BIXOLON®

API 레퍼런스 가이드

Label Printer SDK

Android

Ver. 1.25

www.bixolon.co.kr

목차

저작권 **6**

1. 매뉴얼 안내 **7**

1-1 지원 정보 7

1-2 지원 장치 및 인터페이스 목록 8

1-3 매뉴얼 내 숫자 표기법 10

1-4 각 모델 별 좌표의 사용 범위 11

1-5 패키지 콘텐츠 13

1-5-1 매뉴얼 13

1-5-2 라이브러리 13

1-5-3 샘플 소스 코드 13

1-6 지원 함수 목록 14

2. Label Printer SDK Class Reference **16**

2-1 개요 16

2-2 Methods 16

2-2-1 Bixolon Label Printer 16

2-2-2 Find Bluetooth Printers 19

2-2-3 Find Network Printers 20

2-2-4 Find USB Printers 21

2-2-5 Connect (블루투스 Classic 전용) 22

2-2-6 Connect (블루투스 Classic/BLE) 23

2-2-7 Connect (Wi-Fi/Ethernet) 24

2-2-8 Connect (USB 전용) 25

2-2-9 Is Connected 26

2-2-10 Disconnect 27

2-2-11 Print 28

2-2-12 Begin Transaction Print 29

2-2-13 End Transaction Print 31

2-2-14 Draw Text 33

2-2-15 Draw Vector Font Text 36

2-2-16 Draw 1d Barcode 38

2-2-17 Draw Maxi code 41

2-2-18 Draw PDF 417 42

2-2-19 Draw QR Code 45

2-2-20 Draw Data Matrix 47

2-2-21 Draw Aztec 48

2-2-22 Draw Code 49 50

2-2-23 Draw Coda Block 52

2-2-24 Draw Micro PDF 417 54
2-2-25 Draw IMB Barcode 56
2-2-26 Draw MSI Barcode 57
2-2-27 Draw Plessey Barcode 60
2-2-28 Draw TLC39 Barcode 62
2-2-29 Draw RSS Barcode 64
2-2-30 Draw Block 66
2-2-31 Draw Two Block 67
2-2-32 Draw Circle 69
2-2-33 Draw Bitmap 70
2-2-34 Draw Bitmap 71
2-2-35 Draw Compression Image 72
2-2-36 Draw Compression Image 73
2-2-37 Draw Image 74
2-2-38 Draw Image 76
2-2-39 Draw Base64 Image 78
2-2-40 Draw Image File 80
2-2-41 Get Status 82
2-2-42 Set Auto Cutter 84
2-2-43 Get Printer Information 85
2-2-44 Print Information 86
2-2-45 Initialize Printer 87
2-2-46 Set Orientation 88
2-2-47 Set Character Set 89
2-2-48 Set Printing Type 91
2-2-49 Set Margin 92
2-2-50 Set Back Feed Option 93
2-2-51 Set Buffer Mode 94
2-2-52 Clear Buffer 95
2-2-53 Set Length 96
2-2-54 Set Rewinder 97
2-2-55 Set Speed 98
2-2-56 Set Offset 99
2-2-57 Set Density 100
2-2-58 Set Cutter Position 101
2-2-59 Set Width 102
2-2-60 Execute Direct Io 103
2-2-61 Execute Direct Io 104
2-2-62 Execute Direct Io 105
2-2-63 Execute Direct Io 106
2-2-64 Setup RFID 107
2-2-65 Set RFID Position 108
2-2-66 Set EPC Data Structure 109
2-2-67 Write RFID 110

2-2-68 Set RFID Password 111
2-2-69 Lock RFID 112
2-2-70 Transfer File 113
2-2-71 Transfer File 114
2-2-72 Draw PDF File 115
2-2-73 Set PDF Dpi 116
2-2-74 Get PDF Page 117
2-2-75 Get Count PDF Pages 118
2-2-76 Get PDF Page Height 119
2-2-77 Disable Inactivity Time 120
2-2-78 Firmware Download 121
2-2-79 WLAN Firmware Download 122
2-2-80 Get WLAN Info 123
2-2-81 Get WLAN Info 124
2-2-82 Set WLAN Info 125
2-2-83 Set Binary Certificate File 126
2-2-84 Set Pem Certificate File 127
2-2-85 Update Certificate File 128

3. 상수값 129
3-1 Alignments 129
3-1-1 Device Font Alignment 129
3-1-2 Vector Font Alignment 129
3-2 Barcode HRI 130
3-3 MaxiCode Modes 131
3-4 1D Barcode Types 131
3-5 Barcode Origin Point 131
3-6 Error Correction Level 132
3-7 Data Compression Method 132
3-8 QRCode Model 132
3-9 Code 49 Starting Mode 133
3-10 Codablock Mode 133
3-11 Check Digit Option 134
3-12 RSS Barcode Type 134
3-13 Rotation Degrees 134
3-14 Device Fonts 135
3-15 Vector Fonts 135
3-16 Draw Block Options 136
3-17 Draw Circle Sizes 136
3-18 International Character Set 137
3-19 Code Pages 138
3-20 Printing Type 139
3-21 Media Type 139
3-22 Speed Value 139

3-23 Orientation 139
3-24 Printer Status 140
3-25 Printer Information 140
3-26 Micro PDF 417 Mode List 141
3-27 PDF 417 Barcode HRI 142
3-28 CODE 49 Barcode HRI 142
3-29 PLESSEY Barcode HRI 142
3-30 MSI Barcode HRI 142

4. 부록 143
4-1 개발 환경 설정 143
4-1-1 Manifest 권한 설정 143
4-1-2 안드로이드 장치 연결 144
4-1-3 Android 장치 개발자 옵션 설정 148
4-1-4 Net Configuration Tool 사용 149
4-1-5 gradle 파일 설정 151

저작권

© BIXOLON Co., Ltd. 모든 권한을 소유합니다.

이 사용설명서와 제품에 사용된 저작물은 저작권법에 의해 보호되어 있습니다.
(주)빅솔론의 사전 서면 동의 없이 사용 설명서 및 제품에 사용된 저작물에 대한 일부 또는 전체를 무단으로 복제, 저장, 전송하는 것을 금합니다.
제공된 정보는 본 제품에만 해당되며 다른 제품에 대해서는 적용되지 않습니다.
또한 본 정보 사용으로 인해 발생하는 직/간접적 손해에 대해 책임지지 않습니다.

- 빅솔론 로고는 (주)빅솔론의 등록상표입니다.
- 모든 다른 상표 또는 제품 이름은 해당하는 회사 또는 조직의 상표입니다.

(주)빅솔론은 제품의 기능과 품질 향상을 위하여 지속적인 개선을 하고 있습니다.
이로 인하여 제품의 사양과 매뉴얼의 내용은 사전 통보 없이 변경될 수 있습니다.

사용시 주의사항

프린터와 같은 전자 제품은 정전기에 의해 쉽게 훼손될 수 있습니다.
정전기로부터 프린터를 보호하기 위해서는 프린터 후면 부에 케이블을 연결하거나 제거하기 전에 반드시 프린터 전원을 꺼주세요.
만약 프린터가 정전기로부터 손상을 입었을 경우에는 가까운 구입처에 문의해주세요.

1. 매뉴얼 안내

이 SDK 매뉴얼에는 Android용 응용 프로그램 개발에 필요한 라이브러리에 대한 내용이 기술되어 있습니다.

1-1 지원 정보

• Android OS 6.0 이상

1-2 지원 장치 및 인터페이스 목록

모델명	Wi-Fi	Ethernet	Bluetooth	BLE	USB
SLP-DX420	O	O	X	X	O
SLP-DX423	O	O	X	X	O
SLP-DX220	O	O	O	X	O
SLP-DX223	O	O	O	X	O
SLP-TX420	O	O	X	X	O
SLP-TX423	O	O	X	X	O
SLP-TX220	O	O	X	X	O
SLP-TX223	O	O	X	X	O
SLP-DL410	O	O	X	X	O
SLP-DL413	O	O	X	X	O
SLP-TX400	O	O	X	X	O
SLP-TX403	O	O	X	X	O
SRP-770III	O	O	X	X	O
SRP-E770III	X	O	X	X	O
XT5-40(RFID)	O	O	O	O	O
BT5-40(RFID)	O	O	O	O	O
XT5-43(RFID)	O	O	O	O	O
BT5-43(RFID)	O	O	O	O	O
XT5-46(RFID)	O	O	O	O	O
BT5-46(RFID)	O	O	O	O	O
SPP-L3000	O	X	O	O	O
SPP-L310	O	X	O	O	O
SPP-L410	O	X	O	O	O
XQ-840	X	X	X	X	O
XQ-843	X	X	X	X	O
XQ-840II	X	X	X	X	O
XQ-843II	X	X	X	X	O
XL5-40CT	O	O	O	O	O
XL5-43CT	O	O	O	O	O
XD5-40d	O	O	O	O	O
BD5-40d	O	O	O	O	O
XD5-40IId	O	O	O	O	O
XD5-43d	O	O	O	O	O
BD5-43d	O	O	O	O	O
XD5-43IId	O	O	O	O	O
XD5-40t(RFID)	O	O	O	O	O
BD5-40t(RFID)	O	O	O	O	O
XD5-40IIt(RFID)	O	O	O	O	O
XD5-43t(RFID)	O	O	O	O	O
BD5-43t(RFID)	O	O	O	O	O
XD5-43IIt(RFID)	O	O	O	O	O

XM7-40(RFID)	O	X	O	O	O
XM7-20	O	O	O	O	O
XM7-30	O	O	O	O	O
SRP-S3000_LABEL	O	O	X	X	O
XT3-40	O	O	O	O	O
BT3-40	O	O	O	O	O
XT3-43	O	O	O	O	O
BT3-43	O	O	O	O	O
XM5-30	O	X	O	O	O
XT6-60	O	O	O	O	O
XT6-63	O	O	O	O	O



XM7-20에서 Ethernet 사용시에는 별도의 크래들 필요(옵션)

※ BLE: Bluetooth Low Energy

1-3 매뉴얼 내 숫자 표기법

• 매뉴얼 내의 숫자 표기는 10진수로 작성되어 있으나 “0x” 로 시작하는 숫자 표기의 경우 16진수에 해당됩니다.

[예시]

10진수와 16진수의 구분

값	10진수 표기	16진수 표기
4	4	0x04
10	10	0x0A
76	76	0x4C

1-4 각 모델 별 좌표의 사용 범위

모델명	최소 너비	최대 너비
SLP-DX420	0	864
SLP-DX423	0	1248
SLP-DX220	0	432
SLP-DX223	0	672
SLP-TX420	0	864
SLP-TX423	0	1248
SLP-TX220	0	432
SLP-TX223	0	672
SLP-DL410	0	864
SLP-DL413	0	1248
SLP-TX400	0	864
SLP-TX403	0	1248
SRP-770III	0	832
SRP-E770III	0	832
XT5-40(RFID)	0	832
XT5-43(RFID)	0	1248
XT5-46(RFID)	0	2496
SPP-L3000	0	576
SPP-L310	0	576
SPP-L410	0	832
XQ-840	0	832
XQ-843	0	1232
XQ-840II	0	832
XQ-843II	0	1232
XL5-40CT	0	832
XL5-43CT	0	1248
XD5-40d	0	832
XD5-40IId	0	832
XD5-43d	0	1248
XD5-43IId	0	1248
XD5-40t(RFID)	0	832
XD5-40IIt(RFID)	0	832
XD5-43t(RFID)	0	1232
XD5-43IIt(RFID)	0	1232
XM7-40(RFID)	0	832
XM7-20	0	384
XM7-30	0	576
SRP-S3000_LABEL	0	576
XT3-40	0	864
XT3-43	0	1248
BT3-40	0	864

BT3-43	0	1248
BT5-40	0	832
BT5-43	0	1248
BT5-46	0	2496
BD5-40d	0	832
BD5-43d	0	1248
BD5-40t	0	832
BD5-43t	0	1248
XM5-30	0	576
XT6-60	0	1344
XT6-63	0	1984

1-5 패키지 콘텐츠

1-5-1 매뉴얼

- Manual_Label_Printer_SDK_FOR_Android_API_Reference_Guide_KOR_V*.**
- Manual_Label_Printer_SDK_FOR_Android_API_Reference_Guide_ENG_V*.**

※ 매뉴얼 폴더를 참조하세요.

1-5-2 라이브러리

라이브러리 위치/이름	설명
libs/BixolonLabelPrinter_V[xxx].jar	Label Printer 사용을 위한 라이브러리
libs/libcommon_Vxxx.jar	프린터 제어 코어 라이브러리
libs/jniLibs/ABI 타입/libbxl_common.so	프린터 제어 native 라이브러리

※ 대괄호 안의 내용은 해당 라이브러리의 버전에 따라 변경됩니다.

1-5-3 샘플 소스 코드

샘플 위치/이름	설명
sample/BixolonLabelPrinterSample	프린터 사용에 대한 샘플 어플리케이션

1-6 지원 함수 목록

함수명		비고
General	BixolonLabelPrinter	-
Search	findBluetoothPrinter	-
	findNetworkPrinter	-
	findUsbPrinter	-
Connection	connect	-
	isConnected	-
	disconnect	-
Print	print	-
	beginTransactionPrint	-
	endTransactionPrint	-
Text	drawText	-
	drawVectorFontText	-
Barcode	draw1dBarcode	-
	drawMaxicode	-
	drawPdf417	-
	drawQrCode	-
	drawDataMatrix	-
	drawAztec	-
	drawCode49	-
	drawCodaBlock	-
	drawMicroPDF417	-
	drawIMBBarcode	-
	drawMSIBarcode	-
	drawBarcodePlessey	-
	drawTLC39Barcode	-
	drawRSSBarcode	-
Block & Circle	drawBlock	-
	drawTowBlock	-
	drawCircle	-
Image	drawBitmap	-
	drawCompressionImage	-
	drawBase64Image	-
Status	getStatus	-
Cut	setAutoCutter	커터 장착 모델에 한함
Information	getPrinterInformation	-
	printInformation	-

함수명		비고
Printer Setting	initializePrinter	-
	setOrientation	-
	setCharacterSet	-
	setPrintingType	-
	setMargin	-
	setBackFeedOption	-
	setLength	-
	setBufferMode	-
	clearBuffer	-
	setRewinder	Rewinder 장착 모델에 한함
	setSpeed	-
	setOffset	-
	setDensity	-
	setCutterPositionb	-
	setWidth	-
Direct I/O	executeDirectIo	-
RFID Setting	setupRFID	RFID 지원 모델에 한함
	setRFIDPosition	
	setEPCDataStructure	
	writeRFID	
	setRFIDPassword	
	lockRFID	

2. Label Printer SDK Class Reference

2-1 개요

• BixolonLabelPrinter Class는 프린터를 제어하는 메인 객체입니다.

2-2 Methods

2-2-1 Bixolon Label Printer

BixolonLabelPrinter의 객체를 생성합니다.

[선언]

• void BixolonLabelPrinter(Context context, Handler handler, Looper looper);

[매개변수]

• Context context: 시스템 서비스를 사용하는 UI 컨텍스트

• Handler handler: 이벤트를 수신할 메시지 Handler

• Looper looper: Message Queue를 처리할 Looper Message Queue를 따로 처리하지 않을 경우에는 null 입력



Handler가 작성되지 않은 경우에는 라이브러리에서 보내는 이벤트를 수신할 수 없습니다.

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter;

  private Handler backHandler;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    ...
    backgroundThread.start();
    mPrinter = new BixolonLabelPrinter(this, backHandler, Looper.myLooper());
  }

  Handler mainHandler = new Handler(){
    @Override
    public void handleMessage(Message msg){
      switch(
        case BixolonLabelPrinter.MESSAGE_STATE_CHANGE:
        ...
        break;
        case ...
        ...
      }
    }
  }
  private Thread backgroundThread = new Thread(){
    @Override
    public void run() {
      Looper.prepare();
       backHandler = new Handler(Looper.myLooper()){
         @Override
         public void handleMessage(Message msg) {
           switch(msg.what){
           case BixolonLabelPrinter.MESSAGE_STATE_CHANGE:
              Message message = new Message();
              message.what = msg.what;
              message.obj = msg.obj;
              message.arg1 = msg.arg1;
              ...
              mainHandler.sendMessage(message);
              break;
           case ...
              ...
              break;
           }
```

```
            }
        };
        Looper.loop();
    }
  };
}
```

2-2-2 Find Bluetooth Printers

페어링 된 블루투스 프린터를 검색합니다.

[선언]

• void findBluetoothPrinter();

[반환 값]

• Set<BluetoothDevice>: **페어링 된 프린터 목록**

• null: **페어링 된 블루투스 프린터가 없거나, 휴대폰의 블루투스가 꺼져 있음**

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      …
      mPrinter.findBluetoothPrinter();
   }

   private final Handler mHandler = new Handler() {
      @Override
      public void handleMessage(Message msg) {
         switch (msg.what) {
            …
            case BixolonLabelPrinter.MESSAGE_BLUETOOTH_DEVICE_SET:
               if(msg.obj == null) {
                  Log.i(“TAG”, “Bluetooth Device not found !”);
               } else {
                  Set<BluetoothDevice> devices = (Set<BluetoothDevice>)msg.obj;
               }
               break;
            }
       }
   }
}
```

2-2-3 Find Network Printers

Network 프린터를 검색합니다.

