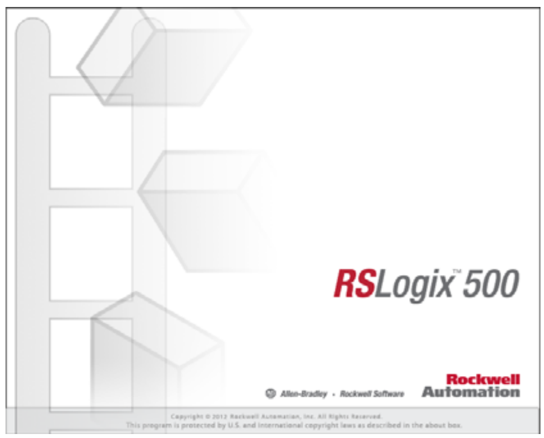
RSLogix 500 Project Report



Processor Type: Bul.1766   MicroLogix 1400 Series A

Processor Name: UNTITLED

Total Memory Used: 338 Instruction Words Used - 163 Data Table Words Used

Total Memory Left: 12096 Instruction Words Left

Program Files: 10

Data Files: 10

Program ID: e6f8

I/O Configuration

0	Bul.1766	MicroLogix 1400 Series A
1	1762-IF2OF2	Analog 2 Chan. Input, 2 Chan. Output
2	1762-IT4	4-Channel Thermocouple Input Module
3		
4		
5		
6		
7		

CHANNEL 0 (SYSTEM) - Driver: DF1 Full Duplex
CHANNEL 0 (SYSTEM) - Driver: DF1 Full Duplex Edit Resource/Owner Timeout: 60
CHANNEL 0 (SYSTEM) - Driver: DF1 Full Duplex Passthru Link ID: 1
CHANNEL 0 (SYSTEM) - Driver: DF1 Full Duplex Write Protected: No
CHANNEL 0 (SYSTEM) - Driver: DF1 Full Duplex Comms Servicing Selection: Yes
CHANNEL 0 (SYSTEM) - Driver: DF1 Full Duplex Message Servicing Selection: Yes
CHANNEL 0 (SYSTEM) - Driver: DF1 Full Duplex 1st AWA Append Character: \d
CHANNEL 0 (SYSTEM) - Driver: DF1 Full Duplex 2nd AWA Append Character: \a

Source ID: 1 (decimal)
Baud: 19200
Parity: NONE
Control Line : No Handshaking
Error Detection: CRC
Embedded Responses: Auto Detect
Duplicate Packet Detect: Yes
ACK Timeout(x20 ms): 50
NAK Retries: 3
ENQ Retries: 3

CHANNEL 1 (SYSTEM) - Driver: Ethernet
CHANNEL 1 (SYSTEM) - Driver: Ethernet Edit Resource/Owner Timeout: 60
CHANNEL 1 (SYSTEM) - Driver: Ethernet Passthru Link ID: 1
CHANNEL 1 (SYSTEM) - Driver: Ethernet Write Protected: No
CHANNEL 1 (SYSTEM) - Driver: Ethernet Comms Servicing Selection: Yes
CHANNEL 1 (SYSTEM) - Driver: Ethernet Message Servicing Selection: Yes

Hardware Address: 00:00:00:00:00:00
IP Address: 0.0.0.0
Subnet Mask: 0.0.0.0
Gateway Address: 0.0.0.0
Msg Connection Timeout (x 1mS): 15000
Msg Reply Timeout (x mS): 3000
Inactivity Timeout (x Min): 30
Bootp Enable: Yes
Dhcp Enable No
SMTP Enable: No
SNMP Enable: Yes
HTTP Enable: Yes
Auto Negotiate Enable: Yes
Port Speed Enable: 10/100 Mbps Full Duplex/Half Duplex
Contact:
Location:

CHANNEL 2 (SYSTEM) - Driver: DF1 Full Duplex
CHANNEL 2 (SYSTEM) - Driver: DF1 Full Duplex Edit Resource/Owner Timeout: 60
CHANNEL 2 (SYSTEM) - Driver: DF1 Full Duplex Passthru Link ID: 1
CHANNEL 2 (SYSTEM) - Driver: DF1 Full Duplex Write Protected: No
CHANNEL 2 (SYSTEM) - Driver: DF1 Full Duplex Comms Servicing Selection: Yes
CHANNEL 2 (SYSTEM) - Driver: DF1 Full Duplex Message Servicing Selection: Yes
CHANNEL 2 (SYSTEM) - Driver: DF1 Full Duplex 1st AWA Append Character: \d
CHANNEL 2 (SYSTEM) - Driver: DF1 Full Duplex 2nd AWA Append Character: \a

Source ID: 1 (decimal)
Baud: 19200
Parity: NONE
Control Line : No Handshaking
Error Detection: CRC
Embedded Responses: Auto Detect
Duplicate Packet Detect: Yes
ACK Timeout(x20 ms): 50
NAK Retries: 3
ENQ Retries: 3

Program File List

Name	Number	Type	Rungs	Debug	Bytes
[SYSTEM]	0	SYS	0	No	0
	1	SYS	0	No	0
MAIN	2	LADDER	10	No	116
D INPUT	3	LADDER	9	No	131
D OUTPUT	4	LADDER	6	No	83
A INPUT	5	LADDER	3	No	254
A OUTPUT	6	LADDER	2	No	121
CONTROLS	7	LADDER	13	No	641
ALARMS	8	LADDER	21	No	920
DISPLAY	9	LADDER	2	No	42

Data File List

Name	Number	Type	Scope	Debug	Words	Elements	Last
OUTPUT	0	O	Global	No	24	8	O:7
INPUT	1	I	Global	No	60	20	I:19
STATUS	2	S	Global	No	0	66	S:65
BINARY	3	B	Global	No	8	8	B3:7
TIMER	4	T	Global	No	24	8	T4:7
COUNTER	5	C	Global	No	3	1	C5:0
CONTROL	6	R	Global	No	3	1	R6:0
INTEGER	7	N	Global	No	12	12	N7:11
FLOAT	8	F	Global	No	6	3	F8:2
TEMP	9	PD	Global	No	23	1	PD9:0

S:1/15 ONLY ENERGIZES ON FIRST SCAN. USED HERE FOR SYSTEM INITIALIZATION, SUCH AS ENSURING THE SYSTEM IS IN "OFF" MODE ON START.

0000

First Pass
S:1
15

SYSTEM MODE
0 = OFF
1 = ON
2 = FAULTED
SYSTEM_MODE
MOV
Move
Source 0
0<
Dest N7:4
1<

BLOWER MODE
0 = OFF
1 = HAND
2 = AUTO
BLOWER_MODE
MOV
Move
Source 2
2<
Dest N7:11
2<

JSR (JUMP SUB-ROUTINE) BLOCKS EXECUTE THE REMAINING PROGRAM FILES IN OUT PLC PROGRAM.

0001

DIGITAL INPUTS
JSR
Jump To Subroutine
SBR File Number U:3

0002

DIGITAL OUTPUTS
JSR
Jump To Subroutine
SBR File Number U:4

0003

ANALOG INPUTS
JSR
Jump To Subroutine
SBR File Number U:5

0004

ANALOG OUTPUTS
JSR
Jump To Subroutine
SBR File Number U:6

0005

CONTROLS
JSR
Jump To Subroutine
SBR File Number U:7

0006

ALARMS
JSR
Jump To Subroutine
SBR File Number U:8

0007 DISPLAY
JSR
Jump To Subroutine
SBR File Number U:9

UNLATCHING THE S:5/0 BIT PERMANENTLY IS A COMMON PRACTICE. THIS PREVENTS MATH ERRORS (SUCH AS DIVIDING BY ZERO) FROM KILLING YOUR PROGRAM DURING EXECUTION.

0008 Overflow Trap
S:5
U
0

0009 END

EACH DIGITAL INPUT IS TIED TO A BIT TO BE USED THROUGHOUT THE PROGRAM. WHILE THE ACTUAL INPUT CAN BE USED, THIS IS PREFERABLE AS IT GREATLY SIMPLIFIES FUTURE CHANGES TO INPUT LOCATIONS.

0000
BUTTON "ON"
I:0
0
Bul.1766
BUTTON "ON" IN BIT
B3:0
0

0001
BUTTON "OFF"
I:0
1
Bul.1766
BUTTON "OFF" IN BIT
B3:0
1

0002
PRESSURE SWITCH IN
I:0
2
Bul.1766
PRESSURE SWITCH IN BIT
PRESSURE_SWITCH
B3:0
2

0003
FLOW SWITCH IN
I:0
3
Bul.1766
FLOW SWITCH IN BIT
FLOW_SWITCH
B3:0
3

0004
LOW-LOW LEVEL SWITCH IN
I:0
4
Bul.1766
LOW-LOW LEVEL SWITCH IN BIT
LVL_LL_SW
B3:0
4

0005
LOW LEVEL SWITCH IN
I:0
5
Bul.1766
LOW LEVEL SWITCH IN BIT
LVL_L_SW
B3:0
5

0006
HIGH LEVEL SWITCH IN
I:0
6
Bul.1766
HIGH LEVEL SWITCH IN BIT
LVL_H_SW
B3:0
6

0007
HIGH-HIGH LEVEL SWITCH IN
I:0
7
Bul.1766
HIGH-HIGH LEVEL SWITCH IN BIT
LVL_HH_SW
B3:0
7

0008
END

EACH DIGITAL OUTPUT IS TRIGGERED BY A SINGLE BIT CONTROLLED ELSEWHERE IN THE PROGRAM. WHILE THE ACTUAL OUTPUT CAN BE USED IN THE CONTROL LOGIC, THIS METHOD IS PREFERABLE AS IT GREATLY SIMPLIFIES FUTURE CHANGES TO OUTPUT LOCATIONS.

0000	BLOWER OUT BIT BLOWER B3:1 0	BLOWER OUT O:0 0 Bul.1766
0001	TANK FILL PUMP OUT BIT PUMP B3:1 1	TANK FILL PUMP OUT O:0 1 Bul.1766
0002	SUPPLY VALVE OUT BIT VALVE B3:1 2	SUPPLY VALVE OUT O:0 2 Bul.1766
0003	HEATER RELAY OUT BIT HEATER B3:1 3	HEATER RELAY OUT O:0 3 Bul.1766
0004	ALARM HORN OUT BIT HORN B3:1 4	ALARM HORN OUT O:0 4 Bul.1766
0005		END

THE SCP (SCALE WITH PARAMETERS) BLOCK DOES THE CONVERSION OF A RAW INPUT SIGNAL INTO A VALUE YOU WANT TO USE IN YOUR PROGRAM. THERE ARE MANY USES FOR THIS BLOCK, AND MANY WAYS TO SCALE SIGNALS. AS THIS PROGRAM IS WRITTEN FOR A PLC CAPABLE OF PROCESSING ANALOG SIGNALS WITH 14 BITS OF RESOLUTION, WE WANT TO MAKE SURE OUR INPUT AND OUTPUT RANGES BOTH HAVE AROUND 16,383 (2^14) DIFFERENT LEVELS. FOR EXAMPLE, 0-100 WOULD BE TERRIBLE IF WE WERE STORING THE RESULT AS AN INTERGER BECAUSE THERE ARE ONLY 101 LEVELS AND WE WOULD BE LOSING MOST OF OUR RESOLUTION (PRECISION). HOWEVER, IF WE STORED THAT 0-100 AS A FLOAT WITH TWO DECIMAL PLACES, IT'S BETTER BECAUSE WE THEN HAVE 10,001 LEVELS. HOWEVER, THIS STILL ISN'T IDEAL.

