

# STRUCTURED CONTROL LANGUAGE

## 1. PODSTAWOWE FUNKCJE

PODSTAWOWE FUNKCJE	
$Q = A$	$\#Q := \#A;$
$Q = A \cap B$	$\#Q := \#A \text{ AND } \#B;$
$Q = A \cup B$	$\#Q := \#A \text{ OR } \#B;$
$Q = \bar{A}$	$\#Q := \text{NOT } \#A;$

PRZYKŁADY	
$Q = \bar{A} \cap B$	$\#Q := \text{NOT } \#A \text{ AND } \#B;$
$Q = A \cup \bar{B}$	$\#Q := \#A \text{ OR NOT } \#B;$
$Q = \bar{A} \cap (B \cap C)$	$\#Q := \text{NOT } \#A \text{ AND } (\#B \text{ OR } \#C);$
$Q = (A \cup B) \cap \bar{C}$	$\#Q := (\#A \text{ OR } \#B) \text{ AND NOT } \#C;$

KONSTRUKCJE	
IF	<pre>IF #S1 THEN   L1 := 1; END_IF;</pre>
IF + ELSE	<pre>IF #S1 AND #S2 THEN   L2 := 1; ELSE   L2 := 0; END_IF;</pre>
IF + IFELS	<pre>IF #S1 AND #B1 THEN   #L1 := 1; ELSIF #S2 THEN   #L2 := 1; END_IF;</pre>
CASE OF	<pre>CASE #krok OF   0:     Y1 := 1;     IF #B2 THEN       #krok := 1;     END_IF;   1:     Y1 := 0;     IF #B1 THEN       #krok := 0;     END_IF; END_CASE;</pre>

RS & SR	
RS (set reset) dominacja resetu	<pre> IF #SET THEN   #ON := 1; END_IF;  IF #RESET THEN   #ON := 0; END_IF; </pre>
RS (reset set) dominacja setu	<pre> IF #RESET THEN   #ON := 1; END_IF;  IF #SET THEN   #ON := 0; END_IF; </pre>

## 2. TON JAKO GENERATOR IMPULSU

```

#TON1(IN:=NOT #TON2.Q,
      Q=>#L1,
      PT:=T#500ms);
#TON2(IN:=#TON1.Q,
      PT:=T#500ms);

```

### 3. CZASÓWKI

<p><b>TP – Time Pulse</b> (impuls zadany)</p> <p>po podaniu sygnału na czasówkę, to ta będzie podawać sygnał przez podany czas</p>	<pre>#TP(IN:=_bool_in_,     PT:=_time_in_,     Q=&gt;_bool_out_,     ET=&gt;_time_out_);</pre> <p>IN – input PT – pulse time Q – output ET – Elapsed Time</p>
<p><b>TON – Time On Delay</b> (opóźnienie włączenia)</p> <p>po podaniu sygnału, zostaje on przekazany dalej po upływie danego czasu sygnał musi być podawany cały przez dany czas aby czasówka zadziałała</p>	<pre>#TON(IN:=_bool_in_,     PT:=_time_in_,     Q=&gt;_bool_out_,     ET=&gt;_time_out_);</pre> <p>IN – input PT – pulse time Q – output ET – Elapsed Time</p>
<p><b>TOF– Time Off Delay</b> (opóźnienie wyłączenia)</p> <p>po podaniu sygnału i jego zniknięciu na czasówkę, będzie on dalej przekazywany przez dany czas</p>	<pre>#TOF(IN:=_bool_in_,     PT:=_time_in_,     Q=&gt;_bool_out_,     ET=&gt;_time_out_);</pre> <p>IN – input PT – pulse time Q – output ET – Elapsed Time</p>
<p><b>TONR – Time On Delay Retentive</b> (opóźnienie włączenia z podtrzymanym czasem)</p> <p>po podaniu sygnału, zostaje on przekazany dalej po upływie danego czasu gdy sygnał zniknie na wejściu czasówki to i tak ona dalej podaje sygnał, aby wyłączyć czasówkę należy podać sygnał na wejście R</p>	<pre>#TONR(IN:=_bool_in_,     R:=_bool_in_,     PT:=_time_in_,     Q=&gt;_bool_out_,     ET=&gt;_time_out_);</pre> <p>IN – input R – reset PT – pulse time Q – output ET – Elapsed Time</p>

#### 4. LICZNIKI

<p><b>CTU – Count Up</b></p> <p>licznik, który dodaje, po spełnieniu warunku PV podaje sygnał na Q</p>	<pre>#CTU(CU:=#S1, R:=#RESET, PV:=5, Q=&gt;#L1);</pre> <p>CU – count up R – reset PV – preset value Q – output CV – current value</p>
<p><b>CTD – Count Down</b></p> <p>licznik, który odejmuje, po spełnieniu warunku CV=0 podaje sygnał na Q</p>	<pre>#CTD(CD:=#S1, R:=#RESET, PV:=5, Q=&gt;#L1);</pre> <p>CD – count down R – reset PV – preset value Q – output CV – current value</p>
<p><b>CTUD Count Up / Down</b></p> <p>licznik, który ma możliwość dodawania i odejmowania, po spełnieniu warunku PV podaje sygnał na wyjście</p>	<pre>#CTUD(CU:=#S1, CD:=#S2, R:=#RESET, LD:=#S3, PV:=10, QU=&gt;#L1, QD=&gt;#L2);</pre> <p>CU – count up CD – count down R – reset PV – preset value LD – load (ładuje PV na CV) QU – output (jeśli PV = CV to da sygnał) QD – output (jeśli PV = 0 to da sygnał ) CV – current value</p>