

# LogicProcessor\_OptimaGSM

© 2016



## OSTRZEŻENIA

---

*Ze względów bezpieczeństwa urządzenie powinno być instalowane tylko przez wykwalifikowanych specjalistów.*

*Przed przystąpieniem do montażu zapoznać się z powyższą instrukcją, czynności połączeniowe należy wykonywać bez podłączonego zasilania.*

*Nie wolno włączać zasilania urządzenia bez podłączonej anteny zewnętrznej (uruchomienie urządzenia bez podłączonej anteny grozi uszkodzeniem układów nadawczych telefonu i utratą gwarancji!).*

*Nie wolno ingerować w konstrukcję bądź przeprowadzać samodzielnych napraw. Należy chronić elektronikę przed wyładowaniami elektrostatycznymi.*

*W celu spełnienia wymagań LVD i EMC należy przestrzegać zasad: zasilania, zabudowy, ekranowania - odpowiednio do zastosowania.*

*Urządzenie jest źródłem fal elektromagnetycznych, dlatego w specyficznych konfiguracjach może zakłócać inne urządzenia radiowe).*

*Firma Ropam elektronik nie ponosi odpowiedzialności za nieprawidłowe działanie sieci GSM i skutków ewentualnych problemów technicznych.*

## OZNAKOWANIE WEEE

---

*Zużytego sprzętu elektrycznego i elektronicznego nie wolno wyrzucać razem ze zwykłymi domowymi odpadami. Według dyrektywy WEEE (Dyrektywy 2002/96/EC) obowiązującej w UE dla używanego sprzętu elektrycznego i elektronicznego należy stosować oddzielne sposoby utylizacji. W Polsce zgodnie z przepisami o zużytym sprzęcie elektrycznym i elektronicznym zabronione jest umieszczanie łącznie z innymi odpadami zużytego sprzętu oznakowanego symbolem określonego kosza. Użytkownik, który zamierza się pozbyć tego produktu, jest obowiązany do oddania ww. do punktu zbierania zużytego sprzętu. Punkty zbierania prowadzone są m.in. przez sprzedawców hurtowych i detalicznych tego sprzętu oraz gminne jednostki organizacyjne prowadzące działalność w zakresie odbierania odpadów.*

*Prawidłowa realizacja tych obowiązków ma znaczenie zwłaszcza w przypadku, gdy w zużytym sprzęcie znajdują się składniki niebezpieczne, które mają negatywny wpływ na środowisko i zdrowie ludzi.*

*Zasilacz centrali współpracuje z akumulatorem 12V DC ołowiowo-kwasowym suchym (SLA, VRL). Po okresie eksploatacji nie należy go wyrzucać, lecz zutylizować w sposób zgodny z obowiązującymi przepisami. (Dyrektywy Unii Europejskiej 91/157/EEC i 93/86/EEC).*



# LogicProcessor\_OptimaGSM

© 2016

*Firma Ropam Elektronik jest wyłącznym właścicielem praw autorskich do materiałów zawartych w dokumentacjach, katalogu i na stronie internetowej, w szczególności do zdjęć, opisów, tłumaczeń, formy graficznej, sposobu prezentacji.*

*Wszelkie kopiowanie materiałów informacyjnych czy technicznych znajdujących się w katalogach, na stronach internetowych czy w inny sposób dostarczonych przez Ropam Elektronik wymaga pisemnej zgody.*

*Wszystkie nazwy, znaki towarowe i handlowe użyte w tej instrukcji i materiałach są własnością stosownych podmiotów i zostały użyte wyłącznie w celach informacyjnych oraz identyfikacyjnych.*

Wydruk: listopad 2016

Wersja:dokumentacji: 1.0.0

## PRODUCENT

*Ropam Elektronik s.c.*

*Polanka 301*

*32-400 Myślenice, POLSKA*

*tel:12-341-04-07*

*tel: 12-272-39-71*

*fax: 12-379-34-10*

*biuro@ropam.com.pl*

*servis@ropam.com.pl*

*www.ropam.com.pl*



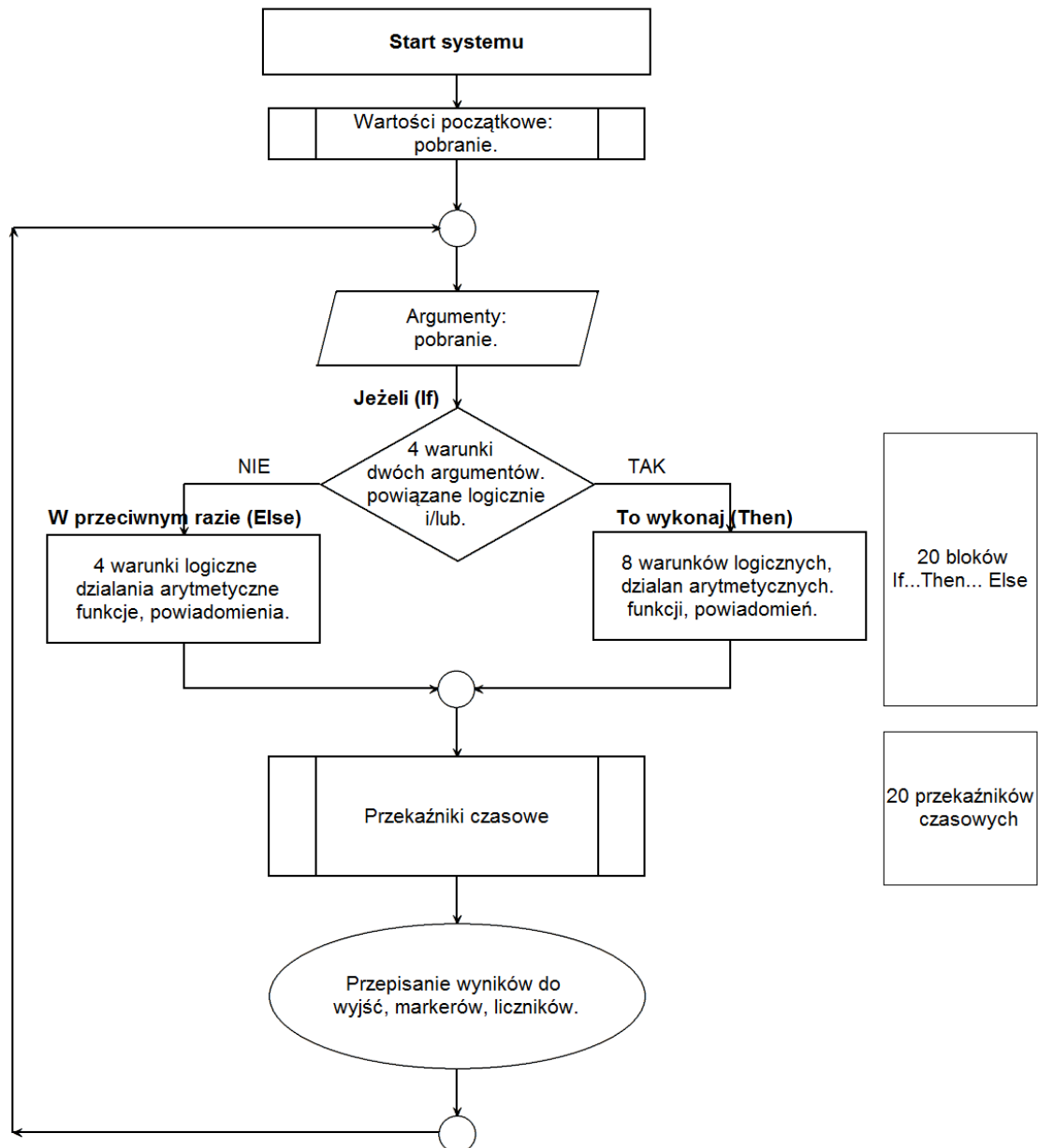
# Spis treści

<b>Rozdział I Zakładka: LogicProcessor</b>	<b>5</b>
1 Funkcje logiczne. ....	6
2 Przekazniki czasowe. ....	18
3 Wartości startowe. ....	22
4 Podgląd skryptu. ....	23
5 Symulator. ....	24
6 Noty aplikacyjne. ....	24

# 1 Zakładka: LogicProcessor

LogicProcessor:

- zaawansowane funkcje logiczne, funkcje arytmetyczne, liczniki, przełączniki czasowe,
- do 20 niezależnych warunków logicznych, (bloki If...Then...Else),
- 20 przełączników czasowych do realizacji funkcji czasowo-logicznych,
- kreator logiki lub edytor skryptu (język skryptowy C), symulator: zadajnik i podgląd skryptu,



**LogicProcessor, funkcje wykonywane są w pętli wg schematu.**

Po restarcie pobierane są wartości początkowe (opcjonalnie).

