

# The Application of Modern PDA Technology for Effective Handheld Solutions in the Retail Industry

S. Coughlan

Department of Electronic Engineering  
National University of Ireland, Galway  
Nun's Island, Galway, Ireland  
sean@excentric.com

J.G. Breslin

Power Electronics Research Centre  
National University of Ireland, Galway  
Nun's Island, Galway, Ireland  
john.breslin@nuigalway.ie

**Abstract**– A modern handheld solution for the retail industry has been designed to replace older proprietary hardware with a PDA-based (Personal Digital Assistant) system. As part of the design, alternative handheld hardware was identified; retail handheld software was written; wireless communication protocols were implemented; and server software and utilities to access product databases were written. The system uses off-the-shelf hardware for which there are multiple suppliers, so the future of the system is secure. The cost to the retailer is potentially one fifth that of previous systems. As well as the lower price tag, the handheld device is competitive due to the amount of features the system boasts. A comprehensive set of user-friendly applications has been created for product creation, checking, etc. The speed of the system is instantaneous, solving a common problem with many handheld implementations on the market today. There is a reduced cost in supporting and maintaining the hardware, and there is also a greater opportunity for future development of the product beyond the retail sector. This system has undergone quality testing and has been deployed successfully in four live shop environments.

## 1 Introduction

A modern PDA-based handheld solution has been designed to replace proprietary barcode scanning handheld systems currently used in the retail industry.

Barcode scanning [1], [2] handheld mobile computing devices have been commonplace in the retail industry for some time. Most of these devices are specifically made for use only in retailing. They come in two forms: networked wireless and “batch” (non-wireless, where data must be synchronised via cable before operation). They tend to run some DOS variant on an x86 architecture.

The applications for these devices might typically contain price check, product lookup and print label functionality. Unfortunately, apart from high cost and weight issues, these systems were traditionally limited to displaying text (no graphical functionality), making user interaction difficult.

As well as these factors, there are a number of other motives for replacing the use of these devices:

- With few suppliers for these purpose-made devices, the cost of buying a new device is high.
- Servicing and maintaining the devices is costly.

- There is a risk that these handheld devices may cease to be produced without notice.
- The software that can be written for these types of devices is limited.

The PDA sector has seen solid growth over the last few years [3], and with it, there has been an increase in the number of fields for which PDA applications have been developed. Mobile handheld computers are providing higher employee productivity and improved customer service levels in a wide variety of industries and sectors.

The increasing adoption of wireless communications as a means to enhance operational efficiency is propelling the demand for such devices. Some of the industries in which mobile computers have been applied include manufacturing, transportation, military, field service and healthcare. The retail industry is an obvious choice for the development of wireless PDA applications, allowing remote creation and maintenance of product information on a central database server.

The possible benefits of replacing proprietary barcode scanning handheld systems with PDAs include:

- There are many PDA suppliers, and many more PDAs are being produced compared to the proprietary devices previously mentioned; therefore the cost of buying a PDA is lower, and because of a competitive growing market, is getting lower.
- Writing software for a PDA is not limited to one language, or one look-and-feel, and is made easier because they are more widely supported and there exists a wide range of pre-existing software.
- A PDA has multiple uses; the consumer is not only going to be buying a retail handheld device, but a productivity tool and mobile computer.
- Once the software for the PDA has been written, there is greater flexibility to improve it later on.

Section II will detail the advantages and disadvantages of potential hardware and software alternatives to previous proprietary systems, and will give an overview of the electronic devices and software technology that comprise the final system design.

In section III, the technical details of the handheld are dealt with in detail, including all of the software technologies in use and how they are applied to turn the handheld into a powerful retailer tool. The software application is explained with the aid of screenshots.

In section IV, the daemon on the server will be described, including detailed information on how the handheld communicates with it, i.e. the TCP/IP protocol that had to be implemented.

Section V explains how sets of database libraries were created to allow any Java application to access ISAM (Indexed Sequential Access Method) [4] databases. The structure of the system database will also be outlined, detailing how the retailer's product information is stored and retrieved.

Section VI will conclude with the technical accomplishments and customer benefits, with details of work that could be carried out to further the system.

## 2 Choice of System Architecture

The basic system architecture involves multiple portable handheld devices communicating via a wireless connection with a central server hosting the product database.

The first objective was to find appropriate hardware for the portable system to run on. Three categories of devices were considered: cheaper DOS-based devices, devices running a Microsoft OS, and Palm-type devices.

The DOS systems were ruled out early on due to the large time required for development and limited future possibilities. Devices running on either Microsoft CE (pocket PC) or XP (tablet PC) were available that incorporated the necessary barcode scanner and wireless network card. The average cost was around two-fifths the price of the previous proprietary device, with powerful software programming potential.

However, Palm devices with scanner and network functionality were available at half that price again. Lowering the cost of the handheld solution was a primary driving factor, and the Symbol Technologies SPT1846 PDA device [6] was finally chosen. The SPT1846 has a resolution of 160x160, comes with 8MB of RAM, runs PalmOS 4.1 and only weighs 0.38kg.

After the hardware was selected, the next objective was to determine what development language would be used to program the PDA device and how that software could be enabled to access information on the server database. There were a number of options available with regards to writing software on the handheld including:

- The source of an existing telnet application for the PalmOS could be modified to use the barcode scanner. However, this would not make use of any graphical or touch screen capabilities of the PDA.
- The open source code of the VNC viewer (Virtual Network Computing) could also be modified to access the barcode scanner. The VNC viewer program allows a remote graphical login to a server computer, and could be modified to allow a PDA device to remotely access the database on the server. However, the speed of the refresh rate of the VNC viewer on the PDA device was tested, and considered too slow for commercial use.
- Software using the PalmOS software development kit (SDK) in C could be programmed giving complete control over the device, with access to all

the native PalmOS application program interfaces (APIs). It would also undoubtedly be the fastest method to use. However, the source code would not have been portable to other architectures.

- The software could be developed in Waba [7], a Java virtual machine (JVM) specifically written for PDAs that is faster than Sun's Java 2 Micro Edition, and is portable as it runs on Linux, Windows, and PalmOS 3/4/5. This option was judged to offer the best opportunities for future development.

SuperWaba is an open source project based on Waba and licensed under the GPL (GNU Public License). It supports the PDA's barcode scanner, so no modification to the SuperWaba source was necessary to enable this functionality. It also supports all the graphical features of the PalmOS, including menu bars, buttons, list boxes, text boxes, keyboard control, checkboxes, radio buttons etc. Development in this language is quick, and can take place in either a Windows or Linux environment. Also because SuperWaba is open source, access to the native PalmOS API is available.

The final issue to be resolved during the design phase of this project was the manner in which SuperWaba applications would access information stored on a database on the server. The most obvious solution (and the one that was adopted) was to write a piece of server software or a daemon that would allow for communication with the PDA device via the TCP/IP protocol over the wireless network. 802.11b was chosen, as it has become the standard wireless networking technology for both business and home applications.

The wireless access point selected is the Symbol Technologies AP4121. It supports auto-channel selection, up to 127 clients, and has a top speed of 11 Mbps; however any 802.11b access point would be equally suitable. This particular access point works up to 90m indoors. The access point is also "WiFi" compliant.

In brief, the final system architecture that was chosen, as illustrated in Fig. 1, is as follows: PalmOS v4.1 runs on the SPT1846 PDA, the SuperWaba virtual machine is installed on PalmOS, and the handheld software application runs on SuperWaba. The handheld software initialises the network card, and communicates with the server (hosting the retailer's product information database) over TCP/IP via the AP4121 wireless access point.

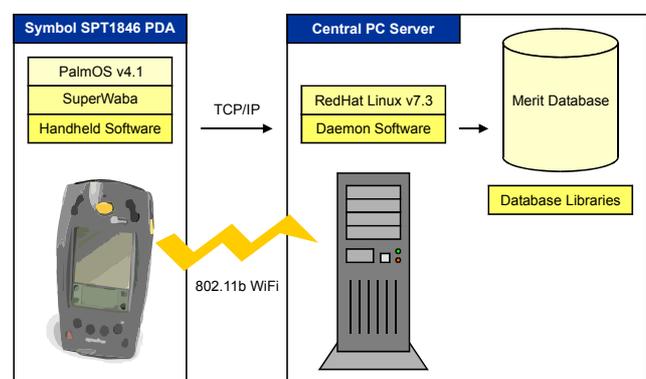


Fig. 1: Hardware and software layers: handheld client, network server and product database.

The server runs the RedHat Linux 7.3 operating system, and a custom Java-coded daemon, that allows network access from PDA clients, is loaded at server start-up. The daemon is a multi-threaded passive server and can therefore handle multiple handheld connections. The daemon allows communication with an ISAM database via a set of custom libraries, and product or other information is returned back to the handheld in real time.

### 3 Handheld Application

Waba [7] is a programming platform, optimised for PDAs and other small devices, that contains a strict subset of the Java language. Therefore, anyone familiar with Java can write Waba, and use the same Java development tools. The Waba SDK source code is available free to the public, which has made it possible for other development projects to use the Waba SDK as a basis for their software.

One such project is called SuperWaba, an extension of Waba with a rich API for use by small device developers. SuperWaba offers reduced libraries and a strict function set due to the constraints of memory and processing speed on PDAs. The time it takes to execute a program on a PDA as compared to a personal computer is significant. The developed handheld application was therefore designed to omit computationally intensive tasks.

The handheld application is proprietary to open source, and was developed in a combination of SuperWaba, C using GCC and the PalmOS SDK. Over 15,000 lines of actual source code were created in a complete Linux build and deployment environment.

The following sub-applications were written and implemented on the handheld:

- **Price Check and Change:** The user is able to scan in a barcode, and get information on the product associated with the barcode. Information includes: product cost, tax rate, items per pack, profit margin, price and description. The user has three means of scanning in a product: by using the barcode scanner, by entering the barcode number into the handheld, or by searching and selecting a product from a product lookup. From here the user can also change the system's sales price of the product, in which case the margin will be updated and the new sales price will be effective immediately on all tills.
- **Purchasing:** This allows the user to create, view or edit purchase orders, goods inwards dockets and returns dockets.
- **Stock Adjust and Count:** This allows the retailer to conduct a stock count, or directly alter the stock level figure.
- **Newspapers and Magazines:** This gives the retailer a system whereby the delivery and returns information on papers and magazines can be accurately recorded on the database.
- **Print Labels:** After scanning the barcode of a product, one may print a corresponding "shelf edge label". One can effortlessly walk around the shop

and after either making a price change or finding a product without a shelf edge label, print the labels out on a remote printer.

- **Supplier Links:** This allows the retailer to associate a particular product with a supplier.
- **Product Maintenance:** This gives the retailer the ability to create a product from the shop floor. It also allows the retailer to modify the existing details of a product on the database.

The range of functionality implemented in the handheld software is shown in the screenshots of Figs. 2 and 3. The system is secure, as each employee must enter an operator login ID and passcode before the sub-applications can be launched.

Also from the menu toolbar, the user can re-login, exit the system, view system details, or change the IP address that the handheld expects the server to be on.

Some custom implementations of many controls exposed by the SuperWaba consortium were necessary and these were returned to the development community. One was the addition of a native method to SuperWaba that allowed for the decoding of special EAN-13+2 barcodes.



Fig. 2: Initial screen showing menu options.



Fig. 3: Purchasing sub-section of the application.

Another custom modification was made to the list box class for instances where the scrollbar of the list box could not be removed. For example, even if there were no entries in the list box, the scrollbar would still exist (disabled). Visual real estate on the PDA screen was at a premium: An extra few characters could be displayed without the scrollbar, so code was written to optionally have a scrollbar when creating an instance of a list box.

#### 4 Networking Daemon and Remote Update

As mentioned previously, the software on the PDA communicates over an 802.11b wireless link to the server. A TCP/IP protocol had to be implemented to allow for communication. The TCP/IP protocol specification describes how the PDA, or any other client, can make requests for data retrieval from the server or make requests for information input into the server. It also describes the appropriate responses for all the requests. Many possible scenarios involving errors and warnings with data retrieval/input are taken into account.

The multi-threaded TCP daemon was written in Java and runs on port 28457. A virtual console screen is allocated on the server that outputs a log of all communications with the daemon. Every action carried out on the handheld device is logged on the server and is effectively stored immediately in the database.

To save processing time on the handheld, some repetitive functions are carried out on the daemon, and passed back to the handheld in a consistent format. One example is when a product lookup takes place on the handheld. Normally, 12 products fit on the screen at a time; the user has the ability to page up and down through this alphabetical list. Instead of the handheld reading 12 separate entries and extracting the product name for each, it simply requests a product lookup from the daemon, and the daemon passes back 12 entries.

A simple remote update and maintenance support utility was also written, allowing the software developer to upgrade software on the retailer's server system via a dial-up modem.

#### 5 Database Libraries

The databases on which product information was stored are ISAM databases [4], a precursor to relational databases. ISAM databases allow records to be accessed either sequentially (in the order they were entered) or randomly (with an index). ISAM databases contain fixed length records. When a database is being created, one must specify the length of a record in the database. After the creation, there is no way of adding a record with a length greater than that specified originally. This is one major disadvantage of ISAM. There are two files associated with every ISAM database: a DAT file and an IDX file. The DAT file contains the records in the database in sequential order, and the IDX file contains the indexes used to access specific records. ISAM updates the index file every time a record is modified or added to the database. Each index in the index file defines a different ordering of the records on a particular key.

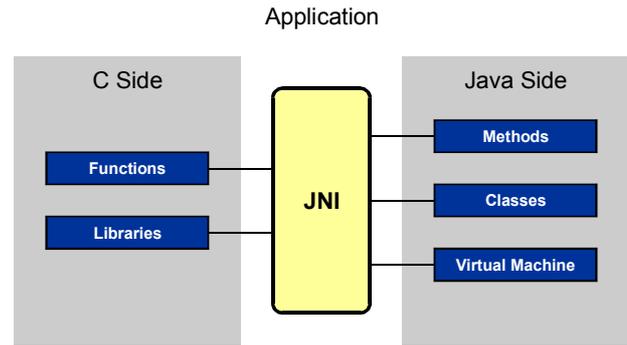


Fig. 4: JNI allows Java code that runs within the JVM to operate with applications and libraries written in other languages.

A product database may have several indexes, based on the information being sought. For example, a description index may order products alphabetically by name, while a price index may order the products by price. For an alphabetical index of product names, the description field of the record would be the key.

The Java Native Interface (JNI) is the native programming interface for Java that comes as part of its SDK. JNI allows Java code that runs within a virtual machine to operate with applications and libraries written in other languages, such as C. Programmers can use the JNI to write native methods to handle situations where part or all of an application cannot be written in Java. For example, one may already have a library or an application written in another programming language that must be made accessible to Java applications.

Since the networking daemon was created in Java, a Java-based method was necessary that could read and write to an ISAM database, as well as use any other features provided by ISAM. To do this proved challenging as the only way to access ISAM databases was to use a C-API provided by IBM called C-ISAM [5], and a shared C library which acted as a wrapper to the static C-ISAM libraries had to be created. Calls to the shared C library could then be made from Java using the JNI as shown in Fig. 4.

#### 6 Conclusions

A modern handheld solution for the retail industry has been developed. The system has undergone quality testing and has been deployed successfully in four live shop environments without any major problems. This system uses off-the-shelf hardware, i.e. the Personal Digital Assistant and the wireless access point. There are multiple possible suppliers for all the hardware used, so the future of the system is secure. Most importantly, the cost to the retailer is potentially one fifth that of previous proprietary systems.

A comprehensive set of user-friendly applications has been written for the PDA. The speed of the system is especially unique; there is no wait between the time a product is scanned and the instant all the product information appears on screen. This is a common problem with many handheld implementations on the market in the retail industry today.

The system can easily be updated to incorporate future functionality such as scanning of products at the back door, support for outer case barcodes or shelf replenishment, and there are many opportunities for future development of the system beyond the retail sector.

## Acknowledgement

The authors gratefully acknowledge the contributions of Michael Hughes and Merit Solutions, Claregalway, Ireland who sponsored this project.

## References

- [1] M. Kuroki, T. Yoneoka, T. Satou, Y. Takagi, T. Kitamura and N. Kayamori, "Bar-code recognition system using image processing", in *Proceedings of the 6th International Conference on Emerging Technologies and Factory Automation Proceedings*, Sep. 1997, pp. 568-572.
- [2] R. Muniz, L. Junco and A. Otero, "A robust software barcode reader using the Hough transform", in *Proceedings of Information Intelligence and Systems*, Oct. 1999, pp. 313-319.
- [3] M. Quan, "PDA market poised for growth after sluggish 2002", *Electronic Engineering Times*, Feb. 2003, [www.eetimes.com/story/OEG20030225S0036](http://www.eetimes.com/story/OEG20030225S0036)
- [4] J.K. Mullin, "An improved index sequential access method using hashed overflow", *Communications of the ACM*, vol. 15, no. 5, May 1972, pp. 301-307.
- [5] IBM Informix C-ISAM library of C functions, [www.ibm.com/software/data/informix/cisam](http://www.ibm.com/software/data/informix/cisam)
- [6] Symbol SPT1800 PDAs, [www.symbol.com/palm](http://www.symbol.com/palm)
- [7] Waba programming platform, [www.wabasoft.com](http://www.wabasoft.com)