Bluetooth

Bluetooth is a way of simple communication uses the UART protocol to make it easy to send and receive data wirelessly , Built in antenna with a range of up to 30 feet.

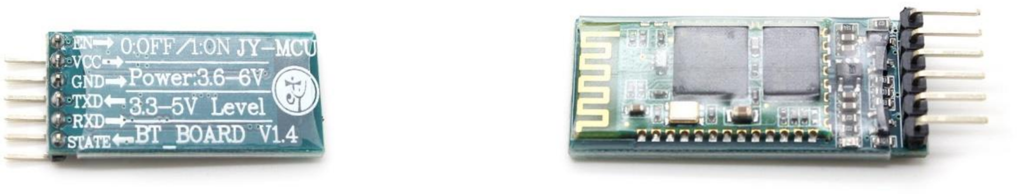

Figure 5-8: Bluetooth module .

In this experiment you’ll need :

1- Bluetooth module .
2- Arduino .
3- 270 Ω Resistor .
4- LED .
5- Wires .

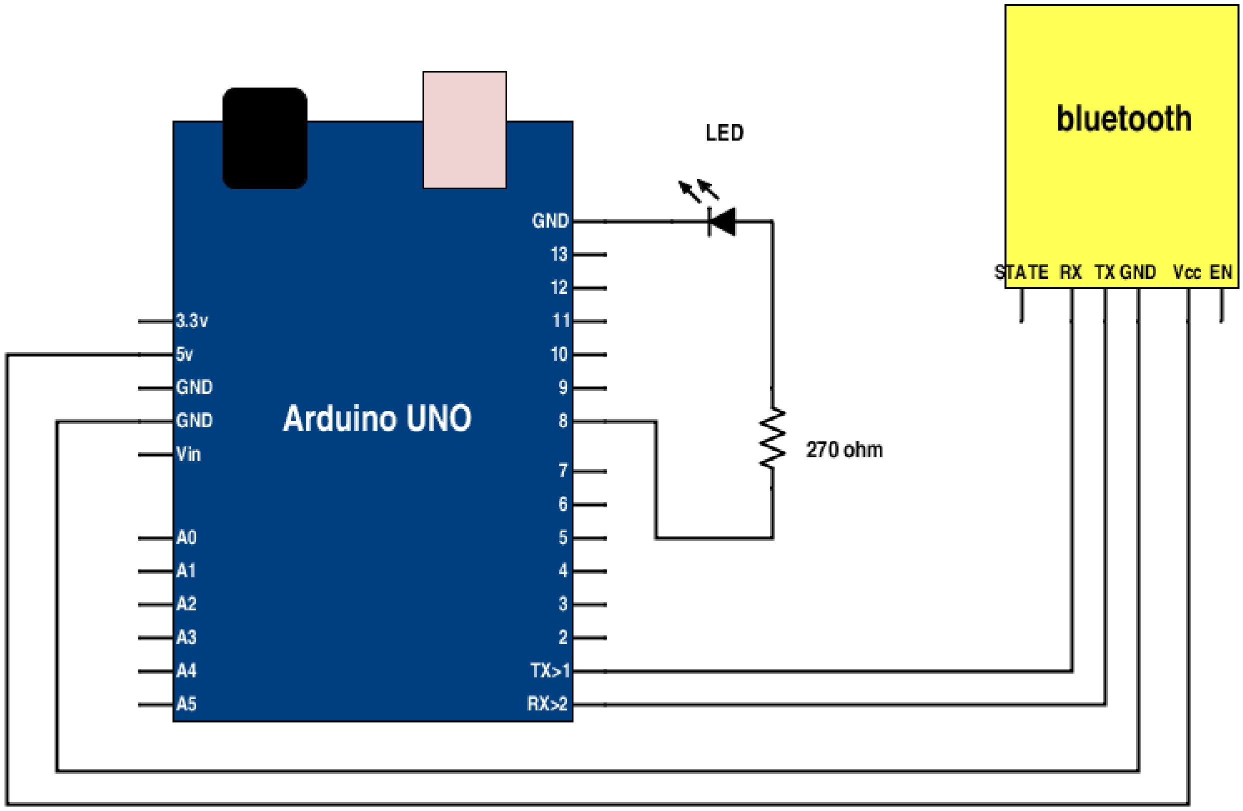

Figure 5-9: control LED by Bluetooth module and arduino connection diagram .

Let`s move to the code :

```
char val; // variable to receive data from the serial port
int ledpin = 8;
void setup()
 {

pinMode(ledpin, OUTPUT);      LED as OUTPUT
  Serial.begin(9600);        // start serial communication at 9600bps
}

void loop() {

  if( Serial.available() )        // if data is available to read
  {
    val = Serial.read();          // read it and store it in 'val'
  }
  if( val == 'H' )                // if 'H' was received
{
    digitalWrite(ledpin, HIGH);  // turn ON the LED
  } else {
    digitalWrite(ledpin, LOW);   // otherwise turn it OFF
  }
  delay(100);                    // wait 100ms for next reading
}
```

Radio Frequency (RF) module

Radio frequency (RF) is the cheapest communication way . the a rate of oscillation in the range of around 3 KHZ to 300 GHZ **.** The transmitter/receiver RF (Tx/Rx) pair operates at a frequency of 434 MHz .

