

**EET 438B**  
**Sequential Control and Data Acquisition**  
**Laboratory 9**  
**Analog Input and Output Using PLCs**

**Laboratory Learning Objectives**

- 1.) Identify comparison and arithmetic PLC instructions in both ladder logic and function block used to manipulate digital data derived from analog inputs
- 2.) Create a program to read analog input values and scale them to meet given specifications.
- 3.) Create a program to write analog output values.
- 4.) Create a program that reads analog inputs processes the data and writes digital outputs and/or analog output to meet a given specification.

**Technical Background**

Modern programmable logic controller have functionality beyond that of the traditional uses such as logic, counting and timing. The new generation of PLCs can take analog inputs from transducers and produce analog outputs to drive other processes. This functionality can be part of the base processor or be added as a plug-in module. A PLC can implement a digital version of the common proportional-integral-derivative (PID) controller by utilizing these access points. PLC languages include functions that allow programmers to access the digitized analog inputs, mathematically modify the result, and write the modified result to an analog output. Logical comparison functions provide a means for testing digitized analog values with greater/less-than and equal/not-equal tests. The results of these tests can modify digital I/O points in programs.

The Micro 800 trainer includes two forms of analog input and one form of analog output. The first type of analog input is a plug-in thermocouple module (Model 2080-TC2) that reads the output of the two, K-type, thermocouples mounted under the PLC on the trainer. The other analog input is a four channel, 14 bit isolated expansion module (Model 2085-IF4) that mounts on the side of the PLC. The Connected Components Workbench software can configure the module to read either voltage or current inputs. A Model 2085-OF4 expansion module provides four channels of 12 bit, isolated analog output. Current or voltage output is possible and set in the PLC programming software.

Table 1 shows the input/output ranges of the analog modules installed on the trainer PLC. The ranges have three supported data formats. These are Raw/Proportional, Engineering Units, and Percent Range. The software allows channel configuration for any of these formats. Table 2 summarizes the characteristics of each data format. Note that the ranges specified in Table 1 do not correspond to the full range values converted by the modules. Table 3 gives the full range

values along with the data formats for each input type. The full range values are used for scaling computations. Table 4 gives the same information for the Model 2085-OF4 module.

Table-1 Analog Expansion Module Input/Output Ranges

| Module Model | Input/Output Type and Range |
|--------------|-----------------------------|
| 2085-IF4     | 0-20 mA<br>4-20 mA          |
| 2085-OF4     | -10-+10 V<br>0-10 V         |

Table-2 Data Format Characteristics

| Data Format       | Description/Characteristics   |
|-------------------|---|
| Raw/Proportional  | The stored value is proportional to the selected input type and scaled to the maximum range allow by the number of bits. Example: $\pm 10$ V dc maps to -32768-+32767.                            |
| Engineering Units | The module scales the analog data to the actual current and voltage ranges specified for the selected input type and span. The resolution of the engineering units is 1 mV or 0.001 mA per count. |
| Percent Range     | The input signal maps to the range 0-100% of normal operating range. Example: 0-10 Vdc maps to 0-100%.  |

Table-3 Valid Data Format Ranges and Full Range Inputs Model 2085-IF4

| Data Format       | Range/Type |               |              |                 |             |
|-------------------|------------|---------------|--------------|-----------------|-------------|
|                   | Rated      | 0-20 mA       | 4-20 mA      | -10 - +10 V     | 0-10 V      |
|                   | Full Range | 0-21 mA       | 3.2-21 mA    | -10.5 - +10.5 V | -0.5-10.5 V |
| Raw/Proportional  |            | -32768-+32767 |              |                 |             |
| Engineering Units |            | 0-21,000      | 3,200-21,000 | -10,500-+10,500 | -500-10,500 |
| Percent Range     |            | 0-10,500      | -500-10,625  | Not Supported   | -500-10,500 |

