УДК: 004.932.2

ИСПОЛЬЗОВАНИЕ ANDROID УСТРОЙСТВА ВМЕСТО СКАНЕРА ШТРИХ-КОДОВ В СИСТЕМАХ УЧЕТА

Кириченко Ю. В.

Киевский политехнический институт, Украина, Киев

Целью данной работы было проектирование и реализация системы сканирования штрих-кода с помощью устройства под управлением ОС Android. В результате проведенной работы были описаны основные требования к данной системе и особенности ее реализации. Также написан программный код на языке программирования Java, при исполнении которого осуществляется распознание штрих-кода и передача его на сервер. Протестированы возможности такой системы. Результаты данной работы могут быть использованы в системах учета, где необходимо преобразовывать штрих-код в цифровой вид и использовать его для хранения вместе с другой информацией. Также были рассмотрены недостатки подобной системы и сформированы цели для дальнейшего исследования этой темы.

Ключевые слова: штрих-код, система учета, сканирование штрих-кода, программный сканер штрих-кода, удешевление учета.

Кириченко Ю. В. Використання Android пристрою замість сканера штрих-кодів в системах обліку / Київський політехнічний інститут, Україна, Київ

Метою даної роботи було проектування і реалізація системи сканування штрих-коду за допомогою пристрою під керуванням ОС Android. В результаті проведеної роботи були описані основні вимоги до даної системи і особливості її реалізації. Також написаний програмний код на мові програмування Java, при виконанні якого здійснюється розпізнавання штрих-коду і передача його на сервер. Протестовані можливості такої

системи. Результати даної роботи можуть бути використані в системах обліку, де необхідно перетворювати штрих-код у цифровий вигляд і використовувати його для зберігання разом з іншою інформацією. Також були розглянуті недоліки подібної системи і сформовані цілі для подальшого дослідження цієї теми.

Ключові слова: штрих-код, система обліку, сканування штрих-коду, програмний сканер штрих-коду, здешевлення обліку.

Kirichenko Y. V. Using Android device instead of barcode scanner / Kyiv Polytechnic Institute, Ukraine, Kyiv

The aim of this work was to design and implement the barcode scanning system on an Android device. As a result of the work described basic requirements for the system and features of its implementation. Also, this article contains the code written in the programming language Java, which is responsible for the barcode recognizing and transferring it to the server. Capabilities of the system were tested. The results of this study can be used in accounting systems, which must convert barcode in digital form and store it together with other information. Also drawbacks of such system discussed and formed targets for further studying.

Keywords: barcode, accounting system, scanning barcode, software barcode scanner, cheaper accounting.

Введение. Общей проблемой в системах учета товаров является потребность во вводе штрих-кода в базу данных. Обычно, ввод осуществляется вручную или с использованием аппаратных устройств. Но в первом случае это влечет за собой большие затраты времени (средняя скорость набора – 4 симв/сек) и необходимость постоянно носить с собой клавиатуру, а во втором – дополнительные инвестиции (сканер, в среднем, стоит больше тысячи гривен).

На сегодняшний день, в связи с повсеместным использованием мобильных устройств под управлением ОС Android (по статистике от Gartner Inc на 2015 год число пользователей устройств на базе Android превысило 1.4 млрд) [1] и наличием как встроенных инструментов, так и внешних бесплатных библиотек для сканирования штрих-кодов, эти проблемы могут быть решены заменой аппаратных сканеров штрих-кодов в системах складского учета на приложение под Android. Это позволит существенно сэкономить на закупках и обслуживании сканера штрих-кодов. Вместо них будет использоваться приложение, которое через камеру устройства сможет распознавать штрих-код. То есть, вместо покупки специализированых устройств, любой смартфон, планшет или другое мобильное устройство под управлением ОС Android может быть использовано как сканер штрих-кодов. Так же, поскольку приложения под Android пишутся на Java, это позволяет разработать одно приложение вместо нескольких для разных устройств, что безусловно позитивно скажется на затратах на разработку и в общем снизит цену проекта.

Задачей данной статьи является создание системы, которая заменит аппаратный сканер штрих-кода.

Система будет состоять из приложения-сервера на стационарном компьютере и клиент-приложения на Android устройстве. Приложение на Android, в свою очередь, будет состоять из двух частей: первая часть сканирует штрих-код и передает второй, которая отправляет штрих-код на сервер.

Для реализации этой системы будет использоваться библиотека Zxing. Zxing – это Android приложение с открытым исходным кодом, которое позволяет пользователям сканировать 1-D или 2-D "графические штрих-коды" с помощью камеры на Android устройстве. Программа декодирует зашифрованные в штрих-коде данные, преобразуя их в код.

Планируемое приложение будет иметь такой принцип работы: приложение вызывает Zxing, Zxing сканирует и декодирует штрих-код, возвращая данные приложению, которое передает раскодирование данные на сервер через Java Socket по Wi-Fi, приложение-сервер выводит штрих-код в консоль.

Для лучшего понимания работы системы ознакомимся с Zxing. Zxing делится на 9 модулей: core – базовая библиотека распознания изображения, javase – специализированный клиент для JavaSE, android – клиент сканера для Android, androidtest – тестовое приложение и тесты классов, android-integration – поддержка интеграции сканера штрих-кодов через Intent, android-core – базовые классы (для android, androidtest, glass), glass – пример простого приложения для Google Glass, zxingorg – исходные коды, используемые для организации работы в zxing.org, zxing.appspot.com – исходные коды, используемые для организации работы веб-генератора штрих-кодов в zxing.appspot.com[2].

Для нашего приложения нужен только модуль android-integration. Классы, которые будут использованы: IntentIntegrator – вспомогательный класс, который позволяет с легкостью интегрировать сканер штрих-кода с помощью Intent, IntentResult – инкапсулирует результат сканирования штрих-кода, вызывается с помощью IntentIntegrator.

Так как в программе будет использоваться java Socket, рассмотрим его устройство и принцип работы. Socket является конечным звеном двусторонней связи между двумя программами, выполняемыми в сети. Socket связан с номером порта, так что TCP слой может идентифицировать приложение, которому были предназначены данные. Каждое соединение TCP может быть однозначно определено ее двумя конечными точками. Конечная точка представляет собой комбинацию IP-адреса и номера порта.

Таким образом, становится возможно создать несколько соединений между клиентом и сервером. Пакет java.net в платформе Java предоставляет класс Socket, который реализует одну сторону двустороннего соединения между программой Java и другой программой в сети. [3]

Так же учитывая использование ОС Android в системе, рассмотрим класс android.app.Activity. Activity является единым классом, сосредоточенным на том, что пользователь может сделать. Почти все виды Activity взаимодействуют с пользователем, поэтому класс Activity отвечает за создание окна, в котором может быть размещен пользовательский интерфейс с использованием setContentView (View). Зачастую Activity взаимодействуют с пользователем через полноэкранные окна, но они так же могут быть представлены в виде всплывающих окон (с помощью темы с windowIsFloating набором) или же быть внедренными во внутрь другого Activity (с использованием ActivityGroup).

Есть два метода, которые используются почти всеми Activity:

- onCreate(Bundle) – инициализация Activity, является наиболее важным методом. В нем обычно вызывается setContentView(int) с ресурсами разметки определенными в пользовательском интерфейсе и используется findViewById(int) для извлечения пользовательского интерфейса, с которым необходимо программно взаимодействовать.
- onPause() – в нем происходит обработка покидания пользователем Activity. Самое главное, что любые изменения, сделанные пользователем, должны быть сохранены в этот момент. [4]

Рассмотрим программную реализацию считывания и декодирования штрих-кода.

Листинг 1

```
public class BarActivity extends ActionBarActivity implements
View.OnClickListener
{
```

```
Button scan;
    TextView scanResult;
    EditText ipEdit;
    @Override
    protected void onCreate(Bundle savedInstanceState)
 {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bar);
        scan=(Button)findViewById(R.id.ScanButton);
        scan.setOnClickListener(this);
        scanResult=(TextView)findViewById(R.id.textView);
        ipEdit=(EditText)findViewById(R.id.editText);
    }
    @Override
    public void onClick(View v)
    {
        if(v.getId()==scan.getId()){
         IntentIntegrator scan=new IntentIntegrator(this);
         scan.initiateScan();
        }
    }
    @Override
    public void onActivityResult(int reqcod,int rescode,Intent intent)
    {
      IntentResult
scanres=IntentIntegrator.parseActivityResult(reqcod,rescode,intent);
        if(scanres!=null){
            scanResult.setText(scanres.getContents());
```

```
            TCPClient a=new
TCPClient(scanres.getContents(),ipEdit.getText().toString(),Integer.valueOf(getSt
ring(R.string.Port)));
            a.start();
        }
    }
```

В представленном классе, после создания Activity, происходит обнаружение всех нужных элементов графического интерфейса и ссылка на них присваивается полям класса, также в кнопке "scan" объект Listener заменяется на экземпляр этого класса (это возможно благодаря реализации интерфейса View.OnClickListener. View.OnClickListener – интерфейс, который используется для обработки нажатия на View [5]). При нажатии на кнопку, через IntentIntegrator, вызывается сканер штрих-кода, который оповещает об окончании сканирования и возвращает результат сканирования программе, вызывая onActivityResult. onActivityResult отвечает за получение результатов распознавания штрих-кода (как только будет найден и распознан код, управление возвращается этому приложению). Дальше из полученных данных выделяются нужные и результат отправляется через TCPClient на сервер.

В листинге 2 показан код класса TCPClient. Он использует java Socket для коммуникации с сервером-приложением.

Листинг 2

```
public class TCPClient extends Thread
{
   private String data;
   private static Socket servercon=null;
   private static DataOutputStream serverwriter=null;
   private String adress;
   private int port;
```

```
    public TCPClient(String data,String adress,int port){
        this.data=data;
        this.adress=adress;
        this.port=port;
    }
    public void run()
    {
        try {
            if(servercon==null) {
                servercon = new Socket(adress, port);
                servercon.setSoTimeout(100);
            }
            if(serverwriter==null)serverwriter=new
DataOutputStream(servercon.getOutputStream());
            serverwriter.writeUTF(data);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public static void close() throws IOException
    {
        if(servercon!=null){
            servercon.close();
        }
    }
}
```

Этот класс наследует Thread и рассчитан на выполнение в отдельном потоке, для того чтобы избежать ошибок выполнения при роботе с Socket.

При запуске устанавливается связь с сервером и выполняется отправка штрих-кода.

В листинге 3 на сервере выполняется получение штрих-кода и его вывод в консоль (вывод в консоль осуществляется в качестве примера использования полученного штрих-кода).

Листинг 3

```
public class Server
{
	private boolean alive=false;
	private int port;
	private ServerSocket serv;
	private Socket clientsocet;
	public Server(int port)
	{
		this.port=port;
	}
	public void start() throws IOException{
	alive=true;
	serv=new ServerSocket(port);
		while(alive){
			clientsocet = serv.accept();
			DataInputStream clientstream =new
DataInputStream(clientsocet.getInputStream());
			String result = clientstream.readUTF();
			System.out.println(result);
		}
	}
	public void stop(){
		alive=false;
```

```
}
}
```

При вызове метода start создается серверный Socket, который отвечает за роботу с клиентским Socket-ом. Вызов метода accept класса ServerSocket приостанавливает выполнение потока до подключения клиента. После того как клиентский Socket подключился, создается DataInputStream, который инкапсулирует в себе функции для удобной работы с потоковыми данными. Далее методом readUTF() класса DataInputStream осуществляется считывание штрих-кода, который после выводится в консоль.

Полученный в результате штрих-код может быть добавлен в базу данных к данным о товаре или просто внесен в какой-нибудь список. Также его можно проанализировать и получить дополнительную информацию, зашифрованную в нем (например, префикс национальной организации, регистрационный номер производителя товара и т.д.).

Выводы. В результате мы получили систему, распознающую штрих-код с помощью устройства под управлением Android и передающую его на сервер. Скорость распознания по результатам тестирования в среднем незначительно ниже аппаратных аналогов. Разработанная система имеет следующие недостатки: возможность обрыва Socket соединения, что приведет к некорректному функционированию приложения, отсутствие возможности автоматического поиска сервера в сети (требуется ввод адреса) и невозможность буферизирования штрих-кодов при отключении от сети для дальнейшей их отправки. Будущая работа над этой темой может быть направлена на устранение выше изложенных недостатков, а так же над расширениями системы до полноценного инструмента складского учета.

Література:

1. Janessa R. Gartner Says Tablet Sales Continue to Be Slow in 2015: Tablet Sales to Reach 8 Percent Growth in 2015 While PC Market to Grow 1 Percent [Electronic resourse]/Janessa Rivera // research lab. - Electronic data . –

[United Kingdom : information technology research and advisory company,2015] - Mode of access: World Wide Web: http://www.gartner.com/newsroom/id/2954317 (viewed on May 11, 2016). – Title from the screen.

2. "Zebra Crossing" [Electronic resourse] // GitHub, Inc.. – 2016. – Mode of access: World Wide Web: https://github.com/zxing/zxing.

3. What Is a Socket? [Electronic resourse] // Oracle. Java, Documentation. – 2015. – Mode of access: World Wide Web: http://docs.oracle.com/javase/tutorial/networking/sockets/definition.html.

4. Activity [Electronic resourse] // Developer.android.com. – 2016. – Mode of access: World Wide Web: http://developer.android.com/intl/ru/reference/android/app/Activity.html.

5. View.OnClickListener [Electronic resourse] // Developer.android.com. – 2016. – Mode of access: World Wide Web: http://developer.android.com/intl/ru/reference/android/view/View.OnClickListener.html.

References:

1. Janessa R. Gartner Says Tablet Sales Continue to Be Slow in 2015: Tablet Sales to Reach 8 Percent Growth in 2015 While PC Market to Grow 1 Percent [Electronic resourse]/Janessa Rivera // research lab. - Electronic data. – [United Kingdom: information technology research and advisory company,2015] - Mode of access: World Wide Web: http://www.gartner.com/newsroom/id/2954317 (viewed on May 11, 2016). – Title from the screen.

2. "Zebra Crossing" [Electronic resourse] // GitHub, Inc.. – 2016. – Mode of access: World Wide Web: https://github.com/zxing/zxing.

3. What Is a Socket? [Electronic resourse] // Oracle. Java, Documentation. – 2015.–Mode of access: World Wide Web: http://docs.oracle.com/javase/tutorial/networking/sockets/definition.html.

4. Activity [Electronic resourse] // Developer.android.com. – 2016. – Mode of access: World Wide Web: http://developer.android.com/intl/ru/reference/android/app/Activity.html.

5. View.OnClickListener [Electronic resourse] // Developer.android.com. – 2016.– Mode of access: World Wide Web: http://developer.android.com/intl/ru/reference/android/view/View.OnClickListener.html.