**Experiment 9**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** CSE (Lateral Entry)  **Section/Group:** 616/A

**Semester:** 6th **Date of Performance:** 08/05/2023

**Subject Name:** Internet of Things Lab **Subject Code:** 20CSP-358

1. **Aim:**

**Real Time application of controlling actuators through bluetooth application using Arduino.**

1. **Objective:**

* Learn about Bluetooth module in detail.
* Learn about IoT programming.
* **Real Time application of controlling actuators through bluetooth (HC-05) application using Arduino**.

1. **Requirements:**

* Arduino Uno R3 board
* HC-05
* Jumper Wires
* 1k Resistor

1. **Procedure:**
2. **Arduino Bluetooth:**

Bluetooth module is a basic circuit set of chip which integrated Bluetooth functions and which can be used in wireless network transmission. Generally, the Bluetooth module can be divided into the following types: data transmission module, remote control module, etc.

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.  
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 – Blueseeed – Dual Model (HM13), you’re getting the following key advantages:  
Bluetooth 4.0 BLE rather than Bluetooth V2.0

1. **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
2. **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.
3. **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.
4. **How to pair Arduino Bluetooth Module with iPhone and Andriod**  
   Since the [**Grove – Blueseeed – Dual model (HM13)**](https://www.seeedstudio.com/Grove-Blueseeed-Dual-model-HM13.html?utm_source=blog&utm_medium=blog) 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!  
    **Step 1:**Power the Bluetooth and configure it as a Peripheral role  
   **Step 2:** Search LightBlue 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
5. **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 initialisation, 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.

1. **Steps/Program:**

#include <SoftwareSerial.h>

SoftwareSerial SUART(10, 11); // SRX = 2, STX = 3

#define ledpin 13 //built-in L of UNO

void setup()

{

pinMode(ledpin, OUTPUT);

digitalWrite(ledpin, LOW); //L is OFF

Serial.begin(9600);

SUART.begin(9600);

}

void loop()

{

if (SUART.available())

{

char x = SUART.read();

if (x == 'Y')

{

digitalWrite(ledpin, HIGH); //L is ON

}

if (x == 'N')

{

digitalWrite(ledpin, LOW); //L is OFF

}

}

if(Serial.available())

{

char c = Serial.read();

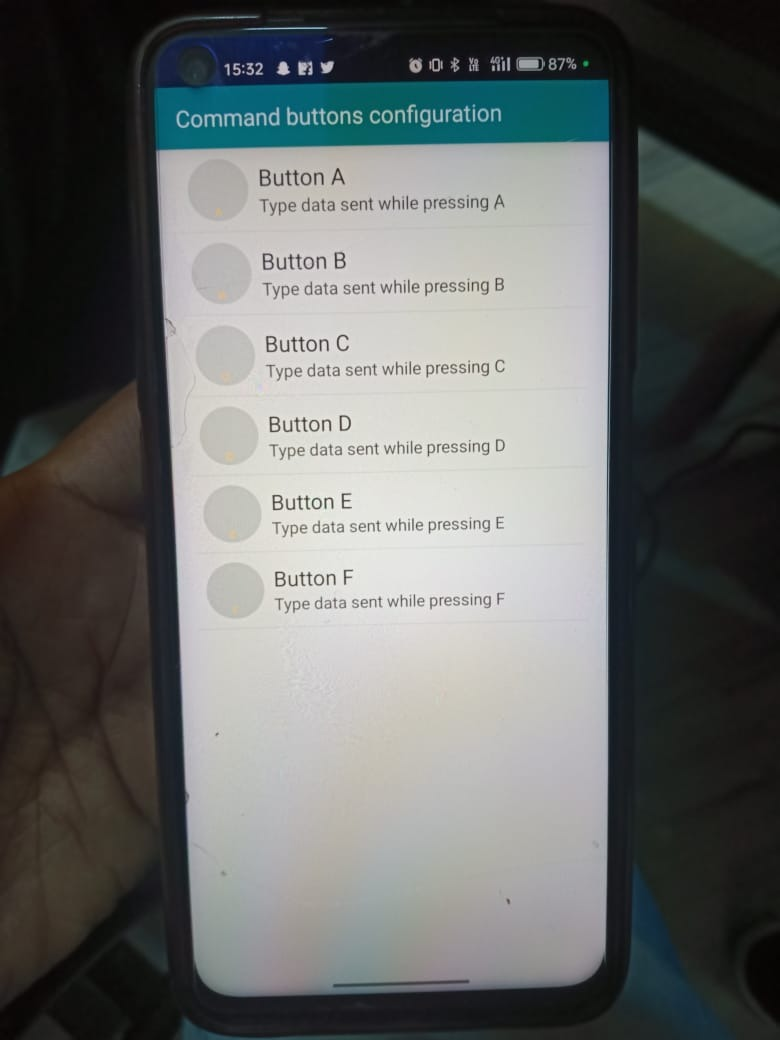
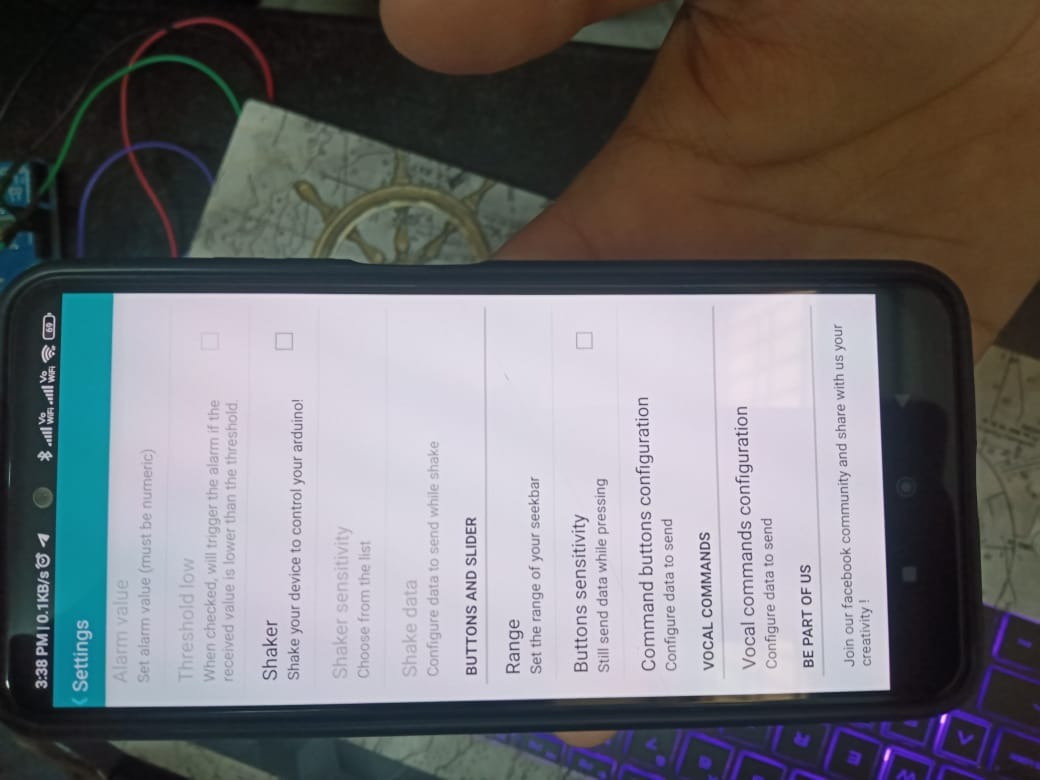
SUART.write(c);

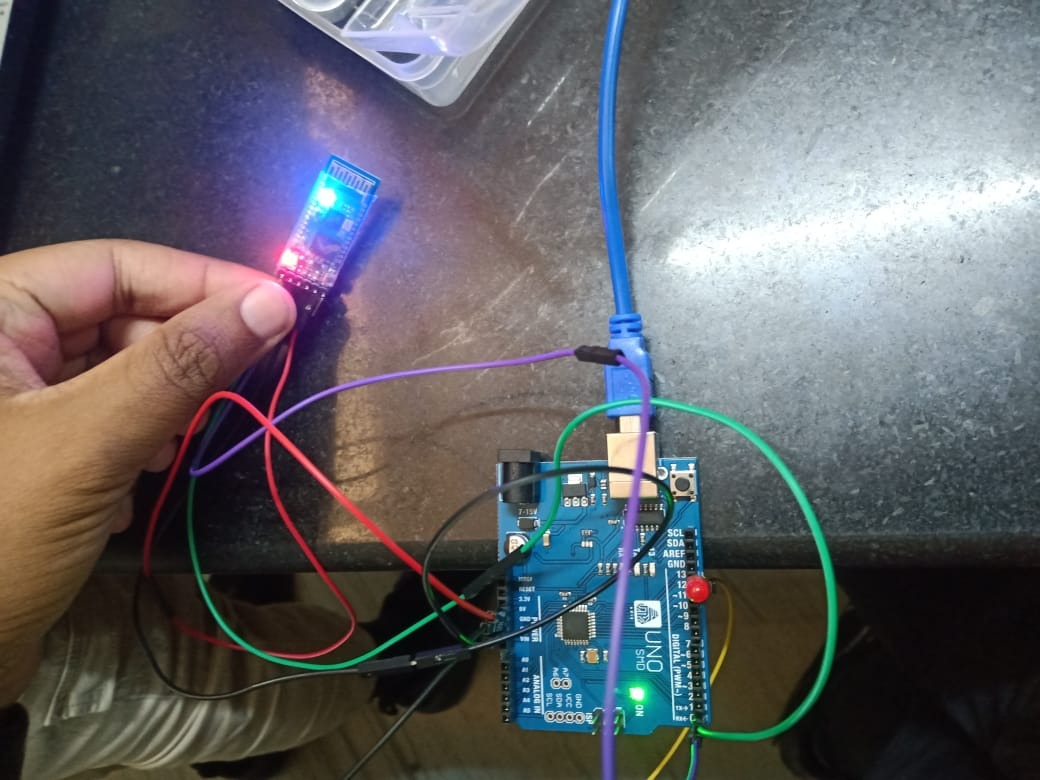
}

}

1. **Output:**





**Learning outcomes (What I have learnt):**

* Learnt about Bluetooth module in detail.
* Learnt about IoT programming.
* Learnt about **Real Time application of controlling actuators through bluetooth (HC-05) application using Arduino**.