White Paper

Implementing a Cost-Effective Human-Machine Interface for Home Appliances

Liquid crystal displays (LCDs) with highly interactive graphical user interfaces (GUIs) are replacing the traditional "mechanical" human-machine interfaces (HMIs) currently found on most home appliances. This paper explores these technologies, as well as Altera, Altia, and Echelon's innovative, total-solution architecture for delivering low-cost, high-performance HMIs on virtually any home appliance or consumer device.

Introduction

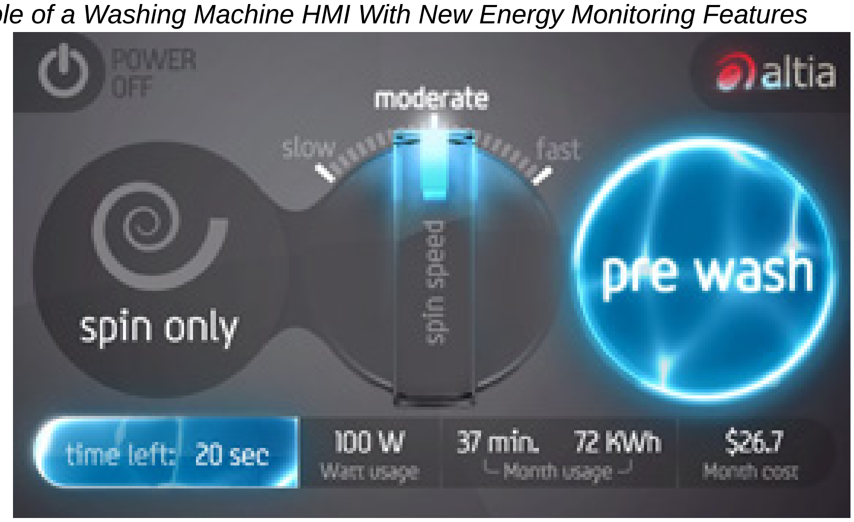
Traditionally, human-machine interfaces (HMIs) for home appliances have been composed of mechanical devices such as buttons and knobs, coupled with display indicators such as light-emitting diodes (LEDs) and vacuum fluorescent displays (VFDs). Today there is a massive transformation occurring throughout the home appliance and consumer device markets. As the cost of liquid crystal displays (LCDs) drops due to the proliferation of LCD technology in televisions, computers, automobiles, and mobile devices, LCDs with highly interactive graphical user interfaces (GUIs) are being deployed as a cost-effective replacement for the traditional "mechanical" HMIs currently found on most home appliances.

The Role of HMIs in Home Appliances

Displays, particularly digital display HMIs, play an increasingly important role as consumers seek easier ways to interact with the myriad of devices in their homes. The trend towards graphical interactivity in devices has been further accelerated by devices such as the Apple iPhone. The iPhone has fundamentally redefined what consumers expect when interacting with a mass-market device. As a result, today's consumer is comfortable touching virtual buttons, sliding their finger across a screen to adjust settings, and even using physical gestures in place of complex HMI actions. In addition, they expect the devices to be "smart" and to deliver unprecedented amounts of information easily, without error messages, and in many cases, without the consumer opening a manual.

In today's competitive consumer device and home appliance markets, manufacturers must add product differentiation at minimal cost to attract buyers and subsequently increase market share. Typically, a new technology begins by being added to the top tier of products, migrates down to the mid-range products, and eventually becomes a standard feature in almost all products. Some recently added technologies include an integrated HDTV for refrigerators, networked appliances, and intelligent energy solutions for washers and dryers (Figure 1).

Figure 1. Example of a Washing Machine HMI With New Energy Monitoring Features



These features all have one thing in common: they require a more advanced display technology to facilitate the human-machine interaction. In many cases, this is accomplished through a touch-screen display. Touch-screen displays enable OEMs to produce modern, sleek appliance controls, setting themselves apart from other brands and meeting current consumer expectations. Using digital display technology in place of mechanical HMIs ultimately leads to more rapid development, increased reuse of componentized architectures, and, over the long term, more economical designs due to advancements in GUI software and programmable logic device (PLD) technology. Touch-screen displays on home appliances are a natural evolution from today's consumer devices and the continuing drive of the "digital lifestyle" market trend.

