

Gainscha Android Bluetooth/Ethernet/USB

Function Library Instructions

1. GTSPL_openPort()

★ Bluetooth

GTSPL_openPort(MacAddress)

■ Description: Start the printer spool.

■ Parameter:

➔ address: String Type, Specifies the Bluetooth MacAddress(BR/EDR MacAddress).

example: " DC:1D:30:00:1D:87"

★ Ethernet

GTSPL_openPort(IP, Port)

■ Description: Start the printer spool.

■ Parameter:

➔ IP: String Type, IP Address, example: "192.168.1.109"

➔ Port: int Type, Port Number, example: 8899

★ USB

GTSPL_openPort(manager, device)

■ Description: Start the printer spool.

■ Parameter:

➔ **manager**: USBManager Class, pass USBManager variables.

➔ **device**: USBDevice Class, pass USBDevice variables.

2. GTSPL_closePort()

- Description: Close printer spool.
- Parameter: None

3. GTSPL_closePort(time)

- Description: Close printer spool.
- Parameter:
 - ➔ time: int Type, delay time, 1000=1 second, example: GTSPL_closePort(2000)

4. GTSPL_setCmdSendMode(mode)

- Description: Set the built-in commands would send to the printer or to file.
- Parameter:
 - ➔ mode: String Type
 - F: Send the built-in commands to file.
(File will be saved in InternalStorage/android/data/packageName/files)
 - P: Send the built-in commands to the printer.

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

- Description: Set up label width, label height, print speed, print density, sensor type, gap/black mark vertical distance and gap/black mark offset distance.
- Parameter:

Parameter	Type	Description
width	int	Set up label width; unit: mm.
height	int	Set up label height; unit: mm.
speed	int	Set up print speed (1~15: print speed at 1"~15"/sec). Selectable print speeds depend on different printer models, and maximum speed is 15"/sec.
density	int	Set up print density(0~15);If the number is bigger, the printout will be darker.

sensor	int	Set up the sensor type for the media. 0: Gap sensor 1: Black mark sensor
sensorDistance	int	Set up vertical gap height of the gap/black mark; unit: mm
sensorOffset	int	Set up offset distance of the gap/black mark, unit: mm, this parameter is set to 0 when the general label is used.
context	Context	Pass current context.

6. GTSPL_clearBuffer(context)

- Description: Clear the printer image buffer.
- Parameter:
 - ➔ context: Context Class, Pass current context.

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

- Description: Use built-in bar code formats to print
- Parameter:

Parameter	Type	Description
x	int	The starting point of the bar code along the X direction, given in dot.
y	int	The starting point of the bar code along the Y direction, given in dot
type	String	Set up the Code Type, refer to Appendix.
height	int	Set up bar code height, given in dots.
readable	int	Set up whether to print human recognizable interpretation (text) or not. 0: prints no interpretation 1: prints interpretation
rotation	int	Set up rotation. 0: rotates 0 degree 90: rotates 90 degree s 180: rotates 180 degrees 270: rotates 270 degrees
narrow	int	Set up narrow bar ratio, refer to Appendix.

wide	int	Set up wide bar ratio, refer to Appendix.
content	String	The bar code content.
context	Context	Pass current context.

8. GTSPL_formFeed(context)

- Description: Jump to next top of label. This function should be used after setup function.
- Parameter:
 - ➔ context: Context Class, Pass current context.

9. GTSPL_noBackFeed(context)

- Description: Set the paper does not backfeed.
- Parameter:
 - ➔ context: Context Class, Pass current context.

10. GTSPL_sendCommand (context, command)

- Description: Sends built-in commands to the printer.
- Parameter:
 - ➔ command: String Type, refer to TSPL programming manual commands for details.
 - ➔ context: Context Class, Pass current context.

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

- Description: Use printer built-in fonts to print.
- Parameter:

Parameter	Type	Description
x	int	The starting point of text (character string) along the X direction, given in dots.
y	int	The starting point of text (character string) along the Y direction, given in dots.
size	String	Built-in font type

		1: 8*/12 dots 2: 12*20 dots 3: 16*24 dots 4: 24*32 dots 5: 32*48 dots TST24.BF2: Traditional Chinese 24*24 TST16.BF2: Traditional Chinese 16*16 TSS24.BF2: Simplified Chinese 24*24 TSS16.BF2: Simplified Chinese 16*16
rotation	int	Set up rotation of the text(character string). 0: rotates 0 degree 90: rotates 90 degrees 180: rotates 180 degrees 270: rotates 270 degrees
x_scale	int	Set up the magnification rate of text (character string) along the X direction, range: 1~8
y_scale	int	Set up the magnification rate of text (character string) along the Y direction, range: 1~8
content	String	The content of text(character string).
context	Context	Pass current context.

12. GTSPL_printLabel(set, copy, context)

- Description: Print the label content.
- Parameter:
 - ➔ set: int Type, set the number of print labels (set).
 - ➔ copy: int Type, set the number of copies printed (copy).
 - ➔ context: Context Class, Pass current context.

13. GTSPL_downloadPCX(filename, context)

- Description: Download the monochrome PCX format file to the printer.
- Parameter:
 - ➔ filename: String Type, File name

(File has to be saved in InternalStorage/android/data/packageName/files)

➔ context: Context Class, Pass current context.

14. GTSPL_downloadBMP(filename, context)

■ Description: Download the monochrome BMP format file to the printer.

■ Parameter:

➔ filename: String Type, File name

(File has to be saved in InternalStorage/android/data/packageName/files)

➔ context: Context Class, Pass current context.

15. GTSPL_downloadTTF(filename, context)

■ Description: Download the True Type Font file to the printer.

■ Parameter:

➔ filename: String Type, File name

(File has to be saved in InternalStorage/android/data/packageName/files)

➔ context: Context Class, Pass current context.

16. GTSPL_printerStatus(delaytime)

■ Description: Response the printer status. (Assign the return value to a String variable.)

■ Parameter:

➔ delaytime: int Type, Set up delay time.

■ Return Value(String Type):

Return Value	Printer Status
00	Normal
01	Head opened
02	Paper Jam
03	Paper Jam and head opened
04	Out of paper
05	Out of paper and head opened

08	Out of ribbon
09	Out of ribbon and head opened
0A	Out of ribbon and paper jam
0B	Out of ribbon, paper jam and head opened
0C	Out of ribbon and out of paper
0D	Out of ribbon, out of paper and head opened
10	Pause
20	Printing
80	Other error

17. **GTSP_L_getSDKVersion (returnWay, context)**

- Description: Response the SDK version.
- Parameter:
 - ➔ retrunWay: int Type, pass 0 will return the SDK version and popup the SDK version Dialog.
 - ➔ context: Context Class, Pass current context.

Gainscha Android Bluetooth Examples

1. Add the following permissions in AndroidManifest.xml first:

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

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

2. Import GTSPS_SDK SDK:

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

3. Example:

```
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\"\n\n");

        mGtsplCmdTest.GTSPL_printLabel(1, 1, this);

        String status = mGtsplCmdTest.GTSPL_printersStatus(1000);

        mGtsplCmdTest.GTSPL_closePort(1000);

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

    }

}
```


Gainscha Android Ethernet Examples

1.Add the following permissions in AndroidManifest.xml first:

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

2.Import GTSP_L_SDK SDK:

```
import com.gainscha.gtsp_l_sdk.GTSP_L_WIFIActivity;
```

3.Example:

```
public class MainActivity extends AppCompatActivity {  
  
    GTSP_L_WIFIActivity mGtspLWIFICmdTest = new GTSP_L_WIFIActivity();  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_main);  
  
        mGtspLWIFICmdTest.GTSP_L_setCmdSendMode("P");  
  
        mGtspLWIFICmdTest.GTSP_L_openPort("192.168.1.109", 8899);  
  
        mGtspLWIFICmdTest.GTSP_L_setup(62, 45, 2, 6, 0, 3, 0, this);  
  
        mGtspLWIFICmdTest.GTSP_L_sendCommand(this, "DIRECTION 1\n\n");  
  
        mGtspLWIFICmdTest.GTSP_L_clearBuffer(this);  
  
        mGtspLWIFICmdTest.GTSP_L_printerFont(100,10,"5",0,1,1,"Print Font 123456",this);  
  
        mGtspLWIFICmdTest.GTSP_L_barcode(30, 30, "128", 100, 1, 0, 2, 2, "barcode1234567", this);  
  
        mGtspLWIFICmdTest.GTSP_L_downloadBMP("CIRCLE.BMP", this);  
  
        mGtspLWIFICmdTest.GTSP_L_sendCommand(this, "PUTBMP 150,30,\"CIRCLE.BMP\"\\r\\n");  
  
        mGtspLWIFICmdTest.GTSP_L_printLabel(1, 1, this);  
  
        String status = mGtspLWIFICmdTest.GTSP_L_printersStatus(1000);  
  
        mGtspLWIFICmdTest.GTSP_L_closePort();  
  
        String version=mGtspLWIFICmdTest.GTSP_L_getSDKVersion(0,this);  
  
    }  
}
```

Gainscha Android USB Examples

1.Import GTSPS_SDK:

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

2.Example:

```
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);

mUSB.GTSPL_closePort();

```

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

```
}
```

Appendix

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		