

## **Ex. No. 1 EXPLORE DIFFERENT COMMUNICATION METHODS WITH IOT DEVICES.**

### **OBJECTIVE:**

To explore different communication methods with IoT devices such as interfacing Arduino to Zigbee module, interfacing Arduino to GSM module and interfacing Arduino to Bluetooth module.

### **REQUIRED COMPONENTS:**

- Arduino UNO
- Arduino IDE
- Zigbee Module
- SIM 900 GSM Module
- Full-Size SIM Card (Unlocked)
- HC-05 Bluetooth Module
- 12V 2Amp Power Supply
- Breadboard
- USB cable for uploading code into Arduino UNO
- LED
- Jumper Wires
- Android App - Arduino Bluetooth Controller

### **BACKGROUND THEORY:**

#### **Zigbee**

Zigbee can be interfaced with external devices very easily. Zigbee can communicate with external devices like sensors and communications devices. There are many wireless modules available in the market like RF, RC transmitter and receiver. But the range is low. Zigbee has a higher range than other wireless modules.

Zigbee is a wireless communication module which uses IEEE 802.15.4 standard. 802.15.4 is an IEEE standard for low power applications of radio frequency. It is used in many products nowadays for wireless communication functionality. It can be used as a transmitter and receiver both. It used serial communication to send and receive data. It

has two series, series1 and series 2. Series 1 is comparatively easy to use and it is recommended for beginners. Series 1 Zigbee module cannot work in a mesh network. Meaning it cannot talk to more than one Zigbees buddies.

### **Working of Zigbee Technology**

Zigbee uses a serial port to send and receive data. So it means it can easily interface with Arduino Uno, any type of microcontroller and computer. Because they all support serial communication and they all have a serial port to send and receive data. It can also communicate with other Zigbee to form a mesh. Zigbee can also be used to make a local area network. It has many applications.

### **Applications of Zigbee**

Some of the famous applications are given below:

- Wireless communication
- wirelessly controlled robot
- remote monitoring system
- Wireless home automation system
- wireless temperature sensor and many others.

Zigbee alone can't do anything. It is used to interface it with some intelligent device like microcontrollers, Arduino and computer. These devices will tell it what to do or what not to do through an already fed program inside microcontrollers and Arduino Uno. These digital devices are not so intelligent. But to make devices intelligent by writing few lines of instructions.

### **Interfacing Zigbee with Arduino**

To interface it, know which series of it is being used. Two Zigbee modules can talk with each other if both are of the same type. To communicate its modules with each other, obtain its modules. Connect one module to Arduino and another module to either sensor or any microcontroller or computer. It is possible to set the configuration of both modules.

## Zigbee Interfacing with Arduino.

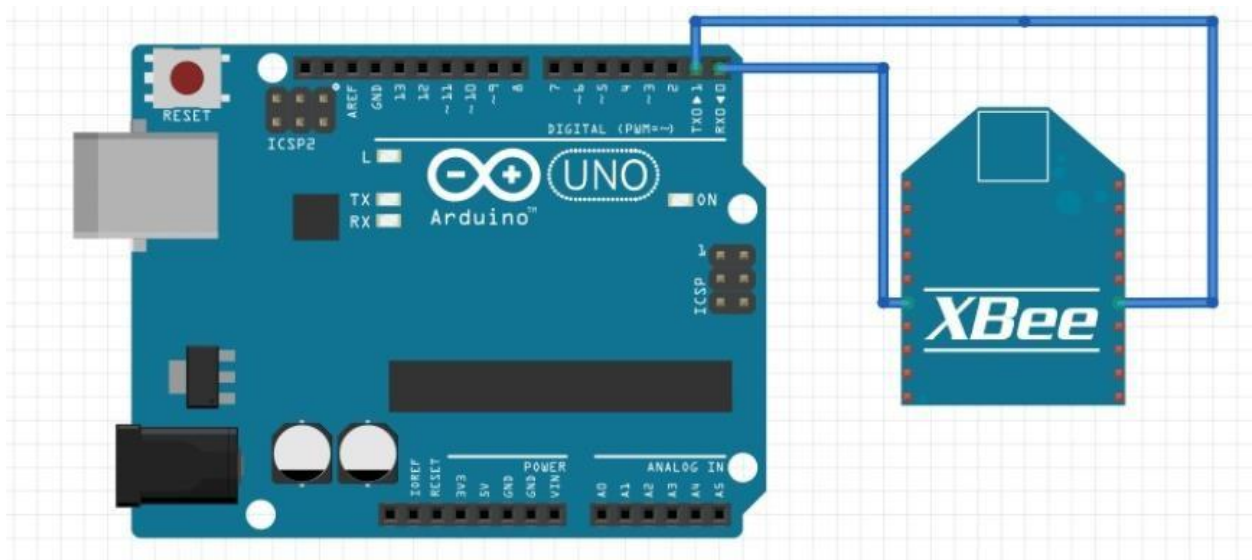


Figure above shows the connection diagram of the module with Arduino. The module should have regulated 5 volts and 3.3 volt. To use the Adafruit XBee Adapter, it has both voltage levels. Otherwise, use a separate power supply. In the above circuit. The TX and RX pins of Zigbee and Arduino are connected to each other. Arduino will send some instructions to Zigbee and according to these instructions Zigbee responds. Similarly, Zigbee receives instructions from other Zigbee to which it has an address. After receiving instructions or data from other Zigbee. It sends data to the arduino through serial pins as shown in connection. Similarly, other modules can be connected with one more Arduino or computer. Same connection diagram is used for other Zigbee and arduino pairs.

### **CODING:**

```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  while (Serial.available() ) {
    Serial.write(Serial.read()); /
```

```
}  
}
```

Upload the above code in Arduino after connecting Arduino with it. Arduino starts receiving whatever Zigbee sends to Arduino through serial communication.

### **Configuring Zigbee's**

Follow the following instructions to configure Zigbee series 1 module:

1. Connect the zigbee's module to the computer serial port using a serial adapter.
2. Download virtual terminal or PuTTY for windows.
3. Set the computer serial port. confirm that it is set to a baud rate of 9,600, 8 data bits, no parity, 1 stop bit.
4. Click on the local Echo box.
5. Click ok.
6. Save the time of the session.
7. Click to the connect button on the virtual terminal.
8. After connection with Zigbee's give the name of the session.
9. Now it is connected with the Zigbee module.
10. Use AT commands to configure the module.

After that type +++ on virtual terminal and after few seconds it will respond with OK after that type these commands:

ATMY1234

ATDL5678

ATDH0

ATID0

ATWR

Zigbee's module is ready to be used. Now connect other zigbee's module with other serial port and repeat the above 10 steps. Other Zigbee module will also respond with commands and following instructions:

ATMY5678

ATDL1234

ATDH0

ATID0

ATWR

Now both zigbees modules are ready to talk with each other.

There are different kinds of GSM modules available on the market. The most popular module based on Simcom SIM900 and Arduino Uno is used. Interfacing a GSM module to Arduino is pretty simple. It is only needed to make 3 connections between the gsm module and Arduino.

### **GSM SIM 900 Module**

- The GSM SIM 900 Module is a type of Arduino Shield, which means it can also be mounted on top of Arduino UNO.
- This is a type of modem, used for long-distance data transmission with the use of GSM technology where there is no internet connectivity.
- This makes it useful in projects which require remote data transmission. Many projects can be made using this module such as call or text message-based triggers which can be used in the day-to-day life.
- Farmers can extensively use this technology in the day-to-day life, such as controlling water pumps with a simple silent call or a text message sitting at the home.
- For example, a gsm based agriculture project. This can also be used to send the data from the module to the mobile phones without the use of the internet.
- Other project ideas include intruder alert notification, Receive timely updates from the sensor to the mobile, etc.

### **Working of GSM SIM 900 Module**

The GSM module uses GSM and GPRS technology to communicate with another device wirelessly. It uses a 2G network to connect with the internet and supports Quad-band (EGSM 900, GSM 850, DCS 1800, PCS1900). Because of this, one can send or receive messages from this or make or receive voice calls using the module by connecting the microphone and speakers to the respective ports given on it. This module can be used for security purposes like gsm-based forest fire alerts and control systems. This Module

also houses an inbuilt RTC to keep track of time, which is very helpful for timer-based applications. The versatility of this module is very high due to its ability to read and send messages without any hassle. Because of the use of AT Commands, it is very easy to configure. because of the presence of an external antenna, the module can also be used in areas with low signal areas.

### GSM SIM 900 Module.



### Interfacing GSM Module with Arduino Circuit Diagram.

**Connect 12V 2A Adapter  
to the GSM**



## Connection Table

Arduino UNO	GSM SIM 900 Module
TX	RX
RX	TX
GND	GND
12 Volt 2 Amp Adaptor	GSM SIM 900 Module
Connect	Connect

## Commands Used for Operation

Command	Description
AT+CSMS	To Select message service
AT+CPMS	To Preferred message storage
AT+CMGF	select Message format
AT+CSCA	Service center address
AT+CSMP	Set text mode parameters in sim
AT+CSDH	To Show text mode parameters in Sim
AT+CSCB	Select cell broadcast message types
AT+CSAS	Save settings in the gsm module
AT+CRES	Restore all settings
AT+CNMI	Message indications to TE
AT+CMGL	To make the list of messages
AT+CMGR	Read new message
AT+CMGS	Send a new message
AT+CMSS	Send message from sim storage
AT+CMGW	Write a message to gsm memory
AT+CMGD	Delete message

## **CODING:**

### **Interfacing GSM Module with Arduino Code**

```
// Download Library of SoftwareSerial link given
// https://github.com/PaulStoffregen/SoftwareSerial
#include <SoftwareSerial.h>

SoftwareSerial SIM900A(10,11); // SoftSerial( RX , TX );
// 10 pin connect to TX of GSM SIM 900 Module
// 11 pin connect to RX of GSM SIM 900 Module

void setup()
{
  SIM900A.begin(9600); // Setting the baud rate of GSM Module
  Serial.begin(9600); // Setting the baud rate of Serial Monitor (Arduino)
  Serial.println ("SIM900A Ready");
  delay(100);
  Serial.println ("Type s to send message or r to receive message");
}

void loop()
{
  if (Serial.available()>0)
  switch(Serial.read())
  {
    case 's':
      SendMessage();
      break;
    case 'r':
      ReceiveMessage();
      break;
  }
  if (SIM900A.available()>0)
  Serial.write(SIM900A.read());
}

void SendMessage()
```



```

{
Serial.println ("Sending Message");
SIM900A.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
delay(1000);
Serial.println ("Set SMS Number");
SIM900A.println("AT+CMGS=\"911234567890\\r"); //Type the Mobile number to send
message
delay(1000);
Serial.println ("Set SMS Content");
SIM900A.println("Good morning, how are you doing?");// Messsage content
delay(100);
Serial.println ("Finish");
SIM900A.println((char)26);// ASCII code of CTRL+Z
delay(1000);
Serial.println ("Message has been sent ->SMS Selesai dikirim");
}
void ReceiveMessage()
{
Serial.println ("SIM900A Membaca SMS");
delay (1000);
SIM900A.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live SMS
delay(1000);
Serial.write ("Unread Message done");
}

```

### **Bluetooth Module (HC-05)**

The Bluetooth module is a device which is used for short range wireless communication to the respective connected device. This module uses serial port protocol for the wireless communication and comes with two configurations that are master and slave. In the master mode the module searches for the other devices to connect and can connect to the other devices. However, in the slave mode the module cannot connect to the devices

by itself. In short the master more the device controls other devices and in slave mode the device is being controlled by some other device. To change the master slave configuration, it is possible to use the AT commands of the Bluetooth module. Moreover, to use the AT mode it is possible to set the baud rate of 38400 and for serial communication use the baud rate of 9600.

- Operating Voltage: 4 V to 6V (have internal 3.3V regulator).
- Operating Current: 30mA
- Integrated antenna and an edge connector.
- Range about 10 meters.
- Configurable in both master and slave modes.
- Pins: STATE, RXD, TXD, GND, VCC, KEY/ENABLE

#### **HC-05 Bluetooth Module.**



#### **Pin Out**

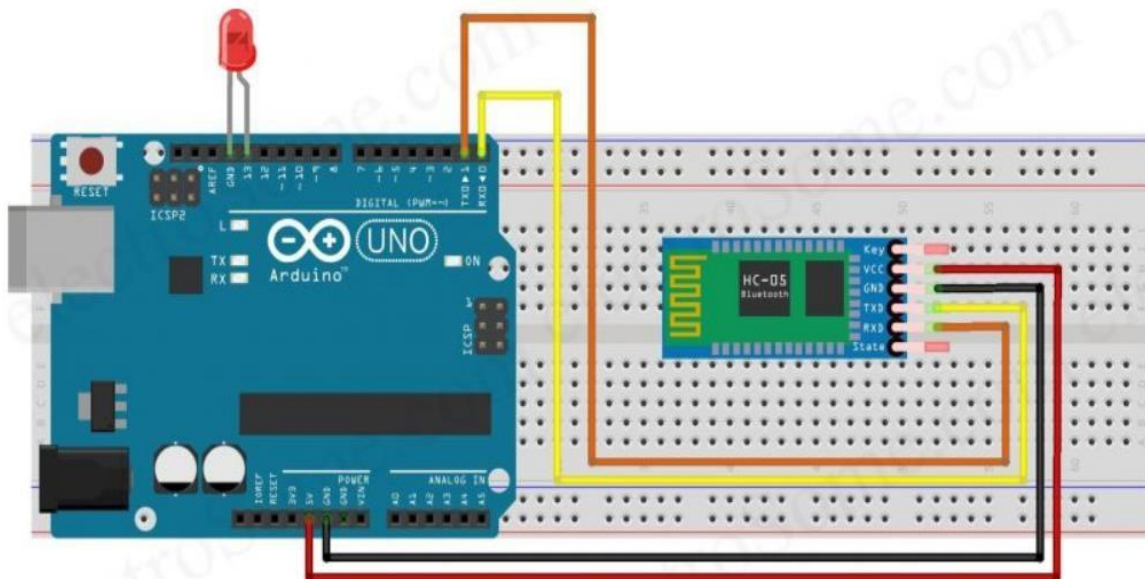
- **STATE:** State pin indicates whether the module is connected or paired with a device. When the module is not connected, this pin will be in LOW state and the on-board LED will be flashing fast. But when the module is paired or connected to

a device, the state pin will be in HIGH state and the on-board LED will be flashing with a delay.

- **RXD:** This is a UART RX pin. This pin is used to send AT commands when the module is in command mode. And it is used to send data to the connected device when the module is in data mode.
- **TXD:** This is a UART TX pin. This pin is used to push out responses to AT command when the module is in command mode. And it is used to push out data sent by the connected device when the module is in data mode.
- **GND:** Power supply -ive.
- **VCC:** Power supply +ive.
- **EN/KEY:** This input is used to switch between command and data mode. If this pin is set HIGH, the module will be in command mode. Similarly, if this pin is set LOW, the module will be in data mode.

## Circuit Diagram

### Interfacing HC-05 Bluetooth Module with Arduino Uno.



## Description

- RXD pin of HC-05 Bluetooth – TXD pin of Arduino Uno
- TXD pin of HC-05 Bluetooth – RXD pin of Arduino Uno

- GND pin of HC-05 Bluetooth – GND pin of Arduino Uno
- VCC pin of HC-05 Bluetooth – 5V output pin of Arduino Uno
- Positive pin of LED - Pin 13 of Arduino Uno
- Negative pin of LED - GND pin of Arduino Uno

### **Arduino Bluetooth Controller**

Installed Arduino Bluetooth app from Google Play Store. This app will act as a Bluetooth remote controller for Arduino. It is very easy to use this app. Open the app and connect to the HC-05 device. Then select the option as switch mode. Now it is possible to control the LED using the app.

### **CODING:**

```
char data = 0;          //Variable for storing received data
void setup()
{
  Serial.begin(9600);    //Sets the data rate in bits per second (baud) for serial data
                        transmission
  pinMode(13, OUTPUT); //Sets digital pin 13 as output pin
}
void loop()
{
  if(Serial.available() > 0)    // Send data only while receiving data:
  {
    data = Serial.read();        //Read the incoming data and store it into variable data
    Serial.print(data);          //Print Value inside data in Serial monitor
    Serial.print("\n");          //New line
    if(data == '1')              //Checks whether value of data is equal to 1
      digitalWrite(13, HIGH);    //If value is 1 then LED turns ON
    else if(data == '0')         //Checks whether value of data is equal to 0
      digitalWrite(13, LOW);     //If value is 0 then LED turns OFF
  }
}
```

### **Working of Bluetooth Module (HC-05)**

- Initialize the serial port (UART) with the default baud rate of HC-05 Bluetooth module.
- Initialize Pin 13 as output pin.
- In the loop() it is possible to keep checking if any data is available to read from the serial port.
- If data is available to read, store it to the variable named "data".
- If the data read is '1' then the LED is turned ON, else the LED will be turned OFF.

### **CONCLUSION:**

Thus the different communication methods with IoT devices such as interfacing Arduino to Zigbee module, interfacing Arduino to GSM module and interfacing Arduino to Bluetooth module were explored successfully.