1. Configuring ESP32

Components:

- DHT11 Sensor: For temperature and humidity.
- BH1750 Sensor: For light intensity detection.
- Relay: To control the lamp.
- Connect Sensors DHT11, BH1750, Relay to ESP32

2. Developing Control Logic: Program Automation Tasks Based on Sensor Input

Automation for the lamp: The automation is based on the light intensity from the BH1750. When the light level is below a threshold, the ESP32 will turn on the lamp. When the light intensity is above this threshold, it turns off the lamp.

3. Creating a Mobile App with MIT App Inventor

- 1. Create a User Interface by adding buttons and labels.
- 2. Add BLE Component to allow BLE connection to the app.

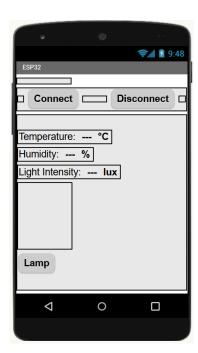
Application Logic:

1. Connect to the ESP32:

Scan for BLE devices and connect to the ESP32. When the ESP32 is connected, the app will show connected status.

2. Control Lamp:

When the Button for the lamp is pressed, sends a BLE command to change the lamp's state to ON or OFF



3. Display Sensor Data:

When BLE command is received from ESP32, display the current temperature, humidity, and light intensity.

MIT App Inventor blocks:

Make sure UUID match in code & in the app inventor or the app will not work!

```
initialize global Service_UUID to differential description of the service description of the
```

```
when LED v. Click

do if BluetoothLE2 v. IsDeviceConnected v. then if LED v. BackgroundColor v. to then set LED v. BackgroundColor v. to call BluetoothLE2 v. WriteStrings serviceUtid characteristicUtid utf16 values v. A v. else if LED v. BackgroundColor v. to call BluetoothLE2 v. WriteStrings serviceUtid characteristicUtid get global RX_char_UUID v. get global RX_char_U
```

```
when Clock1 * Time*

do 0 if BluetoothLE2 BeviceConnected
then call BluetoothLE2. RegisterForStrings
serviceUtid
characteristicUtid get global service_UtiD *
call BluetoothLE2 RegisterForStrings
serviceUtid
characteristicUtid get global service_UtiD *
call BluetoothLE2 RegisterForStrings
serviceUtid get global TX_char_UtiD2 *
call BluetoothLE2 RegisterForStrings
serviceUtid get global service_UtiD *
call BluetoothLE2 RegisterForStrings
serviceUtid get global TX_char_UtiD2 *
call BluetoothLE2 RegisterForStrings
serviceUtid get global
```

```
when EluctoothLE2* StringValues

do (i) if (i) compare texts (get enriceUtid) (ii) get (global service_UUID) (iii) (iii) (iii) (iii) (iii) (iiii) (iiiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiii) (iiiii) (iiii) (iiiii) (iiii) (iii
```

4. Cloud Integration

- 1. Link ESP32 to Arduino IoT Cloud.
- 2. Define variables for Temperature, Humidity, Light Intensity, and Lamp.
- 3. Use the Arduino IoT Cloud dashboard to control the lamp and monitor sensor data remotely.

