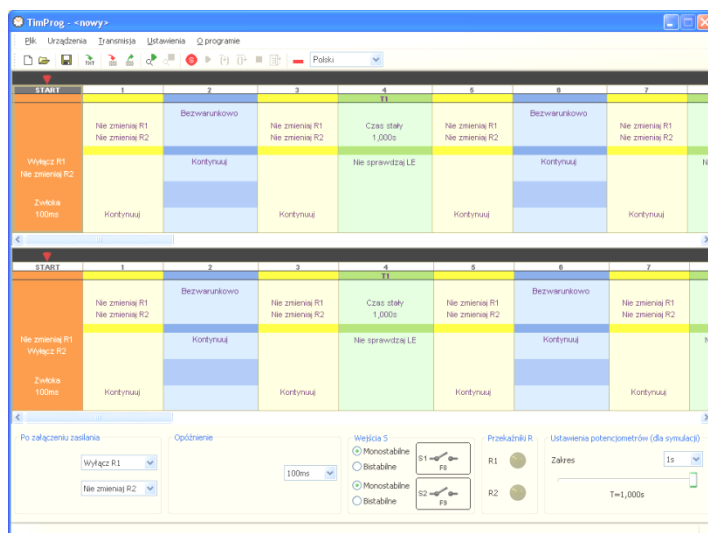




Instrukcja obsługi oprogramowania *TimProg* dla uniwersalnych przekaźników czasowych MTR17-TPA/TPD-U240-XXX



Spis treści

1. Wprowadzenie	3
2. Instalacja.....	4
3. Zasoby programowe.....	6
3.1. Zasoby programowe TPA.....	6
3.2. Zasoby programowe TPD	8
3.3. Zasady definiowania warunku logicznego LE	11
4. Zasady definiowania i analizy funkcji czasowej.....	12
5. Zapis i odczyt konfiguracji	19
6. Ładowanie konfiguracji do przekaźnika	20
7. Zabezpieczenie dostępu do przekaźnika hasłem	21
8. Podgląd bieżącego stanu pracy przekaźnika	22
9. Programowa symulacja pracy przekaźnika	23
10. Eksport ustawień	25
11. Przykładowe funkcje.....	26

1. Wprowadzenie

Przekaźniki programowalne MTR17-TPA-U240-XXX oraz MTR17-TPD-U240-205 są uniwersalnymi układami czasowymi, w których realizowana funkcja czasowa jest definiowana przez użytkownika i ładowana do przekaźnika programowalnego przy pomocy aplikacji *TimProg*. Elastyczna struktura programowa pozwala na prostą i szybką implementację zarówno standardowych, jak i nietypowych funkcji czasowych, umożliwiając budowę układów sterowania dostosowanych do indywidualnych potrzeb odbiorcy. Zastosowanie interfejsu USB pozwala na programowanie przekaźnika przy pomocy standardowego kabla miniUSB, ułatwiając i minimalizując tym samym koszty uruchomienia urządzeń końcowych. Opcje podglądu bieżącego stanu przekaźnika oraz programowa symulacja pracy ułatwiają definiowanie i uruchamianie funkcji czasowych.

Przekaźnik MTR17-TPD-U240-205 posiada dwa niezależne wejścia sterujące S1 i S2, dwa przekaźniki wykonawcze R1 i R2 oraz logiczną strukturę programową pozwalającą na zdefiniowanie dwóch różnych funkcji czasowych wykonujących się równolegle. Rozwiązanie takie umożliwia implementację dwóch przekaźników czasowych umieszczonych w jednej obudowie, obniżając koszty budowy urządzeń końcowych i redukując ilość zajmowanego miejsca w szafie sterowniczej.

W zależności od potrzeb użytkownika, wejścia sterujące S1, S2 oraz przekaźniki wykonawcze R1, R2 mogą być wykorzystane w dwóch różnych lub wyłącznie w jednym programie czasowym. Istnieje możliwość wzajemnego powiązania działania obu programów, tak aby działanie jednego zależało od stanu przekaźników kontrolowanych przez program drugi.

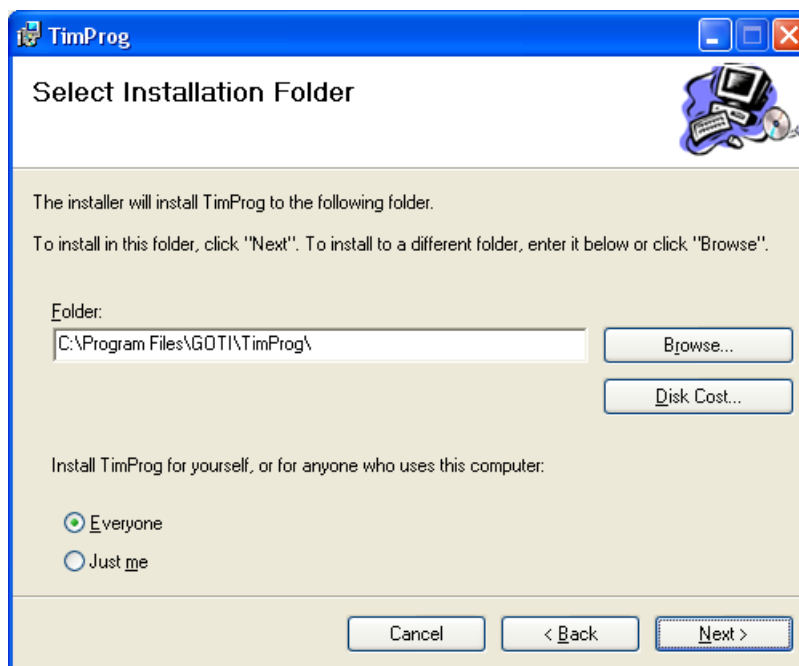
Przekaźnik MTR17-TPD-U240-205 stanowi tanią alternatywę dla prostych aplikacji, w których stopień komplikacji logiki sterującej wymagał stosowania przekaźników programowalnych.

2. Instalacja

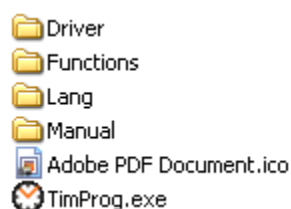
W celu zainstalowania oprogramowania *TimProg* należy pobrać plik instalacyjny z rozszerzeniem *.msi ze strony producenta:

<http://www.dobry-czas.pl/Produkty/Przekazniki-czasowe-programowalne/Przekaznik-MTR17-TPD-U240-205>

Po uruchomieniu pliku pojawi się okno dialogowe, w którym w polu *Folder* możliwy jest wybór docelowej ścieżki instalacji oprogramowania.



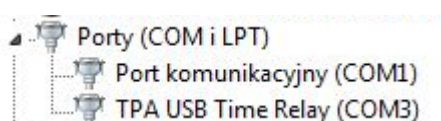
Po zakończeniu instalacji w podanej lokalizacji na dysku widoczny będzie plik programu *TimProg* oraz podkatalogi *Driver*, *Functions*, *Lang* oraz *Manual*.



Instalacja sterownika

Przy pierwszym podłączeniu przekaźnika do komputera, system operacyjny rozpocznie poszukiwanie sterownika urządzenia USB. Należy ręcznie wybrać plik o nazwie *TPx.inf* znajdujący się w podkatalogu *Driver* w miejscu zainstalowania aplikacji.

W wyniku instalacji sterownika do listy urządzeń zostanie dodany wirtualny port szeregowy *TPA USB Time Relay* o numerze zależnym od bieżącej konfiguracji komputera.



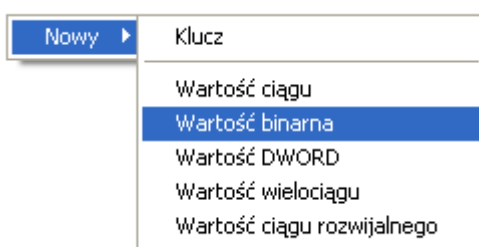
Po pomyślnym zainstalowaniu sterownika oprogramowanie *TimProg* jest gotowe do pracy.

Instalacja sterownika do programowania seryjnego

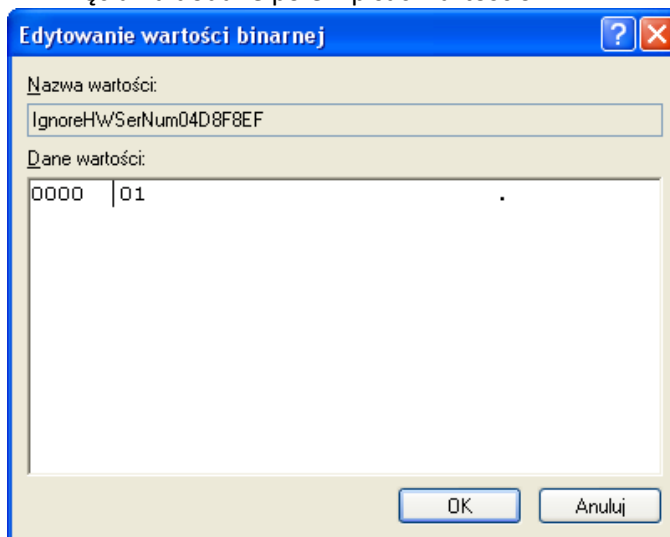
Kontroler USB zastosowany w przekaźniku programowalnym zawiera unikalny identyfikator związany z portem USB, w skład którego wchodzi pola VID, PID oraz numer seryjny. Z tego względu, każde podłączenie kolejnego przekaźnika do komputera powoduje wykrycie nowego urządzenia, co wymaga ponownego zainstalowania sterownika USB. O ile dla użytkownika stosującego pojedyncze sztuki nie stanowi to problemu, o tyle w przypadku programowania większej ilości przekaźników, każdorazowe instalowanie sterownika może stać się uciążliwe.

Możliwe jest takie skonfigurowanie systemu Windows, aby po pierwszej instalacji sterownika przekaźnika programowalnego, nie dokonywał on próby instalacji kolejnych sztuk. W tym celu należy:

1. Uruchomić edytor rejestru systemu Windows komendą *regedit*.
2. Przejść do HKEY_LOCAL_MACHINE→SYSTEM→CurrentControlSet→Control→UsbFlags.
3. Prawym klawiszem myszki wywołać komendę Nowy→Wartość binarna.



4. W nazwie nowego wpisu podać *IgnoreHWSerNum04D8F8EF* i nacisnąć Enter.
5. Po podwójnym kliknięciu na dodane pole wpisać wartość *01*.



Od tego momentu jednorazowe zainstalowanie sterownika przekaźnika programowalnego wystarczy do zaprogramowania dowolnej ilości sztuk wyrobu.

Sterownik przekaźników serii TPA i TPD jest identyczny. Jednorazowa instalacja pozwala na uzyskanie komunikacji z dowolnym typem przekaźnika.

3. Zasoby programowe

Programowalne przekaźniki czasowe w swojej strukturze programowej zawierają cztery rodzaje zasobów: pole *START*, pole *Kontroli stanu R*, *Blok warunkowy* i *Blok czasowy*. W wyniku analizy ustawień wymienionych zasobów przekaźnik realizuje zdefiniowaną funkcję odmierając czas, kontrolując stan wejść S oraz sterując przekaźnikami wykonawczymi R.

W wersji TPA możliwe jest zdefiniowanie wyłącznie jednej funkcji czasowej, sterowanej jednym wejściem S oraz kontrolującej jeden przekaźnik wykonawczy R. Wersja TPD jest podwójnym przekaźnikiem czasowym, posiadającym możliwość definiowania dwóch niezależnych programów, wyposażona w dwa wejścia S1 i S2 oraz dwa przekaźniki wykonawcze R1 i R2.

