TOSHIBA
Leading Innovation

White Paper

How to Quickly Develop a Low-cost Handheld Device

James Yang Ph.D, Sr. Technical Staff Member, Toshiba America Electronic Components, Inc.

Highlights

- Most handheld devices require support for a similar set of functions: display control, user interface and data storage/exchange. All of these functions can be supported by a single microcontroller
- By using a cost-effective microcontroller running on an open source software platform, designers can bring handheld devices to market at low cost
- The design of a sample handheld device, a portable electronic dictionary, is provided in this paper

Abstract

Handheld devices have become the form factor of choice for many consumer (media players, portable dictionaries, portable gaming, etc.), medical and industrial devices. Most handheld devices share several common elements, and using a microcontroller (MCU) that integrates the peripherals needed to support these elements can play a big role in getting a new handheld device to market quickly and cost effectively.

The most common elements in handheld devices are man-machine interfaces, such as LCD controllers and keypad/touch screen controllers and their related drivers and high-level software IP. Additionally, most handheld devices require some sort of storage device controller, for both internal storage media (NOR or NAND Flash memory) and external storage devices such as USB drives or SD (Secure Digital) cards.

Additionally, as handheld devices based on open source software become increasingly popular thanks to their low-cost operating systems, this paper will describe an open source-based system, using µC/OS-II[1] and Nano-X Window System[2].

Finally, all of the various elements of a handheld device will be examined in an actual system-level solution, an e-dictionary based on Toshiba's TMP92CZ26 MCU.

Preface

The basic functions of any handheld device are: user interface input, user interface output and data storage and exchange.

The user interface lets consumers input data into the device for processing (i.e. entering an address into a GPS device, pushing buttons on a portable game device, and moving a scroll wheel on a portable media player). The touch screen is an increasingly popular user interface for handheld devices and in many applications is replacing more traditional user interfaces such as mechanical keys or switches. In light of this, touch screen interfaces will be a particular focus of this paper.

User interface output consists of the display or playback of stored content (video, graphics, etc.). This content is usually presented to the consumer over a color LCD panel, but older technologies like LED indicators and text displays are still used in lower-end handheld devices.

The memory used in handheld devices to store data continues to increase in density at a phenomenal rate in order to give users the ability to store more and more content. In addition to media for content storage, the application code driving today's handheld devices is also growing and requires more memory space. The most cost-effective storage solution for handheld devices is NAND flash, which is the memory technology of choice for both onboard memory and external storage, typically as USB drives or SD cards.

1 See http://www.micrium.com/products/rtos/kernel/licensing.html. License(s) required for commercial use.
2 See http://www.microwindows.org/faq.html. Available for license under the Mozilla Public License (MPL), at http://www.mozilla.org/MPL/MPL-1.1.html, or under the GNU General Public License (GPL), at http://www.gnu.org/copyleft/gpl.html

Now that the basic functionality of a handheld device has been broken down into three categories, a closer look at the most commonly used user interface input, user interface output and memory sub-systems in handheld devices is warranted.

1. Color LCD Controller and Graphic Design

LCD panel-based graphical user interfaces are used in the majority of handheld devices on the market today.

An LCD panel is comprised of a matrix of pixels, and each pixel has a red, green and blue sub-pixel. The brightness of the sub-pixel is controlled by an electrical charge in a capacitor delivered through a transistor. To properly display an image, each sub-pixel needs to be refreshed at a certain time, based on image data held in a frame buffer located in the handheld device's main system memory.

Each pixel on the LCD corresponds to a memory unit in the frame buffer. The size of the unit depends on the color depth (expressed as bits per pixel or BPP). The color bits are split into the three primary colors red, green, and blue (RGB). Typical color depths are 8-bit RGB (R3:G3:B2), 12-bit RGB (R4:G4:B4), 16-bit (R5:G6:B5), 18-bit (R6:G6:B6), and 24-bit (R8:G8:B8). Color depth is often referred to in terms of colors. For example, 24-bit (R8:G8:B8) is =16 Million colors. An MCU used in a handheld device normally has a dedicated bus to transfer display data from system memory to the LCD module. The width of this bus must be able to accommodate the maximum color depth the LCD is designed to support.

The LCD panel has a three-dimensional programming model (see Figure 1.1). Two of these dimensions are the X and Y address of a particular pixel on the LCD, and the third dimension is color depth of the pixel.

In order for a GUI designer to build an application that uses a handheld device's LCD display, a software graphic driver and a hardware LCD controller (LCDC) are required. The software graphic driver maps the 2-D panel locations to linear memory addresses. The LCD controller is a hardware module used to map linear memory locations to 3-D data matrices by constantly accessing data from the frame buffer and sending it to the LCD panel.

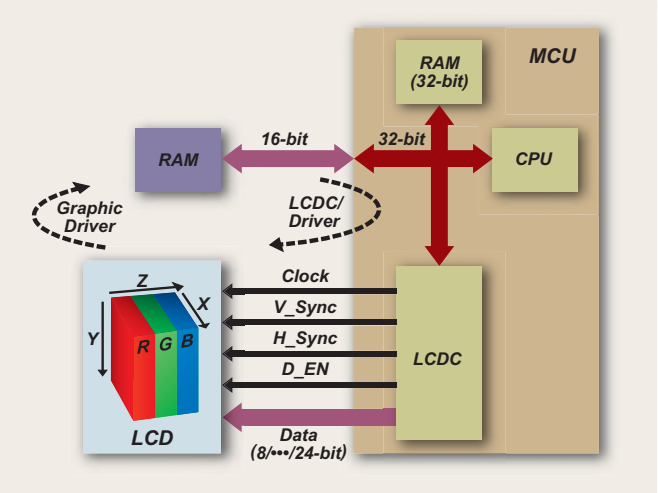

Figure 1.1. LCD system

2. Touch Screen

Touch screen technology addresses the conflicting demands for smaller device size and larger display size by eliminating the need for traditional user interface buttons.

The three most common touch screen technologies are resistive, capacitive and surface acoustic wave. The resistive touch screen technology is the least expensive of the three and provides good reliability, durability, and accuracy. Accordingly, it is a popular touch screen technology for use in handheld devices.

A resistive touch screen is made up of several layers, the most important of which are two thin, metallic, electrically conductive and resistive layers separated by a thin space. When an object touches a resistive touch panel, these two layers touch at the point of contact. The panel then electrically acts similar to two voltage dividers, and the output from the dividers can locate a position (X and Y) on a surface plane. (See Figure 2.1)

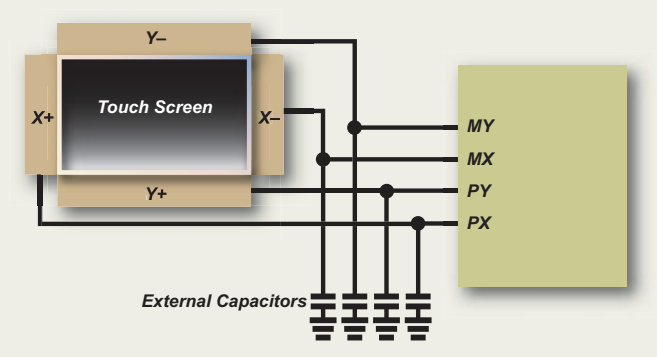

Figure 2.1. Touch-screen basic physics

The touch detection procedure involves two steps: the detection of a touch on the panel (an interrupt to the MCU), followed by the measured position (X/Y) of that touch. To perform these two steps, one of the four pins (PX) is configured as a digital input to generate the interrupt. Each generated X or Y voltage (MX, MY) provides input to an ADC to indicate relative position, while another (PY) becomes either V_{CC}, grounded or open. Figure 2.2 is a flowchart that illustrates this operation.

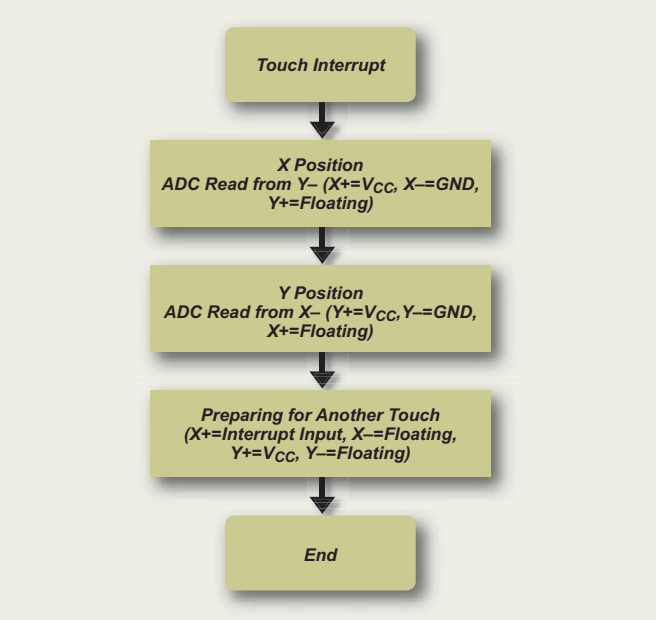

Figure 2.2. Touch-screen operation

3. NAND Controller and Driver

NAND flash memory was introduced by Toshiba in 1998 and is now widely used in handheld devices for on-board storage. Like a hard disk drive, NAND memory uses a block memory management system. Each block consists of a number of pages. The pages are typically 512, 2048 or 4096 bytes in size, plus a few extra bytes (typically 12 to 16) to store error correction code (ECC). The page is a read/write unit, while erase will affect an entire block.

A NAND flash controller (NDFC) generates the control signals required to interface with NAND flash memory, and also includes the ECC calculating circuits. Figure 3.1 illustrates a connection between an MCU and two NAND flash memory chips.

A NAND controller provides registers as interfaces to give programmers access to the NAND flash memory. A typical set of registers is listed in Table 3-1.

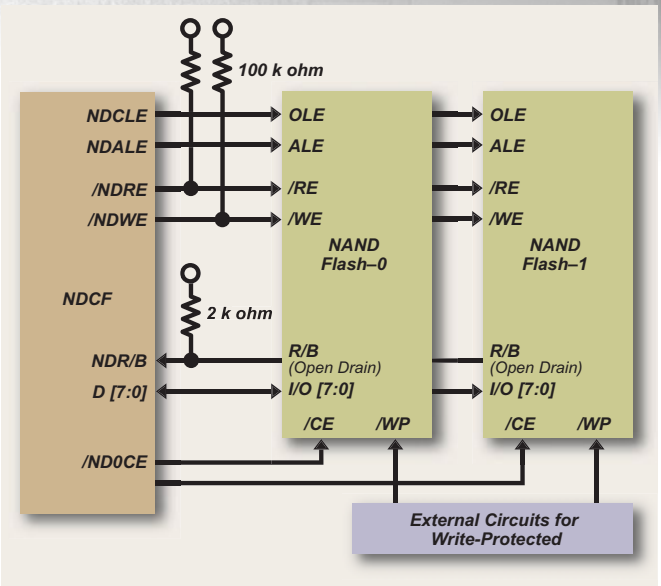

Figure 3.1. Block Diagram of NAND application

Table 3-1 NAND flash registers

Register Name
NAND Flash Data Transfer Register
NAND Flash ECC-code Read Register
NAND Flash Mode Control Register
NAND Flash Status Register
NAND Flash Interrupt Status Register
NAND Flash Interrupt Mask Register
NAND Flash Strobe Pulse with Register
NAND Flash Reset Register

Commands are sent through the control register to perform the four basic operations NAND flash is capable of: initialize, read ID, read (pages), write (pages), and erase (blocks).

4. Removable Storage

Removable storage devices give users the ability to expand on-board storage space, back up data and exchange information. Storage space in the device is divided into linear-addressed blocks (521 bytes per block). The SD card is one of the more popular removable storage media on the market, and will serve as the storage media being used in the following example. A file system is needed to organize user data, and more importantly, to allow information stored on the removable media to be exchanged with different devices. File Allocation Table (FAT) is the most popular file system used in handheld devices.

4a. File system

Figure 4.1. shows a file system consisting of 2 modules:

- API module
- FAT file system module

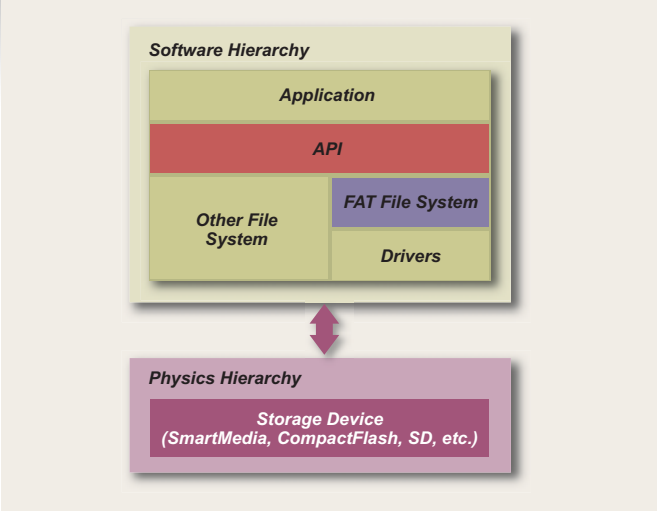

Figure 4.1. System Structure Diagram

When accepting other file systems than FAT, a file operation is possible with a common API by creating a file system in accordance with the lower interface specifications of API module.

4b. SD Controller and Driver

There are three transfer modes supported by SD: SPI (Serial Peripheral Interface) mode (separate serial in and serial out), one-bit SD mode (separate command and data channels and a proprietary transfer format), and four-bit SD mode (uses extra pins plus some reassigned pins) to support four bit wide parallel transfers. SPI is the commonly used interfaces between an MCU and external interface devices. Figure 4.2 shows the hardware connection between an MCU and SD card.

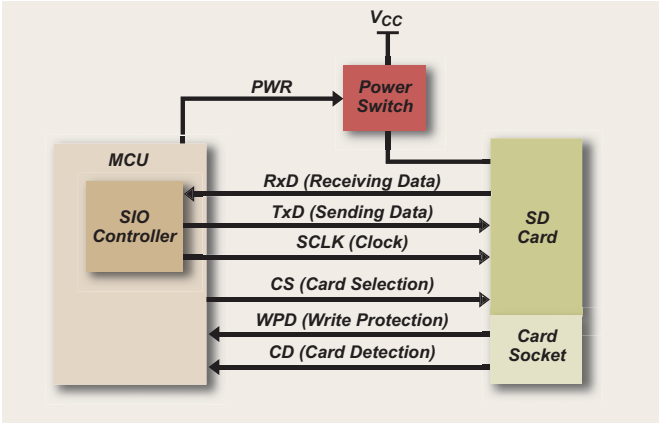

Figure 4.2. SD storage application block diagram

Here is an example of the kinds of APIs an SD driver can provide:

Table 4-2 SD driver APIs

API Name	Description
SDInit	Initializes a driver/host.
SDReadSector	Reads data by sector.
SDWriteSector	Writes data by sector.
SDGetCardSize	Gets a card size.
SDCardOpen	Makes an SD card condition possible to be used.
SDCardClose	Stops an SD card use.
SDGetStatus	Gets status information.

An application program can detect an SD card being taken out or put in a handheld device by using the SDCardOpen and SDCardClose functions.

The operation example below shows a driver used in combination with a file system.

Since the driver functions as a lower module of the file system, the application does not directly access the driver in the read/write operation. An API of the driver will be called from the file system.

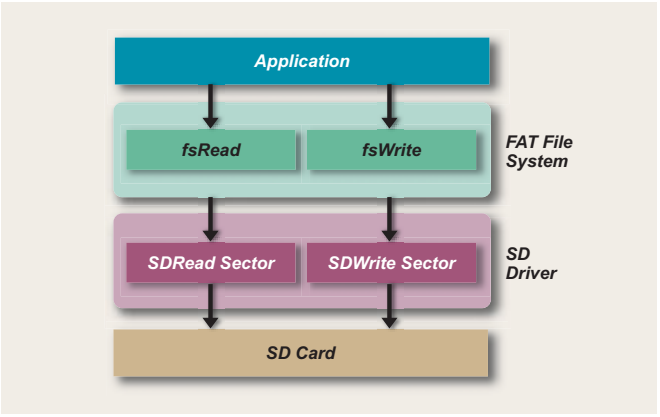

Figure 4.3. Sample operation of a file system and storage device

Reading Flow

1. Calls a file system from an application to read SD card data and opens the file.
2. Calls an SD card driver from a file system and reads SD card data.

Writing Flow

1. Calls a system file from an application to write data in an SD card, and opens the file.
2. Calls an SD card driver from a file system and writes data in an SD card sector.

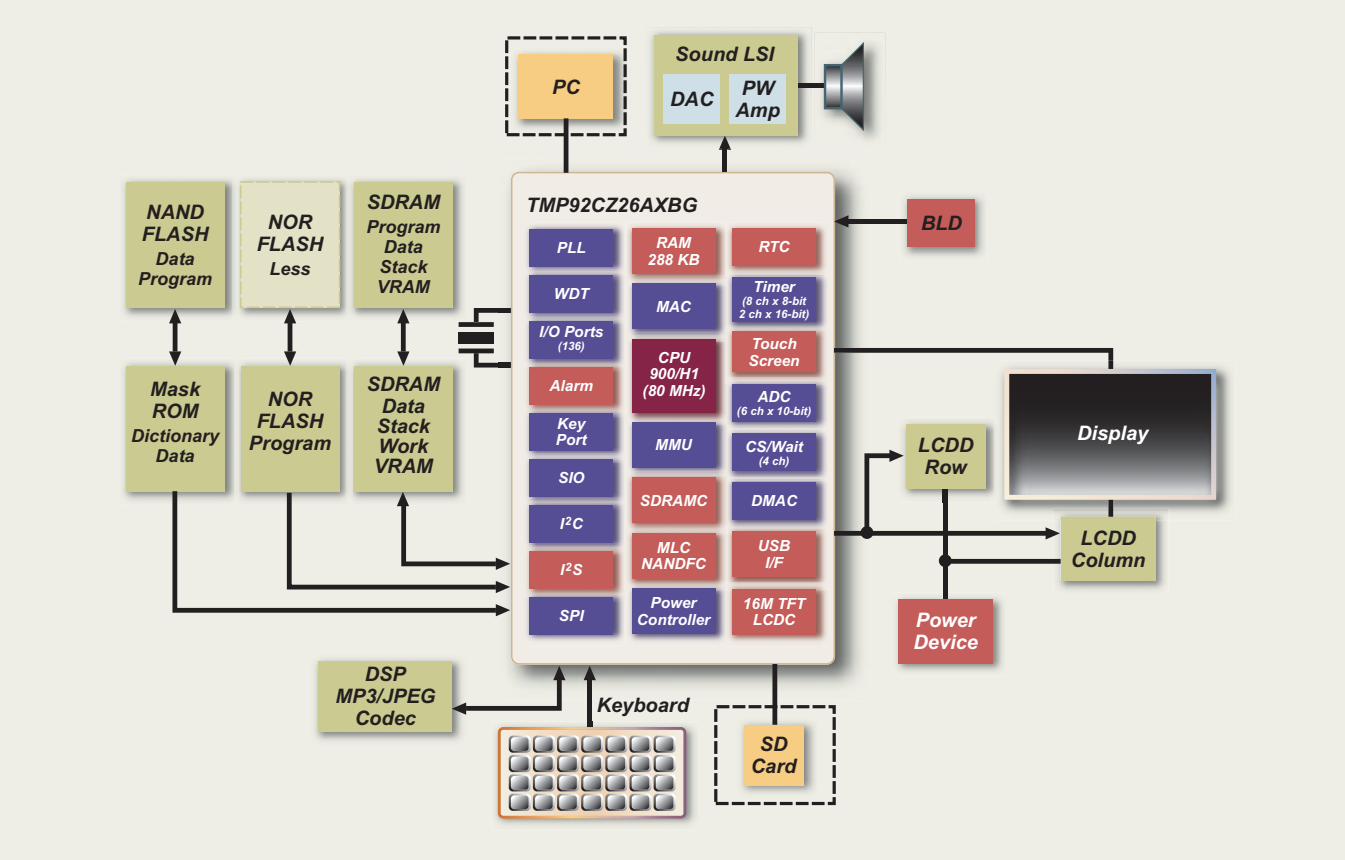

Figure 5.1. Hardware configuration of an e-dictionary based on the Toshiba TMP92CZ26 MCU.

5. Example

To better illustrate many of the concepts described in this paper, here is a sample system-level solution for a portable e-dictionary based on the Toshiba TMP92CZ26 MCU. The system software is based on open-source software, µC/OS-II and Nano-X Window System.

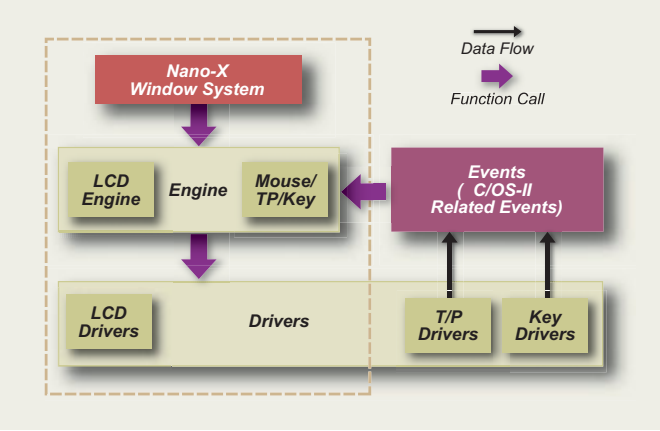

Figure 5.2. Software architecture of the e-dictionary

6. Conclusion

By choosing an MCU that supports the peripherals common to most handheld devices, a device designer can enjoy significant savings in development time and cost. These savings can be improved on if the selected MCU's vendor can provide a designer with a wide selection of software IP to leverage in their design, as well as a hardware reference design.

TAEC Sales Offices

Toshiba America Electronic Components, Inc.

Headquarters

19900 MacArthur Boulevard
Suite 400
Irvine, CA 92612
TEL: 949-623-2900
FAX: 949-474-1330

Toshiba America Electronic Components, Inc.

2590 Orchard Parkway
San Jose, CA 95131
TEL: 408-526-2400
FAX: 408-526-2410

5959 Gateway West,
Suite 405
El Paso, TX 79925
TEL: 915-771-8156

2150 E. Lake Cook Road,
Suite 310
Buffalo Grove, IL 60089
TEL: 847-484-2400
FAX: 847-541-7287

48679 Alpha Drive #120
Wixom, MI 48393
TEL: 248-347-2607
FAX: 248-347-2602

959 Route 46 East
Parsippany, NJ 07054
TEL: 973-541-4715
FAX: 973-541-4716

290 Donald Lynch Blvd.,
Suite 201
Marlborough, MA 01752
TEL: 508-481-0034
FAX: 508-481-8828

3700 Crestwood Parkway,
Suite 1070
Duluth, GA 30096
TEL: 770-931-3363
FAX: 770-931-7602

Toshiba is continuously working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing Toshiba products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a Toshiba product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that Toshiba products are used within specified operating ranges as set forth in the most recent product specifications. Also, please keep in mind the precautions and conditions set forth in the Toshiba Semiconductor Reliability Handbook.
The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by Toshiba for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Toshiba or others. The information herein is subject to change without notice.

www.Toshiba.com/taec

TOSHIBA
Leading Innovation

All trademarks are of their respective manufacturer and may be registered in certain jurisdictions.

© Copyright 1/2009 TAEC