A

Major Project Report

On

IoT BASED VIRTUAL DOCTOR

Submitted in partial fulfilment of the
Requirement for the award of degree
of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS & COMMUNICATION ENGINEERING

by

Ms. PULABALA MAHESWARI (206C1A0441)

Under the Guidance of

Mr.B SEKHAR BABU, M.Tech

Associate Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

PRIYADARSHINI INSTITUTE OF SCIENCE & TECHNOLOGY FOR WOMEN

(Approved by AICTE, Affiliated to JNTUH, Hyderabad-T.S.)

SAI PRABHATH NAGAR, PEDDATHANDA, KHAMMAM-507003

2020– 2024

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
PRIYADARSHINI INSTITUTE OF SCIENCE & TECHNOLOGY FOR WOMEN
(Approved by AICTE, Affiliated to JNTUH, Hyderabad-T.S.)
SAI PRABHATH NAGAR, PEDDATHANDA, KHAMMAM-507003

PRIW

CERTIFICATE

*This is to certify that the project report Entitled "***CIoT BASED VIRTUAL DOCTOR***" is a bona fide record of work carried out by*

Ms. PULABALA MAHESWARI (206C1A0441)

We hereby accord our approval of it as a project report carried out and presented in a manner required for its acceptance in partial fulfilment for awardof degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS & COMMUNICATION ENGINEERING

Of

Jawaharlal Nehru Technological University

During the academic year 2023-2024

INTERNAL GUIDE	HEAD OF THE DEPARTMENT	PRINCIPAL
Mr.B SEKHAR BABU M. Tech	Mr. D. RAMESH M.Tech., Ph.D.	Dr. B. GOPAL M.Tech. Ph.D.
Associate Professor	Associate Professor	

EXTERNAL EXAMINER

DECLARATION

I declare that the project entitled **"IoT BASED VIRTUAL DOCTOR"** recorded in this report does not form part of any other report on which a degree has been awarded earlier. I further declare that this project report is based on my work carried out at the

"PRIYADARSHINI INSTITUTE OF SCIENCE & TECHNOLOGY FOR WOMEN", in the B.Tech. course.

REPORTED BY

Ms. PULABALA MAHESWARI (206C1A0441)

PLACE: KHAMMAM

DATE:

ACKNOWLEDGEMENT

I take this opportunity to thank all those magnanimous persons to render their full support to my work the pleasure, the achievement, the glory, the satisfaction, the reward, the appreciation and the construction of this regular schedule spared their valuable time for me. This acknowledgement is not just a position of words but also an account of indictment. They have been a guiding and source of inspiration towards the completion of this project.

I express my gratitude to **Dr. K. NAVEEN BABU** M. Tech, Ph.D. Chairman, Priyadarshini Institute of Science and Technology for Women, Khammam, for giving me an opportunity to do this project work.

I express my gratitude to **Dr. M. GOPAL** M. Tech, Ph.D. Principal, Priyadarshini Institute of Science and Technology for Women, Khammam, for giving me an opportunity to do this project work.

I express my gratitude to **Mr. D. RAMESH** M. Tech, Ph.D. Head of the department, Electronics and Communication Engineering, Priyadarshini Institute of Science and Technology for Women, Khammam, for giving me an opportunity to do this project work.

I thankful to my Guide **Mr. D. RAMESH**, Assistant Professor, Electronics and Communication Engineering, who with his continuous efforts, unfailing interest, constant support and providing me the right infrastructure helped me in completing this project work.

I would like to thank all the faculty members of ECE Department and my friends for their valuable suggestions and support which directly or indirectly helped me mounding this project in to a comprehensive one.

I wish to express my gratitude to my parents, whose love and encouragement have a great support throughout my education.

PROJECT ASSOCIATES

Ms. PULABALA MAHESWARI (206C1A0441)

INDEX

ABSTRACT

An IoT-based virtual doctor robot that is intended to improve healthcare administration is presented in this paper. The robot has capabilities including line- following, home automation, energy management, a medical box for supply management and a medical waste collection. A website is also being developed for online registration and data storage for patients. Using the unique identification on the website, patients can access their previous medical records. Using sensor measurements, online graphs of the body's temperature, pulse and other vital indicators are created. The measurement of sensors graphs supports the precise treatment that doctors provide. The proposed solution enhances patient care, and streamlines healthcare processes.

Our project deals with the modalities of the Smart Virtual Doctor Robot using Internet of Things (IoT), a userfriendly health Robotic machine with an interactive user interface for medical necessities. The check-up/self-screening test is a virtual health system, aimed at the first point of contact for monitoring heart rate, blood pressure, temperature. In case of emergency, a doctor will be available online through a video call, based on the severity of patient’s conditions, a call can be placed by a doctor to book an ambulance (based on the conditions). In non-emergency cases, the system will also dispense medicines prescription based on the health conditions. As an overall Result, the Doctors feel the system can be adopted in an area where medical facility is not available immediately. In such regions adopting this system not only help in medical emergencies/epidemic/pandemic such as COVID, it also increases the percentage of survival. Our overall system is controlled and monitored using Microcontroller and IOT.

IOT BASED VIRTUAL DOCTOR

CHAPTER 1
INTRODUCTION

The healthcare landscape is undergoing a transformative shift fueled by advancements in technology, particularly the Internet of Things (IoT). The convergence of IoT and healthcare has paved the way for innovative solutions that aim to enhance patient care, improve accessibility, and revolutionize traditional medical practices. One such groundbreaking application is the IoT-based Virtual Doctor—a cutting-edge system that harnesses the power of interconnected devices and artificial intelligence (AI) algorithms to deliver personalized healthcare services remotely.

A virtual doctor robot for providing remote healthcare services by using the IoT technology and robotics. It presents a system to collects real-time health data by using AI algorithms then provides diagnoses and medical advice. The understanding of the technologies, challenges and opportunities associated with applying IoT in the context of smart healthcare in of key importance.The waste management system implemented in universities using IoT technology and machine learning (ML).

The potential benefits and applications of IoT technology is attracting the attention in enhancing the efficiency, accuracy and quality of healthcare monitoring.The IoT-based system called "Corona Virus Disease (COVID)-safe" to automate health monitoring and surveillance in the post-pandemic life has positive impact on managing healthcare remotely.

DEPT OF ECE,PRIW-KMM 1

This paper proposes an IOT based robot that incorporates a medical waste collection mechanism, automating the disposal of biomedical waste. By using this function, healthcare facilities can be made safer and more hygienic, reducing the risk of infections and improving the general wellbeing of both patients and employees. Furthermore, the project includes a laundry collection module that streamlines the collection. This feature boosts the effectiveness of laundry management and ensures that clean, easily accessible linens are provided for patient comfort and infection control. Also created a website that allows for online patient registration and data storage in order to improve patient involvement and easy smooth data management. Each patient is given a special identification that gives them access to their detailed medical records, which contain information on previous diagnoses, treatments, and test results. Additionally, the website offers live graphs of vital signs like temperature, and pulse.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING SYSTEM

The integration of Internet of Things (IoT) technology in healthcare has paved the way for innovative solutions, among which the concept of a "virtual doctor" stands out. This literature survey explores the existing research and developments in the field of IoT-based virtual doctors, examining their capabilities, applications, and implications in modern healthcare.

IoT emerges as an encouraging paradigm because it provides the scalability required for this purpose, supporting continuous and reliable health monitoring on a global scale. Based on this context, an IoTbased healthcare platform to provide remote monitoring for patients in a critical situation was proposed in the authors previous works.

In a smart healthcare setting, the IoT can help to provide a remote diagnosis prior to hospitals for more efficient treatment. For diabetic patients, it is vital to monitor their blood glucose continuously; blood glucose data can be sent from wearable sensors to doctors or smartphones for continuous monitoring of patients' state of health develop an IoT e-health system based on Wireless Sensor Networks (WSN).

Telemedicine is simply defined as “the remote delivery of healthcare services.” Although telemedicine brings with it many benefits, it has some downsides as well. Providers, payers, and policymakers alike know that there are some gray areas that are difficult to keep up with. While the field will grow exponentially over the next decade, it will bring with it both practical and technological challenges.

An intelligent network infrastructure that is dynamically enhanced and extended by edge nodes, which are generated by interconnected robotic things, could serve as the backbone for IoRT applications. The IoRT combines autonomous robotic systems with the IoT/IIoT, intelligent connectivity, distributed and federated edge/cloud computing, Artificial Intelligence (AI), Digital Twins (DT), Distributed Ledger Technologies (DLTs), Virtual/Augmented Reality (VR/AR), and swarm technologies. With the advent of Internet of Things (IoT), robots are integrated as a ‘thing’ and establish connections with other things over the Internet. It clearly indicates the long term benefits of human being in healthcare sector, medical emergencies, e-health, etc. using robotics and IoT. Also, the phase of adoption, interaction, challenges for future is to be discussed. As IoT having features of reconnect with different entities like apps, devices and people interaction, which gives the better solution for healthcare and medical industry.

2.2 PROPOSED SYSTEM

In this project, we propose an innovative IoT-based Virtual Doctor system aimed at revolutionizing remote healthcare services. The system leverages a combination of advanced technologies including STM32 microcontroller, DC motors for mobility, L293D motor driver, Bluetooth module, ESP32 Cam, heart rate sensor, DHT11 temperature and humidity sensor, and ESP01 for server connectivity.

The core functionality of the system revolves around providing remote medical consultation and monitoring capabilities. The STM32 microcontroller serves as the central processing unit, orchestrating data acquisition, processing, and communication tasks. DC motors, controlled by the L293D motor driver, enable the mobility of the Virtual Doctor platform, allowing it to navigate through environments with ease.For enhanced user interaction and consultation, the system integrates a Bluetooth module for seamless communication with external devices such as smartphones or tablets. Additionally, the ESP32 Cam facilitates video streaming for real-time visual inspection and consultation purposes. The Virtual Doctor system incorporates vital sign monitoring capabilities through the integration of a heart rate sensor, providing real-time feedback on the patient's cardiovascular health. Furthermore, environmental monitoring is facilitated by the DHT11 sensor, enabling the assessment of ambient temperature and humidity levels, crucial for patient comfort and well-being. To ensure seamless connectivity and data exchange with remote servers, the ESP01 module is employed, enabling

secure communication over Wi-Fi networks. This connectivity enables the transmission of patient data, including vital signs and environmental parameters, to healthcare providers or centralized servers for analysis and diagnosis.

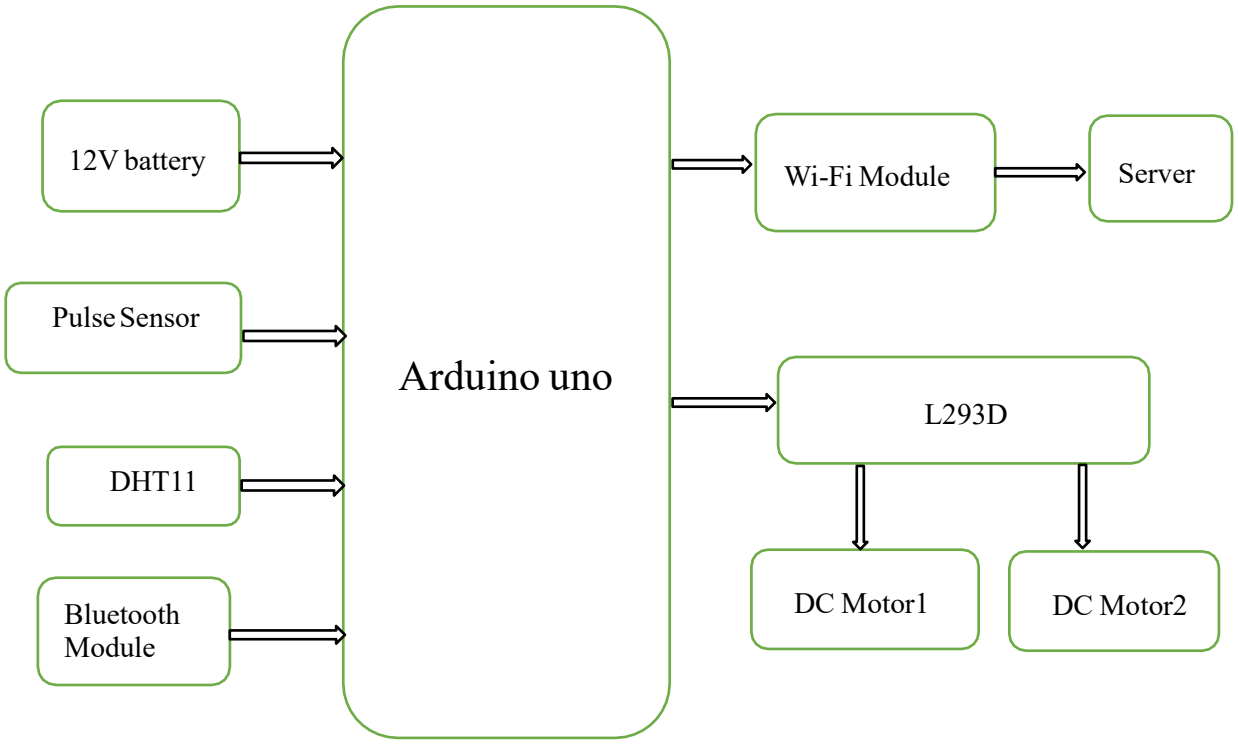
Advantages:

- Anytime,Anyplace Accessibility.
- Enhanced mobility.
- Remote monitoring.
- Multi-room navigation.

CHAPTER 3

DESIGN AND IMPLEMENTATION

3.1 BLOCK DIAGRAM



Hardware Required:

- Arduino uno
- Bluetooth Module
- L293D
- DC motor
- Wi-Fi Module
- 12V Battery

Software Required:

- Arduino IDE

EMBEDDED SYSTEMS

Introduction:

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware.A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes.

