VAR2053
Electronic Signal and Communication Fundamental
電子訊號與通訊基礎

Course Outline
課程大綱

- 軟件開發工具介紹
- 解釋ESP32 藍牙固件結構和功能

Software Development
軟件開發

Software

Applications (Apps)

Firmware

藍牙技術

- 藍牙(英語：Bluetooth) 是一種無線通訊技術標準
  - 使用2.4 GHz至2.485 GHz的頻段來進行通訊
- 它最初的設計是希望建立一個序列資料通訊數據線的無線替代版本
- 用來讓固定與行動裝置，在短距離間交換資料，以形成個人區域網路(PAN-Personal Area Network)
- 它能夠同時連結多個裝置，克服同步的問題

藍牙技術 - 應用

- **汽車**
  - 藍牙免提呼叫系統；車載音訊娛樂系統；監測和診斷機電系統
- **醫療和保健**
  - 血糖監測儀、脈搏血氧儀、心率監視器、哮喘吸入器等產品
- **可穿戴裝置**
  - 智慧型眼鏡、耳機、活動監測儀、兒童和寵物監視器、醫療救助、頭部和手部安裝終端以及攝錄影機

藍牙技術 - 應用

- 消費類電子產品
  - 電視和遊戲系統，家用遊戲機的手柄，包括PS4、PSP Go、 Wii、Switch
- 手機
  - 移動電話和免提裝置之間的無線通訊，這也是最初流行的應用。
- **運動和健身**
  - 健身跟蹤手環和智慧型手錶，瑜伽墊、棒球棍等

藍牙技術 - 應用

- 家居自動化
  - 智慧型家居，室內的照明、溫度、家用電器、窗戶和門鎖等安全系統以及牙刷、鞋墊等日常用品。
- 電腦與周邊設備
  - 滑鼠、鍵盤、耳機、印表機等
- 零售和位置導向式服務
  - 即時定位系統（RTLS），應用"節點"或"標籤"嵌入被跟蹤物品中讀卡機從標籤接收並處理無線訊號以確定物品位置。

藍牙技術

Classic Bluetooth

Bluetooth Low Energy

無線設備傳輸豐富的內容，例如音頻

會與這兩種類型連接的設備

無線設備使用非常少的能量發送少量數據

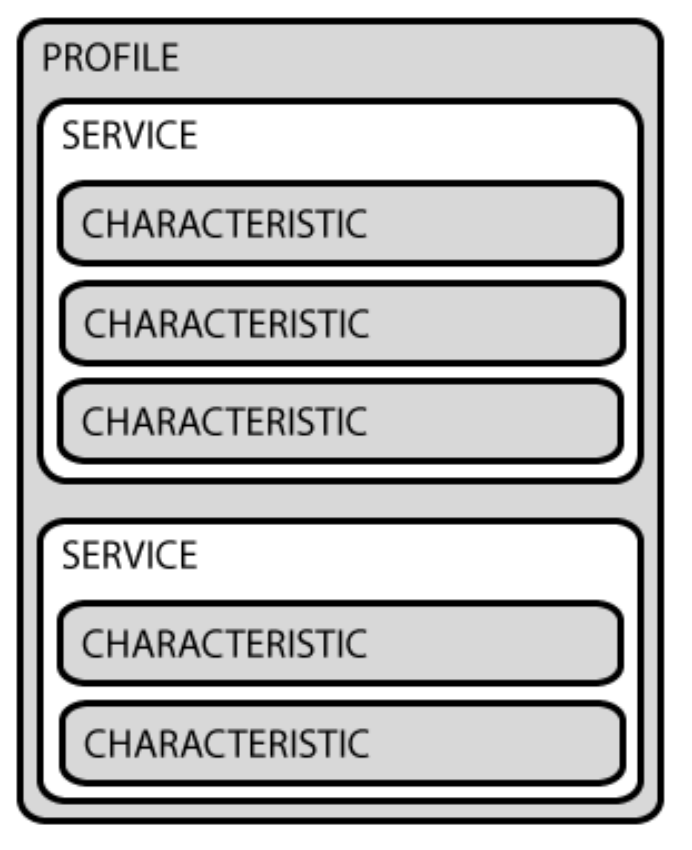
Specifications	Classic Bluetooth	Bluetooth Low Energy (BLE)
Range	100 m	Greater than 100 m
Data Rate	1-3 Mbps	1 Mbps
Application Throughput	0.7 -2.1 Mbps	0.27 Mbps
Frequency	2.4 GHz	2.4 GHz
Security	56/128-bit	128-bit AES with Counter Mode CBC-MAC
Robustness	Adaptive fast frequency hopping, FEC, fast ASK	24-bit CRC, 32-bit Message Integrity Check
Latency	100 ms	6 ms
Time Lag	100 ms	3 ms
Voice Capable	Yes	No
Network Topology	Star	Star
Power Consumption	1 W	0.01 to 0.5 W
Peak Current Consumption	less than 30 mA	less than 15 mA

藍牙技術 - 基本傳輸資料方式

- Android
  - 支持藍牙 Serial Port Profile(SPP)序列資料通信 (Bluetooth Classic)
  - 支持藍牙Generic Attribute Profile (GATT) 通用屬性通信 (Bluetooth Low Energy)
- iOS (iPhone / iPad)
  - 支持藍牙Generic Attribute Profile (GATT) 通用屬性通信 (Bluetooth Low Energy)

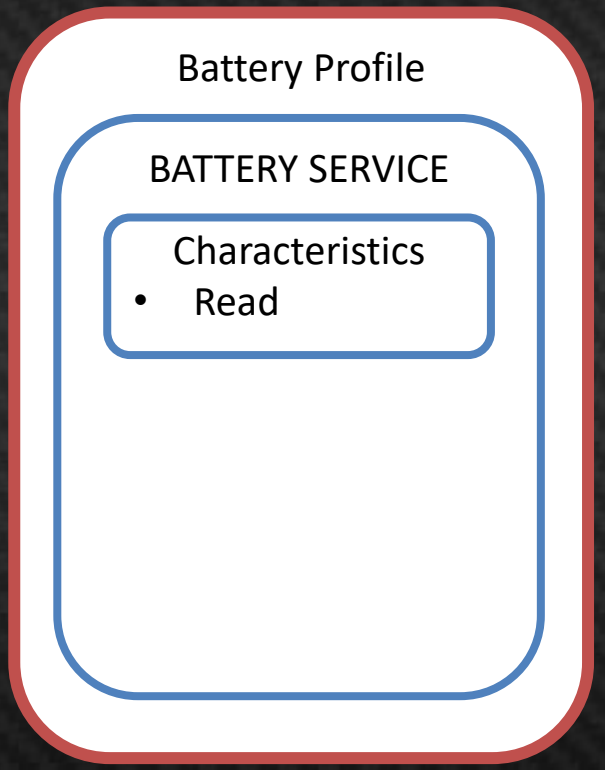
Bluetooth Low Energy - GATT

- GATT (Generic Attribute Profile) 的資料型態
- 由三層資料結構組成
  - Profile
  - Service
  - Characteristic



Bluetooth Low Energy - GATT

- 以百分比形式發送當前設備電池電量
- 允許的範圍是 0 到 100
  - 數值0表示 0% 電池電量
  - 數值100表示 100% 電池電量

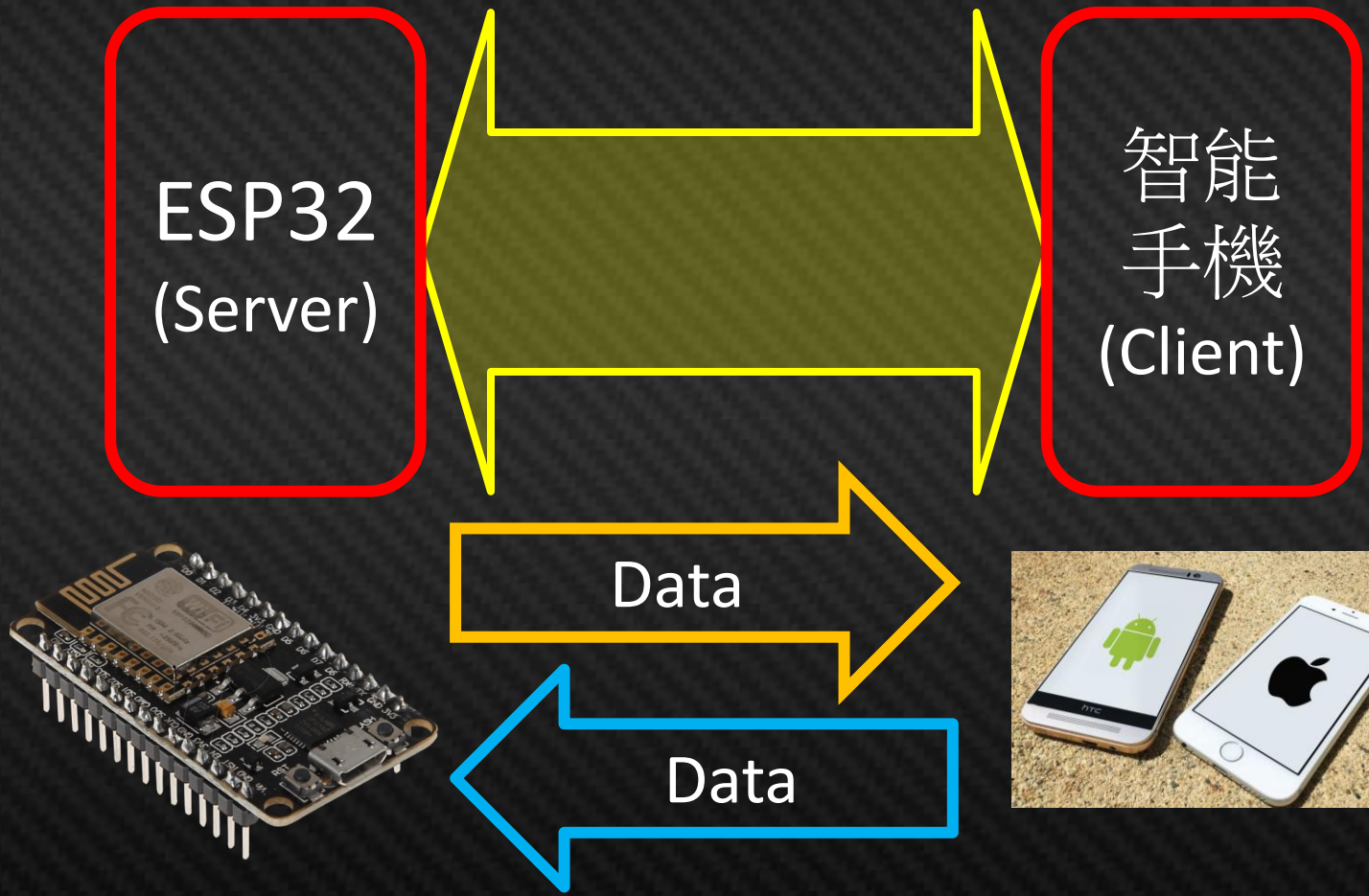


Bluetooth Low Energy - GATT

- Characteristic 可以使用 不同方式交換資料
  - Broadcast (播送)
  - Read (閱讀)
  - Write without response (寫入但不需要回應)
  - Write (寫入)
  - Notify (通知)
  - Indicate (表明)
  - Authenticated Signed Writes (經過身份驗證的寫入)
  - Extended Properties (擴展屬性)

ESP32藍牙固件解構

ESP32固件解構
連接藍牙

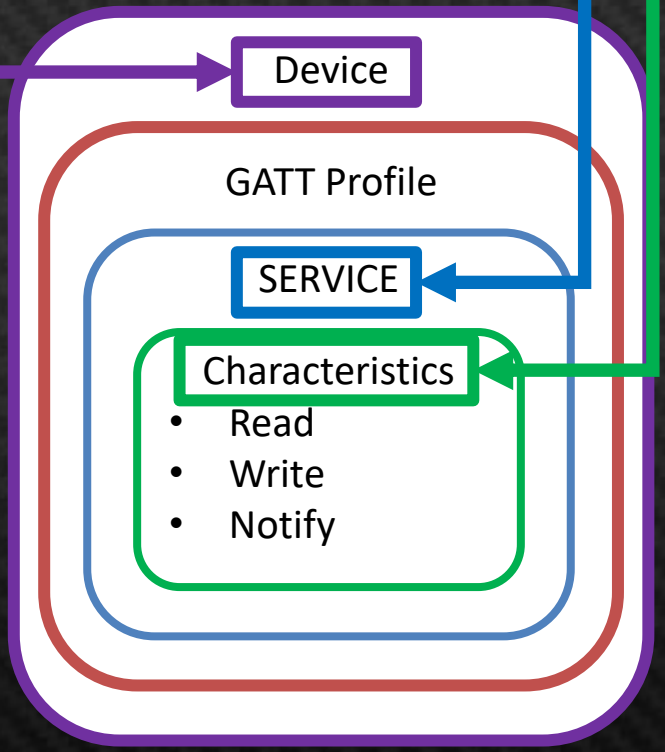


在Arduino IDE中運用藍牙

在Arduino IDE中運用藍牙

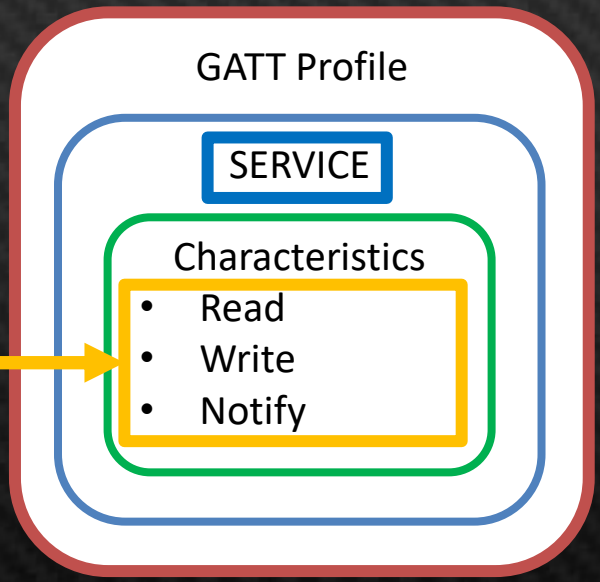
```
#define SERVICE_UUID        "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
#define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
```

```
// Create the BLE Device
BLEDevice::init("SAM CHOY - 04916353");
```

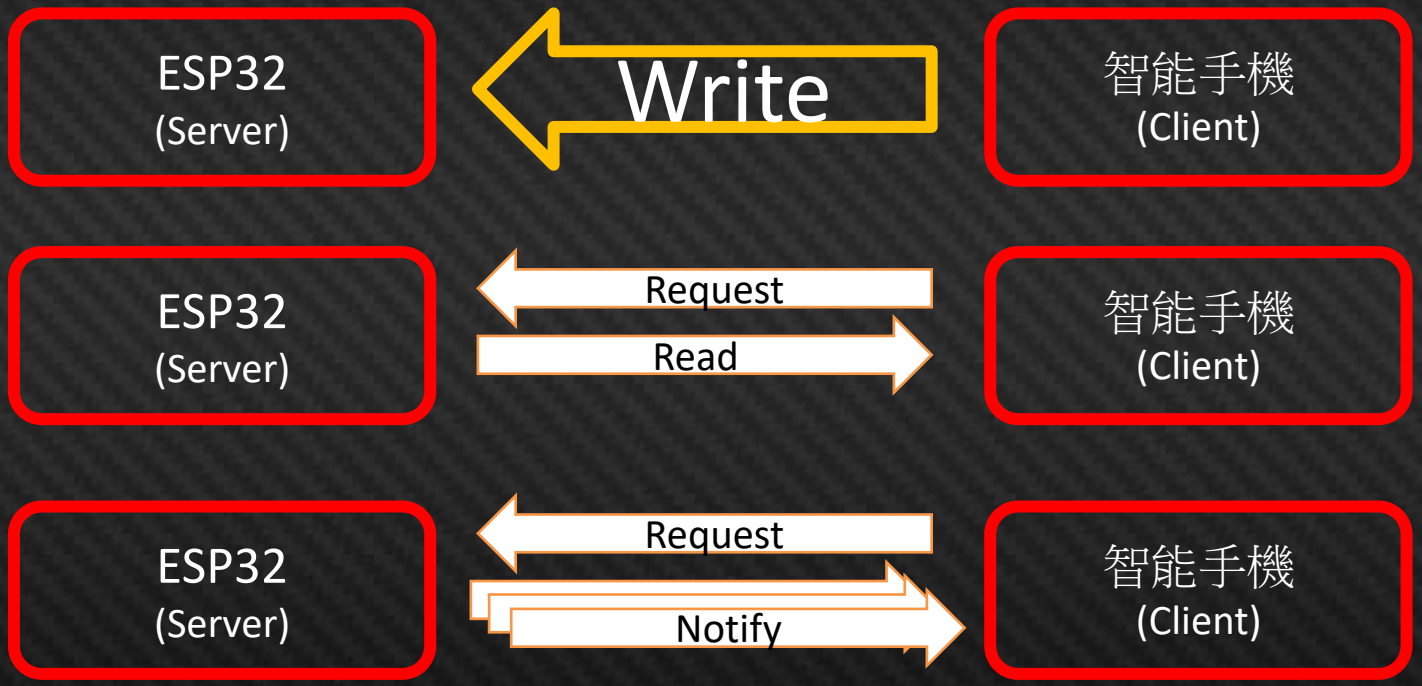


在Arduino IDE中運用藍牙

```
// Create a BLE Characteristic
pCharacteristic = pService->createCharacteristic(
                                        CHARACTERISTIC_UUID,
                                        BLECharacteristic::PROPERTY_READ |
                                        BLECharacteristic::PROPERTY_WRITE |
                                        BLECharacteristic::PROPERTY_NOTIFY
                                      );
```



在Arduino IDE中運用藍牙

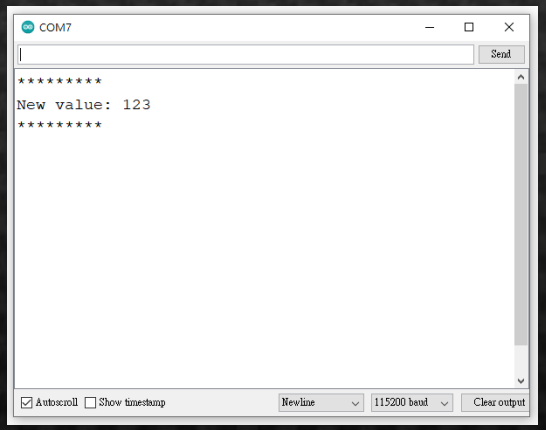


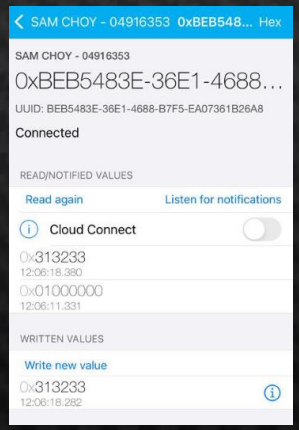
```
// ==========BLE Characteristic status Call back function==========
class MyCallbacks: public BLECharacteristicCallbacks {
    void onWrite(BLECharacteristic *pCharacteristic) {
      std::string value = pCharacteristic->getValue();
      if (value.length() > 0) {
        Serial.print("Incoming Data: ");
        for (int i = 0; i < value.length(); i++)
          Serial.print(value[i]);
        Serial.println();
      }
    }
};
```

ESP32
(Server)

Write

智能手機
(Client)





```
void loop() {
    // ==========notify changed value==========
    // connecting --> record connected device ID
    if (deviceConnected) {
        pCharacteristic->setValue((uint8_t*)&value, 4);
        pCharacteristic->notify();
        value++;
        delay(1000);
    }
    // disconnecting --> restart advertising
    if (!deviceConnected && oldDeviceConnected) {
        delay(500); // Bluetooth stack get things ready time
        pServer->startAdvertising(); // restart advertising
        Serial.println("disconnected and start advertising again");
        oldDeviceConnected = deviceConnected;
    }
    // connecting --> record connected device ID
    if (deviceConnected && !oldDeviceConnected) {
        // do stuff here on connecting
        oldDeviceConnected = deviceConnected;
    }
}
```

通知數據

斷線處理

連接處理

ESP32
(Server)

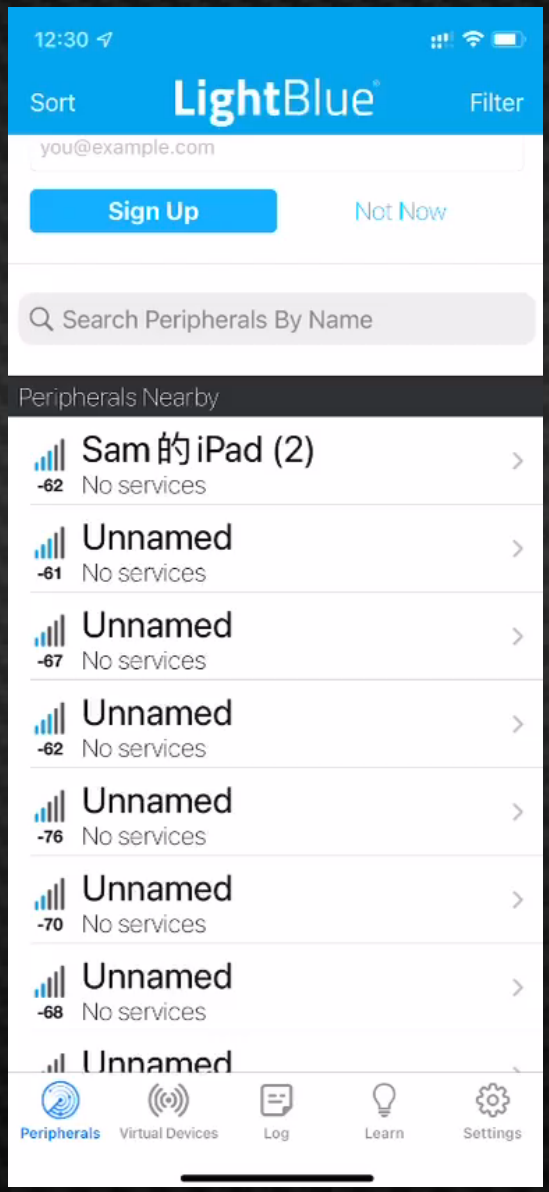
Notify

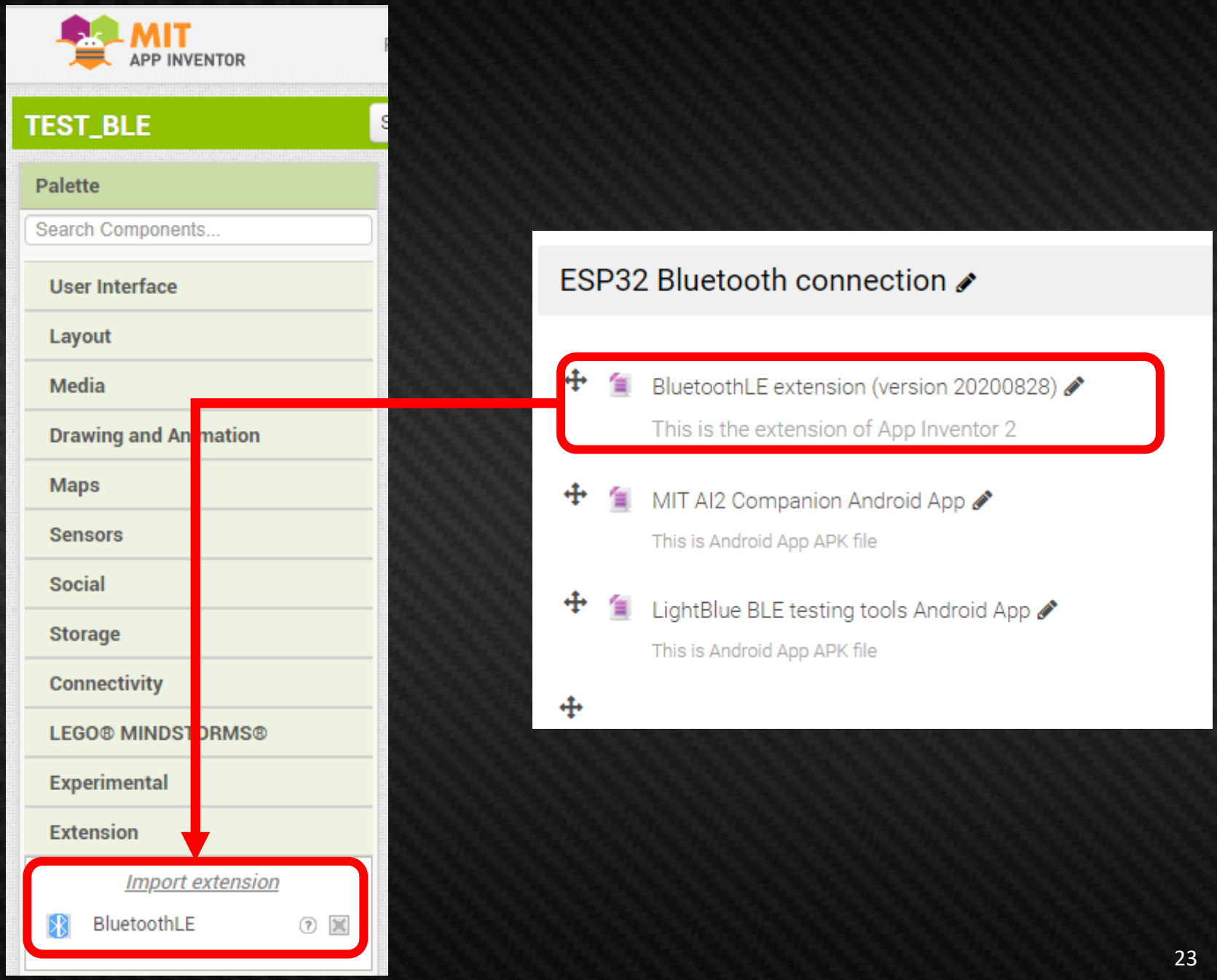
智能手機
(Client)

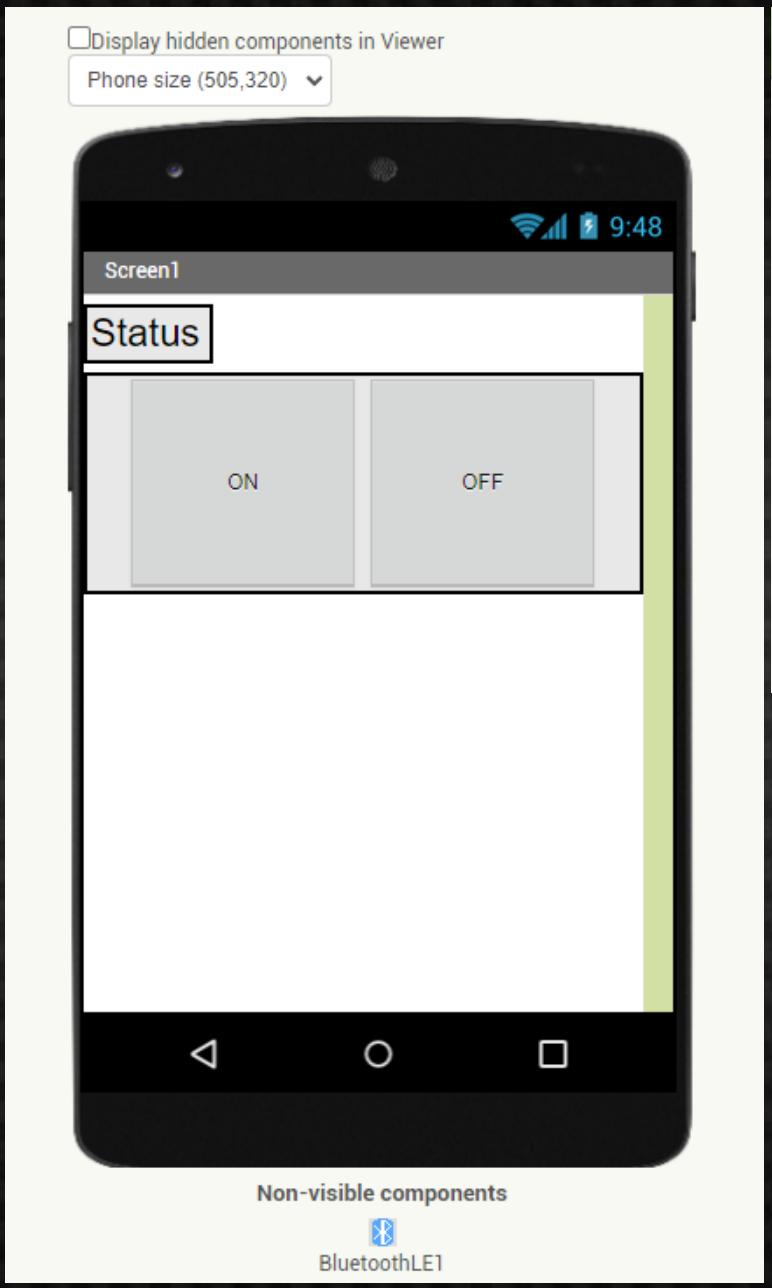
使用 LightBlue App 測試 BLE 設備

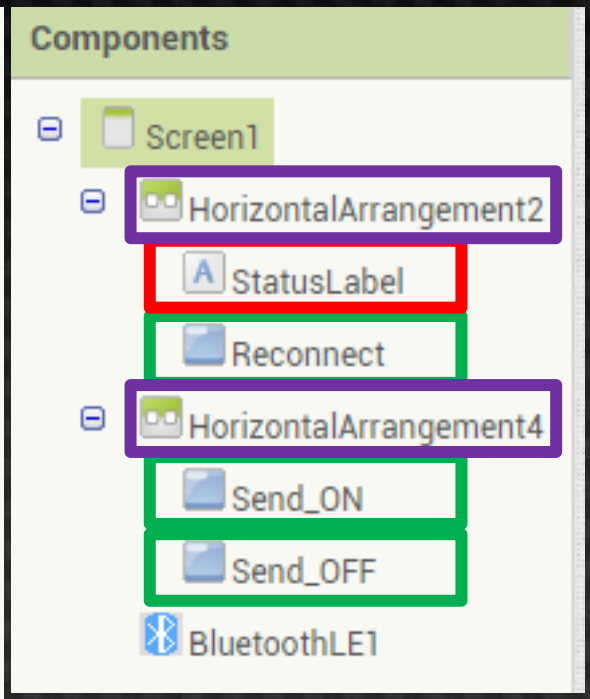
Light Blue (iOS)
BLE testing Tool

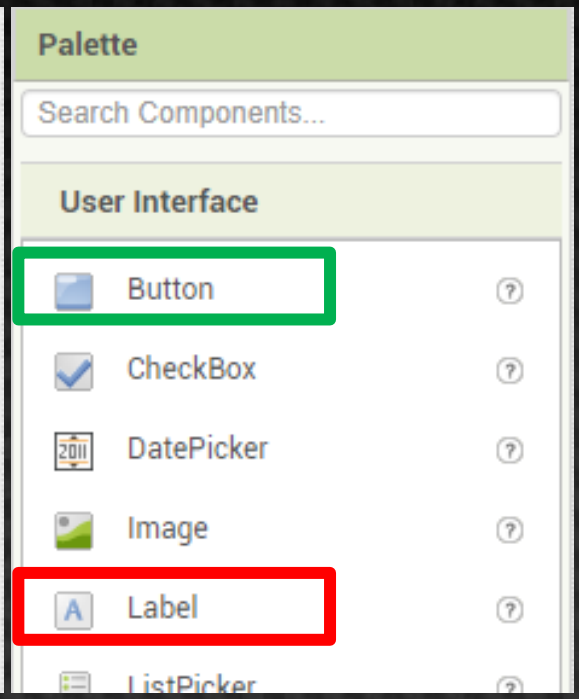
Light Blue
BLE testing Tool

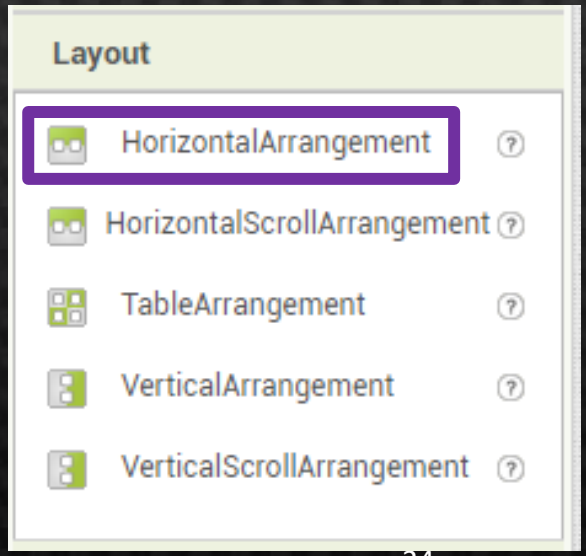


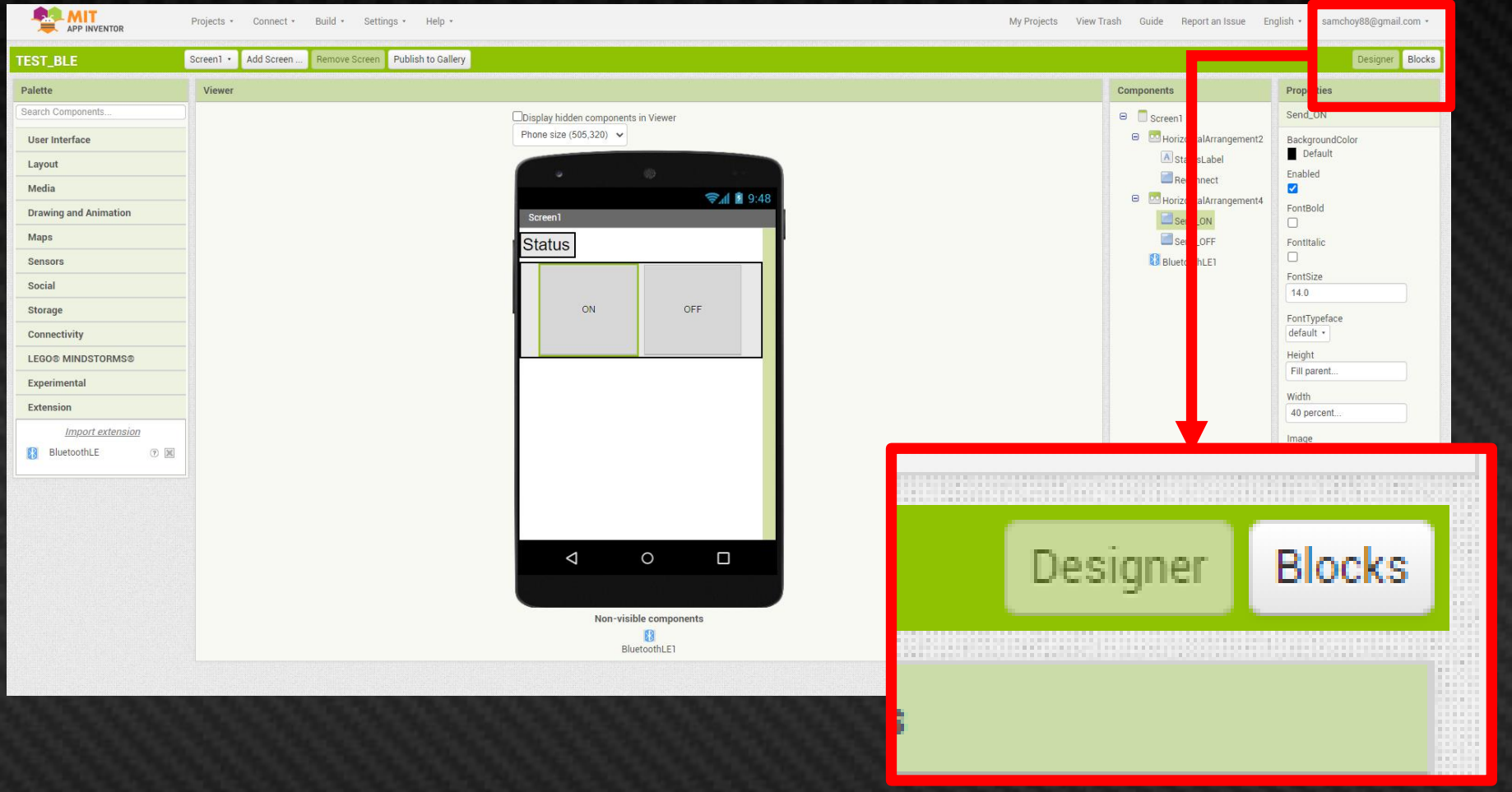

MIT
APP INVENTOR
TEST_BLE
Palette
Search Components...
User Interface
Layout
Media
Drawing and Animation
Maps
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Experimental
Extension
Import extension
BluetoothLE
ESP32 Bluetooth connection
BluetoothLE extension (version 20200828)
This is the extension of App Inventor 2
MIT AI2 Companion Android App
This is Android App APK file
LightBlue BLE testing tools Android App
This is Android App APK file

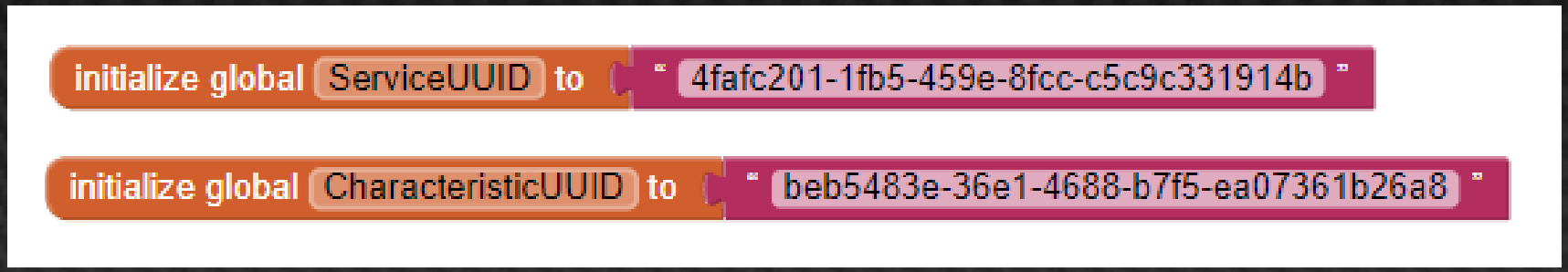

Display hidden components in Viewer
Phone size (505,320)
9:48
Screen1
Status
ON
OFF
Non-visible components
BluetoothLE1


Components
Screen1
HorizontalArrangement2
StatusLabel
Reconnect
HorizontalArrangement4
Send_ON
Send_OFF
BluetoothLE1

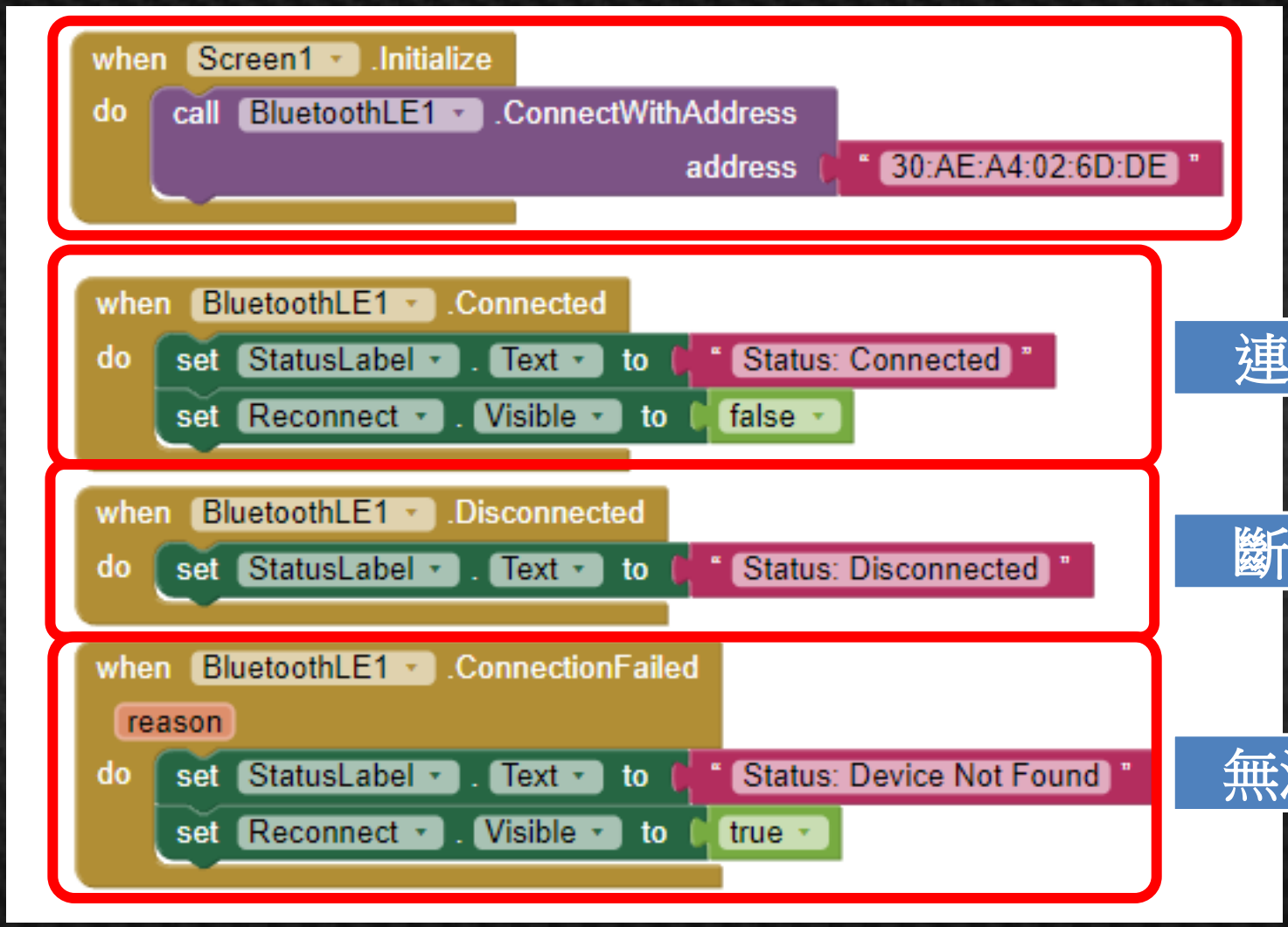

Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker


Layout
HorizontalArrangement
HorizontalScrollArrangement
TableArrangement
VerticalArrangement
VerticalScrollArrangement


MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
samchoy88@gmail.com
TEST_BLE
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Layout
Media
Drawing and Animation
Maps
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Experimental
Extension
Import extension
BluetoothLE
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen1
Status
ON
OFF
Non-visible components
BluetoothLE1
Components
Screen1
Properties
Send_ON
BackgroundColor
Default
Enabled
FontBold
FontItalic
FontSize
14.0
FontTypeface
default
Height
Fill parent...
Width
40 percent...
Image
Designer
Blocks



```
#define SERVICE_UUID        "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
#define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
```


when Screen1 .Initialize
do call BluetoothLE1 .ConnectWithAddress
address " 30:AE:A4:02:6D:DE "
when BluetoothLE1 .Connected
do set StatusLabel . Text to " Status: Connected "
set Reconnect . Visible to false
when BluetoothLE1 .Disconnected
do set StatusLabel . Text to " Status: Disconnected "
when BluetoothLE1 .ConnectionFailed
reason
do set StatusLabel . Text to " Status: Device Not Found "
set Reconnect . Visible to true
連接處理
斷線處理
無法連接處理

重新連接處理

按鈕 A
按下處理

按鈕 B
按下處理