[선언]

• void findNetworkPrinters(int timeout);

[매개변수]

• int timeout: 프린터 검색 시간 (단위: 밀리초)

[반환 값]

• Set<String>: 검색된 Ethernet/Wi-Fi 프린터 목록

• null: 검색된 프린터가 없음

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    ...
    mPrinter.findNetworkPrinters ();
  }

  private final Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
      switch (msg.what) {
        ...
        case BixolonLabelPrinter.MESSAGE_NETWORK_DEVICE_SET:
          if(msg.obj == null) {
              Log.i("TAG", "Network Device not found !");
          } else {
              Set<String> devices = (Set<String>)msg.obj;
          }
          break;
        }
     }
  }
}
```

2-2-4 Find USB Printers

USB 프린터를 검색합니다.

[선언]

• void findUsbPrinters()

[반환 값]

• Set<UsbDevice>: 검색된 USB 목록

• null: 연결된 USB가 없음

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      …
      mPrinter.findUsbPrinter();
   }

   private final Handler mHandler = new Handler() {
      @Override
      public void handleMessage(Message msg) {
         switch (msg.what) {
            …
            case BixolonLabelPrinter.MESSAGE_USB_DEVICE_SET:
               if(msg.obj == null) {
                  Log.i(“TAG”, “USB Device not found !”);
               } else {
                  Set<UsbDevice> devices = (Set<UsbDevice>)msg.obj;
               }
               break;
            }
       }
   }
}
```

2-2-5 Connect (블루투스 Classic 전용)

프린터에 연결을 시도합니다.

[선언]

• void connect(String address)

[매개변수]

• String address: 블루투스 프린터 Mac Address

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    ...
    String address = “74:F0:7F:xx:xx:xx”;
    mPrinter.connect(address);
  }

  private final Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
      switch (msg.what) {
        case BixolonLabelPrinter.MESSAGE_STATE_CHANGE:
          switch (msg.arg1) {
            case BixolonLabelPrinter.STATE_CONNECTED:
              Log.i(”TAG”, “Device is connected !”);
            break;
            case BixolonLabelPrinter.STATE_CONNECTING:
              Log.i(“TAG”, “Device is connecting !”);
            break;
            case BixolonLabelPrinter.STATE_NONE:
              Log.i(“TAG”, “connect is failed or disconnected!”);
            break;
          }
        }
      }
    }
  }
}
```

2-2-6 Connect (블루투스 Classic/BLE)

프린터에 연결을 시도합니다.

[선언]

• void connect(String address, int type)

[매개변수]

• String address: 프린터의 Mac Address

• int type: 블루투스 종류 (0: Classic/1: BLE)

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    ...
    String address = “74:F0:7F:xx:xx:xx”;
    Int type = BixolonLabelPrinter.BLUETOOTH_CLASSIC;
    mPrinter.connect(address, type);
  }

  private final Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
      switch (msg.what) {
        case BixolonLabelPrinter.MESSAGE_STATE_CHANGE:
          switch (msg.arg1) {
            case BixolonLabelPrinter.STATE_CONNECTED:
              Log.i(“TAG”, “Device is connected !”);
            break;
            case BixolonLabelPrinter.STATE_CONNECTING:
              Log.i(“TAG”, “Device is connecting !”);
            break;
            case BixolonLabelPrinter.STATE_NONE:
             Log.i(“TAG”, “connect is failed or disconnected!”);
            break;
          }
      }
    }
  }
}
```

2-2-7 Connect (Wi-Fi/Ethernet)

프린터에 연결을 시도합니다.

[선언]

• void connect(String address, int port, int timeout)

[매개변수]

• String address: 프린터의 IP 주소

• int port: 프린터의 port 번호 (기본값: 9100)

• int timeout: 프린터의 최대 연결 시간

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    …
    String address = “192.168.x.x”;
    Int port = 9100;
    Int timeout = 5000;
    mPrinter.connect(address, port, timeout);
  }

  private final Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
      switch (msg.what) {
        case BixolonLabelPrinter.MESSAGE_STATE_CHANGE:
          switch (msg.arg1) {
            case BixolonLabelPrinter.STATE_CONNECTED:
              Log.i(“TAG”, “Device is connected !”);
            break;
            case BixolonLabelPrinter.STATE_CONNECTING:
              Log.i(“TAG”, “Device is connecting !”);
            break;
            case BixolonLabelPrinter.STATE_NONE:
              Log.i(“TAG”, “connect is failed or disconnected!”);
            break;
          }
      }
    }
  }
}
```

2-2-8 Connect (USB 전용)

프린터에 연결을 시도합니다.

[선언]

• void connect(UsbDevice device)

[매개변수]

• UsbDevice device: 현재 연결된 USB 프린터

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    ...
    mPrinter.connect(UsbDevice);
  }

  private BroadcastReceiver mUsbReceiver = new BroadcastReceiver(){
    @Override
    public void onReceive(Context context, Intent intent) {
      String action = intent.getAction();
      if(UsbManager.ACTION_USB_DEVICE_ATTACHED.equals(action)) {
        Log.i("TAG", "USB is connected !");
      } else if(UsbManager.ACTION_USB_DEVICE_DETACHED.equals(action)) {
        Log.i("TAG", "USB is disconnected !");
      }
    }
  };
}
```

2-2-9 Is Connected

프린터 연결여부를 확인합니다.

[선언]

• boolean isConnected();

[반환 값]

• true: 연결되어 있음

• false: 연결되지 않음

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      ...
      boolean isConnected = mPrinter.isConnected();

      if(isConnected) {
         Log.i("TAG", "Device is connected !");
      } else {
         Log.i("TAG", "connect is failed !");
      }
   }
}
```

2-2-10 Disconnect

프린터 연결을 해제합니다.

[선언]

• void disconnect();

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      disconnect();
   }
}
```

2-2-11 Print

프린터의 버퍼에 쌓여 있는 내용을 인쇄합니다.

[선언]

• void print(int set, int copy);

[매개변수]

• int set: 라벨의 세트 수

• int copy: 인쇄 매수

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      ...

      if(mPrinter.isConnected) {
         int set = 1;
         int copy = 1;
         mPrinter.print(set, copy);
      } else {
         return;
      }
   }
}
```

2-2-12 Begin Transaction Print

Transaction 인쇄를 시작합니다.

[선언]

• void beginTransactionPrint();



- Transaction 인쇄는 프린터 명령어를 별도 buffer에 저장해 두었다 한 번에 프린터로 전송하는 인쇄 방식입니다.
- beginTransactionPrint 메소드 이후의 프린터 명령어는 endTransactionPrint 메소드가 나올 때까지 프린터로 전송되지 않고 Buffer에 저장됩니다.

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    mPrinter.beginTrancsactionPrint();

    mPrinter.drawText("Bixolon Device Font Text test !!"
                      , 100
                      , 80
                      , BixolonLabelPrinter.FONT_SIZE_10
                      , 1
                      , 1
                      , 0
                      , BixolonLabelPrinter.ROTATION_90_DEGREES
                      , false
                      , true
                      , BixolonLabelPrinter.TEXT_ALIGNMENT_LEFT);

     mPrinter.draw1dBarcode("1234567890128"
                            , 90
                            , 200
                            , BixolonLabelPrinter.BARCODE_CODE128
                            , 1
                            , 2
                            , 240
                            , BixolonLabelPrinter.ROTATION_NONE
                            , 0
                            , 0);

    mPrinter.print(1,1);
    mPrinter.endTransactionPrint();
  }
  private final Handler mHandler = handleMessage(msg) -> {
    switch (msg.what) {
      ...
      case BixolonLabelPrinter.MESSAGE_OUTPUT_COMPLETE:
        Toast.makeText(this, "Print complete", Toast.LENGHT_SHORT).show();
        break;
    }
  }
}
```

2-2-13 End Transaction Print

Transaction 인쇄를 끝내고 프린터로 buffer에 저장된 명령어를 전송합니다.

[선언]

• void endTransactionPrint();



핸들러 메시지 BixolonLabelPrinter.MESSAGE_OUTPUT_COMPLETE을 사용해 전송 완료를 확인할 수 있습니다.

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    mPrinter.beginTrancsactionPrint();

    mPrinter.drawText("Bixolon Device Font Text test !!"
                      , 100
                      , 80
                      , BixolonLabelPrinter.FONT_SIZE_10
                      , 1
                      , 1
                      , 0
                      , BixolonLabelPrinter.ROTATION_90_DEGREES
                      , false
                      , true
                      , BixolonLabelPrinter.TEXT_ALIGNMENT_LEFT);

     mPrinter.draw1dBarcode("1234567890128"
                            , 90
                            , 200
                            , BixolonLabelPrinter.BARCODE_CODE128
                            , 1
                            , 2
                            , 240
                            , BixolonLabelPrinter.ROTATION_NONE
                            , 0
                            , 0);
    mPrinter.print(1,1);
    mPrinter.endTransactionPrint();
  }
  private final Handler mHandler = handleMessage(msg) -> {
    switch (msg.what) {
      ...
      case BixolonLabelPrinter.MESSAGE_OUTPUT_COMPLETE:
        Toast.makeText(this, "Print complete", Toast.LENGHT_SHORT).show();
        break;
    }
  }
}
```

2-2-14 Draw Text

이미지 버퍼에 Device Font 문자를 입력합니다.

[선언]

• void drawText(String data, int horizontalPosition, int verticalPosition, int fontSize,
int horizontalMultiplier, int verticalMultiplier, int rightSpace,
int rotation, boolean reverse, boolean bold, int alignment)

[매개변수]

• String data: 출력을 원하는 문자열

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int fontSize: Font 선택

상수명	값	설명
FONT_SIZE_6	48	9 X 15 (dots)
FONT_SIZE_8	49	12 X 20 (dots)
FONT_SIZE_10	50	16 X 25 (dots)
FONT_SIZE_12	51	19 X 30 (dots)
FONT_SIZE_15	52	24 X 38 (dots)
FONT_SIZE_20	53	32 X 40 (dots)
FONT_SIZE_30	54	48 X 76 (dots)
FONT_SIZE_14	55	22 X 34 (dots)
FONT_SIZE_18	56	28 X 44 (dots)
FONT_SIZE_24	57	37 X 58 (dots)
FONT_SIZE_KOREAN1	97	16 X 16 (dots) (ASCII 9 X 15)
FONT_SIZE_KOREAN2	98	24 X 24 (dots) (ASCII 12 X 24)
FONT_SIZE_KOREAN3	99	20 X 20 (dots) (ASCII 12 X 20)
FONT_SIZE_KOREAN4	100	26 X 26 (dots) (ASCII 16 X 30)
FONT_SIZE_KOREAN5	101	20 X 26 (dots) (ASCII 16 X 30)
FONT_SIZE_KOREAN6	102	38 X 38 (dots) (ASCII 22 X 34)
FONT_SIZE_GB2312	109	24 X 24 (dots) (ASCII 12 X 24)
FONT_SIZE_BIG5	110	24 X 24 (dots) (ASCII 12 X 24)
FONT_SIZE_SHIFT_JIS	106	24 X 24 (dots) (ASCII 12 X 24)

• int horizontalMultiplier: Font의 가로 확대 배율 (범위: 1~4)

• int verticalMultiplier: Font의 세로 확대 배율 (범위: 1~4)

• int rightSpace: 문자의 오른쪽 자간 (ex: 5, +3, -10...)

• int rotation: 폰트의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전

• boolean reverse: 역상체 사용 (true: 사용, false: 미사용)

• boolean bold: 굵은 글씨 사용 여부 (true: 사용, false: 미사용)

• int alignment: 정렬 방식

상수명	값	설명
TEXT_ALIGNMENT_NONE	48	정렬 없음
TEXT_ALIGNMENT_LEFT	70	왼쪽 정렬
TEXT_ALIGNMENT_RIGHT	76	오른쪽 정렬
TEXT_ALIGNMENT_RIGHT_TO_LEFT	82	문자열을 오른쪽에서 왼쪽 방향으로 출력



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawText(){
    mPrinter.drawText("Bixolon Device Font Text test !!"
                      , 100
                      , 80
                      , BixolonLabelPrinter.FONT_SIZE_10
                      , 1
                      , 1
                      , 0
                      , BixolonLabelPrinter.ROTATION_90_DEGREES
                      , false
                      , true
                      , BixolonLabelPrinter.TEXT_ALIGNMENT_LEFT);
    mPrinter.print(1, 1);
  }

}
```

2-2-15 Draw Vector Font Text

이미지 버퍼에 Vector Font 문자를 입력합니다.

[선언]

• void drawVectorFontText(String data, int horizontalPosition, int verticalPosition,
int font, int width, int height, int rightSpace,
boolean bold, boolean reverse, boolean italic,
int rotation, int alignment, int direction)

[매개변수]

• String data: 출력을 원하는 문자열

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int font: Font 선택

상수명	값	설명
VECTOR_FONT_ASCII	85	ASCII (1Byte code)
VECTOR_FONT_KS5601	75	KS5601 (2Byte code)
VECTOR_FONT_BIG5	66	BIG5 (2Byte code)
VECTOR_FONT_GB2312	71	GB2312 (2Byte code)
VECTOR_FONT_SHIFT_JIS	74	Shift-JIS (2Byte code)
VECTOR_FONT_OCR_A	97	OCR-A (1Byte code)
VECTOR_FONT_OCR_B	98	OCR-B (1Byte code)

• int width: Font 폭 (범위: 1~1500)

• int height: Font 높이 (범위: 1~1500)

• int rightSpace: 오른쪽 여백 (예: 5, +3, -10...).

• boolean bold: 굵은 글씨 사용 여부 (true: 사용, false: 미사용)

• boolean reverse: 역상체 사용 여부 (true: 사용, false: 미사용)

• boolean Italic: *기울임체* 사용 여부 (true: 사용, false: 미사용)

• int rotation: 폰트의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전

• int alignment: 정렬 방식

상수명	값	설명
VECTOR_FONT_TEXT_ALIGNMENT_LEFT	76	왼쪽 정렬
VECTOR_FONT_TEXT_ALIGNMENT_RIGHT	82	오른쪽 정렬
VECTOR_FONT_TEXT_ALIGNMENT_CENTER	67	가운데 정렬
VECTOR_FONT_TEXT_DIRECTION_LEFT_TO_RIGHT	0	문자열을 왼쪽에서 오른쪽 방향으로 출력
VECTOR_FONT_TEXT_DIRECTION_RIGHT_TO_LEET	1	문자열을 오른쪽에서 왼쪽 방향으로 출력

• int direction: 문자열의 출력 방향



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawVectorFontText (){
    mPrinter.drawVectorFontText("Bixolon Vector Font Text Test"
            , 50
            , 100
            , BixolonLabelPrinter.VECTOR_FONT_KS5601
            , 40
            , 40
            , 0
            , true
            , false
            , false
            , BixolonLabelPrinter.ROTATION_NONE
            , BixolonLabelPrinter.VECTOR_FONT_TEXT_ALIGNMENT_LEFT
            , BixolonLabelPrinter.VECTOR_FONT_TEXT_DIRECTION_LEFT_TO_RIGHT
            );

    mPrinter.print(1, 1);
  }
}
```

2-2-16 Draw 1d Barcode

이미지 버퍼에 1차원 바코드를 입력합니다.

[선언]

• void draw1dBarcode(String data, int horizontalPosition, int verticalPosition,
int barcodeSelection, int narrowBarWidth,
int wideBarWidth, int height, int rotation, int hri,
int quietZoneWidth)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int barcodeSelection: 바코드 종류

상수명	값	설명
BARCODE_CODE39	0	Code39
BARCODE_CODE128	1	Code128
BARCODE_I2OF5	2	Interleaved 2of5
BARCODE_CODABAR	3	Codabar
BARCODE_CODE93	4	Code93
BARCODE_UPC_A	5	UPC-A
BARCODE_UPC_E	6	UPC-E
BARCODE_EAN13	7	EAN13
BARCODE_EAN8	8	EAN8
BARCODE_UCC_EAN128	9	UCC/EAN128
BARCODE_CODE11	10	Code11
BARCODE_PLANET	11	Planet
BARCODE_INDUSTRIAL_2OF5	12	Industrial 2of5
BARCODE_STANDARD_2OF5	13	Standard 2of5
BARCODE_LOGMARS	14	Logmars
BARCODE_UPC_EAN_EXTENSIONS	15	UPC/EAN Extensions
BARCODE_POSTNET	16	Postnet

• int narrowBarWidth: 좁은 바의 너비

• int wideBarWidth: 넓은 바의 너비

• int height: 바코드의 높이

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전

• int hri: 바코드 값의 출력 위치

상수명	값	설명
HRI_NOT_PRINTED	0	HRI를 사용하지 않음
HRI_BELOW_FONT_SIZE_1	1	HRI 위치: 바코드의 하단 Font Size: 1
HRI_ABOVE_FONT_SIZE_1	2	HRI 위치: 바코드의 상단 Font Size: 1
HRI_BELOW_FONT_SIZE_2	3	HRI 위치: 바코드의 하단 Font Size: 2
HRI_ABOVE_FONT_SIZE_2	4	HRI 위치: 바코드의 상단 Font Size: 2
HRI_BELOW_FONT_SIZE_3	5	HRI 위치: 바코드의 하단 Font Size: 3
HRI_ABOVE_FONT_SIZE_3	6	HRI 위치: 바코드의 상단 Font Size: 3
HRI_BELOW_FONT_SIZE_4	7	HRI 위치: 바코드의 하단 Font Size: 4
HRI_ABOVE_FONT_SIZE_5	8	HRI 위치: 바코드의 상단 Font Size: 4

• int quietZoneWidth: 여백 (범위: 0~20)



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  …
  private void drawBarcode (){
    mPrinter.draw1dBarcode(“1234567890128”
                          , 90
                          , 200
                          , BixolonLabelPrinter.BARCODE_CODE128
                          , 1
                          , 2
                          , 240
                          , BixolonLabelPrinter.ROTATION_NONE
                          , 0
                          , 0);

    mPrinter.print(1, 1);
  }
}
```

2-2-17 Draw Maxi code

이미지 버퍼에 MaxiCode 바코드를 입력합니다.

