**Experiment no 1 : Transmit string using UART.**

**Aim:** A UART’s (Universal Asynchronous Receiver/Transmitter) main purpose is to transmit and receive serial String data using serial communication.

**Objective:** To learn how serial communication is achieved using UART. (serial string data)

**Components Required:**

1. Arduino UNO.
2. Arduino UNO USB cable.

**Connections:** Arduino UNO to be connect with computer through USB cable.

**Procedure :**

1. Open the Arduino UNO IDE and create and save a sketch source code as follows
2. Compile sketch.
3. Upload sketch on to Arduino UNO.

**Source Code:**

String name;

int age;

float height;

void setup() {

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

Serial.begin(9600);

}

void loop()

{

// put your main code here, to run repeatedly:

Serial.println(“\n Enter your name:”);

while(Serial.available()==0)

{

}

name=Serial.readString();

Serial.println(“Sending name……”);

delay(10000);

Serial.println(“enter your age:”);

while(Serial.available()==0)

{

}

age=Serial.parseInt();

Serial.println(“Sending age…..”);

delay(10000);

Serial.println(“Enter your Height Feet.Inch :”);

while(Serial.available()==0)

{

}

Height=Serial.parseFloat();

Serial.println(“Sending height….”);

delay(5000);

display();

}

void display()

{

Serial.print(“Hello”);

Serial.print(name);

Serial.println(“your are”);

Serial.print(age);

Serial.print(“years old and”);

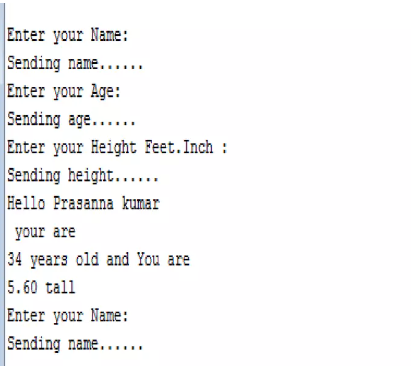
Serial.println(“You are”);

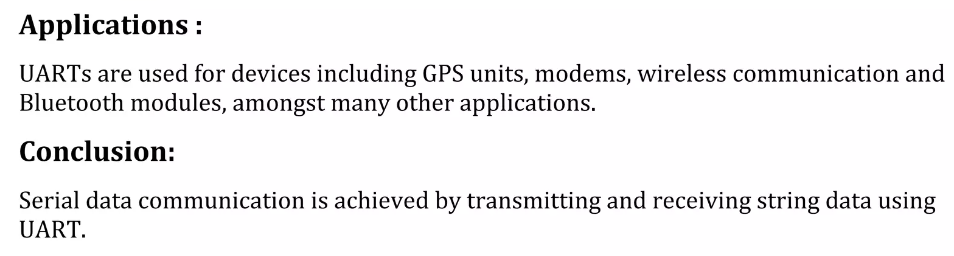
Serial.print(“height”);

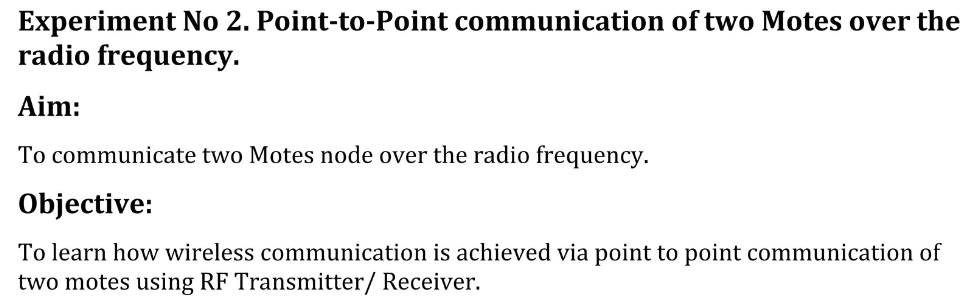
Serial.print(“tall”);

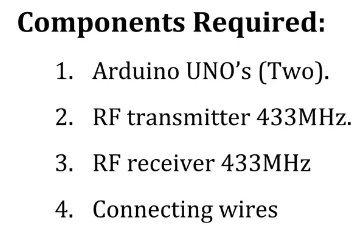
}

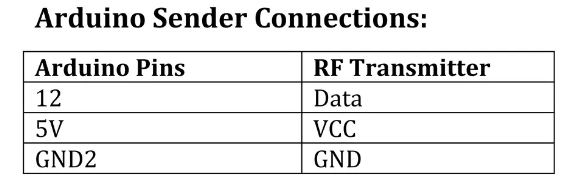
Output:

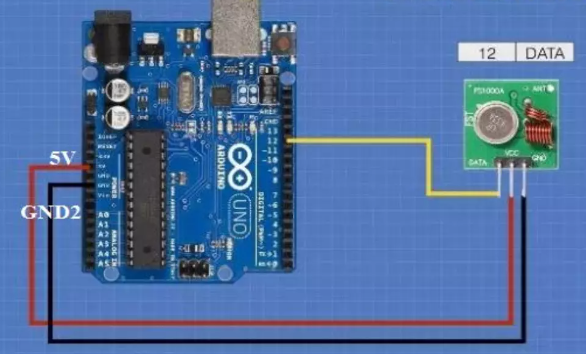


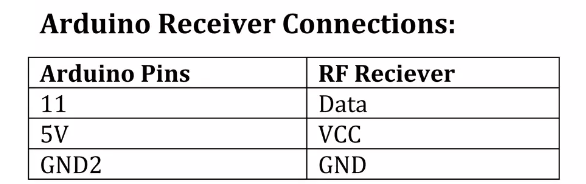


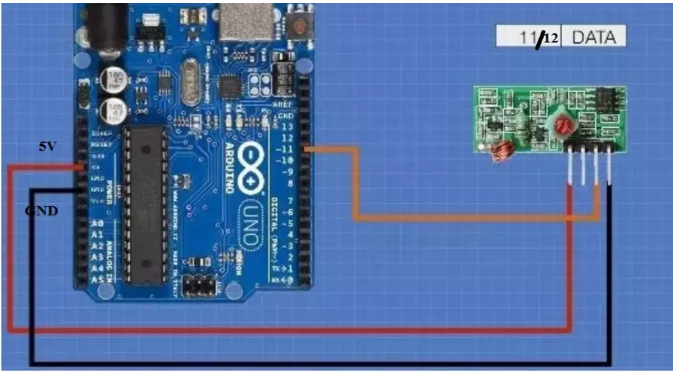


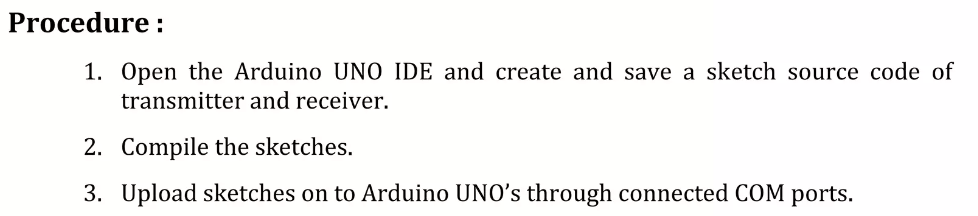




 TRANSMITTER



 RECEIVER



**Source code of Receiver:**

#include<RH\_ASK.h>

#include<SPI.h> // Not actualy used but needed to compile

RH\_ASK driver;

void setup()

{

Serial.begin(9600); // Debugging only

If(!driver.init())

Serial.println(“init failed”);

}

void loop()

{

uint8\_t buf[12];

uint8\_t buflen=sizeof(buf);

if (driver.recv(buf, &buflen))// Non- blocking

{

int i;

// message with a good checksum received, dump it.

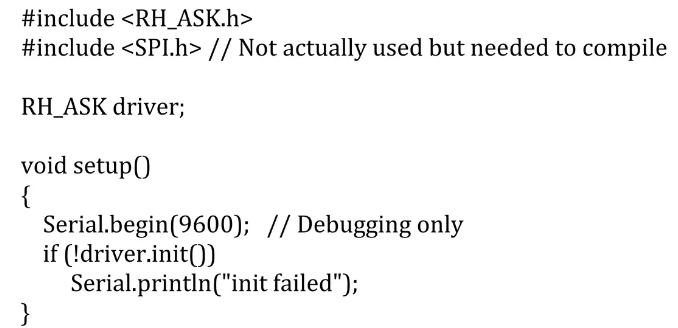
Serial.print(“Message:”);

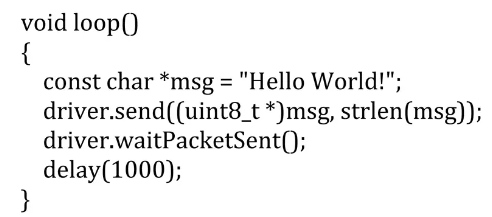
Serial.println(char\*) buf);

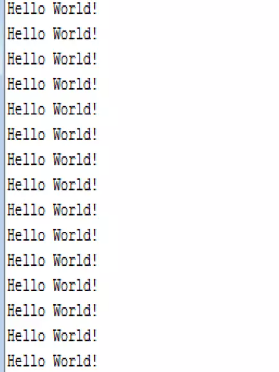
}

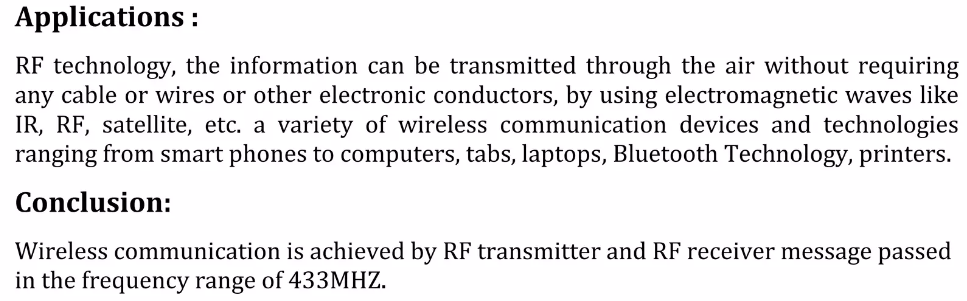
}

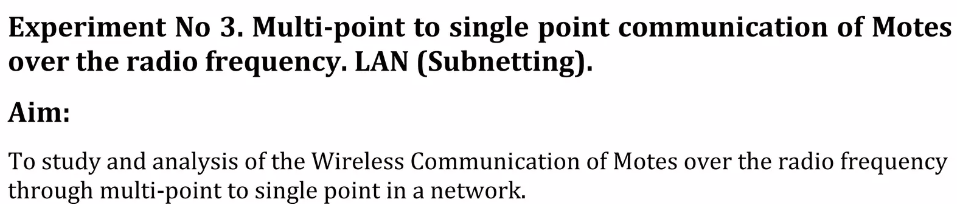
**Source Code of Transmitter:**

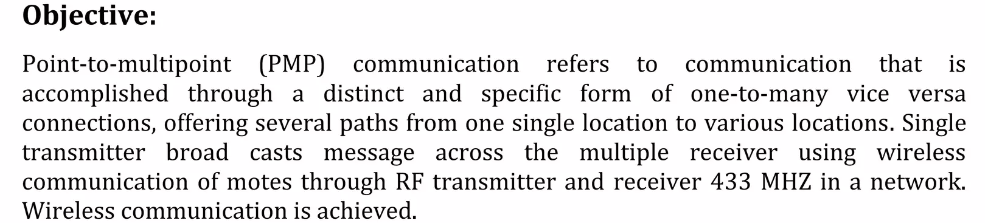


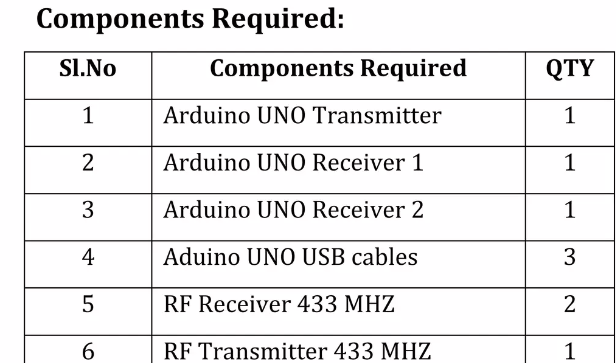




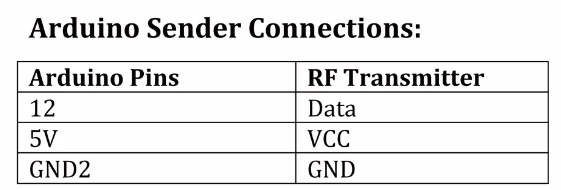


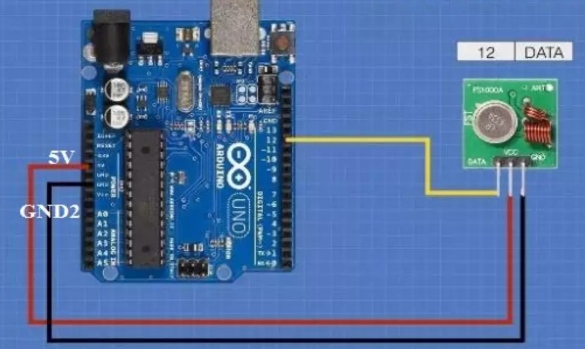


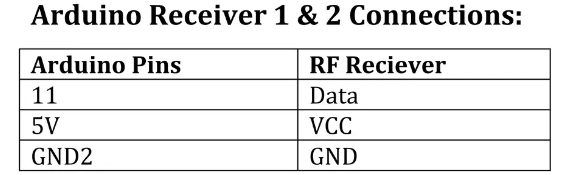


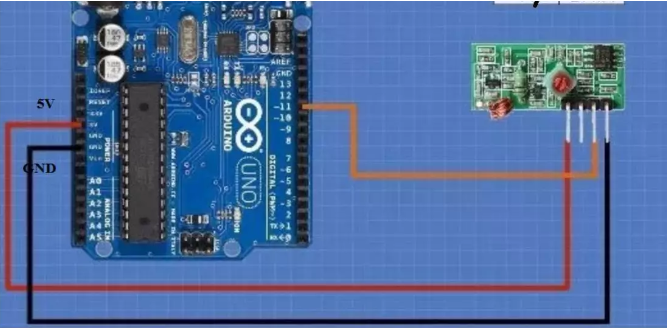


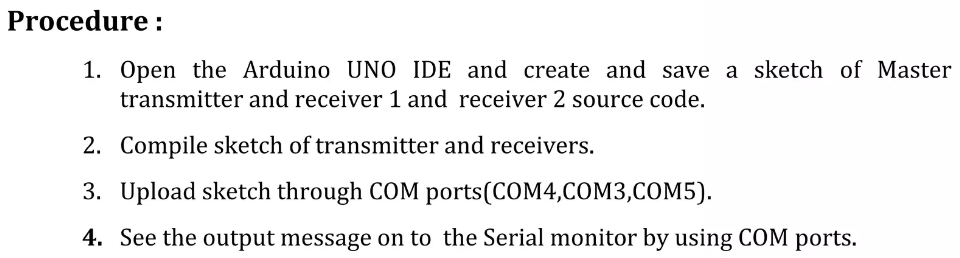


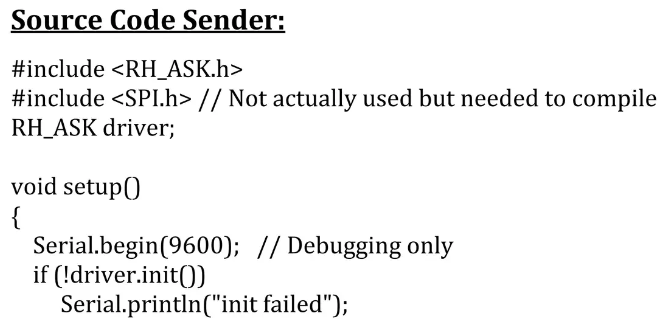


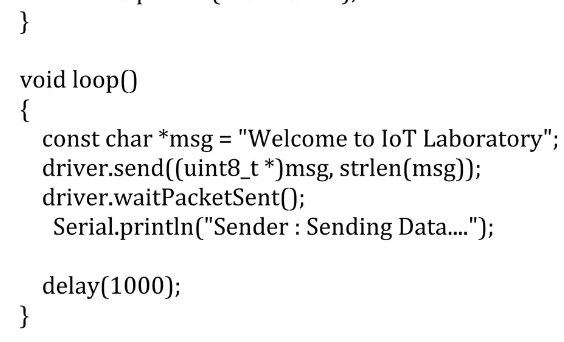
 TRANSMITTER



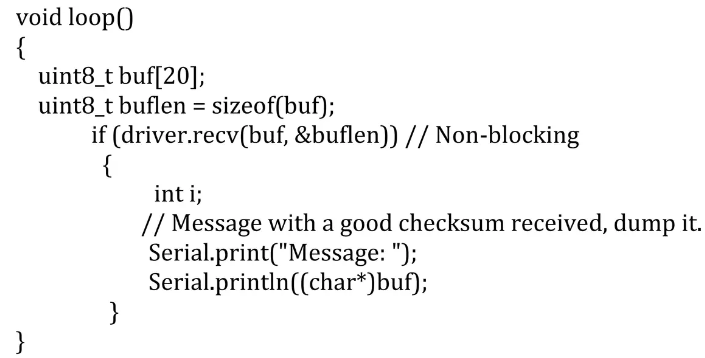
 RECEIVER

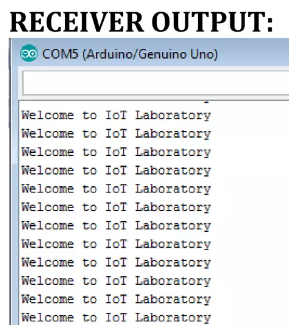




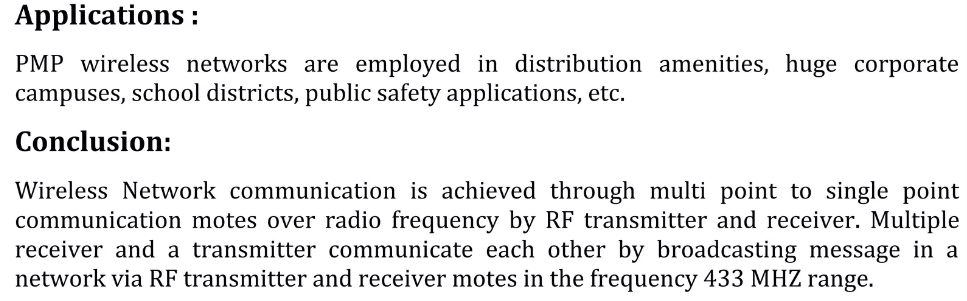


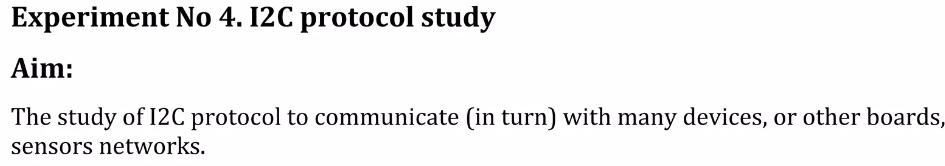


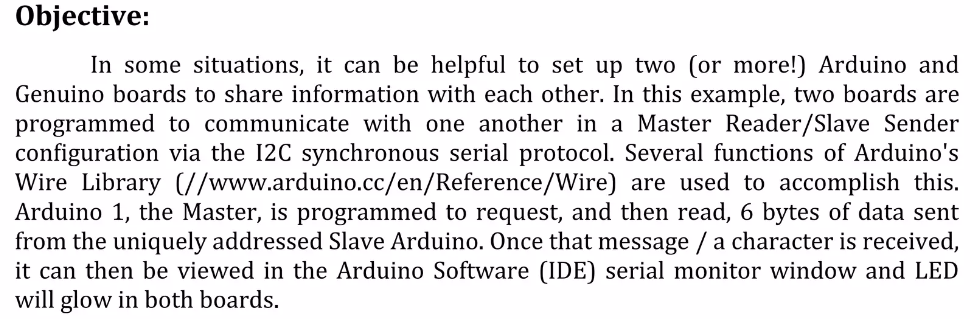


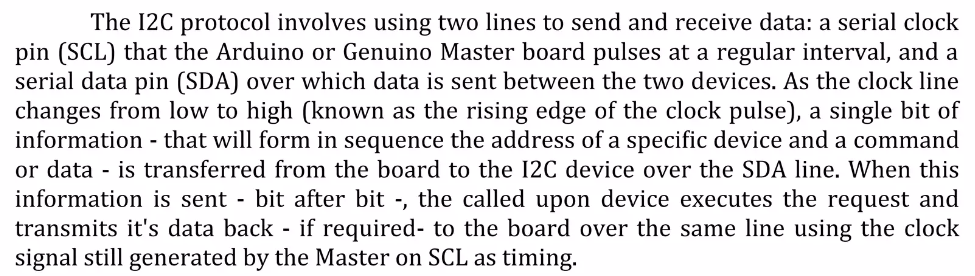


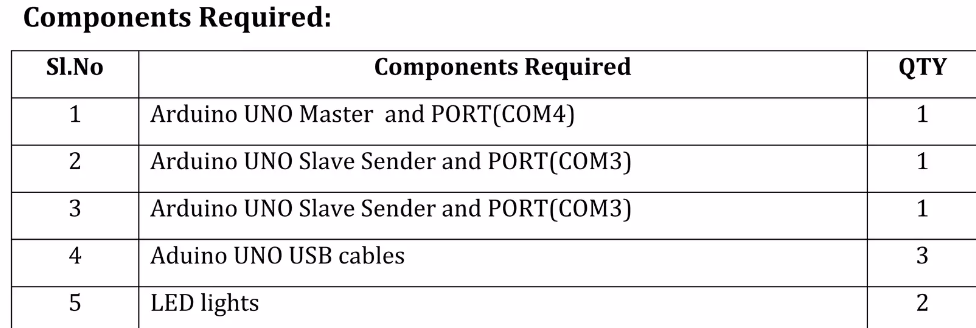


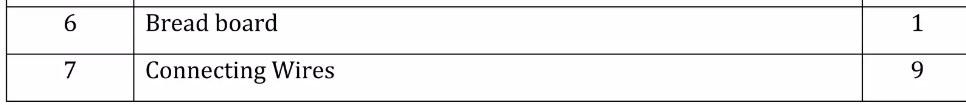


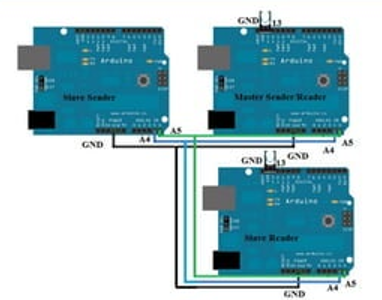


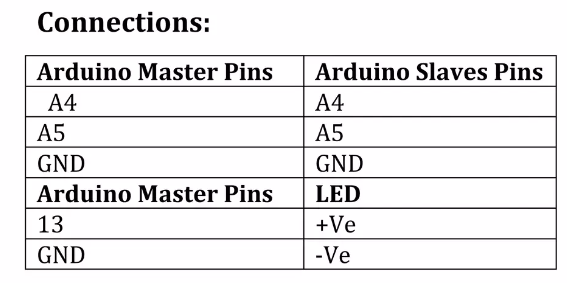


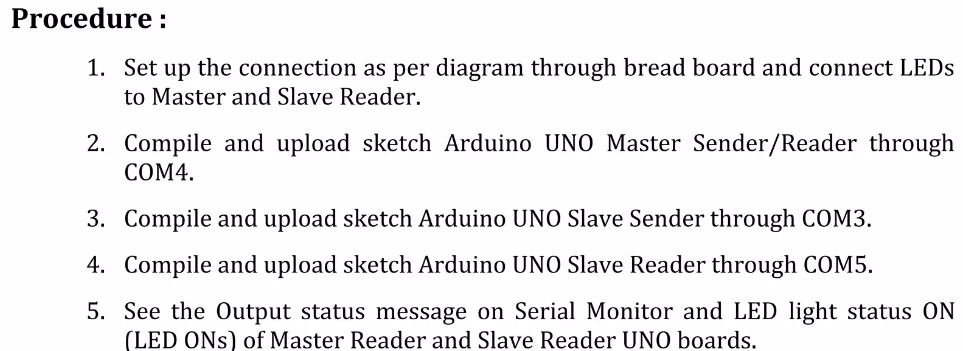


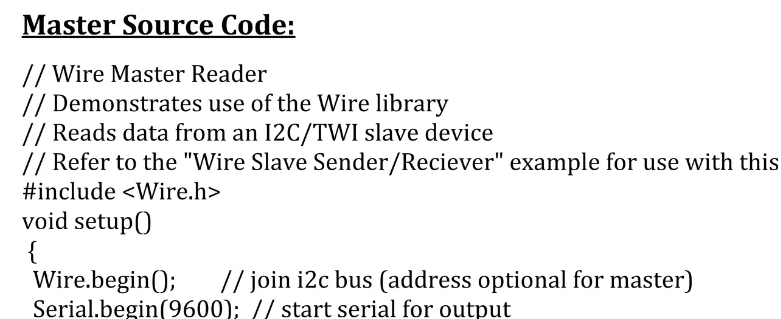


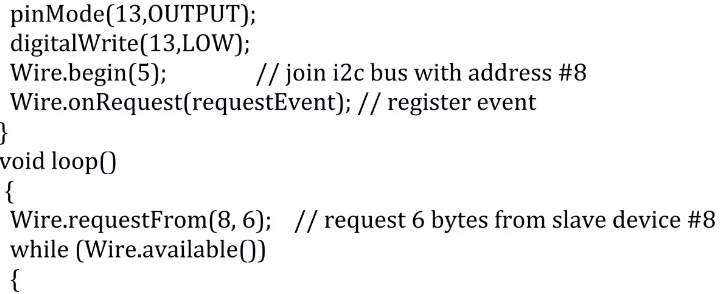


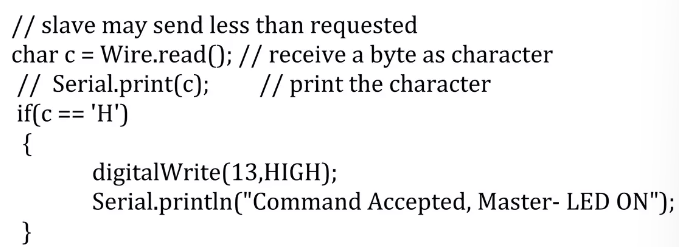


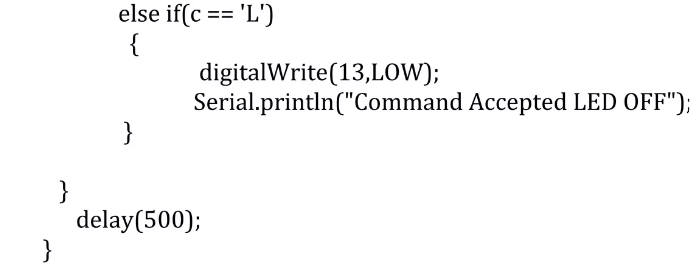


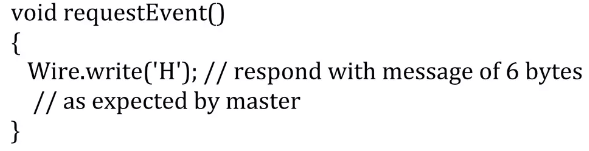


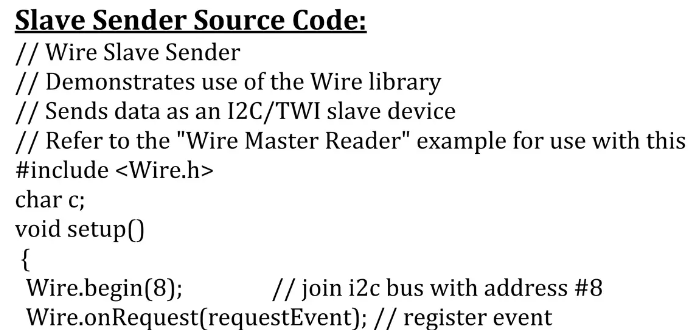


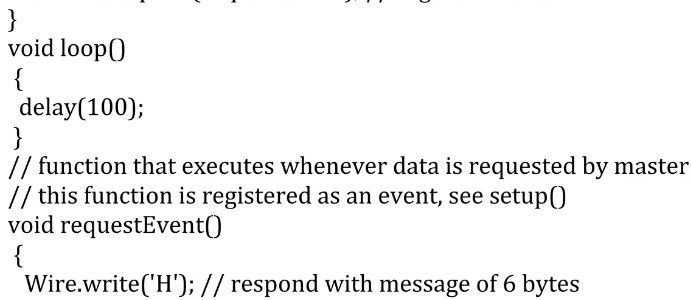




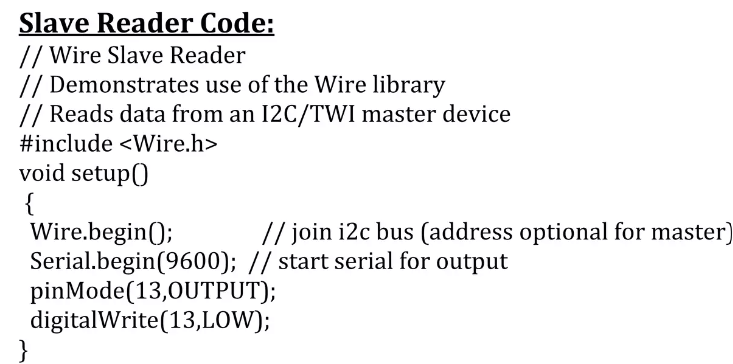




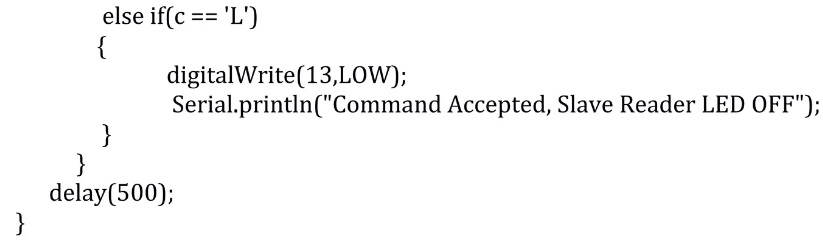


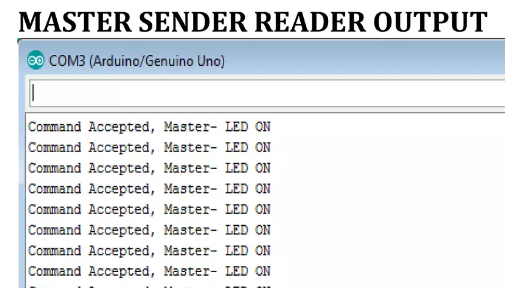


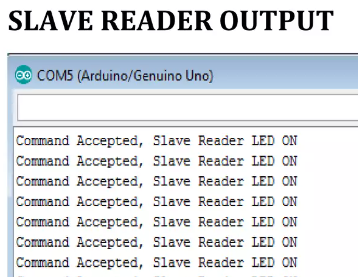


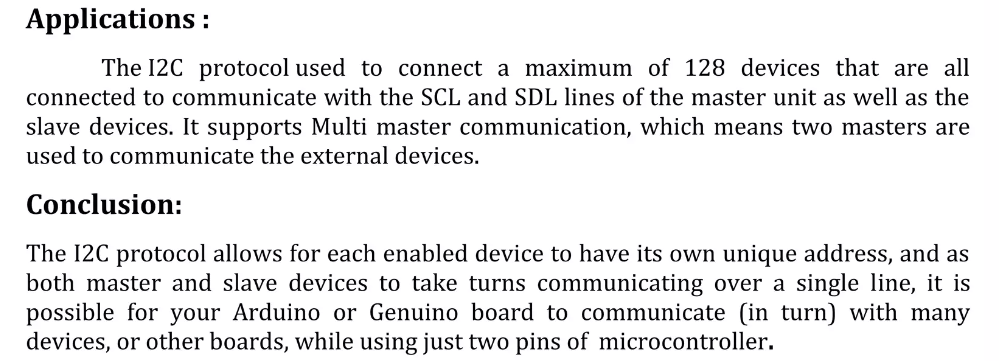


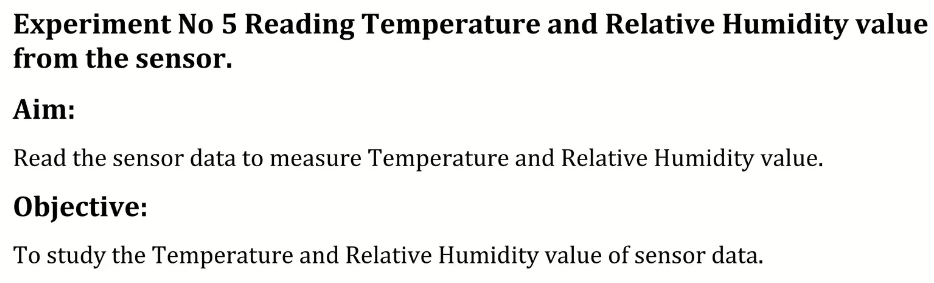


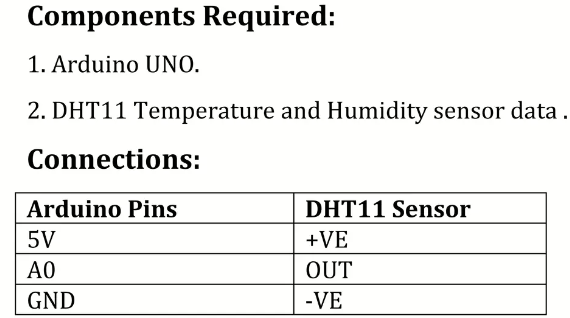


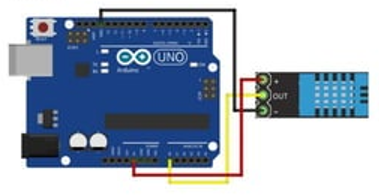


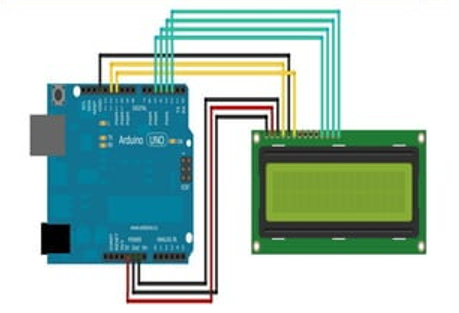


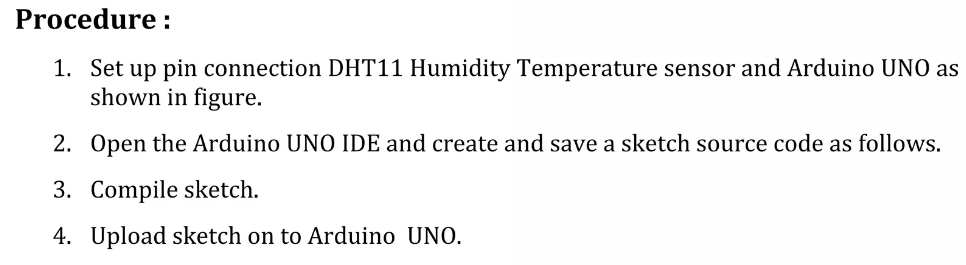


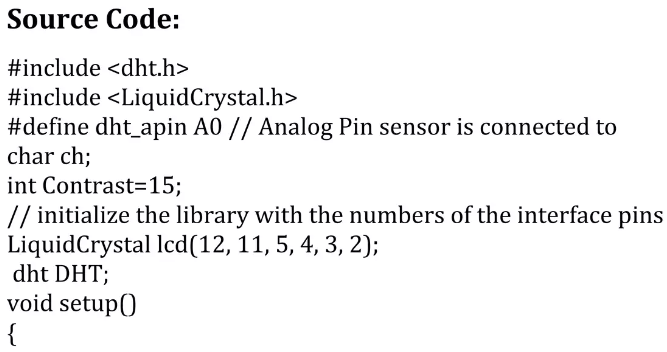


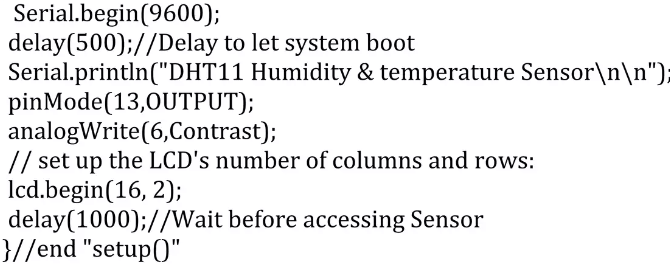


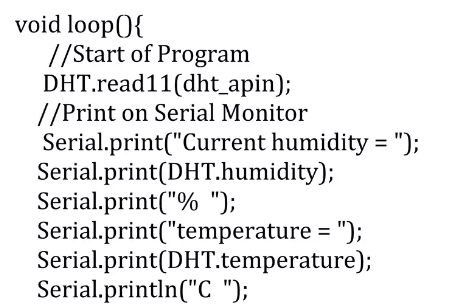


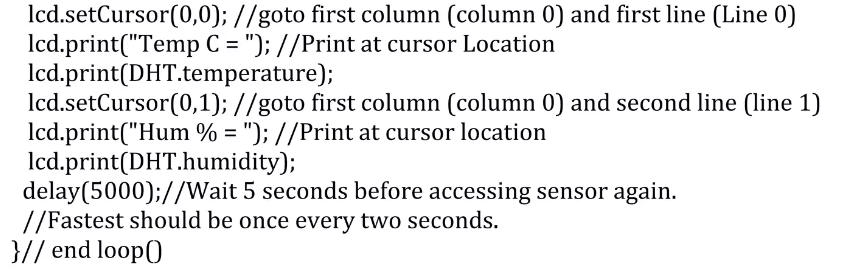












OUTPUT:

