

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,time)

■ 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

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

★ 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_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	Double	設定標籤寬度，單位 mm
height	Double	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	Double	設定間隙/黑標垂直間距高度，單位 mm
sensorOffset	Double	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

7. GTSPL_setDirectionAndMirror(direction, mirror, context)

- 函式說明：設定標籤列印時的出紙方向與是否使用鏡像列印

- 參數說明：

參數	型別	說明
direction	int	設定出紙方向，預設為 0 0：頂端出紙 1：底端出紙
mirror	int	設定是否鏡像列印 0：否 1：是
context	Context	帶入當前畫面的 Context

8. GTSPL_setShift(shiftY, context)

- 函式說明：設定圖像垂直位移距離，數值為正時，圖像會往列印方向移動，數值為負時，圖像會背離列印方向
- 參數說明：
 - ➔ shiftY：int 型別，垂直位移距離，單位為 dot
 - ➔ context：Context 類別，帶入當前畫面的 Context

9. GTSPL_printReverse(x_start, y_start, x_width, y_height, context)

- 函式說明：將指定的區域於列印時反白
- 參數說明：

參數	型別	說明
x_start	int	指定 X 起始座標位置，以點(dot)表示
y_start	int	指定 Y 起始座標位置，以點(dot)表示
x_width	int	指定 X 座標寬度，以點(dot)表示
y_height	int	指定 Y 座標高度，以點(dot)表示
context	Context	帶入當前畫面的 Context

10. GTSPL_setOffset(offset, context)

- 函式說明：設定每次出紙後額外偏移的距離(通常與剝紙模式和裁切模式組合使用)
- 參數說明：
 - ➔ offset：double 型別，額外的出紙偏移，單位為 mm
 - ➔ context：Context 類別，帶入當前畫面的 Context

11. GTSPL_setCutMode(mode, piece, context)

- 函式說明：設定裁切模式與張數
- 參數說明：

參數	型別	說明
mode	int	設定裁切方式，預設為 1 0：反切 1：正切
piece	int	設定裁切張數
context	Context	帶入當前畫面的 Context

12.GTSPL_setAfterPrintAction(mode, context)

- 函式說明：設定列印後動作
- 參數說明：

參數	型別	說明
mode	int	設定列印後動作，預設為 1 0：停在原地 1：撕紙 2：剝紙 3：裁切
context	Context	帶入當前畫面的 Context

13.GTSPL_genericDefault (context)

- 函式說明：將印表機之一般設定值初始化
- 參數說明：
 - ➔ context：Context 類別，帶入當前畫面的 Context

14.GTSPL_sensorDefault (context)

- 函式說明：將印表機之感應器設定值初始化
- 參數說明：
 - ➔ context：Context 類別，帶入當前畫面的 Context

15. 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

16. 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) ， 請參考附件一
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:列印條碼碼文置左 2:列印條碼碼文置中 3:列印條碼碼文置右
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， 請參考附件一
wide	int	設調條碼寬 bar 比例因子， 請參考附件一
content	字串	設定欲列印之條碼內容
context	Context	帶入當前畫面的 Context

17. 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

18. 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

19. 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

20. 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 TST16.BF2: 繁體中文16*16 TSS24.BF2: 簡體中文24*24 TSS16.BF2: 簡體中文16*16
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，1~10
y_scale	int	設定文字Y方向放大倍率，1~10
content	字串	設定欲列印之文字內容
context	Context	帶入當前畫面的 Context

21. GTSPL_qrcode(x, y, size, ECCLevel, cellWidth, mode, rotation, content)

- 函式說明：使用印表機列印 QRcode
- 參數說明：

參數	型別	說明
x	int	QRCode X 方向起始點，以點(dot)表示
y	int	QRCode Y 方向起始點，以點(dot)表示
ECCLevel	字串	容錯率 L : 7% M : 15% Q : 25% H : 30%
cellWidth	int	設定 QRCode 大小，1~10
mode	int	設定自動或手動編碼 A : 自動 M : 手動
rotation	int	設定QRCode旋轉角度 0 : 旋轉0度 90 : 旋轉90度 180 : 旋轉180度 270 : 旋轉270度
content	字串	設定資料內容 資料內容限制： 1) 數字資料: (數字 0~9) 2) 字母資料 數字 0-9 大寫字母 A-Z 9 種其它字元: 空格, \$ % * + - . / :) *如果” A” 是資料內容的第一個字元，那麼資料內容將會被設置為字母數據。 *如果” N” 是資料內容的第一個字元，那麼資料內容將會被設置為數字數據。 * “！” 用來轉換資料的格式，” N” 、” A” 等資料類型可通過”！” 來轉換。

22. 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

23. 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

24. 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

25. GTSPL_download_Not1BitDepthBMP (filename, context)

- 函式說明：將非單色之 BMP 格式圖檔轉檔後下載至印表機
- 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

26. 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

27. 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

28. 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

29. GTSPL_writeUHF (dataFormat,startBlockNo,byteSize,Gen2MemoryBank,datastring, context)

- 函式說明：將資料寫入 UHF 標籤記憶體中
- 參數說明：

參數	型別	說明
dataFormat	string	設定字串資料編碼格式，預設為 H A：ASCII H：Hexadecimal
startBlockNo	int	設定資料區塊起始位置，預設為 2
byteSize	int	設定寫入資料byte長度，預設為1
Gen2Memory Bank	string	設定 Gen2 資料區段，預設為 E R：保留 E：EPC T：TID(Tag ID) U：User
datastring	string	欲寫入之字串資料
context	Context	帶入當前畫面的 Context

30. GTSPL_EPCPWD_Action(action, password, context)

- 函式說明：將 UHF GNE2 的 EPC 資料區塊上鎖或解鎖
- 參數說明：

參數	型別	說明
action	string	設定執行動作 U：解鎖資料區塊 L：上鎖資料區塊 O：永久解鎖資料區塊 P：永久上鎖資料區塊

password	string	密碼，應為 8 hex 字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的 Context

31. GTSPL_TIDPWD_Action(action, password, context)

- 函式說明：將 UHF GNE2 的 TID 資料區塊上鎖或解鎖
- 參數說明：

參數	型別	說明
action	string	設定執行動作 U：解鎖資料區塊 L：上鎖資料區塊 O：永久解鎖資料區塊 P：永久上鎖資料區塊
password	string	密碼，應為 8 hex 字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的 Context

32. GTSPL_USERPWD_Action(action, password, context)

- 函式說明：將 UHF GNE2 的 USER 資料區塊上鎖或解鎖
- 參數說明：

參數	型別	說明
action	string	設定執行動作 U：解鎖資料區塊 L：上鎖資料區塊 O：永久解鎖資料區塊 P：永久上鎖資料區塊
password	string	密碼，應為 8 hex 字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的 Context

33. GTSPL_AccessPWD_Action (action, password, context)

- 函式說明：將 UHF GNE2 的存取密碼進行設定、上鎖或解鎖
- 參數說明：

參數	型別	說明
----	----	----

action	string	設定執行動作 U：解鎖存取密碼 L：上鎖存取密碼 O：永久解鎖存取密碼 P：永久上鎖存取密碼 S：設定存取密碼
password	string	密碼，應為 8 hex 字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的 Context

34. GTSPL_KillPWD_Action (action, password, context)

- 函式說明：將 UHF GNE2 的刪除密碼進行設定、上鎖或解鎖
- 參數說明：

參數	型別	說明
action	string	設定執行動作 U：解鎖刪除密碼 L：上鎖刪除密碼 O：永久刪除存取密碼 P：永久刪除存取密碼 S：設定刪除密碼
password	string	密碼，應為 8 hex 字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的 Context

35. GTSPL_Set_RFIDPorcedure (tagType, rw_position, void_printout, tryEncodie_times,error_handle, speed, retry_times, context)

- 函式說明：RFID 設定
- 參數說明：

參數	型別	說明
tagType	int	設定標籤類型，1~10，預設值為 8 1：EPC Class 1 Generation 2-Q , 8：EPC Class 1 Generation 2-R ,10：UHF-J
rw_position	int	設標籤讀寫位置(標籤頂部起算)，範圍為 0~9999(dot)，預設為 0

void_printout	int	設定無效列印長度(dot)，範圍為0~標籤長度，預設為標籤長度
tryEncodie_times	string	設定最大無效標籤數，範圍為 0~10，預設為 3
error_handle	string	設定無效時採取的動作，預設為 N N：No action(繼續) P：Pause mode(暫停) E：Error mode(停止)
speed	int	設定無效打印速度，範圍 2~10(IPS)，預設值 2(IPS)
retry_times	int	設定標籤重試次數，範圍 0~10，預設值 6
context	Context	帶入當前畫面的 Context

36. GTSPL_Set_RFIDPorcedure (tagType, rw_position, void_printout, tryEncodie_times,error_handle, speed, retry_times,dpi, context)

■ 函式說明：RFID 設定

■ 參數說明：

參數	型別	說明
tagType	int	設定標籤類型，1~10，預設值為 8 1：EPC Class 1 Generation 2-Q, 8：EPC Class 1 Generation 2-R, 10：UHF-J
rw_position	int	設標籤讀寫位置(標籤頂部起算)，範圍為 203dpi:0 ~ 1251 (mm)、300dpi:0 ~ 846 (mm)、600dpi:0 ~ 423 (mm)，預設為 0
void_printout	int	設定無效列印長度(dot)，範圍為0~標籤長度，預設為標籤長度
tryEncodie_times	string	設定最大無效標籤數，範圍為 0~10，預設為 3
error_handle	string	設定無效時採取的動作，預設為 N N：No action(繼續) P：Pause mode(暫停) E：Error mode(停止)
speed	int	設定無效打印速度，範圍 2~10(IPS)，預設值 2(IPS)
retry_times	int	設定標籤重試次數，範圍 0~10，預設值 6
dpi	String	設定打印機的 DPI 203: 203 dpi 300: 300 dpi 600: 600 dpi

context	Context	帶入當前畫面的 Context
----------------	---------	-----------------

37. GTSPL_writeHF (dataFormat,startBlockNo,byteSize,datastring, context)

- 函式說明：將資料寫入 HF 標籤記憶體中
- 參數說明：

參數	型別	說明
dataFormat	string	設定字串資料編碼格式，預設為 H A：ASCII H：Hexadecimal
startBlockNo	int	設定資料區塊起始位置，預設為 2
byteSize	int	設定寫入資料byte長度，預設為1
datastring	string	欲寫入之字串資料
context	Context	帶入當前畫面的 Context

38. GTSPL_printerFontBlock (x, y, width, height, fontname, rotation, x_scale, y_scale, space, align, content, context)

- 函式說明：列印段落文字內容
- 參數說明：

參數	型別	說明
x	string	文字 X 方向起始點，以點(dot)表示
y	string	文字 Y 方向起始點，以點(dot)表示
width	string	設定段落區塊寬度，以點(dot)表示
height	string	設定段落區塊高度，以點(dot)表示
fontname	string	內建字型名稱 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 TST16.BF2: 繁體中文16*16 TSS24.BF2: 簡體中文24*24 TSS16.BF2: 簡體中文16*16
rotation	string	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	string	設定文字 X 方向放大倍率，1~10
y_scale	string	設定文字X方向放大倍率，1~10
space	string	行距，以點(dot)表示
align	string	對齊位置 0：預設(置左) 1：置左 2：置中 3：置右
content	string	設定欲列印之文字內容
context	Context	帶入當前畫面的 Context

39. GTSPL_readUHF(dataFormat,startBlockNo,byteSize,Gen2MemoryBank,context)

■ 函式說明：讀取 UHF 標籤記憶體資料

■ 參數說明：

參數	型別	說明
dataFormat	string	設定字串資料編碼格式，預設為 H A：ASCII H：Hexadecimal
startBlockNo	int	設定資料區塊起始位置，預設為 0
byteSize	int	設定讀取資料byte長度，預設為1
Gen2Memory Bank	string	設定 Gen2 資料區段，預設為 E R：保留 E：EPC T：TID(Tag ID) U：User
context	Context	帶入當前畫面的 Context

■ 回傳字串說明：

dataFormat	回傳字串(範例)
A	標籤資料以 ASCII 顯示 (ex: 24051324000103456400)
H	標籤資料以 Hexadecimal 顯示

	(ex: 3234303531333234303030313033343536343030)
*發生錯誤回傳錯誤代碼，錯誤代碼說明請參考附件二，UHFReaderE710 請參考附件三	

40. GTSPL_readUHF(dataFormat,startBlockNo,byteSize,Gen2MemoryBank,delaytime,context)

- 函式說明：讀取 UHF 標籤記憶體資料(目前只支援 USB 使用 delaytime)
- 參數說明：

參數	型別	說明
dataFormat	string	設定字串資料編碼格式，預設為 H A：ASCII H：Hexadecimal
startBlockNo	int	設定資料區塊起始位置，預設為 0
byteSize	int	設定讀取資料byte長度，預設為1
Gen2Memory Bank	string	設定 Gen2 資料區段，預設為 E R：保留 E：EPC T：TID(Tag ID) U：User
delaytime	int	設定讀取的延遲時間
context	Context	帶入當前畫面的 Context

- 回傳字串說明：

dataFormat	回傳字串(範例)
A	標籤資料以 ASCII 顯示 (ex: 24051324000103456400)
H	標籤資料以 Hexadecimal 顯示 (ex: 3234303531333234303030313033343536343030)
*發生錯誤回傳錯誤代碼，錯誤代碼說明請參考附件二，UHFReaderE710 請參考附件三	

41. GTSPL_readUHFQ(dataFormat, PCReturnStatus, CRCReturnStatus,context)

- 函式說明：讀取 UHF 標籤記憶體資料
- 參數說明：

參數	型別	說明
dataFormat	string	設定字串資料編碼格式，預設為 H

		A : ASCII H : Hexadecimal
PCReturnStatus	int	PC 返回狀態，預設為 0 0：不回傳 PC 值 1：回傳 PC 值
CRCReturnStatus	int	CRC-16 返回狀態，預設為 0 0：不回傳 CRC-16 1：回傳CRC-16
context	Context	帶入當前畫面的 Context

■ 回傳字串說明：

dataFormat	回傳字串(範例)
A	標籤資料以 ASCII 顯示 (ex: 24051324000103456400)
H	標籤資料以 Hexadecimal 顯示 (ex: 3234303531333234303030313033343536343030)
*發生錯誤回傳錯誤代碼，錯誤代碼說明請參考附件二，UHFReaderE710 請參考附件三	

42. GTSPL_LabelCalibration (Context context)

- 函式說明：執行 RFID Tag 校準動作
- 參數說明：
 - ➔ context：Context 類別，帶入當前畫面的 Context

43. GTSPL_rfidSetupDefault (Context context)

- 函式說明：將 RFID 設定值初始化
- 參數說明：
 - ➔ context：Context 類別，帶入當前畫面的 Context

44. GTSPL_writeGJB(String dataFormat, int startBlockNo, int byteSize, String GJBMemoryBank, String datastring, String writePWD, Context context)

- 函式說明：將資料寫入 UHF GJB 標籤記憶體中

■ 參數說明：

參數	型別	說明
dataFormat	string	設定字串資料編碼格式，預設為 H A：ASCII H：Hexadecimal
startBlockNo	int	設定資料區塊起始位置，GJB 預設為 1
byteSize	int	設定寫入資料byte長度，預設為1
Gen2MemoryBank	string	設定 GJB 資料區段，預設為 E R：安全區 E：EPC T：TID(Tag ID) U：User
datastring	string	欲寫入之字串資料
writePWD	string	寫入密碼，應為 8 hex 字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的 Context

45. GTSPL_readGJB(String dataFormat, int startBlockNo, int byteSize, String GJBMemoryBank, String ReadPWD, Context context)

■ 函式說明：讀取 UHF GJB 標籤記憶體資料，需用字串變數接收回傳訊息

■ 參數說明：

參數	型別	說明
dataFormat	string	設定字串資料編碼格式，預設為 H A：ASCII H：Hexadecimal
startBlockNo	int	設定資料區塊起始位置，預設為 0
byteSize	int	設定讀取資料byte長度，預設為1
Gen2MemoryBank	string	設定 GJB 資料區段，預設為 E R：安全區 E：EPC T：TID(Tag ID) U：User
ReadPWD	string	讀取密碼，應為 8 hex 字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的 Context

■ 回傳字串說明：

dataFormat	回傳字串(範例)
A	標籤資料以 ASCII 顯示 (ex: 24051324000103456400)
H	標籤資料以 Hexadecimal 顯示 (ex: 3234303531333234303030313033343536343030)

46. GTSPL_Set_GJB_Pwd(String pwdArea, String action, String pwdSet, String writePWD, Context context)

- 函式說明：設定 UHF GJB 各密碼區之新密碼
- 參數說明：

參數	型別	說明
passwordArea	string	設定密碼區域，預設為 W K：Kill 刪除 W：Write 寫入 R：Read 讀取 S：Status 狀態
action	string	設定動作 S：Set Password
pwdSet	string	設定密碼區域的新密碼，應為8 hex字元(0~9,A,B,C,D,E,F)
writePWD	string	目前之寫入密碼，應為 8 hex 字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的 Context

47. GTSPL_Set_GJB_Status(String GJBMemoryBank, String action, String statusPWD, Context context)

- 函式說明：設定 UHF GJB 各資料區塊讀寫狀態
- 參數說明：

參數	型別	說明
Gen2MemoryBank	string	設定 GJB 資料區段，預設為 E F：安全區 E：EPC T：TID(Tag ID) U：User
action	string	設定狀態，預設為 A A=Lock0(可讀可寫)

		B=Lock1(可讀不可寫) C=Lock2(不可讀可寫) D=Lock3(不可讀不可寫)
statusPWD	string	狀態密碼，應為8 hex字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的Context

48. GTSPL_Kill_GJB_Tag(String kill_PWD, Context context)

- 函式說明：刪除 UHF GJB 標籤
- 參數說明：

參數	型別	說明
kill_PWD	string	刪除密碼，應為 8 hex 字元(0~9,A,B,C,D,E,F)
context	Context	帶入當前畫面的 Context

49. GTSPL_WifiFrequency (String Frequency, Context context)

- 函式說明：使用兼容 5G 頻段 WIFI 模塊時，可用於切換使用頻段
- 參數說明：

參數	型別	說明
Frequency	string	設定模塊頻段 2.4G：使用 2.4G 頻段 5G：使用 5G 頻段 BOTH：使用雙頻頻段
context	Context	帶入當前畫面的 Context

50. GTSPL_printBMP (String x, String y, int width, int height, int mode, String filename, Context context)

- 函式說明：將圖片轉為單色點陣圖，使用印表機直接列印
- 參數說明：

參數	型別	說明
x	string	文字 X 方向起始點，以點(dot)表示
y	string	文字 Y 方向起始點，以點(dot)表示
width	int	圖片寬度，以位元組(byte)表示
height	int	圖片高度，以點(dot)表示

mode	int	圖片格式 0: OVERWRITE 1: OR 2: XOR
filename	string	檔案名稱 (檔案需存在內部儲存空間/android/data/packageName/files資料夾下) 圖檔僅支援以下格式： 1. BMP (Bitmap)：位圖格式 2. JPG (JPEG)：壓縮的圖像格式 3. PNG (Portable Network Graphics)：無損壓縮的圖像格式 4. GIF (Graphics Interchange Format)：支援多張圖片的格式，通常用於動畫 5. TIFF (Tagged Image File Format)：高品質的無損壓縮圖像格式 6. ICO (Icon)：圖示格式，用於顯示檔案、程式或資料夾的圖示 7. WMF (Windows Metafile)：Windows 繪圖文件格式 8. EMF (Enhanced Metafile)：擴展的 Windows 繪圖文件格式
context	Context	帶入當前畫面的 Context

51. GTSPL_printBMP_Compression (String x, String y, int width, int height, String filename,

Context context)

- 函式說明：將圖片轉為單色點陣圖，壓縮後再使用印表機列印
- 參數說明：

參數	型別	說明
x	string	文字 X 方向起始點，以點(dot)表示
y	string	文字 Y 方向起始點，以點(dot)表示
width	int	圖片寬度，以位元組(byte)表示
height	int	圖片高度，以點(dot)表示
filename	string	檔案名稱 (檔案需存在內部儲存空間/android/data/packageName/files資料夾下) 圖檔僅支援以下格式： 1. BMP (Bitmap)：位圖格式 2. JPG (JPEG)：壓縮的圖像格式 3. PNG (Portable Network Graphics)：無損壓縮的圖像格式 4. GIF (Graphics Interchange Format)：支援多張圖片的格式，通常用

		於動畫 5. TIFF (Tagged Image File Format)：高品質的無損壓縮圖像格式 6. ICO (Icon)：圖示格式，用於顯示檔案、程式或資料夾的圖示 7. WMF (Windows Metafile)：Windows 繪圖文件格式 8. EMF (Enhanced Metafile)：擴展的 Windows 繪圖文件格式
context	Context	帶入當前畫面的 Context

52. GTSPL_autoConnect (MacAddress, isManual)

- 函式說明：非手動關閉的情況下，7 秒過後在指定藍芽的 MAC 位址(請使用 BR/EDR MacAddress)，開啟輸出埠(目前只支援 BT 使用)
- 只支持 GE2408DE1168 的機種
- 參數說明：

➔ **MacAddress**：address：字串型別，指定連線的藍芽位址(Bluetooth MacAddress)，

如："DC:1D:30:00:1D:87"

➔ **isManual**：Boolean 類別，確認是否為手動關閉

53. GTSPL_labelCalibration(width, height, sensor, sensorDistance, tagType, context)

- 函式說明：設定標籤的寬度、高度、感應器類別、間隙/黑標垂直間距、標籤類型，設定完後執行 RFID Tag 校準動作
- 只支持 GE2408DE1168 的機種
- 參數說明：

參數	型別	說明
width	Double	設定標籤寬度，單位 mm
height	Double	設定標籤高度，單位 mm
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	Double	設定間隙/黑標垂直間距高度，單位 mm
tagType	int	設定標籤類型，1~10，預設值為 8 1：EPC Class 1 Generation 2-Q，8：EPC Class 1 Generation 2-R，10：UHF-J
context	Context	帶入當前畫面的 Context

54. GTSPL_setBicolor(color, density, context)

- 函式說明：設定雙色指令
- 僅支持部分機種，詳情請洽代理商
- 參數說明：

參數	型別	說明
color	string	設定顏色 R：紅色 B：黑色
density	int	設定列印濃度，1~15，數字越大列印結果越深(color 為黑色時設定為 0)
context	Context	帶入當前畫面的 Context

Android Bluetooth 範例說明

1.在 build.gradle(Module)中的 dependencies{}選擇要使用 jar 檔或是 aar 檔

啟用 jar 檔:

```
implementation fileTree(dir: 'libs', include: ['*.jar'])
```

啟用 aar 檔:

```
implementation fileTree(dir: 'libs', include: ['*.aar'])
```

2.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" />
```

3.匯入 GTSPS_SDK：

```
import com.gto.gtspl_sdk.GTSPLActivity;
```

4.範例程式：

```
public class MainActivity extends Activity {  
  
    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_autoConnect("DC:1D:30:00:1D:87", false);  
  
        mGtsplCmdTest.GTSPL_setup(62, 45, 2, 6, 0, 3, 0, this);  
  
        mGtsplCmdTest.GTSPL_setup(62.0, 45.0, 2, 6, 0, 3.0, 0.0, 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_qrcode(300, 100, "H", 4, "A", 0, "ABCabc123", this);  
    }  
}
```

```

mGtsplCmdTest.GTSPL_downloadBMP("CIRCLE.BMP", this);

mGtsplCmdTest.GTSPL_sendCommand(this, "PUTBMP 150,30,\"CIRCLE.BMP\"\\r\\n");

mGtsplCmdTest.GTSPL_download_Not1BitDepthBMP("printTest4.BMP", MainActivity.this);

mGtsplCmdTest.GTSPL_sendCommand(MainActivity.this, "PUTBMP

10,10,\"PrintTest4.BMP\"\\r\\n");

String sBlock = "We stand behind our products with one of the most comprehensive support
programs in the Auto-ID industry.";

mGtsplCmdTest.GTSPL_printFontBlock("15", "15", "790", "90", "0", "0", "8", "8", "20",
"2",sBlock,this);

mGtsplCmdTest.GTSPL_printLabel(1, 1, this);


//初始化

mGtsplCmdTest.GTSPL_genericDefault(); //一般設定初始化

mGtsplCmdTest.GTSPL_sensorDefault(); //感應器初始化

mGtsplCmdTest.GTSPL_rfidSetupDefault(); //rfid 設定初始化


//修改 Wifi 頻段

mGtsplCmdTest.GTSPL_WifiFrequency("5G");


//打印設定功能(需搭配打印功能執行)

mGtsplCmdTest.GTSPL_setDirectionAndMirror(1, 0); //設定打印方向與鏡像

mGtsplCmdTest.GTSPL_setShift(50); //設定垂直偏移

mGtsplCmdTest.GTSPL_printReverse(10, 10, 160, 160); //設定反白

mGtsplCmdTest.GTSPL_setOffset(20); //設定出紙偏移

mGtsplCmdTest.GTSPL_setCutMode(0, 2); //設定切刀模式與張數

mGtsplCmdTest.GTSPL_setAfterPrintAction(2); //設定列印後動作

mGtsplCmdTest.GTSPL_printLabel(1, 1);

```

```
//Bitmap 打印
```

```
mGtsplCmdTest.GTSPL_printBMP(-500,30,400,300,1,"CIRCLE.BMP ",this);
```

```
mGtsplCmdTest.GTSPL_printBMP_Compression(-500,30,400,300," CIRCLE.BMP ",this);
```

```
mGtsplCmdTest.GTSPL_printLabel(1, 1);
```

```
//雙色打印
```

```
mGtsplCmdTest.GTSPL_setBicolor("R",7, this);
```

```
//GEN2 RFID
```

```
mGtsplCmdTest.GTSPL_writeUHF("H", 2, 12, "E", "414142424343444445454646", this);
```

```
mGtsplCmdTest.GTSPL_GTSPL_EPCPWD_Action("U", "12345678", this);
```

```
mGtsplCmdTest.GTSPL_GTSPL_TIDPWD_Action("L", "12345678", this);
```

```
mGtsplCmdTest.GTSPL_USERPWD_Action("L", "12345678", this);
```

```
mGtsplCmdTest.GTSPL_AccessPWD_Action("S", "12345678", this);
```

```
mGtsplCmdTest.GTSPL_KillPWD_Action("S", "12345678", this);
```

```
mGtsplCmdTest.GTSPL_Set_RFIDPorcedure(8, 8, 32, 3, "N", 2, 2, this);
```

```
mGtsplCmdTest.GTSPL_Set_RFIDPorcedure(8, 8, 32, 3, "N", 2, 2, "203", this);
```

```
mGtsplCmdTest.GTSPL_writeHF("H", 2, 12, "414142424343444445454646", this);
```

```
mGtsplCmdTest.GTSPL_printLabel(1, 1, this);
```

```
String uhfData = mGtsplCmdTest.GTSPL_readUHF("H",2,12,"E", this);
```

```
String uhfData = mGtsplCmdTest.GTSPL_readUHFQ("H",0,0, this);
```

```
//GJB RFID 設定密碼
```

```
//帶入寫入密碼，設定新的讀取密碼
```

```
mGtsplCmdTest.GTSPL_Set_GJB_Pwd("R","S","87654321","12345678",this);
```

```
//帶入寫入密碼，設定新的寫入密碼
```

```

mGtsplCmdTest.GTSPL_Set_GJB_Pwd("W","S","87654321","12345678",this);

//帶入寫入密碼，設定新的刪除密碼

mGtsplCmdTest.GTSPL_Set_GJB_Pwd("K","S","87654321","12345678",this);

//帶入寫入密碼，設定新的狀態密碼

mGtsplCmdTest.GTSPL_Set_GJB_Pwd("S","S","87654321","12345678",this);

mGtsplCmdTest.GTSPL_printLabel(1, 1, this);


//GJB RFID 設定不同資料區塊的寫入讀取狀態

mGtsplCmdTest.GTSPL_Set_GJB_Status("E","C","11112222",this);

mGtsplCmdTest.GTSPL_printLabel(1, 1, this);


//GJB RFID 寫入 EPC 資料

mGtsplCmdTest.GTSPL_writeGJB("H",2,12,"E","404041414242434344444545","12345678",this);

mGtsplCmdTest.GTSPL_printLabel(1, 1, this);


//GJB RFID 讀取 EPC 資料

String GJBData=mGtsplCmdTest.GTSPL_readGJB("H",2,12,"E","33334444",MainActivity.this);


//GJB RFID 刪除標籤

mGtsplCmdTest.GTSPL_Kill_GJB_Tag("11224455",this);

mGtsplCmdTest.GTSPL_printLabel(1,1,this);


//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);

String status = mGtsplCmdTest.GTSPL_printersStatus(1000);

mGtsplCmdTest.GTSPL_closePort(1000);

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


//RFID 自動校準

mGtsplCmdTest.GTSPL_LabelCalibration(this);


//UHFReaderE710 自動校準

mGtsplCmdTest.GTSPL_labelCalibration(62.0, 45.0, 0, 3.0, 8,this)
}
}

```

Android Ethernet 範例說明

1.在 build.gradle(Module)中的 dependencies{}選擇要使用 jar 檔或是 aar 檔

啟用 jar 檔:

```
implementation fileTree(dir: 'libs', include: ['*.jar'])
```

啟用 aar 檔:

```
implementation fileTree(dir: 'libs', include: ['*.aar'])
```

2.需先於 AndroidManifest.xml 設定下列權限：

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

3.匯入 GTSP_SDK：

```
import com.gto.gtspl_sdk.GTSPLWIFIActivity;
```

4.範例程式：

```
public class MainActivity extends Activity {  
  
    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",8899,5000);  
  
        mGtsplWIFICmdTest.GTSPL_setup(62, 45, 2, 6, 0, 3, 0, this);  
  
        mGtsplWIFICmdTest.GTSPL_setup(62.0, 45.0, 2, 6, 0, 3.0, 0.0, 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_qrcode(300, 100, "H", 4, "A", 0, "ABCabc123", this);  
  
        mGtsplWIFICmdTest.GTSPL_downloadBMP("CIRCLE.BMP", this);  
  
        mGtsplWIFICmdTest.GTSPL_sendCommand(this, "PUTBMP 150,30,\"CIRCLE.BMP\"\\r\\n");  
    }  
}
```

```

mGtsplWIFICmdTest.GTSPL_download_Not1BitDepthBMP("printTest4.BMP",
MainActivity.this);

mGtsplWIFICmdTest.GTSPL_sendCommand(MainActivity.this, "PUTBMP
10,10,\"PrintTest4.BMP\\r\\n\");

String sBlock = "We stand behind our products with one of the most comprehensive support
programs in the Auto-ID industry.";

mGtsplWIFICmdTest.GTSPL_printFontBlock("15", "15", "790", "90", "0", "0", "8", "8", "20",
"2",sBlock,this);

mGtsplWIFICmdTest.GTSPL_printLabel(1, 1, this);


//初始化
mGtsplWIFICmdTest.GTSPL_genericDefault(); //一般設定初始化
mGtsplWIFICmdTest.GTSPL_sensorDefault(); //感應器初始化
mGtsplWIFICmdTest.GTSPL_rfidSetupDefault(); //rfid 設定初始化


//修改 Wifi 頻段
mGtsplWIFICmdTest.GTSPL_WifiFrequency("5G");


//打印設定功能(需搭配打印功能執行)
mGtsplWIFICmdTest.GTSPL_setDirectionAndMirror(1, 0); //設定打印方向與鏡像
mGtsplWIFICmdTest.GTSPL_setShift(50); //設定垂直偏移
mGtsplWIFICmdTest.GTSPL_printReverse(10, 10, 160, 160); //設定反白
mGtsplWIFICmdTest.GTSPL_setOffset(20); //設定出紙偏移
mGtsplWIFICmdTest.GTSPL_setCutMode(0, 2); //設定切刀模式與張數
mGtsplWIFICmdTest.GTSPL_setAfterPrintAction(2); //設定列印後動作
mGtsplWIFICmdTest.GTSPL_printLabel(1, 1);

```

//Bitmap 打印

```
mGtsplWIFICmdTest.GTSPL_printBMP(-500,30,400,300,1,"CIRCLE.BMP ",this);  
mGtsplWIFICmdTest.GTSPL_printBMP_Compression(-500,30,400,300," CIRCLE.BMP ",this);  
mGtsplWIFICmdTest.GTSPL_printLabel(1, 1);
```

//雙色打印

```
mGtsplWIFICmdTest.GTSPL_setBicolor("R",7, this);
```

//GEN2 RFID

```
mGtsplWIFICmdTest.GTSPL_writeUHF("H", 2, 12, "E", "414142424343444445454646", this);  
mGtsplWIFICmdTest.GTSPL_EPCPWD_Action("L", "12345678", this);  
mGtsplWIFICmdTest.GTSPL_TIDPWD_Action("L", "12345678", this);  
mGtsplWIFICmdTest.GTSPL_USERPWD_Action("L", "12345678", this);  
mGtsplWIFICmdTest.GTSPL_AccessPWD_Action("S", "12345678", this);  
mGtsplWIFICmdTest.GTSPL_KillPWD_Action("S", "12345678", this);  
mGtsplWIFICmdTest.GTSPL_Set_RFIDPorcedure(8, 8, 32, 3, "N", 2, 2, this);  
mGtsplWIFICmdTest.GTSPL_Set_RFIDPorcedure(8, 8, 32, 3, "N", 2, 2, "203", this);  
mGtsplWIFICmdTest.GTSPL_writeHF("H", 2, 12, "414142424343444445454646", this);  
mGtsplWIFICmdTest.GTSPL_printLabel(1, 1, this);  
String uhfData = mGtsplWIFICmdTest.GTSPL_readUHF("H",2,12,"E", this);  
String uhfData = mGtsplWIFICmdTest.GTSPL_readUHFQ("H",0,0,this);
```

//GJB RFID 設定密碼

//帶入寫入密碼，設定新的讀取密碼

```
mGtsplWIFICmdTest.GTSPL_Set_GJB_Pwd("R","S","87654321","12345678",this);
```

//帶入寫入密碼，設定新的寫入密碼

```
mGtsplWIFICmdTest.GTSPL_Set_GJB_Pwd("W","S","87654321","12345678",this);
```



```

//帶入寫入密碼，設定新的刪除密碼
mGtsplWIFICmdTest.GTSPL_Set_GJB_Pwd("K","S","87654321","12345678",this);

//帶入寫入密碼，設定新的狀態密碼
mGtsplWIFICmdTest.GTSPL_Set_GJB_Pwd("S","S","87654321","12345678",this);

mGtsplWIFICmdTest.GTSPL_printLabel(1, 1, this);


//GJB RFID 設定不同資料區塊的寫入讀取狀態
mGtsplWIFICmdTest.GTSPL_Set_GJB_Status("E","C","11112222",this);

mGtsplWIFICmdTest.GTSPL_printLabel(1, 1, this);


//GJB RFID 寫入 EPC 資料
mGtsplWIFICmdTest.GTSPL_writeGJB("H",2,12,"E","404041414242434344444545","123456
78",this);

mGtsplWIFICmdTest.GTSPL_printLabel(1, 1, this);


//GJB RFID 讀取 EPC 資料
String GJBData =
mGtsplWIFICmdTest.GTSPL_readGJB("H",2,12,"E","33334444",MainActivity.this);


//GJB RFID 刪除標籤
mGtsplWIFICmdTest.GTSPL_Kill_GJB_Tag("11224455",this);

mGtsplWIFICmdTest.GTSPL_printLabel(1,1,this);


//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";

mGtsplWIFICmdTest.GTSPL_clearBuffer(this);

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

mGtsplWIFICmdTest.GTSPL_printLabel(1, 1, this);

String status = mGtsplWIFICmdTest.GTSPL_printersStatus(1000);

mGtsplWIFICmdTest.GTSPL_closePort();

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


//RFID 自動校準

mGtsplWIFICmdTest.GTSPL_LabelCalibration(this);


//UHFReaderE710 自動校準

mGtsplWIFICmdTest.GTSPL_LabelCalibration(62.0, 45.0, 0, 3.0, 8,this)
}
}

```

Android USB 範例說明

1. 在 `build.gradle(Module)` 中的 `dependencies{}` 選擇要使用 `jar` 檔或是 `aar` 檔

啟用 `jar` 檔:

```
implementation fileTree(dir: 'libs', include: ['*.jar'])
```

啟用 `aar` 檔:

```
implementation fileTree(dir: 'libs', include: ['*.aar'])
```

2. 匯入 `GTSP_SDK` :

```
import com.gto.gtspl_sdk.GTSPLUsbActivity;
```

3. 範例程式 :

```
public class MainActivity extends Activity {

    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_setup(62.0, 45.0, 2, 3, 0, 3.0, 0.0, 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_qrcode(300, 100, "H", 4, "A", 0, "ABCabc123", 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_download_Not1BitDepthBMP("printTest4.BMP", MainActivity.this);  
  
mUSB.GTSPL_sendCommand(MainActivity.this, "PUTBMP10,10,\"PrintTest4.BMP\"\\r\\n");
```

```
String sBlock = "We stand behind our products with one of the most comprehensive  
  
support programs in the Auto-ID industry."  
  
mUSB.GTSPL_printFontBlock("15", "15", "790", "90", "0", "0", "8", "8", "20",  
"2",sBlock,this);  
  
mUSB.GTSPL_printLabel(1, 1, this);
```

//初始化

```
mUSB.GTSPL_genericDefault(); //一般設定初始化  
mUSB.GTSPL_sensorDefault(); //感應器初始化  
mUSB.GTSPL_rfidSetupDefault(); //rfid 設定初始化
```

//修改 Wifi 頻段

```
mUSB.GTSPL_WifiFrequency("5G");
```

//打印設定功能(需搭配打印功能執行)

```
mUSB.GTSPL_setDirectionAndMirror(1, 0); //設定打印方向與鏡像  
mUSB.GTSPL_setShift(50); //設定垂直偏移  
mUSB.GTSPL_printReverse(10, 10, 160, 160); //設定反白  
mUSB.GTSPL_setOffset(20); //設定出紙偏移  
mUSB.GTSPL_setCutMode(0, 2); //設定切刀模式與張數
```

```
mUSB.GTSPL_setAfterPrintAction(2); //設定列印後動作
```

```
mUSB.GTSPL_printLabel(1, 1);
```

```
//Bitmap 打印
```

```
mUSB.GTSPL_printBMP(-500,30,400,300,1,"CIRCLE.BMP ",this);
```

```
mUSB.GTSPL_printBMP_Compression(-500,30,400,300," CIRCLE.BMP ",this);
```

```
mUSB.GTSPL_printLabel(1, 1);
```

```
//雙色打印
```

```
mUSB.GTSPL_setBicolor("R",7, this);
```

```
//GEN2 RFID
```

```
mUSB.GTSPL_writeUHF("H", 2, 12, "E", "414142424343444445454646", this);
```

```
mUSB.GTSPL_EPCPWD_Action("L", "12345678", this);
```

```
mUSB.GTSPL_TIDPWD_Action("L", "12345678", this);
```

```
mUSB.GTSPL_USERPWD_Action("L", "12345678", this);
```

```
mUSB.GTSPL_AccessPWD_Action("S", "12345678", this);
```

```
mUSB.GTSPL_KillPWD_Action ("S", "12345678", this);
```

```
mUSB.GTSPL_Set_RFIDPorcedure(8, 8, 32, 3, "N", 2, 2, this);
```

```
mUSB.GTSPL_Set_RFIDPorcedure(8, 8, 32, 3, "N", 2, 2, "203", this);
```

```
mUSB.GTSPL_writeHF("H", 2, 12, "414142424343444445454646", this);
```

```
mUSB.GTSPL_printLabel(1, 1, this);
```

```
String uhfData = mUSB.GTSPL_readUHF("H",2,12,"E", this);
```

```
String uhfData = mUSB.GTSPL_readUHF("H",2,12,"E", 4000, this);
```

```
String uhfData = mUSB.GTSPL_readUHFQ("H",0,0,this);
```

```
//GJB RFID 設定密碼
```

```

//帶入寫入密碼，設定新的讀取密碼

mUSB.GTSPL_Set_GJB_Pwd("R","S","87654321","12345678",this);

//帶入寫入密碼，設定新的寫入密碼

mUSB.GTSPL_Set_GJB_Pwd("W","S","87654321","12345678",this);

//帶入寫入密碼，設定新的刪除密碼

mUSB.GTSPL_Set_GJB_Pwd("K","S","87654321","12345678",this);

//帶入寫入密碼，設定新的狀態密碼

mUSB.GTSPL_Set_GJB_Pwd("S","S","87654321","12345678",this);

mUSB.GTSPL_printLabel(1, 1, this);


//GJB RFID 設定不同資料區塊的寫入讀取狀態

mUSB.GTSPL_Set_GJB_Status("E","C","11112222",this);

mUSB.GTSPL_printLabel(1, 1, this);


//GJB RFID 寫入 EPC 資料

mUSB.GTSPL_writeGJB("H",2,12,"E","404041414242434344444545","12345678",this);

mUSB.GTSPL_printLabel(1, 1, this);


//GJB RFID 讀取 EPC 資料

String GJBData = mUSB.GTSPL_readGJB("H",2,12,"E","33334444",MainActivity.this);


//GJB RFID 刪除標籤

mUSB.GTSPL_Kill_GJB_Tag("11224455",this);

mUSB.GTSPL_printLabel(1,1,this);


//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);

String status = mUSB.GTSPL_printersStatus(1000);

mUSB.GTSPL_closePort();

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

```
//RFID 自動校準
```

```
mUSB.GTSPL_LabelCalibration(this);
```

```
//UHFReaderE710 自動校準
```

```
mUSB.GTSPL_labelCalibration(62.0, 45.0, 0, 3.0, 8, this)
```

```
}
```


附件一

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		

附件二

錯誤代碼	錯誤代碼說明
100	其他錯誤
101	超過記憶體空間
102	記憶體被鎖住
103	讀取功率不足
104	非特定的錯誤
105	CRC錯誤
106	寫入中若發生錯誤時，回覆已寫入多少 words 數
107	寫入中若 Tag 標籤回覆錯誤時，錯誤碼加上已寫入多少 words 數
108	沒有標籤存在
109	指令格式錯誤
110	設定電源強度失敗
111	設定法規失敗

附件三

錯誤代碼	錯誤代碼說明
00	沒有查詢到電子標籤
05	訪問密碼錯誤
FA	有電子標籤,但通信不暢,操作失敗
FB	無電子標籤可操作
FC	電子標籤返回錯誤
FD	命令長度錯誤
FE	不合法的命令
FF	參數錯誤