Practical – 2

Aim: What's GPIO & it's use, Interfacing & Programming of LED & RGB LED with Arduino UNO.

1) What's GPIO in IoT?

- General Purpose Input Output (GPIO) refers to one or more open-ended digital IO pins found on many electronic devices and circuit boards. As the words "General Purpose" suggest, these open-ended pins can be used for a variety of purposes, most commonly, to collect input such as that from sensors or to assert an output signal to control functionality on other device. Given this open-endedness, GPIO pins are often referred to as "uncommitted pins" since their purpose/functionality is often left up to the developer to determin.
- GPIO can be used in three modes:
- **input**
- **output**
- **UART interface**

- **GPIO input**

This is the default mode, in which the beacon receives input from the connected device via GPIO. You can imagine a button broadcasting its status (on/off) through the beacon.

In this configuration, the beacon will broadcast received data in the Estimote Telemetry packet. This means information about binary states of two GPIO pins. In other words, the beacon will be advertising two 0/1 values.

If you want to learn how to enable Estimote Telemetry, read:

What is a beacon protocol? Can beacons broadcast multiple packets simultaneously?

- **GPIO output**

In the output mode, beacon delivers data to the connected device via GPIO. You could for example switch a LED lamp on or off with the beacon controlled from a mobile app.

In this configuration, the beacon will deliver data from two pins about their binary states to the connected device.

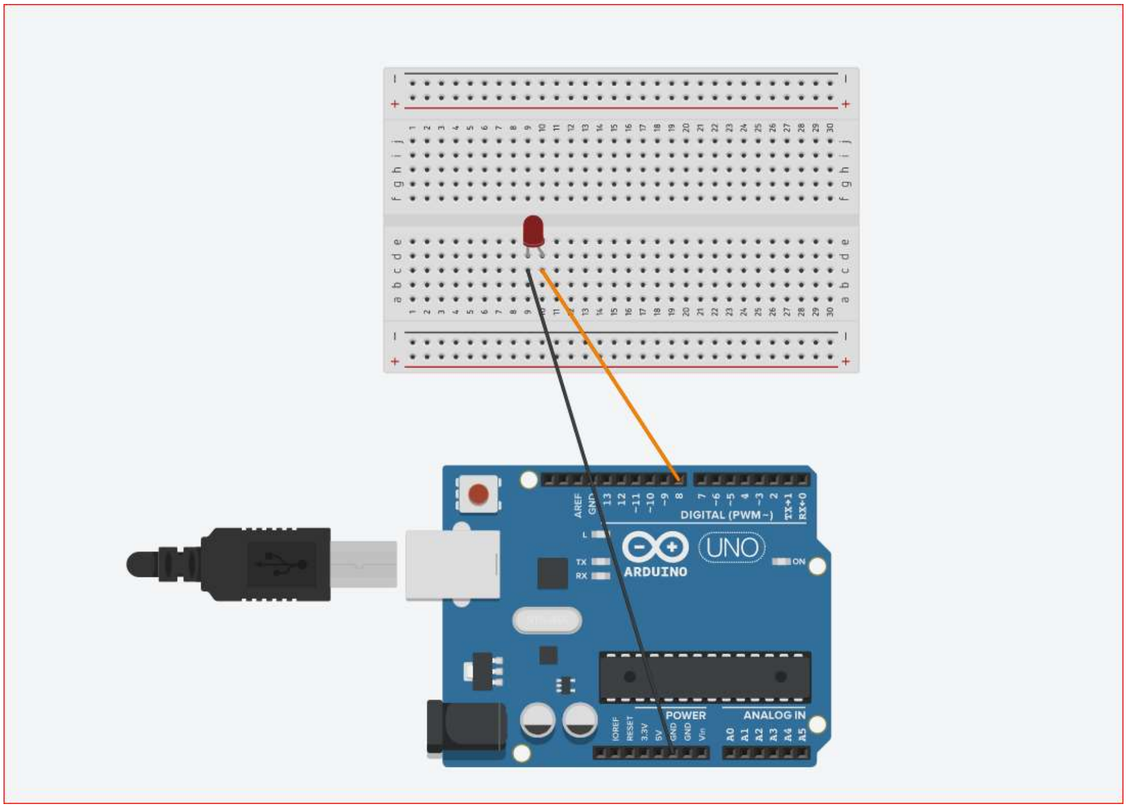
- **UART**

If you configure GPIO as an UART interface, you'll be able to define custom advertising packets for an Estimote Beacon. This feature will soon be enabled in the SDK.