Table-4 Valid Data Format Ranges and Full Range Inputs Model 2085-OF4

| Data Format       | Range/Type |               |              |                 |          |
|-------------------|------------|---------------|--------------|-----------------|----------|
|                   | Rated      | 0-20 mA       | 4-20 mA      | -10 - +10 V     | 0-10 V   |
|                   | Full Range | 0-21 mA       | 3.2-21 mA    | -10.5 - +10.5 V | 0-10.5 V |
| Raw/Proportional  |            | -32768-+32767 |              |                 |          |
| Engineering Units |            | 0-21,000      | 3,200-21,000 | -10,500-+10,500 | 0-10,500 |
| Percent Range     |            | 0-10,500      | -500-10,625  | Not Supported   | 0-10,500 |

### Data Format Value Conversions

Integer values represent the analog input/output data for all ranges and input/output types. Scaling equations convert the data format values into measured analog values and vice-versa. Equation (1) shows a scaling equation for voltage inputs while Equation (2) gives the equation for current inputs

$$DF - DF_{\min} = \left( \frac{DF_{\max} - DF_{\min}}{V_{\max} - V_{\min}} \right) \cdot (V_{\text{in}} - V_{\min}) \quad (1)$$

$$DF - DF_{\min} = \left( \frac{DF_{\max} - DF_{\min}}{I_{\max} - I_{\min}} \right) \cdot (I_{\text{in}} - I_{\min}) \quad (2)$$

Where:

- DF = data format value for given input V or I,
- DF<sub>min</sub> = minimum data format value for given range,
- DF<sub>max</sub> = maximum data format value for given range,
- V<sub>min</sub> = minimum voltage value for given range,
- V<sub>max</sub> = maximum voltage value for given range,
- I<sub>min</sub> = minimum current value for given range,
- I<sub>max</sub> = maximum current value for given range,
- V<sub>in</sub> = input voltage value,
- V<sub>max</sub> = input current value.

**Example 1:** Find the analog value that represents the raw/proportional data value of -10,550. The input range is 4-20 mA.

Derive the equation for this range from (2) using the following values

$$\begin{array}{ll} DF_{\max} = 32,767 & DF_{\min} = -32768 \\ I_{\max} = 21 \text{ mA} & I_{\min} = 3.2 \text{ mA} \quad \text{Use full range values.} \\ DF = -10,550 & I_{\text{in}} = ??? \end{array}$$

Place the known values into (2) and simplify.

$$DF - (-32767) = \left( \frac{(32768) - (-32767)}{21.0 - 3.2 \text{ mA}} \right) \cdot (I_{\text{in}} - 3.2 \text{ mA})$$

$$DF + 32768 = \left( \frac{65535}{17.8 \text{ mA}} \right) \cdot (I_{\text{in}} - 3.2 \text{ mA})$$

$$DF + 32768 = 3681.742 / \text{mA} \cdot (I_{\text{in}} - 3.2 \text{ mA})$$

$$DF = (3681.742 / \text{mA}) I_{\text{in}} - 11782 - 32768$$

$$DF = (3681.742 / \text{mA}) I_{\text{in}} - 44550$$

Since we have a known value of DF, we must solve the above equation for  $I_{\text{in}}$ .

$$DF = (3681.742 / \text{mA}) I_{\text{in}} - 44550$$

$$DF + 44550 = (3681.742 / \text{mA}) I_{\text{in}}$$

$$\frac{(DF + 44550)}{3681.742 / \text{mA}} = I_{\text{in}}$$

$$I_{\text{in}} = 0.0002716 \text{ mA} \cdot DF + 12.1$$

Now place the given value of DF into the equation solved for  $I_{\text{in}}$  and find the value associated with  $DF = -10,550$ .

$$I_{\text{in}} = 0.0002716 \text{ mA} \cdot DF + 12.1$$

$$I_{\text{in}} = 0.0002716 \text{ mA} \cdot (-10,550) + 12.1$$

$$I_{\text{in}} = 9.235 \text{ mA}$$

The value of input current is 9.235 mA.

**Example 2:** A  $\pm 10$  Vdc analog input has a value of -5.67 V. What is the value of the data format if the channel uses the raw/proportional format.

