PROGRAM GRAVEL (\* Gravel measurement and loading system \*)

VAR\_INPUT

OFF\_PB : BOOL ;

ON\_PB : BOOL ;

FILL\_PB : BOOL ;

SIREN\_ACK : BOOL ;

LOAD\_PB : BOOL ; (\* Load truck from bin \*)

JOG\_PB : BOOL ;

LAMP\_TEST : BOOL ;

TRUCK\_ON\_RAMP : BOOL ; (\* Optical sensor \*)

SILO\_EMPTY\_LS : BOOL ;

BIN\_EMPTY\_LS : BOOL ;

SETPOINT : BYTE ; (\* 2-digit BCD \*)

END\_VAR

VAR\_OUTPUT

CONTROL\_LAMP : BOOL ;

TRUCK\_LAMP : BOOL ;

SILO\_EMPTY\_LAMP : BOOL ;

CONVEYOR\_LAMP : BOOL ;

CONVEYOR\_MOTOR : BOOL ;

SILO\_VALVE : BOOL ;

BIN\_VALVE : BOOL ;

SIREN : BOOL ;

BIN\_LEVEL : BYTE ;

END\_VAR

VAR

BLINK\_TIME : TIME; (\* BLINK ON/OFF time \*)

PULSE\_TIME : TIME; (\* LEVEL\_CTR increment interval \*)

RUNOUT\_TIME: TIME; (\* Conveyor running time after loading \*)

RUN\_IN\_TIME: TIME; (\* Conveyor running time before loading \*)

SILENT\_TIME: TIME; (\* Siren silent time after SIREN\_ACK \*)

OK\_TO\_RUN : BOOL; (\* 1 = Conveyor is allowed to run \*)

(\* Function Blocks \*)

BLINK: TON; (\* Blinker OFF period timer / ON output \*)

BLANK: TON; (\* Blinker ON period timer / blanking pulse \*)

PULSE: TON; (\* LEVEL\_CTR pulse interval timer \*)

SIREN\_FF: RS;

SILENCE\_TMR: TP; (\* Siren silent period timer \*)

END\_VAR

VAR RETAIN LEVEL\_CTR : CTU ; END\_VAR

(\* Program body \*)

(\* Major operating states \*)

INITIAL\_STEP START : END\_STEP

TRANSITION FROM START TO FILL\_BIN

:= FILL\_PB & CONTROL.X ; END\_TRANSITION

STEP FILL\_BIN: SILO\_VALVE(N); END\_STEP

TRANSITION FROM FILL\_BIN TO START

:= NOT FILL\_PB OR NOT CONTROL.X ; END\_TRANSITION

TRANSITION FROM FILL\_BIN TO LOAD\_WAIT := LEVEL\_CTR.Q ;

END\_TRANSITION

STEP LOAD\_WAIT : END\_STEP

TRANSITION FROM LOAD\_WAIT TO RUN\_IN

:= LOAD\_PB & OK\_TO\_RUN ; END\_TRANSITION

STEP RUN\_IN : END\_STEP

TRANSITION FROM RUN\_IN TO LOAD\_WAIT := NOT OK\_TO\_RUN ;

END\_TRANSITION

TRANSITION FROM RUN\_IN TO DUMP\_BIN

:= RUN\_IN.T > RUN\_IN\_TIME;

END\_TRANSITION

STEP DUMP\_BIN: BIN\_VALVE(N); END\_STEP

TRANSITION FROM DUMP\_BIN TO LOAD\_WAIT := NOT OK\_TO\_RUN ;

END\_TRANSITION

TRANSITION FROM DUMP\_BIN TO RUNOUT := BIN\_EMPTY\_LS ;

END\_TRANSITION

STEP RUNOUT : END\_STEP

TRANSITION FROM RUNOUT TO LOAD\_WAIT := NOT OK\_TO\_RUN ;

END\_TRANSITION

TRANSITION FROM RUNOUT TO START

:= RUNOUT.T >= RUNOUT\_TIME ; END\_TRANSITION

(\* Control state sequencing \*)

INITIAL\_STEP CONTROL\_OFF: END\_STEP

TRANSITION FROM CONTROL\_OFF TO CONTROL

:= ON\_PB & NOT OFF\_PB ; END\_TRANSITION

STEP CONTROL: CONTROL\_ACTION(N); END\_STEP

ACTION CONTROL\_ACTION:

BLINK(EN:=CONTROL.X & NOT BLANK.Q, PT := BLINK\_TIME) ;

BLANK(EN:=BLINK.Q, PT := BLINK\_TIME) ;

OK\_TO\_RUN := CONTROL.X & TRUCK\_ON\_RAMP ;

CONVEYOR\_MOTOR :=

OK\_TO\_RUN & OR(JOG\_PB, RUN\_IN.X, DUMP\_BIN.X, RUNOUT.X);

END\_ACTION

TRANSITION FROM CONTROL TO CONTROL\_OFF := OFF\_PB ;

END\_TRANSITION

(\* Monitor Logic \*)

INITIAL\_STEP MONITOR: MONITOR\_ACTION(N); END\_STEP

ACTION MONITOR\_ACTION:

CONVEYOR\_LAMP := CONVEYOR\_MOTOR & BLINK.Q ;

CONTROL\_LAMP := CONTROL.X OR LAMP\_TEST ;

TRUCK\_LAMP := TRUCK\_ON\_RAMP OR LAMP\_TEST ;

SILO\_EMPTY\_LAMP := BLINK.Q & SILO\_EMPTY\_LS OR LAMP\_TEST ;

SILENCE\_TMR(IN:=SIREN\_ACK, PT:=SILENT\_TIME) ;

SIREN\_FF(S:=SILO\_EMPTY\_LS, R1:=SILENCE\_TMR.Q) ;

SIREN := SIREN\_FF.Q1 ;

PULSE(IN:=FILL\_BIN.X & NOT PULSE.Q, PT:=PULSE\_TIME) ;

LEVEL\_CTR(R := BIN\_EMPTY\_LS, CU := PULSE.Q,

PV := BCD\_TO\_INT(SETPOINT)) ;

BIN\_LEVEL := INT\_TO\_BCD(LEVEL\_CTR.CV) ;

END\_ACTION

END\_PROGRAM

CONFIGURATION GRAVEL\_CONTROL

RESOURCE PROC1 ON PROC\_TYPE\_Y

PROGRAM G : GRAVEL

(\* Inputs \*)

(OFF\_PB := %I0.0 ,

ON\_PB := %I0.1 ,

FILL\_PB := %I0.2 ,

SIREN\_ACK := %I0.3 ,

LOAD\_PB := %I0.4 ,

JOG\_PB := %I0.5 ,

LAMP\_TEST := %I0.7 ,

TRUCK\_ON\_RAMP := %I1.4 ,

SILO\_EMPTY\_LS := %I1.5 ,

BIN\_EMPTY\_LS := %I1.6 ,

SETPOINT := %IB2 ,

(\* Outputs \*)

CONTROL\_LAMP => %Q4.0,

TRUCK\_LAMP => %Q4.2,

SILO\_EMPTY\_LAMP => %Q4.3,

CONVEYOR\_LAMP => %Q5.3,

CONVEYOR\_MOTOR => %Q5.4,

SILO\_VALVE => %Q5.5,

BIN\_VALVE => %Q5.6,

SIREN => %Q5.7,

BIN\_LEVEL => %B6) ;

END\_RESOURCE

END\_CONFIGURATION