DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE
(AUTONOMOUS)
(Approved by AICTE & Affiliated to Anna University, Chennai)
Accredited with 'A' Grade by NAAC, Accredited by TCS
Accredited by NBA with BME, ECE & EEE
PERAMBALUR - 621 212. Tamil Nadu.
website : www.dsengg.ac.in

# U20CS704
# INTERNET OF THINGS LABORATORY

**B.E/B.Tech Degree Practical Examination**

**Name** ……

**Branch** ……

**Semester** ……

**Reg.No.** ……

**Academic Year** ……

# DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE & Affiliated to Anna University, Chennai)
Accredited with 'A' Grade by NAAC, Accredited by TCS
Accredited by NBA with BME, ECE & EEE
**PERAMBALUR - 621 212. Tamil Nadu.**
website : www.dsengg.ac.in

**Name** ……

**Reg.No.** ……

**Sem. /Branch** ……

**Sub.Code/Subject** ……

Certified that this is the Bonafide record of the Practical done for the above subject in the Laboratory during the period

……

**Staff in Charge** **Head of the Department**

Submitted for the University Practical Examination held at **DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE (AUTONOMOUS)**, PERAMBALUR on……

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

# TABLE OF CONTENTS

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**Vision and Mission of the Department:**

**Vision**

To produce globally competent, socially responsible professionals in the field of Computer Science and Engineering.

**Mission**

M1: Impart high quality experiential learning to get expertise in modern software tools

M2: Inculcate industry exposure and build inter disciplinary research skills.

M3: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M4: Acquire Innovative skills and promote lifelong learning with a sense of societal and ethical responsibilities

**State the Program Educational Objectives (PEOs)**

**PEO 1:** Graduates of the programme will develop proficiency in identifying, formulating, and resolving complex computing problems.

**PEO 2:** Graduates of the programme will achieve successful careers in the field of computer science and engineering, pursue advanced degrees, or demonstrate entrepreneurial success.

**PEO 3**: Graduates of the programme will cultivate effective communication skills, teamwork abilities, ethical values, and leadership qualities for professional engagement in industry and research organizations.

**EXP : 1**

**Date:** **Arduino Installation**

---

**Aim:**

Getting Started with IoT (Arduino) and perform necessary software installation

**Arduino:**

- Arduino is a platform that makes it easy for you to build projects using electronics.
- IoT is a way of using electronics - to make electronic modules talk to each other remotely andwirelessly (often using a Cloud) to solve problems.
- Now, Arduino can also help you easily build IoT projects in two ways: Using traditional Arduino boards and attaching communication breakout modules (like nRF, Bluetooth,WiFi,LoRA,GSM, etc) to them.

- Arduino is a micro controller that can be connected to one or more sensors and help you capture the data or information and then pass it on to processor. If you know the full stack of IoT then you shouldalso look at Raspberry.

- RaspPi is a microprocessor so the basic difference between Arduino and RasPi is that RaspPi is controller plus processor and Arduino is just a micro controller.

- They suit the need for different use cases. You can easily read online about this both.

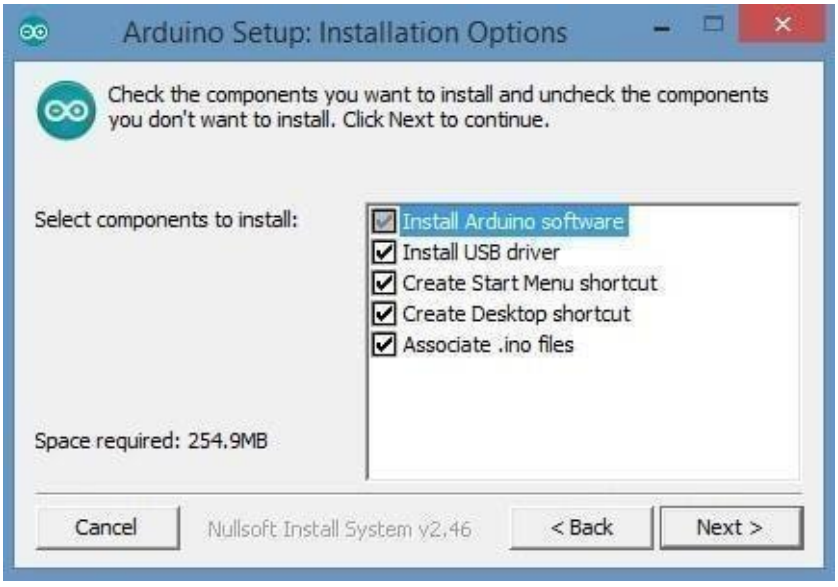
**Download and install the Arduino software (Arduino IDE 1.8.15)**

- Go to the Arduino website and click the download link to go to the download page.
- After downloading, locate the downloaded file on the computer and extract the folder from thedownload zipped file. Copy the folder to a suitable place such as your desktop.

