

Project 1

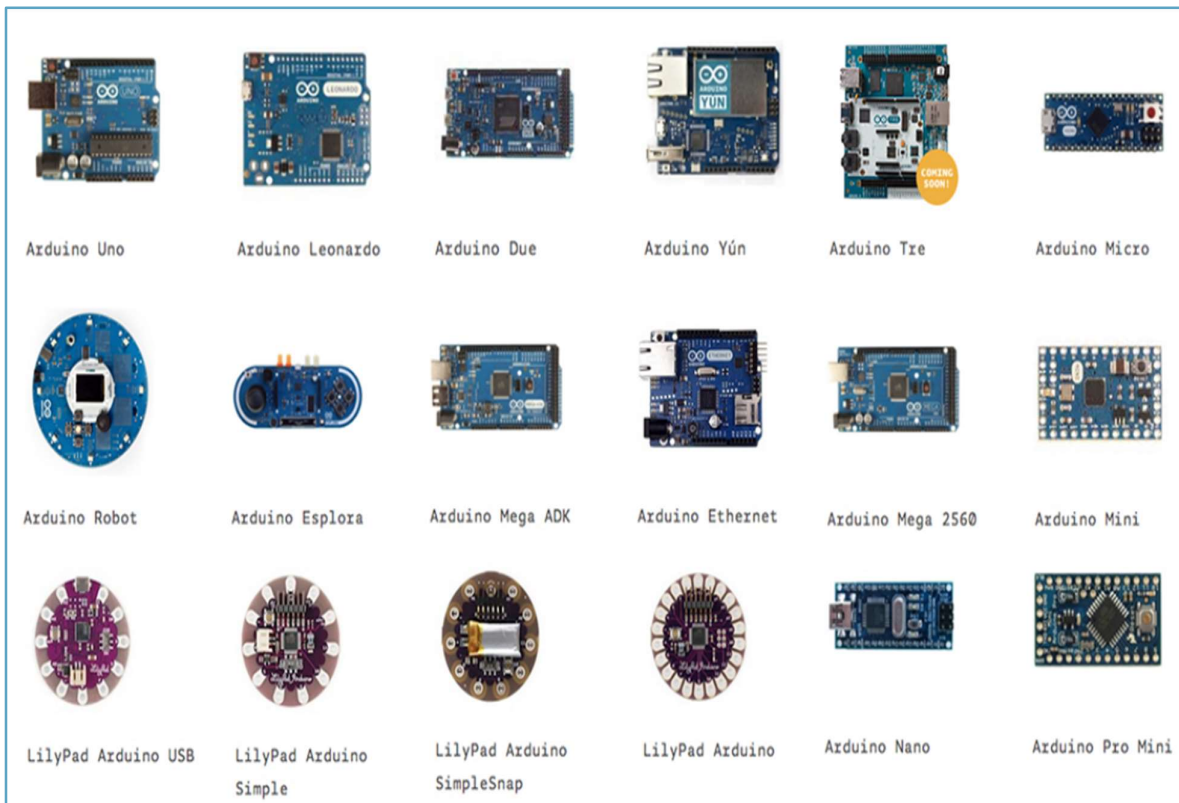
INTRODUCTION TO ARDUINO AND PROTEUS 8

Overview on Arduino Uno (Blinking A LED)

Overview on Arduino Uno

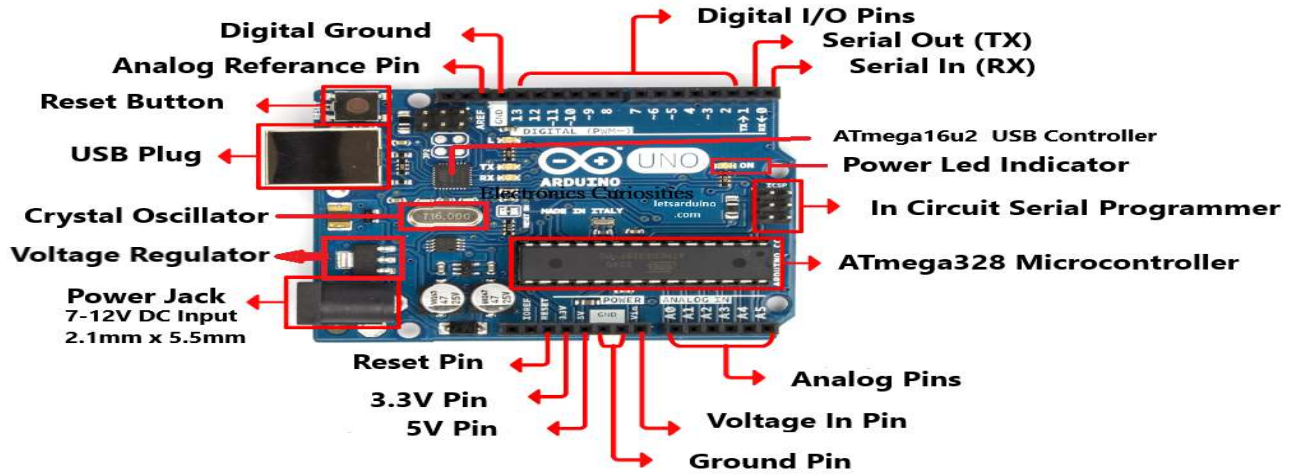
What is Arduino?

Arduino is an open source hardware prototyping platform designed around the popular (**ATMEL**) microcontroller family, and it includes a simple software development environment (**Arduino IDE**). Arduino bridges the computational world with the physical world, it can simply connect the sensors and actuators with a computer. Basically, you can write code to monitor and control various electronic components such as motors, thermostats, lights, switches, and many more. As Arduino is open source, there are a large number of compatible Arduino boards, just as there are many official Arduino models, with special functions as shown below. Arduino Uno is the most popular and used Arduino model.



Arduino Uno

Arduino Uno ("**Uno**" means **one in Italian**) is an **ATMEL** microcontroller board based on the **ATmega328P** (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button.

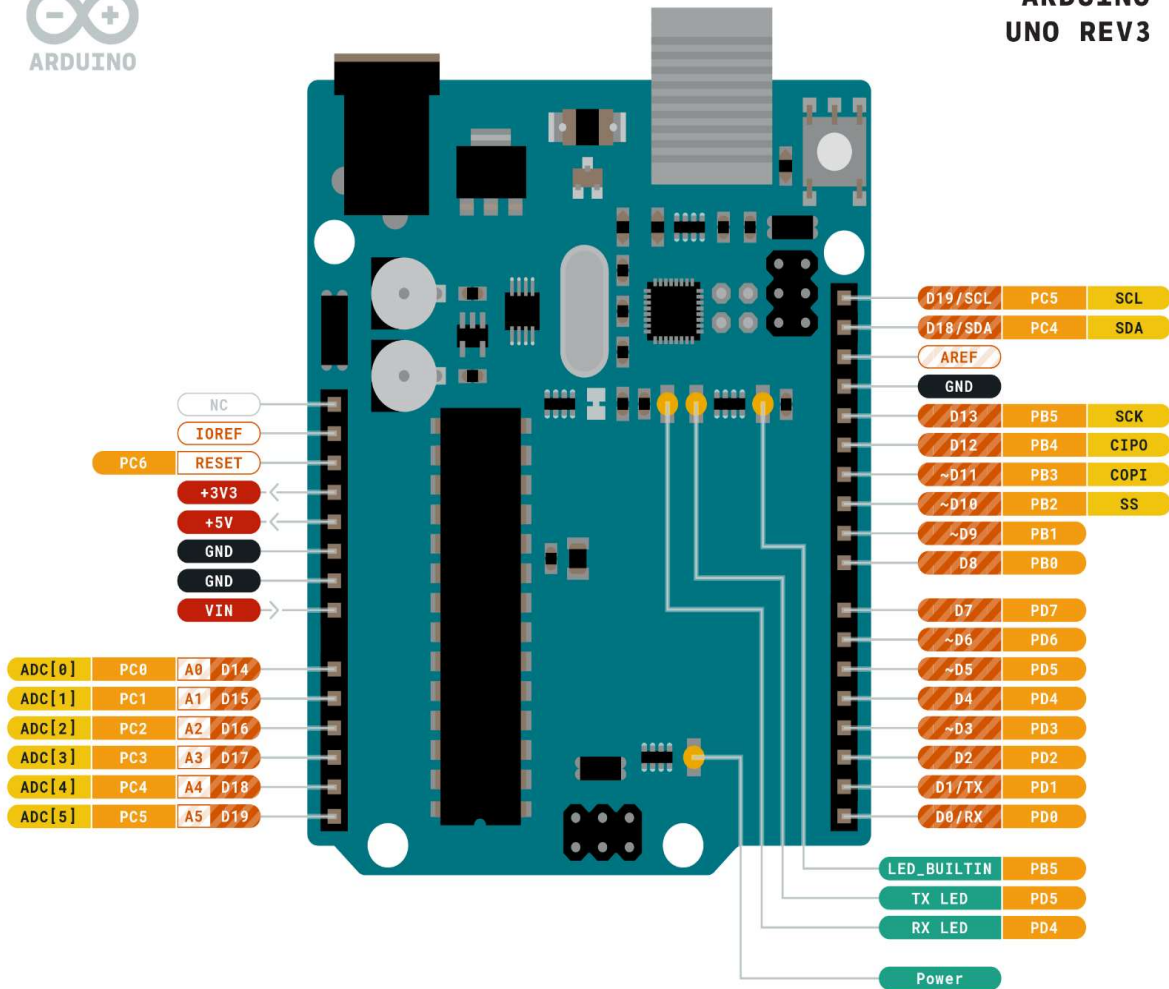


Tech specs

MICROCONTROLLER	ATmega328P
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMIT)	6-20V
DIGITAL I/O PINS	14 (of which 6 provide PWM output)
PWM DIGITAL I/O PINS	6
ANALOG INPUT PINS	6
DC CURRENT PER I/O PIN	20 mA
DC CURRENT FOR 3.3V PIN	50 mA
FLASH MEMORY	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
CLOCK SPEED	16 MHz
LED_BUILTIN	13
LENGTH	68.6 mm
WIDTH	53.4 mm
WEIGHT	25 g



ARDUINO UNO REV3



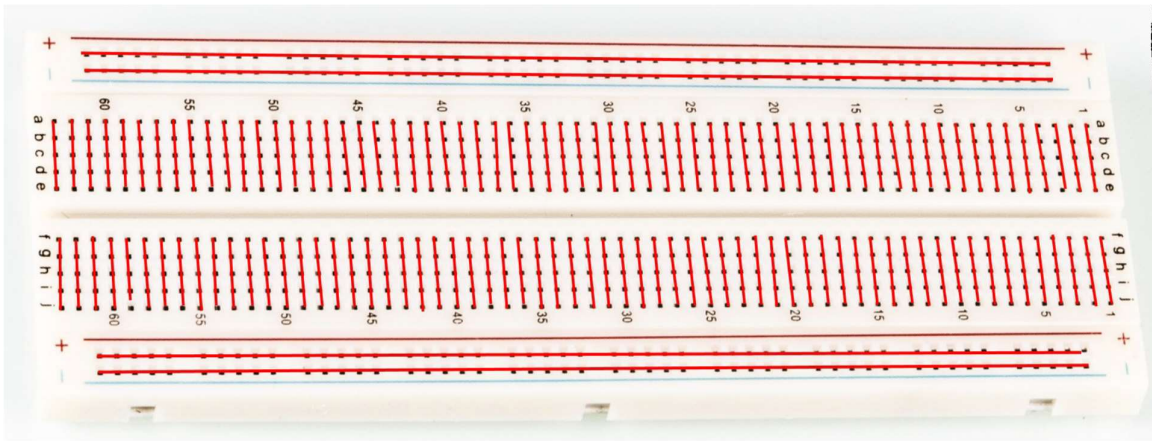
Ground	Internal Pin	Digital Pin	Microcontroller's Port
Power	SWD Pin	Analog Pin	
LED	Other Pin	Default	

ARDUINO.CC

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1886, Mountain View, CA 94042, USA.