0000

LEVEL SENSOR IN
LIM
Limit Test
Low Lim 0
0<
Test I:1.0
10800<
High Lim 16383
16383<

TANK LEVEL
(SCALED 0-100%)
TANK_LEVEL
SCP
Scale w/Parameters
Input I:1.0
10800<
Input Min. 0.0
0.0<
Input Max. 16383.0
16383.0<
Scaled Min. 0.0
0.0<
Scaled Max. 100.0
100.0<
Output F8:0
65.92199<

LEVEL SENSOR IN
LES
Less Than (A<B)
Source A I:1.0
10800<
Source B 0
0<

TANK LEVEL
(SCALED 0-100%)
TANK_LEVEL
MOV
Move
Source 0.0
0.0<
Dest F8:0
65.92199<

LEVEL SENSOR IN
GRT
Greater Than (A>B)
Source A I:1.0
10800<
Source B 16383
16383<

TANK LEVEL
(SCALED 0-100%)
TANK_LEVEL
MOV
Move
Source 100.0
100.0<
Dest F8:0
65.92199<

SOMETIMES, STRANGE SIGNALS CAN COME INTO OUR SYSTEM AS A RESULT OF NOISE, FAULTY SENSORS / WIRING, OR FAULTY MODULES. TO KEEP MY VALUES IN A USEFUL RANGE IN SUCH CASES, I'VE PUT CONDITIONAL LOGIC AROUND THE SCP BLOCK TO HANDLE ANOMALOUS INPUTS. THE LIM (LIMIT) BLOCK CHECKS TO SEE IF MY RAW INPUT IS WITHIN THE DESIRED RANGE BEFORE SCALING. A LES (LESS THAN <) BLOCK AND A GRT (GREATER THAN >) BLOCK ON SEPARATE BRANCHES SET MINIMUM OR MAXIMUM VALUES RESPECTIVELY IF AND ONLY IF THE RAW INPUT SIGNAL IS OUT OF RANGE. THE LAST BRANCH ON THE TEMPERATURE RUNG (0001) IS JUST STORING THE RAW INPUT (WHICH IS ALREADY SCALED FOR PID 0-16383) TO AN INTERGER REGISTER IN MEMORY FOR USE IN A PID CONTROL LOOP. THIS PREVENTS ME FROM REFERRING TO THE ACTUAL INPUT WITHIN THE REST OF THE PROGRAM. THUS, IF IN THE FUTURE WE NEED TO REMAP OUR IO, WE ONLY HAVE TO MAKE CHANGES IN THE IO PROGRAM FILES INSTEAD OF HUNTING EACH ADDRESS THROUGHOUT THE ENTIRE PROGRAM.

0001

THERMOCOUPLE IN
LIM
Limit Test
Low Lim 0
0<
Test I:2.0
3000<
High Lim 16383
16383<

TEMPERATURE
(SCALED -40 TO 750F)
TEMPERATURE
SCP
Scale w/Parameters
Input I:2.0
3000<
Input Min. 0.0
0.0<
Input Max. 16383.0
16383.0<
Scaled Min. -40.0
-40.0<
Scaled Max. 750.0
750.0<
Output F8:1
104.6621<

THERMOCOUPLE IN
LES
Less Than (A<B)
Source A I:2.0
3000<
Source B 0
0<

TEMPERATURE
(SCALED -40 TO 750F)
TEMPERATURE
MOV
Move
Source -40.0
-40.0<
Dest F8:1
104.6621<

THERMOCOUPLE IN
GRT
Greater Than (A>B)
Source A I:2.0
3000<
Source B 16383
16383<

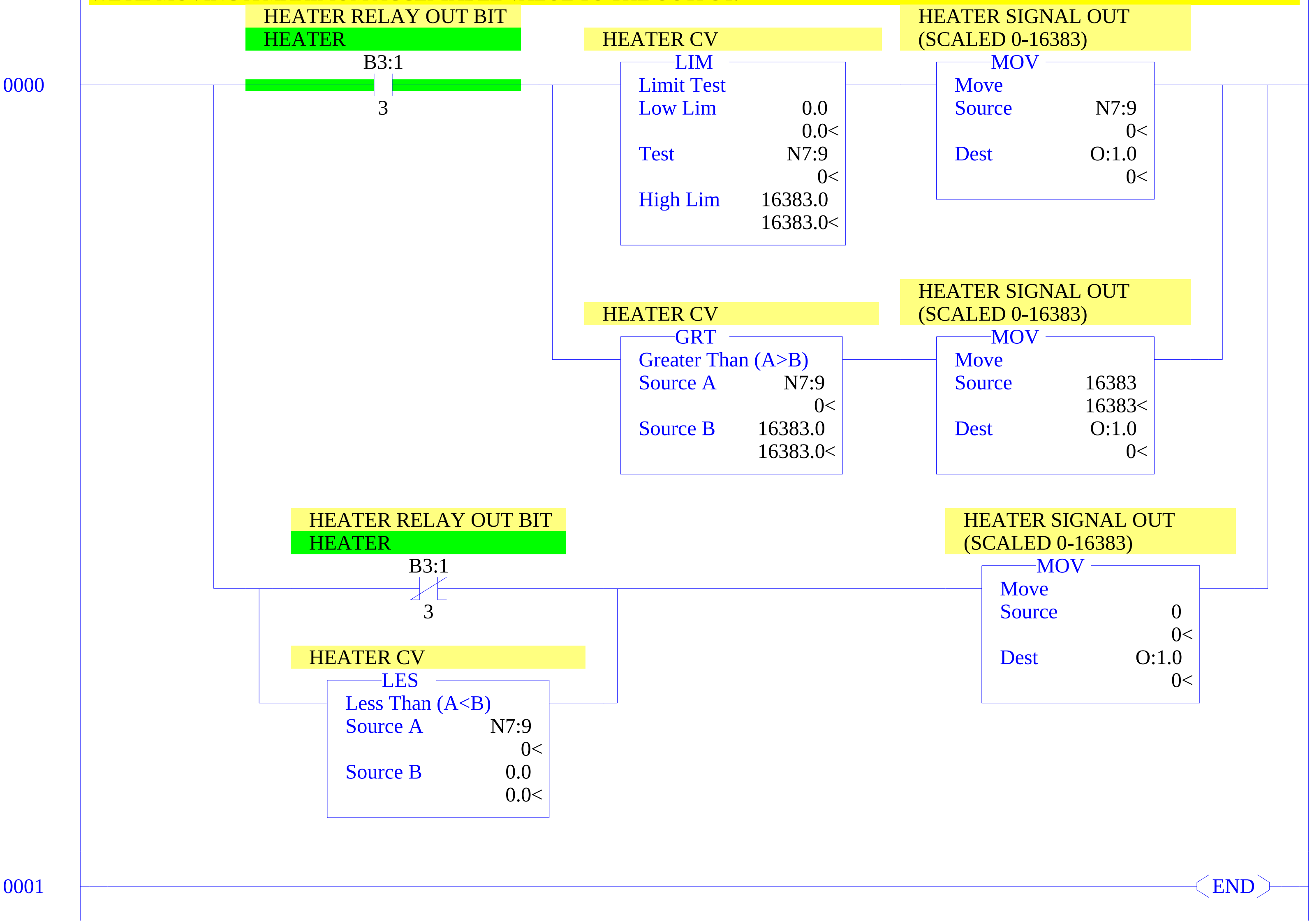
TEMPERATURE
(SCALED -40 TO 750F)
TEMPERATURE
MOV
Move
Source 750.0
750.0<
Dest F8:1
104.6621<

TEMPERATURE PV
MOV
Move
Source I:2.0
3000<
Dest N7:8
3000<

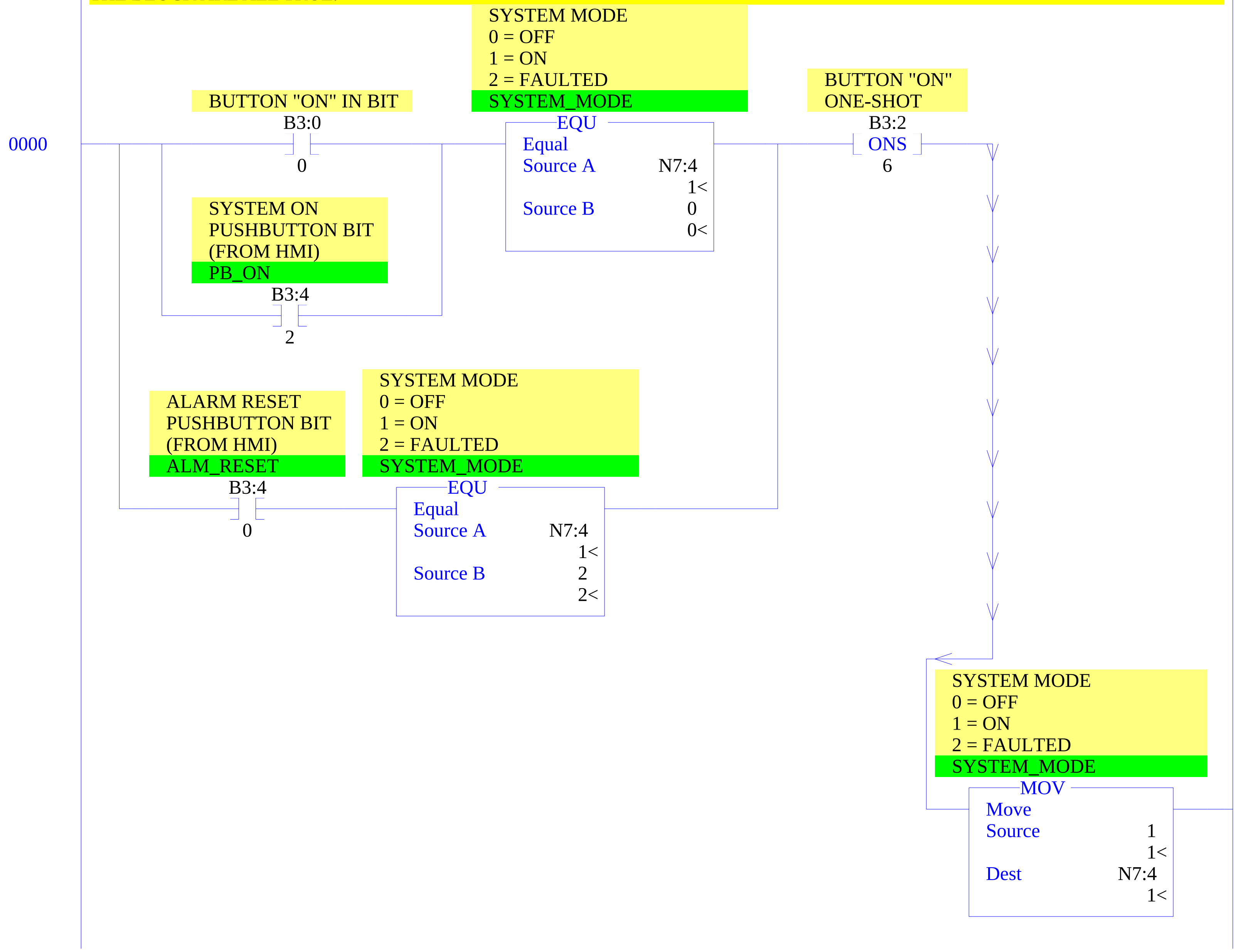
0002

END

THIS RUNG IS CONTROLLING THE ANALOG OUTPUT SIGNAL TO TELL OUR HEATER HOW MUCH HEAT TO PRODUCE. THE OUTPUT MODULE CONVERTS OUR 0-16383 PID VALUE INTO A 4-20mA SIGNAL. HERE WE ARE CHECKING TO SEE IF THE HEATER IS ENERGIZED. IF IT ISN'T, OR IF FOR SOME REASON OUR PID CV (CONTROL VARIABLE) IS BELOW 0, WE'RE SENDING A ZERO OUTPUT TO THE HEATER - NO HEAT. IF HOWEVER THE HEATER IS ENERGIZED, WE CHECK TO SEE IF THE PID CV IS WITHIN RANGE OR OVER RANGE. IF IT'S IN RANGE, WE MOVE OUR VALUE DIRECTLY TO THE OUTPUT (WHICH IS CONFIGURED FOR A PID RANGE OF 0-16383). ON THE OTHER HAND, IF FOR SOME REASON OUR CV IS OVER THE RANGE, WE'RE MOVING A MAXIMUM ACCEPTABLE VALUE TO THE OUTPUT.



HERE WE LOOK FOR THE PHYSICAL "ON" BUTTON OR THE HMI SYSTEM ON PUSHBUTTON TO BE PRESSED BY THE OPERATOR. IF AND ONLY IF THE SYSTEM IS IN THE "OFF" MODE, WE CHANGE THE MODE TO ON BY MOVING THE VALUE "1" INTO THE ADDRESS N7:4 (AN ARBITRARILY CHOSEN INTERGER LOCATION TO BE USED THROUGHOUT THE CONTROL LOGIC AND HMI DISPLAYS TO INDICATE SYSTEM MODE). THE EQU (EQUALS) BLOCK COMPARES THE VALUE STORED IN N7:4 TO THE VALUE "0" AND IS EVALUATED AS TRUE WHEN THESE TWO VALUES ARE THE SAME. THE ONS (ONE-SHOT) BLOCK ALLOWS THE OUTPUT TO THE RIGHT TO EXECUTE FOR ONLY ONE SCAN FOR EACH TIME THE CONDITIONS TO THE LEFT OF THE BLOCK ARE ALL TRUE.



HERE WE LOOK FOR THE PHYSICAL "OFF" BUTTON OR THE HMI SYSTEM OFF PUSHBUTTON TO BE PRESSED BY THE OPERATOR. WHENEVER PRESSED, WE CHANGE THE MODE TO OFF BY MOVING THE VALUE "0" INTO THE ADDRESS N7:4.

0001

BUTTON "OFF" IN BIT
B3:0
1

SYSTEM OFF
PUSHBUTTON BIT
(FROM HMI)
PB_OFF
B3:4
3

BUTTON "OFF"
ONE-SHOT
B3:2
ONS
7

SYSTEM MODE
0 = OFF
1 = ON
2 = FAULTED
SYSTEM_MODE

MOV	
Move	
Source	0
	0<
Dest	N7:4
	1<

HERE WE LOOK FOR AN ALARM CONDITION TO. WHEN ANY ALARM OCCURS, WE CHANGE THE MODE TO FAULTED BY MOVING THE VALUE "2" INTO THE ADDRESS N7:4.

0002

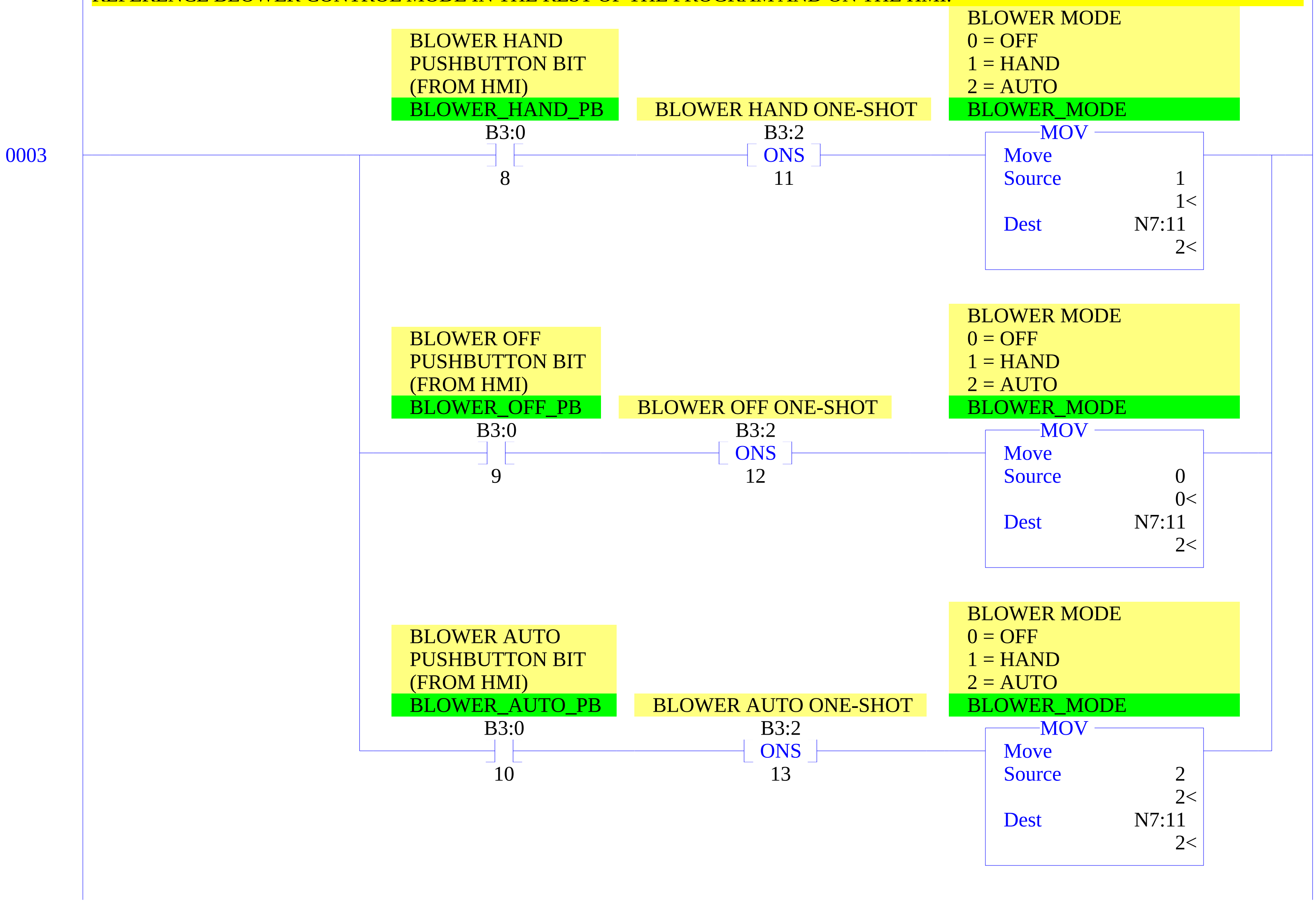
GENERAL ALARM
NOTIFICATION BIT
B3:7
3

"FAULTED"
ONE-SHOT
B3:2
ONS
10

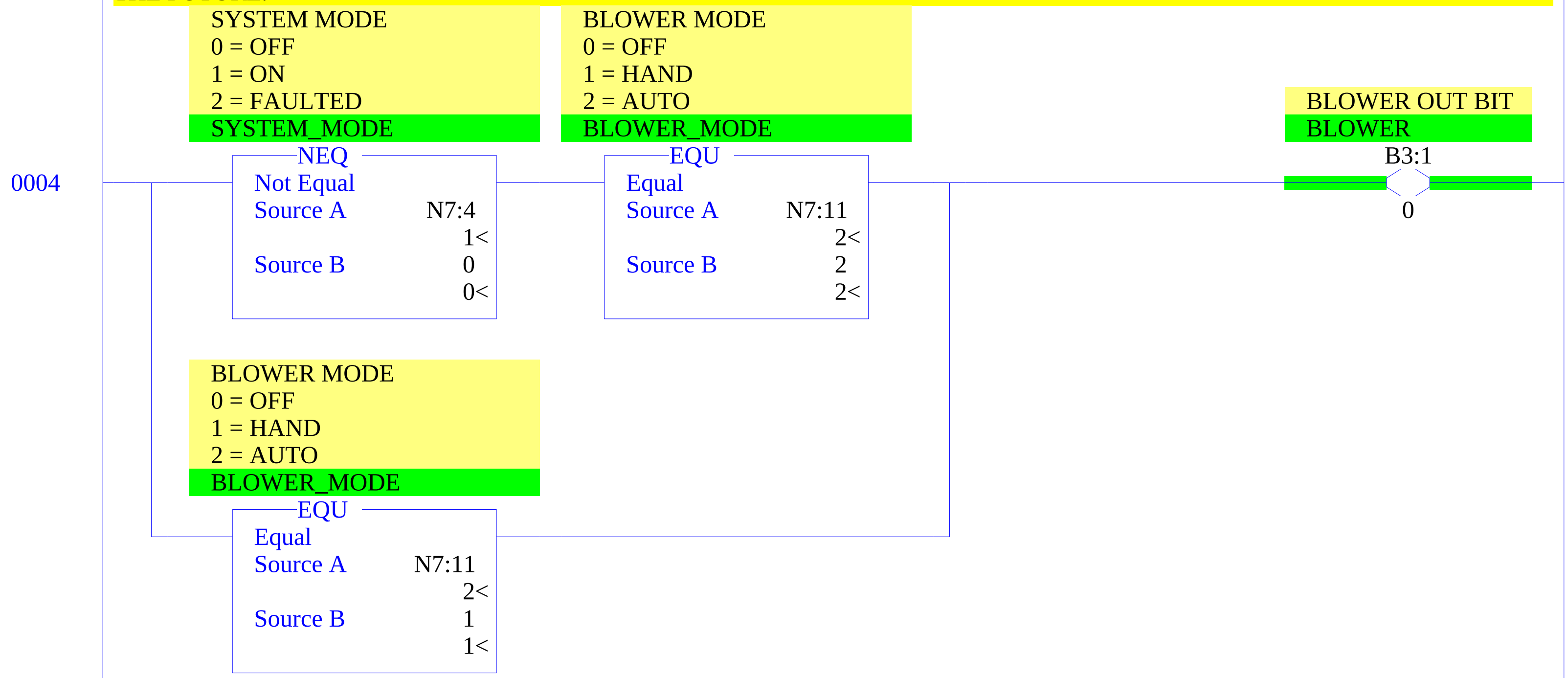
SYSTEM MODE
0 = OFF
1 = ON
2 = FAULTED
SYSTEM_MODE

MOV	
Move	
Source	2
	2<
Dest	N7:4
	1<

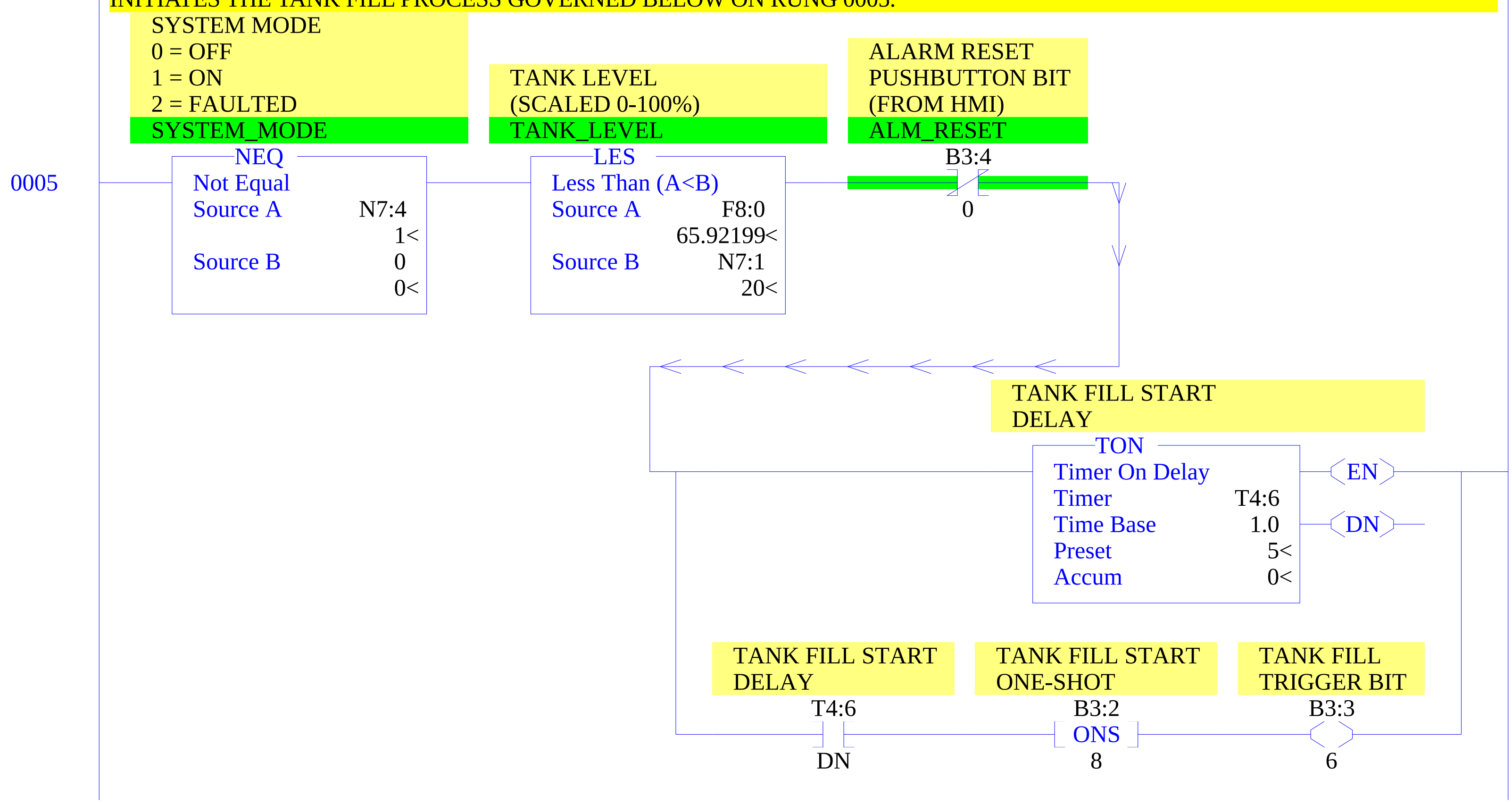
HERE IS A SIMPLE DEMONSTRATION OF AN HOA (HAND - OFF - AUTO) CONTROL SCHEME FOR A DIGITAL DEVICE. IN THIS RUNG WE'RE USING PUSHBUTTONS ON OUR HMI TO SET A STATUS WORD (IN THIS CASE THE INTERGER N7:11) TO REFERENCE BLOWER CONTROL MODE IN THE REST OF THE PROGRAM AND ON THE HMI.



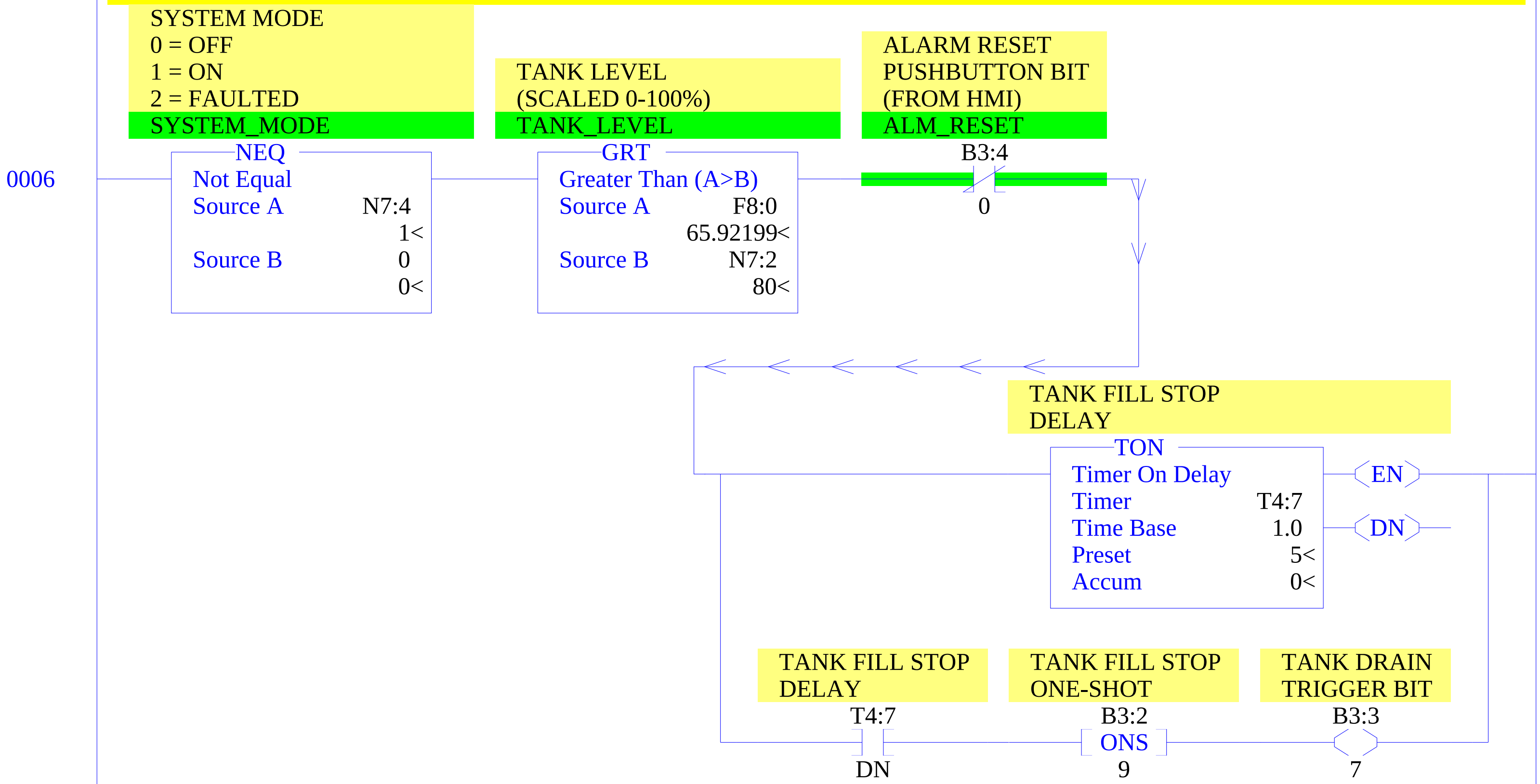
ON THIS RUNG, WE ENERGIZE OUR BLOWER ANYTIME THE SYSTEM IS IN A MODE OTHER THAN OFF --- AND --- THE BLOWER IS IN AUTO CONTROL. WE ALSO ENERGIZE THE BLOWER IF THE BLOWER IS IN HAND MODE REGARDLESS OF THE SYSTEM MODE. THE NEQ (NOT EQUALS) BLOCK IS EVALUATED AS TRUE ANYTIME THE VALUE STORED IN N7:4 AND THE VALUE "0" ARE DIFFERENT. SOMETHING IMPORTANT TO NOTE IS THAT EACH OUTPUT OR DIGITAL MEMORY LOCATION IS ONLY EVER ENERGIZED IN ONE PLACE THROUGHOUT THE ENTIRE PROGRAM! WHILE THIS IS NOT IMPERATIVE, IT IS AN EXCELLENT PRACTICE AS IT MAKES THE PROGRAM SMALLER, SIMPLER, AND MUCH EASIER TO READ, TROUBLESHOOT AND MODIFY IN THE FUTURE.



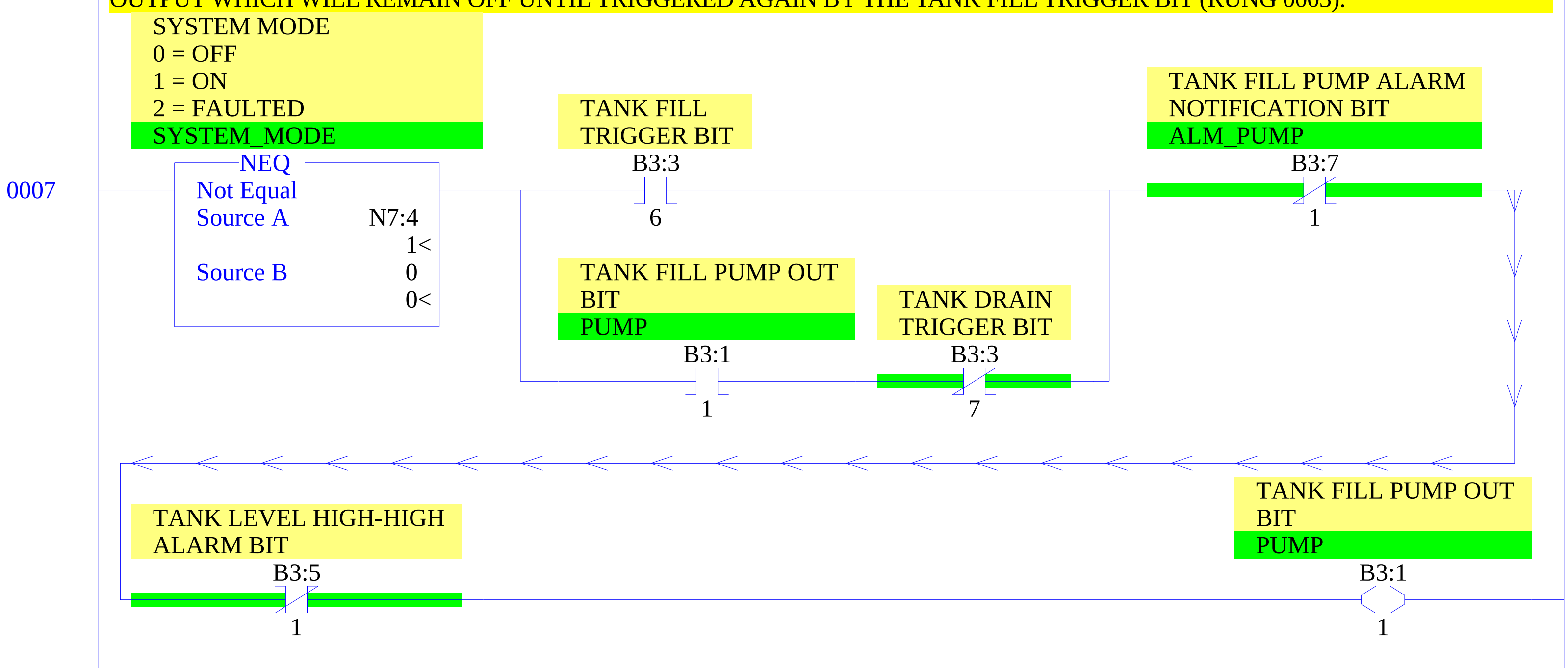
ON THIS RUNG, AS LONG AS THE SYSTEM IS NOT OFF, WE COMPARE OUR TANK LEVEL (F8:0) TO N7:1 WHICH IS DECLARED AS "TANK LOW LEVEL SETPOINT" IN THE PROGRAM. IF THE ACTUAL TANK LEVEL IS LOWER THAN THIS LOW SETPOINT, WE ENERGIZE A DELAY TIMER ON THE TOP BRANCH. IF OUR TANK LEVEL REMAINS BELOW THE LOW SETPOINT FOR FIVE STRAIGHT SECONDS, THE TIMER TIMES OUT AND ENERGIZED IT'S OWN DN (COUNTED DOWN) BIT WHICH IS BEING EVALUATED ON THE LOWER BRANCH OF THIS RUNG. WHEN IT IS TRUE / CLOSED, WE ENERGIZE A TRIGGER BIT WHICH INITIATES THE TANK FILL PROCESS GOVERNED BELOW ON RUNG 0005.



THIS IS THE RECIPROCAL RUNG TO 0003. IT IS USED TO INITIATE THE TANK DRAIN PROCESS THUS DISABLING OUT PUMP UNTIL NEEDED AGAIN. THE MECHANICS OF THIS RUNG ARE FUNDAMENTALLY IDENTICAL TO THOSE OF RUNG 0003. THE DIFFERENCE HERE IS THAT WE ARE COMPARING OUR TANK LEVEL TO A HIGH SETPOINT TO SEE IF THE TANK LEVEL IS GREATER.



SINCE OUR TANK FILL TRIGGER BIT IS PRECEDED BY A ONS (ONE-SHOT) ON RUNG 003, IT WILL ONLY BE ENERGIZED FOR ONE SCAN. ON THIS RUNG, ASSUMING THE SYSTEM IS NOT OFF AND WE DO NOT HAVE AN ACTIVE PUMP ALARM OR HIGH-HIGH LEVEL ALARM, THE TANK FILL PUMP WILL ENERGIZE. THE SUB-BRANCH WHICH BYPASSES THE TANK FILL TRIGGER BIT PROVIDES A HOLD-IN TO KEEP THE PUMP RUNNING AFTER B3:3/6 DE-ENERGIZES (WHICH IT WILL DO IMMEDIATELY). HERE, THE TANK FILL PUMP OUT BIT (B3:1/1) HOLDS ITSELF IN SO LONG AS THE TANK DRAIN TRIGGER BIT (B3:3/7 - ENERGIZED ON RUNG 0004) REMAINS DE-ENERGIZED. ONCE B3:3/7 ENERGIZES, IT WILL TERMINATE THE PUMP'S OUTPUT WHICH WILL REMAIN OFF UNTIL TRIGGERED AGAIN BY THE TANK FILL TRIGGER BIT (RUNG 0003).



HERE WE'RE CONTROLLING THE SUPPLY VALVE LEADING OUT FROM OUR TANK TO WHATEVER WE'RE SUPPLYING. (DRINKING WATER FOR A TRAILER PARK IN INDIANA PERHAPS?) ASSUMING OUR SYSTEM IS NOT TURNED OFF, AND OUR WATER LEVEL ISN'T SO LOW THAT WE HAVE A LOW-LOW LEVEL ALARM ACTIVE, THE VALVE WILL BE ENERGIZED OPEN, AND THE PEOPLE CAN DRINK. WE WOULDN'T WANT TO RUN THE TANK DRY FOR RISK OF SEDIMENT (CRUD THAT SETTLES AT THE BOTTOM OF THE TANK) GETTING INTO THE SUPPLY LINE, WHICH WOULD TASTE HORRIFIC.

SYSTEM MODE
0 = OFF
1 = ON
2 = FAULTED
SYSTEM_MODE

TANK LEVEL LOW-LOW
ALARM BIT

SUPPLY VALVE OUT
BIT
VALVE

0008

NEQ
Not Equal
Source A N7:4
1<
Source B 0
0<

B3:5
0

B3:1
2

ON RUNG 0007 WE'RE CONTROLLING OUR HEATER RELAY THE SAME WAY WE DID OUR SUPPLY VALVE IN THE PREVIOUS RUNG. WHILE THE SYSTEM IS NOT OFF AND WE DON'T HAVE A HIGH-HIGH TEMPERATURE ALARM, THE HEATER RELAY IS ENERGIZED. (THE AMOUNT OF HEAT OUTPUT IS BEING CONTROLLED BY A SEPARATE ANALOG SIGNAL.)

SYSTEM MODE
0 = OFF
1 = ON
2 = FAULTED
SYSTEM_MODE

HIGH-HIGH
TEMPERATURE
ALARM BIT

HEATER RELAY OUT BIT
HEATER

0009

NEQ
Not Equal
Source A N7:4
1<
Source B 0
0<

B3:5
5

B3:1
3

THIS IS UNDOUBTEDLY THE MOST COMPLICATED AND ENIGMATIC RUNG IN THE ENTIRE PROGRAM. THIS RUNG IS CONTROLLING THE ANALOG OUTPUT TO THE HEATER, VIA A PID (PROPORTIONAL INTEGRAL DIFFERENTIAL) CONTROL LOOP. IN SIMPLEST TERMS, THIS IS HOW IT WORKS: THE PID CONTROL IS GIVEN A SETPOINT. IN THIS CASE, THE OPERATOR ENTERS ONE INTO THE HMI. THIS IS THE EQUIVALENT OF TURNING YOUR OVEN DIAL TO 350. THAT IS THE DESIRED TEMPERATURE. THE PID THEN READS THE PV (PROCESS VARIABLE) WHICH TELLS IT WHAT THE TEMPERATURE ACTUALLY IS (INSIDE THE OVEN FOR INSTANCE). DEPENDING ON A VERY COMPLEX SET OF EQUATIONS CONSIDERING SEVERAL PROGRAMMER-CONFIGURABLE VARIABLES TOO COMPLEX TO GET INTO IN THIS TUTORIAL, IT ADJUSTS THE CV (CONTROL VARIABLE) UP OR DOWN TO TRY TO MAINTAIN THE TEMPERATURE AT THE SETPOINT. THE CV IS WHAT WE USE IN THE ANALOG OUTPUT PROGRAM FILE TO ESTABLISH A SIGNAL TO SEND TO THE HEATER. THE MOV (MOVE) BLOCK HERE IS JUST STORING THE OPERATOR'S SETPOINT INTO THE PID SO IT CAN BE EVALUATED THERE. THERE ARE BOOKS, PROGRAMS AND ENTIRE COURSES ON HOW TO SETUP AND 'TUNE' PID'S. FOR SMALLER APPLICATIONS OR APPLICATIONS BEING USED BY LESS-THAN-FLUENT OPERATORS, MANY PROGRAMMERS RIGHTLY SHY AWAY FROM USING PID'S TO CONTROL PROCESS IN ALL BUT CRITICAL PROCESSES.

SYSTEM MODE
0 = OFF
1 = ON
2 = FAULTED
SYSTEM_MODE

TEMPERATURE CONTROL LOOP

0010

NEQ
Not Equal
Source A N7:4
1<
Source B 0
0<

MOV
Move
Source N7:6
90<
Dest PD9:0.SPS
0<

TEMPERATURE CONTROL LOOP

PID
PID
PID File PD9:0
Process Variable N7:8
Control Variable N7:9
Setup Screen <

IS THERE AN ALARM? THEN LET'S SOUND A LOUD AND ANNOYING HORN THROUGHOUT THE PLANT!

SYSTEM MODE
0 = OFF
1 = ON
2 = FAULTED
SYSTEM_MODE

GENERAL ALARM NOTIFICATION BIT

ALARM HORN OUT BIT
HORN

0011

NEQ
Not Equal
Source A N7:4
1<
Source B 0
0<

B3:7
3

B3:1
4

0012

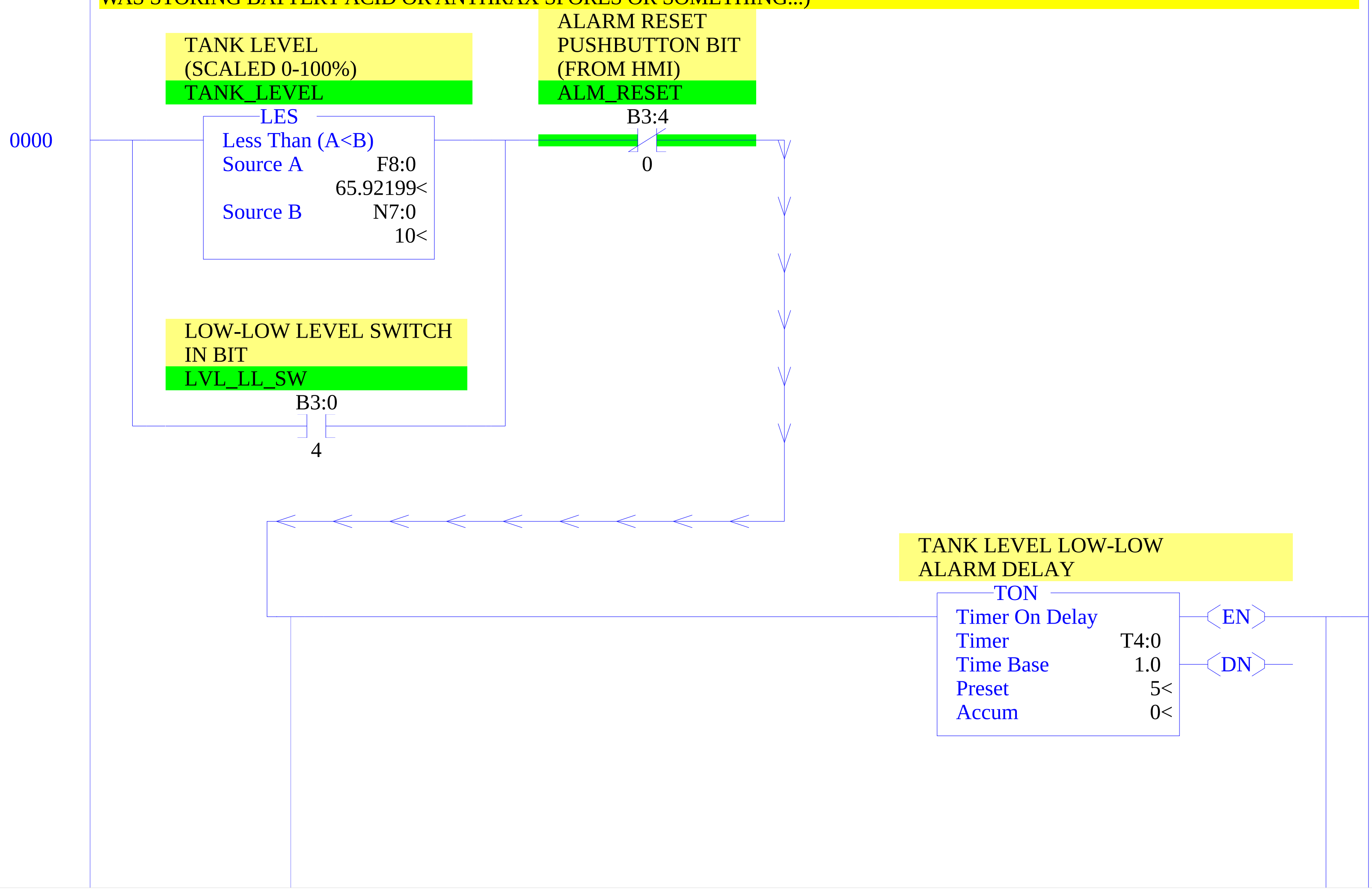
END

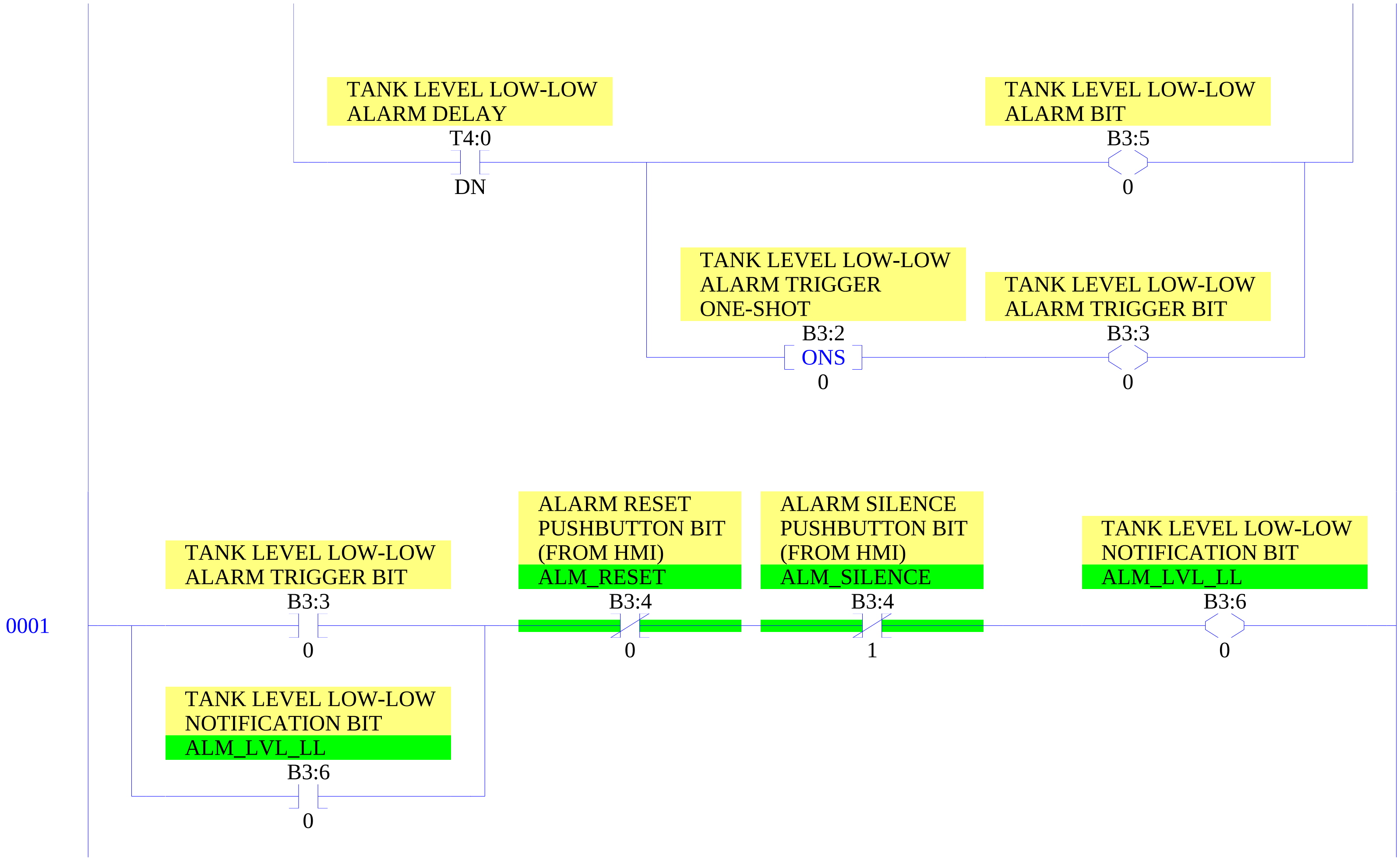
OK, NOW FOR ALARMS. THERE ARE MANY WAYS TO CREATE, TERMINATE AND MANAGE ALARMS. SOME WAYS ARE WAY BETTER, MORE PRACTICAL, MORE USEFUL, AND MORE OPERATOR-FRIENDLY THAN OTHERS! SOME PROGRAMMERS LET THE PLC HANDLE ALARMS, OTHERS DELEGATE THAT FUNCTIONALITY TO THEIR HMI PROGRAM. SOME HMI'S DO IT WELL, OTHERS DON'T. I'VE WRITTEN THIS ALARM LOGIC ASSUMING THE PLC WILL BE HANDLING ALL ALARM FUNCTIONALITY, WHICH MEANS THIS PROGRAM IS SUITABLE FOR USE WITH A MUCH WIDER VARIETY OF HMI'S. MORE IMPORTANTLY, YOU WILL BE ABLE TO LEARN A LOT ABOUT PRACTICAL ALARM MANAGEMENT.

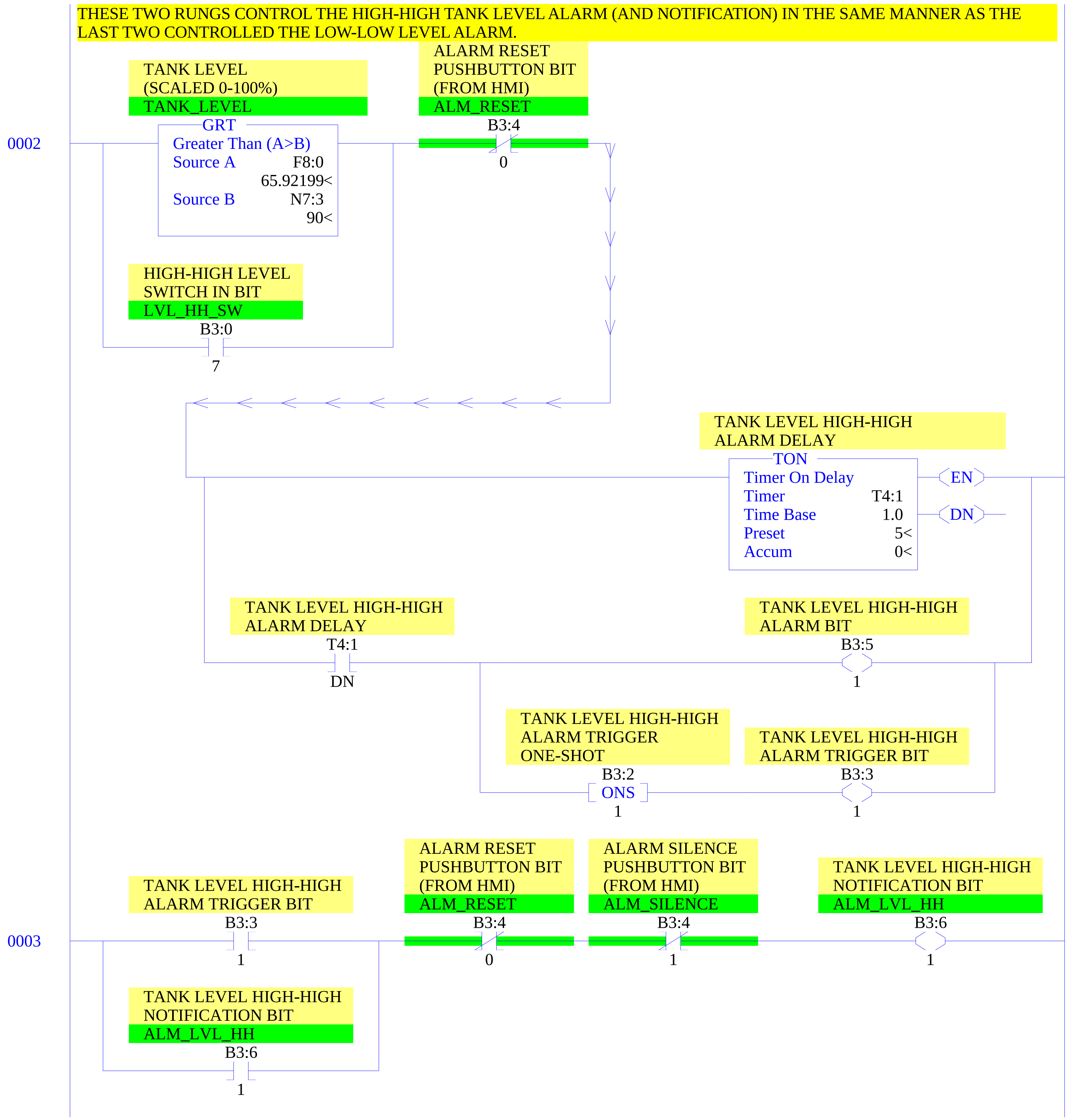
EACH ALARM IN THIS PROGRAM WILL BE GOVERNED BY TWO OR THREE RUNGS. THIS FIRST ALARM IS GOVERNED BY TWO. FOR THE TANK ALARMS, WE ASSUME WE HAVE DUAL LEVEL CONTROLS IN OUR TANK. IMAGINE WE HAVE FOUR LEVEL SWITCHES (FLOAT SWITCHES), ONE AT THE LOW-LOW LEVEL, ONE AT THE LOW LEVEL, ONE AT THE HIGH LEVEL AND ANOTHER AT THE HIGH-HIGH LEVEL. WE ALSO HAVE AN ANALOG SENSOR (RADAR OR HYDROSTATIC PRESSURE FOR INSTANCE). ON THE LEFT OF OUR RUNG, WE ARE EXAMINING BOTH CONTROLS, EITHER OF WHICH CAN TELL US WE HAVE A LOW-LOW LEVEL. IF OUR ANALOG-MEASURED TANK LEVEL (F8:0) IS LOWER THAN OUR LOW-LOW LEVEL SETPOINT (STORED IN N7:0), --- OR --- IF OUR LOW-LOW LEVEL SWITCH IS CLOSED, THEN WE ENERGIZE OUR DELAY TIMER. ONCE THE FIVE SECONDS ON THE TIMER EXPIRE, WE'LL TRIGGER OUR LOW-LOW ALARM BIT --- AND --- A SEPARATE LOW-LOW NOTIFICATION BIT (IN THE RUNG BELOW - 0001) USING THE FAMILIAR ONE-SHOT AND TRIGGER BIT (JUST LIKE WE USED IN CONTROLS 0004 AND 0005).

IN THIS ALARM, ONCE THE LOW-LOW LEVEL CONDITION DISAPPEARS, THE ALARM DOES AS WELL - AUTOMATICALLY, --- BUT --- THE NOTIFICATION REMAINS!!! WHY? IF A LEVEL ALARM OCCURS AT 3 O'CLOCK IN THE MORNING AND THEN FIXES ITSELF, DO WE WANT TO SHUT THE SYSTEM DOWN UNTIL SOMEBODY SHOWS UP AT TEN? MEANWHILE, A LOT OF PEOPLE HAVE NO WATER. OF COURSE NOT. SO WE'LL LET THE ALARM CLEAR ITSELF, --- BUT --- WE WANT THE NOTIFICATION TO STILL BE THERE WHEN THE OPERATOR SHOWS UP FOR WORK. THAT WAY, HE CAN SEE THAT SOMETHING NEEDS HIS ATTENTION. THE NOTIFICATION BIT REMAINS ENERGIZED UNTIL HE PRESSES THE ALARM RESET --- OR --- THE ALARM SILENCE BUTTON ON THE HMI. THESE ARE DIFFERENT. ONE (ALARM SILENCE) MAKES THE HORNS SHUT OFF AND THE MESSAGES ON THE HMI DISAPPEAR. THE OTHER (ALARM RESET) CAN ACTUALLY RESET ALARMS THAT ARE LOCKED IN. THE ALARM RESET WILL ALSO RESET THE ALARM TIMERS AND MAKE THE ALARMS DISAPPEAR --- HOWEVER --- IF THE CONDITION THAT PRECIPITATED THE ALARM REMAINS, THE ALARM AND NOTIFICATION WILL REAPPEAR ONCE THE DELAY TIMER COUNTS DOWN ANEW.

WE WON'T BE LOCKING OUR LEVEL ALARMS IN, BUT YOU CAN BET WE'LL BE LOCKING IN OUR PUMP AND TEMPERATURE ALARMS! THOSE CAN BECOME EXPENSIVE / DANGEROUS IF LEFT UNATTENDED (AS COULD LEVEL ALARMS IF THE TANK WAS STORING BATTERY ACID OR ANTHRAX SPORES OR SOMETHING...)

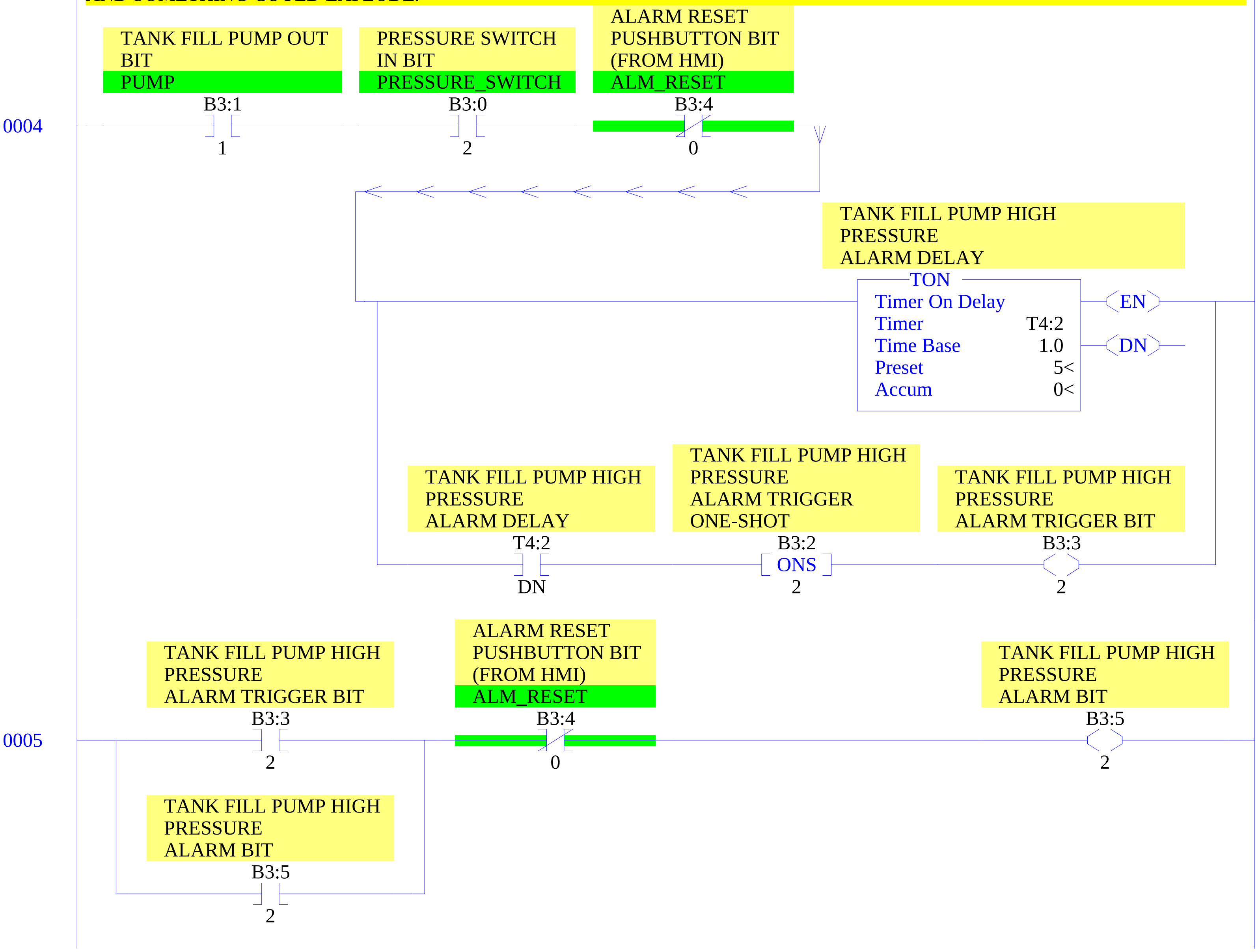


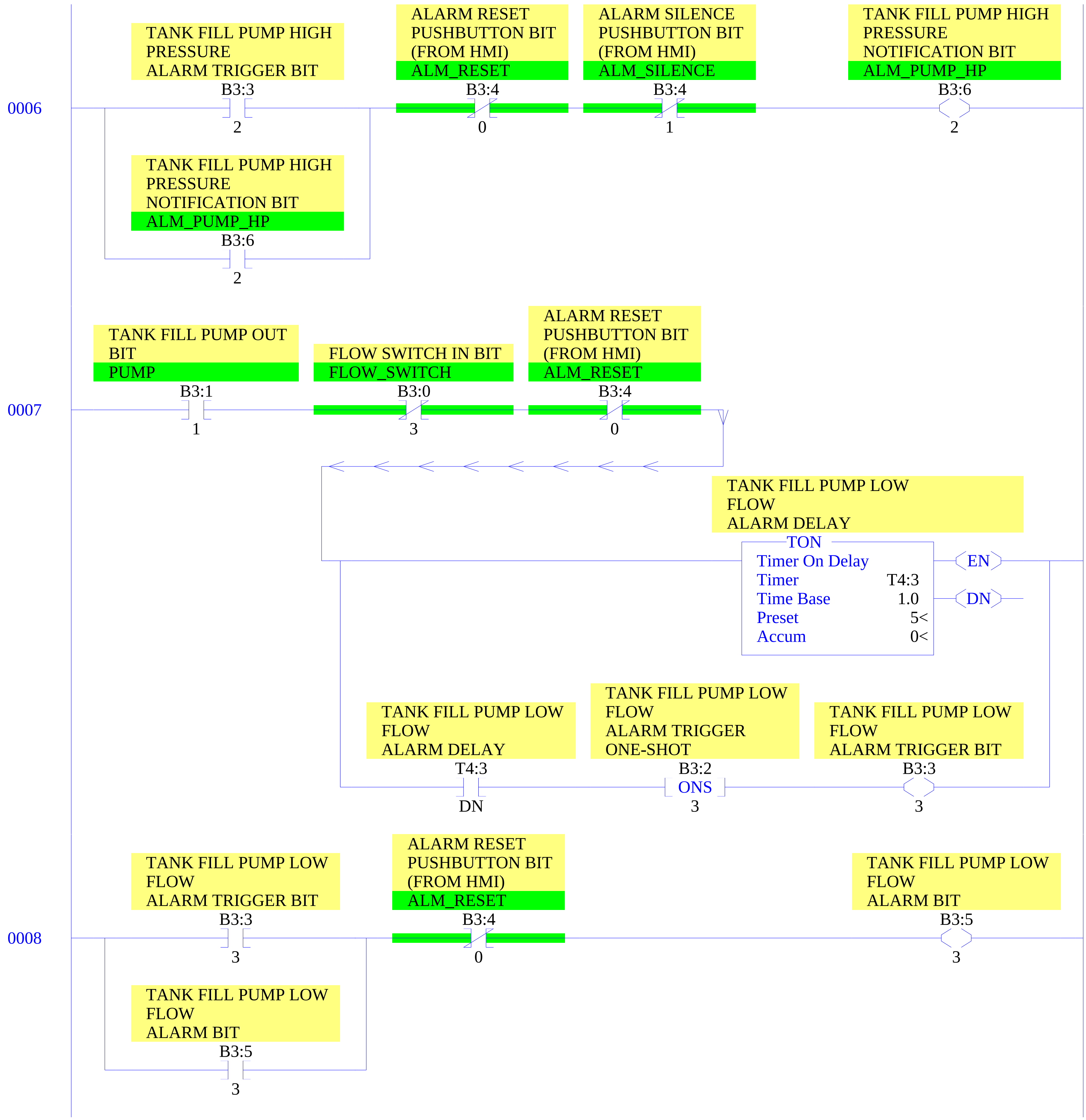

TANK LEVEL LOW-LOW ALARM DELAY
T4:0
DN
TANK LEVEL LOW-LOW ALARM BIT
B3:5
0
TANK LEVEL LOW-LOW ALARM TRIGGER ONE-SHOT
B3:2
ONS
0
TANK LEVEL LOW-LOW ALARM TRIGGER BIT
B3:3
0
0001
TANK LEVEL LOW-LOW ALARM TRIGGER BIT
B3:3
0
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
ALARM SILENCE PUSHBUTTON BIT (FROM HMI)
ALM_SILENCE
B3:4
1
TANK LEVEL LOW-LOW NOTIFICATION BIT
ALM_LVL_LL
B3:6
0
TANK LEVEL LOW-LOW NOTIFICATION BIT
ALM_LVL_LL
B3:6
0

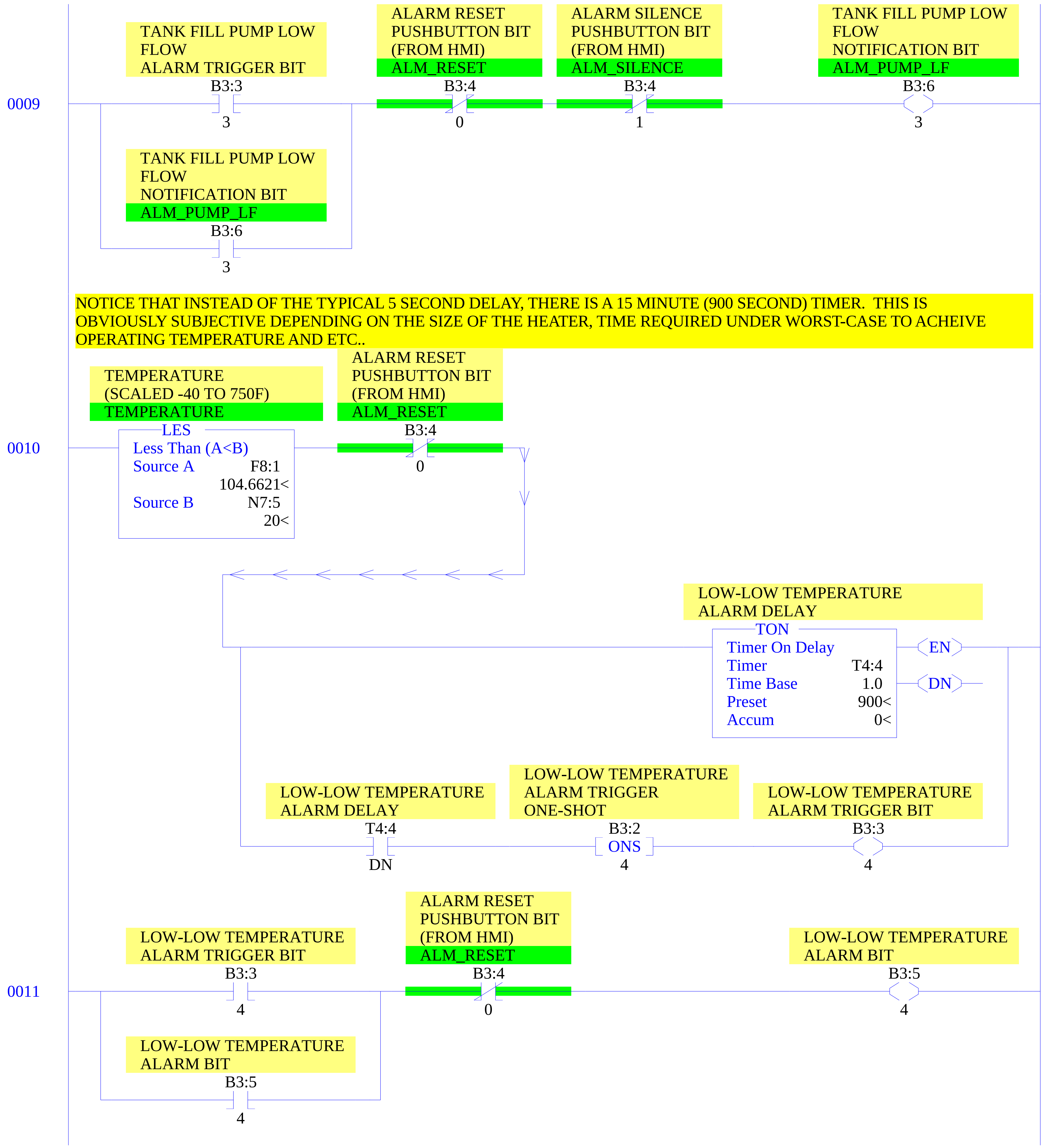

THESE TWO RUNGS CONTROL THE HIGH-HIGH TANK LEVEL ALARM (AND NOTIFICATION) IN THE SAME MANNER AS THE LAST TWO CONTROLLED THE LOW-LOW LEVEL ALARM.
0002
TANK LEVEL
(SCALED 0-100%)
TANK_LEVEL
GRT
Greater Than (A>B)
Source A F8:0
65.92199<
Source B N7:3
90<
ALARM RESET
PUSHBUTTON BIT
(FROM HMI)
ALM_RESET
B3:4
0
HIGH-HIGH LEVEL
SWITCH IN BIT
LVL_HH_SW
B3:0
7
TANK LEVEL HIGH-HIGH
ALARM DELAY
TON
Timer On Delay
Timer T4:1
Time Base 1.0
Preset 5<
Accum 0<
EN
DN
TANK LEVEL HIGH-HIGH
ALARM DELAY
T4:1
DN
TANK LEVEL HIGH-HIGH
ALARM BIT
B3:5
1
TANK LEVEL HIGH-HIGH
ALARM TRIGGER
ONE-SHOT
B3:2
ONS
1
TANK LEVEL HIGH-HIGH
ALARM TRIGGER BIT
B3:3
1
0003
TANK LEVEL HIGH-HIGH
ALARM TRIGGER BIT
B3:3
1
ALARM RESET
PUSHBUTTON BIT
(FROM HMI)
ALM_RESET
B3:4
0
ALARM SILENCE
PUSHBUTTON BIT
(FROM HMI)
ALM_SILENCE
B3:4
1
TANK LEVEL HIGH-HIGH
NOTIFICATION BIT
ALM_LVL_HH
B3:6
1
TANK LEVEL HIGH-HIGH
NOTIFICATION BIT
ALM_LVL_HH
B3:6
1

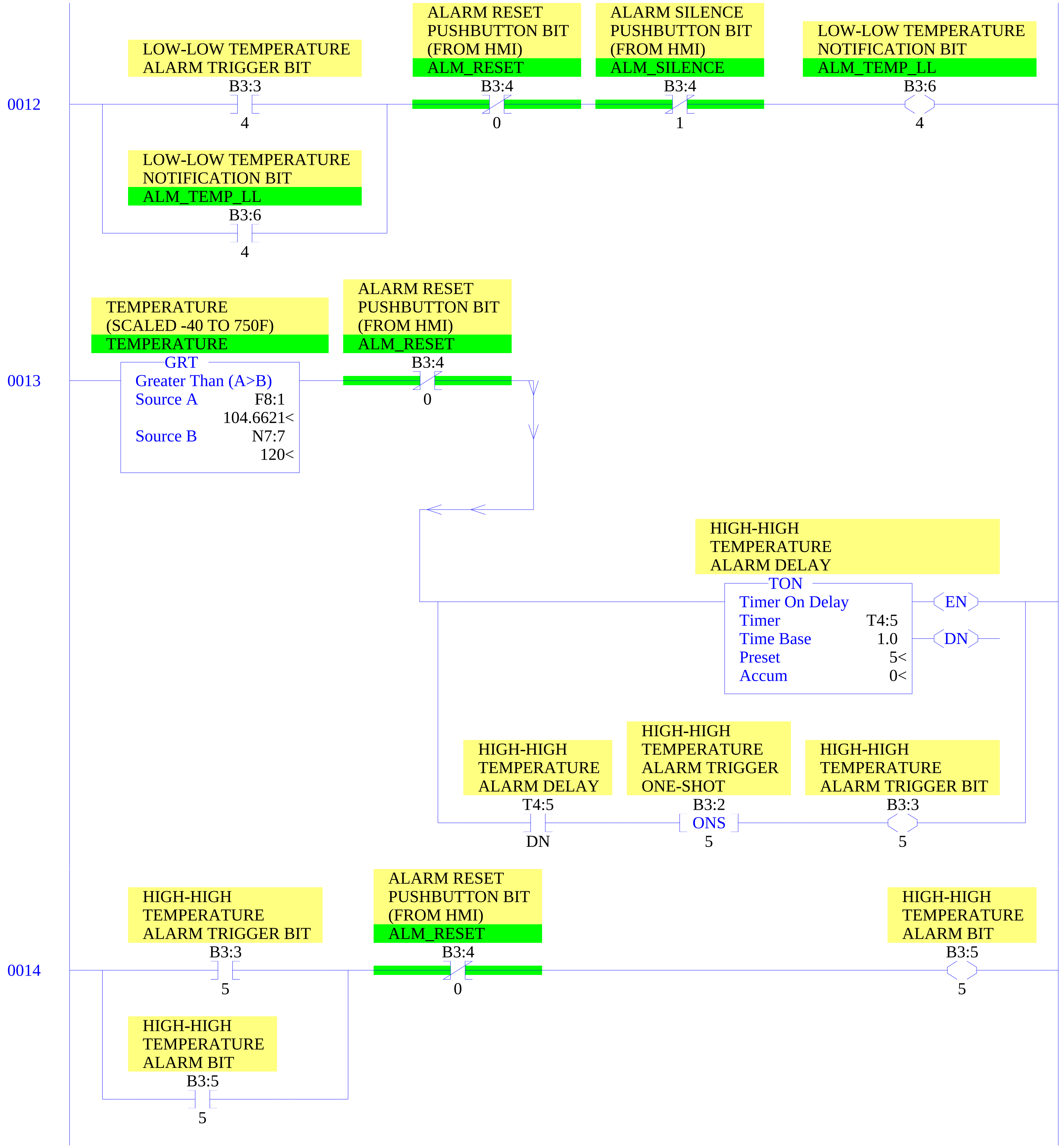
FROM HERE ON DOWN, THE ALARMS WILL BE GOVERNED BY THREE RUNGS EACH. THE FIRST RUNG WILL TRIGGER THE ALARM. THE SECOND RUNG WILL GOVERN THE ALARM BIT. THE THIRD RUNG WILL GOVERN THE NOTIFICATION BIT. UNLIKE THE LEVEL ALARMS, THE PUMP AND TEMPERATURE ALARMS WILL HAVE A HOLD-IN. THUS, EVEN IF THE CONDITION DISAPPEARS, THE ALARM WILL REMAIN UNTIL THE OPERATOR PRESSES THE ALARM RESET BUTTON (AND OF COURSE THE PROBLEM IS CORRECTED).

IF A PUMP IS OVER-PRESSURIZING A PIPE, WE WANT THE OPERATOR TO CHECK IT OUT. OTHERWISE, SOMETHING COULD EXPLODE.
IF A PUMP IS NOT CREATING FLOW, WE WANT THE OPERATOR TO CHECK IT OUT. OTHERWISE, THE PUMP COULD CAVITATE.
IF A HEATER IS OVER- HEATING, WE WANT THE OPERATOR TO CHECK IT OUT. OTHERWISE, SOMETHING COULD CATCH ON FIRE.
IF A HEATER IS NOT CREATING HEAT, WE WANT THE OPERATOR TO CHECK IT OUT. OTHERWISE, A SENSOR COULD BE BAD AND SOMETHING COULD EXPLODE.




0006
TANK FILL PUMP HIGH PRESSURE ALARM TRIGGER BIT
B3:3
2
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
ALARM SILENCE PUSHBUTTON BIT (FROM HMI)
ALM_SILENCE
B3:4
1
TANK FILL PUMP HIGH PRESSURE NOTIFICATION BIT
ALM_PUMP_HP
B3:6
2
TANK FILL PUMP HIGH PRESSURE NOTIFICATION BIT
ALM_PUMP_HP
B3:6
2
0007
TANK FILL PUMP OUT BIT
PUMP
B3:1
1
FLOW SWITCH IN BIT
FLOW_SWITCH
B3:0
3
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
TANK FILL PUMP LOW FLOW ALARM DELAY
TON
Timer On Delay
Timer T4:3
Time Base 1.0
Preset 5<
Accum 0<
EN
DN
TANK FILL PUMP LOW FLOW ALARM DELAY
T4:3
DN
TANK FILL PUMP LOW FLOW ALARM TRIGGER ONE-SHOT
B3:2
ONS
3
TANK FILL PUMP LOW FLOW ALARM TRIGGER BIT
B3:3
3
0008
TANK FILL PUMP LOW FLOW ALARM TRIGGER BIT
B3:3
3
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
TANK FILL PUMP LOW FLOW ALARM BIT
B3:5
3
TANK FILL PUMP LOW FLOW ALARM BIT
B3:5
3


0009
TANK FILL PUMP LOW FLOW ALARM TRIGGER BIT
B3:3
3
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
ALARM SILENCE PUSHBUTTON BIT (FROM HMI)
ALM_SILENCE
B3:4
1
TANK FILL PUMP LOW FLOW NOTIFICATION BIT
ALM_PUMP_LF
B3:6
3
TANK FILL PUMP LOW FLOW NOTIFICATION BIT
ALM_PUMP_LF
B3:6
3
NOTICE THAT INSTEAD OF THE TYPICAL 5 SECOND DELAY, THERE IS A 15 MINUTE (900 SECOND) TIMER. THIS IS OBVIOUSLY SUBJECTIVE DEPENDING ON THE SIZE OF THE HEATER, TIME REQUIRED UNDER WORST-CASE TO ACHEIVE OPERATING TEMPERATURE AND ETC..
0010
TEMPERATURE (SCALED -40 TO 750F)
TEMPERATURE
LES
Less Than (A<B)
Source A F8:1
104.6621<
Source B N7:5
20<
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
LOW-LOW TEMPERATURE ALARM DELAY
TON
Timer On Delay
Timer T4:4
Time Base 1.0
Preset 900<
Accum 0<
EN
DN
LOW-LOW TEMPERATURE ALARM DELAY
T4:4
DN
LOW-LOW TEMPERATURE ALARM TRIGGER ONE-SHOT
B3:2
ONS
4
LOW-LOW TEMPERATURE ALARM TRIGGER BIT
B3:3
4
0011
LOW-LOW TEMPERATURE ALARM TRIGGER BIT
B3:3
4
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
LOW-LOW TEMPERATURE ALARM BIT
B3:5
4
LOW-LOW TEMPERATURE ALARM BIT
B3:5
4


0012
LOW-LOW TEMPERATURE ALARM TRIGGER BIT
B3:3
4
LOW-LOW TEMPERATURE NOTIFICATION BIT
ALM_TEMP_LL
B3:6
4
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
ALARM SILENCE PUSHBUTTON BIT (FROM HMI)
ALM_SILENCE
B3:4
1
LOW-LOW TEMPERATURE NOTIFICATION BIT
ALM_TEMP_LL
B3:6
4
0013
TEMPERATURE (SCALED -40 TO 750F)
TEMPERATURE
GRT
Greater Than (A>B)
Source A F8:1
104.6621<
Source B N7:7
120<
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
HIGH-HIGH TEMPERATURE ALARM DELAY
TON
Timer On Delay
Timer T4:5
Time Base 1.0
Preset 5<
Accum 0<
EN
DN
HIGH-HIGH TEMPERATURE ALARM DELAY
T4:5
DN
HIGH-HIGH TEMPERATURE ALARM TRIGGER ONE-SHOT
B3:2
ONS
5
HIGH-HIGH TEMPERATURE ALARM TRIGGER BIT
B3:3
5
0014
HIGH-HIGH TEMPERATURE ALARM TRIGGER BIT
B3:3
5
HIGH-HIGH TEMPERATURE ALARM BIT
B3:5
5
ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0
HIGH-HIGH TEMPERATURE ALARM BIT
B3:5
5

0015

HIGH-HIGH TEMPERATURE ALARM TRIGGER BIT
B3:3
5

HIGH-HIGH TEMPERATURE NOTIFICATION BIT
ALM_TEMP_HH
B3:6
5

ALARM RESET PUSHBUTTON BIT (FROM HMI)
ALM_RESET
B3:4
0

ALARM SILENCE PUSHBUTTON BIT (FROM HMI)
ALM_SILENCE
B3:4
1

HIGH-HIGH TEMPERATURE NOTIFICATION BIT
ALM_TEMP_HH
B3:6
5

THIS IS A CATEGORY ALARM BIT. IF WE HAVE A LOW-LOW OR HIGH-HIGH LEVEL ALARM, WE WILL ENERGIZE THIS BIT. THIS MAY BE READ BY ANOTHER SYSTEM OVER A NETWORK OR DISPLAYED ON THE HMI.

0016

TANK LEVEL LOW-LOW NOTIFICATION BIT
ALM_LVL_LL
B3:6
0

TANK LEVEL HIGH-HIGH NOTIFICATION BIT
ALM_LVL_HH
B3:6
1

TANK LEVEL ALARM NOTIFICATION BIT
ALM_TANK
B3:7
0

THIS IS A CATEGORY ALARM BIT. IF WE HAVE A LOW FLOW OR HIGH PRESSURE ALARM, WE WILL ENERGIZE THIS BIT. THIS MAY BE READ BY ANOTHER SYSTEM OVER A NETWORK OR DISPLAYED ON THE HMI.

0017

TANK FILL PUMP HIGH PRESSURE NOTIFICATION BIT
ALM_PUMP_HP
B3:6
2

TANK FILL PUMP LOW FLOW NOTIFICATION BIT
ALM_PUMP_LF
B3:6
3

TANK FILL PUMP ALARM NOTIFICATION BIT
ALM_PUMP
B3:7
1

THIS IS A CATEGORY ALARM BIT. IF WE HAVE A LOW-LOW OR HIGH-HIGH TEMPERATURE ALARM, WE WILL ENERGIZE THIS BIT. THIS MAY BE READ BY ANOTHER SYSTEM OVER A NETWORK OR DISPLAYED ON THE HMI.

0018

LOW-LOW TEMPERATURE NOTIFICATION BIT
ALM_TEMP_LL
B3:6
4

HIGH-HIGH TEMPERATURE NOTIFICATION BIT
ALM_TEMP_HH
B3:6
5

TEMPERATURE ALARM NOTIFICATION BIT
ALM_TEMP
B3:7
2

IF WE HAVE AN ALARM IN ANY CATEGORY (TANK, PUMP OR TEMPERATURE), WE WILL ENERGIZE THE GENERAL ALARM BIT. THIS MAY BE READ BY ANOTHER SYSTEM OVER A NETWORK OR DISPLAYED ON THE HMI.

0019

TANK LEVEL ALARM
NOTIFICATION BIT
ALM_TANK
B3:7
0

TANK FILL PUMP ALARM
NOTIFICATION BIT
ALM_PUMP
B3:7
1

TEMPERATURE ALARM
NOTIFICATION BIT
ALM_TEMP
B3:7
2

GENERAL ALARM
NOTIFICATION BIT
B3:7
3

0020

END

LAD 9 - DISPLAY --- Total Rungs in File = 2

HEATER OUTPUT FOR
DISPLAY (0-100%)
HEATER_OUTPUT

0000

SCP	
Scale w/Parameters	
Input	N7:9
	0<
Input Min.	0
	0<
Input Max.	16383
	16383<
Scaled Min.	0
	0<
Scaled Max.	100
	100<
Output	N7:10
	0<

0001 END