



Experiment 3.2

Student Name: Nabha Varshney

Branch: B.E.-CSE

Semester: 6th

Subject Name: Internet of Things Lab

UID: 20BCS4995

Section/Group: 20BCS704-A

Date of Performance: 02/05/2023

Subject Code: 20CSP-358

1. Aim:

Real Time application of controlling actuators through Bluetooth application using Arduino.

2. Objective:

- Learn about interfacing.
- Learn about IoT programming.

3. Requirements:

8 Male/Male Jumper Wires

1 HC-05 Bluetooth Module

1 (5 mm) LED: Red

1 Arduino UNO

1 Resistor 1k ohm

Apps and platforms

1 Arduino IDE

1 MIT App Inventor

4. Procedure:

About the interfacing of Arduino & Bluetooth

Let's get started right away with the Arduino Bluetooth Tutorial, where I'll provide all you need to know about pairing Arduino with Bluetooth. Ranging from hardware configurations to programming and using it with your iPhone or Android devices!

1. Arduino Bluetooth:



Grove-Blueseed-Dual model(HM13)

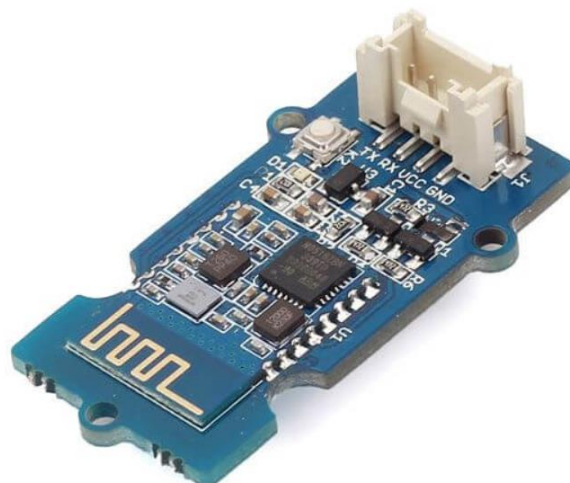
Flexible and packed with high Bluetooth transmission speed, the Grove – Blueseed LE – Dual Model (HM13) uses a CSR dual-mode Bluetooth chip, with the ARM architecture single chip that supports AT instructions.

This allows users like yourself to have control over the serial baud rate, equipment name, and pairing password.

You may be wondering; why not the HC-06 or HC-05 Arduino Bluetooth module instead? Well, with the Grove – Blueseed – Dual Model (HM13), you're getting the following key advantages:

Bluetooth 4.0 BLE rather than Bluetooth V2.0

Grove – Base Shield



Hardware configurations:

Step 1: Connect the Grove – Blueseed – Dual model (HM13) to a Grove port on the Grove – Base Shield via the Grove cable

Step 2: Plug Grove – Base Shield into your Arduino board

Step 3: Connect your Arduino to PC via USB cable

Understanding the software:

Conventions

In EDR mode, only the slave can be configured while either master or slave can be in BLE mode.

Factory default setting:EDR Name HMSoft, Slave role, PinCode 1234

BLE Name HMSoft, Slave role, PinCode 000000

Baud: 115200, N, 8, 1;

AT Command format:Uppercase AT command format. string format, without any other symbol. (e.g. \r or \n).

Any incorrect command would get no response.

After understanding the software configurations, here's how to configure Bluetooth with a PC. For hardware connection, do refer to the "Hardware configurations" section. You'll find that the flashing blue LED on the module illustrates no connection is set up

Step 1: Open a serial terminal and set Baud Rate: 115200, Databits: 8, Stopbits: 1, and no flow control like above

