Cubinote Bluetooth SDK

Version 1.0 Beta by Simon Wei

Aug 30, 2018

Overview

Cubinote Bluetooth SDK provides multiple platforms APIs for third-party applications to print with Cubinote via Bluetooth.

Supported Platforms

Windows, Mac, iOS, Android.

APIs

1. CubinoteBLE_OpenSession

   Open the session of accessory to read and write on iOS and Mac.

   1) Definition:

   iOS: CubinoteBLE_OpenSession(device: EAAcessory, callback: (String) -> Void)) -> String

   2) Parameters:

   device: EAAcessory: The accessory device which Cubinote is connected or paired.

   callback: The callback function to handle notifications.

   3) Return value (JSON):

   ```
   {
          "errorCode": int, //see Error Codes
   }
   ```

2. CubinoteBLE_CloseSession

   Close the opened session of accessory on iOS and Mac.

   1) Definition:

   iOS: CubinoteBLE_CloseSession() -> String

2) Return value (JSON):

```
{
        “errorCode”: int, //see Error Codes
}
```

3. CubinoteBLE_GetStatus

Read the status of Cubinote which is connected to the specified port.

1) Definition:

Windows: String CubinoteBLE_GetStatus(String portName)

Android: String CubinoteBLE_GetStatus(BluetoothDevice device)

iOS: CubinoteBLE_GetStatus() -> String

2) Parameters:

portName: String; The name of the port to which Cubinote is connected or paired.

device: BluetoothDevice; The Bluetooth device which Cubinote is connected or paired.

3) Return value (JSON):

Success:

```
{
    "command": 5,
    "printerState": int,
    "busy": int,        //0-idle; 1-printing
    "printTableID": int,     //the ID of current printing note
    "packageCount": int,         //the package count of current printing note
    "packageNo": int             //the number of current printing note
}
```

Failed:

```
{
        “errorCode”: int, see Error Codes
}
```

4. CubinoteBLE_Set

Set the options of Cubanite which is connected to the specified port.

1) Definition:

Windows: String CubinoteBLE_Set(String portName, int led, int buz, int speed, int languageId)

Android: String CubinoteBLE_Set(BluetoothDevice device, int led, int buz, int speed, int languageId)

iOS: CubinoteBLE_Set(led: Int, buz: Int, speed: Int, languageId: Int) -> String

2) Parameters:

portName: String; The name of the port to which Cubinote is connected or paired.

device: BluetoothDevice; The Bluetooth device which Cubinote is connected or paired.

led: int; Turn off|on of the led on Cubinote. 0-off; 1-on.

buz: int; Turn off|on of the buz in Cubinote. 0-off; 1-on.

speed: int; Set the print speed of Cubinote. 0-21; 0-slowest; 21-fastest.

languageId: int; Change the language. 0-Simplified Chinese; 1-English.

3) Return value (JSON):

Success:

```
{
    "command": 2,
    "led": int,          //0-off; 1-on
    "buz": int,          //0-off; 1-on
    "speed": int,        //0-21
    "languageId": int,          //0-Simplified Chinese; 1-English
}
```

Failed:

```
{
      "errorCode": int, see Error Codes
}
```

5. CubinoteBLE_Print_BWImage

Print a **Monochrome bitmap image** with the Cubanite which is connected to the specified port.

Important notes:

- The image is a Monochrome bitmap, that is, a Monochrome bitmap, and 1 bit corresponds to one pixel, and the pixel is black or white. Therefore, Non-Mono images need to be processed and converted by the 3rd party app before sending to Cubinote to print.
- The width must be equal or less than to the number of pixels of Cubinote, which is 576 pixels.
- The bitmap image needs to rotate the picture 180 degrees clockwise and then flip it horizontally before sending to Cubinote.

1) Definition:

Windows: String CubinoteBLE_Print_BWImage(String portName, byte[] pImage)

Android: String CubinoteBLE_Print_BWImage(BluetoothDevice device, byte[] pImage)

iOS: CubinoteBLE_Print_BWImage(pImage: NSData) -> String

2) Parameters:

portName: String; The name of the port to which Cubinote is connected or paired.

device: BluetoothDevice; The Bluetooth device which Cubinote is connected or paired.

pImage: byte[]; The data of monochrome bitmap image to be printed.

pImage: NSData; The data of monochrome bitmap image to be printed.

3) Return value (JSON):

Success:

```
{
    "command": 3,
    "msgType": 1,                //Message type
    "printID": int,    //the ID of current printing note
    "packageCount": int,         //the package count of current printing note
    "packageNo": int             //the number of current printing note
    "result": 1                  //1:OK
}
```

Failed:

```
{
    "errorCode": int. //see Error Codes.
}
```

6. CubinoteBLE_Print_Content

Print **Structed contents** (see Class InnerContent) with the Cubanite which is connected to the specified port.

Important notes:

- When adding a Monochrome bitmap image into TextItem, it must be Base64 encoded into a string. But other contents mustn't be Base64 encoded, such as text, QR code.
- The image is a Monochrome bitmap, that is, a Monochrome bitmap, and 1 bit corresponds to one pixel, and the pixel is black or white. Therefore, Non-Mono images need to be processed and converted by the 3rd party app before sending to Cubinote to print.
- The width of image must be equal or less than to the number of pixels of Cubinote, which is 576 pixels.
- The bitmap image needs to rotate the picture 180 degrees clockwise and then flip it horizontally before sending to Cubinote.

1) Definition:

Windows: String CubinoteBLE_Print_Content(String portName, InnerContent inerContent)

Android: String CubinoteBLE_Print_Content(BluetoothDevice device, InnerContent inerContent)

iOS: CubinoteBLE_Print_Content(inerContent: InnerContent) -> String

2) Parameters:

portName: String; The name of the port to which Cubinote is connected or paired.

device: BluetoothDevice; The Bluetooth device which Cubinote is connected or paired.

inerContent: InnerContent; A class that contains TextItems to be printed. For the definitions of InnerContent and TextItem. See them below.

3) Return value (JSON):

Success:

```
{
    "command": 3,
    "msgType": 1,                    //Message type
    "printID": int,    //the ID of current printing note
    "packageCount": int,             //the package count of current printing note
    "packageNo": int                 //the number of current printing note
    "result": 1                      //1:OK
}
```

Failed:

```
{
      "errorCode": int. //see Error Codes.
}
```

Classes

1. CubinoteBLE

All the APIs, public methods and Error Codes are encapsulated int the class CubinoteBLE.

1) Error Codes
   - i. int CubinoteBLE_OK = 0; //Operation is successful.
   - ii. int CubinoteBLE_ERR_Invalid_DeviceName = -1; //Device is invalid.
   - iii. int CubinoteBLE_ERR_Open_Device_Failed = -2; //Open device failed.
   - iv. int CubinoteBLE_ERR_Parameter_Led = -3; //Value of led is invalid.
   - v. int CubinoteBLE_ERR_Parameter_Buz = -4; //Value of buz is invalid.
   - vi. int CubinoteBLE_ERR_Parameter_Speed = -5; //Value of speed is invalid.
   - vii. int CubinoteBLE_ERR_Parameter_LanguageId = -6; //Value of languageId is invalid.
   - viii. int CubinoteBLE_ERR_InvalidOperationException = -7; //Exception InvalidOperationException
   - ix. int CubinoteBLE_ERR_ArgumentOutOfRangeException = -8; //Exception ArgumentOutOfRangeException
   - x. int CubinoteBLE_ERR_ArgumentNullException = -9; //Exception ArgumentNullException
   - xi. int CubinoteBLE_ERR_ArgumentException = -10; //Exception ArgumentException
   - xii. int CubinoteBLE_ERR_IOException = -11; //Exception IOException

xiii. int CubinoteBLE_ERR_TimeoutException = -12; //Exception TimeoutException
xiv. int CubinoteBLE_ERR_UnauthorizedAccessException = -13; //Exception UnauthorizedAccessException
xv. int CubinoteBLE_ERR_Exception = -14; //Exception
xvi. int CubinoteBLE_ERR_Session_Not_Opened = -15; //Session isn't opened, please open it first
xvii. int CubinoteBLE_ERR_Session_Busy = -16; //Session is busy, try it later

2. InnerContent

The InnerContent class contains a list of TextItems to be printed.

Properties:

1) textList: List of TextItem. Call textList.Add() to append a TextItem.

Construction:

1) InnerContent(): Create an instance of InnerContent with an empty textList.

2) InnerContent(TextItem item): Create an instance of InnerContent and insert item into the textList.

3. TextItem

The TextItem class describes an item of printing content which could be a text with style, a material, a QR code or a monochrome bitmap image.

Properties:

1) basetext: String. A string that holds the content of text or base64 encoded monochrome bitmap image.

2) bold: int; The basetext is bold or not. 0-not, default; 1-bold.

3) fontSize: int; The font size of the basetext. 1-regular, default; 2-big.

4) iconID: int; The ID of material. Default is 0.

5) printType: int; The type of TextItem. 1-text; 2-reserved; 3-QR code; 4-material; 5-image, default.

Construction:

1) TextItem(String basetext): Create an image item. The basetext is the base64 encoded.

2) TextItem(String basetext, int printType): Create a text(printType=1), or image (printType=5), or QR code(printType=3) item.

3) TextItem(int iconID): Create a material item.