Touch-Screen Technology

A touch screen is a display that can detect the presence and location of a "touch" within the physical constrains of the display area. Usually this is accomplished by the use of a person's finger or a plastic stylus. More recent implementations of touch technology use "multi-touch" sensing technology, whereby a single person uses two fingers to manipulate an object as on an Apple iPhone or multiple people can collaboratively interact on a single screen as with Microsoft's Surface Computing technology.

Invented in the late 1960s, touch technology was first deployed in computer-assisted learning terminals for corporate research labs and in commercial kiosk systems. However, it did not achieve mass-market popularity until the introduction of recent products, primarily in the mobile device markets. Consumers first embraced this technology in personal digital assistants (PDAs) such as the Palm Pilot, but it was not until the arrival of the Apple iPhone, with its seamless blend of touch interactivity combined with compelling, dynamic graphics, that the demand for this technology really began to escalate.

A touch screen primarily consists of three integrated components: a sensor, a module, and a display. The touch-screen sensor is the key component of the device, it allows the position of a finger or stylus touch to be identified and communicated to the underlying system. The touch-screen module includes the touch-screen sensor, a controller IC, and software. The touch-screen display consists of the module integrated with a display panel.

There are various types of touch-screen sensors in use today, including strain gauge, optical imaging, dispersive signal, acoustic pulse, surface acoustic wave, capacitive, and resistive sensors. Resistive sensors are the most widely deployed due to their low cost, with more than two-thirds of touch-screen component manufacturers producing resistive sensor-based touch screens. Resistive touch-screen displays also are not affected by external elements such as water, light, or dust while delivering high-resolution images and accurate small-target activation. Resistive touch screens use a controller on glass coated with indium tin oxide (ITO) or a plastic film overlay on the display surface to produce the touch connection. To activate the touch screen, finger or stylus pressure closes the air gap between the plastic film and the underlying substrate, thus producing a voltage change on the ITO film and glass. The touch position is calculated by the controller IC to determine the user's exact position on the sensor, which then is used by the underlying application software to determine the user's intent and perform the appropriate action.

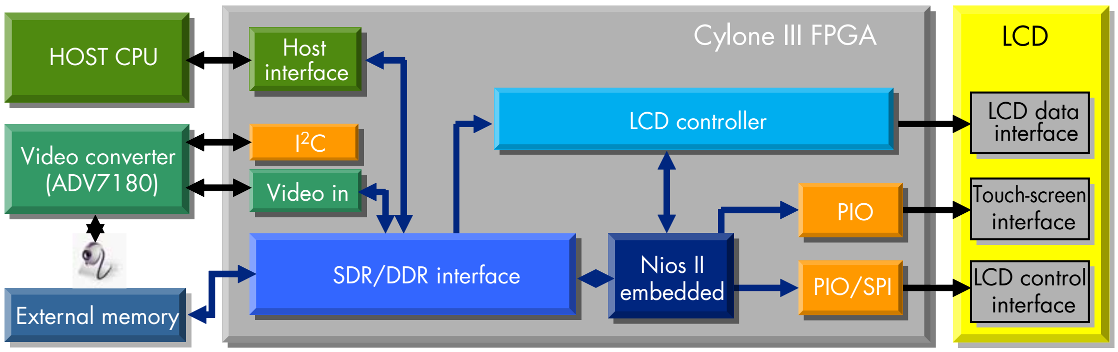
Touch-Screen Implementation With FPGAs

The trend towards highly interactive GUIs is driving a general requirement for more powerful device processors. Unfortunately, many low-cost microcontroller units (MCUs) on the market today do not meet the meet the performance characteristics necessary to deliver a compelling interactive, animated GUI on an LCD screen. In addition, existing MCUs often do not include on-chip support for peripherals, graphics acceleration, or LCD displays, making the total package a very expensive solution when assembled out of individual discreet components.

FPGAs, devices composed of an array of logic cells that can be configured to perform a variety of functions, make a better choice for LCD touch-screen display implementation than MCUs due their higher performance and flexibility. When combined with an embedded soft processor core, FPGAs easily support both the MCU general-processing functions and the functionality of other external devices. These devices offer unprecedented scalability, as they can be adapted dynamically for different screen sizes, image resolutions, peripherals, and GUIs.

Due to their programmability, FPGAs primarily were used to verify design concepts and build initial product prototypes. However, due to semiconductor submicron technology advancements, low-cost FPGAs, such as the Altera® Cyclone® product series, have been widely deployed in million of units of consumer electronic products, such as digital televisions, set-top boxes and DVD recorders. The further penetration of FPGAs into home appliance designs is a natural progression of this trend. Figure 2 shows a 65-nm low-power, low-cost Cyclone III FPGA enabling a LCD touch-screen display. With a built-in LCD controller and touch-screen interface, the Cyclone III FPGA helps eliminate the additional LCD controller and graphics processor that would otherwise be needed to implement such a design.

Figure 2. Touch-Screen Display Implementation With Cyclone III FPGA

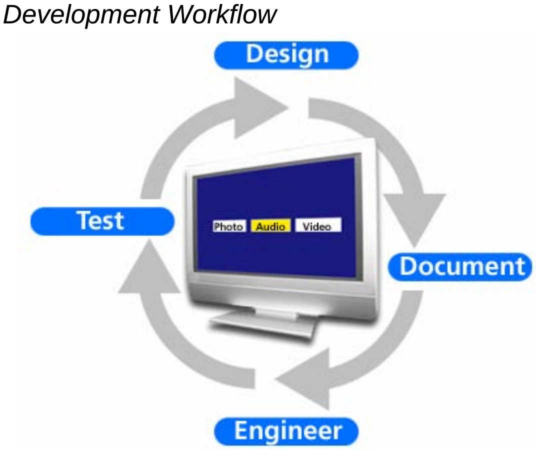


Using Next-Generation Technology to Make GUI Development Easier

Traditional GUI application development can be remarkably error prone and labor intensive, with OEMs claiming as much as 70 percent of quality issues due to the GUI implementation. These deficiencies can be attributed to inefficiencies in the workflow process, trade-offs made due to time and cost pressures, and the complexities of developing truly high-performance graphics software on an embedded system.

Typically, the GUI design and application is created by a design team that is separate from the software engineering team. The design document is passed to the software engineering team, which attempts to meet these requirements using handwritten code. The result then is passed to quality assurance, to compare against the original design specification. Deviations are identified as bugs and returned to the software engineering team to be addressed in the next software build. This development/quality assurance process (Figure 3) cycles until a stable, acceptable match between the software and design document is achieved. However, so many trade-offs may have been made that the resulting application and GUI do not match the original design specification. Due to time, cost pressures, or actual platform performance characteristics, it usually is not possible to address the remaining deficiencies, so they simply are accepted as known issues and the software is released as is.

Figure 3. Traditional Software Development Workflow



Further changes attempted by the design team for even small details such as text localization cause software engineering changes, and thus have the potential to bring instability into the system. For each GUI modification, big or small, at least one complete cycle through design, software engineering, and test is necessary to ensure a quality result and to guard against the unexpected complications of changing code. After the initial version is complete, even producing different versions with minor changes requires significant testing. The inevitable move to the next lower-cost hardware platform means that the software engineering team will repeat this entire process for a new set of software application program interfaces (APIs), discarding all the work in the previous product. As a result, much time is spent creating derivative products and, in many cases, the consistency of the GUI look-and-feel is compromised or lost.

While it is not possible to eliminate all the inefficiencies of the development process, there are now ways to use tools and packaged technology solutions to shorten the length of these repetitive cycles, achieve higher and more predictable graphics performance from an embedded systems design, and future-proof the design to port more easily between hardware generations.

Approaches to GUI Development

There are multiple approaches to GUI development. The most common approach, hand-written code, is also the most expensive method to productize a graphical application. It requires longer development times, minor GUI changes are cumbersome, and the code must be rewritten nearly from scratch for each device implementation. The second approach, a code generator tool, provides an easy-to-use interface to build the GUI and define behavior, but generates generic code that requires significant handwritten modifications to enable it to run and perform well. The third approach utilizes a secondary scripting language and an interpretive engine to process the script at run time, requiring large amounts of platform resources that are prone to severe performance issues and errors.