Skrypt następnie pobiera dane argumentów.

Następnie wykonywane jest do 20 bloków: **Jeżeli... To wykonaj.. W przeciwnym przypadku (If... Then...Else).**

Warunki **Jeżeli'** to maks. 8 warunków między dwoma argumentami, każdy wiersz jest powiązany

logiką 'i/lub' (AND/OR) z kolejnym.

W przypadku spełnienia warunku **'If'** wykonywana jest blok **'Then'** w przeciwnym przypadku **'Else'**. Warunki **"To wykonać (Then)"** to maksymalnie 8 wierszy logicznych, arytmetycznych lub powiadomienia na panelach (Print= okno informacyjne na panelu TPR-1x/TPR-4x, HINT=dolny pasek, lub SMS-ów).

Warunki **'W przeciwnym przypadku' (Else)** maksymalnie 4 to warunkowe funkcje jeżeli nie zostanie spełniony warunek 'If'.

Następnie wykonywane jest do 20 przełączników czasowych.

Na końcu pętli wyniki są przepisywane do wyjść fizycznych, markerów, liczników.

**Uwagi:**

Funkcje wykonywane są w pętli wg schematu. Fizyczne wyjścia używane (wyniki funkcji) w LogicProcessor muszą mieć zaznaczone sterowanie "LogicProcessor".

## 1.1 Funkcje logiczne.

Argumenty		
Symbol	Opis	Wartości
<b>I1÷I48</b>	stan wejść, przyjmuje wartość binarną 0 lub 1,	0= wejście nienaruszone 1= wejście naruszone
<b>O1÷O32</b>	stan wyjść fizycznych, przyjmuje wartość binarną 0 lub 1	0= wyjście nieaktywne 1= wyjście aktywne
<b>tk1÷tk4</b>	wskaźniki timerów/kalendarzy, przyjmuje wartość binarną 0 lub 1	1= timer załączony (ON->OFF) 0= timer wyłączony (OFF->ON).
<b>vi1÷vi4</b>	wskaźnik sygnału wideo dla FGR-4 (CH1÷CH4), przyjmuje wartość binarną 0 lub 1	0= brak sygnału wideo 1= sygnał wideo poprawny
<b>mv1÷mv4</b>	detekcja ruchu w sygnale wideo (FGR-4), przyjmuje wartość binarną 0 lub 1	0= brak detekcji ruchu 1= ruch wykryty
<b>ac</b>	wskaźnik awarii napięcia podstawowego zasilania (AC), przyjmuje wartość binarną 0 lub 1	0= napięcie podstawowe obecne 1= napięcie podstawowe nieobecne
<b>bf</b>	wskaźnik awarii akumulatora zasilania awaryjnego, status pobierany z zasilacza nadzorowanego, przyjmuje wartość binarną 0 lub 1,	0= brak awarii 1= awaria akumulatora
<b>uzv</b>	wartość napięcie zasilania centrali DC [mV]	xxxx
<b>uzi</b>	wartość prądu na wyjściu zasilacza PSR-ECO [mA]	xxxx
<b>log</b>	wskaźnik zalogowania modemu do sieci GSM, przyjmuje wartość binarną 0 lub 1	0= modem niezalogowany do sieci GSM 1= modem zalogowany do sieci GSM
<b>jmg</b>	wskaźnik zagłuszania sieci GSM (jamming), przyjmuje wartość binarną 0 lub 1	0= brak zagłuszania GSM 1= zagłuszanie sieci GSM
<b>nlv</b>	poziom sieci GSM 1-4, tzw. 'kreski'	1÷4
<b>tha1÷tha4</b>	wskaźnik temperatury dla progu A jeżeli wybrany tryb H, przyjmuje wartość binarną 0 lub 1	1 = temp. > H 0 = temp < (H-histereza)
<b>tla1÷tla4</b>	wskaźnik temperatury dla progu A jeżeli wybrany tryb L, przyjmuje wartość binarną 0 lub 1	1 = temp. < L 0 = temp > (L+histereza)
<b>thb1÷thb4</b>	wskaźnik temperatury dla progu B jeżeli wybrany tryb H, przyjmuje wartość binarną 0	1 = temp. > H 0 = temp < (H-histereza)

	lub 1	
<b>tlb1÷tlb4</b>	wskaźnik temperatury dla progów B jeżeli wybrany tryb L, przyjmuje wartość binarną 0 lub 1	1 = temp. < L 0 = temp > (L+histereza)
<b>t1v÷t4v</b>	wartość temperatury z czujników T1-T4, interwał pomiaru co 60s, wartość [°C], liczba całkowita ze znakiem	xxxx
<b>ft1÷ft4</b>	wskaźnik awarii czujnika temp., przyjmuje wartość binarną 0 lub 1	0= brak awarii 1= awaria czujnika temp.
<b>ail</b>	wskaźnik wartości wejścia analogowego AI dla progów L, przyjmuje wartość binarną 0 lub 1	1 = (ail < L) 0 = [ail > (L+histereza)]
<b>aih</b>	wskaźnik wartości wejścia analogowego AI dla progów H, przyjmuje wartość binarną 0 lub 1	1 = (aih > H) 0 = [aih < (H-histereza)]
<b>M1÷M16</b>	wartość markerów, przyjmuje wartość binarną 0 lub 1	0= marker wartość 0 1= marker wartość 1
<b>L1÷L8</b>	liczniki wartości całkowitych, 8 niezależnych liczników	-2 147 483 648 ÷ 2 147 483 647
<b>as1÷as4</b>	wskaźnik czuwania pełnego w systemie dla danej strefy, przyjmuje wartość binarną 0 lub 1,	0= brak czuwania pełnego (dozoru) 1= czwanie pełne (dozór)
<b>an1÷an4</b>	wskaźnik czuwania nocnego w systemie dla danej strefy, przyjmuje wartość binarną 0 lub 1, obiekt typu Binary Value	0= brak czuwania (dozoru) 1= czwanie nocne (dozór)
<b>al1÷al4</b>	wskaźnik alarmu w danej strefie, przyjmuje wartość binarną 0 lub 1,	0= brak alarmu 1= stan alarmu
<b>ta1÷ta4</b>	wskaźnik sabotażu w danej strefie, przyjmuje wartość binarną 0 lub 1,	0= brak sabotażu 1= stan sabotażu
<b>fn1÷fn4</b>	stan wyjść przekaźnikowych w panelach dotykowych o adresach TP1-TP4, przyjmuje wartość binarną 0 lub 1,	0= wyjście nieaktywne 1= wyjście aktywne
<b>uid</b>	ID numer kodu użytkownika 1-32 wprowadzony w panelu,	1÷32
<b>abf</b>	wskaźnik niskiego napięcia dowolnego zarejestrowanego urządzenia radiowego w systemie: Aero, RF-4, przyjmuje wartość binarną 0 lub 1	0= brak awarii 1= awaria baterii w urządzeniu RF
<b>alf</b>	wskaźnik utraty połączenia bezprzewodowego w systemie Aero, przyjmuje wartość binarną 0 lub 1	0= łączność bezprzewodowa Aero 1= brak połączenia Aero
<b>sec</b>	czas pracy centrali [s] od czasu restartu, dokładność 1%	sec= xx
<b>fcd</b>	kod awarii xx (patrz SMS STAN)	00= brak awarii xx= awaria
<b>aiv</b>	wartość napięcia wejścia analogowego AI [mV]	xxxx
<b>p1÷p8</b>	wartość poboru mocy z modułów IOE-IQPL [W] (inteligentne gniazda w systemie SmartPLC)	1÷2500
<b>ev1÷ev8</b>	zmiana stanu wejścia łącznika IOE/IO-IQPLC, przyjmuje wartość: 0,1,2,	0= brak 1= przycisk wciśnięty >30ms 2= przycisk wciśnięty >800ms
<b>ip1÷ip8</b>	stanu wejścia łącznika IOE/IO-IQPLC, odświeżany co interwał kontroli łączności, przyjmuje wartość binarną 0 lub 1	0= włącznik wyłączony 1= włącznik włączony
<b>kb1÷kb5</b>	numer aktywnego kanału pilota (modułu RF-4, Keyfob-Aero), przyjmuje wartość binarną 0 lub 1	0= kanał pilota (przycisk) nieaktywne 1= kanał pilota (przycisk) aktywny

		(nadaje)
<b>kfi</b>	numer aktywnego pilota modulu (RF-4, Keyfob-Aero)	1÷42: RF-4 1-16: Keyfob Aero
<b>ctr</b>	stan połączenia ze stacją monitoringu MSR-1	"0" - brak połączenia "1" - połączenie aktywne
<b>tr1, tr2</b>	wskaźnik załączenia termostatu	"0" - grzanie wyłączone, "1" - grzanie załączone
<b>0</b>	wartość binarna 0, obiekt typu Binary Value	0
<b>1</b>	wartość binarna 1	1

#### Funkcja logiczne (If)

Symbol	Opis	Nazwa
<b>==</b>	zwraca prawdę, jeżeli oba argumenty mają identyczną wartość.	równość
<b>!=</b>	zwraca prawdę, jeżeli oba argumenty mają różne wartości	nierówność
<b>_ </b>	zwraca prawdę, jeżeli oba argumenty mają zbocze narastające	równość; zbocze narastające
<b>-- </b>	zwraca prawdę, jeżeli oba argumenty mają zbocze opadające	równość; zbocze opadające
<b>&gt;</b>	zwraca prawdę, jeżeli lewy argument ma większą wartość od prawego	większe niż
<b>&lt;</b>	zwraca prawdę, jeżeli lewy argument ma mniejszą wartość od prawego	mniejsze niż
<b>&gt;=</b>	zwraca prawdę, jeżeli lewy argument ma większą lub równą wartość prawemu	większe lub równe
<b>&lt;=</b>	zwraca prawdę, jeżeli lewy argument ma mniejszą lub równą wartość prawemu	mniejsze lub równe

#### Wynik (Output)

Symbol	Opis	Wartości logiczne
<b>O1÷O32</b>	stan wyjść fizycznych, przyjmuje wartość binarną 0 lub 1	0= wyjście nieaktywne 1= wyjście aktywne
<b>M1÷M16</b>	wartość markerów, przyjmuje wartość binarną 0 lub 1	0= marker wartość 0 1= marker wartość 1
<b>L1÷L8</b>	liczniki wartości całkowitych, 8 niezależnych liczników	-2 147 483 648 ÷ 2 147 483 647

#### Funkcja logiczne, arytmetyczne, powiadomienia. (Then, Else).


Symbol	Opis	Tabela prawdy		
<b>AND</b>	<b>iloczyn logiczny: A1÷A8</b> jest to układ logiczny, który spełnia następujące funkcje: na wyjściu pojawia się sygnał '1' wtedy i tylko wtedy, kiedy wszystkie n sygnały wejściowe posiadają wartość logiczną '1'	A1	An	Output
		0	0	0
		0	1	0
		1	0	0
		1	1	1




<b>OR</b>	<b>suma logiczna: <math>A1 \div A8</math></b> jest to układ sumy logicznej, który daje na wyjściu sygnał '1', jeżeli tę wartość ma co najmniej jeden z sygnałów. Oznacza to, że '0' pojawia się wtedy i tylko wtedy, kiedy oba sygnały są wartości '0'	A1	An	Output
		0	0	0
		0	1	1
		1	0	1
		1	1	1
<b>NAND</b>	<b>zanegowany iloczyn logiczny (NOT AND): <math>A1 \div A8</math></b> jest to układ sumy logicznej, który daje na wyjściu sygnał '1', jeżeli tę wartość ma n-1 sygnałów wejściowych. Oznacza to, że '0' pojawia się wtedy i tylko wtedy, kiedy wszystkie sygnały są wartości '1'	A1	An	Output
		0	0	1
		0	1	1
		1	0	1
		1	1	0
<b>NOR</b>	<b>zanegowana suma logiczna (NOT OR); <math>A1 \div A8</math></b> jest to układ logiczny, który spełnia następujące funkcje: na wyjściu pojawia się sygnał '1' wtedy i tylko wtedy, kiedy wszystkie n sygnały wejściowe posiadają wartość logiczną '0'	A1	An	Output
		0	0	1
		0	1	0
		1	0	0
		1	1	0
<b>XOR</b>	<b>alternatywa wykluczająca: <math>A1 \div A8</math></b> jest to układ na którego wyjściu pojawia się sygnał '1', wtedy i tylko wtedy, gdy jeden z sygnałów wejściowych będzie miał '1'. W przypadku gdy sygnały są równe '0' lub więcej niż jeden ma wartość '1' na wyjściu sygnał będzie równy '0'.	A1	An	Output
		0	0	0
		0	1	1
		1	0	1
		1	1	0
<b>NOT</b>	<b>negacja: A1</b> jest to układ na którego wyjściu pojawia się sygnał '1', wtedy i tylko wtedy, gdy wejście ma sygnał '0', jeżeli na wejściu pojawi się '1' to wyjście ma '0'	A1	Output	
		0	1	
		1	0	
<b>=</b>	<b>przypisanie; A1</b> jest to układ, który przepisuje wartość sygnału wejściowego na wyjście	A1	Output	
		0	0	
		1	1	
<b>-- </b>	<b>zbczce opadające: A1</b> jest to układ, który wygeneruje na wyjściu '1' wtedy i tylko wtedy gdy wejście zmienia stan '1' na '0'	A1	Output	
		1->0	1	
		1	0	
		0	0	
<b>_ </b>	<b>zbczce narastające: A1</b> jest to układ, który wygeneruje na wyjściu '1' wtedy i tylko wtedy gdy wejście zmienia stan '0' na '1'	A1	Output	
		0->1	1	
		1	0	
		0	0	
<b>+</b>	<b>dodawanie: <math>A1 \div A2</math></b> funkcja dodaje argumenty i wpisuje wynik do licznika Lx			
<b>-</b>	<b>odejmowanie: <math>A1 \div A2</math></b> funkcja odejmuje argumenty i wpisuje wynik do licznika Lx			
<b>/</b>	<b>dzielenie: <math>A1 \div A2</math></b> funkcja dzieli dwa argumenty i wpisuje wynik do licznika Lx			
<b>*</b>	<b>mnożenie: <math>A1 \div A2</math></b> funkcja mnoży dwa argumenty i wpisuje wynik do licznika Lx			
<b>WAIT</b>	<b>czekaj: A1</b> funkcja zatrzymuje petlę na czas argumentu [ms] lub podanej wartości			

<b>PRINT</b>	<b>wyświetl informacje: A1÷A2</b> funkcja wyświetla okno z informacją na panelach dotykowych, jako argument A1 można podać tekst info a argument A2 inny argument systemowy np. moc, funkcja łączy A1 i A2	
<b>HINT</b>	<b>wyświetl informacje na pasku: A1÷A2</b> funkcja wyświetla na dolnym pasku paneli dotykowych informację, jako argument A1 można podać tekst info a argument A2 inny argument systemowy np. moc, funkcja łączy A1 i A2	
<b>SMS</b>	<b>wyślij SMS: A1÷A2</b> funkcja generuje SMS pod wskazane numery, jako argument A1 można podać tekst i numery telefonu w formie '\$1,2,3,4,5,6,7,8' a argument A2 inny argument systemowy np. moc, funkcja łączy A1 i A2	

**PRZYKŁADY:**

<b>Nazwa funkcji</b>	<b>PRINT</b>												
<b>Przeznaczenie</b>	Funkcja wypisuje zadany komunikat na oknie panelu TPR. Tworzone jest okno gdzie wyświetlane są komunikaty wraz z godziną ich wystąpienia. Okno posiada historię 7 ostatnich wpisów. Najnowsze wpisy zastępują te starsze. Zastosowanie funkcji PRINT												
<b>Składnia</b>	PRINT(napis,x) lub PRINT(napis) <ul style="list-style-type: none"> <li>• Napis objęty znakami ” tekst który ma zostać wyświetlony max 20 znaków</li> <li>• x dodatkowy parametr zmiennej która ma zostać wyświetlona na końcu napisu tekst</li> </ul>												
<b>Przykład z kreatora</b>	<table border="1"> <thead> <tr> <th>Lp</th> <th>Wynik do</th> <th>Funkcja</th> <th>A1</th> <th>A2</th> <th>A3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>---</td> <td>PRINT</td> <td>Stan wejścia 1</td> <td>I1</td> <td>---</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	A2	A3	1	---	PRINT	Stan wejścia 1	I1	---
Lp	Wynik do	Funkcja	A1	A2	A3								
1	---	PRINT	Stan wejścia 1	I1	---								
<b>Przykład skrypt</b>	<i>PRINT("Stan wejścia 1",I1);</i>												
<b>Uwagi</b>	Aby panel TPR wyświetlał komunikaty PRINT z centrali należy zaznaczyć opcję  w ustawieniach konfiguracyjnych panela TPR												

<b>Nazwa funkcji</b>	<b>HINT</b>
<b>Przeznaczenie</b>	Funkcja wypisuje zadany komunikat na dolnej panelu TPR. Wyświetlany napis nie jest zapamiętywany, następane wywołanie funkcji HINT() lub inny komunikat systemowy spowoduje nadpisanie wyświetlanego komunikatu.
<b>Składnia</b>	HINT(napis,x) lub HINT(napis)

	<ul style="list-style-type: none"> <li>• Napis objęty znakami " " tekst który ma zostać wyświetlony max 20 znaków</li> <li>• x dodatkowy parametr zmiennej która ma zostać wyświetlona na końcu napisu tekst</li> </ul>												
<b>Przykład z kreatora logiki</b>	<table border="1"> <thead> <tr> <th>Lp</th> <th>Wynik do</th> <th>Funkcja</th> <th>A1</th> <th>A2</th> <th>A3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>---</td> <td>HINT</td> <td>Zasilanie centrali w mV</td> <td>uzv</td> <td>---</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	A2	A3	1	---	HINT	Zasilanie centrali w mV	uzv	---
Lp	Wynik do	Funkcja	A1	A2	A3								
1	---	HINT	Zasilanie centrali w mV	uzv	---								
<b>Przykład skrypt</b>	HINT("Zasilanie centrali w mV ",uzv); HINT("Awaria oświetlenia LED");												
<b>Uwagi</b>	<p>Aby panel TPR wyświetlał komunikaty HINT należy zaznaczyć opcję</p>  <p>w ustawieniach konfiguracyjnych panela TPR</p>												

<b>Nazwa funkcji</b>	<b>SMS</b>												
<b>Przeznaczenie</b>	Funkcja powoduje wysłanie smsa o dowolnej treści, do określonej grupy adresatów.												
<b>Składnia</b>	<p><b>SMS(napis,x)</b></p> <ul style="list-style-type: none"> <li>• napis objęty znakami " " treść smsa</li> <li>• x dodatkowy parametr zmiennej która ma zostać dołożona na końcu napisu wysyłanego smsa</li> </ul> <p>Możliwe jest kilka wariantów wywołania funkcji np.</p> <ol style="list-style-type: none"> <li>1. SMS("Za duza wilgotnosc ") wysle sms pod 1szy zaprogramowany numer</li> <li>2. SMS("Za duza wilgotnosc \$2,3") wysle 2 smsy pod zaprogramowany numer 2 i numer 3 znak \$ separuje treść od numerów adresatów</li> <li>3. SMS(Za duza wilgotnosc \$2,3",aiv) jak w punkcie 2, dodatkowo zostanie dołączona wartość zmiennej aiv ( wartość mierzonego napięcia na wejściu A1)</li> </ol>												
<b>Przykład z kreatora logiki</b>	<table border="1"> <thead> <tr> <th>Lp</th> <th>Wynik do</th> <th>Funkcja</th> <th>A1</th> <th>A2</th> <th>A3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>---</td> <td>SMS</td> <td>Alarm , wysoka wilgotnosc</td> <td>aiv</td> <td>---</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	A2	A3	1	---	SMS	Alarm , wysoka wilgotnosc	aiv	---
Lp	Wynik do	Funkcja	A1	A2	A3								
1	---	SMS	Alarm , wysoka wilgotnosc	aiv	---								
<b>Przykładowy skrypt</b>	<p><i>Jednorazowe wysłanie sms gdy wartość napięcia na wejściu analogowym A1 przekroczy 5V</i></p> <pre>int aiv; int M1; main(){ while(1){ aiv=getai(1); if(aiv&gt;5000&amp;&amp;M1==0){ SMS("Alarm, wysoka wilgotnosc ",aiv); M1=1; }; }; };</pre>												

<b>Nazwa funkcji</b>	WAIT(x)										
<b>Przeznaczenie</b>	Funkcja powoduje zatrzymanie wykonywania programu na zadaną liczbę ms (1000ms=1s)										
<b>Składnia</b>	WAIT(x) x opóźnienie w ms										
<b>Przykład z kreatora logiki</b>	<table border="1"> <thead> <tr> <th>Lp</th> <th>Wynik do</th> <th>Funkcja</th> <th>A1</th> <th>A2</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>---</td> <td>WAIT</td> <td>1000</td> <td>---</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	A2	1	---	WAIT	1000	---
Lp	Wynik do	Funkcja	A1	A2							
1	---	WAIT	1000	---							
<b>Przykład skryptu</b>	<p><i>Cykliczne załączanie/wyłączanie wyjścia O8 1s/0.5s gdy czuwa 1sza strefa</i></p> <pre>int as1; main(){ while(1){ gbenv(); if(as1==1){ seto(8,1); <b>WAIT(1000);</b> seto(8,0); <b>WAIT(500);</b> }; }; };</pre>										
<b>Uwagi</b>	Funkcja blokuje wykonywanie skryptu na zadany czas, jeżeli jest to niedopuszczalne to należy użyć przekaźników czasowych które nie blokują wykonywania skryptu (działają asynchronicznie)										

<b>Nazwa funkcji</b>	ARMF(x)										
<b>Przeznaczenie</b>	Funkcja uzbraja zadaną strefę w czuwaniu pełnym										
<b>Składnia</b>	ARMF(x) x- numer zazbrajanej strefy 1-4										
<b>Przykład z kreatora logiki</b>	<table border="1"> <thead> <tr> <th>Lp</th> <th>Wynik do</th> <th>Funkcja</th> <th>A1</th> <th>A2</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>---</td> <td>ARMF</td> <td>1</td> <td>---</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	A2	1	---	ARMF	1	---
Lp	Wynik do	Funkcja	A1	A2							
1	---	ARMF	1	---							
<b>Przykład skryptu</b>	<p><i>Automatyczne uzbrojenie/rozbrojenie strefy 3 gdy czuwają strefy 1 i 2</i></p> <pre>int as1; int as2; int as3; main(){ while(1){ gbenv(); if(as1==1&amp;&amp;as2==1){ if(as3==0){</pre>										

	<pre> ARMF(3); }; } else {     if(as3==1){         DISARM(3);     }; }; }; } </pre>
<b>Uwagi</b>	<b>Dostępne od wersji oprogramowania v1.8 centrali</b>

<b>Nazwa funkcji</b>	<b>DISARM(x)</b>										
<b>Przeznaczenie</b>	Funkcja rozbraja zadaną strefę w czuwaniu pełnym										
<b>Składnia</b>	DISARM(x) x- numer rozbrajanej strefy 1-4										
<b>Przykład z kreatora logiki</b>	<table border="1"> <thead> <tr> <th>Lp</th> <th>Wynik do</th> <th>Funkcja</th> <th>A1</th> <th>A2</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>---</td> <td>ARMF</td> <td>1</td> <td>---</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	A2	1	---	ARMF	1	---
Lp	Wynik do	Funkcja	A1	A2							
1	---	ARMF	1	---							
<b>Przykład skryptu</b>	<p><i>Automatyczne rozbrojenie strefy 1 gdy czuwają strefy 1 i 2</i></p> <pre> int as1; int as2; int as3; main(){ while(1){ gbenv(); if(as1==1&amp;&amp;as2==1){     if(as3==0){         ARMF(3);     }; } else {     if(as3==1){         DISARM(3);     }; }; }; } </pre>										
<b>Uwagi</b>	<b>Dostępne od wersji oprogramowania v1.8 centrali</b>										