  - GPIO is found in many use cases ranging from toggling devices like actuators or heaters, to reading the state of an external device's switch, and even implementing various communication protocols. So with such flexibility, it's not hard to imagine their significance in embedded systems such as IoT devices and robotics.

2) GPIO Interfacing and programming demo: *LED* Blink using Arduino UNO --> It should include: Tinkercad circuit connection setup image, code, output (screenshot/serial monitor/image of activity).

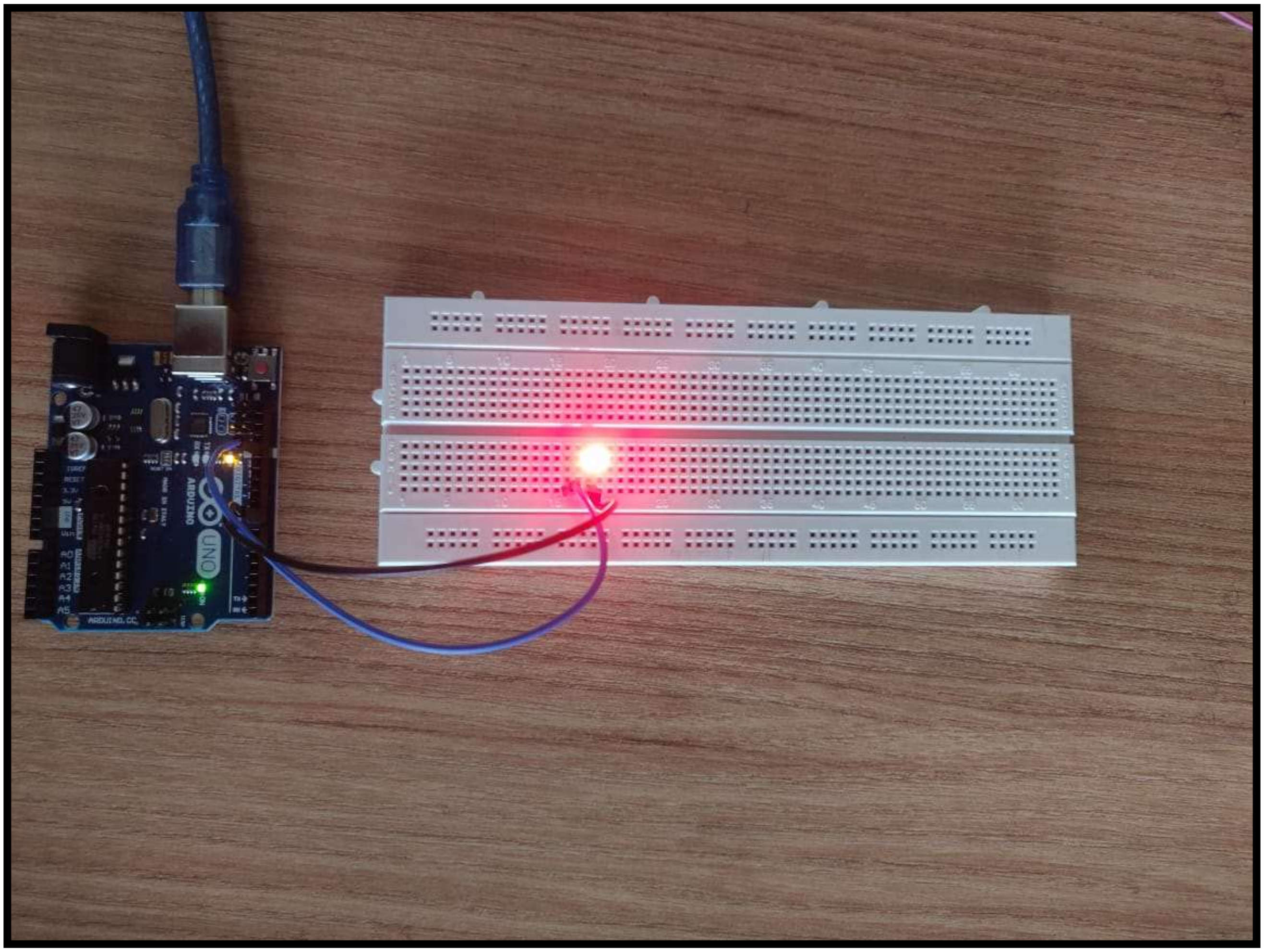
SETUP:



CODE:

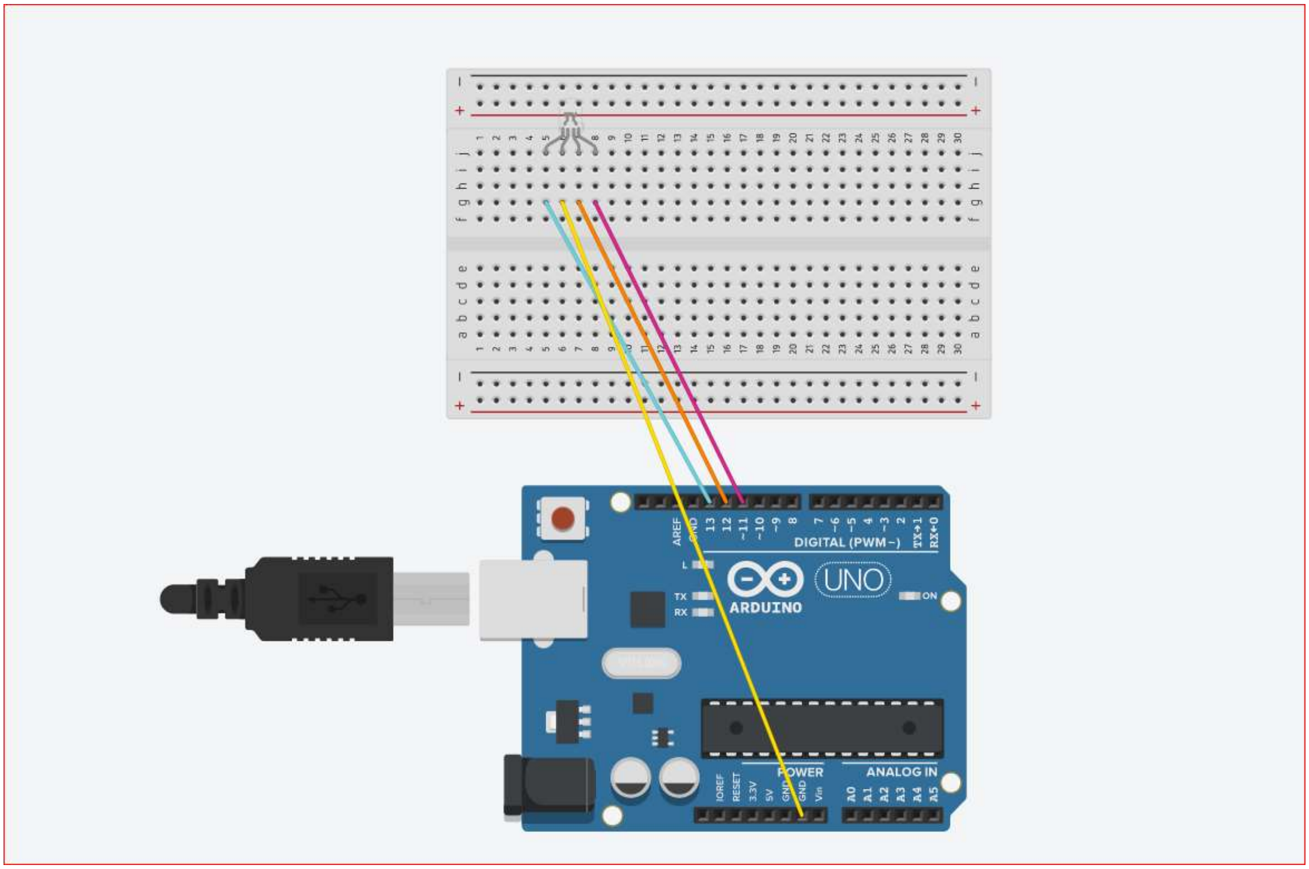
```
int pinSetup=8;
void setup()
{
  pinMode(pinSetup, OUTPUT);
}

void loop()
{
  digitalWrite(pinSetup, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(pinSetup, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

OUTPUT:



3) **GPIO Interfacing and programming demo: *RGB LED* Blink using Arduino UNO --> It should include: Tinkercad circuit connection setup image, code, output (screenshot/serial monitor/image of activity).**

SETUP:



CODE:

```
 void setup()

{

  pinMode(13, OUTPUT);

  pinMode(12, OUTPUT);

  pinMode(11, OUTPUT);

}


void loop()

{

  digitalWrite(13, HIGH);

  digitalWrite(12, LOW);

  digitalWrite(11, LOW);
  delay(1000); // Wait for 1000 millisecond(s)


  digitalWrite(13, LOW);

  digitalWrite(12, HIGH);

  digitalWrite(11, LOW);

  delay(1000); // Wait for 1000 millisecond(s)


  digitalWrite(13, LOW);

  digitalWrite(12, LOW);

  digitalWrite(11, HIGH);

  delay(1000); // Wait for 1000 millisecond(s)

}
```

OUTPUT: