PROGRAM PLC\_PRG

VAR

CODE\_1 : Code\_1 ; //Instance of function Code\_1

IN1 : LREAL;

IN2 : LREAL;

IN3 : LREAL;

IN4 : LREAL;

IN5 : LREAL;

IN6 : LREAL;

IN7 : LREAL;

END\_VAR

IN1 := 5;

IN2 := 5 ;

IN3 := 5 ;

IN4 := 5 ;

IN5 := 5 ;

IN6 := 5 ;

IN7 := 5 ;

CODE\_1 (In1 := IN1, In2 := IN2, In3 := IN3, In4 := IN4, In5 := IN5, In6 := IN6, In7 := IN7);

FUNCTION\_BLOCK Code\_1

VAR\_INPUT

In1: LREAL;

In2: LREAL;

In3: LREAL;

In4: LREAL;

In5: LREAL;

In6: LREAL;

In7: LREAL;

END\_VAR

VAR\_OUTPUT

Out1: ARRAY[0..1] OF BOOL;

Out2: BOOL;

END\_VAR

VAR

Input\_Multip : LREAL ;

Input\_Multip\_int : DINT ;

INPUTTOINDEX\_3D\_1 : InputToIndex\_3D ; //first instance of function InputToIndex\_3D

INPUTTOINDEX\_3D\_2 : InputToIndex\_3D ; //second instance OF FUNCTION InputToIndex\_3D

rtb\_Switch: LREAL;

rtb\_RelationalOperator: BOOL;

Array\_: ARRAY [1..100] OF LREAL := [100(10)];

END\_VAR

(\* Outputs for Atomic SubSystem: '<Root>/Code\_1' \*)

(\* Switch: '<S1>/Switch' incorporates:

\* Gain: '<S1>/Gain2'

\* Product: '<S1>/Product'

\* Sum: '<S1>/Sum'

\* Trigonometry: '<S1>/Trigonometric Function' \*)

Input\_Multip := In1 \* In2 \* In3 \* In4 \* In5 \* In6 \* In7 ;

Input\_Multip\_int := LREAL\_TO\_DINT (Input\_Multip);

IF In5 >= 0.5 THEN

IF Input\_Multip > 1000 THEN

INPUTTOINDEX\_3D\_1(Input1 := In1,Input2 := In2 ,Input3 := In3, Input4 := In4, Input5 := In5 , Input6 := In6 , Input7 := In7);

rtb\_Switch := SIN(((0.017453292519943295 \* In3) + In2) + In4) \* In1;

ELSE

INPUTTOINDEX\_3D\_2(Input1 := In1,Input2 := In2 ,Input3 := In3, Input4 := In4, Input5 := In5 , Input6 := In6 , Input7 := In7);

Array\_ [Input\_Multip\_int] := Input\_Multip;

rtb\_Switch := SIN(((0.017453292519943295 \* In3) + In2) + In4) \* In1;

END\_IF;

ELSE

rtb\_Switch := In6;

END\_IF;

(\* End of Switch: '<S1>/Switch' \*)

(\* RelationalOperator: '<S1>/Relational Operator' \*)

rtb\_RelationalOperator := rtb\_Switch >= In7;

(\* End of Outputs for SubSystem: '<Root>/Code\_1' \*)

(\* Outport: '<Root>/Out1' incorporates:

\* Logic: '<S1>/BL4' \*)

Out1[0] := rtb\_RelationalOperator;

(\* Outputs for Atomic SubSystem: '<Root>/Code\_1' \*)

Out1[1] := NOT rtb\_RelationalOperator;

(\* Outport: '<Root>/Out2' incorporates:

\* Gain: '<S1>/Gain3'

\* RelationalOperator: '<S1>/Relational Operator1' \*)

Out2 := ( -rtb\_Switch) >= In7;

(\* End of Outputs for SubSystem: '<Root>/Code\_1' \*)

FUNCTION\_BLOCK InputToIndex\_3D

VAR\_INPUT

Input1 : LREAL ;

Input2 : LREAL ;

Input3 : LREAL ;

Input4 : LREAL ;

Input5 : LREAL ;

Input6 : LREAL ;

Input7 : LREAL ;

END\_VAR

VAR\_OUTPUT

Array\_3D : ARRAY [1..100,1..100,1..100] OF LREAL := [100(1),100(1),100(1)];

Array\_1D : ARRAY [1..100] OF LREAL := [100(1)];

END\_VAR

VAR

Index1 : LREAL ;

Index2 : LREAL ;

Index3 : LREAL ;

Index1\_int : DINT := 0;

Index2\_int : DINT := 0;

Index3\_int : DINT := 0;

i : DINT;

j : DINT ;

k : DINT ;

END\_VAR

Index1 := (Input1)\*(Input2);

Index2 := (Input3)\*(Input4);

Index3 := (Input5)\*(Input6);

Index1\_int := LREAL\_TO\_DINT (Index1);

Index2\_int := LREAL\_TO\_DINT (Index2);

Index3\_int := LREAL\_TO\_DINT (Index3);

FOR i:= 0 TO Index1\_int DO

FOR j:= 0 TO Index2\_int DO

FOR k:= 0 TO Index3\_int DO

Array\_3D[i,j,k] := Input7;

END\_FOR

END\_FOR

END\_FOR