3.1. Zasoby programowe TPA

Zasoby programowe przekaźników serii TPA przedstawiono w tabeli poniżej.

Zasoby programowe TPA

Zasób	Ilość	Opis
START	1	<ol style="list-style-type: none"> Umożliwia zdefiniowanie stanu początkowego przekaźnika wykonawczego R po załączeniu napięcia zasilającego. <ul style="list-style-type: none"> R=0 R=1 Określa dodatkowe opóźnienie po załączeniu zasilania, umożliwiające detekcję stanu sygnału wejściowego S przed rozpoczęciem realizacji zaprogramowanej funkcji (patrz „minimalny czas trwania impulsu” w danych technicznych). <ul style="list-style-type: none"> 0ms (brak opóźnienia) 50ms 100ms 150ms
Kontrola stanu R	18	<ol style="list-style-type: none"> Definiuje operację na przekaźniku wykonawczym. <ul style="list-style-type: none"> Nie zmieniaj stanu R Wyłącz R Włącz R Neguj R (zmiana stanu na przeciwny) Po wykonaniu operacji na przekaźniku określa następny realizowany etap programu. <ul style="list-style-type: none"> Kontynuuj (kolejny etap) Skocz do... (wykonuje skok do podanego etapu)
Blok warunkowy	9	<ol style="list-style-type: none"> Sprawdza ustawiony warunek i wykonuje skok do podanego etapu. <ul style="list-style-type: none"> Bezwarunkowo (wykonuje operację skoku bez sprawdzania warunku) Czekaj na S=0 Czekaj na S=1 Czekaj na zmianę S z 1 na 0 Czekaj na zmianę S z 0 na 1 Czekaj na dowolną zmianę S Jeżeli S=0 Jeżeli S=1 Jeżeli R=0 Jeżeli R=1 Definiuje etap lub etapy, do których nastąpi skok. Dla czterech ostatnich warunków należy podać dwa miejsca skoków – pierwszy jest istotny, gdy warunek jest spełniony, drugi dla sytuacji przeciwnej.
Blok czasowy	8	<ol style="list-style-type: none"> Pozwala na ustawienie odmierzanego czasu. <ul style="list-style-type: none"> Zakresy: 1s, 10s, 1m, 10m, 1h, 10h, 100h Płynna regulacja 0,1...1,0 wartości zakresu Możliwość pobrania ustawień z potencjometrów zamontowanych na panelu przekaźnika W trakcie odmierzania czasu możliwa jest kontrola styku S. Spełnienie określonego warunku powoduje natychmiastowe zatrzymanie licznika czasu i skok do etapu ustawionego w Bloku czasowym. <ul style="list-style-type: none"> Nie sprawdzaj styku S Skocz jeżeli S=0 Skocz jeżeli S=1 Skocz jeżeli zmiana S z 0 na 1 Skocz jeżeli zmiana S z 1 na 0 Skocz jeżeli dowolna zmiana S Po całkowitym zakończeniu odmierzania nastawionego czasu program przechodzi do kolejnego etapu znajdującego się po bieżącym bloku czasowym.

START Pole *START* jest pierwszym blokiem analizowanym po załączeniu zasilania, który definiuje stan początkowy przekaźnika wykonawczego R. Ustawienie parametrów *Wyłącz R* lub *Włącz R* powoduje, że praca programu rozpocznie się odpowiednio od wyłączonego lub załączonego wyjścia przekaźnika.

W polu *START* definiowane jest także wstępne opóźnienie, które może być potrzebne na analizę stanu wejścia sterującego S. Pomiar napięcia na zasilaniu DC maksymalnie 45ms, a dla AC – 90ms. Oznacza to, że po podłączeniu zasilania przez ten czas stan S będzie równy 0 nawet wtedy, gdy do zasilania S

podane jest napięcie. Jeżeli z pewnych przyczyn stan S jest istotny od początku działania programu, należy ustawić wstępne opóźnienie, które pozwoli na pomiar rzeczywistej wartości S przed rozpoczęciem analizy właściwej funkcji czasowej. Jest to szczególnie ważne w sytuacji, gdy program ma reagować na zmianę stanu S z 0 na 1. Zmiana ta może być błędnie wykryta nie w wyniku pojawienia się napięcia na zasisku S, ale na skutek aktualizacji pomiaru. Możliwe wartości opóźnienia to 0ms (brak zwłoki), 50ms, 100ms i 150ms.

Kontrola stanu R

Pole *Kontroli stanu przekaźnika R* służy do załączania i wyłączenia wyjścia przekaźnika. Możliwe są cztery ustawienia: *Nie zmieniaj stanu R*, *Wyłącz R*, *Włącz R* i *Neguj R*. Opcja *Neguj R* zamienia stan wyjścia na przeciwny. Pole kontroli posiada również miejsce skoku, czyli numer etapu, do którego program ma przejść po wykonaniu operacji na przekaźniku. Wartość *Kontynuuj* powoduje przejście do następnego etapu, natomiast opcja *Skocz do* pozwala na dalszą analizę programu od dowolnego podanego miejsca. Czas analizy pola *Kontroli stanu R* jest pomijalnie mały.

Blok warunkowy

Blok warunkowy służy do sprawdzenia wybranego warunku i realizacji skoku do zdefiniowanego etapu. Możliwe są następujące ustawienia:

- **Bezwarunkowo** – warunek ten oznacza, że program zawsze wykona skok do podanego miejsca. Ustawienie miejsca skoku na *Kontynuuj* jest równoznaczne z przejściem do etapu o numerze o 1 większym od bieżącego.
- **Czekaj na S=0** – warunek ten zatrzymuje wykonywanie programu do momentu, aż stan wejścia sterującego S przyjmie wartość logiczną 0. Po wykryciu stanu 0 następuje skok do etapu zdefiniowanego przez użytkownika.
- **Czekaj na S=1** – warunek ten zatrzymuje wykonywanie programu do momentu, aż stan wejścia sterującego S przyjmie wartość logiczną 1. Po wykryciu stanu 1 następuje skok do etapu zdefiniowanego przez użytkownika.
- **Czekaj na zmianę S z 0 na 1** – warunek powoduje zatrzymanie wykonywania programu do momentu, aż zostanie wykryta zmiana stanu wejścia S z 0 na 1. Jest to równoznaczne z podaniem napięcia na wejście S. Po wykryciu zmiany następuje skok do etapu zdefiniowanego przez użytkownika.
- **Czekaj na zmianę S z 1 na 0** – warunek powoduje zatrzymanie wykonywania programu do momentu, aż zostanie wykryta zmiana stanu wejścia S z 1 na 0. Jest to równoznaczne z odłączeniem napięcia od wejście S. Po wykryciu zmiany następuje skok do etapu zdefiniowanego przez użytkownika.
- **Czekaj dowolną zmianę S** – warunek powoduje zatrzymanie wykonywania programu do momentu, aż zostanie wykryta zmiana stanu wejścia S z 0 na 1 lub z 1 na 0. Po wykryciu zmiany następuje skok do etapu zdefiniowanego przez użytkownika.
- **Jeżeli S=0** – warunek powoduje sprawdzenie stanu wejścia sterującego S. W zależności od bieżącej wartości S program może przejść do jednego z dwóch podanych etapów.
- **Jeżeli S=1** – warunek powoduje sprawdzenie stanu wejścia sterującego S. W zależności od bieżącej wartości S program może przejść do jednego z dwóch podanych etapów.
- **Jeżeli R=0** – warunek powoduje sprawdzenie stanu przekaźnika wykonawczego R. W zależności od bieżącej wartości R program może przejść do jednego z dwóch podanych etapów.
- **Jeżeli R=1** – warunek powoduje sprawdzenie stanu przekaźnika wykonawczego R. W zależności od bieżącej wartości R program może przejść do jednego z dwóch podanych etapów.

Blok czasowy

Blok czasowy jest elementem służącym do odmierzenia czasu. Analiza *Bloku czasowego* oznacza, że program zatrzyma się na nastawiony czas, po odmierzeniu którego przejdzie zawsze do kolejnego etapu znajdującego się bezpośrednio za bieżącym *Blokiem czasowym*. Czas może być zdefiniowany w programie poprzez ustawienie źródła czasu na *Czas stały*. Wówczas parametrami bloku są *zakres*, który może przyjmować wartości *1s, 10s, 1m, 10m, 1h, 10h, 100h* oraz liniowy przelicznik przyjmujący wartości od 0,1 do 1,0. Przykładowo opóźnienie wynoszące 30 minut może być ustawione poprzez podanie zakresu *1h* oraz liniowego przelicznika 0,5, co daje $1h \cdot 0,5 = 30min$. Przy konfiguracji opóźnienia program wyświetla bieżący czas ustawiony przez użytkownika. Na panelu czołowym przekaźnika znajdują się także dwa potencjometry służące do ustawiania czasu opóźnienia. Jeżeli w programie jako źródło czasu podamy *Potencjometry*, program ustawi opóźnienie zależne od aktualnego ustawienia zakresu oraz skali liniowej na panelu czołowym.

W trakcie pomiaru czasu możliwa jest analiza stanu wejścia sterującego *S* poprzez sprawdzenie warunku podanego przez użytkownika. Jeżeli warunek zostanie spełniony, następuje przerwanie odmierzenia czasu i natychmiastowy skok do etapu podanego w konfiguracji *Bloku czasowego*. Możliwe jest określenie następujących warunków:

- **Nie sprawdzaj S** – warunek powoduje pominięcie analizy stanu wejścia *S*. Przejście do kolejnego etapu możliwe jest wyłącznie po zakończeniu odmierzenia nastawionego czasu.
- **Jeżeli S=0** – w przypadku wykrycia stanu logicznego 0 na wejściu sterującym *S* następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.
- **Jeżeli S=1** – w przypadku wykrycia stanu logicznego 1 na wejściu sterującym *S* następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.
- **Jeżeli zmiana S z 0 na 1** – w przypadku wykrycia zmiany stanu logicznego z 0 na 1 na wejściu sterującym *S* następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.
- **Jeżeli zmiana S z 1 na 0** – w przypadku wykrycia zmiany stanu logicznego z 1 na 0 na wejściu sterującym *S* następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.
- **Jeżeli dowolna zmiana S** – w przypadku wykrycia zmiany stanu logicznego z 0 na 1 lub z 1 na 0 na wejściu sterującym *S* następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.

3.2. Zasoby programowe TPD

Zasoby programowe przekaźników serii TPD są znacznie rozszerzone w porównaniu z wersją TPA, z tego też względu zostały opisane w oddzielnym podrozdziale.

