

Map a Temperature Input to a Current Output



Overview

Use the programming language ScriptBasic on the DXM Controller to map inputs to outputs of different scales. The ScriptBasic program (NullSpanV2.sb) and associated XML configuration file (NullSpanV2.xml) show an example of how this is accomplished.

When converting between different input/output types, define these four key parameters:

Null

On the input side, Null defines the starting point.

In this example, the Null value is $-20\text{ }^{\circ}\text{C}$ (the starting of the output range).

Span

Defines the total range of the input.

In this example, Span is 80. The temperature range starts at $-20\text{ }^{\circ}\text{C}$ and therefore ends at $+60\text{ }^{\circ}\text{C}$.

Output Range mA

Defines the total output range.

In this example the output is a current (mA) output with a total range of $20000\text{ }\mu\text{A}$, or 20 mA. For a 4–20 mA output, the range is $16000\text{ }\mu\text{A}$, or 16 mA.

Output Range mA Offset

If the output range starts at zero, the offset is zero.

For example, 0–20mA. If the range is 4–20mA, the output offset is 4 mA.

For our mapping example, the input temperature range $-20\text{ }^{\circ}\text{C}$ to $+60\text{ }^{\circ}\text{C}$ is mapped to a 0–20 mA output. In the ScriptBasic program, the four variables that define the input to output mapping are:

```
!*****  
!*** User Variables ***  
!*****  
iNull1 = -40  
iSpan1 = 80  
Range_mAOffset1 = 0  
Range_mA1 = 20000
```

The DXM Configuration Tool defines Local Register 1 as the temperature input (Temp Input1). The DXM Configuration Tool's **Local Registers > Local Registers in Use** screen shows the simple settings for the registers. This configuration is useful to read the registers on the DXM Controller LCD screen.

#	Register Name	Register Group	Units	Signed	Constant or Timer	Cloud Permissions	LCD Permissions	Protocol Conversion	Log Files	Read Rules
1	InputValue1		C	<input checked="" type="checkbox"/>	None	None	Read	None	None	Modbus RTU Read Temp
2	InputValue2		C	<input checked="" type="checkbox"/>	None	None	Read	None	None	
11	OutputValue1		mA	<input type="checkbox"/>	None	None	Read	None	None	
12	OutputValue2		mA	<input type="checkbox"/>	None	None	Read	None	None	

A Read Rule reads a temperature/relative humidity sensor and places the temperature reading (in degrees C) into Local Register 1. The rule reads the temperature value from Modbus Slave 8 register 2 and writes the data into Local Register 1.

Read Rules | Write Rules | J1939 | Modbus TCP

Add Read Rule | Delete Last Rule

Read Temp | From slave ID 8 | read | 1 registers starting at | 2 through 2 | to local registers starting at | 1 through 1

Read Settings

Remote type: Holding register

Frequency: 00:00:01.000 hh:mm:ss.fff

Scaling: Scale value 0.000000 | Scale offset 0

Apply offset before scale value

Error condition: Apply value 0 after 0 read failures

Floating point: Swap words

On register: 0

Local Registers Names

InputValue1

Load a Configuration File or ScriptBasic Program

1. Before you can load a configuration file or ScriptBasic program, connect the DXM Controller a PC running the DXM Configuration Tool.
2. Select the port from the **Device > Connection Settings** menu.
3. To upload a ScriptBasic program:
 - a) Go to the **Settings > Scripting** screen.
 - b) Click **Upload file**.
 - c) Select the file to upload to the DXM Controller and click **Ok**.
 - d) Set the program to run at start-up time by selecting the file name and clicking **Add Selected to Startup**.
4. To upload a configuration XML file, go to the **Device > Send Configuration to Device** menu.
The configuration file currently loaded into the DXM Configuration Tool uploads to the DXM Controller.

ScriptBasic Program to Map a Temperature Input to an Analog Output

If necessary, you can set the Null and Span parameters and the output Range parameters within the program. The rest of the ScriptBasic program is ready to run. The input values of temperature are expected to be 16-bit signed numbers. Twos complement format is used to define signed numbers.

The function `ConvertTemp1` reads Local Register 1, checks to see if it is negative, checks to see if the input number is out of range of the defined Null and Span parameters, then converts the number.

The basic idea is to find where the input is within the input range (percentage) and then apply that same percentage to the output.

The input Local Register is read into the program by the `GETREG` function. Since ScriptBasic doesn't know if the number is positive or negative, it must look at the upper bit in the 16-bit number. If the number is negative, the program performs a twos complement on the number then make it negative.

Because numbers are stored multiplied by 20, the program divides by 20 to get the actual temperature value.

The input ratio is calculated and then checked for boundary conditions. From this input ratio, the output is calculated. The various values are printed out to the console output. The `SETREG` function writes the calculated outputs to the I/O board analog output and to a Local Register so the user can see the value on the LCD screen.

```
FUNCTION ConvertTemp1
  'Read from Local Registers for testing.
  InputReading1 = 0xFFFF AND (GETREG(Input1_reg, LocalReg, HoldingReg))
  '
  ' 2's complement if it's negative.
  IF (0x8000 AND InputReading1) THEN
    InputReading1 = (0xFFFF XOR InputReading1) +1
    InputScaled1 = (InputReading1 / TempMult) *-1
  ELSE
    InputScaled1 = (InputReading1 / TempMult)
  END IF
  '
  InputRatio1 = (InputScaled1 - iNull1) / iSpan1
  ' Check for values outside the range looking at the ratio
  IF InputRatio1 < 0 THEN
    InputRatio1 = 0
  ELSE IF InputRatio1 > 1 THEN
    InputRatio1 = 1
  END IF
  '
  'Output scale
  Output_mA1 = (Range_mA1 * InputRatio1) + Range_mAOffset1
  '
  PRINT "InputReading1: ", InputReading1, " InputScaled1: ", InputScaled1, " Input ratio1: ", InputRatio1, " -
  Output mA1: ", Output_mA1, "\n"
  ' Write I/O board Analog output;
  WrErr = SETREG(Output1_reg, Output_mA1, IOBoardSID, HoldingReg)
  WrErr = SETREG(LR_Out1_reg, Output_mA1, LocalReg, HoldingReg)
END FUNCTION
```

There are two conversion functions in the ScriptBasic program that provide mapping for both analog outputs of the DXM Controller.

```
'Main loop
WHILE(1)
  'Read input1 Local Register and convert to mA output range
  ConvertTemp1
  'Read input2 Local Register and convert to mA output range
  ConvertTemp2
  SLEEP(1)
WEND
```

To view the ScriptBasic console output, go to the DXM Configuration Tool's **Device > View Serial Activity** menu.