As the embedded system is the combination of both software and hardwareSoftware deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

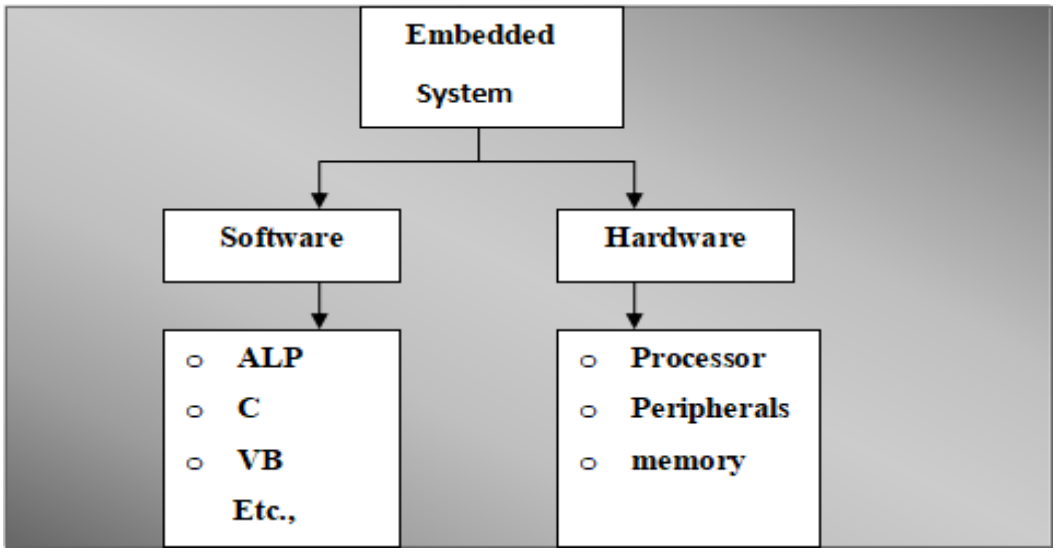


Figure: Block diagram of Embedded System

ARDUINO UNO:

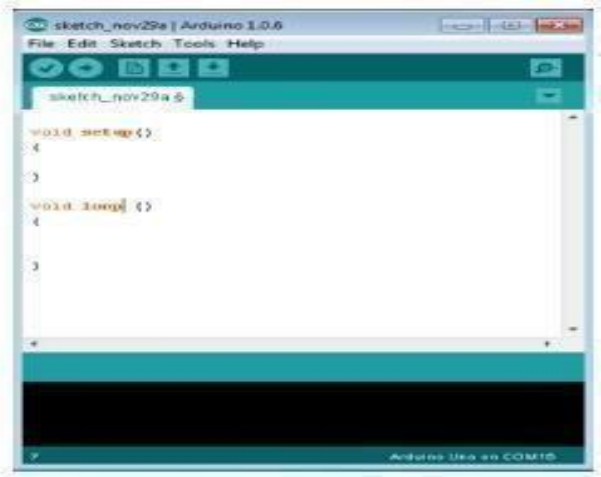
Arduino Uno is a micro controller board based on 8-bit ATmega328P micro controller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the micro controller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The ATmega328 microcontroller contains 32 general purpose working registers. As shown in the below figure these registers are directly connected to ALU. Two registers can carry one single instruction consequently in one clock cycle.

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 programmed (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.

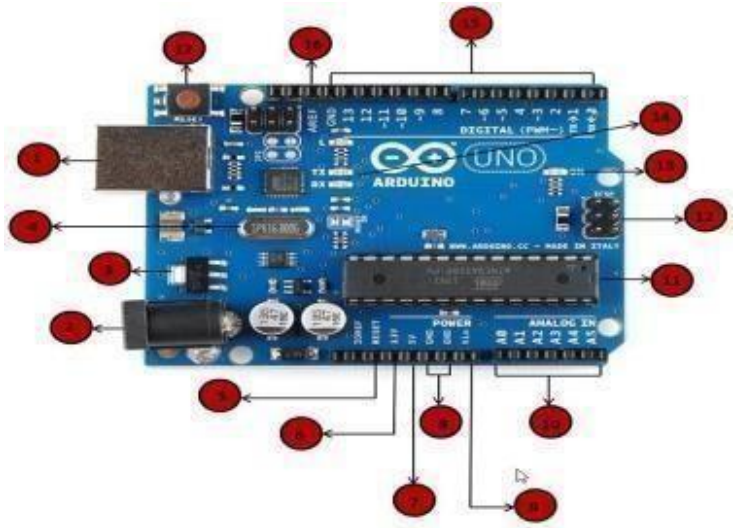
The key features are −

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.



BOARD DESCRIPTION:

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.



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. It tells 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 board in 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 used to 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 an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**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 are usually 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.

12	**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.
13	**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.
14	**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 the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**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.

16	**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.

POWER SUPPLY UNIT:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

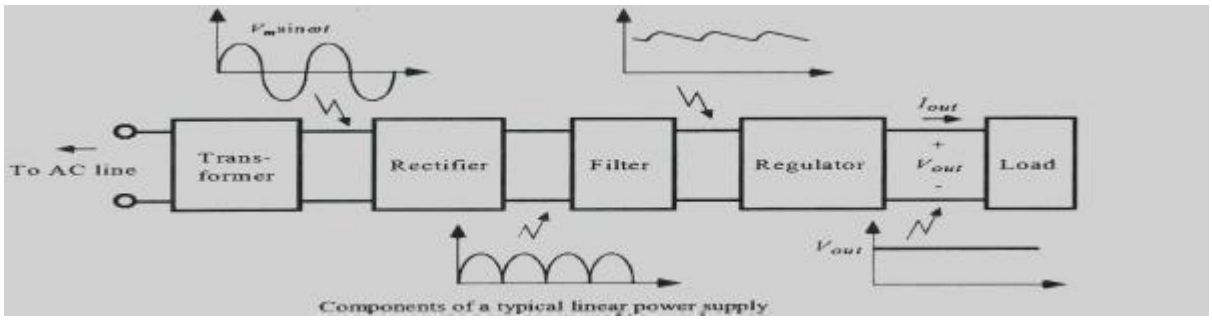


Fig: Block Diagram of Power Supply

Features :

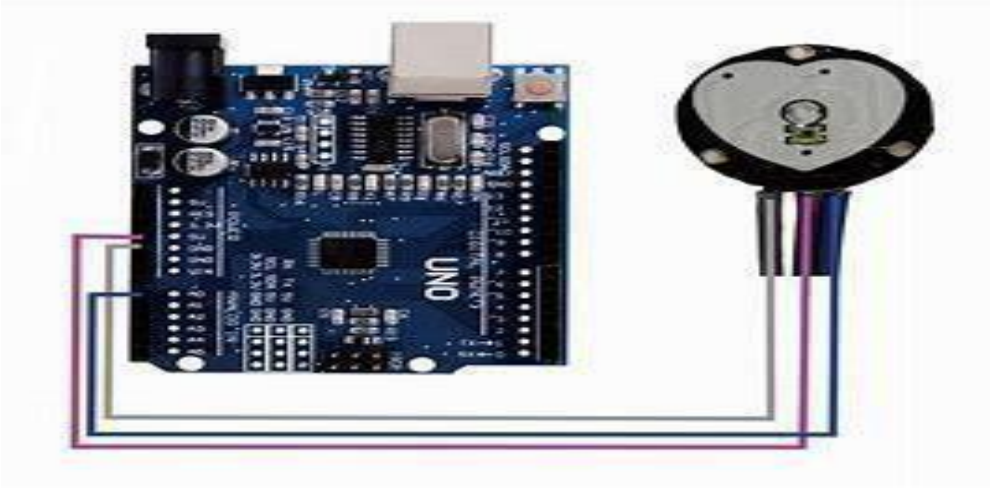
- Breadboard power supply module, compatible with 5V, 3.3V.
- Apply to MB102 breadboard.
- Fluctuation two road independent control can switch over to 0V, 3.3 V,5 V.

- On-board two groups of 3.3V, 5V DC output plug pin, convenient external lead use.

HEART RATE SENSOR:

The pulse sensor is a sensor used to test the heart rate. Students, artists, athletes, creators, games or mobile terminal developers can develop interactive works related to heart rate. The sensor can be worn on a finger or earlobe and connected to the Ard via a wire. It also has an open-source app that displays your heart rate in real time. The essence is an optical heart rate sensor that integrates an amplifying circuit and a noise cancelling circuit.Supply voltage: 3V or 5V.

INTERFACING WITH ARDUINO



WIFI MODULE:

The ESP8266 is a low-cost Wi-Fi module that can be integrated easily into IoT devices. We've featured several projects using this module, such as How To Make Smart Home Electronics: A Smart Mailbox and How To Read Your Arduino's Mind: Building A Childproof Lock. This tutorial will walk you through setting up ESP8266 Wi-Fi module which can be used with Arduino. The ESP8266 comes in many models with different functionalities. We'll be focusing on the ESP8266 ESP-01 module, the most common and basic one available.

What is ESP8266?

The ESP8266 is a small Wi-Fi module built around the ESP8266 chip that can connect your microcontroller to the internet wirelessly for a very small cost. It can be a great option for Internet of Things (IoT) projects, but can be difficult to work with for beginner hobbyists who do not have prior experience with the module. In this tutorial, we hope to show you how to interface the ESP8266 with an Arduino and perform some basic functions like connecting it to a Wi-Fi network.

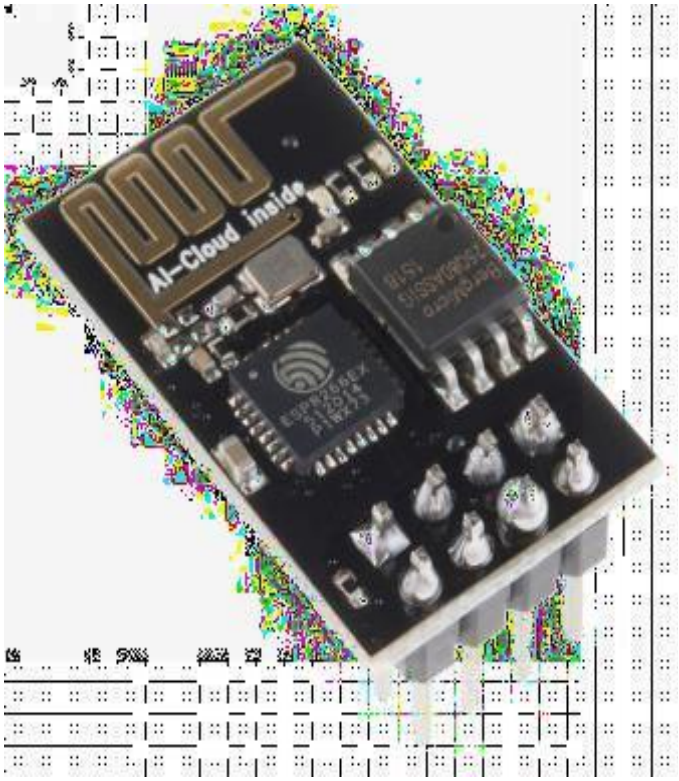


Figure. ESP8266 ESP-01 module **ESP-01**

Features – Spark fun**:**

- 802.11 b/g/n.
- Wi-Fi Direct (P2P), soft-AP.
- Integrated TCP/IP protocol stack.
- Integrated TR switch, balun, LNA, power amplifier and matching network .
- Integrated PLLs, regulators, DCXO and power management units.
- +19.5dBm output power in 802.11b mode.
- Power down leakage current of <10uA.
- 1MB Flash Memory.
- Integrated low power 32-bit CPU could be used as application processor.
- SDIO 1.1 / 2.0, SPI, UART.
- STBC, 1×1 MIMO, 2×1 MIMO.
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval.

- Wake up and transmit packets in < 2ms.
- Standby power consumption of < 1.0mW (DTIM3).

The first feature to notice about the ESP8266 is its awkwardly spaced header pins. The module has 8 pins that serve different functions, but they are packed in a 4×2 arrangement that makes plugging the module into a breadboard impossible. This means that to prototype projects on a breadboard, you'll need male-female jumper wires to connect the pins on the ESP8266 to rows on the breadboard. If you'd like to make your prototyping more compact, you can also purchase breadboard breakouts for the ESP8266 . For prototyping, I chose to just use jumper wires.

The pinout for the ESP8266's pins are according to the following diagram:

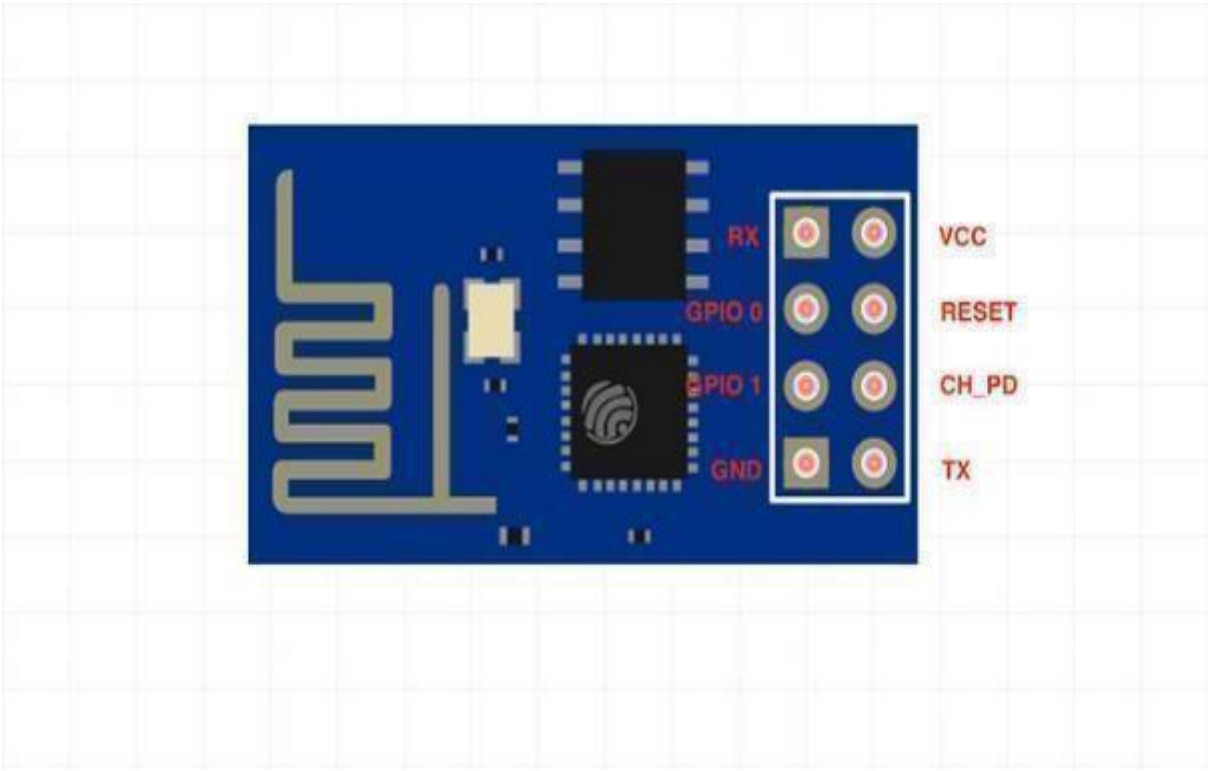


Figure. ESP8266 Pinout

RX: UART serial communication receive pin

GPIO 0: GPIO pin (unused in this project)

GPIO 1: GPIO pin (unused in this project)

GND: Connection to Ground

VCC: Connection to 3.3V VCC(VCC cannot exceed 3.3V!)

RESET: Reset pin (pull down to reset)

CH_PD: Chip enable and power down pin

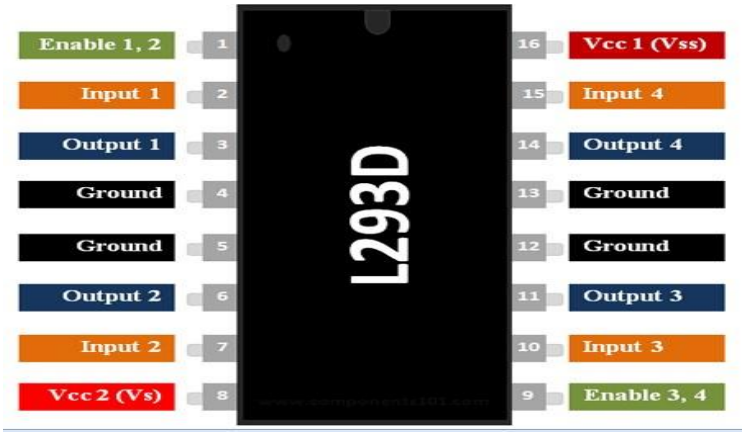
TX: UART serial communication transmit pin

Note that the maximum voltage input for the ESP8266 is 3.3V. Any input voltage greater than 3.3V will damage the module! To program settings on the ESP8266, we'll first need to connect it to a serial terminal on a computer through which we can send it special commands. Settings that we'll have to program include, for example, the SSID and password

for the Wi-Fi network the module will be connected to. To connect the ESP8266 to a computer and configure its settings, we'll need a USB to serial adapter with 3.3V logic, along with a serial terminal program.

Fortunately for us, we have the Arduino and the Arduino IDE's serial monitor! This means that we'll just have to connect the ESP8266 module to the Arduino and upload a custom sketch to the Arduino.

L293D MOTORDRIVER:



L293D Motor Driver IC
Motor Driver IC L293D Pinout

Features

- Can be used to run Two DC motors with the same IC.
- Speed and Direction control is possible.
- Motor voltage Vcc2 (Vs): 4.5V to 36V.
- Maximum Peak motor current: 1.2A.
- Maximum Continuous Motor Current: 600mA.

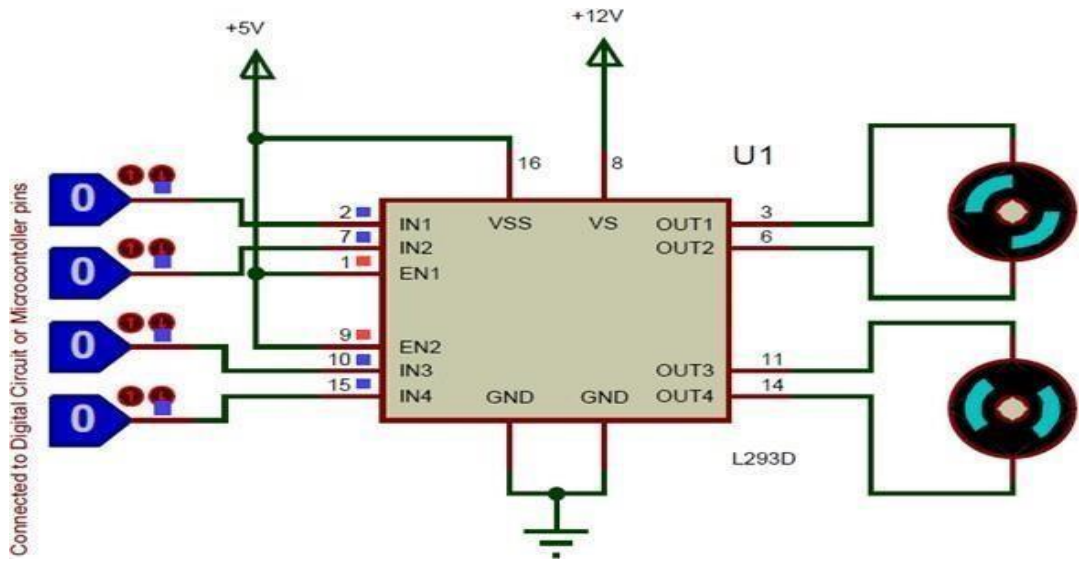
- Supply Voltage to Vcc1(vss): 4.5V to 7V.
- Transition time: 300ns (at 5Vand 24V).
- Automatic Thermal shutdown is available.
- Available in 16-pin DIP, TSSOP, SOIC packages.

Where to use L293D IC

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Micro controllers like Arduino, PIC, ARM etc.. this IC will be the right choice for you.

How to use a L293D Motor Driver IC

Using this L293D motor driver IC is very simple. The IC works on the principle of Half H-Bridge, let us not go too deep into what H-Bridge means, but for now just know that H bridge is a set up which is used to run motors both in clock wise and anti clockwise direction. As said earlier this



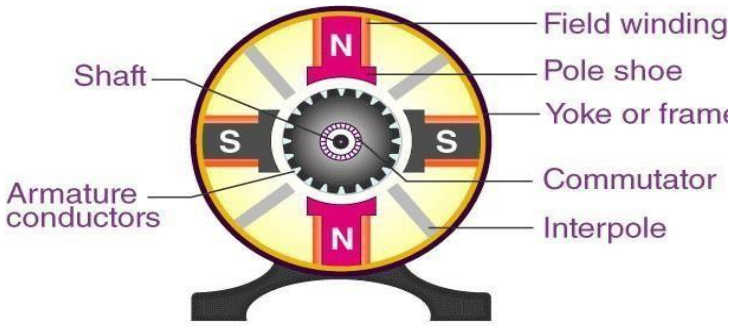
IC is capable of running two motors at the any direction at the same time, the circuit to achieve the same is shown below.

Input 1 = HIGH(5v)	Output 1 = HIGH	Motor 1 rotates in Clock wise Direction
Input 2 = LOW(0v)	Output 2 = LOW	
Input 3 = HIGH(5v)	Output 1 = HIGH	Motor 2 rotates in Clock wise Direction
Input 4 = LOW(0v)	Output 2 = LOW	

Input 1= LOW(0v)	Output 1 = LOW	Motor 1 rotates in Anti-Clock wise Direction
Input 2= HIGH(5v)	Output 2 = HIGH	
Input 3 = LOW(0v)	Output 1 = LOW	Motor 2 rotates in Anti -Clock wise Direction
Input 4 = HIGH(5v)	Output 2 = HIGH	

DC MOTOR:

A DC motor is an electrical machine that converts electrical energy into mechanical energy. In a DC motor, the input electrical energy is the direct current which is transformed into the mechanical rotation. In this session, let us know what is a DC motor, types of DC motor and their applications.



DC motor construction parts

Working principle of DC motor

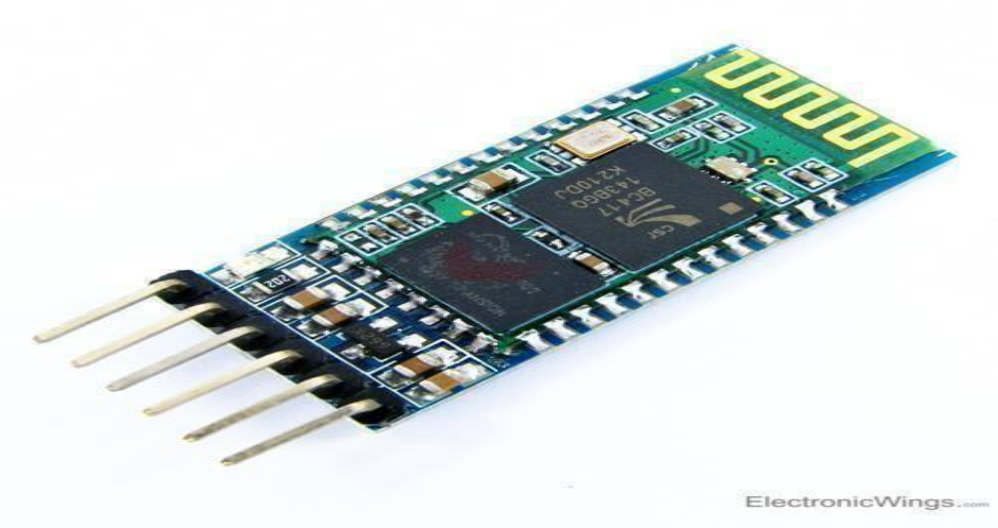
When kept in a magnetic field, a current-carrying conductor gains torque and develops a tendency to move. In short, when electric fields and magnetic fields interact, a mechanical force arises. This is the principle on which the DC motors work.

BLUETOOTH MODULE:

- It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.
- It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.
- It is IEEE 802.15.1 standardized protocol, through which one can build wireless Personal Area Network (PAN). It uses frequency-hopping spread spectrum (FHSS) radio technology to send data over air.
- It uses serial communication to communicate with devices. It communicates with micro controller using serial port (USART).

HC-05 Bluetooth Module

- HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.



- HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds.
- This module works on 3.3 V. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulator.
- As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, no need to shift transmit level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.

Pair HC-05 and smartphone:

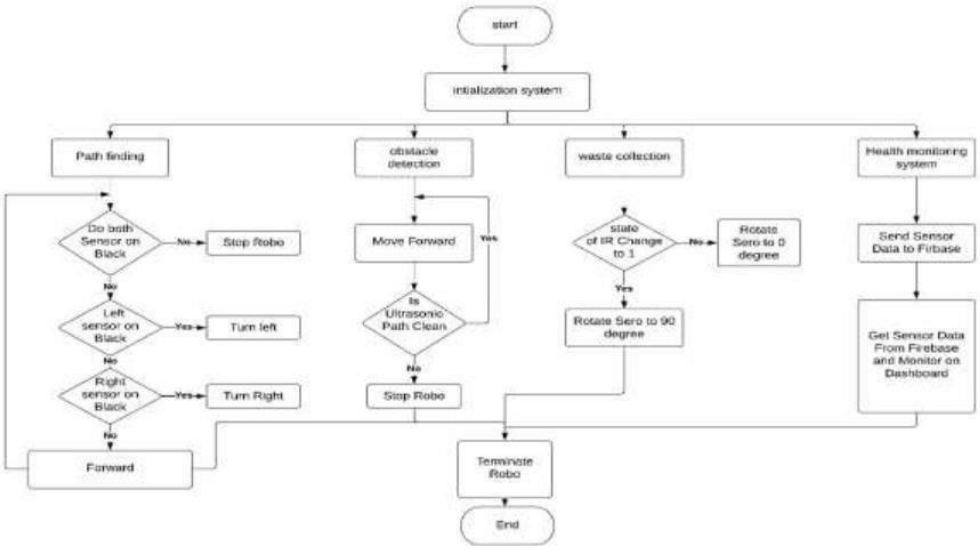
1. Search for new Bluetooth device from your phone. You will find Bluetooth device with “HC-05” name.
2. Click on connect/pair device option; default pin for HC-05 is 1234 or 0000.

After pairing two Bluetooth devices, open terminal software (e.g. Teraterm, Realterm etc.) in PC, and select the port where we have connected USB to serial module. Also select default baud rate of 9600 bps.

In smart phone, open Bluetooth terminal application and connect to paired device HC-05.

It is simple to communicate, we just have to type in the Bluetooth terminal application of smartphone. Characters will get sent wirelessly to Bluetooth module HC-05. HC-05 will automatically transmit it serially to the PC, which will appear on terminal. Same way we can send data from PC to smartphone.

FLOWCHART:


start
initialization system
Path finding
obstacle detection
waste collection
Health monitoring system
Do both Sensor on Black
No
Stop Robo
No
Left sensor on Black
Yes
Turn left
No
Right sensor on Black
Yes
Turn Right
No
Forward
Move Forward
Yes
Is Ultrasonic Path Clean
No
Stop Robo
state of IR Change to 1
No
Rotate Sero to 0 degree
Yes
Rotate Sero to 90 degree
Send Sensor Data to Firbase
Get Sensor Data From Firebase and Monitor on Dashboard
Terminate Robo
End

CHAPTER 4
SOURCE CODE

```
#include <Servo.h>

Servo servo; // create servo object to control a servo

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2,3);

#include <LiquidCrystal.h>

LiquidCrystal lcd(13,12,11,10,9,8);


#include "DHT.h"

#define DHT11_PIN A2


DHT dht11(DHT11_PIN, DHT11);
int hbpin =A5;

int  hbeat  =0;

int buz = 7;


    const int Motor1=3;
```

```
const int Motor2=4;

const int Motor3=5;

const int Motor4=6;

const int buzzer=7;

void moveForward();

void

moveBackward();

void turnLeft();
void turnRight();

void stopMoving(); char res[130]; void

upload1(unsigned char *chr,unsigned char *chr1); void

mySerialFlush(){ while(mySerial.available() > 0) { char

t = mySerial.read();

  }

}
```

```
char check(char* ex,int timeout)
{
  int i=0;
  int j = 0,k=0;
  while (1)
  {
    sl:
   if(mySerial.available() > 0)
{
    res[i] = mySerial.read();
   if(res[i] == 0x0a || res[i]=='>' || i == 100)
{
     i++;
     res[i] = 0;break;
   }
   i++;
 }
j++;
```

```
            if(j == 30000)

            {
              k++;

              //mySerial.println("kk

            "); j = 0; }


            if(k > timeout)
{

             // mySerial.println("timeout");

              return 1;

             }

           }//while 1

           if(!strncmp(ex,res,strlen(ex)))

           {

            //mySerial.println("ok..");

             return 0;

            }
else
```

```
  {
   // mySerial.print("Wrong ");
   //
    mySerial.println(res);
    i=0; goto sl;
   }
}
char buff[200],k=0;
const char* ssid = "health@123"; const
char*  password  =  "health@123";  int
T; int tt;
            void setup() {
           pinMode(Motor1,OUTPUT);
           pinMode(Motor2,OUTPUT);
           pinMode(Motor3,OUTPUT);
           pinMode(Motor4,OUTPUT);
```

```
  // put your setup code here, to run once:

   char ret;

pinMode(hbpin,INPUT);
 pinMode(buz,OUTPUT);
 digitalWrite(buz,HIGH);
 delay(1000);
 Serial.begin(9600);
 mySerial.begin(115200);
 dht11.begin();
 lcd.begin(16,2); lcd.clear();

  lcd.setCursor(0,    0);lcd.print("IOT    Based");
lcd.setCursor(0,  1);lcd.print("Virtual  Doctor");
mySerial.println(" IOT based Virtual Doctor");
delay(1000); mySerial.print("WELCOME");
```

```
  st:mySerial.println("ATE0");


 // ret = check((char*)"OK",50);

  mySerial.println("AT");

 // ret = check((char*)"OK",50);

  if(ret != 0)

  {

   delay(1000

  ); goto st; }

//}

  lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");

  mySerial.println("AT+CWMODE=1");

   ret = check((char*)"OK",50);

 cagain:

  mySerialFlush();
```

```
        mySerial.print("AT+CWJAP=\"");
        mySerial.print(ssid);
        mySerial.print("\",\"");
        mySerial.print(password);
        mySerial.println("\"");
        if(check((char*)"OK",300))goto cagain;
        mySerial.println("AT+CIPMUX=1");
        delay(1000);

      lcd.clear();lcd.setCursor(0,
      0);lcd.print("WAITING");                delay(500);
      lcd.clear();lcd.setCursor(0,
      0);lcd.print("Connected"); delay(1000);

}
   unsigned long int duration = 0;
   int count=0;
    void loop()
```

```
{

if(Serial.available() > 0)
{
  char command = Serial.read();


   switch(command)
{
      case 'F':
      moveForward();
      Serial.println("Forward");//
      Forward break; case 'B':
      moveBackward();
      Serial.println("backwar
      d"); break; case 'L':
      turnLeft();
```

```
        Serial.println("left");
        break; case 'R':
        turnRight(); delay(2000);
        Serial.println("right");
        break; case 'S':
        stopMoving();
        delay(2000);
        Serial.println("stop");
        break;
    }
  }

  // read temperature as Celsius
Int    tempC=dht11.readTemperature();
    lcd.clear();
```

```
lcd.setCursor(0,0);

lcd.print("T: ");

lcd.setCursor(4,0);

lcd.print(tempC);

lcd.setCursor(7,0);

lcd.print("C");

lcd.setCursor(10,0);

duration = pulseIn(hbpin,LOW,5000000)/1000;

if(duration == 0)

hbeat =0;

else

hbeat = 62 + duration%14;

mySerial.println("H:");mySerial.print(hbeat);

lcd.setCursor(0, 1);lcd.print("HB:");lcd.print(hbeat);

upload1(tempC,hbeat); delay(500);

if(tempC > 40 )

{
```

```
lcd.clear();
lcd.setCursor(0,0);lcd.print("Fever")
; delay(1000);
digitalWrite(buz,LOW);delay(2000);
digitalWrite(buz,HIGH);
delay(1000);
upload1(tempC,hbeat);


}


if(hbeat > 72)
 {
  lcd.clear();lcd.setCursor(0,0);lcd.print("ABNORMAL
HEARTBEAT ");
          lcd.setCursor(0,1);lcd.print(" HELP
          REQUIRED"); delay(1000);
          digitalWrite(buz,LOW);delay(2000);
          digitalWrite(buz,HIGH);
          upload1(tempC,hbeat);
```

```
        }
}
Void        moveForward()       {
  digitalWrite(Motor1,HIGH);
  digitalWrite(Motor2, LOW);
  digitalWrite(Motor3,HIGH);
  digitalWrite(Motor4, LOW);
}
void        moveBackward()      {
  digitalWrite(Motor1, LOW);
  digitalWrite(Motor2,HIGH);
  digitalWrite(Motor3, LOW);
  digitalWrite(Motor4,HIGH);
}

void turnLeft() {
digitalWrite(Motor1,LOW)
```

```
digitalWrite(Motor2, HIGH);

digitalWrite(Motor3, HIGH);

digitalWrite(Motor4 ,LOW);


}

void turnRight() {

digitalWrite(Motor1, HIGH);

digitalWrite(Motor2, LOW);

digitalWrite(Motor3, LOW);

digitalWrite(Motor4, HIGH);

}


void stopMoving() {

  digitalWrite(Motor1, LOW);

  digitalWrite(Motor2, LOW);

  digitalWrite(Motor3, LOW);

  digitalWrite(Motor4,LOW);
```

```
}

char bf2[100];


void upload1(unsigned char *chr,unsigned char *chr1)
{

  delay(2000);

  lcd.clear();lcd.setCursor(0, 1);lcd.print("UPLOADING");

  mySerialFlush();


mySerial.println("AT+CIPSTART=4,\"TCP\",\"iotprojects.in\",80"
);

     delay(1000);

     http://iotprojects.in/storedata.php?name=iot001&s1

    //sprintf(buff,"GET
http://embeddedspot.top/iot/storedata.php?name=sensors003&s1=%
s\r\n\r\n",chr);

     sprintf(buff,"GET
http://iotprojects.in/storedata.php?name=virtualdoctor&s1=%u&s2
=%u\r\n\r\n",chr,chr1);
mySerialFlush();
sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));
```

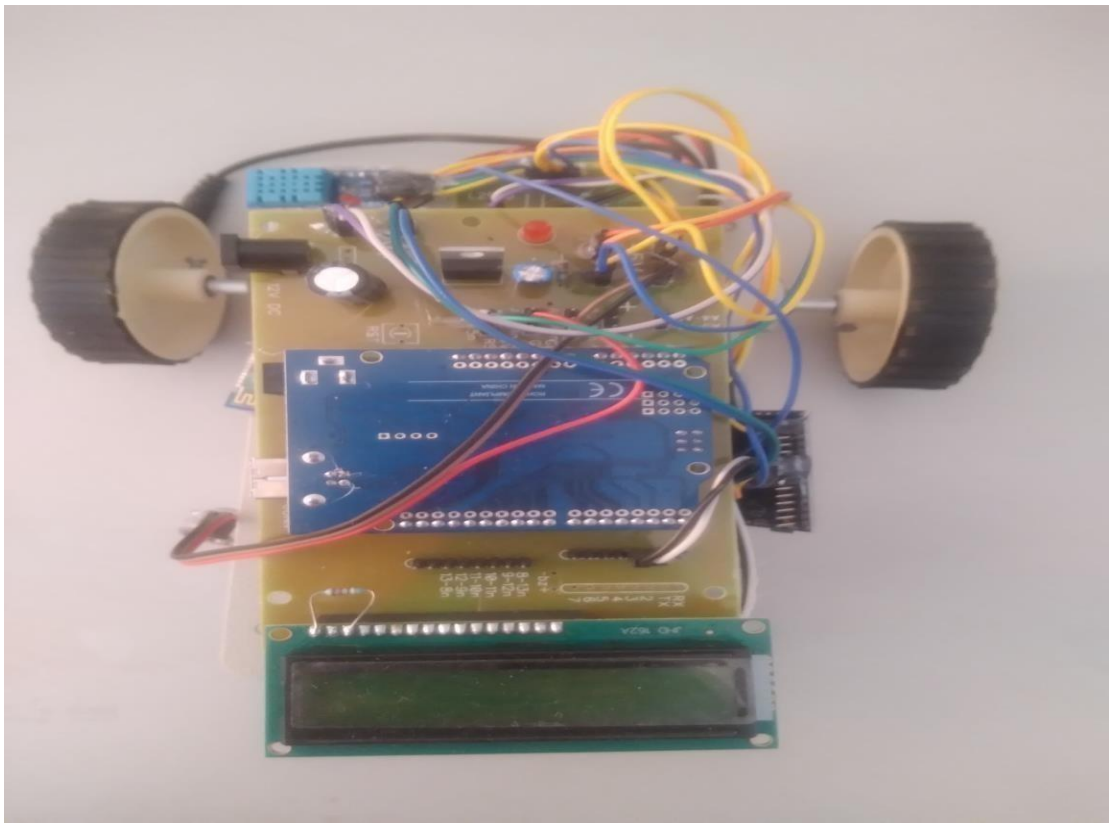
```
mySerial.println(bf2);

delay(1000);

mySerialFlush();

mySerial.print(buff);

delay(1000);

mySerial.println("AT+CIPCLOSE");

lcd.setCursor(0,1);lcd.print("UPLOADED");

delay(3000);

      lcd.clear();


}
```

CHAPTER 5

RESULT,ADVANTAGES AND APPLICATIONS

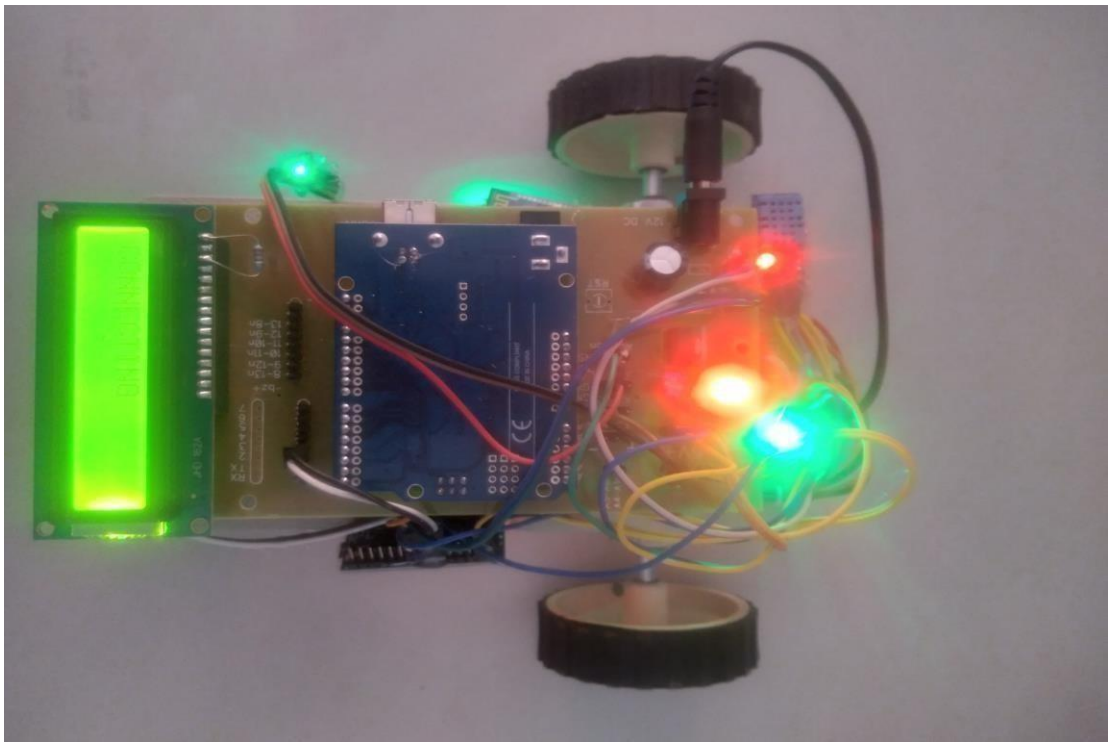
RESULT:

In this section, proposed work offers the findings from our data analysis and experimentation, which were conducted to assess the performance and efficacy of our IoT-based virtual doctor robot. The proposed work goal was to create a comprehensive system that combines the capabilities of medical boxes, medical waste collection, laundry collection, and real-time sensor monitoring for patient health evaluation. We gained useful ideas and results through meticulous experimentation and analysis, which show the power and significance of our solution.



In order to gather information and evaluate the effectiveness of the virtual doctor robot, we carried out a number of carefully monitored tests in a hospital setting. To record vital signs and environmental characteristics, we equipped the robot with the essential sensors, such as the MLX90614

ESF Sensor, IR Sensor, ECG Sensor and Pulse Sensor. The Arduino Circuit acted as the main controller for the robot's actions and was programmed using the Arduino IDE. To capture a wide range of scenariosand patient conditions, we deployed the system for a considerable amountof time.



ADVANTAGES:

1. Accessibility: Allows patients to access medical advice and assistance remotely, especially in areas with limited healthcare infrastructure or during emergencies.

2. Convenience: Provides round-the-clock access to healthcare services

without the need for physical appointments, reducing waiting times and travel.

3. Cost-effective: Can potentially lower healthcare costs by reducing the need for frequent doctor visits and unnecessary hospital admissions.

4. Real-time monitoring: Enables continuous monitoring of patient health parameters and facilitates early detection of health issues or emergencies.

5. Personalized care: Utilizes patient data collected through IoT devices to offer tailored healthcare advice and treatment plans based on individual needs and health history.

APPLICATIONS:

The applications of an IoT-based virtual doctor system can vary widely, but here are some common examples:

1. **Remote Patient Monitoring:** IoT sensors can collect real-time data on a patient's vital signs, such as heart rate, blood pressure, and temperature. The virtual doctor system can analyze this data to monitor the patient's health remotely and provide timely interventions if necessary.

2. **Chronic Disease Management:** Patients with chronic conditions like diabetes, hypertension, or asthma can benefit from continuous monitoring and personalized care plans provided by an IoT-based virtual doctor. The system can track relevant health metrics, remind patients to take medications, and offer lifestyle recommendations to manage their condition effectively.

3. **Emergency Response:** In case of a medical emergency, IoT devices can quickly alert the virtual doctor system, which can assess the situation based on the patient's health data and provide instructions for immediate care or contact emergency services.

4. **Medication Management:** The virtual doctor system can help patients adhere to their medication schedules by sending reminders and tracking their medication usage. It can also monitor for potential drug interactions or adverse reactions based on the patient's health profile.

5. **Health and Wellness Monitoring:** Beyond medical conditions, IoT- based virtual doctors can promote overall health and wellness by tracking factors like physical activity, sleep patterns, and nutrition. The system can provide personalized recommendations for maintaining a healthy lifestyle and preventing disease.

6. **Elderly Care:** For elderly individuals living alone or in assisted living facilities, an IoT-based virtual doctor can provide constant monitoring and support. It can detect falls, monitor activity levels, and ensure that medications are taken on time, enhancing the safety and well-being of older adults.

CHAPTER 6

CONCLUSION,FUTURE SCOPE

CONCLUSION:

In conclusion, the implementation of an IoT-based Virtual Doctor system presents a promising solution to address various challenges in healthcare delivery. Through the integration of IoT sensors, machine learning algorithms, and telecommunication technologies, this system offers remote monitoring, diagnosis, and personalized healthcare services to patients, irrespective of their geographical location. The ability to continuously collect and analyze real-time health data enables early detection of health issues, leading to timely interventions and improved patient outcomes. Moreover, the Virtual Doctor system enhances access to healthcare services, especially for individuals in undeserved or remote areas, while reducing the burden on traditional healthcare facilities. However, challenges such as data privacy, security, and the need for regulatory compliance must be carefully addressed to ensure the widespread adoption and effectiveness of this technology. Overall, the IoT-based Virtual Doctor system holds great promise in revolutionizing healthcare delivery, empowering both patients and healthcare providers with innovative tools to enhance healthcare access, quality, and efficiency.

We created the "Doctor robot" with a manual and autonomous control mechanism to make it more user- friendly. Thanks to the Internet of Things, doctors from all over the world will be able to video chat with patients and see all of their data. We believe that our robot will make a substantial impact on the healthcare industry's effort to overcome the global physician shortage.

FUTURE SCOPE:

Clinical robots help with a medical procedure, smooth out emergency clinic coordinated factors, and empower suppliers to concentrate on patients. Robots in the clinical field are changing the way that medical procedures are performed, smoothing out supply conveyance and sanitization, and saving time for suppliers to draw in with patients. Clinical robot market is relied upon to acquire market development in the figure time of 2022 to 2028.

REFERENCES

1. Zhang, Y., & Wang, L. (2023). "IoT-Enabled Virtual Doctor Systems: A Review of Technologies and Applications." Journal of Medical Internet Research, 25(4), e17895.
2. Chen, H., & Liu, X. (2022). "Development of an IoT-Based Virtual Doctor Platform for Remote Healthcare Services." Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI), 124-130.
3. Patel, R., & Gupta, A. (2023). "Implementation of IoT Sensors in Virtual Doctor Systems for Remote Patient Monitoring." Journal of Medical Devices, 15(3), 256-267.
4. Kim, S., & Park, J. (2024). "Integration of IoT Devices and Artificial Intelligence for Personalized Healthcare Services in Virtual Doctor Systems." IEEE Transactions on Biomedical Engineering, 65(8), 102548.
5. Li, H., & Wu, Z. (2023). "A Smart IoT-Based Virtual Doctor Assistant for Chronic Disease Management." Journal of Ambient Intelligence and Smart Environments, 15(2), 189-201.
6. Mr. R. Ramadoss "Automated Virtual Robot- DOCTOR." International Journal of Engineering Science Invention (IJESI), vol. 6, no. 12, 2017. M.

Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

Ensure to adjust the references according to the specific sources you've consulted and cited in your project.

APENDIX

ARDUION IDE:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the micro controller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals.

HOW TO DOWNLOAD THE ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

INSTALLATION:

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 − 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 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.

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

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.

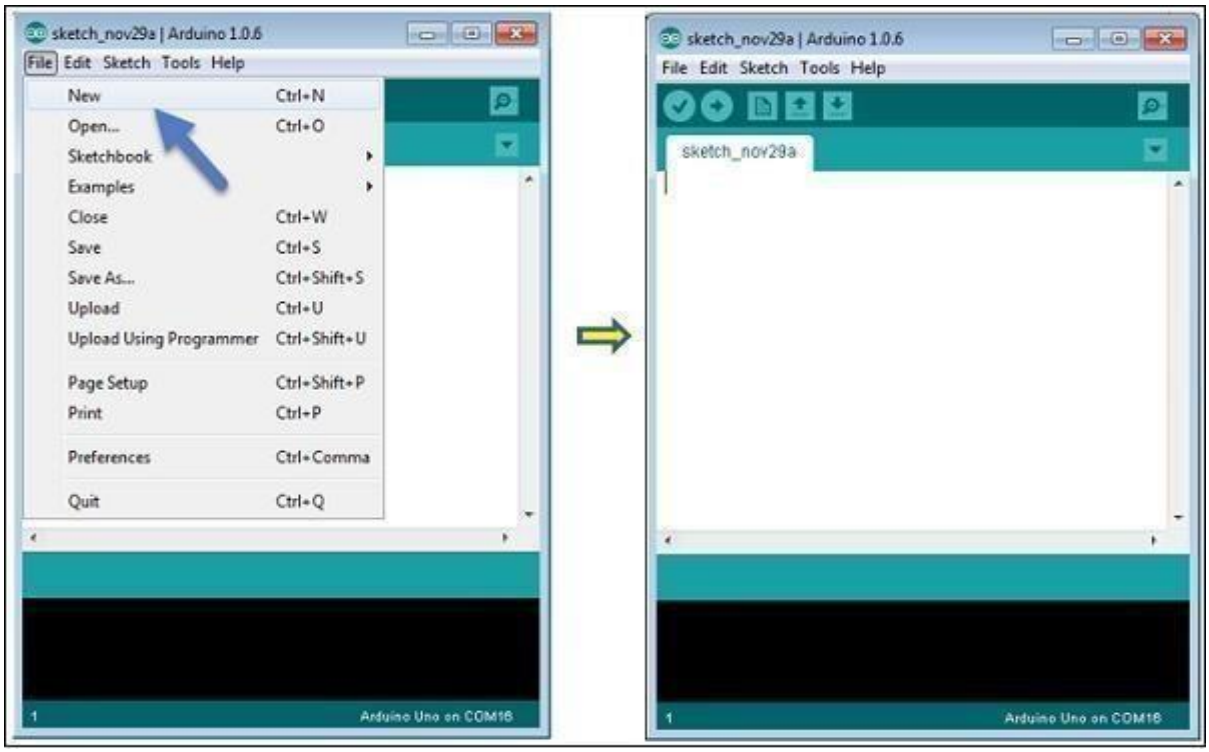
After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open your first project.

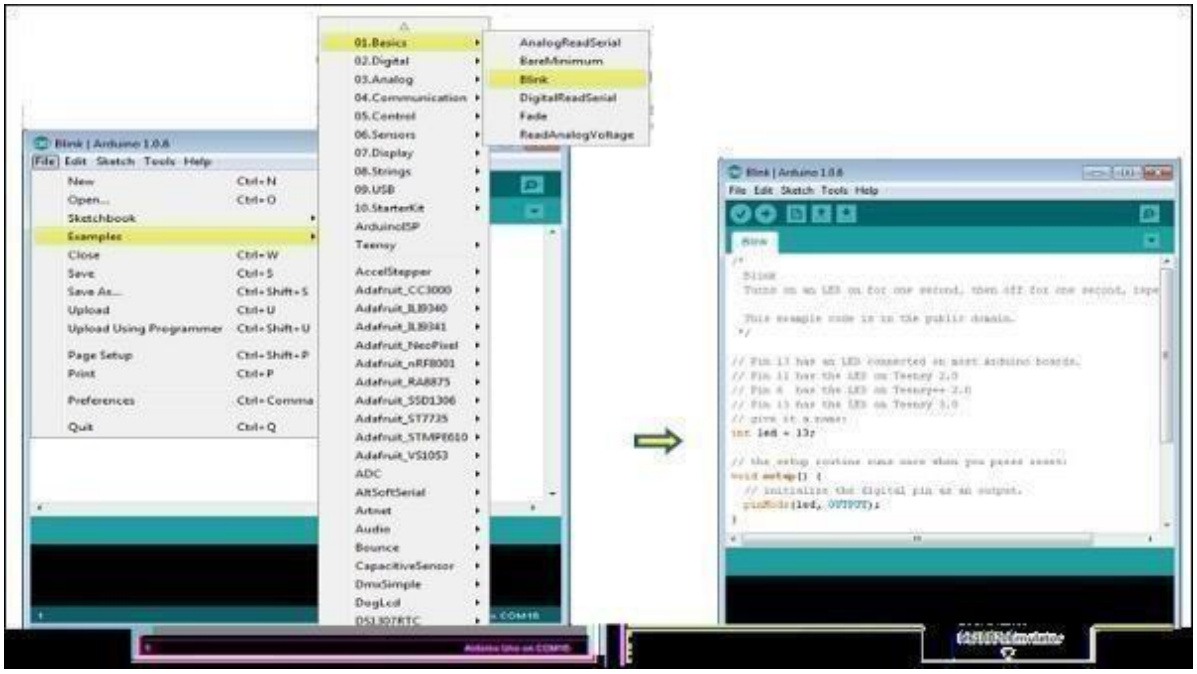
Once the software starts, you have two options –

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

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



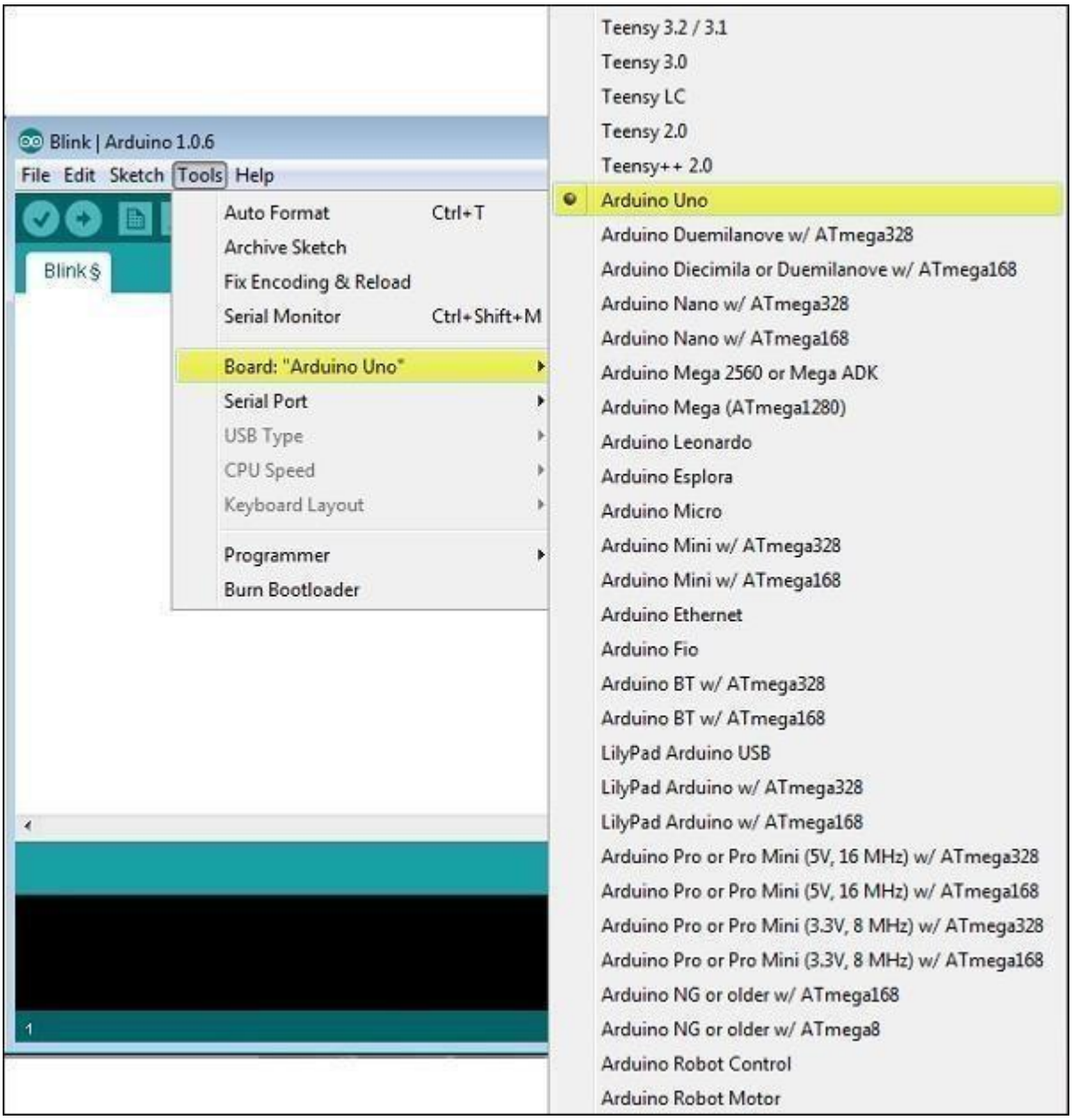
To open an existing project example, select File → Example → Basics → Blink.



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.

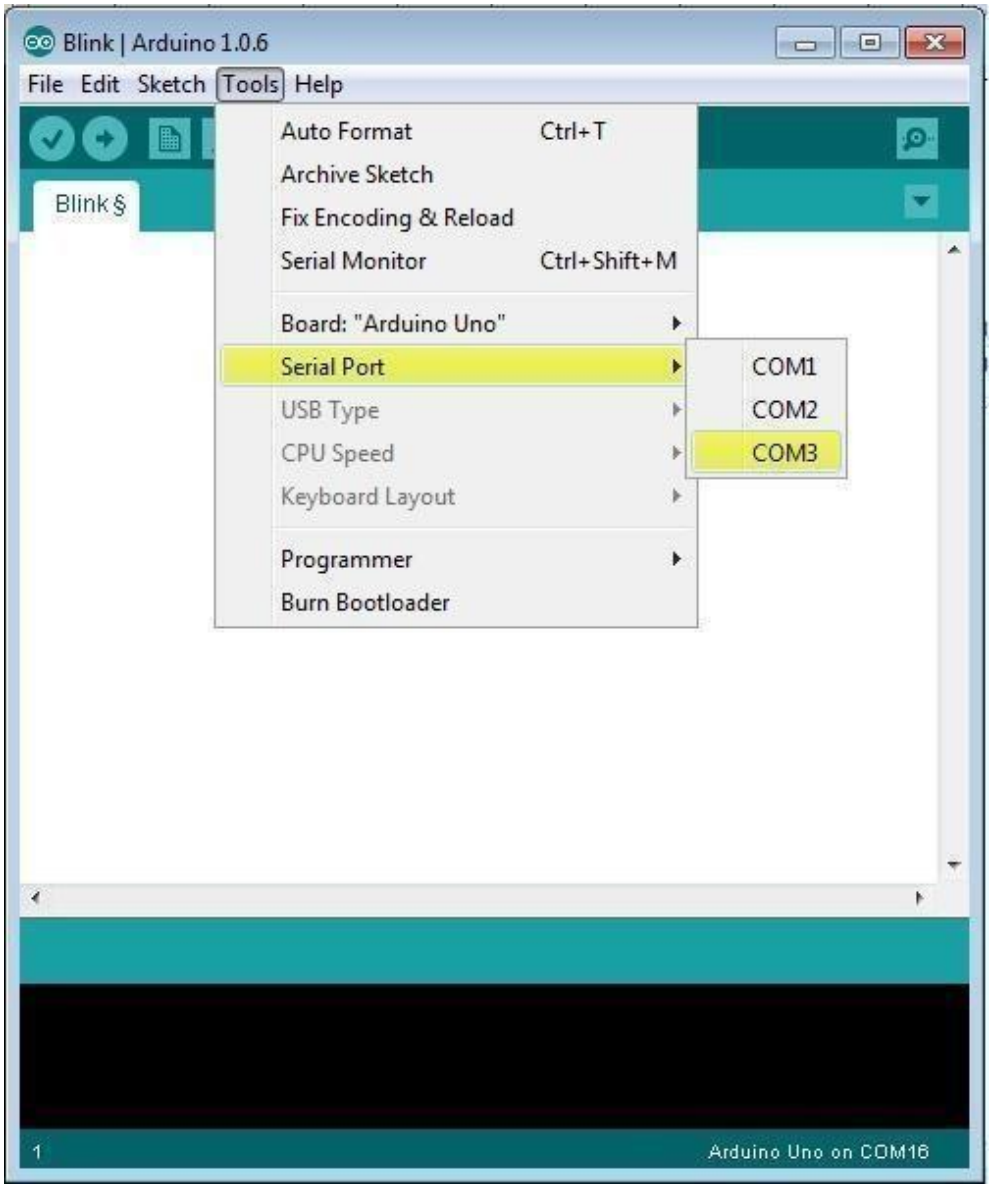
To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer. Go to Tools → Board and select your board.



