VAR:

// PRESSURE VAR

PPV\_0: INT; // Pressure PPV\_0 at A-1

PPV\_1: INT; // Pressure PPV\_1 at A-2

PPV\_2: INT; // Pressure PPV\_2 at A-3

PPV\_3: INT; // Pressure PPV\_3 at A-4

PPV\_4: INT; // Pressure PPV\_4 at TA-3 and TA-4

PPV\_5: INT; // Pressure PPV\_5 at TA-1 and TA-2

PPV\_6: INT; // Pressure PPV\_6 at TA-5

PPV\_0\_STATUS: Bool;

PPV\_1\_STATUS: Bool;

PPV\_2\_STATUS: Bool;

PPV\_3\_STATUS: Bool;

PPV\_4\_STATUS: Bool;

PPV\_5\_STATUS: Bool;

PPV\_6\_STATUS: Bool;

// SOLENOID VAR

S\_1: INT; // SOLENOID READINGS

S\_2: INT;

S\_3: INT;

S\_4: INT;

S\_5: INT;

// TEMPEARTUE VAR

PT0: INT; // TEMPEATURE OF PT0 MEASURED AT NOZZLE 1

PT1: INT; // TEMPEATURE OF PT1 MEASURED AT NOZZLE 1

PT2: INT; // TEMPEATURE OF PT2 MEASURED AT NOZZLE 2

PT3: INT; // TEMPEATURE OF PT3 MEASURED AT NOZZLE 2

PT4: INT; // TEMPEATURE OF PT4 MEASURED AT TA\_1 AND TA\_2

PT5: INT; // TEMPEATURE OF PT5 MEASURED AT TA\_1 AND TA\_2

PT6: INT; // TEMPEATURE OF PT6 MEASURED AT TA\_3 AND TA\_4

PT7: INT; // TEMPEATURE OF PT7 MEASURED AT TA\_3 AND TA\_4

PT8: INT; // TEMPEATURE OF PT8 MEASURED AT TA\_5

// Constant Definition

CONSTANT PRESSURE\_NOTOPEN: = FALSE;

CONSTANT RRESSURE\_OPEN: = TRUE;

CONSTANT PROPULSION\_TANK\_MAX\_PRESSURE: = 4550;

CONSTANT PILOT\_AIR\_MAX\_PRESSURE: = 250;

CONSTANT SOLENOID\_MAX\_VOLTAGE: = 24;

CONSTANT SYS\_FAILURE: FALSE;

CONSTANT NORMAL: = TRUE;

CONSTANT NOMINAL: = ;

CONSTANT TBD: = ;

// SYSTEM VAR

POPSYS: Bool; // PROPULSION SYSTEM STATUS

END\_VAR

// PRESSURE PROBLEM

IF PPV\_0 <> SICK AND PPV\_0 == UNDEFINED THEN

POPSYS: = SYS\_FAILURE;

PPV\_0\_STATUS = PRESSURE\_NOTOPEN;

ELSIF PPV\_1 <> SICK AND PPV\_1 == UNDEFINED THEN

POPSYS: = SYS\_FAILURE;

PPV\_1\_STATUS = PRESSURE\_NOTOPEN;

ELSIF PPV\_2 <> SICK AND PPV\_2 == UNDEFINED THEN

POPSYS: = SYS\_FAILURE;

PPV\_2\_STATUS = PRESSURE\_NOTOPEN;

ELSIF PPV\_3 <> SICK AND PPV\_3 == UNDEFINED THEN

POPSYS: = SYS\_FAILURE;

ELSIF PPV\_4 <> SICK AND PPV\_4 == UNDEFINED THEN

POPSYS: = SYS\_FAILURE;

ELSIF (PPV\_0 <> UNDEFINED AND PPV\_1 <> UNDEFINED AND PPV\_2 <> UNDEFINED AND PPV\_3 <> UNDEFINED) AND (PPV\_0 – PPV\_2 > 30 OR PPV\_0 – PPV\_3 > 30 OR PPV\_1 - PPV\_2 > 30 OR PPV\_1 – PPV\_3 > 30) THEN

POPSYS: = SYS\_FAILURE;

ELSIF (PT\_0 <> UNDEFINED AND PT\_1 <> UNDEFINED AND PT\_2 <> UNDEFINED AND PT\_3 <> UNDEFINED) AND (PT\_0 – PT\_2 > 40 OR PT\_0 – PT\_3 > 40 OR PT\_1 - PT\_2 > 40 OR PT\_1 – PT\_3 > 40) THEN

POPSYS: = SYS\_FAILURE;

ELSIF (PPV\_5 > PROPULSION\_TANK\_MAX\_PRESSURE) OR (PPV\_4 > PROPULSION\_TANK\_MAX\_PRESSURE) OR (PPV\_6 > PILOT\_AIR\_MAX\_PRESSURE) THEN

POPSYS: = SYS\_FAILURE;

END\_IF

// Solenoid values

IF S\_1 < SOLENOID\_MAX\_VOLTAGE OR S\_2 < SOLENOID\_MAX\_VOLTAGE OR S\_3 < SOLENOID\_MAX\_VOLTAGE OR S\_4 < SOLENOID\_MAX\_VOLTAGE THEN

POPSYS: = SYS\_FAILURE

END\_IF

// TEMPERATURE PROBLEM

IF (PT4 == NOMINAL AND PT5 == NOMINAL) AND (PT4 + PT5)/2 <> TBD THEN

POPSYS: = SYS\_FAILURE;

ELSIF (PT4 == SICK OR PT5 == SICK) AND (PT4 <> TBD OR PT5 <> TBD) THEN

POPSYS: = SYS\_FAILURE;

ELSIF (PT6 == NOMINAL AND PT7 == NOMINAL) AND (PT6 + PT7)/2 <> TBD THEN

POPSYS: = SYS\_FAILURE;

ELSIF (PT6 == SICK OR PT7 == SICK) AND (PT6 <> TBD OR PT7 <> TBD) THEN

POPSYS: = SYS\_FAILURE;

ELSIF PT8 == NOMINAL AND PT8 <> TBD THEN

POPSYS: = SYS\_FAILURE;

END\_IF