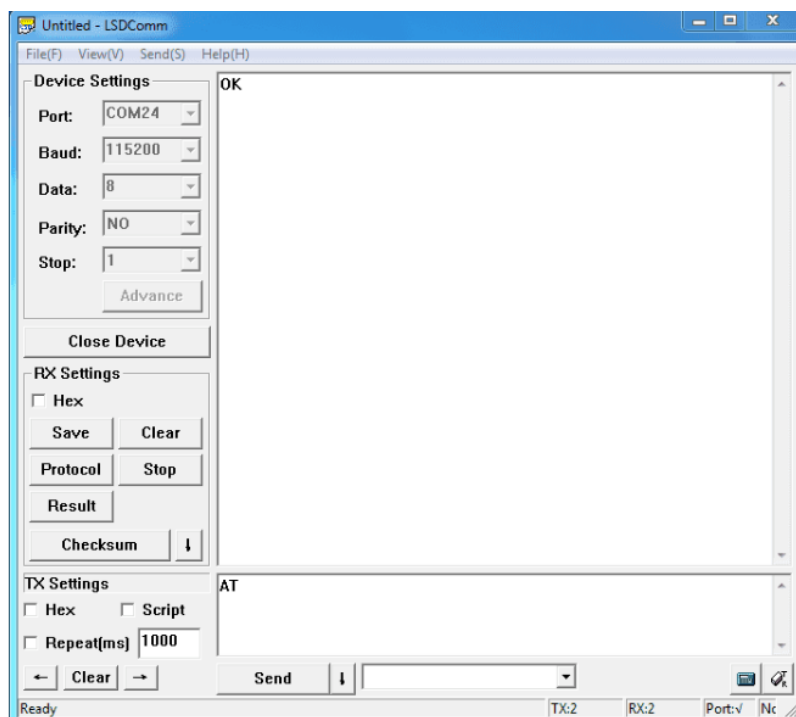
Step 2: Send "AT" to Bluetooth with the serial terminal to check if you receive an "OK"

The Bluetooth only respond AT commands either when:No connection is set up

All commands were seen as string and sent out

*You can distinguish the above status in step 2 through LED indications.

It should look something like this after the above steps:



Here are some useful configurations that can be sent:

Test serial connection, send "AT", will return "OK".

Restore factory settings, send "AT+RENEW", return "OK+RENEW".

Reset baud rate of serial port, send "AT+BAUD2", return "OK+Set:2".

Enable authentication, send "AT+AUTH1", return "OK+Set:1".

Reset the Bluetooth, send "AT+RESET", return "OK+RESET".

5. Code/Program-

```
#include<SoftwareSerial.h>
SoftwareSerial bluetoothSerial(10, 11);

const int led1=2;
const int led2=3;
const int led3=4;
String inputString;
String buttonId;
int buttonState=0;
int pinNum=0;

void setup(){
    //set the digital pins
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
    //start communication with bluetooth module
    bluetoothSerial.begin(9600);
    //set all pins to off as default to start with
    digitalWrite(led1,LOW);
    digitalWrite(led2,LOW);
    digitalWrite(led3,LOW);
}

<p>void loop(){<br>  if( bluetoothSerial.available()){
    while( bluetoothSerial.available()){
        char inChar = (char) bluetoothSerial.read(); //read the input
        inputString += inChar;          //make a string of the characters coming in serial

    }</p><p style="margin-left: 20px;">          //if it ends with "/"# it is the end of the
command</p><p>  if(inputString.endsWith("/#")){
        inputString = inputString.substring(0,inputString.length()-2); //get rid of the "/"#
at the end

        for(int i=0;i<inputString.length();i++){          //split to get id and state</p><p>
if(inputString.charAt(i)==' '){
            buttonId = inputString.substring(0,i); //get the id of the button</p><p>
style="margin-left: 20px;">          //get the state of the button ,if its 0 switch it off and if
its 1 turn it on</p><p style="margin-left: 20px;">          buttonState =
inputString.substring(i+1).toInt();
        }
    }

    inputString = ""; //reset the input data for the next time
```

