WAKE TECH
Building, Engineering, & Skilled Technologies

Lab 1: Using IO and Data Types

BAT-212: Logic and Programming

This material is based upon work supported by the National Science Foundation Advanced Technical Education grant program, A New Technician Training Program for Advanced Building Technologies, DUE-2000190.

The opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Lab 1: Using IO and Data Types

Objectives

Upon completion of this activity the student will be able to:

1. Build a functioning wire sheet in Sedona using basic input/output and data type conversion Sedona components.
2. Measure voltage ranges of inputs and outputs. Compare to values on controller webpage.
3. Use Sedona blocks to convert between data types.
4. Write a clear, concise, complete lab report.

Parts and Equipment

- Sedona Application Editor from Contemporary Controls
- BASemulator
- BAS22/22Scontroller
- Controller

References

- Contemporary Controls product page: https://www.ccontrols.com/basautomation/bascontrolfirmware.htm
- BAScontrol22 User Manual (BAS22 UM): https://www.ccontrols.com/pdf/um/UM-BASC22V4.pdf
- Sedona Open Control Reference Manual https://www.ccontrols.com/pdf/RM-SEDONA00.pdf
- Wikipedia article on fixed point numbers: https://en.wikipedia.org/wiki/Fixed-point_arithmetic
- Wikipedia article on floating point numbers: https://en.wikipedia.org/wiki/Single-precision_floating-point_format

Background

In this lab you will experiment with different types of inputs and outputs (I/O) used on the BAScontrol22/22S as well as different data types used in the wiresheets.

I/O Types

We will be experimenting with sensor and control I/O only; the networking connections will be discussed elsewhere. The controller user manual lists 3 areas of consideration for I/O, as discussed below.

The first area is the electrical specifications for the I/O circuits on the controller board.

1. Universal Input: There are 8 UI channels. There are five possible configurations for the UI channels. Refer to Table 2.2.
2. Binary Input: There are 4 binary input only channels. Refer to Table 2.3.
3. Analog Output: There are 4 analog output channels. Refer to Table 2.
4. Binary Output: There are 6 binary output channels. Refer to Table 2.5.

The second area is the channel configurations for the I/O to the field devices. The input devices used for this lab are two switches, an analog sensor, and a thermistor. The output devices are three LEDs and an actuator. The channel configurations are set via the controller's webpage. Access and navigation of the webpage are described in Chap 5 of the user manual. Sec 5.4 describes the channel configuration.

1. Setting the channel type for the UI channels is necessary. The other channels have only one type.
2. It is good practice to set default values for the outputs.
3. We do not need to give information for the BACnet object at this stage, but it can be helpful to you in debugging your system to give an object name, units, and other information depending on the object type.

The third area is the electrical connections to the field devices. A circuit of some kind is needed to provide the electrical signal. Determine the type of circuit needed based on I/O type and device type. This is discussed in Chap 4 of the manual.

1. Universal Input: refer to section 4.2 in BAS22 UM for detailed information.
2. Binary Input: refer to section 4.3 in BAS22 UM for detailed information.
3. Analog Output: refer to section 4.4 in BAS22 UM for detailed information.
4. Binary Output: refer to section 4.5 in BAS22 UM for detailed information.

Data Types

Data inside the program can be converted from one type to another. Some blocks only accept one data type, which means that sometimes data will need to be converted. We will be using three data types: Boolean, Float, and Integer.

1. Boolean: The Boolean data type has two possible values, either *true* or *false*. This data type is used for Binary Inputs and Binary Outputs, as well as intermediate values used within the wiresheet.
2. Float: A float value uses a decimal point, therefore can represent a fractional value. This data type is used with Universal Inputs for channel types analog voltage, thermistor, and resistor as well as Analog Output. The float data type uses a 32-bit floating point format. This implies a sign bit, 8 bits for an exponent, and 24 bits for the significant bits (leading bit assumed to be 1, therefore 23 bits stored). This provides 6 to 9 significant decimal digits. The optional link above provides a reasonable explanation of the format if you are interested.
3. Integer: The integer data type is not directly generated by the physical I/O channels but is used to simplify calculations (integer math requires fewer processor cycles than

floating point math) or for display. One use of integers is to truncate floating point values and in provided automatically in the UI Sedona block. Another use for integers is for fixed point calculations. Fixed point refers to the practice of multiplying a float value by a fixed number with the understanding that the lower place digits are actually fractional values. For instance, we may wish to use a temperature measurement and include two places to the right of the decimal point. We would then multiply all our temperatures by 100 and truncate values to the right of the decimal, with the understanding that 7898 should be interpreted as 78.98 deg. This allows for more efficient storage and calculations using the data. Refer to the referenced Wiki article for a fuller explanation (optional).

Refer to the Sedona reference manual 3.1 – 3.5 for explanation of the data types and the blocks used for conversion between the blocks.

PROCEDURE

Testing Physical I/O

Step 1:

You will connect field devices in the enclosures and VAV boxes to the controller. Table 1 below lists the different I/O types to be tested. Select appropriate pins on the BAS controller for each point listed. Available devices should have been discussed be your instructor. Select an appropriate device for connection to each point in the table. In and out in the table is defined relative to the BAS controller. Fill out the Connection Point and Field Device columns in Table 1 below.

Step 2:

Set the IP address for your computer so that you can connect to the BAS controller. The default address of the controller is 192.168.92.68. Open the Command Prompt. Use ipconfig/all to verify the change of the computer's address. Use ping to verify the controller's address. If it doesn't work, use Angry IP scanner (app is available on the computer) to find the controller's address, and use that address to ping the controller. Ask your instructor if you cannot find the controller's address. **Include a screenshot of the command window with the computer IP address and the controller address highlighted in your lab report.**

Step 3:

Open the web page of the controller by typing the controller's IP address in the search bar. Configure the UI terminals via the webpage. To do this, click on the title above the channel box. Set the channel type and give a name identifying the field device connected to the channel. Change the names for the other I/O channels used also. **Include a screenshot of the webpage with completed changes in your lab report.**

Step 4:

Use the Sedona Application Editor to create a wiresheet with appropriate IO blocks for each point. Wire the controller connections according to Table 1. Note that the circuit for the binary output requires that you wire a power source in series with the relay. In Figure 1 below, the binary contacts are the poles on either side of the switch. 24 volts DC is available in the enclosure.

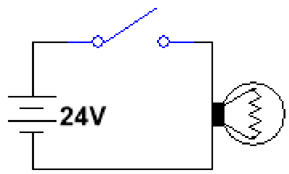

Figure 1: Circuit for binary output

Step 5:

Measure 2 different voltages across each connected I/O on the controller. Record multimeter measurements across the two connectors for each controller connection used. For Binary IO, include measurements for open and close. For the thermistor and for Analog IO, include a low and a high mid-range measurement. Include the corresponding webpage values for each point. Note you will use the webpage to force the output values. Are any of your measurements unexpected? **Include an explanation of the meaning of your measurements related to the web page values in your lab report.**

Step 6:

Include the table, an image of the wiresheet, and good photos of the wiring on the controller in your lab report.

Field Device	Connection point BAS Controller	Point Description	DMM Voltage Measurement A	B	Webpage value A	B
		Binary in				
		Binary out				
		Analog in				
		Analog out				
		Thermistor				
		UI Binary				

Table 1: Points list for field devices to controller I/O types

Data Type conversion:

In this section you will experiment with data types. Start the BAS emulator. Open the Sedona editor and connect to the emulator. Place a UI block for each of the following types of input: binary, analog, and thermistor. Use the web interface to force input values to test the block responses. **Describe** the block for each type of input for your lab report, that is, for UI Analog, UI binary and UI Therm 10KT2. What are legal input values for the different data types? Include your answers in a concise paragraph in your lab report.