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

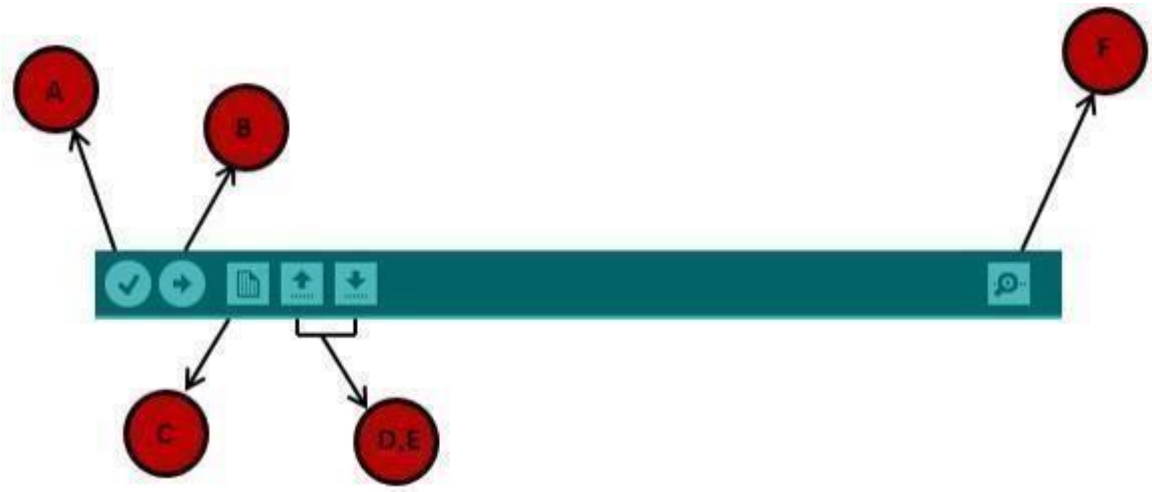
Step 7 − Select your serial port.

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.



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 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.

SOFTWARE:

Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possess 2D CAD drawing feature. It deserves to bear the tagline “From concept to completion”.

About Proteus

It is a software suite containing schematic, simulation as well as PCB designing. 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.

Features

ISIS has wide range of components in its library. It has sources, signal generators,measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps,discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

Starting New Design

Step 1: Open ISIS software and select New design in File menu

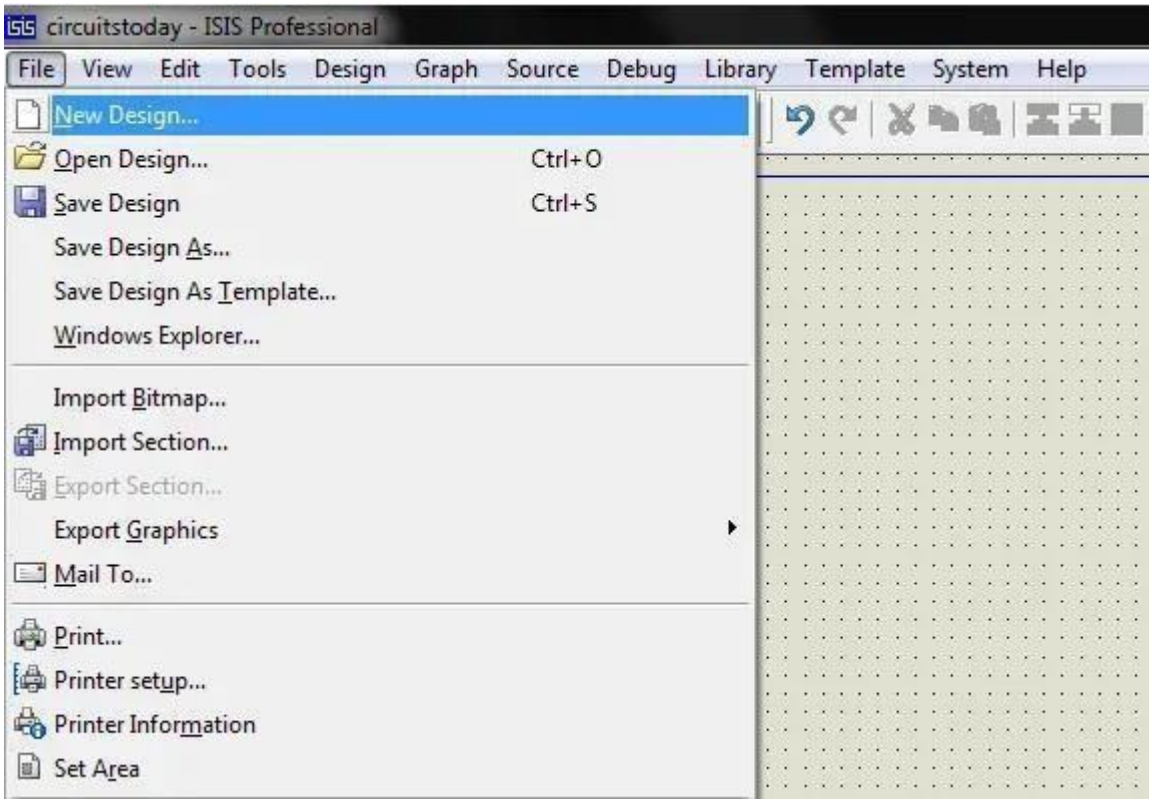


Fig : Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a PopUp appears asking to select the template. It is similar to selecting the paper size while printing. For now, select default or according to the layout size of the circuit.

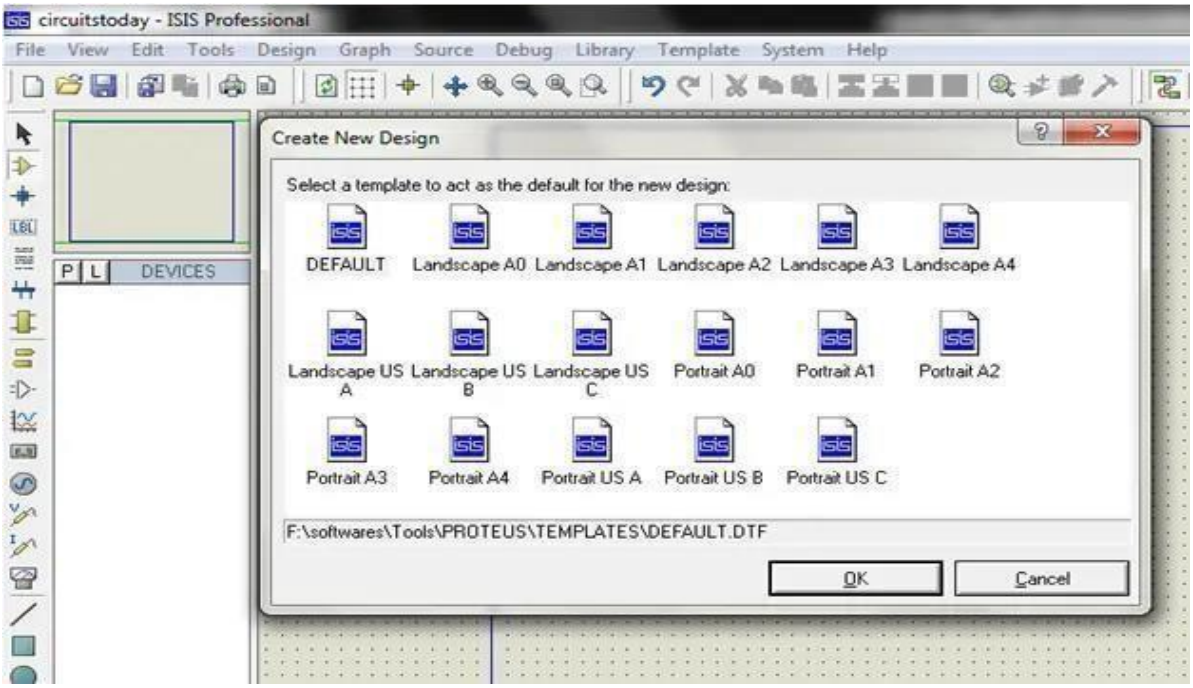


Fig : Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

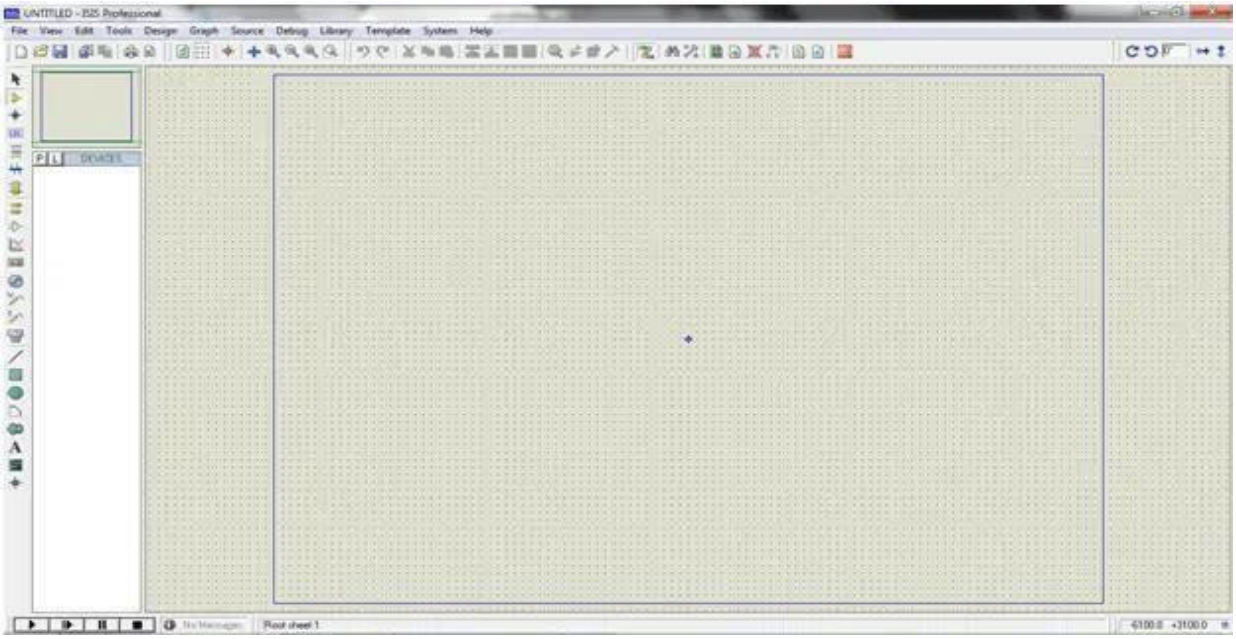


Fig : Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

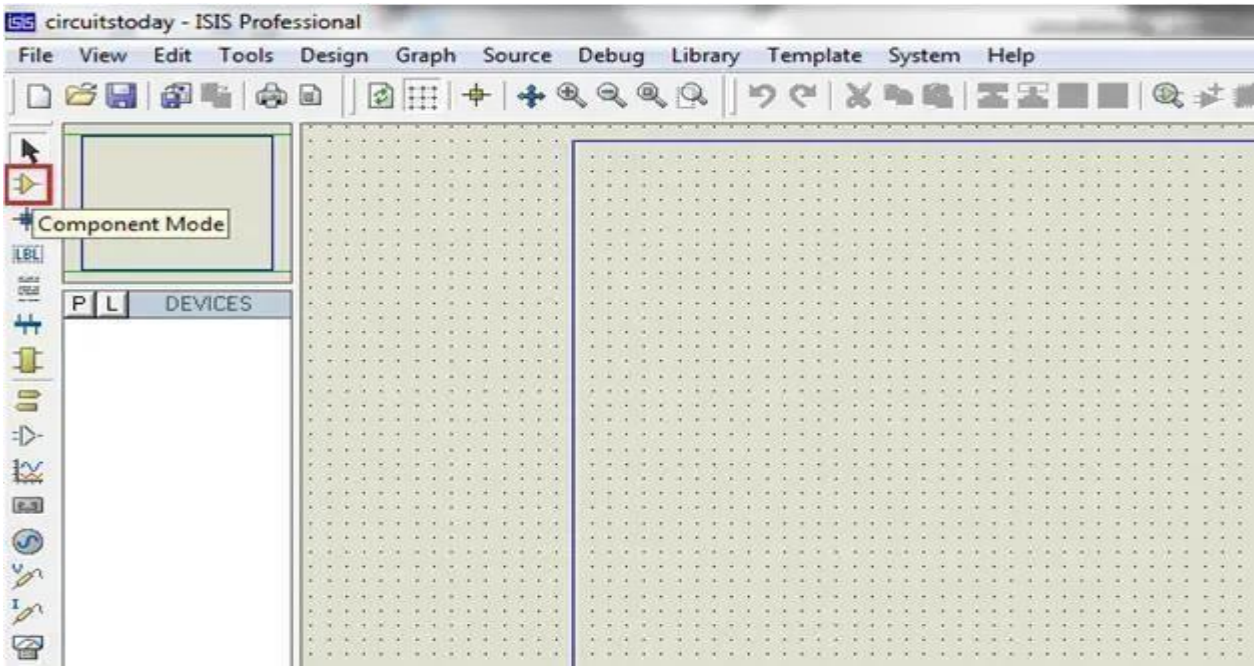


Fig : Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option and enter a part name.

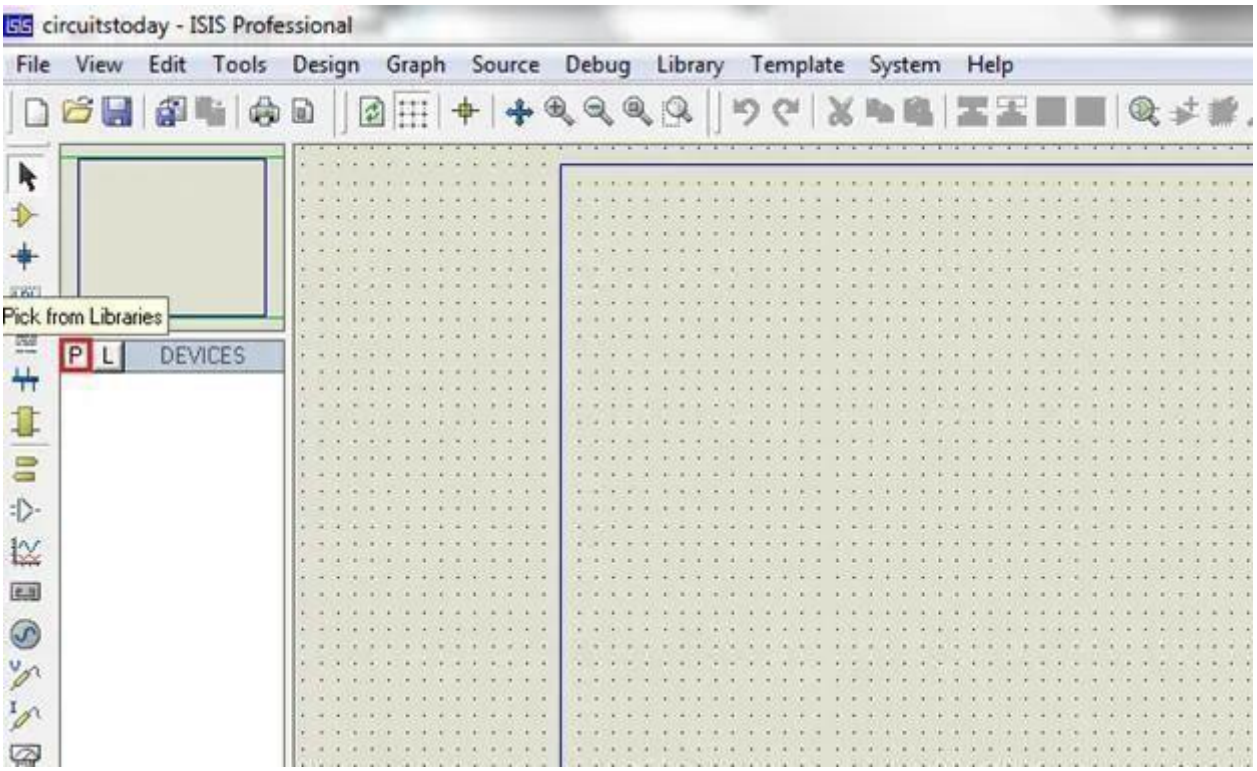


Fig : Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

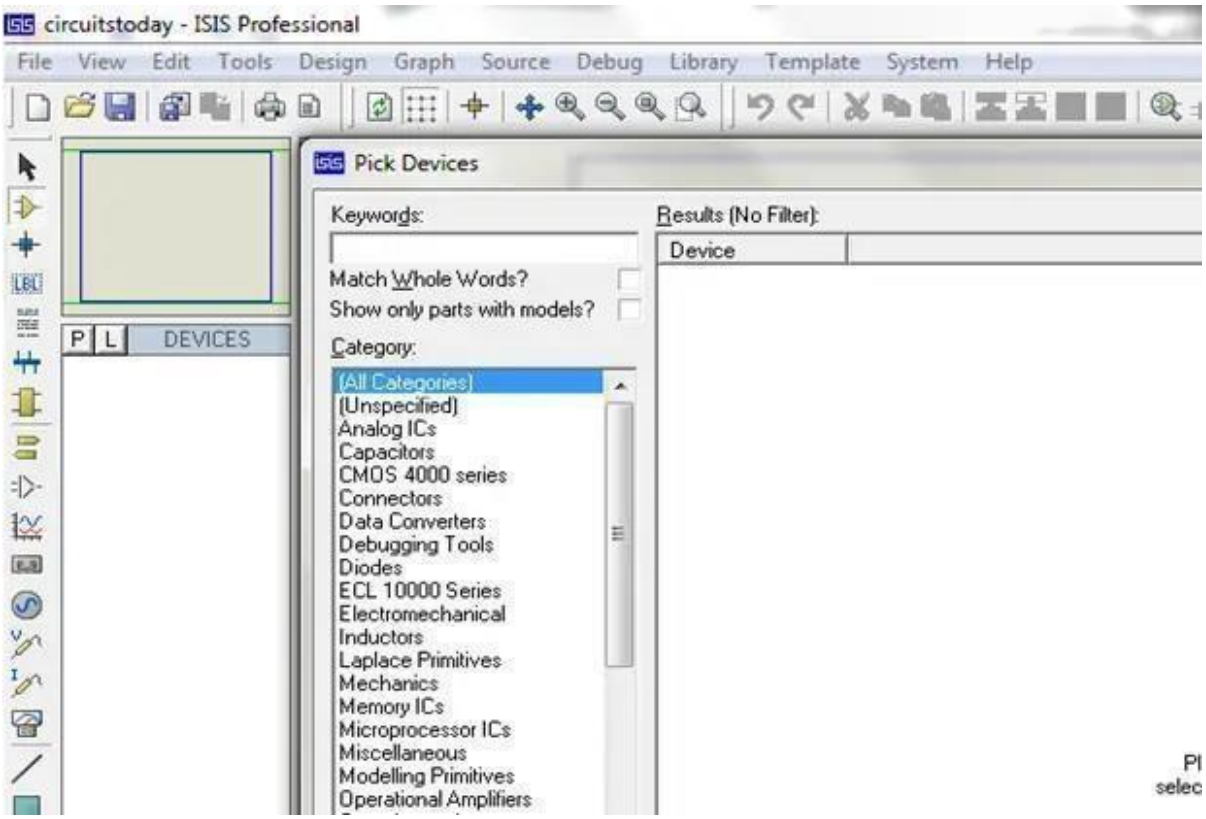


Fig : Keywords Textbox

Example shows selection of push button. Select the components accordingly.

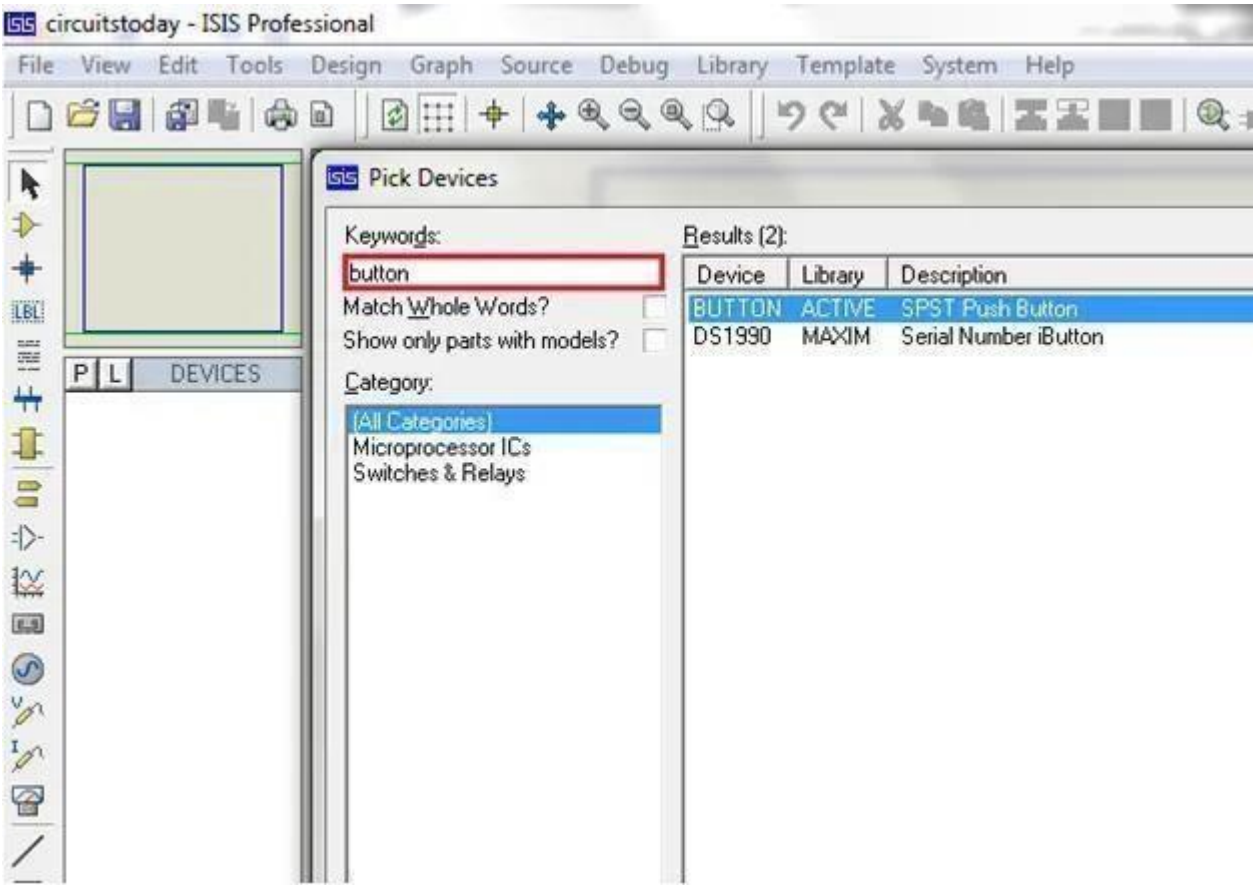


Fig: Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

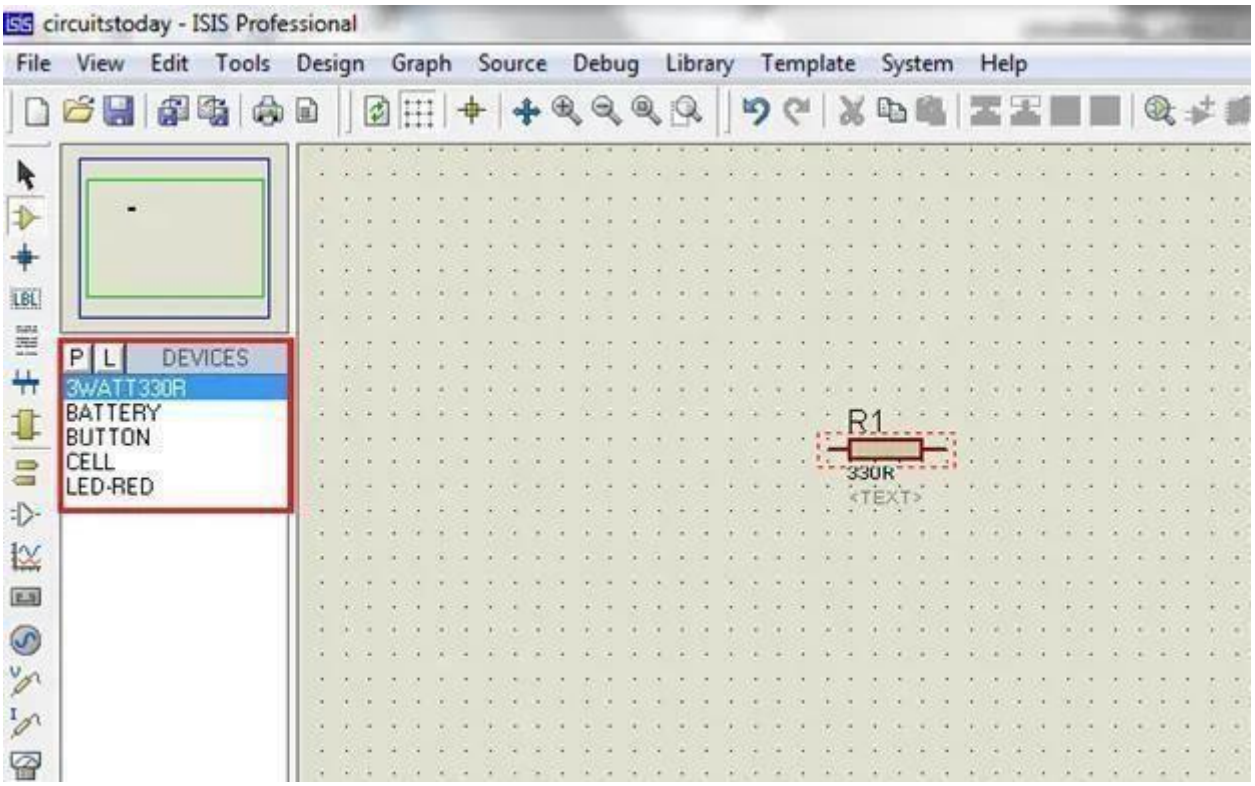


Fig : Component Selection

Step 8: Place all the required components and route the wires i.e., make connections.Either selection mode above the component mode or component mode allows to connect through wires.

