GITAM

(DEEMED TO BE UNIVERSITY)

(Estd. u/s 3 of the UGC Act, 1956)

VISAKHAPATNAM • HYDERABAD • BENGALURU

NAAC accredited with A+ Grade

G.U. Regd. No. 122010325012

Roll No. 122010325012

CERTIFICATE

Certified that this is the bonafied record of practical work done in

13 *Experiments by* S. Jahnavi

of B.Tech./M.Tech./B.Arch./M.Sc./M.C.A./B.B.M./B.B.A./M.B.A./ M.H.R.M./LLB/BA, LLB, B.Sc., B.Com (Hons), B.Pharm, M.Pharm in

the Internet Of Things *Laboratory of*

Department of CSE (CS)

During the year 2021-22

Signature of Head of the Department

Date :

Signature of Laboratory In-Charge

Date :

INDEX

Name : S. Talwar

Class & Branch : B.Tech, CSE (CS)

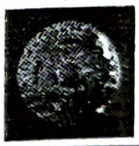
Expt. No.	Date	Title of the Experiment	Page No.	Date of Submission	Teacher's Signature	Remarks
1.		LED Control using Arduino	1-3			
2.		Temperature Sensor Data Reading	4-9			
3.		Control DC Motor	10-15			
4.		RGB Sensor	15-20			
5.		GAS Sensor	20-28			
6.		Home Automation using sensor	28-31			
7.		Sending Data to cloud	31-45			
8.		Build an ESP8266	45-48			
9.		DTH11 with Arduino	48-53			
10.		RGB LED Module	53-57			
11.		Heart Beat Sensor	57-60			

INDEX

Name : _____

Class & Branch : _____

Expt. No.	Date	Title of the Experiment	Page No.	Date of Submission	Teacher's Signature	Remarks
12.		Controlling LEDs using Thingspeak and NodeMCU	64-70			
13.		Raspberry Pi OS Installation	71-80			



Experiment No. 1

SHEET No. 1

Aim: Control the LED with Arduino Board and tinkercad software.

Objectives: Student should get the knowledge of Arduino Board and control of output device (LED)

Outcomes: Student will be Write program using Arduino IDE for Blinking LED

Hardware Requirements:

- 1 x Breadboard
- 1 x Arduino Board
- 1 x LED
- 1 x 330Ω Resistor
- 2 x Jump wires

Procedure:

1. Create a new Account in www.tinkercad.com & login with existing Gmail account.
2. click on goto create project & create a new project.
3. Open the project 1 and add a description of your project as LED blinking
4. Go to Create menu and select circuit.
5. Select the Arduino and breadboard and place it in design area.
6. Select the component LED and resistor, make connections as shown in above fig. Configure the resistor value as

Preferences

Settings Network

Sketchbook location:

C:\Users\sail\Documents\Arduino

Browse

Editor language: System Default: (requires restart of Arduino)

Editor font size: 14

Interface scale: Automatic % (requires restart of Arduino

Theme: Default theme (requires restart of Arduino)

Show verbose output during: compilation upload

Compiler warnings: None

- [ ] Display line numbers
- [x] Verify code after upload
- [x] Check for updates on startup
- [ ] Use accessibility features
- [ ] Enable Code Folding
- [ ] Use external editor
- [x] Save when verifying or uploading

Additional Boards Manager URLs: https://arduino.esp8266.com/stable/package_esp8266com_index.json

C:\Users\sail\Documents\ArduinoData\preferences.txt

OK Cancel

330 ohms.

7. Attach the LED to an output pin of the Arduino D13

8. Once the circuit connections are ready, programming the Arduino can be done in three ways.

- Using Codeblocks
- Using Codeblocks + text programing
- With text programing

Now from the Code menu select blocks. The default program to blinking led with code blocks is provided. Press start simulation, you'll notice the blinking of LED with 1Sec.

Now from start code menu select the text programing mode and plate the below program to blink the led.

```
int led = 13;

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

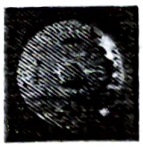
My Channels

Click Here

Name | Created | Updated

Help

Examples



Let's try using a different pin of the Arduino. Say D7. Move the red jumper lead from pin D13 to pin D7 and modify the following line near the top of the Sketch:

```
int led = 13;
int led = 07;
```

Upload the modified sketch to your Arduino board and the LED should still be blinking but this time using pin D7.

30/5/22

Experiment No. 2

SHEET No. 4

Aim: Read data from a sensor. Experiment with both analog and digital sensors.

Objectives: Student should get the knowledge of Temperature Sensor & IR sensor.

Outcomes: Student will be developed programs using Arduino IDE and Arduino Board for Temperature sensor & IR sensor.

Theory:

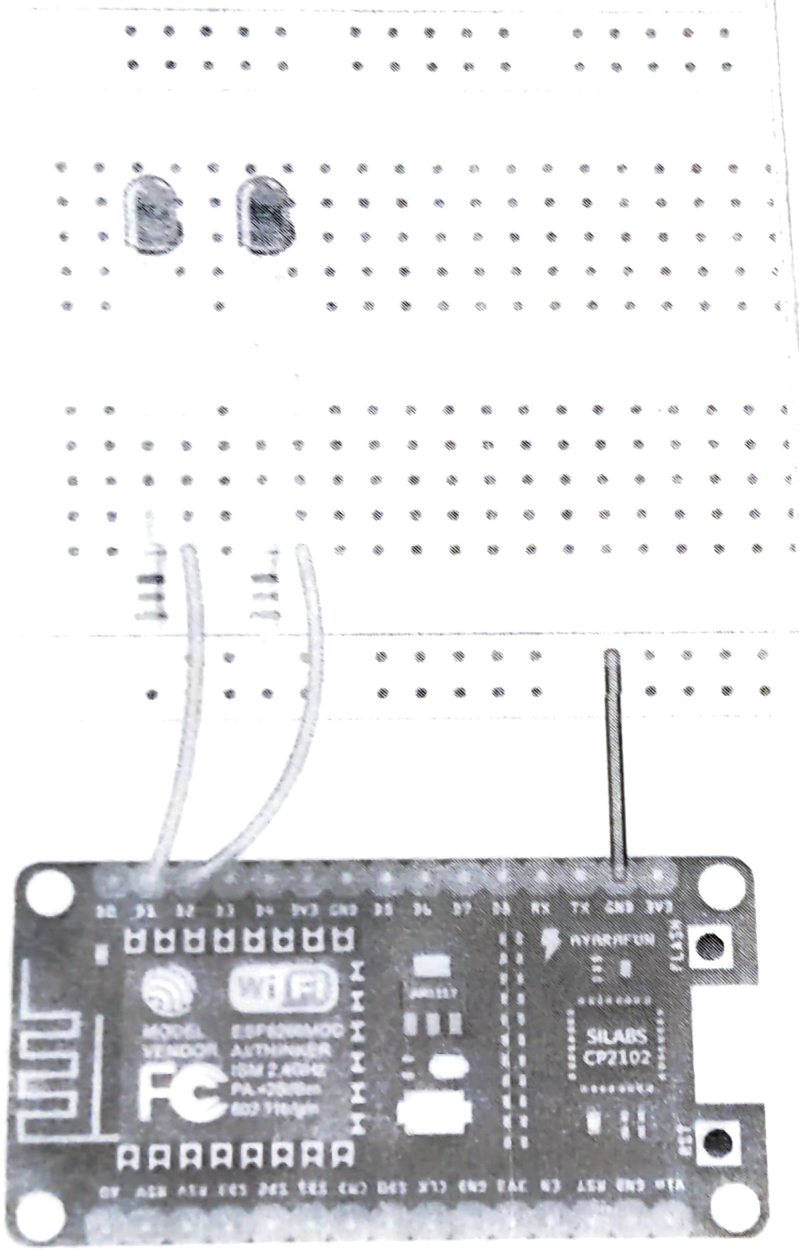
connecting to Temperature Sensor:

There sensors have little chips in them and while they're not that delicate, they do need to be handled properly. Be careful static electricity when handling them and make sure the power supply is connected up correctly and is between 2.7 and 5.5V DC - so don't try to use a 9v battery.

They come in a "TO-92" package meant the chip is housed in a plastic semi-cylinder with three legs. The legs can be bent easily to allow the sensor to be plugged into a breadboard. You can also solder to the pins to connect long wires.

Reading Analog Temperature Data:

* The sensor has 3 pins, 2 pins are used to power the sensor & third one is the analog output. Remember that you can use anywhere 2.7V & 5.5V as the power supply.


WiFi
SILABS
CP2102

No matter what supply you use, the analog voltage reading will range from about 0V to 1.75V
Voltage at pin in millivolts = (reading from ADC) * 5000/1024
This formula converts the number 0-1023 from the ADC into 0-5000mV (=5V) If you're using a 3.3V Arduino.
voltage at pin in millivolts = (reading ADC) * (3300/1024)

Centigrade temp = [(analog voltage in mV) - 500]/10

Simple Thermometer

Part-I

Hardware Requirements:
- 1 × Breadboard
- 1 × Arduino Uno
- 1 × Temprature Sensor
- 1 × 330Ω Resistor
- 2 × Jumper wires.

1. Connect the circuit as show in fig.
2. Create a variable name reading and acquire the analog value of A0 into it.

Code:

```
int SensorPin = 0;

void setup()
{
```

```
    Serial.begin(9600);
}

void loop()
{
int reading = analogRead(SensorPin);
float voltage = (reading * 5.0) / 1024.0

Serial.print(voltage); Serial.println("volts");
float temperatureC = (voltage - 0.5) * 100.
Serial.print(temperatureC); Serial.println("degrees C");

float tempF = (temperatureC * 9.0/5.0) + 32.0;
Serial.print(tempF); Serial.println("degrees F");
delay(1000);
}
```

Part - II

Components Required:

There are the part you need to built the thermometer

1 x Arduino UNO Board
1 x Breadboard
1 x 16x2 LCD display
1 x LM36 & LM35 analog temperature sensor
1 x 10K potentiometer - for adjusting the brightness of the display
Jumpers - to connect everything.

Step 1: Building the thermometer
By following the Fritzing schematic above, plug the LCD in the breadboard and then connect it to the Arduino board with jumpers. After that plug the potentiometer and the sensor in the breadboard, connect the left and the right pins.

Step 2: Programming the Arduino
To get it work you have to use one of the two codes below. Upload it to your Arduino using the integrated development environment, for short IDE.

Code:

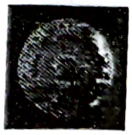
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

int sensor_input;
float temp;

void setup()
{
   Serial.begin(9600);
   lcd.begin(16, 2);
}

void loop()
{
```

```
measure_temp();
display_temp();
}
void measure_temp()
{
  Sensor_Input = analogRead(A0);
  temp = (float) sensor_Input/1024;
  temp = temp * 5;
  temp = temp - 0.5;
  temp = temp * 100;
}
void display_temp()
{
  lcd.setCursor(0,0);
  Serial.print("Temp in Celcius");
  Serial.print(temp);
  Serial.println();
  lcd.print("Temp in Celcius");
  lcd.setCursor(5,1);
  lcd.print(temp);
  delay(500);
}
```

Part - IV

Components Required:

* Arduino UNO
* TMP36 (Temperature sensor)
* Resistor - 03



* LED - Green, Yellow, Red
* Connecting wires

Connection:
* Connect VCC pin of the TMP36 sensor to the 5V pin
* Connect GND pin of TMP36 sensor to the GND terminal
* Connect the green LED anode pin to pin no. of 4
* Connect the yellow LED anode pin to no. 3
* Connect the RED LED anode pin to pin no. 2
* Connect cathode of all the LEDs to Arduino GND.

Code Explanation:
* After declaring the variables drag out set block and set block and set to the following blocks Step to set the variable to "map" of converting.
* Step 1: In the Math section, take out a "map" block and merge two arithmetic blocks inside the first field.
* Step 2: Set the temperature range -40 to 125.
* Step 3: Go to input section & take out an "analog read pin A0" block, fix it into the first arithmetic block inside the "map" block.
* Step 4: Now set the math's blocks to "(read analog pin A0 - 20) x 3.04"

Conclusion: We had developed programs using Arduino IDE and Arduino Board for temperature sensor in different mode

30/5/22

Experiment-3

SHEET No. 10

Aim: To control DC motor with H-Bridge and PWM

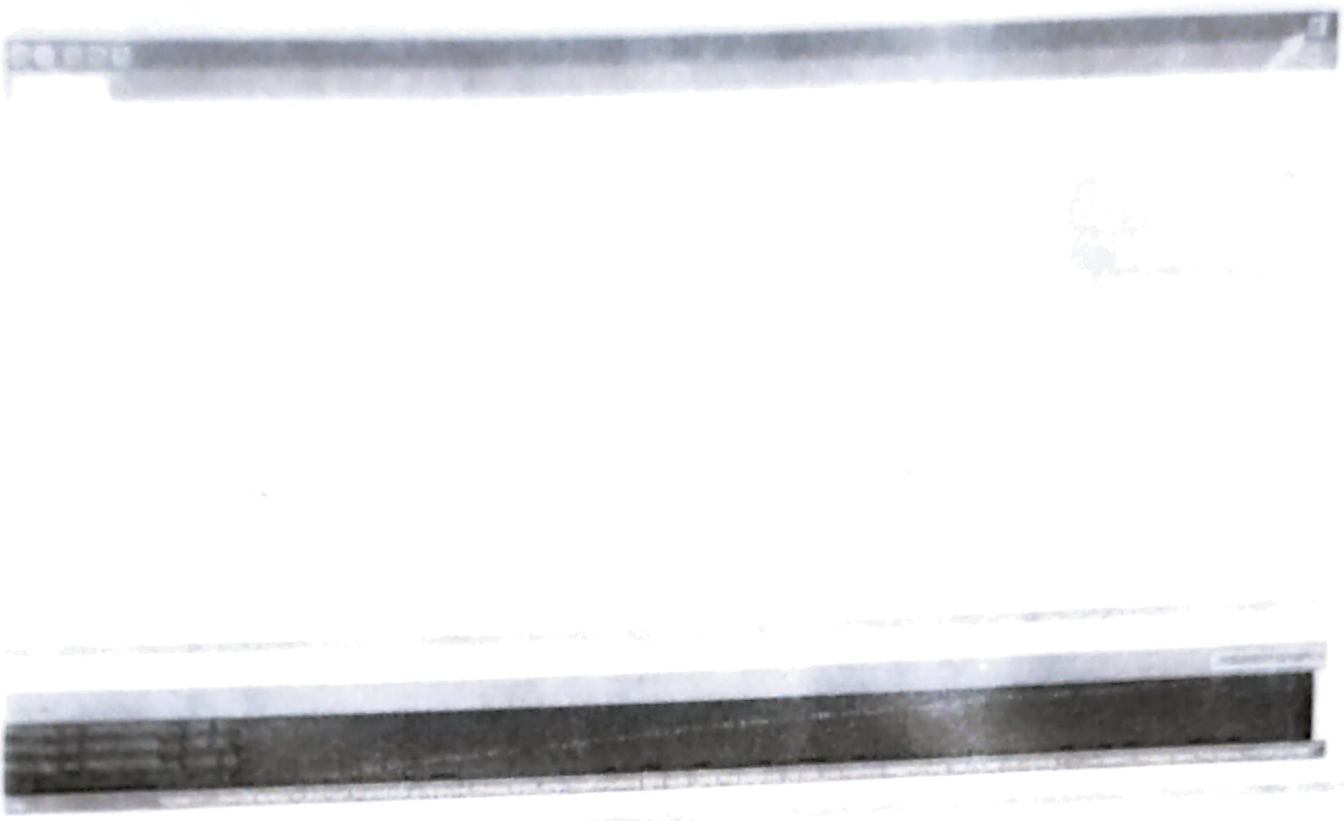
Objectives: Student should get the knowledge of DC motor.

Outcomes: Student will be developed programs using Arduino IDE and Arduino Board for control DC motor.

Hardware Requirements:
- 1 × DC Motor
- 1 × Arduino Uno
- 1 × H-Bridge motor driver
- Jumper wires

Theory:
The H-bridge is an electronic circuit that looks like the letter H. An H-bridge is used to drive a load, such as a brushed DC motor, in both directio

Applications
* Most important application of the H-Bridge circuit as the motor driver circuit
* In Robotics, H-Bridge is used.
* It is also used in Inverter circuit
* In modern battery charger for electrical vechicles.



Procedure:

* First, we have to place Arduino board, L293D and DC motor
* Give connections as shown in fig.
* Connections:
  -> 5v - 8th pin & also short 8th pin with 16th pin
  -> Both grounds - 4th, 5th pins.
  -> 10th pin - 1st pin
  -> 6th pin - 2nd pin
  -> 7th pin - 7th pin
  -> Red - 6th pin (output 2)
  -> Black - 3rd pin (Output 1)

Code:

```
void setup()
{
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(10, OUTPUT);
  digitalWrite(10, HIGH);
  Serial.begin(9600);
}
void loop()
{
  if(Serial.available())
  {
    char ch = Serial.read();
```

```
if (ch == 'a')
{
  digitalWrite(6, HIGH);
  digitWrite(7, LOW);
}
else if (ch == 'c')
{
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
}
else
{
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
}
}
}
```

Hardware Requirements:

- 1 x DC motor
- 1 x Arduino Uno
- 1 x NPN transistor (BJT)
- Jump Wires
- 4 x AA Batteries
- 1 x resistor (50 ohms)

Theory:

PWM is an efficient way to vary the speed and power of electric DC motors. Here two drivers are described for 24V (15 V to 30 V) motors with a max current upto 80A. These drivers can for example to be used to vary the speed of small eletric vechicles. The first driver use a power MOSFET transistor switched at 20 kHz. The second method is preferred due to its higher frequency.

Applications:

PWM is used in many applications, ranging from communication to power control and conversion. For example, the PWM is commonly used to control the speed of eletric motors.

Procedure:

1) First we have to place Arduino board, four - 1.5v AA Battery, a resistor, a NPN transistor and a DC Motor on the design area.
2) Next step is to give connection as shown Fig.
3) Connections
   1) Resistor terminal 1 - D3 of Arduino
   2) Resistor terminal 2 - base of NPN transistor.
   3) Collector of NPN transistor.
   4) Emitter of NPN transistor.
   5) Terminal 1 of DC motor.
4) Code:

```
int pwmpin = 3;
```

```
int val = 20;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  analogWrite(pwmpin, val);
}
```

Output:

565 rpm - 655 rpm

Result: Hence, the control of DC motor using Arduino with both H-Bridge and PWM is shown.

30/5/22

Experiment-4

Aim: To control LED with Temperature Sensor.

Objectives: Student should know how to assemble a temperature-controlled LED with Arduino

Outcomes: Student will be developed programs using Arduino IDE and Arduino Board for LED with Temperature Sensor.

