

Lab: 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: 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, and 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
- BAS22controller
- Trainer board

REFERENCES

- Contemporary Controls product page: <https://www.ccontrols.com/basautomation/bascontrolfirmware.htm>
- BAScontrol22 User Manual: <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#:~:text=In%20computing%2C%20fixed%2Dpoint%20is,1%2F100%20of%20dollar

BACKGROUND

In this lab you will experiment with different types of inputs and outputs (I/O) used on the BAScontrol22. 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. There are five possible configurations for the universal inputs. There are also 4 binary input only channels. There are analog output and binary output channels. Read the details of the electrical specifications in Chap 2 of the user manual.

The second area is the channel configurations for the I/O to the field devices. The input devices provided on the trainer board are two switches, an analog sensor, and a thermistor. The output devices are two 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. Setting the channel type for the UI is necessary. It is good practice to set default values for the outputs. 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.

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. The integer data type is not 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 for fixed point calculations. Fixed point refers to the practice of multiplying a real value by a fixed number with the understanding that the lower place values 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 of 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. Refer to the Sedona reference manual 2.9 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 found on the trainer 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. Identify the type of I/O appropriate for the field devices on the trainer. Select a device for connection to each point in the table. In and out in the table is defined relative to the BAS controller.

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. Try to open the web page of the controller using this address. If it doesn't work, use Angry IP to find the controller's address, and open the webpage. Configure the UI terminals via the webpage.

Step 3:

Use the Sedona Application Editor to create a wiresheet with appropriate IO blocks for each point. Wire the controller connections according to the table below. 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 is available on the trainer.

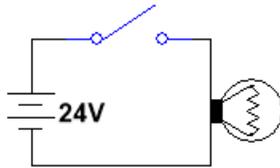


Figure 1: Circuit for binary output

Step 4:

Measure 2 different voltages across each connected I/O on the controller. For Binary IO, include measurements open and close. For the thermistor and for Analog IO, include a low and a high mid-range measurement. Record multimeter measurements across the two connectors for each controller connection used. 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 physical meaning of your measurements in your lab report.

Step 5:

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

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

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

Data Type conversion:

In this section you will experiment with data types. Start the BAS emulator. Open the Sedona editor and place a UI block for each type 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, UI Therm 10KT2, and BI. What are legal input values for the different data types? Are any of the input types natively integer, that

is, are the received values whole numbers? Include your answers in a concise paragraph in your lab report.

The block **B2F** implements binary-to-float conversion. The testing layout 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 and record the output, 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.

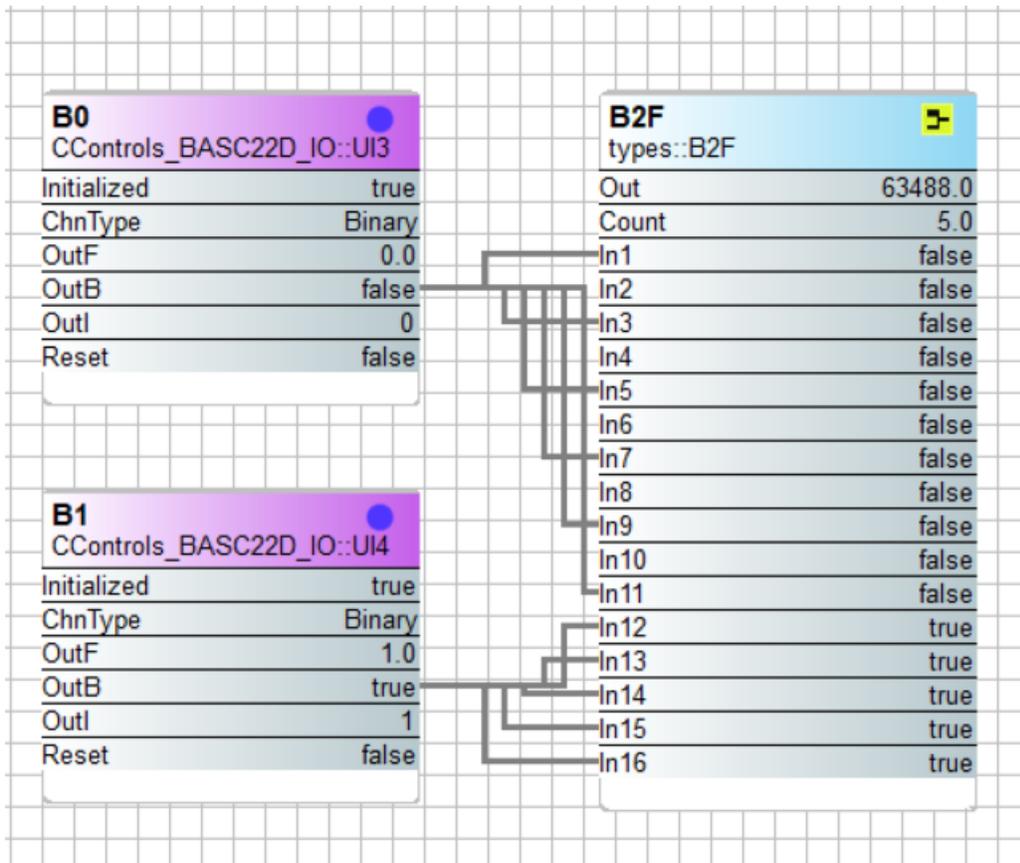


Figure 2: B2F test layout

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? When would you use one versus the other?
2. What does it mean for a device to sink current?