# INSTALLING THE ARDUINO IDE ONWindows PCs

1. Visit http://www.arduino.cc/en/main/software to download the latest Arduino IDE version for your computer's operating system. There are versions for Windows, Mac, and Linux systems. At the download page, click on the "Windows Installer" option for the easiest installation.
2. Save the .exe file to your hard drive.
3. Open the .exe file.
4. Click the button to agree to the licensing agreement:

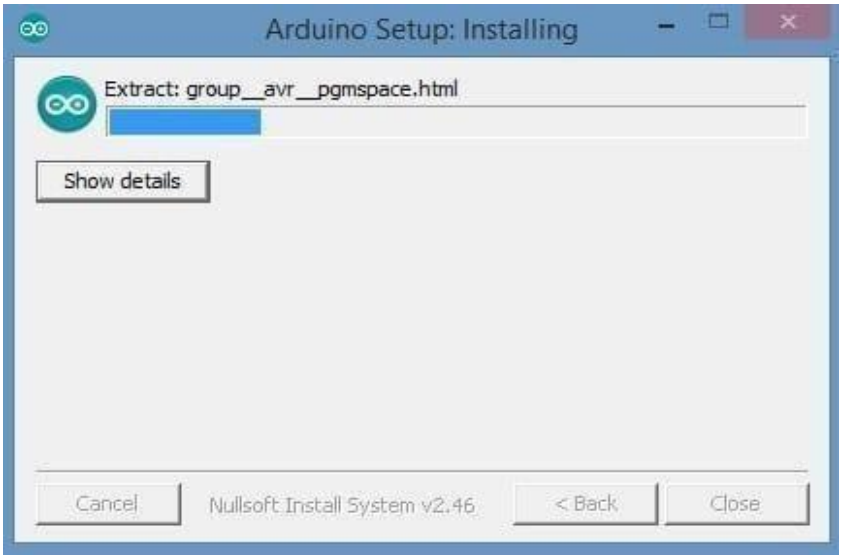

5. Decide which components to install, and then click "Next":

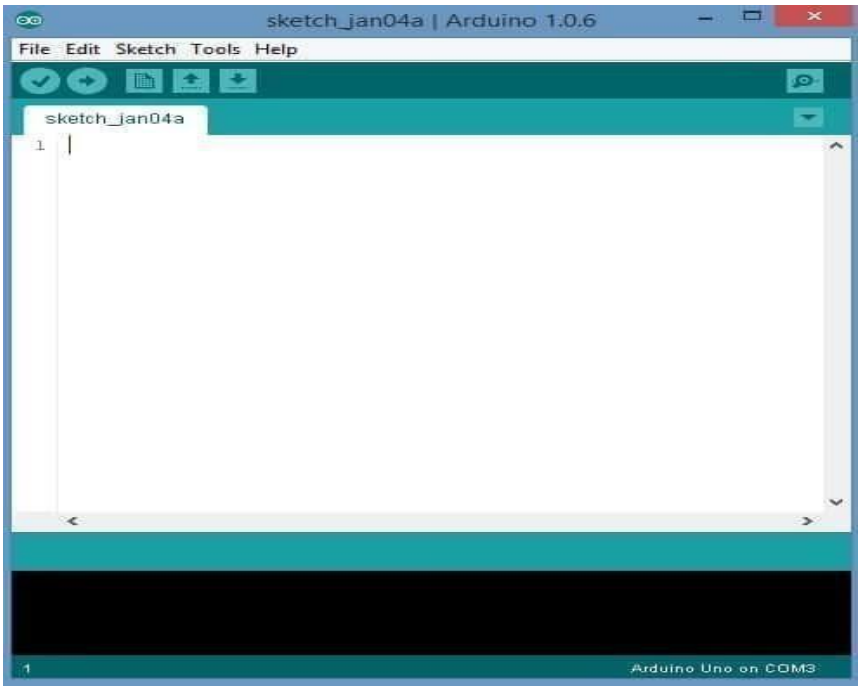


