

QR 二维码信息解码器 (QRecorder) 使用说明书 (软件操作文档)

1. 引言

QR 二维码 (QR code)^[1]是由日本 Denso-Wave 公司 1994 年研制出的一种矩阵二维码符号。由于其信息存储量大、识别快速, 现已成为使用最为普遍的条码符号之一。21 世纪以来, QR 码在机械制造、物流运输、电子商务等领域得到了广泛地应用, 大大提高了各行业的生产效率。随着智能手机的普及, QR 码的应用也深入到了普通人的生活, 在移动支付、电子票务等场景中被广泛使用。

1.1 编写目的

在识读 QR 二维码时，用户经常有分段识读信息的需求。传统的 QR 二维码扫描器只能简单转译二维码所包含的字符串信息，并不能根据分隔符将各字段信息分段识读。为了解决这个问题，作者基于 Python3 开发了专门用于识读含分段信息的 QR 二维码，并基于用户自定义信息将其自动快速整理成电子表格文件存储。

1.2 项目背景

作者在一所高校从事采购和实验室管理工作，在实际工作中经常要统计汇总大量的财务报销记录。传统使用电子表格人工录入每笔报销单的信息（如项目号、报销人、报销金额等）费时费事，而且容易出错。经分析发现，学校财务管理系统在生成报销单时每张报销单上均生成一个 QR 二维码（图 1.a），扫码发现 QR 码被解译为一个长字符串，以“;”或“|”等字符分隔字段，各字段存储着该报销单各项关键信息，包括报销单号、报销人工号、手机号、部门号、项目号、报销金额等。因此，作者计划开发一个二维码读码和解码的软件，自动解译并分字段存储 QR 二维码信息。



a. 财务报销单上的二维码 b. 商品包装上的二维码 c. 宣传材料上的二维码

图 1 各类 QR 二维码、条形码应用示例

除此应用外, QR 二维码(条形码)的解码应用在各行业广泛需要(图 1.b, 图 1.c), 字符信息中分隔符样式、字段名称和格式限制要求也千差万别。为了使软件具有通用性, QRecorder 软件在编写时将源代码与读码配置文件分开放置。用户根据特定规则可编写适合自身读码要求的配置文件(以.ini 文件存储), 使该软件满足各应用场景的需要。

1.3 定义(专门术语的定义和缩写词的原意)

- **Zmodem 协议:** Zmodem 协议是一种面向字符的通讯控制规程, 可用于传送一个或一批文件。该协议于 1986 年在 Xmodem 协议、Ymodem 协议基础上产生, 是一种完善、实用的 modem 协议^[2-3]。
- **GTIN:** GTIN (Global Trade Item Number) 是国际编码系统中应用最广泛的标识代码, 是为全球贸易项目提供唯一标识的一种代码^[4]。它有四种不同的编码结构: GTIN-13、GTIN-14、GTIN-8 和 GTIN-12。
- **正则表达式:** 正则表达式 (Regular Expression) 是计算机科学的一个概念, 表示一种文本模式。正则表达式使用单个字符串来描述、匹配一系列匹配某个句法规则的字符串, 通常被用来检索、替换那些符合某个模式(规则)的文本。在本程序中, 正则表达式符合 Python 版本的正则表达式语法规则。
- **configparser 模块:** Python 程序的一个扩展包, 用于读取脚本文件中用户设定的一系列参数。
- **配置文件:** 用户编写的对二维码(条形码)信息进行解译的脚本文件, 语法规则符合 configparser 模块对脚本文件的语法规范, 以“ini”为文件扩展名。
- **数据文件:** 存储解译后二维码(条形码)信息的“csv”格式文件。程序运行时, 每成功读取一条二维码(条形码)信息后会在该文件中新增一条记录并进行增量存储。

2. 软件性能

2.1. 文本编码格式

本程序输入、输出数据均为 UTF-8 编码的文本类型文件。

2.2. 时间特性

本程序对输入数据、输出数据的处理时间由待读取二维码解码信息长度和运行软件主机性能决定。一般来讲读取并解译、存储单条记录信息的时间不超过 0.5 秒。

3. 运行环境

3.1 硬件

推荐配置：

CPU：主频 > 2.5 GHz (x86 架构)

5 内存：> 1 GB (DDR4)

硬盘：因数据量确定

网络带宽：> 5Mbps (峰值)

3.2 支持软件

操作系统：Linux (CentOS 7/8)

10 Windows

(如开启打印功能，操作系统须支持 Zmodem 通讯协议)

4. 使用说明

4.1 程序安装

QRecorder 为免安装的绿色软件，只包含一个程序文件：

15 QRecorder/QRecorder.exe (Linux 系统/Windows 系统)。在运行时，QRecorder 必须结合自定义的配置文件 (*.ini) 才能正常使用。配置文件存储目录须与主程序相同。

4.2 配置文件

20 QR 二维码 (条形码) 中的字符串被读取后，需进行解译后再存储到数据文件中。解译规则由配置文件中所存储的参数进行定义。配置文件的编写语法须符合 Python 语言 configparser 模块的语法规则。在本程序中：第一行以 “[DECODE]” 开头作为保留字；之后分行定义解译二维码字符串的 11 个参数 (见表 1)。

25 表 1 配置文件各参数语法表

序号	关键字	定义功能	赋值语法规则
1	name	配置文件名 (必填)	与 ini 文件名同名，字符型数据，必填项
2	delimiters	分隔符样式 (可空)	符合 Python 版本的正则表达式的匹配规范。用 [...] 表示匹配一组分隔符。值为字符型数据。
3	minIntervalTime	读码最小时间间隔 (可空)	二维码扫描器读码过程中有时出现同一条码重复扫描的情形。为了避免信息重复录入，可设定两次读码的最小时间间隔

			隔，单位为秒。值为数值型数据。
4	HEADER	字段名（必填）	定义由分隔符切分后的各字段的名称，字段名间用“,”分割。值为字符型数据。
5	replaceStrs	替代字符串（可选）	用于对读取的各字段数据进行二次编辑，将特定字符串替换为指定字符，便于后续应用的解译。替换表达式的语法为“匹配字符串”/”替换后的字符串”。如有多个字符串样式待替换，表达式间用“;”隔开。值为字符型数据。
6	BarcodeSites	条形码分割规则（可选）	除识别二维码外，程序还能对一维的条形码（Barcode）数据进行解译。条形码的字段划分不依靠分隔符，而通过字符位置定义。如“1-3 位”的字符代表“字段 1”，如“4-6 位”的字符代表“字段 2”...写作 1-3,4-6,...值为字符型数据。
7	note	备注字段名（可选）	除分割并定义各字段外，软件还允许自定义一个备注字段，用户可以在读取每一条字符后，手动录入该字段的值。值为字符型数据。
8	isPrint	打印字段名（可选）	在默认情况下，软件读取、解译每一条二维码信息并增量存储在一个 csv 文件中。如用户需要在读取、解译每一条信息后下载、打印该记录，则可定义打印字段名。定义规则同“HEADER”参数。但字段名须为“HEADER”中已定义的

			字段名的子集。值为字符型数据。
9	exception	异常值判断规则（可选）	软件在读取、解译后通过该参数定义各字段的异常值，并给出提示。含有异常值的记录不能存储在数据文件中。 exception 的定义须符合 python 的语法规范。
10	api_prefix	api 接口前缀（可选）	使用外部 api 接口对读取到的条形码信息指定字段进行进一步解译。api_prefeix 用于定义第三方 api 接口的前缀。
11	api_output	api 查询字段（可选）	第三方 api 接口待查询的字段用 api_output 参数进行定义。该字段的值必须是 HEADER 参数定义的其中一个字段。

附：

配置文件语法示例 1 (CNUapRecorder.ini)

[DECODE]

```

name = CNUapRecords
delimiters = [;|; \n]
minIntervalTime = 2
HEADER = TYPE,ID,STAFF,MOBILE,CNY,DEP,ITEM
replaceStrs = YY03/授权采购;03,yy03/授权采购;03,YY/授权采购,yy/授权采购
BarcodeSites =
note = COMPANY
isPrint = TYPE,ID,STAFF,MOBILE,CNY,DEP,ITEM,COMPANY
exception =(dictRec['ID'].startswith('032021') == False) or (len(dictRec['DEP'])
    not in [3,5]) or (len(dictRec['ITEM']) not in [7,9,11,13])
api_prefix =
api_output =

```

配置文件语法示例 2 (IDs.ini)

[DECODE]

```
name = IDs
delimiters =
minIntervalTime = 2
HEADER = IDs,PROVINCE,CITY,Birthday,Others
replaceStrs =
BarcodeSites = 1-18
note =
isPrint =
exception = (len(Barcode) != 18)
api_prefix =
api_output =
```

配置文件语法示例 3 (GTIN.ini)

[DECODE]

```
name = GTIN
delimiters =
minIntervalTime = 2
HEADER = GTIN
replaceStrs =
BarcodeSites =1-13
api_prefix =
https://way.jd.com/jisuapi/barcode2?appkey=a34611de6c6727de0b4840cddec5b2
b9&barcode=
api_output = GTIN
note =
isPrint =
exception = (len(Barcode) not in [13,14])
```

4.3 数据文件

- QRecorder 的输出文件为一个 csv 格式文件，用于增量存储每一条读取、解译后的二维码信息。该文件自动存储在与配置文件同名的文件夹中且该文件夹与 QRecorder 程序处在同一目录中。

为了方便用户排查二维码信息解析错误，程序在读取二维码字符串

后，会在数据文件同目录下自动生成一个“raws”文件夹，并对每一个运行过的配置文件每日生成一个日志文件，用于收集当天针对该配置文件读取到的每一条二维码原始信息。该日志文件的主要作用是：如因配置文件编写错误或程序异常等原因导致对二维码信息解译异常，可以将原始读取字符串和解译后的字符串进行比对，分析解译失败的原因。

如果用户在配置文件设置了记录打印的功能，程序还将在数据文件增量存储后针对每条二维码信息记录生成一个 txt 文件，记录中待打印字段列表由配置文件的 isPrint 参数指定。如果程序运行在服务器端，每个待打印的 txt 文件生成后通过 Zmodem 通讯协议自动传输到用户本地终端的 prtfiles 文件夹中。该文件夹与数据文件同目录。用户可以使用文档编辑器查看、打印 txt 信息，也可以撰写脚本自动打印 txt 文件。

4.4 出错和恢复

出错信息 1：“提示符光标不动，程序无响应”。

含意：程序运行异常。

措施：按 Ctrl+C 组合键强制退出程序或强制重启 Shell 管理器。

4.5 求助查询

在使用中有任何疑问或发现程序中的 bug，请通过电子邮件（zhanyj@cnu.edu.cn）与作者联系。

5. 运行说明

5.1 运行表

QRecorder <-i[ifile] xx.ini>

参数释义：

-i[ifile]：-i 配置文件名前的保留字符，可以使用“-i”或“--ifile”两种形式表示（必填项）。

xx.ini：配置文件名，“xx”为文件名，扩展名为“ini”（必填项）。

以上参数的使用须符合 Python configparser 模块的使用规范。

5.2 屏幕提示信息解析

（0）初始化

显示启动信息并确认运行参数。

QRecorder V1.0

①

Designed by Ph.D ZHAN YONGJIA, CAPITAL NORMAL UNIVERSITY.

②

Author: Yongjia Zhan (Capital Normal University) ③

E-mail: zhanyj@cnu.edu.cn ④

USAGE: QRecorder -i[--ifile] <inputfile.ini> ⑤

5 Press Ctrl+C or exit the program. ⑥

注释:

①【软件名(版本号)】②【功能简介】③【作者姓名、单位】④【E-mail】

10 ⑤【命令行运行参数表】⑥【运行异常处理方法】

(1) 读取并解析二维码信息

****Scanning QRcode_XXX:** ①

注释:

15 ① 输入二维码字符串信息。可以使用二维码扫描设备输入,也可以使用键盘输入且支持“复制/粘贴”功能。“XXX”为本次输入二维码的顺序号,从“001”开始。

(2) 录入备注字段值(可选)

****Input [note]:** ①

20 注释:

① 输入二维码字符串附加的备注字段值。“[note]”为配置文件中定义的备注字段名。

25 6. 用户操作举例

6.1 示例 1

读取、解译存储《财务报销单》上的二维码信息。字符串解译规则见上文配置文件语法示例 1 (CNUapRecorder.ini)

(1) 命令行:

30 **Qrecorder.exe -i CNUapRecorder.ini**

(2) 运行提示信息

QRecorder V1.0

¹ 灰色底纹的字符为程序运行时的命令行显示信息。

Designed by Ph.D ZHAN YONGJIA, CAPITAL NORMAL UNIVERSITY

Author: Yongjia Zhan

E-mail: zhanyj@cnu.edu.cn

USAGE: 'QRecorder -i[--ifile] <inputfile>

5 Press Ctrl+C or exit the program

The config file(CNUapRecorder.ini) is loaded successfully!

Appending to the existing file: CNUapRecords.csv

10 (Last modified Time: Sun Feb 27 16:03:47 2022)

**Scanning QRcode_001:YY;03202106100035;5507;13512345678;15600;126|2155181

**Input COMPANY :AAA 公司

15 {

"DATETIME": "2022-08-05 14:04:57",

"USER": "zhanyongjia",

"TYPE": "授权采购",

"ID": "03202106100035",

20 "STAFF": "5507",

"MOBILE": "13512345678",

"CNY": "15600",

"DEP": "126",

"ITEM": "2155181",

25 "COMPANY": "AAA 公司"

}

Waiting for printing (based on running the StartMonitoring.ps1).....

**Scanning QRcode_002:YY;03202106100035;5810;13987654321;7000;008|2255188

30 **Input COMPANY :BBB 公司

{

"DATETIME": "2022-08-05 14:38:47",

"USER": "zhanyongjia",

"TYPE": "授权采购",

35 "ID": "03202106100035",

```

"STAFF": "5810",
"MOBILE": "13987654321",
"CNY": "7000",
"DEP": "008",
5  "ITEM": "2255188",
   "COMPANY": "BBB 公司"
}
Waiting for printing (based on running the StartMonitoring.ps1).....
-----
10 **Scanning QRcode_003:YY;03202306100035;5810;13987654321;7000;008|2255188
   **Input COMPANY :CCC 公司
      Warning! The QRcode/Barcode is an exception!
(dictRec['ID'].startswith('032021') == False) or (len(dictRec['DEP']) not in [3,5]) or
(len(dictRec['ITEM']) not in [7,9,11,13]))
15  QRcode: 授权采购;03202306100035;5810;13987654321;7000;008|2255188
   Barcode: 授权采购;03202306100035;5810;13987654321;7000;008|2255188

.....

20 **Scanning QRcode_XXX:exit
   GoodBye!
-----

```

程序运行后，自动显示启动并确认配置文件中所列的运行参数加载正常。之后，程序会循环要求用户输入二维码信息及备注字段值（可选），并打印输出结果或显示解译错误警告，直至获取到“exit”指令后才会停止执行并退出。

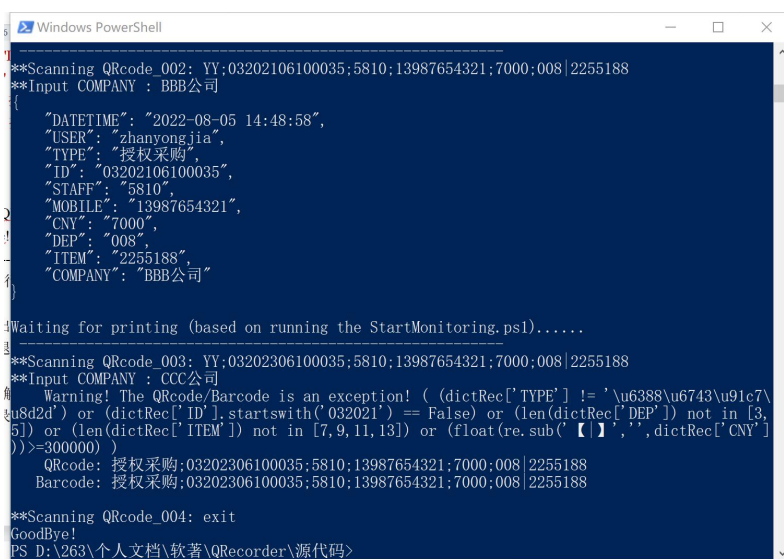
（3）结果记录解读和异常结果警告

二维码信息和备注字符串（可选）录入后，程序会以 JSON 字符串的形式将解译后的二维码信息或错误警告显示在运行终端（如 Winodws PowerShell, CMD, XShell 等）上（见图 1）。

解译结果记录包含三个部分：自动生成字段、二维码信息读取字段和备注字段。自动生成字段：程序在解译过程中会自动将二维码读取的时间信息和运行该程序的操作系统用户名二个字段添加到结果记录中，以方便用户排查错误和信息溯源。二维码信息读取字段：此部分为结果

记录的主体。在本示例中，配置文件的“**delimiters**”参数定义了“;”、“|”、“;”和“\n”共4个字符为分隔符，因此示例中的二维码字符串被分割符切分为7个部分，分别赋值给配置文件中“**HEADER**”参数定义的“**TYPE**”，“**ID**”，“**STAFF**”，“**MOBILE**”，“**CNY**”，“**DEP**”，“**ITEM**”的7个字段。需要注意的是，由于配置文件的“**replaceStrs**”参数的定义，字符串中的“**YY**”字符被替换为“授权采购”后赋值给“**TYPE**”字段。备注字段：由于配置文件的“**note**”参数定义了备注字段名为“**COMPANY**”，因此程序在获取二维码信息后，还要求用户输入备注字段的值。备注字段的值作为最后一个字段整合到结果记录中。

异常结果警告是对异常字段值的提示。在本示例中，配置文件的**exception**参数使用Python表达式语法规定了多个结果字段的合法性规则，如输入的二维码信息经解译后发现某一字段的值违反合法性规则，则在运行终端上显示“异常结果警告”，该记录也不会存储到结果数据文件中，以保证数据的存储质量。如在本示例中，定义了ID字段须以“032021”作为字符串的开头，而003号二维码的信息分割符切割后对该字段的赋值为“03202306100035”，违反该字段值的合法性，所以显示了异常结果警告。



```
Windows PowerShell
**Scanning QRcode_002: YY:03202106100035;5810;13987654321;7000;008|2255188
**Input COMPANY : BBB公司
{
  "DATETIME": "2022-08-05 14:48:58",
  "USER": "zhanyongjia",
  "TYPE": "授权采购",
  "ID": "03202106100035",
  "STAFF": "5810",
  "MOBILE": "13987654321",
  "CNY": "7000",
  "DEP": "008",
  "ITEM": "2255188",
  "COMPANY": "BBB公司"
}

Waiting for printing (based on running the StartMonitoring.ps1).....
**Scanning QRcode_003: YY:03202306100035;5810;13987654321;7000;008|2255188
**Input COMPANY : CCC公司
Warning! The QRcode/Barcode is an exception! ((dictRec['TYPE'] != '\u6388\u6743\u91c7\u82d2') or (dictRec['ID'].startswith('032021') == False) or (len(dictRec['DEP']) not in [3, 5]) or (len(dictRec['ITEM']) not in [7, 9, 11, 13]) or (float(re.sub('【】', '', dictRec['CNY'])) >= 300000))
QRcode: 授权采购:03202306100035;5810;13987654321;7000;008|2255188
Barcode: 授权采购:03202306100035;5810;13987654321;7000;008|2255188
**Scanning QRcode_004: exit
GoodBye!
PS D:\263\个人文档\软著\QRecorder\源代码>
```

图2 示例1运行截图

6.2 示例2

读取、解译存储身份证二维码信息。字符串解译规则见上文配置文件语法示例1 (IDs.csv)

(1) 命令行:

```
Qrecorder.exe -i IDs.ini
```

(2) 运行提示信息

QRecorder V1.0

Designed by Ph.D ZHAN YONGJIA, CAPITAL NORMAL UNIVERSITY

Author: Yongjia Zhan

5 E-mail: zhanyj@cnu.edu.cn

USAGE: ', 'QRecorder -i[--ifile] <inputfile>

Press Ctrl+C or exit the program

The config file(IDs.ini) is loaded successfully!

10 -----
Appending to the existing file: IDs.csv

(Last modified Time: Wed Aug 3 14:36:04 2022)

**Scanning QRcode_001:110106200001052419

15 {

"DATETIME": "2022-08-06 14:28:45",

"USER": "zhanyongjia",

"IDs": "110106200001052419",

"PROVINCE": "110",

20 "CITY": "106",

"Birthday": "20000105",

"Others": "2419",

""; ""

}

25 **Scanning QRcode_002:110108200105052110

{

"DATETIME": "2022-08-06 14:30:32",

"USER": "zhanyongjia",

"IDs": "110108200105052110",

30 "PROVINCE": "110",

"CITY": "108",

"Birthday": "20010505",

"Others": "2110",

""; ""

35 }

```
**Scanning QRcode_003:110101999
```

```
Warning! The QRcode/Barcode is an exception! ( (len(Barcode) != 18) )
```

```
QRcode: 110101999,110,101,999,110,,101,,
```

```
Barcode: 110101999
```

```
5 **Scanning QRcode_004:110101999
```

```
Warning! The current interval time (0s) between 004 and 005 is less than 2s
```

```
.....
```

```
**Scanning QRcode_XXX:exit
```

```
10 GoodBye!
```

```
-----
```

程序运行后，自动显示启动并确认配置文件中所列的运行参数加载正常。之后，程序会循环要求用户输入条形码信息并显示解译错误警告，直至获取到“exit”指令后才会停止执行并退出。在本示例中，输入的条形码信息是我国《居民身份证》号码。

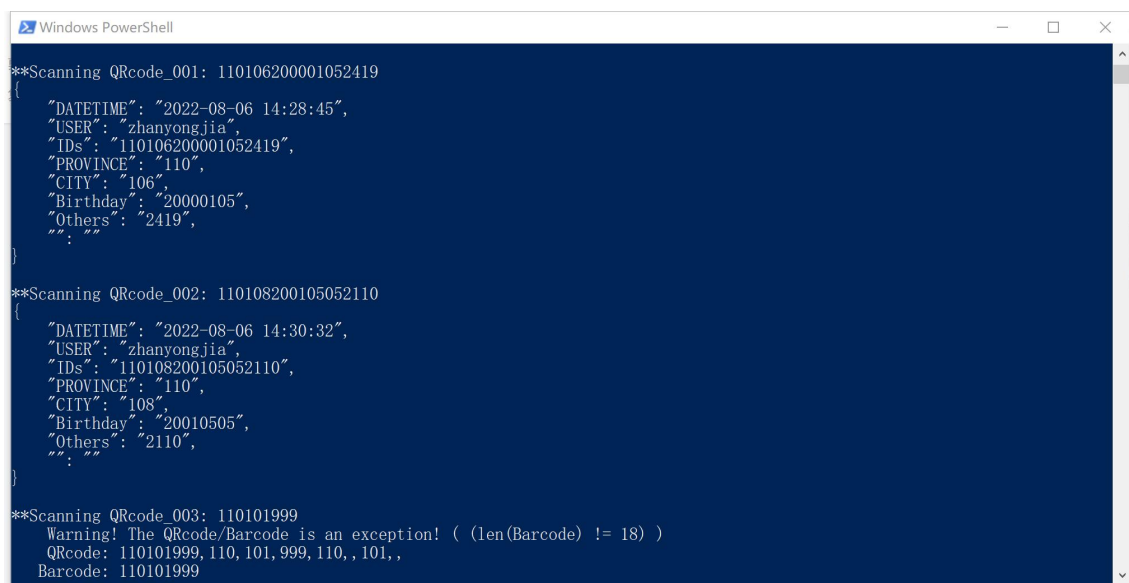
(3) 结果记录解读和异常结果警告

身份证号码信息读取后，程序会根据配置文件 (IDs.ini) 定义的规则进行解译并以 JSON 字符串的形式将结果记录或错误警告显示在运行终端上 (见图 2)。

解译结果记录包含三个部分：自动生成字段、条形码字段和配置文件定义的条形码字符串切分字段。自动生成字段：与示例 1 相同，程序自动将二维码读取的时间信息和运行该程序的操作系统用户名二个字段添加到结果记录中。条形码字段和切分字段：在示例 2 中，配置文件的“BarcodeSites”参数定义了条形码字段的切分规则，以字符串的“1-3 位”、“4-6 位”、“7-14 位”、“15-18 位”分别赋值给配置文件“HEADER”参数定义的除“IDs”字段外的“PROVINCE”，“CITY”，“Birthday”和“Others”字段。

异常结果警告。在示例 2 中，配置文件的 exception 参数规定了条形码信息的合法性规则为字符串长度须为 18，如输入的二维码信息经解译后发现某一字段的值违反合法性规则 (如：“110101999”的字符串长度仅为 9)，则运行终端上显示“异常结果警告”，该记录也不会存储到结果数据文件中。此外，示例 2 的配置文件的“minIntervalTime”参数定义了连续两次读码的最小时间间隔为 2s，则如果发生因条形码扫描设备故障对某一条码段段时间重复扫描时，只能够成功读取第 1 次扫描获取的信

息，第 2 次读码信息不能存储且显示异常警告。



```

Windows PowerShell

**Scanning QRcode_001: 110106200001052419
{
  "DATETIME": "2022-08-06 14:28:45",
  "USER": "zhanyongjia",
  "IDs": "110106200001052419",
  "PROVINCE": "110",
  "CITY": "106",
  "Birthday": "20000105",
  "Others": "2419",
  "": ""
}

**Scanning QRcode_002: 110108200105052110
{
  "DATETIME": "2022-08-06 14:30:32",
  "USER": "zhanyongjia",
  "IDs": "110108200105052110",
  "PROVINCE": "110",
  "CITY": "108",
  "Birthday": "20010505",
  "Others": "2110",
  "": ""
}

**Scanning QRcode_003: 110101999
Warning! The QRcode/Barcode is an exception! ( (len(Barcode) != 18) )
QRcode: 110101999, 110, 101, 999, 110,, 101,,
Barcode: 110101999
  
```

图 3 示例 2 运行截图

6.3 示例 3

本示例可读取各类商品包装上 GTIN 格式的条码信息，并通过第三方 API 程序接口进一步解译商品条码所代表的商品信息。字符串解译规则见上文配置文件语法示例 3 (GTIN.ini)。

(1) 命令行:

```
QRecorder.exe -i GTIN.ini
```

(2) 运行提示信息

```

-----
QRecorder V1.0
Designed by Ph.D ZHAN YONGJIA, CAPITAL NORMAL UNIVERSITY
Author: Yongjia Zhan
E-mail: zhanyj@cnu.edu.cn
USAGE: 'QRecorder -i[--ifile] <inputfile>'
Press Ctrl+C or exit the program
-----
The config file(GTIN.ini) is loaded successfully!
-----
Appending to the existing file: GTIN.csv
(Last modified Time: Sun Feb 27 15:57:46 2022)
-----

**Scanning QRcode_001: 6972663245859
  
```

```
{
  "DATETIME": "2022-08-07 13:34:00",
  "USER": "zhanyongjia",
  "GTIN": "6972663245859",
  "ApiName": "",
  "ApiValue": "{ 'code': '10000', 'charge': False, 'remain': 0, 'msg': '查询成功', 'result': { 'status': '210', 'msg': '没有信息', 'result': '' }, 'requestId': '93f5bee83d3b4395a13145b397da1ec3' }"
}
```

10 **Scanning QRcode_002: 6901721496230

```
{
  "DATETIME": "2022-08-07 13:36:19",
  "USER": "zhanyongjia",
  "GTIN": "6901721496230",
  "ApiName": "",
  "ApiValue": "{ 'code': '10000', 'charge': False, 'remain': 0, 'msg': '查询成功', 'result': { 'status': 0, 'msg': 'ok', 'result': { 'barcode': '6901721496230', 'name': '麦斯威尔三合一速溶咖啡 特浓 单条', 'unspsc': '10000115(速溶咖啡)', 'brand': '麦斯威尔', 'type': '13g', 'origincountry': '中国大陆', 'assemblycountry': '中国大陆', 'description': '', 'keyword': 'MaxwellHouse 3in1 特浓 1s', 'pic': 'http://api.jisuapi.com/barcode2/upload/202110/18204339_26026.png', 'netcontent': '13g', 'company': '臻饮贸易(上海)有限公司' } }, 'requestId': '594d1d8a9e544288bef7599a7a439954' }"
}
```

25 **Scanning QRcode_003: 12345678

```
{
  "DATETIME": "2022-08-07 13:36:43",
  "USER": "zhanyongjia", "GTIN": "12345678",
  "ApiName": ""
}
Warning! The QRcode/Barcode is an exception! ( (len(Barcode) not in [13,14]))
QRcode: 12345678,12345678,,
Barcode: 12345678
```

35

.....

****Scanning QRcode _XXX:exit**

GoodBye!

5

程序运行后，自动显示启动并确认配置文件中所列的运行参数加载正常。之后，程序会循环要求用户输入条形码信息并显示解译信息或错误警告，直至获取到“exit”指令后才会停止执行并退出。在示例 3 中，配置文件设置的合法条码是打印在商品包装上符合 GTIN 编码格式的商品条形码（图 3）。

10

（3）结果记录解读和异常结果警告

GTIN 格式的商品条码信息读取后，程序会根据配置文件（GTIN.ini）定义的规则进行解译并以 JSON 字符串的形式将结果记录或错误警告显示在运行终端上（见图 4）。

15

解译结果记录包含三个部分：自动生成字段、条形码字段和第三方 API 接口返回的该条码商品信息字段。自动生成字段：与示例 1、2 相同，程序自动将二维码读取的时间信息和运行该程序的操作系统用户名二个字段添加到结果记录中。条形码字段：返回配置文件 HEADER 参数所规定的字段名。在示例 3 中，只有“GTIN”一个字段，用于存储扫描获取的商品 GTIN 商品条形码的数值。第三方 API 接口返回的该条码商品信息字段：示例 3 配置文件中 api_prefix 参数和 api_output 参数定义了条码读取后使用第三方 API 接口查询的 URL 地址和输出字符串的显示格式。在本示例中，第 1 次扫描的条码“6972663245859”调用第三方 API 接口成功但是没有找到该商品的信息。第 2 次扫描的条码“6901721496230”调用第三方 API 接口成功且成功返回了商品的各项信息（为某品牌的一款速溶咖啡）”。

20

25

异常结果警告。在示例 3 中，配置文件的 exception 参数规定了 GTIN 条形码信息的合法性规则为字符串长度须为 13 位或 14 位，如输入的二维码信息经解译后发现某一字段的值违反合法性规则（如第 3 次扫描获取的“12345678”的字符串长度仅为 8），则运行终端上显示“异常结果警告”，该记录也不会存储到结果数据文件中。

30



图 3 扫描到示例 3 中的 GTIN 商品条形码

```
Windows PowerShell
"GTIN-13": "6972663245859",
"ApiName": "",
"ApiValue": "{ 'code': '10000', 'charge': False, 'remain': 0, 'msg': '查询成功', 'result': { 'status': '210', 'msg': '没有信息', 'result': '' }, 'requestId': '93f5bee83d3b4395a13145b397dalec3' }"

**Scanning QRcode_002: 6901721496230
[ 'DATETIME': '2022-08-07 13:36:19', 'USER': 'zhanyongjia', 'GTIN-13': '6901721496230', 'ApiName': '' ]

"DATETIME": "2022-08-07 13:36:19",
"USER": "zhanyongjia",
"GTIN-13": "6901721496230",
"ApiName": "",
"ApiValue": "{ 'code': '10000', 'charge': False, 'remain': 0, 'msg': '查询成功', 'result': { 'status': 0, 'msg': 'ok', 'result': { 'barcode': '6901721496230', 'name': '麦斯威尔三合一速溶咖啡 特浓 单条', 'unspse': '10000115(速溶咖啡)', 'brand': '麦斯威尔', 'type': '13g', 'origincountry': '中国大陆', 'assemblycountry': '中国大陆', 'description': '', 'keyword': 'Maxwell House 3in1 特浓 1s', 'pic': 'http://api.jisuapi.com/barcode2/upload/202110/18204339_26026.png', 'netcontent': '13g', 'company': '臻饮贸易(上海)有限公司' }, 'requestId': '594d1d8a9e544288bef7599a7a439954' } }"

**Scanning QRcode_003: 12345678
[ 'DATETIME': '2022-08-07 13:36:43', 'USER': 'zhanyongjia', 'GTIN-13': '12345678', 'ApiName': '' ]
Warning: The QRcode/Barcode is an exception! ( len(Barcode) not in [13,14] )
QRcode: 12345678, 12345678,
Barcode: 12345678

**Scanning QRcode_004:
```

图 4 示例 3 运行截图

参考文献

- 5 [1]31 ISO/IEC-JTC-1/SC. Automatic identification and data capture techniques — QR Code bar code symbology specification: ISO/IEC 18004:2015. [S], 2015.
- [2]吴时霖,周奕. Zmodem 协议的 Petri 网描述验证及其随机性能评价[J]. 计算机工程, 1994, (2): 33-40.
- 10 [3]孙金秋,怀红旗. ZMODEM 协议在文件传输中的应用与实现[J]. 计算机技术与发展, 2008, (5): 241-244.

[4]中国物品编码中心. 全球贸易项目代码 (GTIN) [EB/OL].
<http://ancc.org.cn/Knowledge/BarcodeArticle.aspx?id=183>.