**CHAPTER-5**

# **SOFTWARE REQUIREMENT**

Arduino IDE (Integrated Development Environment) is required to program the NODE MCU board.

## **5.1 PROGRAMMING ARDUINO**

Once Arduino IDE is installed on the computer, connect the board with computer using USB cable. Now open the Arduino IDE and

choose the correct board by selecting Tools>Boards>Arduino/Genuine Uno, and choose the correct Port by selecting Tools>Port. NODE MCU is programmed using Arduino programming language based on Wiring. To get it started with NODE MCU board and blink the built-in LED, load the example code by selecting Files>Examples>Basics>Blink. Once the example code (also shown below) is loaded into your IDE, click on the ‘upload’ button given on the top bar. Once the upload is finished, you should see the Arduino’s built-in LED blinking. Below is the example code for blinking:

## **5.2 ARDUINO – INSTALLATION**

After learning about the main parts of the NODE MCU board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board. In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB

cable.

## Step 1: About Arduino board

First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use NODE MCU, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.



Figure.5.1 USB Cable

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image



Figure.5.2 A TO MINI-B CABLE

## Step 2: Download Arduino IDE Software

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software,which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

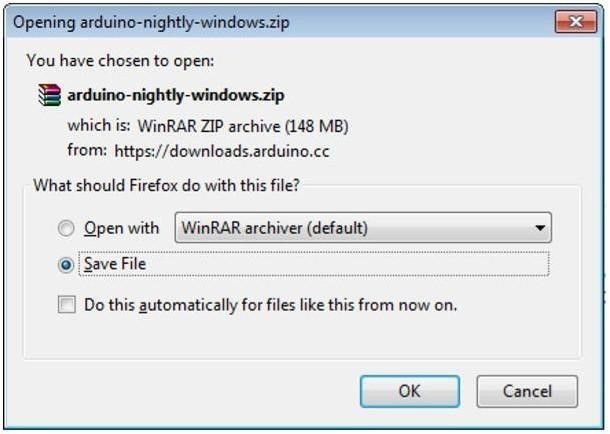


Figure.5.3 Opening Arduino

Step 3: Power up your board

The NODE MCU, Mega, Duemilanove, Arduino, and nano technology automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

## Step 4: Launch Arduino IDE

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an

infinity label (application.exe). Double-Click the icon to start the IDE

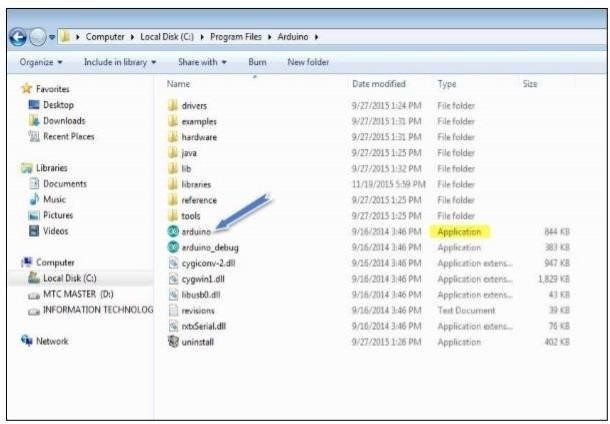
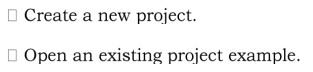


Figure.5.4Launching Arduino IDE

Step 5: Open your first project

Once the software starts, you have two options:



To create a new project, select File --> New

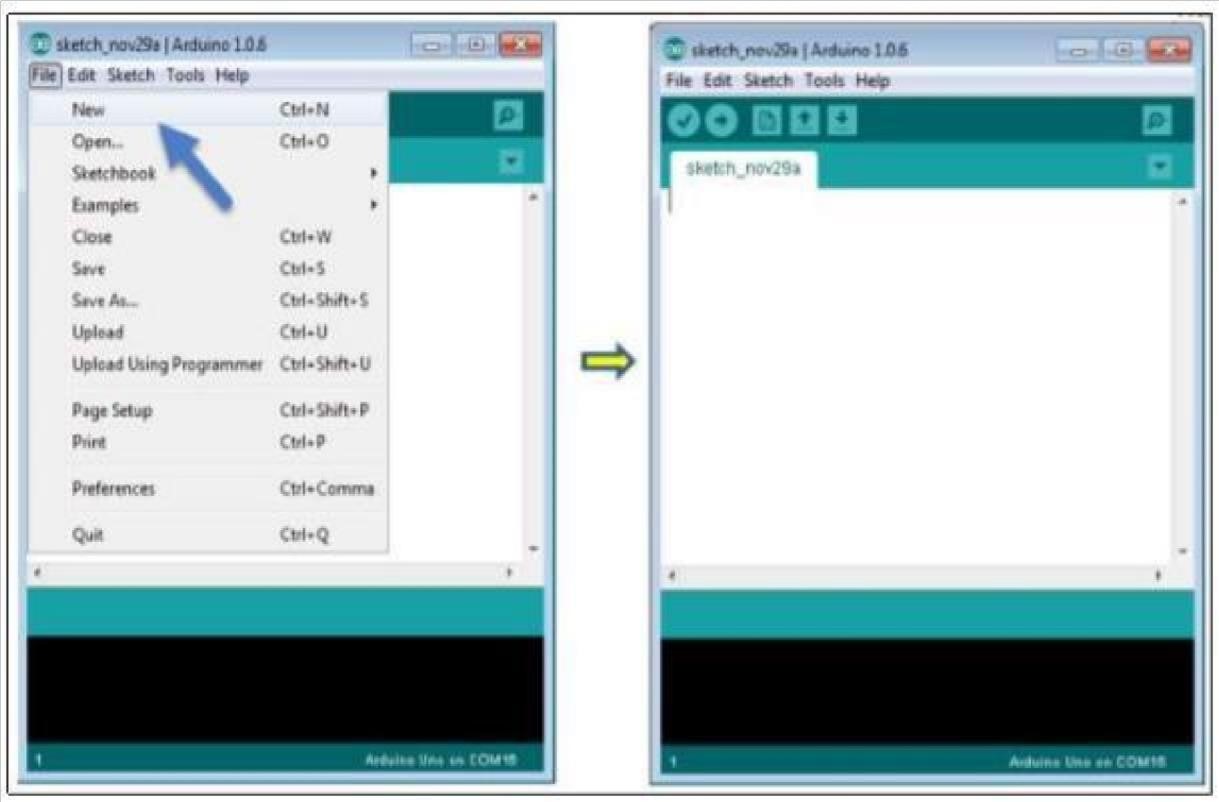


Figure.5.5 Arduino Software Opening New File

To open an existing project example, select File -> Example ->

Basics->Blink.

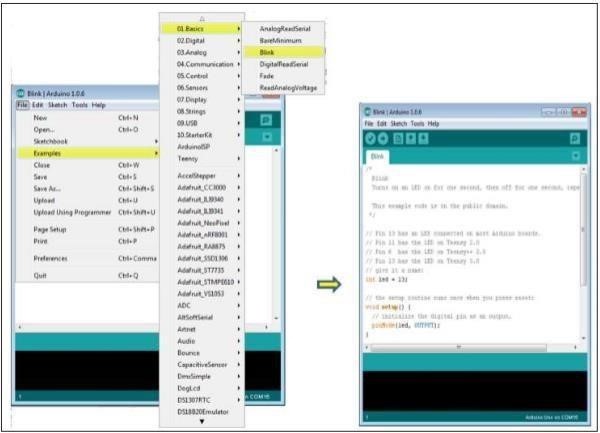


Figure 5.6 Selecting File Type

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any

other example from the list.

## Step 6: Select your Arduino board

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools -> Board and select your board.

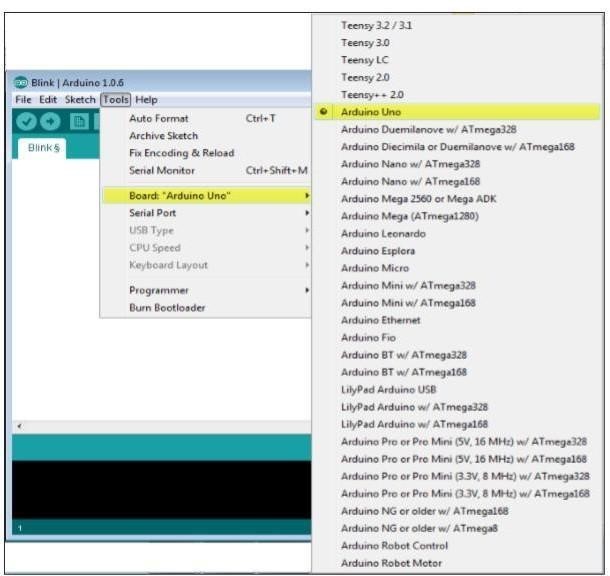


Figure 5.7 Selecting Board Type

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7: Select your serial port

Select the serial device of the Arduino board. Go to Tools -> Serial Port menu.This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and

select that serial port.

