A PORTABLE MULTIMEDIA INFORMATION DEVICE
IN A WIRELESS OPTICAL DATA LINK

Grantham Pang and Gordon Poon

Abstract

This paper describes the design and implementation of a portable multimedia information device that receives information from a free space optical data link established using visible light LEDs. The visible light, which is transmitted from a transmitter made up of LEDs, is modulated and encoded with data and information. The portable receiver is consisted of a demodulation circuit and an interface to a handheld PC. Software applications have also been developed on this hardware platform so that the mobile PC device can receive data, audio and graphics through the visible light from the transmitter, which is located at some distance from the device. With the high frequency switching of the LEDs, the transmitter appears to be constantly illuminated.

1. Introduction

This paper is developed based on the idea that the visible light emitted by light emitting diodes (LEDs) can be modulated and encoded with data and information. This new technology makes use of the fast switching capability of LEDs to establish a free space optical data link. Essentially, visible light becomes a communication medium for the broadcasting of information. Current studies [1,2] have already shown that Light Emitting Diode (LED) has many advantages over incandescent light including money and energy saving, and with much easier maintenance. Recent research has developed the idea on the dual use of the LED based information device and traffic signs [3,4]. Information can be transmitted optically to a photodiode-based receiver in a wireless way. This is similar to infrared LEDs for optical wireless communication [5,6], but it also has many differences. While infrared LED operates at 780-950nm, visible light LED operates in the visible light spectrum and it is utilized for communication. This idea leads to the dual use of many display, signaling devices that are constructed from LEDs [7].

Pang et al. [8] describe the idea, design, evaluation and implementation of LED wireless communication systems. Both analog audio broadcast and digital data communication system are covered. An intelligent traffic light was fabricated and evaluation experiments were conducted. The viewing angle (half power) is found to be 30°. For digital signal, the transmission rate can reach 500 kbps and the throughput is around 2000 bytes/s. For an optical power at the receiver of 0.8 μW, the bit error rate (BER) is around 9×10^{-8}.

In this paper, the focus is on a multimedia portable information device that is developed for receiving information when there is a direct line of sight (LOS) between the lighting transmitter and the receiver device. The transmitter, in addition to its normal function of being an illumination, indication or display device, can also be used as an information broadcasting device. This information beacon makes use of the visible portion of the EM spectrum, which is unregulated by FCC or any countries in the world.

2. Hardware and Protocol Design

The developed multimedia information device is modular and configurable. Figure 1 shows that the whole hardware data-line can be classified into three levels: asynchronous level, synchronous level and Manchester level. Each layer can establish the connection by other means. For example, a lap-link cable can be inserted in the Asynchronous Level and replaces the hardware transmitter and receiver. On the other hand, the hardware transmitter and receiver can replace the lap-link cable in any simplex application. This modular design has eased the development and trouble-shooting procedures. When working at the high level (information server and software receiver

0098 3063/00 $10.00

kernel) debugging, a serial lap-link cable is used instead of the hardware transmitter and receiver to localize software problem. When trouble-shooting the hardware transmitter and receiver, dump terminal program can be used to send out byte patterns, which can isolate failures from the developing software. In the final design, the following settings are established:
Transmitter baud = 19200 baud / sec
Receiver baud=57600 baud / sec
Manchester data rate=28800 bit / sec.

Figure 2 gives a picture of the Palm PC that is used as the portable device that displays the multimedia information. Figure 3 shows the back of the handheld-PC with the interface circuit that demodulates the visible light signal. The LED transmitter that is constructed for the broadcasting of information is shown in Figure 4. It is constructed with 192 ultra bright white light LEDs and resembles a table lamp. The LEDs are modulated through another interface circuitry connected to the information server, which is a desktop PC.

The information server supports generic file transfer and dynamic file transfer scheduling. A user can decide the files to be transferred in a specific sequential order. In some cases, the user may suggest that some files are more important than others, and the priority of those files can be increased by increasing the number of instances of those files in the transfer profile. With other factors remaining constant, the probability of those important files being received would be increased.

A methodology, called "wrapper", is used to encapsulate all the file information necessary for the reconstruction of that file in the destination computer. This makes the source copy and the destination copy identical not only for the file content, but also with the file times and attributes. The file CRC appended at the end of the file content is used to check for the file integrity. Figure 5 shows how the wrapper is arranged.

