

Microcontroller and DSP System

Various Applications Using Arduino Microcontroller....

Third Year, 1st Semester

Lecture No.7

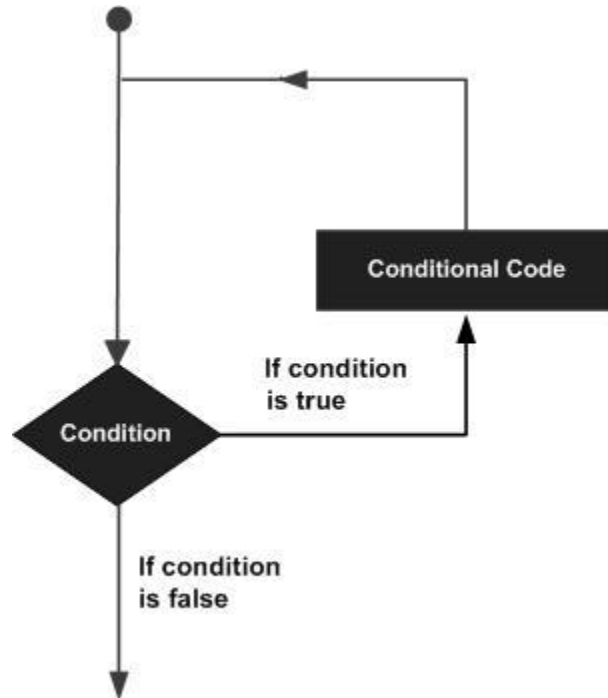
Ass. Lecturer: Yousif Allbadi
M.Sc. of Communications Engineering
yousifallbadi@uodiyala.edu.iq

University of Diyala
College of Engineering
Department of Communications Engineering
2020-2021

Arduino – Control Statements

Decision making structures require that the programmer specify one or more conditions to be evaluated or tested by the program. It should be along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Following is the general form of a typical decision-making structure found in most of the programming languages.



Control Statements are elements in Source Code that control the flow of program execution. They are:

- If statement
- If ...else statement
- If...else if ...else statement
- switch case statement
- Conditional Operator?

if statement

It takes an expression in parenthesis and a statement or block of statements. If the expression is true then the statement or block of statements gets executed otherwise these statements are skipped.

If ...else statement

An **if** statement can be followed by an optional else statement, which executes when the expression is false.

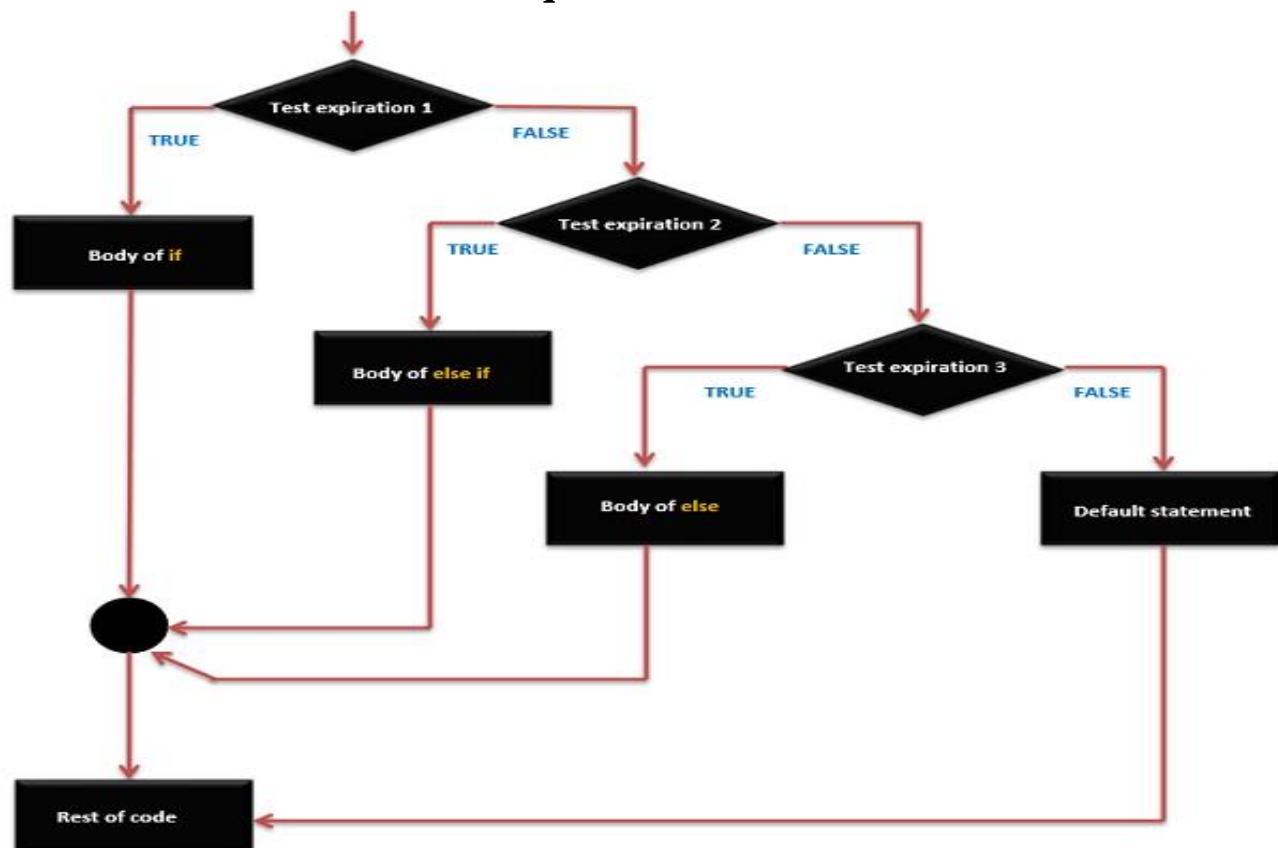
if...else if ...else statement

The **if** statement can be followed by an optional **else if...else** statement, which is very useful to test various conditions using single if...else if statement.

When using **if...else if...else** statements, keep in mind

- An **if** can have zero or one else statement and it must come after any else if' s.
- An **if** can have zero to many else if statements and they must come before the else.
- Once an **else if** succeeds, none of the remaining else if or else statements will be tested.

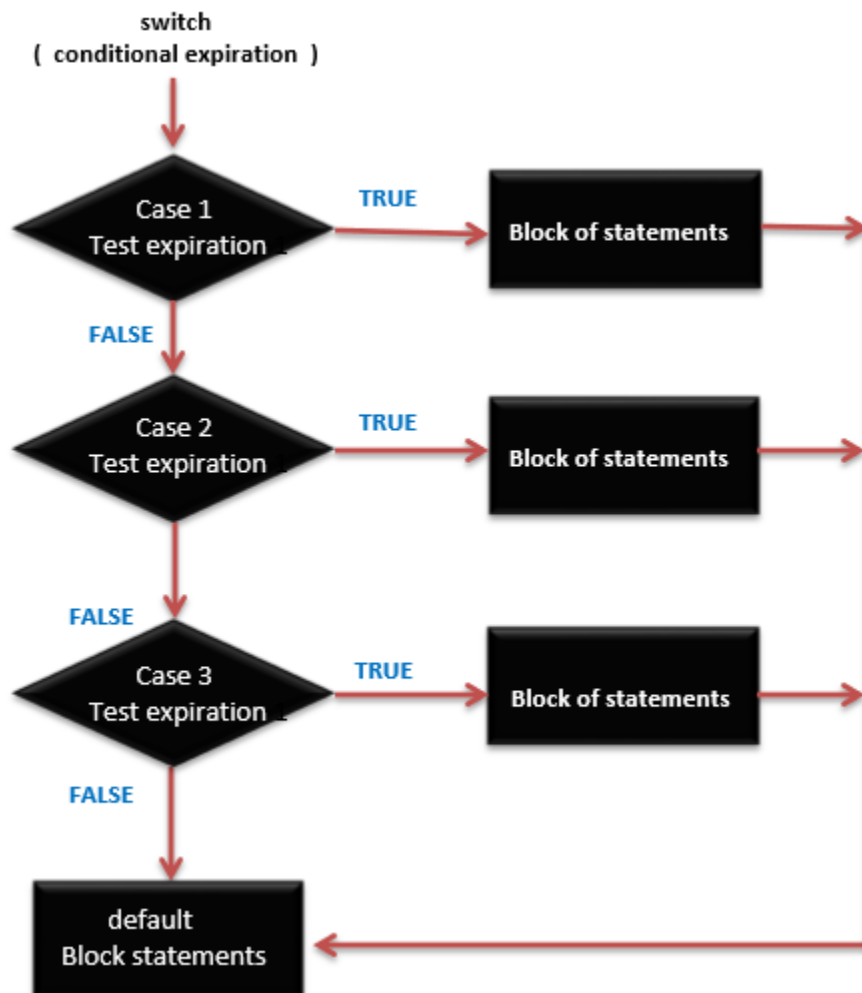
if...else Statement – Execution Sequence



Switch Case Statement

Similar to the if statements, **switch...case** controls the flow of programs by allowing the programmers to specify different codes that should be executed in various conditions. In particular, a **switch** statement compares the value of a variable to the values specified in the **case** statements. When a case statement is found whose value matches that of the variable, the code in that case statement is run.

The **break** keyword makes the switch statement exit, and is typically used at the end of each case. Without a break statement, the switch statement will continue executing the following expressions ("falling-through") until a break, or the end of the switch statement is reached.



Rules of Conditional Operator

- expression1 must be a scalar expression; expression2 and expression3 must obey one of the following rules:
- Both expressions have to be of arithmetic type.
- expression2 and expression3 are subjected to usual arithmetic conversions, which determines the resulting type.

Both expressions have to be of void type. The resulting type is void.

Arduino – Loops

Programming languages provide various control structures that allow for more complicated execution paths.

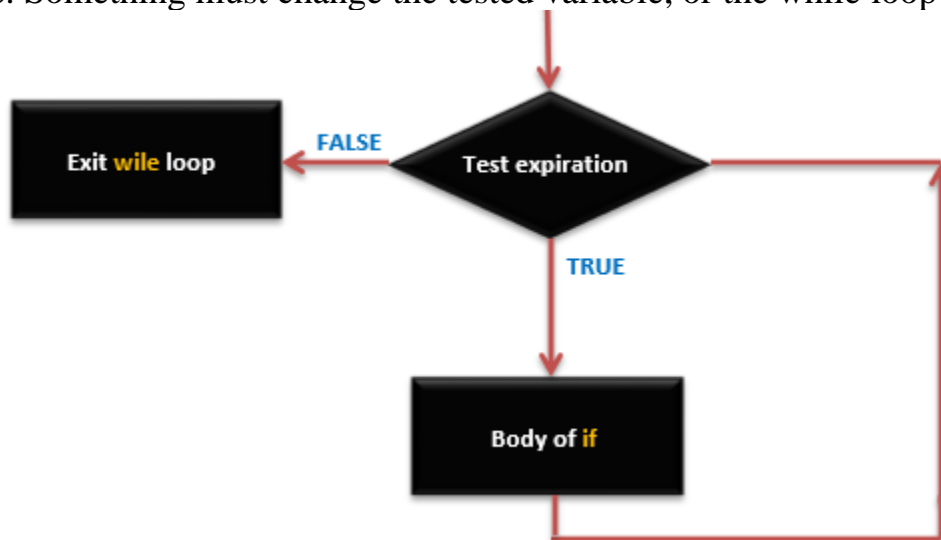
A loop statement allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages

C programming language provides the following types of loops to handle looping requirements.

- while loop
- do...while loop
- for loop
- nested loop
- infinite loop

while loop

while loops will loop continuously, and infinitely, until the expression inside the parenthesis, () becomes false. Something must change the tested variable, or the while loop will never exit.



do...while loop

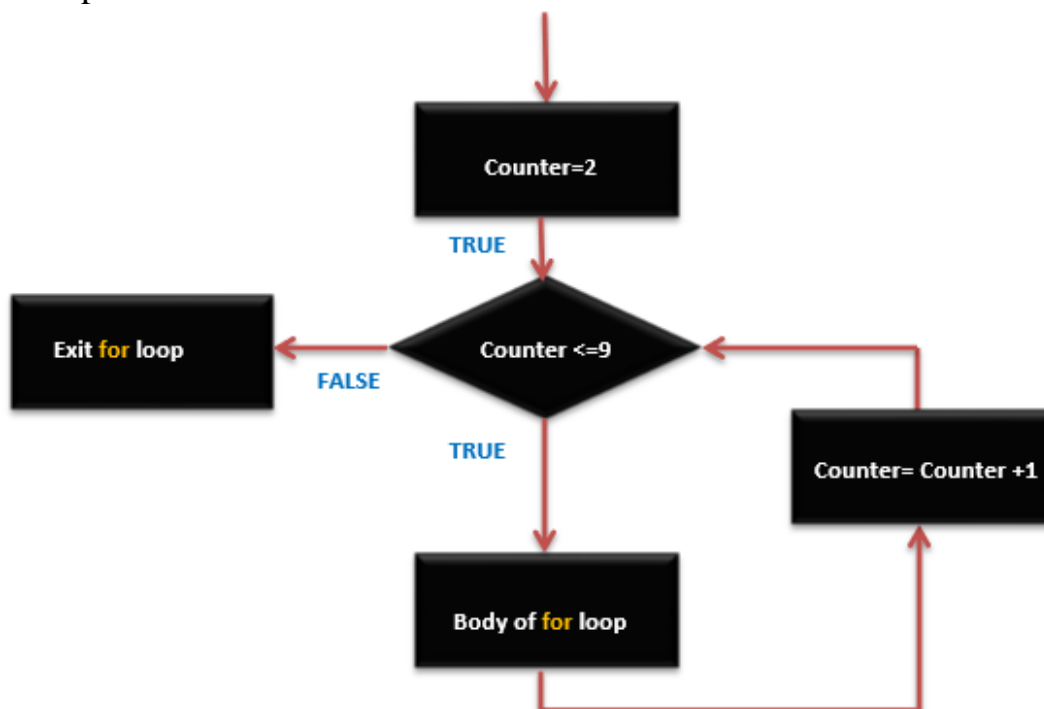
The **do...while** loop is similar to the while loop. In the while loop, the loop-continuation condition is tested at the beginning of the loop before performed the body of the loop. The do...while statement tests the loop-continuation condition after performed the loop body. Therefore, the loop body will be executed at least once.

When a **do...while** terminates, execution continues with the statement after the while clause. It is not necessary to use braces in the do...while statement if there is only one statement in the body. However, the braces are usually included to avoid confusion between the while and do...while statements.

for loop

A **for loop** executes statements a predetermined number of times. The control expression for the loop is initialized, tested and manipulated entirely within the for-loop parentheses. It is easy to debug the looping behavior of the structure as it is independent of the activity inside the loop.

Each **for** loop has up to three expressions, which determine its operation. The following example shows general for loop syntax. Notice that the three expressions in the for-loop argument parentheses are separated with semicolons.



Nested Loop

C language allows you to use one loop inside another loop.

Infinite loop

It is the loop having no terminating condition, so the loop becomes infinite.