

# CENTER FOR SKILL AND ENTREPRENEURSHIP DEVELOPMENT

(CSED)

INDUSTRIAL INTERNET OF THINGS

PROGRAM CODE: IIOT-3

PROGRAM NAME: SMART INDUSTRIAL CONNECTIVITY FOR IIOT

TOPIC: BLUETOOTH  
INTERFACING WITH ARDUINO



+91 20 27 276939

+91 79722 67361



425, Pride Purple Square, Kalewadi Chowk,  
Aundh Ravet Road, Wakad, Pune: 411057  
Maharashtra, India



info@dcsplm.com  
www.DYSMECH.com

## Table of Contents

Bluetooth Interfacing with Arduino .....	3
How Does it Work?? .....	3
Code .....	4
Code Explanation .....	5
Output .....	5
How to use the App? .....	6
References .....	8

## BLUETOOTH INTERFACING WITH ARDUINO

The circuit is so simple and small, there are only a few connections to be made

Arduino Pins                  Bluetooth Pins

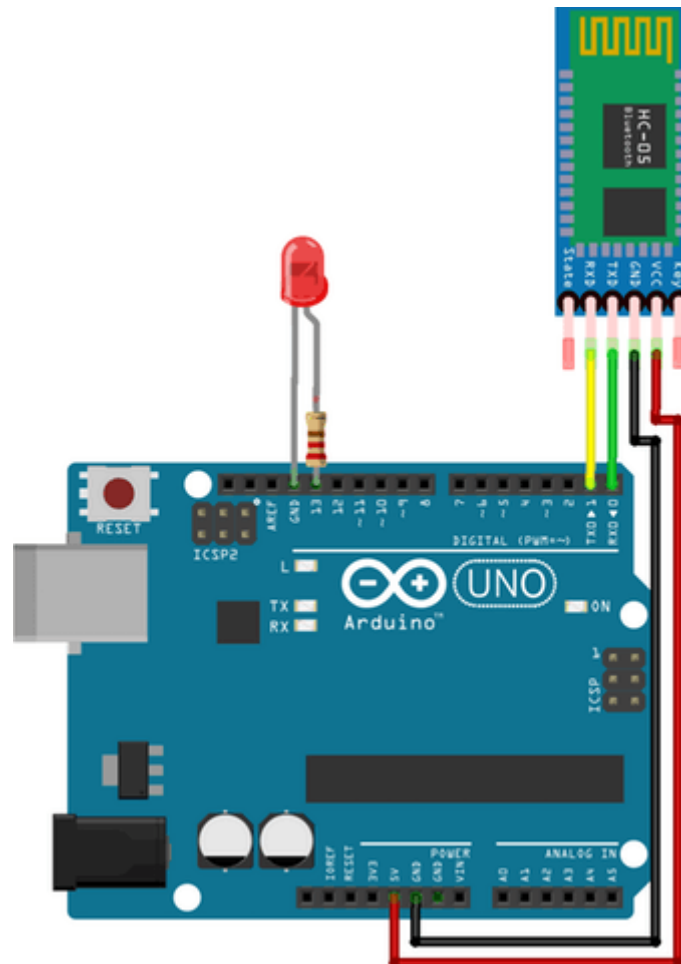
RX (Pin 0)                  TX

TX (Pin 1)                  RX

5V                          VCC

GND                        GND

Connect a LED negative to GND of Arduino and positive to pin 13 with a resistance valued between 220Ω – 1KΩ. And you're done with the circuit



Note: Don't Connect RX to RX and TX to TX of Bluetooth to Arduino you will receive no data, Here TX means Transmit and RX means Receive

### HOW DOES IT WORK??

HC 05/06 works on serial communication. Here the android app is designed to send serial data to the Bluetooth module when a button is pressed. The Bluetooth module at the other end receives

the data and sends it to Arduino through the TX pin of the Bluetooth module(RX pin of Arduino). The Code fed to Arduino checks the received data and compares it. If received data is 1 the LED turns on turns OFF when received data is 0