[선언]

• void drawMaxicode(String data, Int horizontalPosition, int verticalPosition,
int mode)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• Int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• Int mode: MaxiCode의 Mode

상수명	값	설명
_SDK_MAXICODE_MODE_0	0	MaxiCode Mode 0
_SDK_MAXICODE_MODE_2	2	MaxiCode Mode 2
_SDK_MAXICODE_MODE_3	3	MaxiCode Mode 3
_SDK_MAXICODE_MODE_4	4	MaxiCode Mode 4



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawBarcode (){
    mPrinter.drawMaxicode("Bixolon Maxicode Test"
                          , 100
                          , 100
                          , BixolonLabelPrinter.MAXICODE_MODE2);
    mPrinter.print(1, 1);
  print();
  }
}
```

2-2-18 Draw PDF 417

이미지 버퍼에 PDF417 바코드를 입력합니다.

[선언]

• void drawPdf417(String data, int horizontalPosition, int verticalPosition,
Int maxRow, int maxColumn, int errorCorrection\,
int compression, int hri, int orignPoint, int width,
int height, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int maxRow: 최대 가로 여백 (범위: 3~90)

• int maxColumn: 최대 세로 여백 (범위: 1~30)

• int errCorrection: 에러 보정 레벨

상수명	값	설명
PDF417_ERROR_CORRECTION_LEVEL0	0	에러 보정 레벨 0
PDF417_ERROR_CORRECTION_LEVEL1	1	에러 보정 레벨 1
PDF417_ERROR_CORRECTION_LEVEL2	2	에러 보정 레벨 2
PDF417_ERROR_CORRECTION_LEVEL3	3	에러 보정 레벨 3
PDF417_ERROR_CORRECTION_LEVEL4	4	에러 보정 레벨 4
PDF417_ERROR_CORRECTION_LEVEL5	5	에러 보정 레벨 5
PDF417_ERROR_CORRECTION_LEVEL6	6	에러 보정 레벨 6
PDF417_ERROR_CORRECTION_LEVEL7	7	에러 보정 레벨 7
PDF417_ERROR_CORRECTION_LEVEL8	8	에러 보정 레벨 8

• int compression: Data Compression Method

상수명	값	설명
DATA_COMPRESSION_TEXT	0	2char/codeword
DATA_COMPRESSION_NUMERIC	1	2.93 char/codeword
DATA_COMPRESSION_BINARY	2	1.2bytes/codeword

• int hri: HRI 출력 여부 선택

상수명	값	설명
PDF417_HRI_NOT_PRINTED	0	HRI 인쇄하지 않음
PDF417_HRI_BELOW_BARCODE	1	바코드 아래에 인쇄

• int barcodeOriginPoint: 바코드의 시작 지점

상수명	값	설명
BARCODE_ORIGIN_POINT_CENTER	0	기준점을 바코드의 한가운데로 지정합니다.
BARCODE_ORIGIN_POINT_UPPER_LEFT	1	기준점을 바코드의 좌측 상단으로 지정합니다.

• int width: 모듈 폭 (범위: 2~9)

• int height: 바코드의 높이 (범위: 4~99)

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void drawBarcode (){

      mPrinter.drawPdf417("Bixolon PDF417 Test"
                           , 100
                           , 100
                           , 10
                           , 5
                           , BixolonLabelPrinter.PDF417_ERROR_CORRECTION_LEVEL0
                           , BixolonLabelPrinter.DATA_COMPRESSION_TEXT
                           , BixolonLabelPrinter.PDF417_HRI_NOT_PRINTED
                           , BixolonLabelPrinter.BARCODE_ORIGIN_POINT_CENTER
                           , 5
                           , 50
                           , BixolonLabelPrinter.ROTATION_NONE);
      mPrinter.print(1, 1);
   }
}
```

2-2-19 Draw QR Code

이미지 버퍼에 QRCode 바코드를 입력합니다.

[선언]

• void drawQrCode(String data, int horizontalPosition, int verticalPosition,
int model, int eccLevel, int size, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int model: QRCode Model

상수명	값	설명
QR_CODE_MODEL1	1	QR Model 1
QR_CODE_MODEL2	2	QR Model 2

• int eccLevel: 에러 보정 레벨

상수명	값	설명
PDF417_ERROR_CORRECTION_LEVEL0	0	에러 보정 레벨 0
PDF417_ERROR_CORRECTION_LEVEL1	1	에러 보정 레벨 1
PDF417_ERROR_CORRECTION_LEVEL2	2	에러 보정 레벨 2
PDF417_ERROR_CORRECTION_LEVEL3	3	에러 보정 레벨 3
PDF417_ERROR_CORRECTION_LEVEL4	4	에러 보정 레벨 4
PDF417_ERROR_CORRECTION_LEVEL5	5	에러 보정 레벨 5
PDF417_ERROR_CORRECTION_LEVEL6	6	에러 보정 레벨 6
PDF417_ERROR_CORRECTION_LEVEL7	7	에러 보정 레벨 7
PDF417_ERROR_CORRECTION_LEVEL8	8	에러 보정 레벨 8

• int size: 바코드의 크기 (범위: 1~9)

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void drawBarcode (){
      mPrinter.drawQrCode("Bixolon QR Code Test"
                          , 100
                          , 150
                          , BixolonLabelPrinter.QR_CODE_MODEL1
                          , BixolonLabelPrinter.ECC_LEVEL_15
                          , 1
                          , BixolonLabelPrinter.ROTATION_NONE);
      mPrinter.print(1, 1);
   }
}
```

2-2-20 Draw Data Matrix

이미지 버퍼에 Data Matrix를 입력합니다.

[선언]

• void drawDataMatrix(String data, int horizontalPosition, int verticalPosition,
int size, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int size: 바코드의 크기 (범위: 1~4)

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawBarcode (){
    mPrinter.drawDataMatrix("Bixolon DataMatrix Test"
                            , 50
                            , 100
                            , 2
                            , BixolonLabelPrinter.ROTATION_NONE);
    mPrinter.print(1, 1);
  }
}
```

2-2-21 Draw Aztec

이미지 버퍼에 Aztec 바코드를 입력합니다.

[선언]

• void drawAztec(String data, int horizontalPosition, int verticalPosition,
int size, boolean extendedChannel, int eccLevel,
boolean menuSymbol, int numberOfSymbols,
String optionalID, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int size: 바코드의 크기 (범위: 1~10)

• boolean extendedChannel: 확장 채널 코드 사용여부

• int eccLevel: 에러 보정 레벨

Error control and symbol size/type	값
Default error correction level	0
Error correction percentage	1~99

• boolean menuSymbol: Menu symbol 사용 여부를 선택합니다. (true: 사용, false: 미사용)

• int numberOfSymbols: Number of symbols for structured append: (1 ~ 26)

• String optionalID: Optional ID field for structured append: ID field string
(Maximum 24 character)

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void drawBarcode (){
      mPrinter.drawAztec("Bixolon Aztec Test"
                         , 50
                         , 100
                         , 4
                         , false
                         , BixolonLabelPrinter.ECC_LEVEL_15
                         , false
                         , 10
                         , null
                         , BixolonLabelPrinter.ROTATION_NONE);
      mPrinter.print(1, 1);
   }
}
```

2-2-22 Draw Code 49

이미지 버퍼에 Code49 바코드를 입력합니다.

[선언]

• void drawCode49(String data, int horizontalPosition, int verticalPosition,
int widthNarrow, int widthWide, int height, int hri,
int startingMode, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int widthNarrow: 좁은 바의 너비

• int widthWide: 넓은 바의 너비

• int height: 바코드의 높이

• int hri: 바코드의 HRI 출력 위치

상수명	값	설명
CODE49_HRI_NOT_PRINTED	0	HRI 인쇄하지 않음
CODE49_HRI_BELOW_BARCODE	1	바코드 아래에 인쇄
CODE49_HRI_ABOVE_BARCODE	2	바코드 위에 인쇄

• int startingMode: Starting Mode

상수명	값	설명
CODE49_STRING_MODE_REGULAR_ALPHANUMERIC	0	Regular Alphanumeric Mode
CODE49_STRING_MODE_MULTIPLE_READ_ALPHANUMERIC	1	Multiple Read Alphanumeric
CODE49_STRING_MODE_REGULAR_NUMERIC	2	Regular Numeric Mode
CODE49_STRING_MODE_GROUP_ALPHANUMERIC	3	Group Alphanumeric Mode
CODE49_STRING_MODE_REGULAR_ALPHANUMERIC_SHIFT1	4	Regular Alphanumeric Shift 1
CODE49_STRING_MODE_REGULAR_ALPHANUMERIC_SHIFT2	5	Regular Alphanumeric Shift 2
CODE49_STRING_MODE_AUTOMATIC_MODE	7	Automatic Mode

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawBarcode (){
    mPrinter.drawCode49("Bixolon Code49 Test"
                        , 100
                        , 50
                        , 2
                        , 4
                        , 100
                        , BixolonLabelPrinter.HRI_BELOW_FONT_SIZE_1
                        , BixolonLabelPrinter. CODE49_STRING_MODE_AUTOMATIC_MODE
                        , BixolonLabelPrinter.ROTATION_NONE);
    mPrinter.print(1, 1);
  }
}
```

2-2-23 Draw Coda Block

이미지 버퍼에 CodaBlock 바코드를 입력합니다.

[선언]

• void drawCodaBlock(String data, int horizontalPosition, int verticalPosition,
int widthNarrow, int widthWide, int height,
boolean securityLevel, Int dataColumns, char mode,
int encode)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int widthNarrow: 좁은 바의 너비

• int widthWide: 넓은 바의 너비

• int height: 바코드의 높이

• boolean securityLevel: securityLevel 사용 여부 (true: 사용, false: 미사용)

• int dataColumns: 한 줄당 문자 수 (범위: 2~62)

• char mode: 바코드 출력 모드

상수명	값	설명
CODABLOCK_MODE_A	'A'	Code 39의 Character set을 사용합니다.
CODABLOCK_MODE_E	'E'	Code 128의 Character set을 사용합니다.
CODABLOCK_MODE_F	'F'	Code 128의 Character set을 사용합니다. Function 1이 자동으로 추가됩니다.

• int encode: 인코딩을 수행할 행 수

모드	값
CODABLOCK_MODE_A	1 ~ 18
CODABLOCK_MODE_E	2 ~ 4
CODABLOCK_MODE_F	2 ~ 4

이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawBarcode (){
    mPrinter.drawCodaBlock("Bixolon CodaBlock Test"
                           , 100
                           , 100
                           , 2
                           , 4
                           , 100
                           , false
                           , 10
                           , BixolonLabelPrinter.CODABLOCK_MODE_A
                           , 10);
    mPrinter.print(1, 1);
  }
}
```

2-2-24 Draw Micro PDF 417

이미지 버퍼에 Micro PDF417 바코드를 입력합니다.

[선언]

• void drawMicroPDF417(String data, int horizontalPosition, int verticalPosition,
int moduleWidth, int height, int mode, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int moduleWidth: module 너비 (범위: 2~8)

• int height: Barcode의 높이 (범위: 1~99)

• int mode: MicroPDF417 Barcode의 Mode

Mode	Number of Data Columns	Number of Data Rows	% of Cws for EC	Max Alpha Characters	Max Digits
0	1	11	64	6	8
1	1	14	50	12	17
2	1	17	41	18	26
3	1	20	40	22	32
4	1	24	33	30	44
5	1	28	29	38	55
6	2	8	50	14	20
7	2	11	41	24	35
8	2	14	32	36	52
9	2	17	29	46	67
10	2	20	28	56	82
11	2	23	28	64	93
12	2	26	29	72	105
13	3	6	67	10	14
14	3	8	58	18	26
15	3	10	53	26	38
16	3	12	50	34	49
17	3	15	47	46	67
18	3	20	43	66	96
19	3	26	41	90	132
20	3	32	40	114	167
21	3	38	39	138	202

22	3	44	38	162	237
23	4	6	50	22	32
24	4	8	44	34	49
25	4	10	40	46	67
26	4	12	38	58	85
27	4	15	35	76	111
28	4	20	33	106	155
29	4	26	31	142	208
30	4	32	30	178	261
31	4	38	29	214	313
32	4	44	28	250	366
33	4	4	50	14	20

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawBarcode (){
    mPrinter.drawMicroPDF417("Bixolon MicroPDF417 Test"
                             , 100
                             , 100
                             , 4
                             , 100
                             , 0
                             , BixolonLabelPrinter.ROTATION_NONE);
    mPrinter.print(1, 1);
  }
}
```

2-2-25 Draw IMB Barcode

이미지 버퍼에 IMB 바코드 출력을 요청합니다.

[선언]

• void drawIMBBarcode(String data, int horizontalPosition, int verticalPosition,
boolean hri, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• boolean hri: HRI 출력 여부 (true: 출력 false: 미출력)

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawBarcode (){
    mPrinter.drawIMBBarcode("Bixolon Inteligent Mail Barcode Test"
                            , 100
                            , 100
                            , true
                            , BixolonLabelPrinter.ROTATION_NONE);
    mPrinter.print(1, 1);
  }
}
```

2-2-26 Draw MSI Barcode

이미지 버퍼에 MSI 바코드를 입력합니다.

[선언]

• void drawMSIBarcode(String data, int horizontalPosition, int verticalPosition,
int widthNarrow, int widthWide, int height,
int checkDigit, boolean printCheckDigit, int hri, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int widthNarrow: 좁은 바의 너비

• int widthWide: 넓은 바의 너비

• int height: 바코드의 높이

• int checkDigit: checkDigit Option 선택

상수명	값	설명
MSI_BARCODE_CHECKDIGIT_NONE	0	Check Digit 없음
MSI_BARCODE_CHECKDIGIT_1MOD10	1	Check Digit 1 Mod 10
MSI_BARCODE_CHECKDIGIT_2MOD10	2	Check Digit 2 Mod 10
MSI_BARCODE_CHECKDIGIT_1MOD11_AND_1MOD_10	3	Check Digit 1 Mod 10

• int printCheckDigit: HRI에 Check Digit를 포함 여부 (true: 포함 false: 미포함)

• int hri: 바코드의 Data 값의 출력 위치

상수명	값	설명
HRI_NOT_PRINTED	0	HRI를 사용하지 않음
HRI_BELOW_FONT_SIZE_1	1	HRI 위치: 바코드의 하단 Font Size: 1
HRI_ABOVE_FONT_SIZE_1	2	HRI 위치: 바코드의 상단 Font Size: 1
HRI_BELOW_FONT_SIZE_2	3	HRI 위치: 바코드의 하단 Font Size: 2
HRI_ABOVE_FONT_SIZE_2	4	HRI 위치: 바코드의 상단 Font Size: 2
HRI_BELOW_FONT_SIZE_3	5	HRI 위치: 바코드의 하단 Font Size: 3
HRI_ABOVE_FONT_SIZE_3	6	HRI 위치: 바코드의 상단 Font Size: 3
HRI_BELOW_FONT_SIZE_4	7	HRI 위치: 바코드의 하단 Font Size: 4
HRI_ABOVE_FONT_SIZE_5	8	HRI 위치: 바코드의 상단 Font Size: 4

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawBarcode (){
    mPrinter.drawMSIBarcode("943457842"
                            , 100
                            , 100
                            , 2
                            , 4
                            , 100
                            , BixolonLabelPrinter.MSI_BARCODE_CHECKDIGIT_1MOD10
                            , true
                            , BixolonLabelPrinter.HRI_BELOW_FONT_SIZE_1
                            , BixolonLabelPrinter.ROTATION_NONE);
    mPrinter.print(1, 1);
  }
}
```

2-2-27 Draw Plessey Barcode

이미지 버퍼에 Plessey 바코드를 입력합니다.

[선언]

• void drawPlesseyBarcode(String data, int horizontalPosition, int verticalPosition,
int widthNarrow, int widthWide, int height,
boolean printCheckDigit, int hri, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int widthNarrow: 좁은 바의 너비

• int widthWide: 넓은 바의 너비

• int height: 바코드의 높이

• printCheckDigit: HRI에 Check Digit를 포함 여부 (true: 포함 false: 미포함)

• int hri: 바코드의 Data 값의 출력 위치

상수명	값	설명
PLESSEY_BARCODE_HRI_NOT_PRINTED	0	HRI 인쇄하지 않음
PLESSEY_BARCODE_HRI_BELOW_BARCODE	1	바코드 아래에 인쇄
PLESSEY_BARCODE_HRI_ABOVE_BARCODE	2	바코드 위에 인쇄

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void drawBarcode (){
      mPrinter.drawPlesseyBarcode("12345678"
                                  , 100, 100, 2
                                  , 4
                                  , 100
                                  , true
                                  , BixolonLabelPrinter.HRI_BELOW_FONT_SIZE_1
                                  , BixolonLabelPrinter.ROTATION_NONE);
      mPrinter.print(1, 1);
   }
}
```

2-2-28 Draw TLC39 Barcode

이미지 버퍼에 TLC39 바코드를 입력합니다.

[선언]

• void drawTLC39Barcode(String data, int horizontalPosition, int verticalPosition,
int widthNarrow, int widthWide, int height,
int rowHeightOfMicroPDF417,
int narrowWidthOfMicroPDF417, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int widthNarrow: Code39 좁은 바의 너비

• int widthWide: Code39 넓은 바의 너비

• int height: Code39 바코드의 높이

• int rowHeightOfMicroPDF417: microPDF417 행 높이

• int narrowWidthOfMicroPDF417: microPDF417 좁은 바의 폭

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void drawBarcode (){
      mPrinter.drawTLC39Barcode("123456,ABCD12345678901234"
                                    , 100
                                    , 100
                                    , 2
                                    , 4
                                    , 100
                                    , 3
                                    , 2
                                    , BixolonLabelPrinter.ROTATION_NONE);
      mPrinter.print(1, 1);
   }
}
```

2-2-29 Draw RSS Barcode

이미지 버퍼에 RSS 바코드를 입력합니다.

[선언]

• void drawRSSBarcode(String data, int horizontalPosition, int verticalPosition,
int barcodeType, int magnification, int separator,
int BarHeight, int SegmentWidth, int rotation)

[매개변수]

• String data: 출력을 원하는 바코드 값

• int horizontalPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int barcodeType: RSS 바코드 종류

상수명	값	설명
BARCODE_TYPE_RSS14	0	RSS14
BARCODE_TYPE_RSS14_TRUNCATED	1	RSS14 truncated
BARCODE_TYPE_RSS14_STACKED	2	RSS14 stacked
BARCODE_TYPE_RSS14_STACKED_OMNIDIRECTIONAL	3	RSS14 Stacked omnidirectional
BARCODE_TYPE_RSS_LIMITIED	4	RSS limited
BARCODE_TYPE_RSS_EXPANDED	5	RSS Expanded
BARCODE_TYPE_RSS_UPCA	6	RSS UPC A
BARCODE_TYPE_RSS_UPCE	7	RSS UPC E
BARCODE_TYPE_RSS_EAN13	8	EAN13
BARCODE_TYPE_RSS_EAN8	9	EAN 8
BARCODE_TYPE_RSS_UCC_EAN128_CCAB	10	EAN128 CC-A/B
BARCODE_TYPE_RSS_UCC_EAN128_CCC	11	EAN128 CC-C

• int magnification: 확대비율 (범위: 1~10)

• int separator: separator 높이 (범위: 1~2)

• int barHeight: 바코드의 높이

• int segmentWidth: segmentWidth (범위: 0~22)

• int rotation: 바코드의 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전

이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void drawBarcode (){
      mPrinter .drawRSSBarcode("12345678901|this is composite info"
                                    , 100
                                    , 100
                                    , BixolonLabelPrinter.BARCODE_TYPE_RSS14
                                    , 2
                                    , 1
                                    , 20
                                    , 10
                                    , BixolonLabelPrinter.ROTATION_NONE);
      mPrinter.print(1, 1);
   }
}
```

2-2-30 Draw Block

이미지 버퍼에 라인, 블록, 박스 및 사선을 입력합니다.

[선언]

• void drawBlock(int horizontalStartPosition, int verticalStartPosition,
int horizontalEndPosition, int verticalEndPosition, int option,
int thickness)

[매개변수]

• int horizontalStartPosition: 수평 start 포지션(X)

(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalStartPosition: 수직 start 포지션(Y)

(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int horizontalEndPosition: 수평 end 포지션(X)

(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalEndPosition: 수직 end 포지션(Y)

(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int option: 블록 출력 옵션

• int thickness: 두께

이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawBox (){
    mPrinter.drawBlock(0
                       , 0
                       , 100
                       , 100
                       , BixolonLabelPrinter.BLOCK_OPTION_BOX
                       , 3);
    mPrinter.print(1,1);
  }
}
```

2-2-31 Draw Two Block

이미지 버퍼에 2개의 Block을 입력합니다.

[선언]

• void drawTowBlock(int horizontalStartPosition, int verticalStartPosition,
int horizontalEndPosition, int verticalEndPosition,
int option, int horizontalStartPositionSquare2,
int verticalStartPositionSquare2,
int horizontalEndPositionSquare2,
int verticalEndPositionSquare2, int optionSquare2)

[매개변수]

• int horizontalStartPosition: 첫 번째 블록의 수평 start 포지션(X)
(참조:1-4 각 모델 별 좌표의 사용 범위)

• int verticalStartPosition: 첫 번째 블록의 수직 start 포지션(Y)
(참조:1-4 각 모델 별 좌표의 사용 범위)

• int horizontalEndPosition: 첫 번째 블록의 수평 end 포지션(X)
(참조:1-4 각 모델 별 좌표의 사용 범위)

• int verticalEndPosition: 첫 번째 블록의 수직 end 포지션(Y)
(참조:1-4 각 모델 별 좌표의 사용 범위)

• int option: 첫 번째 블록의 블록 출력 옵션

• int horizontalStartPositionSquare2: 두 번째 블록의 수평 start 포지션(X)
(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalStartPositionSquare2: 두 번째 블록의 수직 start 포지션(Y)
(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int horizontalEndPositionSquare2: 두 번째 블록의 수평 end 포지션(X)
(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalEndPositionSquare2: 두 번째 블록의 수직 end 포지션(Y)
(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int optionSquare2: 두 번째 블록의 블록 출력 옵션

이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void drawBox (){
      mPrinter.drawTowBlock(100
                           , 100
                           , 300
                           , 300
                           , BixolonLabelPrinter.BLOCK_OPTION_LINE_EXCLUSIVE_OR
                           , 400
                           , 400
                           , 500
                           , 500
                           , BixolonLabelPrinter.BLOCK_OPTION_LINE_EXCLUSIVE_OR);
      mPrinter.print(1,1);
   }
}
```

2-2-32 Draw Circle

이미지 버퍼에 원을 입력합니다.

[선언]

• drawCircle(int horizontalStartPosition, int verticalStartPosition, int size,
int multiplier)

[매개변수]

• int horizontalStartPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalStartPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int size: 출력 옵션

상수명	값	설명	
CIRCLE_SIZE_DIAMETER5	1	40 × 40 (dot)	5 mm
CIRCLE_SIZE_DIAMETER7	2	56 × 56 (dot)	7 mm
CIRCLE_SIZE_DIAMETER9	3	72 × 72 (dot)	9 mm
CIRCLE_SIZE_DIAMETER11	4	88 × 88 (dot)	11 mm
CIRCLE_SIZE_DIAMETER13	5	104 × 104 (dot)	13 mm
CIRCLE_SIZE_DIAMETER21	6	168 × 168 (dot)	21 mm

• int multiplier: 확대 (범위: 1~4)

이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawCircle (){
    mPrinter.drawCircle(50
                        , 50
                        , BixolonLabelPrinter.CIRCLE_SIZE_DIAMETER5
                        , 1);
    mPrinter.print(1,1);
  }
}
```

2-2-33 Draw Bitmap

이미지 버퍼에 이미지를 입력합니다.

[선언]

• drawBitmap(String pathname, int horizontalStartPosition, int verticalStartPosition,
int width, int level, boolean dithering)

[매개변수]

• String pathname: 이미지 경로

• int horizontalStartPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalStartPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int width: 출력할 이미지의 너비 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int level: 출력할 이미지의 밝기

• boolean dithering: 디더링 적용 여부 (true: 적용, false: 미적용)



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  …
  private void drawImage (){
  String path = “/…”;
    mPrinter.drawBitmap(path
                        , 100
                        , 100
                        , 100
                        , 20
                        , true);
    mPrinter.print(1,1);
  }
}
```

2-2-34 Draw Bitmap

이미지 버퍼에 이미지를 입력합니다.

[선언]

• drawBitmap(Bitmap bitmap, int horizontalStartPosition, int verticalStartPosition,
int width, int level, boolean dithering)

[매개변수]

• Bitmap bitmap: 출력할 이미지 객체

• int horizontalStartPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalStartPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int width: 출력할 이미지의 너비 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int level: 출력할 이미지의 밝기

• boolean dithering: 디더링 적용 여부 (true: 적용, false: 미적용)

이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawImage (){
  Bitmap bitmap = ...;
    mPrinter.drawBitmap(bitmap
                        , 100
                        , 100
                        , 100
                        , 20
                        , true);
    mPrinter.print(1,1);
  }
}
```

2-2-35 Draw Compression Image

이미지 버퍼에 이미지 출력을 요청합니다. (이미지 압축 알고리즘 적용)

[선언]

• drawCompressionImage(Bitmap bitmap, int horizontalStartPosition,
int verticalStartPosition, int width, int level,
boolean dithering)

[매개변수]

• Bitmap bitmap: Bitmap 객체

• int horizontalStartPosition: 시작위치의 X 좌표

(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalStartPosition: 시작위치의 Y 좌표

(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int width: 출력할 너비

(참조: 1-4 각 모델 별 좌표의 사용 범위)

• int level: 출력할 이미지의 밝기

• boolean dithering: 디더링 적용 여부 (true: 적용, false: 미적용)



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawCompressImage (){
    mPrinter.drawCompressionImage(bitmap
                                  , 0
                                  , 10
                                  , 200
                                  , 200
                                  , 80
                                  , true)
    mPrinter.print(1,1);
  }
}
```

2-2-36 Draw Compression Image

이미지 버퍼에 이미지 출력을 요청합니다. (이미지 압축 알고리즘 적용)

[선언]

• drawCompressionImage(String path, int horizontalStartPosition,
int verticalStartPosition, int width, int level,
boolean dithering)

[매개변수]

• String path: 이미지 경로

• int horizontalStartPosition: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int verticalStartPosition: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int width: 출력할 너비 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int level: 출력할 이미지의 밝기

• boolean dithering: 디더링 적용 여부 (true: 적용, false: 미적용)



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void drawCompressImage (){
    mPrinter.drawCompressionImage(path
                                  , 0
                                  , 10
                                  , 200
                                  , 200
                                  , 80
                                  , true)
    mPrinter.print(1,1);
  }
}
```

2-2-37 Draw Image

라벨 프린터 버퍼에 이미지를 저장합니다.

[선언]

• drawImage(Bitmap imageData, int startPosX, int startPosY, int width, int threshold, int ditheringType, int compressType)

[매개변수]

• Bitmap imageData: 인쇄하려는 이미지 데이터

• int startPosX: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int startPosY: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int width: 출력할 너비 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int threshold: 출력할 이미지의 threshold 값 (0~255 범위를 벗어나면 128로 처리)

• int ditheringType: 이미지 인쇄 시 디더링 타입

• int compressType: 사용할 압축 알고리즘

상수명	값	설명
LABEL_IMAGE_NONE	0	기본값
LABEL_IMAGE_RLE	1	RLE
LABEL_IMAGE_LZMA	2	LZMA

이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출.

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   private void drawImage (){
      mPrinter.drawImage(imageData
                              , 0
                              , 10
                              , 200
                              , 200
                              , 80
                              , 0
                              , 0);
      mPrinter.print(1,1);
   }
}
```

2-2-38 Draw Image

라벨 프린터 버퍼에 이미지를 저장합니다.

[선언]

• drawImage(Bitmap imageData, int startPosX, int startPosY, int width,
int threshold, int ditheringType, int compressType, int rotation)

[매개변수]

• Bitmap imageData: 인쇄하려는 이미지 데이터

• int startPosX: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int startPosY: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int width: 출력할 너비 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int threshold: 출력할 이미지의 threshold 값 (0~255 범위를 벗어나면 128로 처리)

• int ditheringType: 이미지 인쇄 시 디더링 타입

• int compressType: 사용할 압축 알고리즘

상수명	값	설명
LABEL_IMAGE_NONE	0	기본값
LABEL_IMAGE_RLE	1	RLE
LABEL_IMAGE_LZMA	2	LZMA

• int rotation: 이미지 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출.

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void drawImage (){
      mPrinter.drawImage(imageData
                         , 0
                         , 10
                         , 200
                         , 200
                         , 80
                         , 0
                         , 0
                         , BixolonLabelPrinter.ROTATION_NONE);
      mPrinter.print(1,1);
   }
}
```

2-2-39 Draw Base64 Image

이미지 버퍼에 Base64 이미지 출력을 요청합니다.

[선언]

• drawBase64Image(String base64Img, boolean isTransparent,
int stratPosX, int startPosY, int width, int threshold,
int ditheringType, int compressType, int rotation)

[매개변수]

• String base64Img: base64 이미지 데이터

• Boolean isTransparent: 이미지 transparent 적용 여부

• int startPosX: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int startPosY: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int width: 출력할 너비 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int threshold: 출력할 이미지의 threshold 값 (0~255 범위를 벗어나면 128로 처리)

• int ditheringType: 이미지 인쇄 시 디더링 타입

• int compressType: 사용할 압축 알고리즘

상수명	값	설명
LABEL_IMAGE_NONE	0	기본값
LABEL_IMAGE_RLE	1	RLE
LABEL_IMAGE_LZMA	2	LZMA

• int rotation: 이미지 회전 방향

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   private void drawBase64Image (){
      mPrinter.drawBase64Image(imageData
                              , false
                              , 0
                              , 10
                              , 200
                              , 200
                              , 80
                              , 0
                              , 0
                              , BixolonLabelPrinter.ROTATION_NONE);
      mPrinter.print(1,1);
}
```

2-2-40 Draw Image File

라벨 프린터 버퍼에 이미지 파일을 저장합니다.

[선언]

• drawImageFile(String filename, int startPosX, int startPosY, int width, int threshold, int ditheringType, int compressType)

[매개변수]

• String path: 인쇄하려는 이미지 파일 경로

• int startPosX: 시작위치의 X 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int startPosY: 시작위치의 Y 좌표 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int width: 출력할 너비 (참조: 1-4 각 모델 별 좌표의 사용 범위)

• int threshold: 출력할 이미지의 threshold 값

• int ditheringType: 디더링 적용 여부

• int compressType: 사용할 압축 알고리즘

상수명	값	설명
LABEL_IMAGE_NONE	0	기본값
LABEL_IMAGE_RLE	1	RLE
LABEL_IMAGE_LZMA	2	LZMA



이미지 버퍼의 내용을 인쇄하려면 **2-2-11 print API** 호출.

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   private void drawImageFile (){
      mPrinter.drawImageFile(filename
                              , 0
                              , 10
                              , 200
                              , 200
                              , 80
                              , 0
                              , 0)
      mPrinter.print(1,1);
   }
}
```

2-2-41 Get Status

프린터의 상태를 체크합니다.

[선언]

• void getStatus(boolean checkImageBuffer)

[매개변수]

• boolean checkImageBuffer: **결과 값 2바이트 여부** (true: 2byte, false: 1byte)

[결과 값]

상수명	값	설명
STATUS_NORMAL	0x00	이상 없음
STATUS_1ST_BYTE_PAPER_EMPTY	0x80	용지 없음
STATUS_1ST_BYTE_COVER_OPEN	0x40	용지 커버 열림
STATUS_1ST_BYTE_CUTTER_JAMMED	0x20	Cutter 걸림
STATUS_1ST_BYTE_TPH_OVERHEAT	0x10	헤더 과열
STATUS_1ST_BYTE_AUTO_SENSING_FAILURE	0x08	갭 검출 실패
STATUS_1ST_BYTE_RIBBON_END_ERROR	0x04	리본 없음
STATUS_2ND_BYTE_BUILDING_IN_IMAGE_BUFFER	0x00000080	이미지 버퍼에 라벨 구성 중
STATUS_2ND_BYTE_PRINTING_IN_IMAGE_BUFFER	0x00000040	이미지 버퍼에 있는 라벨 인쇄 중
STATUS_2ND_BYTE_PAUSED_IN_PEELER_UNIT	0x00000020	인쇄된 라벨이 peeler에서 붙어있음

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...

  private void getStatus(){
    mPrinter.getStatus(true);
  }
  private final Handler mHandler = new Handler(){
    @Override
    public void handleMessage(Message msg){
      switch (msg.what){
      ...
      case BixolonLabelPrinter.MESSAGE_READ :
      switch (msg.arg1){
        case BixolonLabelPrinter.PROCESS_GET_STATUS:
        byte[] report = (byte[]) msg.obj;

        //1Byte Message
        if((report[0] & BixolonLabelPrinter.STATUS_1ST_BYTE_PAPER_EMPTY)
                == BixolonLabelPrinter.STATUS_1ST_BYTE_PAPER_EMPTY){
            Log.i("TAG", "Paper is Empty");
        }
          if((report[0] & BixolonLabelPrinter.STATUS_1ST_BYTE_COVER_OPEN)
                  == BixolonLabelPrinter.STATUS_1ST_BYTE_COVER_OPEN){
            Log.i("Printer cover open");
        }

        ...
        //2Byte Message
         if(report.length == 2){
          if((report[1] &
             BixolonLabelPrinter.STATUS_2ND_BYTE_PAUSED_IN_PEELER_UNIT)==
                BixolonLabelPrinter.STATUS_2ND_BYTE_PAUSED_IN_PEELER_UNIT){
             Log.i("TAG", " Issued label is paused in peeler unit");
          }
          ...
         }
         break;
        }
      break;
      }
    }
  }
}
```

2-2-42 Set Auto Cutter

Auto Cutter가 장착된 모델의 AutoCutter 설정을 변경합니다.

[선언]

• void setAutoCutter(boolean enabled, int cuttingPeriod)

[매개변수]

• boolean enabled: AutoCutter의 사용 여부 (true: 사용 false: 미사용)

• int cuttingPeriod: 커팅 간격 설정

커팅 간격을 2로 설정한 후 2-2-11 print 파라미터를 3으로 입력하여 3장 반복 인쇄 시 2장 인쇄 후 커팅, 마지막 1매 인쇄 후 커팅 합니다.

- 오토 커터가 장착된 경우에만 사용 가능 합니다.
- 오토 커터가 장착되지 않은 경우 enable을 항상 NO로 설정하십시오.

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void cutterSetting(){
      mPrinter.setAutoCutter(true, 10);
   }
}
```

2-2-43 Get Printer Information

모델명, 펌웨어 버전과 같은 다양한 프린터의 정보를 요청합니다.

[선언]

• void getPrinterInformation(int param)

[매개변수]

• int param: 프린터에 요청할 정보

상수명	값	설명
PRINTER_INFORMATION_MODEL_NAME	0	연결된 프린터의 모델명
PRINTER_INFORMATION_FIRMWARE_VERSION	2	연결된 프린터의 펌웨어 버전

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void getPrinterInformation(){
      mPrinter.getPrinterInformation(
                  BixolonLabelPrinter.PRINTER_INFORMATION_MODEL_NAME);
   }
}
```

2-2-44 Print Information

프린터의 정보를 출력합니다.

[선언]

• void printInformation()

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void getPrinterInformation(){
      mPrinter.printInformation();
   }
}
```

2-2-45 Initialize Printer

프린터 설정을 초기화합니다.

[선언]

• void initializePrinter();

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.initializePrinter();
   }
}
```

2-2-46 Set Orientation

출력 시 방향을 설정합니다.

[선언]

• void setOrientation(int orientation);

[매개변수]

• int orientation: 출력 방향

상수명	값	설명
ORIENTATION_TOP_TO_BOTTOM	84	위에서 아래로 출력
ORIENTATION_BOTTOM_TO_TOP	66	아래에서 위로 출력

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setOrientation(BixolonLabelPrinter.ORIENTATION_BOTTOM_TO_TOP);
   }
}
```

2-2-47 Set Character Set

internationalCharacterSet 및 코드페이지를 설정합니다.

[선언]

• void setCharacterSet(int internationalCharacterSet, int codePage);

[매개변수]

• int internationalCharacterSet: 문자셋

상수명	값	설명
INTERNATIONAL_CHARACTER_SET_USA	0	U.S.A
INTERNATIONAL_CHARACTER_SET_FRANCE	1	France
INTERNATIONAL_CHARACTER_SET_GERMANY	2	Germany
INTERNATIONAL_CHARACTER_SET_UK	3	U.K
INTERNATIONAL_CHARACTER_SET_DENMARK1	4	Denmark I
INTERNATIONAL_CHARACTER_SET_SWEDEN	5	Sweden
INTERNATIONAL_CHARACTER_SET_ITALY	6	Italy
INTERNATIONAL_CHARACTER_SET_SPAIN1	7	Spain I
INTERNATIONAL_CHARACTER_SET_NORWAY	8	Norway
INTERNATIONAL_CHARACTER_SET_DENMARK2	9	Denmark II
INTERNATIONAL_CHARACTER_SET_JAPAN	10	Japan
INTERNATIONAL_CHARACTER_SET_SPAIN2	11	Spain II
INTERNATIONAL_CHARACTER_SET_LATIN_AMERICA	12	Latin America
INTERNATIONAL_CHARACTER_SET_KOREA	13	Korea
INTERNATIONAL_CHARACTER_SET_SLOVENIA_CROATIA	14	Slovenia/Croatia
INTERNATIONAL_CHARACTER_SET_CHINA	15	China

• int codePage: 코드페이지

상수명	값	설명	
CODE_PAGE_CP437_USA	0	CP437	U.S.A
CODE_PAGE_CP850_LATIN1	1	CP850	Latin1
CODE_PAGE_CP852_LATIN2	2	CP 852	Latin2
CODE_PAGE_CP860_PORTUGUESE	3	CP 860	Portuguese
CODE_PAGE_CP863_CANADIAN_FRENCH	4	CP 863	Canadian French
CODE_PAGE_CP865_NORDIC	5	CP 865	Nordic
CODE_PAGE_WCP1252_LATIN1	6	WCP 1252	Latin I
CODE_PAGE_CP865_WCP1252_EUROPEAN_COMBINED	7	CP 865 + WCP 1252	European Combined
CODE_PAGE_CP857_TURKISH	8	CP 857	Turkish
CODE_PAGE_CP737_GREEK	9	CP 737	Greek
CODE_PAGE_WCP1250_LATIN2	10	WCP 1250	Latin 2
CODE_PAGE_WCP1253_GREEK	11	WCP 1253	Greek
CODE_PAGE_WCP1254_TURKISH	12	WCP 1254	Turkish
CODE_PAGE_CP855_CYRILLIC	13	CP 855	Cyrillic
CODE_PAGE_CP862_HEBREW	14	CP 862	Hebrew
CODE_PAGE_CP866_CYRILLIC	15	CP 866	Cyrillic
CODE_PAGE_WCP1251_CYRILLIC	16	WCP 1251	Cyrillic
CODE_PAGE_WCP1255_HEBREW	17	WCP 1255	Hebrew
CODE_PAGE_CP928_GREEK	18	CP 928	Greek
CODE_PAGE_CP864_ARABIC	19	CP 864	Arabic
CODE_PAGE_CP775_BALTIC	20	CP 775	Baltic
CODE_PAGE_WCP1257_BALTIC	21	WCP1257	Baltic
CODE_PAGE_CP858_LATIN1_EURO	22	CP858	Latin 1 + Euro

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void printerSetting (){
    mPrinter.setCharacterSet(
                    BixolonLabelPrinter.INTERNATIONAL_CHARACTER_SET_USA
                   , BixolonLabelPrinter.CODE_PAGE_CP437_USA);
  }
}
```

2-2-48 Set Printing Type

프린터의 출력 타입을 설정합니다.

[선언]

• void setPrintingType(int type)

[매개변수]

• int type: 출력 타입

상수명	값	설명
PRINTING_TYPE_DIRECT_THERMAL	100	용지 직접 감열
PRINTING_TYPE_THERMAL_TRANSFER	116	리본 감열

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setPrintingType (
            BixolonLabelPrinter.PRINTING_TYPE_DIRECT_THERMAL);
   }
}
```

2-2-49 Set Margin

이미지 버퍼의 여백을 설정합니다.

[선언]

• void setMargin(int horizontalMargin, int verticalMargin);

[매개변수]

• int horizontalMargin: 가로 여백

• int verticalMargin: 세로 여백

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setMargin(10, 10);
   }
}
```

2-2-50 Set Back Feed Option

인쇄를 시작하기 전에 Back-Feed 수행 여부를 설정합니다.

[선언]

• void setBackFeedOption(boolean enabled, int quantity);

[매개변수]

• boolean enabled: Back Feed 활성화 여부 (true: 활성화, false: 비활성화)

• int quantity: Back-Feed 길이 (0: 기본값)

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setBackFeedOption(true, 10);
   }
}
```

2-2-51 Set Buffer Mode

이미지 버퍼의 모드를 설정합니다.

[선언]

• void setBufferMode(boolean doubleBuffering);

[매개변수]

• boolean doubleBuffering: 더블 버퍼 활성화 여부 (true: 활성화, false: 비활성화)

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setBufferMode (false);
   }
}
```

2-2-52 Clear Buffer

이미지 버퍼의 내용을 비우고 새로운 라벨을 만들 준비를 합니다.

[선언]

• void clearBuffer();

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.clearBuffer ();
   }
}
```

2-2-53 Set Length

프린터의 용지 길이와 갭/블랙마크의 길이, 용지 종류를 설정합니다.

[선언]

• void setLength(int labelLength, int gapLength, int mediaType, int offsetLength);

[매개변수]

• int labelLength: 용지 길이

• int gapLength: 갭의 길이나 블랙마크의 두께

• int mediaType: 용지 타입

상수명	값	설명
MEDIA_TYPE_GAP	71	갭 용지
MEDIA_TYPE_CONTINUOUS	67	연속 용지
MEDIA_TYPE_BLACK_MARK	66	블랙마크 용지

• int offsetLength: 갭 또는 블랙마크와 절취선과의 간격

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setLength (1216
                          , 1
                          , BixolonLabelPrinter.MEDIA_TYPE_GAP
                          , 1);
   }
}
```

2-2-54 Set Rewinder

리와인더의 사용 여부를 선택합니다.

[선언]

• void setRewinder(boolean enabled)

[매개변수]

• boolean enabled: 리와인더 사용 여부 (true: 사용, false: 미사용)



- 리와인더가 장착된 경우에만 사용 가능 합니다.
- 리와인더가 장착되지 않은 경우 enabled를 항상 false로 설정하십시오.

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setRewinder (false);
   }
}
```

2-2-55 Set Speed

프린터의 출력 속도를 설정합니다.

[선언]

• void setSpeed(int speed);

[매개변수]

• int speed: 출력 속도

상수명	값	설명
SPEED_25IPS	0	초당 2.5인치 출력
SPEED_30IPS	1	초당 3인치 출력
SPEED_40IPS	2	초당 4인치 출력
SPEED_50IPS	3	초당 5인치 출력
SPEED_60IPS	4	초당 6인치 출력
SPEED_70IPS	5	초당 7인치 출력
SPEED_80IPS	6	초당 8인치 출력

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setSpeed(BixolonLabelPrinter.SPEED_30IPS);
   }
}
```

2-2-56 Set Offset

갭 또는 블랙마크와 절취선과의 간격을 설정합니다.

[선언]

• void setOffset(int offset);

[매개변수]

• int offset: 간격 (범위: -100~100)

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setOffset (40);
   }
}
```

2-2-57 Set Density

인쇄 농도를 설정합니다.

[선언]

• void setDensity(int density)

[매개변수]

• int density: 인쇄 농도 (범위: 0~20)

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setDensity(10);
   }
}
```

2-2-58 Set Cutter Position

라벨의 커팅 위치를 설정합니다.

[선언]

• void setCutterPosition(int position)

[매개변수]

• int position: 커팅 위치 (범위: -100~100)

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setCutterPosition (0);
   }
}
```

2-2-59 Set Width

이미지 버퍼의 너비를 설정합니다.

[선언]

• void setWidth(int labelWidth)

[매개변수]

• int labelWidth: 이미지 버퍼 너비

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting (){
      mPrinter.setWidth (800);
   }
}
```

2-2-60 Execute Direct Io

프린터로 명령어를 직접 전송합니다.

명령어 생성을 위해서는 당사의 SLCS(Programming) 매뉴얼을 참조하십시오.

[선언]

• executeDirectIo(String command, boolean hasResponse, int responseLength)

[매개변수]

• String command: SLCS를 통해 생성한 명령어 데이터

• boolean hasResponse: 반환 값 유무 (true: 있음, false: 없음)

• int responseLength: 반환 값의 길이

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printDirectIo (){
      mPrinter.executeDirectIo("CB\n" +
                                    "SW800\n" +
                                    "SM10,0\n" +
                                    "BD100,300,300,500,O\n" +
                                    "BD400,300,700,500,B,30\n" +
                                    "P1", false, 0);
   }
}
```

2-2-61 Execute Direct Io

프린터로 명령어를 직접 전송합니다.

명령어 생성을 위해서는 당사의 SLCS(Programming) 매뉴얼을 참조하십시오.

[선언]

• executeDirectIo(String command, ResponseType responseType,
int responseLength)

[매개변수]

• String command: SLCS를 통해 생성한 명령어 데이터

• ResponseType responseType: 반환 값의 타입

타입명	설명
NONE	반환 값 없음
STATUS_CHECK	상태 체크 값
CRLF	반환 값 이후 CRLF
NULL	반환 값 이후 NULL
RAW_DATA	바이트 배열

• int responseLength: 반환 값의 길이

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printDirectIo (){
      mPrinter.executeDirectIo("CB\n" +
                                    "SW800\n" +
                                    "SM10,0\n" +
                                    "BD100,300,300,500,O\n" +
                                    "BD400,300,700,500,B,30\n" +
                                    "P1", ResponseType.NONE, 0);
   }
}
```

2-2-62 Execute Direct Io

프린터로 Byte 배열의 명령어를 직접 전송합니다.

명령어 생성을 위해서는 당사의 SLCS(Programming) 매뉴얼을 참조하십시오.

[선언]

• executeDirectIo(byte[] command, boolean hasResponse, int responseLength)

[매개변수]

• byte[] command: SLCS를 통해 생성한 명령어의 Byte 배열

• boolean hasResponse: 반환 값 유무 (true: 있음, false: 없음)

• int responseLength: 반환 값의 길이

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void printDirectIo (){
    String command ="SS3\n" + // Set Speed to 5 ips
                    "SD20\n" +      // Set Density level to 20
                    "SW800\n" +     // Set Label Width 800
                    "SOT\n" +       // Set Printing Orientation from Top to Bottom
                    "T26,20,0,1,1,0,0,N,N,'Font - 6 pt'\n" +
                    "T26,49,1,1,1,0,0,N,N,'Font - 8 pt'\n" +
                    "T26,81,2,1,1,0,0,N,N,'Font - 10 pt'\n" +
                    "T26,117,3,1,1,0,0,N,N,'Font - 12 pt'\n" +
                    "T26,156,4,1,1,0,0,R,N,'Font - 15 pt'\n" +
                    "T26,200,5,1,1,0,0,N,N,'Font - 20 pt'\n" +
                    "T26,252,6,1,1,0,0,N,N,'Font - 30 pt'\n" +
                    "P1";

    mPrinter.executeDirectIo(command.getBytes(), false, 0);
  }
}
```

2-2-63 Execute Direct Io

프린터로 Byte 배열의 명령어를 직접 전송합니다.

명령어 생성을 위해서는 당사의 SLCS(Programming) 매뉴얼을 참조하십시오.

[선언]

• executeDirectIo(byte[] command, ResponseType resonseType, int responseLength)

[매개변수]

• byte[] command: SLCS를 통해 생성한 명령어의 Byte 배열

• ResponseType responseType: 반환 값 타입

타입명	설명
NONE	반환 값 없음
STATUS_CHECK	상태 체크 값
CRLF	반환 값 이후 CRLF
NULL	반환 값 이후 NULL
RAW_DATA	반환 값 있음

• int responseLength: 반환 값의 길이

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  ...
  private void printDirectIo (){
    String command ="SS3\n" + // Set Speed to 5 ips
                    "SD20\n" +      // Set Density level to 20
                    "SW800\n" +     // Set Label Width 800
                    "SOT\n" +       // Set Printing Orientation from Top to Bottom
                    "T26,20,0,1,1,0,0,N,N,'Font - 6 pt'\n" +
                    "T26,49,1,1,1,0,0,N,N,'Font - 8 pt'\n" +
                    "T26,81,2,1,1,0,0,N,N,'Font - 10 pt'\n" +
                    "T26,117,3,1,1,0,0,N,N,'Font - 12 pt'\n" +
                    "T26,156,4,1,1,0,0,R,N,'Font - 15 pt'\n" +
                    "T26,200,5,1,1,0,0,N,N,'Font - 20 pt'\n" +
                    "T26,252,6,1,1,0,0,N,N,'Font - 30 pt'\n" +
                    "P1";

    mPrinter.executeDirectIo(command.getBytes(), ResponseType.NONE, 0);
  }
}
```

2-2-64 Setup RFID

RFID트랜스폰더 타입, 재시도 코딩 횟수, 재시도 라벨수량, 송수신 강도를 설정합니다.

[선언]

• setupRFID(int rfidType, int numberOfRetries, int numberOfLabel, int radioPower)

[매개변수]

• int rfidType: RFID트랜스폰더 Type (범위: 0~5)

• int numberOfRetries: 코딩 실패 시 재시도 횟수 설정 (범위: 1~10)

• int numberOfLabel: RFID라벨 Write 실패 시 재시도 라벨 수 (범위: 1~5)

• int radioPower: 송수신 강도 조절 (범위: 0~30)

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   private void setupRFID (){
      int rfidType = 5;
      int numberOfRetries = 2;
      int numberOfLabel = 2;
      int radioPower = 21;

      mPrinter. setupRFID (rfidType, numberOfRetries, numberOfLabel, radioPower);
   }
}
```

2-2-65 Set RFID Position

RFID 라벨에 코딩 위치를 직접 설정합니다.

[선언]

• setRFIDPosition(int transPosition)

[매개변수]

• int transPosition: RFID 라벨 코딩 위치 (y축 값)

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void setRFIDPosition (){
      int transposition = 320;

      mPrinter. setRFIDPosition (transPosition);
   }
}
```

2-2-66 Set EPC Data Structure

EPC Data를 쓰기 위한 EPC Data 구조를 정의합니다.

[선언]

• setEPCDataStructure(int totalSize, String fieldSize)

[매개변수]

• int totalSize: field의 총 비트 량

• String fieldSize: 각 field당 비트 량

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   private void setEPCDataStructure (){
      int totalSize = 64;
      String fieldSize = “8,8,8,8,8,8,8,8”;

      mPrinter.setEPCDataStructure (totalSize, fieldSize);
   }
}
```

2-2-67 Write RFID

RFID라벨을 Write 합니다.

[선언]

• writeRFID(int dataType, int startingBlockNumber, int dataLength, String data)

[매개변수]

• int dataType: RFID 데이터 타입 (A: Ascii, H: Hexadecimal, E: EPC, U: User영역)

• int startingBlockNumber: block number (범위: 4~10)

• int dataLength: byte 수 (범위: 2~12)

• String data: write 할 데이터



- RFID 데이터 타입을 EPC로 설정할 경우 setEPCDataStructure를 통해 EPC데이터 구조 설정 후 사용 가능
- 데이터 타입이 EPC인 경우 startingBlockNumber, dataLength 파라미터는 무시

[예제]

```
Public Class SampleClass{
  BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
  …
  private void writeRFID(){
    int dataType = ‘E’;
    int startingBlockNumber = 4;
    int dataLength = 8;
    String data = “1,2,3,4,5,6,7,8”;

    mPrinter.writeRFID (dataType, startingBlockNumber, dataLength, data);
  }
}
```

2-2-68 Set RFID Password

RFID Access Password와 Kill Password를 설정합니다.

[선언]

• setRFIDPassword(String oldAccessPwd, String oldKillPwd, String newAccessPwd, String newKillPwd)

[매개변수]

• String oldAccessPwd: 현재 사용 중인 Access Password

• String oldKillPwd: 현재 사용 중인 Kill Password

• String newAccessPwd: 수정할 Access Password

• String newKillPwd: 수정할 Kill Password

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   private void setRFIDPassword (){
      String oldAccessPwd = “0000”;
      String oldKillPwd = “0000”;
      String newAccessPwd = “1111”;
      String newKillPwd = “1111”;

      mPrinter.setRFIDPassword (oldAccessPwd, oldKillPwd,
                                    newAccessPwd, newKillPwd);
   }
}
```

2-2-69 Lock RFID

Access Password로 Kill, Access, EPC Data를 Lock 시킵니다.

[선언]

• lockRFID();

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   private void lockRFID (){

      mPrinter. lockRFID ();
   }
}
```

2-2-70 Transfer File

파일을 전송합니다.

[선언]

• transferFile(String filePath);

[매개변수]

• String filePath: file 경로

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   private void transferFile (String filePath){
      String filename = “exampleFileName”
      mPrinter. transferFile (filePath + “/” + fileName);
   }
}
```

2-2-71 Transfer File

파일을 전송합니다.

[선언]

• transferFile(String filePath, TransferFileCallback callback);

[매개변수]

• String filePath: file 경로

• TransferFileCallback callback: 파일 전송 진행률과 결과를 확인하기 위한 Callback

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   BixolonLabelPrinter.TransferFileCallback callback = new
BixolonLabelPrinter.TransferFileCallback() { ... }
   ...
   private void transferFile (String filePath){
      String filename = “exampleFileName”
      mPrinter. transferFile (filePath + “/” + filename, callback);
   }
}
```

2-2-72 Draw PDF File

PDF를 인쇄합니다.

[선언]

• drawPDFFile(Uri uri, int horizontalStartPosition, int verticalStartPosition, int page, int width, int level, boolean dithering, boolean compress);

[매개변수]

• Uri uri: pdf file uri

• int horizontalStartPosition: 수평 시작 위치

• int verticalStartPosition: 수직 시작 위치

• int page: 인쇄할 PDF page

• int width: 인쇄 너비

• int level: 이미지 스레숄드 값

• boolean dithering: 디더링 여부

• boolean compress: 압축 여부

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   @Override
   Public void onActivityResult(int requestCode, int resultCode, Intent data) {
       //…

       Uri uri = data.getData();
       int page = 1;
       int width = 800;
       int level = 100;
       boolean dithering = false;
       boolean compress = false;
       mPrinter.drawPDFFile(
         uri, 0, 0, page, width, level, dithering, compress);
       mPrinter.print(1, 1);
   }
}
```

2-2-73 Set PDF Dpi

PDF dpi를 설정합니다.

[선언]

• setPdfDpi(int dpi);

[매개변수]

• int dpi: dpi value

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   mPrinter.setPdfDpi(300);

}
```

2-2-74 Get PDF Page

PDF의 특정 페이지를 얻어옵니다.

[선언]

• getPdfPage(Uri uri, int pageNum);

[매개변수]

• Uri uri: pdf file uri

• int pageNum: 불러올 PDF page 번호

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   @Override
   Public void onActivityResult(int requestCode, int resultCode, Intent data) {
       //…

       Uri uri = data.getData();
       int pageNum = 1;
       Bitmap bitmap = mPrinter.getPdfPage(uri, pageNum);
   }
}
```

2-2-75 Get Count PDF Pages

PDF의 전체 페이지 수를 불러옵니다.

[선언]

• getCountPdfPages(Uri uri);

[매개변수]

• Uri uri: pdf file uri

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   @Override
   Public void onActivityResult(int requestCode, int resultCode, Intent data) {
       //...

       Uri uri = data.getData();
       int totalPages = mPrinter.getCountPdfPages(uri);
   }
}
```

2-2-76 Get PDF Page Height

PDF의 특정 페이지 높이를 얻어옵니다.

[선언]

• getPdfPageHeight(Uri uri, int index);

[매개변수]

• Uri uri: pdf file uri

• int index: 높이를 얻어올 PDF page 번호

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   @Override
   Public void onActivityResult(int requestCode, int resultCode, Intent data) {
       //…

       Uri uri = data.getData();
       int index = 1;
       int height = mPrinter.getPdfPageHeight(uri, index);
   }
}
```

2-2-77 Disable Inactivity Time

Wi-Fi/Ethernet 연결 시, 연결유지 기능 활성화 설정 기능입니다.

[선언]

• disableInactivityTime(boolean b);

[매개변수]

• boolean b: false는 연결유지 기능 비활성화, true는 연결유지 기능 활성화

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting() {
      mPrinter.disableInactivityTime(false);
   }
}
```

2-2-78 Firmware Download

연결한 프린터에 펌웨어 파일을 다운로드 합니다.

[선언]

• firmwareDownload(String filePath, FirmwareCallbackFunc callback);

[매개변수]

• String filePath: 펌웨어 파일 경로

• FirmwareCallbackFunc callback: 펌웨어 다운로드 진행 상황 확인을 위한 callback

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...

   private XDownloader.FirmwareCallbackFunc firmwareCallback = new
XDownloader.FirmwareCallbackFunc() { ... }

   ...

   private void updateFirmware(String filePath) {
      mPrinter.firmwareDownload(filePath, firmwareCallback);
   }
}
```

2-2-79 WLAN Firmware Download

연결한 프린터에 WLAN 모듈 펌웨어 파일을 다운로드 합니다.

[선언]

• wlanFirmwareDownload(String filePath, TransferFileCallback callback);

[매개변수]

• String filePath: WLAN 모듈 펌웨어 파일 경로

• TransferFileCallback callback: 파일 전송 진행률과 결과를 확인하기 위한 Callback

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);

   private BixolonLabelPrinter.TransferFileCallback callback = new
BixolonLabelPrinter.TransferFileCallback() { ... }
   private void updateWlanFirmware(String filePath) {
      mPrinter.wlanFirmwareDownload(filePath, callback);
   }
}
```

2-2-80 Get WLAN Info

연결한 프린터로부터 WLAN 세부 정보를 불러옵니다.

[선언]

• getWlanInfo();

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void printerSetting() {
      mPrinter.getWlanInfo();
   }
}
```

2-2-81 Get WLAN Info

nct 파일로부터 WLAN 세부 정보를 불러옵니다.

[선언]

• getWlanInfo(String filePath);

[매개변수]

• String filePath: nct 형식 파일의 경로

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void loadWlanSetting(String filePath) {

      mPrinter.getWlanInfo(filePath);
   }
}
```

2-2-82 Set WLAN Info

연결된 프린터의 WLAN 세부 정보를 설정합니다.

[선언]

• setWlanInfo(WlanInfo wlanInfo);

[매개변수]

• WlanInfo wlanInfo: WLAN 세부 정보

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   …
   private void printerSetting() {
      WlanInfo wlanInfo = mPrinter.getWlanInfo();
      wlanInfo.setSystemName(“BIXOLON”);

      mPrinter.setWlanInfo(wlanInfo);
   }
}
```

2-2-83 Set Binary Certificate File

연결된 프린터에 인증서 전송/업데이트 명령어가 작성된 파일을 전송합니다.

[선언]

• setBinaryCertificateFile(WlanInfo wlanInfo, String filePath);

[매개변수]

• WlanInfo wlanInfo: WLAN 세부 정보

• String filePath: 파일 경로

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void certificateSetting(String filePath) {
      WlanInfo wlanInfo = mPrinter.getWlanInfo();
      mPrinter.setBinaryCertificateFile(wlanInfo, filePath);
   }
}
```

2-2-84 Set Pem Certificate File

연결된 프린터에 인증서 pem 파일을 전송합니다.

[선언]

• setPemCertificateFile(WlanInfo wlanInfo, PemFileType pemFileType, String filePath);

[매개변수]

• WlanInfo wlanInfo: 인증서를 저장할 WLAN 세부 정보

• PemFileType pemFileType: 인증서 타입

타입명	설명
CA	CA 인증서
CLIENT_KEY	Client Key 인증서
CLIENT_CERT	Client PEM 인증서

• String filePath: 파일 경로

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void sendPemFile(String filePath, PemFileType type) {
      WlanInfo wlanInfo = mPrinter.getWlanInfo();
      mPrinter.setPemCertificateFile(wlanInfo, type, filePath);
   }
}
```

2-2-85 Update Certificate File

연결된 프린터에 인증서 업데이트 명령어를 전송합니다.

[선언]

• updateCertificateFile();

[예제]

```
Public Class SampleClass{
   BixolonLabelPrinter mPrinter = new BixolonLabelPrinter(this, mHandler, null);
   ...
   private void wlanUpdate() {
      mPrinter.updateCertificateFile();
   }
}
```

3. 상수값

제공하는 SDK에서 사용하는 상수 값으로 “BixolonLabelPrinter.java” 파일에 정의되어 있습니다.

3-1 Alignments

3-1-1 Device Font Alignment

디바이스 폰트의 정렬 값을 정의하는 속성입니다.

상수명	값	설명
TEXT_ALIGNMENT_NONE	48	정렬 없음
TEXT_ALIGNMENT_LEFT	70	왼쪽 정렬
TEXT_ALIGNMENT_RIGHT	76	오른쪽 정렬
TEXT_ALIGNMENT_RIGHT_TO_LEFT	82	문자열을 오른쪽에서 왼쪽 방향으로 출력

3-1-2 Vector Font Alignment

벡터폰트의 정렬 값을 정의하는 속성입니다.

상수명	값	설명
VECTOR_FONT_TEXT_ALIGNMENT_LEFT	76	왼쪽 정렬
VECTOR_FONT_TEXT_ALIGNMENT_RIGHT	82	오른쪽 정렬
VECTOR_FONT_TEXT_ALIGNMENT_CENTER	67	가운데 정렬
VECTOR_FONT_TEXT_DIRECTION_LEFT_TO_RIGHT	0	문자열을 왼쪽에서 오른쪽 방향으로 출력
VECTOR_FONT_TEXT_DIRECTION_RIGHT_TO_LEET	1	문자열을 오른쪽에서 왼쪽 방향으로 출력

3-2 Barcode HRI

• 아래 상수는 바코드 출력 시 HRI의 위치와 글꼴의 속성지정에 사용됩니다.

상수명	값	설명
HRI_NOT_PRINTED	0	HRI를 사용하지 않음
HRI_BELOW_FONT_SIZE_1	1	HRI 위치: 바코드의 하단 Font Size: 1
HRI_ABOVE_FONT_SIZE_1	2	HRI 위치: 바코드의 상단 Font Size: 1
HRI_BELOW_FONT_SIZE_2	3	HRI 위치: 바코드의 하단 Font Size: 2
HRI_ABOVE_FONT_SIZE_2	4	HRI 위치: 바코드의 상단 Font Size: 2
HRI_BELOW_FONT_SIZE_3	5	HRI 위치: 바코드의 하단 Font Size: 3
HRI_ABOVE_FONT_SIZE_3	6	HRI 위치: 바코드의 상단 Font Size: 3
HRI_BELOW_FONT_SIZE_4	7	HRI 위치: 바코드의 하단 Font Size: 4
HRI_ABOVE_FONT_SIZE_5	8	HRI 위치: 바코드의 상단 Font Size: 4

3-3 MaxiCode Modes

• Maxi code 바코드 출력 시 사용하는 바코드 옵션지정에 사용됩니다.

상수명	값	설명
_SDK_MAXICODE_MODE_0	0	MaxiCode Mode 0
_SDK_MAXICODE_MODE_2	2	MaxiCode Mode 2
_SDK_MAXICODE_MODE_3	3	MaxiCode Mode 3
_SDK_MAXICODE_MODE_4	4	MaxiCode Mode 4

3-4 1D Barcode Types

• 1차원 바코드 출력 시 사용하는 바코드 옵션에 대한 속성입니다.

상수명	값	설명
BARCODE_CODE39	0	Code39
BARCODE_CODE128	1	Code128
BARCODE_I2OF5	2	Interleaved 2of5
BARCODE_CODABAR	3	Codabar
BARCODE_CODE93	4	Code93
BARCODE_UPC_A	5	UPC-A
BARCODE_UPC_E	6	UPC-E
BARCODE_EAN13	7	EAN13
BARCODE_EAN8	8	EAN8
BARCODE_UCC_EAN128	9	UCC/EAN128
BARCODE_CODE11	10	Code11
BARCODE_PLANET	11	Planet
BARCODE_INDUSTRIAL_2OF5	12	Industrial 2of5
BARCODE_STANDARD_2OF5	13	Standard 2of5
BARCODE_LOGMARS	14	Logmars
BARCODE_UPC_EAN_EXTENSIONS	15	UPC/EAN Extensions
BARCODE_POSTNET	16	Postnet

3-5 Barcode Origin Point

• 바코드 위치의 기준점에 대한 속성입니다.

상수명	값	설명
BARCODE_ORIGIN_POINT_CENTER	0	기준점을 바코드의 한가운데로 지정합니다.
BARCODE_ORIGIN_POINT_UPPER_LEFT	1	기준점을 바코드의 좌측 상단으로 지정합니다.

3-6 Error Correction Level

• 바코드의 손상에 대비한 에러 보정레벨에 대한 속성입니다.

상수명	값	설명
PDF417_ERROR_CORRECTION_LEVEL0	0	에러 보정 레벨 0
PDF417_ERROR_CORRECTION_LEVEL1	1	에러 보정 레벨 1
PDF417_ERROR_CORRECTION_LEVEL2	2	에러 보정 레벨 2
PDF417_ERROR_CORRECTION_LEVEL3	3	에러 보정 레벨 3
PDF417_ERROR_CORRECTION_LEVEL4	4	에러 보정 레벨 4
PDF417_ERROR_CORRECTION_LEVEL5	5	에러 보정 레벨 5
PDF417_ERROR_CORRECTION_LEVEL6	6	에러 보정 레벨 6
PDF417_ERROR_CORRECTION_LEVEL7	7	에러 보정 레벨 7
PDF417_ERROR_CORRECTION_LEVEL8	8	에러 보정 레벨 8

3-7 Data Compression Method

• Data 압축 속성입니다.

상수명	값	설명
DATA_COMPRESSION_TEXT	0	2char/codeword
DATA_COMPRESSION_NUMERIC	1	2.93 char/codeword
DATA_COMPRESSION_BINARY	2	1.2bytes/codeword

3-8 QRCode Model

• QR Code 출력 시 사용하는 바코드 옵션에 대한 속성입니다.

상수명	값	설명
QR_CODE_MODEL1	1	QR Model 1
QR_CODE_MODEL2	2	QR Model 2

3-9 Code 49 Starting Mode

• Code 49 바코드 출력 시 사용하는 Starting Mode 대한 속성입니다.

상수명	값	설명
CODE49_STRING_MODE_REGULAR_ALPHANUMERIC	0	Regular Alphanumeric Mode
CODE49_STRING_MODE_MULTIPLE_READ_ALPHANUMERIC	1	Multiple Read Alphanumeric
CODE49_STRING_MODE_REGULAR_NUMERIC	2	Regular Numeric Mode
CODE49_STRING_MODE_GROUP_ALPHANUMERIC	3	Group Alphanumeric Mode
CODE49_STRING_MODE_REGULAR_ALPHANUMERIC_SHIFT1	4	Regular Alphanumeric Shift 1
CODE49_STRING_MODE_REGULAR_ALPHANUMERIC_SHIFT2	5	Regular Alphanumeric Shift 2
CODE49_STRING_MODE_AUTOMATIC_MODE	7	Automatic Mode

3-10 Codablock Mode

• 바코드 출력 시 사용하는 바코드 옵션에 대한 속성입니다.

상수명	값	설명
CODABLOCK_MODE_A	'A'	Code 39의 Character set을 사용합니다.
CODABLOCK_MODE_E	'E'	Code 128의 Character set을 사용합니다.
CODABLOCK_MODE_F	'F'	Code 128의 Character set을 사용합니다. Function 1이 자동으로 추가됩니다.

3-11 Check Digit Option

• MSI 바코드 출력 시 선택 가능 한 Check digit 속성입니다.

상수명	값	설명
MSI_BARCODE_CHECKDIGIT_NONE	0	Check Digit 없음
MSI_BARCODE_CHECKDIGIT_1MOD10	1	Check Digit 1 Mod 10
MSI_BARCODE_CHECKDIGIT_2MOD10	2	Check Digit 2 Mod 10
MSI_BARCODE_CHECKDIGIT_1MOD11_AND_1MOD_10	3	Check Digit 1 Mod 10

3-12 RSS Barcode Type

• RSS 바코드 출력 시 바코드 타입에 대한 속성입니다.

상수명	값	설명
BARCODE_TYPE_RSS14	0	RSS14
BARCODE_TYPE_RSS14_TRUNCATED	1	RSS14 truncated
BARCODE_TYPE_RSS14_STACKED	2	RSS14 stacked
BARCODE_TYPE_RSS14_STACKED_OMNIDIRECTIONAL	3	RSS14 Stacked omnidirectional
BARCODE_TYPE_RSS_LIMITIED	4	RSS limited
BARCODE_TYPE_RSS_EXPANDED	5	RSS Expanded
BARCODE_TYPE_RSS_UPCA	6	RSS UPC A
BARCODE_TYPE_RSS_UPCE	7	RSS UPC E
BARCODE_TYPE_RSS_EAN13	8	EAN13
BARCODE_TYPE_RSS_EAN8	9	EAN 8
BARCODE_TYPE_RSS_UCC_EAN128_CCAB	10	EAN128 CC-A/B
BARCODE_TYPE_RSS_UCC_EAN128_CCC	11	EAN128 CC-C

3-13 Rotation Degrees

• 출력물의 회전에 대한 속성입니다.

상수명	값	설명
ROTATION_NONE	0	회전하지 않음
ROTATION_90_DEGREES	1	90도 회전
ROTATION_180_DEGREES	2	180도 회전
ROTATION_270_DEGREES	3	270도 회전

3-14 Device Fonts

• Device Font 속성입니다.

상수명	값	설명
FONT_SIZE_6	48	9 X 15 (dots)
FONT_SIZE_8	49	12 X 20 (dots)
FONT_SIZE_10	50	16 X 25 (dots)
FONT_SIZE_12	51	19 X 30 (dots)
FONT_SIZE_15	52	24 X 38 (dots)
FONT_SIZE_20	53	32 X 40 (dots)
FONT_SIZE_30	54	48 X 76 (dots)
FONT_SIZE_14	55	22 X 34 (dots)
FONT_SIZE_18	56	28 X 44 (dots)
FONT_SIZE_24	57	37 X 58 (dots)
FONT_SIZE_KOREAN1	97	16 X 16 (dots) (ASCII 9 X 15)
FONT_SIZE_KOREAN2	98	24 X 24 (dots) (ASCII 12 X 24)
FONT_SIZE_KOREAN3	99	20 X 20 (dots) (ASCII 12 X 20)
FONT_SIZE_KOREAN4	100	26 X 26 (dots) (ASCII 16 X 30)
FONT_SIZE_KOREAN5	101	20 X 26 (dots) (ASCII 16 X 30)
FONT_SIZE_KOREAN6	102	38 X 38 (dots) (ASCII 22 X 34)
FONT_SIZE_GB2312	109	24 X 24 (dots) (ASCII 12 X 24)
FONT_SIZE_BIG5	110	24 X 24 (dots) (ASCII 12 X 24)
FONT_SIZE_SHIFT_JIS	106	24 X 24 (dots) (ASCII 12 X 24)

3-15 Vector Fonts

• Vector Font 속성입니다.

상수명	값	설명
VECTOR_FONT_ASCII	85	ASCII (1Byte code)
VECTOR_FONT_KS5601	75	KS5601 (2Byte code)
VECTOR_FONT_BIG5	66	BIG5 (2Byte code)
VECTOR_FONT_GB2312	71	GB2312 (2Byte code)
VECTOR_FONT_SHIFT_JIS	74	Shift-JIS (2Byte code)
VECTOR_FONT_OCR_A	97	OCR-A (1Byte code)
VECTOR_FONT_OCR_B	98	OCR-B (1Byte code)

3-16 Draw Block Options

• Draw Block Method 사용시 그리기 옵션과 관련된 속성입니다.

상수명	값	설명
BLOCK_OPTION_LINE_OVERWRITING	79	Line Overwriting
BLOCK_OPTION_LINE_EXCLUSIVE_OR	69	Line Exclusive OR
BLOCK_OPTION_LINE_DELETE	68	Line Delete
BLOCK_OPTION_SLOPE	83	Slope (a oblique line)
BLOCK_OPTION_BOX	66	Box

3-17 Draw Circle Sizes

• Draw Circle Method 사용시 크기와 관련된 속성입니다.

상수명	값	설명	
CIRCLE_SIZE_DIAMETER5	1	40 × 40 (dot)	5 mm
CIRCLE_SIZE_DIAMETER7	2	56 × 56 (dot)	7 mm
CIRCLE_SIZE_DIAMETER9	3	72 × 72 (dot)	9 mm
CIRCLE_SIZE_DIAMETER11	4	88 × 88 (dot)	11 mm
CIRCLE_SIZE_DIAMETER13	5	104 × 104 (dot)	13 mm
CIRCLE_SIZE_DIAMETER21	6	168 × 168 (dot)	21 mm

3-18 International Character Set

• International Character Set 속성입니다.

상수명	값	설명
INTERNATIONAL_CHARACTER_SET_USA	0	U.S.A
INTERNATIONAL_CHARACTER_SET_FRANCE	1	France
INTERNATIONAL_CHARACTER_SET_GERMANY	2	Germany
INTERNATIONAL_CHARACTER_SET_UK	3	U.K
INTERNATIONAL_CHARACTER_SET_DENMARK1	4	Denmark I
INTERNATIONAL_CHARACTER_SET_SWEDEN	5	Sweden
INTERNATIONAL_CHARACTER_SET_ITALY	6	Italy
INTERNATIONAL_CHARACTER_SET_SPAIN1	7	Spain I
INTERNATIONAL_CHARACTER_SET_NORWAY	8	Norway
INTERNATIONAL_CHARACTER_SET_DENMARK2	9	Denmark II
INTERNATIONAL_CHARACTER_SET_JAPAN	10	Japan
INTERNATIONAL_CHARACTER_SET_SPAIN2	11	Spain II
INTERNATIONAL_CHARACTER_SET_LATIN_AMERICA	12	Latin America
INTERNATIONAL_CHARACTER_SET_KOREA	13	Korea
INTERNATIONAL_CHARACTER_SET_SLOVENIA_CROATIA	14	Slovenia/Croatia
INTERNATIONAL_CHARACTER_SET_CHINA	15	China

3-19 Code Pages

• Code Page 속성입니다.

상수명	값	설명	
CODE_PAGE_CP437_USA	0	CP437	U.S.A
CODE_PAGE_CP850_LATIN1	1	CP850	Latin1
CODE_PAGE_CP852_LATIN2	2	CP 852	Latin2
CODE_PAGE_CP860_PORTUGUESE	3	CP 860	Portuguese
CODE_PAGE_CP863_CANADIAN_FRENCH	4	CP 863	Canadian French
CODE_PAGE_CP865_NORDIC	5	CP 865	Nordic
CODE_PAGE_WCP1252_LATIN1	6	WCP 1252	Latin I
CODE_PAGE_CP865_WCP1252_EUROPEAN_COMBINED	7	CP 865 + WCP 1252	European Combined
CODE_PAGE_CP857_TURKISH	8	CP 857	Turkish
CODE_PAGE_CP737_GREEK	9	CP 737	Greek
CODE_PAGE_WCP1250_LATIN2	10	WCP 1250	Latin 2
CODE_PAGE_WCP1253_GREEK	11	WCP 1253	Greek
CODE_PAGE_WCP1254_TURKISH	12	WCP 1254	Turkish
CODE_PAGE_CP855_CYRILLIC	13	CP 855	Cyrillic
CODE_PAGE_CP862_HEBREW	14	CP 862	Hebrew
CODE_PAGE_CP866_CYRILLIC	15	CP 866	Cyrillic
CODE_PAGE_WCP1251_CYRILLIC	16	WCP 1251	Cyrillic
CODE_PAGE_WCP1255_HEBREW	17	WCP 1255	Hebrew
CODE_PAGE_CP928_GREEK	18	CP 928	Greek
CODE_PAGE_CP864_ARABIC	19	CP 864	Arabic
CODE_PAGE_CP775_BALTIC	20	CP 775	Baltic
CODE_PAGE_WCP1257_BALTIC	21	WCP1257	Baltic
CODE_PAGE_CP858_LATIN1_EURO	22	CP858	Latin 1 + Euro

3-20 Printing Type

• 프린터 출력 방식 설정에 대한 속성입니다.

상수명	값	설명
PRINTING_TYPE_DIRECT_THERMAL	100	용지 직접 감열
PRINTING_TYPE_THERMAL_TRANSFER	116	리본 감열

3-21 Media Type

• 출력용지 설정에 대한 속성입니다.

상수명	값	설명
MEDIA_TYPE_GAP	71	갭 용지
MEDIA_TYPE_CONTINUOUS	67	연속 용지
MEDIA_TYPE_BLACK_MARK	66	블랙마크 용지

3-22 Speed Value

• 프린터 인쇄속도 설정에 대한 속성입니다.

상수명	값	설명
SPEED_25IPS	0	초당 2.5인치 출력
SPEED_30IPS	1	초당 3인치 출력
SPEED_40IPS	2	초당 4인치 출력
SPEED_50IPS	3	초당 5인치 출력
SPEED_60IPS	4	초당 6인치 출력
SPEED_70IPS	5	초당 7인치 출력
SPEED_80IPS	6	초당 8인치 출력

3-23 Orientation

• 출력 방향에 대한 속성입니다.

상수명	값	설명
ORIENTATION_TOP_TO_BOTTOM	84	위에서 아래로 출력
ORIENTATION_BOTTOM_TO_TOP	66	아래에서 위로 출력

3-24 Printer Status

• 프린터 오류 상태에 대한 속성입니다.

상수명	값	설명
STATUS_NORMAL	0x00	이상 없음
STATUS_1ST_BYTE_PAPER_EMPTY	0x80	용지 없음
STATUS_1ST_BYTE_COVER_OPEN	0x40	용지 커버 열림
STATUS_1ST_BYTE_CUTTER_JAMMED	0x20	Cutter 걸림
STATUS_1ST_BYTE_TPH_OVERHEAT	0x10	헤더 과열
STATUS_1ST_BYTE_AUTO_SENSING_FAILURE	0x08	갭 검출 실패
STATUS_1ST_BYTE_RIBBON_END_ERROR	0x04	리본 없음
STATUS_2ND_BYTE_BUILDING_IN_IMAGE_BUFFER	0x00000080	이미지 버퍼에 라벨 구성 중
STATUS_2ND_BYTE_PRINTING_IN_IMAGE_BUFFER	0x00000040	이미지 버퍼에 있는 라벨 인쇄 중
STATUS_2ND_BYTE_PAUSED_IN_PEELER_UNIT	0x00000020	인쇄된 라벨이 peeler에서 붙어있음

3-25 Printer Information

• 프린터의 정보에 대한 속성입니다.

상수명	값	설명
PRINTER_INFORMATION_MODEL_NAME	0	연결된 프린터의 모델명
PRINTER_INFORMATION_FIRMWARE_VERSION	2	연결된 프린터의 펌웨어 버전

3-26 Micro PDF 417 Mode List

• MicroPDF417 바코드의 모드에 대한 속성입니다.

Mode	Number of Data Columns	Number of Data Rows	% of Cws for EC	Max Alpha Characters	Max Digits
0	1	11	64	6	8
1	1	14	50	12	17
2	1	17	41	18	26
3	1	20	40	22	32
4	1	24	33	30	44
5	1	28	29	38	55
6	2	8	50	14	20
7	2	11	41	24	35
8	2	14	32	36	52
9	2	17	29	46	67
10	2	20	28	56	82
11	2	23	28	64	93
12	2	26	29	72	105
13	3	6	67	10	14
14	3	8	58	18	26
15	3	10	53	26	38
16	3	12	50	34	49
17	3	15	47	46	67
18	3	20	43	66	96
19	3	26	41	90	132
20	3	32	40	114	167
21	3	38	39	138	202
22	3	44	38	162	237
23	4	6	50	22	32
24	4	8	44	34	49
25	4	10	40	46	67
26	4	12	38	58	85
27	4	15	35	76	111
28	4	20	33	106	155
29	4	26	31	142	208
30	4	32	30	178	261
31	4	38	29	214	313
32	4	44	28	250	366
33	4	4	50	14	20

3-27 PDF 417 Barcode HRI

• PDF417 바코드의 HRI 에 대한 속성입니다.

상수명	값	설명
PDF417_HRI_NOT_PRINTED	0	HRI 인쇄하지 않음
PDF417_HRI_BELOW_BARCODE	1	바코드 아래에 인쇄

3-28 CODE 49 Barcode HRI

• CODE49 바코드의 HRI 에 대한 속성입니다.

상수명	값	설명
CODE49_HRI_NOT_PRINTED	0	HRI 인쇄하지 않음
CODE49_HRI_BELOW_BARCODE	1	바코드 아래에 인쇄
CODE49_HRI_ABOVE_BARCODE	2	바코드 위에 인쇄

3-29 PLESSEY Barcode HRI

• PLESSEY 바코드의 HRI 에 대한 속성입니다.

상수명	값	설명
PLESSEY_BARCODE_HRI_NOT_PRINTED	0	HRI 인쇄하지 않음
PLESSEY_BARCODE_HRI_BELOW_BARCODE	1	바코드 아래에 인쇄
PLESSEY_BARCODE_HRI_ABOVE_BARCODE	2	바코드 위에 인쇄

3-30 MSI Barcode HRI

• MSI 바코드의 HRI 에 대한 속성입니다.

상수명	값	설명
MSI_BARCODE_HRI_NOT_PRINTED	0	HRI 인쇄하지 않음
MSI_BARCODE_HRI_BELOW_BARCODE	1	바코드 아래에 인쇄
MSI_BARCODE_HRI_ABOVE_BARCODE	2	바코드 위에 인쇄

4. 부록

4-1 개발 환경 설정

4-1-1 Manifest 권한 설정

• 블루투스 권한

- BLUETOOTH: 커넥션 요구, 커넥션 수락, 데이터 전송 등의 블루투스 통신 권한
- BLUETOOTH_ADMIN: 디바이스 검색 시작, 블루투스 설정 조작 권한

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

• Wi-Fi 권한

- ACCESS_WIFI_STATE: Wi-Fi 연결 여부 확인 권한
- CHANGE_WIFI_STATE: Wi-Fi 상태 변경 여부 확인 권한
- CHANGE_WIFI_MULTICAST_STATE: Wi-Fi 멀티캐스트 모드 사용 권한

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE" />
```

• Internet 권한

- INTERNET: 네트워크 소켓 오픈 권한
- ACCESS_NETWORK_STATE: 네트워크 정보 접근 권한

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

• 저장소 권한

- WRITE_EXTERNAL_STORAGE: 외부저장소 사용 권한

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

• USB 권한

- ACTION_USB_DEVICE_ATTACHED: USB 연결 권한

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
    <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
</intent-filter>
<meta-data
    android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
    android:resource="@xml/device_filter" />
```

4-1-2 안드로이드 장치 연결

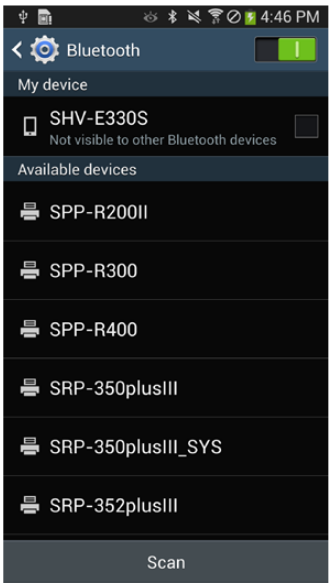
아래 스크린 샷은 Android 4.2 스마트폰에서 캡쳐 한 화면입니다. 스크린샷과 항목 이름은 Android 운영체제 또는 장치에 따라 다를 수 있습니다.

1. Bluetooth

1) Settings을 선택합니다.

2) Android 기기의 블루투스가 켜져 있고, 프린터의 전원이 켜져 있어야 합니다.

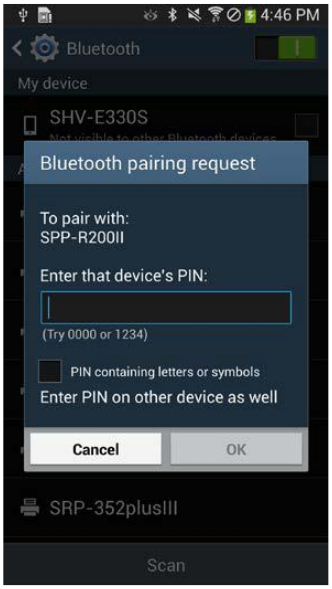
3) 설정을 위해 블루투스를 선택합니다.



4) Scan을 선택합니다.

5) 연결하기 위한 프린터를 선택하고 페어링 합니다.

6) PIN code를 입력합니다. ㈜빅솔론의 초기 PIN code는 “0000” 입니다.



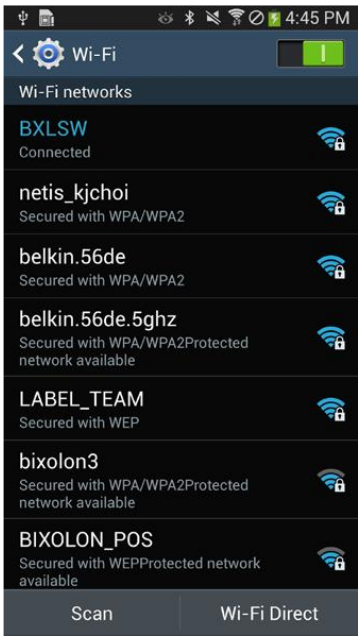
2. Wi-Fi

1) 프린터를 네트워크 AP(Access Point)에 연결하고 IP 주소를 할당하거나 DHCP로 설정합니다. ㈜빅솔론의 프린터는 초기에 Ad-hoc으로 설정되어 있기 때문에, 마스터CD에 포함되어 있는 Net Configuration Tool로 최초 한 번은 설정을 해야 됩니다.
Net Configuration Tool은 ㈜빅솔론 홈페이지(www.bixolon.co.kr)에서도 다운로드 받으실 수 있습니다.

2) Settings를 선택합니다.

3) Android 기기의 Wi-Fi가 켜져 있고, 프린터의 전원이 켜져 있어야 합니다.

4) ㈜빅솔론의 프린터가 연결되어 있는 네트워크와 동일한 네트워크에 연결합니다.



5) Android 장치를 프린터의 TCP/IP 포트에 연결하기 위해 추가 설정은 필요하지 않습니다.

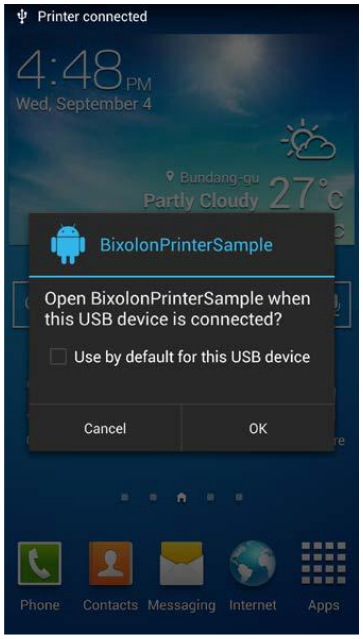
3. USB

1) Android 장치는 OS 버전 3.1 이상부터 USB 주변 장치 연결이 가능합니다.

2) Android 장치에 ㈜빅솔론의 특정 드라이버나 프린터 소프트웨어가 설치되어 있을 필요는 없습니다.

3) 필요한 USB 케이블은 스마트폰이나 태블릿에 따라 달라질 수 있습니다.
대부분의 Android 장치는 A to B USB 케이블을 제공하지 않습니다. Mini/Micro USB 케이블이나 아답터/dock이 필요할 수 있습니다. 사용하려는 안드로이드 장치에 알맞은 케이블을 사용하는지 확인해야 합니다.

4) ㈜빅솔론의 프린터를 처음 연결하는 경우, Android 장치에 따라서 아래와 같은 메시지가 나타날 수 있습니다.



5) USB 주변 장치를 연결하기 위해서는 아래와 같은 코드를 AndroidManifest.xml과 res/xml/device_filter.xml에 입력해야 합니다.

[AndroidManifest.xml]

```
<intent-filter>
    <action
android:name=”android.hardware.usb.action.USB_DEVICE_ATTACHED”/>
</intent-filter>
<meta-data
    android:name=”android.hardware.usb.action.USB_DEVICE_ATTACHED”
    android:resource=”@xml/device_filter” />
```

[device_filter.xml]

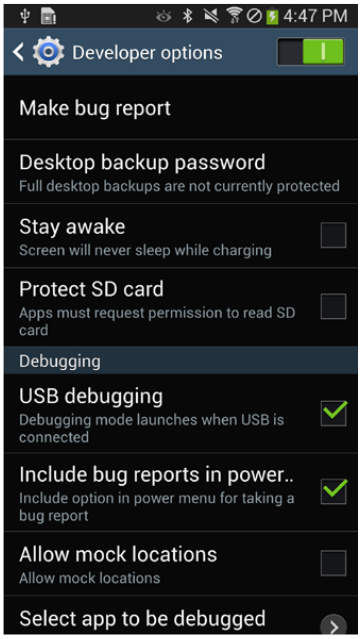
```
<?xml version=”1.0” encoding=”utf-8”>
<resources>
    <usb-device
        class=”7”
        protocol=”2”
        subclass=”1”
        vendor-id=”5380” />
</resources>
```

4-1-3 Android 장치 개발자 옵션 설정

1) Settings를 선택합니다.

2) Developer options를 선택합니다.

3) USB debugging을 활성화합니다.



4) Developer options가 보이지 않을 경우

4-1 Settings를 선택합니다.

4-2 About device를 선택합니다.

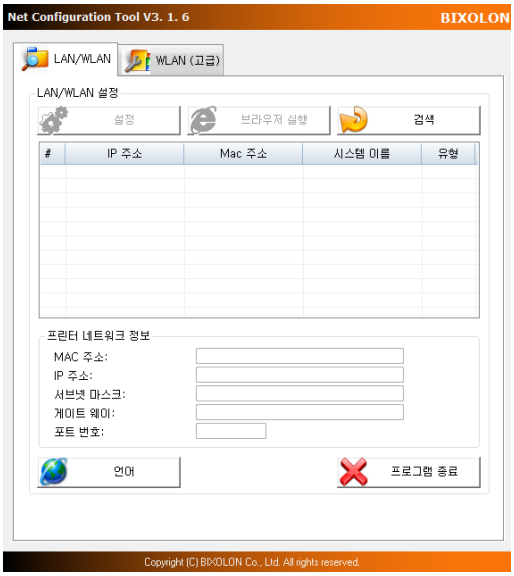
4-3 Software info를 선택합니다.

4-4 Build number를 연타하여 Developer options을 활성화시킵니다.

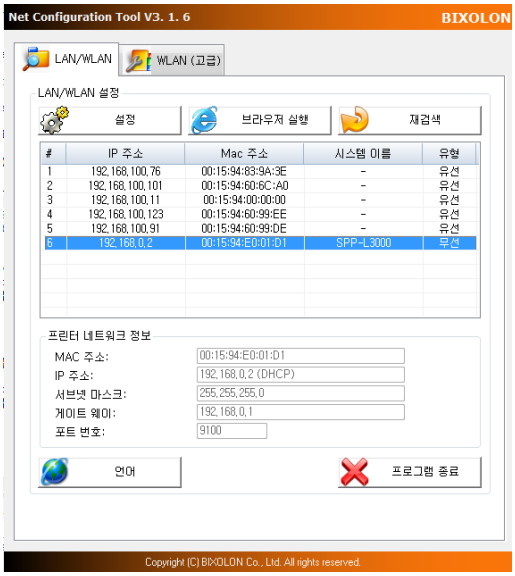
4-1-4 Net Configuration Tool 사용

1) ㈜빅솔론의 웹사이트에서 Net Configuration Tool을 다운로드 받아 설치합니다.

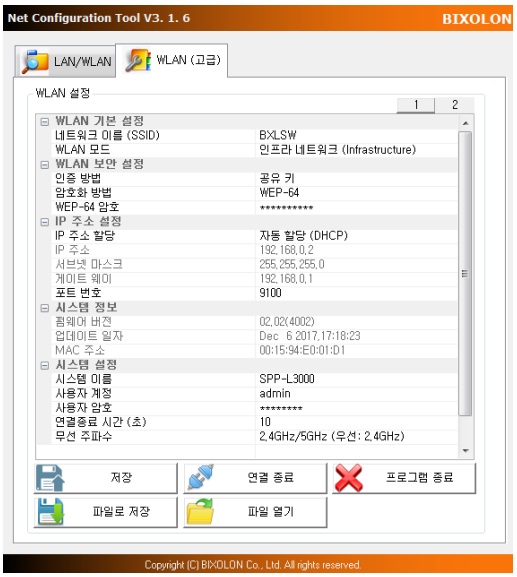
2) 다운로드 받은 Net Configuration Tool을 실행합니다.



3) 검색을 눌러 현재 AP에 연결되어 있는 프린터를 검색합니다.



4) 연결하고자 하는 프린터가 검색되지 않을 시, WLAN(고급) 탭으로 이동하여 장치를 연결합니다.



5) 프린터를 연결할 네트워크 설정을 입력합니다.

6) 설정 완료 후 저장을 누르고, 프린터를 재시작 합니다.

7) 프린터의 Interface 항목의 WLAN으로 진입하여 IP 주소를 확인합니다.

4-1-5 gradle 파일 설정

- Java로 작성된 프로젝트에 V2.0.0 이후의 Label SDK로 업데이트하기 위해선 gradle 파일 내용의 일부 수정이 필요할 수 있습니다.

1. App 수준 gradle 파일 수정

1) 상단 plugin block에 'org.jetbrains.kotlin.android' 추가

2) 하단 dependencies block에 implementation 'androidx.core:corektx:${kotlinversion}' 추가

3) 필요 시, dependencies block에 appcompat 버전 수정

4) 필요 시, 중단 android block에 아래 문구 추가

```
compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
```

5) 프로젝트에 설정한 Kotlin 버전에 맞게 sdk version 수정

2. Project 수준 gradle 파일 수정

1) buildscript 블록과 allprojects 블록 중간에 아래 문구 추가
plugins {id 'org.jetbrains.kotlin.android' version ${kotlin_version} apply false}

2) buildscript 및 allprojects 블록 내 repositories에 아래 문구 추가
maverCentral() 또는 maven { url 'https://maven.google.com' }

3) buildscript 블록 내 dependencies 블록 gradle 버전 6.1.1 이상으로 수정

3. gradle.properties 파일 수정

위 파일 내 아래 문구 추가
android.useAndroidX=true
android.enableJetifier=true

- gradle 파일 수정 내용은 적용할 프로젝트의 현재 설정에 따라 달라질 수 있습니다. 위 내용을 참고하여, 프로젝트에 맞게 적용 바랍니다.
- 위 내용 외 추가적인 수정 내용이 필요할 수 있습니다.
- gradle 파일 수정 내용 적용 예시는 배포 파일에 포함된 Sample 코드를 참고 바랍니다.

개정 이력

Ver.	날짜	설명
1.00	2018-05-23	신규제작
1.01	2019-01-28	지원 모델 추가 버퍼에 이미지 입력함수 추가(Path 사용) 이미지 버퍼에 이미지 입력 시 디더링 적용 옵션 추가
1.02	2019-03-04	XD3-40d 지원 모델 추가
1.03	2019-04-22	XQ-840 지원 모델 추가 DataMatrix 미지원 옵션 제거
1.04	2019-06-24	XD3-40d 지원 모델 제거 XQ-843, XL5-40/43, XD5-40d/43d 지원 모델 추가
1.05	2019-09-26	Pdf 인쇄 api 삭제
1.06	2019-10-17	XL5-40CT/43CT 모델명 변경
1.07	2019-11-27	XD5-40t, XD5-43t, XM7-40 지원 모델 추가
1.08	2020-03-12	XM7-20 지원 모델 추가
1.09	2020-09-15	RFID 지원 여부 표기 지원 함수 목록 추가 RFID api 추가
1.10	2020-10-05	SRP-S3000_LABEL 지원 모델 추가
1.11	2020-12-01	drawImage, drawImageFile 함수 추가
1.12	2021-03-05	SRP_S3000_LABEL 블루투스 인터페이스 지원 여부 변경
1.13	2021-03-18	Rotation 적용 가능한 drawImage API 추가
1.14	2021-06-14	API 추가 - beginTransactionPrint, endTransactionPrint Noto Sans 폰트 적용
1.15	2021-07-30	-XM7-40 RFID, XT3-40/43 지원 내용 추가
1.16	2021-11-23	-drawBase64Image API 설명 추가 - 설명 문구 변경 : “참조: 1-3 각 모델 별 좌표의 사용 범위” -> “참조: 1-4 각 모델 별 좌표의 사용 범위”

1.17	2022-04-13	-XM7-30 지원 내용 추가 - calibrateRFID API 내용 삭제
1.18	2022-07-15	- transferFile API 지원 내용 추가 - gradle 파일 설정 내용 추가
1.19	2022-09-06	drawPDFFile, setPdfDpi, getPdfPage, getCountPdfPages, getPdfPageHeight API 설명 추가
1.20	2023-05-17	disableInactivityTime API 추가
1.21	2024-01-05	XQ-840II, XQ-843II 지원 모델 추가
1.22	2024-08-20	executeDirectIO API 추가 firmwareDownload, wlanFirmwareDownload, getWlanInfo, setWlanInfo, setBinaryCertificateFile, setPemCertificateFile, updateCertificateFile API 추가
1.23	2025-01-15	XD5-40IId, XD5-43IId, XD5-40IIt, XD5-43IIt 지원 모델 추가
1.24	2025-04-24	BT3-40, BT3-43, BT5-40, BT5-43, BT5-46, BD5-40d, BD5-43d, BD5-40t, BD5-43t 지원 모델 추가
1.25	2025-09-15	XM5-30, XT6-60, XT6-63 지원 모델 추가