MANAV RACHNA INTERNATIONAL INSTITUTE OF RESEARCH AND STUDIES

FACULTY OF ENGINEERING AND TECHNOLOGY

Practical File

for

____Semester (Academic Year**_____)**

SUBJECT NAME: _____

SUBJECT CODE: _____

Submitted By:-

Student Name: _____

Roll No.: _____

Branch: _____

Section:_____

Submitted To:-

Faculty Name: _____

Designation: _____

Department: _____

_____

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Practical No.	Aim of Practical	Date on which Conducted	Teacher's Signature

Practical No.	Aim of Practical	Date on which Conducted	Teacher's Signature

Practical No.	Aim of Practical	Date on which Conducted	Teacher's Signature

EXPERIMENT NO – 1(a)

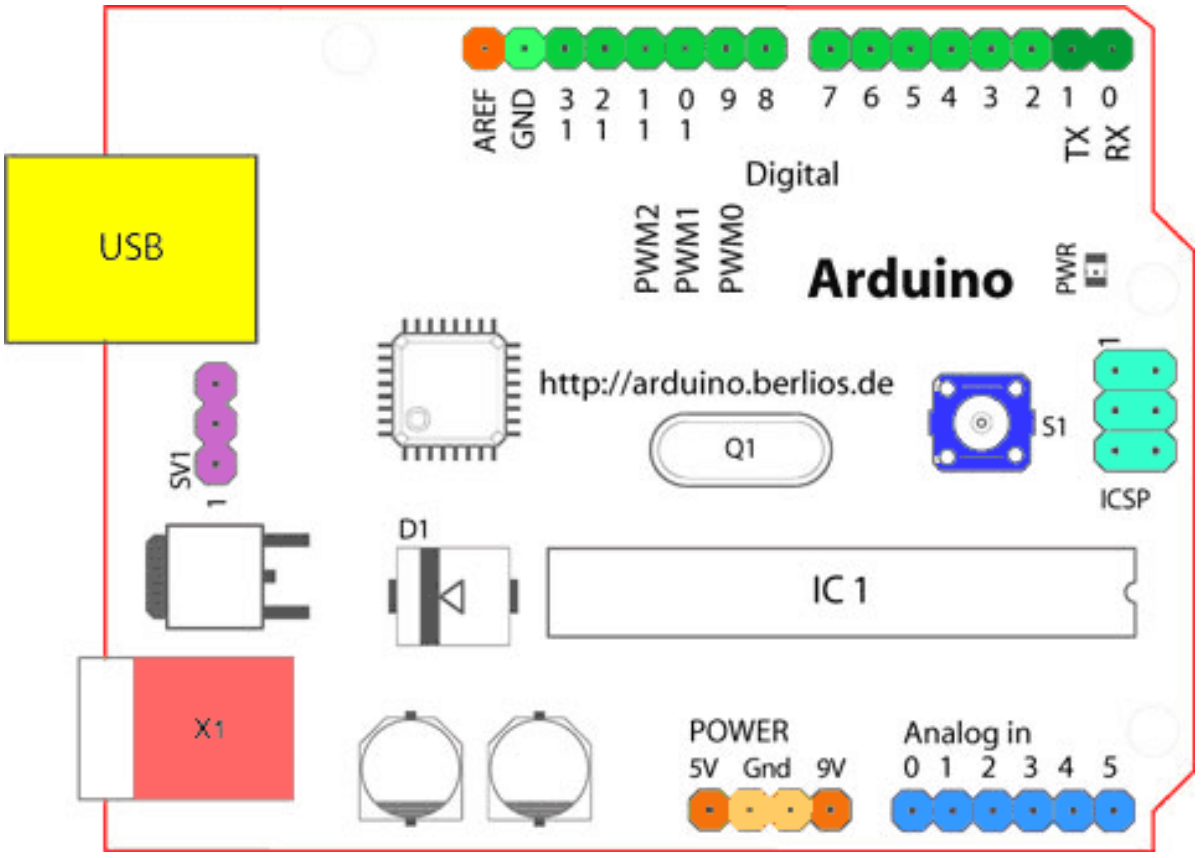
AIM – Introduction to Arduino.