## CODE

We will upload the program first because you will receive an error for sure if you try to upload the sketch after making the connection. The reason is that the TX and RX pins are used while uploading the program to the microcontroller. You can see the TX RX LEDs near the onboard LED blinks continuously at the time of uploading the program.

```
char data = 0; //Variable for storing received data
void setup()
{
    Serial.begin(9600); //Sets the baud for serial data transmission
    pinMode(13, OUTPUT); //Sets digital pin 13 as output pin
}
void loop()
{
    if(Serial.available() > 0) // Send data only when you receive
data:
    {
        data = Serial.read(); //Read the incoming data and
store it into variable data
        Serial.print(data); //Print Value inside data in
Serial monitor
        if(data == '1') // Checks whether value of data
is equal to 1
            digitalWrite(13, HIGH); //If value is 1 then LED turns ON
        else if(data == '0') // Checks whether value of data
is equal to 0
            digitalWrite(13, LOW); //If value is 0 then LED turns
OFF
    }
}
```

## CODE EXPLANATION

```
char data = 0;
```

Initially, we created a variable called data to store data received from the Bluetooth module.

```
void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}
```

Inside `setup()` Serial communication is initialized with a baud rate of 9600. Since the HC-05/06 Bluetooth module communicates to Arduino through serial communication. `pinMode(13, OUTPUT);` sets digital pin 13 as OUTPUT pin. Since we are using a LED, which is an output device.

```
if(Serial.available() > 0)
{
  data = Serial.read();
  Serial.print(data);
}
```

The `Serial.available()` function returns the number of characters available for reading from the serial port.

If the value is greater than 0, which means it's receiving Serial data. Then the received characters are stored in the variable `data`.

```
if(data == '1')
  digitalWrite(13, HIGH);
else if(data == '0')
  digitalWrite(13, LOW);
}
```

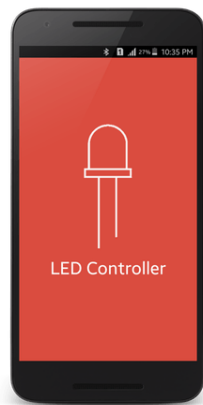
If the received data equal to `1` then digitalpin pin 13 is made HIGH and made LOW when data equal to `0`

## OUTPUT

- Open the serial monitor and watch the received data

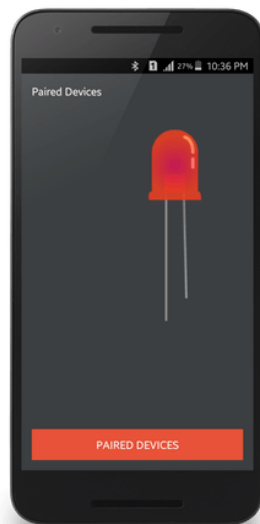
## HOW TO USE THE APP?

- Develop Application using MIT AppInventor
- Pair your device with HC 05/06 Bluetooth module1) Turn ON HC 05/06 Bluetooth module2) Scan for available device3) Pair to HC 05/06 by entering default password 1234 OR 0000
- Install LED application on your android device
- Open the Application



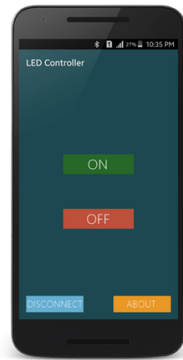
Splash screen

- Press paired devices
- Select your Bluetooth module from the List (HC 05)



Paring screen

- After connecting successfully
- Press ON button to turn ON LED and OFF button to turn OFF the LED



- Disconnect button to disconnect from Bluetooth module

## REFERENCES

1. <https://learn.sparkfun.com/tutorials/bluetooth-basics/all>
2. <https://www.u-blox.com/en/publication/white-paper/use-case-possibilities-bluetooth-low-energy-iot-applications>
3. <https://create.arduino.cc/projecthub/mayooghgirish/arduino-bluetooth-basic-tutorial-d8b737>
4. <http://mgprojecthub.com/arduino-bluetooth-basic-tutorial/>
5. <https://circuitdigest.com/microcontroller-projects/controlling-raspberry-pi-gpio-using-android-app-over-bluetooth>
6. <https://circuitdigest.com/fullimage?i=inlineimages/bluetooth-paring-android-phone-with-raspberry-pi.gif>