Experiment No. - 9

Student Name: Vivek Kumar
Branch: BE-CSE(LEET)
Semester: 6th
Subject Name: IOT Lab

UID: 21BCS8129
Section/Group:20BCS-ST-801/B
Date of Performance: 05/05/2023
Subject Code: 20CSP-358

1. **Aim:**
   Real Time application of controlling actuators through Bluetooth application using Arduino.

2. **Objective:**
   a. Learn about Gas sensor in detail.
   b. LED and LCD
   c. Learn about IoT programming.
   d. Measure the pollution in the air.

3. **Requirements:**
   a. Arduino Uno R3 board
   b. 3 LED
   c. 3 220Ω registers
   d. USB or 5V Power Supply
   e. Solder less breadboard
   f. Jumper Wires
   g. HC-05 Bluetooth module
   h. Most importantly your phone and a downloaded Bluetooth app (Arduino Bluetooth Controller, which offers many different features)

4. **Procedure:**

Bluetooth module connection:

- Connect the BT module's Rx pin to pin 11 on the Arduino
- Connect the BT module's Tx pin to pin 10 on the Arduino
- Connect up the Gnd and Vcc (5v) to the Arduino

Led's connection:

- Connect all the cathodes (short pin) of the led to Gnd
- Connect each anode to a 220Ω resistor
- Connect a resistor to Arduino pin 2,3 and 4

Flexible and packed with high Bluetooth transmission speed, the Grove – Blueseeed LE – Dual Model (HM13) uses a CSR dual-mode Bluetooth chip, with the ARM architecture single chip that supports AT instructions.
If the led on the Bluetooth Module is blinking quickly then it is ready to pair to your phone, if not then check your connections.

Hardware configurations:

Step 1: Connect the Grove – Blueseeed – 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.

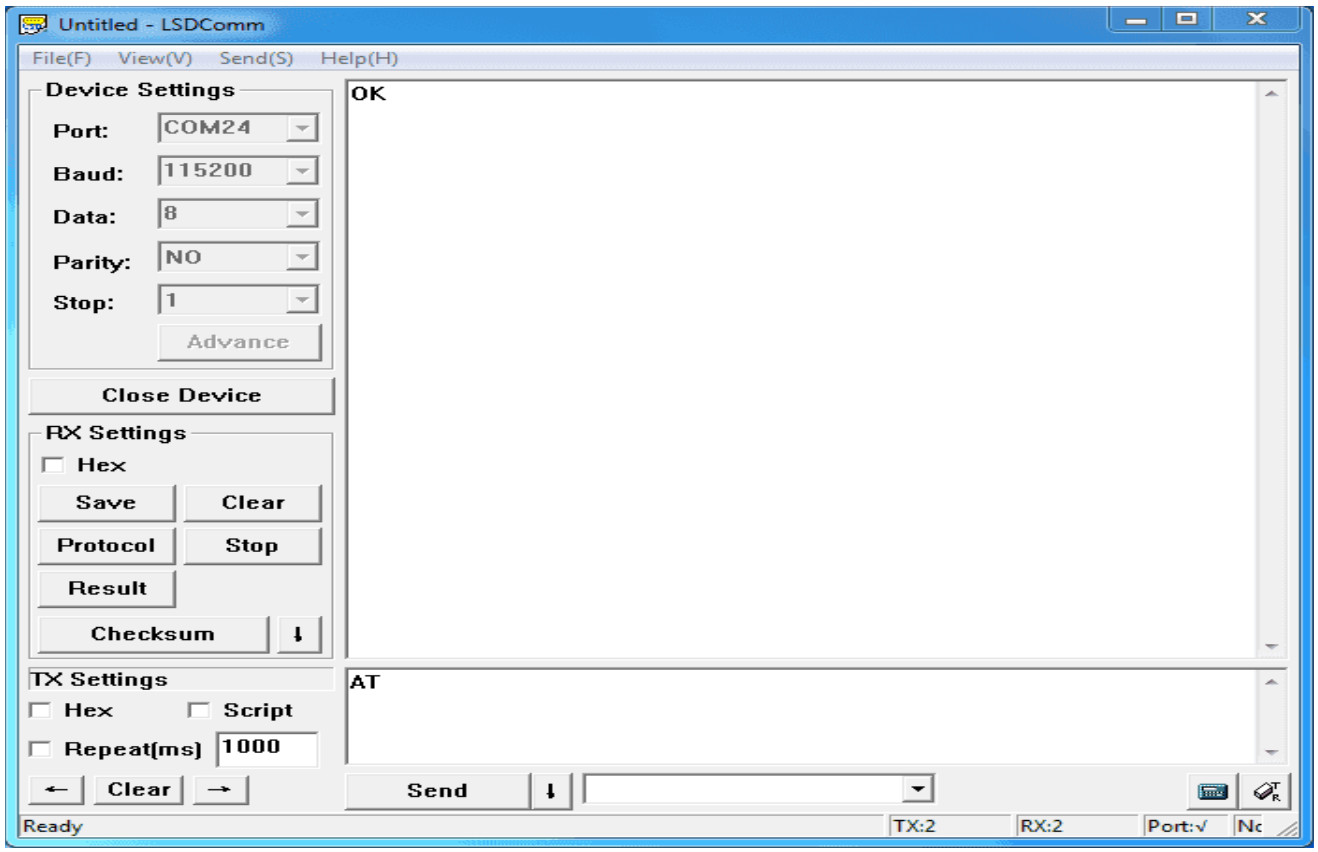
How to program on the Arduino Bluetooth Module

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.



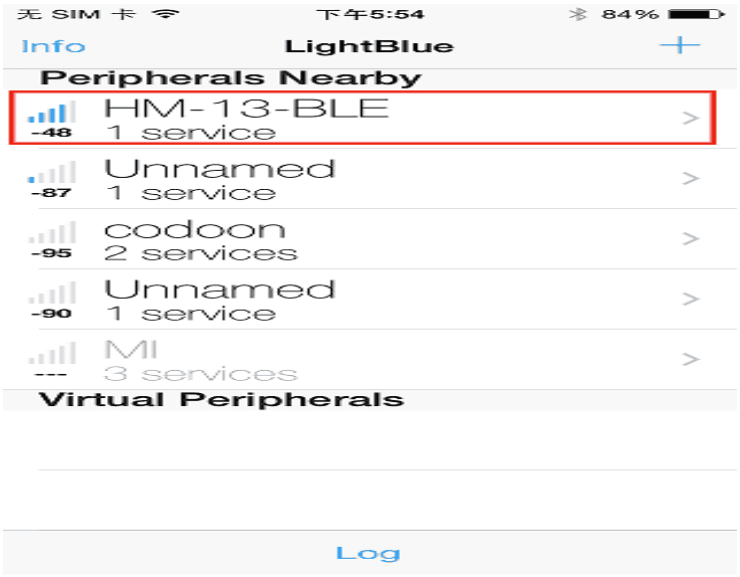
Here are some useful configurations that can be sent:
Test serial connection, send “AT”, will return “OK”.
Restore factory settings, send “AT+RENEW”, return “OK+RENEW”.
Reset baud rate of serial port, send “AT+BAUD2”, return “OK+Set:2”.
Enable authentication, send “AT+AUTH1”, return “OK+Set:1”.
Reset the Bluetooth, send “AT+RESET”, return ”OK+RESET”.
Query firmware version, send “AT+VERS?”, return “OK+Get:HMSoftV217”.
Query MAC of EDR, send “AT+ADDE?”, return “OK+Get:000E0E002074”.
Query MAC of BLE, send “AT+ADDB?”, return “OK+Get:000E0B002074”.
Set the name of EDR, send “AT+NAMEHM-13-EDR”, return “OK+Set:HM-13-EDR”.
Set the name of BLE, send “AT+NAMEHM-13-BLE”, return “OK+Set:HM-13-BLE”.
Set the password of EDR, send “AT+PINE123451”, return “OK+Set:123451”.
Set the password of BLE, send “AT+PINB123451”, return “OK+Set:123451”.
Enable discovery and connectable, send “AT+SCAN0”, return “OK+Set:0”.
Enable notify information of connection, send “AT+NOTI1”, return “OK+Set:1”.
Notify information include address, send “AT+NOTP1”, return “OK+Set:1”.
Enable user key, send “AT+PIO01”, return “OK+Set:1”.
Set to Central mode, send “AT+ROLB1”, return “AT+ROLB1”.
Set to Peripheral mode, send “AT+ROLB0”, return “AT+ROLB0”.

We used two Bluetooth that were connected with the PC, with one set as central while the other as Peripheral. Several seconds later, they find each other, and the LED stops flashing connected!

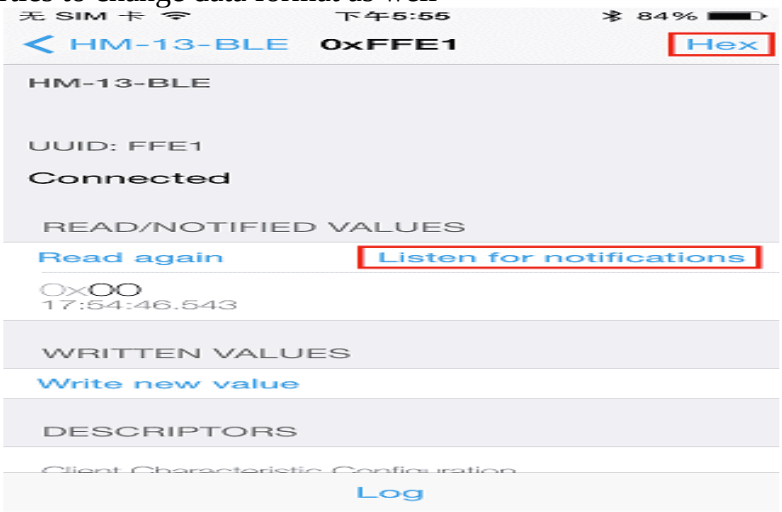
How to pair Arduino Bluetooth Module with iPhone and Andriod
Since the Grove – Blueseeed – Dual model (HM13) have two protocol: Bluetooth EDR (Enhanced Data Rate) and Bluetooth Low Energy (BLE), it can communicate with either Andriod or iPhones!
For this part of the tutorial, we’ll use an iPhone to demonstrate how you can interact with Bluetooth!

*Note: The tutorial below is run on an older version of ios but it should still work the same

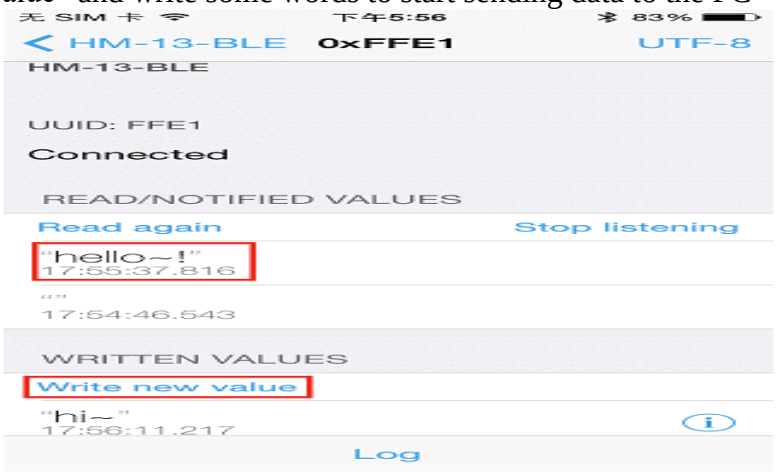
Step 1: Power the Bluetooth and configure it as a Peripheral role
Step 2: Search Light Blue in the App Store and install it
Step 3: Launch the app, and connect to “HM-13-BLE”



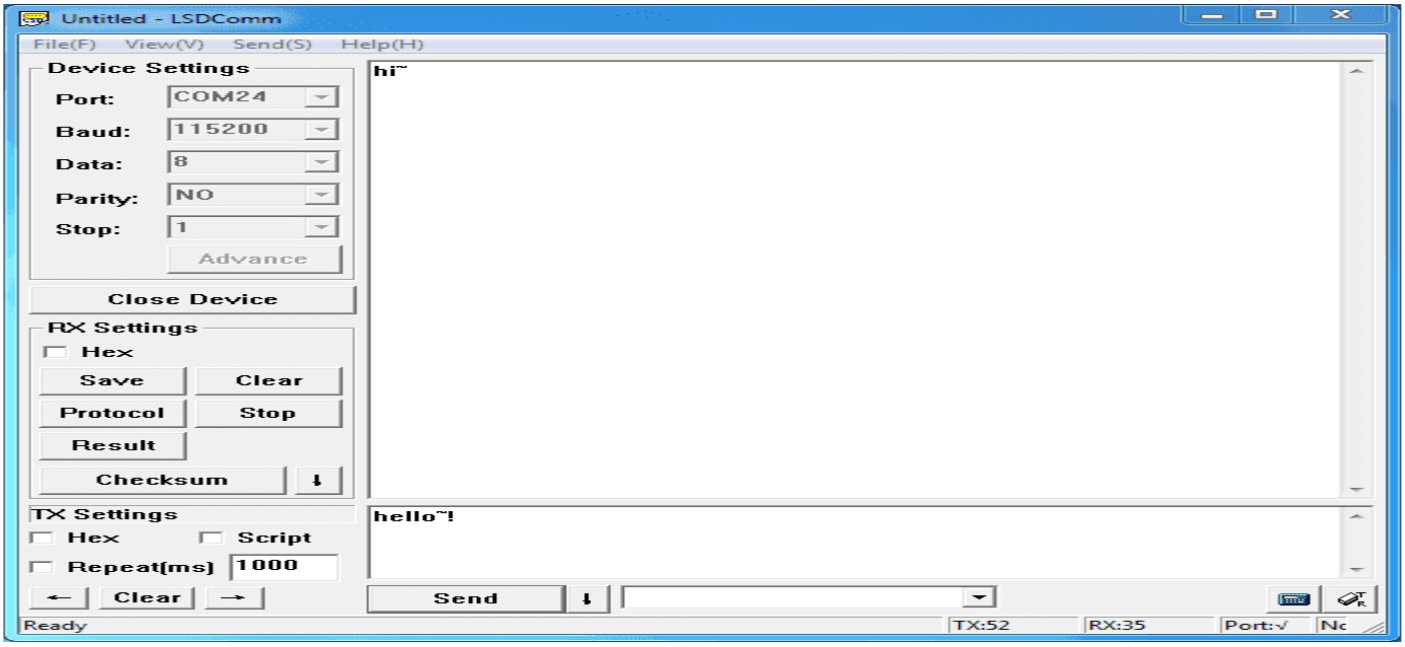
Step 4: Touch on properties and hit "listen for notifications" to enable data receiving There's a "Hex" key on the top right under properties to change data format as well



Step 5: Hit "Write new value" and write some words to start sending data to the PC



With the serial terminal, you can transfer data from the PC to iPhone as well:



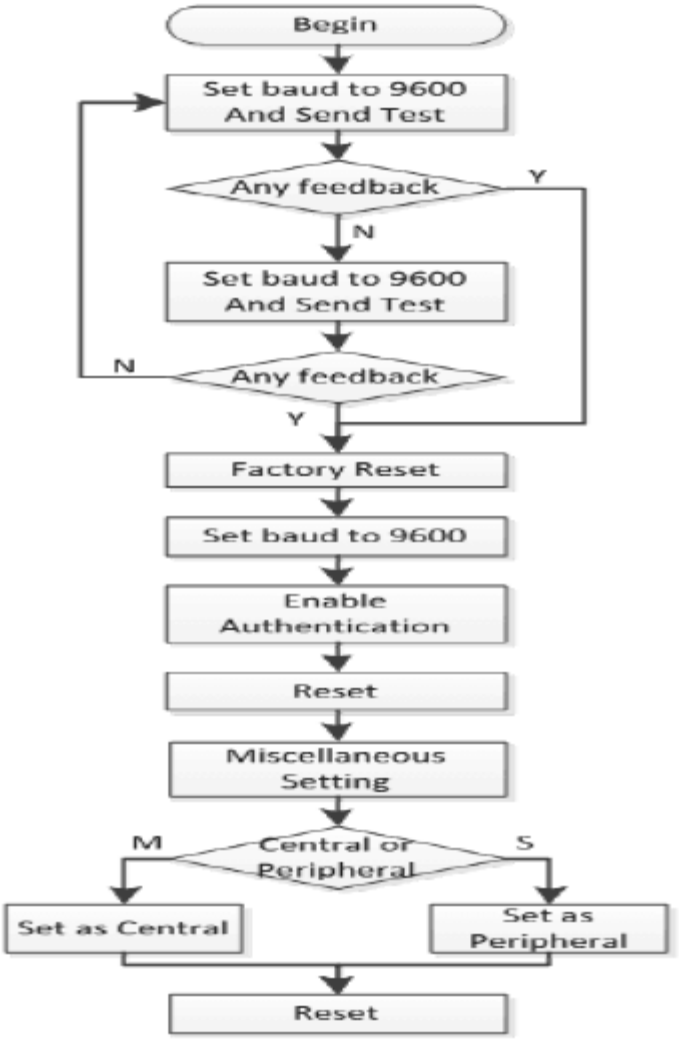
Bluetooth Data transmission guide between two Arduino boards

Now after all the above steps, are you ready to code? In this final section, we'll use two Arduino Uno and a pair of Bluetooth modules to get started!

Step 1: Set up the connection mentioned in the hardware configurations section

Step 2: Assign the Bluetooth to the Central role by modifying the text to "#define MASTER 1" The program of Central and Peripheral use the same code but there's a difference in the micro define at the beginning of the program

Step 3: Follow the flow chart below for initialization of the program

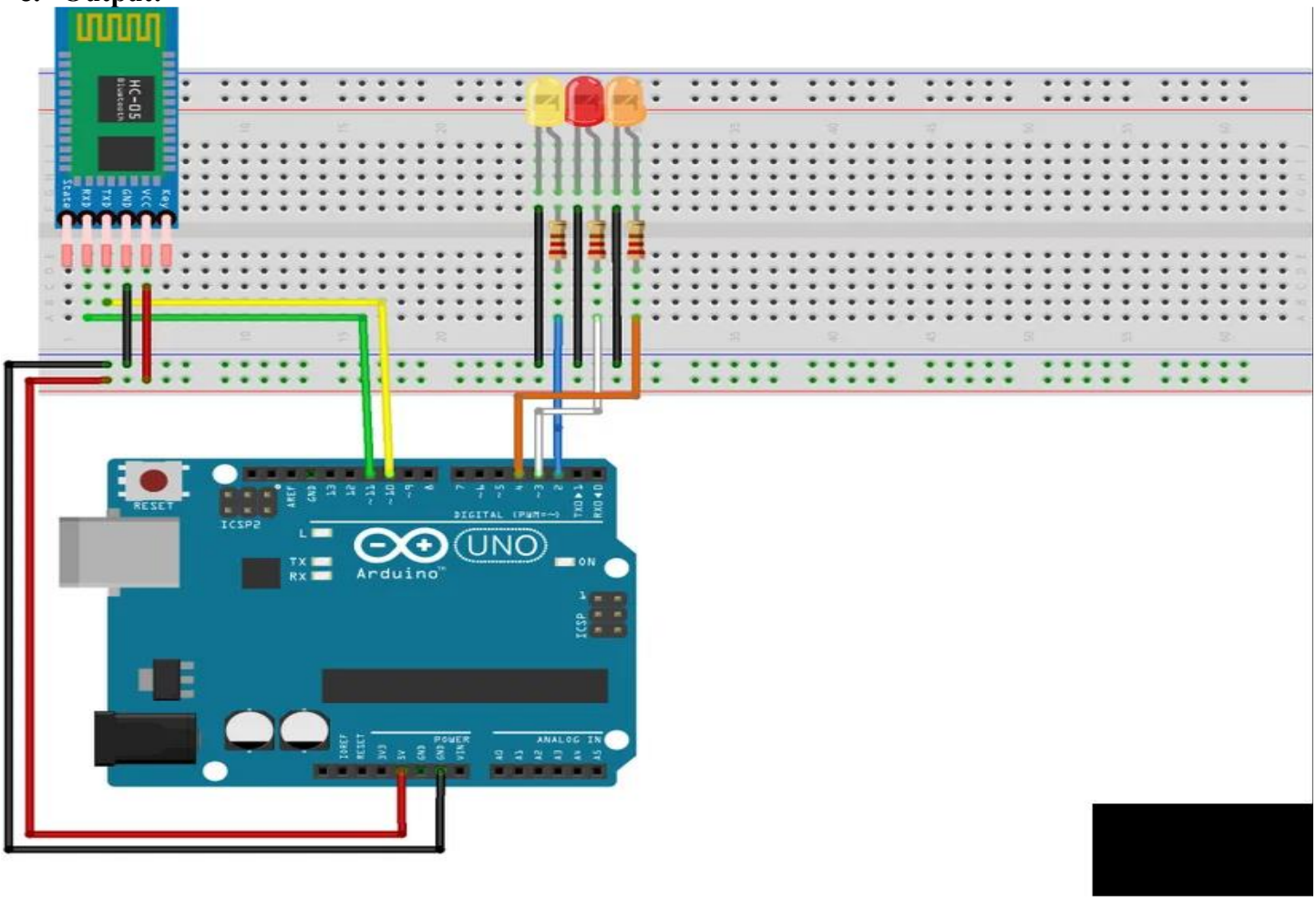


5. Steps/Program:

```
const int LED = 5;
char switchstate;
void  setup() {
   //Here the code only runs once.
   Serial.begin(9600);
   pinMode(LED, OUTPUT);
}
void loop() {
//This code repeats. This  is  our  main  code.
   while(Serial.available()>0){
      //code to be executed only when Serial.available()>0
      switchstate = Serial.read();
      Serial.print(switchstate);
      Serial.print("\ ");
      delay(15);
```

```
        if(switchstate == '1'){
            //Checking if the value from app is '1'
            digitalWrite(5, HIGH);
        } else if(switchstate == '0'){
            //Else, if the vaue from app is '0',
            digitalWrite(5, LOW);
            //Write the component on pin 5(LED) low.
        }
    }
}
```

6. Output:



Learning outcomes (What I have learnt):

- Learnt about HC-05 Bluetooth Module
- Learnt about IoT programming.
- Controlled the device using Mobile in Lab.

Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):

Sr. No.	Parameters	Marks Obtained	Maximum Marks

1.	Worksheet completion including writing learning objectives/Outcomes. (To be submitted at the end of the day).		
2.	Post-Lab Quiz Result.		
3.	Student Engagement in Simulation/Demonstration/Performance and Controls/Pre-Lab Questions.		
	Signature of Faculty (with Date):	Total Marks Obtained:	