Hardware Requirements:
- 1 x Arduino Uno
- 3 x Resistors
- Jump Wires
- 1x Temperature Sensor
- 1x LED

Applications of Temperature Sensor:
* Home appliances - Kettles, toasters, washing machines, dishwashers and coffee machines will all contain temperature sensors.
* Computers - within computers there are temperature sensors to ensure the system does not overheat.
* Industrial Equipment - temperature sensors used within these applications will need to be robust as the
* Warming Electrical Radiators: NTC thermistors are used to control the heat on eletric radiators.

Procedure:

1) Place Arduino board, Temperature sensor and LED
2) Give connection.
3) Connections for temperature sensor
- Pin 1 of the temperature sensor goes to 5v
- Pin 2 of the temperature sensor goes to analog pin A0
- Pin 3 of the temperature sensor goes to ground
4) Connections for LED:
- LED is connected to digital pin 11, 9 & 10
- 2nd pin is connected to ground.

Code:

```
int v_TempSen = 0;

void setup()
{
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(A0, INPUT);
}

Void loop()
{
  analogWrite(11, 51);
  analogWrite(10, 204);
  analogWrite(9, 0);
```

```
v_TempSen = -40 + 0.488155 * (analogRead(A0) - 20);
if(v-TempSen >= 50) {
   analogWrite(11, 255);
   analogWrite(10,0);
   analogWrite(9,0);
}
if(v-TempSen >=30) {
   analogWrite(4, 255);
   analogWrite(10, 204);
   analogWrite(9, 0);
}
if(v-TempSen <=10) {
   analogWrite(4, 51);
   analogWrite(10, 102);
   analogWrite(9, 255);
}
delay(10);
}
```

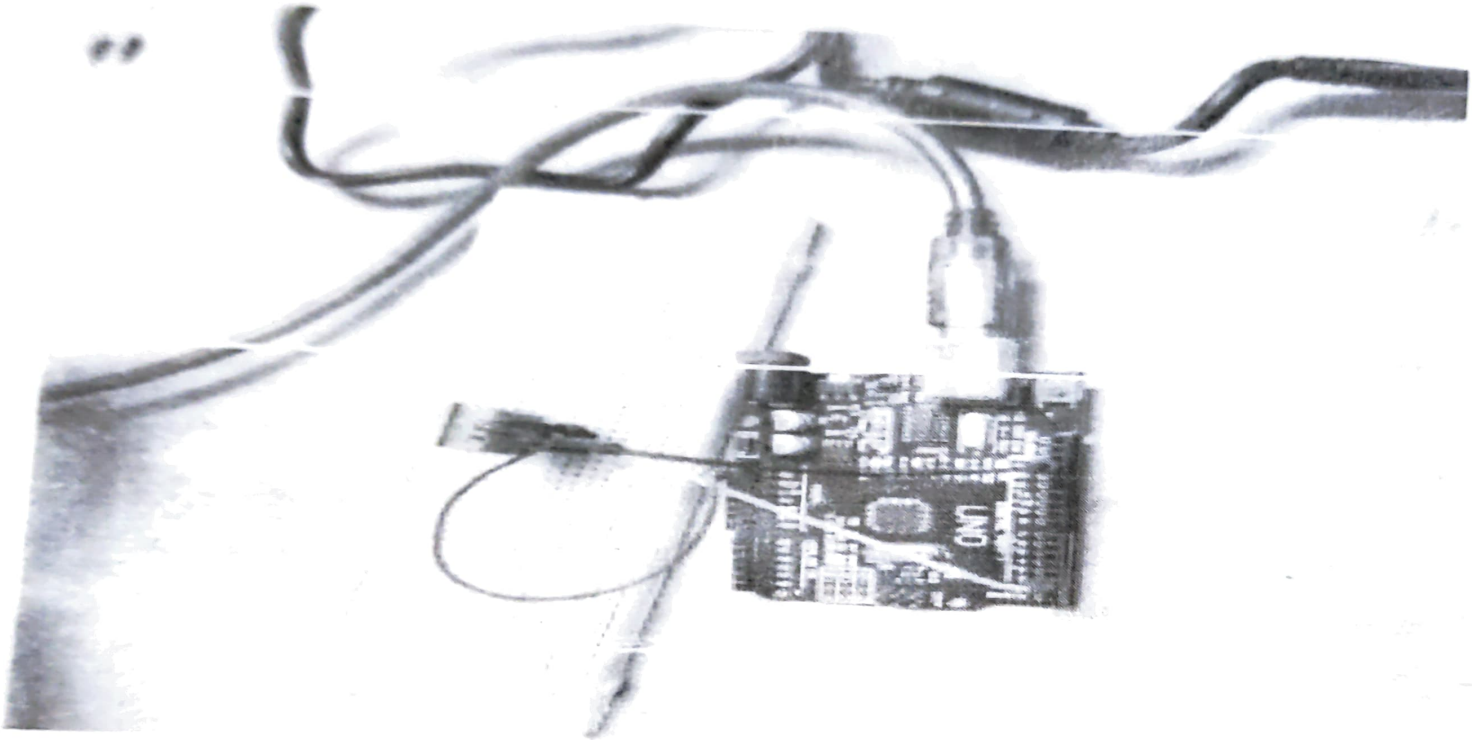
B) Ultrasonic Sensor-based Car parking System:

Aim: To control car parking using Ultrasonic Sensor.

Objectives: Student should have an idea about using ultrasonic sensor at the time of parking a car

Outcomes: To develop an intelligent, user friendly automated car parking system.

Exp 9



Hardware Requirements:
- 1x Ultra Sonic Distance Sensor
- 1x Piezo
- 1x Breadboard
- 3x LED
- 1x A tiny
- 1x Power Supply
- Jump Wires

Applications of Ultrasonic Sensors:
- They are used within food and beverage to measure liqui
- They can be used within manufacturing for an automa
- Used for control maximizing efficiency on the factor
- Used for Anti-Collision Detection, People Detection.

Theory:
Ultrasonic sensors technology is used to observe car parki
The Ultrasonic is installed above each parking lot to chec
the state of the available lot in each park, the use
of ultrasonic sensors of facilitates the implementation of
the high-scale systems at low cost.

Procedure:
1) First, we have to place ultrasonic Distance Sensor,
Piezo, Breadboard, Power Supply and LED.
2) Next step is to give connection as shown in fig.

3) Connections:

- The ground of led, power supply and ultrasonic sensor should be connected to the ground of ATtiny.
- Power of the ultrasonic sensor is connected to power
- Trigger, echo and 2 led are connected to PB 1, PB0, PB5, PB3.
- Piezo is connected to ground and pb4

Code:

```
#define Green 5
#define Yellow 3
#define Red 2
#define buzzer 4

const int trigPin = 1;
const int echoPin = 0;
long duration;
int distance;

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(Green, OUTPUT);
  pinMode(Yellow, OUTPUT);
  pinMode(Red, OUTPUT);
  pinMode(buzzer, OUTPUT); }
```

```
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034/2;

  if(distance >200) {
     digitalWrite(Green, HIGH);
     digitalWrite(buzzer, LOW);
     digitalWrite(Yellow, LOW);
     digitalWrite(Red, LOW);
  }
  if(distance <=200 && distance >100) {
       digitalWrite(Green, LOW);
       digitalWrite(buzzer, LOW);
       digitalWrite(Yellow, HIGH);
       digitalWrite(Red, LOW);
  }
}
```

Result: Hence the use of ultra sonic sensor and temperature sensor is shown in the above experiment

30/5/22

Experiment-5

SHEET No. 21

Aim: To make a circuit using gas sensor and understand its working

Objectives: Student should get the knowledge of gas sensor

Outcomes: Student will develop program using Arduino IDE and Arduino Board for gas sensor.

Hardware Requirements:
- 1 x LCD.
- 1 x Arduino
- 1 x hal sensor
- 1 x Pero sensor
- 4 x resistors
- 2 x led

Theory:
- This sensor is capable of detecting H_2, LPG, CH_4, Alcohol and Smoke
- Due to it's high sensitivity and fast response time, measurements can be taken as quick as desired.
- When any flammable gas flows through this sensor the coil inside this sensor burns and so, the resistance of the coil decreases. Hence, the output voltage starts increasing, which can be detected using a micro controller.

Procedure:

- Place Arduino board the gas sensor, pero sensor & Led in the design area.
- Give connections as shown in the fig.

Code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(5,6,8,9,10,11);
 int redled = 2;
 int greenled = 3;
int buzzer = 4;
 int sensor = A0;
 int sensorThresh = 400;

void setup()
{
  pinMode(redled, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(sensor, INPUT);
  Serial.begin(9600);
  lcd.begin(16, 2);
}
void loop() {
  int analogValue = analogRead(sensor);
  Serial.print(analogValue);
  if (analogValue > sensorThresh)
```

```
void loop(){
  // read without samples.
  byte temperature = 0;
  byte humidity = 0;
  int err = SimpleDHTErrSuccess;
  if ((err = dht11.read(pinDHT11, &temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
Serial.print("Read DHT11 failed, err="); Serial.println(err);delay(1000);
   return;
  }
Serial.print((int)temperature); Serial.print(" *C, ");
Serial.print((int)humidity); Serial.println(" H");
  // DHT11 sampling rate is 1HZ.
delay(1500);
}
```

```
{
  digitalWrite(redled, HIGH);
  digitalWrite(greenled, LOW);
  tone(buzzer, 1000, 300);
  lcd.clear();
  lcd.setCursor(0, 1);
lcd.print("ALERT");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("EVACUATE");
  delay(1000);
}
```

Result:
Hence, here we have used the gas sensor to know how it works and how it reacts in different situation

Experiment- 6

SHEET No. 24

Aim: To build a home automation using sensors.

Objectives: Student should get knowledge of the sensors used in home automation.

Outcomes:
Student will get to know about the purpose of a home automation system is to streamline how the home functions. Considering some of their benifits: Remote Access.

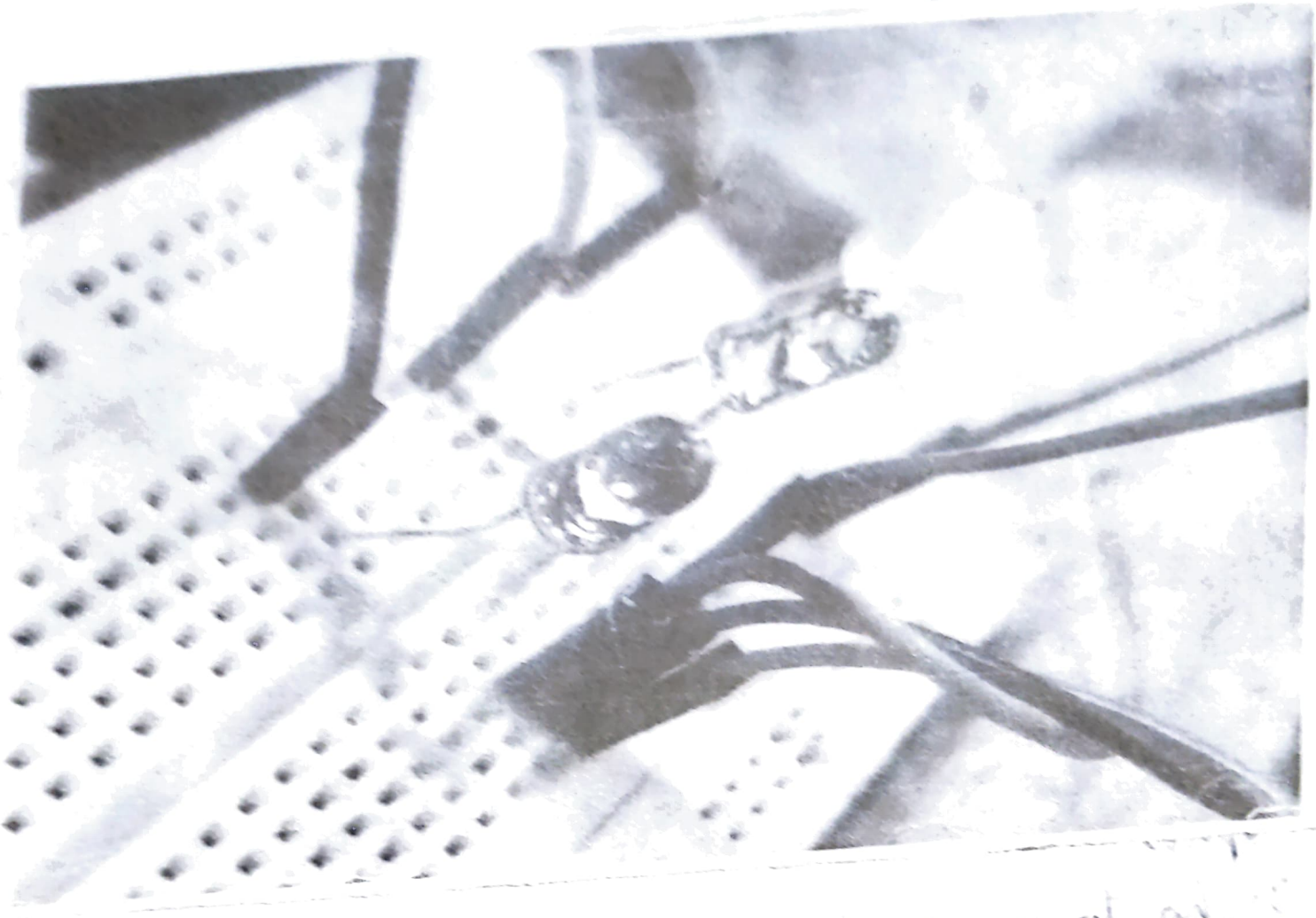
Theory:
Home automation is a network of hardware, communication and electronic interfaces that work to integrate everyday devices with one another via the internet. Each devices has sensors and is connected through WiFi, so you can manage them from your smartphone, or tablet whether you're at home or miles away.
- Sensors can monitor changes in daylight, temperature or motion detection.
- Controllers refer to the devices - personal computers, tablets or smartphones - used to send and receive messages about the status of automated features in your home.
- Actuators may be light switches, motors or motorized valves that control the actual mechanism. They are programmed to be activated by a remote command from a controller.

Hardware Requirements:

- Arduino UNO
- 16*2 LCD display
- Smoke detector MQ6
- PIR
- LDR
- Ultrasonic
- DC Power source
- Relay board of 2 channel

Procedure:

1. Open Tinkercad, then login with your Tinkercad account
2. Go to circuits module and click on create new circuit option.
3. Go to components and search for Arduino.
4. Place Arduino in your workspace.
5. Now search for ultrasonic sensor as well as a servo motor.
6. Connect 5v of servo to 5v of Arduino.
7. Connect GND to Arduino GND.
8. Connect 5v of ultrasonic sensor to 5v of Arduino.
9. Connect GND of ultrasonic sensor to GND of Arduino
10. Connect Trigger pin to Arduino pin D6
11. Connect Signal pin of servo to pin D7 of Arduino
12. Import a PIR sensor from the components menu.
13. Import a DC motor.
14. Connect signal pin from PIR sensor to pin D9.
15. Simultaneously connect 5V & GND to Arduino.



16. Import a relay from Components menu.
17. Connect -ve to relay NO terminal.
18. Connect common terminal to GND of power supply.
19. Add a switch for manual control
20. Connect coil of relay to Arduino D10 & GND.
21. Connect coil to Arduino D10 & GND.
22. Import a photo diode
23. Connect terminal 1 of LDR to A0
24. Add a 3K resistor between terminal 1 & Vcc.
25. Import 2 bulbs and 2 relays from components menu.
26. Connect relay and both bulbs in series.
27. Import a gas sensor and a passive buzzer.
28. Connect R_1, H2 & R_2 to vcc of gas sensor.
29. Connect a 5K resistor between A1 and H1
30. Connect A2 to Arduino A1
31. Connect buzzer in GND and D8
32. Now run the code for the circuit to begin working

Code:

```
#include <Servo.h>
Int output1Value = 0;
int sen1value = 0;
int sen2value = 0;
int const gas_sensor = A1;
int const LDR = A0;
int limit = 400;
```

```
long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH);
}

Servo Servo_7;
void setup()
{
  Serial.begin(9600);
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  pinMode(13, OUTPUT);
  Servo_7.attach(7, 500, 2500);

  pinMode(8, OUTPUT);
  pinMode(9, INPUT);
  pinMode(10, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(3, OUTPUT);
}
```

KY039 | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Manage Libraries... Ctrl+Shift+I

Serial Monitor Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

WiFi101 / WiFiNINA Firmware Updater

Board: "Arduino Uno"

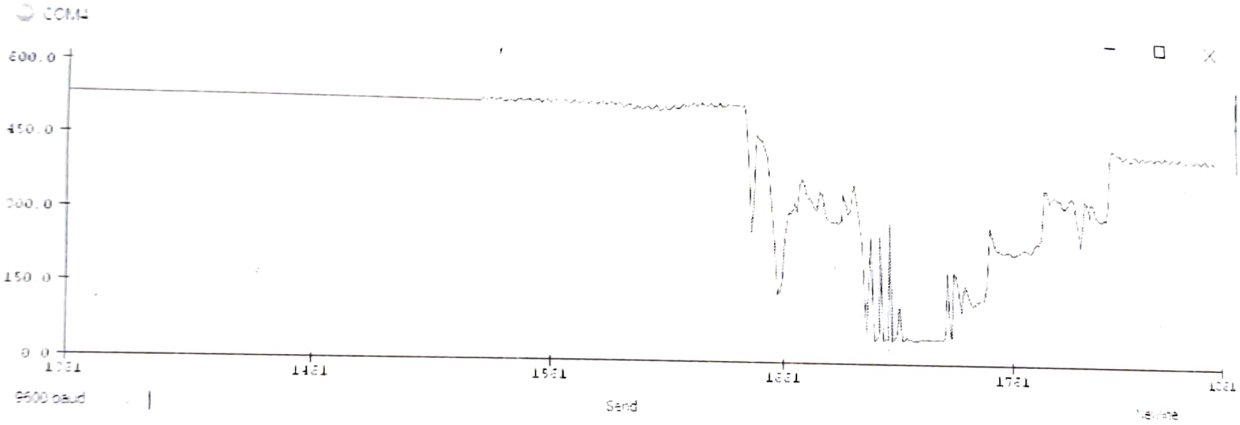
Port: "COM4"

Get Board Info

Programmer: "AVRISP mkII"

Burn Bootloader

```
pulse
eum
i
eum
pulse = eum
(100);
```



```
void loop() {
 int val1 = analogRead(LDR);
  if(val1 >500) {
      digitalWrite(13, LOW);
  }
  else {
      digitalWrite(13, HIGH);
      Serial.print("Bulb OFF = ");
      Serial.print(val1);
}
 Sen2Value = digitalRead(9);
 if(Sen2Value == 0) {
    digitalWrite(10, LOW);
    digitalWrite(4, HIGH);
    digitalWrite(3, LOW);
    Serial.print("\n NO Motion Detected");
}
 else {
   Servo_7.write(0);
   Serial.print("\n Door closed!");
   Serial.print(Sen1Value);
   Serial.print("\n");
}
 delay(10);
}
```

Result:

At the intensity of light decreases bulbs are turns ON and vice versa. If Someone entred in home, fan will automatically turned ON. Also, we can control it manually with switch. In home, if LPG gas is leaked the alarm circuit gets activated and buzzer makes sound. Ultrasonic sensor will set up on the top of the main door if any one comes near to door the door will automatically opens.

Experiment- 7

SHEET No. 30

Aim: Create any cloud platform account, explore IoT Services and register a thing on the platform.

Objectives: Student should get the knowledge of use of cloud for IoT.

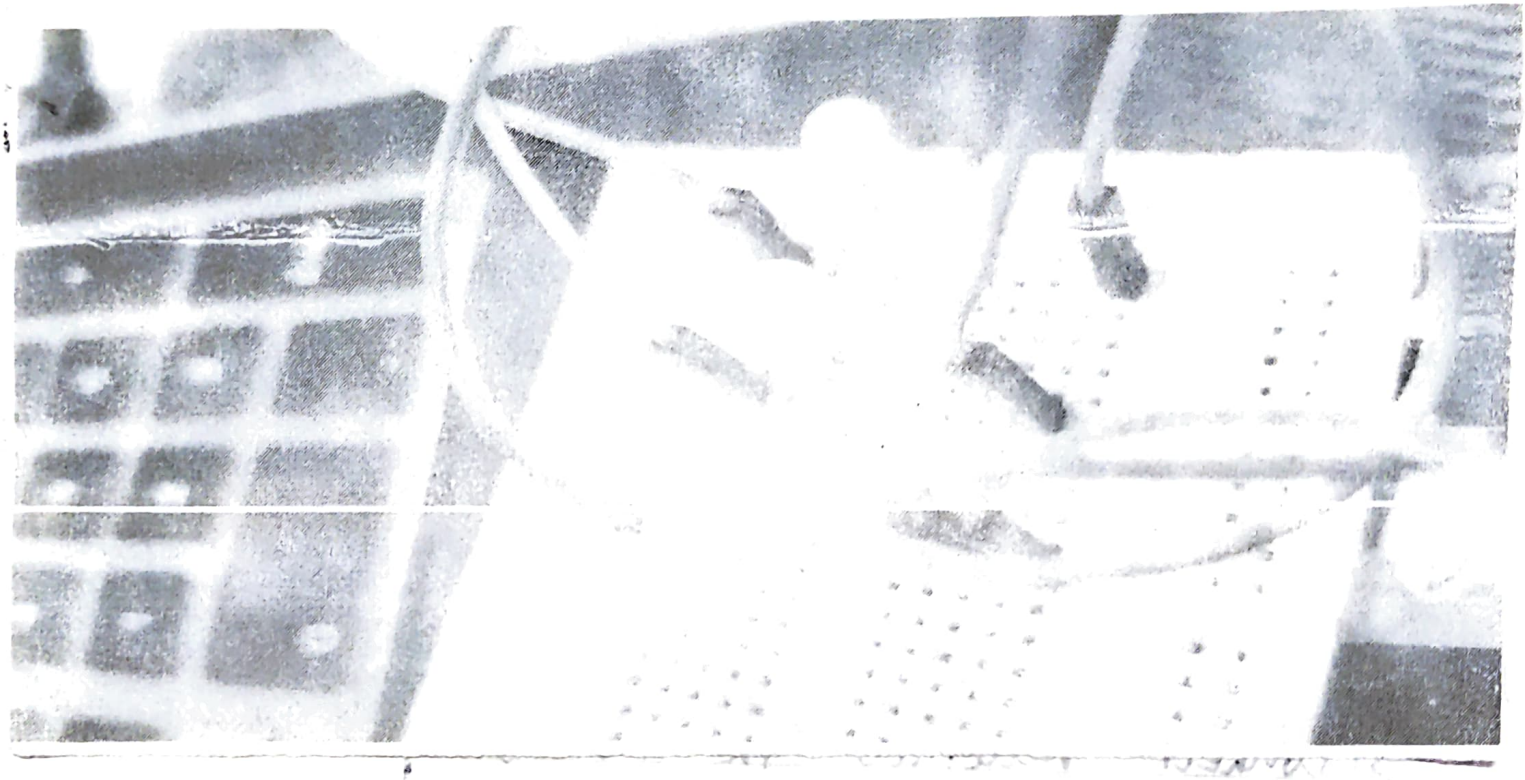
Outcomes: Student will be understood to create a cloud account for Arduino board and register it on the cloud.

Hardware Requirements:
- NodeMCU
- USB Cable

Procedure:
1) Create a new account in www.thinger.io
2) Login with your account
3) Connect nodeMCU in the USB port.
4) Click on devices and click on add device

Give the following information:
Device Type: Generic Thinger Device
Device ID: MynodeMCU
Device Description: LED Control
Click on Generate random credentials.

Click add device



Now your device is registered

Steps to install Arduino IDE:

1) Go to https://www.arduino.cc/en/software.

After Arduino IDE is successfully installed on your PC Open Arduino IDE

2) Install ESP8266 Add-on in Arduino IDE

-) In your Arduino IDE, go to File > Preferences

-) Open Boards Manager. Go to Tools > Board > Boards Manage

-) Search for ESP8266 and press install button for the "ESP8266" by EsP8266 Community"

-) That's it.

Experiment-8

SHEET No. 33

Aim: Push sensor data to cloud.

Objectives: Student should get the knowledge of use of cloud for IoT

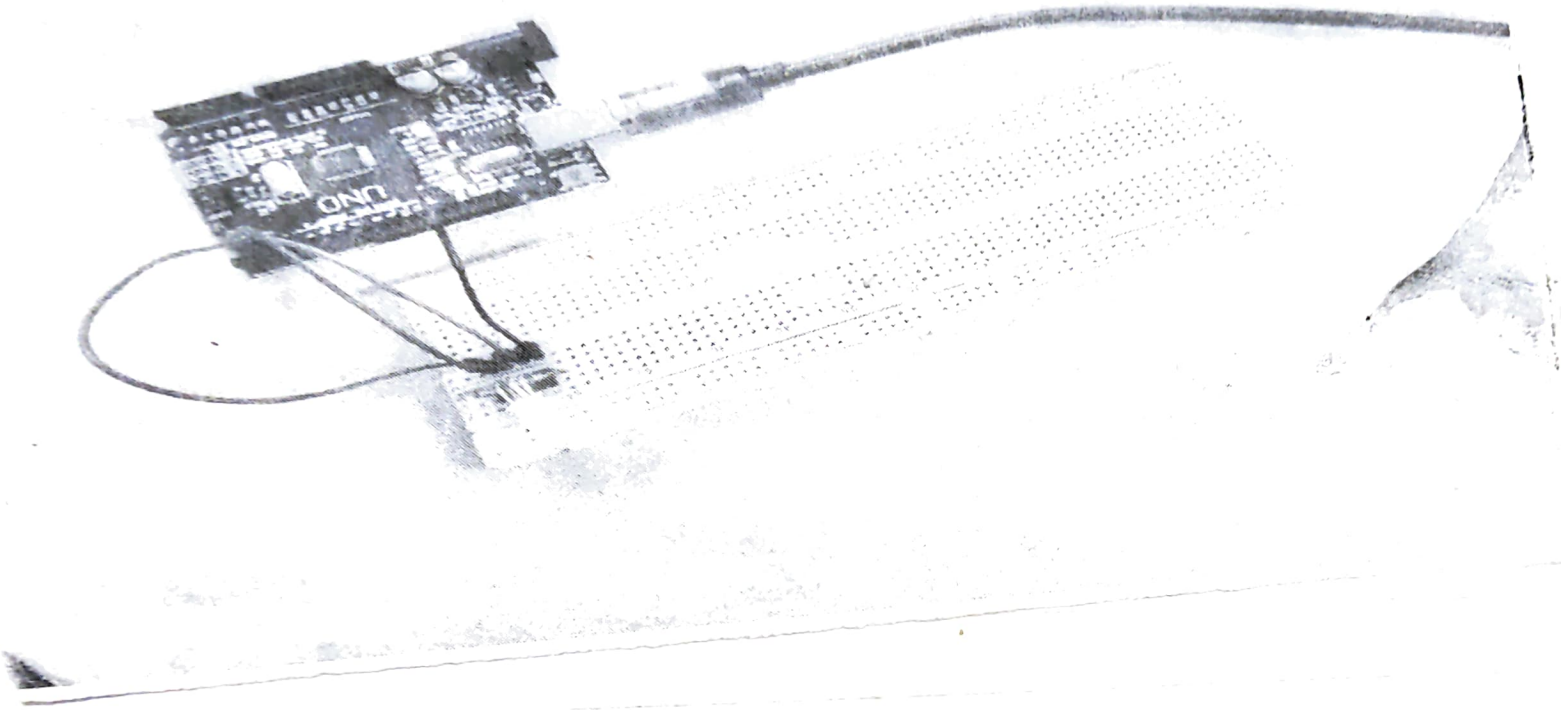
Outcomes: Student will be understood to how to acquire data from sensor and transferring it to cloud.

Hardware Requirements:
- NodeMCU
- USB Cable
- Breadboard
- Lm35 sensor
- Connecting wire (Orange, Yellow, Green)

Procedure:

Connection for Hardware:
- Insert lm35 onto the breadboard.
- Connect yellow wire to the right end pin of the sensor, green wire to middle pin and orange wire to the left end pin of lm35 sensor.
- Now take NodeMCU, insert
  - Yellow -> GND
  - Green -> A0 pin
  - Orange -> 3V3 pin
- After Connecting to hardware, proceed to cloud.

* Create a new account in www.thingser.io


UNO

Aim: Push sensor data to cloud.

Objectives: Student should get the knowledge of use of cloud for IoT

Outcomes: Student will be understood to how to acquire data from sensor and transferring it to cloud.

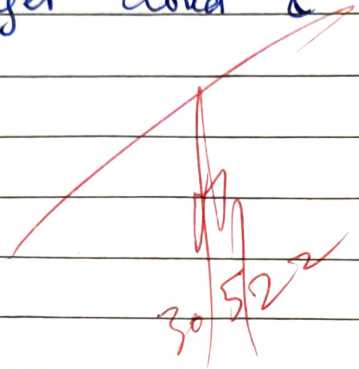
Hardware Requirements:
- NodeMCU
- USB Cable
- Breadboard
- Lm35 sensor
- Connecting wire (Orange, Yellow, Green)

Procedure:
Connection for Hardware:
- Insert lm35 onto the breadboard.
- Connect yellow wire to the right end pin of the sensor, green wire to middle pin and orange wire to left end pin of lm35 sensor.
- Now take NodeMCU, insert
  - Yellow -> GND
  - Green -> A0 pin
  - Orange -> 3V3 pin
- After connecting to hardware, proceed to cloud.

* Create a new account in www.thingser.io


11:28
ESP8266 LED
ON
ON

* add device
* Program the below Sketch to nodemcu
* Add widget to dashboard
* click on dashboard and new widget
* Save and select gauge menu
* Change gauge settings
* Now you can observe the sensed value on guage & chart widget cloud & calculate average.



Experiment - 9

SHEET No. 35

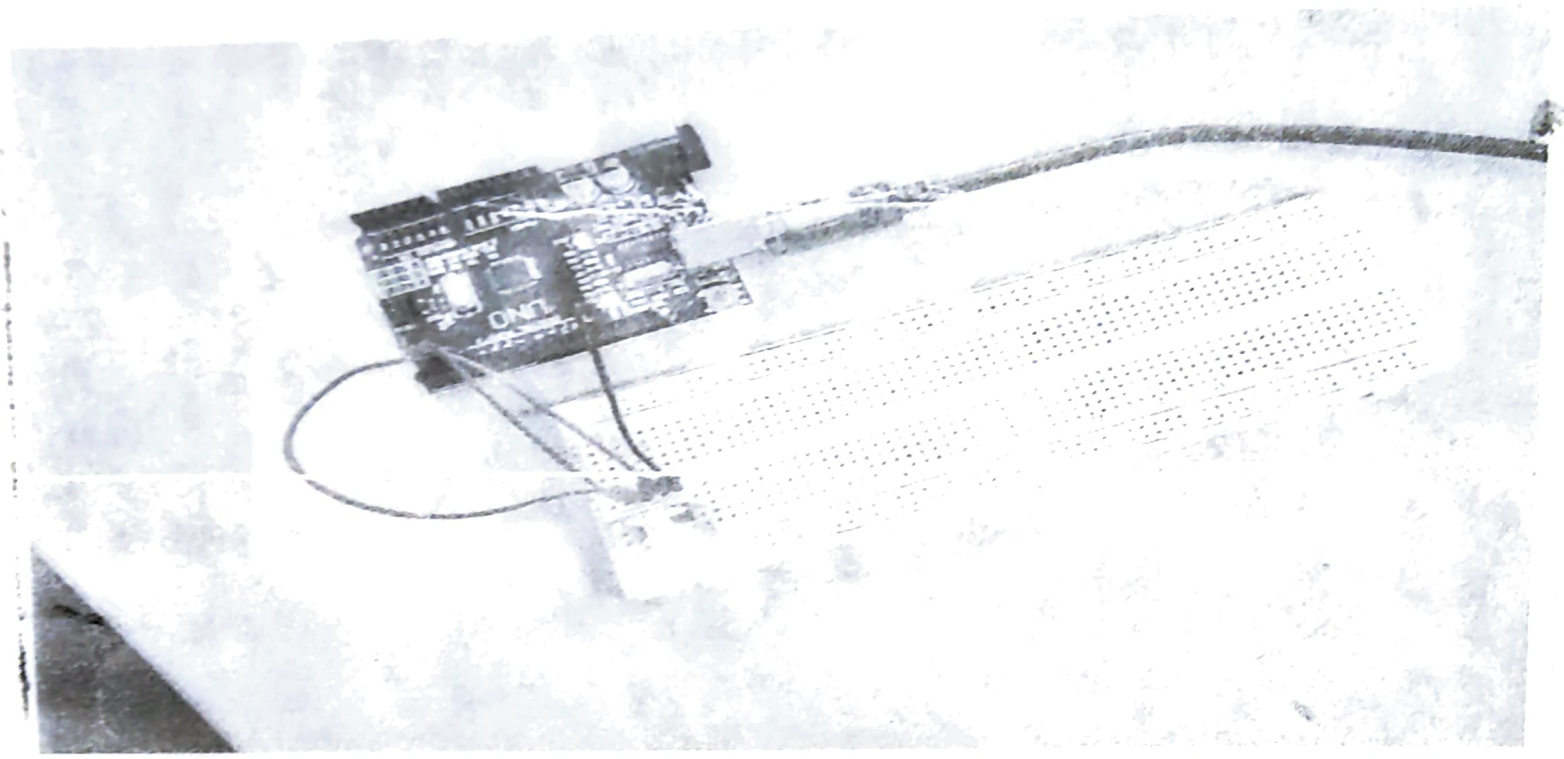
Aim: To build an ESP8266 Web Server Circuit using NodeMCU

Apparatus:
- NodeMCU (ESP8266)
- USB Type-A
- LEDs - 2x
- Breadboard
- Wiel - 1x
- Pc with good internet connection.

Procedure:
1) Download and install Arduino IDE
2) Install the ESP8266 add-on for the IDE.
3) click "OK" button
4) Tools > Board > Boards Manager
5) Select ESP8266 menu and install "ESP2866 by ESP2866 "
6) Took > Board & choose ESP8266 board.
7) Download & copy the following code on to the IDE.
8) Change the code inside the void loop
9) Connect the connections on the Breadboard.
10) Progress of uploading code into NodeMCU
11) Open Serial Monitor & wait for 30s
12) Copy IP address & paste in a device. Control LEDs

Code:

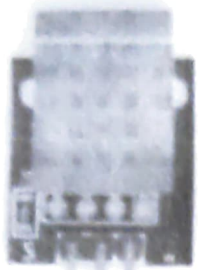
```
#include <ESP2866WiFi.h>
```



```
const char* ssid = "WiFi name";
const char* pass = "Password";

WiFiServer(80);        // web server port: 80

String header;
String output5State = "off";
String output4State = "On";

const int output5 = 5;
const int output4 = 4;
unsigned long currentTime = millis();
unsigned long previousTime = 0;
const long timeoutTime = 2000;

void setup() {
  Serial.begin(9600);
  pinMode(output5, OUTPUT);
  pinMode(output4, OUTPUT);

  digitalWrite(output5, LOW);
  digitalWrite(output4, LOW);

  Serial.print("connecting to");
  Serial.println(&ssid);
  WiFi.begin(ssid, password);
  delay(500); }
```

```
    Serial.println("WiFi Connected");
    Serial.println("IP address");
    Serial.println(WiFi.localIP());
    Server.begin();
}

void loop() {
    WiFiClient client = Server.available();
    if (client) {
        Serial.println("New Client");
        String currentLine = "";
        currTime = millis();
        prevTime = currentTime();

    while(client.connected() && currentTime - prevTime <= timeoutTime) {
            currTime = millis();
            if(currentTime) {
                char c = client.read();
                header += c;
                if (c == '\n') {
                    if(currentTime == 0) {
                        client.println("HTTP/1.1 200 OK");
                        client.println("connection: close");
                    }
                }
            }
        }
    if (header.indexOf("GET /5/low") >= 0) {
        Serial.println("GPIO 5 on");
```

3 PIN



Vcc (3.3V) Out GND

```
int pinDHT11 = 2;

SimpleDHT11 dht11;

void setup() {#include <SimpleDHT.h>

// for DHT11,

//      VCC: 5V or 3V

//      GND: GND

//      DATA: 2

// start working...

Serial.println("Temperature and Humidity Data");

Serial.begin(9600);}
```

```
output5State = "on";
digitalWrite(output5, HIGH);
} else if (header.indexOf("GET /5/off") >= 0) {
        Serial.println("GPIO 5 off");
        output5State = "off";
        digitalWrite(output5, LOW);
} else {
        currentLine = "";
}
header = "";
client.stop();
Serial.println("Client disconnected.");
}
}
```

Theory:

NodeMCU has Station mode using which it can connect to the existing Wi-Fi network & can act as an HTTP server with an IP address allocated by that network. NodeMCU gets IP from the Wi-Fi router to which it is connected.

Experiment - 10

Aim: Temperature/Humidity sensor using DHT11 with Arduino.

Materials:
- Arduino Uno
- KY-015 Arduino
- Jumper Wires
- Breadboard
- Cable (USB 2.0)

Theory:
In this project, we will use KY-015 Arduino DHT 11 temperature sensor and display it to the Arduino IDE Serial Monitor. This project is really simple and shouldn't takes us very long. The DHT 11 uses just one signal wire to transmit data to the Arduino. The DHT11 is a basic, ultra low-cost

Code:
```
#include <SimpleDHT.h>
int pinDHT11 = 2;
SimpleDHT11 dht11;

Void setup() {
  Serial.println("Temperature & Humidity Data");
  Serial.begin(9600);
}
void loop() {
   byte temperature = 0;
```

6.click on compile, after completion click on Sketch.