Breadboard:

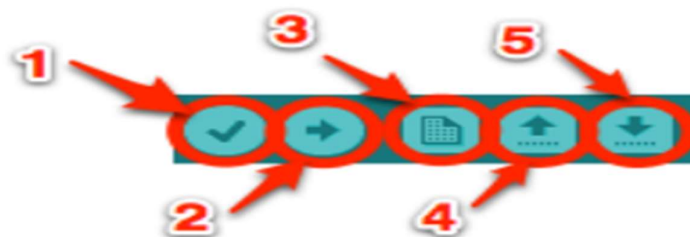
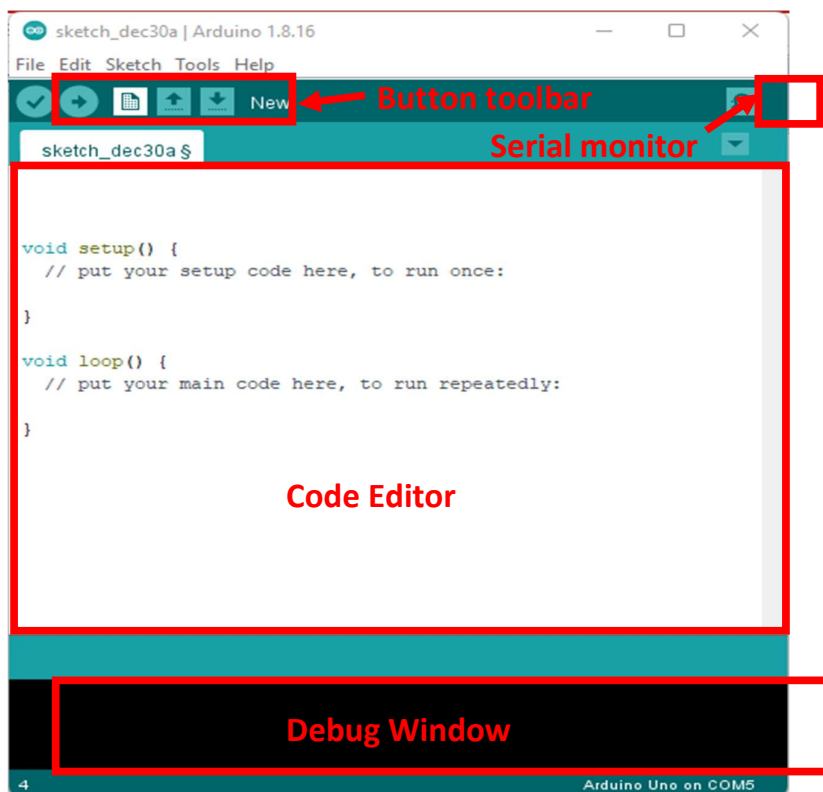
A Breadboard is a helpful tool to build circuits without any soldering. Certain contacts are connected with each other. Therefore, it is possible to connect many cables with each other without soldering or screwing them together. The image below shows in color, which contacts are connected.



Arduino IDE

Arduino IDE is a multi-platform (Mac, Windows and Linux) open-source software used to create, compile and send programs (Sketch) to the Arduino Board. The IDE can be downloaded from the following address: <http://arduino.cc>.

Arduino IDE Interface:



Button 1: verify (compile) Sketch for possible errors.

Button 2: Upload (send) Sketch to the Arduino Board.

Button 3: create a new Sketch.

Button 4: open an existing Sketch.

Button 5: Save Sketch.

Programing Structure of an Arduino Sketch

The structure of an Arduino sketch consists of 3 main sections as explained below:

The screenshot shows the Arduino IDE interface with a sketch named 'Sweep \$'. The code is as follows:

```
#include <Servo.h>

Servo myservo;

int pos = 0;

void setup() {
  myservo.attach(9);
  pinMode(10, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) {
    // in steps of 1 degree
    myservo.write(pos);
    delay(15);
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    myservo.write(pos);
    delay(15);
  }
}
```

1. Declaration Section
This section used for declare variables, constants, Pin Numbers and to include Libraries (C++ header files).

1. Setup Section
In this his section, the setup () function is used to initialize serials, configure Pin mode, Pin Numbers, using Libraries ...etc.
This function will only run once after each power up or reset of the Arduino board.

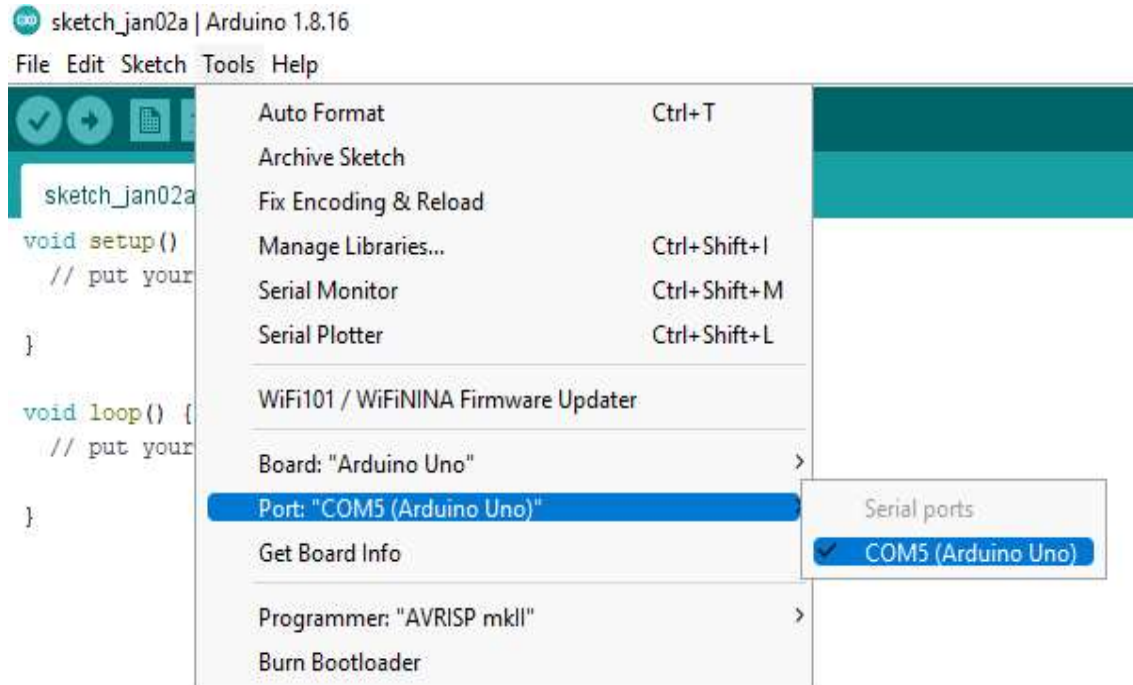
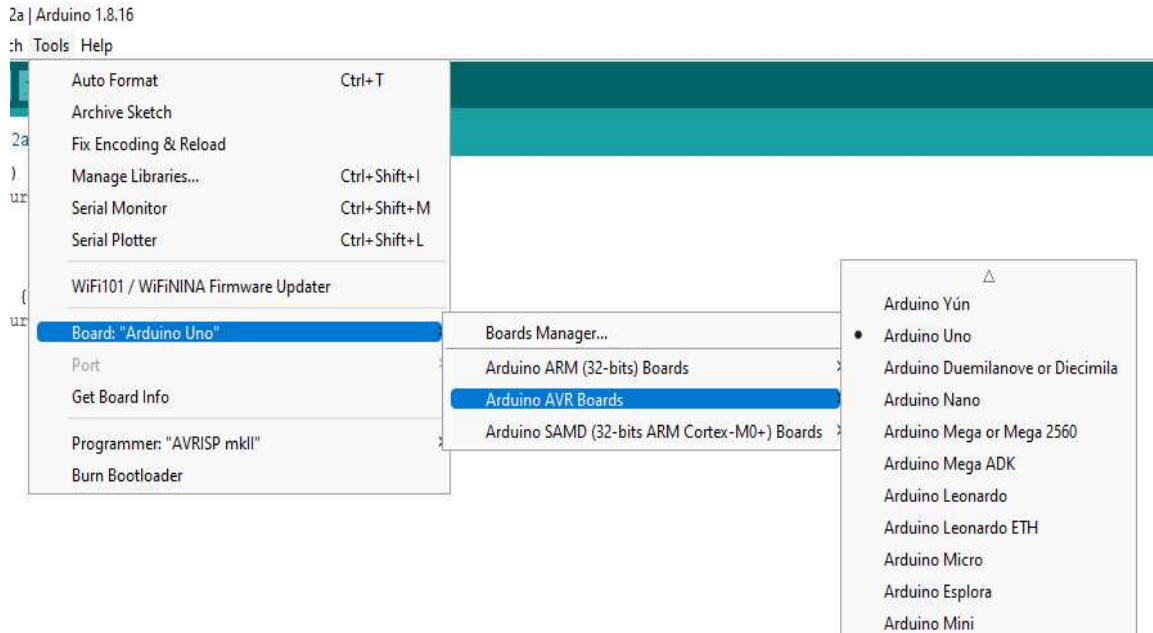
1. Loop Section
The main program of the sketch will runs in this section.
The Loop () Function will run infinitely.