<b>Nazwa funkcji</b>	<b>ARMN(x)</b>
<b>Przeznaczenie</b>	Funkcja uzbraja zadaną strefę w czuwaniu nocnym (działają tylko wejścia z flagą czuwanie nocne)
<b>Składnia</b>	ARMN(x) x- numer zabrajananej strefy 1-4

Przykład z kreatora logiki	<b>Lp</b>	<b>Wynik do</b>	<b>Funkcja</b>	<b>A1</b>
	1	---	ARMN	1
Przykład skryptu				
Uwagi	Dostępne od wersji oprogramowania v1.8 centrali			

Nazwa funkcji	seto(x,y)								
Przeznaczenie	Funkcja ustawiająca zadane wyjście centrali								
Składnia	seto(x,y) <ul style="list-style-type: none"> <li>• x numer wyjścia które ma zostać załączone/wyłączone</li> <li>• y wartość logiczna 1 = wyjście załączone 0 = wyjście wyłączone</li> </ul>								
Przykład z kreatora logiki	<table border="1"> <tr> <td><b>Lp</b></td> <td><b>Wynik do</b></td> <td><b>Funkcja</b></td> <td><b>A1</b></td> </tr> <tr> <td>1</td> <td>01</td> <td>=</td> <td>tk1</td> </tr> </table>	<b>Lp</b>	<b>Wynik do</b>	<b>Funkcja</b>	<b>A1</b>	1	01	=	tk1
<b>Lp</b>	<b>Wynik do</b>	<b>Funkcja</b>	<b>A1</b>						
1	01	=	tk1						
Przykład skryptu	<i>Załączanie wyjścia 01 zgodnie ze stanem Timera 1</i> <pre>int O1; int tk1; main(){ while(1){ gbenv(); O1=tk1; seto(1,O1); }; };</pre>								
Uwagi	Aby dostęp do wyjścia był możliwy należy dla sterowanego wyjścia włączyć opcję w zakładce wyjścia <ul style="list-style-type: none"> <li><input type="checkbox"/> Potwierdzenie pulsami zał./wył. czuwania</li> <li><input checked="" type="checkbox"/> Logic processor</li> <li><input type="checkbox"/> Awaria zbiorcza</li> </ul> Czas załączenia wyjścia jest wtedy nadpisywany przez moduł logiki.								

Nazwa funkcji	geto(x)
Przeznaczenie	Funkcja pobierająca stan logiczny wybranego wyjścia centrali
Składnia	y=geto(x) <ul style="list-style-type: none"> <li>• x – zmienna logiczna do której będzie przypisany stan wyjścia 0=wyłączone 1= załączone</li> <li>• y – numer wyjścia stan którego ma zostać odczytany 1-32</li> </ul>

Przykład z kreatora logiki	<b>Lp</b>	<b>A1</b>	<b>Funkcja</b>	<b>A2</b>	<b>Logika</b>		
	1	O1	==	1	---	+	-
Przykład skryptu	<i>Informacja na panelu TPR o załączeniu wyjścia O1</i> <pre>int O1; int O1p; main(){ while(1){ O1=geto(1); if((O1p==0&amp;&amp;O1==1)){ HINT("Zalaczylo sie wyjscie O1"); }; O1p=O1; }; };</pre>						
Uwagi							

Nazwa funkcji	geti(x)								
Przeznaczenie	Funkcja pobierająca stan logiczny wybranego wejścia centrali								
Składnia	<pre>y=geti(x)</pre> <ul style="list-style-type: none"> <li>y – zmienna logiczna do której będzie przypisany stan wejścia wartość 1= naruszone 0=nienaruszone</li> <li>x – numer wejścia stan którego ma zostać odczytany 1-48</li> </ul>								
Przykład z kreatora logiki	<table border="1"> <thead> <tr> <th>Lp</th> <th>Wynik do</th> <th>Funkcja</th> <th>A1</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>O1</td> <td>=</td> <td>I1</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	1	O1	=	I1
Lp	Wynik do	Funkcja	A1						
1	O1	=	I1						
Przykład skryptu	<i>Przepisanie stanu wejścia I1 na wyjście O1 (śledzenie wejścia)</i> <pre>int I1; int O1; main(){ while(1){ I1=geti(1); O1=I1; seto(1,O1); }; };</pre>								
Uwagi									

Nazwa funkcji	gett(x)
Przeznaczenie	Funkcja pobierająca wartość temperatury z czujnika tsr-1

e															
<b>Składnia</b>	<code>y=gett(x)</code> <ul style="list-style-type: none"> <li>• y – zmienna do której będzie przypisana wartość temperatury z czujnika</li> <li>• x – numer czujnika 1-4</li> </ul>														
<b>Przykład z kreatora logiki</b>	<table border="1"> <thead> <tr> <th>Lp</th> <th>A1</th> <th>Funkcja</th> <th>A2</th> <th>Logika</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>t1v</td> <td>==</td> <td>10</td> <td>---</td> <td>+</td> <td>-</td> </tr> </tbody> </table>	Lp	A1	Funkcja	A2	Logika			1	t1v	==	10	---	+	-
Lp	A1	Funkcja	A2	Logika											
1	t1v	==	10	---	+	-									
<b>Przykład skryptu</b>	<p><i>Informacja na panelu TPR o temperaturze ujemnej mierzonej przez czujnik temp. Nr 1</i></p> <pre>int t1v; main(){ while(1){ <b>t1v=gett(1);</b> if(t1v&lt;0){ HINT("Jest mroz T=",t1v); } else { HINT("Temperatura dodatnia"); }; WAIT(30000); }; };</pre>														
<b>Uwagi</b>	<p><b>Funkcja zwraca tylko wartość całkowitą temperatury w stopniach Celcjusza</b></p> <p><b>Gdy brak czujnika lub awaria to funkcja zwraca -999</b></p>														

<b>Nazwa funkcji</b>	<code>getenv()</code>
<b>Przeznaczenie</b>	Funkcja pobierająca wartość wszystkich zmiennych systemowych
<b>Składnia</b>	<code>getenv()</code>
<b>Przykład z kreatora logiki</b>	
<b>Przykład skryptu</b>	<pre>int uzv; main(){ while(1){ <b>gbenv();</b> HINT("napiecie zasilania U[mV]",uzv); WAIT(1000); }; };</pre>
<b>Uwagi</b>	<p><b>Funkcja zwraca tylko wartość całkowitą temperatury w stopniach Celcjusza</b></p> <p><b>Gdy brak czujnika lub awaria to funkcja zwraca -999</b></p>



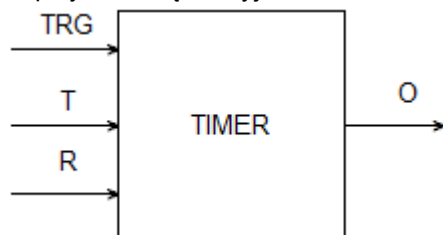
<b>Nazwa funkcji</b>	gettw(x)
<b>Przeznaczenie</b>	Funkcja pobierająca wartość temperatury z czujnika bezprzewodowego RHT-Aero (1-8)
<b>Składnia</b>	y=gettw(x) <ul style="list-style-type: none"> <li>• y – zmienna do której będzie przypisana wartość temperatury z czujnika</li> <li>• x – numer czujnika 1-8</li> </ul>
<b>Przykład z kreatora logiki</b>	
<b>Przykład skryptu</b>	<i>Informacja na panelu TPR o temperaturze ujemnej mierzonej przez czujnik temp. Nr 1</i> <pre>int twv; main(){ while(1){ twv=gettw(1); if(twv&lt;0){ HINT("Jest mroz T=",twv); } else { HINT("Temperatura dodatnia"); }; }; WAIT(30000); }; };</pre>
<b>Uwagi</b>	<b>Funkcja zwraca tylko wartość całkowitą temperatury w stopniach Celcjusza</b> <b>Gdy błąd odczytu to funkcja zwraca -999</b>