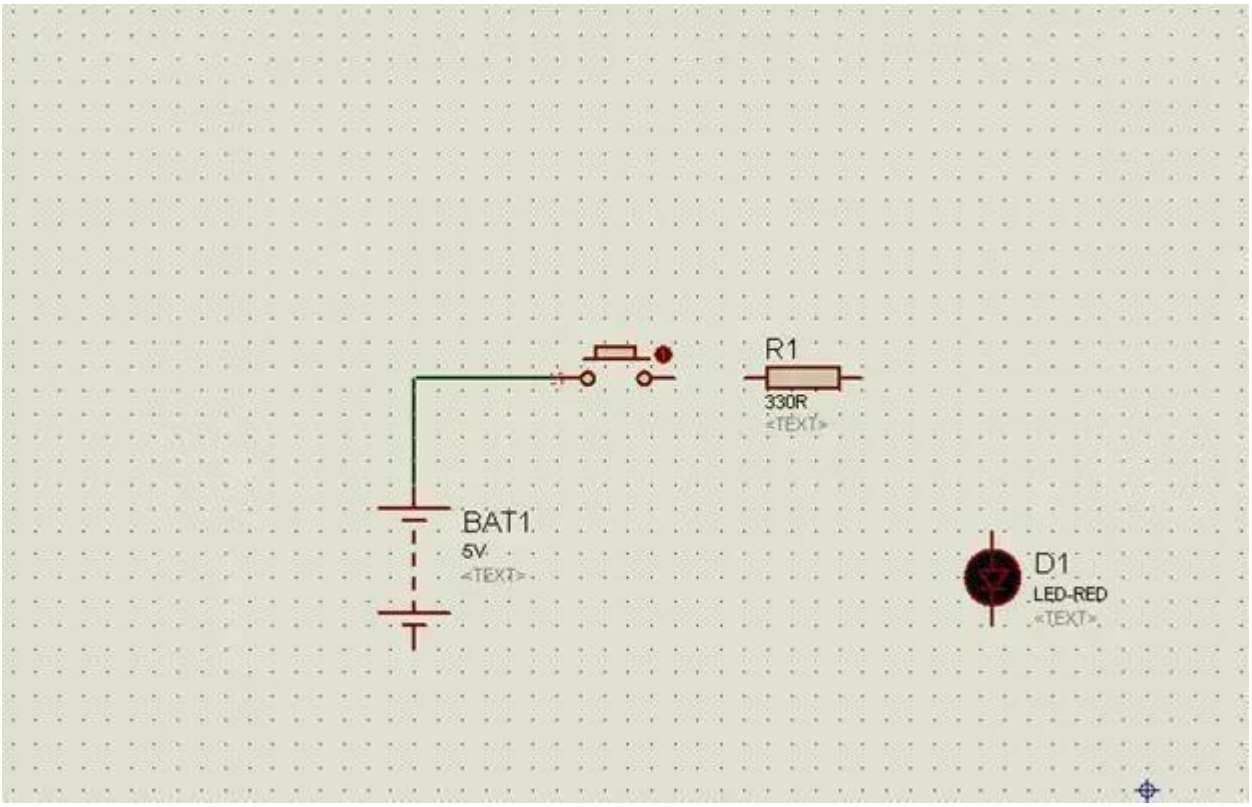
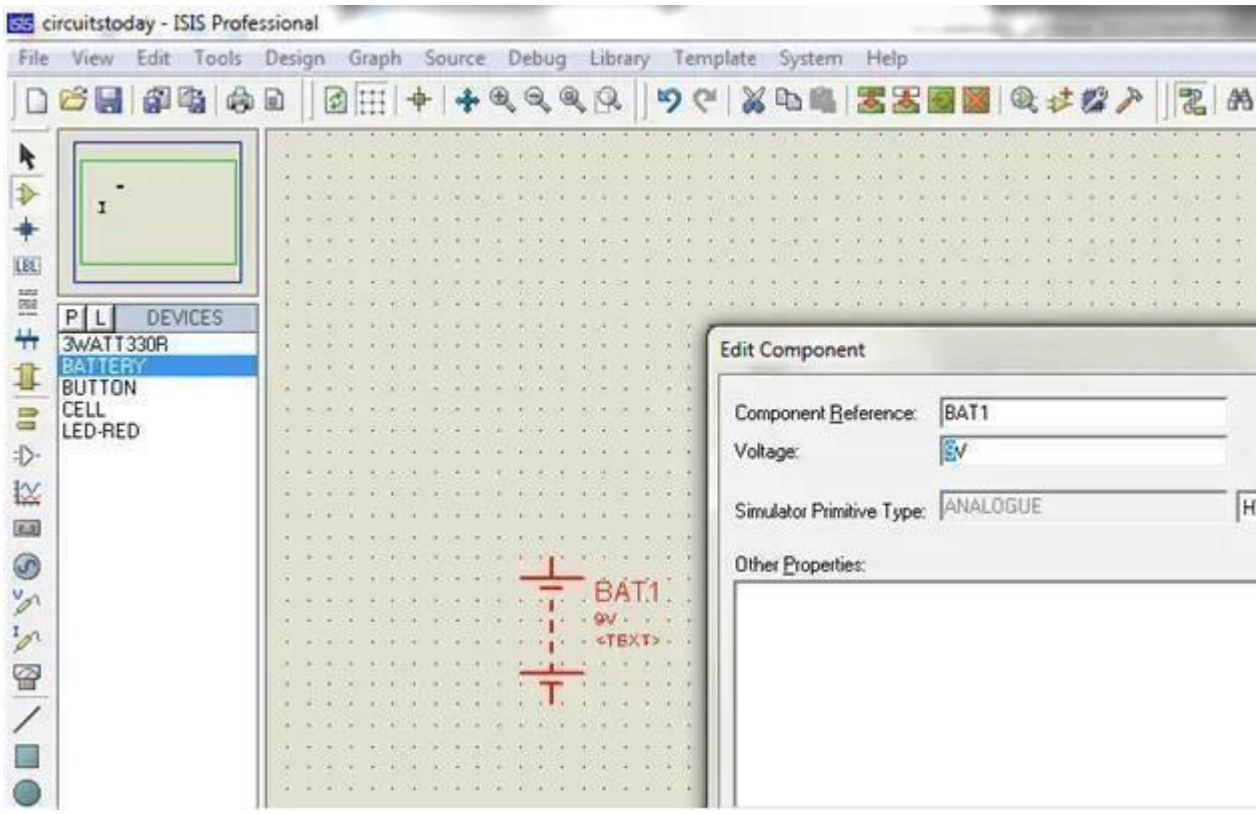


Fig: Component Properties Selection

Step 9: Double click on the component to edit the properties of the components and click on Ok.



Step 10: After connecting the circuit, click on the play button to run the simulation.

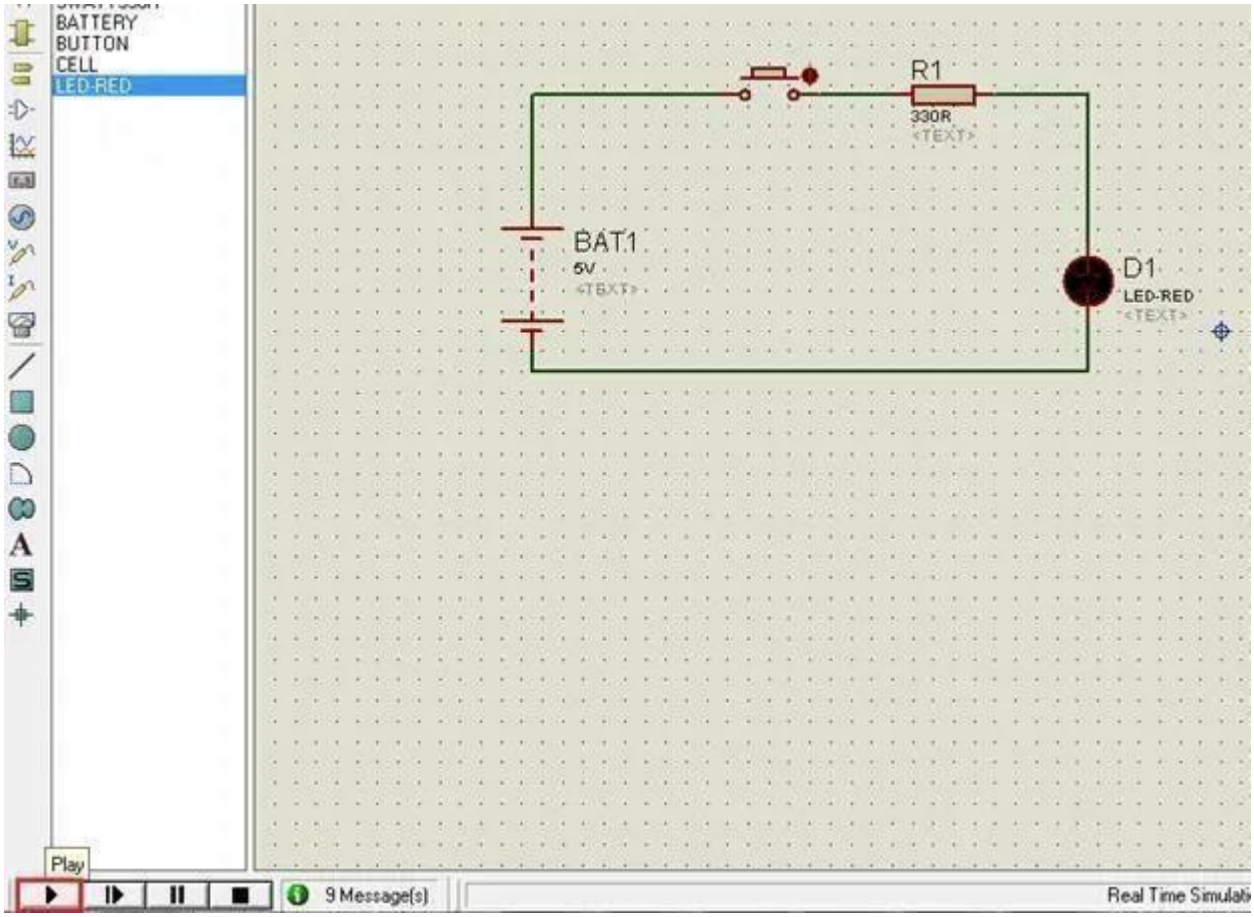


Fig: Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

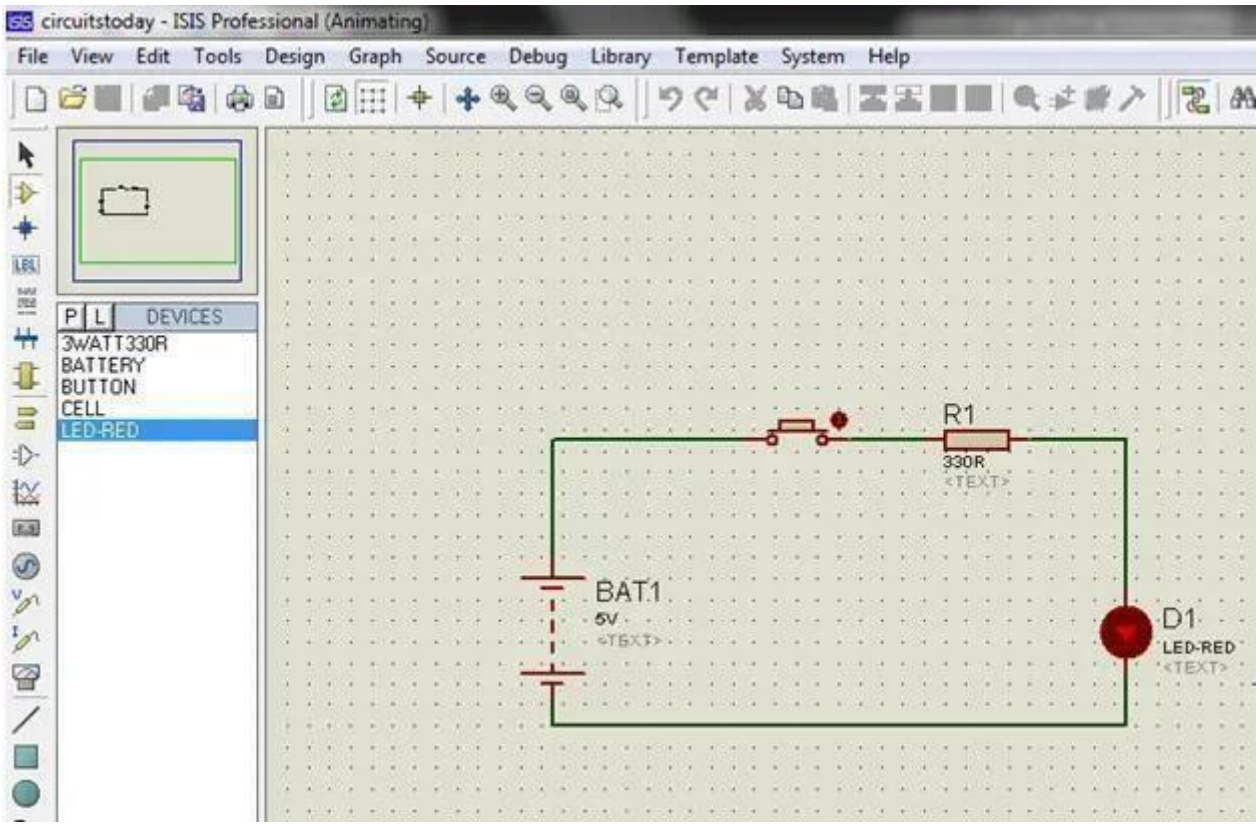


Fig : Simulation Animating

Simulation can be stepped, paused or stopped at any time.

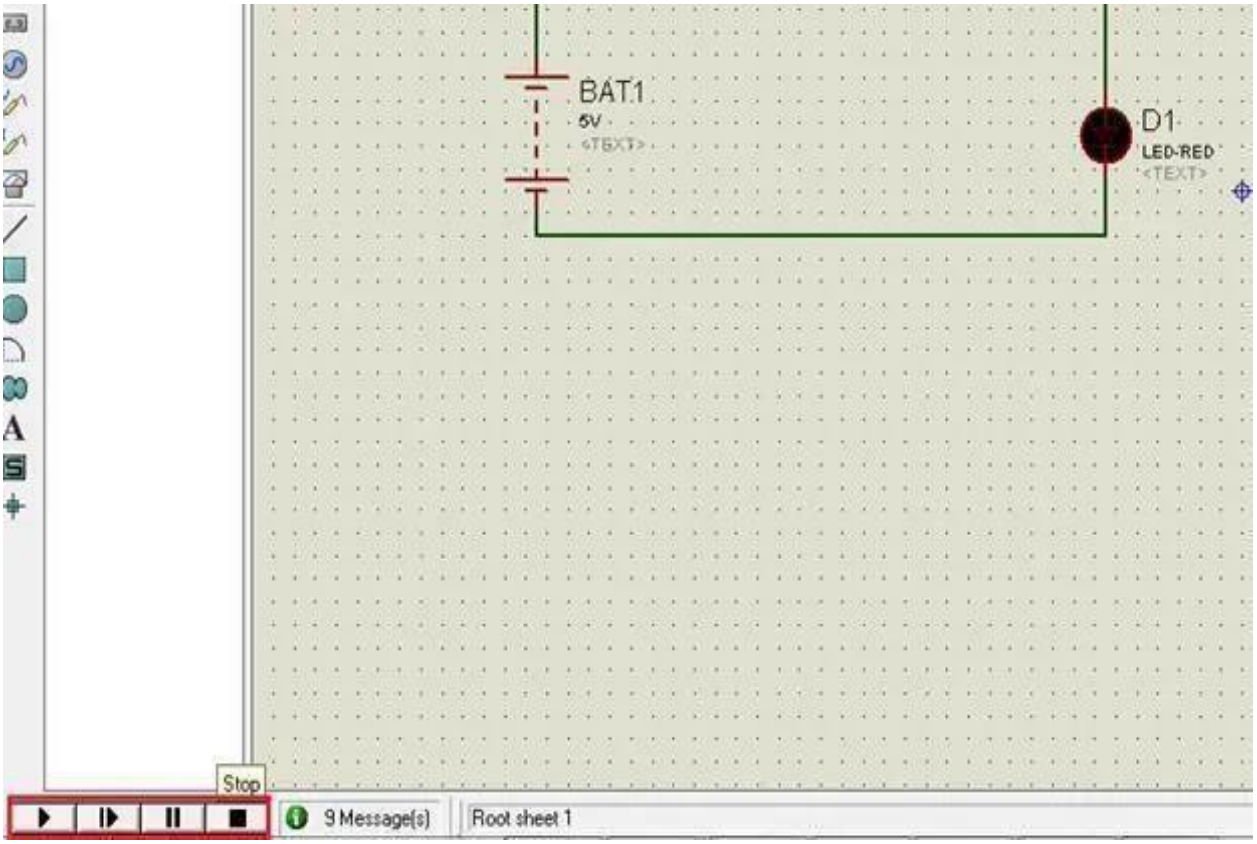


Fig: Simulation Step-Pause-Stop