(\* Following taken directly from the IEC 61131.3 draft standard \*)

(\*

\* An IEC 61131-3 IL and ST compiler.

\*

\* Based on the

\* FINAL DRAFT - IEC 61131-3, 2nd Ed. (2001-12-10)

\*

\*)

(\*

\* This is part of the library conatining the functions

\* and function blocks defined in the standard.

\*

\* Counter function blocks

\* -----------------------

\*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

(\* \*)

(\* C T U \*)

(\* \*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

FUNCTION\_BLOCK CTU

VAR\_INPUT

CU : BOOL;

R : BOOL;

PV : INT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : INT;

END\_VAR

VAR

CU\_T: R\_TRIG;

END\_VAR

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

END\_IF ;

Q := (CV >= PV) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTU\_DINT

VAR\_INPUT

CU : BOOL;

R : BOOL;

PV : DINT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : DINT;

END\_VAR

VAR

CU\_T: R\_TRIG;

END\_VAR

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

END\_IF ;

Q := (CV >= PV) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTU\_LINT

VAR\_INPUT

CU : BOOL;

R : BOOL;

PV : LINT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : LINT;

END\_VAR

VAR

CU\_T: R\_TRIG;

END\_VAR

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

END\_IF ;

Q := (CV >= PV) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTU\_UDINT

VAR\_INPUT

CU : BOOL;

R : BOOL;

PV : UDINT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : UDINT;

END\_VAR

VAR

CU\_T: R\_TRIG;

END\_VAR

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

END\_IF ;

Q := (CV >= PV) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTU\_ULINT

VAR\_INPUT

CU : BOOL;

R : BOOL;

PV : ULINT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : ULINT;

END\_VAR

VAR

CU\_T: R\_TRIG;

END\_VAR

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

END\_IF ;

Q := (CV >= PV) ;

END\_FUNCTION\_BLOCK

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

(\* \*)

(\* C T D \*)

(\* \*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

FUNCTION\_BLOCK CTD

VAR\_INPUT

CD : BOOL;

LD : BOOL;

PV : INT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : INT;

END\_VAR

VAR

CD\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

IF LD THEN CV := PV ;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF ;

Q := (CV <= 0) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTD\_DINT

VAR\_INPUT

CD : BOOL;

LD : BOOL;

PV : DINT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : DINT;

END\_VAR

VAR

CD\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

IF LD THEN CV := PV ;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF ;

Q := (CV <= 0) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTD\_LINT

VAR\_INPUT

CD : BOOL;

LD : BOOL;

PV : LINT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : LINT;

END\_VAR

VAR

CD\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

IF LD THEN CV := PV ;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF ;

Q := (CV <= 0) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTD\_UDINT

VAR\_INPUT

CD : BOOL;

LD : BOOL;

PV : UDINT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : UDINT;

END\_VAR

VAR

CD\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

IF LD THEN CV := PV ;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF ;

Q := (CV <= 0) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTD\_ULINT

VAR\_INPUT

CD : BOOL;

LD : BOOL;

PV : ULINT;

END\_VAR

VAR\_OUTPUT

Q : BOOL;

CV : ULINT;

END\_VAR

VAR

CD\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

IF LD THEN CV := PV ;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF ;

Q := (CV <= 0) ;

END\_FUNCTION\_BLOCK

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

(\* \*)

(\* C T U D \*)

(\* \*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

FUNCTION\_BLOCK CTUD

VAR\_INPUT

CU : BOOL;

CD : BOOL;

R : BOOL;

LD : BOOL;

PV : INT;

END\_VAR

VAR\_OUTPUT

QU : BOOL;

QD : BOOL;

CV : INT;

END\_VAR

VAR

CD\_T: R\_TRIG;

CU\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF LD THEN CV := PV ;

ELSE

IF NOT (CU\_T.Q AND CD\_T.Q) THEN

IF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF;

END\_IF;

END\_IF ;

QU := (CV >= PV) ;

QD := (CV <= 0) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTUD\_DINT

VAR\_INPUT

CU : BOOL;

CD : BOOL;

R : BOOL;

LD : BOOL;

PV : DINT;

END\_VAR

VAR\_OUTPUT

QU : BOOL;

QD : BOOL;

CV : DINT;

END\_VAR

VAR

CD\_T: R\_TRIG;

CU\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF LD THEN CV := PV ;

ELSE

IF NOT (CU\_T.Q AND CD\_T.Q) THEN

IF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF;

END\_IF;

END\_IF ;

QU := (CV >= PV) ;

QD := (CV <= 0) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTUD\_LINT

VAR\_INPUT

CU : BOOL;

CD : BOOL;

R : BOOL;

LD : BOOL;

PV : LINT;

END\_VAR

VAR\_OUTPUT

QU : BOOL;

QD : BOOL;

CV : LINT;

END\_VAR

VAR

CD\_T: R\_TRIG;

CU\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF LD THEN CV := PV ;

ELSE

IF NOT (CU\_T.Q AND CD\_T.Q) THEN

IF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF;

END\_IF;

END\_IF ;

QU := (CV >= PV) ;

QD := (CV <= 0) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTUD\_UDINT

VAR\_INPUT

CU : BOOL;

CD : BOOL;

R : BOOL;

LD : BOOL;

PV : UDINT;

END\_VAR

VAR\_OUTPUT

QU : BOOL;

QD : BOOL;

CV : UDINT;

END\_VAR

VAR

CD\_T: R\_TRIG;

CU\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF LD THEN CV := PV ;

ELSE

IF NOT (CU\_T.Q AND CD\_T.Q) THEN

IF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF;

END\_IF;

END\_IF ;

QU := (CV >= PV) ;

QD := (CV <= 0) ;

END\_FUNCTION\_BLOCK

FUNCTION\_BLOCK CTUD\_ULINT

VAR\_INPUT

CU : BOOL;

CD : BOOL;

R : BOOL;

LD : BOOL;

PV : ULINT;

END\_VAR

VAR\_OUTPUT

QU : BOOL;

QD : BOOL;

CV : ULINT;

END\_VAR

VAR

CD\_T: R\_TRIG;

CU\_T: R\_TRIG;

END\_VAR

CD\_T(CD);

CU\_T(CU);

IF R THEN CV := 0 ;

ELSIF LD THEN CV := PV ;

ELSE

IF NOT (CU\_T.Q AND CD\_T.Q) THEN

IF CU\_T.Q AND (CV < PV)

THEN CV := CV+1;

ELSIF CD\_T.Q AND (CV > 0)

THEN CV := CV-1;

END\_IF;

END\_IF;

END\_IF ;

QU := (CV >= PV) ;

QD := (CV <= 0) ;

END\_FUNCTION\_BLOCK