Pętla x {Błąd x I-Wzmacnienie(1/B x Czas pętli)}]



2. Proste przykłady

Przykład 1. ADD (1) (dodawanie dwóch zdefiniowanych stałych i wynik w postaci częstotliwości zadanej).

▪ **1-1. Lista kontrolna:**

- wyjście w postaci = ((A+B)+C).
- Jeśli C wynosi 0x0000, wartość zostanie rozpoznana jako 0.

- Jeżeli blok zostanie sparametryzowany nieprawidłowo, wyjście User Output@ w postaci -1.

▪ **1-2. Ustawienie parametrów:**

- US-31~33 Void Para1~3 → Wartości stałe całkowite (uwaga na skalowanie!!!)
- UF-2~4 User Input 1-A~C → adresy komunikacyjne stałych Void Para1~3

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	5 : 1.0S
US-11	Link User Out 1	1101
US-01	User Seq. Con	0 : Stop

Kod	Nazwa	Ustawienie
US-31	Void Para 1	3000
US-32	Void Para 2	2000
US-33	Void Para 3	0

2. Proste przykłady

Kod	Nazwa	Ustawienie
UF-01	User Func 1	1:ADD
UF-02	User Input 1-A	1D1F
UF-03	User Input 1-B	1D20
UF-04	User Input 1-C	1D21
UF-05	User Output 1	-

- **1-3. Test**
 - Ustaw US-01 (User Seq Con) na 1(Run).
 - Sprawdź czy UF-05 wynosi 5000.
 - Sprawdź, że częstotliwość wynosi 50Hz.

2. Proste przykłady

Przykład 1. Dodawanie (2) (dodawanie stałych i wynik w postaci częstotliwości zadanej)

■ 2-1. Ustawienie parametrów

- US-31~33 Void Para1~2 → Wartości stałe całkowite.
- User Input @-A~C → adresy komunikacyjne wartości stałych: Void Para1~3.

Kod	Nazwa	Ustawienie	Kod	Nazwa	Ustawienie	Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes	UF-01	User Func 1	1:ADD	UF-06	User Func 2	1:ADD
US-02	User Seq. Loop Time	5 : 1.0S	UF-02	User Input 1-A	1101	UF-07	User Input 2-A	1D1F
US-11	Link User Out 1	1101	UF-03	User Input 1-B	1D20	UF-08	User Input 2-B	1D20
US-01	User Seq. Con	0 : Stop	UF-04	User Input 1-C	0000	UF-09	User Input 2-C	0000
			UF-05	User Output 1	-	UF-10	User Output 2	-

Kod	Nazwa	Ustawienie
US-31	Void Para 1	0
US-32	Void Para 2	200

■ 2-2. Test

- Ustaw US-01 (User Seq Con) na 1(Run).
- Sprawdź czy częstotliwość zwiększa się o 4Hz co każdą sekundę

2. Proste przykłady

Przykład 2. MPYDIV (operacja złożona mnożenia i dzielenia. Wynik końcowy jako czas hamowania dla funkcji JOG)

■ **3-1. Lista kontrolna:**

- Wyjście w postaci: $= (A \times B) / C$.
- Jeżeli C wynosi 0x0000, na wyjściu wynik operacji: (AXB).
- Jeżeli konfiguracja bloku jest nieprawidłowa, wyjście User Output@ wynosi -1.

■ **3-2. Ustawienie parametrów:**

- US-31~33 Void Para1~3 → Ustawienie wartości stałych całkowitych.
- UF-2~4 User Input 1-A~C → adresy komunikacyjne stałych Void Para1~3.

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	5 : 0.02 s
US-11	Link User Out 1	110D
US-31	Void Para 1	100
US-32	Void Para 2	100
US-33	Void Para 3	200

Kod	Nazwa	Ustawienie
US-01	User Seq. Con	0 : Stop
UF-01	User Func 1	8 : MpyDiv
UF-02	User Input 1-A	1D20
UF-03	User Input 1-B	1D20
UF-04	User Input 1-C	1D21

2. Proste przykłady

Przykład 2. MPYDIV (operacja złożona mnożenia i dzielenia. Wynik końcowy jako czas hamowania dla funkcji JOG)

- **3-3. Test**
 - Ustaw US-01 (User Seq Con) na 1(Run).
 - Sprawdź czy UF-05 wynosi 50.
 - Sprawdź czy Jog Dec time wynosi 5sekund.

2. Proste przykłady

Przykład 4. COMPARE-GT (operacja porównania, większy niż. Wynik w postaci częstotliwości kluczenia. Jeśli $A > B$ to wynikiem jest C, które nadpisuje aktualną wartość częstotliwości kluczenia).