Derive the equation for this range from (1) using the following values

|                                   |   |
|-----------------------------------|---|
| $DF_{\text{max}} = 32,767$        | $DF_{\text{min}} = -32768$                                |
| $V_{\text{max}} = 10.5 \text{ V}$ | $V_{\text{min}} = -10.5 \text{ V}$ Use full range values. |
| $DF = ???$                        | $V_{\text{in}} = -5.67 \text{ V}$                         |

Substitute in the values for the input/output ranges and simplify the equation.

$$DF - DF_{\text{min}} = \left( \frac{DF_{\text{max}} - DF_{\text{min}}}{V_{\text{max}} - V_{\text{min}}} \right) (V_{\text{in}} - V_{\text{min}})$$

$$DF - (-32768) = \left( \frac{32767 - (-32768)}{10.5 - (-10.5)} \right) (V_{in} - (-10.5))$$

$$DF + 32768 = \left( \frac{65535}{21} \right) (V_{in} + 10.5)$$

$$DF + 32768 = (3121/V)(V_{in} + 10.5)$$

$$DF + 32768 = (3121/V)V_{in} + 32771$$

$$DF = (3121/V)V_{in} + 3$$

Now insert the value of  $V_{in}$  into the derived equation to find the digital representation of the analog input voltage rounded to an integer.

$$DF = (3121/V)V_{in} + 3$$

$$DF = (3121/V)(-5.67) + 3$$

$$DF = -17693$$

### Analog I/O Memory Mapping

All PLC input/output points have corresponding locations in the processors memory. Analog input card channels have 14-bit representations of analog inputs that map into the global variables identified in the form:  $\_IO\_Xx\_AI\_yy$ . The parameter x indicates the expansion slot number, 1-4 and yy indicates the channel number 00 to 03 for a four input card. The analog input cards also have status words that indicate the condition of the expansion module and specify alarms. Table 5 show the status mapping for the 2085-IF4 analog input module. The global variables in the form of  $\_IO\_Xx\_ST\_yy$  identify the status words in the PLC memory. The x parameter is the slot number of the card and yy is the status word (00-02). Programmers access status bits using the address format  $\_IO\_Xx\_ST\_yy.zz$ , with zz indicating the bit number (00-15). Status words 1 and 2 contain upper and lower alarm bits that toggle when the input reaches a user-defined input level. Programmers can set these alarm levels while configuring the PLC hardware in the Connected Components Software.

Table-5 2085-IF4 Analog Input Card Status Word Mapping

| Word     | 15       | 14 | 13   | 12       | 11  | 10  | 9   | 8  | 7        | 6    | 5    | 4   | 3   | 2   | 1  | 0 |
|----------|----------|----|------|----------|-----|-----|-----|----|----------|------|------|-----|-----|-----|----|---|
| Status 0 | PU       | GF | CRC  | Reserved |     |     |     |    |          |      |      |     |     |     |    |   |
| Status 1 | Reserved |    | HHA1 | LLA1     | HA1 | LA1 | DE1 | S1 | Reserved | HHA0 | LLA0 | HA1 | LA1 | DE1 | S1 |   |
| Status 2 | Reserved |    | HHA3 | LLA3     | HA3 | LA3 | DE3 | S3 | Reserved | HHA2 | LLA2 | HA3 | LA3 | DE3 | S3 |   |

#### Table Legend

CRC = CRC error. Bit is set when error on data received

DE# = Data error on channel #. Bit set when input channels not receiving data

GF = General Fault. Bits set on memory errors or errors in reserved bits

HA# = High Alarm Over-range. Bits set when channel # input exceeds preset high limit.

HHA# = High-High Alarm Over-range. Bits set when channel # input exceeds preset high-high limit

LA# = Low Alarm Under-range. Bits set when channel # input falls below preset low limit.

**Table Legend (Continued)**

LLA# = Low-Low Under-range. Bits set when channel # input falls below preset low-low limit.

S# = Channel Fault. Bits set for channel # when input is open, have data error or out-of-range.

The memory map for the 2085-OF4 analog output module is similar to that of the analog input module. Analog outputs map to the global variables of the form `_IO_Xx_AO_yy`, where x identifies the slot number (1-4) and yy is the channel number (00-03). The analog output has control bits identified by the global variables `_IO_Xx_CO_00.zz`, where x represents the slot number (1-4) and zz the bit number (00-12). Table-6 shows the bit mapping of the control word in PLC memory. The CEx bits are written during the run mode to clear any DAC hardware errors resetting the associated channel. The UUX and UOX bits clear the under and over range alarms for channel x. The alarm conditions must be eliminated for the UUX and UOX bits to have an effect.

Table-6 2085-OF4 Analog Output Card Control Word Mapping

| Word      | 15       | 14 | 13 | 12 | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|-----------|----------|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Control 1 | Reserved |    |    |    | CE3 | CE2 | CE1 | CE0 | UU3 | UO3 | UU2 | UO2 | UU1 | UO1 | UU0 | UO0 |

Table 7 shows the status words for the 2085-OF4 analog output module. The first four status words in the memory map hold the binary value of the analog output. Status words 4 to 6

Table-7 2085-OF4 Analog Output Card Status Word Mapping

| Word     | 15       | 14 | 13  | 12       | 11       | 10 | 9  | 8  | 7        | 6  | 5  | 4  | 3        | 2  | 1  | 0  |
|----------|----------|----|-----|----------|----------|----|----|----|----------|----|----|----|----------|----|----|----|
| Status 4 | PU       | GF | CRC | Reserved |          |    |    |    | E3       | E2 | E1 | E0 | S3       | S2 | S1 | S0 |
| Status 5 | Reserved |    | U3  | O3       | Reserved |    | U2 | O2 | Reserved |    | U1 | O1 | Reserved |    | U0 | O0 |
| Status 6 | Reserved |    |     |          |          |    |    |    |          |    |    |    |          |    |    |    |

**Table Legend**

CRC = CRC error. Bit is set when error on data received

PU = Unexpected MCU reset.

GF = General Fault. Bits set on memory errors or errors in reserved bits.

Ex = DAC hardware error. Broken wire or high resistance load on channel x

Ox = Over-range flag. Attempting to drive DAC channel x beyond high output limit. Output set to highest DAC value

Ux = Under-range flag. Attempting to drive DAC channel x below lowest output limit. DAC output set to low limit.

Sx = Channel fault. Indicates a fault on channel x.

identify error conditions for the analog output module. These flags include upper and lower range limits, hardware error flags, and channel faults.

## Thermocouple Plug-In Input Module (2080-TC2)

The PLC trainers have thermocouple interface modules installed that allow the direct reading of thermocouples. The modules include cold junction compensation that accounts for the dissimilar metal junction voltages produced at the module inputs. The module has two input channels for thermocouples. The channel input parameters are set using the Connected Components Workbench software. Figure 1 shows the Connected Component Workbench screen for configuring the thermocouple module. The programmer selects the data update rate and the thermocouple type to complete channel configuration.

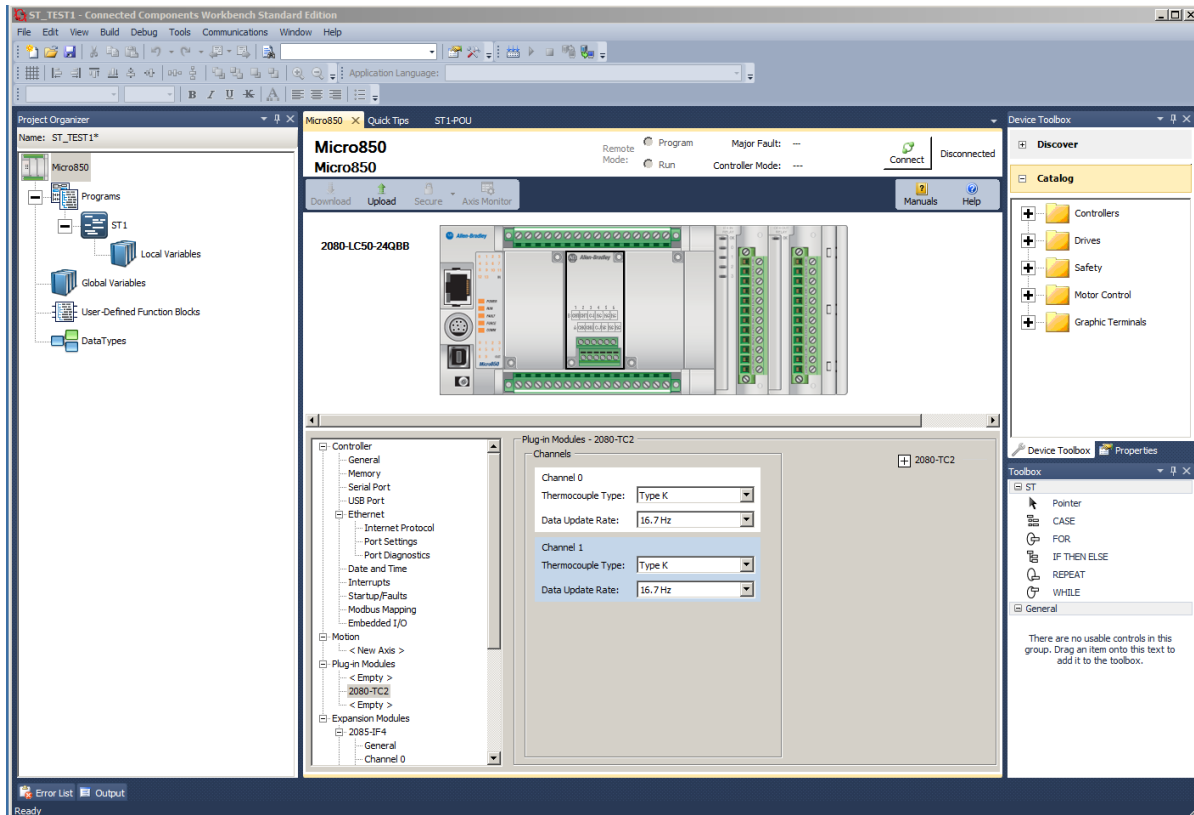


Figure 1. Configuration of 2080-TC2 Thermocouple Plug-in Module.

Programmers access the temperature data and input channel status information through global variables. These variables have the general form `_IO_Px_AI_yy`. The parameter `x` indicates the plug-in module location (0-3) and `yy` indicates the word offset in the PLC memory mapping. Table 8 gives the data mapping for the 2080-TC2 module.

The thermocouple module uses a firmware mapping of the raw data to temperature in degrees C to insure accuracy. Equation (3) shows the relationship between the raw data count and the temperature. Use equation (4) to find the temperature for a given data count.

$$D = 10 \cdot (T + 270.0) \quad (3)$$

Where: D = data count  
T = temperature in degrees C

$$T = \frac{(D - 2700)}{10} \quad (4)$$

Table-8 2080-TC2 Two Channel Thermocouple Data Mapping

| Word Offset | 15                         | 14  | 13       | 12 | 11  | 10  | 9   | 8  | 7        | 6  | 5  | 4  | 3        | 2 | 1 | 0 |
|-------------|----------------------------|-----|----------|----|-----|-----|-----|----|----------|----|----|----|----------|---|---|---|
| 00          | Channel 0 Temperature Data |     |          |    |     |     |     |    |          |    |    |    |          |   |   |   |
| 01          | Channel 1 Temperature Data |     |          |    |     |     |     |    |          |    |    |    |          |   |   |   |
| 02          | Channel 0 Information      |     |          |    |     |     |     |    |          |    |    |    |          |   |   |   |
|             | UKT                        | UKR | Reserved |    |     |     |     | OR | UR       | OC | DI | CC | Reserved |   |   |   |
| 03          | Channel 1 Information      |     |          |    |     |     |     |    |          |    |    |    |          |   |   |   |
|             | UKT                        | UKR | Reserved |    |     |     |     | OR | UR       | OC | DI | CC | Reserved |   |   |   |
| 04          | System Information         |     |          |    |     |     |     |    |          |    |    |    |          |   |   |   |
|             | Reserved                   |     |          |    | SOR | SUR | COC | CE | Reserved |    |    |    |          |   |   |   |

#### Table Legend

UKT = Unknown sensor type.

UKR = Unknown sensor update rate.

OR = Over-range on input channel. Maximum temperature count for selected sensor type

UR = Under-range on input channel. Minimum temperature count for selected sensor type

OC = Open circuit sensor

DI = Data Illegal. Data field in illegal and cannot be used by user.

CC = Code Calibrated. Set bit indicates temperature data calibrated by system calibration coefficient.

SOR = System Over-range. Set bit indicates over-range error with environment temperature above 70 C.

SUR = System Under-range. Set bit indicates under-range error with environment temperature below -20 C.

COC = CJC open. Set bit indicates that CJC sensor is not connected.

CE = Calibration error. Set bit indicates module is not accurate. Default value is 0.

## PLC Programs Utilizing Analog I/O

Programmable logic controllers with analog I/O can handle a wide range of on/off control applications and even implement digitized proportional-integral-derivative (PID) control in slow-speed applications. The global variables defined in the previous sections give programmers access to the data values. A programmer can use the global variables in programs to make control decisions or compute analog outputs based on input values. Any supported PLC language can utilize the analog I/O simply by including the global variables in the program structure. The Micro800 series PLCs have a number of arithmetic functions that can perform computations using the analog inputs. These include logarithms trigonometric and inverse trigonometric functions. Compare instructions such as greater/less than and equal/not equal can convert analog



inputs into Boolean outputs for control of digital output points. Process control instructions take analog inputs and compute derivatives and integrals and scale input values. The Micro800 series programming languages support an advanced PID controller function that takes analog inputs and produces analog outputs to drive final control elements of process control loops. The Micro800 Programmable Controllers General Instructions reference manual (Rockwell Publication No. 2080-RM001B-ENE) give the details of all these instructions and shows examples of their usage for ladder, function block and structured text programs. Access the course website or the learning management system for copies of this document.

Figure 2 shows comparator blocks in a ladder diagram program used to test an analog input value. The ladder diagram comparator blocks have an enable Boolean inputs that activates the block. They are wired to a constant TRUE from the left-hand rung in this example. They will test their inputs on every processor scan and update the outputs based on the analog input on pin i1 (`_IO_X1_AI_00`) and the constants on pin i2. The range of the analog data depends on the range type selected in the module setup. The constants on pins i2 must represent scaled values of the