12 Arduino Uno on COM5

Upload a sketch to Arduino Board

After writing and compiling the code, it can be uploaded to the Arduino board as follow:

First select the right Arduino board from the menu toolbar: **Tools > Board > Arduino AVR Boards > Arduino UNO**, then select the right serial port as: **Tools > Port > COMx**. After that the code can be uploaded using the upload button.

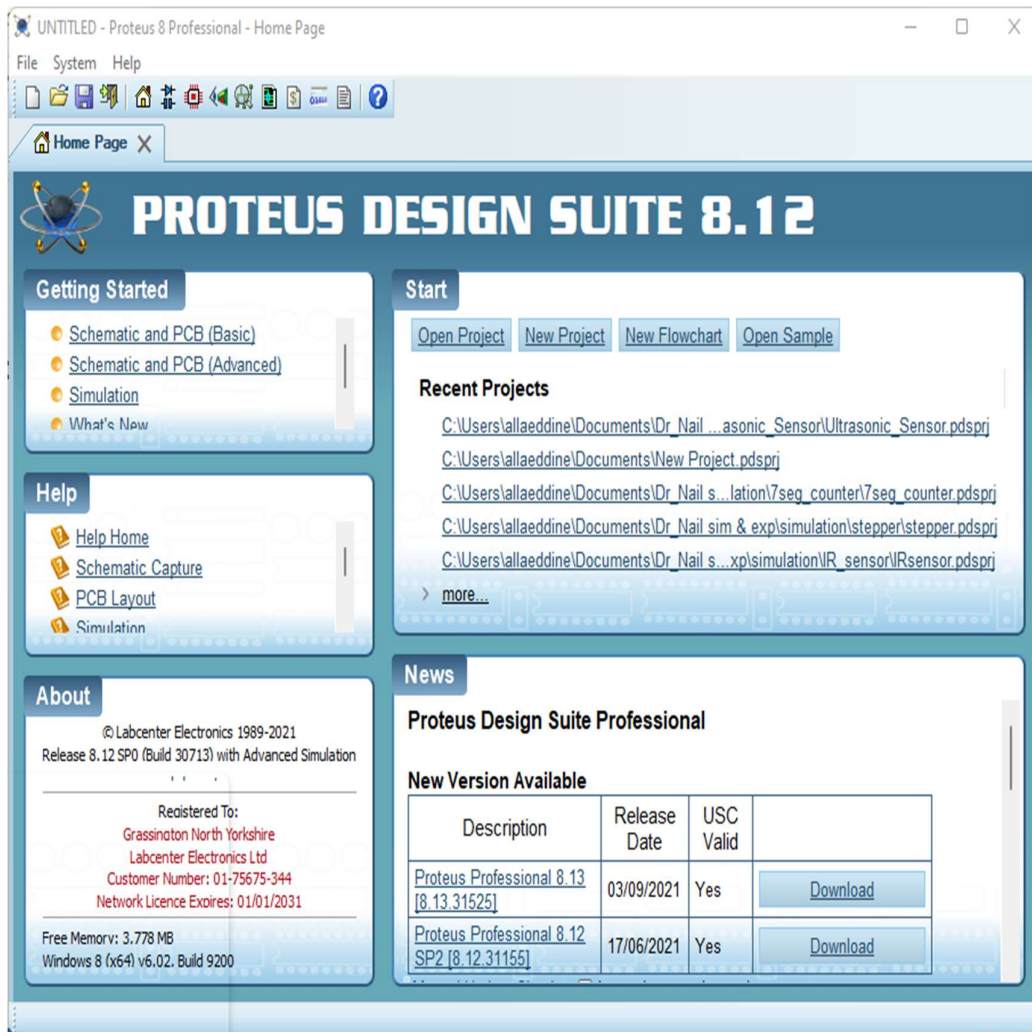


What is Proteus?

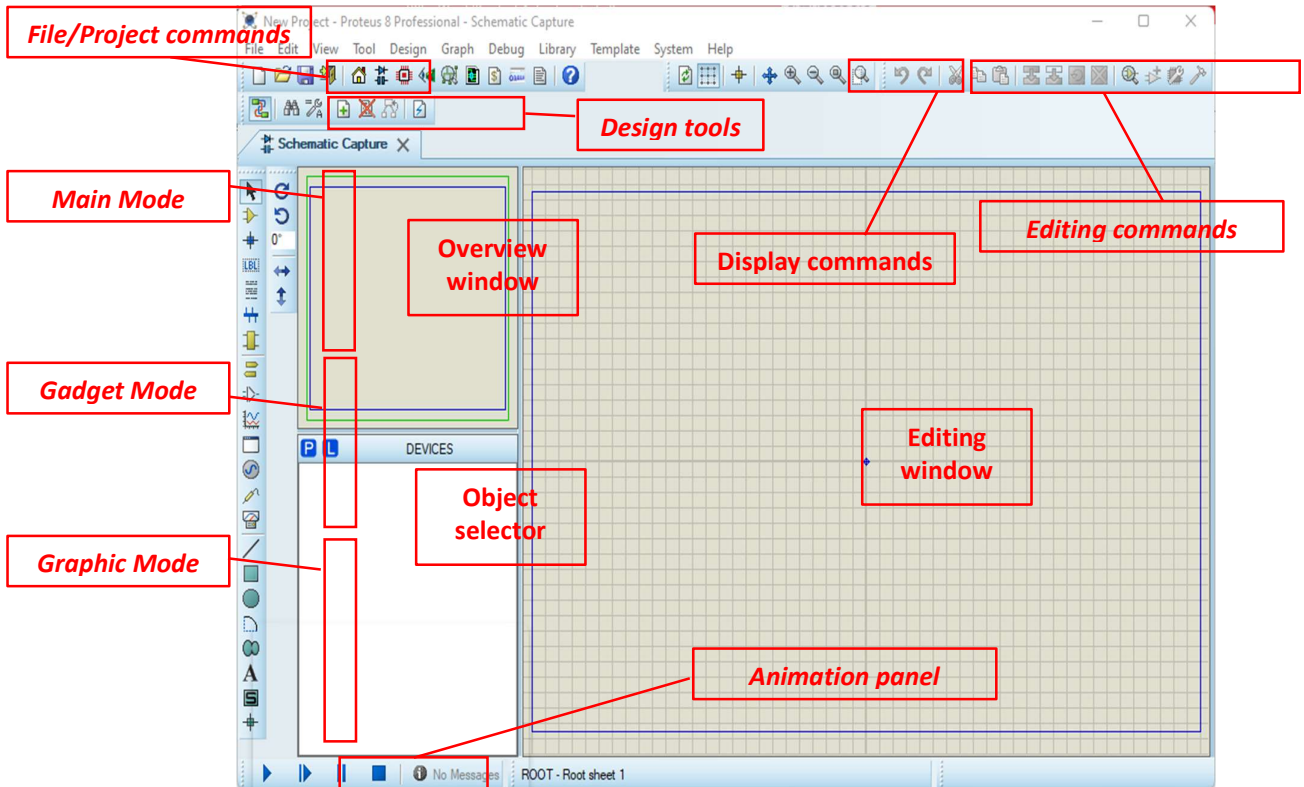
Proteus Design Suite (designed by **Labcenter Electronics Ltd.**) is a software tool set, mainly used for creating schematics, simulating Electronics, Embedded Circuits, Microcontrollers, and designing PCB (**Print Circuit Board**) Layouts.

About Proteus

- 1) It is a software suite containing schematic, simulation as well as PCB designing.
- 2) **ISIS** is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.
- 3) **ARES** is used for **PCB** designing. It has the feature of viewing output in 3D view of the designed PCB along with components.



Proteus ISIS Interface



PROJECT 1: Blinking A LED

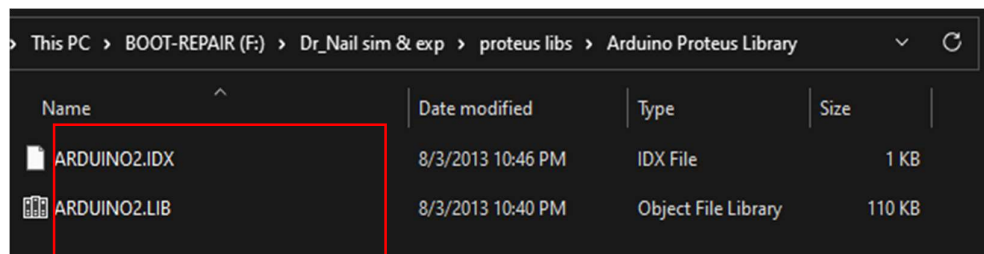
Objective: light up the LED for 1 second, then turn it off for another 1 second.

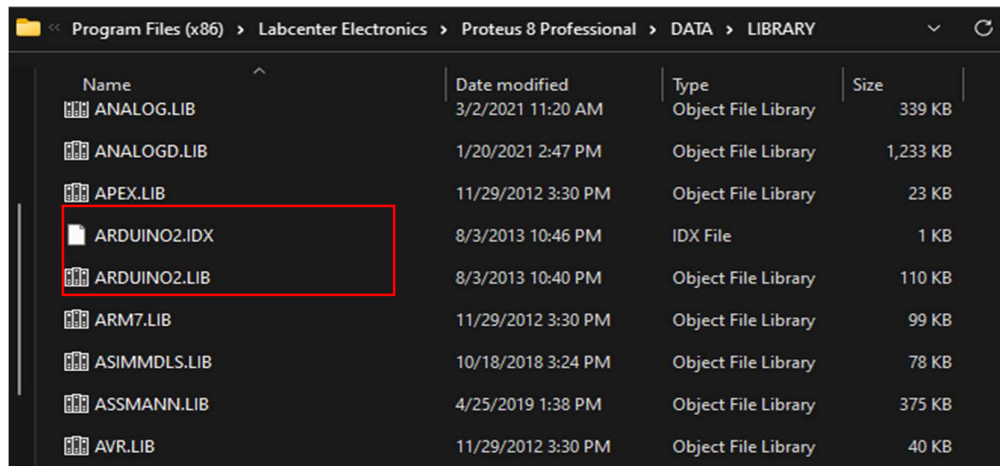
Required components: Arduino Uno, LED, 220ohm Resistor, bread board, 2 wires (jumper cables).

Simulation using Proteus


Install Arduino Library on Proteus

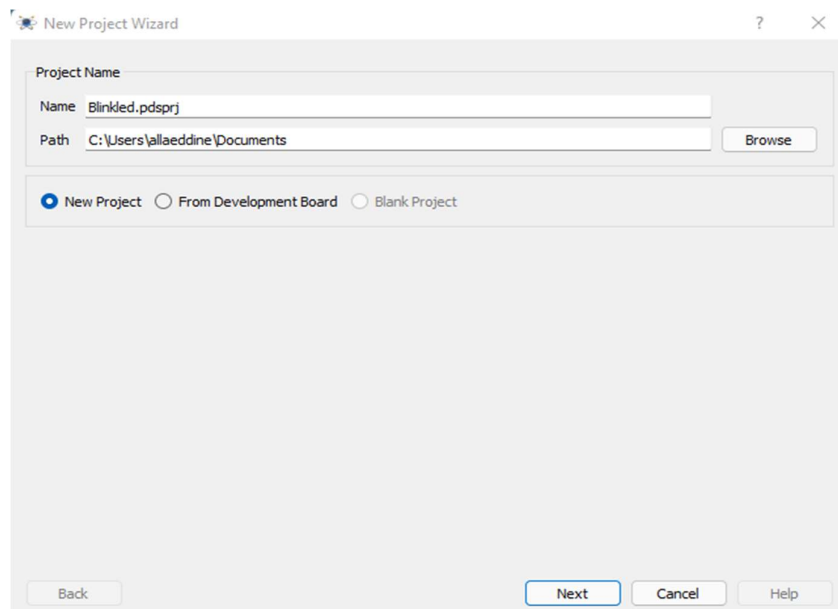
After downloading the Arduino Libraries extract the files, the copy the (*.LIB, *.IDX) files to the following folder **C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\DATA\LIBRARY**



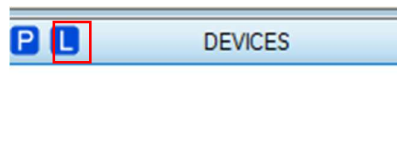


Building the Circuit

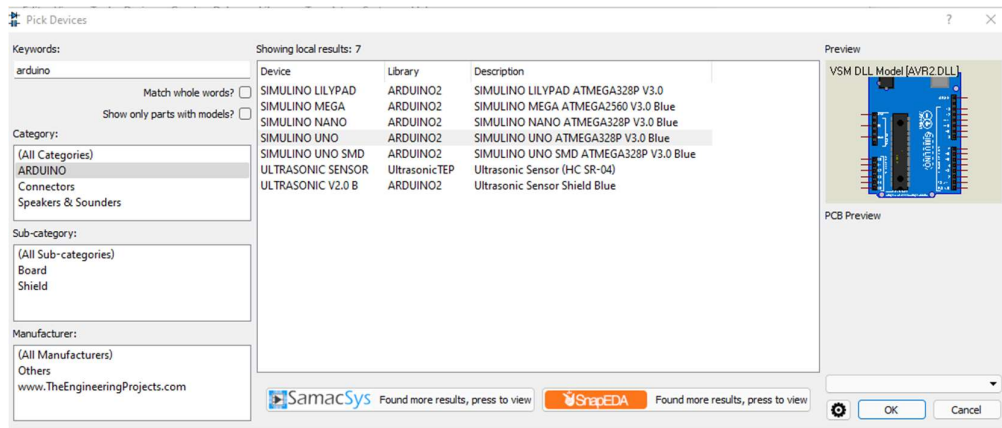
After installing Arduino libraries, open the Proteus software then click on  to start a new project a window form will shows up, select the name of the project then click on **next > next > next > finish**.



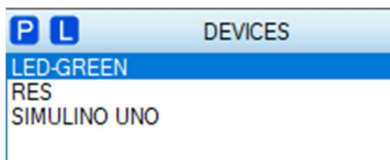
Now, to insert the circuit components click on **P** the **pickdevice** in the **Object Selector** area.



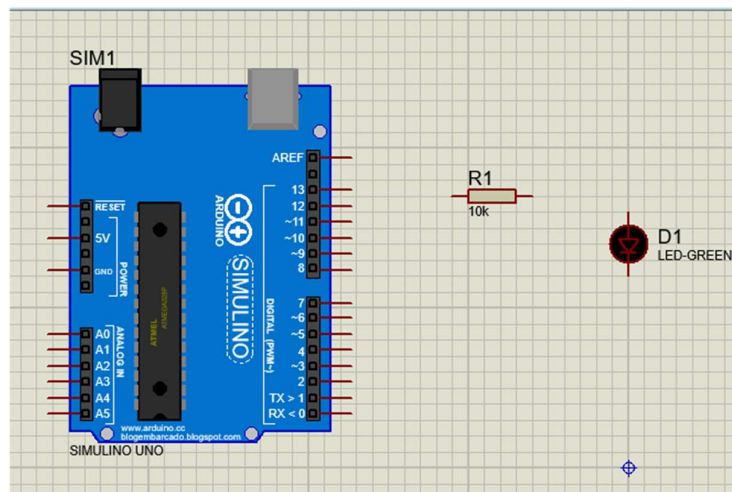
The following window will appears, then on the **keywords** field search for Arduino, select **SIMULINO UNO** double click to add it to the **Object Selector**.




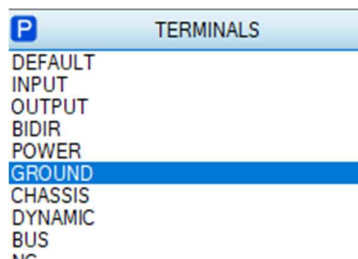
Using the same steps to add Resistor and LED Green components.



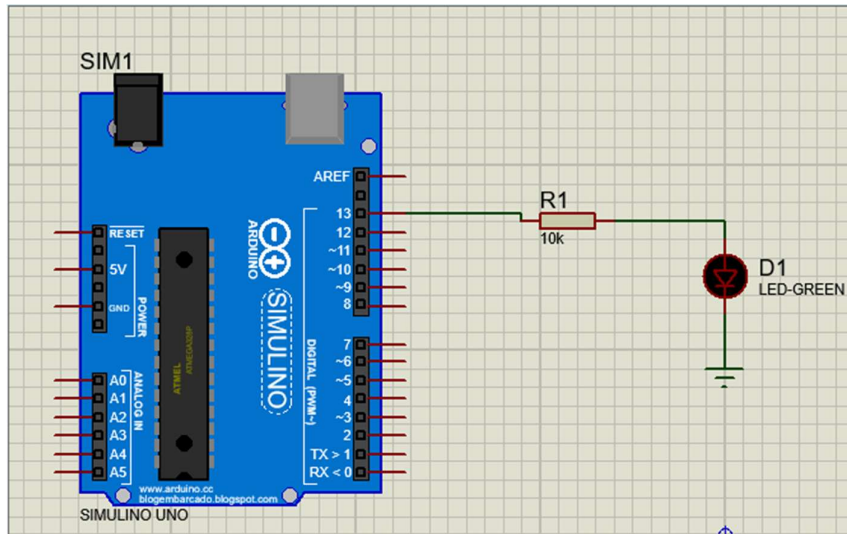
Using the mouse left click select the component from **Object Selector** list and add to the **Editing Window**.