```
byte humidity = 0;
int err = SimpleDHTErrSuccess;

if (err = dht11.read(pinDHT11, &temperature, &humidity, NULL)
  != SimpleDHTErrSuccess) {
    Serial.print("Read DHT11 failed, err=");
    Serial.println(err); delay(1000);
    return;
}
Serial.print((int)temperature);
Serial.print((int)humidity);
delay(500);
}
```

* DHT11 is a humidity and temperature sensor. It can be used as humanity sensor as well as the temperature sensor.
* In 3pin dht11 sensor already 10K ohm resistor is added inside the module. The operating voltage of this module is 3.3V
*

Result: We will use KY-015 Arduino DHT11 Temperature/Humidity sensor and display it to the Arduino IDE serial monitor.

30/5/22

Experiment-11

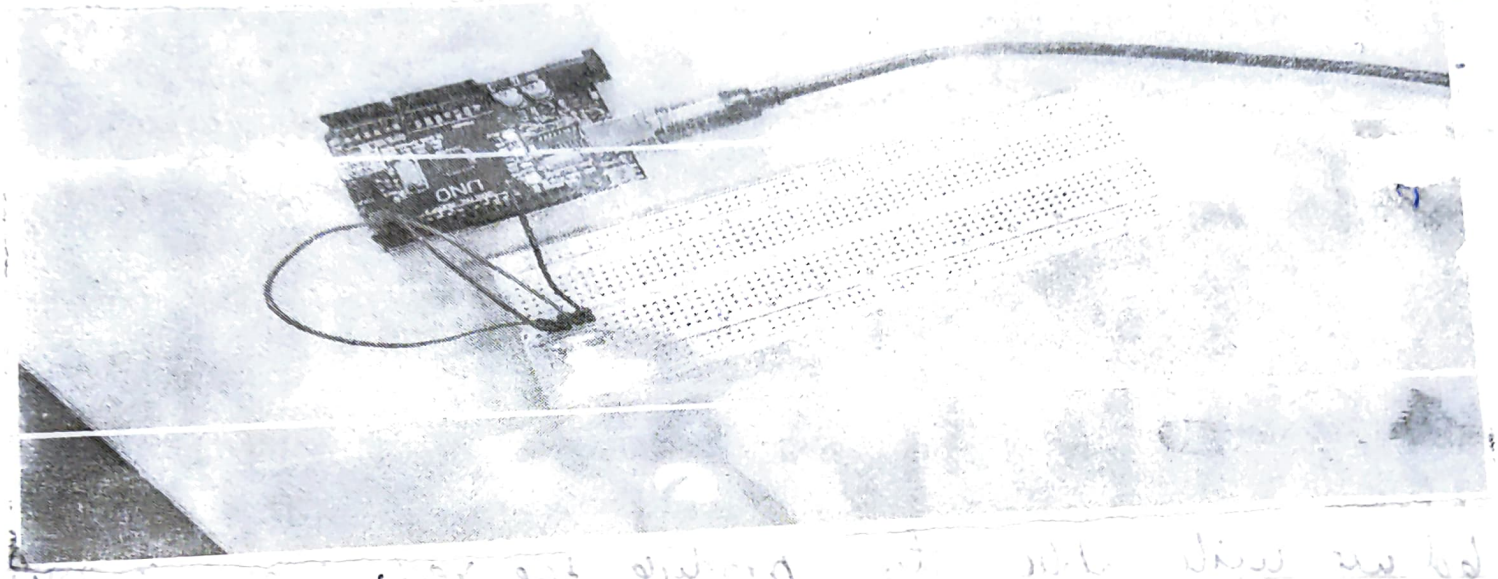
SHEET No. 41

Aim: To interface RGB led module with Arduino uno

Hardware Requirements:
- Breadboard
- Arduino Uno
- Jumper wires
- RGB led module

Theory:
RGB led is a combination of blue led, green led and red led. It is also called tri-color or multi-color led. With this led we will able to produce the red, green & blue. We can also produce different colors by configuring the intensity of each led.

Procedure:
- Take a breadboard and a RGB LED module connect the module to the breadboard.
- Connections:
  * Connect Green pin to D4
  * Connect Blue pin to D3
  * Connect Red pin to D2
  * Connect cathode to the GND of Arduino.
  * Upload the given C code to the Arduino and you can observe the rgb module changes light color for every 1 Second.

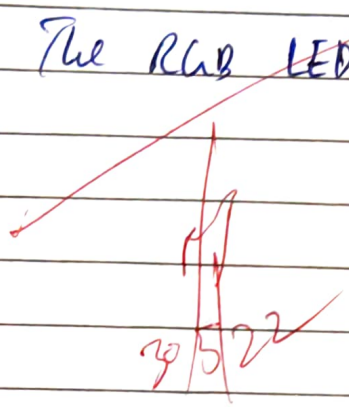

UNO

Code:

```
const int red = 2;
const int green = 3;
const int blue = 4;

void setup() {
  Serial.begin(9600);
  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(blue, OUTPUT);
}

void loop() {
  Serial.println("Now red is on");
  digitalWrite(red, HIGH);
  digitalWrite(green, LOW);
  digitalWrite(blue, LOW);
  delay(100);
  Serial.println("Green is on");
  digitalWrite(green, HIGH);
  digitalWrite(red, LOW);
  digitalWrite(blue, LOW);
  delay(100);
  Serial.println("Now blue is on");
  digitalWrite(blue, HIGH);
  digitalWrite(red, LOW);
  digitalWrite(green, LOW);
```

```
    delay(100);
}
```

Result: The RGB LED module changes light colors every second.



Experiment - 12

Aim: Control LED's using ThingSpeak and NodeMCU with Arduino

Software Requirements:
- Create a new account in ThingSpeak
- Install Arduino IDE in your PC & Laptop
- Proper Internet Connection

Hardware Requirements:
- NodeMCU
- Breadboard
- LED's
- Jumping wires
- USB Cable

Theory:
Thingspeak is IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data.

Hardware Connections:
- Connect NodeMCU to your Laptop Computer using USB
- Using jumping wires connect D1, D2, D3 pins of NodeMCU to the LED1, LED2, LED3 on breadboard respectively.
- Using jumping wires give a common ground connection for the three LEDs placed on Breadboard.

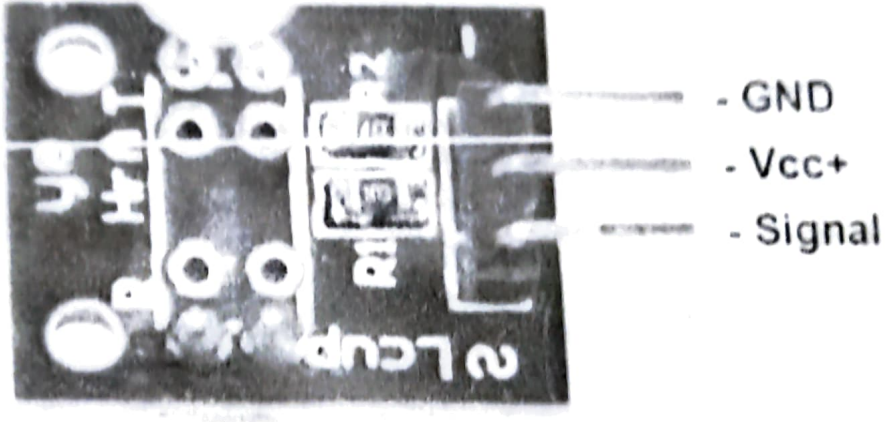
Procedure:

Steps: Setting ThingSpeak

- First login to the Thingspeak Server.
- If you are now new user then create the new account
- After login show this type of webpage but if you are new.
- After clicking the new channel, first write the name of the name channel.
- If you want to give the description then write the description and select the field 1 and don't forget to click on checkbox, after clicking on the checkbox scroll the page and click on Save button.
- Give the channel name as "LED control" and description as 'Controlling LED using NodeMcu and ThingSpeak'
  * Field 1 - led 1
  * Field 2 - led 2
  * Field 3 - led 3
- Click on Save channel
- Now show your led 1, 2, 3 as charts
- Now make the channel as public views so that anyone can views your channel and keep your channel ID safe.
- Now we are done with the ThingSpeak part.
- Open IDE & install ThingSpeak and ESP8266WiFi Libraries

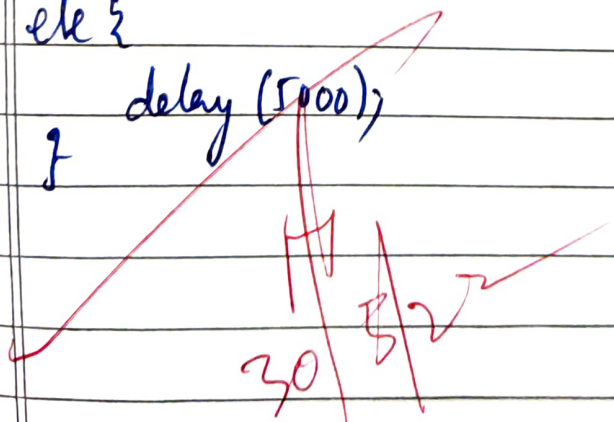
Code:

```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
```


- GND
- Vcc+
- Signal

```
const char* ssid = "ssid";
const char* pass = "password";
unsigned long channel = 1381140;
unsigned int led1 = 1;
unsigned int led2 = 2;
unsigned int led3 = 3;

WiFiClient client;

void setup() {
  Serial.begin(115200);
  delay(100);

  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  digitalWrite(D1, 0);
  digitalWrite(D2, 0);
  digitalWrite(D3, 0);

  Serial.print("Connecting");
  Serial.println(ssid);

  WiFi.begin(ssid, pass);

  while(WiFi.status() != WL_CONNECTED) {
      delay(500); }
```

```
    Serial.println("WiFi Connected");
    Serial.println(WiFi.localIP());
    Serial.println(WiFi.gatewayIP());
    ThingSpeak.begin(client);
}

void Loop() {
    int led_1 = ThingSpeak.readFloatField(channel, led1);
    int led_2 = ThingSpeak.readFloatField(channel, led2);
    int led_3 = ThingSpeak.readFloatField(channel, led3);

    if(led_1 == 1){
        digitalWrite(D1, 1);
        Serial.println("D1 is On!");
    }
else if(led_2 == 1){
        digitalWrite(D2, 1);
}
else if(led_3 == 1){
        digitalWrite(D3, 1);
    }
else {
        delay(5000);
}
```



Observation:

- Using Thingspeak platform the data is monitored in NodeMCU according to data the LED's will be controlled by changing the field values in the URL and the LED's will blink according to field values.

Result:

We will control LEDs from anywhere using the API key provided to upload the data in ThingSpeak platform.

Experiment-12

Aim: To download and install Raspberry OS and connect to it.

Procedure:

- Insert SD card to PC using SD card reader
- Format SD card.
- Visit www.raspberrypi.com/software and download "Raspberry Pi Imager".
- Install the file
- Click on "choose OS" and choose "Raspberry Pi OS (32 bit)"
- click on "choose Storage" and choose "SD card".
- Click on advance settings
- Configulation all the setings
- click on Save
- click on write
- Warning as shown fig if pop-up just click Yes.
- It will write OS and verify
- After successful completion click on continue.
- After this, you just need to plug you sd card.
- Turn on hotspot and connect.
- Now power on Rpberry Pi
- Download putty.
- After downloading putty, Install it.
- click Accept.
- Login using.
  * username: pi
  * passwd: raspberry

Install Raspberry Pi OS using Raspberry Pi Imager

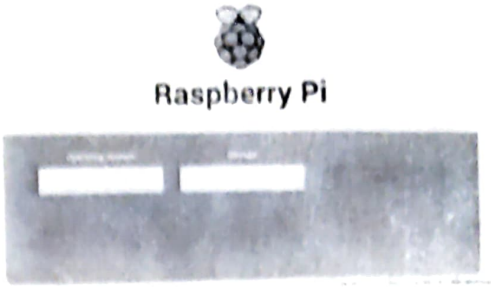
Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Watch our video to learn how to install an operating system using Raspberry Pi Imager.

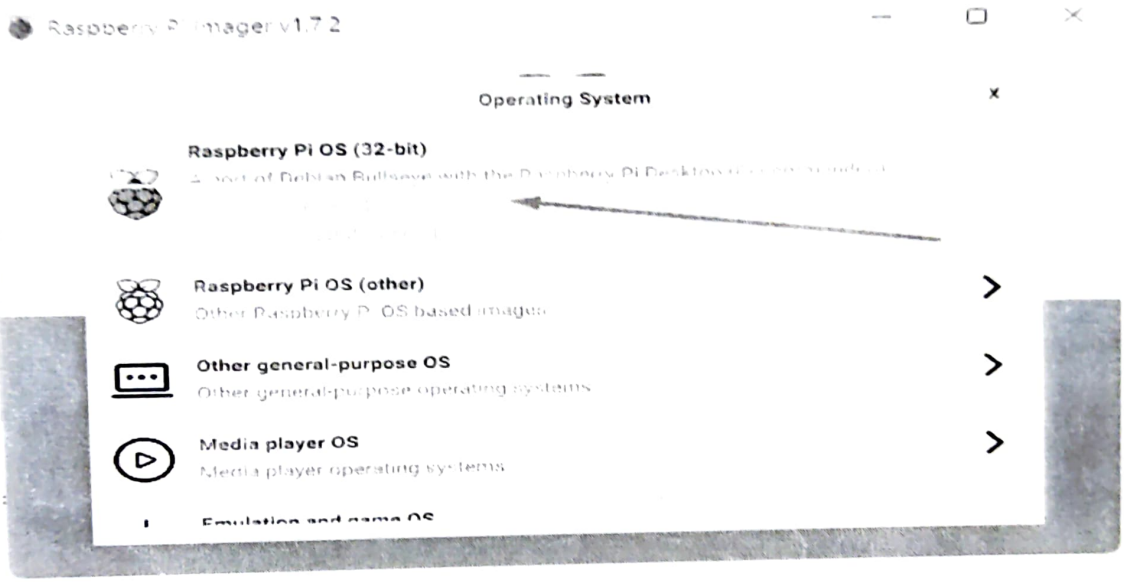
Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the computer's SD card reader, and open Raspberry Pi Imager.

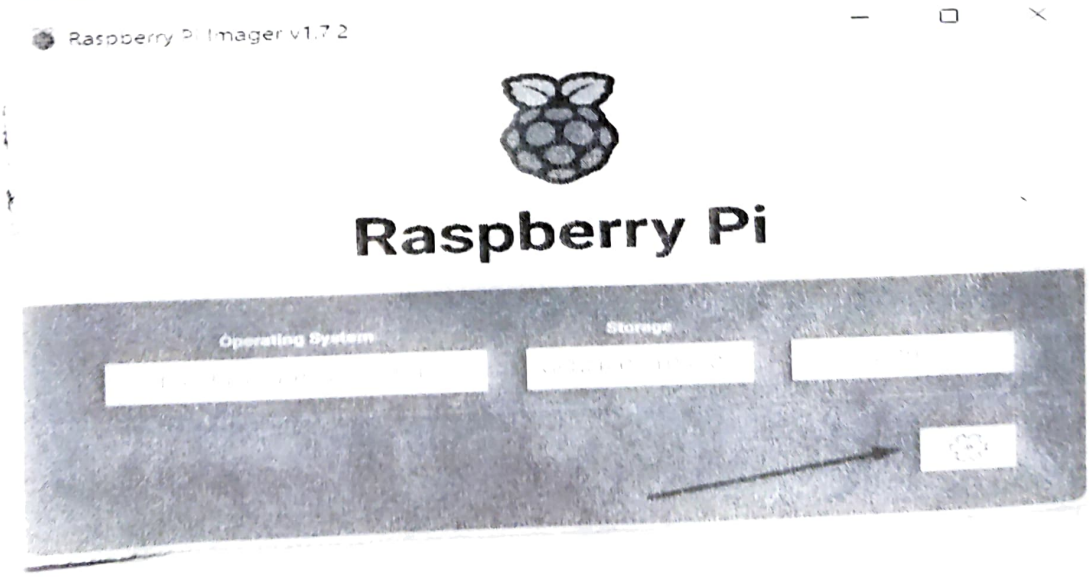
Download for macOS

Download for Ubuntu for x86

Install Raspberry Pi OS (Legacy)

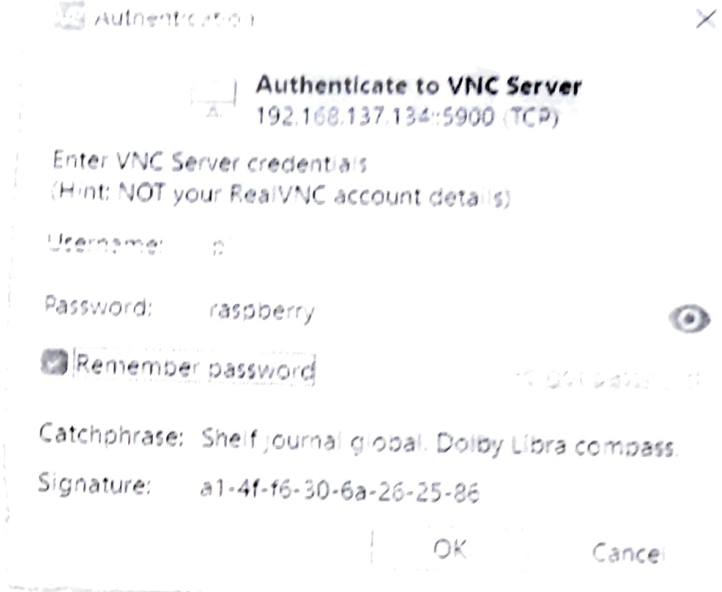






- Once connected
- Type command to go into the config menu
sudo raspi-config
- options > VNC > Enable it.
- click enter on "VNC" & select yes to enable. you'll get like this after VNC enabled.
- Now goto display options & select VNC resolution
- Select 1980 x 1080
- click on finish
- It will ask to reboot click yes
- It will reboot.
- If you are an IOS user you can download VNC viewer
- Open VNC viewer, put IP of raspberry Pi in address bar.
- later put username and password & click on.
- Now you can able to use raspberry using your laptop.

M 1 / 5 / 22
30


Authentication
Authenticate to VNC Server
192.168.137.134::5900 (TCP)
Enter VNC Server credentials
(Hint: NOT your RealVNC account details)
Username: p
Password: raspberry
Remember password
Catchphrase: Shelf journal global Dolby Libra compass
Signature: a1-4f-f6-30-6a-26-25-86
OK
Cancel