

Gainscha Android Bluetooth/Ethernet/USB 函式庫使用說明

1. GTSPL_openPort()

★ Bluetooth

GTSPL_openPort(MacAddress)

- 函式說明：指定藍芽的 MAC 位址(請使用 BR/EDR MacAddress)，開啟輸出埠
- 參數說明：
 - ➔ address：字串型別，指定連線的藍芽位址(Bluetooth MacAddress)，
如："DC:1D:30:00:1D:87"

★ Ethernet

GTSPL_openPort(IP,Port)

- 函式說明：指定印表機的 IP 位址與連接埠，開啟輸出埠
- 參數說明：
 - ➔ IP：字串型別，指定連線的 IP 位址，如："192.168.1.109"
 - ➔ Port：int 型別，指定連線的連接埠如：8899

★ USB

GTSPL_openPort(manager,device)

- 函式說明：指定 USB 相關變數，開啟輸出埠
- 參數說明：
 - ➔ manager：USBManager 型別，輸入 USBManager 變數
 - ➔ device：USBDevice 型別，輸入 USBDevice 變數

2. GTSPL_closePort()

- 函式說明：關閉輸出埠
- 參數說明：無

3. GTSPL_closePort(time)

■ 函式說明：關閉輸出埠

■ 參數說明：

➔ time：int 型別，延遲時間，1000=1 秒，範例：GTSPL_closePort(2000)

4. GTSPL_setCmdSendMode(mode)

■ 函式說明：設定命令傳送至印表機或檔案

■ 參數說明：

➔ mode：字串型別

F:將命令傳送至檔案

(檔案位置在內部儲存空間/android/data/packageName/files 下)

P:將命令傳送至印表機

5. GTSPL_setup(width, height, speed, density, sensor, sensorDistance, sensorOffset, context)

■ 函式說明：設定標籤的寬度、高度、列印速度、列印熱度、感應器類別、間隙/黑標垂直間距、間隙/黑標偏移距離

■ 參數說明：

參數	型別	說明
width	int	設定標籤寬度，單位 mm
height	int	設定標籤高度，單位 mm
speed	int	設定列印速度，1~15，代表每秒 1~15 吋列印速度(隨機型不同會有不同列印最高上限，最高為每秒 15 吋列印速度)
density	int	設定列印濃度，0~15，數字越大列印結果越黑
sensor	int	設定使用感應器之類別； 0：表示使用間隙感測器(gap sensor) 1：表示使用黑標感測器(black mark sensor)
sensorDistance	int	設定間隙/黑標垂直間距高度，單位 mm
sensorOffset	int	設定間隙/黑標垂直間距高度，單位 mm，此參數若使用一般標籤時均設為 0
context	Context	帶入當前畫面的 Context

6. GTSPL_clearBuffer(context)

■ 函式說明：清除圖像緩衝

■ 參數說明：

➔ context：Context 類別，帶入當前畫面的 Context

7. GTSPL_barcode(x, y, type, height, readable, rotation, narrow, wide, content, context)

■ 函式說明：使用印表機內建條碼列印

■ 參數說明：

參數	型別	說明
x	int	條碼 X 方向起始點，以點(dot)表示
y	int	條碼 Y 方向起始點，以點(dot)表示
type	字串	設定條碼類型(Code Type) ， 請參考附件
height	int	設定條碼高度，高度以點來表示
readable	int	設定是否列印條碼碼文 0:不列印 1:列印
rotation	int	設定條碼旋轉角度 0：旋轉0度 90：旋轉90度 180：旋轉180度 270：旋轉270度
narrow	int	設定條碼窄 bar 比例因子， 請參考附件
wide	int	設定條碼寬 bar 比例因子， 請參考附件
content	字串	設定欲列印之條碼內容
context	Context	帶入當前畫面的 Context

8. GTSPL_formFeed(context)

■ 函式說明：跳頁，該函式需在 setup 後使用

■ 參數說明：

➔ context：Context 類別，帶入當前畫面的 Context

9. GTSPL_noBackFeed(context)

■ 函式說明：設定紙張不回吐

■ 參數說明：

➔ context：Context 類別，帶入當前畫面的 Context

10. GTSPL_sendCommand (context, command)

■ 函式說明：送內建指令到印表機

■ 參數說明：

➔ command：字串型別，設定指令內容，詳細指令請參考 TSPL 使用手冊

➔ context：Context 類別，帶入當前畫面的 Context

11. GTSPL_printerFont(x, y, size, rotation, x_scale, y_scale, content, context)

■ 函式說明：使用印表機內建文字列印

■ 參數說明：

參數	型別	說明
x	int	文字 X 方向起始點，以點(dot)表示
y	int	文字 Y 方向起始點，以點(dot)表示
size	字串	內建字型名稱，共五種 1: 8*12 dots 2: 12*20 dots 3: 16*24 dots 4: 24*32 dots 5: 32*48 dots TST24.BF2: 繁體中文24*24 TSS24.BF2: 簡體中文24*24
rotation	int	設定文字旋轉角度 0：旋轉0度 90：旋轉90度 180：旋轉180度 270：旋轉 270 度
x_scale	int	設定文字 X 方向放大倍率，1~8
y_scale	int	設定文字Y方向放大倍率，1~8

content	字串	設定欲列印之文字內容
context	Context	帶入當前畫面的 Context

12. GTSPL_printLabel(set, copy, context)

- 函式說明：列印標籤內容
- 參數說明：
 - ➔ set：int 型別，設定列印標籤式數(set)
 - ➔ copy：int 型別，設定列印標籤份數(copy)
 - ➔ context：Context 類別，帶入當前畫面的 Context

13. GTSPL_downloadPCX(filename, context)

- 函式說明：下載單色 PCX 格式圖檔至印表機
- 參數說明：
 - ➔ filename：字串型別，檔案名稱
(檔案需存在內部儲存空間/android/data/packageName/files 資料夾下)
 - ➔ context：Context 類別，帶入當前畫面的 Context

14. GTSPL_downloadBMP(filename, context)

- 函式說明：下載單色 BMP 格式圖檔至印表機
- 參數說明：
 - ➔ filename：字串型別，檔案名稱
(檔案需存在內部儲存空間/android/data/packageName/files 資料夾下)
 - ➔ context：Context 類別，帶入當前畫面的 Context

15. GTSPL_downloadTTF(filename, context)

■ 函式說明：下載 True Type Font 字型至印表機

■ 參數說明：

➔ filename：字串型別，檔案名稱

(檔案需存在內部儲存空間/android/data/packageName/files 資料夾下)

➔ context：Context 類別，帶入當前畫面的 Context

16. GTSPL_printerStatus(delaytime)

■ 函式說明：回傳印表機狀態，需用字串變數接收回傳訊息

■ 參數說明：

➔ delaytime：int 型別，設定延遲時間

■ 回傳字串說明：

回傳字串	印表機狀態
00	就緒
01	上蓋開啟
02	卡紙
03	卡紙且上蓋開啟
04	標籤用盡
05	標籤用盡且上蓋開啟
08	碳帶用盡
09	碳帶用盡且上蓋開啟
0A	碳帶用盡且卡紙
0B	碳帶用盡、卡紙且上蓋開啟
0C	碳帶用盡且標籤用盡
0D	碳帶用盡、標籤用盡且上蓋開啟
10	暫停
20	列印中
80	其他錯誤

17. GTSPL_getSDKVersion (returnWay, context)

- 函式說明：回傳此 SDK 版本號
- 參數說明：
 - ➔ returnWay：int 型別，輸入 0 除返回 SDK 版本號外，會跳出 SDK 版本訊息
 - ➔ context：Context 類別，帶入當前畫面的 Context

Gainscha Android Bluetooth 範例說明

1.需先於 AndroidManifest.xml 設定下列權限：

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />  
  
<uses-permission android:name="android.permission.BLUETOOTH" />
```

2.匯入 GTSPL_SDK：

```
import com.gainscha.gtspl_sdk.GTSPLActivity;
```

3.範例程式：

```
public class MainActivity extends AppCompatActivity {  
  
    GTSPLActivity mGtsplCmdTest = new GTSPLActivity();  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_main);  
  
        mGtsplCmdTest.GTSPL_setCmdSendMode ("P");  
  
        mGtsplCmdTest.GTSPL_openPort ("DC:1D:30:00:1D:87");  
  
        mGtsplCmdTest.GTSPL_setup(62, 45, 2, 6, 0, 3, 0, this);  
  
        mGtsplCmdTest.GTSPL_sendCommand(this, "DIRECTION 1\n\n");  
  
        mGtsplCmdTest.GTSPL_clearBuffer(this);  
  
        mGtsplCmdTest.GTSPL_printerFont(100, 100, "3", 0, 1, 1, "Print Font 123456", this);  
  
        mGtsplCmdTest.GTSPL_barcode(30, 30, "128", 100, 1, 0, 2, 2, "barcode1234567", this);  
  
        mGtsplCmdTest.GTSPL_downloadBMP("CIRCLE.BMP", this);  
  
        mGtsplCmdTest.GTSPL_sendCommand(this, "PUTBMP 150,30,\"CIRCLE.BMP\"\r\n");  
  
        mGtsplCmdTest.GTSPL_printLabel(1, 1, this);  
  
        String status = mGtsplCmdTest.GTSPL_printersStatus(1000);  
  
        //簡中打印  
  
        String stString="默认简体中文测试";  
  
        mGtsplCmdTest.GTSPL_clearBuffer(this);  
  
        mGtsplCmdTest.GTSPL_printerFont(100, 10, "TSS24.BF2", 0, 1, 1, stString, this);  
  
    }  
}
```



```
mGtsplCmdTest.GTSPL_printLabel(1, 1, this);  
  
//繁中打印  
String ttString="默認繁體中文測試";  
  
mGtsplCmdTest.GTSPL_clearBuffer(this);  
  
mGtsplCmdTest.GTSPL_printerFont(100, 10, " TST24.BF2", 0, 1, 1, ttString, this);  
  
mGtsplCmdTest.GTSPL_printLabel(1, 1, this);  
  
mGtsplCmdTest.GTSPL_closePort(1000);  
  
String version= mGtsplCmdTest .GTSPL_getSDKVersion(0,this);  
  
}  
  
}
```

Gainscha Android Ethernet 範例說明

1.需先於 AndroidManifest.xml 設定下列權限：

```
<uses-permission android:name="android.permission.INTERNET"/>
```

2.匯入 GTSP_SDK：

```
import com.gainscha.gtspl_sdk.GTSPLWIFIActivity;
```

3.範例程式：

```
public class MainActivity extends AppCompatActivity {  
  
    GTSPLWIFIActivity mGtsplWIFICmdTest = new GTSPLWIFIActivity();  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_main);  
  
        mGtsplWIFICmdTest.GTSPL_setCmdSendMode("P");  
  
        mGtsplWIFICmdTest.GTSPL_openPort("192.168.1.109",8899);  
  
        mGtsplWIFICmdTest.GTSPL_setup(62, 45, 2, 6, 0, 3, 0, this);  
  
        mGtsplWIFICmdTest.GTSPL_sendCommand(this, "DIRECTION 1\n\n");  
  
        mGtsplWIFICmdTest.GTSPL_clearBuffer(this);  
  
        mGtsplWIFICmdTest.GTSPL_printerFont(100, 10, "5", 0, 1, 1, "Print Font 123456", this);  
  
        mGtsplWIFICmdTest.GTSPL_barcode(30, 30, "128", 100, 1, 0, 2, 2, "barcode1234567", this);  
  
        mGtsplWIFICmdTest.GTSPL_downloadBMP("CIRCLE.BMP", this);  
  
        mGtsplWIFICmdTest.GTSPL_sendCommand(this, "PUTBMP 150,30,\"CIRCLE.BMP\"\\r\n");  
  
        mGtsplWIFICmdTest.GTSPL_printLabel(1, 1, this);  
  
        String status = mGtsplWIFICmdTest.GTSPL_printersStatus(1000);  
  
        //簡中打印  
  
        String stString="默认简体中文测试";  
  
        mGtsplWIFICmdTest.GTSPL_clearBuffer(this);  
  
        mGtsplWIFICmdTest.GTSPL_printerFont(100, 10, "TSS24.BF2", 0, 1, 1, stString, this);  
  
        mGtsplWIFICmdTest.GTSPL_printLabel(1, 1, this);  
  
    }  
}
```

```
//繁中打印
String ttString="默認繁體中文測試";

mGtspIWIFICmdTest.GTSPL_clearBuffer(this);

mGtspIWIFICmdTest.GTSPL_printerFont(100, 10, " TST24.BF2", 0, 1, 1, ttString, this);

mGtspIWIFICmdTest.GTSPL_printLabel(1, 1, this);

mGtspIWIFICmdTest.GTSPL_closePort();

String version=mGtspIWIFICmdTest.GTSPL_getSDKVersion(0,this);
}
}
```

Gainscha Android USB 範例說明

1.匯入 GTSPL_SDK :

```
import com.gainscha.gtspl_sdk.GTSPLUsbActivity;
```

2.範例程式 :

```
public class MainActivity extends AppCompatActivity {

    GTSPLUsbActivity mUSB = new GTSPLUsbActivity();

    private static final String ACTION_USB_PERMISSION = "com.android.example.USB_PERMISSION";

    private static UsbManager mUsbManager;

    private static PendingIntent mPermissionIntent;

    private static boolean hasPermissionToCommunicate = false;

    private static UsbDevice mDevice;

    private final BroadcastReceiver mUsbReceiver = new BroadcastReceiver() {

        public void onReceive(Context context, Intent intent) {

            String action = intent.getAction();

            if (ACTION_USB_PERMISSION.equals(action)) {

                synchronized (this) {

                    UsbDevice device = intent.getParcelableExtra(UsbManager.EXTRA_DEVICE);

                    if (intent.getBooleanExtra(UsbManager.EXTRA_PERMISSION_GRANTED, false)) {

                        if (device != null) {hasPermissionToCommunicate = true;}}

                }

            }

        }

    };

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
```

```

setContentView(R.layout.activity_main);

mUsbManager = (UsbManager) getSystemService(Context.USB_SERVICE);

mPermissionIntent = PendingIntent.getBroadcast(this, 0, new
Intent(ACTION_USB_PERMISSION), 0);

IntentFilter filter = new IntentFilter(ACTION_USB_PERMISSION);

registerReceiver(mUsbReceiver, filter);

HashMap<String, UsbDevice> deviceList = mUsbManager.getDeviceList();

Iterator<UsbDevice> deviceIterator = deviceList.values().iterator();

while (deviceIterator.hasNext()) {

    mDevice = deviceIterator.next();

    if (mDevice.getVendorId() == 1137) {break;}

}

mPermissionIntent = PendingIntent.getBroadcast(MainActivity.this, 0, new
Intent(ACTION_USB_PERMISSION), PendingIntent.FLAG_ONE_SHOT);

mUsbManager.requestPermission(mDevice, mPermissionIntent);

mUSB.GTSPL_setCmdSendMode("P");

mUSB.GTSPL_openPort(mUsbManager, mDevice);

mUSB.GTSPL_setup(62, 45, 2, 3, 0, 3, 0, this);

mUSB.GTSPL_sendCommand(this, "DIRECTION 1\r\n");

mUSB.GTSPL_clearBuffer(this);

mUSB.GTSPL_barcode(30, 30, "128", 100, 1, 0, 2, 2, "barcode9463521", this);

mUSB.GTSPL_printerFont(100, 50, "2", 0, 1, 1, "PrintFontTest123", this);

mUSB.GTSPL_downloadBMP("LOGO.BMP", this);

mUSB.GTSPL_sendCommand(this, "PUTBMP 100,80,\"LOGO.BMP\"\\r\\n");

mUSB.GTSPL_printLabel(1, 1, this);

String status = mUSB.GTSPL_printersStatus(1000);

//簡中打印

```

```
String stString="默认简体中文测试";

mUSB.GTSPL_clearBuffer(this);

mUSB.GTSPL_printerFont(100, 10, "TSS24.BF2", 0, 1, 1, stString, this);

mUSB.GTSPL_printLabel(1, 1, this);

//繁中打印

String ttString="默認繁體中文測試";

mUSB.GTSPL_clearBuffer(this);

mUSB.GTSPL_printerFont(100, 10, "TST24.BF2", 0, 1, 1, ttString, this);

mUSB.GTSPL_printLabel(1, 1, this);

mUSB.GTSPL_closePort();

String version= mUSB.GTSPL_getSDKVersion(0,this);

}
```

附件

Code Type	Description	Narrow : Width					Max. data length
		1:1	1:2	1:3	2:5	3:7	
128	Code 128, switching code subset automatically.	V					
128M	Code 128, switching code subset manually.	V					
EAN128	EAN128, switching code subset automatically.	V					
EAN128M	EAN128M, switching code subset manually.	V					
25	Interleaved 2 of 5.		V	V	V		Length is even
25C	Interleaved 2 of 5 with check digit.		V	V	V		Length is odd
25S	Standard 2 of 5.		V	V	V		
25I	Industrial 2 of 5.		V	V	V		
39	Code 39, switching standard and full ASCII mode automatically.		V	V	V		
39C	Code 39 with check digit.		V	V	V		
93	Code 93.			V			
EAN13	EAN 13.	V					12
EAN13+2	EAN 13 with 2 digits add-on.	V					14
EAN13+5	EAN 13 with 5 digits add-on.	V					17
EANB	EAN 8.	V					7
EANB+2	EAN 8 with 2 digits add-on.	V					96
EANB+5	EAN 8 with 5 digits add-on.	V					12
CODA	Codabar.		V	V	V		
POST	Postnet.	V					5,9,11
UPCA	UPC-A.	V					11
UPCA+2	UPC-A with 2 digits add-on.	V					13
UPA+5	UPC-A with 5 digits add-on.	V					16
UPCE	UPC-E.	V					6
UPCE+2	UPC-E with 2 digits add-on.	V					8
UPE+5	UPC-E with 5 digits add-on.	V					11
MSI	MSI.		V	V	V		
MSIC	MSI with check digit.		V	V	V		
PLESSEY	PLESSEY.		V	V	V		
CPOST	China post.					V	
ITF14	ITF14.		V	V	V		13
EAN14	EAN14.	V					13
11	Code 11.		V	V	V		
TELEPEN	Telepen. *Since V6.89EZ.		V	V	V		
TELEPENN	Telepen number. *Since V6.89EZ.		V	V	V		

PLANET	Planet. *Since V6.89EZ.	V					
CODE49	Code 49. *Since V6.89EZ.	V					
DPI	Deutsche Post Identcode. *Since V6.91EZ.		V	V	V		11
DPL	Deutsche Post Leitcode. *Since V6.91EZ.		V	V	V		13
LOGMARS	A special use of Code 39. *Since V6.88EZ.		V	V	V		