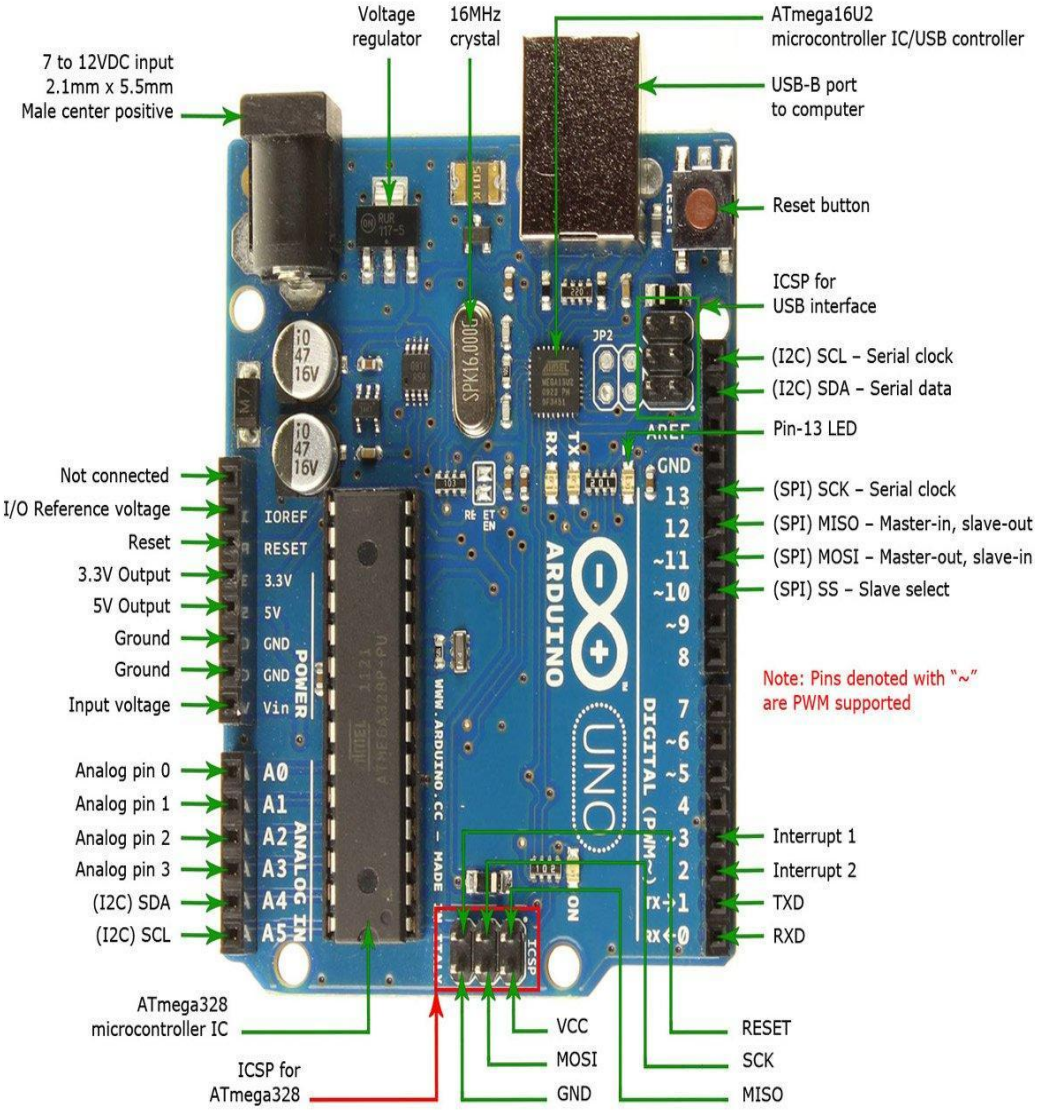
Arduino –

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

Arduino Components –



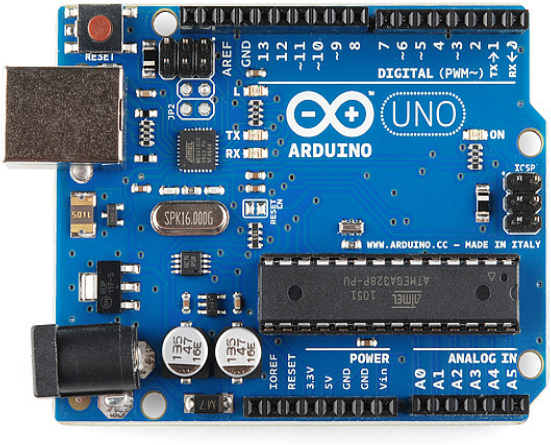


- **Different Boards of Arduino –**

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well-suited to someone new to the world of Arduino:

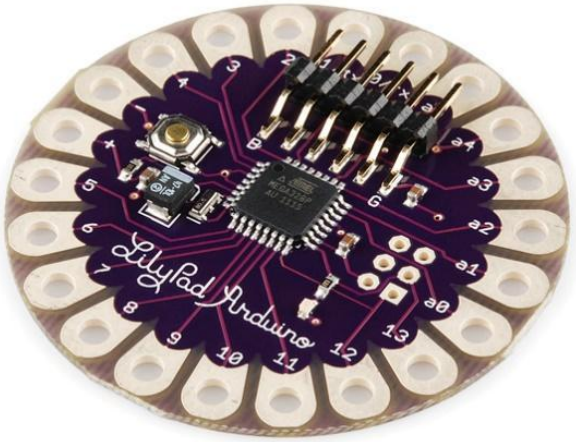
Arduino Uno (R3) -

The Uno is a great choice for your first Arduino. It's got everything you need to get started, and nothing you don't. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, a reset button and more. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.



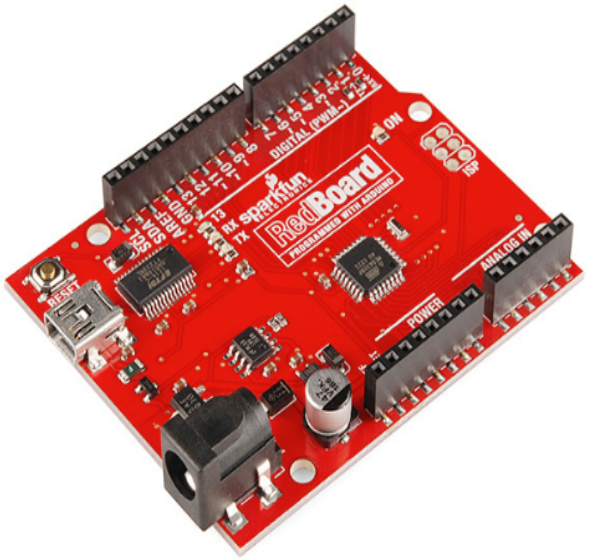
LilyPad Arduino -

This is LilyPad Arduino main board! LilyPad is a wearable e-textile technology developed by Leah Buechley and cooperatively designed by Leah and SparkFun. Each LilyPad was creatively designed with large connecting pads and a flat back to allow them to be sewn into clothing with conductive thread. The LilyPad also has its own family of input, output, power, and sensor boards that are also built specifically for e-textiles. They're even washable!



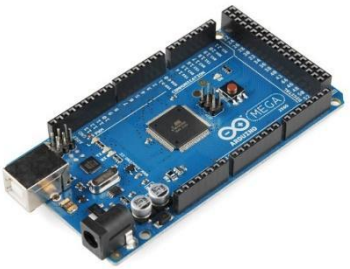
RedBoard -

The RedBoard can be programmed over a USB Mini-B cable using the Arduino IDE. It'll work on Windows 8 without having to change your security settings (we used signed drivers, unlike the UNO). It's more stable due to the USB/FTDI chip we used, plus it's completely flat on the back, making it easier to embed in your projects. Just plug in the board, select "Arduino UNO" from the board menu and you're ready to upload code. You can power the RedBoard over USB or through the barrel jack. The on-board power regulator can handle anything from 7 to 15VDC.



Arduino Mega (R3) -

The Arduino Mega is like the UNO's big brother. It has lots (*54!*) of digital input/output pins (14 can be used as PWM outputs), 16 analog inputs, a USB connection, a power jack, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The large number of pins make this board very handy for projects that require a bunch of digital inputs or outputs (like lots of LEDs or buttons).

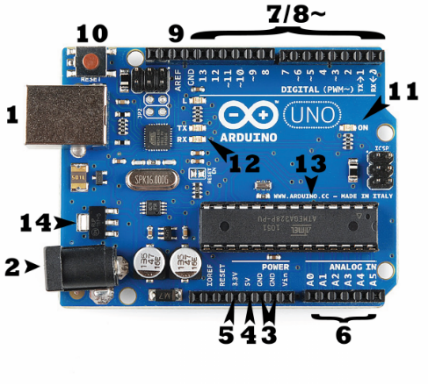


Arduino Leonardo

The Leonardo is Arduino's first development board to use one microcontroller with built-in USB. This means that it can be cheaper and simpler. Also, because the board is handling USB directly, code libraries are available which allow the board to emulate a computer keyboard, mouse, and more!

Components And its Roles -

There are many varieties of Arduino boards that can be used for different purposes. Some boards look a bit different from the one below, but most Arduinos have the majority of these components in common:



Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labeled **(1)** and the barrel jack is labeled **(2)**.

The USB connection is also how you will load code onto your Arduino board.

NOTE: Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- **GND (3)**: Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog (6)**: The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
- **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM).
- **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear -- once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC.

Voltage Regulator

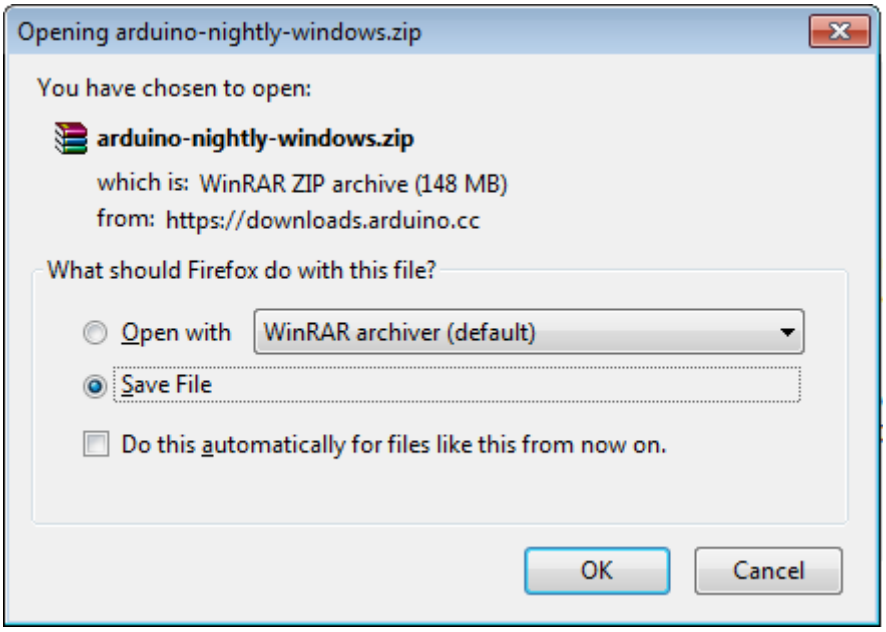
The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says -- it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

EXPERIMENT NO – 1(b)

AIM – Install the Arduino Software IDE

Step 1− 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.



