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
* HC-05 Bluetooth module
* 12 V 2 Amp Power Supply
* Breadboard.
* USB cable
* 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 communication 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. Series1 is comparatively easy to use and it is recommended for beginners. Series1 Zigbee module cannot work in a mesh network. Meaning it cannot talk to more than on 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 :

* 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.

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 Arduino 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 more/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 whether 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 connection 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 transmissions 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 fire alerts and control systems. This module also houses an inbuilt RTC to keep track of time which is very helpful for timer based application.

The versality 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.

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 broad cast 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);
  Serial.begin(9600);
  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");
 delay(1000);
 Serial.println ("Set SMS Number");
 SIM900A.println ("AT+CMGS=\"911234567890\"\r");
 message
 delay(1000);
 Serial.println ("Set SMS content");
 SIM900A.println ("Good morning, how are you doing?");
 delay (100);
 Serial.println ("Finish");
 SIM900A.println (Char) 26);
 delay (1000);
 Serial.println ("Message has been sent -> SMS selesai
                                          dikirim");
}
void Receive Message ()
{
 Serial.println ("SIM900A Membaca SMS");
 delay (1000);
 SIM900A.println ("AT+CNMI=2,2,0,0,0");
 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 mode 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. However, 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 : 4V to 6V (have internal 3.3 V regulator)
* Operating Current : 30mA.
* Integrated antenna and an edge connector.
* Range about 10 meters.
* Configureble in both master and slave modes.
* Pins : STATE, RXD, TXD, GND, VCC, KEY/ENABLE.

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:

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
void setup () {
 Serial . begin (9600);   // serial data
// transmission
 pinMode (13, OUTPUT);
}
 void loop() {
  if (Serial . available () > 0) {
   data = Serial . read();
   Serial . print (data);
   Serial . print ("\n");
   if (data = = '1') {
   digitalWrite (13, HIGH); }
   else if (data = = '0')
   digitalWrite (13, LOW);
 }
}
```

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.

Ex. no: 2. OBSTACLE DETECTION AND ALERT SYSTEM USING INFRARED SENSOR AND ARDUINO.

OBJECTIVE :

To develop a simple application which detects obstacle and gives alert using Infrared Sensor and Arduino.

REQUIRED COMPONENTS :

* Arduino UNO
* Arduino IDE
* IR Sensor.
* Breadboard.
* USB cable for uploading code into Arduino UNO.
* Jumper wires.

BACKGROUND THEORY :

An infrared proximity sensor or IR Sensor is an electronic device that emits infrared lights to sense some aspect of the surroundings and can be employed to detect the motion of an object. As this is a passive sensor, it can only measure infrared radiation. This sensor is very common in the electronic industry to design an obstacle avoidance robot or any other proximity detection-based system.

IR Sensor Pinout :

The IR sensor has a 3-pin connector that interfaces it to the outside world. The connections are as follows :

VCC -> The power supply pin for the IR sensor which we connect to the 5V pin on the Arduino.

OUT -> a 5V TTL logic output pin. LOW indicates no motion is detected ; HIGH mean motion is detected.

GND -> should be connected to the ground of the Arduino.

WORKING :

The working of the IR sensor module is very simple, it consists of two main components : the first is the IR transmitter section and the second is the IR receiver section. In the transmitter section, IR led is used and in the receiver section, a photodiode is used to receive infrared signal and after some signal processing and conditioning, the output will be obtained.

An IR proximity sensor works by applying a voltage to the onboard Infrared light Emitting Diode which in turn emits infrared light. This light propagates through the air and hits an object, after that the light gets reflected in the photodiode sensor. If the object is close, the reflected light will be stronger, If the object is far away, the reflected light will be weaker. If you look closely toward the module. When the sensor becomes active it sends a corresponding Low signal through the output pin that can be sensed by an Arduino or any kind of microcontroller. to execute a particular task. The one cool thing about this module is that it has two onboard LED's built-in, one of which lights on when power is available and another one turns on when the circuit gets triggered.

IR MOTION SENSOR MODULE - Parts.

IR sensor is used to detect proximity or to build obstacle avoidance robots. This sensor is popular among beginners as these are low power, low cost, rugged, and feature a wide sensing range that can be trimmed down to adjust the sensitivity.

This sensor has three pins two of which are power pins leveled VCC and GND and the other one is the sense / data pin. It has an onboard power LED and a signal LED the power LED turns on when power is applied to the board the signal LED turns on when the circuit is triggered. This board also has a comparator op-amp that is responsible for converting the incoming analog signal from the photodiode to a digital signal. It also have a sensitivity adjustment potentiometer; with that, the sensitivity of the device can be adjusted. Last and finally, it has the photodiode and IR emitting LED pair which all together make the total IR proximity sensor module.

CONNECTION DIAGRAM :

CODING :

