Unit-6

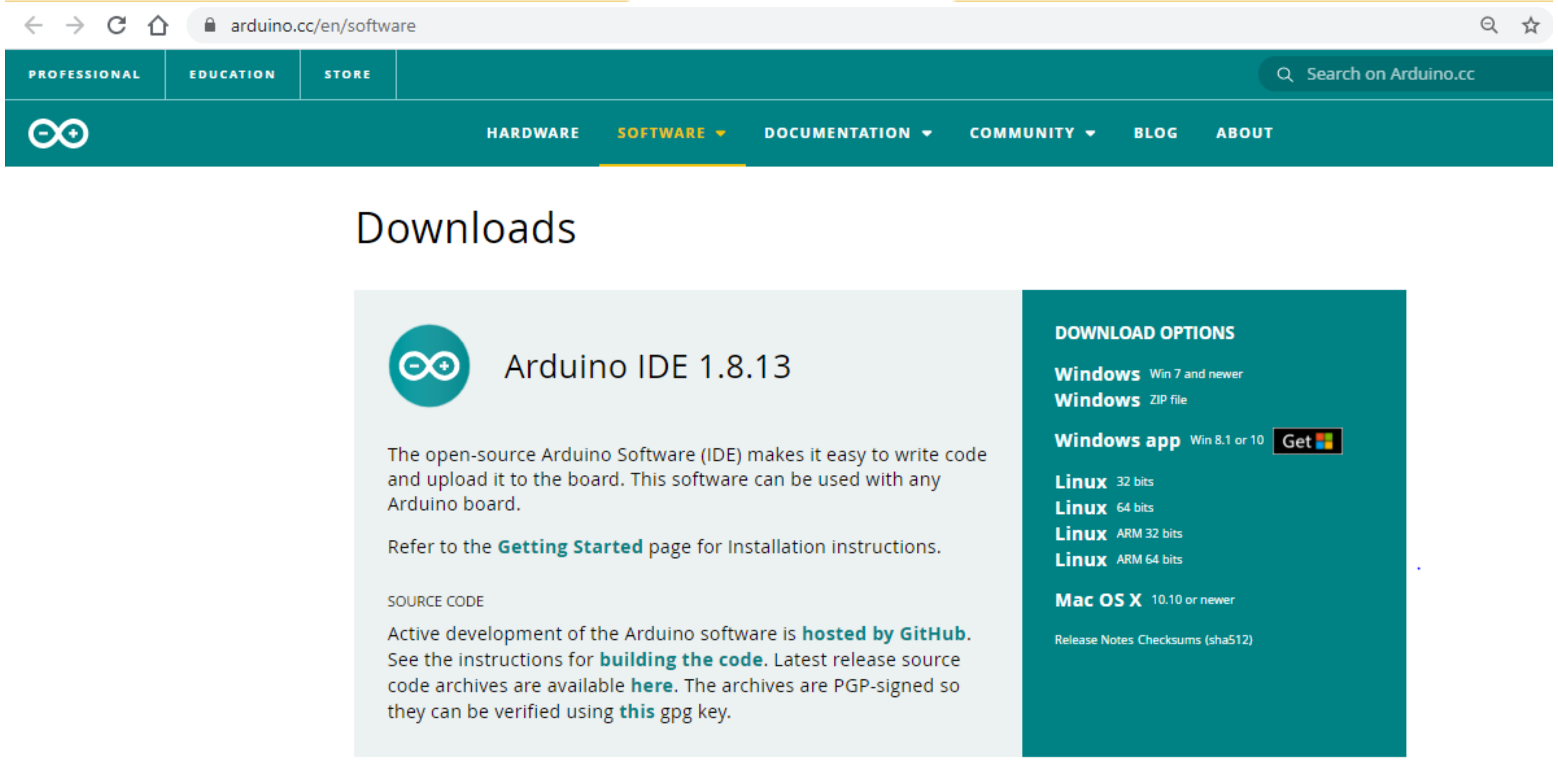
Arduino and Raspberry Pi

Arduino

Section - 1

Programming the Arduino

- Programming of Arduino is done in Arduino IDE – Integrated Development Environment.
- The software helps to connect, upload and communicate with Arduino hardware.
- The Arduino software is an open-source platform.
- The steps to install the Arduino software into the computer are as following:
- Download Arduino IDE from http://arduino.cc/en/Main/Software



Downloads

Arduino IDE 1.8.13

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

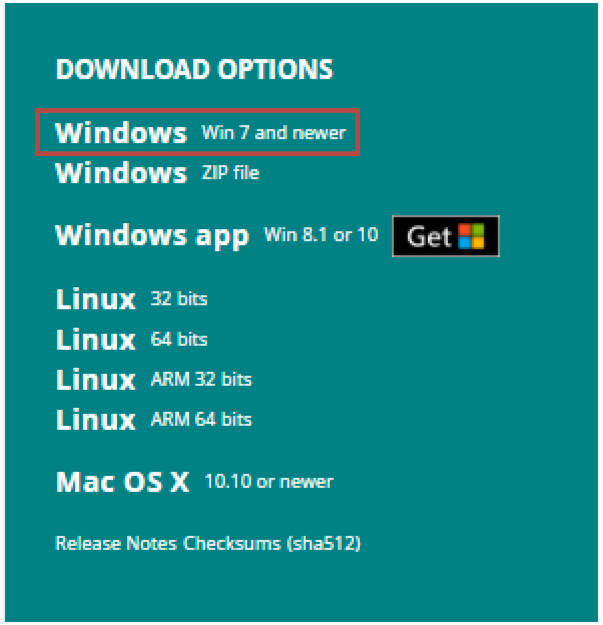
Refer to the **Getting Started** page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available **here**. The archives are PGP-signed so they can be verified using **this** gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file
Windows app Win 8.1 or 10 Get
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits
Mac OS X 10.10 or newer
Release Notes Checksums (sha512)

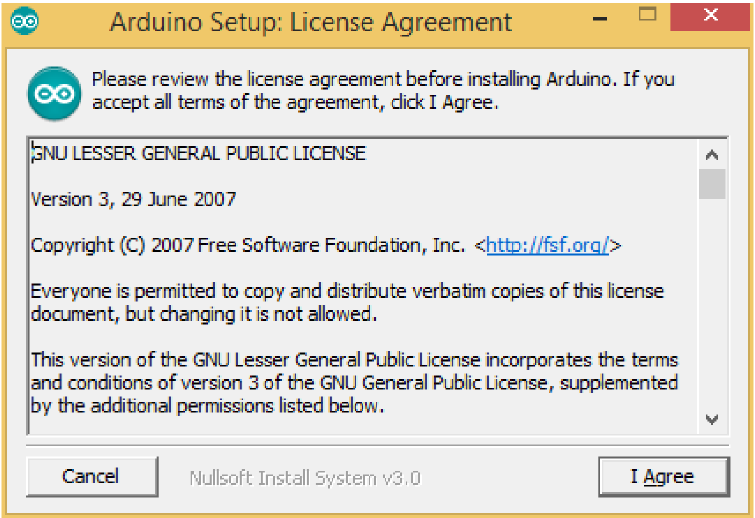


DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file
Windows app Win 8.1 or 10 Get
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits
Mac OS X 10.10 or newer
Release Notes Checksums (sha512)

Installation process of Arduino

- After downloading the installer double click and install the software by agreeing to License Agreement.
- Select the components required to be installed and click on next. Make sure to have free space in hard drive which is mentioned in the software.
- Select the path where you want to install Arduino IDE and click on “Install”
- Now wait for few minutes until installation process gets completed.
- The dialogue box will show the “Completed” status of the installation. Click on “Close” button.



Arduino Setup: License Agreement

Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

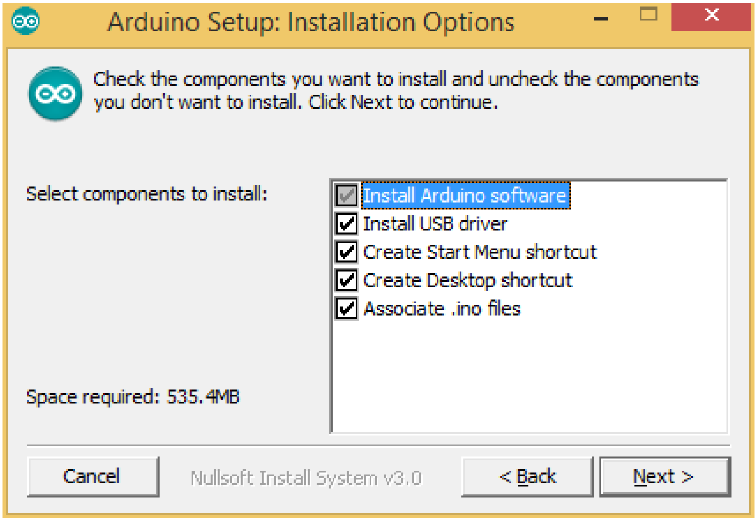
GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.



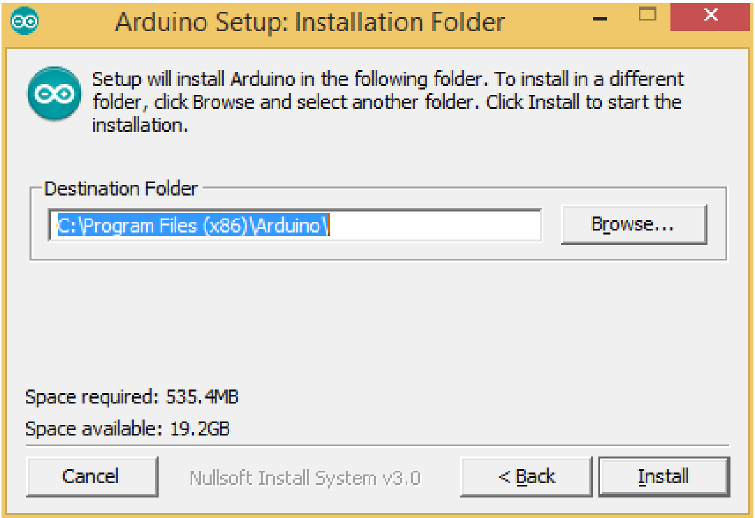
Arduino Setup: Installation Options

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- ☑ Install Arduino software
- ☑ Install USB driver
- ☑ Create Start Menu shortcut
- ☑ Create Desktop shortcut
- ☑ Associate .ino files

Space required: 535.4MB



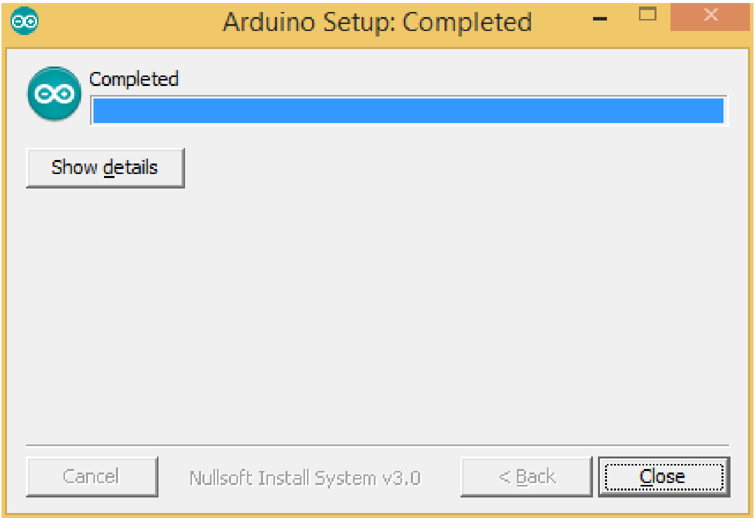
Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\Arduino\