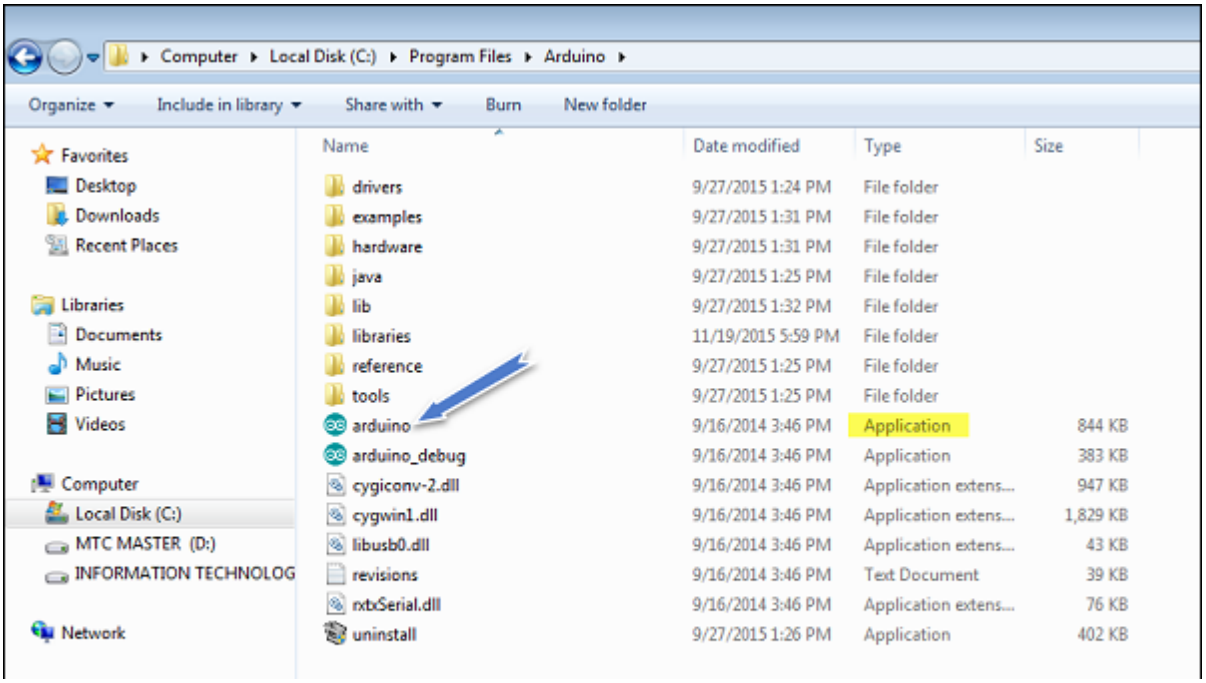
Step 2 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano 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 3 − 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.

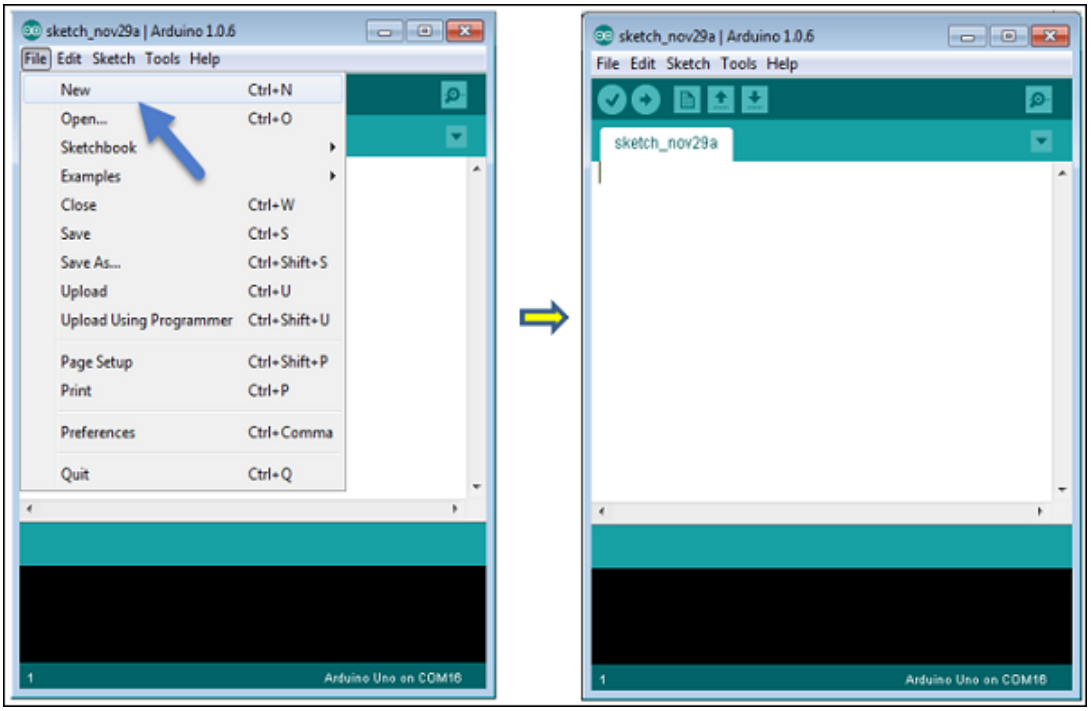


Step 4 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



Arduino Language –

Arduino programming language can be divided in three main parts: functions, values (variables and constants), and structure.

Functions

For controlling the Arduino board and performing computations.

Digital I/O
- digitalRead()
- digitalWrite()
- pinMode()

Analog I/O
- analogRead()
- analogReference()
- analogWrite()

Zero, Due & MKR Family
- analogReadResolution()
- analogWriteResolution()

Math
- abs()
- constrain()
- map()
- max()
- min()
- pow()
- sq()
- sqrt()

Trigonometry
- cos()
- sin()
- tan()

Random Numbers
- random()
- randomSeed()

Bits and Bytes
- bit()
- bitClear()
- bitRead()
- bitSet()
- bitWrite()
- highByte()
- lowByte()

Variables

Arduino data types and constants.

Constants
- HIGH | LOW
- INPUT | OUTPUT | INPUT_PULLUP
- LED_BUILTIN
- true | false
- Floating Point Constants

Data Types
- array
- bool
- boolean
- byte
- char

Variable Scope & Qualifiers
- const
- scope
- static
- volatile

Structure

The elements of Arduino (C++) code.

Sketch

loop()
setup()

Control Structure

break
continue
do...while
else
for
goto

Arithmetic Operators

% (remainder)
* (multiplication)
+ (addition)
- (subtraction)
/ (division)
= (assignment operator)

Comparison Operators

!= (not equal to)
< (less than)

Pointer Access Opera

& (reference operator)
* (dereference operato

Bitwise Operators

& (bitwise and)
<< (bitshift left)
>> (bitshift right)
^ (bitwise xor)
| (bitwise or)
~ (bitwise not)

EXPERIMENT NO – 1(c)

AIM - Simulation Software – Tinkercad

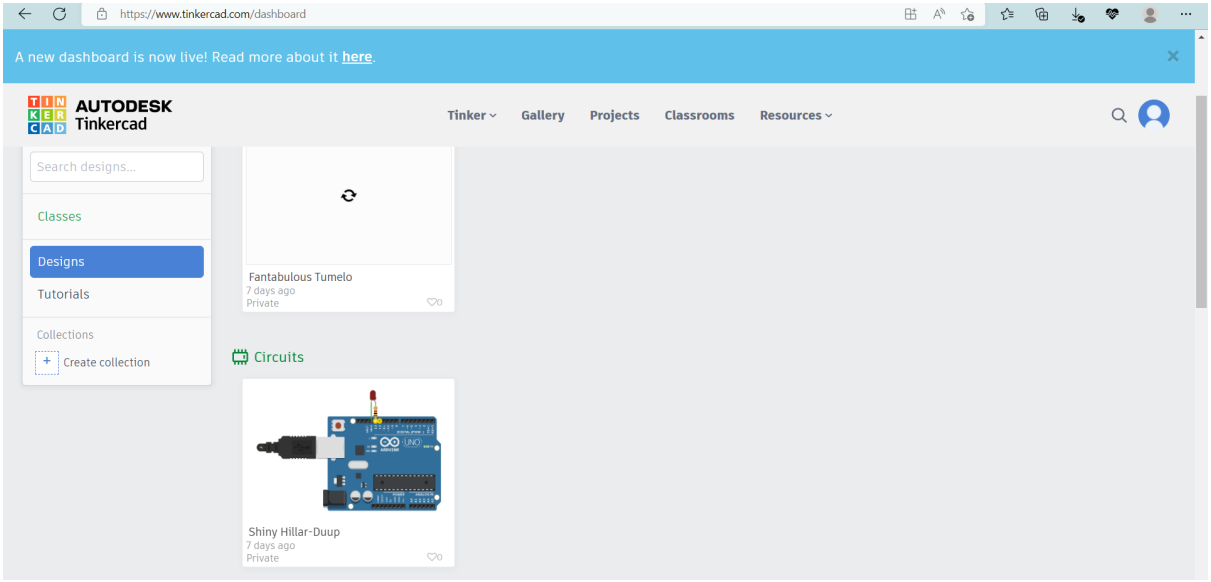
Simulation software's -

Companies that engage in e-commerce rely on software for a variety of things, from ensuring security, operating servers, and managing customer relationships to providing visitors with online shopping carts and payment systems. By definition, software consists of instructions that tell computers what to do. Although it's customary to view hardware and software as separate and distinct elements that work in tandem to make a computers operate, the two are actually enmeshed together at many different levels. Software normally is placed in one of two different categories: system software and applications. Applications are software programs that users apply to various tasks, ranging from enjoyment to productivity. One particularly useful kind of application allows companies to generate simulations—models or pictures of what might happen in different situations.

Tinkercad

Tinkercad is an online collection of software tools from Autodesk that enable complete beginners to create 3D models. This CAD software is based on constructive solid geometry (CSG), which allows users to create complex models by combining simpler objects together. As a result, this 3D modeling software is user-friendly and currently enjoyed by many, particularly teachers, kids, hobbyists, and designers. Best of all, it's free and you only need an internet connection to answer it. The software allows users to create models that are compatible with 3D printing, a great option for beginners to the technology.