<b>Nazwa funkcji</b>	getthw()
<b>Przeznaczenie</b>	Funkcja pobierająca wartość wilgotności z czujnika bezprzewodowego RHT-Aero (1-8)
<b>Składnia</b>	y=getthw(x) <ul style="list-style-type: none"> <li>• y – zmienna do której będzie przypisana wartość wilgotności z czujnika</li> <li>• x – numer czujnika 1-8</li> </ul>
<b>Przykład z kreatora logiki</b>	
<b>Przykład skryptu</b>	<i>Informacja na panelu TPR o wilgotności mierzonej przez czujnik wilg. Nr 1</i> <pre>int thv; main(){ while(1){ thv=getthw(1);</pre>

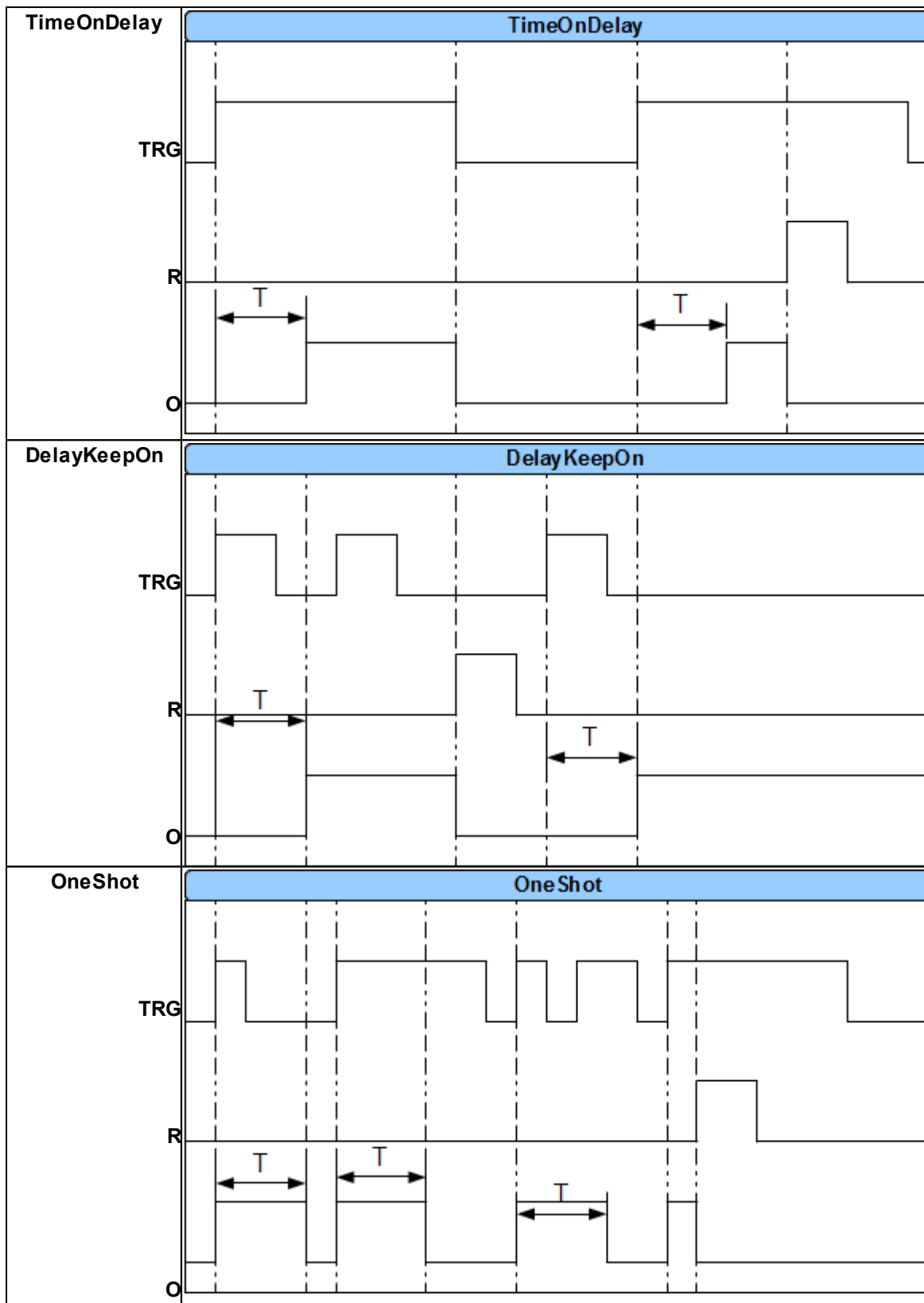
	<pre> iff(thv&gt;70){ HINT("Za duza wilg. H=",thv); } else { HINT("Wilgotnosc OK."); }; }; WAIT(30000); }; }; </pre>
<b>Uwagi</b>	<b>Funkcja zwraca tylko wartość całkowitą wilgotności w %. Gdy błąd odczytu to funkcja zwraca 255.</b>

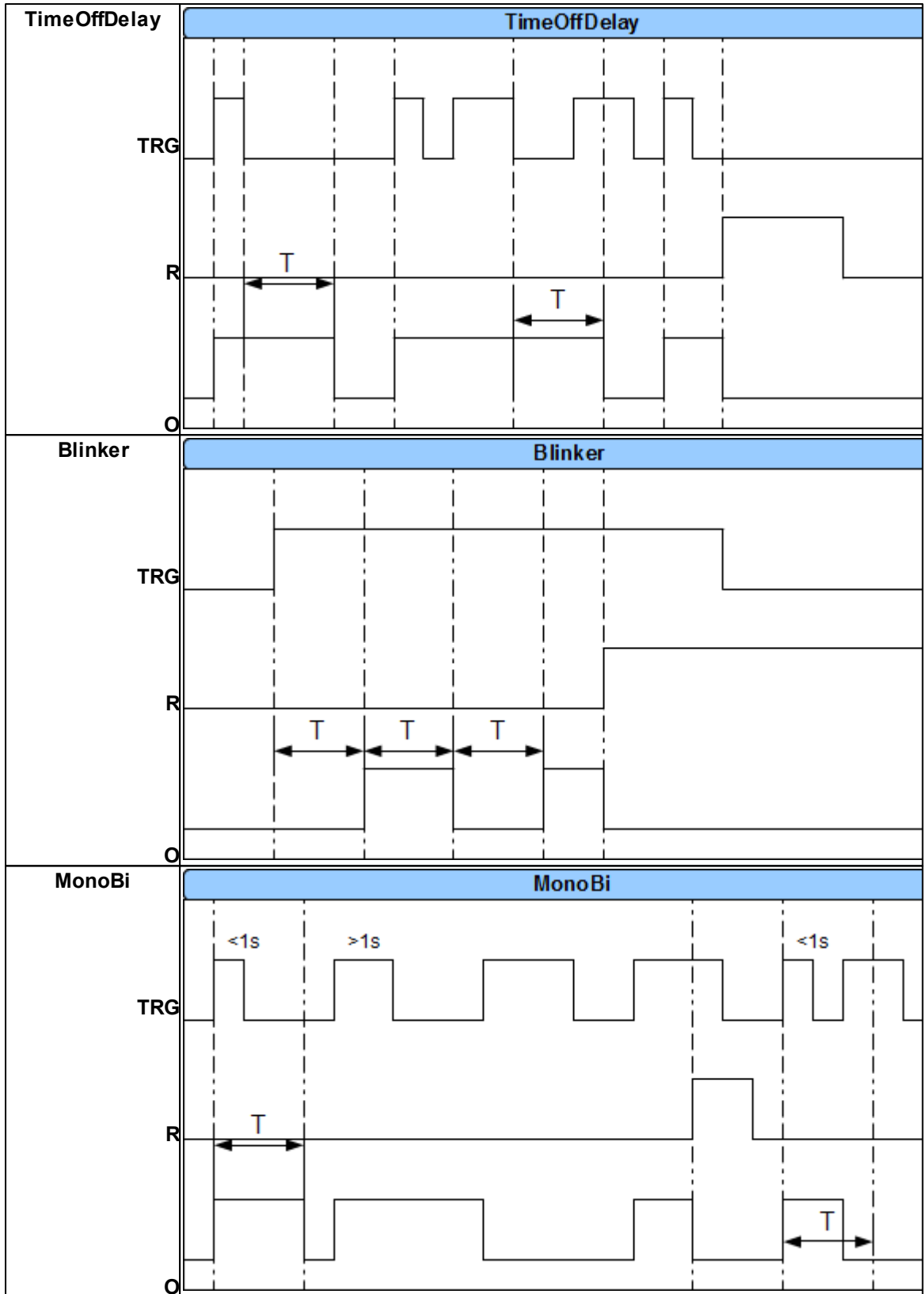
## 1.2 Przełączniki czasowe.

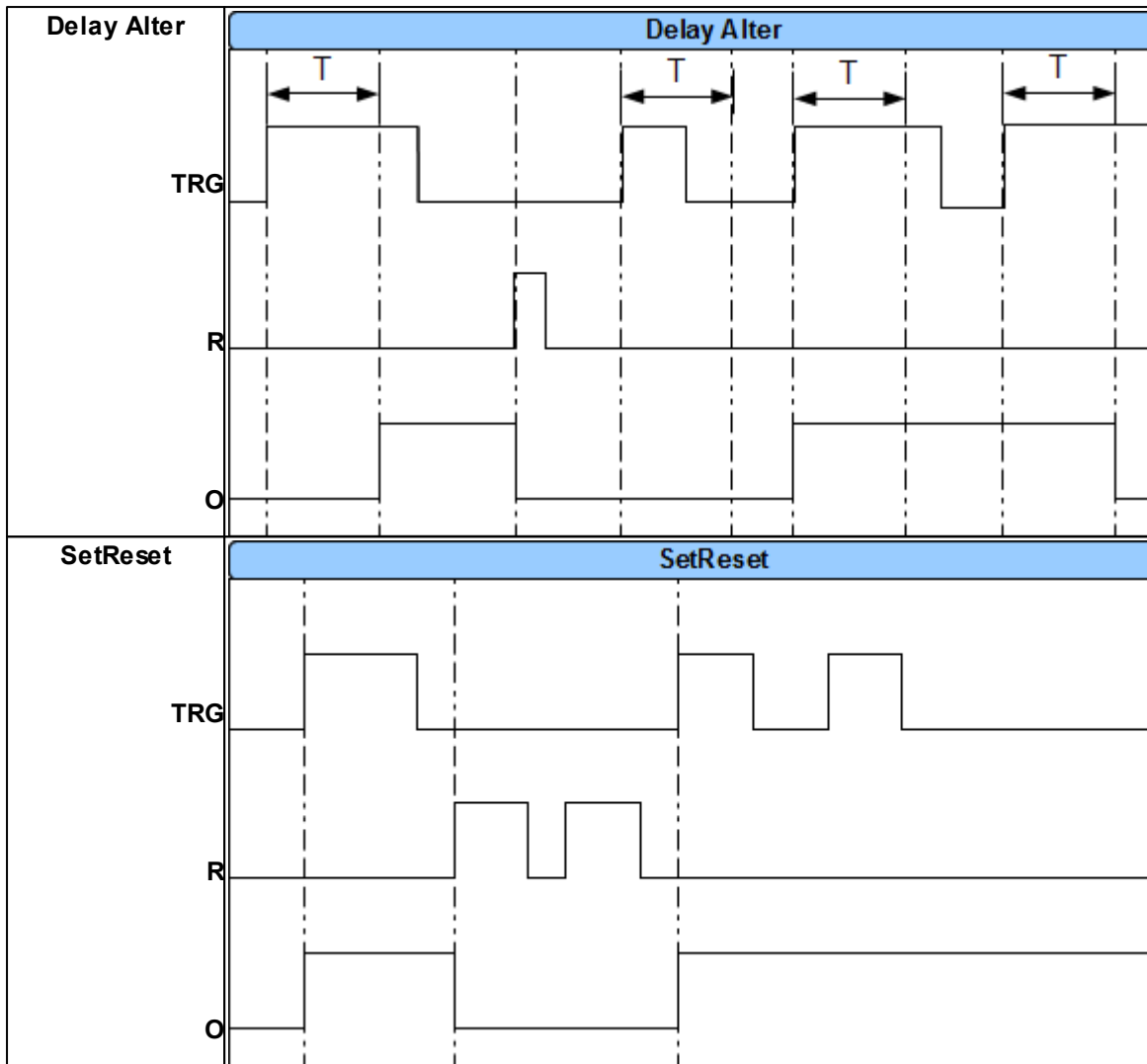
Funkcje czasowo-logiczne pozwalają na wykonanie programowanych przełączniki czasowe, wyzwalania i reset timerów (bloków) identyczne jak argumenty w funkcjach logicznych a wyniki zapisywane są do wyjść lub markerów,



Symbol	Parametr	Opis
TRG	Trigger	sygnał wyzwalający
T	Czas	czas timera, funkcji
R	Reset	sygnał resetujący
O	Output	wyjście funkcji
TIMER	Typ timera	typ funkcji czasowo/licznikowej

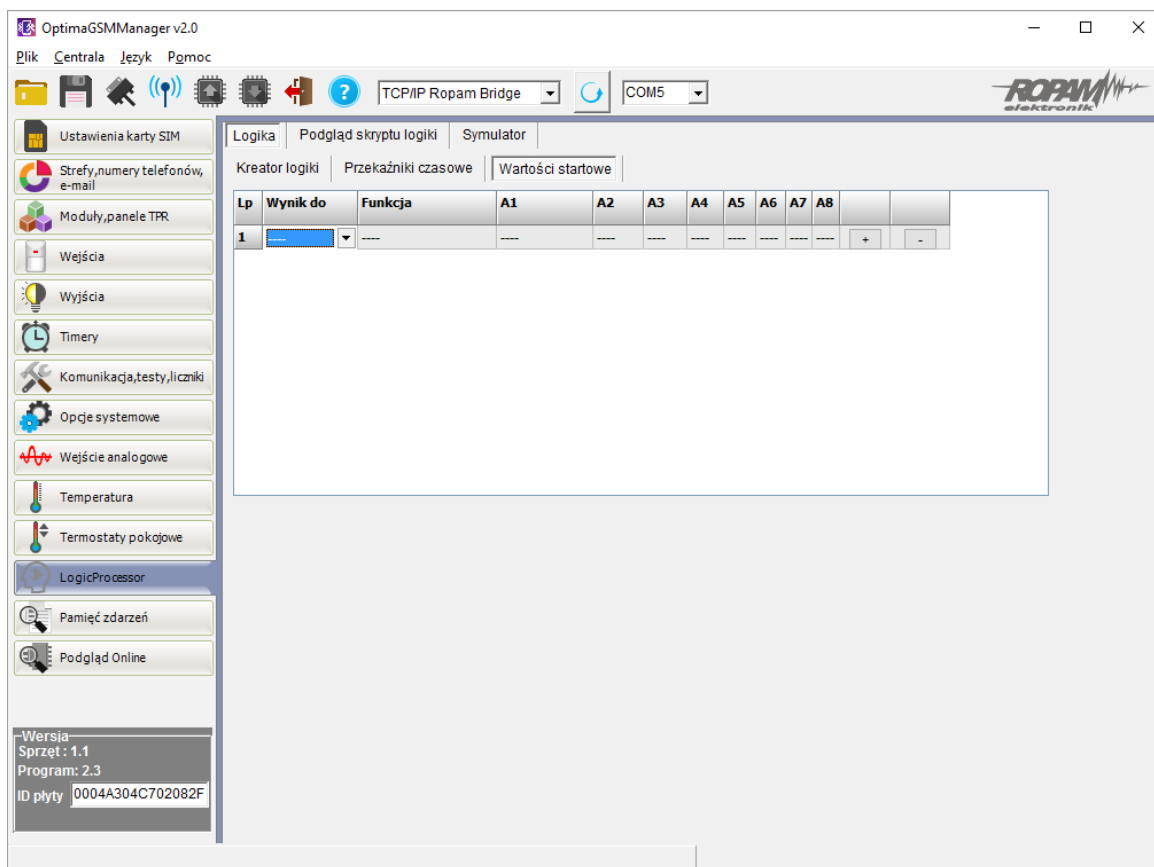




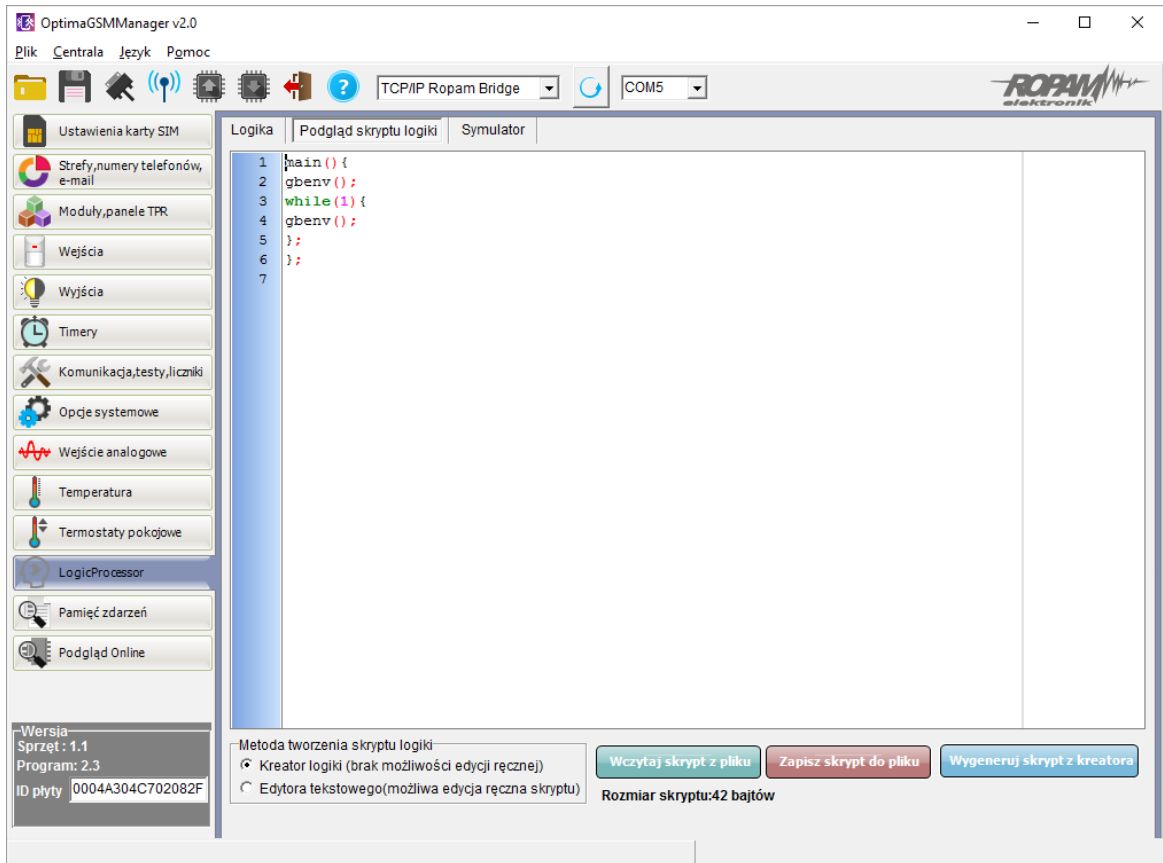


### 1.3 Wartości startowe.

W celu uniknięcia stanów nieustalonych podczas startu skryptu LP, można ustawić wartości startowe dla skryptu bazujące na zasobach systemu przeznaczonych dla Logic Processora.



## 1.4 Podgląd skryptu.

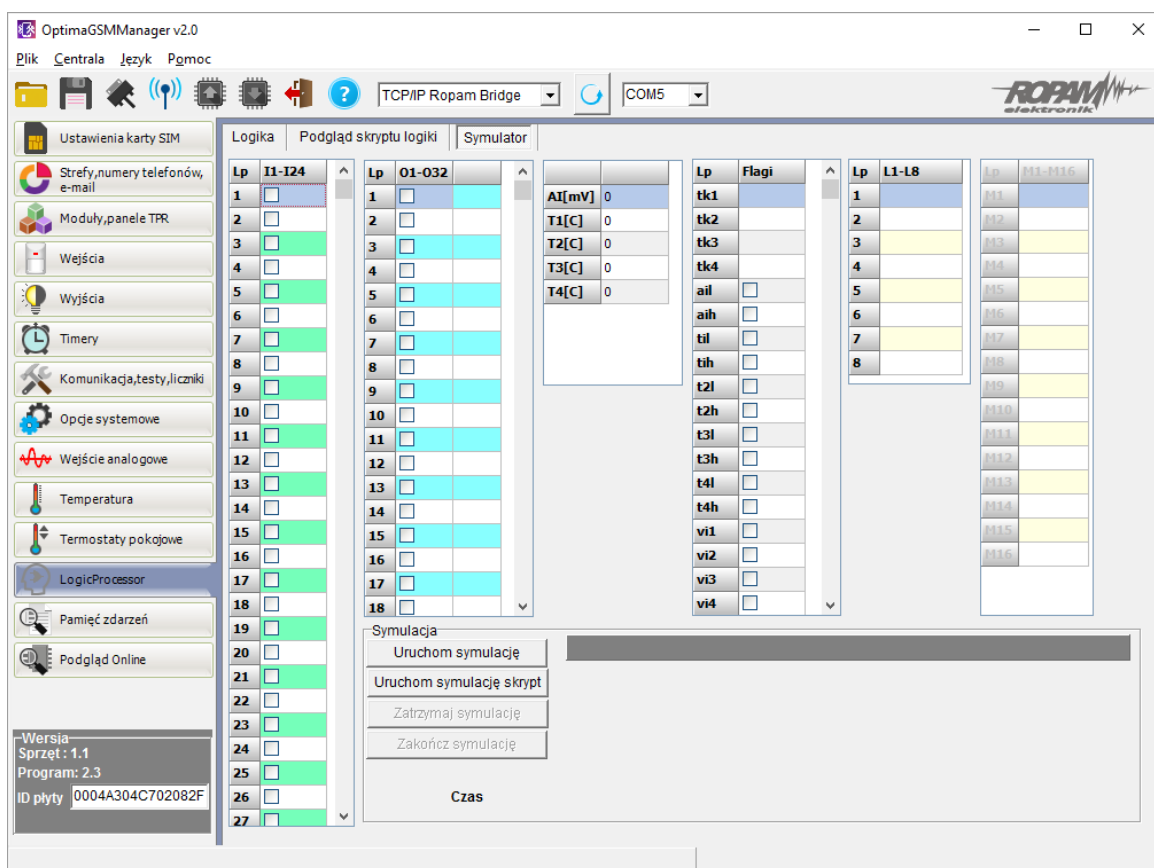


The screenshot displays the OptimaGSMMManager v2.0 application window. The interface includes a menu bar (Plik, Centrala, Język, Pomoc), a toolbar with various icons, and a sidebar with numerous configuration options such as 'Ustawienia karty SIM', 'Strefy,numery telefonów, e-mail', and 'LogicProcessor'. The main area is divided into three tabs: 'Logika', 'Podgląd skryptu logiki', and 'Symulator'. The 'Podgląd skryptu logiki' tab is active, showing a code editor with the following C-like code:

```
1 main () {  
2   gbenv ();  
3   while (1) {  
4     gbenv ();  
5   };  
6 };  
7
```

At the bottom of the window, there is a section for 'Metoda tworzenia skryptu logiki' with two radio buttons: 'Kreator logiki (brak możliwości edycji ręcznej)' (selected) and 'Edytora tekstowego(możliwa edycja ręczna skryptu)'. To the right of these buttons are three action buttons: 'Wczytaj skrypt z pliku', 'Zapisz skrypt do pliku', and 'Wygeneruj skrypt z kreatora'. Below these buttons, the text 'Rozmiar skryptu:42 bajtów' is displayed.

## 1.5 Symulator.



## 1.6 Noty aplikacyjne

Dane do serwera FTP z danymi technicznymi, notami aplikacyjnymi, firmware.  
Wszystko w jednym miejscu dostępne przez klienta FTP

serwer: <ftp.ropam.com.pl>

login: anonymous@ropam.com.pl

hasło: pozostaw puste pole i kliknij OK



---

Notatki:



# LogicProcessor\_OptimaGSM

