

## Application Note—AN0410A

---

### Measuring outside air temperature

This application note describes several methods for determining outside air temperature with values from multiple sensors or averaging several readings from one sensor.

---

Outside air temperature is used for many purposes such as setting airflow for an economizer, calculating boiler water temperature, or enabling chillers.

To improve accuracy or to provide redundancy the temperature readings from two or more sensors may be compared or averaged to determine an optimal value. In addition, averaging several readings will also provide some filtering of fluctuations due to local sensor variation.

#### Installing outside sensors

The examples described in this application note do not require a particular sensor. When installing sensors, keep the following items in mind.

- ◆ Mount and connect the sensors as described in the installation guides provided with the sensors and the controllers.
- ◆ Sensors mounted on the north side of a building and out of direct sunlight will be the most accurate.
- ◆ The sensors need not be connected to the same controller. However, the example programs will have to be modified to reference the correct point or object for sensors not physically connected to the controller in which the Control Basic program is running.
- ◆ All sensors must be configured for the same unit of measure, either degrees Fahrenheit or degrees Celsius.

The KMC Controls model STE-1451 is suitable for the applications in the example Control Basic programs.



*Illustration 1 STE-1451 outside air temperature sensor*

## **Choosing between two sensors**

The simplest method to determine outside air temperature with multiple sensors is to use the Control Basic MIN function. This method assumes that the lower temperature value will be the correct temperature.

### **Example for KMD controllers**

The following Control Basic example is shown with WinControl XL Plus descriptions.

```
10 OutsideAirTemp = MIN( TempSensor1 , TempSensor2 )
```

**TempSensor1 and TempSensor2** These points may be either physical inputs or variables.

**OutsideAirTemp** This variable holds the results of the calculation. The value stored in this variable is typically used by other controllers on the network as the single global value for outside air temperature.

### **Control Basic examples for BACnet controllers**

The following Control Basic example is entered with mnemonics for the BACnet objects in KMD controllers.

```
10 AV10 = MIN( AI1 , AI2 )
```

**Inputs AI1 and AI2** The input objects may be either physical controller analog inputs or an analog value object that represents a temperature input.

**Value object AV10** This value object stores the results of the calculation. The present value of this object is typically used by other controllers on the network as the single global value for outside air temperature.

## Averaging values from multiple sensors

Averaging the values of multiple sensor will provide some filtering of fluctuations caused by local sensor variation. The following example averages the values of two sensors and then calculates a value that can be shared with the rest of the network devices. It also tests each sensor for a valid response. If the temperature reading is out of range for a sensor, the sensor is flagged and its reading is ignored.

### Example for KMD controllers

The following Control Basic example is shown with WinControl XL Plus labels.

*Example:*

```
10 OAT = AVG( TMPSNS1 , TMPSNS2 )
20 IF TMPSNS1 > 140 OR TMPSNS1 < -40 THEN START A ELSE STOP A
30 IF TMPSNS2 > 140 OR TMPSNS2 < -40 THEN START B ELSE STOP B
40 IF A THEN OAT = TMPSNS2
50 IF B THEN OAT = TMPSNS1
60 DALARM A , 20 , OUTSIDE SENSOR FAILURE ON Sensor 1
70 DALARM B , 20 , OUTSIDE SENSOR FAILURE ON Sensor 2
```

**TMPSNS1 and TMPSNS2** These points may be either physical inputs or variables.

**OAT** This variable holds the results of the calculation. The value stored in this variable is typically used by other controllers on the network as the single source for outside air temperature.

### Control Basic examples for BACnet controllers

The following Control Basic example is entered with mnemonics for the BACnet objects in KMD controllers.

**Analog inputs AI1 and AI2** The input objects may be either physical controller analog inputs or an analog value object that represents a temperature input.

**Value object AV21** This value object holds the results of the calculation. The value stored in this object is typically used by other controllers on the network as the single source for outside air temperature.

*Example:*

```
10 AV21 = AVG( AI1 , AI2 )
20 IF AI1 > 140 OR AI1 < -40 THEN START A
30 IF AI2 > 140 OR AI2 < -40 THEN START B
40 IF A THEN AV21 = AI2
50 IF B THEN AV21 = AI1
```

## Calculating a running average for an outside temperature

To minimize the effects of temperature fluctuations, use a running average value of outside air temperature. The following program measures the outside temperature at 10-minute intervals, stores the readings in local variables and then calculates the average temperature for the last hour.

### Example for KMD controllers

The following Control Basic example is entered with WinControl XL Plus descriptions.

```
10 IF INTERVAL( 0:10:00 ) THEN A = B , B = C , C = D , D = E , E = F , F = TempSensor1
20 IF POWER-LOSS OR A = 0 THEN A = TempSensor1 , B = TempSensor1 , C = TempSensor1 , D = TempSensor1 , E = TempSensor1 , F = TempSensor1
30 OATAverage = AVG( A , B , C , D , E , F )
100 END
```

**TempSensor1** Outside air temperature

**OATAverage** The average outside temperature over the past 60 minutes. This value is a stable temperature reading well suited for use in temperature resets.

**Local variables** The local variables (A, B, C....) are held only in memory by the controller. After a power loss or upon sending the program, the values of these variables become 0. Line 20 keeps these values to a realistic average for the initial hour after the program is sent and in the event of power failure in the controller.

### Control Basic examples for BACnet controllers

The following Control Basic example is entered with mnemonics for the BACnet objects in KMD controllers.

```
10 IF INTERVAL( 0:10:00 ) THEN A = B , B = C , C = D , D = E , E = F , F = AI1
20 IF POWERLOSS OR A = 0 THEN A = AI1 , B = AI1 , C = AI1 , D = AI1 , E = AI1 , F = AI1
30 AV23 = AVG( A , B , C , D , E , F )
100 END
```

**Input AI1** This object may be either a physical input or value object that represents the outside air temperature.

**Value object AV23** The average outside temperature over the past 60 minutes. This value is a stable temperature reading well suited for use in temperature resets.

**Local variables** The local variables (A, B, C....) are held only in memory of the controller. After a power loss or upon sending the program, the values of these variables become 0. Line 20 keeps these values to a realistic average for the initial hour after the program is sent and in the event of power failure in the controller.