The fourth approach, called the binary GUI approach, combines a professional GUI building tool, a suite of powerful software APIs, some hand written code, and a high-performance embedded graphics engine. A PC-based tool enables the designer to create a pixel-perfect GUI and output it to a single binary data file. The exact pixel-for-pixel design then is passed directly from the designer’s desk to the developer, with no additional translation necessary. The developer then uses an advanced high-performance GUI engine and the associated APIs to manipulate and display the graphical data stored in the binary data file. Not only does this approach consume very few platform resources, but it allows further changes to the GUI to be made, with no or very few changes to code or logic. Furthermore, the applications can be easily modified and reused from model to model and year to year, even across different hardware designs.

“Energy Aware” Appliance Platform

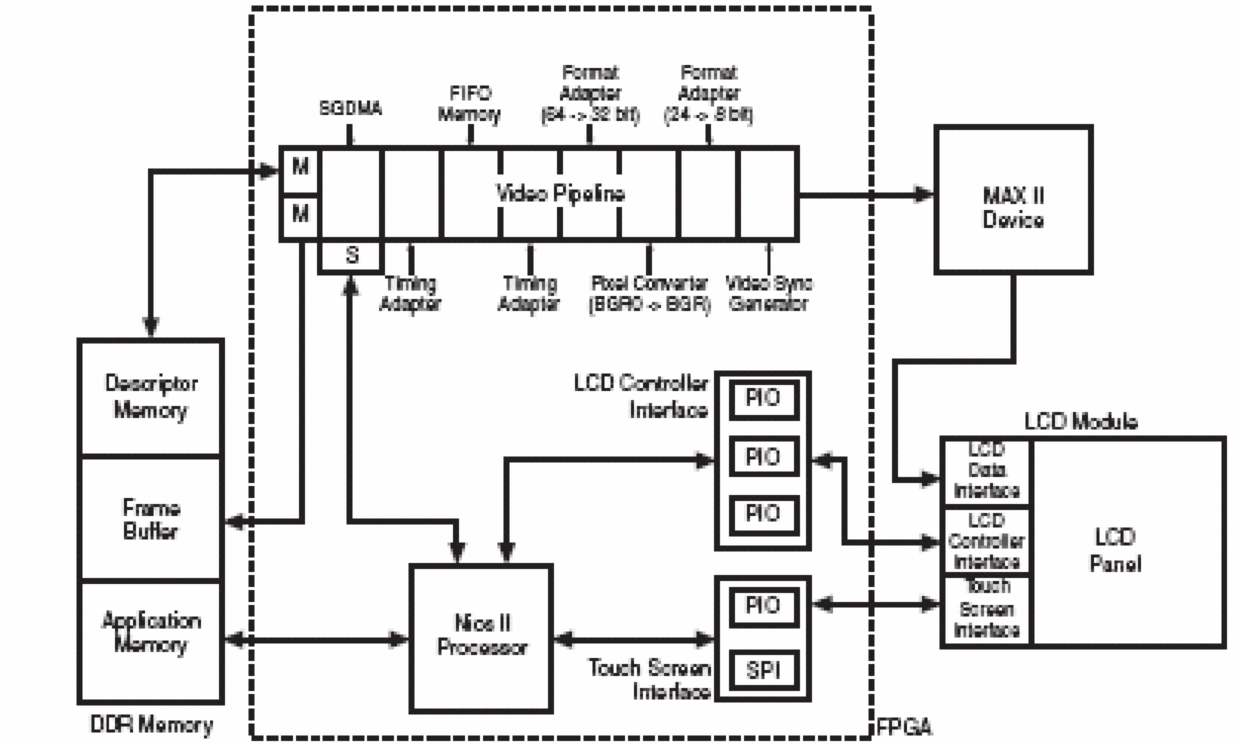
To help manufacturers familiarize themselves with the advantages of using a FPGA-based HMI solution for home appliances, Altera, Echelon, and Altia introduced the Energy Aware appliance platform, targeting the premium white-goods market. This fully integrated reference hardware platform consists of an Altera Nios® Embedded Evaluation Kit (NEEK) connected to an Echelon Power Line Smart Transceiver board, a 4.3" LCD touch-screen module, and a glass back that displays the interior of the kit.

The platform is powered by an Altera Cyclone III FPGA with an Altera Nios II embedded soft processor core. These devices together act as the host for all system software, including the Echelon ShortStack® API for power line networking and the Altia high-performance GUI engine and tools for HMI and LCD touch-screen control. With a built-in LCD controller, a touch screen interface, internal video pipeline, and integrated high-performance GUI engine and tools, the Cyclone III FPGA allows system designers to eliminate an external LCD driver and other graphics components, resulting in reduced bill of materials (BOM) cost and shorter time to market. This system design approach represents a significant advantage when navigating the transition from a mechanical HMI to the next generation of digital LCD touch-screen HMIs for the home-appliance market.

NEEK LCD Controller

Figure 4 provides a high-level, hierarchical view of the peripherals and interfaces that implement the NEEK LCD controller design. The video pipeline in Cyclone III FPGA, the LCD touch-screen module, and the MAX® II CPLD are the major components of the NEEK LCD controller.

Figure 4. NEEK LCD Controller Subsystem



Video Pipeline

The video pipeline is responsible for driving data signals on the LCD module data bus and for reading frame buffer data generated by the Nios II processor. A series of specialized Avalon® Streaming (ST) peripherals allows data units to be converted between buses with different widths, in this case, a 24-bit red, green, and blue (RGB) pixel input stream to an 8-bit pixel output stream in which each RGB color component is transmitted separately. The video sync generator peripheral transmits pixel data to the LCD touch-screen module by sequencing the control and data signals for the data bus of the module.

LCD Touch-Screen Module

The LCD touch-screen module consists of three major components:

- The LCD graphical data interface, which includes a 24-bit RGB data bus and some control signals, carries video data to the LCD module.
- The touch-screen interface consists of a Serial Peripheral Interface (SPI) and a parallel I/O (PIO) peripheral. The SPI communicates with the Analog Devices AD7843 touch-screen digitizer chip to signal “touching” events, a single PIO line captures interrupt events, and the Nios II processor runs software that drives both peripherals.
- The LCD controller interface configures the module by having the controller chip communicate protocol for sending and receiving data via a simple, proprietary three-wire interface of a general-purpose PIO peripheral. This peripheral is controlled by a Hardware Abstraction Layer (HAL) software driver running on the Nios II processor.

MAX II CPLD

The MAX II CPLD provides voltage translation between the Cyclone III FPGA and the 2.5V inputs and 3.3V outputs of the many peripherals to which it connects. It also serves as a color demultiplexer between the FPGA and the LCD module: it accepts the 8-bit time-division-multiplexed (TDM) stream from the FPGA and converts back to the 24-bit, parallel RGB format to be displayed on the LCD module.

Note that designers may eliminate the MAX II CPLD from their LCD controller designs if they do not need voltage translation or multiplexing /demultiplexing functions.

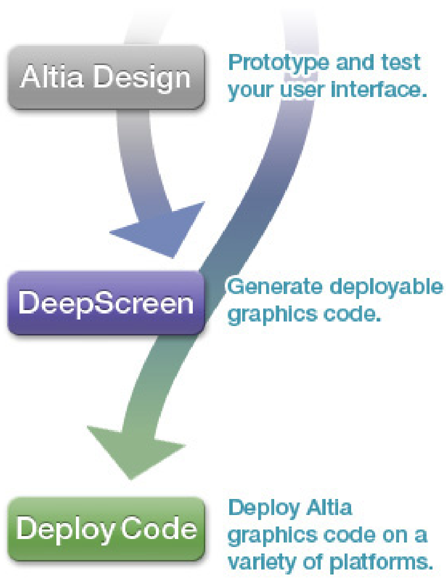
In addition to the NEEK hardware, Altera also provides the source code for touch screen software API, LCD module software API and video pipeline subsystem API. For additional details on the NEEK LCD controller and its hardware or software components, refer to *AN 527: Implementing an LCD Controller*.

Enabling Rapid GUI Development

Altia's suite of GUI development and code generation software delivers a cohesive and streamlined method for developing high-impact GUIs for next-generation home appliances. Prototypes can be created from artists' graphic assets or from a comprehensive collection of pre-built library components. The Altia Design prototype can be integrated to a simulation model and shared with development teams, managers, domain experts, and customers.

Once a GUI is complete, the prototype is turned into C code using Altia's DeepScreen. This application generates code for the FPGA by leveraging a Nios II processor for all graphical operations. As shown in Figure 5, the software tools enable developers to seamlessly create stunning color graphics comprised of a full set of rendering features—vector objects, bitmaps, text, alpha blending, and transformations like scaling and rotation.

Figure 5. Altia Design...DeepScreen...Deploy



The tools suite offers a number of advantages during GUI development. Altia Design provides users with the capability to use widely available graphics tools to create custom graphics. In this design environment, graphics can be created without the need for programming. Integrating a prototype with state and modeling tools offers development teams an easy way to demonstrate to customers and managers, allowing for valuable feedback early in development, thus preventing redesign due to misinterpretation of a written specification. The prototype serves as a clear and decisive method for communicating requirements for product behavior and functionality. Changes to the prototype can be made easily, saving time and the need for costly hardware mock-ups.

The DeepScreen code generator produces the same graphics that were used during development, so management and customers get the exact GUI on the final product that was approved during development. DeepScreen simplifies GUI code generation by generating graphics code in minutes rather than months. The graphics code can be deployed on different products and models, so manufacturers gain the cost advantage of creating a single GUI for an entire generation of products. This code can be deployed on both low- and high-power hardware.

Summary

Increasingly, graphical HMIs are a required element in consumer devices. Today, LCD touch screens are gaining traction rapidly in the home-appliance market. The cost of LCD screens is dramatically declining due to the popularity of devices like the Apple iPhone, and touch controls with animated GUI features are an expected high-end feature. The Energy Aware appliance platform, jointly developed by Altera, Altia, and Echelon, delivers a new, cost-effective approach to bring 21st-century HMI features to home appliances. This unique platform employs a suite of technologies targeted to reduce the total BOM cost and shorten overall development cycles by:

- *Eliminating external graphics devices:* Previously, developers needed to add an external LCD controller and graphics driver devices to MCU-based home-appliance designs to enable touch-screen display capabilities. With the Cyclone III FPGA and the Nios II embedded processor, external components can be incorporated into a single footprint, resulting in lower system BOM cost and higher design-integration flexibility.
- *Increasing silicon design flexibility:* Due to the nature of FPGA programmability, home-appliance developers can add new features, incorporate peripherals into the core silicon, and even address hardware quality issues without affecting the hard costs of production.
- *Reducing GUI development time and cost:* The Altia GUI development platform reduces the overall software development time and cost while dramatically changing the look-and-feel of an appliance GUI.
- *Enabling low-cost product line diversification:* Using skin technology, patented text rendering, and proven localization tools enables multi-brand and multi-model GUI differentiation from a single ROM image.

Further Information

Altera

- *AN 527: Implementing an LCD Controller*:
  www.altera.com/literature/an/an527.pdf
- *Energy-Aware Appliance Platform: A New Approach to Home Energy Control*:
  www.altera.com/literature/wp/wp-01077-energy-aware-appliance-platform.pdf
- *FPGAs Enable Energy-Efficient Motor Control in Next-Generation Smart Home Appliances:*
  www.altera.com/literature/wp/wp-01084-fpga-energy-efficient-home-appliance-motor-control.pdf

Altia

- Altia Design
  www.altia.com/products_design.php
- Altia DeepScreen:
  www.altia.com/products_ds.php

Echelon

- ShortStack API:
  www.echelon.com/products/development/shortstack
- Power Line Smart Transceivers:
  www.echelon.com/products/transceivers/powerline

Acknowledgements

- Audrey Brouwer, Technical Marketing Manager, Consumer Business Unit, Altera Corporation
- Jason Williamson, Director of Engineering, Altia Incorporated
- Steve Nguyen, Director Corporate Marketing, Echelon Corporation

101 Innovation Drive
San Jose, CA 95134
www.altera.com