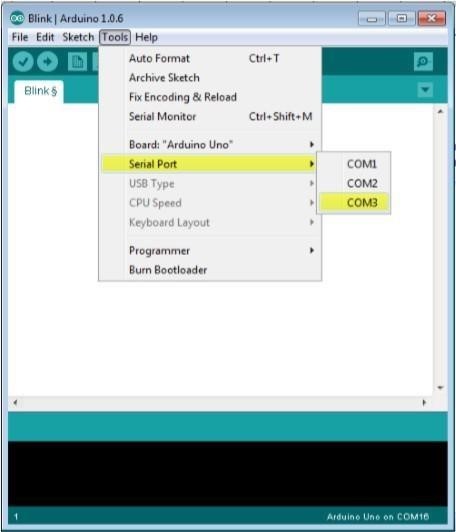


Figure 5.8 Selecting Serial Port

## Step 8: Upload the program to your board

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the

Arduino IDE toolbar

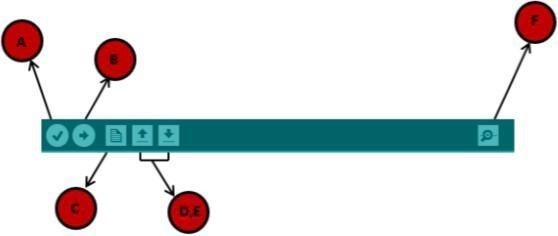


Figure 5.9 Uploading Program To Board

1. Used to check if there is any compilation error.
2. Used to upload a program to the Arduino board.
3. Shortcut used to create a new sketch.
4. Used to directly open one of the example sketch.
5. Used to save your sketch.
6. Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

**Note**: If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

### **5.3 ARDUINO – PROGRAM STRUCTURE**

We will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is opensource. 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

Function:



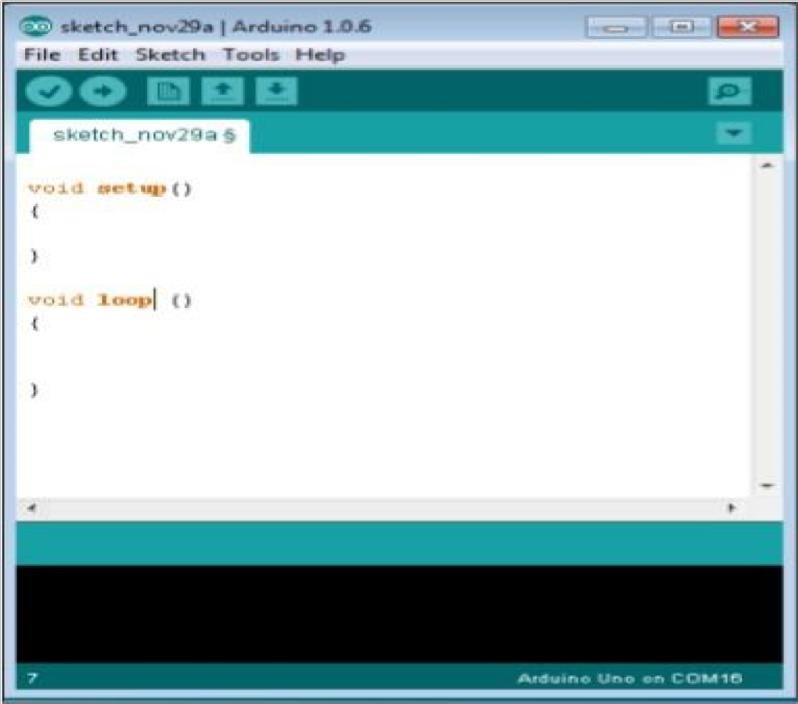


Figure 5.10 Program Structure



**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:



**PURPOSE:** After creating a setup () function, which initializes and set 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: