Institute of Technology, Carlow

B.Sc. Hons. In Software Engineering

CW208

Research Manual

Project Title:	Student ID Card Barcode Recognition For Android Mobile Phone
Name:	Long Long
Login ID:	C00131028
Supervisor:	Christophe Meudec
Date:	1 December 2009

Table of Contents

1. Abstract

I’m writing this research report to introduce the works of mobile barcode recognition I’ve done so far. In this report, several parts are set to make it a clear expression, they are: introduction, product aim, barcode knowledge, feasibility on mobile and conclusion.

The part Introduction covered the definition of barcode and barcode history, some common usages of barcode and a brief introduction of mobile barcode recognition.

The part Product aim is a description of the project goal and the general function of the product is described.

In the part Barcode knowledge, different barcode systems, barcode structure, barcode reading principle, encoding rules and decoding scheme were covered.

The part Feasibility on mobile is a research of if it is feasible to develop and run the product software on Android mobiles (GPhone), including hardware support, Android OS and development kit.

Finally, the part Conclusion is an overview. A special mention of a better recognition process and the current student ID card used around CarlowIT was coverd.

2. Introduction

2.1 Barcode & its History

“A **barcode** is an optical machine-readable representation of data, which shows certain datas on certain products. Originally, barcodes represented data in the widths (lines) and the spacings of parallel lines, and may be referred to as linear or 1D (1 dimensional) barcodes or symbologies. They also come in patterns of squares, dots, hexagons and other geometric patterns within images termed 2D (2 dimensional) matrix codes or symbologies. Although 2D systems use symbols other than bars, they are generally referred to as barcodes as well. Barcodes can be read by optical scanners called barcode readers, or scanned from an image by special software.

In 1948 Bernard Silver (1924–63), a graduate student at Drexel Institute of Technology in Philadelphia, overheard the president of a local food chain asking one of the deans to research a system to automatically read product information during checkout. Silver told his friend Norman Joseph Woodland (1921-) about the request, and they started working on a variety of systems. Their first working system used ultraviolet ink, but this proved to fade and was fairly expensive.

Convinced that the system was workable with further development, Woodland quit his position at Drexel, moved into his father's apartment in Florida, and continued working on the system. His next inspiration came from Morse code, and he formed his first barcode from sand on the beach when "I just extended the dots and dashes downwards and made narrow lines and wide lines out of them." To read them, he adapted technology from optical soundtracks in movies, using a 500-watt light bulb shining through the paper onto an RCA935 photomultiplier tube (from a movie projector) on the far side. He later decided that the system would work better if it were printed as a circle instead of a line, allowing it to be scanned in any direction.

On 20 October 1949 Woodland and Silver filed a patent application for "Classifying Apparatus and Method", in which they described both the linear and bullseye printing patterns, as well as the mechanical and electronic systems needed to read the code. The patent was issued on 7 October 1952 as US Patent 2,612,994. In 1951, Woodland moved to IBM and continually tried to interest IBM in developing the system. The company eventually commissioned a report on the idea, which concluded that it was both feasible and interesting, but that processing the resulting information would require equipment that was some time off in the future.

In 1952 Philco purchased their patent, and then sold it to RCA the same year. In 1963 Silver died in a car accident.”[1]

Table t1 involves the information of some common used barcodes ordered by year.

Year	Barcode Name	Inventor or Co.	Special Meaning
1949	Bull's Eye Code	N. Joe Woodland, Bernard Silver	The first barcode
1972	Codabar	Monarch Marking System	
1973	UPC	IBM	The first extensive used barcode
1974	Code 39	David C. Allias (Intermec)	First commercial used alpha-numeric barcode
1976	EAN	EAN Association	
1981	Code 128		
1983	Code 93		

Table t1: some common used barcodes[2]

2.2 Usages of barcode

2.2.1 Sales

Portable POS system

"Description:

> The Portable POS system can read a bar code for sales goods to display merchandise information such as the product name and sales price. Combined with a pocket printer, the system can also issue a receipt.

Advantages:

- The portable POS system allows speedy sales operation.
- As it is portable, POS can be realized in a vehicle, train or temporary shop.
- Sales performance data is processed at a host for sales analysis."

Merchandise replenishment system

"Description:

> Have the merchandise replenishment system read the bar code for the merchandise to be replenished and enter an ordering quantity. Send the accumulated ordering data to the host to complete order placement.

Advantages:

- The merchandise replenishment system allows detailed merchandise inventory control.
- Merchandise can be accurately and quickly replenished."

2.2.2 Logistics

Warehouse shipping/receiving control system

“Description:

The warehouse shipping/receiving control system can read bar codes attached to shipping/receiving products to perform checking on the spot.

Advantages:

- The warehouse shipping/receiving control system allows real time delivery status control.
- Delivery time can be accurately managed.”

Package tracking control system

“Description:

The package tracking control system can read bar codes of package slips to control delivery status. Data from the delivery destination can even be sent via PHS data communication.

Advantages:

- The package tracking control system allows real time delivery status control.
- Delivery time can be accurately managed.”

2.2.3 Manufacturing

“**Operation process control system**

Description:

The operation process control system reads bar-coded worker numbers, part numbers, process numbers, etc. before and after operation.

Advantages:

- The operation process control system facilitates progress supervision, which leads to quality improvement.
- Operation hours per process can be analyzed to achieve optimal line allocation.”

All information about the usages of barcode is referenced to Ref.[3]

2.3 Mobile barcode recognition

A description of mobile barcode recognition by *Justin Montgomery*:

“Almost every item available in stores has a bar code for tracking, and sales purposes. A new idea involving mobile bar code technology could give the consumer instant reviews about any product they're thinking of purchasing in real-time while holding the item in the store aisle.

The idea is that the consumer can simply take a picture of the bar code on the product, and the phone would automatically identify the product via an integrated java program, and search the internet for related blog posts, news reviews, comparisons, etc. and report them back to the consumer in real-time to read up on the product and make sure it’s really the right one. Consumers can see what other people who have bought the same product really thought about it, and what problems may have occurred. Think about doing an instant price check on the product to see if it might be cheaper somewhere else, or maybe cheaper to purchase online. This could prove to be a valuable asset to price, and safety conscious people.”[4]

Justin’s idea is that take a picture of barcode and save it, and then decode the picture of barcode and identify the product. This is one of the methods of process for barcode recognition. While the process of another could be like: user runs the barcode recognition software on the phone; the software active the webcam of the phone; user scan the barcode of products (don’t need to take photos); the software analyses the pixels captured from the webcam, decoding them and work out the binary code for the barcode and show it out to user. It’s an improved method for no need extra memory for picture storing. Well, we can also alter the process as: take picture; work out binary code; then delete the picture automatically when complete. But in the viewpoint of mine, the first method is better because it’s faster for all information of pixels storing in Cache rather than in SD card or phone memory.

3. Product Aims

As Wang, Jian said in his report *“Camera-based barcode recognition”*
, “use of barcodes is becoming more widespread, including use of barcodes for identifying many different types of things, including, but not limited to commercial goods, such as groceries, product packages of various types, printed reading material.”[5] The mobile barcode recognition became more and more popular, such as for commercial goods and etc.

The aim of mine is to develop a mobile barcode recognition for student ID card used in CarlowIT. If we pay a brief attention on our student ID card, we may find the barcode. It provides a unique graphic identifier to encode a set of digits or characters for machine, and also provides me an idea about my project.

The idea is that we simply run the product software on the Android mobile phone, and the software will active the webcam of the phone. When we focus the webcam to the barcode on student ID card, the program gets the pixels and analyses them, (don’t need to take photos), then figures out the binary code for the current barcode. For getting the detail information of the student who owns the current student ID card, the program transfers the recognized binary code to the college web server and gets the relevant student information back to the program and displays it onto the screen.

4. Barcode knowledge

4.1 Different barcode systems

EAN-13

“**EAN-13** (European Article Number Bar Code) is used world-wide for marking retail goods. The symbol encodes 13 characters: the first two or three are a country code which identify the country in which the manufacturer is registered (not necessarily where the product is actually made). The country code is followed by 9 or 10 data digits (depending on the length of the country code) and a single digit checksum. 2-digit and 5-digit supplemental barcodes may be added for a total of 14 or 17 data digits.”[6]

EAN-8

“**EAN-8** is a shortened version of the EAN-13 code. It includes a 2 or 3 digit country code, 4 of 5 data digits (depending on the length of the country code), and a checksum digit. While it is possible to add a 2-digit or 5-digit extension barcode, the primary purpose of the EAN-8 code is to use as little space as possible.

Unlike the UPC-E symbol, which compresses data that could also be printed as a full-size UPC-A symbol by squeezing out zeroes, the data digits in an EAN-8 symbol specifically identify a particular product and manufacturer. Since a limited number of EAN-8 codes are available in each country, they are issued only for products with insufficient space for a normal EAN-13 symbol. For example, a 2-digit country code permits a total of only 100,000 item numbers.”[7]

Code 39

“**Code 39**, the first alpha-numeric symbology to be developed, is still widely used-especially in non-retail environments. It is the standard bar code used by the United States Department of Defense, and is also used by the Health Industry Bar Code Council (HIBCC). Code 39 is also known as "3 of 9 Code" and "USD-3". Code 39 is a discrete, variable-length symbology. It is self-checking in that a single print defect cannot transpose one character into another valid character.”[8]

Code 128

“**Code 128** is a very effective, high-density symbology which permits the encoding of alphanumeric data. The symbology includes a checksum digit for verification, and the barcode may also be verified character-by-character verifying the parity of each data byte. This symbology has been widely implemented in many applications where a relatively large amount of data must be encoded in a relatively small amount of space. It's specific structure also allows numeric data to be encoded at, effectively, double-density."[9]

UPC-A

"UPC-A is used for marking products which are sold at retail in the USA. The barcode identifies the manufacturer and specific product so point-of- sale cash register systems can automatically look up the price."[10]

UPC-E

"UPC-E is a variation of the UPC-A symbol that is used for number system 0. By suppressing zeroes, UPC-E codes can be printed in a very small space and are used for labeling small items."[11]

Codabar

"Codabar is a low density numeric bar code. It covers 16 characters: 0-9, plus "-", ".", ":", "$", "/", and "+". The use of four separate Start and Stop characters allows useful information to be encoded by characters normally considered as overhead. Bar codes can be of variable length and do not require a checksum. We offer the Rationalized Codabar with a wide to narrow ratio of 3, allowing the most widely readable bar codes. Our set includes both standard and human readable versions of this font."[12]

Table t2 involves the features of the barcode mentioned above.

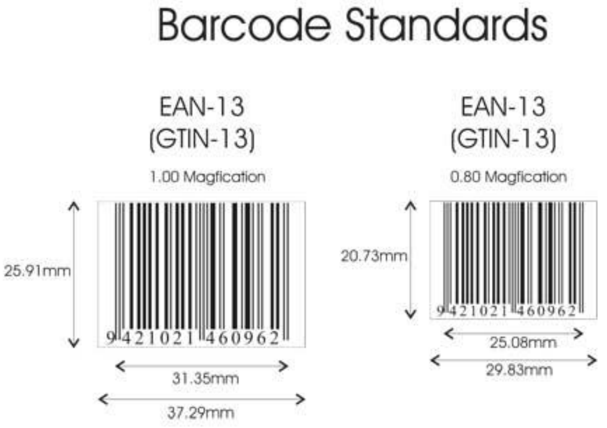
Symbology	Continuous or discrete	Uses
UPC	Continuous	Worldwide retail, GS1 approved
Codabar	Discrete	Old format used in libraries, blood banks, airbills
Code 39	Discrete	Various
Code 128	Continuous	Various
EAN 13, EAN 8	Continuous	Worldwide retail, GS1 approved

Table t2: the features of popular barcode[1]

4.2 Barcode structure

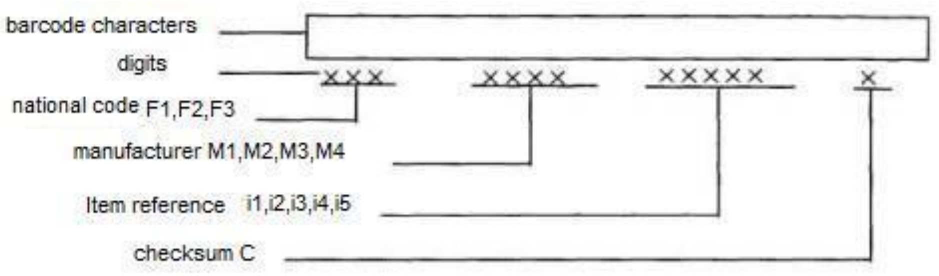
To make an easy understanding, an example of EAN-13 is taken to explain the structure of barcode.

EAN-13 outlook:



Pic p1: EAN-13 outlook

EAN-13 format:



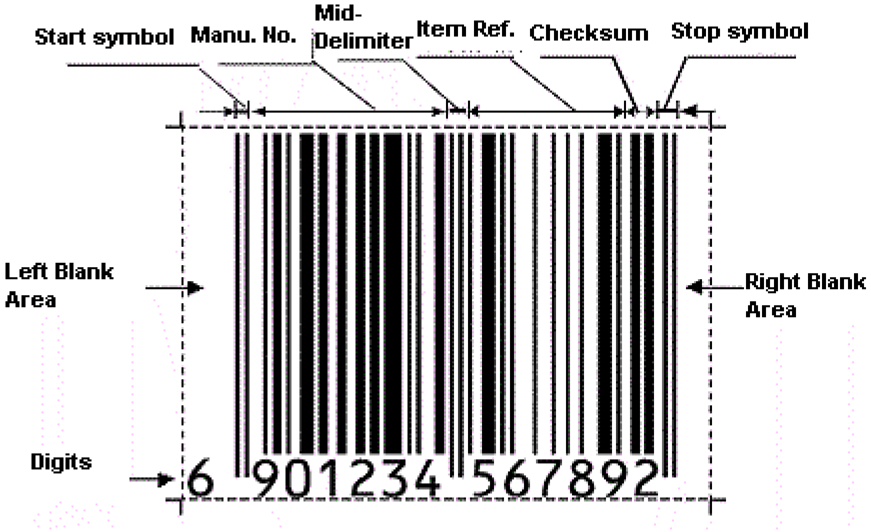
Pic p2: EAN-13 format

- National code,
  the first two (for EU countries) or three digits (for other countries) identify the national code, such as China(690), Japan(49*) or Australia(93*) etc.
- Manufacturer number,
  consisting of four, five or six digits depending on number of GTIN-13s required by the manufacturer to identify different product lines.
- Item reference,
  consisting of two to six digits.
- Checksum,
  a single checksum digit. The check digit is computed modulo 10, where the weights in the checksum calculation alternate 1 and 3. In particular, since the weights are relatively prime to 10 the EAN system will detect all single digit

errors. But since the difference of consecutive weights is even, the EAN system does not detect all adjacent transposition errors.

Additionally, the complete number is used as a reference key to look up information about the product line held on a database; the number is never normally broken down into its components within users' systems.[13]

EAN-13 structure:



Pic p3: EAN-13 structure

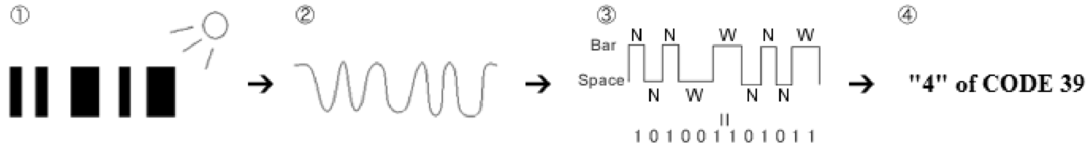
The GTIN(EAN)-13 encoded in the bar code has these components:

- Left blank area,
  with minimum width of 11 modules(bar or blank).
- Start symbol,
  with width of 3 modules, involves the start of barcodes.
- Manufacturer number(left data),
  with width of 42 modules(presenting 6 digits).
- Middle delimiter,
  separates the left data and right data, with width of 5 modules.
- Item reference(right data),
  with width of 35 modules(presenting 5 digits).
- Checksum,
  with width of 7 modules.
- Stop symbol,
  with width of 3 modules.
- Right blank area,
  with minimum width of 7 modules.
- Digits,
  for people recognizing. Additionally, the first digit of national code F1(named prefix-code) decides the parity(odd or even) of the left data.[14]

The left and the right blank area are actually buffer areas for both start and end reading the barcode. It is extremely important that make sure the widths of them are big enough for recognition. Otherwise, the barcode may not be recognized or misread.[15]

4.3 Barcode reading principles (scanning)

A brief description of barcode scanning is found on Website Denso Wave as below.



Pic p4: barcode reading process

1) A bar code consists of white and black bars. Data retrieval is achieved when bar code scanners shine a light at a bar code, capture the reflected light and replace the black and white bars with binary digital signals.
2) Reflections are strong in white areas and weak in black areas. A sensor receives reflections to obtain analog waveforms.
3) The analog signal is converted into a digital signal via an A/D converter. (Binarization)
4) Data retrieval is achieved when a code system is determined from the digital signal obtained. (Decoding process)[16]

4.4 Encoding rule

Encoding

EAN-13 is encoded using binary code. The binary codes for data symbol, start symbol, stop symbol and middle delimiter show in Table t3 as below:

Digit values	Binary representation		
	Left data		Right data
	Odd	Even	Even
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100
Start symbol	101		
Middle delimiter	01010		
Stop symbol	101		

Table t3: the EAN-13 encode[14]

Left data has parity(odd or even). Its parity is decided by the prefix-code F1.(It's one of the characters of EAN-13 barcode).

The combinatorial method of prefix-code and left data parity shows in Table t4:

Prefix code	Parity(odd or even) of left data					
0	O	O	O	O	O	O
1	O	O	E	O	E	E
2	O	O	E	E	O	E
3	O	O	E	E	E	O
4	O	E	O	O	E	E
5	O	E	E	O	O	E
6	O	E	E	E	O	O
7	O	E	O	E	O	E
8	O	E	O	E	E	O
9	O	E	E	O	E	O

Table t4: the combinatorial method of the left data symbol parity arrange[14]

From Table t4 it is clear that, this mapping of left data parity arrange and the prefix-code F1 is one to one corresponding and it is fixed for all time. For instance, the national code of China is “690”, so its prefix-code is “6”, then the left data parity arrange is “OEEEOO”

Checksum

“Checksum is used to prevent misreading of those barcodes which were printed in poor quality or damaged during the transportation.”

Steps of checking calculation of checksum:

1) Set checksum digit as the first bit. Arrange the barcode digits in reverse order(including the checksum digit).

   e.g. •pre-condition: Assume the barcode digits are “*6901038100578*”

   •post-condition: The new sequence is “*8750018301096*”.

2) Calculate the sum of even bit digits starting from the second bit, and then multiplied the result by 3.

   •post-condition: $N_1 = 3 \times (7+0+1+3+1+9)=63$

3) Calculate the sum of odd bit digits starting from the third bit.

   •post-condition: $N_2 = 5+0+8+0+0+6=19$

4) Add N_1 and N_2, get the sum N_3.

   •post-condition: $N_3 = N_1 + N_2 = 82$

5) Divide N_3 by 10 and get the remainder R. Mod R by 10, then get the complement of the result.

   •post-condition: $R = N3 / 10 = 2$

   $$Complement\ C = 10 - (2 \bmod 10) = 8$$

6) Match C and the Checksum digit, if they match the encoding correct.

   •post-condition: *C = 8 = Checksum*, so the encoding is correct

All the information about encoding rule is reference to Ref.[14]

4.5 Decoding scheme

4.5.1 Color Grayscale treatment

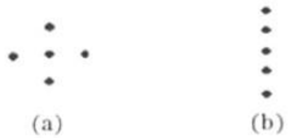
"Bit pictures are the pictures which are displayed and saved using bit map. Grey-scale picture is the kind of bit picture that only contains information of brightness, rather than contains both information of brightness and color."

All existing colors can be composed by red, green and blue (RGB). And we can transform colorful pictures into gray-scale picture by this formula:

$$Y = 0.299R + 0.587G + 0.114B$$

4.5.2 Median filter

The purpose of median filter is to wipe off the single dots so that the grey-scale picture becomes more clear to deal with.



Pic p5: Median filter sampling

Pic p5 (a) is a traditional 5-dot sampling window. But considering the shape of barcode, an improved sampling method (b) is coming out.

The specific step is to pick the target dot and its nearest 4 dots; sorting these 5 dots; selecting the middle one and covering the target dot by it.

4.5.3 Thresholding Processing

Purpose: to wipe off the unvalued information and reserve valued shape of barcode.

1) Count the times that grey-scale values happened.
2) Calculate average grey-scale value.
3) Get critical value (multiply the average grey-scale value by 0.6)
4) The grey-scale value of pixels which is bigger than the critical value is treated as black dot. While those smaller than the critical value are treated as white dot.

4.5.4 Image trimming

Purpose: to get the exact boundary of barcode.
Scan the binarized image starting at the left side to the right. It will meet left blank area and right blank area. Get the image between the left blank area and the right blank area as the valued part of image, and trim the others away.

4.5.5 Image rotating

Purpose: rotate the deflective image into the horizontal and vertical coordinate.

1) Set rotation grad (PI/180)
2) Record the distance of the opposite vertical angles
3) Rotate the image with one rotation grad every time
4) If the distance of the opposite vertical angles is bigger it of before, rotation finished and get the distance value. Otherwise, repeat 3).

4.5.6 Binarization

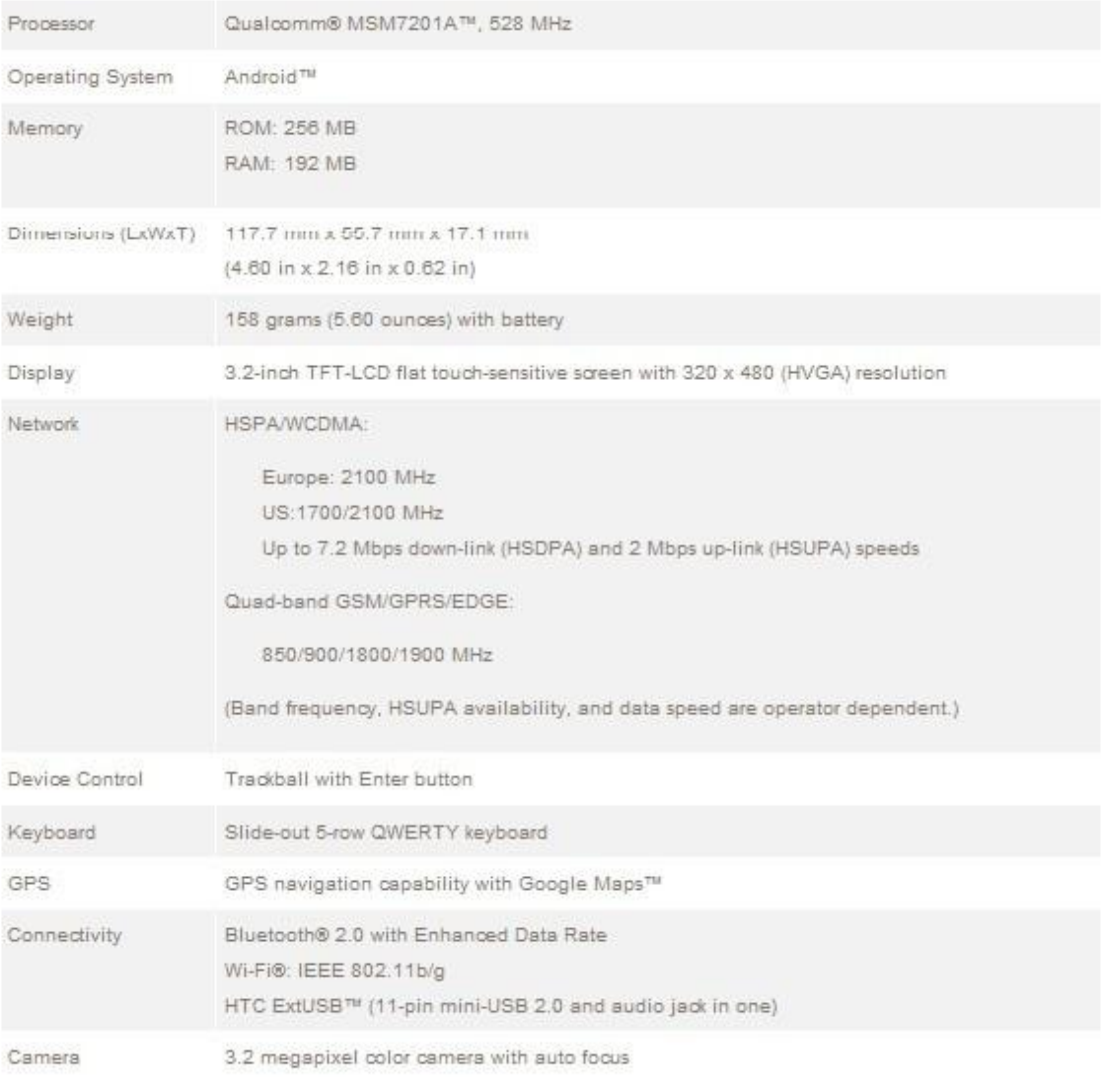
Scan the valued image. When the black dots in a column are more than 75% out of the total dots, this column is a bar, and set this column “1”. Otherwise, it’s a blank, and set it “0”.

All the information about decoding scheme is reference to Ref.[17]

5. Feasibility on Mobile

5.1 Hardware supports

A specification of the HTC G1 mobile platform:



Processor	Qualcomm® MSM7201A™, 528 MHz
Operating System	Android™
Memory	ROM: 256 MB RAM: 192 MB
Dimensions (LxWxT)	117.7 mm x 55.7 mm x 17.1 mm (4.60 in x 2.16 in x 0.62 in)
Weight	158 grams (5.60 ounces) with battery
Display	3.2-inch TFT-LCD flat touch-sensitive screen with 320 x 480 (HVGA) resolution
Network	HSPA/WCDMA: Europe: 2100 MHz US:1700/2100 MHz Up to 7.2 Mbps down-link (HSDPA) and 2 Mbps up-link (HSUPA) speeds Quad-band GSM/GPRS/EDGE: 850/900/1800/1900 MHz (Band frequency, HSUPA availability, and data speed are operator dependent.)
Device Control	Trackball with Enter button
Keyboard	Slide-out 5-row QWERTY keyboard
GPS	GPS navigation capability with Google Maps™
Connectivity	Bluetooth® 2.0 with Enhanced Data Rate Wi-Fi®: IEEE 802.11b/g HTC ExtUSB™ (11-pin mini-USB 2.0 and audio jack in one)
Camera	3.2 megapixel color camera with auto focus

Pic p6: HTC G1 specification [18]

- ✓ **CPU speed**(528 MHz)
- ✓ **Webcam support** (3.2 megapixel color camera with auto focus)
- ✓ **Networking support** (Wi-Fi: IEEE 802.11b/g)

5.2 Platform - Android Operating System

5.2.1 What is Android

“**Android** is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.”

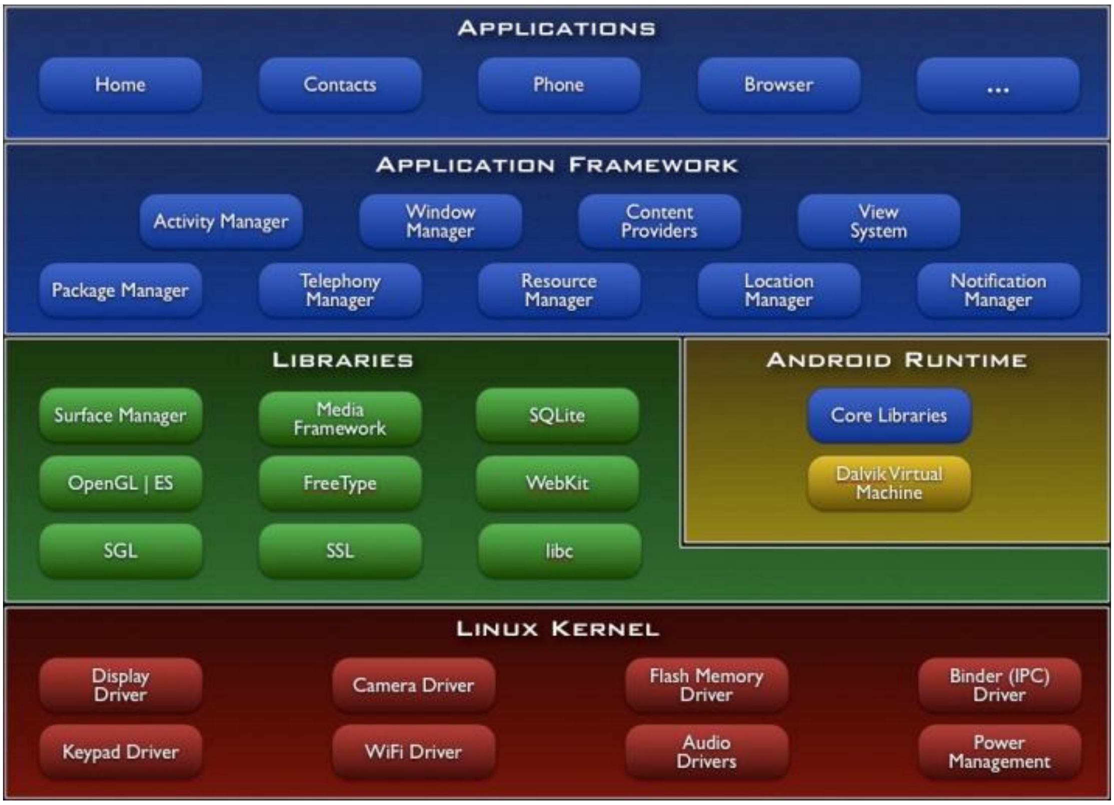
5.2.2 Features

The features were described on Android Developers website:

- **“Application framework** enabling reuse and replacement of components
- **Dalvik virtual machine** optimized for mobile devices
- **Integrated browser** based on the open source WebKit engine
- **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- **GSM Telephony** (hardware dependent)
- **Bluetooth, EDGE, 3G, and WiFi** (hardware dependent)
- **Camera, GPS, compass, and accelerometer** (hardware dependent)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE”

5.2.3 Android Architecture

The diagram below from Android Developers website shows the major components of the Android operating system:



Pic p7: major components of the Android operating system[19]

Applications
"Android will ship with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. All applications are written using the Java programming language."

Application Framework
"Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.
Underlying all applications is a set of services and systems, including:

- A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser
- Content Providers that enable applications to access data from other

applications (such as Contacts), or to share their own data

- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files
- A Notification Manager that enables all applications to display custom alerts in the status bar
- An Activity Manager that manages the lifecycle of applications and provides a common navigation backstack"

Libraries

"Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

- System C library - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices
- Media Libraries - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG
- Surface Manager - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications
- LibWebCore - a modern web browser engine which powers both the Android browser and an embeddable web view
- SGL - the underlying 2D graphics engine
- 3D libraries - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer
- FreeType - bitmap and vector font rendering
- SQLite - a powerful and lightweight relational database engine available to all applications"

Android Runtime

"Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool.

The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management."

Linux Kernel

"Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack."

All information about Android operating system above is referenced to Ref.[19]

5.3 Development kit – Eclipse with ADT

There is a great convenience we developing Android Apps in Eclipse with Android Development Tools. A short description from Android Developers website:

> "The Android Development Tools (ADT) plugin for Eclipse adds powerful extensions to the Eclipse integrated development environment. It allows you to create and debug Android applications easier and faster. If you use Eclipse, the ADT plugin gives you an incredible boost in developing Android applications:
>
> - It gives you access to other Android development tools from inside the Eclipse IDE. For example, ADT lets you access the many capabilities of the DDMS tool: take screenshots, manage port-forwarding, set breakpoints, and view thread and process informationd irectly from Eclipse.
> - It provides a New Project Wizard, which helps you quickly create and set up all of the basic files you'll need for a new Android application.
> - It automates and simplifies the process of building your Android application.
> - It provides an Android code editor that helps you write valid XML for your Android manifest and resource files.
> - It will even export your project into a signed APK, which can be distributed to users."[20]

6. Conclusion

A better process

A don't-need-to-take-photo recognition scheme is really attractive. Two benefits I mentioned before:

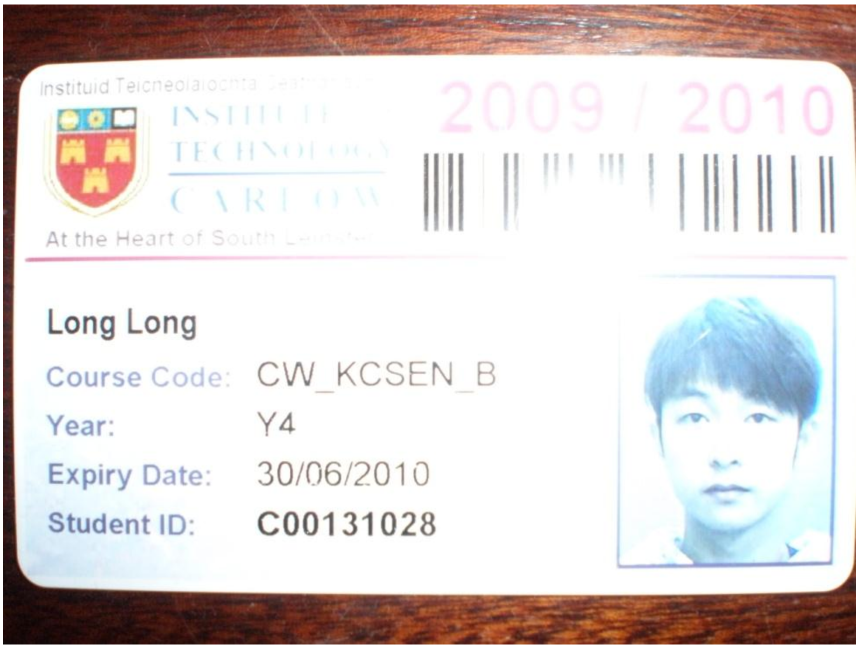
- Faster, for all information of pixels storing in Cache rather than in SD card or phone memory.
- Memory save; do not need to store photos in the phone.

Additionally, an extremely big benefit:

- It's dynamic, if barcodes cannot be recognized because of deflection, user just moves the phone a little bit and let it try again, instead of setting a very complex algorithm into the software to figure it out(like rotating image into the horizontal and vertical coordinate).

End user benefit from this, because it saves a lot of CPU time figuring out those rotating problems and the entire software is more flexible.

Student ID card used in CarlowIT:



Instituid Teicneolaiochta

INSTITUTE TECHNOLOGY CARLOW

At the Heart of South Leinster

2009 / 2010

Long Long

Course Code: CW_KCSEN_B

Year: Y4

Expiry Date: 30/06/2010

Student ID: C00131028

Pic p8: student ID card of CarlowIT

The barcode located on the top right of student ID card involves a unique identifier of the card holder.

Once developed, mobile barcode recognition will be very useful and can be widely used around the college such as for quick checkout for meetings, student card valid checking etc.

7. Reference

1. "Barcode - Wikipedia, the free encyclopedia." WIKIPEDIA. Web. <http://en.wikipedia.org/wiki/Barcode>.

2. "Brif introduction of 1D barcode." SYSTRON INCORPRATED. Web. <http://www.systron.com.cn/tiaoxingma/book1-2.htm>.

3. "Bar code usage." DENSO WAVE INCORPORATED. Web. <http://www.denso-wave.com/en/adcd/fundamental/barcode/case.html>.

4. Montgomery, Justin. "Mobile Barcode Recognition, Mobile Ticketing, and Mobile Coupon Technology – Real World Uses." RIFCOMOBILE. Web. <http://www.rifcomobile.com/articles/mobile-barcode-recognition-mobile-ticketing-and-mobile-coupon-technology-real-world-uses>.

5. Wang, Jian, and Chunhui Zhang. "Camera-based barcode recognition." PPTENT STORM. Web. <http://www.patentstorm.us/patents/7273177/description.html>.

6. "What is Barcode EAN-13." QKSOFT. Web. <http://www.qksoft.com/qk-barcode-generator/cean-13.html>.

7. "What is BarCode EAN-8." QKSOFT. Web. <http://www.qksoft.com/qk-barcode-generator/cean-8.html>.

8. "What is Barcode C39." QKSOFT. Web. <http://www.qksoft.com/qk-barcode-generator/c39.html>.

9. "What is Barcode 128." QKSOFT. Web. <http://www.qksoft.com/qk-barcode-generator/c128.html>.

10. "What is UPC-A." QKSOFT. Web. <http://www.qksoft.com/qk-barcode-generator/cupc-a.html>.

11. "What is UPC-E." QKSOFT. Web. <http://www.qksoft.com/qk-barcode-generator/cupc-e.html>.

12. "Choosing a bar code." ELFRING FONTS. Web. <http://www.elfring.com/barcode.htm>.

13. "European Article Number." Wikipedia. Web.
<http://en.wikipedia.org/wiki/European_Article_Number>.

14. "1-D Barcode Recognition Tech." *DDVIP.COM*. Web.
<http://tech.ddvip.com/2009-02/1235742950109947.html>.

15. "Usage of Blank Areas of Barcode." *Guangzhou Government Portal Site*. Web.
<http://www.gz.gov.cn/vfs/content/newcontent.jsp?contentId=488795>

16. "What is a bar code scanner?" DENSO WAVE INCORPORATED. Web.
<http://www.denso-wave.com/en/adcd/fundamental/barcode/scanner.html>.

17. "1-D Barcode Recognition System Design and Achieve" *Shuchen ZHONG* & *Xiaohong CHEN*, Journal of ChengDo University of Information Technology. Vol.21 Suppl (Dec.2006).

18. "Products T-Mobile G1 Specification." *HTC*. Web.
<http://www.htc.com/www/product/g1/specification.html>.

19. "What is Android." *Android Developers*. Web.
<http://androidappdocs.appspot.com/guide/basics/what-is-android.html>.

20. "Developing In Eclipse, with ADT." *Android Developers*. Web.
<http://androidappdocs.appspot.com/guide/developing/eclipse-adt.html>.