

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 (reset set) SR (set reset) nie ma możliwości dominacji setu ani resetu	<pre>IF #SET THEN #ON := 1; END_IF; IF #RESET THEN #ON := 0; END_IF;</pre>

2. CZASÓWKI

<p>TP – Time Pulse (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=>_bool_out_, ET=>_time_out_);</pre> <p>IN – input PT – pulse time Q – output ET – Elapsed Time</p>
<p>TON – Time On Delay (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=>_bool_out_, ET=>_time_out_);</pre> <p>IN – input PT – pulse time Q – output ET – Elapsed Time</p>
<p>TOFF – Time Off Delay (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=>_bool_out_, ET=>_time_out_);</pre> <p>IN – input PT – pulse time Q – output ET – Elapsed Time</p>
<p>TONR – Time On Delay Retentive (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=>_bool_out_, ET=>_time_out_);</pre> <p>IN – input R – reset PT – pulse time Q – output ET – Elapsed Time</p>

3. TON JAKO GENERATOR IMPULSU

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

4. LICZNIKI

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