- **4-1. Lista kontrolna:**
 - Wyjście w postaci: = jeśli ($A > B$) to C w innym wypadku 0.
 - Jeśli C wynosi 0x0000 i warunek jest spełniony, na wyjściu 1 (prawda)).
 - Jeżeli konfiguracja bloku jest nieprawidłowa, wyjście User Output@ wynosi -1.
- **4-2. Ustawienie parametrów:**
 - US-31~33 Void Para1~3 → Ustawienie wartości stałych całkowitych.
 - UF-2~4 User Input 1-A~C → adresy komunikacyjne stałych Void Para1~3.

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	1 : 0.02 s
US-11	Link User Out 1	1404
US-31	Void Para 1	1000
US-32	Void Para 2	500
US-33	Void Para 3	50

Kod	Nazwa	Ustawienie
US-01	User Seq. Con	0 : Stop
UF-01	User Func 1	10 : Compara-GT
UF-02	User Input 1-A	1D1F
UF-03	User Input 1-B	1D20
UF-04	User Input 1-C	1D21

2. Proste przykłady

Przykład 4. COMPARE-GT (operacja porównania, większy niż. Wynik w postaci częstotliwości kluczenia. Jeśli $A > B$ to wynikiem jest C, które nadpisuje aktualną wartość częstotliwości kluczenia).

- **4-3. Test**
 - Ustaw US-01 (User Seq Con) na 1(Run).
 - Sprawdź czy UF-05 wynosi 50.
 - Sprawdź czy częstotliwość kluczenia wynosi 5kHz

2. Proste przykłady

Przykład 5. TIMER (obsługa timera, wyjście w postaci zmiennej częstotliwości zadanej).

■ **5-1. Lista kontrolna:**

- Wyjście = Loop Time(US-02) x A(Max Loop)
- Wejście-A: Max ilość pętli. Jeśli A jest ujemne, zostanie rozpoznane jako 0.
- Wejście -B: Start/Stop timera, jeśli B = 1, timer stopuje. Jeśli = 0, timer startuje.
- Wejście -C: wybór trybu. Jeśli C wynosi 0x0000, zostanie rozpoznane jako 0.

■ **5-2. Ustawienie parametrów:**

- US-31~33 Void Para1~3 → Ustawienie wartości stałych całkowitych.
- UF-2~4 User Input 1-A~C → adresy komunikacyjne stałych Void Para1~3

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	1 : 0.02 s
US-11	Link User Out 1	1101
US-31	Void Para 1	1000
US-32	Void Para 2	0
US-33	Void Para 3	1

Kod	Nazwa	Ustawienie
US-01	User Seq. Con	0 : Stop
UF-01	User Func 1	14 : Timer
UF-02	User Input 1-A	1D1F
UF-03	User Input 1-B	1D20
UF-04	User Input 1-C	1D21

2. Proste przykłady

Przykład 5. TIMER (obsługa timera, wyjście w postaci zmiennej częstotliwości zadanej).

- **5-3. Test**
 - Ustaw US-01 (User Seq Con) na 1(Run).
 - Sprawdź czy UF-05 zmienia się pomiędzy 0~1000.
 - Sprawdź czy częstotliwość zadana zmienia się pomiędzy 0~10Hz.

2. Proste przykłady

Przykład 6. ANDOR (operacja złożona z bramek logicznych And i OR, wyjście w postaci 16 bitów).

■ **6-1. Lista kontrolna:**

- Wyjście w postaci: $= ((A \& B) \text{ OR } C)$
- Wejścia: A, B, C na wyjściu słowo 16 bitowe (kod binarny).
- Jeżeli C wynosi 0x0000, operacja tylko na A, B.

■ **6-2. Ustawienie parametrów:**

- US-31~33 Void Para1~3 → Ustawianie wartości stałych całkowitych.
- UF-2~4 User Input 1-A~C → Adresy komunikacyjne stałych Void Para1~3.

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	1 : 0.02 s
US-31	Void Para 1	15
US-32	Void Para 2	12
US-33	Void Para 3	14

Kod	Nazwa	Ustawienie
US-01	User Seq. Con	0 : Stop
UF-01	User Func 1	19 : ANDOR
UF-02	User Input 1-A	1D1F
UF-03	User Input 1-B	1D20
UF-04	User Input 1-C	1D21

■ **6-3. Test:**

- Ustaw US-01 (User Seq Con) na 1(Run).
- Sprawdź czy UF-05 iwynosi 000E (0000 0000 0000 1110).

2. Proste przykłady

Przykład 7. SWITCH (operacja na funkcji SWITCH, wyjście w postaci zmienionej częstotliwości kluczowania).

■ **7-1. Lista kontrolna:**

- Wyjście w postaci: = jeśli (A) to B inaczej C.
- User Input-A wynosi 1(Prawda) lub 0 (Fałsz).
- Jeśli A wynosi 1, na wyjściu B. W innym wypadku, na wyjściu C.
- Wejście B jest zawsze większe równe C.

■ **7-2. Ustawienie parametrów:**

- US-31~33 Void Para1~3 → Ustawienie wartości stałych
- UF-2~4 User Input 1-A~C → Adresy komunikacyjne stałych Void Para1~3

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	1 : 0.02 s
US-11	Link User Out 1	1404
US-31	Void Para 1	1
US-32	Void Para 2	120
US-33	Void Para 3	30

Kod	Nazwa	Ustawienie
US-01	User Seq. Con	0 : Stop
UF-01	User Func 1	20 : SWITCH
UF-02	User Input 1-A	1D1F
UF-03	User Input 1-B	1D20
UF-04	User Input 1-C	1D21

2. Proste przykłady

Przykład 7. SWITCH (operacja na funkcji SWITCH, wyjście w postaci zmienionej częstotliwości kluczenia).

- **7-3. Test:**
 - Ustaw US-01 (User Seq Con) na 1(Run).
 - Sprawdź czy UF-05 wynosi 120.
 - Sprawdź czy częstotliwość kluczenia zmienia się na wartość 12kHz.

2. Proste przykłady

Przykład 8. BITTEST (operacja testowania bitów)

■ 8-1. Lista kontrolna:

- Wejście A wskazuje na testowany bajt.
- Wejście B wskazuje testowany bit.
- Operacja nie wykorzystuje wejścia C.

■ 8-2. Ustawienie parametrów:

- US-31~32 Void Para1~2 → Ustawianie wartości stałych całkowitych
- UF-2~3 User Input 1-A~C → Adresy komunikacyjne stałych Void Para1~2.

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	1 : 0.02 s
US-31	Void Para 1	15
US-32	Void Para 2	5

Kod	Nazwa	Ustawienie
US-01	User Seq. Con	0 : Stop
UF-01	User Func 1	21 : BITTEST
UF-02	User Input 1-A	1D1F
UF-03	User Input 1-B	1D20

■ 8-3. Test

- Ustaw US-01 (User Seq Con) na 1(Run).
- Sprawdź czy UF-05 wynosi 0.

2. Proste przykłady

Przykład 9. LIMIT (funkcja limitu, wyjście w postaci limitu górnego i dolnego częstotliwości zadanej)

■ **9-1. Lista kontrolna**

- Wejście A jest wartością ograniczaną.
- Wejście B jest granicą górną wartości A.
- Wejście C jest granicą dolną wartości A.
- Parametr B musi być większy równy C.

■ **9-2. Ustawienie parametrów:**

- US-31~32 Void Para1~2 → Ustawianie wartości stałych.
- UF 3~4 User Input 1-A~C → adresy komunikacyjne wartości stałych Void Para1~2.

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	1 : 0.02 s
US-11	Link Uset Out 1	1101
US-31	Void Para 1	4000
US-32	Void Para 2	2000

Kod	Nazwa	Ustawienie
US-01	User Seq. Con	0 : Stop
UF-01	User Func 1	15 : LIMIT
UF-02	User Input 1-A	1101
UF-03	User Input 1-B	1D1F
UF-04	User Input 1-C	1D20

2. Proste przykłady

Przykład 9. LIMIT (funkcja limitu, wyjście w postaci limitu górnego i dolnego częstotliwości zadanej)

- **9-3. Test**
 - Ustaw US-01 (User Seq Con) na 1(Run).
 - Zmieniaj losowo wartość częstotliwości zadanej w iUS-05 i sprawdź czy zmiana ma miejsce pomiędzy górną i dolną nastawioną granicą (20 i 40 Hz).

2. Proste przykłady

Przykład 10. LOWPASSFILTER (filtr dolnoprzepustowy)

■ 10-1. Lista kontrolna:

- Wyjście w postaci A po osiągnięciu stałej czasowej filtra B.
- Wejście B jest wzmacnieniem (wyjście A gdy filtr uzyska BxUS-02 (czas po którym A uzyskuje 63,3% wartości)).
- Wejście C mówi o działaniu filtra. Jeżeli wynosi 0 – rozpoczęcie operacji.

■ 10-2. Ustawienie parametrów:

- US-31~33 Void Para1~3 → Ustawianie wartości stałych całkowitych.
- UF-2~4 User Input 1-A~C → adresy komunikacyjne wartości stałych Void Para1~3

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	1 : 0.02 s
US-11	Link Uset Out 1	1101
US-31	Void Para 1	600
US-32	Void Para 2	100
US-33	Void Para 3	0

Kod	Nazwa	Ustawienie
UF-01	User Func 1	24 : LOWPASS-FILER
UF-02	User Input 1-A	1D1F
UF-03	User Input 1-B	1D20
UF-04	User Input 1-C	1D21
US-01	User Seq. Con	0 : Stop

2. Proste przykłady

Przykład 10. LOWPASSFILTER (filtr dolnoprzepustowy)

- **10-3. Test:**
 - Ustaw US-01 (User Seq Con) na 1(Run).
 - Sprawdź czy na stawiony czas: US-02(0.02Sec) X UF-03(100) powoduje osiągnięcie 63,3% częstotliwości zadanej. Jeżeli częstotliwość zadana wynosi 60Hz, po 2 sekundach wartość ta powinna osiągnąć poziom $63,3\% \times 60\text{Hz} = 38,2\text{Hz}$.

2. Proste przykłady

Przykład 11. UPCOUNT (zliczanie w przód)

■ 11-1. Lista kontrolna:

- Po otrzymaniu wyzwolenia na wejściu (A), wyjście zliczane w przód zgodnie z warunkami C.
- Wejście A jest wyzwalające (wartość 0 lub 1).
- Wejście B startuje operacje. Jeżeli B=0 , start operacji. Jeśli B=1, stop operacji, na wyjściu 0.
- Wejście C określa warunek. Jeśli C=0, zliczanie po zmianie A z 0 na 1. Jeśli C=1 zliczanie po zmianie A z 1 na 0. Jeśli C=2 zliczanie po dowolnej zmianie A (z 0 na 1 lub z 1 na 0).

■ 11-2. Ustawienie parametrów:

- US-31~33 Void Para1~3 → Ustawianie wartości stałych, całkowitych.
- UF-2~4 User Input 1-A~C → Adresy komunikacyjne wartości stałych Para1~3

Kod	Nazwa	Ustawienie
APP-02	User Seq. En	1 : Yes
US-02	User Seq. Loop Time	1 : 0.02 s
US-31	Void Para 1	10
US-32	Void Para 2	0
US-33	Void Para 3	2

Kod	Nazwa	Ustawienie
US-01	User Seq. Con	0 : Stop
UF-01	User Func 1	27 : UPCOUNT
UF-02	User Input 1-A	1D1F
UF-03	User Input 1-B	1D20
UF-04	User Input 1-C	1D21

2. Proste przykłady

Przykład 11. UPCOUNT (zliczanie w przód)

- **11-3. Test:**
 - Ustaw US-01 (User Seq Con) na 1(Run).
 - Sprawdź czy US-05 wzrasta, po zmianie wartości w US-31 (Void Para1).

2. Proste przykłady

Lista ważniejszych adresów komunikacyjnych ze skalą dla falownika S100)

Adres (Hex)	Parametr	Jednostka	Skala
1101	Target Freq.	Hz	0.01
1103	Acc Time	Sec	0.1
1104	Dec Time	Sec	0.1
1505	Analog Input 1	%	0.01
1606	Analog Output 1	%	0.1
155A	Multi-func. Input Status	Bit	-
0385	Virtual Multi-func. Input	Bit	-

Adres (Hex)	Parametr
1E19	User Output 5
1E1E	User Output 6
1E23	User Output 7
1E28	User Output 8
1E2D	User Output 9
1E32	User Output 10
1E37	User Output 11

Adres (Hex)	Parametr
1D1F	Void Para 1
1D20	Void Para 2
1D21	Void Para 3
1D22	Void Para 4
1D23	Void Para 5
1D24	Void Para 6
1D25	Void Para 7



ANIRO

2. Proste przykłady

Adres (Hex)	Parametr	Jednostka	Skala
1629	Multi-func. Output Status	Bit	-
1404	Carrier Freq.	kHz	0.1
0009	Output Current	A	0.1
000A	Output Freq.	Hz	0.01
000B	Output Volt.	V	1
000C	DC-link Volt.	V	1
000D	Output Power	kW	0,1
000E	Operation Status	-	-
000F	Fault Info.	-	-

Adres (Hex)	Parametr
1D26	Void Para 8
1D27	Void Para 9
1D28	Void Para 10
1D29	Void Para 11
1D30	Void Para 12
1E05	User Output 1
1E0A	User Output 2
1E0F	User Output 3
1E14	User Output 4

Adres (Hex)	Parametr
1E3C	User Output 12
1E41	User Output 13
1E46	User Output 14
1E4B	User Output 14
1E50	User Output 16
1E55	User Output 17
1E0A	User Output 18
1E55	User Output 17
1E0A	User Output 18



ANIRO Sp. z o.o.

ul. B. Chrobrego 64, 87- 100 Toruń
tel. 56 657 63 63, aniro@aniro.pl

www.aniro.pl



Dziękuję.