Our protocol caters for transmission loss and has the ability to resume file transfer. Each frame has 384 bytes and starts with a frame header, in which there is frame identification information stored. The layout of the frame header is shown in figure 6. The protocol supports multiple files transmission at the same instance, with the Frame code indicating which file a frame belongs to. The 16-bit frame code places a maximum limit of 65536 (2^{16}) files to be transmitted concurrently, but this limit is seldom reached. The Frame count is the sequence number of the frame, which can be used to tell the software receiver kernel where this frame is stored in the destination file. The total frame count is the total number of frames that a file should have. The 16-bit total frame count raises another limitation on the file size, which is around 20Mb, as a file can only have 65535 ($2^{16}-1$) frames. This maximum limit on file size is relatively large for mobile devices of the current technology. A recognizable frame structure is used to cater for a leaky channel. If byte leakage occurs in the current frame, we can recognize which is the next frame and neglect the current frame. The recognizable frame structure is shown in figure 7. The number of bytes that is used for data is 328, which gives an efficiency of 85.42%.

3. Transmitter Hardware and firmware

Figure 8 shows the block diagram of the transmitter. The data/signal flow is indicated by the arrows. The asynchronous data stream comes from the serial port, and they are level shifted to TTL 0-5V by a voltage shifter. The asynchronous TTL data stream enters the MCU by the Serial Communication Interface (SCI), which is an asynchronous RS232 interface standard without hand-shaking. After the double buffering inside the MCU, a synchronous data stream is clocked out through the Serial Peripheral Interface (SPI). The SPI is a typical serial interface used by MCU. The Manchester encoder samples the synchronous data, together with the timer output. Finally, the Manchester code is produced at the LED driver and the BNC outlet. The fabrication is done on a multi-purpose circuit board as shown in figure 9.

The transmitter firmware runs in full duplex mode, with data coming in from the SCI at the same time when data is going out through the SPI. Two handling methods are used to

handle the subsystems: polling is used to handle the SCI, and interrupt is used for the SPI.   Figure 10 shows the flow chart of the main routine inside the transmitter firmware. In the initialization stage, the registers and variables have to be initialized. The next stage is a looping stage, which loops for every byte from the SCI interface. Whenever a byte is received, it is stored into the buffer.   When a buffer has been filled up, the writing buffer needs to be swapped with the reading buffer, and the SPI interface would drain the buffered data.   After the swap, the writing buffer is empty while the reading buffer is full of data to be dispatched.

4.   Receiver Hardware and firmware

Figure 11 shows the receiver block diagram. The data/signal flow is indicated by the arrows.   The bit stream from the visible light signal is first received and amplified by the analog circuits.   The rectified bit stream then goes into the Manchester decoder, where carrier locking, synchronization and byte alignment take place.   The NRZ data stream from the Manchester decoder is then clocked into the MCU.   After the internal buffering inside MCU, it pumps the buffer out.   The data stream is level shifted to the RS232C standard ± 12V by voltage shifter.   The power circuits provide the MCU a rectified 5V DC source from 3V batteries.   The reset circuit is connected to the serial port via voltage shifter, which enables the receiver application software to automate the process. The fabrication of the hardware receiver is on a PCB, which is shown in figure 12.   Figure 3 shows how it is attached to the Palm PC.

The receiver firmware runs in half duplex mode, with data going out through the SCI interface after the data coming in from the SPI interface has filled up the internal buffer. Both SCI and SPI subsystem are handled by polling.   Figure 13 shows the main flow chart of the receiver firmware.   First, the registers and variables are initialized in the initialization stages.   Next is a byte alignment process.   Then, there is a buffering/looping stage. The incoming bytes are stored into the buffer.   When the buffer is full, the firmware would move to the next stage, in which the MCU is switched from the receiving mode to transmitting mode.   There

are routines to dump data out.   After the dumping operation, the MCU would wait for the next incoming byte.

5.   Software Receiver Kernel

The primary function of the software receiver kernel in the Palm PC is to reconstruct identical destination files in the persistent storage on the target platform.   Its data source is coming through the serial port from the hardware receiver.   There are three threads running concurrently, namely, **portReadThread**, **framingThread** and **filingThread**.   These three threads are in two pairs of producer-consumer relations. **PortReadThread** produces arrays of byte for **framingThread**, whereas **framingThread** produces files data for the **filingThread** to consume.

The **portReadThread** is responsible for reading the serial port after an entire buffer array of bytes are received, and passing the array to the **framingThread**.   In some cases, the **portReadThread** may produce a new array of byte faster than **framingThread** consumes, and therefore a buffering array would standby in the interface between the two threads to prevent data overflow.   The shared data between the two threads would raise another problem   –   thread synchronization.   Hence, the statements accessing the overflow buffer were protected by a Critical Section object.

The **framingThread** identifies frame structure in the byte array produced by **portReadThread**.   After the frame structure is recognized, **framingThread** extracts the frame data and verifies them by frame CRC. If the CRC verification is passed, the verified frame will be arranged in a **CPtrArray** object. When all the frames of a file are verified, **framingThread** will pass a **CPtrArray** object and fires an event to **filingThread**. The **filingThread** then uses the information contained in the file stamp to reconstruct an identical file on the persistent storage.   The **filingThread** skips the reconstruction if an identical file that already exists.   No matter if the new file is reconstructed or not, the associated memory is freed up before the routine ends.

6. The Application and Results

There are numerous applications that can be developed from the proposed optical communication technology and multimedia information receiver reported in this paper. In an outdoor application, the portable Palm PC can receive multimedia information from an LED display or LED traffic lights. For indoors, any lighting source that is from LEDs can potentially carry multimedia information. Thus, any indoor lighting can be used as an information beacon, which suggests on its dual use as both an illumination and broadcasting device. Many applications have been developed to demonstrate the ease of using the software receiver kernel architecture. In this section, an example application of using visible light to broadcast information of some promotion items in a trade show or exhibition hall is shown. An application developer does not require any knowledge about the hardware or the information server.

An application is assumed to open the serial port for information. To make use of the software receiver kernel, the application simply declares an instance of a class of objects for processing. Some parameters will need to be passed when the receiver kernel is initialized. They include the handle of the serial port and the window handle of the main frame of the application. Then, the frame size (in our case, 384) is also passed along with the message ID. When the receiver kernel receives a new data frame and needs to construct a new file, it will notify the application by placing messages in the message queue of the application's main frame. Upon receiving the messages, it would be the work of the developer to write the message handler for the application. The application is responsible for setting the serial port parameters, such as baud rate, flow control settings and so on. Finally, the application has to close the serial port before it terminates.

The application information is arranged in pages by their category, and the pages are arranged in a tabbed pane fashion. The connection panel (as shown in figure 14) is usually the first panel that a user would see. A user can choose to work offline or to work

online. When the user decides to work offline, the application serves as an information browser. It would display the cached data acquired in last online session. To ensure the robustness of the receiver unit, the hardware receiver is reset every time a new connection is made. The connection panel issues a hardware receiver reset signal by supplying a positive pulse on the RTS line.

Under the connection panel, three panels can be shown: data, image and audio panels. A data panel is shown in figure 15 and a user can obtain the digital data that can be transmitted through the visible light. An image panel is shown in figure 16. A user can see a graphic or image showing in the panel. As the data volume is usually large for bitmap files, transmission speed is an issue of concern. The panel supports bitmap of different color depth, and the table below shows the estimated transmission time for different type of bitmap files.

Color depth	File size	Frames occupied	Shortest transfer time
256 color	74kb	231	~45s
16 color	37kb	115	~25s
Black/white	10kb	27	6s

An audio panel is shown on figure 17 and a user can select any file on the list for playing. If the checkbox "AutoPlay when new file(s) arrive" is checked, a new incoming file will be played automatically upon its arrival. Similar to the case of image panel, the data volume of audio file is another concern. A software called Mobile Voice is used to compress and decompress audio data. The table below compares the difference in size between the Mobile Voice and PCM format.

Format	Parameters	Duration	File size	Shortest transfer time
Mobile voice	8kHz, 8bit, mono	4.32s	2kb	~2s
PCM	8kHz, 8bit, mono	4.32s	34kb	~25s

In another files panel, all the received files are listed.   As the files in the cache directory will be deleted periodically, if the user wants to retain a "persistence" copy of the files, a copy should be made to another directory.   If there are new file format that cannot be handled by the software kernel, the user can manipulate the file by a third party software.

7. Conclusions

In this paper, the design and implementation of a portable multimedia receiver is described. It is attached to a Palm PC and can be used to demodulate information encoded in visible light. The multimedia information can then be shown on the handheld PC. The protocol is designed to be both upward and downward compatible.   Later versions of the protocol can append extra information after the file CRC bytes.   With the open architectural design of the software kernel, it eases the development of many new applications.   For example, applications can be developed for a museum, exhibition hall, supermarket or department store environment.   The detailed information on individual item or product can be broadcast via a plurality of LEDs which are also used for the purpose of illumination or display.   This technology makes use of the direct LOS property of visible light communication.   Thus, the lighting source is also an indication of the location of the transmitter.

References

1. G.B. Stringfellow and M.G. Craford, "High Brightness Light Emitting Diodes", Semiconductors and Semimetals Vol. 48, *Academic Press,* 1997.

2. M.G. Craford, "LEDs Challenge the Incandescents", IEEE Circuits and Devices, pp. 24-29, September 1992.

3. G. Pang, T. Kwan, C.H. Chan, H. Liu, "LED Traffic Lights as a Communication Device", Proceedings of the International Conference on Intelligent Transportation Systems, Tokyo, Japan, pp.788-793, 5-8 Oct. 1999.

4. G. Pang, C.H. Chan, H. Liu, T. Kwan, "Dual use of LEDs: Signaling and Communications in ITS", Proceedings of the 5th World Congress on Intelligent Transport Systems, Seoul, Korea, 12-16 Oct.1998.

5. T.S. Chu and M.J. Gans, "High speed infrared local wireless communication", *IEEE Communications Magazine,* pp. 4-10, August 1997.

6. J.M. Kahn and J.R. Barry, "Wireless Infrared Communications", *Proceedings of the IEEE,* Vol. 85, No.2, pp.265-298, Feb 1997.

7. John O'Connell, "The Philadelphia Story", Traffic Technology International, Aug/Sept '97, p.106-110, 1997.

8. G. Pang, C.H. Chan, K.L. Ho, T. Kwan, Edward Yang, "Visible light communication for audio systems", IEEE Transactions on Consumer Electronics, Vol. 45, No. 4, November 1999.

BIOGRAPHIES

Grantham K.H. Pang obtained his Ph.D. degree from the University of Cambridge in 1986. He was with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, from 1986 to 1996 and joined the Department of Electrical and Electronic Engineering at The University of Hong Kong in 1996. Since 1988, he published more than 100 technical papers and has authored or co-authored five books.   His research interests include machine vision for defect detection, optical communications, expert systems for control system design, intelligent control and intelligent transportation systems.

Dr. Pang was the Organizing Chair of the 1996 IEEE Symposium on Computer-Aided Control System Design. Dr. Pang is an Editor of the International Journal of Intelligent Control and Systems, Machine Intelligence and Robotic Control, and Control and Computers. He is a Chartered Electrical Engineer, and a member of the IEE, HKIE and IEEE.

Gordon Poon received the B.Eng. degree in Electrical Engineering from The University of Hong Kong in 2000. Currently, he is a graduate student at Stanford University.
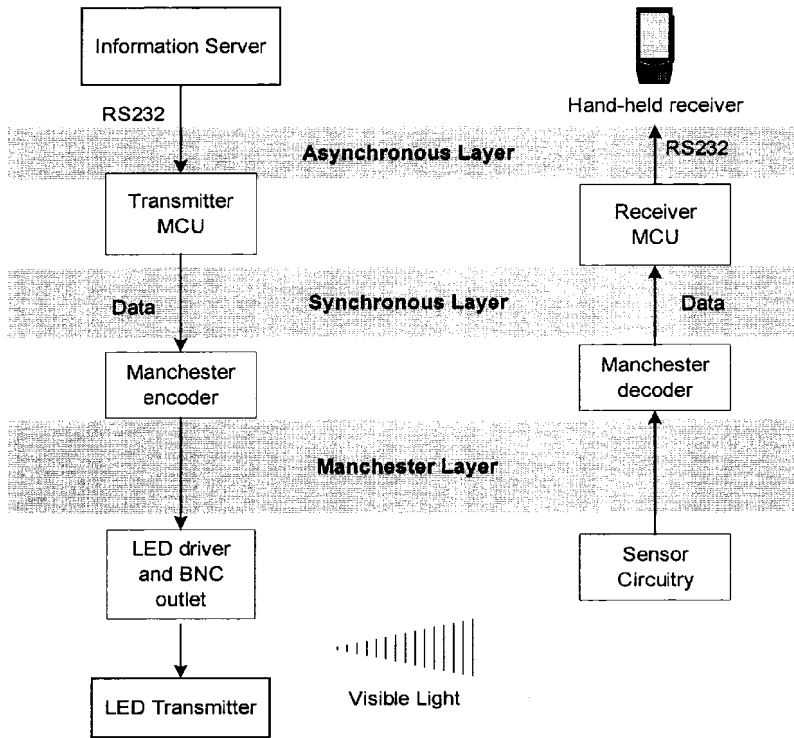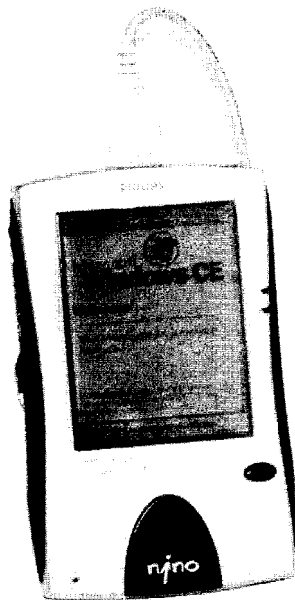
Figure 1

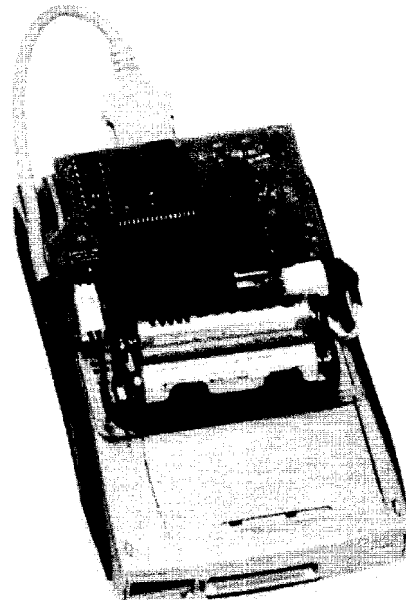Layers of data-link hardware

Figure 2 Multimedia Information Device

Figure 3 Receiver Module
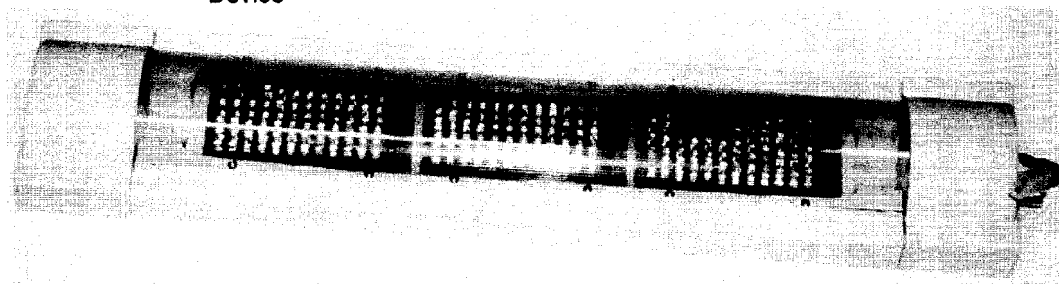
Figure 4 Visible Light LED Transmitter

File stamp	File Content	File CRC

Creation Date	Last Access Date	Last Modification Date	File Attribute	Filename Length	FileName	File Length
8 bytes	8 bytes	8 bytes	4 byte	1 byte	max 255 bytes	4 bytes
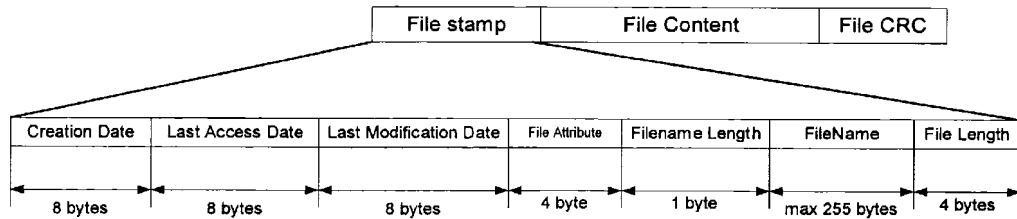
Figure 5 Layout of the file wrapper

Frame Header

Frame code	Frame Count	Total Frame Count
2 bytes	2 bytes	2 bytes

Frame ID
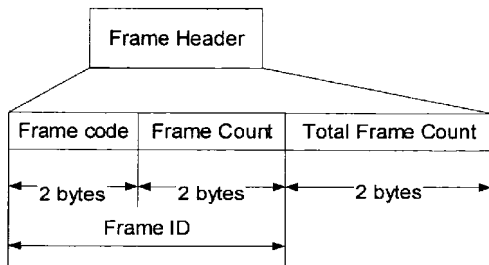
Figure 6   Frame header structure

SYNC	STX	FH	STX		STX			STX		STX	SYNC

12 bytes
1   8
41 (Stx+Data)
2 bytes
Total frame length = 384 bytes; Efficiency 85.42%
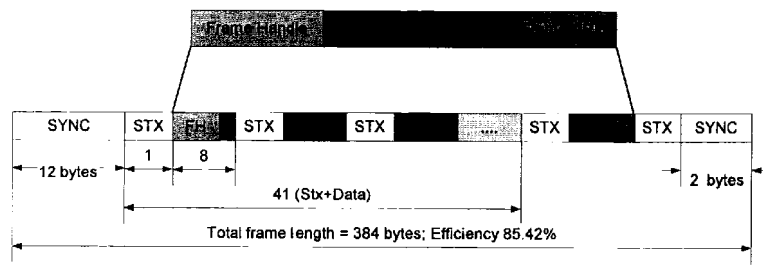
Figure 7   The frame structure

TTL RS232 interface

SCI

Reset circuit   →   MCU Double Buffer   Crystal pins   Crystal

Reset, mode pins

SPI   Timer   I/O   DIP SW

Manchester encoder

LED driver & BNC outlet
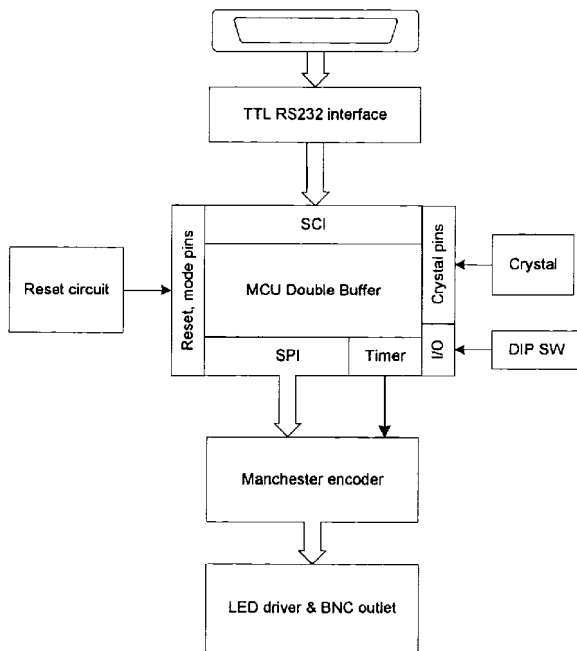
Figure 8   Block diagram of hardware transmitter
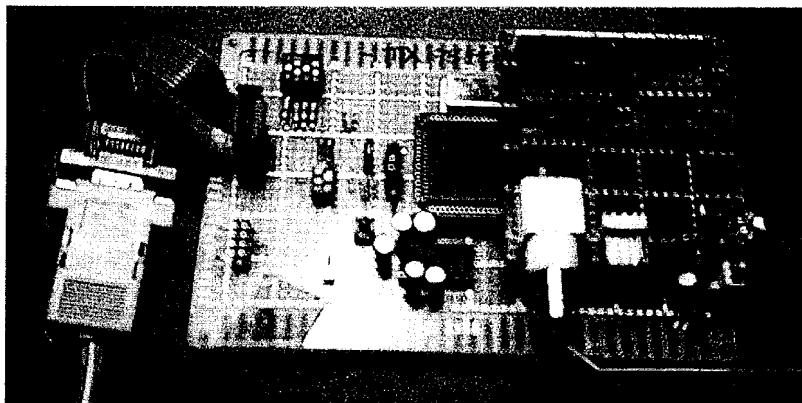
Figure 9

The appearance of the hardware transmitter.

BNC socket

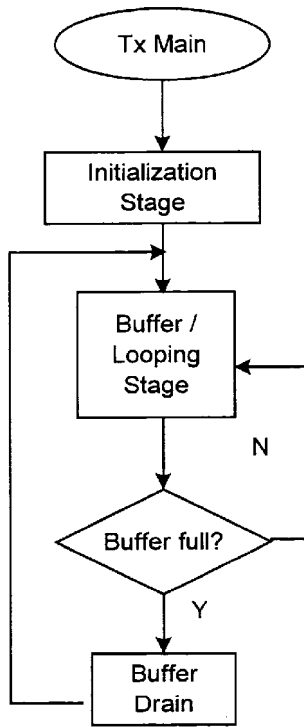**Figure 10** Transmitter firmware main
routine flow chart

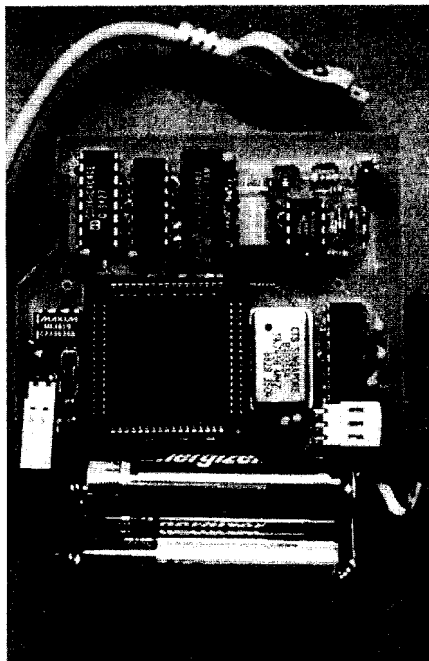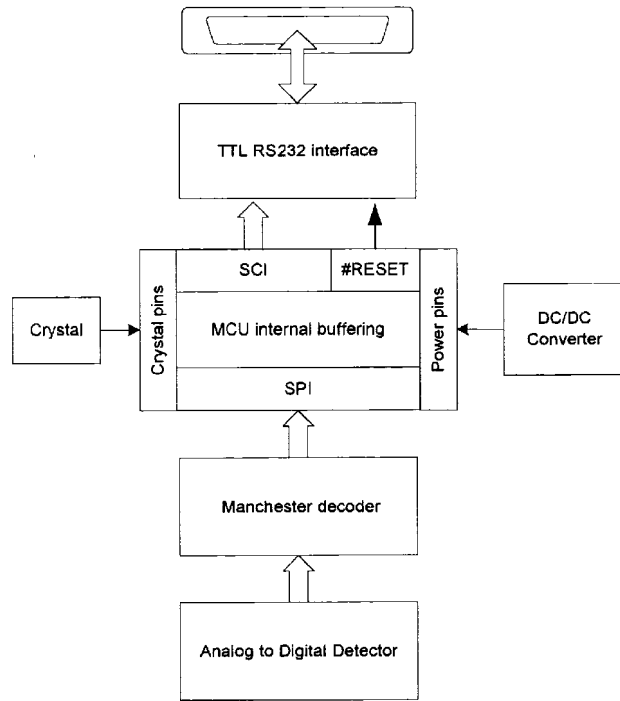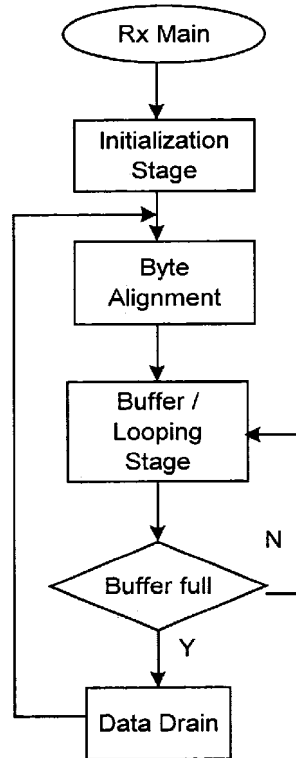**Figure 11** Block diagram of hardware receiver

