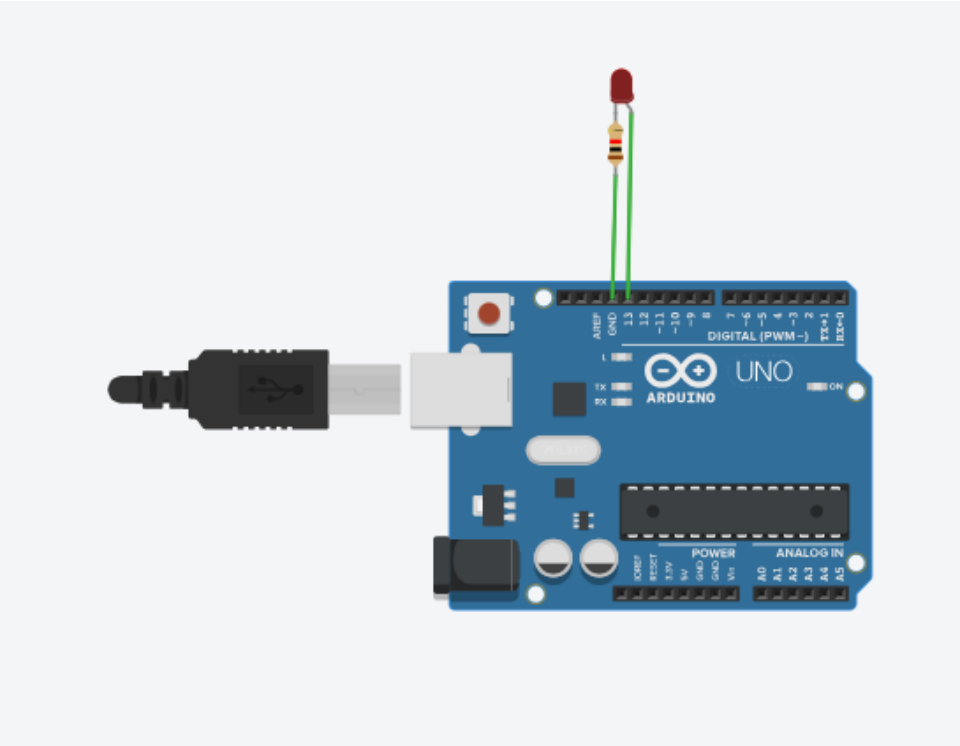
IoT Lab Answers

question 1 :

Design an IoT Circuit for on/off the LED bulb with a delay of 5 sec

answer:



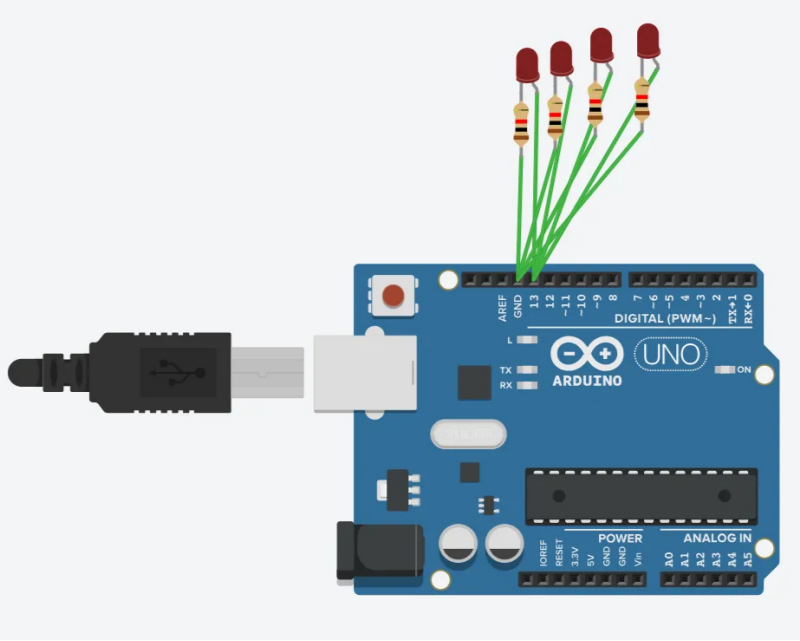

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);  // Turn on LED
  delay(5000);             // Wait for 5 seconds
  digitalWrite(13, LOW);   // Turn off LED
```

```
  delay(5000);              // Wait for 5 seconds
}
```

question 2:

2. **Switch on/off four LED bulbs in parallel manner**

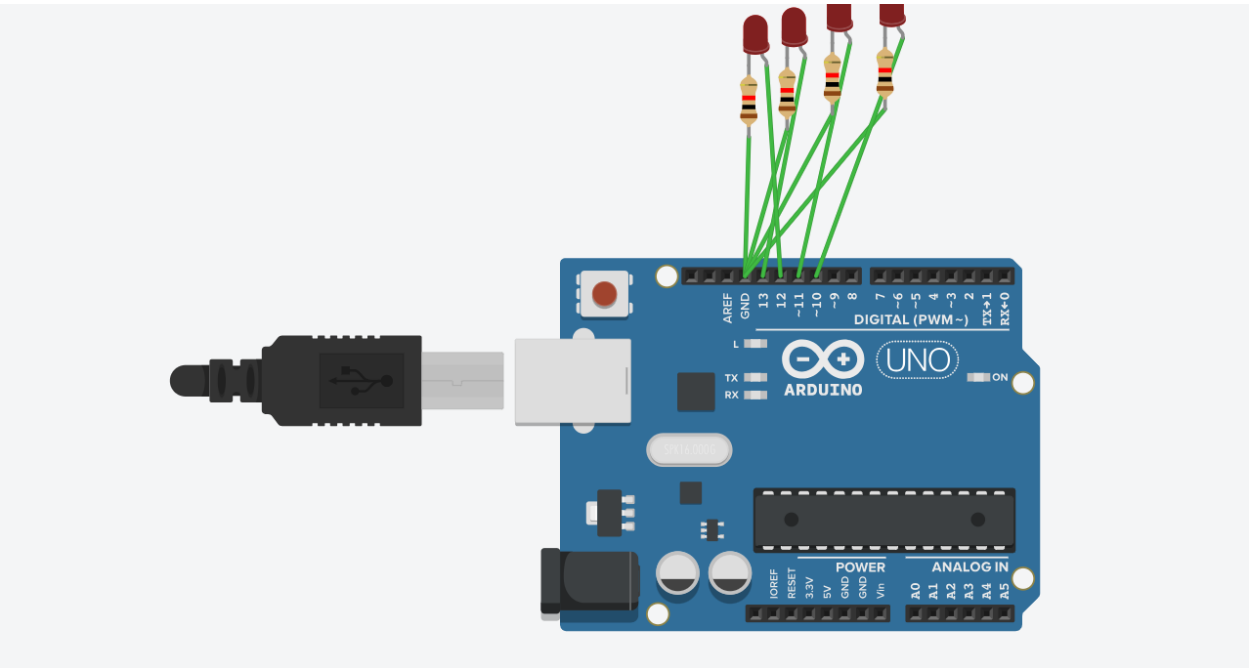
answer:




```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // Turn on all LEDs
  delay(1000);            // Wait for 1 second
  digitalWrite(13, LOW);  // Turn off all LEDs
  delay(1000);            // Wait for 1 second
}
// Connect all LEDs in " parallel ", each having a resistor, to
```

3. **Switch on/off four LED bulbs in series manner**

answer:

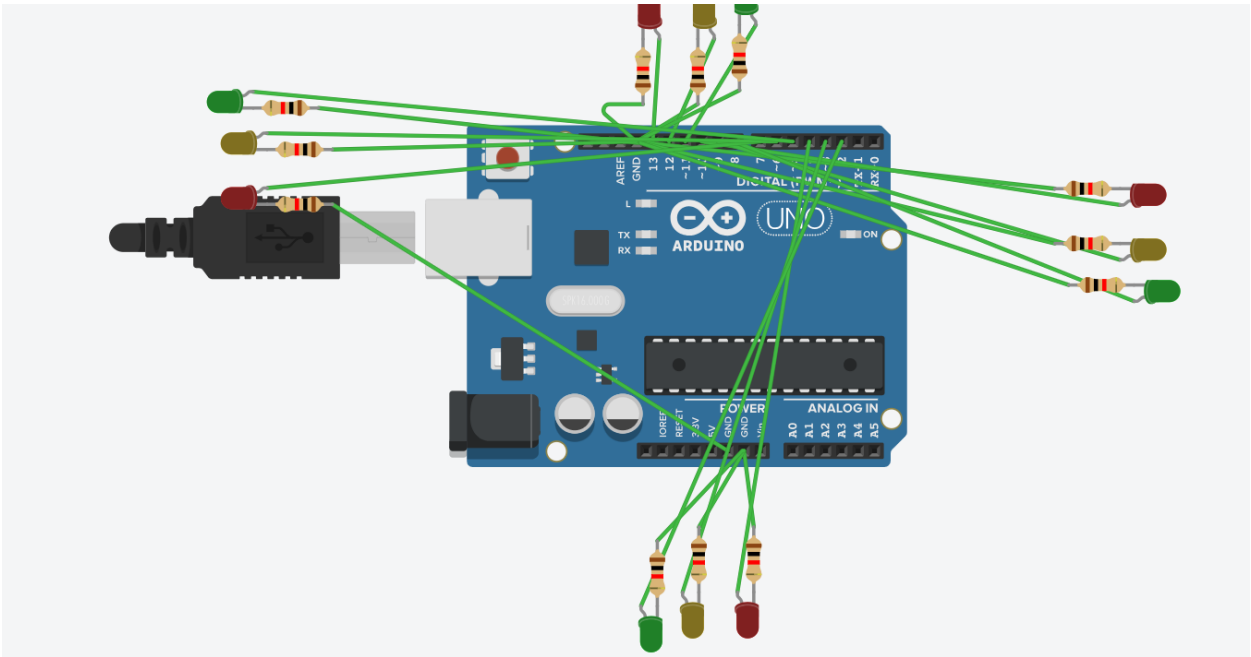



```
void setup()
{
  pinMode(13,OUTPUT);
   pinMode(12,OUTPUT);
   pinMode(11,OUTPUT);
   pinMode(10,OUTPUT);
}
void loop(){
digitalWrite(13,HIGH);
  delay(1000);
  digitalWrite(13,LOW);
  delay(1000);
  digitalWrite(12,HIGH);
  delay(1000);
  digitalWrite(12,LOW);
  delay(1000);
  digitalWrite(11,HIGH);
```

```
  delay(1000);
  digitalWrite(11,LOW);
  delay(1000);
  digitalWrite(10,HIGH);
  delay(1000);
  digitalWrite(10,LOW);
  delay(1000);
}
/* Connect all LEDs in " series ", each having a resistor,
to digital pins in serial manner 13,12,11,10 */
```

4. Four-way Junction Traffic Controller System

answer:




```
int red1 = 13, yellow1 = 12, green1 = 11;
int red2 = 10, yellow2 = 9, green2 = 8;
int red3 = 7, yellow3 = 6, green3 = 5;
int red4 = 4, yellow4 = 3, green4 = 2;
```

```
void setup() {
  for (int i = 2; i <= 13; i++) {
    pinMode(i, OUTPUT);
  }
}

void loop() {
  controlLights(red1, yellow1, green1, red2, yellow2, green2, re
  controlLights(red2, yellow2, green2, red1, yellow1, green1, re
  controlLights(red3, yellow3, green3, red1, yellow1, green1, re
  controlLights(red4, yellow4, green4, red1, yellow1, green1, re
}

void controlLights(int mainRed, int mainYellow, int mainGreen,
                   int cross1Red, int cross1Yellow, int cross1Gr
                   int cross2Red, int cross2Yellow, int cross2Gr
                   int cross3Red, int cross3Yellow, int cross3Gr
  // Main direction green light on, others red
  digitalWrite(mainGreen, HIGH);  // Green on
  digitalWrite(mainYellow, LOW);  // Yellow off
  digitalWrite(mainRed, LOW);     // Red off
  digitalWrite(cross1Red, HIGH);  // Other directions red
  digitalWrite(cross2Red, HIGH);
  digitalWrite(cross3Red, HIGH);

  delay(5000);  // Main green light duration

  // Switch to yellow before changing
  digitalWrite(mainGreen, LOW);
  digitalWrite(mainYellow, HIGH);  // Yellow on
  delay(3000);                     // Yellow duration

  // Reset lights after yellow period
  digitalWrite(mainYellow, LOW);
```

```
  digitalWrite(mainRed, HIGH);     // Turn red on for main
}
```

5. Display Temperature and Humidity in Serial Monitor

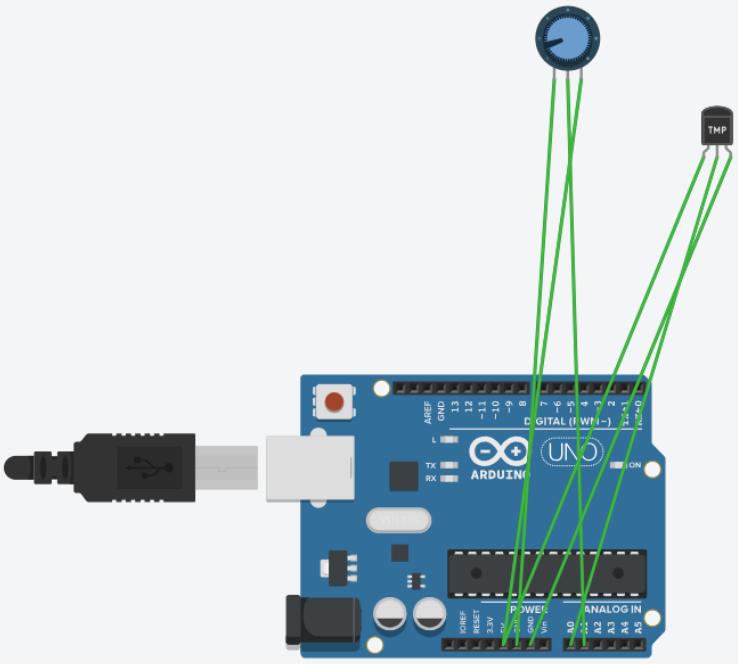
answer :

hardware

```
#include <DHT.h>

#define DHTPIN 2       // Pin connected to the DATA pin of the D
#define DHTTYPE DHT11  // DHT11 or DHT22; change to DHT22 if usi

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);   // Start serial communication
  dht.begin();          // Initialize the DHT sensor
}

void loop() {
  float humidity = dht.readHumidity();         // Read humidity
  float temperature = dht.readTemperature();   // Read temperatu

  // Check if any readings failed
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Print the readings to the Serial Monitor
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
```

```
  Serial.print(temperature);
  Serial.println(" °C");

  delay(2000);  // Wait 2 seconds before taking another reading
}
```

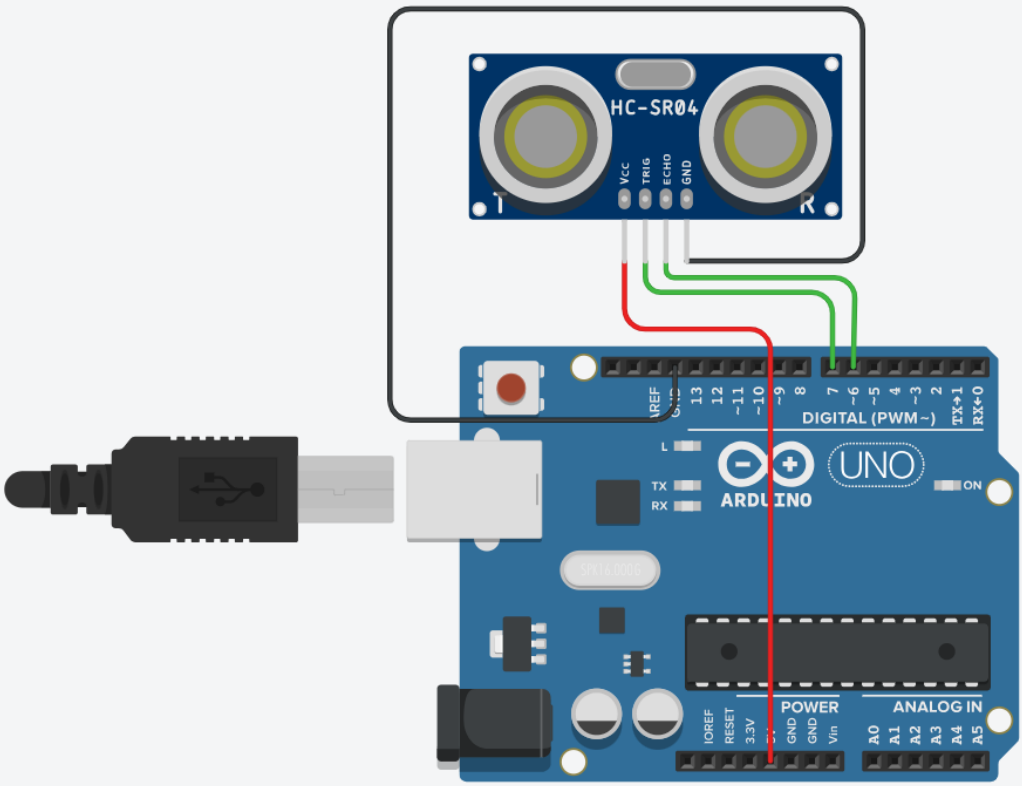
Tinkercad




```
/*
This code records the temperature through testing the mV put out
It records in both Celcius and Fahrenheit.
It can only detect from -40 degrees C to 125 degrees C or -40 de
to 257 degrees F
The Humidity is simulated by a potentiometer by being mapped int
*/
```

```
const int analogIn = A0;
int humiditysensorOutput = 0;
// Defining Variables
int RawValue= 0;
double Voltage = 0;
double tempC = 0;
double tempF = 0;

void setup(){
  Serial.begin(9600);
  pinMode(A1, INPUT);
}

void loop(){

  RawValue = analogRead(analogIn);
  Voltage = (RawValue / 1023.0) * 5000; // 5000 to get millivots
  tempC = (Voltage-500) * 0.1; // 500 is the offset
  tempF = (tempC * 1.8) + 32; // convert to F
  Serial.print("Raw Value = " );
  Serial.print(RawValue);
  Serial.print("\t milli volts = ");
  Serial.print(Voltage,0); //
  Serial.print("\t Temperature in C = ");
  Serial.print(tempC,1);
  Serial.print("\t Temperature in F = ");
  Serial.println(tempF,1);
  humiditysensorOutput = analogRead(A1);
  Serial.print("Humidity: "); // Printing out Humidity Percentag
  Serial.print(map(humiditysensorOutput, 0, 1023, 10, 70));
  Serial.println("%");

  delay(5000);  //iterate every 5 seconds

}
```

6. Display Distance between Object and Sensor




```
const int trigPin=7;
const int echoPin=6;
float duration,distance;
void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
```

```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(2);

  digitalWrite(trigPin, LOW);
  duration=pulseIn(echoPin,HIGH);

  distance=(duration*0.0343)/2;

  Serial.print("\t Distance: ");
  Serial.print(distance);      // cm
  // Serial.print("\t Time:");
  // Serial.print(duration);
  Serial.println(" ");
  delay(1000);
}
```

7. Display Temperature and Humidity in Cloud Database.

```
#include <WiFi.h>
#include <DHT.h>
#include "ThingSpeak.h"

#define DHTPIN 4             // Pin connected to DHT sensor
#define DHTTYPE DHT11        // DHT sensor type (DHT11)

DHT dht(DHTPIN, DHTTYPE);    // Initialize DHT sensor
WiFiClient client;           // Initialize WiFi client

// WiFi and ThingSpeak credentials
const char* ssid = "Praveen";                  // WiFi SSID
const char* password = "123456789";            // WiFi password
unsigned long channelID = 539141;              // ThingSpeak Char
const char* writeAPIKey = "0SY6SUIDJ47NLIXN"; // ThingSpeak Writ

float humidity, temperature;
```

```
void setup() {
    Serial.begin(115200);
    dht.begin();  // Start DHT sensor

    // Connect to WiFi
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nWiFi connected.");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());

    ThingSpeak.begin(client);  // Initialize ThingSpeak
}

void loop() {
    // Read humidity and temperature from the DHT sensor
    humidity = dht.readHumidity();
    temperature = dht.readTemperature();

    // Display readings in Serial Monitor
    Serial.print("Temperature: ");
    Serial.println(temperature);
    Serial.print("Humidity: ");
    Serial.println(humidity);

    // Upload data to ThingSpeak
    ThingSpeak.setField(1, temperature);
    ThingSpeak.setField(2, humidity);
    int writeSuccess = ThingSpeak.writeFields(channelID, writeAP
```

```
    if (writeSuccess == 200) {
        Serial.println("Data successfully uploaded.");
    } else {
        Serial.println("Failed to upload data. Error code: " + S
    }

    delay(10000); // Delay before next upload (ThingSpeak limits
}
```

Display distance of object in cloud.

```
#include <WiFi.h>
#include "ThingSpeak.h"

#define TRIG_PIN 5         // Pin connected to the ultrasonic se
#define ECHO_PIN 18        // Pin connected to the ultrasonic se

WiFiClient client;           // Initialize WiFi client

// WiFi and ThingSpeak credentials
const char* ssid = "Praveen";                 // WiFi SSID
const char* password = "123456789";           // WiFi password
unsigned long channelID = 539141;             // ThingSpeak Char
const char* writeAPIKey = "0SY6SUIDJ47NLIXN"; // ThingSpeak Writ

float distance;

void setup() {
    Serial.begin(115200);

    // Set up ultrasonic sensor pins
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);

    // Connect to WiFi
```

```
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nWiFi connected.");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());

    ThingSpeak.begin(client);  // Initialize ThingSpeak
}

void loop() {
    // Measure distance using ultrasonic sensor
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    // Read the echo pin and calculate distance
    long duration = pulseIn(ECHO_PIN, HIGH);
    distance = duration * 0.034 / 2; // Convert to centimeters

    // Display distance in Serial Monitor
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");

    // Upload distance to ThingSpeak
    ThingSpeak.setField(1, distance);
    int writeSuccess = ThingSpeak.writeFields(channelID, writeAP

    if (writeSuccess == 200) {
```

```
        Serial.println("Data successfully uploaded.");
    } else {
        Serial.println("Failed to upload data. Error code: " + S
    }

    delay(10000); // Delay before next upload (ThingSpeak limits
}
```

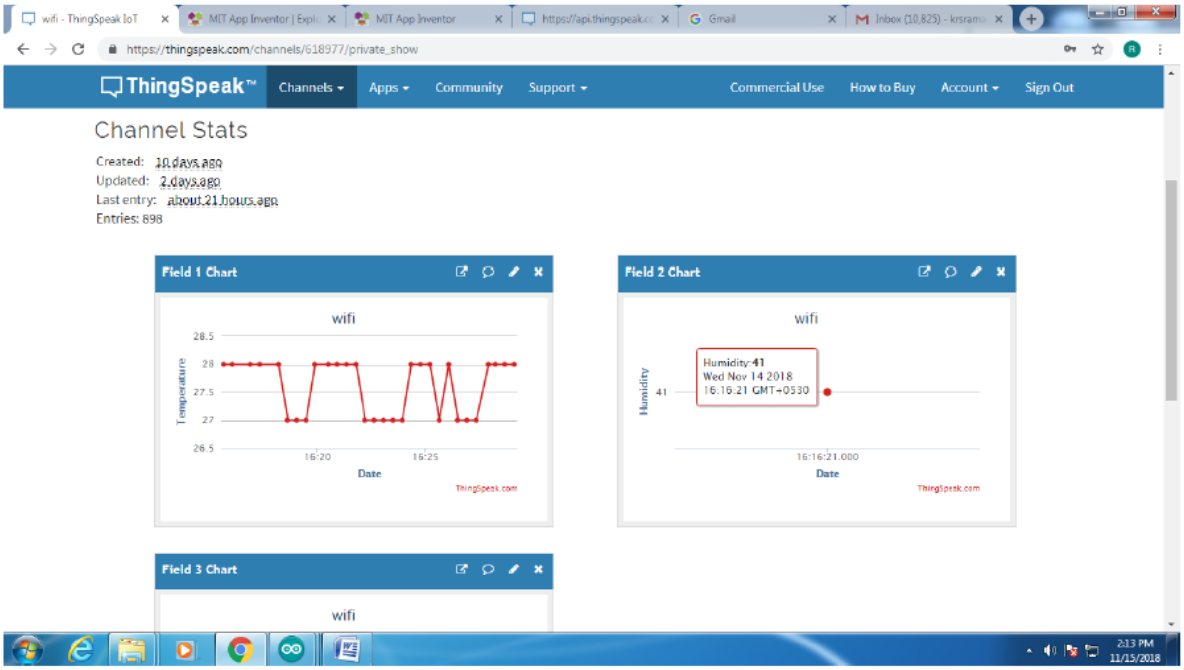
8. Display temp and humd in mobile app. (IoT)

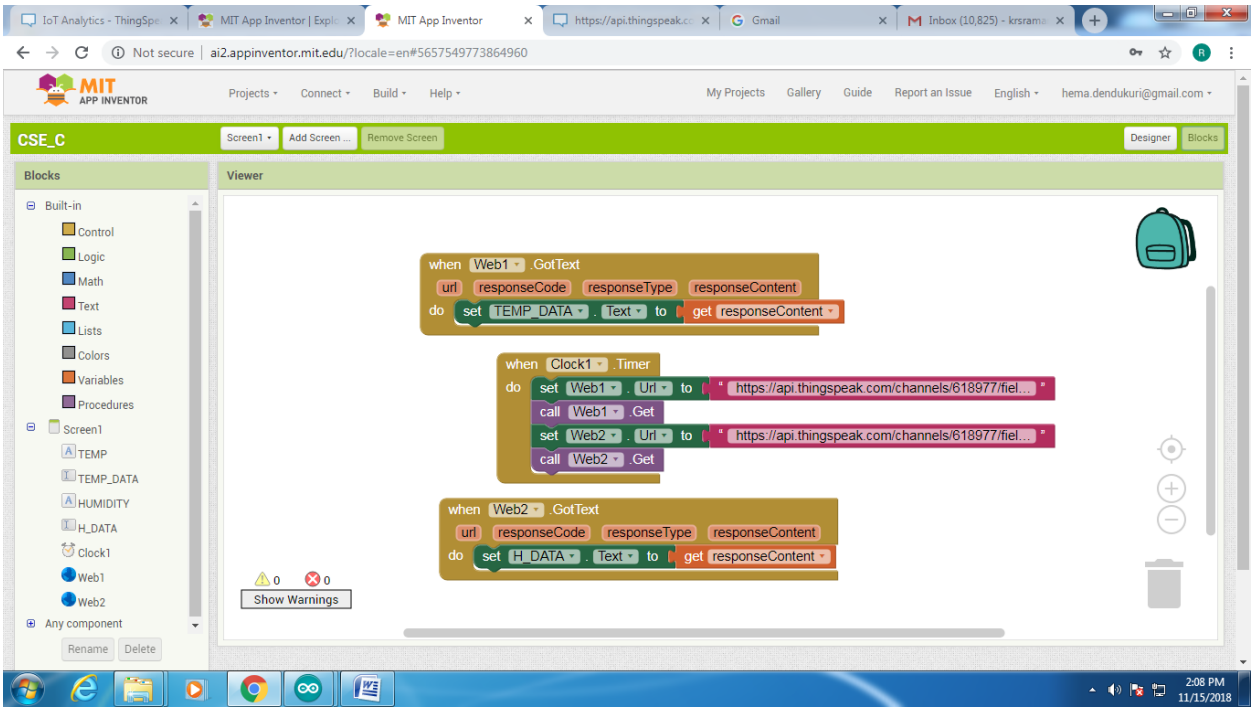
URL for Temperature field: **https://api.thingspeak.com/channels/618977/fields/1/last?result=0**

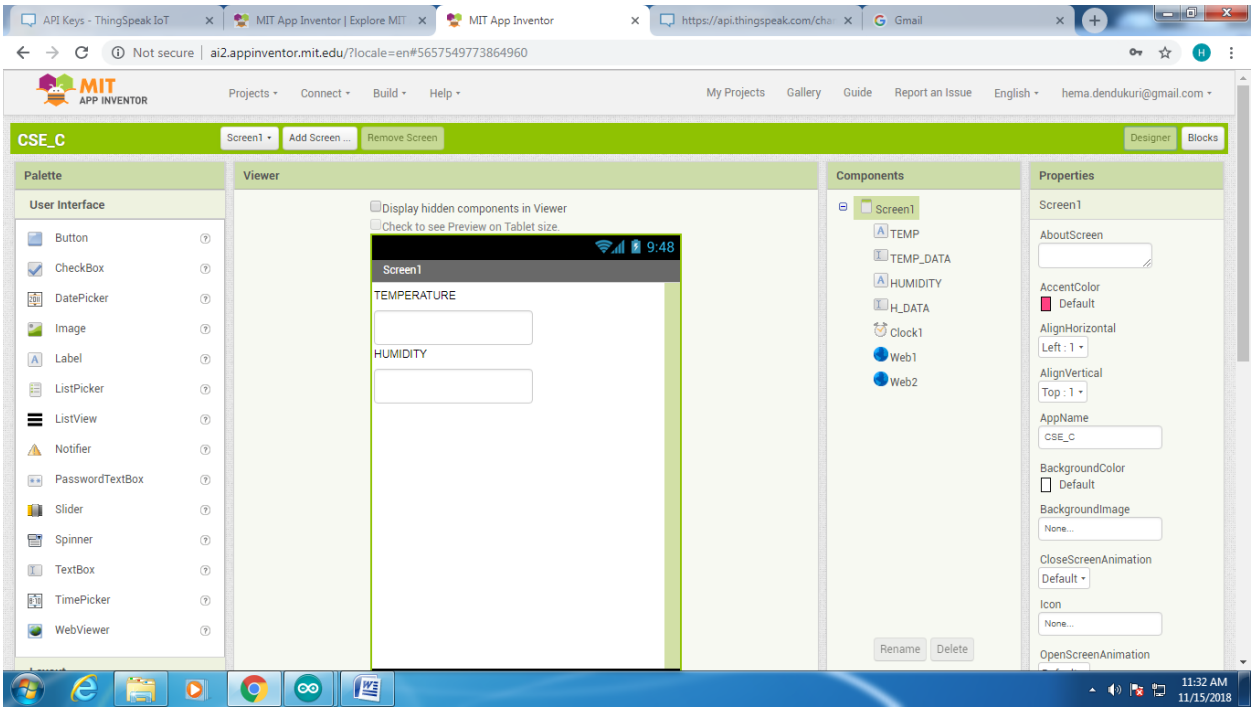
URL for Humidity field: **https://api.thingspeak.com/channels/618977/fields/2/last?result=0**

Cloud space using ThingSpeak.com host:

Channel Creation and Data view in ThingSpeak host








```
#include <ESP8266WiFi.h>    // or #include <WiFi.h> for ESP32
#include <DHT.h>

#define DHTPIN D2            // GPIO pin where the DHT sensor is
```

```
#define DHTTYPE DHT11        // Define DHT type (DHT11 or DHT22)
DHT dht(DHTPIN, DHTTYPE);

const char* ssid = "Your_SSID";             // Replace with your
const char* password = "Your_PASSWORD";     // Replace with your
const char* server = "api.thingspeak.com";
String apiKey = "YOUR_API_KEY";             // Replace with your

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");
  dht.begin();
}

void loop() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  if (WiFi.status() == WL_CONNECTED) {
    WiFiClient client;

    if (client.connect(server, 80)) {
      String postStr = apiKey;
      postStr += "&field1=";
      postStr += String(temperature);
```

```
      postStr += "&field2=";
      postStr += String(humidity);
      postStr += "\r\n\r\n";

      client.print("POST /update HTTP/1.1\n");
      client.print("Host: " + String(server) + "\n");
      client.print("Connection: close\n");
      client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
      client.print("Content-Type: application/x-www-form-urlenc
      client.print("Content-Length: ");
      client.print(postStr.length());
      client.print("\n\n");
      client.print(postStr);

      Serial.println("Temperature and humidity data sent to Thi
    }
    client.stop();
  }

  delay(20000);  // Update every 20 seconds (ThingSpeak limit)
}
```

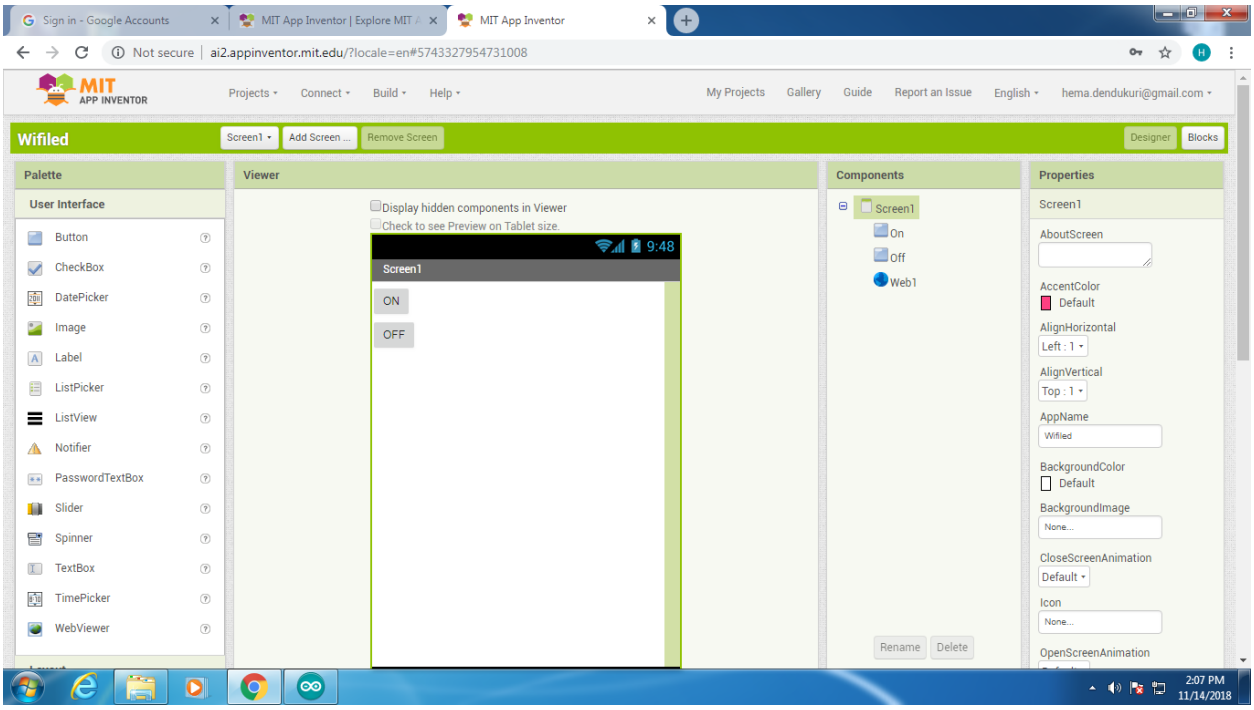
9. Distance in mobile application

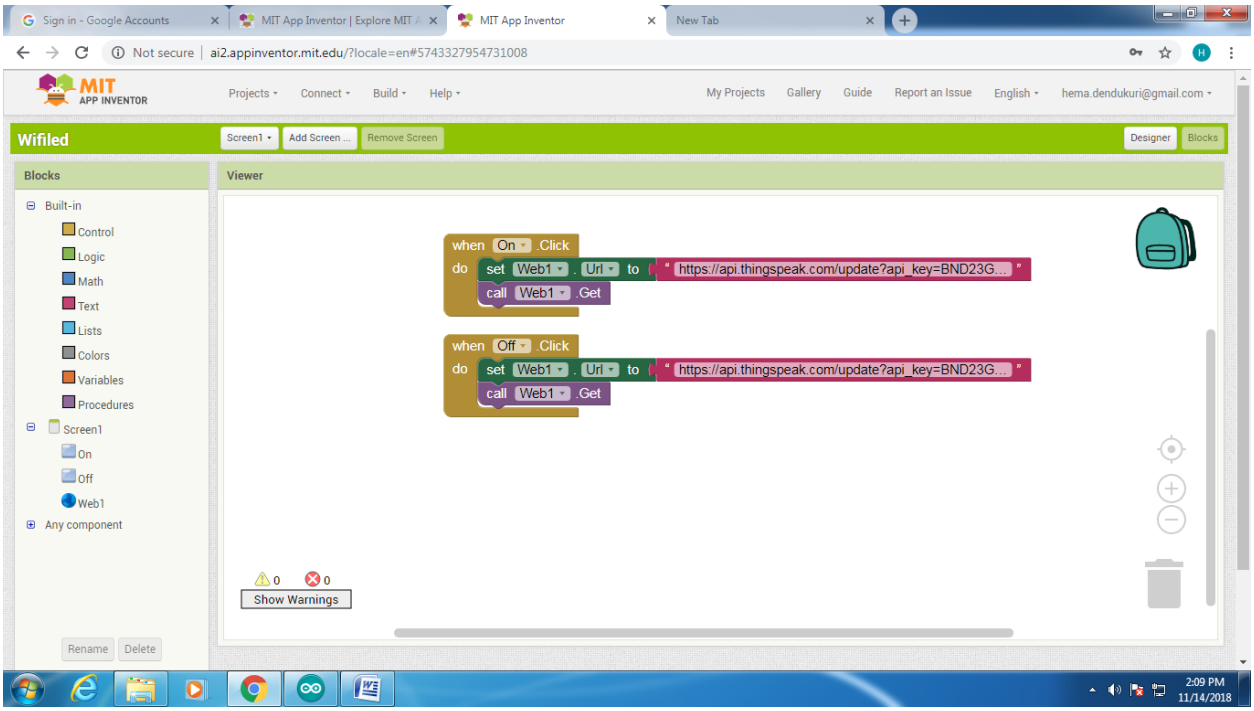
```
#include <WiFi.h>  // Or #include <WiFi.h> if using ESP32

#define trigPin 9
#define echoPin 10

const char* ssid = "Your_SSID";                 // Replace with you
const char* password = "Your_PASSWORD";         // Replace with you
const char* server = "api.thingspeak.com";
String apiKey = "YOUR_API_KEY";                 // Replace with you

void setup() {
```

```
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  // Measure Distance
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

  // Send data to ThingSpeak
  if (WiFi.status() == WL_CONNECTED) {
    WiFiClient client;

    if (client.connect(server, 80)) {
      String postStr = apiKey;
      postStr += "&field1=";
      postStr += String(distance);
      postStr += "\r\n\r\n";

      client.print("POST /update HTTP/1.1\n");
```

```
      client.print("Host: " + String(server) + "\n");
      client.print("Connection: close\n");
      client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
      client.print("Content-Type: application/x-www-form-urlenc
      client.print("Content-Length: ");
      client.print(postStr.length());
      client.print("\n\n");
      client.print(postStr);

      Serial.println("Distance data sent to ThingSpeak");
    }
    client.stop();
  }

  delay(20000);  // Update every 20 seconds (ThingSpeak limit)
}
```

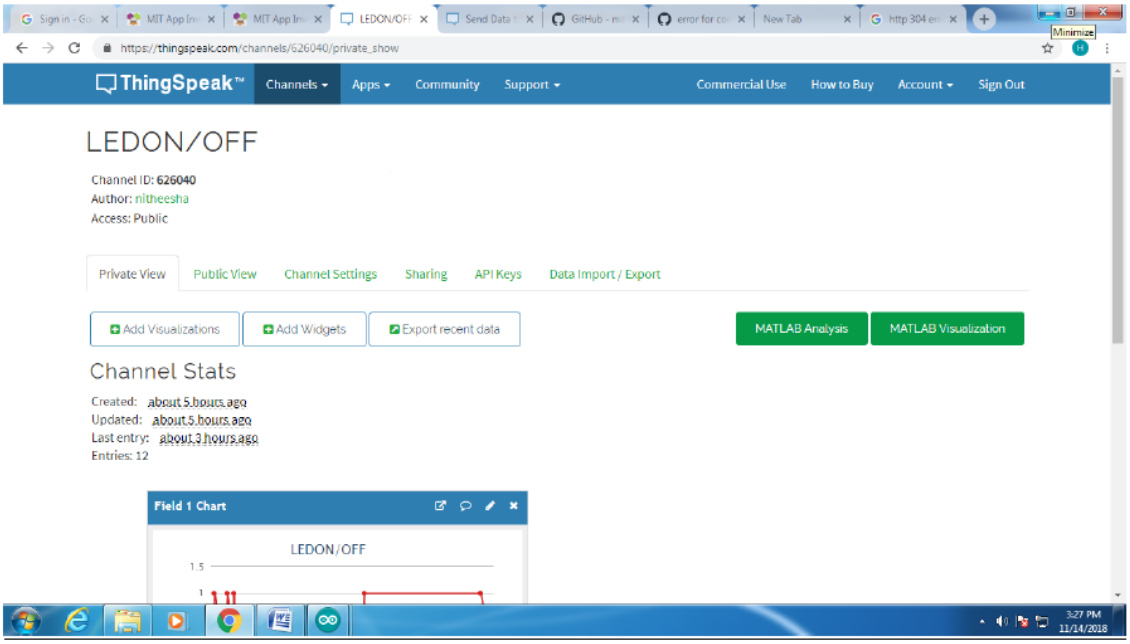
10. wiFi on/off light (mooooooooooo)

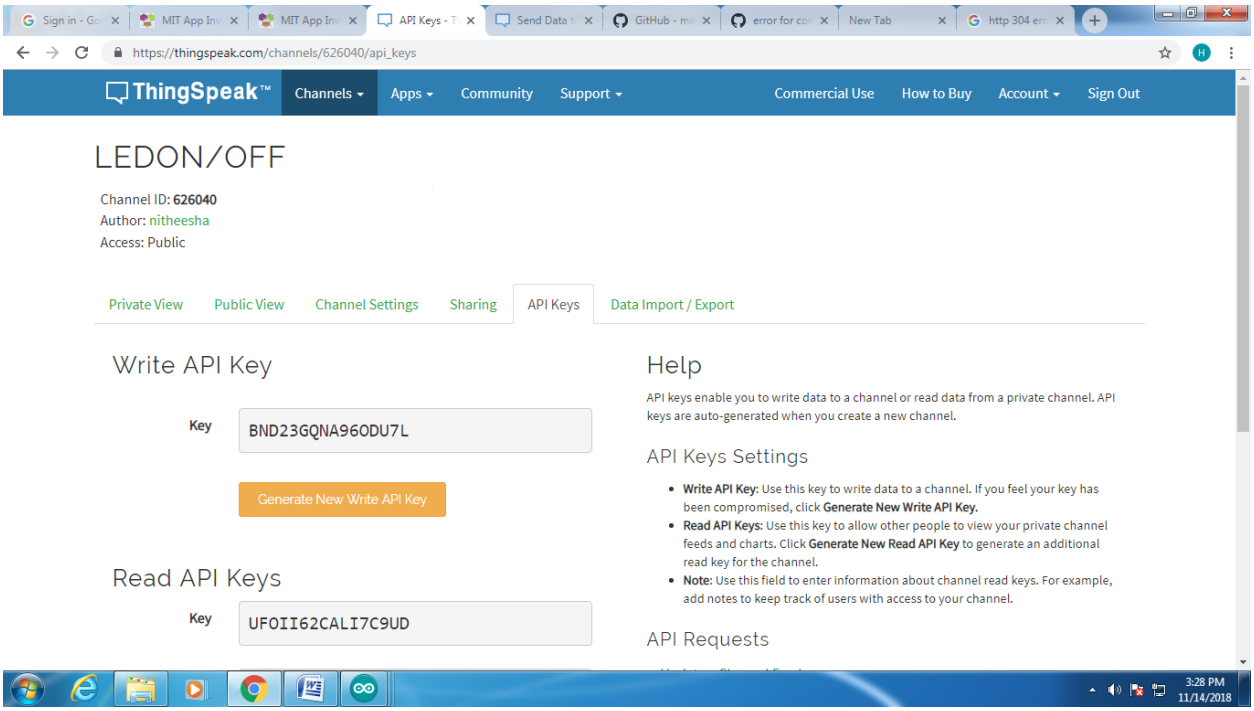



Sign in - Google Accounts
MIT App Inventor | Explore MIT
MIT App Inventor
New Tab
Not secure | ai2.appinventor.mit.edu/?locale=en#5743327954731008
MIT APP INVENTOR
Projects
Connect
Build
Help
My Projects
Gallery
Guide
Report an Issue
English
hema.dendukuri@gmail.com
Wifiled
Screen1
Add Screen ...
Remove Screen
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Colors
Variables
Procedures
Screen1
On
Off
Web1
Any component
Rename
Delete
Viewer
when On .Click
do set Web1 . Url to " https://api.thingspeak.com/update?api_key=BND23G... "
call Web1 .Get
when Off .Click
do set Web1 . Url to " https://api.thingspeak.com/update?api_key=BND23G... "
call Web1 .Get
0
0
Show Warnings
2:09 PM
11/14/2018

URL for ON button:
https://api.thingspeak.com/update?api_key=BND23GQNA96ODU7L&field1=1

URL for OFF button:
https://api.thingspeak.com/update?api_key=BND23GQNA96ODU7L&field1=0

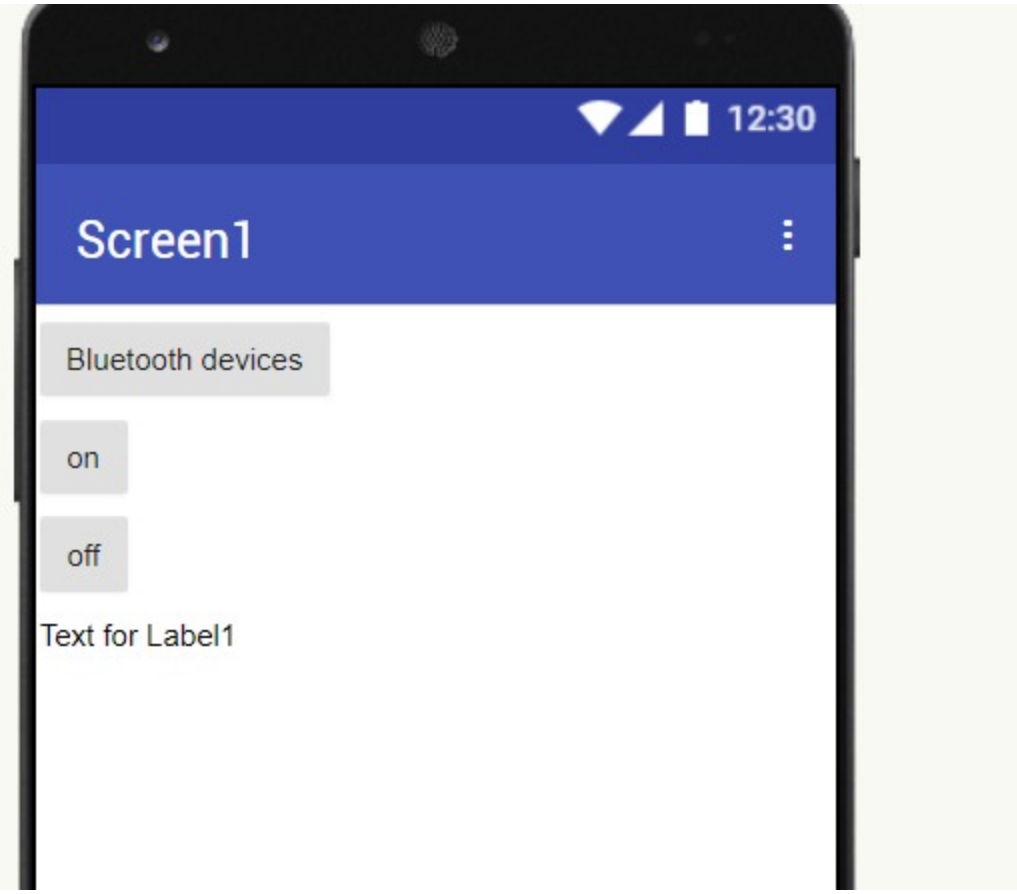


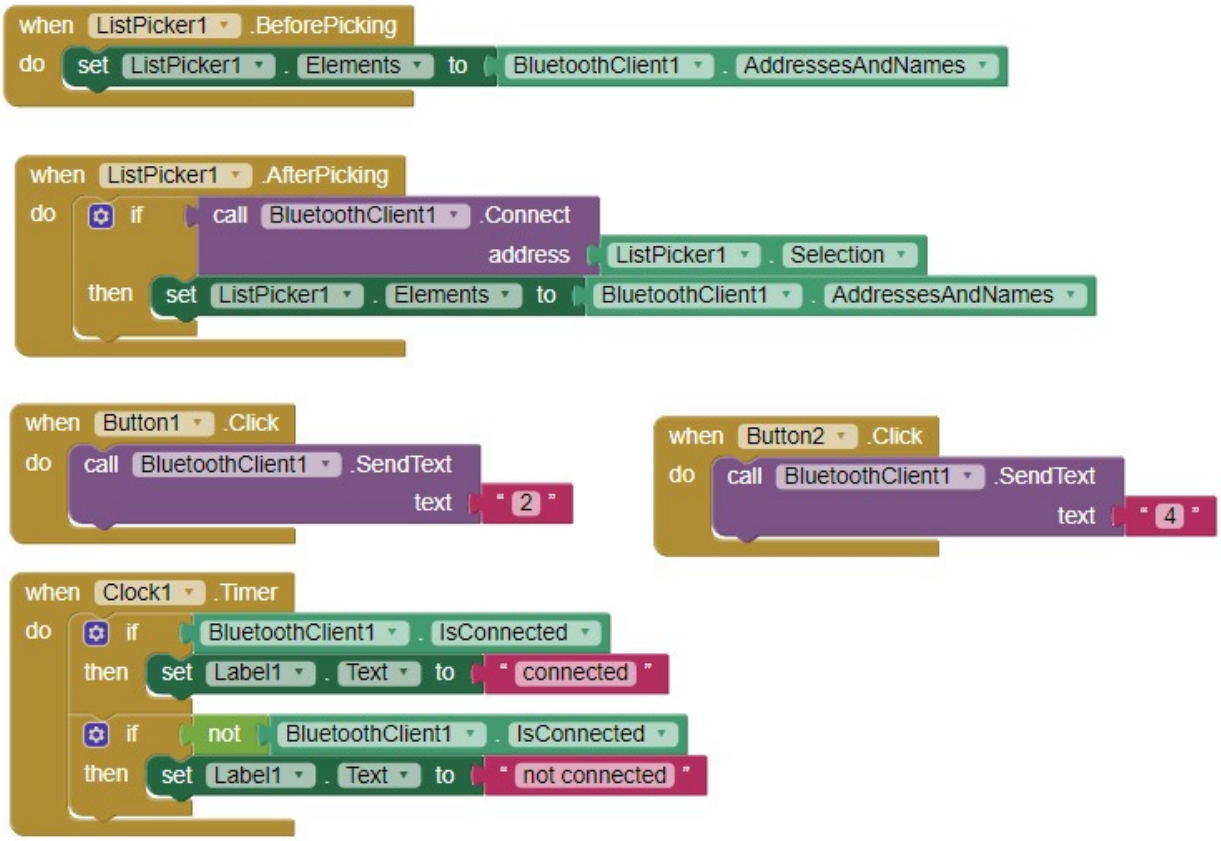


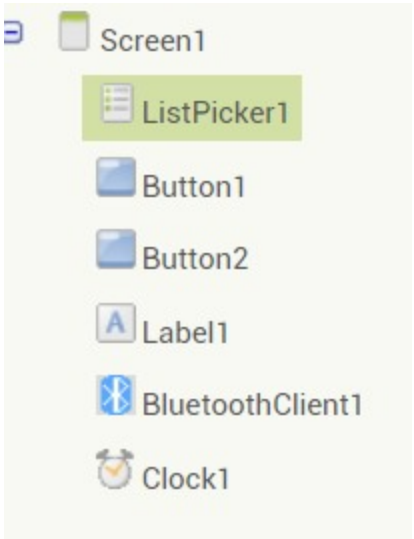

```
#include <WiFi.h>
#include <ThingSpeak.h>  // Ensure this library is installed

WiFiClient client;

const char* ssid = "ONLYCSE"; // Your Wi-Fi SSID
const char* password = "keepsmile"; // Your Wi-Fi password

const char* host = "api.thingspeak.com"; // API host
const char* privateKey = "UOZ3XFFHY6J345IO"; // Read API key
const char* privateKey1 = "9HO22G0EY0D8XYD4"; // Write API key

void setup() {
  Serial.begin(115200);
  pinMode(4, OUTPUT);  // Set LED as output

  // Start ThingSpeak client
  ThingSpeak.begin(client);
```

```
  // Connect to Wi-Fi
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  // Read the latest value from field 1 of your ThingSpeak chann
  int d = ThingSpeak.readIntField(626082, 1);  // Channel ID and

  Serial.print("Field value: ");
  Serial.println(d);

  if (d == 1) {
    digitalWrite(4, HIGH);  // Turn LED ON
    Serial.println("LED ON");
  } else {
    digitalWrite(4, LOW);   // Turn LED OFF
    Serial.println("LED OFF");
  }

  delay(1000); // Delay to avoid hitting the rate limit of Thing
}
```

11. led on/off using bluetooth (mo)








```
#include  "BluetoothSerial.h"
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENA
#error Bluetooth is not enabled! Please run make menuconfig to a
#endif
```

```
BluetoothSerial SerialBT;
String state;
void setup()
{
    pinMode(16, OUTPUT);
    Serial.begin(115200);
    SerialBT.begin("21b91a05b5"); // Bluetooth device name
    Serial.println("The device started, now you can pair it witl
}
void loop()
{
    if (Serial.available())
    {
        SerialBT.write(Serial.read());
        Serial.println("hello");
    }
    if (SerialBT.available())
    {
        state = SerialBT.read();
        Serial.print("State :");
        Serial.println(state);
        if (state.equals("49"))
        {
            digitalWrite(16, HIGH);
            Serial.println("Light On");
        }
        // if the state is 'LED1OFF' the led1 will turn off
        else if (state.equals("48"))
        {
            digitalWrite(16, LOW);
            Serial.println("Light Off");
        }
    }
    state = "";
```

```
    delay(200);
}
```