```
// Arduino IR sensor Code.
int IRSensor =9;
int LED = 13;
void setup() {
  Serial.begin (115200);
  Serial.println ("Serial Working");
  pinMode (IRSensor, INPUT);
  pinMode (LED, OUTPUT);
}
void loop() {
   int SensorStatus = digitalRead (IRsensor);
  if (sensorStatus == 1)
  {
    digitalWrite (LED, LOW);
    Serial.println ("Motion Ended!");
  }
  else
  {
    digitalWrite (LED, HIGH);
    Serial.println ("Motion Detected!");
  }
}
```

EXP. NO: 4

DATE:

OBJECTIVE:

To control LEDs, relay & buzzer using Blynk app.

REQUIRED COMPONENTS:

* Arduino Uno x 1
* HC-05 Bluetooth module x 1
* Jumper wires x 6-10
* Buzzer x 1
* Led x 1

BACKGROUND THEORY:

Blynk is a user friendly Internet of Things (IoT) platform that empowers developers to effortlessly create custom mobile applications for controlling and monitoring IoT devices. Central to Blynk is its intuitive mobile app, available on both Android and iOS, which features a drag and drop interface for easily designing graphical user interfaces. Compatible with a variety of popular microcontrollers like Arduino and ESP8266, Blynk facilitates cloud connectivity, allowing users to remotely interact with their IoT projects. The platform offers a range of widgets for diverse functionality, and its energy system ensures efficient resource management. With API integration and an active community, Blynk serves as a versatile and accessible tool for IoT enthusiasts and developers.

PROCEDURE :

Step 1 : Circuit Diagram connections.

Step 2 :

HC 05 Bluetooth module.

Module Arduino Vcc 5V

GND GND

Tx Digital pin. Rx Digital pin 11.

- Buzzer

- LED

module Arduino

+ve Digital pin 3

-ve Gnd

Step 3 : Blink App.

1) Go to Play Store from your android phone and download and install Blynk App.

2) Open Blynk App.

3) Sign up / Register.

4) Now click on Create new project.

5) Project name - "Give your project a name".

6) choose device - "Arduino Uno".

7) Connection Type - "Bluetooth".

8) Now click "Create".

9) As soon as you click "create", a mail called "dispatcher" is sent to you by blynk.

10) Open mail and copy the "Auth token".

11) Now visit blynk website.

12) Now there you will find "Download blynk library". Now click on it follow the instructions and download blynk library.

13) Now extract the downloaded library and copy it to C:\Program Files(x86)\Arduino\libraries.

14) Now again visit blynk website.

1. Under "Flash" you will find "Sketch Builder", click on "Sketch Builder", On left side you will find.

- Board = Arduino.
- Connection = HC05/HC06
- Example = Getting Started / BlynkBlink

2. Now copy example and paste it to the Arduino IDE.

3. Now paste the "Auth token" (mailed by blynk" in the place of "Your Auth" and remove jumpers from pin 10 and pin 11 from Arduino and upload the code to board.

Step 4: Creating Interface in App.

- Click on "add widget" (+) and select a button.
- now click on button.
- Give name to button say "led".
- under OUTPUT tab.

→ Click pin and select the pin to which led is connected arduino, here it is digital pin 13, hence select digital and under pin D13. And click continue.

- under MODE tab...

→ Select whether you want this button as "push button" or switch.

- Click back.
- Click on "add widget" (+) and select "slider".
- click on "slider".

→ Name the slider say "buzzer"

→ under OUTPUT tab...

Select the pin no to which buzzer is connected to arduino, here it is digital pin D3. Click "continue".

→ under SEND ON RELEASE tab... set it off

→ click back.

- Click on "add widget" (+) and select "Bluetooth".
- Now close the app.
- Now power your Arduino.
- Turn on Bluetooth of your phone and search for "HC-05", now pair the device with default key "1234".
- After successful pairing. Open Blynk APP, select the project you have created select bluetooth.

tap on connect "bluetooth device" here you should find "HC-05" select it.

now you should see # HC-05 connected. and now hit back.

now in right most corner you should see "play" button adjacent to "Add widget", hit "play".

now press led it should turn on led and move the slider accordingly buzzer should sound.

INFERENCE: