3.8 Arduino Programming Structure

The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

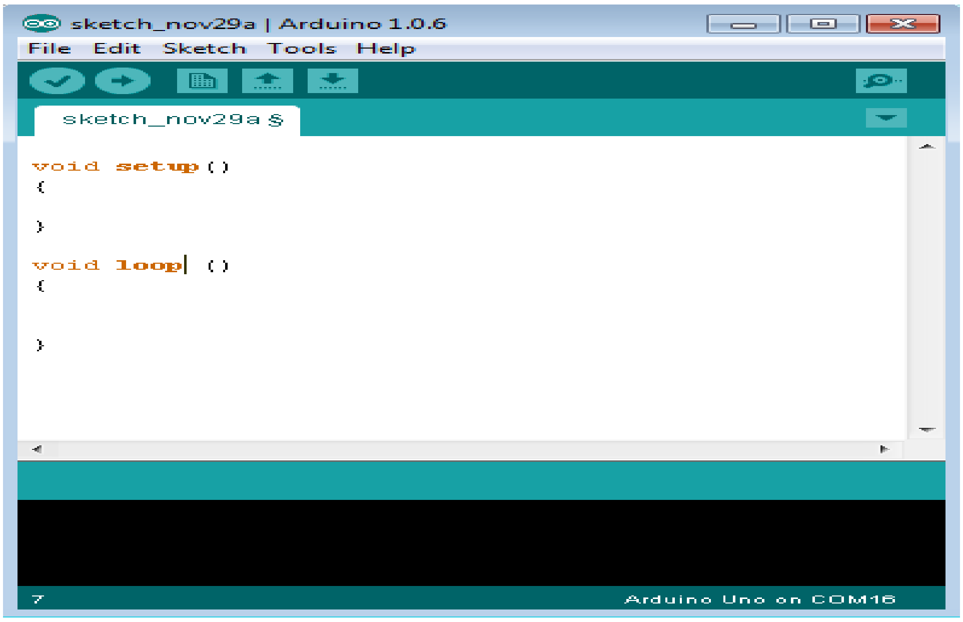
Sketch − The first new terminology is the Arduino program called “**sketch**”.

Structure

Arduino programs can be divided in three main parts: **Structure, Values** (variables and constants), and **Functions**. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the **Structure**. Software structure consist of two main functions −

- Setup( ) function
- Loop( ) function



```
void setup ()
{

}

void loop ()
{


}
```

Void setup ( ) {

}

- **PURPOSE** − 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** − -

Void Loop ( ) {

}

- **PURPOSE** − 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** − -

3.9 Sketches:

Start the Arduino IDE application. Open the Blink example sketch: Select **File → Examples → 01.Basics → Blink** on the top Arduino IDE menu.

Arduino Comments in Sketches

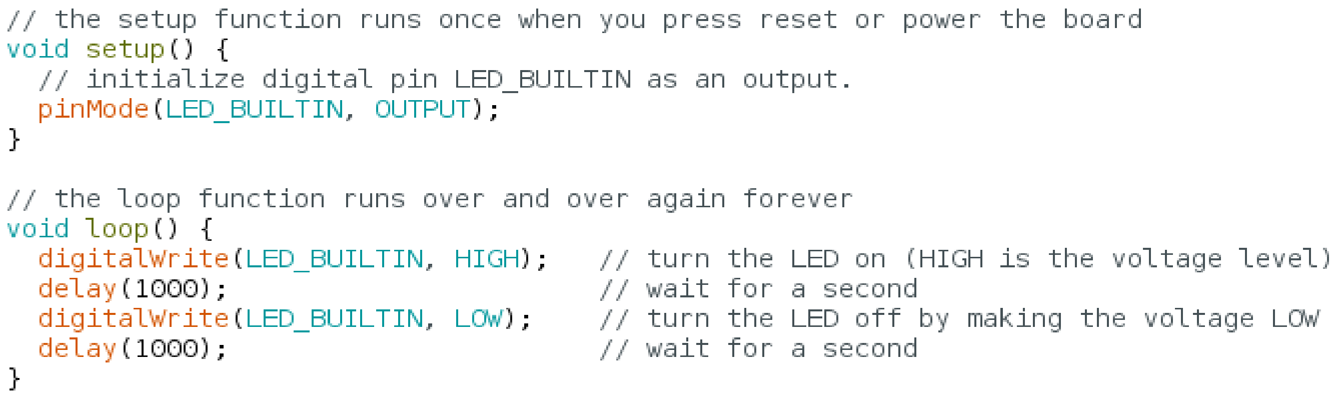
The Blink sketch contains a big comment block at the top. Scroll down to find the code that blinks the on-board LED. Comments are shown in a gray color between the /* and */ characters. Text typed between an opening forward slash and asterisk /* and closing asterisk forward slash */ is ignored by the software tools that run when the sketch is uploaded to an Arduino board. Place any comments or notes that you want between these characters. This type of comment can span multiple lines. The sketch name and description are contained in the top comment block of the Blink sketch. Detailed information about the on-board LED, and a history of the sketch follows.

A second type of comment starts with double forward slashes //. In this case, everything after the // is part of the comment. This is a single-line comment that turns everything after the // on the same line into a comment. Consequently the next line after the comment is not part of the comment.

Arduino Blink Sketch Code

The image below shows the Blink sketch code with the top comment block removed. As can be seen, it contains a mix of single-line comments and code. The code is everything in the image that is not a comment. Comments describe what is happening in the sketch.

When a sketch is loaded to an Arduino board, it is first built by software tools that automatically run. The build process consists of preprocess, compile and link stages. This converts the code from human readable text to something that runs on the Arduino board.



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

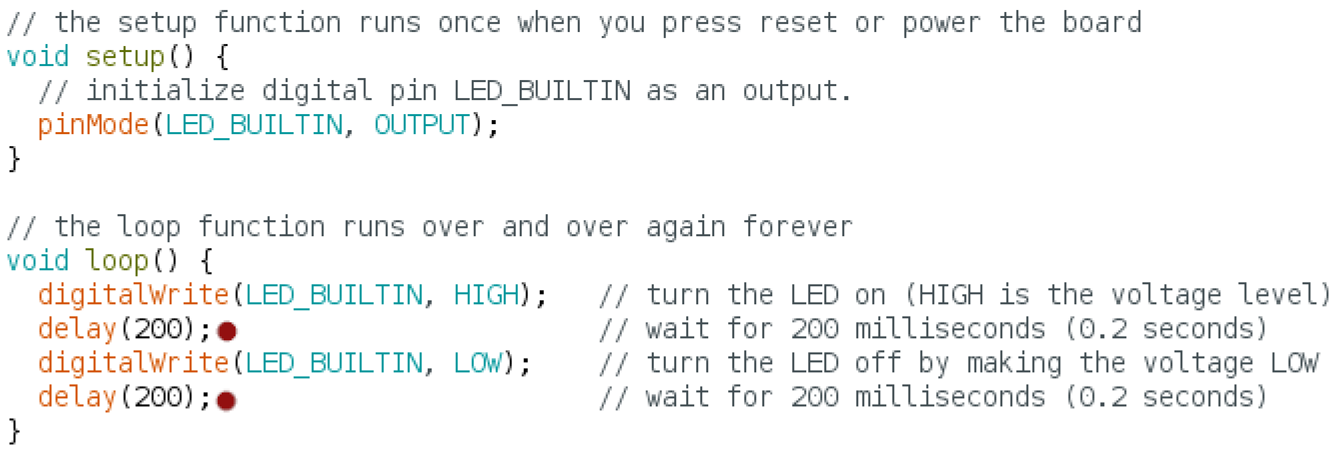
Blink Sketch Code is one of the Arduino Sketches for Beginners

Arduino Sketch Syntax Highlighting

Colored words in the sketch are a result of the Arduino IDE using syntax highlighting. Syntax highlighting is the highlighting of Arduino language keywords, definitions and functions. This is more easily understood after more is learned about writing Arduino sketches.

Modify the Arduino Blink Sketch

Change the rate that the on-board LED blinks at in the Blink sketch, as follows. Change 1000 to **200** in both instances it is found in the code. The following image shows the modified Arduino Blink sketch. Red dots in the image mark the modified lines of code. Notice that the comments at the right of each modified line of code are updated to reflect the changes made in the code.



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(200);●                       // wait for 200 milliseconds (0.2 seconds)
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(200);●                       // wait for 200 milliseconds (0.2 seconds)
}
```

Modified Blink Sketch Code

Click the Upload button in the Arduino IDE. This uploads the modified sketch to the target Arduino board. Notice that the on-board L LED blinks faster. As a result of the modified code, the LED is now on for 0.2 seconds and off for 0.2 seconds. The on and off times were originally 1000 milliseconds, also written 1000ms, which is one second (1s). This is because there are 1000ms in 1s. When 1000 is changed to 200 in the sketch, the LED on and off times change from 1000ms to 200ms. 200ms is 0.2s.

Save the Modified Sketch

Arduino example sketches are read-only. This means that they cannot be overwritten. When an example sketch is modified and saved, it must be saved to a new location. Three ways of saving a sketch in the Arduino IDE are firstly, click the Save toolbar icon (the arrow pointing down). Hover the mouse cursor over any toolbar icon and its name is shown at the right of the icons. Secondly, save the file using the keyboard shortcut **Ctrl + S** (hold down the Ctrl key and then press the s key). Thirdly, select **File → Save** on the top Arduino IDE menu bar. Because the file is read-only, the IDE prompts to save the file to a different location. Use the dialog box that opens to save the Blink sketch to your Arduino folder. Change the name to something like **Blink_Fast** befor
Start the Arduino IDE application. Open the Blink example sketch: Select **File → Examples → 01.Basics → Blink** on the top Arduino IDE menu.

Arduino Comments in Sketches

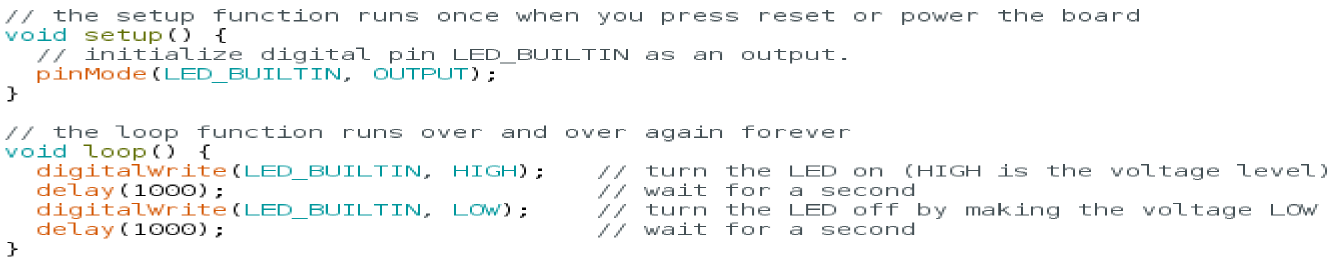
The Blink sketch contains a big comment block at the top. Scroll down to find the code that blinks the on-board LED. Comments are shown in a gray color between the /* and */ characters. Text typed between an opening forward slash and asterisk /* and closing asterisk forward slash */ is ignored by the software tools that run when the sketch is uploaded to an Arduino board. Place any comments or notes that you want between these characters. This type of comment can span multiple lines. The sketch name and description are contained in the top comment block of the Blink sketch. Detailed information about the on-board LED, and a history of the sketch follows. After that, find a link to the Arduino Blink sketch tutorial page.

A second type of comment starts with double forward slashes //. In this case, everything after the // is part of the comment. This is a single-line comment that turns everything after the // on the same line into a comment. Consequently the next line after the comment is not part of the comment.

Arduino Blink Sketch Code

The image below shows the Blink sketch code with the top comment block removed. As can be seen, it contains a mix of single-line comments and code. The code is everything in the image that is not a comment. Comments describe what is happening in the sketch.

When a sketch is loaded to an Arduino board, it is first built by software tools that automatically run. The build process consists of preprocess, compile and link stages. This converts the code from human readable text to something that runs on the Arduino board.



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

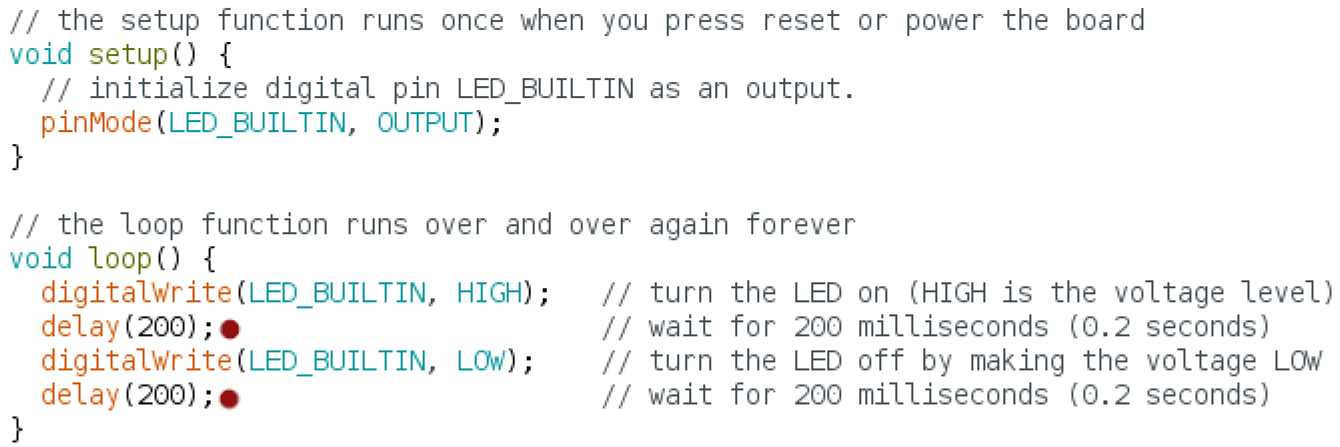
Blink Sketch Code is one of the Arduino Sketches for Beginners

Arduino Sketch Syntax Highlighting

Colored words in the sketch are a result of the Arduino IDE using syntax highlighting. Syntax highlighting is the highlighting of Arduino language keywords, definitions and functions. This is more easily understood after more is learned about writing Arduino sketches.

Modify the Arduino Blink Sketch

Change the rate that the on-board LED blinks at in the Blink sketch, as follows. Change 1000 to **200** in both instances it is found in the code. The following image shows the modified Arduino Blink sketch. Red dots in the image mark the modified lines of code. Notice that the comments at the right of each modified line of code are updated to reflect the changes made in the code.



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(200);●                       // wait for 200 milliseconds (0.2 seconds)
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(200);●                       // wait for 200 milliseconds (0.2 seconds)
}
```

Modified Blink Sketch Code

Click the Upload button in the Arduino IDE. This uploads the modified sketch to the target Arduino board. Notice that the on-board L LED blinks faster. As a result of the modified code, the LED is now on for 0.2 seconds and off for 0.2 seconds. The on and off times were originally 1000 milliseconds, also written 1000ms, which is one second (1s). This is because there are 1000ms in 1s. When 1000 is changed to 200 in the sketch, the LED on and off times change from 1000ms to 200ms. 200ms is 0.2s.

Save the Modified Sketch

Arduino example sketches are read-only. This means that they cannot be overwritten. When an example sketch is modified and saved, it must be saved to a new location.

- Three ways of saving a sketch in the Arduino IDE are firstly, click the Save toolbar icon (the arrow pointing down). Hover the mouse cursor over any toolbar icon and its name is shown at the right of the icons.
- Secondly, save the file using the keyboard shortcut **Ctrl + S** (hold down the Ctrl key and then press the s key). Thirdly, select **File → Save** on the top Arduino IDE menu bar. Because the file is read-only, the IDE prompts to save the file to a different location.
- Use the dialog box that opens to save the Blink sketch to your Arduino folder. Change the name to something like **Blink_Fast** before for saving.