

## Experiment 9

Student Name: Jatin

UID: 20BCS5951

Branch: CSE

Section/Group: 605-B

Semester: 6<sup>th</sup>

Subject Name: IOT LAB

Subject Code: 20CSP358

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

Components Required:

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

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

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

Step 2: Circuit

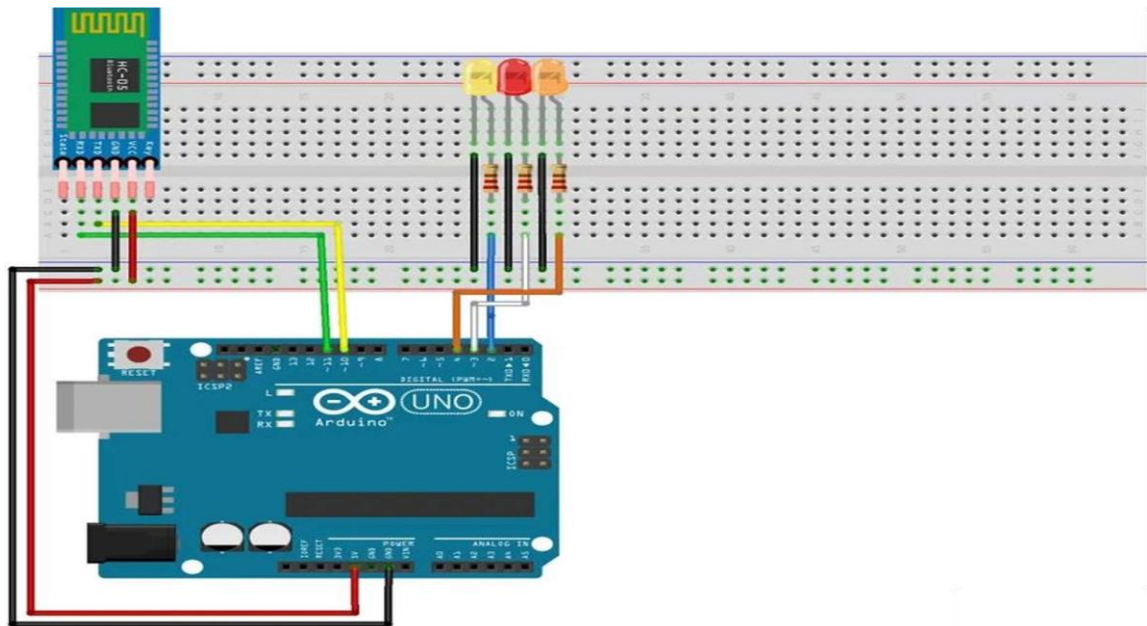


Figure: Circuit Diagram of HC05 interfacing with Arduino

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:

```
1 /*Code to control an LED via bluetooth through a phone app. 2
Project by MukeshArvindh. Code by MukeshArvindh.*/
3
4 /*If you are going to copy and paste the code, then do not forget
5 to delete the void setup() and void loop() function from the 6 sketch you are using
before doing so,as the functions already
```

```
7 exist in this sketch. Copy-pasting the comments will not 8
cause any changes in the outcome or code.*/
9
10 /*Note: This code has
been compiled and checked multiple times, and has 11 proven to be accurate. The product also
works as intended.*/ 12
13 //Bluetooth uses serial communication. So, we use many serial functions 14 //in this sketch.
15 const int LED = 5;
16 /*Declaring that there is an LED on pin 5 of the arduino board. We use 17 const as we will
not change this. You don't have to name it LED. You
18 can even put your name instead.*/
19 char switchstate;
20 /*declaring that there is a variable called switchstate, which will
21 hold a character value. This is due to programming of the app, which
22 will send a text value to arduino. If we use 'int' instead of 23 'char' the code will not work
properly.*/
24 void setup() { //Here the code only runs once.
25 Serial.begin(9600);
26 /*To start serial communication at a rate of 9600 bits per second. This 27 is the default rate
anyways.*/
28 pinMode(LED, OUTPUT);
29 //Declaring that the LED is an output.
30 }
31 void loop() { //This code repeats. This is our main code.
32 while(Serial.available() > 0) {
33 //code to be executed only when Serial.available() > 0

34 /*Serial.available() > 0 is to check if there is any reading from the
35 HC-05 Bluetooth module.*/
36 switchstate = Serial.read();
37 /*The character we had declared earlier is now being assigned a value- 38 the value of
whatever Serial.read() is.*/
39 //Serial.read() is to read the value coming from app.
40 Serial.print(switchstate);
41 //This will print the value onto the Serial monitor.
42 Serial.print("\n");
43 };
44 //This moves to the next line after every new line printed.
```

```

45 delay(15);
46 /*Gives a break of 15 milliseconds. Delay is for human eye, and for
47 speed of some computers, as some will crash at high speeds.*/
48 if(switchstate == '1'){//Checking if the value from app is '1'
49 digitalWrite(5, HIGH);
50 //If it is, write the component on pin 5(LED) high. 51 }
52 else if(switchstate == '0'){//Else, if the value from app is '0', 53 digitalWrite(5,
LOW);//Write the component on pin 5(LED) low.
54 }
55 }
56 }

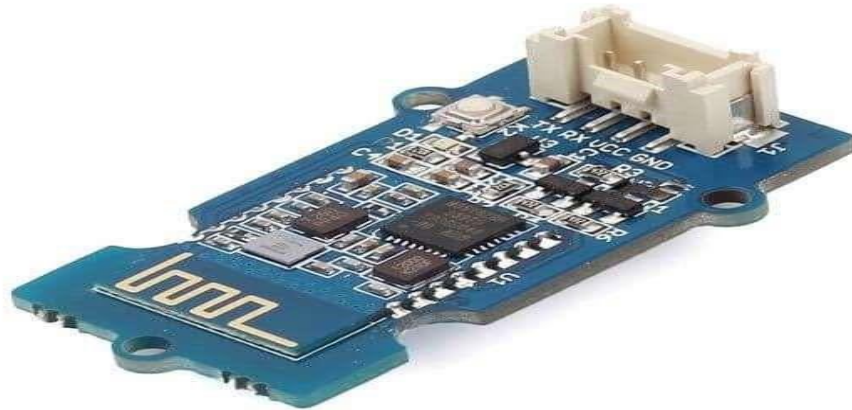
```

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:





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 to have control over the serial baud rate, equipment name, and pairing password!

## 2. 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

## 3. 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.

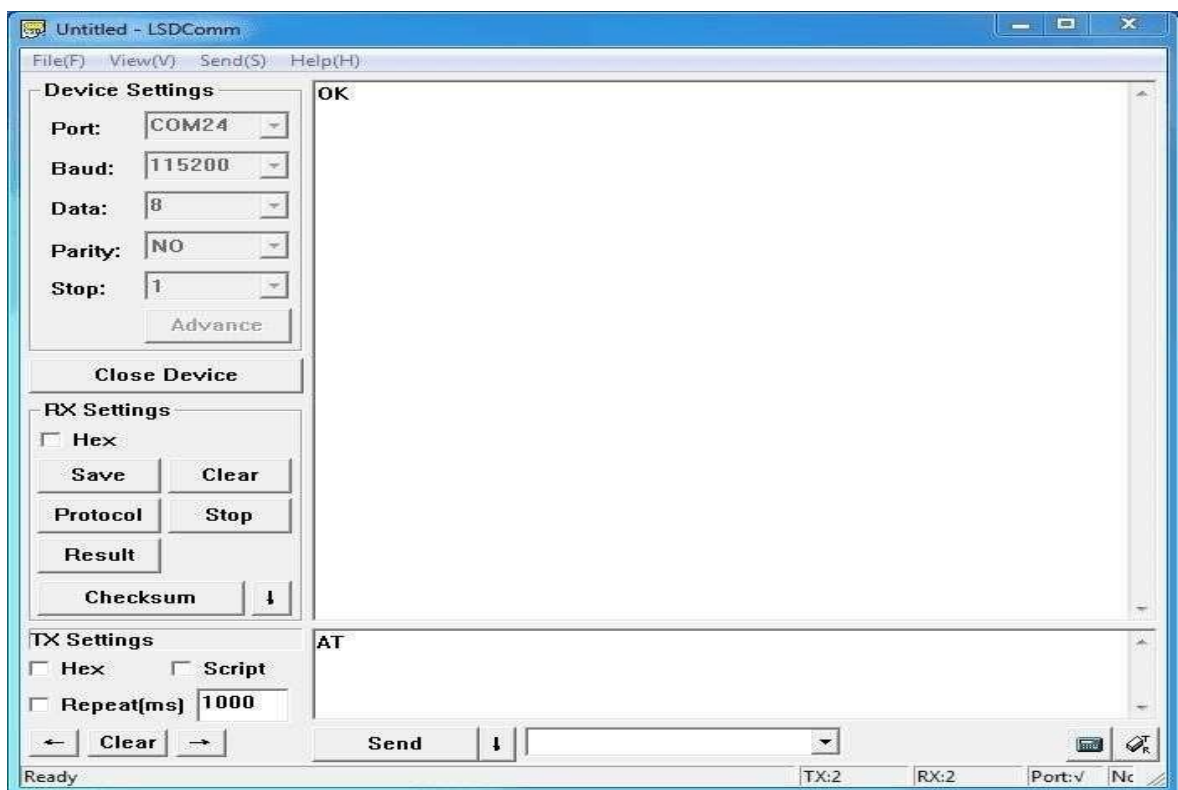
## 4. How to program on the Arduino Bluetooth Module

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!

## 5. 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 Android 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

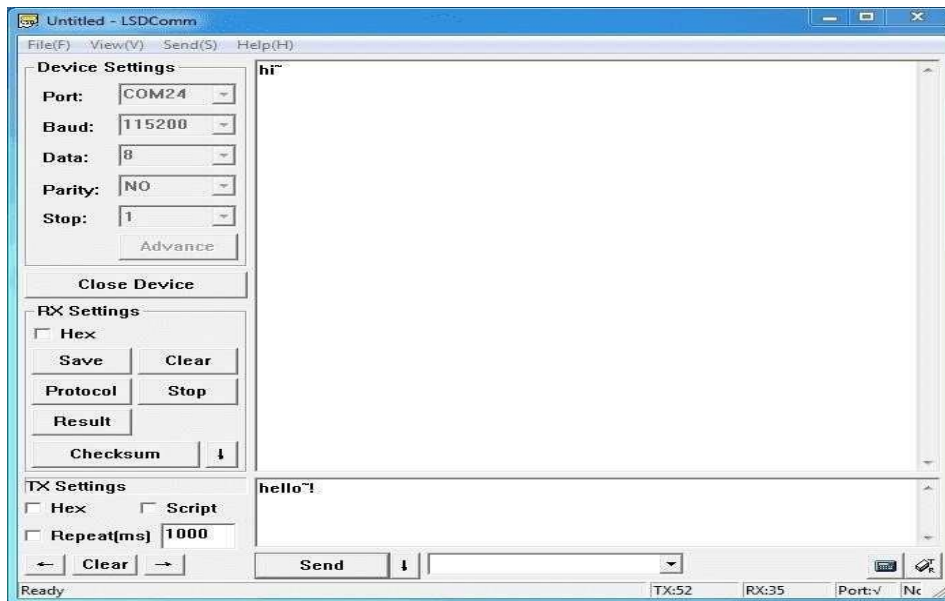


Step 5: Hit “Write new value” and write some words to start sending data to the PC



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





## 6. Bluetooth Data transmission guide between two Arduino boards

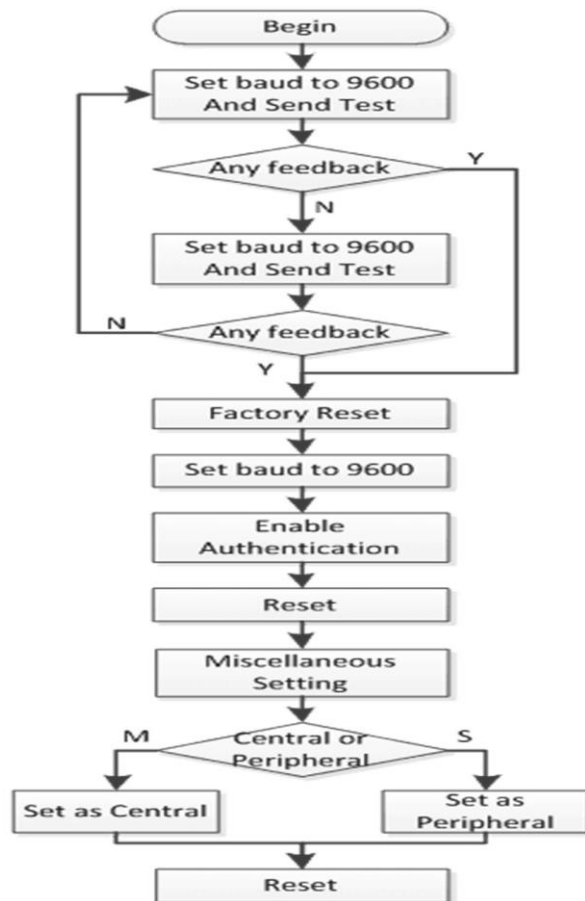
Now after all the above steps, are you ready to code? In this final section, we'll use two Arduino Uno and a pair of Bluetooth modules to get started!

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