6. Select which folder to install the program to, then click "Install":


7. Wait for the program to finish installing, and then click "Close":

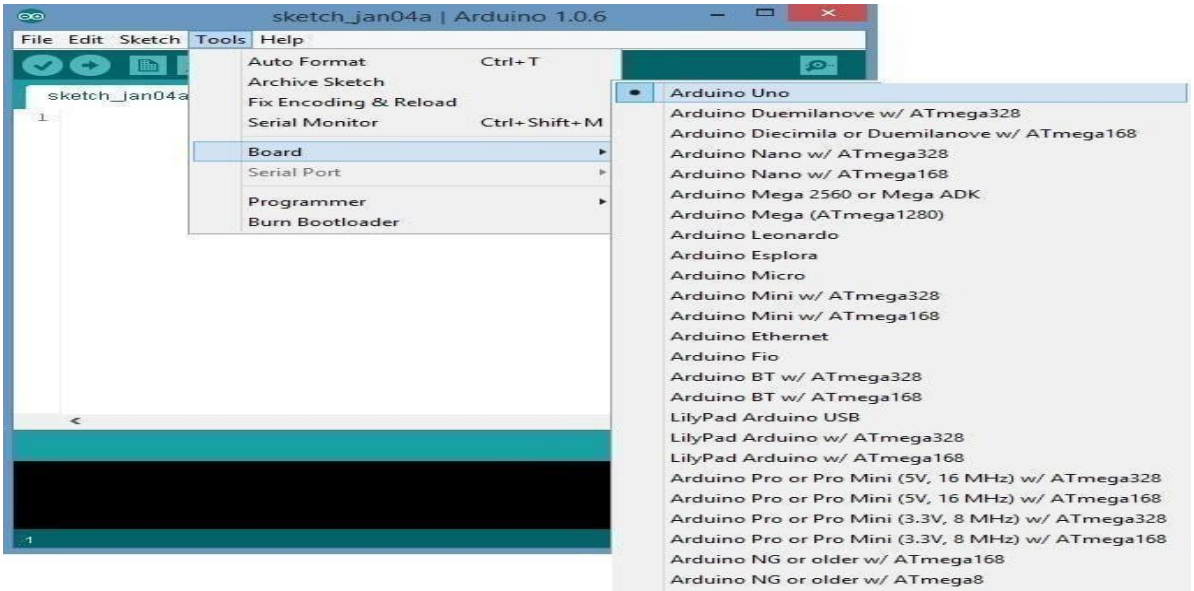
8. Now find the Arduino shortcut on your Desktop and click on it. The IDE will open up and you'll seethecode editor:





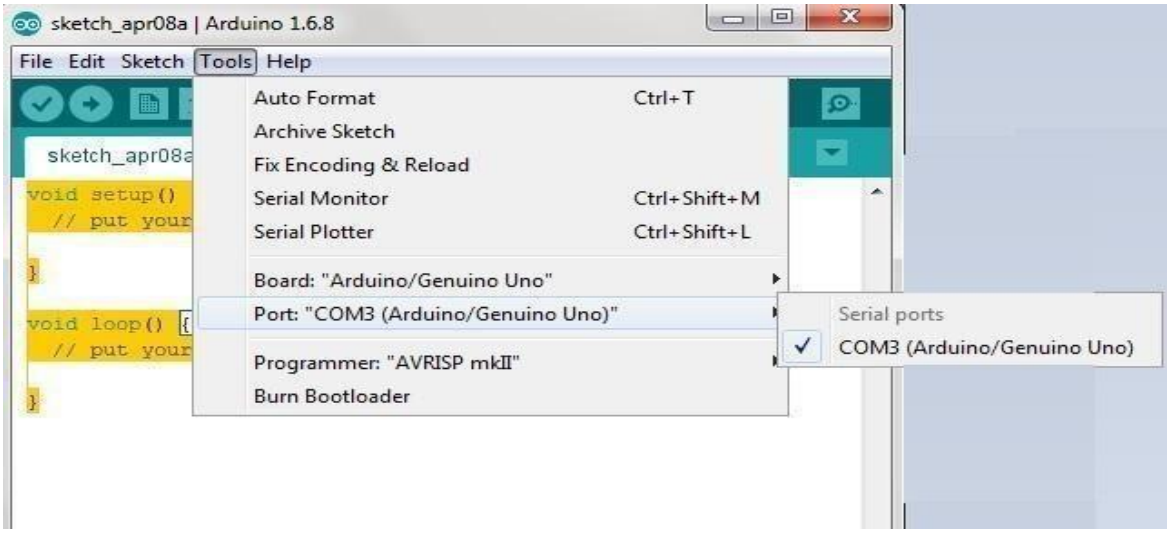

## CONFIGURING THE ARDUINO IDE

The next thing to do is to make sure the software is set up for your particular Arduino board. Go to the "Tools" drop-down menu, and find "Board". Another menu will appear where you can select from a list of Arduino models. I have the Arduino Uno R3, so I chose "Arduino Uno".



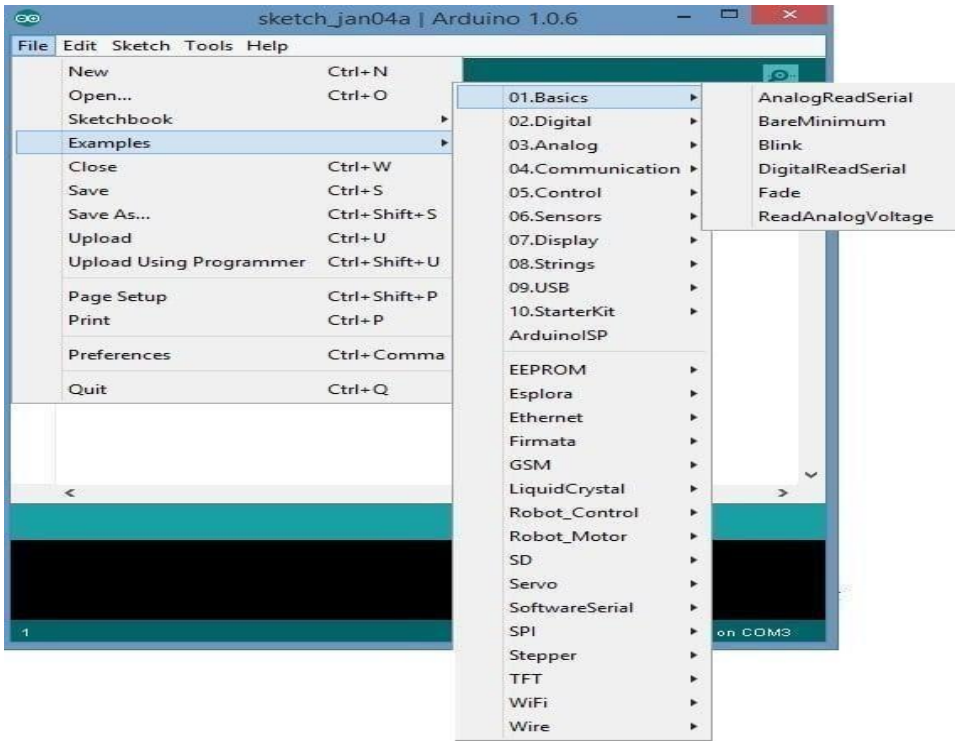

Selecting arduino board

Selecting arduino port



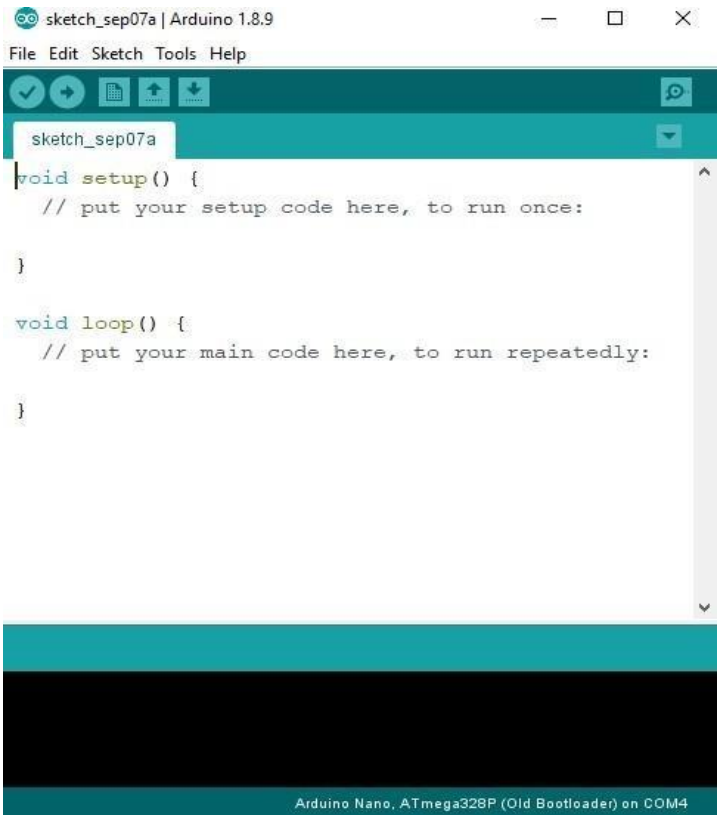

## EXPLORING THE ARDUINO IDE

If you want, take a minute to browse through the different menus in the IDE. There is a good variety of example programs that come with the IDE in the "Examples" menu. These will help you get started with your Arduino right away without having to do lots of research:



**Running the Arduino IDE Software**

This is a display of the Arduino IDE Software. The application is ready to be used to create amazing projects.



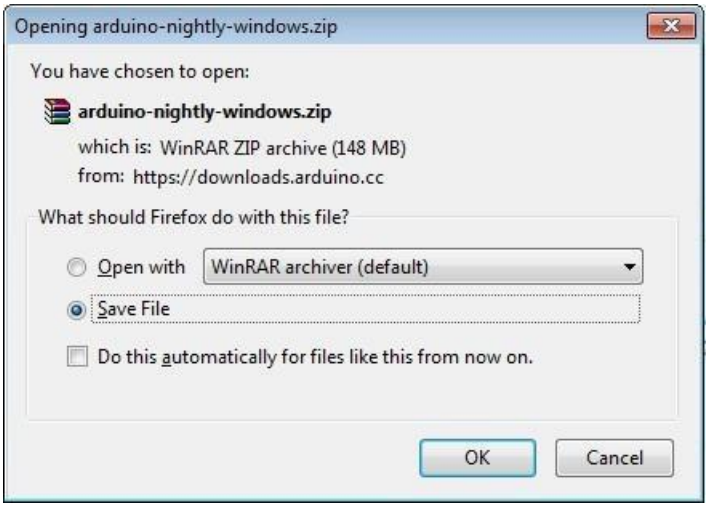

**Steps to connect Arduino board:**

**Step 1** − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will needa standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

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

**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.




**Step 3 − Power up your board.**

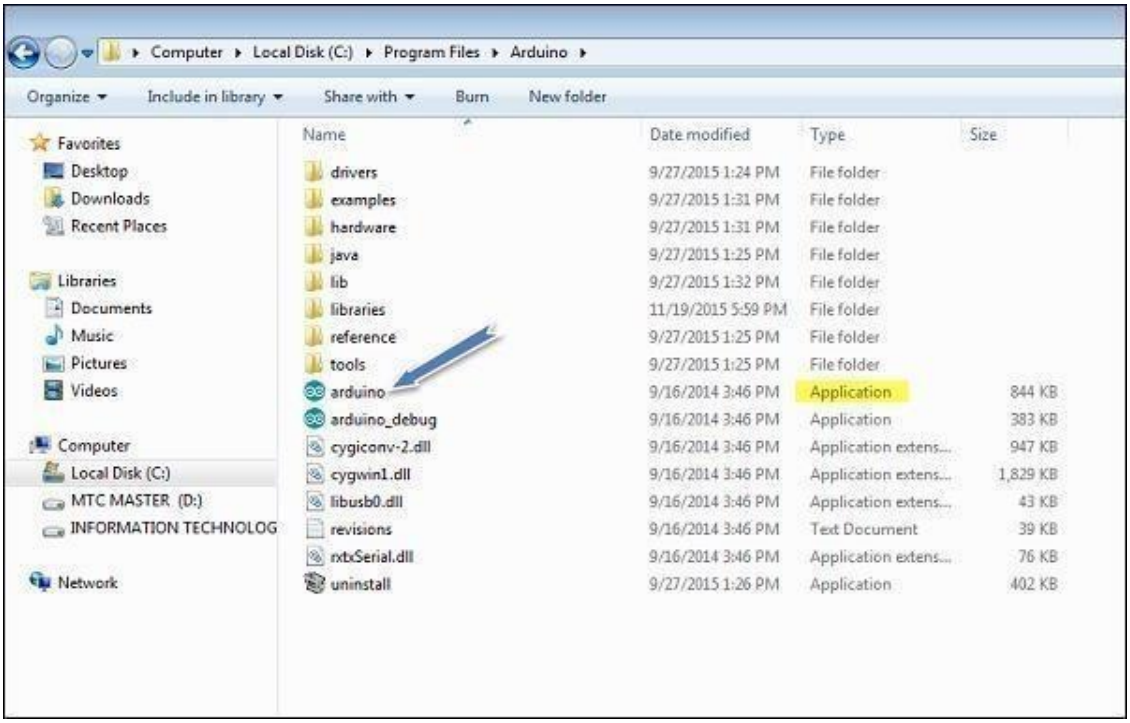
The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have tomake sure that the board is configured to draw power from the USB connection. The power source

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.

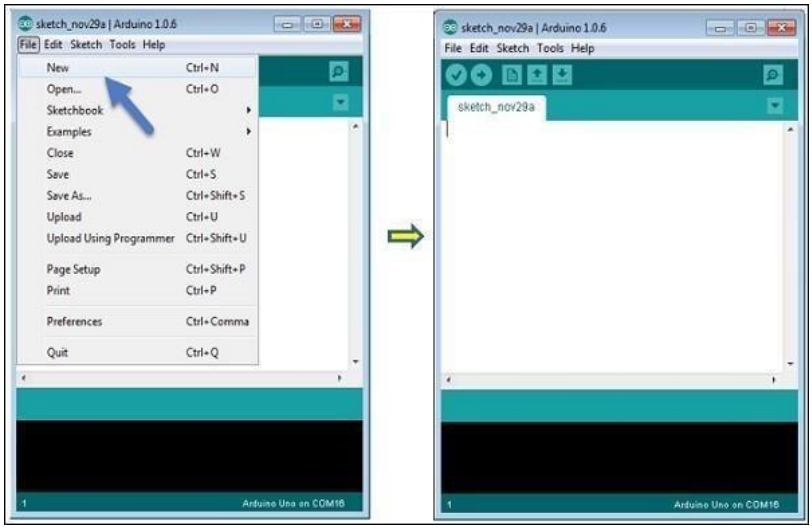



**Step 5 − Open your first project.**

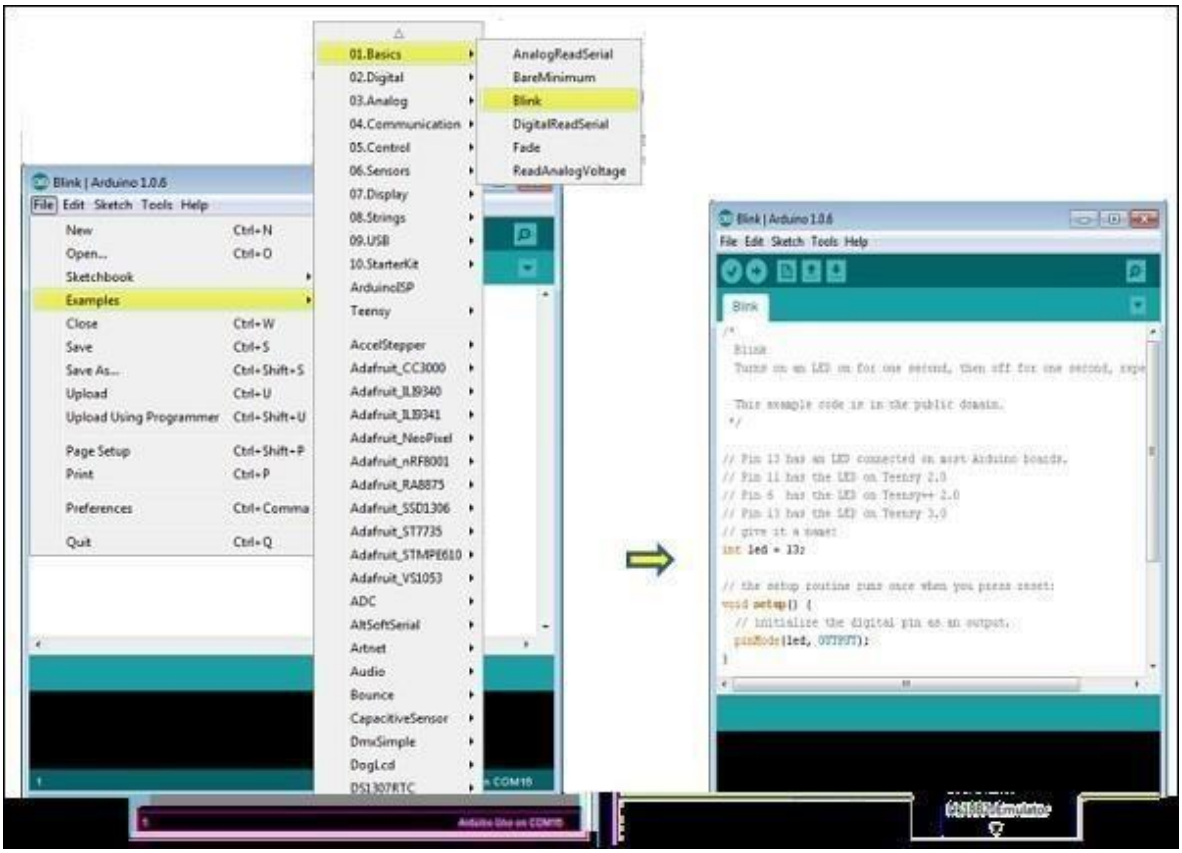
Once the software starts, you have two options −

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

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




To open an existing project example, select File → Example → Basics → Blink.

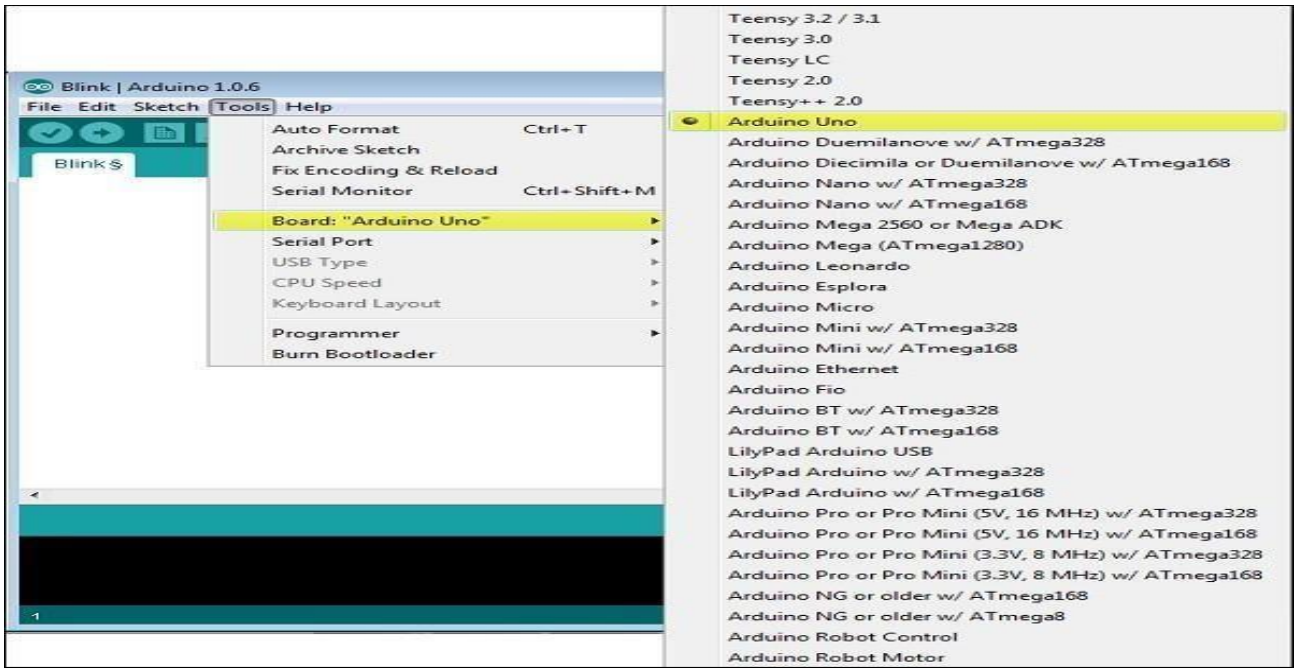



Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

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.

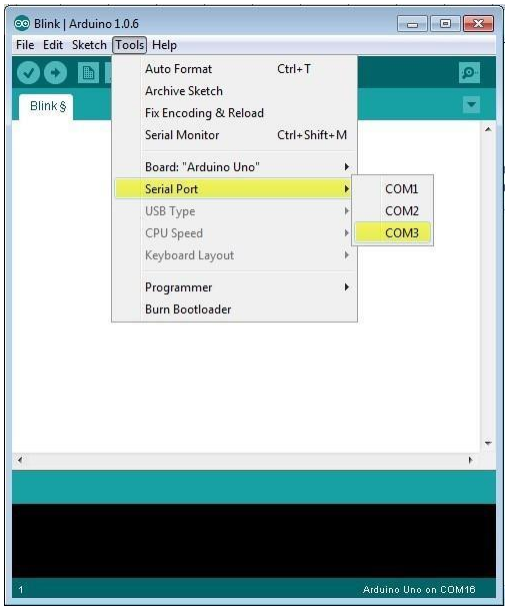
Go to Tools → Board and select your board.




Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matchingthe 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.



**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.


**A** − Used to check if there is any compilation error.

**B** − Used to upload a program to the Arduino board.

**C** − Shortcut used to create a new sketch.

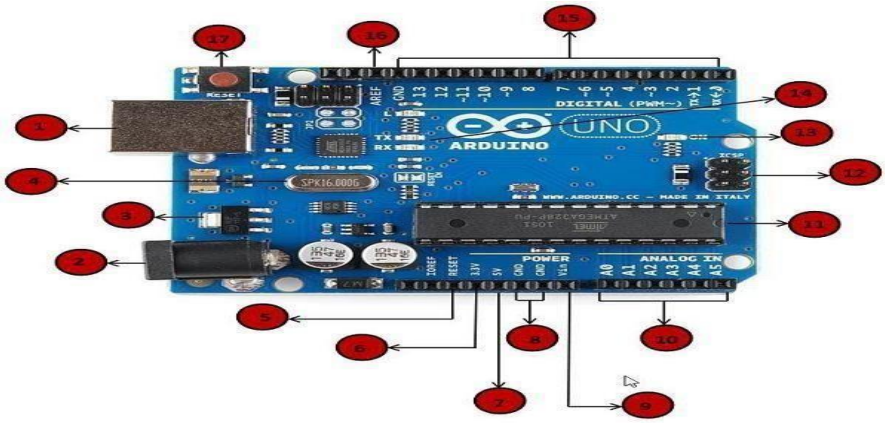
**D** − Used to directly open one of the example sketch.

**E** − Used to save your sketch.

**F** − 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 TXLEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

**Arduino uno Board**



**1. Power USB**

Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).

**2. Power (Barrel Jack)**

Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

**3. Voltage Regulator**

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

**4. Crystal Oscillator**

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. Ittells us that the frequency is 16,000,000 Hertz or 16 MHz.