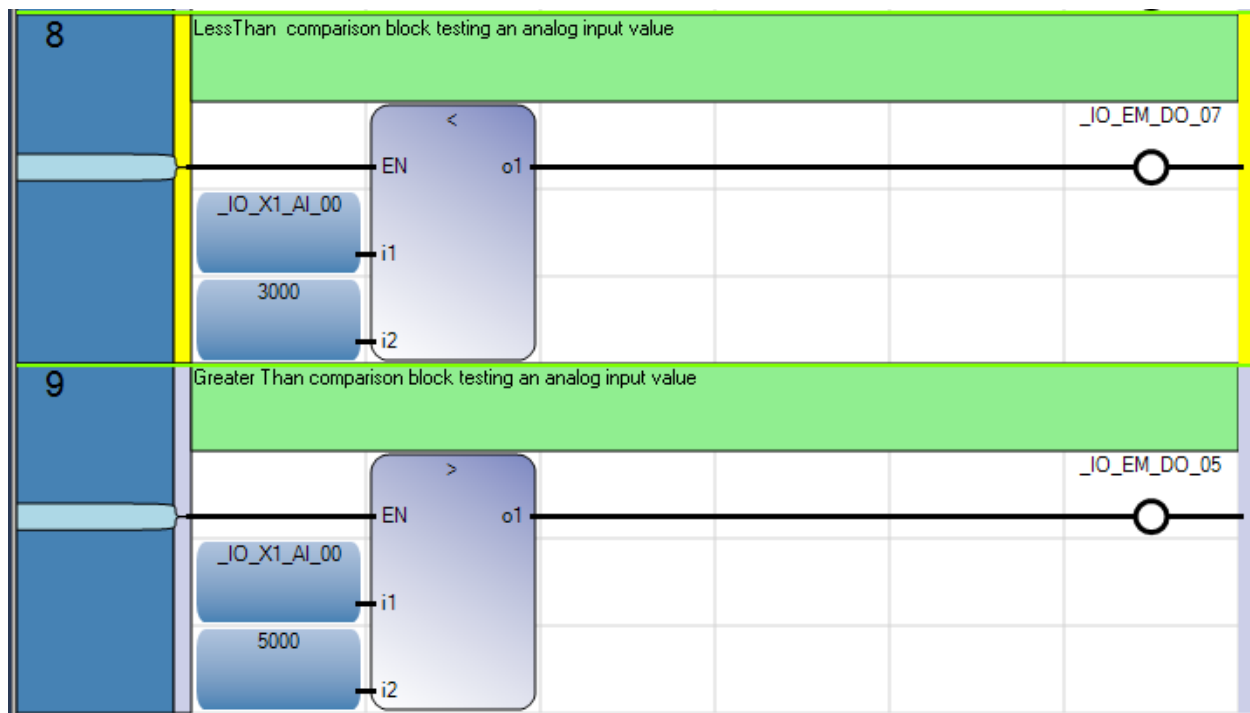


Figure 2. Ladder Diagram Comparators Testing Analog Input .

actual analog input values. The less than comparator's output in rung 8 controls the digital I/O point `_IO_EM_DO_07` while the greater than block controls point `_IO_EM_DO_05`.

Figure 3 shows the same code sample written as a Functional Block Diagram (FBD) program. Notice that the comparator blocks do not have enable input in this programming implementation. The analog input variable `_IO_X1_AI_00` feeds values to both comparator blocks. This program is functionally equivalent to the one shown in Figure 2. Figure 4 shows the initial program

written as structured text (ST). This produces the most concise program, but required detailed knowledge of programming languages and structures.

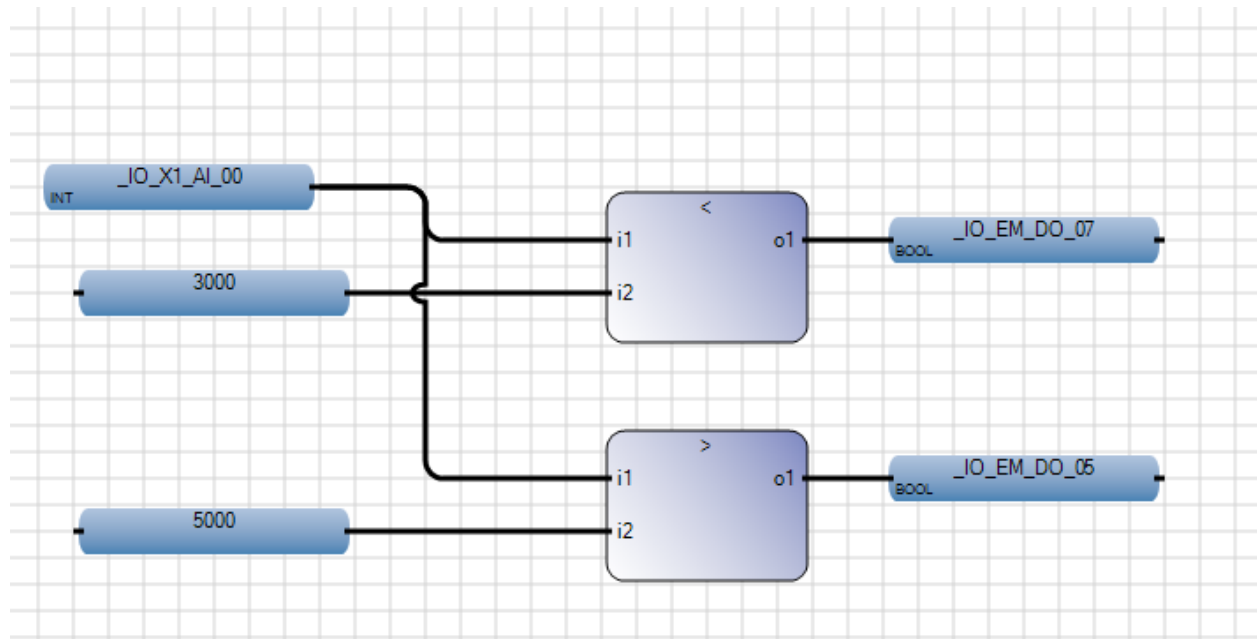


Figure 3. FD Language Representation of Analog Comparator Tests.

```
(* Structured test program to read analog inputs *)
_IO_EM_DO_07:=(_IO_X1_AI_00 < 3000);  (*less than test *)
_IO_EM_DO_05:=(_IO_X1_AI_00 > 5000);  (*greater than test *)
```

Figure 4. ST Language Representation of Analog Comparator Tests.

The trainer has a variable voltage source connected to the 2085-IF4 analog input module channel 0. The potentiometer on the trainer front panel controls the voltage input to the analog channel. There are three other analog input channels, but they require an external voltage or current. The 2085-OF4 analog output module channel A0 connects to a 0-10 Vdc panel meter located on the trainer. Lab projects can use this meter to monitor the analog output in programming projects. The module has three other unused analog output channels available.

## Analog Input Programming Project

- 1.) Create a bar graph display on the trainer using the analog input channel AI0 of the 2085-IF4 module. This channel connects to the potentiometer on the trainer front panel. The input range is 0-10 Vdc. The program should light digital outputs D0, D2, D4 and D6 as the input voltage increases from 0-10 Vdc with D0 representing the lowest range and D6 the highest range. The input voltage should be divided into equally spaced trip point levels over the analog input range. Write the program using Ladder Diagram instructions. Check the input channel configuration before programming to verify the input range.
- 2.) Recode the program in part 1 using Function Block diagram instructions.
- 3.) Recode the program in part 1 using Structured Text instructions.

## Analog Output Programming Project

- 1.) Program the PLC analog output module to simulate the operation of a three bit DAC with an output range of 0-10 Vdc. Use channel AO0 of the 2085-OF4 that connects to the dc panel meter to display the DAC output. Use the digital input selector switches DI5, DI6, and DI7 to simulate the 3-bit binary input. Make DI5 the least significant bit on the inputs. Light the digital outputs DO0, DO2, DO4 to indicate the positions of the selector switches. Write the program using any one of the supported PLC languages.
- 2.) Test the program by completing the table below. Save the results for future reference.

**Simulated DAC Program Test Results**

| DI7     | DI6     | DI5     | Output Voltage (Vdc) |
|---------|---------|---------|----------------------|
| OFF (0) | OFF (0) | OFF (0) |                      |
| OFF (0) | OFF (0) | ON (1)  |                      |
| OFF (0) | ON (1)  | OFF (0) |                      |
| OFF (0) | ON (1)  | ON (1)  |                      |
| ON (1)  | OFF (0) | OFF (0) |                      |
| ON (1)  | OFF (0) | ON (1)  |                      |
| ON (1)  | ON (1)  | OFF (0) |                      |
| ON (1)  | ON (1)  | ON (1)  |                      |

## **Lab 9 Assessment**

Complete and submit the following items for grading and perform the listed actions to complete this laboratory assignment.

- 1.) Complete the online quiz over Lab 9 technical background.
- 2.) Code the analog input and output programs into the PLC trainer
- 3.) Demonstrate working programs to the lab TA
- 4.) Write short descriptions (one page double-spaced or less) describing the operation of the program
- 5.) Submit pdf files of the working programs and the appended descriptions to the Lab 9 dropbox.