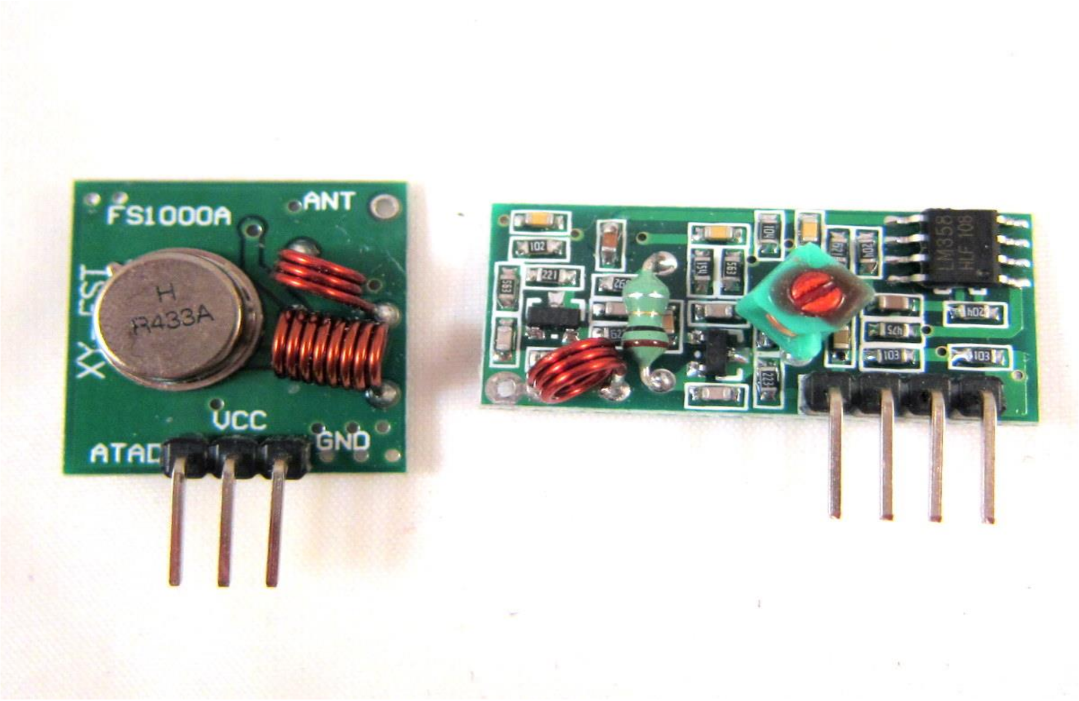

Figure 5-10 : Transmitter module + Receiver module .

In this experiment you'll need :

1- Transmitter module .
2- Receiver module .
3- Two arduino .
4- Wires .

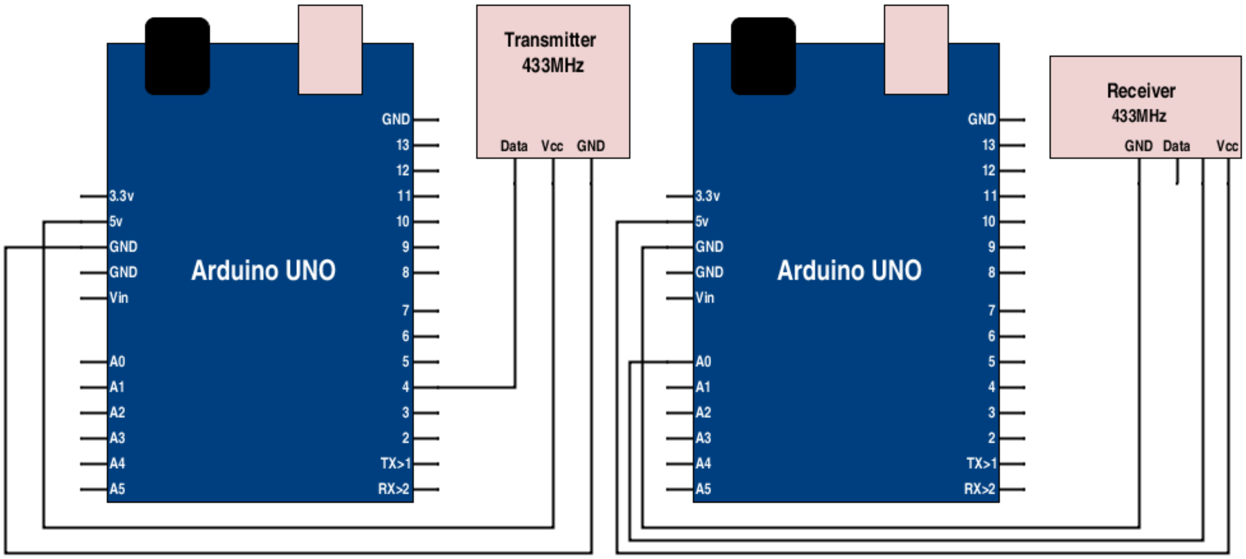

Figure 5-10 : Transmitter + Receiver with arduino connection diagram.

Now let`s move to the code :

We have two code one for Transmitter and one for receiver .

Code for Transmitter :

```
#include <VirtualWire.h>

char *controller;

void setup() {

  pinMode(13,OUTPUT);

vw_set_ptt_inverted(true); //
vw_set_tx_pin(4);

vw_setup(4000);// speed of data transfer Kbps

}

void loop(){

controller="1"  ;

vw_send((uint8_t *)controller, strlen(controller));

vw_wait_tx(); // Wait until the whole message is gone

digitalWrite(13,1);

delay(2000);

controller="0"  ;

vw_send((uint8_t *)controller, strlen(controller));

vw_wait_tx(); // Wait until the whole message is gone

digitalWrite(13,0);

delay(2000);

}
```

Code for Receiver :

```
#include <VirtualWire.h>

char *controller;

void setup() {

  pinMode(13,OUTPUT);

vw_set_ptt_inverted(true); //

vw_set_tx_pin(A0);
vw_setup(4000);// speed of data transfer Kbps
}

void loop(){
controller="1"  ;
vw_send((uint8_t *)controller, strlen(controller));
vw_wait_tx(); // Wait until the whole message is gone
digitalWrite(13,1);
delay(2000);
controller="0"  ;
vw_send((uint8_t *)controller, strlen(controller));
vw_wait_tx(); // Wait until the whole message is gone
digitalWrite(13,0);
delay(2000);

}
```

Radio-Frequency Identification (RFID)

RFID – RC522 uses the SPI interface .serves the same purpose as a bar code or a magnetic strip on the back of a credit card or ATM card; it provides a unique identifier for that object (ID for object). the RFID device must be scanned to retrieve the identifying information.

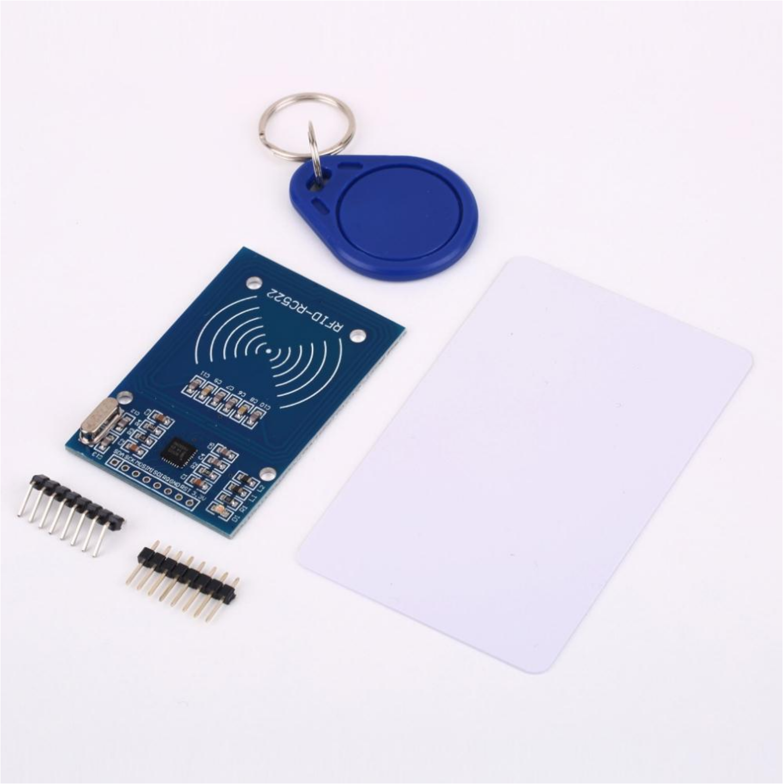

Figure 5-11 : RFID-RC522 module.

In this experiment you'll need :

1- RFID-RC522 module.
2- Arduino Uno .
3- Card .
4- Wires.

To connect the RFID-RC522 to arduino you have to connect :

- Reset pin » Pin 5 .
- SS pin » Pin 10 .
- MOSI pin » Pin 11 .
- MISO pin » Pin 12 .
- SCK pin » Pin 13 .
- GND pin » GND .
- 3.3v pin » 3.3v .

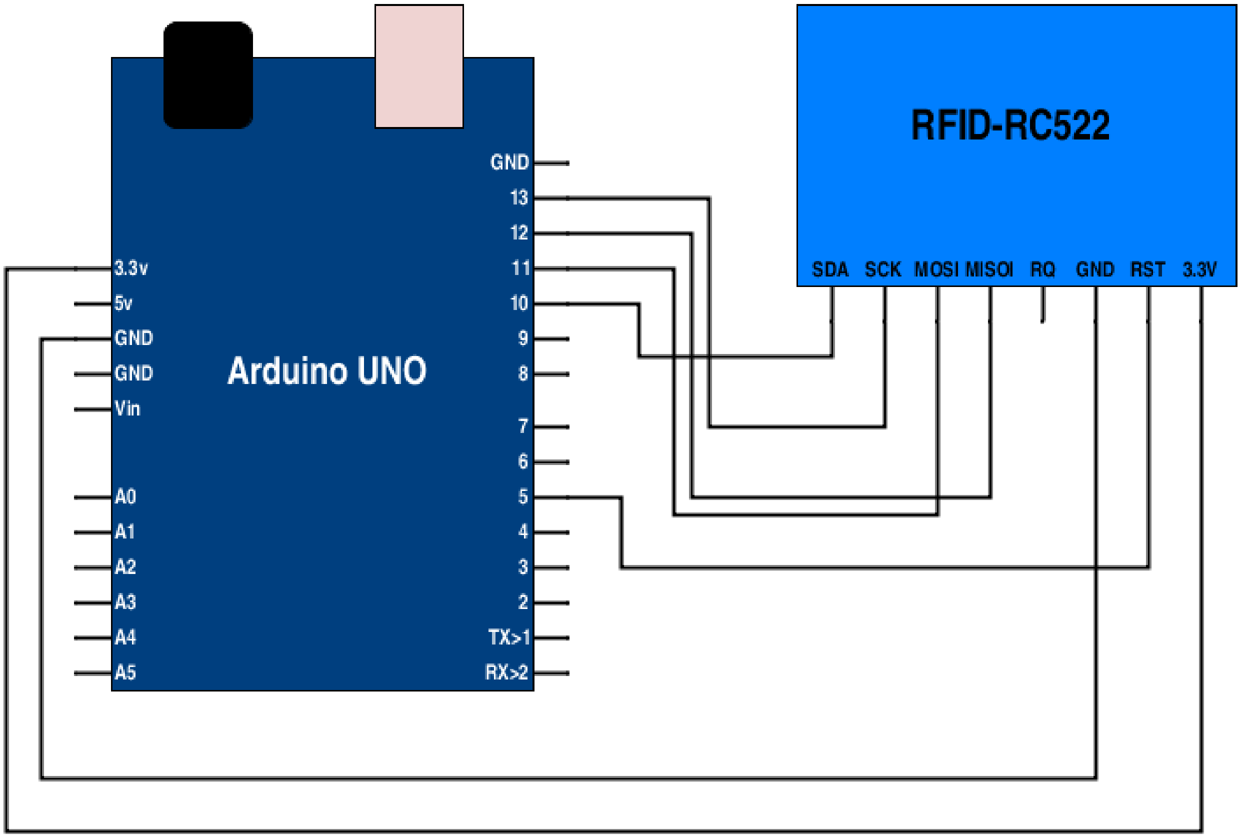

Figure 5-12 : RFID-RC522 module with arduino connection diagram.

Now let`s move to the code :

```
#include <SPI.h>   //   Library already build in arduino .
#include <RFID.h> // Arduino library you have to download it .
#define SS_PIN 10
#define RST_PIN 9
RFID rfid(SS_PIN, RST_PIN);
int x;
// Setup variables:
    int serNum0;
    int serNum1;
    int serNum2;
    int serNum3;
    int serNum4;
void setup()
{
  pinMode(5,OUTPUT);
  Serial.begin(9600);
  SPI.begin();
  rfid.init();
}
void loop()
{
  if (rfid.isCard()) {
  if (rfid.readCardSerial()) {
 if (rfid.serNum[0] != serNum0
  && rfid.serNum[1] != serNum1
  && rfid.serNum[2] != serNum2
  && rfid.serNum[3] != serNum3
&& rfid.serNum[4] != serNum4
            ) {
   /* With a new cardnumber, show it. */
  Serial.println(" ");
Serial.println("Card found");
  serNum0 = rfid.serNum[0];
  serNum1 = rfid.serNum[1];
  serNum2 = rfid.serNum[2];
  serNum3 = rfid.serNum[3];
  serNum4 = rfid.serNum[4];
   x=rfid.serNum[0];
  //Serial.println(" ");
   Serial.println("Cardnumber:");
   Serial.print("Dec: ");
  Serial.print(rfid.serNum[0],DEC);
    Serial.print(", ");
   Serial.print(rfid.serNum[1],DEC);
   Serial.print(", ");
    Serial.print(rfid.serNum[2],DEC);
      Serial.print(", ");
Serial.print(rfid.serNum[3],DEC);
  Serial.print(", ");
Serial.print(rfid.serNum[4],DEC);
 Serial.println(" ");
Serial.print("Hex: ");
Serial.print(rfid.serNum[0],HEX);
Serial.print(", ");
Serial.print(rfid.serNum[1],HEX);
Serial.print(", ");
Serial.print(rfid.serNum[2],HEX);
Serial.print(", ");
Serial.print(rfid.serNum[3],HEX);
Serial.print(", ");
Serial.print(rfid.serNum[4],HEX);
 Serial.println(" ");
   } else {
/* If we have the same ID, just write a dot. */
  Serial.print(".");
    }
    }
         }
rfid.halt();
delay(500);
Serial.println(x);
  if (x==173){
```

```
Serial.println("medal");

digitalWrite(5,HIGH);

delay(500);

  }

if (x==82){

  Serial.println("card");

digitalWrite(5,LOW);

delay(500);

    }

}
```

Wireless 433MHz :

HC-11 wireless serial port communication module(433Mhz transceiver) Its working frequency band is 433.4-473.0MHz , and the communication distance is 1,000m in open space.

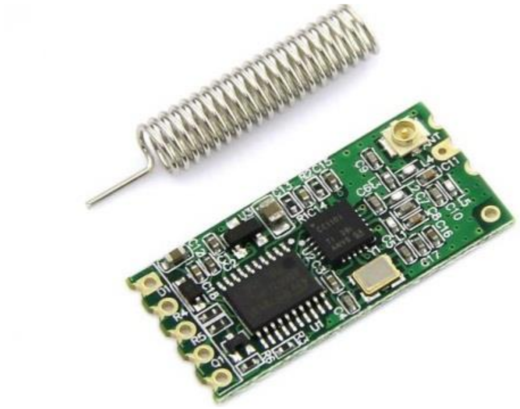

Figure : wireless 433MHz.

We have three experiment :

In the first experiment you'll need :

1. Two arduino Uno.
2. Two 433Mhz wireless .
3. Wires .
4. LED .

See connection diagram :

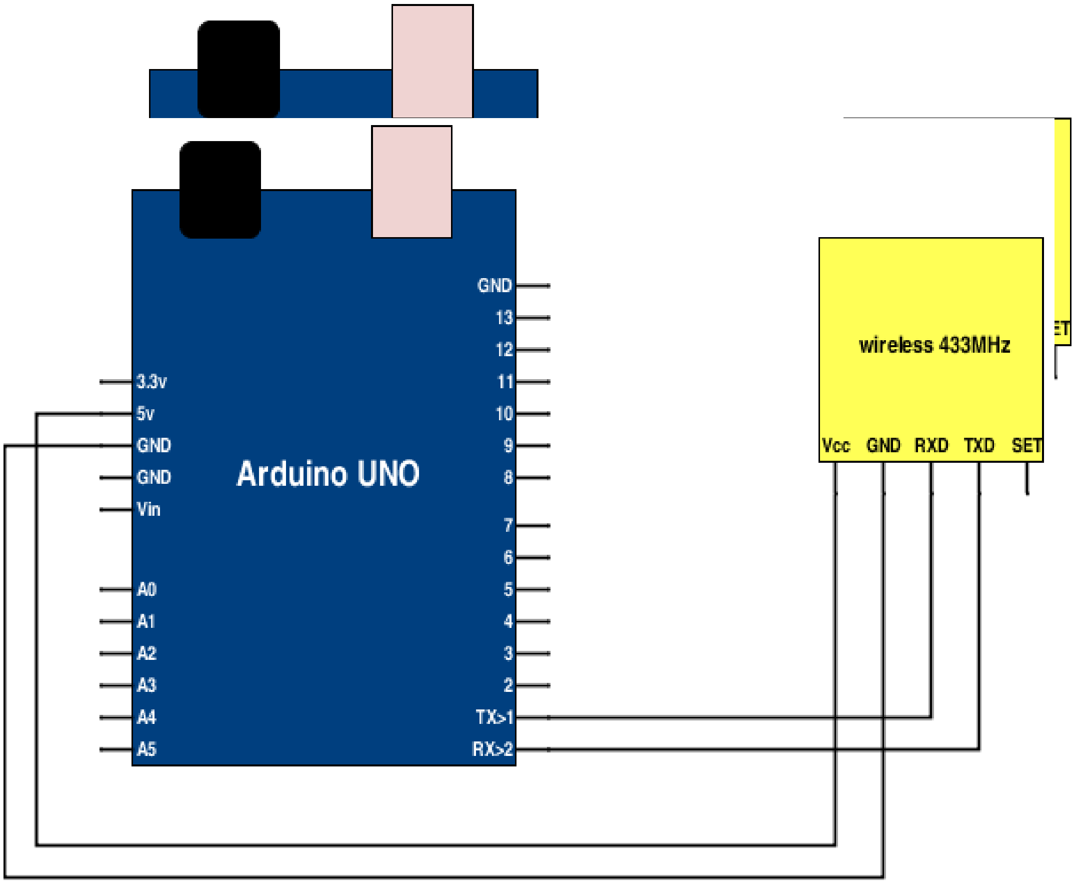

Figure :

Transmitter connection diagram .

Figure : receiver connection diagram .

Now let`s move to the code :

Transmitter code :

```
void setup(){
 Serial.begin(9600);
}
void loop(){
  Serial.write("1");
  delay(1000);
  Serial.write("0");
  delay(1000);
}
```

Receiver code :

```
int x;
void setup(){
 Serial.begin(9600);
  pinMode(13,OUTPUT);   }
void loop(){
 x=Serial.read();
 if (x=='1'){
   digitalWrite(13,1);  }
if (x=='0'){
   digitalWrite(13,0);   }
}
```

In the second experiment you'll need :

1. Two arduino Uno.
2. Two 433Mhz wireless .
3. Temperature sensor (LM35).
4. Wires .

See connection diagram :

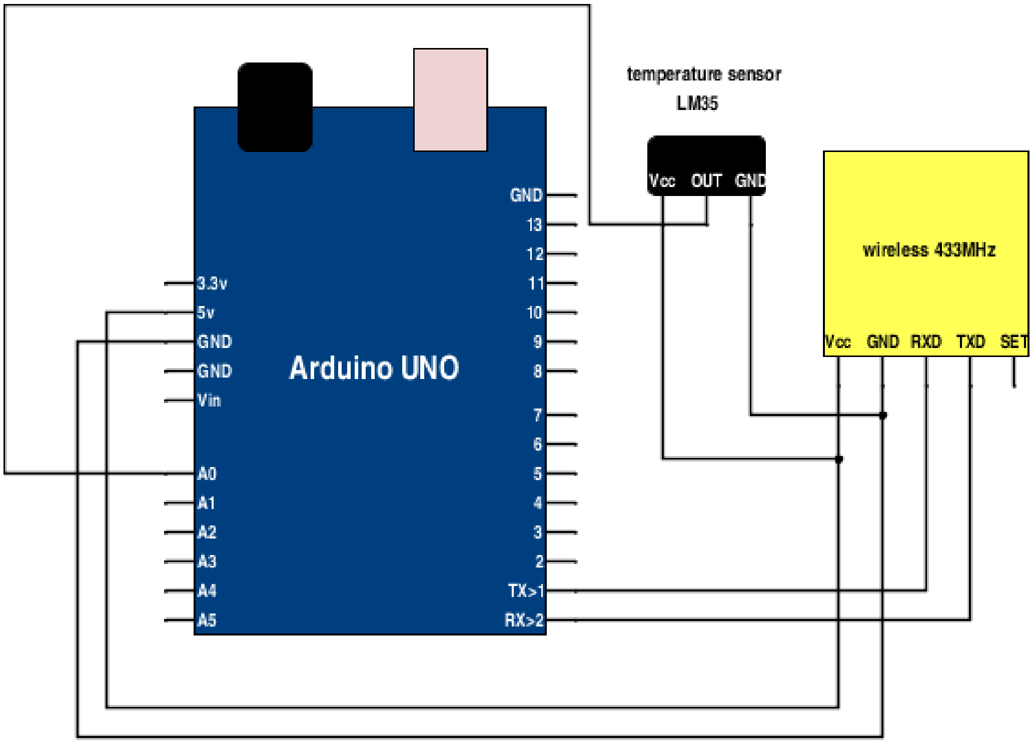

Figure   : Transmitter connection diagram .

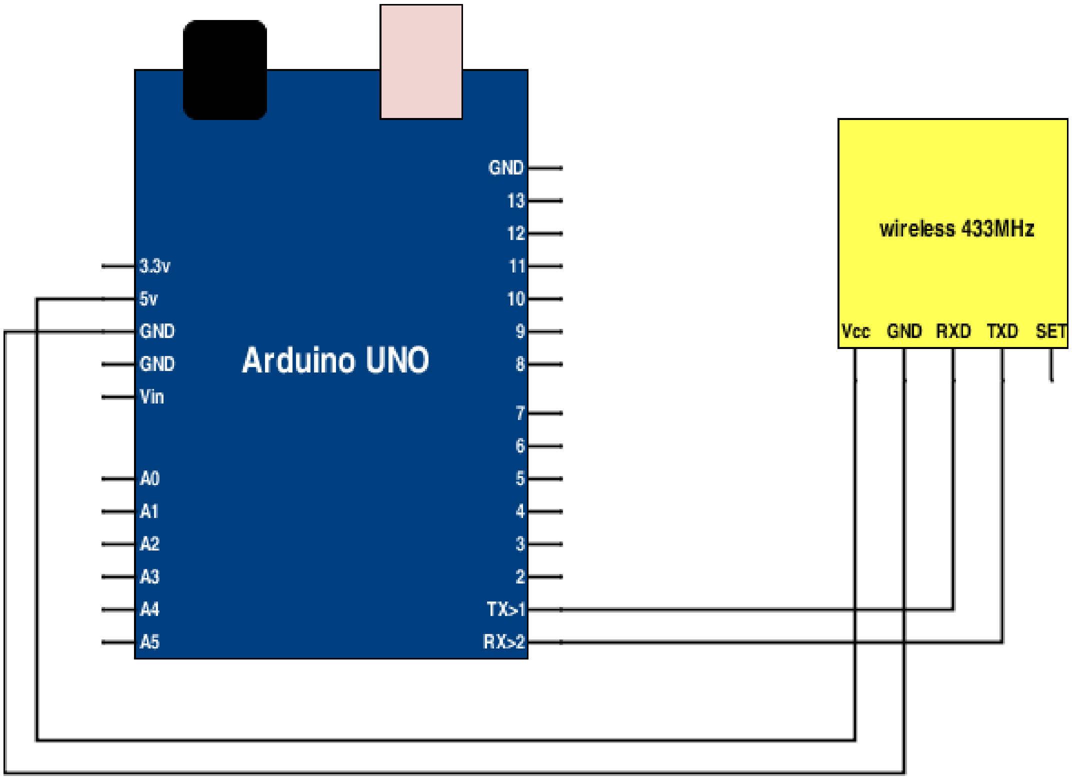

Figure : receiver connection diagram .

Now let`s move to the code :

Transmitter code :

```
int tempC;

 int tempPin = 0;

 void setup()  {

  Serial.begin(9600);

 }

 void loop()  {

  tempC = analogRead(tempPin);

 tempC = (5.0 * tempC * 100.0)/1024.0;

 Serial.write(tempC);

 delay(30);   }
```

Receiver code :

```
int tempC;
void setup()  {
Serial.begin(9600);
 }
 void loop()  {
   if (Serial.available()){
 tempC=Serial.read();
 Serial.println(tempC);
   }
   delay(30);
 }
```

In the third experiment you’ll need :

1. Two arduino Uno.
2. Two 433Mhz wireless .
3. Two analog sensor .
4. Wires .

Now let`s move to the code :

Transmitter code :

```
int val1,val2;
void setup() {
Serial.begin(9600);
}
void loop() {
val1 = analogRead(1);
val2 = analogRead(0);
Serial.write(">");
Serial.write(val1);
Serial.write(val2);
delay(30);    }
```

Receiver code :

```
byte start;
int val1,val2;
void setup(){
  Serial.begin(9600);  }
void loop(){
 if (Serial.available()){
  start=Serial.read();
  if (start=='>'){
   val1=Serial.read();
  val2=Serial.read();   }
delay(30);   }  }
```

GPS module :

Global positioning system module It enables you to find the position on the map, and give you the latitude and longitude .

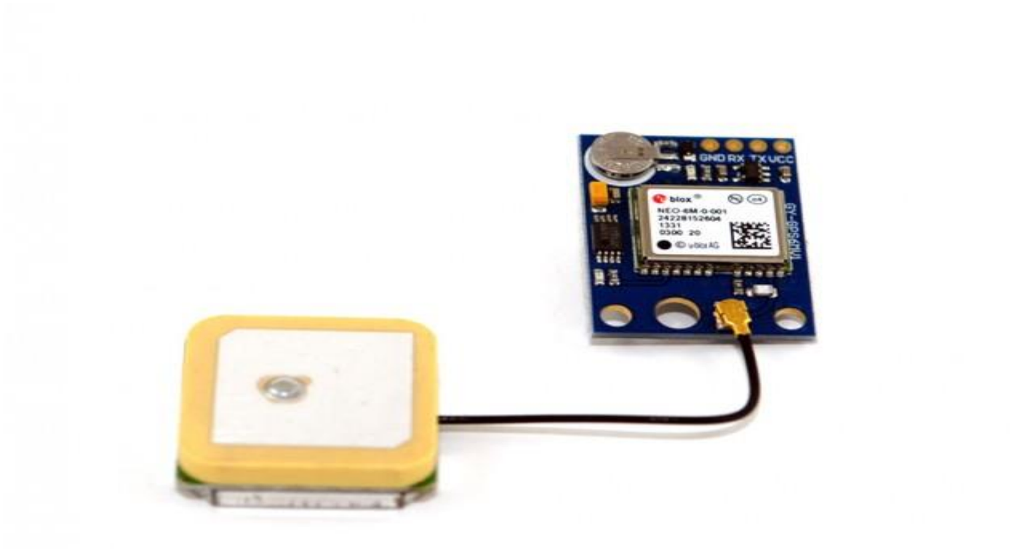

Figure : gps module .

In this experiment you will need :

1- **Arduino board .**
2- **GPS module .**
3- **Wires .**

See connection diagram :

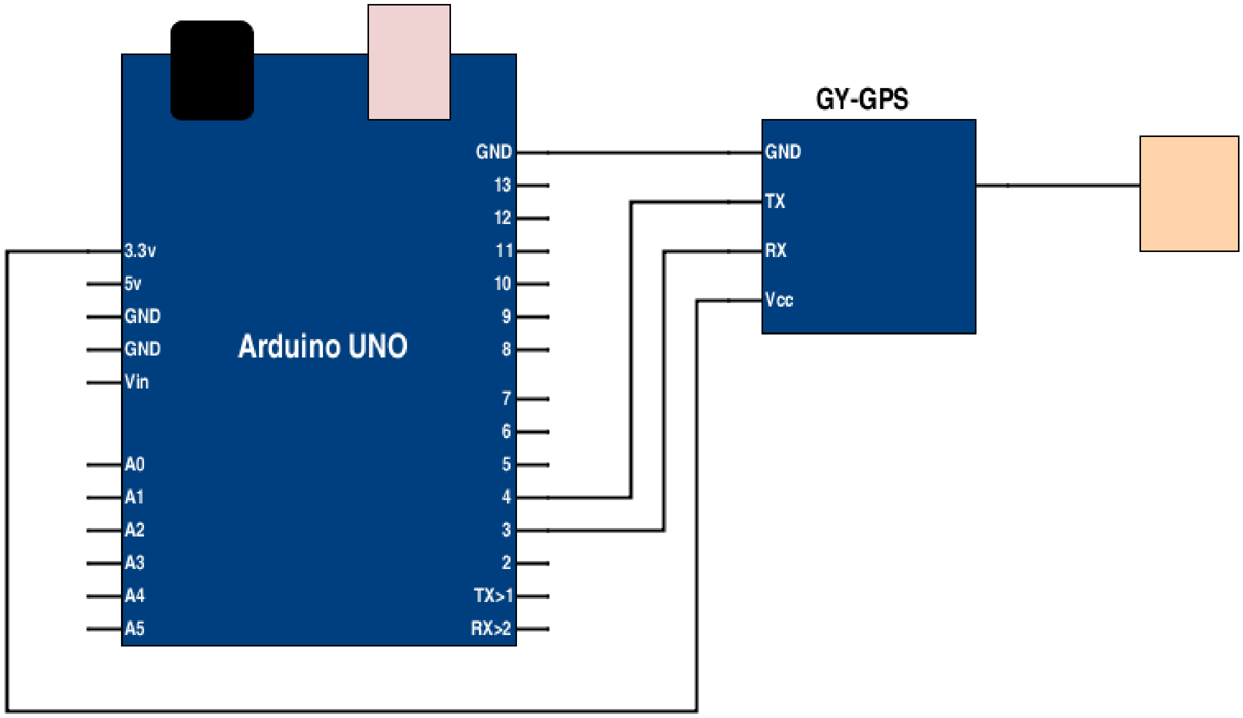

Figure : GPS module with arduino connection diagram .

Now let`s move to the code :

```
void setup() {
Serial.begin(9600);
Serial1.begin(9600);
}
void loop() {
if (Serial1.available()) {
  int GPS_data = Serial1.read();
  Serial.write(GPS_data);
}
}
```

GSM module :

Enable user to send and receive massages like mobile phone ,you can find may version of GSM like GSM SIM900 ,GSM SIM908 .

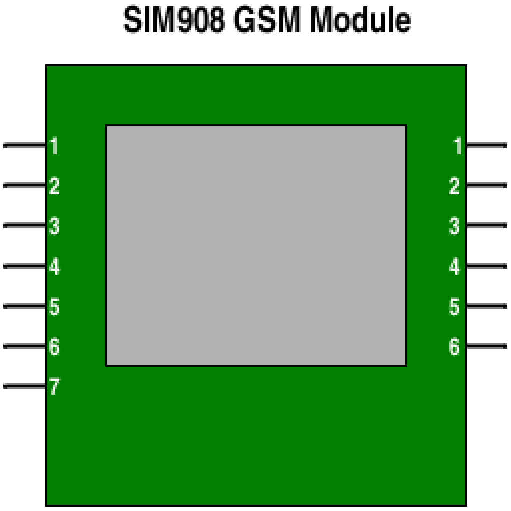

Figure : GSM module .

In this experiment you will need :

1- **Arduino board .**
2- **GSM module .**
3- **SIM card .**
4- **Wires .**

**** Connection between GSM to Arduino :**

Pin 3 (TX pin) → RX pin .

Pin 4 (RX pin) → Tx pin .

Pin 5 (GND pin) → GND pin .

Pin 6 (VCC pin) → VCC pin .

See connection diagram :

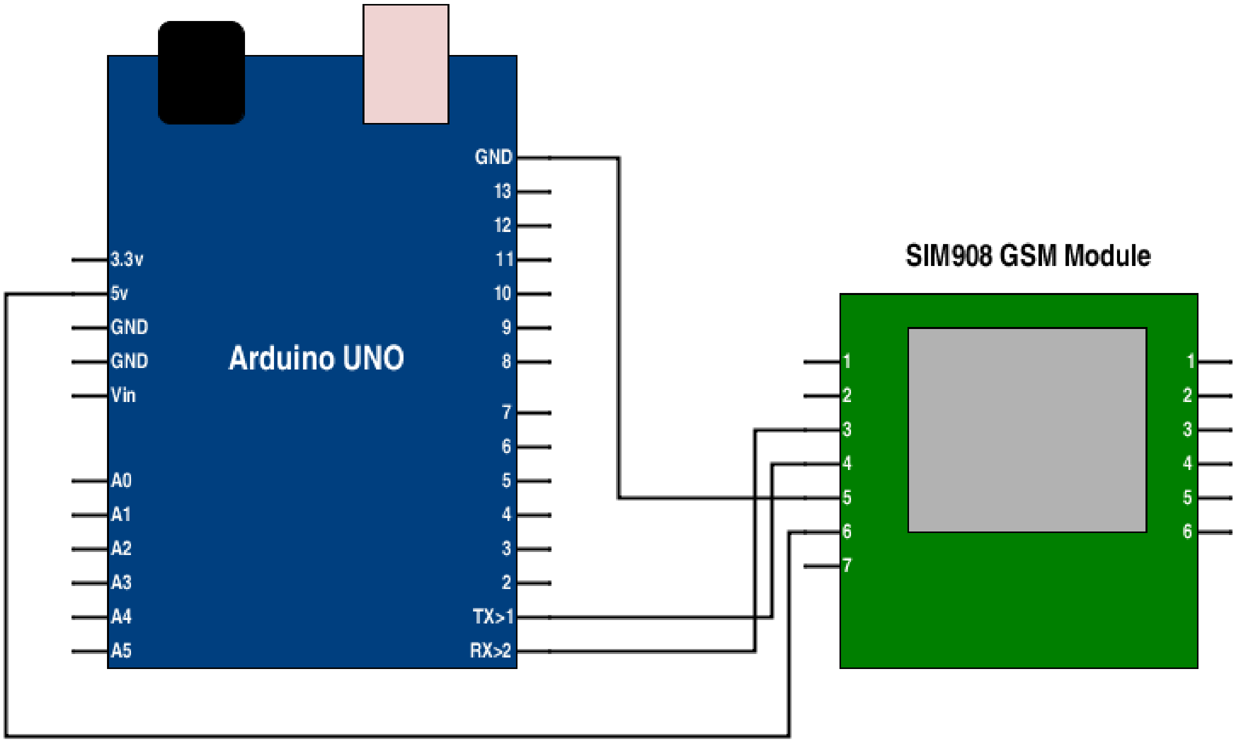

figure : GSM Module with arduino connection diagram .

Now let`s move to the code :

```
#include "SIM900.h"
#include <SoftwareSerial.h>
#include "sms.h"
SMSGSM sms;
boolean started=false;
void setup() {
Serial.begin(9600);
 Serial.println("GSM Shield testing.");
if (gsm.begin(2400)){
   Serial.println("\nstatus=READY");
   started=true;
 }
else Serial.println("\nstatus=IDLE");

  if(started){
    if (sms.SendSMS("+6148123123123", "SMS from my phone"))
      Serial.println("\nSMS sent OK");
  }

};

void loop()
{

}
```

SD card :

Sd card module used for reading and saving analog value from arduino .

In this experiment we will save two analog potentiometer reading to sd card .

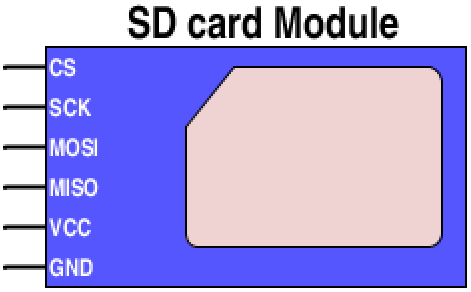

Figure : SD card Module .

In this experiment you will need :

1- **Arduino board .**
2- **SD card module .**
3- **Wires .**
4- **Two potentiometer .**

See connection diagram :

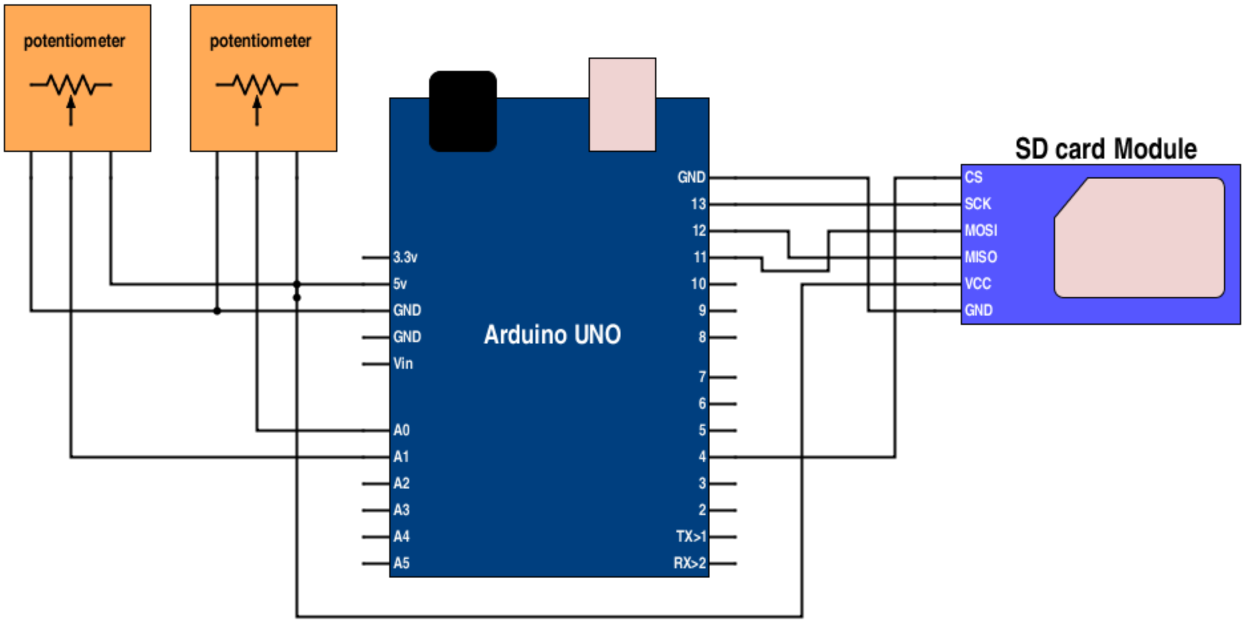

figure : SD card with Arduino connection diagram .

Now let`s move to the code :

```
#include <SPI.h>

#include <SD.h>

const int chipSelect = 4;

void setup() {

  Serial.begin(9600);

  Serial.print("Initializing SD card...");

  if (!SD.begin(chipSelect)) {

    Serial.println("Card failed, or not present");

    return;

  }

  Serial.println("card initialized.");

}
```

```
void loop() {
  String dataString = "";
  for (int analogPin = 0; analogPin < 2; analogPin++) {
    int sensor = analogRead(analogPin);
    dataString += String(sensor);
    if (analogPin < 1) {
      dataString += ",";
    }
  }
  File dataFile = SD.open("datalog.txt", FILE_WRITE);
  if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    Serial.println(dataString);
  }
  else {
    Serial.println("error opening datalog.txt");
  }
}
```