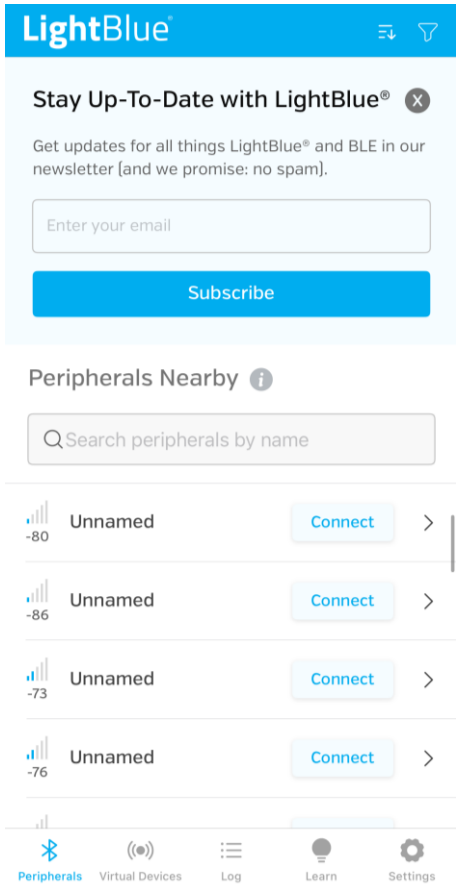
# How to Control LEDs and Buzzers with LightBlue

# Software Used

Please install the following applications on your smartphone.

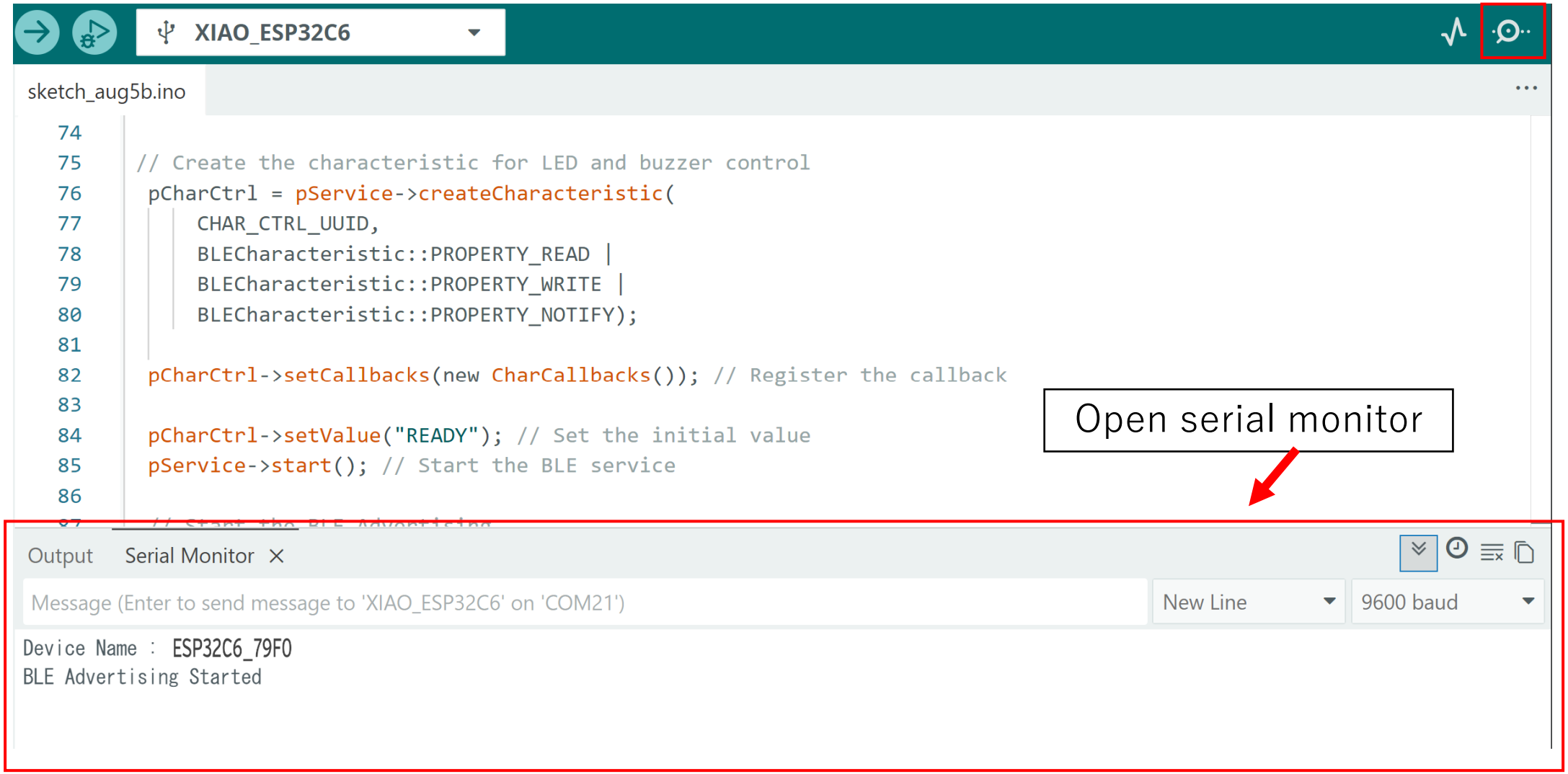
https://punchthrough.com/lightblue/

If this screen appears, you're good to go.



# How to Open Serial Monitor

Click here

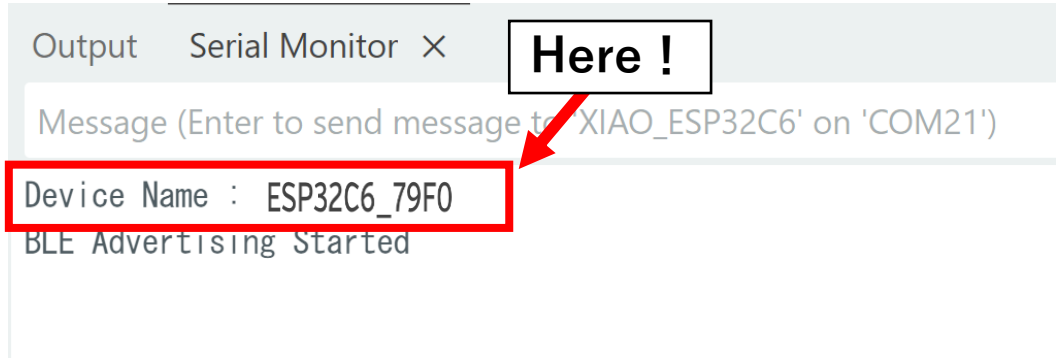


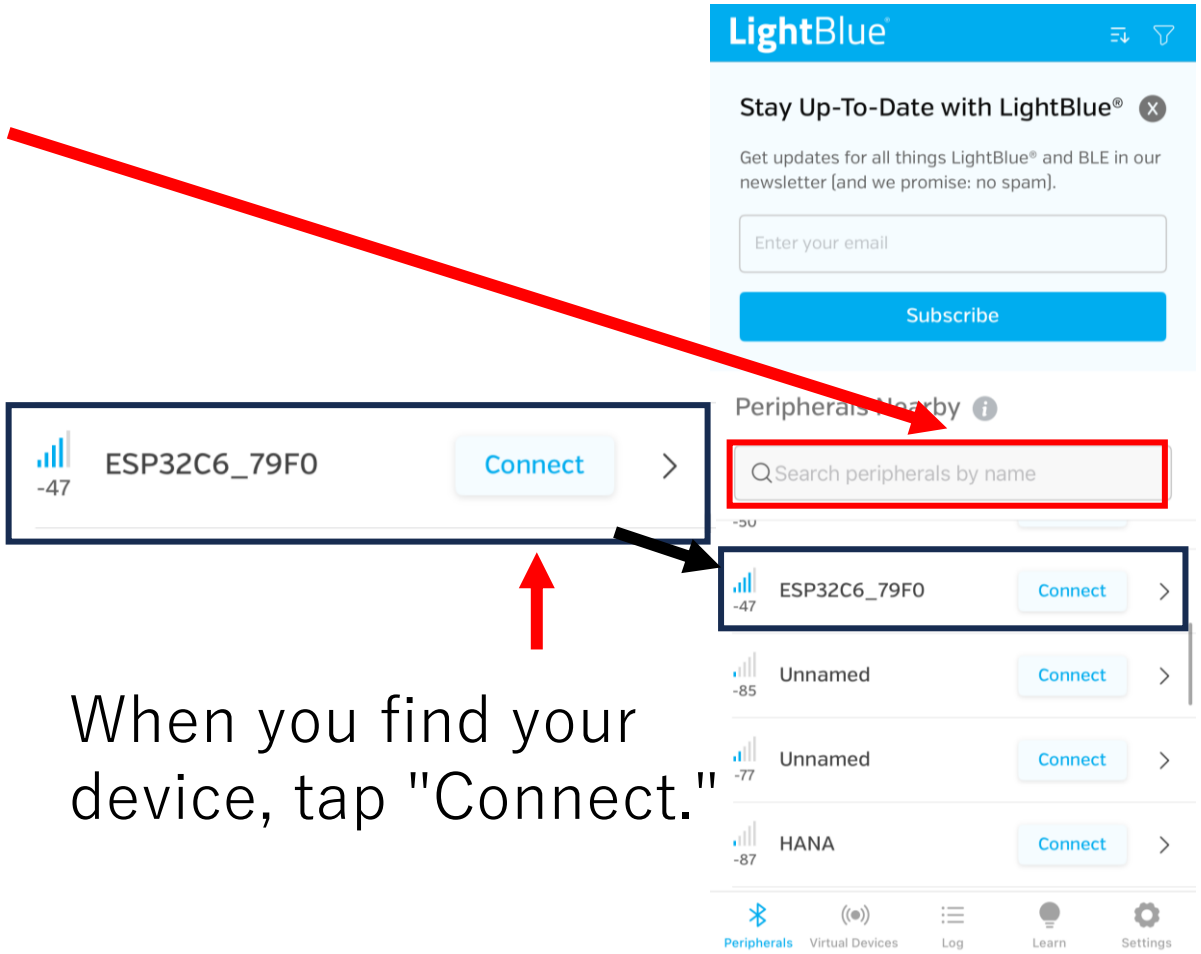
# How to Use the App

①Use the Serial Monitor to check the name of your device.

②Find your device name.

(Searching is also available.)

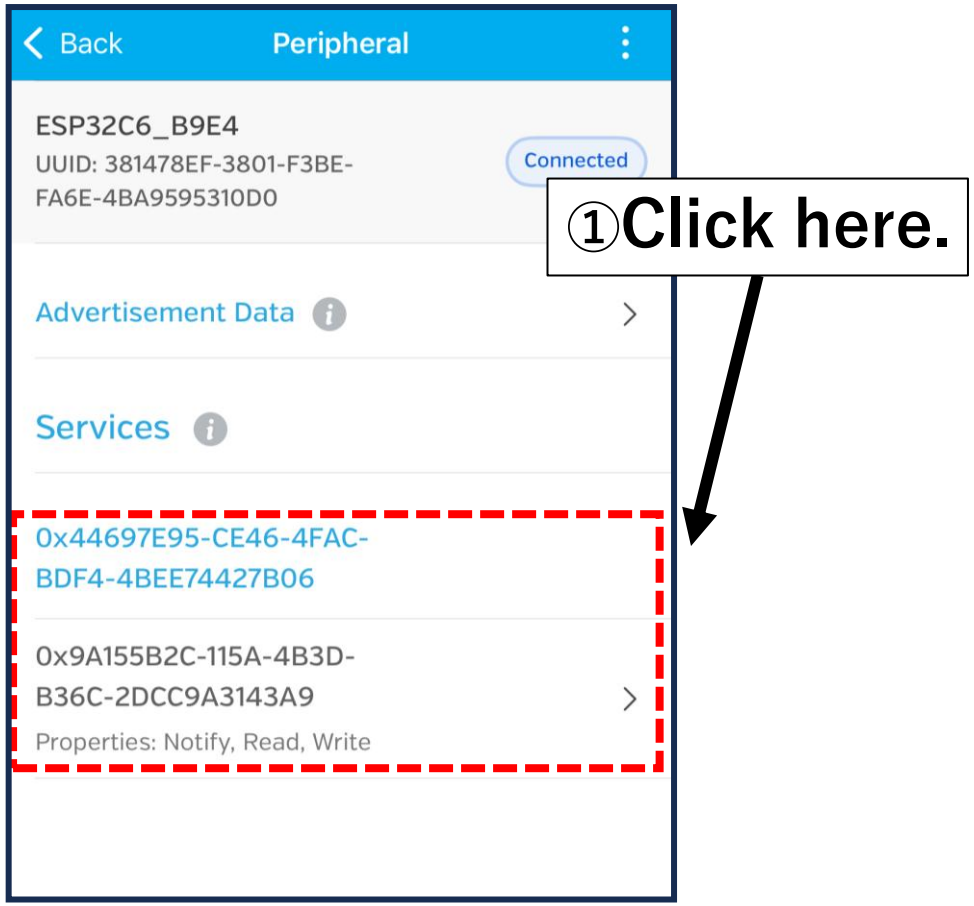


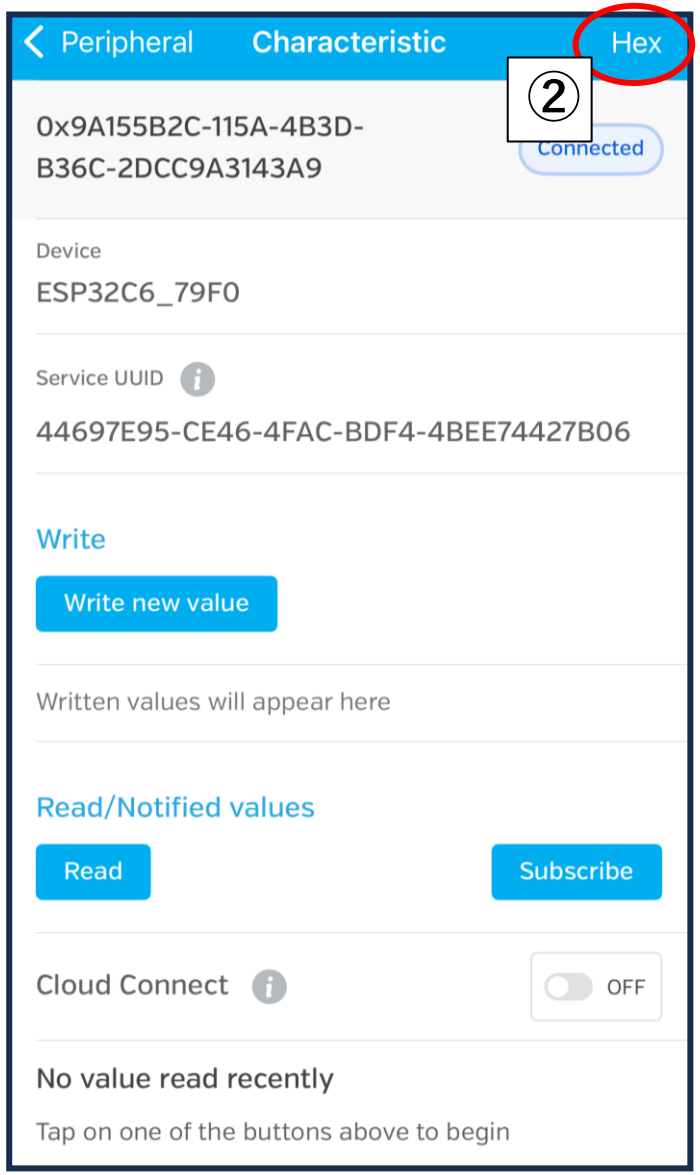



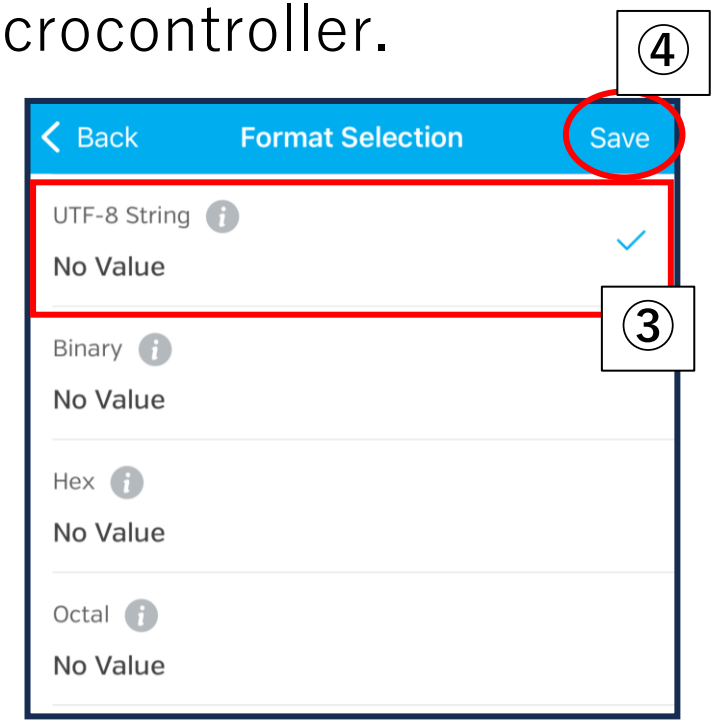

When you find your device, tap "Connect."

# Bluetooth Control of a Microcontroller !

Use a Bluetooth connection to send commands to the microcontroller.

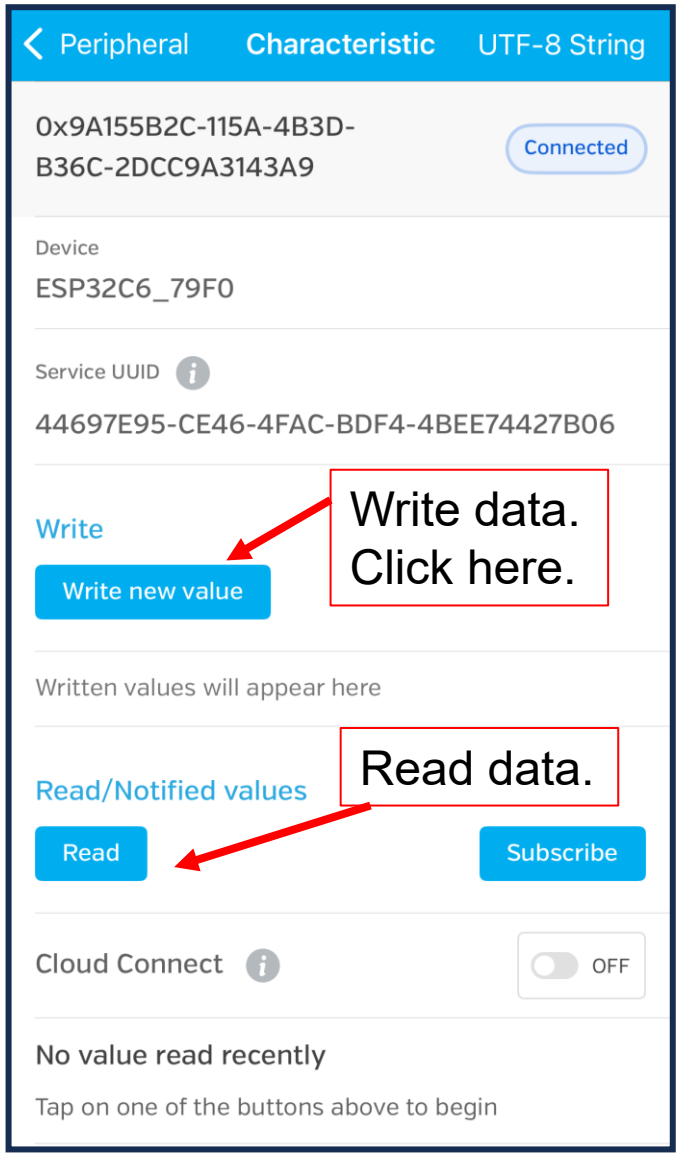


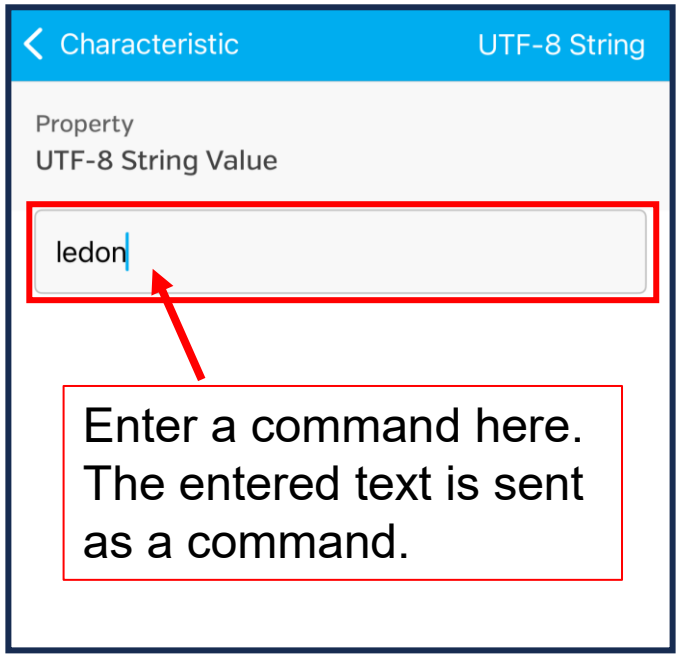




**Change the display settings.**
Click **"Hex"** in the upper-right corner, select **"UTF-8 String"**, and then click **"Save"**.

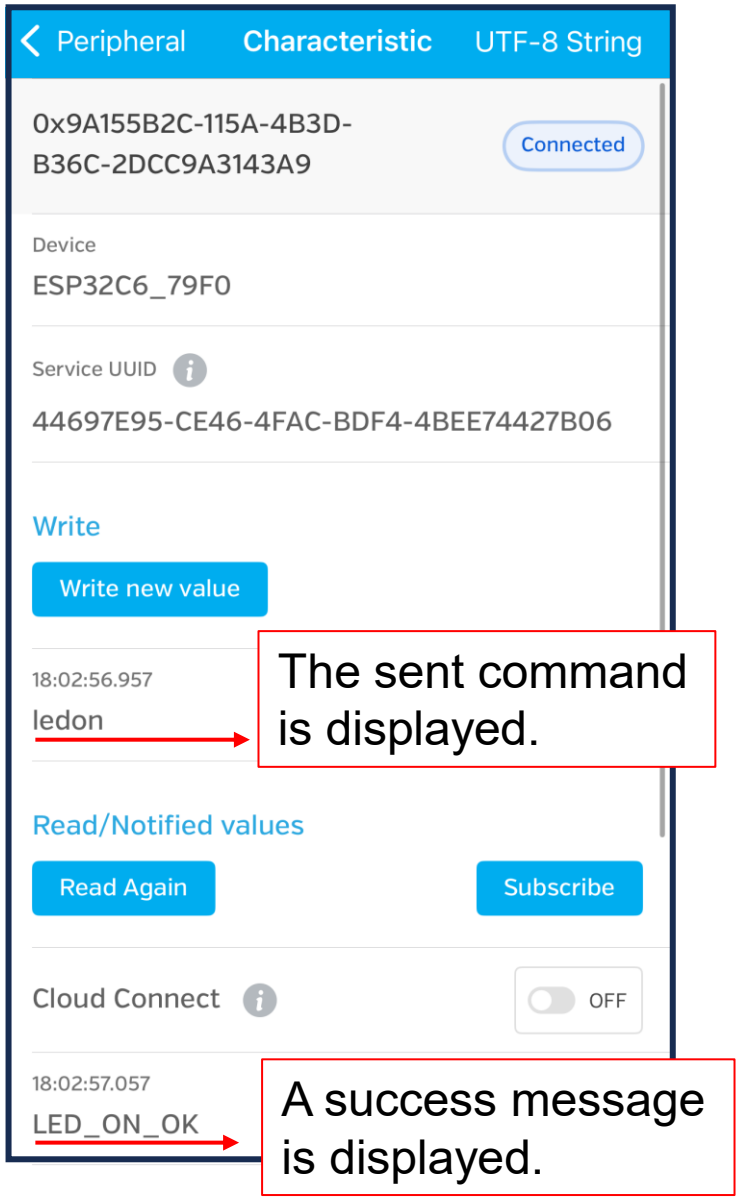
# Bluetooth Control of a Microcontroller !




| Command | Action |
|---|---|
| ledon | Turn on the LED |
| ledoff | Turn off the LED |
| bz | Sound the buzzer |

Uppercase and lowercase letters are both accepted.



# Program Details(Overall Structure)

○**Settings and Variables　[ #define / Global Variables ]**

Configure the LED and buzzer pin numbers and　communication UUIDs.

○**Receive Action Definition　[ class CharCallbacks ]**

Define the actions to perform when data is received from the smartphone (e.g., turn on the LED or activate the buzzer).

○**Initialization　[void setup()]**

Runs once at startup. After creating the device name, it initializes the BLE server, service, and characteristic, registers the data reception callback, and starts BLE advertising to enter the standby state.

# Program Details(Callback Function)

The callback function (CharCallbacks) is automatically executed whenever data is received from a BLE device, such as a smartphone.

```
class CharCallbacks : public BLECharacteristicCallbacks {
  void onWrite(BLECharacteristic *pChar) override {
    String value = pChar->getValue();
    value.toLowerCase();
    Serial.print("Received: ");
    Serial.println(value);
    if (value == "ledon") {
      digitalWrite(LED_PIN, HIGH);
      // Notify the execution result
      pChar->setValue("LED_ON_OK");
      pChar->notify();
    }
  }
};
```

←"onWrite" is called when data is received.

←Read the received data as a string.

←Accept both uppercase and lowercase commands.

←Display the command on the Serial Monitor.

←If the command is "ledon", turn the LED on (HIGH).

←Send "LED_ON_OK" to the smartphone.

←Notify the smartphone that the command was executed.


**Command received!**
**Run the callback function!**
**Turn on the LED!**