Zasoby programowe TPD (zasoby są identyczne dla każdego z dwóch dostępnych programów)

Zasób	Ilość	Opis				
START	1	1. Umożliwia zdefiniowanie stanu początkowego przekaźników wykonawczych R1 oraz R2 po załączeniu napięcia zasilającego. <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid #ccc; padding: 2px;"> <ul style="list-style-type: none"> • Wyłącz Rx • Nie zmieniaj Rx </td> <td style="border: 1px solid #ccc; padding: 2px;"> <ul style="list-style-type: none"> • Włącz Rx </td> </tr> <tr> <td colspan="2" style="border: none; padding: 2px;">* Rx oznacza dowolny przekaźnik R1 lub R2</td> </tr> </table>	<ul style="list-style-type: none"> • Wyłącz Rx • Nie zmieniaj Rx 	<ul style="list-style-type: none"> • Włącz Rx 	* Rx oznacza dowolny przekaźnik R1 lub R2	
		<ul style="list-style-type: none"> • Wyłącz Rx • Nie zmieniaj Rx 	<ul style="list-style-type: none"> • Włącz Rx 			
* Rx oznacza dowolny przekaźnik R1 lub R2						
		2. Określa dodatkowe opóźnienie po załączeniu zasilania, umożliwiające detekcję stanu sygnałów wejściowych Sx przed rozpoczęciem realizacji zaprogramowanej funkcji (patrz „minimalny czas trwania impulsu” w danych technicznych). <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid #ccc; padding: 2px;"> <ul style="list-style-type: none"> • 0ms (brak opóźnienia) • 100ms </td> <td style="border: 1px solid #ccc; padding: 2px;"> <ul style="list-style-type: none"> • 50ms • 150ms </td> </tr> </table>	<ul style="list-style-type: none"> • 0ms (brak opóźnienia) • 100ms 	<ul style="list-style-type: none"> • 50ms • 150ms 		
<ul style="list-style-type: none"> • 0ms (brak opóźnienia) • 100ms 	<ul style="list-style-type: none"> • 50ms • 150ms 					

Kontrola stanu R

18

- Definiuje operację na przekaźniku wykonawczym.
 - Nie zmieniaj stanu Rx
 - Włącz Rx
 - Wyłącz Rx
 - Neguj Rx (zmiana stanu na przeciwny)
- Po wykonaniu operacji na przekaźniku określa następnym realizowany etap programu.
 - Kontynuuj (kolejny etap)
 - Skocz do... (wykonuje skok do podanego etapu)

Blok warunkowy

9

- Sprawdza ustawiony warunek i wykonuje skok do podanego etapu.
 - Bezwarunkowo (wykonuje operację skoku bez sprawdzania warunku)
 - Czekaj na LE=1
 - Czekaj na zmianę LE z 1 na 0
 - Jeżeli LE=0
 - Czekaj na LE=0
 - Czekaj na zmianę LE z 0 na 1
 - Czekaj na dowolną zmianę LE
 - Jeżeli LE=1
- Definiuje etap lub etapy, do których nastąpi skok. Dla dwóch ostatnich warunków należy podać dwa miejsca skoków – pierwszy jest istotny, gdy warunek jest spełniony, drugi dla sytuacji przeciwnej.

LE (*Logical Expression*) jest wyrażeniem logicznym maksymalnie trzech zmiennych definiowanym przez użytkownika. Dostępne są operatory AND, OR i XOR oraz zestaw argumentów S1, ~S1, S2, ~S2, R1, ~R1, R2 i ~R2. Symbol ~ oznacza operator negacji. Bardziej szczegółowy opis wyrażań logicznych opisano w rozdziale 3.3.

Blok czasowy

8

- Pozwala na ustawienie odmierzanego czasu.
 - Zakresy: 1s, 10s, 1m, 10m, 1h, 10h, 100h
 - Możliwość pobrania ustawień z potencjometrów zamontowanych na panelu przekaźnika
 - Płynna regulacja 0,1...1,0 wartości zakresu
- W trakcie odmierzania czasu możliwa jest kontrola ustawionego warunku LE, którego spełnienie powoduje natychmiastowe zatrzymanie licznika czasu i skok do etapu ustawionego w Bloku czasowym.
 - Nie sprawdzaj LE
 - Jeżeli LE=1
 - Jeżeli zmiana LE z 1 na 0
 - Jeżeli LE=0
 - Jeżeli zmiana LE z 0 na 1
 - Jeżeli dowolna zmiana LE
- Po całkowitym zakończeniu odmierzania nastawionego czasu program przechodzi do kolejnego etapu znajdującego się po bieżącym bloku czasowym.

START

Pole START jest pierwszym blokiem analizowanym po załączeniu zasilania, który definiuje stan początkowy przekaźników wykonawczych R1 i R2 (w dalszej części określane jako Rx). Ustawienie parametrów *Wyłącz Rx* lub *Włącz Rx* powoduje, że praca programu rozpocznie się odpowiednio od wyłączonego lub załączonego wyjścia przekaźnika. Ponieważ przekaźnik TPD posiada dwa niezależnie działające programy zawierające pola START, możliwa jest sytuacja, w której wybrany program ingeruje w stan jednego przekaźnika Rx i nie powinien modyfikować stanu drugiego Rx. W tym celu wprowadzono opcję *Nie zmieniaj Rx*. Zadaniem użytkownika jest takie ustawienie pól START, aby dany przekaźnik Rx był ustawiany wyłącznie jeden raz (w wybranym programie). W innym przypadku wystąpi sytuacja logicznie niepoprawna.

W polu START definiowane jest także wstępne opóźnienie, którego funkcja została opisana w zasobach programowych przekaźników serii TPA.

Kontrola stanu R

Pole *Kontroli stanu przekaźników R* (R1 lub R2) służy do załączania i wyłączania ich wyjść. Możliwe są cztery ustawienia: *Nie zmieniaj stanu Rx*, *Wyłącz Rx*, *Włącz Rx* i *Neguj Rx*. Opcja *Neguj Rx* zamienia stan wyjścia na przeciwny. Każdy blok *Kontroli stanu R* może modyfikować oba dostępne przekaźniki, pozwalając na ich wykorzystanie wyłącznie w jednym lub w obu programach niezależnie. Należy zwrócić uwagę na taką konfigurację Rx, aby modyfikacja stanu przekaźnika w jednym z programów, nie wpłynęła w nieoczekiwany sposób na realizację funkcji czasowej wykonywanej w programie drugim. Pole kontroli posiada również miejsce skoku, czyli numer etapu, do którego program ma przejść po wykonaniu operacji na przekaźniku. Opcja ta została opisana w zasobach programowych przekaźników TPA, przy czym możliwy jest skok wyłącznie w ramach bieżącego programu.

Blok warunkowy

Blok warunkowy służy do sprawdzenia ustawionego warunku logicznego LE (*Logical Expression*). Po spełnieniu warunku następuje operacja skoku do wybranego etapu w ramach bieżącego programu. Możliwe są następujące ustawienia:

- **Bezwarunkowo** – warunek ten oznacza, że program zawsze wykona skok do podanego miejsca. Ustawienie miejsca skoku na *Kontynuuj* jest równoznaczne z przejściem do etapu o numerze o 1 większym od bieżącego.
- **Czekaj na LE=0** – warunek ten zatrzymuje wykonywanie programu do momentu, aż stan warunku przyjmie wartość logiczną 0. Po wykryciu stanu 0 następuje skok do etapu zdefiniowanego przez użytkownika.
- **Czekaj na LE=1** – warunek ten zatrzymuje wykonywanie programu do momentu, aż stan warunku przyjmie wartość logiczną 1. Po wykryciu stanu 1 następuje skok do etapu zdefiniowanego przez użytkownika.
- **Czekaj na zmianę LE z 0 na 1** – warunek powoduje zatrzymanie wykonywania programu do momentu, aż zostanie wykryta zmiana stanu warunku LE z 0 na 1. Po wykryciu zmiany następuje skok do etapu zdefiniowanego przez użytkownika.
- **Czekaj na zmianę LE z 1 na 0** – warunek powoduje zatrzymanie wykonywania programu do momentu, aż zostanie wykryta zmiana stanu warunku z 1 na 0. Po wykryciu zmiany następuje skok do etapu zdefiniowanego przez użytkownika.
- **Czekaj dowolną zmianę LE** – warunek powoduje zatrzymanie wykonywania programu do momentu, aż zostanie wykryta zmiana stanu warunku LE z 0 na 1 lub z 1 na 0. Po wykryciu zmiany następuje skok do etapu zdefiniowanego przez użytkownika.
- **Jeżeli LE=0** – warunek powoduje sprawdzenie stanu LE. W zależności od bieżącej wartości LE program może przejść do jednego z dwóch podanych etapów.
- **Jeżeli LE=1** – warunek powoduje sprawdzenie stanu LE. W zależności od bieżącej wartości LE program może przejść do jednego z dwóch podanych etapów.

Blok czasowy

Zasady ustawiania czasu są identyczne jak dla przekaźników serii TPA.

W trakcie pomiaru czasu możliwa jest analiza stanu warunku logicznego LE. Jeżeli warunek zostanie spełniony, następuje przerwanie odmierzenia czasu i natychmiastowy skok do etapu ustawionego w *Bloku czasowym*. Możliwe jest określenie następujących warunków:

- **Nie sprawdzaj LE** – warunek powoduje pominięcie analizy stanu warunku. Przejście do kolejnego etapu następuje wyłącznie po zakończeniu odmierzenia nastawionego czasu.
- **Jeżeli LE=0** – w przypadku wykrycia stanu wyrażenia LE=0 następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.
- **Jeżeli LE=1** – w przypadku wykrycia stanu wyrażenia LE=1 następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.
- **Jeżeli zmiana LE z 0 na 1** – w przypadku wykrycia zmiany stanu wyrażenia z 0 na 1 następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.
- **Jeżeli zmiana LE z 1 na 0** – w przypadku wykrycia zmiany stanu wyrażenia z 1 na 0 następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.
- **Jeżeli dowolna zmiana LE** – w przypadku wykrycia dowolnej zmiany stanu wyrażenia następuje natychmiastowe przerwanie odmierzenia czasu i skok do etapu ustawionego w *Bloku czasowym*.

3.3. Zasady definiowania warunku logicznego LE

Etapy *Bloków warunkowych* oraz *Bloków czasowych* wykorzystują wyrażenia logiczne LE (*Logical Expression*), które pozwalają na uzależnienie działania funkcji czasowych od stanów wejść sterujących Sx oraz stanów przekaźników wykonawczych Rx.

Wyrażenie logiczne składa się z maksymalnie trzech wymienionych niżej argumentów:

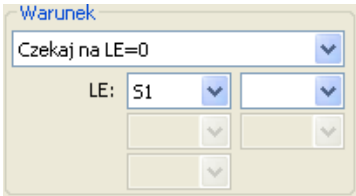
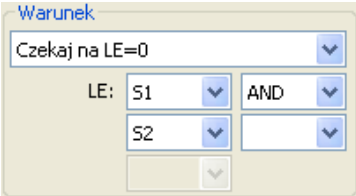
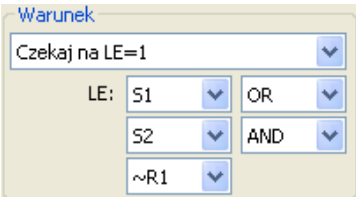
$$S1, \sim S1, S2, \sim S2, R1, \sim R1, R2 \text{ i } \sim R2,$$

przy czym znak „~” oznacza symbol negacji.

Wyrażenie logiczne LE może składać się z operacji logicznych AND, OR lub XOR, których tabele prawdy przedstawiono w tabeli poniżej.

Argument 1	Argument 2	Wynik AND	Wynik OR	Wynik XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

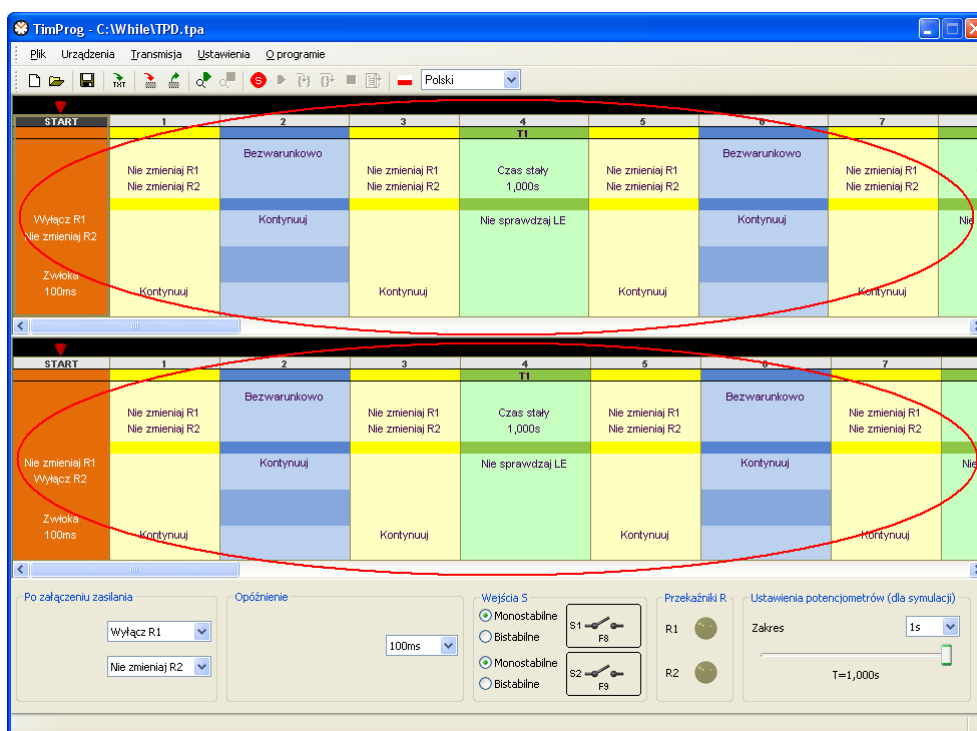
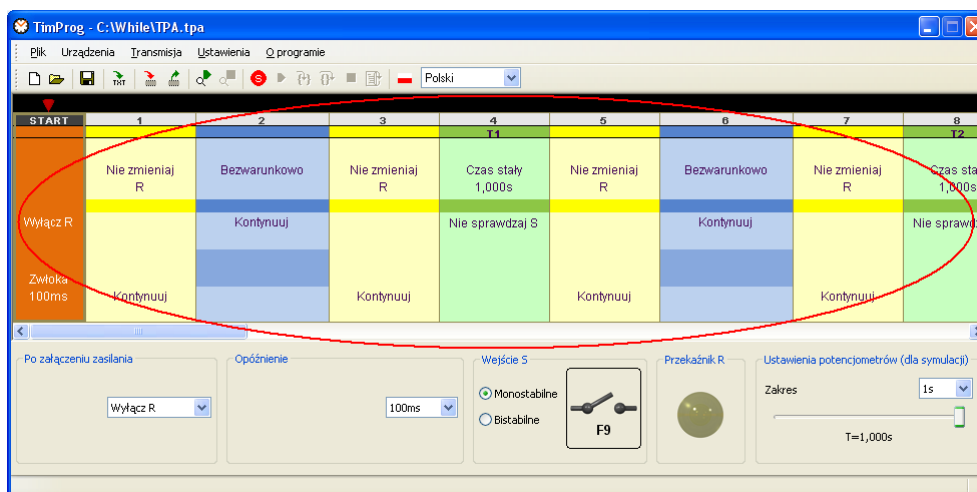
Operacje logiczne wykonywane są w kolejności ich zapisu i nie jest zachowywana kolejność działań wynikająca z ogólnie przyjętych zasad. W celu zdefiniowania pożądanego wyrażenia może wystąpić konieczność zmiany kolejności zapisu argumentów. Kilka przykładowych operacji logicznych zostało opisanych poniżej.

Ustawienie w programie	Wyrażenie logiczne LE
	<p>LE = S1</p> <p>Wyrażenie logiczne składa się z jednego argumentu S1. Podany warunek zostanie spełniony, jeżeli S1=0 (wejście S1 nieaktywne).</p>
	<p>LE = S1 AND S2</p> <p>Wyrażenie logiczne składa się z dwóch argumentów S1 oraz S2. Podany warunek zostanie spełniony, jeżeli przynajmniej jedno wejście Sx przyjmie stan 0.</p>
	<p>LE = (S1 OR S2) AND ~R1</p> <p>Wyrażenie logiczne składa się z trzech argumentów S1, S2 i ~R1. Podany warunek zostanie spełniony, jeżeli przynajmniej jedno z wejść S1 i S2 przyjmie wartość 1, a przekaźnik wykonawczy R1 będzie wyłączony.</p>

4. Zasady definiowania i analizy funkcji czasowej

Oprogramowanie *TimProg* posiada interfejs użytkownika, przy pomocy którego możliwe jest zdefiniowanie schematu działania funkcji czasowych lub sekwencyjnych. W zależności od wybranego typu przekaźnika TPA lub TPD, w oknie aplikacji pojawią się odpowiednio jedno lub dwa pola programu konfigurowanego przez użytkownika. Każdy program składa się z opisanych wcześniej elementów logicznych: pola *START*, *Kontroli stanu R*, *Bloków warunkowych* oraz *Bloków czasowych*. Na samym końcu każdego programu umieszczony jest blok *STOP*, który nie posiada żadnych wartości konfiguracyjnych i służy wyłącznie do zatrzymania działania danego programu.

Wygląd aplikacji dla przekaźników TPA i TPD z zaznaczonymi polami programów przedstawiono poniżej.

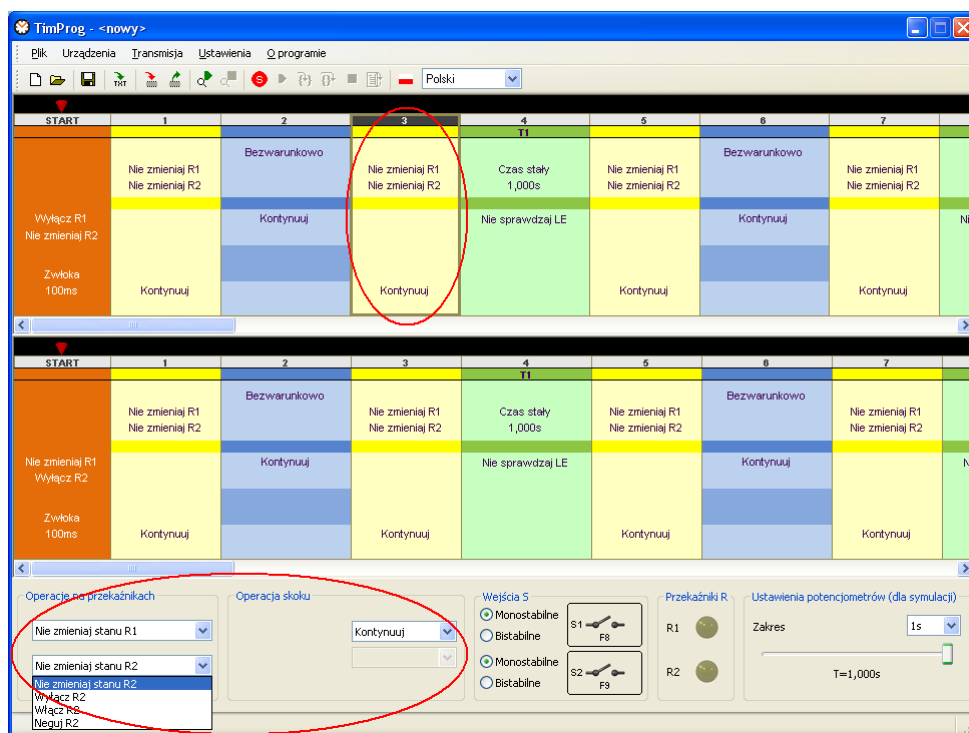


Wszystkie elementy logiczne przekaźnika, poza polami *START* i *STOP*, posiadają swój unikalny numer z zakresu od 1 do 35, zwany w dalszej części instrukcji numerem **Etapu** lub w uproszczeniu **Etapem**. Etap należy rozumieć jako jeden z występujących bloków, który służy do wykonania określonej operacji logicznej. Ilość wykorzystanych w programie bloków zależy wyłącznie od użytkownika, który definiując ich parametry określa sposób działania funkcji czasowej.

Konfiguracja etapów – zasady ogólne

Wartości konfiguracyjne poszczególnych etapów różnią się w zależności od wybranej wersji przekaźnika TPA lub TPD, jednak ogólne zasady ustawiania parametrów są w obu przypadkach identyczne. Z tego względu obsługę programu omówiono na przykładzie wersji TPD.

W celu zdefiniowania ustawień przekaźnika, należy lewym przyciskiem myszki kliknąć na wybrany etap. Zostanie on otoczony ramką, a w dolnej części aplikacji pojawią się opcje konfiguracyjne właściwe dla zaznaczonego pola. Na poniższym rysunku przedstawiono przykład konfiguracji pola *Kontroli stanu R* stanowiącego etap numer 3 programu pierwszego.



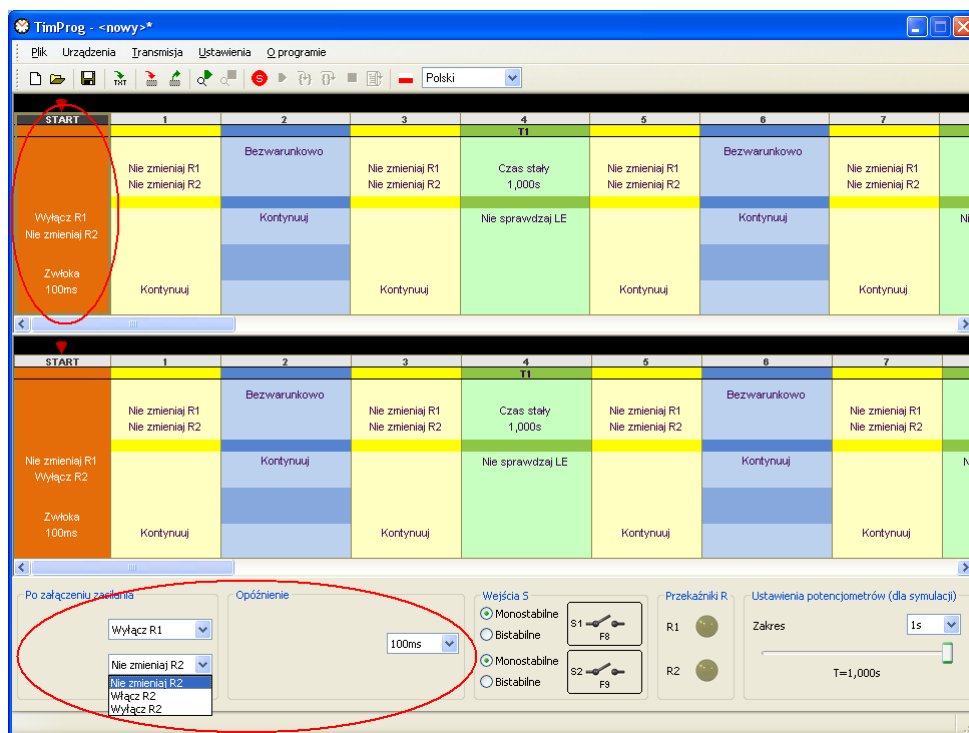
Z dostępnych parametrów należy wybrać pożądane wartości konfiguracyjne. Nie wszystkie etapy muszą zostać w programie skonfigurowane i wykorzystywane. W prawdzie są one zawsze aktywne, ale ustawienia skoków dokonane przez użytkownika mogą wpłynąć na liczbę i kolejność analizy pól.

Konfiguracja pola START

Po kliknięciu na etap START, w dolnej części aplikacji uaktywnią się pola operacji wykonywanej po załączeniu zasilania oraz opcji wstępnego opóźnienia.

W poniższym przykładzie przekaźnik czasowy rozpocznie pracę od wyłączonych obu przekaźników wykonawczych Rx (styki NO rozwarte), a przejście do następnego etapu zostanie opóźnione o 100ms. Cel wprowadzenia opóźnienia został opisany w rozdziale 3. Dla pola START kolejnym wykonywanym etapem jest zawsze blok *Kontroli stanu R* o numerze 1.

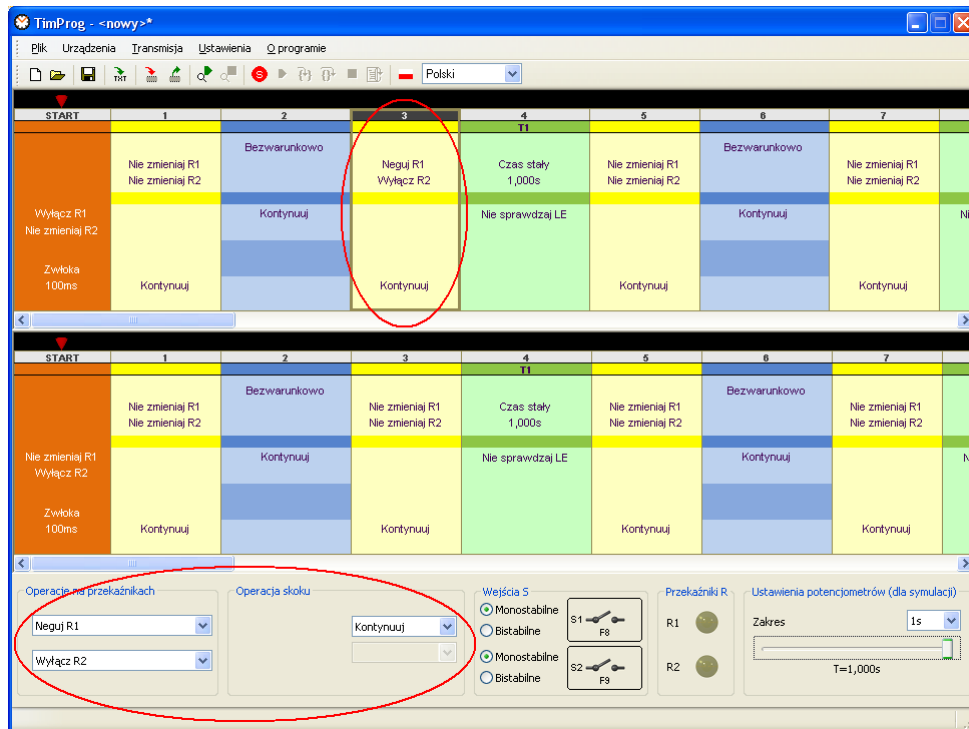
Każdy z dwóch programów przekaźnika posiada własne pola startowe, które mogą być wykorzystane do ponownego uruchomienia zdefiniowanych funkcji czasowych. Należy tak skonfigurować parametry pól START, aby ustawienia przekaźników wzajemnie się nie wykluczały. Dopuszczalna, ale logicznie niepoprawna jest sytuacja, w której ten sam przekaźnik jest w jednym programie włączony a w drugim wyłączony. Można przyjąć zasadę, że ustawienie lub skasowanie Rx w jednym z programów, wymusza konfigurację „Nie zmieniaj Rx” w drugim.



Konfiguracja pola Kontroli stanu R

Pole *Kontroli stanu przekaźników R* (oznaczone kolorem żółtym) zostało wprowadzone w celu modyfikacji stanu styków Rx i opisane w rozdziale 3. Podczas analizy omawianego pola, w pierwszej kolejności program wprowadzi zmiany w ustawieniach przekaźników Rx, a następnie w ramach bieżącego programu wykona skok do etapu ustawionego w polu konfiguracyjnym *Operacja skoku*. Można zdefiniować dowolny numer etapu wraz z polami *START* i *STOP*. Skok do etapu *STOP* oznacza natychmiastowe zatrzymanie pracy danego programu. Ponowne uruchomienie zatrzymanego programu możliwe jest jedynie po wyłączeniu i ponownym załączeniu napięcia zasilającego. Jeżeli w *Operacji skoku* ustawimy wartość *Kontynuuj*, program przejdzie do następnego etapu znajdującego się tuż za bieżącym blokiem. Jest to ogólna zasada dotycząca wszystkich ustawień skoków dostępnych w *Blokach warunkowych*, *Blokach czasowych* i polach *Kontroli stanu R*. Czas analizy omawianego pola jest pomijalnie mały i nie wpływa na końcową wartość odmierzonego czasu. W poniższym przykładzie analiza etapu 3 spowoduje zmianę stanu przekaźnika R1 na przeciwny, wyłączenie R2 i w dalszej kolejności przejście do etapu 4 (*Kontynuuj*).

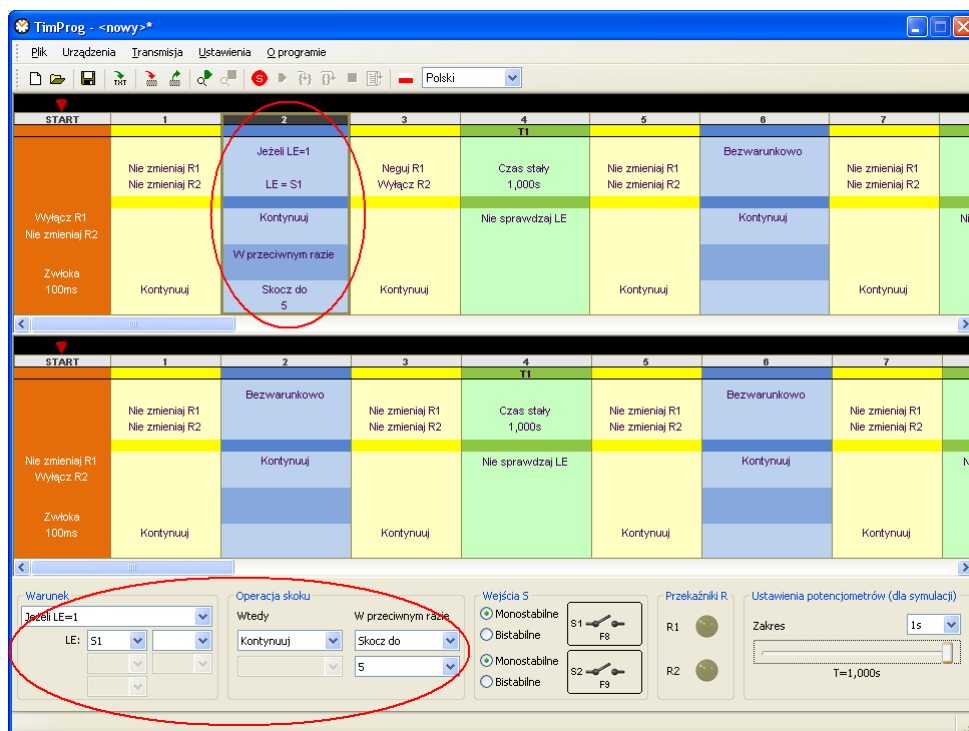
Należy pamiętać, że przekaźniki Rx mogą być wykorzystywane w obu programach równocześnie. Przewrótanie Rx w jednym programie, może w nieoczekiwany sposób wpłynąć na działanie programu drugiego, jeżeli Rx jest tam również obsługiwany.



Konfiguracja bloku warunkowego

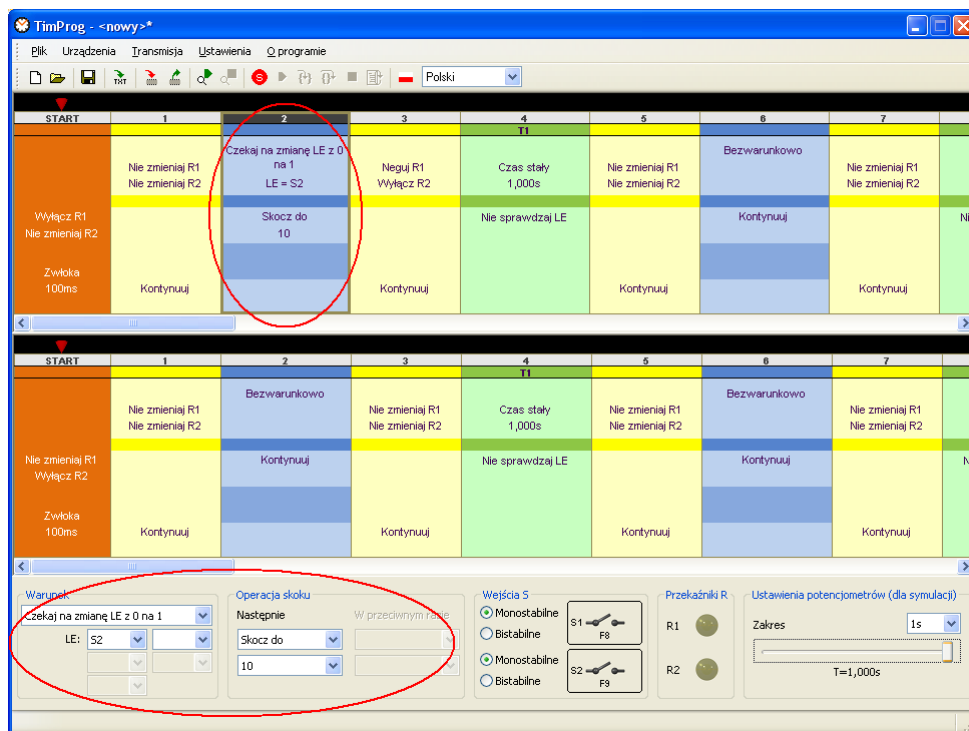
Blok warunkowy (kolor niebieski) jest elementem, w którym w zależności od wybranego typu warunku sprawdzany jest stan wejścia sterującego S i wyjścia R (dla TPA) lub stan wyrażenia logicznego LE (dla TPD). Funkcją tego bloku jest wprowadzenie możliwości warunkowego wykonania programu lub zatrzymanie realizacji funkcji do momentu wystąpienia określonego zdarzenia. Po spełnieniu ustawionego warunku program wykonuje skok do etapu o wybranym numerze.

Przykładową operacją logiczną jest sprawdzenie, czy w momencie analizy Bloku warunkowego stan wejścia S1 był równy 1.



Jeżeli warunek był spełniony, czyli $S1=1$, nastąpi przejście do kolejnego etapu (opcja *Kontynuuj*). W przeciwnym przypadku, gdy $S1=0$, program wykona skok do etapu numer 5 i od tamtego miejsca będzie analizował dalsze pola.

Blok warunkowy pozwala na zatrzymanie analizy programu w oczekiwaniu na zdarzenie, przykładowo na zmianę stanu wyrażenia logicznego LE z 0 na 1. W poniższym Bloku warunkowym wyrażenie logiczne ustawione jest na $LE = S2$, co jest równoznaczne z oczekiwaniem na zmianę stanu $S2$ z 0 na 1.



Program będzie oczekiwał na moment pojawienia się napięcia na $S2$. Gdy to nastąpi, wykona operację skoku do podanego miejsca, czyli w powyższym przykładzie do etapu o numerze 10. Ponieważ oba programy czasowe wykonują się niezależnie, oczekiwanie na zdarzenie w jednym z nich nie ma wpływu na działanie funkcji zdefiniowanej w programie drugim.

Możliwe jest jednak powiązanie działania obu programów poprzez kontrolę stanu przekaźników Rx. Jeżeli stan Rx jest modyfikowany w programie pierwszym, to program drugi, poprzez odpowiednią konfigurację *Bloków warunkowych*, może uzależnić swoje działanie od bieżącej wartości Rx.

W sytuacji, gdy jakkolwiek analiza warunku jest niepotrzebna, należy ustawić opcję *Bezwarunkowo*. Podczas analizy nastąpi natychmiastowy skok do podanego etapu, co wraz z parametrem *Kontynuuj* jest równoznaczne z pominięciem *Bloku warunkowego*.

Konfiguracja bloku czasowego

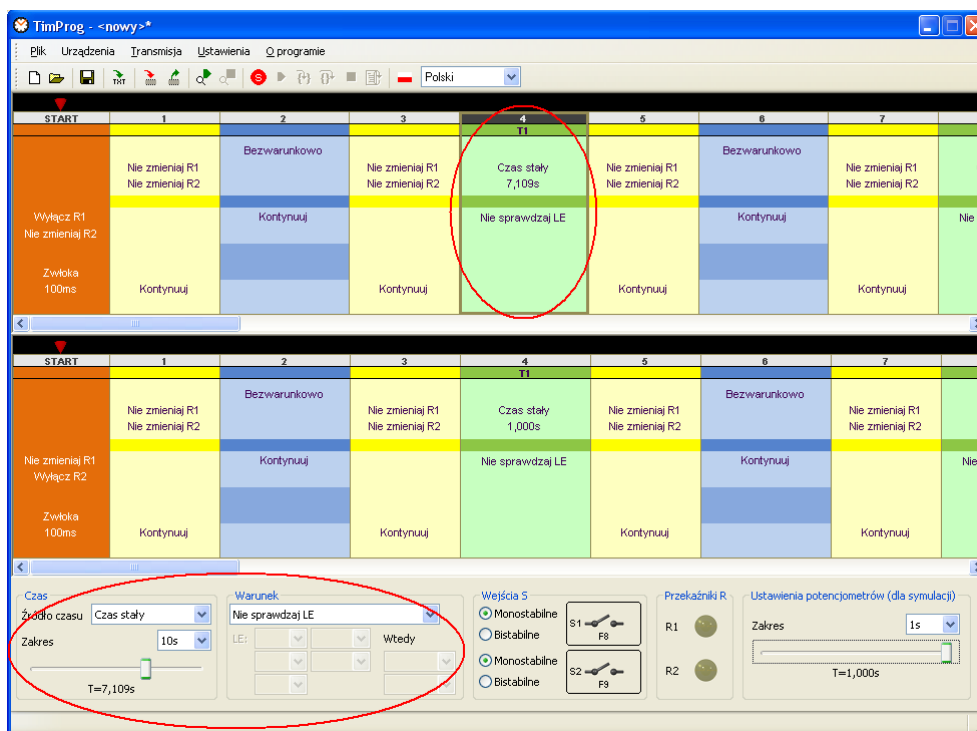
Blok czasowy (kolor zielony) jest elementem wykorzystywanym w programie do odmierzenia czasu. Dozwolone są dwa sposoby określania opóźnienia: *Czas stały* i *Potencjometry*. *Czas stały* pozwala na wprowadzenie z góry określonego opóźnienia poprzez konfigurację *Zakresu* oraz liniowej skali czasu od 0,1 do 1,0. Skonfigurowane opóźnienie jest wyświetlane w dolnej części i wynosi:

$$Czas = skala\ liniowa \cdot Zakres$$

Panel czołowy przekaźnika posiada dwa potencjometry odpowiadające *Zakresowi* oraz liniowej regulacji skali czasu. W sytuacji, gdy docelowe opóźnienie nie jest znane lub może zająć potrzeba jego modyfikacji, jako *Źródło czasu* należy wybrać opcję *Potencjometry*. Wówczas odmierzony czas nie będzie na stałe wpisany w program, a jego wartość ustalona zostanie na podstawie położenia potencjometrów w chwili rozpoczęcia odmierzenia czasu. Ułatwia to dostrojenie przekaźnika przy pomocy zwykłego śrubokrętu bez potrzeby podłączania komputera i modyfikacji oprogramowania.

W trakcie pracy przekaźnika, podczas analizy Bloku czasowego, program zatrzyma się na omawianym polu na nastawiony czas, a po jego odmierzeniu przejdzie zawsze do następnego etapu znajdującego się bezpośrednio za *Blokiem czasowym*.

W trakcie realizacji niektórych funkcji czasowych może zająć potrzeba reakcji na pewne zdarzenia, występujące w trakcie odmierzenia czasu. Stąd *Blok czasowy* wyposażono w opcję kontroli wyrażenia logicznego LE (wersja TPD) lub stanu styku sterującego S (wersja TPA). Spełnienie warunku określonego w bloku czasowym powoduje natychmiastowe zatrzymanie licznika czasu oraz przejście do etapu ustawionego w *Bloku czasowym*. W poniższym przykładzie czas *T1* etapu o numerze 4 ustawiono na 7,109s, natomiast wyrażenie logiczne LE nie jest sprawdzane uniemożliwiając tym samym wcześniejsze przerwanie odmierzenia czasu.



Jeżeli program ma rozpocząć działanie od początku (od pola START), gdy przekaźnik R2 sterowany w programie drugim zmieni swój stan z 0 na 1. Wówczas konfiguracja pola czasowego wygląda następująco:

The screenshot displays the TimProg software interface for configuring a relay control program. The main workspace shows a ladder logic diagram with eight steps (START, 1, 2, 3, 4, 5, 6, 7). Step 4 is highlighted with a red circle and contains a time delay element (T1) with the following configuration:

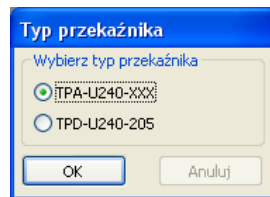
- Time Source:** Czas stały (Constant time)
- Time Value:** 0m:43s
- Condition:** Jeżeli zmiana LE z 0 na 1 (If change of LE from 0 to 1)
- Action:** Skocz do START (Jump to START)

The configuration panel at the bottom of the window is also highlighted with a red circle and shows the following settings:


- Time Source:** Czas stały
- Condition:** Jeżeli zmiana LE z 0 na 1
- Time Value:** T=0m:43s
- Relay R2:** Monostabilne (Monostable)
- Relay R1:** Bistabilne (Bistable)
- Relay R2:** Bistabilne (Bistable)
- Time Value:** T=1,000s


5. Zapis i odczyt konfiguracji


Po uruchomieniu aplikacji *TimProg* pojawia się okno wyboru, pozwalające otworzyć program w trybie do obsługi przekaźników serii TPA lub TPD.




Po zaakceptowaniu wyboru w oknie aplikacji pojawiają się jeden lub dwa programy użytkownika wypełnione polami ustawionymi na wartości domyślne. Są one tak dobrane, aby załadowanie ich do przekaźnika spowodowało przejście przez wszystkie kolejne bloki i zatrzymanie się na polu *STOP*.

W każdym momencie możliwe jest całkowite skasowanie programu przekaźnika i ustawienie go na wartości domyślne za pomocą komendy *Plik*→*Nowy* lub poprzez ikonkę . Przy otwieraniu nowego programu należy podać ponownie typ przekaźnika TPA lub TPD, dla jakiego chcemy otworzyć nową konfigurację.

Zapis bieżącej konfiguracji na dysku komputera odbywa się poprzez opcje *Plik*→*Zapisz*  lub *Plik*→*Zapisz jako*. Domyślnym rozszerzeniem pliku programu przekaźnika jest *.tpa.


Odczyt istniejącej konfiguracji można dokonać poprzez *Plik*→*Otwórz* , jednak spowoduje to utratę bieżących ustawień programu i zastąpienie ich wartościami umieszczonymi w pliku *.tpa.

6. Ładowanie konfiguracji do przekaźnika

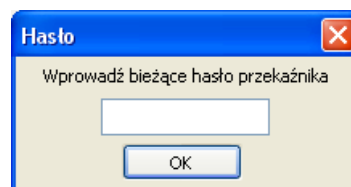
Aplikacja *TimProg* umożliwia zapis konfiguracji do nieulotnej pamięci przekaźnika, która nie zostanie skasowana nawet po odłączeniu napięcia zasilającego. W celu dokonania zapisu należy w dowolnym momencie uruchomić komendę *Transmisja* → *Wyślij ustawienia* . Praca przekaźnika zostanie zatrzymana, nowe wartości załadowane, po czym nastąpi uruchomienie programu od pola *START*.



Podczas ładowania programu należy zachować ostrożność, aby zatrzymanie pracy przekaźnika i rozpoczęcie realizacji nowego programu nie wpłynęły na bezpieczeństwo pracy urządzenia, w którym przekaźnik jest zainstalowany.

W dowolnym momencie pracy możliwy jest odczyt ustawień z przekaźnika i aktualizacja pól aplikacji *TimProg* (komenda *Transmisja* → *Odczytaj ustawienia* lub ikonka ). Odczyt programu nie wpływa na pracę przekaźnika i realizację funkcji.

W przypadku dostępu do zabezpieczonych przekaźników serii TPD, program zapyta o hasło znajdujące się w przekaźniku. Odczyt lub zapis konfiguracji możliwy jest wyłącznie po podaniu prawidłowego hasła dostępu.

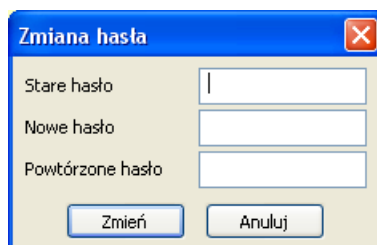


7. Zabezpieczenie dostępu do przekaźnika hasłem

i Zabezpieczenie dostępu do programu hasłem dotyczy wyłącznie przekaźników serii TPD.

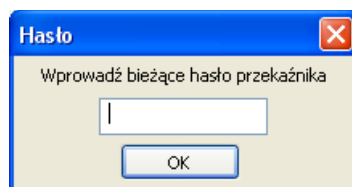
Aplikacja *TimProg* umożliwia zabezpieczenie dostępu do programu znajdującego się w przekaźniku TPD za pomocą hasła. Chroni to bieżącą konfigurację przed modyfikacją przez niepowołane osoby oraz przed nieautoryzowanym kopiowaniem stworzonego rozwiązania.

W nowym przekaźniku hasło nie jest ustawione i podczas prób dostępu do programu użytkownika aplikacja *TimProg* używa domyślnie hasła pustego. Jego ustawienie możliwe jest za pomocą komendy *Transmisja* → *Zmień hasło*.

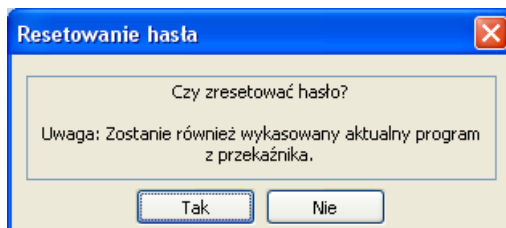


W celu zmiany hasła należy podać hasło bieżące oraz dwa razy nowe, składające się maksymalnie z ośmiu znaków. W przypadku modyfikacji przekaźnika, w którym hasło nie było dotychczas używane, pole *Stare hasło* należy pozostawić puste. Po ustawieniu hasła zostanie ono zapamiętane w aplikacji *TimProg* na potrzeby bieżącej sesji i nie ma potrzeby ponownego jego wpisywania przy każdym dostępie do przekaźnika.

Przy pierwszej próbie dostępu do zabezpieczonego przekaźnika, aplikacja *TimProg* poprosi o wprowadzenie bieżącego hasła.





W przypadku zagubienia hasła zabezpieczającego, dostęp do przekaźnika nie jest możliwy. Aplikacja *TimProg* pozwala odblokować przekaźnik komendą *Transmisja* → *Resetuj hasło i program...*, ale wraz z resetem hasła kasowany jest również zapisany w przekaźniku program użytkownika.

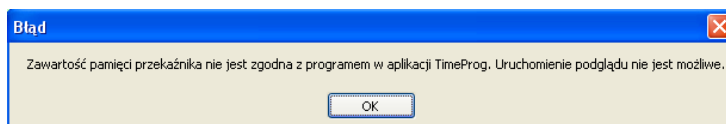


Po zresetowaniu przekaźnika nowe hasło jest puste.

8. Podgląd bieżącego stanu pracy przekaźnika

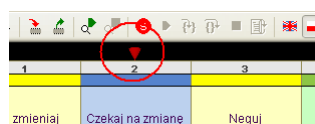
Aplikacja *TimProg* umożliwia podgląd bieżącego stanu pracy przekaźnika bez wpływu na jego pracę. Ułatwia to proces definiowania funkcji czasowych i analizę ich działania w rzeczywistym urządzeniu. Po podłączeniu kabla USB i załączeniu napięcia zasilającego należy wywołać funkcję *Transmisja* → *Uruchom podgląd* lub kliknąć na ikonkę . Do przerywania podglądu służy komenda *Transmisja* → *Zatrzymaj podgląd* lub .

Przy starcie podglądu sprawdzana jest zgodność programu zapisanego w przekaźniku z bieżącą konfiguracją aplikacji *TimProg*, gdyż w przypadku rozbieżności dalsze śledzenie programu jest bezcelowe. Pojawienie się komunikatu

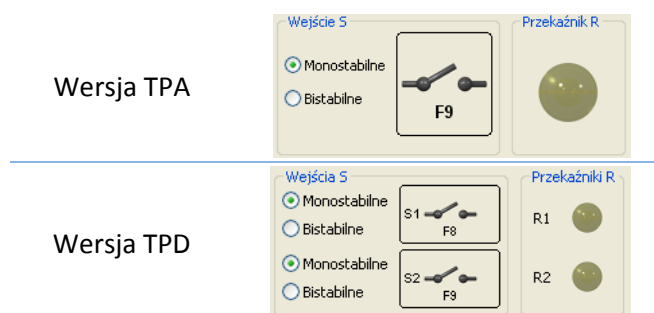


oznacza, że w przekaźniku znajduje się inny program niż w aplikacji.

W trakcie podglądu działania pracy przekaźnika, przy pomocy czerwonej strzałki, aplikacja wskazuje bieżący etap, w którym znajduje się program.



Ponadto aktualne wartości stanów wejść sterujących Sx oraz przekaźników wykonawczych Rx wskazywane są w dolnej części aplikacji. Zamknięcie styku Sx oznacza, że na jego wejście zostało podane napięcie sterujące.




W chwili analizy *Bloku czasowego*, wartość odmierzonego czasu jest odświeżana i wyświetlana w aktualnym polu w kolorze niebieskim. Jest to wartość bieżąca, a nie docelowa ustawiona w programie!




Aplikacja *TimProg* odświeża ustawienia przekaźnika cztery razy na sekundę. Oznacza to, że etapy, których czas analizy jest znikomo mały, mogą zostać pominięte przy śledzeniu wskaźnika określającego numer bieżącego etapu.

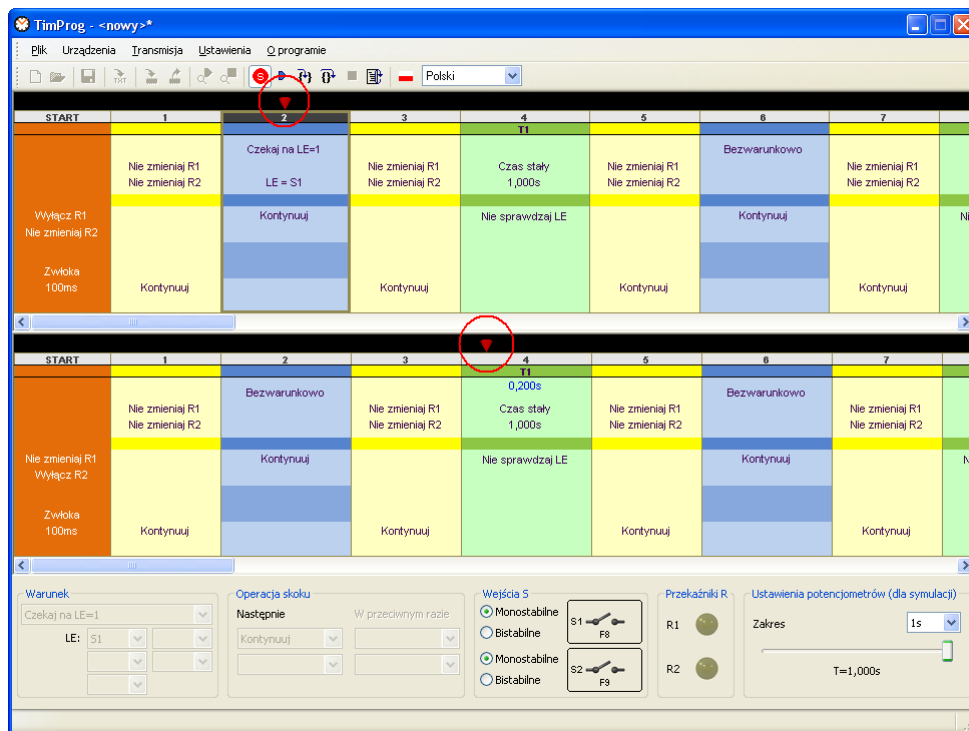
9. Programowa symulacja pracy przekaźnika

Aplikacja *TimProg* umożliwia programową symulację pracy zdefiniowanych funkcji czasowych bez konieczności podłączenia przekaźnika czasowego. Wszelkie próby i stworzenie własnego programu możliwe są od razu po instalacji aplikacji, bez konieczności zakupu gotowego wyrobu.

Wejście w tryb symulacji odbywa się przy pomocy komendy *Transmisja* → *Tryb symulacji* lub ikonki . Tryb symulacji nie oznacza automatycznego rozpoczęcia analizy programu, a jedynie aktywację komend, które pozwalają śledzić działanie zdefiniowanych funkcji.

Praca ciągła

Opcja *Transmisja* → *Uruchom symulację* lub ikona  powoduje rozpoczęcie analizy programu w sposób ciągły. To znaczy, że z określoną szybkością aplikacja będzie przechodzić przez poszczególne etapy według ustawień wprowadzonych przez użytkownika, a czerwone strzałki wskażą bieżące pola.

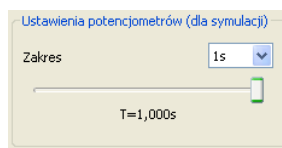


W dolnej części okna aplikacji znajdują się elementy kontrolne – wejścia sterujące S_x oraz żółte lampki odzwierciedlające stan przekaźników wykonawczych R_x .



Przy pomocy myszki lub klawiszami skrótów użytkownik może zmieniać stan wejść S_x i w ten sposób wpływać na działanie przygotowanego programu. Dla wygody obsługi przyciski kontrolne S_x posiadają dwa tryby pracy. W trybie monostabilnym wejścia S_x załączone są tak długo, jak naciśnięty będzie odpowiadający im klawisz skrót lub przycisk myszki. Praca bistabilna pozwala na naprzemienne, trwałe załączanie i wyłączenie tych wejść.


Ustawienie *Źródła czasu* z potencjometrów w *Blokach czasowych* oznacza, że jego wartość nie jest znana, gdyż zależy wyłącznie od położenia pokręteł na panelu czołowym przekaźnika. Na potrzeby symulacji należy ustawić żądany czas w panelu umieszczonym się w prawym, dolnym rogu aplikacji.




Do zatrzymania pracy ciągłej służy opcja *Transmisja* → *Zatrzymaj symulację* lub ikonka .


Należy zwrócić uwagę, że rzeczywisty czas odmierzany w symulacji nie jest zgodny z ustawionym. Wartość czasu w każdym kroku symulacji jest zwiększana o pewien procent bieżącego zakresu, aby analiza bloków czasowych nie trwała zbyt długo.

Praca krokowa


Symulacja programu w trybie ciągłym odbywa się na tyle szybko, że uchwycenie szczegółów działania funkcji może być kłopotliwe. W tym celu wprowadzono opcję pracy krokowej uruchamianej poprzez *Transmisja* → *Krokuj* lub ikonkę . Każdorazowe wywołanie tej opcji powoduje uruchomienie jednego kroku symulacji, co odpowiada analizie pojedynczego pola *Kontroli stanu R*, *Bloku warunku* lub w *Bloku czasowym* zwiększenia licznika czasu o pewien procent wartości ustawionego zakresu.

Analiza *Bloku czasowego* wymaga często wielu kroków symulacji, co jest związane z inkrementacją licznika o pewien procent wartości zakresu. Jest to zwykle na tyle dużo, że wprowadzono opcję *Transmisja* → *Przeskocz* , której wywołanie powoduje przerwanie odmierzania czasu i przejście symulacji do następnego etapu znajdującego się po *Bloku czasowym*.

Reset symulacji

Zatrzymanie symulacji uruchomionej w trybie ciągłym ustawia czerwony wskaźnik nad ostatnim analizowanym polem, a wszystkie parametry programu są pamiętane. Ponowne uruchomienie symulacji powoduje kontynuację działania programu od miejsca zatrzymania. Rozpoczęcie symulacji od początku możliwe jest wyłącznie po uprzednim wywołaniu polecenia *Transmisja* → *Reset* lub kliknięciu na ikonkę .

10. Eksport ustawień

Opcja *Plik* → *Eksport dokumentu*  służy do zapisu ustawień programu do pliku tekstowego. Ułatwia to przygotowanie dokumentacji urządzenia, gdyż opis konfiguracji może być łatwo skopiowany do dowolnego edytora.

Przykładowy fragment pliku ustawień przekaźnika TPA jest następujący:

```
Etap: START
-----
    Po załączeniu zasilania      : Wyłącz R
    Opóźnienie                   : 100ms

Etap: 1
-----
    Operacja na przekaźniku R    : Nie zmieniaj stanu R
    Operacja skoku               : Kontynuuj

Etap: 2
-----
    Warunek                      : Czekaj na zmianę S z 0 na 1
    Operacja skoku               : Kontynuuj

Etap: 3
-----
    Operacja na przekaźniku R    : Nie zmieniaj stanu R
    Operacja skoku               : Kontynuuj

Etap: 4
-----
    Źródło czasu                 : Czas stały
    Ustawiony czas              : 1,000s
    Warunek                      : Nie sprawdzaj S

Etap: 5
-----
    Operacja na przekaźniku R    : Neguj R
    Operacja skoku               : Skocz do 2
```

11. Przykładowe funkcje

W miejscu instalacji oprogramowania *TimProg* znajduje się podkatalog *Functions*, w którym umieszczono plik w formacie PDF z opisem wszystkich typowych funkcji czasowych oferowanych w wyrobach firmy Dobry Czas Sp. z o.o. Katalog ten zawiera także gotowe pliki w formacie *.tpa dla przekaźników serii TPA.

Pierwszy program – odmierzenie czasu zadziałania TB

Funkcja TB generująca pojedynczy impuls o czasie trwania T jest jedną z podstawowych funkcji oferowanych przez producentów przekaźników czasowych. Jej wykres jest następujący:



Odmierzanie czasu zadziałania (TB) - po załączeniu zasilania U przekaźnik wykonawczy R zostaje załączony i pozostaje w tym stanie przez czas T . Po upływie czasu T przekaźnik R zostaje na stałe wyłączony. Rozpoczęcie kolejnego cyklu pracy możliwe jest po wyłączeniu i ponownym podaniu napięcia zasilającego.

Prześledźmy wybrany sposób implementacji powyższej funkcji w programowalnych przekaźnikach czasowych TPA oraz TPD.

TPA	TPD	
		<p>W pierwszym kroku ustawmy pola <i>START</i>. Jak widać z wykresu funkcji przekaźnik wykonawczy R jest załączony od razu po podaniu napięcia zasilającego. Oznacza to, że w konfiguracji <i>START</i> należy wybrać parametr <i>Włącz R</i>. W przypadku wersji TPD będziemy korzystać wyłącznie z programu górnego i przekaźnika $R1$, gdzie go ustawiamy. W programie dolnym pozostawiamy <i>Nie zmieniaj R1</i>.</p> <p>Ze względu na fakt, iż nie korzystamy z wejścia sterującego S, możemy pominąć wstępne opóźnienie ustawiając zwłokę na $0ms$.</p>
		<p>Po wykonaniu analizy pola <i>START</i> program przekaźnika przechodzi do etapu o numerze 1 będącego polem typu <i>Kontrola stanu R</i>. Przekaźnik R mamy już ustawiony i nie ma potrzeby zmiany jego stanu, dlatego etap ten należy pominąć ustawiając parametry <i>Nie zmieniaj R (Rx)</i> oraz <i>Kontynuuj</i>.</p> <p>Dla wersji TPD kończymy niewykorzystywany program dolny wykonując skok do <i>STOP</i>.</p>

		<p>W dalszej kolejności program przechodzi do <i>Bloku warunkowego</i> o numerze 2. Żaden warunek nie jest potrzebny, dlatego blok również należy pominąć ustawiając <i>Bezwarunkowo</i> i <i>Kontynuuj</i>. Program kontynuował będzie pracę przechodząc do etapu 3.</p> <p>Dalsze przykłady ustawień pól dla przekaźnika TPD dotyczą będą programu górnego, gdyż dolny jest nieużywany.</p>
		<p>Kolejnym etapem jest <i>Kontrola stanu R</i>. Tak jak poprzednio nie ma potrzeby dokonywania żadnej zmiany, dlatego etap pomijamy przechodząc do <i>Bloku czasowego</i> o numerze 4.</p>
		<p>Przeszliśmy w końcu do miejsca, w którym nasz program zatrzyma się na czas 5s. W prawdzie analiza poprzednich etapów też zajmuje czas procesora, ale jest on pomijalnie mały. Pamiętajmy, że wcześniej w polu <i>START</i> ustawiliśmy przekaźnik R (R1) na załączony i takim stanie będzie znajdował się teraz przez całe 5s. Po odmierzeniu czasu program przechodzi do następnego etapu znajdującego się bezpośrednio za nim (<i>generalna zasada dla Bloków czasowych</i>).</p>
		<p>Odmierzaliśmy już 5s i przyszedł czas na wyłączenie przekaźnika R (R1). Znaleźliśmy się w polu <i>Kontroli stanu R</i>, w którym ustawiamy <i>Wyłącz R (R1)</i>. Realizacja funkcji dobiegła końca, dlatego zatrzymujemy działanie przekaźnika wykonując skok do pola <i>STOP</i>.</p>
		<p>Program zakończył swoje działanie. Aby go jeszcze raz uruchomić wymagane jest wyłączenie i ponowne załączenie napięcia zasilającego.</p>

Okno zawierające gotowe programy dla przekaźników TPA i TPD wyglądają następująco:

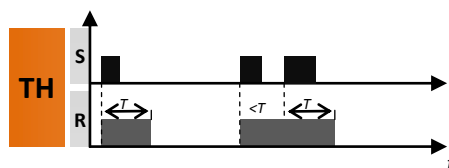
START	1	2	3	4	5
				T1	
Włącz R	Nie zmieniaj R	Bezwarunkowo	Nie zmieniaj R	Czas stały 5,000s	Wyłącz R
Zwłoka 0ms	Kontynuuj	Kontynuuj	Kontynuuj	Nie sprawdzaj S	Skocz do STOP

START	1	2	3	4	5
				T1	
Włącz R1 Nie zmieniaj R2	Nie zmieniaj R1 Nie zmieniaj R2	Bezwarunkowo	Nie zmieniaj R1 Nie zmieniaj R2	Czas stały 5,000s	Wyłącz R1 Nie zmieniaj R2
Zwłoka 0ms	Kontynuuj	Kontynuuj	Kontynuuj	Nie sprawdzaj LE	Skocz do STOP

START	1	2	3	4	5
				T1	
Nie zmieniaj R1 Wyłącz R2	Nie zmieniaj R1 Nie zmieniaj R2	Bezwarunkowo	Nie zmieniaj R1 Nie zmieniaj R2	Czas stały 1,000s	Nie zmieniaj R1 Nie zmieniaj R2
Zwłoka 0ms	Skocz do STOP	Kontynuuj	Kontynuuj	Nie sprawdzaj LE	Kontynuuj

Generacja impulsu z przedłużaniem wyzwalana zboczem narastającym na styku S

Mamy już za sobą pierwszy program generujący pojedynczy impuls. Zmodyfikujmy go tak, żeby przekaźnik załączał się tylko w momencie podania napięcia na wejście sterujące S, a w trakcie odmierzenia czasu możliwe było jego przedłużenie. Wykres funkcji jest następujący:



Generacja impulsu z przedłużaniem wyzwalana zboczem narastającym na styku S (TH) - w momencie wystąpienia narastającego zbocza na styku S przekaźnik wykonawczy R zostaje załączony na czas T. Ewentualne zbocze narastające na styku S podane w trakcie odmierzenia czasu powoduje rozpoczęcie odliczania czasu T od początku.

Opis kolejnych etapów programu:

TPA	TPD	
		<p>Przekaźnik wykonawczy R pozostaje wyłączony po podaniu napięcia zasilającego, choć nie jest to na wykresie zaznaczone. Wybieramy parametr <i>Wyłącz R</i> dla TPA.</p> <p>Dla wersji TPD ustawiamy <i>Wyłącz R1</i> w programie górnym i <i>Nie zmieniaj R1</i> w programie dolnym.</p> <p>W programie korzystamy także z wejścia S (S1). Ponieważ bezpośrednio po załączeniu zasilania rzeczywisty stan S nie jest znany, dlatego wprowadzamy opóźnienie 100ms, aby dać procesorowi czas na dokonanie pomiaru.</p>
		<p>Po wykonaniu analizy pola <i>START</i> program przekaźnika przechodzi do etapu o numerze 1 będącego polem typu <i>Kontrola stanu R</i>. Przekaźnik R (R1) mamy już ustawiony i nie ma potrzeby zmiany jego stanu, dlatego etap ten należy pominąć ustawiając parametry <i>Nie zmieniaj R (R1)</i> oraz <i>Kontynuuj</i>.</p> <p>Dla wersji TPD kończymy niewykorzystywany program dolny wykonując skok do STOP.</p>

		<p>Kontynuując program przeszedł do <i>Bloku warunkowego</i>. Ponieważ chcemy, aby odmierzenie czasu rozpoczęło się po aktywacji S (S1), dlatego zatrzymujemy się w tym miejscu aż do momentu, jak stan S (S1) zmieni się z 0 na 1. Dla wersji TPA ustawiamy <i>Czekaj na zmianę S z 0 na 1</i>, po wykryciu której przechodzimy do kolejnego etapu. W operacji skoku konfigurujemy zatem <i>Kontynuuj</i>. Dla wersji TPD korzystamy już wyłącznie z programu górnego, gdzie czekamy na zmianę wyrażenia logicznego LE z 0 na 1, przy czym LE=S1.</p>
		<p>Zmiana na wejściu S (S1) została wykryta, dlatego włączamy przekaźnik wykonawczy R (R1) i <i>Kontynuujemy</i> przechodząc do <i>Bloku czasu</i>.</p>
		<p>Tutaj każemy przekaźnikowi zatrzymać się na 5s, ale wprowadzamy dodatkową kontrolę. Jeśli podczas odmierzenia czasu wykryjemy zmianę na wejściu S (S1) z 0 na 1, wykonujemy skok do etapu 4, czyli faktycznie do bieżącego. Dzięki temu czas zacznie liczyć się od nowa i uzyskamy wymagane przedłużenie. Jeżeli jednak zmiana nie nastąpi, program przejdzie do następnego etapu po 5s.</p>
		<p>Przekaźnik odmierzył nastawiony czas i styki R (R1) należy wyłączyć. Ustawiamy zatem <i>Wyłącz R (R1)</i> oraz wykonujemy skok do etapu 2, w którym oczekiwać będziemy na ponowne załączenie wejścia S (S1).</p>

Wygląd gotowego programu dla przekaźników TPA oraz TPD wygląda następująco:

START	1	2	3	4	5
Wyłącz R Zwłoka 100ms	Nie zmieniaj R	Czekaj na zmianę S z 0 na 1	Włącz R	Czas stały 5,000s	Wyłącz R
		Kontynuuj		Jeżeli zmiana S z 0 na 1	
	Kontynuuj		Kontynuuj	Skocz do 4	Skocz do 2

START	1	2	3	4	5
Wyłącz R1 Nie zmieniaj R2 Zwłoka 100ms	Nie zmieniaj R1 Nie zmieniaj R2	Czekaj na zmianę LE z 0 na 1 LE = S1	Włącz R1 Nie zmieniaj R2	Czas stały 5,000s	Wyłącz R1 Nie zmieniaj R2
		Kontynuuj		Jeżeli zmiana LE z 0 na 1 LE = S1	
	Kontynuuj		Kontynuuj	Skocz do 4	Skocz do 2

START	1	2	3	4	5
Nie zmieniaj R1 Wyłącz R2 Zwłoka 100ms	Nie zmieniaj R1 Nie zmieniaj R2	Bezwarunkowo	Nie zmieniaj R1 Nie zmieniaj R2	Czas stały 1,000s	Nie zmieniaj R1 Nie zmieniaj R2
		Kontynuuj		Nie sprawdzaj LE	
	Skocz do STOP		Kontynuuj		Kontynuuj

W poniższej instrukcji pokazano jedynie krótkie przykłady definiowania funkcji czasowych. Jeżeli w programie użytkownik pominię *Bloki czasowe* i wykorzysta odpowiednio *Bloki warunkowe*, powstać może przekaźnik sekwencyjny, np. naprzemiennie załączający i wyłączający wyjście po wykryciu impulsu na wejściu kontrolnym S. Ilość zasobów oraz różnorodność ustawień dają bardzo duże możliwości konfiguracyjne, pozwalając na implementację schematów sterowania niespotykanych w typowych przekaźnikach czasowych.



Dobry Czas Sp. z o.o. 51-315 Wrocław ul. Miłostowska 7/6
 ☎ +48 71 729 95 90
 ✉ marketing@dobry-czas.pl

www.dobry-czas.pl