The block **B2F** implements binary-to-float conversion. The testing layout is shown in Figure 2. Input block B0 is used to provide 0's to the individual bits, and input block B1 is used to provide 1's to selected bits. The particular setting shown has 0's going to bits 1 thru 11, and 1's going to bits 12 thru 16, resulting in a float type output of 63488.0. You should change the connections to test different values. For instance, if you change the input of bit 1 from 0 to 1, how does the output change? What is the range of float numbers that can result from the conversion? To test this, change all the inputs to 0, record the output, and **include a screen shot of the wiresheet in your lab report**. Then change them all to 1 and record the output. Which input bit is the least significant bit (LSB) and which is the most significant bit (MSB)? What does the count value in the block represent? **Include your answers in a concise paragraph in your lab report.**

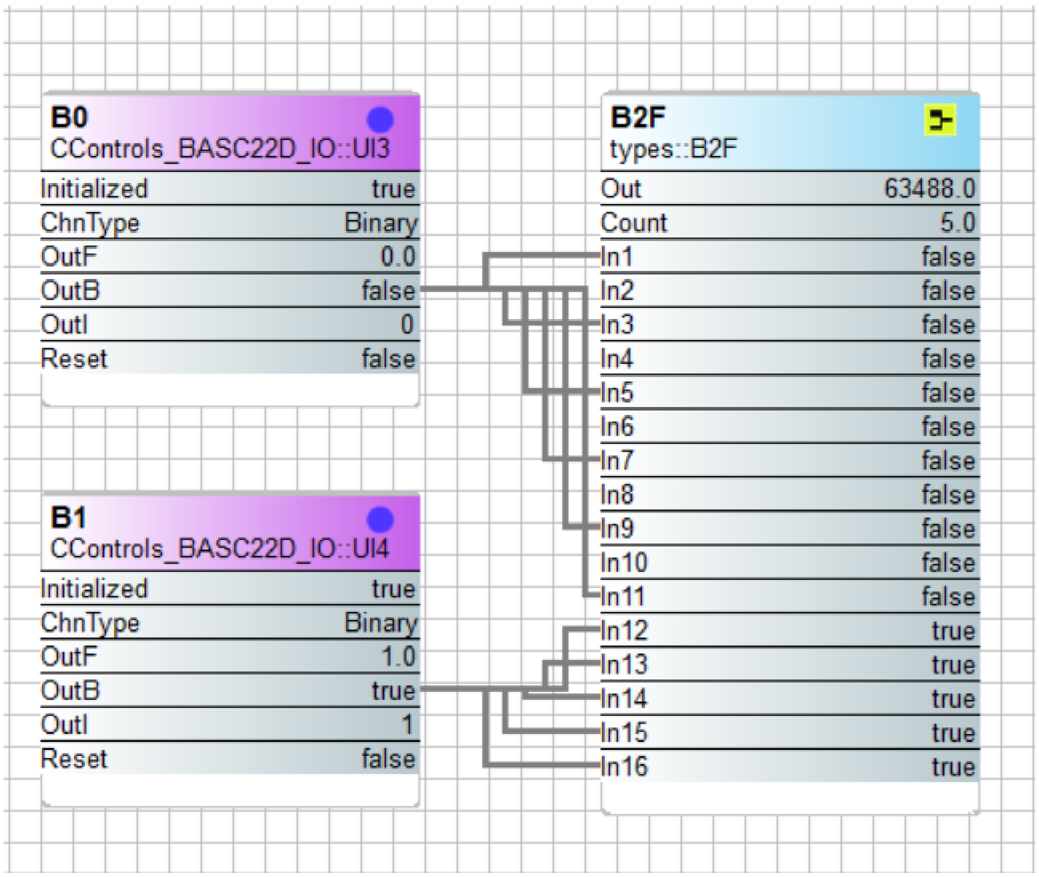

Figure 2: B2F test layout

The block **F2B** implements float-to-binary conversion. If you connect the **F2B** to an analog UI block, the valid range of values is 0 to 10 volts. How many bits in the F2B are affected by the analog input over the range of valid values? Thinking in terms of fixed floating point, how might you make use of the full range of binary values in representing the analog signal? **Include your answers in a concise paragraph in your lab report.**

QUESTIONS

1. What is the difference between a UI used as binary input versus the dedicated binary input? Read about these inputs in the manual to give a complete answer. When would you use one versus the other?
2. What does it mean for a device to sink current?