Figure 12   The appearance of the
completed PCB receiver

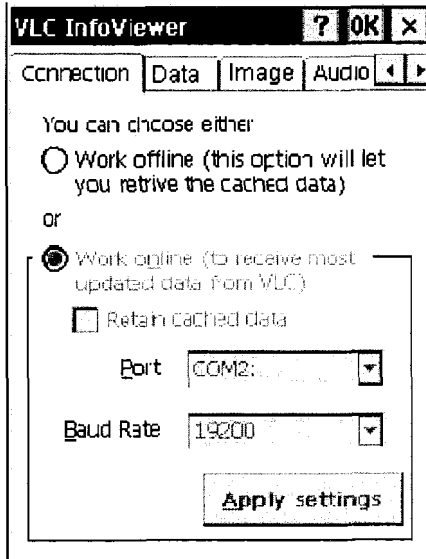**Figure 13** The flow chart of receiver firmware
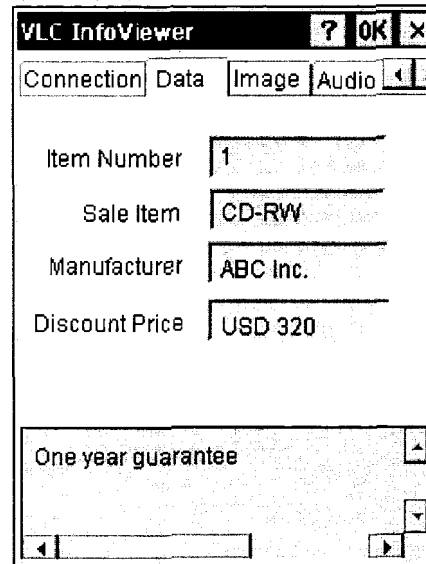
Figure 14 The Connection panel
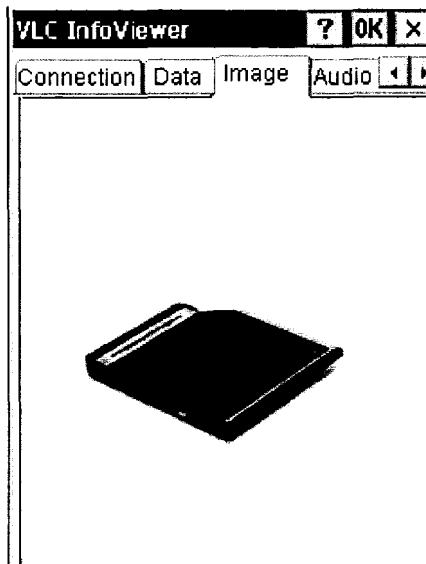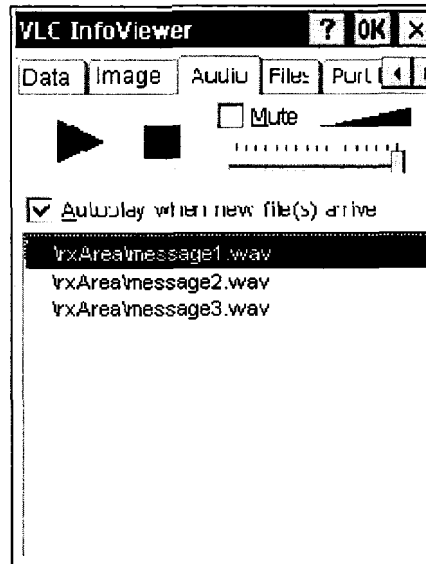

Figure 15 The Data panel

Figure 16 The Image panel

Figure 17 The Audio panel