Space required: 535.4MB

Space available: 19.2GB

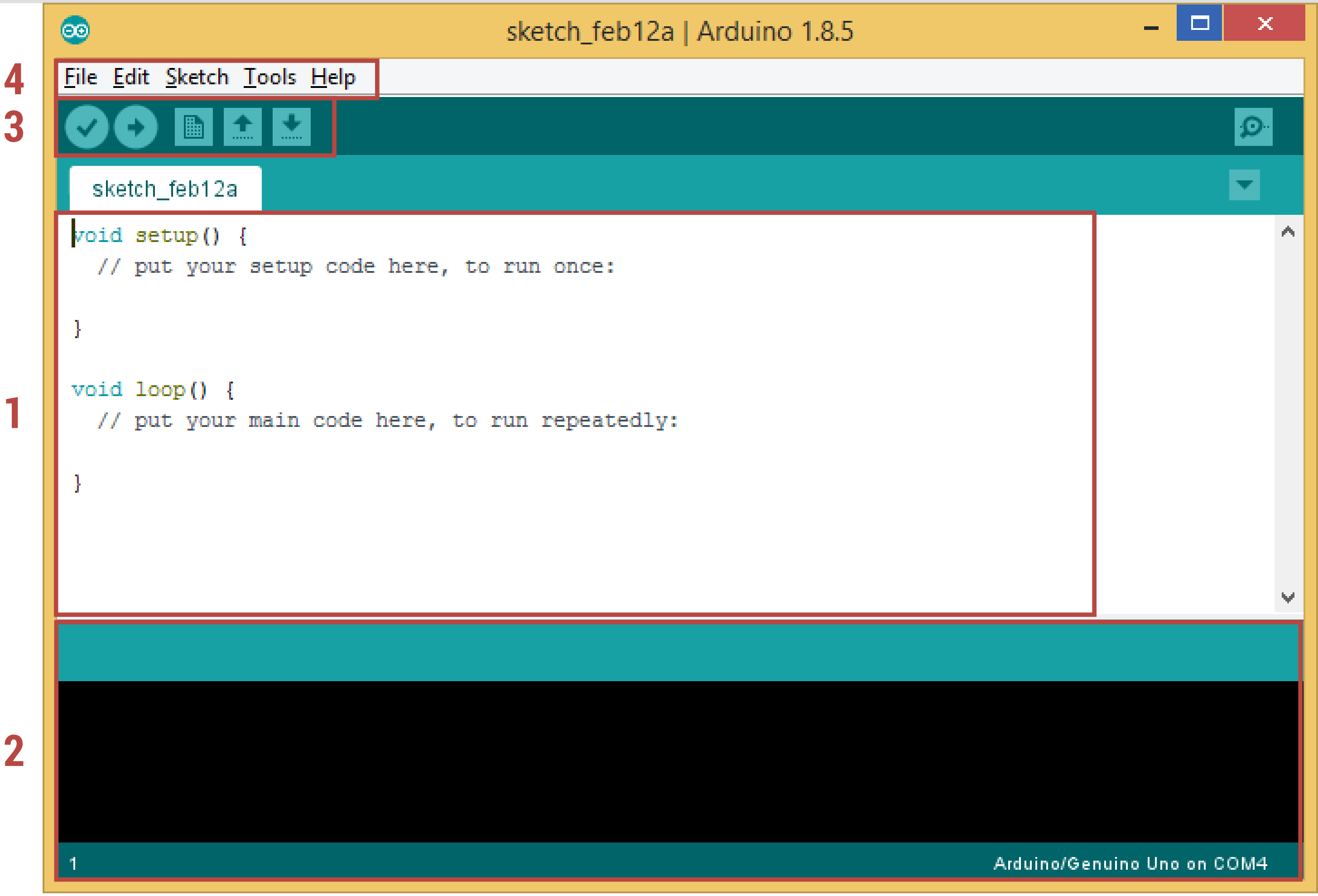


Arduino Setup: Completed

Completed

Arduino IDE Explanation

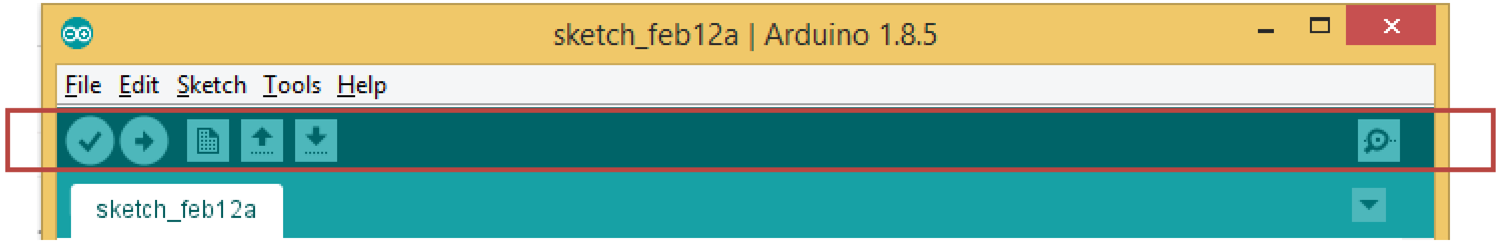
- Open Arduino IDE software. Here, we can find different sections.
  1. Editor – It is the space where we can write the code
  2. A text Console – for displaying the messages.
  3. A toolbar with buttons – given for common functions to perform.
  4. Menu bar – A menu bar with series of menus
- Note that, the name of the default file is sketch_[currentdate] with the alphabet ‘a’ and every new file has consecutive alphabet.


sketch_feb12a | Arduino 1.8.5
4
File Edit Sketch Tools Help
3
sketch_feb12a
1
2
1
Arduino/Genuino Uno on COM4

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Arduino IDE Explanation – Cont.

- The functions of buttons in toolbar can be explained as following:



Verify
Checks your code for errors compiling it.

Upload
Compiles your code and uploads it to the configured board.

New
Creates a new sketch.

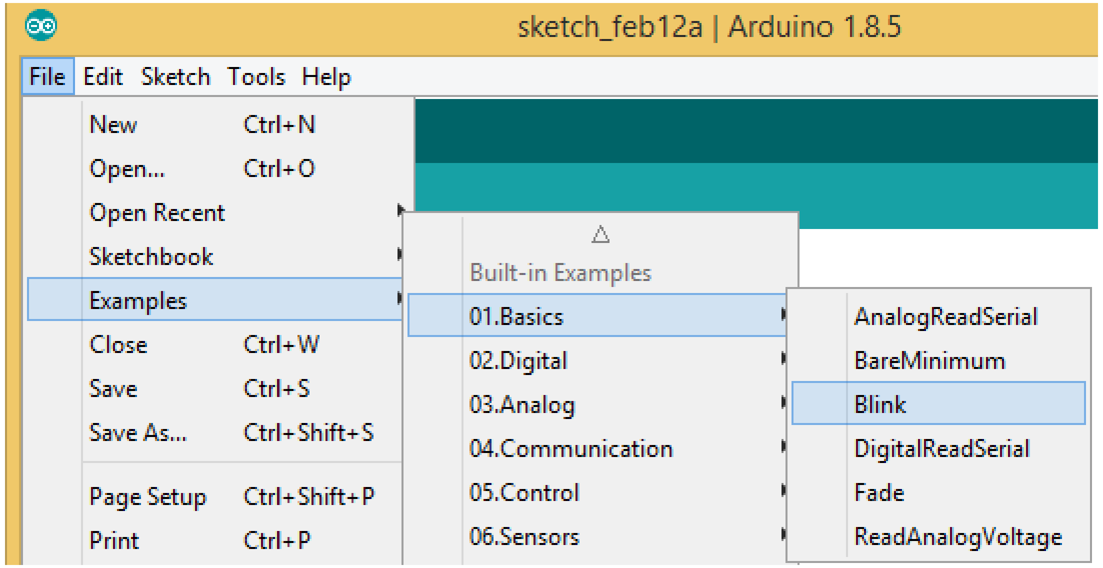
Open
Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

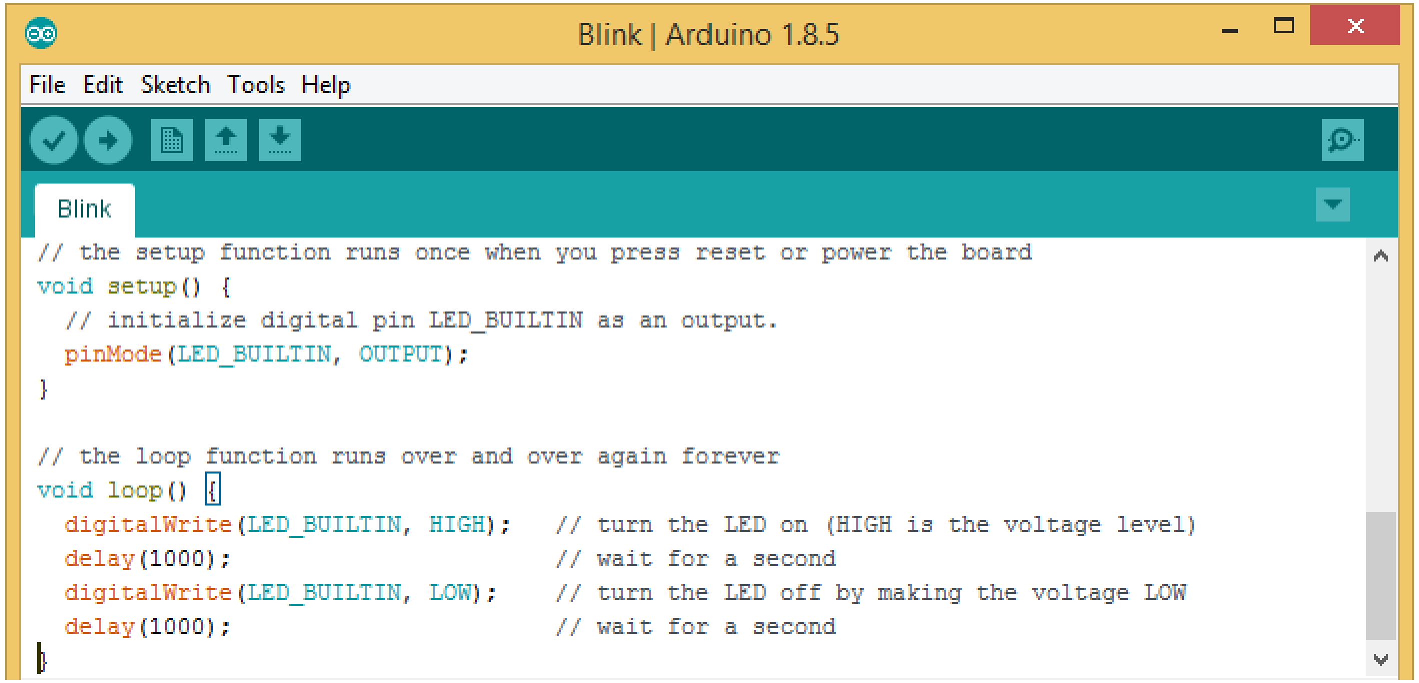
Save
Saves your sketch.

Serial Monitor
Opens the serial monitor.

Programming the First Code

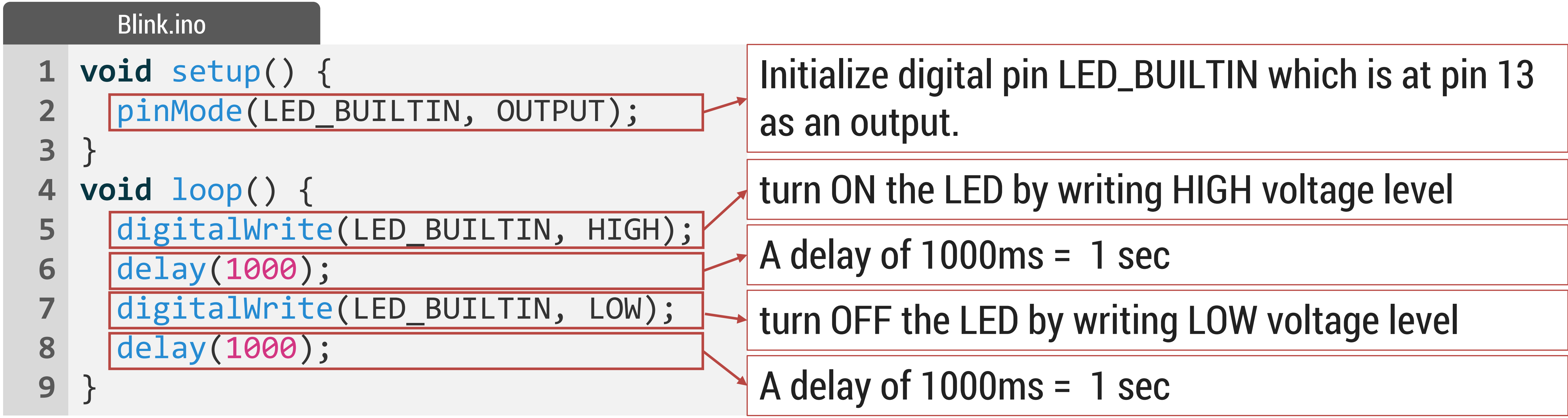
- The IDE is already loaded with few basic examples. Let us take a basic example of LED blinking which blinks built-in LED of Arduino UNO with a 1-second interval.
- Go to File > Examples > 01.Basics > Blink as shown in the figure.
- It will load the prewritten example in a new window with the title “Blink”.





```
// 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
}
```

Code Explanation



Blink.ino

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

Initialize digital pin LED_BUILTIN which is at pin 13 as an output.

turn ON the LED by writing HIGH voltage level

A delay of 1000ms = 1 sec

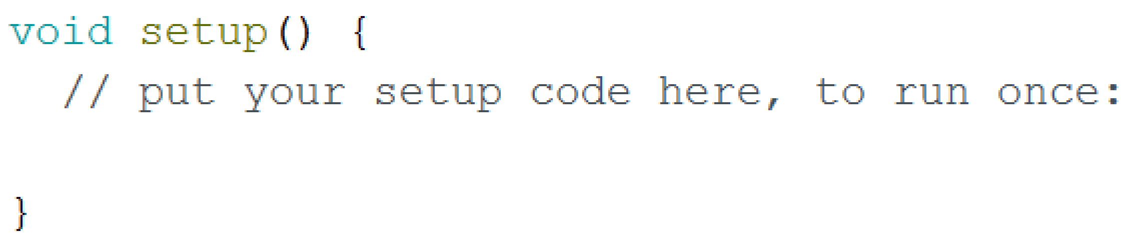
turn OFF the LED by writing LOW voltage level

A delay of 1000ms = 1 sec

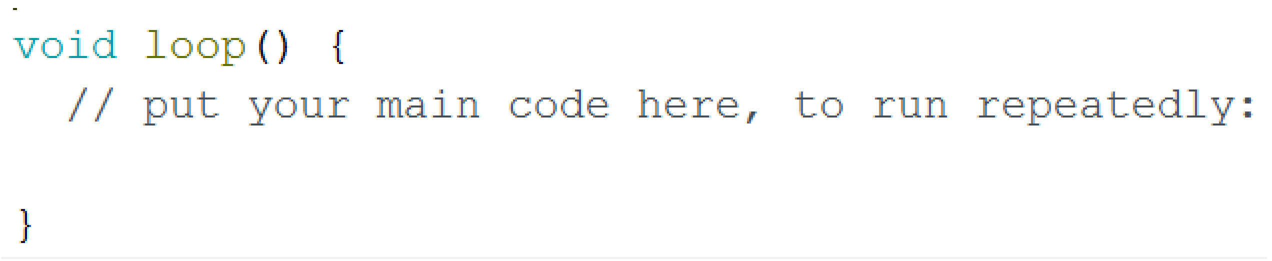
- We should initialize the pin as input or output once only. Therefore the syntax for defining the pin as output is written in void setup ( ) function.
- For blinking of LED, the LED should turn on and off at the regular interval of time. Hence, the instructions for turning on, turning off and delay for both on time and off time are written in the void loop ( ) function

Code Explanation

- Arduino code is always written with two mandatory functions.
  - void setup ( )
  - void loop ( )
- The instructions which are to be performed once in the program are written in void setup ( ) function.
- The instructions written in void loop ( ) are executed infinite times until the next restart of Arduino board.



```
void setup() {
  // put your setup code here, to run once:

}
```



```
void loop() {
  // put your main code here, to run repeatedly:

}
```

- Comments are important for any programming language. They are ignored while execution of the program
  - Anything written after // in Arduino program is considered as a single line comment
  - Anything written between /* and */ is considered as multiple line comments.

Functions in Arduino IDE

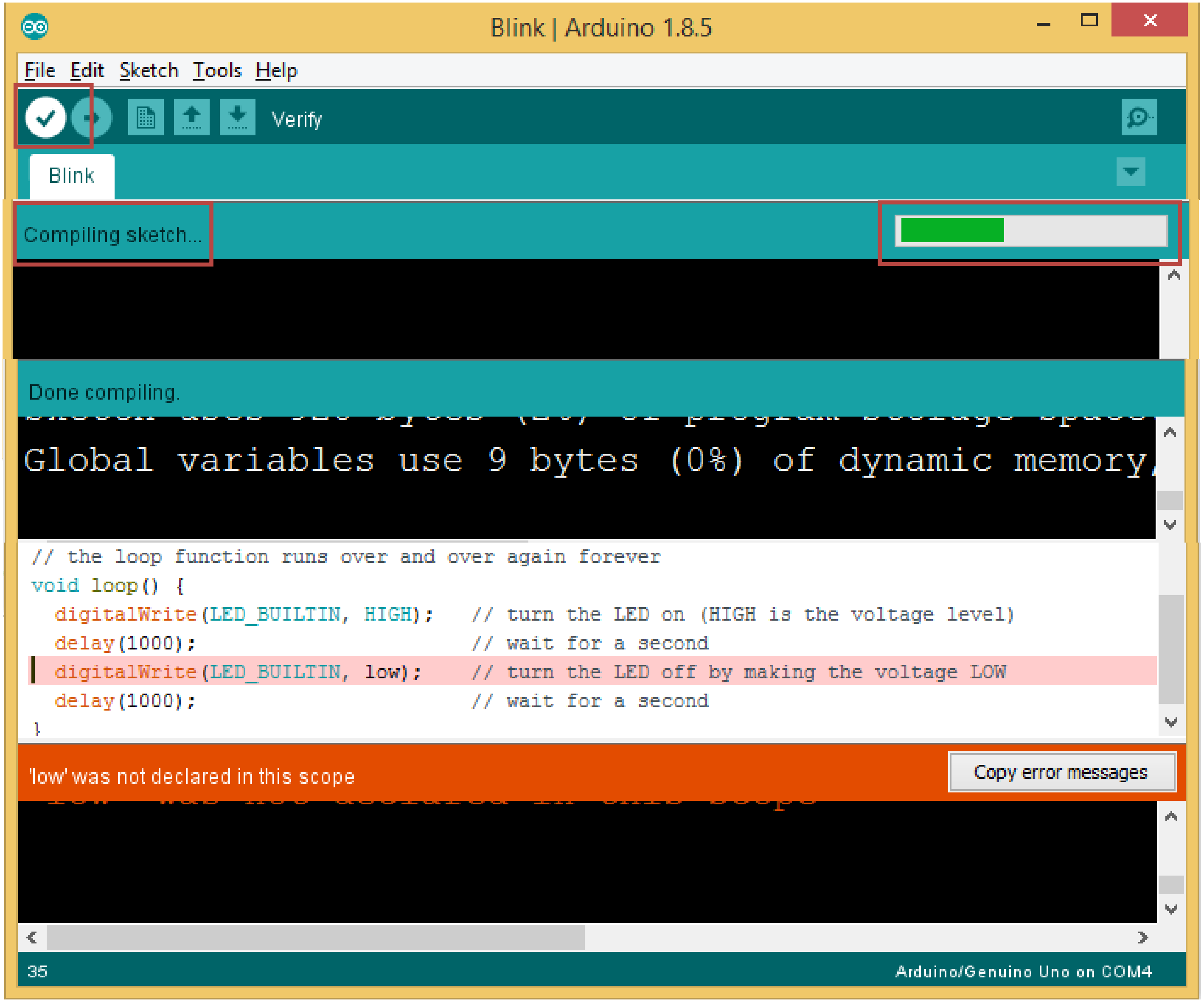
- pinMode( ): Configures the specified pin to behave either as an input or an output.
  - Syntax : pinMode(pin, mode)
  - Parameters :
    - pin: the Arduino pin number to set the mode of.
    - Mode: INPUT or OUTPUT
- digitalWrite( ) : Write a HIGH or a LOW value to a digital pin.
  - Syntax : digitalWrite(pin, value)
  - Parameters :
    - pin : the Arduino pin number.
    - Mode : HIGH or LOW
- What is HIGH or LOW?
  - **HIGH**: - Also considered as Logic ‘1’ or Logic High. It will set 5V on the configured pin of Arduino.
  - **LOW**: - Also considered as Logic ‘0’ or Logic Low. It will set 0V on the configured pin of Arduino.

Functions in Arduino IDE (Cont.)

- delay( ): It pauses the program for the amount of time (in milliseconds) specified as the parameter. For example, if we want to hold the program for 1 second before executing the next instruction. A delay of 1000ms = 1s is passed in the parameter.
  - Syntax : delay(ms)
  - Parameters :
    - ms: the number of milliseconds to pause

Compiling the Code

- To compile a code in Arduino click on the “Verify” button in menu bar.
- While compiling the code, a message “Compiling sketch” is shown with its progress in the status bar.
- If there is no error in the code, the message “Done compiling” is displayed in the status bar.
- If an error is present in the code it will show the message of error with highlighted syntax where the error is present.



Compiling sketch...

Done compiling.

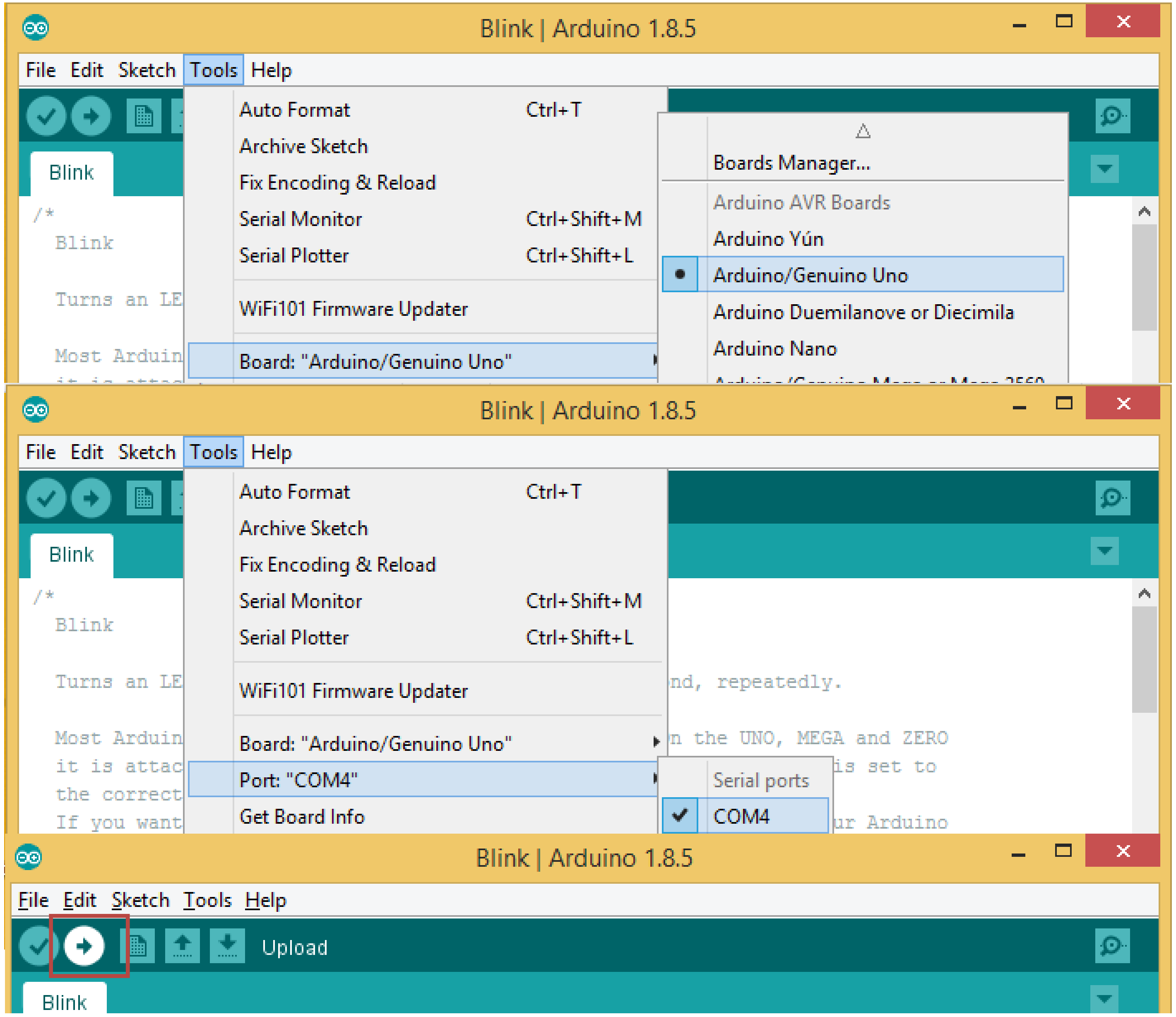
```
Global variables use 9 bytes (0%) of dynamic memory,
```

```
// 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
}
```

'low' was not declared in this scope

Upload the first code in Arduino Hardware

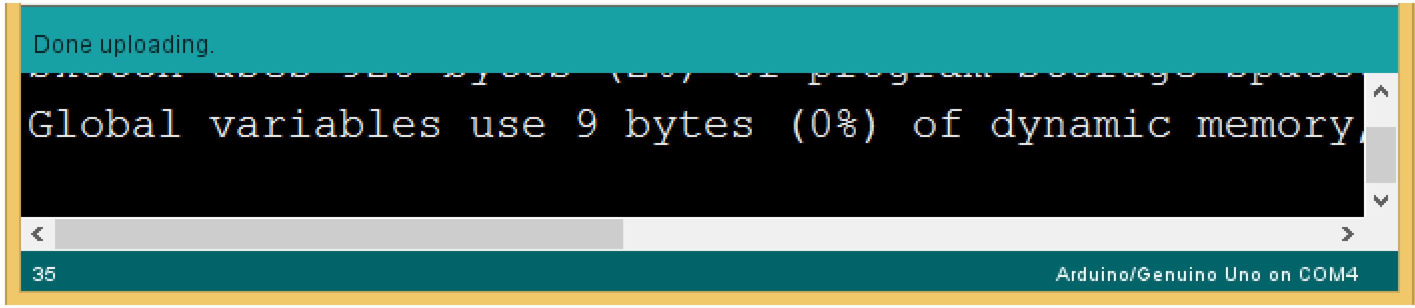
- Select the proper board type: First, we need to select the appropriate board type in which we need to upload the program.
- Go to Tools > Board menu and select the board which is used as the hardware.
- Select port: Select the serial device of the board from the Tools > Port menu.
- This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).
- Upload the program: Now, simply click the "Upload" button in the environment.



```
/*
  Blink

  Turns an LE
  Most Arduin
```

Upload the first code in Arduino Hardware

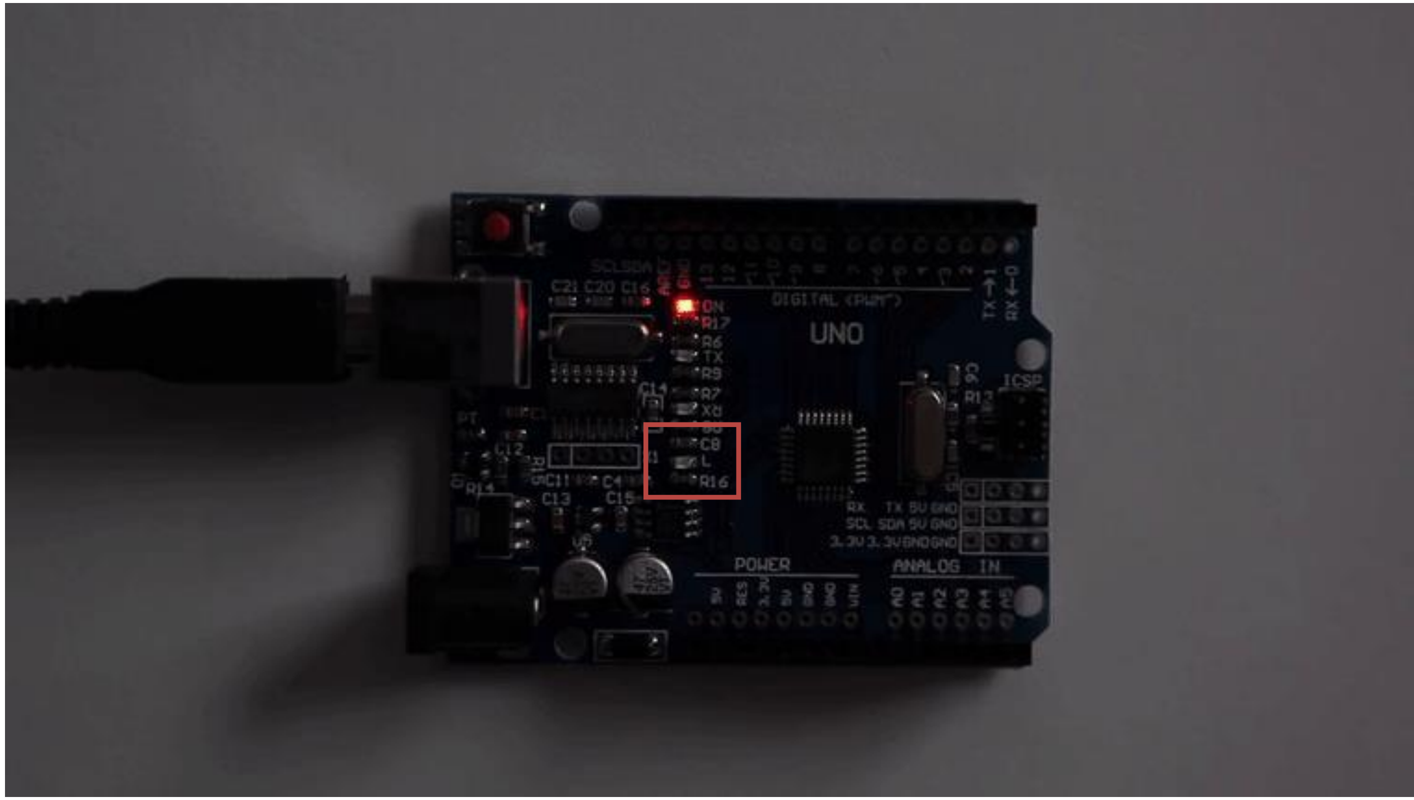
- While uploading process, TX and RX LED will be flashing.
- If uploading is done successfully, it will show a message “Done uploading” in the status bar.
- Once the program is uploaded, the built-in LED of the Arduino board will flash with one second delay.
- It means LED turns ON for 1 second and LED turns OFF for 1 second.



Done uploading.

```
Global variables use 9 bytes (0%) of dynamic memory,
```

35 Arduino/Genuino Uno on COM4



Arduino Technology Architecture and Its Advantages

watelectronics.com/arduino-technology-architecture-and-applications

admin

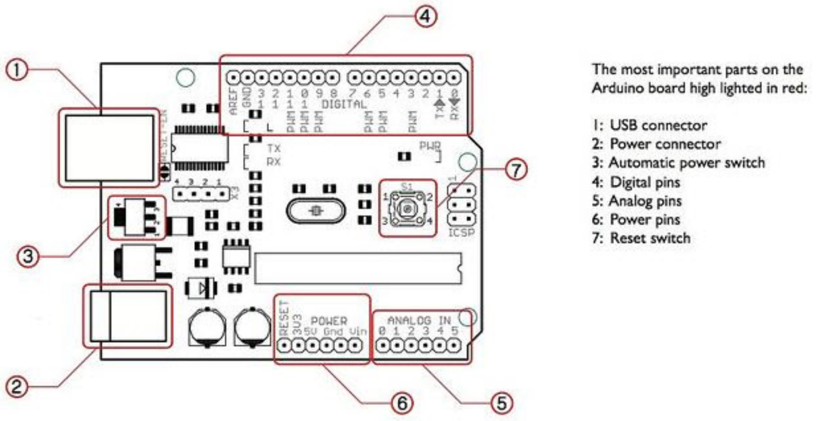
January 8, 2021

Arduino Board

An Arduino board is a one type of microcontroller based kit. The first Arduino technology was developed in the year 2005 by David Cuartielles and Massimo Banzi. The designers thought to provide easy and low cost board for students, hobbyists and professionals to build devices. Arduino board can be purchased from the seller or directly we can make at home using various basic components. The best examples of Arduino for beginners and hobbyists includes motor detectors and thermostats, and simple robots. In the year 2011, Adafruit industries expected that over 3lakhs Arduino boards had been produced. But, 7lakhs boards were in user's hands in the year 2013. Arduino technology is used in many operating devices like communication or controlling.

Arduino Technology

A typical example of the Arduino board is Arduino Uno.It includes an ATmega328 microcontroller and it has 28-pins



The most important parts on the Arduino board high lighted in red:

1: USB connector
2: Power connector
3: Automatic power switch
4: Digital pins
5: Analog pins
6: Power pins
7: Reset switch

Arduino Pin Diagram

The pin configuration of the Arduino Uno board is shown in the above. It consists of 14-digital i/o pins. Wherein 6 pins are used as pulse width modulation o/ps and 6 analog i/ps, a USB connection, a power jack, a 16MHz crystal oscillator, a reset button, and an ICSP header. Arduino board can be powered either from the personal computer through a USB or external source like a battery or an adaptor. This board can operate with an external supply of 7-12V by giving voltage reference through the IORef pin or through the pin Vin.

Digital I/Ps

It comprises of 14-digital I/O pins, each pin take up and provides 40mA current. Some of the pins have special functions like pins 0 & 1, which acts as a transmitter and receiver respectively. For serial communication, pins-2 & 3 are external interrupts, 3,5,6,9,11 pins delivers PWM o/p and pin-13 is used to connect LED.

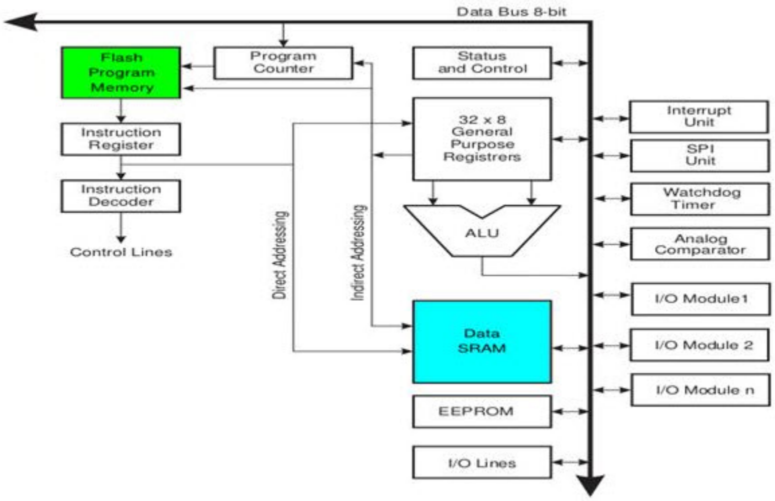
Analog i/ps: It has 6-analog I/O pins, each pin provide a 10 bits resolution.

Aref: This pin gives a reference to the analog i/ps.

Reset: When the pin is low, then it resets the microcontroller.

Arduino Architecture

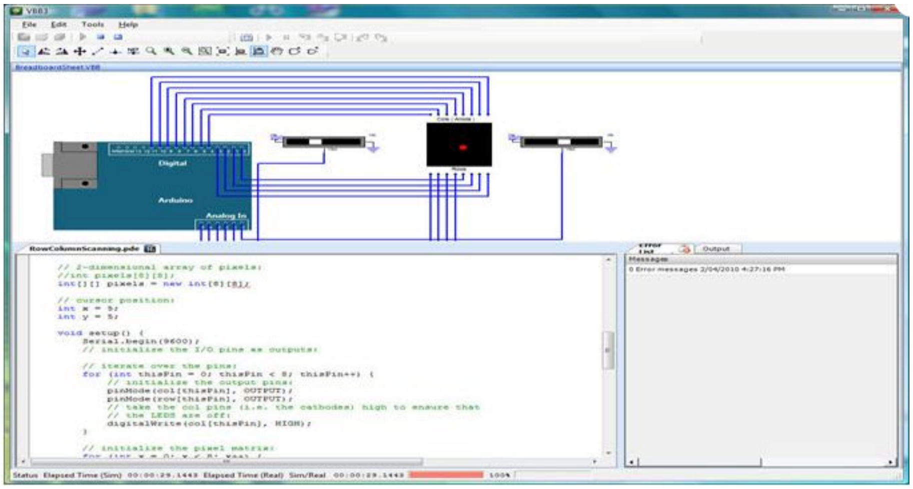
Basically, the processor of the Arduino board uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories such as program memory and data memory. Wherein the data is stored in data memory and the code is stored in the flash program memory. The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM 1kb of EPROM and operates with a 16MHz clock speed.



Arduino Architecture

How to program an Arduino?

The main advantage of the Arduino technology is, you can directly load the programs into the device without the need of a hardware programmer to burn the program. This is done because of the presence of the 0.5KB of boot loader, that allows the program to be dumped into the circuit. The Arduino tool window contains a toolbar with a various buttons like new, open, verify, upload and serial monitor. And additionally it comprises of a text editor (employed to write the code), a message space (displays the feedback) like showing the errors, the text console, that displays the o/p & a series of menus just like the file, tool menu & edit.



```
// 2-dimensional array of pixels:
//int pixels[8][8];
int[][] pixels = new int[8][8];

// cursor position:
int x = 5;
int y = 5;

void setup() {
    Serial.begin(9600);
    // initialize the I/O pins as outputs:

    // iterate over the pins:
    for (int thisPin = 0; thisPin < 8; thisPin++) {
        // initialize the output pins:
        pinMode(col[thisPin], OUTPUT);
        pinMode(row[thisPin], OUTPUT);
        // take the col pins (i.e. the cathodes) high to ensure that
        // the LEDS are off:
        digitalWrite(col[thisPin], HIGH);
    }

    // initialize the pixel matrix:
    for (int x = 0; x < 8; x++) {
```

0 Error messages 2/04/2010 4:27:16 PM

Status Elapsed Time (Sim) 00:00:29.1443 Elapsed Time (Real) Sim/Real 00:00:29.1443 100%

Arduino Program

- Programming into the Arduino board is called as sketches. Each sketch contains of three parts such as Variables Declaration, Initialization and Control code. Where, Initialization is written in the setup function and Control code is written in the loop function.
- The sketch is saved with .ino and any operation like opening a sketch, verifying and saving can be done using the tool menu.
- The sketch must be stored in the sketchbook directory.
- Select the suitable board from the serial port numbers and tools menu.
- Select the tools menu and click on the upload button, then the boot loader uploads the code on the microcontroller.

Basic Functions of Arduino Technology

- Digital read pin reads the digital value of the given pin.
- Digital write pin is used to write the digital value of the given pin.
- Pin mode pin is used to set the pin to I/O mode.
- Analog read pin reads and returns the value.
- Analog write pin writes the value of the pin.
- Serial. Begins pin sets the beginning of serial communication by setting the rate of bit.

Please refer to this link for Arduino MCQs

Advantages of Arduino Technology

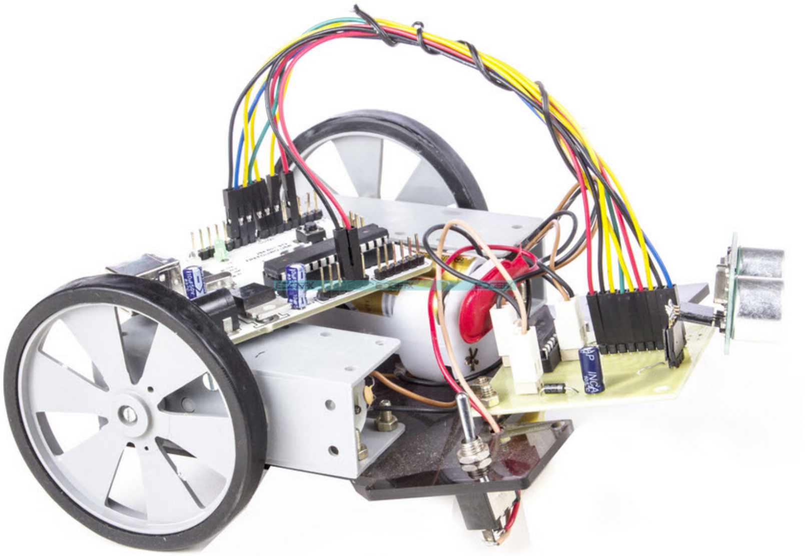
- It is cheap

- It comes with an open supply hardware feature that permits users to develop their own kit
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programing language libraries and may be extended and changed.
- For beginners, it is very simple to use.

Applications of Arduino Technology

The Obstacle Avoidance Robot Operated with Arduino

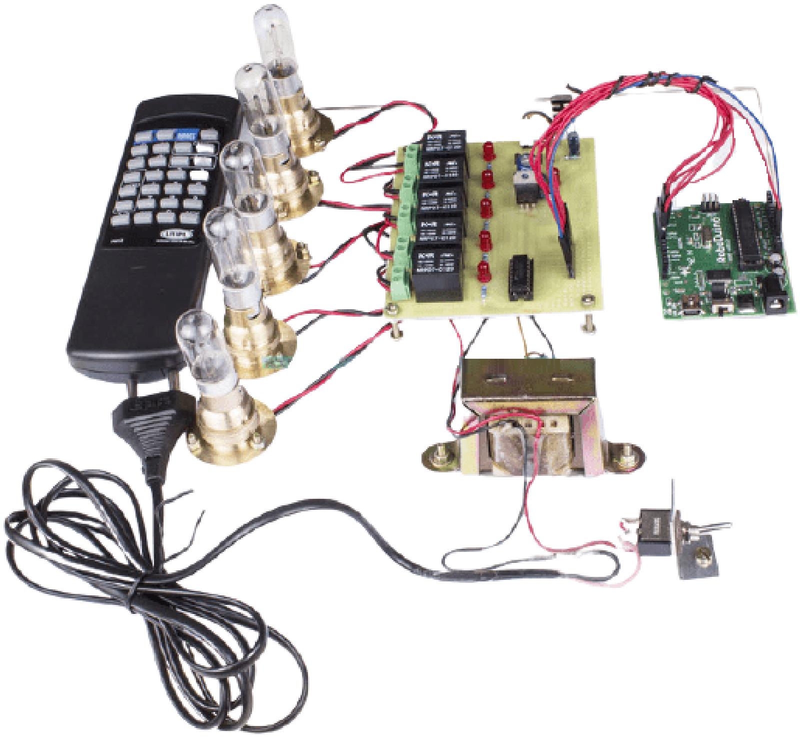
The main concept of this project is to design a robot using ultrasonic sensors to avoid the obstacle. A robot is a machine and it is a combination of programs instructions and motors. It can perform some task with some guidance or automatically. This robotic vehicle has an intelligence which is built inside of the robot. When a obstacle problem comes ahead of it then, it guides itself. This robot is designed with a microcontroller from Atmel family of Aduino board.



Arduino Operated Obstacle Avoidance Robot Project

Arduino based Controlling of Electrical Appliances using IR

The main goal of this project is to control the electrical appliances using an IR remote. This project uses the TV remote to transmit the coded data, then it is received by a sensor which is interfaced to the control unit. The proposed system controls the electrical loads depending on the transmitted data from the remote. Operating electrical appliances are very difficult for handicapped or senior people. This project gives the solution by integrating household appliances to a control unit which can be operated with a TV remote.

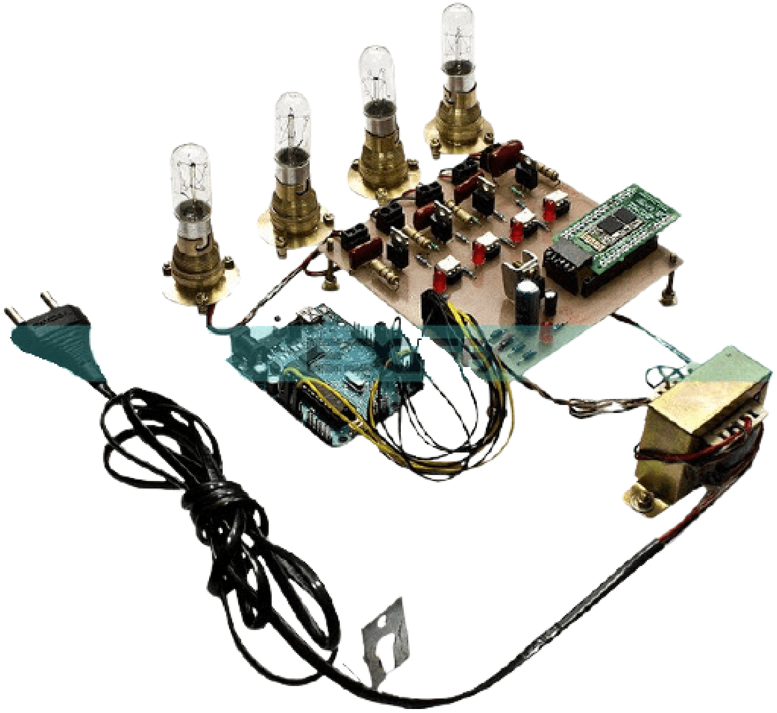


Arduino based Electrical Appliances Control using IR Project

Know more about Arduino MCQs.

Arduino based Home Automation

The main goal of this project is to design a home automation system using an Arduino board with Bluetooth being controlled remotely by any Android OS based smart phone. This home automation system provides a modern solution with smart phones.In order to achieve this, a Bluetooth device is attached to the Arduino board at the receiver side and while on the transmitter side, a GUI application on the smart phone sends ON/OFF commands to the receiver where loads are connected. By touching the particular location on the graphical user interface (GUI), the different loads can be remotely turned ON/OFF via this technology. When we touch the exact location on the GUI, then the loads can be turned ON/OFF remotely. The loads works with an Arduino board through Thyristors and Opto-Isolators using triacs.



Arduino based Home Automation Project

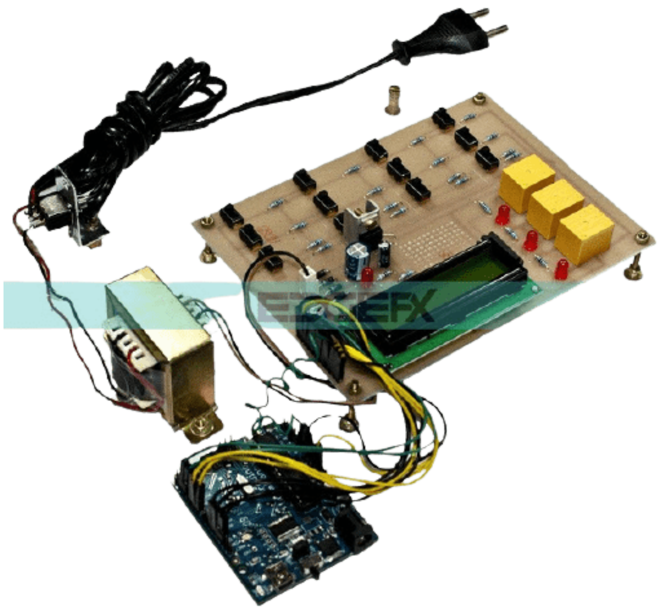
Underground Cable Fault Recognition using the Arduino Board

The main concept of this project is to find the location of the underground cable fault from the base station in kms using an Arduino technology. The underground cable fault is a common problem in many urban areas. When a fault occurs in the underground cable, at that time the repairing of that cable is very difficult due to not knowing the location of the cable.

Please refer to this link to know more about Arduino Projects for Engineering Students

This project is built with a set of resistors to represent the cable length in kilometers and fault creation is designed with a set of switches at every known kilometer to cross check the exactness of the same. When a fault occurs at a particular distance and the particular phase is displayed on an LCD which is interfaced to the Arduino board.

This is all about Arduino technology and its applications. So this is the basic information regarding Arduino which can be used for many applications like controlling of actuators for example generators, motors & based on the input of the sensors. Moreover, any questions regarding this topic or electrical and electronic projects, please give your feedback by commenting in the comment section below. Here is a question for you, What is the physical location of the U1A symbol on Arduino Uno?



Arduino based Underground Cable Fault Detection Project