**5, 17- Arduino Reset**

You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO boardin two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

**6,7,8,9: Pins (3.3, 5, GND, Vin)**

- 3.3V (6) − Supply 3.3 output volt
  5V (7) − Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.
- GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be usedto ground your circuit.
- Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

**10. Analog pins**

The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from ananalog sensor like the humidity sensor or temperature sensor and convert it into a digital value that

can beread by the microprocessor.

**Main microcontroller**

Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers areusually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

**11. ICSP pin**

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK,RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

**12. Power LED indicator**

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

**13. TX and RX LEDs**

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending theserial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

**14. Digital I/O**

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.

**15. AREF**

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

**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 consists of two main functions −

- Setup( ) function
- Loop( ) function

```
Void setup ( ) {


}
```

- **PURPOSE** − The **setup()** function is called when a sketch starts. Use it to initialize the variables, pinmodes, start using libraries, etc. The setup function will only run once, after each power up or resetof 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** − -

**Result:**

Thus, the Arduino IDE has been successfully installed.

**EXP : 2**

**Date:** **Turn ON and OFF the LEDs.**

**AIM:**

To write a program to switch light on when the input is 1 andswitch the light off when the input is 0 using Arduino.

**COMPONENTS REQUIRED:**

1. Arduino
2. Breadboard
3. Jumper wires
4. Resistor
5. LED

**ALGORITHM:**

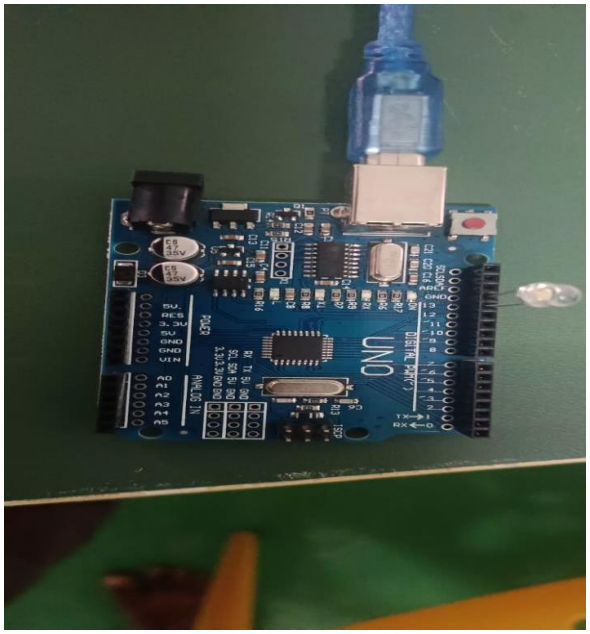
- Start the process.
- Connect micro USB power input to Arduino.
- Connect HDMI to the system to act as monitor for Arduino.
- Connect USB port 2.0 to mouse and keyboard.
- Enter the coding in the terminal for installing python and GPTO.
- Open new sketch →enter coding →save.
- Save file location → open terminal → paste file location in terminal → press enter.
- In the terminal window to get output enter 0 or 1, to switch light ON when the input is 1 and switch light OFF when the input is0 in breadboard using Arduino.
- Stop the process.

**CODING:**

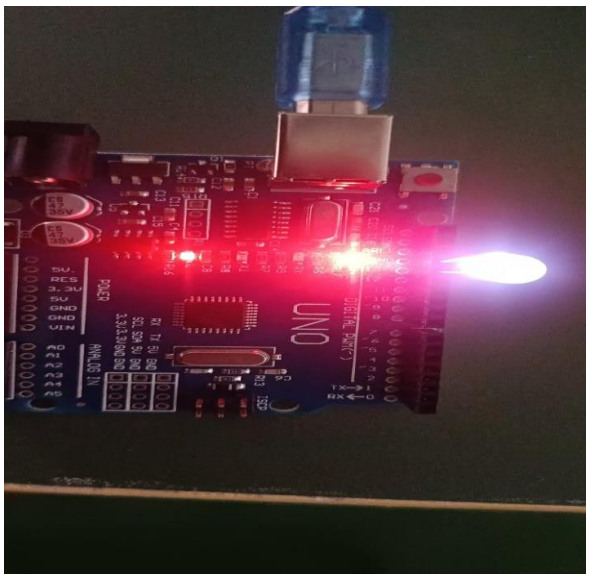
```
void setup() {
// put your setup code here, to run once:
pinMode(13,OUTPUT);
}
void loop() {
// put your main code here, to run repeatedly:
digitalWrite(13,HIGH);
delay(10000);
digitalWrite(13,LOW);
delay(10000);
}
```

## OUTPUT:

**Input**



**Output**



## RESULT:

Thus the output to switch light ON/OFF using Arduino has been successfully executed.