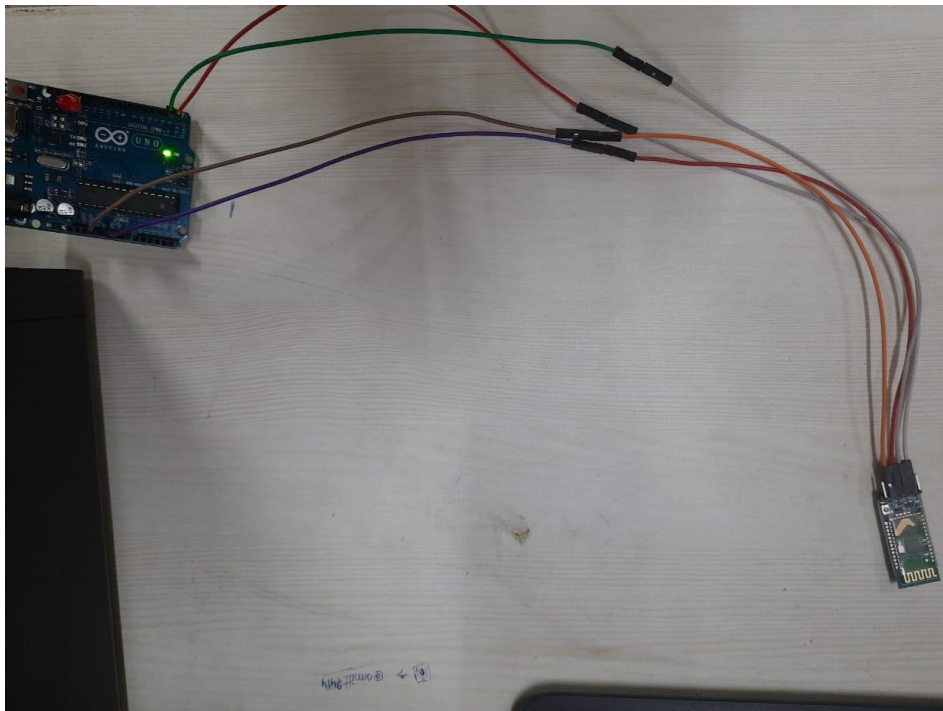
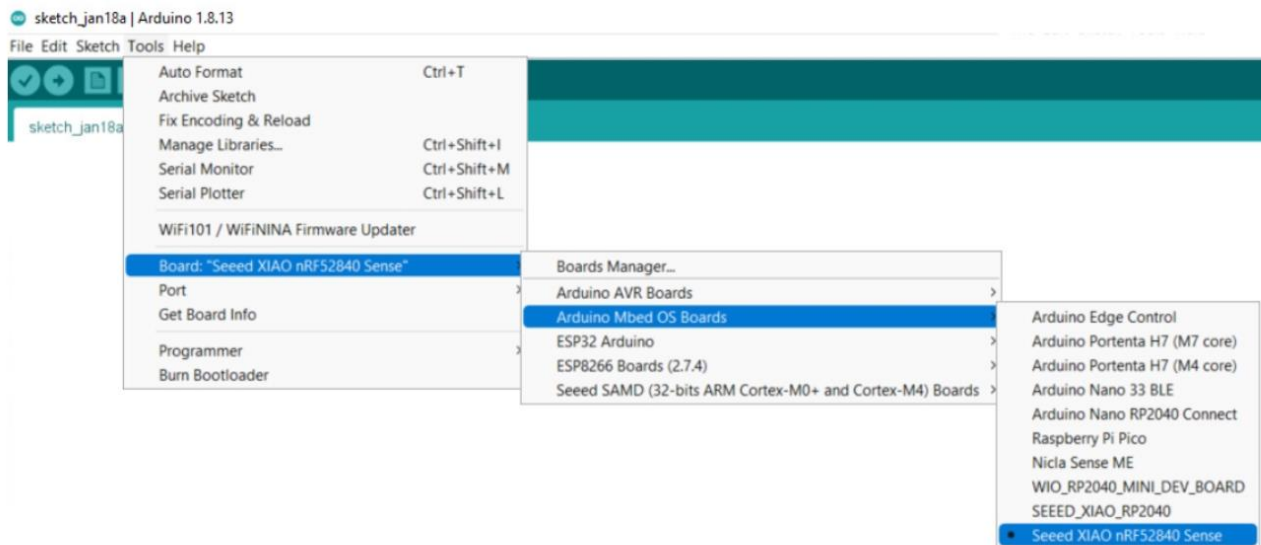
```

} //get pin number and then convert the string to an
int pinNum = buttonId.substring(1).toInt();
digitalWrite( pinNum,buttonState);

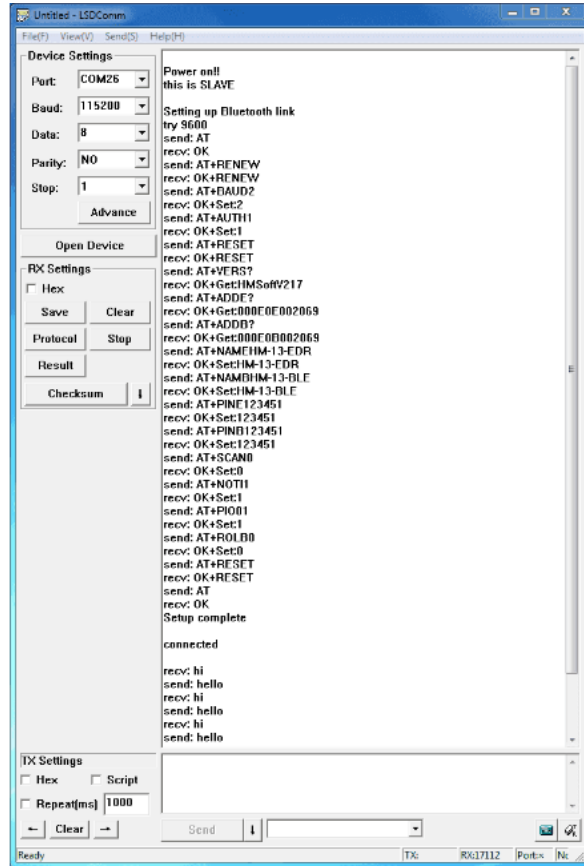
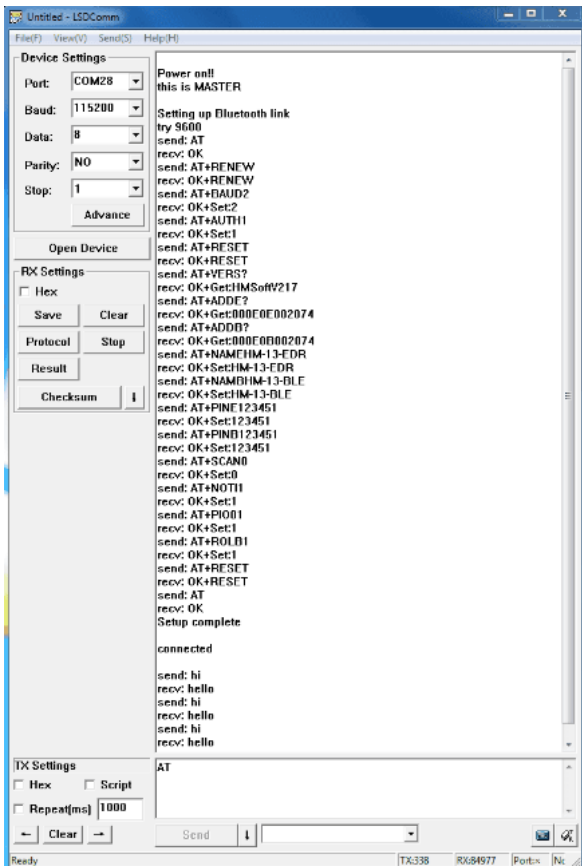
}

```

6. Output



Circuit of the experiment



Result-

In this experiment we learn how to display data generated by Real Time application of controlling actuators through Bluetooth application using Arduino. We learn how to connect Bluetooth with Arduino and how to access different features of Arduino using Bluetooth and use Grove-Blueseed-Dual model(HM13).

Learning Outcomes(What I have learnt):

- Learnt how to use Arduino with Bluetooth.
- Leant how to use the software using Grove-Blueseed-Dual model(HM13).
- Learnt how to do code on Arduino board and use different sensors and actuators.