



SNS COLLEGE OF ENGINEERING

(Autonomous)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



19OE204- IoT System Architecture

Arduino - Basics of Embedded C Programming for Arduino

**Prepared by,
K.Sangeetha
Assistant Professor/ECE
SNS College of Engineering**



Arduino - Program Structure

Arduino programs can be divided in three main parts: **Structure**, **Values** (variables and constants), and **Functions**.

Structure:

Software structure consist of two main functions –

Setup() function

Loop() function

Sketch – The first new terminology is the Arduino program called “sketch”.

The **setup()** function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

INPUT – -

OUTPUT – -

RETURN – -

After creating a **setup()** function, which initializes and sets the initial values, the **loop()** function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

INPUT – -

OUTPUT – -

RETURN – -



DATA TYPES

The following table provides all the data types that you will use during Arduino programming.

void	Boolean	char	Unsigned char	byte	int	Unsigned int	word
long	Unsigned long	short	float	double	array	String-char array	String-object



ARDUINO

void

The void keyword is used only in function declarations. It indicates that the function is expected

to return no information to the function from which it was called.

Example

```
Void Loop ( ) { // rest of the code }
```

Boolean

A Boolean holds one of two values, true or false. Each Boolean variable occupies one byte of memory.

Char

A data type that takes up one byte of memory that stores a character value. Character literals are written in single quotes like this: 'A' and for multiple characters, strings use double quotes: "ABC".



ARDUINO

unsigned char

Unsigned char is an unsigned data type that occupies one byte of memory. The unsigned char data type encodes numbers from 0 to 255.

byte

A byte stores an 8-bit unsigned number, from 0 to 255.

int

Integers are the primary data-type for number storage. int stores a 16-bit (2-byte) value.

The **int** size varies from board to board.

On the Arduino Due, for example, an **int** stores a 32-bit (4-byte) value.



ARDUINO

Unsigned int

Unsigned ints (unsigned integers) are the same as int in the way that they store a 2 byte value. Instead of storing negative numbers, however, they only store positive values.

Word

On the Uno and other ATMEGA based boards, a word stores a 16-bit unsigned number.

On the Due and Zero, it stores a 32-bit unsigned number.

Long

Long variables are extended size variables for number storage, and store 32 bits (4 bytes).



ARDUINO

unsigned long

Unsigned long variables are extended size variables for number storage and store 32 bits (4 bytes). Unlike standard longs, unsigned longs will not store negative numbers).

short

A short is a 16-bit data-type. On all Arduinos (ATMega and ARM based), a short stores a 16-bit (2-byte) value.

float

Data type for floating-point number is a number that has a decimal point. Floating-point numbers are often used to approximate the analog and continuous values.

double

On the Uno and other ATMEGA based boards, Double precision floating-point number occupies four bytes. That is, the double implementation is exactly the same as the float, with no gain in precision. On the Arduino Due, doubles have 8-byte (64 bit) precision



ARDUINO – VARIABLES AND CONSTANT

What is Variable Scope?

Variables in C programming language, which Arduino uses, have a property called scope. A scope is a region of the program and there are three places where variables can be declared. They are –

- Inside a function or a block, which is called local variables.
- In the definition of function parameters, which is called formal parameters.
- Outside of all functions, which is called global variables.



ARDUINO - OPERATORS

An operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C language is rich in built-in operators and provides the following types of operators –

Arithmetic Operators

Comparison Operators

Boolean Operators

Bitwise Operators

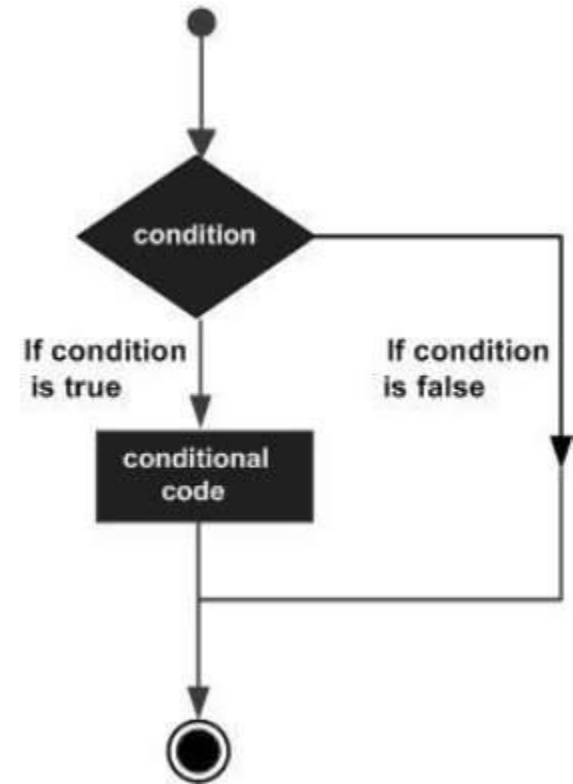
Compound Operators



ARDUINO – CONTROL STATEMENTS

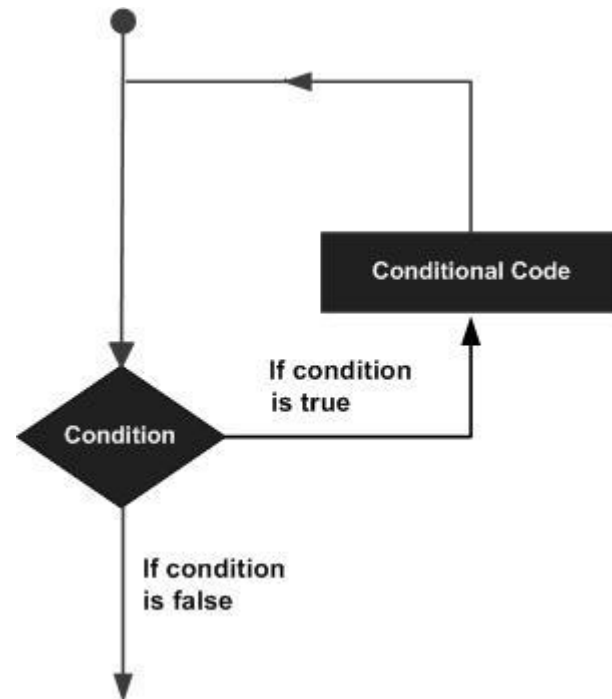
Decision making structures require that the programmer specify one or more conditions to be evaluated or tested by the program. It should be along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition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

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



ARDUINO - LOOP

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 –



ARDUINO - FUNCTIONS

Functions allow structuring the programs in segments of code to perform individual tasks. The typical case for creating a function is when one needs to perform the same action multiple times in a program.

Standardizing code fragments into functions has several advantages –

Functions help the programmer stay organized. Often this helps to conceptualize the program.

Functions codify one action in one place so that the function only has to be thought about and debugged once.

This also reduces chances for errors in modification, if the code needs to be changed.

Functions make the whole sketch smaller and more compact because sections of code are reused many times.

They make it easier to reuse code in other programs by making it modular, and using functions often makes the code more readable.



Thank you