# 一、前言

BH-400设备实现批量采集标签的功能。

# 二、设备使用场景分析（用例）

## 1. 场景一：批量采集标签

### 1.1 使用流程分析

使用界面进行简单说明：

Step 1 用户输入批次号、标签总数量，点击“开始批量采集”

Step 2 界面向用户展示已采集标签张数

Step 3 当采集张数等于输入总张数，采集完成

Step 4 展示采集结果







### 1.2 软件协作图



## 2. 场景二：获取批量采集批次列表

### 1.1 使用流程分析

使用界面进行简单说明：

Step 1 进入查询页面，页面自动向服务器请求批次列表，展示返回结果



### 1.2 软件协作图



## 3. 场景三：获取某批次采集的标签信息列表

### 1.1 使用流程分析

使用界面进行简单说明：

Step 1 用户点击某一批次采集

Step 2 页面向服务器请求该次采集的标签信息列表，展示返回结果



### 1.2 软件协作图



## 4. 场景四：修改标签Id

BH-400算法识别标签Id内容有误，或者识别失败时，需要人为修正或补充。

### 1.1 使用流程分析

Step 1浏览器向服务器请求某次采集的标签信息列表，展示返回结果。单个标签状态有待识别标签Id、识别标签Id失败等（状态定义见第四章的第4小节）

Step 2 对识别有误或识别失败的，用户输入正确的标签Id，发送到Web Server完成修改



### 1.2 软件协作图



## 5. 场景五：审核

某批次采集完成后，需要人工审核采集结果，审核通过后，由Web Server上传兆日服务器。

### 1.1 使用流程分析

Step 1浏览器向服务器请求某次采集的标签信息列表，展示返回结果

Step 2 用户可以选择审核通过（有标签Id识别失败的，需要修正后，才能选择审核通过）或者作废



### 1.2 软件协作图



## 6. 场景六：获取一张预览图

### 1.1 使用流程分析

Step 1 浏览器向Web Server请求获取一张预览图。设备采集预览图完成后，Web Server返回采集结果（此过程为浏览器同步阻塞等待采集完成）

### 1.2 软件协作图



## 7. 场景七：采集一张纸纹图

### 1.1 使用流程分析

Step 1 用户点击“采集一张纸纹图”

Step 2 页面向Web Server请求采集一张纸纹图。设备采集纸纹完成后，Web Server返回采集结果（此过程为浏览器同步阻塞等待采集完成）

Step 3 页面向Web Server发送审核通过请求，其中批次号为第2步返回的“LabelId”和“CaptureTime”组合成的字符串

### 1.2 软件协作图



# 三、整体软件框图



说明：

1. BH-400软件分为两部分：

（1）Web Server：对外提供http接口。采集图像功能通过调用纸纹设备SDK接口实现。

（2）纸纹设备SDK：提供采集图像等接口。批量采集的触发源为GPIO口信号（硬触发）。

2. 图像存储与管理：纸纹设备SDK拍照后保存图像数据到硬盘，Web Server根据客户端的请求进行读取和删除。

3. 用户使用浏览器操作BH-400设备。

# 四、Web Server接口定义

## 1. 浏览器调用Web Server接口方式

浏览器访问Web Server接口，是通过前后端分离的方式：

（1）首先在浏览器地址栏输入网址http://ip:port，向Web Server 请求静态页面。Web Server收到请求，返回预先部署的Web页面。

（2）浏览器得到Web页面后，用户可以点击相关按钮，触发浏览器通过http协议向Web Server发送请求。

## 2. 接口公共内容

**（1）Header内容**

|  |  |
| --- | --- |
| **参数** | **值** |
| Content-Type | application/json;charset=utf-8 |
| Authorization | 身份验证信息  目前可以先不填写 |

**（2）请求内容结构**

无固定格式

**（3）返回内容结构**

|  |
| --- |
| {  "ErrCode": 0,  "ErrMsg": "OK",  "Data": {} } |
|
|
|
|

说明：

ErrCode：错误码

ErrMsg：错误信息

Data：业务数据

## 3. 接口定义

### 3.1 开始批量采集

**（1）接口描述**

本接口用于开启批量采集。

HTTP 方法：POST

请求URL：

http://ip:port/api/v1/batchcapture/start

**（2）输入参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| BatchId | String | 是 | 批次号 |
| TotalLabelNum | Integer | 是 | 该批次的标签总数 |

**（3）输出参数**

成功/失败。

**（4）示例**

输入示例

|  |
| --- |
| {  "BatchId": “202312061033”,  “TotalLabelNum”: 100 } |
|
|
|
|

输出示例

|  |
| --- |
| {  “ErrCode”: 0,  “ErrMsg”: “OK” } |
|
|
|
|

### 3.2 查询批量采集状态

**（1）接口描述**

本接口用于查询批量采集状态。

HTTP 方法：POST

请求URL：

http://ip:port/api/v1/batchcapture/checkstatus

**（2）输入参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| BatchId | String | 是 | 批次号 |
| GetCapturedLabelInfos | Boolean | 否 | 是否返回已采集标签信息列表，默认值为false |
| Offset | Integer | 否 | 分页的偏移量，默认值为0  示例值：0 |
| Limit | Integer | 否 | 分页单页限制数目，默认值为20，最大值100  示例值：5 |

**（3）输出参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| CapturedLabelsNum | Integer | 是 | 已采集标签张数 |
| Status | Integer | 是 | 采集状态，1表示正在采集，2表示采集完成 |
| LabelInfos | LabelInfo 数组 | 否 | 已采集标签信息列表  LabelInfo定义见下面 |

LabelInfo定义：

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| Index | Integer | 是 | 标签序号 |
| Status | Integer | 是 | 标签的状态，定义见第4节 |
| LabelId | String | 是 | 标签Id |
| ImgUrl | String | 是 | 标签图像url  示例值：http://ip:port/api/v1/imgs/20231129143030/1.jpg |

**（4）示例**

输入示例

|  |
| --- |
| {  "BatchId": "20231129144622" } |
|
|
|
|

输出示例

|  |
| --- |
| {  "ErrCode": 0,  "ErrMsg": "OK",  "Data": {  "CapturedLabelsNum": 51,   "Status": 1  } } |
|
|
|
|

### 3.3 获取批量采集批次列表

**（1）接口描述**

本接口用于获取批量采集批次列表。

HTTP 方法：POST

请求URL：

http://ip:port/api/v1/batchcapture/getallbatches

**（2）输入参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| Status | Integer数组 | 否 | 批次的整体状态，定义见第4节 |
| StartTime | Integer | 否 | 查询起始时间，秒级Unix时间戳  示例值：1695193830 |
| EndTime | Integer | 否 | 查询结束时间，秒级Unix时间戳  示例值：1695197431 |
| Offset | Integer | 否 | 分页的偏移量，默认值为0  示例值：0 |
| Limit | Integer | 否 | 分页单页限制数目，默认值为20，最大值100  示例值：5 |

**（3）输出参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| AllBatches | BatchInfo 数组 | 是 | 所有批次信息  BatchInfo定义见下面 |
| TotalCount | Integer | 是 | 总数目 |
| TotalPageSize | Integer | 是 | 总页数 |

BatchInfo定义：

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| BatchId | String | 是 | 批次号 |
| CaptureTime | Integer | 是 | 采集时间，秒级Unix时间戳  示例值：1695197431 |
| TotalCount | Integer | 是 | 采集总张数 |
| Status | Integer | 是 | 批次的整体状态，定义见第4节 |

**（4）示例**

输入示例

|  |
| --- |
| {  "StartTime": 1695193830,  "EndTime": 1695197431,  "Offset": 0,  "Limit": 2 } |
|
|
|
|

输出示例

|  |
| --- |
| {  "ErrCode": 0,  "ErrMsg": "OK",  "Data": {  "AllBatches": [  {  "BatchId": "20231129144622",  "CaptureTime": 1695197431,  "TotalCount": 100,  "Status": 2  },  {  "BatchId": "20231129155002",  "CaptureTime": 1695197566,  "TotalCount": 100,  "Status": 1  }  ],   "TotalCount": 10  "TotalPageSize": 1  } } |
|
|
|
|

### 3.4 获取某批次采集的标签信息列表

**（1）接口描述**

本接口用于获取某批次采集的标签信息列表。

HTTP 方法：POST

请求URL：

http://ip:port/api/v1/batchcapture/getonebatchdetail

**（2）输入参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| BatchId | String | 是 | 批次号 |
| Offset | Integer | 否 | 分页的偏移量，默认值为0  示例值：0 |
| Limit | Integer | 否 | 分页单页限制数目，默认值为20，最大值100  示例值：5 |

**（3）输出参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| LabelInfos | LabelInfo 数组 | 是 | 某批次标签信息列表  LabelInfo定义见下面 |
| TotalCount | Integer | 是 | 总数目 |
| TotalPageSize | Integer | 是 | 总页数 |

LabelInfo定义：

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **必选** | **描述** |
| Index | Integer | 是 | 标签序号 |
| Status | Integer | 是 | 标签的状态，定义见第4节 |
| LabelId | String | 是 | 标签Id |
| ImgUrl | String | 是 | 标签图像url  示例值：http://ip:port/api/v1/imgs/20231129143030/1.jpg |
| IssuedTime | Integer | 否 | 录入时间，秒级Unix时间戳  示例值：1695197431 |

**（4）示例**

输入示例

|  |
| --- |
| {  "TransId": "20231129144622",  "Offset": 0,  "Limit": 2 } |
|
|
|
|

输出示例

|  |
| --- |
| {  "ErrCode": 0,  "ErrMsg": "OK",  "Data": {  "TransId": "20231129144622",  "LableInfos": [  {  "Index": 1,  "Status": 1,  "LabelId": "",  "ImgUrl": "http://ip:port/api/v1/imgs/20231129144622/123.jpg",  "IssuedTime": 1695197431  },  {  "Index": 2,  "Status": 3,  "LabelId": "456",  "ImgUrl": "http://ip:port/api/v1/imgs/20231129144622/456.jpg",  "IssuedTime": 1695197431  }  ],   "TotalCount": 100  "TotalPageSize": 10  } } |
|
|
|
|

### 3.5 修改标签Id

**（1）接口描述**

本接口用于修改标签Id。

HTTP 方法：POST

请求URL：

http://ip:port/api/v1/batchcapture/modifylabelid

**（2）输入参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| BatchId | String | 是 | 批次号 |
| Index | Integer | 是 | 标签序号 |
| LabelId | String | 是 | 标签Id，非空 |

**（3）输出参数**

成功/失败。

**（4）示例**

输入示例

|  |
| --- |
| {  "BatchId": "20231129144622",  "Index": 1,  "LabelId": "123" } |
|
|
|
|

输出示例

|  |
| --- |
| {  "ErrCode": 0,  "ErrMsg": "OK" } |
|
|
|
|

### 3.6 审核

**（1）接口描述**

本接口用于审核某批次采集结果。

HTTP 方法：POST

请求URL：

http://ip:port/api/v1/batchcapture/checkresult

**（2）输入参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| BatchId | String | 是 | 批次号 |
| Result | Integer | 是 | 审核结果，1表示通过，2表示作废 |

**（3）输出参数**

成功/失败。

**（4）示例**

输入示例

|  |
| --- |
| {  "BatchId": "20231129144622",  "Result": 1 } |
|
|
|
|

输出示例

|  |
| --- |
| {  "ErrCode": 0,  "ErrMsg": "OK" } |
|
|
|
|

### 3.7 获取图像

**（1）接口描述**

本接口用于获取图像。

HTTP 方法：GET

请求URL：

http://ip:port/api/v1/imgs/xxxxx/xxx.jpg

以上URL为获取原图，还可以获取缩略图（列表建议获取缩略图），请求URL：

http://ip:port/api/v1/imgs/xxxxx/xxx.jpg?imgtype=thumbnail

**（2）输入参数**

无

**（3）输出参数**

返回报文的Header为：

|  |  |
| --- | --- |
| **参数** | **值** |
| Content-Type | image/jpeg |

报文体是图像数据。

**（4）示例**

html页面通过<img src=”img\_url”>可以自动加载显示图像。

### 3.8 获取一张预览图

**（1）接口描述**

本接口用于获取一张预览图。

HTTP 方法：POST

请求URL：

http://ip:port/api/v1/batchcapture/getpreviewimg

**（2）输入参数**

无

**（3）输出参数**

返回报文的Header为：

|  |  |
| --- | --- |
| **参数** | **值** |
| Content-Type | image/jpeg |

报文体是图像数据。

**（4）示例**

输入示例

|  |
| --- |
| { } |
|
|
|
|

输出示例

|  |
| --- |
| 图像数据 |
|
|
|
|

### 3.9 采集一张纸纹图

**（1）接口描述**

本接口用于采集一张纸纹图。

HTTP 方法：POST

请求URL：

http://ip:port/api/v1/capture/get

**（2）输入参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| LabelId | String | 否 | 标签Id  有此参数，被用作采集纸纹的Id  无此参数，需要识别纸纹Id |

**（3）输出参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **必选** | **描述** |
| LabelId | String | 是 | 标签Id |
| ImgUrl | String | 是 | 标签图像url  示例值：http://ip:port/api/v1/imgs/20231129143030/1.jpg |
| CaptureTime | Integer | 是 | 采集时间，秒级Unix时间戳  示例值：1695197431 |

**（4）示例**

输入示例

|  |
| --- |
| {  "LabelId": “123” } |
|
|
|
|

输出示例

|  |
| --- |
| {  "ErrCode": 0,  "ErrMsg": "OK",  "Data": {  "LabelId": "123",  "ImgUrl": "http://ip:port/api/v1/imgs/20231129143030/123.jpg",  "CaptureTime": 1695197431  } } |
|
|
|
|

## 4. 状态定义

**（1）单批次的整体状态**

|  |  |
| --- | --- |
| **状态值** | **状态说明** |
| 1 | 待识别标签Id |
| 2 | 识别标签Id失败 |
| 3 | 待审核 |
| 4 | 已作废 |
| 5 | 待上传 |
| 6 | 上传完成 |

说明：

1）单批次存在未识别Id的标签，状态为“待识别标签Id”

2）单批次的标签Id识别完成，且存在识别失败的，状态为“识别标签Id失败”

3）单批次的标签Id识别完成，且都识别成功的（或者用户通过协议修改），状态为“待审核”

4）单批次审核通过的，状态为“待上传”，作废的，状态为“已作废”

5）单批次的标签信息都上传兆日服务器成功，状态为“上传完成”

特别说明：

1）状态 [1, 2, 3] 可以理解为待审核，为1和2时，审核通过和作废按钮置灰，为3时，按钮正常

2）状态 [5, 6] 可以理解为已审核，为5时，表示待注册，为6时，表示注册成功

**（2）单批次的单个标签状态**

|  |  |
| --- | --- |
| **状态值** | **状态说明** |
| 1 | 待识别标签Id |
| 2 | 识别标签Id失败 |
| 3 | 待上传 |
| 4 | 上传完成 |

说明：

1）单个标签未识别Id，状态为“待识别标签Id”

2）单个标签识别Id失败，状态为“识别标签Id失败”

3）单个标签识别Id成功，状态为“待上传”

4）单个标签上传成功，状态为“上传完成”

## 5. 数据库表设计

设计两张表，表一表示完成批量采集的批次信息（capture\_infos），表二表示批次对应的标签信息（label\_infos）。表一和表二是1:N关系。

表一：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段说明** | **字段类型/长度** | **主键/索引/外键** | **允许为空** | **备注** |
| 1 | ID | 自增主键 | INTTEGER | 自增主键 | NOT NULL |  |
| 2 | BATCH\_ID | 批次号 | VARCHAR (128) |  | NOT NULL |  |
| 3 | TOTAL\_COUNT | 总数目 | INTTEGER |  | NOT NULL |  |
| 4 | LABEL\_WIDTH | 标签图像宽 | INTTEGER |  |  |  |
| 5 | LABEL\_HEIGHT | 标签图像高 | INTTEGER |  |  |  |
| 6 | CAPTURE\_TIME | 采集时间 | INTTEGER |  |  |  |
| 7 | STATUS | 该批次的整体状态 | INTTEGER |  | NOT NULL | 见上一节“状态定义” |
| 8 | REVERSE | 保留 | VARCHAR (256) |  |  |  |

表二：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段说明** | **字段类型/长度** | **主键/索引/外键** | **允许为空** | **备注** |
| 1 | ID | 自增主键 | INTTEGER | 自增主键 | NOT NULL |  |
| 2 | BATCH\_ID | 批次号 | VARCHAR (128) |  | NOT NULL |  |
| 3 | INDEX | 序号 | INTTEGER |  | NOT NULL |  |
| 4 | STATE | 标签状态 | INTTEGER |  | NOT NULL |  |
| 5 | LABEL\_ID | 标签Id | VARCHAR (256) |  |  |  |
| 6 | ISSUED\_TIME | 录入时间 | INTTEGER |  |  |  |
| 7 | BATCH\_STATUS | 批次状态 | INTTEGER |  | NOT NULL |  |

## 6. 状态码

|  |  |  |
| --- | --- | --- |
| 状态码 | 英文名称 | 中文描述 |
| 200 | OK | 请求成功。一般用于GET与POST请求 |
| 400 | Bad Request | 客户端请求的语法错误，服务器无法理解 |
| 401 | Unauthorized | 请求要求用户的身份认证 |
| 402 | Payment Required | 保留，将来使用 |
| 403 | Forbidden | 服务器理解请求客户端的请求，但是拒绝执行此请求 |
| 404 | Not Found | 服务器无法根据客户端的请求找到资源（网页）。通过此代码，网站设计人员可设置"您所请求的资源无法找到"的个性页面 |
| 500 | Internal Server Error | 服务器内部错误，无法完成请求 |

## 7. 错误码

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误信息** | **描述** |
| 1/-1 | Unknown error | 服务器内部错误，请再次请求， 如果持续出现此类错误，请联系技术支持团队 |

# 五、SDK接口定义

// 图像类型定义

#define IMG\_TYPE\_PAPER 1 // 纸纹图

#define IMG\_TYPE\_HD 2 // 高清图

// 图像输出格式定义

#define IMG\_FORMAT\_JPEG 1

#define IMG\_FORMAT\_RAW 2

// 图像采集参数结构体

struct CapParam

{

// 图像类型

int imgType;

// 图像输出格式

int imgFormat;

};

// 图像输出方式相关参数

struct ImgRetModeParam

{

// 图像输出方式：1为落盘（此方式，filePath不能为空），2为返回buffer（此方式，outBuf不能为空）

int imgRetMode;

// 图像存储路径

char\* filePath;

// 图像输出buffer

unsigned char\* outBuf;

// 图像buffer长度

int outBufLen;

// 图像宽

int imgWidth;

// 图像高

int imgHeight;

};

// 识别二维码相关参数

struct IdentifyIdParam

{

// 是否需要识别二维码

bool needIdentifyId;

// 二维码内容

std::string id;

};

## 1. 开始批量采集

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称: StartBatchCapture

\* 功能描述: 开始批量采集

\* 输入参数: capParam - 图像采集参数，包括图像类型、图像输出格式

\* dir - 图像存储目录，例如"/home/sino/bh400/20231129144622"

\* firstLabelId - 首张标签Id

\* cb - 采集过程回调函数指针

\* 回调函数类型：void cb(int code, int num)

\* 回调函数参数：code - 处理结果（0: 成功；非0: 错误码）

\* num - 已采集标签张数

\* 输出参数: 无

\* 返回值 : 0 - 成功；非0 - 错误码

\* 其它说明:

\* 修改日期 版本号 修改人 修改内容

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int StartBatchCapture(CapParam& capParam, char\* dir, int firstLabelId, void(\*cb)(int, int));

说明：

图像存储路径为：dir + 标签Id + “.raw(jpg)”，例如”/home/sino/bh400/20231129144622/1.raw”

## 2. 结束批量采集

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称: StopBatchCapture

\* 功能描述: 结束批量采集

\* 输入参数: 无

\* 输出参数: 无

\* 返回值 : 0 - 成功；非0 - 错误码

\* 其它说明:

\* 修改日期 版本号 修改人 修改内容

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int StopBatchCapture();

## 3. 采集一张图像

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称: CaptureImg

\* 功能描述: 采集一张图像

\* 输入参数: capParam - 图像采集参数，包括图像类型、图像输出格式

\* retModeParam – 图像输出方式相关参数

\* idfIdParam – 识别二维码相关参数

\* downSample – 下采样系数

\* 输出参数: 无

\* 返回值 : 0 - 成功；非0 - 错误码

\* 其它说明:

\* 修改日期 版本号 修改人 修改内容

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int CaptureImg(CapParam& capParam, ImgRetModeParam& retModeParam, IdentifyIdParam& idfIdParam, float downSample = 1.0f)

## 4. 识别二维码

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称: DecodeLabelId

\* 功能描述: 识别标签Id

\* 输入参数: path - 图像路径

\* imgWidth – 图像宽

\* imgHeight – 图像高

\* imgType – 图像类型，1为raw，2为jpg

\* id – 输出的id

\* 输出参数: 无

\* 返回值 : 0 - 成功；非0 - 错误码

\* 其它说明:

\* 修改日期 版本号 修改人 修改内容

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int CBH400Device::DecodeLabelId(char\* path, int imgWidth, int imgHeight, int imgType, std::string& id);

## 5. raw图转jpg图

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称: Raw2Jpeg

\* 功能描述: raw转jpg

\* 输入参数: inRawPath - raw图路径

\* rawWidth – raw图宽

\* rawHeight – raw图高

\* outJpgPath – 输出的jpg图路径

\* downSample – 下采样系数

\* quality – 压缩系数

\* 输出参数: 无

\* 返回值 : 0 - 成功；非0 - 错误码

\* 其它说明:

\* 修改日期 版本号 修改人 修改内容

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int CBH400Device::Raw2Jpeg(char\* inRawPath, int rawWidth, int rawHeight, char\* outJpgPath, float downSample = 1.0f, float quality = 0.95f);