Tinkercad is a good alternative to other 3D modeling software such as SketchUp or Fusion360—another solution from Autodesk—if you do not need the more advanced features of these solutions. Actually, Autodesk acquired Tinkercad in 2013, two years after it was launched by former Google engineer Kai Backman and his cofounder Mikko Mononen. The software's main advantage over the other two software is that it is free, while still offering more modeling freedom than what first meets the eye! It is currently available in 16 languages.



Main features of TinkerCAD are:

- File Exportation
- File Editing
- 3D Designs
- Circuits
- Codeblocks
- Presets

- **Tinkercad Benefits** –

The main benefits of TinkerCAD are low production costs, quick product completion, and user-friendly interface.

- **Low production costs**

TinkerCAD allows businesses to produce more products with fewer expenses. This software stores your product prototypes in the cloud, so you don't need to pay for large warehouses. Moreover, TinkerCAD enables you to allocate a lower budget on materials, since you can view and review your items' designs before proceeding with productions.

TinkerCAD also helps you to employ a smaller workforce since the software does most of the work. Machines are even the ones responsible for 3D printing and laser cutting. Since TinkerCAD is free software, there is no need for subscriptions. Hence, TinkerCAD makes your business more productive without exhausting your funds.

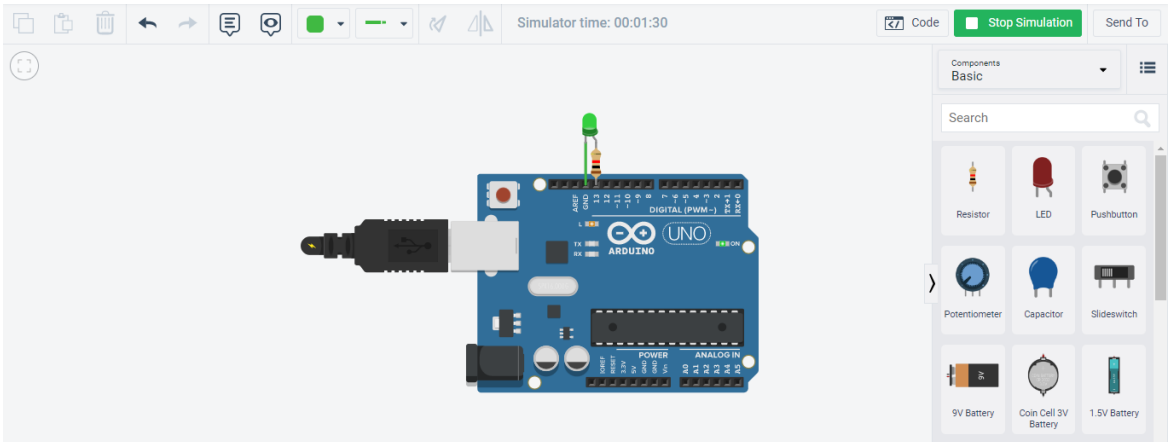
- **Quick product completion**

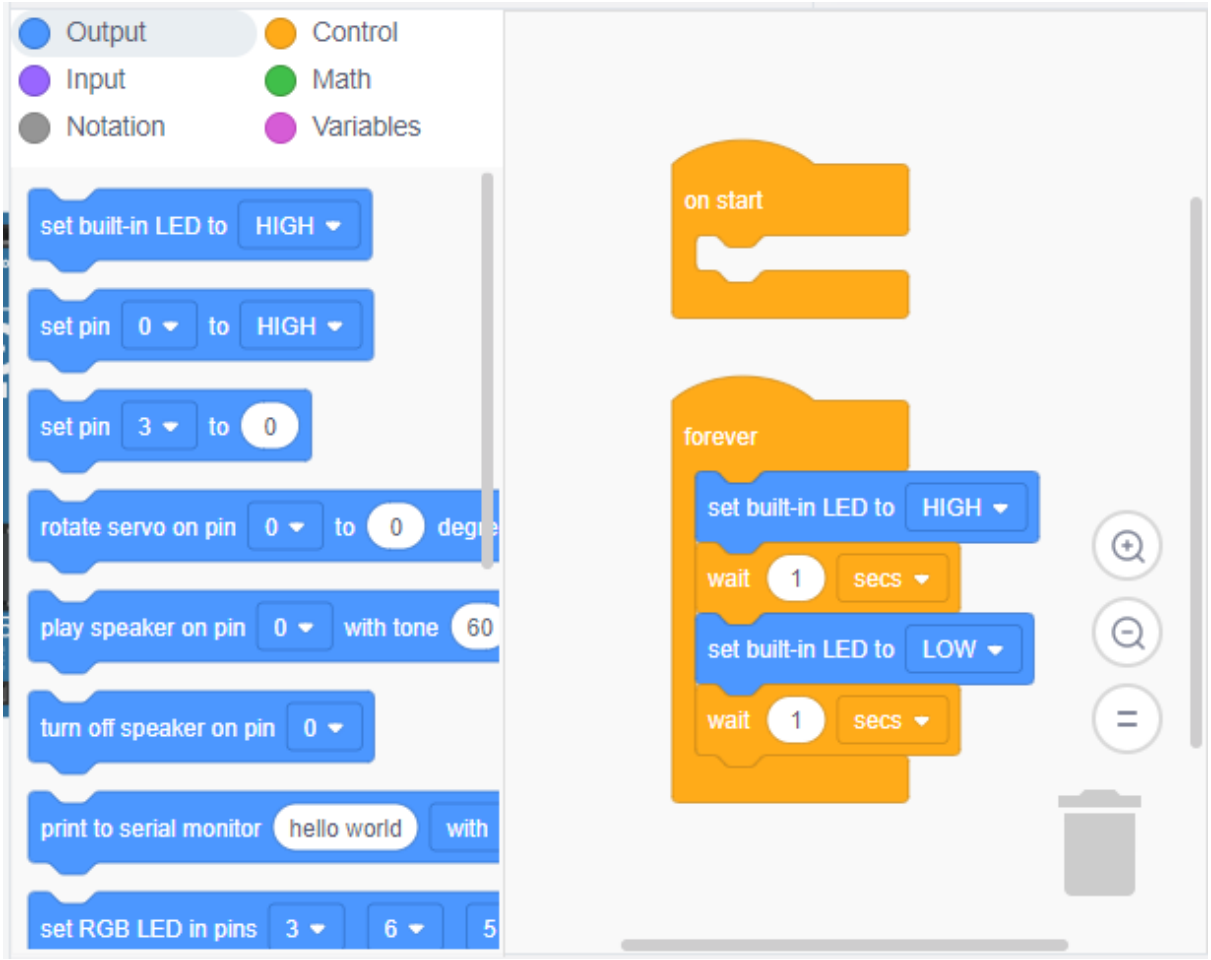
This CAD software allows you to review and enhance your products' designs. TinkerCAD enables you to produce quality prototypes due to its features, such as controllable work plane and accurate scale size. Thus, you can create quality models without compromise on quality.

TinkerCAD also connects you to 3D printing companies that can create real-life versions of your designs. You can also choose to ship the completed items to your company's address.

- **User-friendly interface**

This platform provides a user-friendly interface with options such as 3D design, circuits, and code blocks. TinkerCAD also allows for a 2D and 3D input, which permits you to integrate other models into your new plans. With TinkerCAD, you can view the design of your prototype from almost any angle. It also enables you to customize existing presets to create a unique design for your product.


Simulator time: 00:01:30
Code
Stop Simulation
Send To
Components
Basic
Search
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
9V Battery
Coin Cell 3V Battery
1.5V Battery
ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN


Output
Control
Input
Math
Notation
Variables
set built-in LED to HIGH
set pin 0 to HIGH
set pin 3 to 0
rotate servo on pin 0 to 0
play speaker on pin 0 with tone 60
turn off speaker on pin 0
print to serial monitor hello world with
set RGB LED in pins 3 6
on start
forever
set built-in LED to HIGH
wait 1 secs
set built-in LED to LOW
wait 1 secs

EXPERIMENT NO. – 2

AIM: Blink LED Light for Particular Duration.

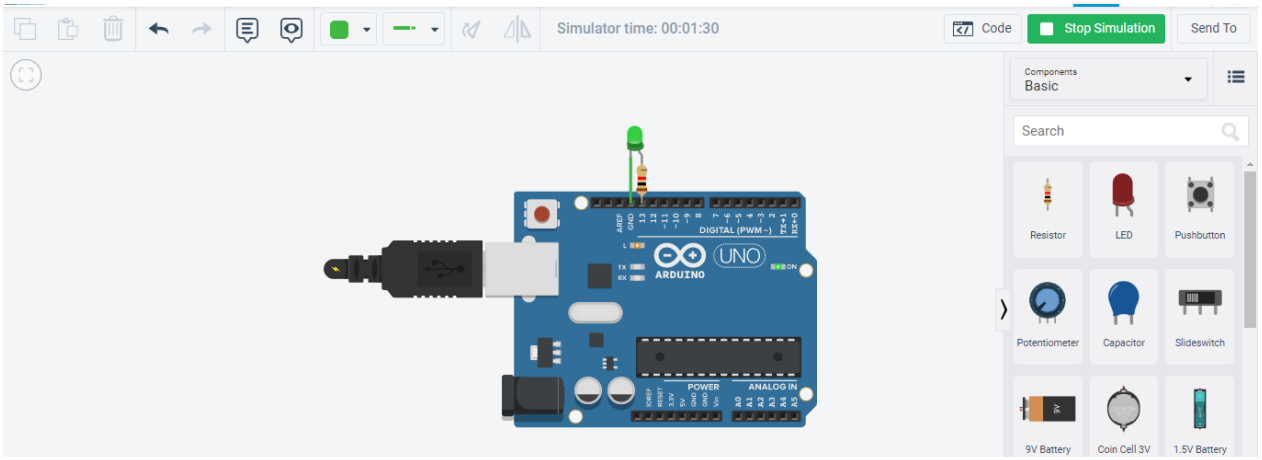
Hardware Required-

- LED
- Arduino Uno
- 220Ω resistor

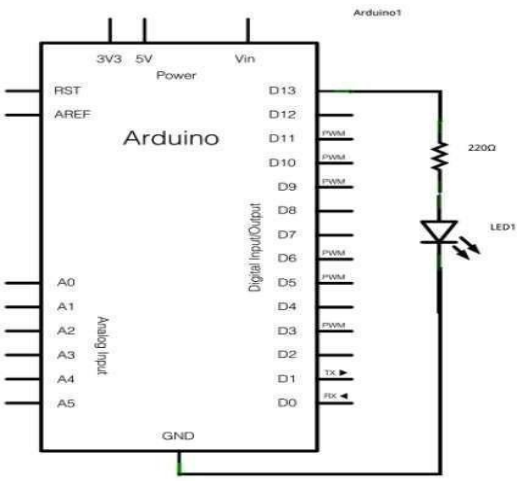
Circuit

The LED is connected to 13th pin of Arduino. Connect the long leg(+ve) of the LED to the other end of the resistor. Connect the short leg(-ve) of the LED to GND. The value of the resistor in series with the LED maybe of a different value than 220 ohms; the LED will lit up also with values up to 1K ohm.

Tinkercad –

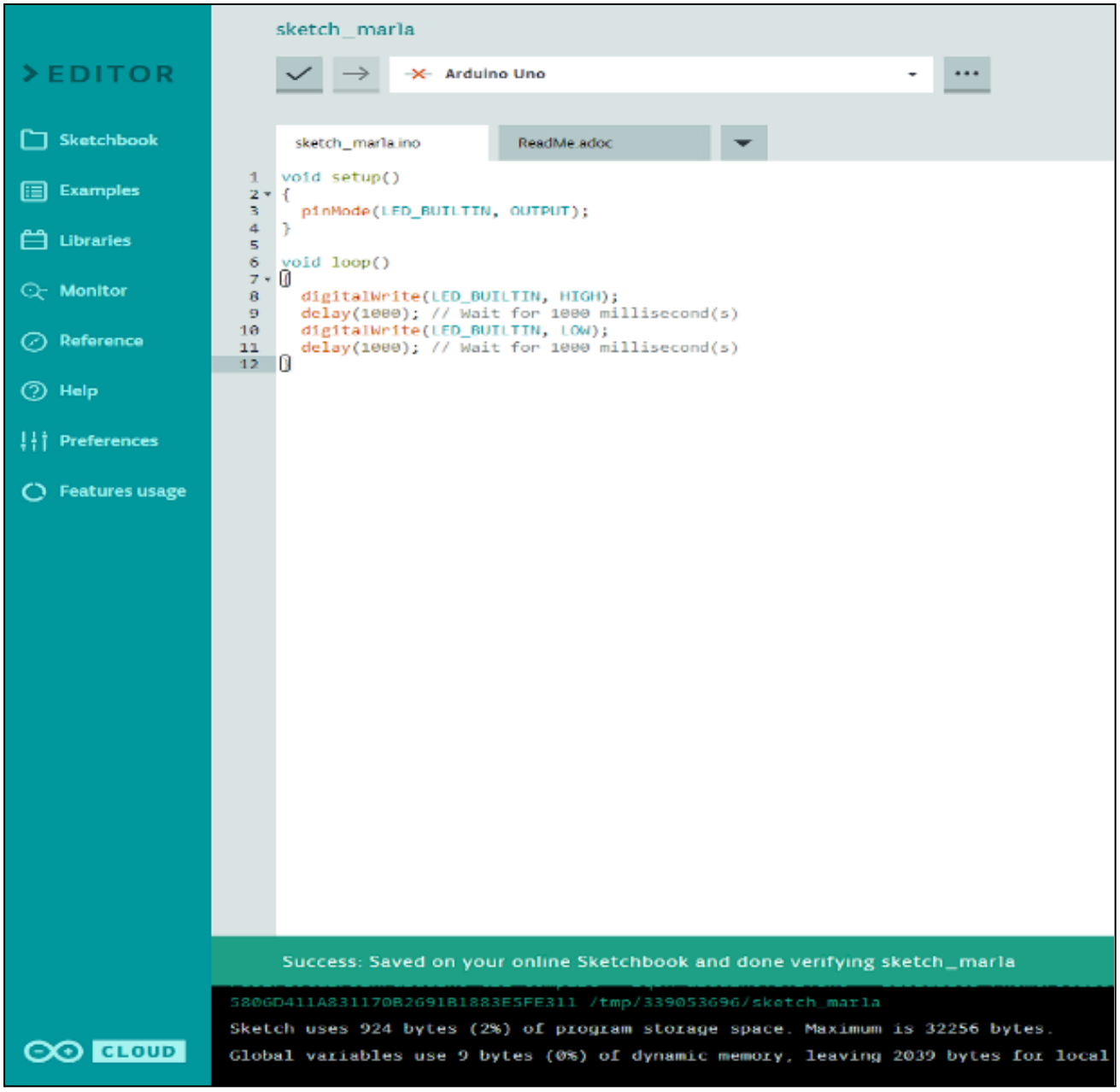


SCHEMATIC:



CODE:

```
// C++ code
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  delay(3000); // delay for 3 seconds
  digitalWrite(13, LOW);
  delay(3000); // delay for 3 seconds
}
```



Implementation -