# Decision Making

CASE STRUCTURE

When creating a program, you may want to execute (or bypass) code based on an input. The input may be user input (for example, pushing a front panel button) or input that is a result of the code in your program (for example, the reading of a sensor).

The Case Structure is a programming structure that has one or more subdiagrams, or cases, only one of which executes when the structure executes. The value wired to the selector terminal determines which case to execute. This structure is located on the Functions ► Structures palette. First, select the structure from the Structures palette, then use your cursor to drag a selection rectangle to size the structure.
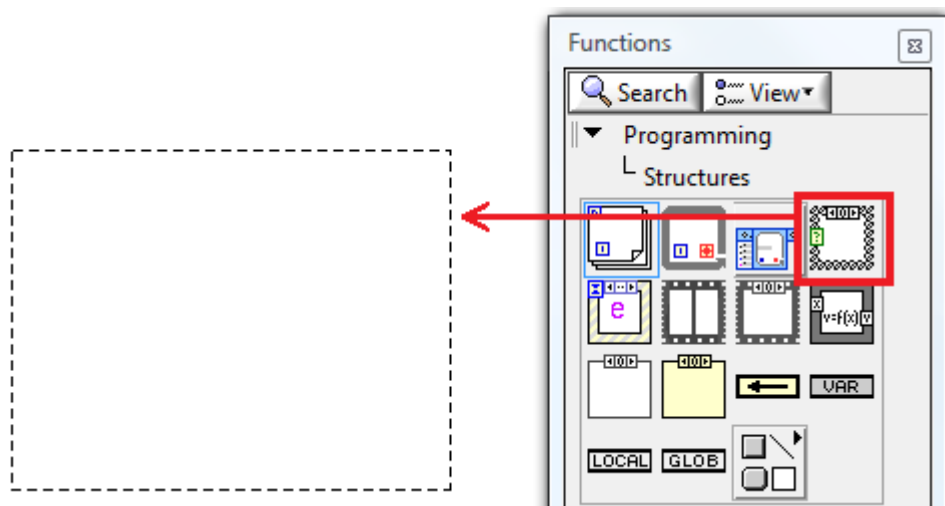

When you release the mouse button, the Case Structure boundary is set. The structure can be resized as necessary. Then add your code inside each subdiagram, or case, of the structure. The default is two cases, but there can be multiple cases. The hidden cases can be viewed by clicking the right or left arrows at the top of the case structure.
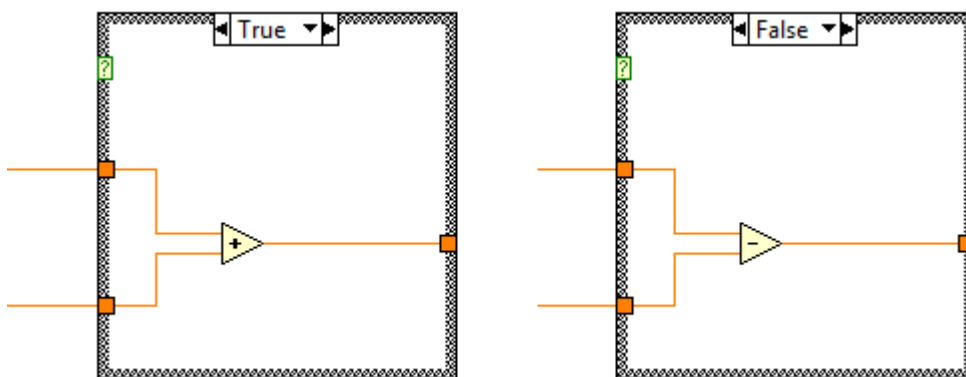
The diagram above is not complete because the case selector terminal has not been wired. The case selector terminal appears as a small question mark (?) on the left side of the case structure. The input value wired to the selector terminal determines which case to execute and can be

Boolean values, string, integer, or enumerated type (a control that provides a list of items). A case structure with a Boolean wired to its selector terminal has a maximum of two cases; all other data types allow two or more cases.
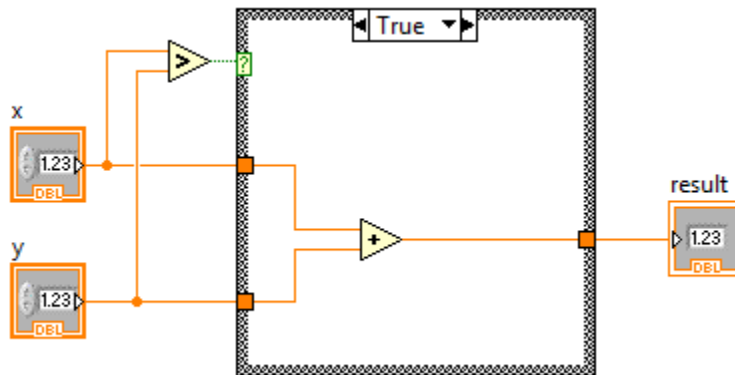
The figure above is now completely wired. The case selector input in this example is a Boolean value that is based on the result of a comparison of the two numeric values, x and y. If x is greater than y, the True case is executed and the two values are added. If x is not greater than y, the False case is executed (this case is not shown in the figure above), and the y numeric is subtracted from x.

It is important to note that the Case Structure can also accommodate lists and ranges of values. For lists, use commas to separate values. For numeric ranges, specify a range as 10..20, meaning all numbers from 10 to 20 inclusively. You can also use open-ended ranges. For example, ..100 represents all numbers less than or equal to 100, and 100.. represents all numbers greater than or equal to 100. For string ranges, a range of a..c includes all strings beginning with a or b, but not c. A range of a..c,c includes the ending value of c.

Placing a Case Structure within a Case Structure (i.e., creating a nested Case Structure) increases the decision making power. For example, you may need to use two numeric values to determine what message to send to the user, but you only want to send the message if the user has pressed a button.
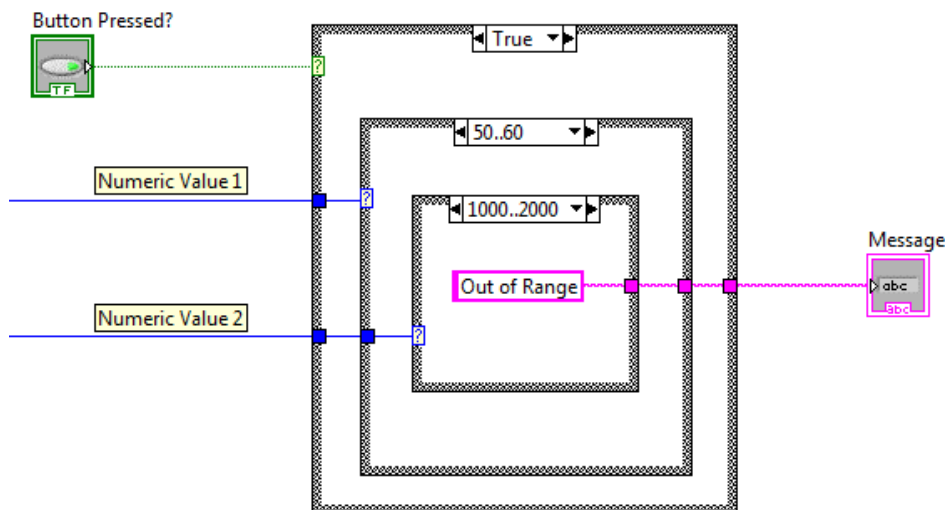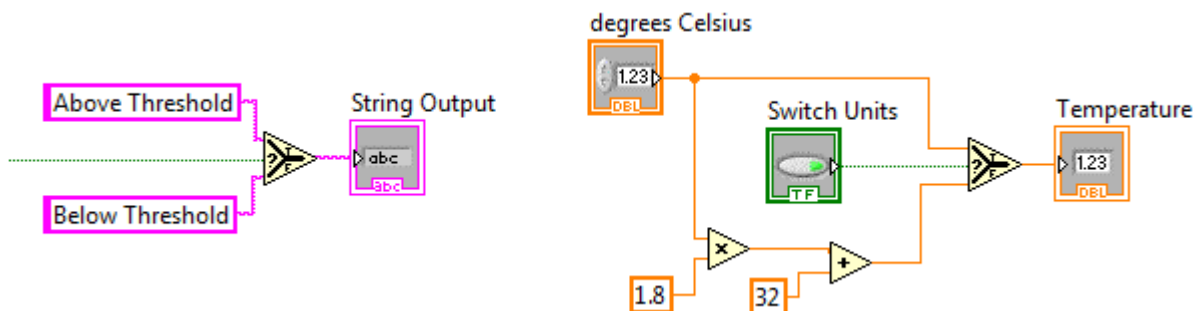
SELECT

The Select function is found in the Functions ► Programming ► Comparison palette.
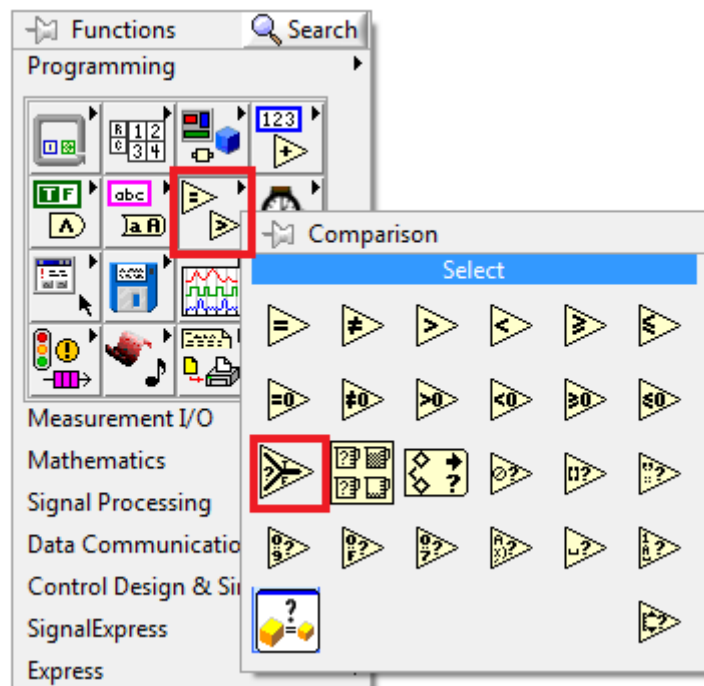
This function returns one of two values, depending on the value of the conditional input. If the conditional input is TRUE, this function returns the value wired to the "t" terminal. If it is FALSE, this function returns the value wired to the "f" terminal. In other words, your program selects one of two values.

The figure above shows two examples of using the Select function. In the example on the left, the Boolean value input determines what string is passed to the String Output. If the Boolean value is True, the String Output indicator will display the "Above Threshold" string. If it is False, it will display the "Below Threshold" string. In the example on the right, the Boolean control called "Switch Units" determines if the value is displayed in Celsius or Fahrenheit units.