**“Experiment 3.2”**

**Aim:**

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

**Objective:**

1. Learn about interfacing.
2. Learn about IOT programming.

**Components Required:**

1. 8 Male/Male Jumper Wires
2. 1 HC-05 Bluetooth Module
3. 1 (5 mm) LED: Red
4. 1 Arduino UNO
5. 1 Resistor
6. 1k ohm

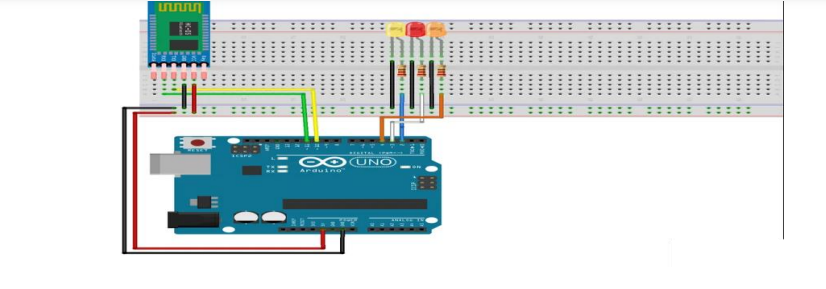
**Apps and platforms:**

1. Arduino IDE
2. MIT App Inventor

**Step 1 Here is what you need to control Led's with Bluetooth:**

* Arduino
* HC-05 Bluetooth module
* Solder less breadboard
* 3 Led's
* 3 220Ω resistors
* Wires
* Most importantly your phone and a downloaded Bluetooth app (Arduino Bluetooth Controller, which offers many different features)

**Step 2: Circuit**

****

**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

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

**Code:**

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.

}

}

}

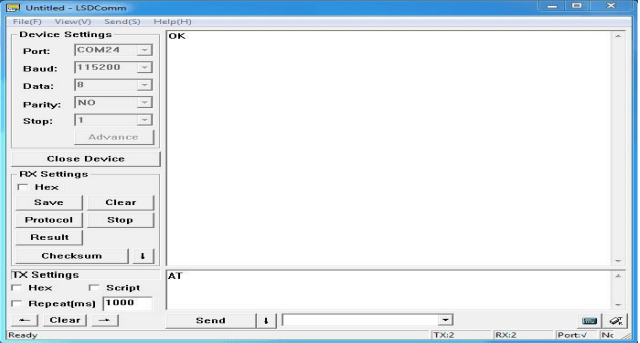
1. **How to program on the Arduino Bluetooth Module**

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

****

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!

1. **How to pair Arduino Bluetooth Module with iPhone and Android**

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!

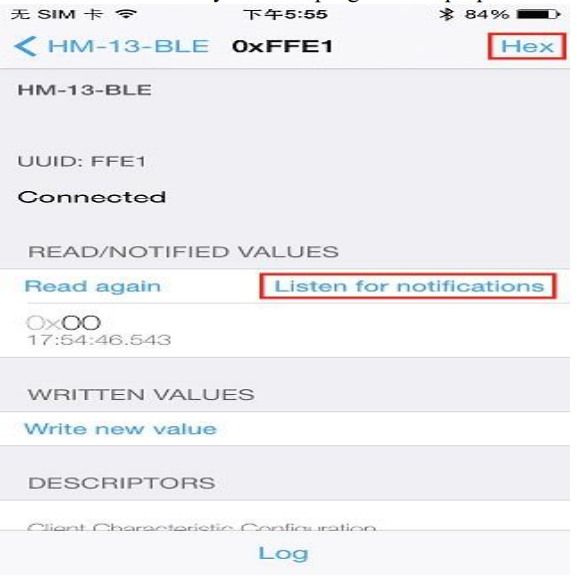
**Note:** The tutorial below is run on an older version of ios but it should still work the same

**Step 1:** Power the Bluetooth and configure it as a Peripheral role

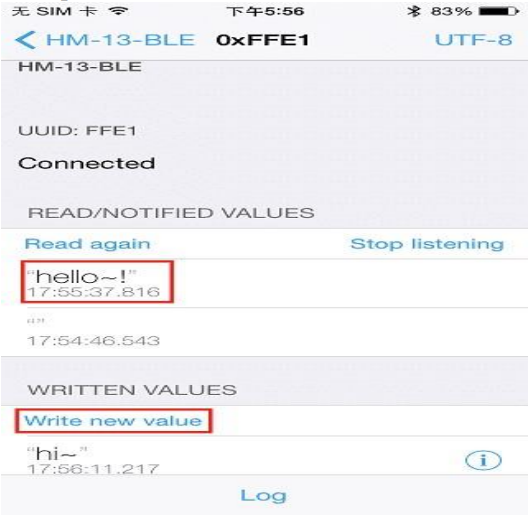
**Step 2:** Search Light Blue in the App Store and install it

**Step 3:** Launch the app, and connect to “HM-13-BLE”

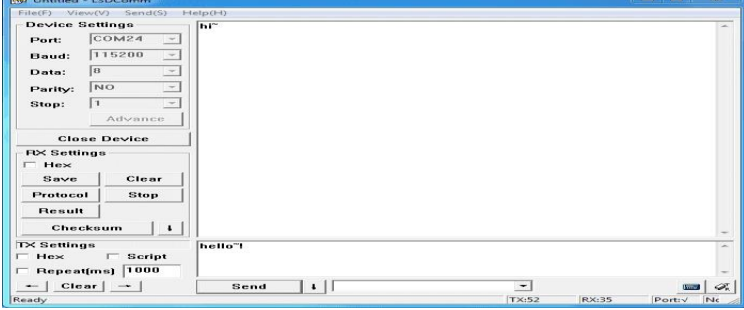
**Step 4:** Touch on properties and hit “listen for notifications” to enable data receiving There’s a “Hex” key on the top right under properties to change data format as well

****

**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:

****

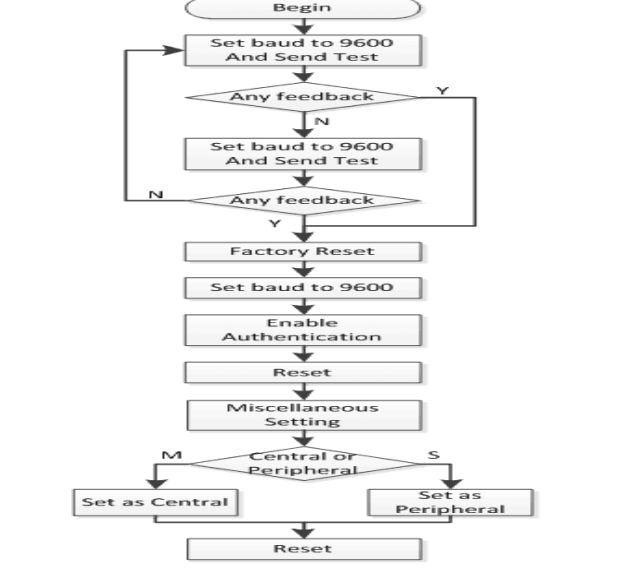
1. **Bluetooth Data transmission guide between two Arduino boards**

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

****

After the initialization, the Central and Peripheral will do different things; Central will send a message to the Peripheral interval and print what’s received from the Peripheral while the Peripheral only responds to the central

**Step 4:** Download the test code and open HM-13\_SW.ino with Arduino IDE, compile and download to Arduino Uno. Remember to configure the Bluetooth to the different role by modifying the macro at the beginning

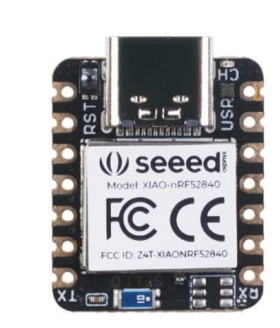
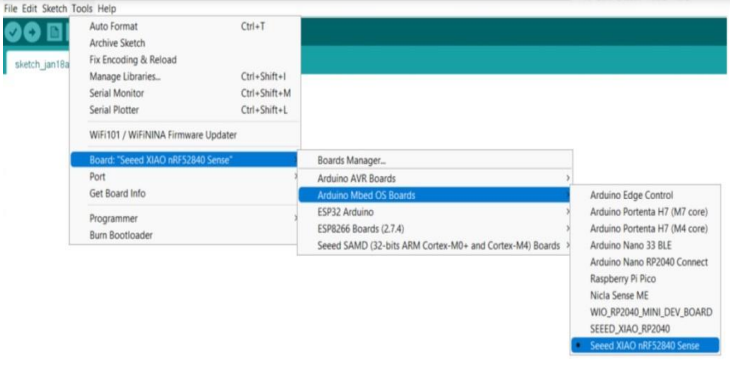
**Step 5:** After the program is downloaded, open two serial terminal windows and wait for the Bluetooth connection. A connection is indicated by: LEDs on the Bluetooth modules will flash for a few seconds, stop flashing, and kept on. According to the program written, the Central will now send a message to the Peripheral continually and get feedback every time.

**Arduino-supported MCU with Bluetooth 5.0**

As the first wireless product in the Seeed XIAO family, Seeed XIAO BLE & BLE Sense has equipped a powerful Nordic nRF52840 MCU which is designed in **a Bluetooth 5.0 module,** built around a 32-bit ARM® Cortex™-M4 CPU with Floating-Point Unit(FPU) operating at 64Mhz.

**Seeed XIAO BLE nRF52840 -**Supports Arduino / MicroPython -Bluetooth5.0 with Onboard Antenna

**Seeed XIAO BLE nRF52840 Sense** – TinyML/TensorFlow Lite- IMU / Microphone – Bluetooth5.0

** **