



## NOTA APLIKACYJNA

<b>APLIKACJA</b>	Funkcje logiczne w LogicProcessor
<b>URZĄDZENIA:</b>	OptimaGSM lub OptimaGSM-PS
<b>WERSJA</b>	1.0PC
<b>DATA</b>	22.09.2015

Opis dostępnych funkcji logicznych. Oprócz funkcji arytmentyczno-logicznych moduł LogicProcessora obsługuje zestaw funkcji wykorzystujących specyficzne zasoby centrali . Umożliwia to stworzenie zupełnie nowych funkcjonalności lub dopasowanie już istniejących do specyficznych wymagań klientów. Jest to unikatowa cecha centrali, która elastycznością wyróżnia się na rynku wobec urządzeń do tego segmentu.

### Opis dostępnych funkcji (wersja oprogramowania v1.9)

<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>	Aby panel TPR wyświetlał komunikaty HINT należy zaznaczyć opcję  w ustawieniach konfiguracyjnych panela TPR												

<b>Nazwa funkcji</b>	<b>SMS</b>												
<b>Przeznaczenie</b>	Funkcja powoduje wysłanie smsa o dowolnej treści , do określonej grupy adresatów.												
<b>Składnia</b>	SMS(napis,x) <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>SMS("Za duza wilgotnosc ") wysle sms pod 1szy zaprogramowany numer</li> <li>SMS("Za duza wilgotnosc \$2,3") wysle 2 smsy pod zaprogramowany numer 2 i numer 3 znak \$ separuje treść od numerów adresatów</li> <li>SMS(Za duza wilgotnosc \$2,3",aiv) jak w punkcie 2, dodatkowo zostanie dołączona wartość zmiennej aiv ( wartość mierzonego napięcia na wejściu AI)</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 AI 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>	<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>										

<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){ <b>ARMF(3);</b> }; } else { if(as3==1){ DISARM(3); }; }; };</pre>										

	}; }
Uwagi	Dostępne od wersji oprogramowania v1.8 centrali

Nazwa funkcji	ARMN(x)								
Przeznaczenie	Funkcja uzbraja zadaną strefę w czuwaniu nocnym (działają tylko wejścia z flagą czuwanie nocne)								
Składnia	ARMN(x) x- numer zazbrajanej strefy 1-4								
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>---</td> <td>ARMN</td> <td>1</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	1	---	ARMN	1
Lp	Wynik do	Funkcja	A1						
1	---	ARMN	1						
Przykład skryptu									
Uwagi	Dostępne od wersji oprogramowania v1.8 centrali								

Nazwa funkcji	DISARM(x)								
Przeznaczenie	Funkcja rozbrajająca zadaną strefę								
Składnia	DISARM(x) x- numer rozbrajanej strefy 1-4								
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>---</td> <td>DISARM</td> <td>4</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	1	---	DISARM	4
Lp	Wynik do	Funkcja	A1						
1	---	DISARM	4						
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"> <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>tk1</td> </tr> </tbody> </table>	Lp	Wynik do	Funkcja	A1	1	O1	=	tk1
Lp	Wynik do	Funkcja	A1						
1	O1	=	tk1						
Przykład skryptu	<i>Załączanie wyjścia O1 zgodnie ze stanem Timera 1</i> <pre>int O1; int tk1; main(){ while(1){ gbenv(); O1=tk1; seto(1,O1); }; };</pre>								

<b>Uwagi</b>	<p>Aby dostęp do wyjścia był możliwy należy dla sterowanego wyjścia włączyć opcję w zakładce wyjścia</p> <p><input type="checkbox"/> Potwierdzenie pulsami zał./wył. czuwania</p> <p><input checked="" type="checkbox"/> Logic processor</p> <p><input type="checkbox"/> Awaria zbiorcza</p> <p>Czas załączenia wyjścia jest wtedy nadpisywany przez moduł logiki.</p>

<b>Nazwa funkcji</b>	geto(x)																			
<b>Przeznaczenie</b>	Funkcja pobierająca stan logiczny wybranego wyjścia centrali																			
<b>Składnia</b>	<p>y=geto(y)</p> <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>																			
<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>O1</td> <td>==</td> <td>1</td> <td>---</td> <td>+</td> <td>-</td> </tr> </tbody> </table>						Lp	A1	Funkcja	A2	Logika			1	O1	==	1	---	+	-
Lp	A1	Funkcja	A2	Logika																
1	O1	==	1	---	+	-														
<b>Przykład skryptu</b>	<p>Informacja na panelu TPR o załączeniu wyjścia O1</p> <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>																			
<b>Uwagi</b>																				

<b>Nazwa funkcji</b>	geti(x)										
<b>Przeznaczenie</b>	Funkcja pobierająca stan logiczny wybranego wejścia centrali										
<b>Składnia</b>	<p>y=geti(y)</p> <ul style="list-style-type: none"> <li>x – zmienna logiczna do której będzie przypisany stan wejścia wartość 1= naruszone 0=nienaruszone</li> <li>y – numer wejścia stan którego ma zostać odczytany 1-48</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> </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								
<b>Przykład skryptu</b>	<p>Przepisanie stanu wejścia I1 na wyjście O1 (śledzenie wejścia)</p> <pre>int I1; int O1; main(){ while(1){ I1=geti(1); O1=I1; }; };</pre>										

	seto(1,O1); }; };
<b>Uwagi</b>	

<b>Nazwa funkcji</b>	gett(x)						
<b>Przeznaczenie</b>	Funkcja pobierająca wartość temperatury z czujnika tsr-1						
<b>Składnia</b>	y=gett(y) <ul style="list-style-type: none"> <li>x – zmienna do której będzie przypisana wartość temperatury z czujnika</li> <li>y – numer czujnika 1-4</li> </ul>						
<b>Przykład z kreatora logiki</b>	<b>Lp</b>	<b>A1</b>	<b>Funkcja</b>	<b>A2</b>	<b>Logika</b>		
	1	t1v	==	10	---	<input data-bbox="933 683 981 716" type="button" value="+"/>	<input data-bbox="1029 683 1077 716" type="button" value="-"/>
<b>Przykład skryptu</b>	<i>Informacja na panelu TPR o temperaturze ujemnej mierzonej przez czujnik temp. Nr 1</i> <pre>int t1v; main(){ while(1){ t1v=gett(1); if(t1v&lt;0){ HINT("Jest mroz T=",t1v); } else { HINT("Temperatura dodatnia"); }; }; WAIT(30000); }; };</pre>						
<b>Uwagi</b>	<b>Funkcja zwraca tylko wartość całkowitą temperatury w stopniach Celcjusza Gdy brak czujnika lub awaria to funkcja zwraca -999</b>						

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