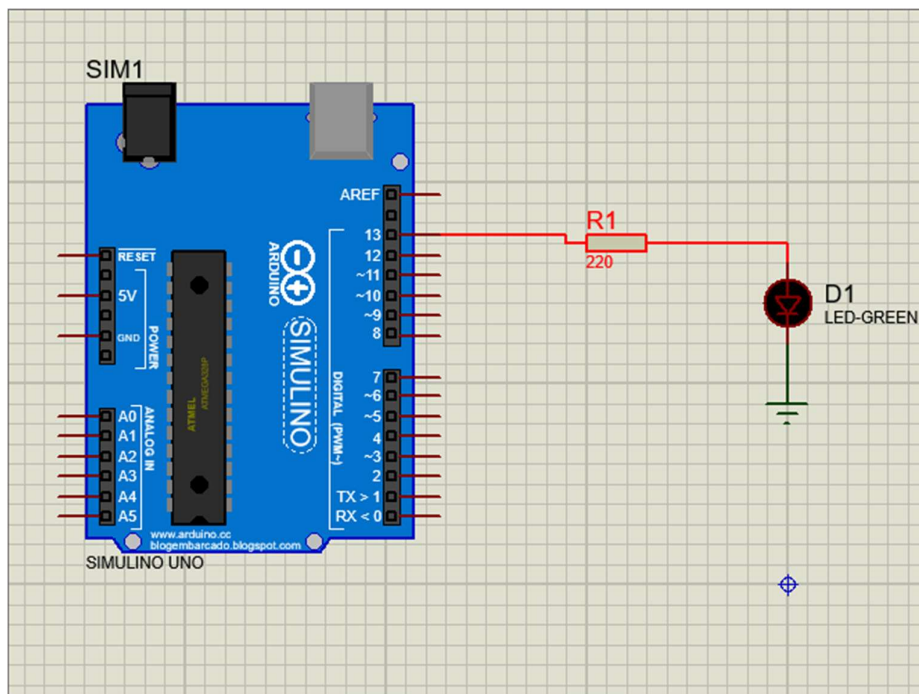
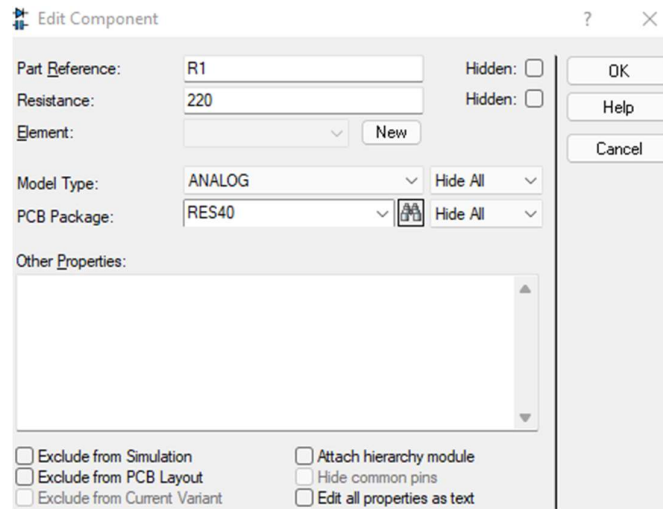
Click on  then select the **Ground** from the **Object Selector**, left click on the **Ground** to add to **Editing window**. Use the mouse to create connections (wires) between the components.



The circuit schematic should look like the image below



Double click on the resistor device and change its value to 220ohm.



Arduino Code

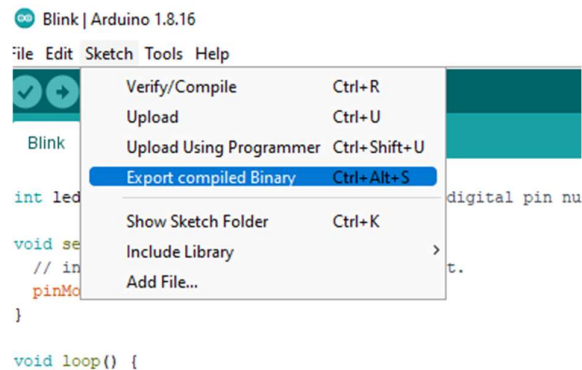
Open the Arduino IDE and write the following code:

```
int led13 = 13; //define a variable with the digital pin number


void setup() {
  // initialize digital pin led13 as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

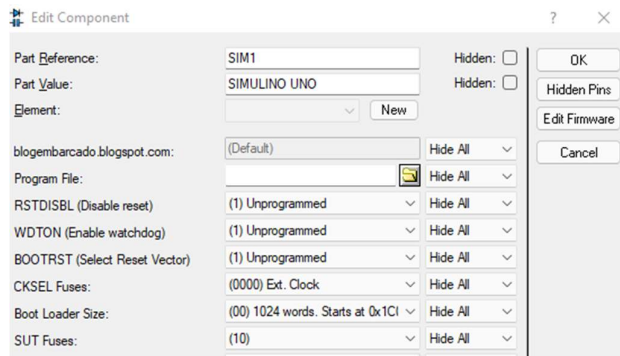
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Click on the compile button to verify the code. Save the code into *.INO file, then on the Arduino IDE navigate to **Sketch > Export compiled Binary** to extract the *.HEX that will be used for simulation on the Proteus environment.

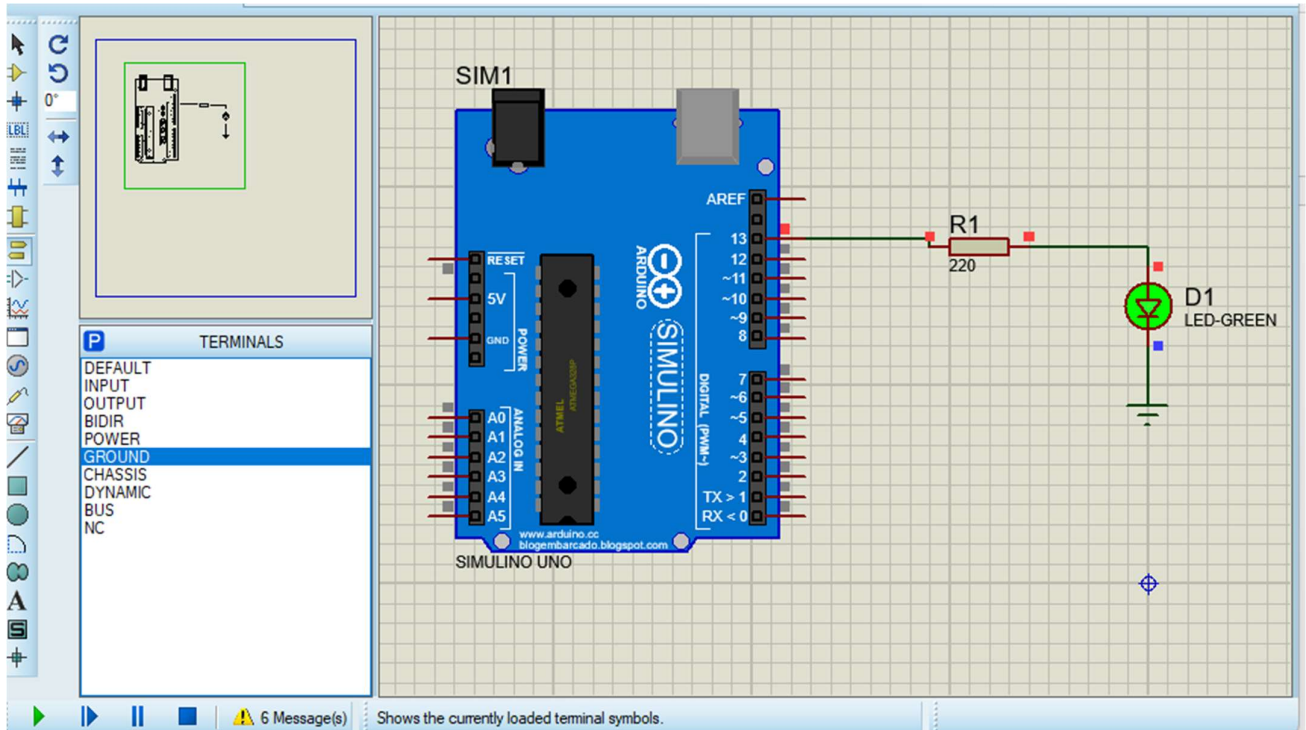


Name	Date modified	Type	Size
Blink	1/3/2022 12:31 PM	Arduino file	1 KB
Blink.ino.standard.hex	1/3/2022 12:32 PM	HEX File	3 KB
Blink.ino.with_bootloader.standard.hex	1/3/2022 12:32 PM	HEX File	4 KB

Go back to the Proteus Simulator, double click on the Arduino board then select the **Program File** field and click on  to select the extracted Arduino code *.Hex file. Then click on the **ok** Button.

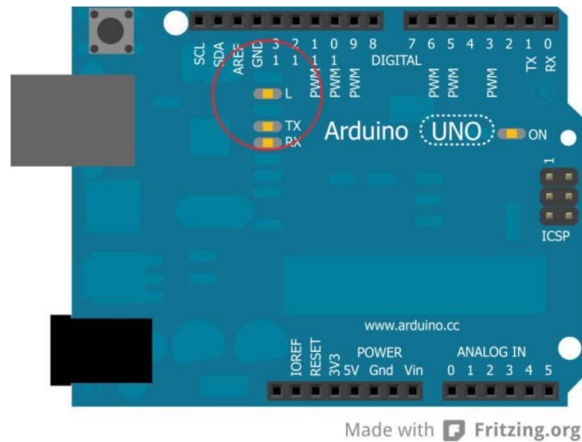


To run the simulation click on  in the **animation panel**.

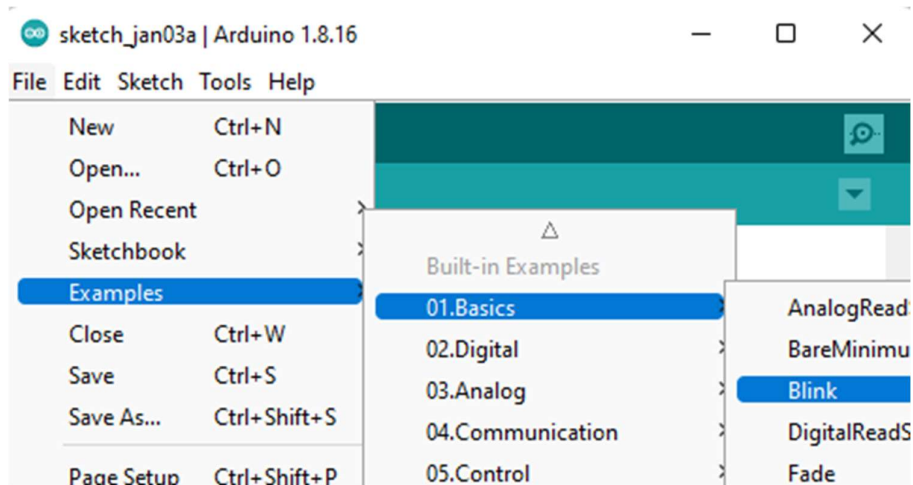


Implementation part

Blink the Arduino Built-In LED

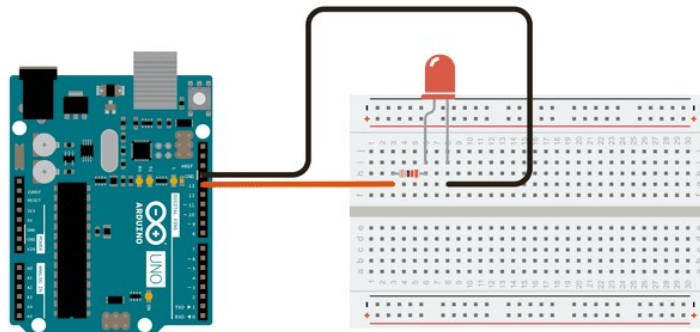


Connect the Arduino board through a USB cable to the computer. Then open the Arduino IDE, navigate to **File > Examples > Basics > Blink**. Then use upload Button to send the code to the Arduino Board.



Blinking LED using Arduino

Use the equipment's to implement the following circuit:



Connect the Arduino board to the computer then open up the Arduino IDE, compile the code used in the simulation then use the upload button to send it to Arduino.