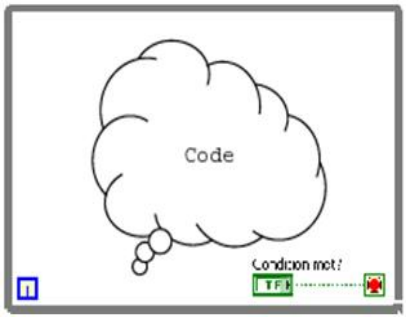
Chapter
4

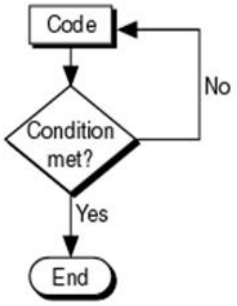
Working with Loops

LOOPS

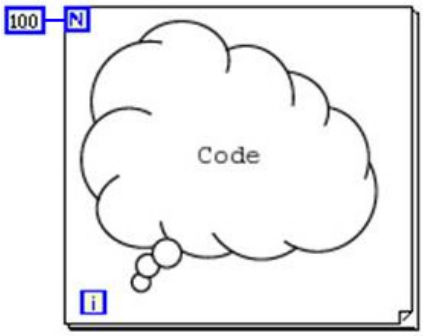
Two common structures used in LabVIEW programming are the While Loop and the For Loop. Both While and For Loops are located on the Functions ► Structures palette.

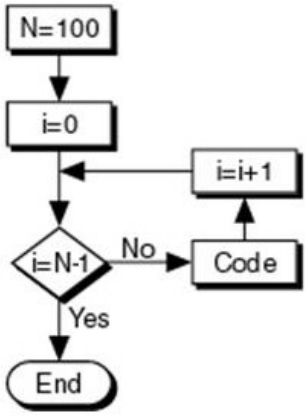
A While Loop is a control flow statement used to execute a block of LabVIEW code repeatedly until a given Boolean condition is met. First, you execute the code, and then the conditional terminal is evaluated. A While Loop does not have a set iteration count; thus, a While Loop executes indefinitely if the condition never occurs.





A For Loop is a control flow statement used to execute code a set number of times. The value in the count terminal (an input terminal represented by the N), indicates how many times to repeat the code.

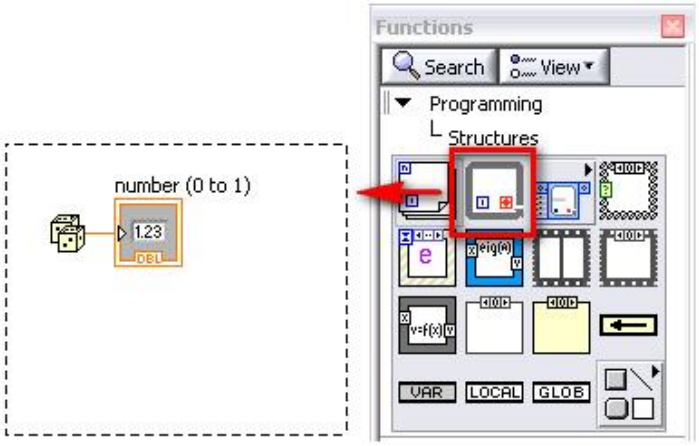




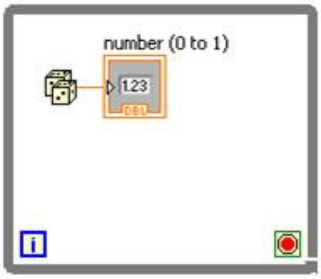
Both the While Loop and For Loop have an iteration terminal, i. The iteration terminal is an output terminal that contains the number of completed iterations. The iteration count always starts at zero. Therefore, the first time the code executes, the iteration terminal returns 0; the second time, the code executes the iteration terminal returns 1. If you repeat the code 100 times, the iteration terminal would end with a value of 99.

To place a loop on the block diagram, you will first select the loop from the Structures palette, and then use the cursor to drag a selection rectangle around (lasso) the section of the block diagram you want to repeat. For example, select the While Loop structure, and use the cursor to

drag a selection rectangle around code that generates a random number. The Random Number (0–1) function is found in the Numeric palette.

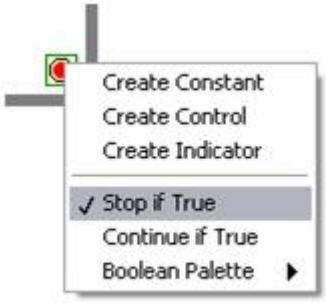


When you release the mouse button, a While Loop boundary encloses the section you have selected. The loop can be resized as necessary and additional code can be dropped inside the loop.

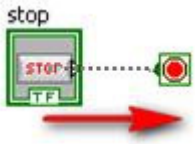


The diagram above is not complete because the While Loop's conditional terminal must be wired. The conditional terminal defines when the loop stops. There are two settings for the conditional terminal: Stop if True, , and Continue if True, . When set to Stop if True, the While Loop runs until a Boolean value of true is sent to the conditional terminal. If the conditional terminal is configured to Continue if True, the While Loop runs only if a Boolean value of true is sent to the terminal.

To switch the conditional terminal between Continue if True and Stop if True, right-click on the conditional terminal, and check the corresponding setting. In most cases, it is best to leave the conditional terminal in the default configuration of Stop if True. This keeps the While Loop logic consistent.



Wire a Boolean control (such as a STOP button) to the conditional terminal so that you can control the execution of the While Loop. An easy way to add a STOP button to a While Loop is to right-click on the conditional terminal and select Create ► Control from the shortcut menu. When the STOP button is pressed, a true value is passed to the conditional terminal causing the While Loop to stop execution. You can wire any Boolean data to the conditional terminal to control the execution of a While Loop.



When working with hardware, it may be critical to stop the hardware prior to stopping the execution of the While Loop. For example, if there is code within the loop to continually collect data, and the loop is terminated without properly stopping the data collection process first, the hardware will not be properly shut down. The proper method would be to stop the hardware, and then terminate the execution of the While Loop as displayed in the example code below.