Embedded COMPUTING DESIGN

www.embedded-computing.com

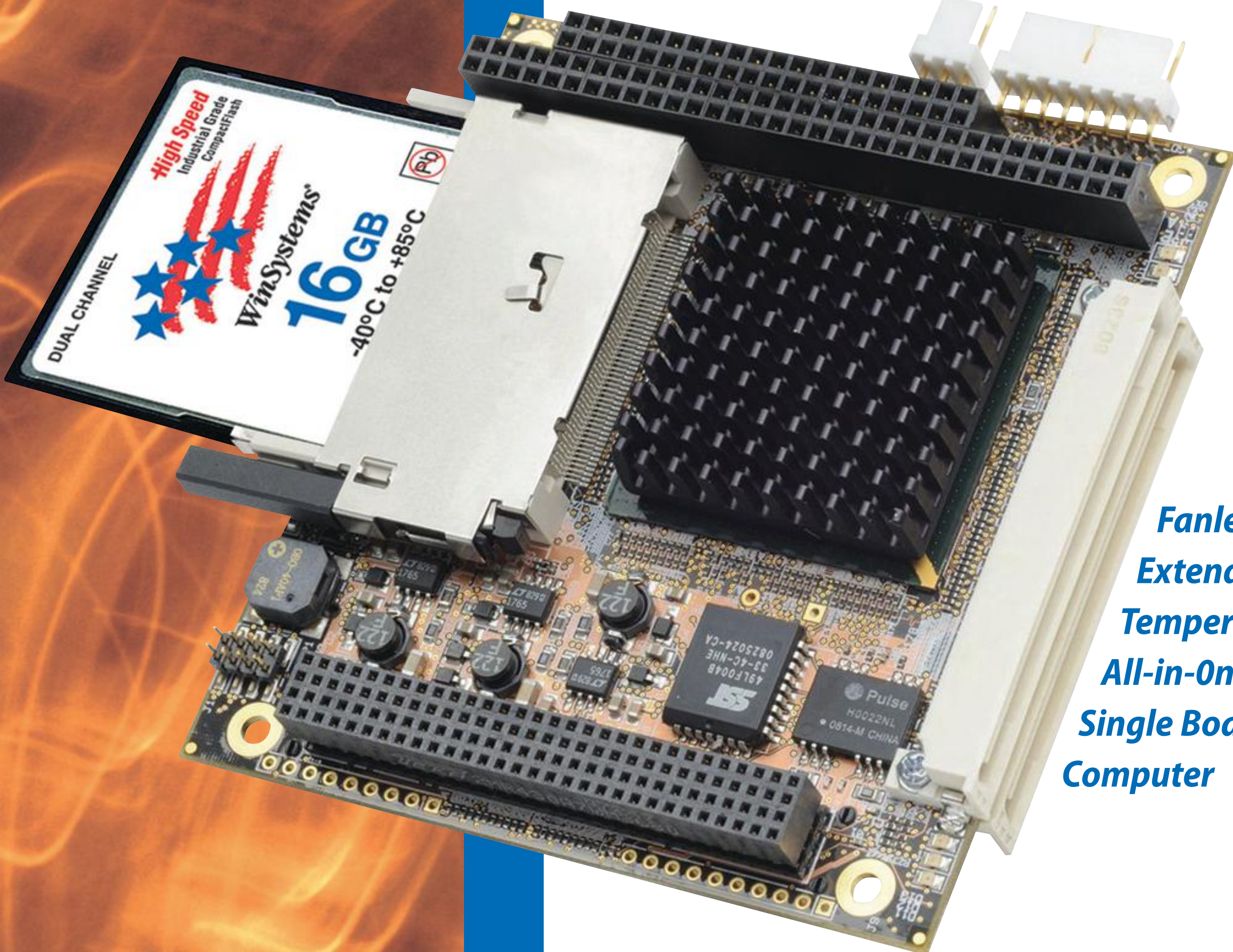
@embedded_mag
Embedded Computing Design

FEBRUARY 2011
VOLUME 9 • NUMBER 1

Software and standards: Keep 'em open

www.opensystemsmedia.com

PC/104-*Plus* SBC Supports Industrial Control, HMI, Networking and I/O



Fanless Extended Temperature All-in-One Single Board Computer

The PPM-LX800 SBC is available on a 6.5-inch Open Frame Panel PC.

If your application needs a compact, industrial PC with fanless -40°C to +85°C operation, check out WinSystems' PPM-LX800.

Designed for embedded, space-limited, and low-power applications requiring reliability and long-term availability, the PPM-LX800 runs Linux, Windows® XP embedded, and other x86 real-time operating systems. Plus it comes with a wealth of on-board I/O functions and free software drivers.

Features include:

- AMD LX800; x86-compatible CPU
- Compact size, 3.6" x 3.8" (90mm x 96mm)
- Socket for a bootable CompactFlash card
- High-resolution video controller provides
  - CRT resolutions up to 1920 x 1440
  - Panel resolutions up to 1600 x 1200
- Custom splash screen on start up
- 10/100 Mbps Intel Ethernet controller
- Four serial ports with FIFO
- Two USB 2.0 ports
- 16 Lines of digital I/O
- PS/2 keyboard and mouse controller plus LPT port
- +5V only operation
- Ultra DMA100 EIDE controller for 1 or 2 devices
- PC/104 and PC/104-*Plus* expansion connectors
- AC97 Audio with MIC, Line In, and Line Out
- -40°C to +85°C extended operational temperature
- Long-term product availability
- Responsive and knowledgeable technical support
- Quick Start Kits offered for easy software development

Contact our factory application engineers for additional product information, custom configurations, and pricing.

Call 817-274-7553 or
Visit www.winsystems.com/PPM-LX800

Ask about our 30-day product evaluation

WinSystems®
An Employee-Owned Company

715 Stadium Drive • Arlington, Texas 76011
Phone 817-274-7553 • FAX 817-548-1358
E-mail: info@winsystems.com

EMBEDDED PCS · GPS MODULES · COMPACT FLASH · SUMIT · PANEL PCS · I/O MODULES · ENCLOSURES · SBCS

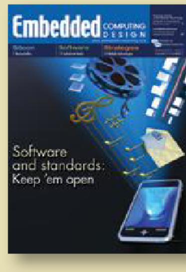
Embedded COMPUTING DESIGN

www.embedded-computing.com

February 2011 | Volume 9 • Number 1

On the cover

"Open source software and open standards are now broadly accepted in the embedded systems industry and should be considered on each new project," argues new *Embedded Computing Design* Editorial Director Warren Webb in his inaugural "Tracking Trends in Embedded Technology" column. Page 5.



ENVIROINK
The inks used to print the body of this publication contain a minimum of 20%, by weight, renewable resources.

2011 OpenSystems Media ® © 2011 Embedded Computing Design
All registered brands and trademarks within *Embedded Computing Design* magazine are the property of their respective owners.

ISSN: Print 1542-6408, Online: 1542-6459

Tracking Trends in Embedded Technology

By Warren Webb

@warrenwebb wwebb@opensystemsmedia.com

Software and standards: Keep 'em open

As embedded system performance expectations and data rates skyrocket, embedded design teams are turning to open source software and open standards to lower development costs and shorten the time to market. With potential cost savings on licensing fees, source code, development tools, and recurring royalties, designers and managers alike have become staunch open source fans. Even the often cited "lack of support" complaint has dwindled as in-house experience grows and a host of third-party experts offer their services. In fact, a growing industry has evolved around each new open source project to provide development teams with technical support, system integration, software training, documentation, and custom enhancements. Along with the software, embedded designers also rely on open standards to ensure an ample selection of pre-engineered, off-the-shelf hardware components to satisfy at least some of the requirements of each new project. These standards not only guarantee that hardware from multiple vendors will operate together, but also make it possible for designers to second source critical system components. Open source software and open standards are now broadly accepted in the embedded systems industry and should be considered on each new project.

In this issue of *Embedded Computing Design*, we present a close-up look at some widely used open source software and the associated development tools that may come in handy on your next project. For example, Vlad Buzov, a software architect at Mentor Embedded Professional Services, leads us through the steps necessary to port the popular Android operating system to applications other than mobile handsets. He covers all the details required to remove the phone software stack and add new device drivers dedicated to custom hardware. Continuing with the open software theme, Leigh Williamson of IBM Rational delves into the software development tools available for Android along with their features and limitations. Leigh describes how to combine the free Google Android Software Development Kit with a commercial product such as the IBM Rational Team Concert to enable collaborative development with task management and source control. Although Android is currently all the rage, Maciej Halasz, director of product management at Timesys Corporation, leads us through the relatively new MeeGo operating system for consumer electronics devices. MeeGo is Linux-based and targets portable or small form factor devices such as netbooks, handsets, in-vehicle infotainment, and connected TVs. Whether you are a first-time user or a veteran, these articles offer plenty of insight and tips that you can use as you tackle open source software development.

Although open source is a large part of this issue, we could not pass up the opportunity to present the latest mobile technology trends in our strategies section. Ed Chrumka, senior product manager at Nuance Communications, describes the multitude of communications, navigation, and infotainment services already available in the connected car and hints at the future. As states roll out laws banning driver distractions, Nuance speech recognition software has become popular in applications allowing drivers to verbally search the Web, dictate e-mail messages, update social networking sites, and navigate to street addresses or points of interest.

None of these software sensations would be of any use in a development project without the right embedded processor in place. To track the latest in microcontrollers, we covered an OpenSystems Media E-cast entitled "Take your Design to the Next Level with Intel Atom," where experts presented details on the first Intel architecture created specifically for embedded designs. Charlie Wu of Advantech explained the E6xx features such as low power, scalable performance, and flexible I/O, while Joe Broxson of Adeneo Embedded demonstrated the startup and customization options available with Windows Embedded Standard.

While the articles presented in this issue are timely and educational, they are only a small tip of a huge and growing iceberg called the embedded systems industry. I am very excited and honored to be the editorial director and work with the great team here at Embedded Computing Design. Our job will be to track the trends in this enormous industry and deliver timely news and information that may affect your design. However, in order to get this done, we need your help. Please give us your feedback on the information that we deliver and offer your suggestions. If you have an idea for a technical article that would be of interest to our community, please do not hesitate to send me a short article abstract.

Warren Webb, Editorial Director
wwebb@opensystemsmedia.com

Reader resources

Subscribe to magazines and E-letters:
www.opensystemsmedia.com/subscriptions

E-casts:
ecast.opensystemsmedia.com

White Papers:
whitepapers.opensystemsmedia.com

Videos:
videos.opensystemsmedia.com

Get news via RSS:
www.embedded-computing.com/news/rss.xml

Advertiser resources

Submit information:
submit.opensystemsmedia.com

Media kits:
www.opensystemsmedia.com/publications

Join the discussion

Comment on blogs, articles, news, and more

Twitter: @embedded_mag

LinkedIn: www.linkedin.com/groups?gid=1802681

Advertiser Information

Embedded COMPUTING DESIGN

OpenSystems media.

CompactPCI Advanced & Micro TCA | DSP-FPGA.com | INDUSTRIAL EMBEDDED SYSTEMS | Military EMBEDDED SYSTEMS | PC/104 and small form factors | VME and Critical Systems

Embedded Group

Warren Webb, Editorial Director
wwebb@opensystemsmedia.com

Don Dingee, Embedded Technologist
ddingee@opensystemsmedia.com

Jennifer Hesse, Assistant Managing Editor
jhesse@opensystemsmedia.com

Monique DeVoe, Web Content Editor

Hermann Strass, European Representative
hstrass@opensystemsmedia.com

Konrad Witte, Senior Web Developer

Matt Jones, Web Developer

Christine Capuano, Web/Editorial Assistant

Steph Sweet, Creative Director

David Diomede, Art Director

Joann Toth, Senior Designer

Phyllis Thompson, Circulation/Office Manager
subscriptions@opensystemsmedia.com

Reprints and PDFs

Nan Holliday
800-259-0470, republish@opensystemsmedia.com

Sales Group

Dennis Doyle, Senior Account Manager
ddoyle@opensystemsmedia.com

Tom Varcie, Senior Account Manager
tvarcie@opensystemsmedia.com

Rebecca Barker, Strategic Account Manager
rbarker@opensystemsmedia.com

Christine Long, Digital Content Manager
clong@opensystemsmedia.com

International Sales
Elvi Lee, Account Manager – Asia
elvi@aceforum.com.tw

Regional Sales Managers
Barbara Quinlan, Midwest/Southwest
bquinlan@opensystemsmedia.com

Denis Seger, Southern California
dseger@opensystemsmedia.com

Sydele Starr, Northern California
sstarr@opensystemsmedia.com

Ron Taylor, East Coast/Mid Atlantic
rtaylor@opensystemsmedia.com

Editorial/Business Office

16626 E. Avenue of the Fountains, Ste. 203 Fountain Hills, AZ 85268

Tel: 480-967-5581 ■ **Fax:** 480-837-64661 ■ **Website:** www.opensystemsmedia.com

Publishers: John Black, Michael Hopper, Wayne Kristoff

Vice President Editorial: Rosemary Kristoff

Vice President Marketing & Sales: Patrick Hopper
phopper@opensystemsmedia.com

Business Manager: Karen Layman

Embedded microcontroller update

By Warren Webb

Warren reports on a recent OpenSystems Media E-cast covering microcontroller architecture for embedded designs, "Take your Design to the Next Level with Intel Atom[1]." Curt Schwaderer, technology editor for OpenSystems Media, moderated. Speakers were Charlie Wu, product manager for Advantech's Embedded Single Board Computer Product Line and Joe Broxson, senior embedded engineer at Adeneo Embedded.

Curt opened the E-cast by pointing out the wide range of embedded systems in use today and the one thing they all have in common: the need for higher performance and shorter development times. In response to these needs, Intel has developed the Atom processor with a high performance level and flexible I/O while keeping thermal and board footprints within range for embedded requirements. Curt mentioned an ARC Advisory Group report naming the Intel Atom as a key market driver for improving the performance of future embedded systems.

Charlie Wu of Advantech then presented a detailed description of the Intel Atom processor E6xx series including features such as low power, scalable performance, flexible I/O, extended temperature, and a large software ecosystem. Charlie noted that the E6xx series is available in a wide range of off-the-shelf modules and industrial PCs to simplify and shorten embedded design projects. He also presented technical information on two new Advantech modules designed around the E6xx series: The SOM-6764 COM Express 95 x 95 mm module with type 2 pin-out for compact/rugged

Figure 1 | The SOM-6764 COM Express 95 x 95 mm module from Advantech is based on the Intel Atom E6xx series processor.

applications (Figure 1) and the PCM-9364 3.5-inch single board computer for vehicle, signage, and industrial control applications (Figure 2).

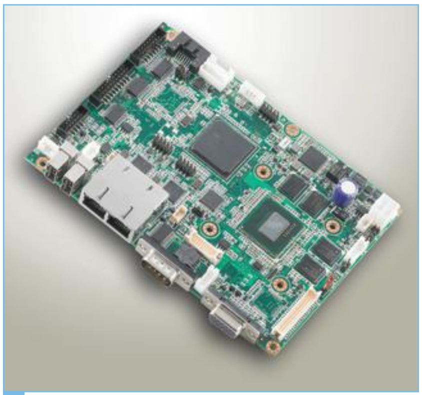


Figure 2 | Vehicle, signage, and industrial control applications are among those targeted by Advantech's PCM-9364 3.5 inch SBC, based on the Intel Atom E6xx processor.

Next, Curt introduced a pre-recorded video entitled "Getting to Market Quickly with Windows Embedded Standard 7 and the Intel Atom Processor E6xx Series" hosted by Joe Broxson of Adeneo Embedded on behalf of Microsoft. During the video presentation Joe demonstrated the startup and customization options available with Windows Embedded Standard. He went through the step-by-step process necessary to create a custom firmware image for a digital signage application. The Windows Embedded development toolset allows the designer to pick from predesigned templates or create a completely unique software configuration.

Joining the speakers for the Q&A was Joe Sunga, application manager for Advantech. Following are excerpts from that session:

CS: Are certain board form factors better or standard for certain embedded markets? For example, is there any advantage to using COM Express versus PC/104?

CW: There are a few different form factors that Advantech offers in single board computers such as the PC/104 in 3.5-inch and also 5.25-inch. The COM Express is more for custom design purposes. You can use COM Express for virtually anything that you want to design. We have seen vehicle computers, medical applications, and embedded signage applications using COM Express. The PC/104 in particular, or that interface connection, is more for traditional industrial control or ruggedized applications such as transportation. It is also very popular for machining applications.

CS: So if I needed a board for some kind of reference design, I could take a COM Express board from Advantech and use that for software development. Is that an option?

CW: Yes, that is an option. With the COM Express board, we also offer a standard carrier board that you can use to do your development on, and it will come with most of the standard PC I/O functions.

CS: OK, and is that how I would add my own I/O to the board if I need I/O interfaces that didn't show up on it?

CW: Once you have been developing your application software using the carrier development board then you can either start doing your own design using the carrier board that you like using the form factor and functionality, or Advantech will also help develop the carrier board for you if you [so] specified.

CS: What if I need 64-bit support for my embedded application? Is that available?

JB: Yes, it is available with Windows Embedded Standard 7. You just have to take the same precautions that you would with the desktop, meaning that you have to ensure that you have all of the correct 64-bit drivers. But once you have done that you can use both 64- and 32-bit applications on the Windows Embedded Standard system.

CS: With regard to the build environment do users have the option of activating specific devices?

JB: Here you are talking about Windows activation. When you buy a new version of Windows and install it on a computer, it asks you if you would like to activate using a key. With Windows Embedded Standard 7 there is no product activation. You are issued a key from Microsoft when you sign up for licensing and you use that key when you create your image. If you use that key when you create the image, there is nothing else to do. You simply create the image, deploy it, and then you work with a licensed distributor to purchase stickers that you apply to your devices. Those stickers are the license. You don't have to do activation like you would with the retail, off-the-shelf version of Windows.

CS: They say they make their devices for a regulated industry. Does the tool chain integrate with source control?

JB: Yes, it is very common that if you are in a regulated environment making devices that have to be certified for FDA or for other areas, you need to make sure that your systems are controlled as far as change control. With Windows Embedded Standard 7 it is very easy to do that. We have this thing called a distribution share, which is simply a directory that contains all the different components, or packages, which you can apply or use within your system as well as drivers or other things. That entire directory can be checked in as source control as well as your project files that define which different file packages or settings are included in your device. So it is extremely easy to do source control with Windows Embedded Standard 7. This is a large improvement over previous versions of Windows Embedded Standard.

CS: What is the status of the Windows 7 support for the Advantech boards? Are the drivers ready specifically for the E6xx series and your circuit boards and how much off-the-shelf software is there for the Advantech form factor?

JS: Based on Intel's roadmap, most of the drivers are available for Windows 7, but some of them are not including the graphics driver. That's currently still in beta stage, and what they are predicting is that everything will be ready early this year.

CS: There's a question about Bluetooth and its importance to things such as in-car applications. Does the Atom support that and are the drivers there, or does the developer have to worry about creating drivers to enable the desired I/O?

JS: Atom does not support Bluetooth, but it has a bunch of I/O expandabilities through the PCI Express bus where someone could actually use a Bluetooth adapter with PCI Express, and they could develop their drivers for that. Or if they are using an IOH there is also the option of using the UART. They could interface their Bluetooth through the serial bus, and the drivers would be inherent to the serial bus. I think there are also seven USB 2.0 ports on there, so you could also interface through USB.

CS: Another I/O capability is CANbus. What is that I/O capability for and where can developers find more information on CANbus?

CW: The CANbus has been used mostly in the vehicle applications. More recently we found that the CANbus has been used in medical applications as well. My understanding is that the CANbus is used for communications on specific measurements on vehicle sensors. We currently have PC/104 modules specifically to support CANbus. And of course the great thing with the Atom E6xx processor is that we can now start supporting CANbus directly on the board.

CS: What video and resolution and frame rates are planned to be ported onto the E6xx platform for both encoding and decoding?

JB: There is no limitation imposed by Windows 7.

JS: A display controller utilizes the Atom graphics engine and from there the controller will support 18-bit and 24-bit single-channel LVDS. The maximum resolution is 1280 x 768 at 60 Hz or you can utilize the SDVO output, which has a maximum resolution of 1280 x 1024 at 85 Hz. And if you are talking about frame rate for video encode, that can be done at up to 720p at 30 frames per second using the H.264 codec or MPEG-4. Other codecs can actually be done up to 1080p at 60 frames per second and 720p at 60 frames per second.

CS: Regarding the iManager and the SUSI interface from Advantech, can you give a flavor for the kind of service APIs that are available and what functions they offer? And what kinds of operating systems are available for it? Can you run Linux, Windows 7, and other embedded operating systems?

CW: First I would like to answer that the iManager is an embedded controller to monitor the system input voltage and other power management signals such as S3 and S4, and then accurately program and control the power-on sequence logic for the chipset. That makes the power sequencing more accurate and also makes the power-on more reliable. It also has additional features such as watchdog timer, hardware monitor, and GPIO controlled SMBus. The SUSI on the other hand is the software interface for the iManager at this point. The features include the monitoring portion (watchdog timer, hardware monitor, and hardware control). The control portion is SMBus,

TS-WIFIBOX-2

A Complete Solution for 802.11g WiFi Applications

qty 1 $185

Powered by a
250 MHz ARM9 CPU

- Low power (3.2 watts), fanless
- Power via 5-12 VDC, USB, PoE (opt.)
- 64MB DDR-RAM
- 256MB ultra-reliable XNAND drive
- Micro-SD Card slot
- RS-232, RS-485, CAN, RTC
- Header with SPI and 11 DIO
- 480Mbit/s USB, Ethernet, PoE option
- Boots Linux 2.6.24 in < 3 seconds
- Un-brickable, boots from SD or flash
- Customizable FPGA - 5K LUT
- Optional DIN mountable enclosure

Ideal for gateway or firewall, protocol converter, web server, WiFi audio, and unattended remote applications

- Over 25 years in business
- Never discontinued a product
- Engineers on Tech Support
- Open Source Vision
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products ship next day

Technologic SYSTEMS

We use our stuff.

visit our TS-7800 powered website at

www.embeddedARM.com

(480) 837-5200

I2C Bus, and GPIO. The display portion includes brightness control, on/off control, etc. Currently the operating system support that we have is Windows XP Embedded. Soon we will have Windows 7 or Windows 7 Embedded. It also supports Windows CE and Linux.

CS: Why should we use the Windows 7 Embedded operating system instead of the desktop Windows 7?

JB: The primary reason that people prefer the Windows 7 Embedded Standard is that it is broken down into these packages where you can only pick the portions of the OS that you need. You can actually create an operating system image that is as small as 600 megabytes. However if you are going through the OEM channel and using the standard OEM OS image it is upwards of 16 gigabytes because of the items that it includes. You get other options for customization such as the unbranded startup screen as well as other embedded enabling features. In addition, the licensing cost is typically lower for the Embedded Standard than it is for full OEM SKUs or the desktop retail SKUs. Benefits include the lack of activation. You don't have to activate each device. You just have to put your key in once and download the image to a thousand devices and you are ready to go. Another reason is that you don't have to use the Windows shell. There is no startup; it starts with what you want to see in your embedded application.

CS: In terms of things that users know and love from Windows such as the Winsock API and standard Windows APIs for graphics: Are those available on Windows Embedded, and how hard is it to take a Windows desktop application and port it to Windows Embedded?

JB: The difficulty in porting is minimal. The API set is identical. However the operating system is broken down into modules that have to be selected. So as long as you have all the dependencies that are required, then your application will just run. For example if you have an application the uses the .NET Framework3.5, you have to include that in your image.

You just load the template on top of that and go ahead and build it. Creating a custom template is no harder than creating a project.

CS: Current design requirements are for RAM on a single board computer. Does Advantech have an E6xx board with both RAM and Flash included on the SBC? If not, is there a way to get Flash through an I/O interface?

CW: Regarding the design requirement on the memory on the SBC, currently for the 2.5-inch single board computer and the PC/104 form factor computer we have CFast design, not on board, but through a CFast socket. At this moment we do not have a product that has both memory and Flash on board, but this is something that we can investigate.

CS: Are there default builder packages that you can plug into the Windows Embedded environment? So if I want an Advantech SBC part number xyz, I could load it into this builder and take in or out the components that I want? Or do I need to build this up myself?

JS: Depending on the Advantech model number and software, say for Windows Embedded Standard 7, we have a board support package that can be worked through, so you can get an idea of what kind of components the board utilizes. And from there if a custom design were needed, then you could basically depopulate the components you do not use. That board support package has everything for the components that are on the Advantech board. On some boards we have an expansion slot and if the designer adds something there, it would have to be added to the board support package.

JB: The way it works with Embedded Standard is that we have a tool that will analyze the target and identify what drivers are needed on it and so on. What you can do is take the output file from that and import it into the tool and then do your minimal configuration. Just the things you need to get the OS booted. There is a component called eCore that goes along with that. You have eCore and all the drivers required, and you save that as a template. You can actually save that project as a new template. And by doing that it gives you the ability to use that as a starting point. Let's say you were working with a specific board, then you would save that base model as a template and whenever you want to start on a new project you would just start up the tool and load up the template for that board. On top of that you can load the template for some sort of display device, or set-top box, or whatever you are trying to create. You just load the template on top of that and go ahead and build it. Creating a custom template is no harder than creating a project. You just save that project in the right place, and it becomes a new template.

CS: If you have a board with multiple configurations, how do you handle that in a template environment?

JB: There are multiple ways to handle that. Just ensure that all of the drivers for all the different components are available in the image. It is a bit of an advanced step as to how the phases work, but you would simply specify that all of those drivers would be added during this offline-servicing phase. All of the drivers are on the hard drive so that whenever the device comes up, any other components will be recognized during plug and play and then they will be available on the system. Then you have your base image with all these optional drivers that are also available on the system when you install that image. If the hardware is there it will automatically be installed by the Windows installer the first time the image boots.

CS: Can you give a few more details on how to create and install a custom splash screen for branding purposes?

JB: There are two pieces to it. The first is the standard boot screen and you can go with the Windows flowering screen if you like or you can switch it to a black screen. Also, a second screen comes up. Now that second screen is typically the log-on screen, and you have the ability to add a package to your image that allows you to exchange the background – and all the things seen there – with your own background. You would supply a PNG file that would become the background, and all the Windows messages would appear on top of that. You have a lot of control over that startup process and what you see there.

CS: What are the I/O hub options? Are there any third-party companion chips if they are needed?

JS: Intel developed the chip code-named Topcliff, which is essentially the EG20T. That is what Advantech utilizes for the I/O hub on this. The I/O has two SATA II ports, seven USB 2.0 ports, a Gigabit Ethernet, one CANbus, one I2C, and one SPI bus. It also has eight general purpose I/O ports, four UARTs and also options for two SD or MMC interfaces. Intel does not normally certify third-party I/O, but there are IOH vendors out there such as OKI Semiconductor, STMicroelectronics, and Realtek. To make things easier, there is no front side bus. It just uses a standard PCI Express bus, so you can install anything using that standard.

CS: In terms of the 1080 HD video, via the SDVO, is there any documentation available to support this design requirement?

CW: At this moment, I am not aware that there is a readily available document, however we do have reference designs where we take SDVO and we turn that into various signal outputs including LVDS, HDMI, and also DVI-D. So as long as the video output can support the 1080p and we have a reference design available to our customer who wants to make a custom design, we can provide the 1080 SDVO video support.

CS: How do you use the video encoder/decoder hardware on the board?

JS: I know for video decode, the application software must be compliant with GStreamer or MPlayer framework. The application must use the API those frameworks provide to utilize a video decoder. For video encoding, the application must also be compliant with the GStreamer framework.

JB: Windows 7 does provide the same encoding and decoding pipelines on the embedded as it does on the others. So as long as the device drivers and the codecs that take care of handling hardware acceleration are available for a piece of hardware, then Windows 7 can absolutely take advantage of it.

CS: Is Advantech doing any boards with Tunnel Creek only and no Topcliff or any other IOH solution?

CW: At this moment, for the single board computers, functional in PC/104 form factors, we don't have that offering because it is more general purpose and they will all come with Topcliff, the EG20T, which is the I/O hub. However, we do have the computer modules for the SOM product line in the works, and we will have a couple of products to support the Tunnel Creek only or E6xx series only without the Topcliff IOH. Basically the Atom E6xx series can stand alone, and IOH is usually required for lower-level functions. The IOH connects to storage solutions, the USB client functions, CANbus, I2C, and the general-purpose I/Os. If the customer is developing portable applications, they will usually want to incorporate the storage connectivity such as SD, SDIO, or the MMC card. It is not necessary to add an EG20T or a third-party I/O hub unless one of the functions mentioned is required for the application. **ECD**

[1] Register to view the archived E-cast, Take your Design to the Next Level with Intel Atom with slides and audio at http://bit.ly/fHQvdw.

***Warren Webb**, an Editorial Director at OpenSystems Media, is an internationally known industry analyst and editor. He is the former technical editor of* EDN *magazine. He worked on the magazine's Embedded Webblog commenting on board-level embedded hardware, development tools, and software. Additionally, Warren wrote in-depth technical articles on hardware design, software development, and emerging technologies.*

ECD in 2D: See how Intel's E6xx processor series can elevate your designs in a recent E-cast. Use your smartphone, scan this code, watch a video: http://bit.ly/fHQvdw

The benefits of developing Android applications using commercial Eclipse-based solutions

By Leigh Williamson

While the Android SDK provides a great starting point for an individual developer of Android code, it is missing features that facilitate the collaboration and coordination needed when a team is developing an Android application. By integrating the device-specific, native platform SDK with a compatible commercial development solution, agile teams can achieve tremendous efficiencies and higher-quality results.

The Android Software Development Kit (SDK), which Google provides for free, is a great starting point for developing an Android-based smart device application. The SDK contains a variety of useful materials for developers, including extensive documentation, tutorials, samples, best practice guidance, and an array of tools for numerous development purposes.

The SDK's set of Java APIs gives application developers access to native functions that Android-based devices support, such as 2D and 3D graphics, multimedia codecs, telephony features, and location services. A device emulator in the SDK allows developers to try out their code directly from the development environment without requiring a physical device. And the SDK has an Eclipse plug-in that exposes the Android APIs and SDK tools in a rich Integrated Development Environment (IDE).

Opening the door to collaboration

For an individual developer of Android code, the SDK is valuable and is becoming more so as it is being extended with new features all the time. However, it is missing features that facilitate the collaboration and coordination needed when a development team is creating the application.

By integrating the device-specific, native platform SDK from Google with a compatible commercial development solution, agile teams can achieve tremendous efficiencies and better results. Integrating the native Android SDK with a commercial development environment opens the door to seamless source control, iterative application planning,

effortless work item management, and a host of enterprise-quality development capabilities for an Android application.

For instance, many Android applications are structured as hybrid Web applications, where part of the application runs on an application server on the network delivering data to the device from an enterprise storage system, perhaps a mainframe computer. Another part of the hybrid application runs on the device itself, displaying the data it receives across the network and formatting it for the device form factor while accessing the device's services such as GPS, camera, and accelerometer to deliver a rich and well-performing user experience.

Such a hybrid application is typically created by a small team comprising a few developers of the fundamental business logic and Web application components, a few User Interface (UI) developers, a user experience designer, a couple of testers, and a team leader or manager. Let's consider how this team can leverage the Android SDK in an environment that allows each member to efficiently communicate and collaborate.

The integrated Eclipse environment

The Android SDK, or more precisely, the Android Development Tools Eclipse plug-in that is part of the SDK, can be combined with an Eclipse-based commercial collaborative development product such as IBM Rational Team Concert (RTC). There is much information available that documents how to get the Android SDK and RTC working within the same Eclipse "shell." See Figure 1 for an illustration of what the Android SDK looks like when integrated with RTC.

The commercial IDE (RTC) offers integrated work item management, which allows the team leader to define work for the Android UI developers and assign those tasks to them, separately from the work assigned to the application logic developers and the other team members, including work assigned to the testers. The code changes associated with a particular work item are tied together into a specific change set that is delivered in one shot, so the full code change can be tracked as a unit. As the developers edit files inside their IDE, the change set is automatically maintained. The developers don't have to do anything special to produce the change set other than edit the files they need to work on.

The change sets can be shared amongst members of the team before being fully integrated with the main code stream. So a change set altering the format of data supplied by the Web application can be shared with the UI developer working on the logic that displays the new data without affecting the rest of the team. Once both UI code changes and Web app code changes are deemed ready, they can be integrated in one synchronized task into the mainline code stream for the rest of the team to use.

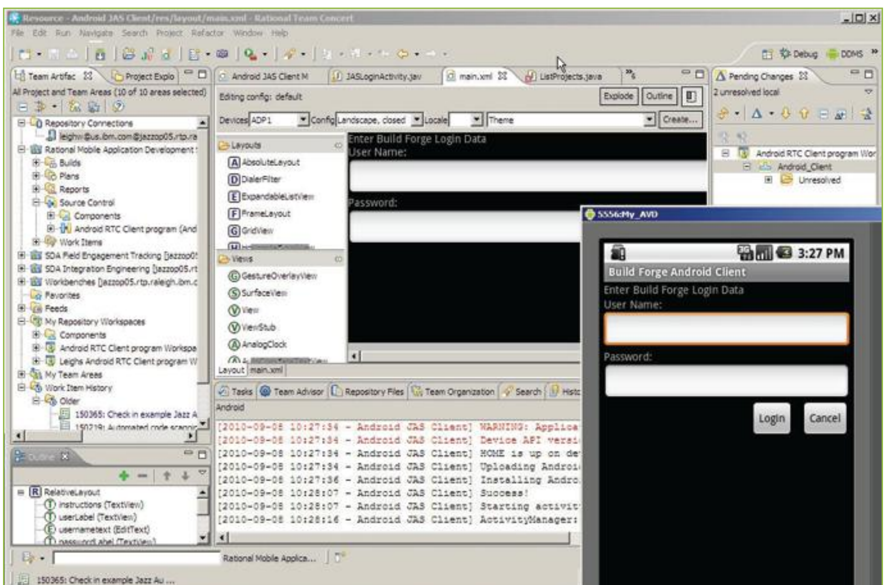


Figure 1 | Rational Team Concert enables a development team to easily create and manage Android apps throughout their life cycle.

Developers working on the Web application can execute the part of the application that runs on the device from their own IDE using the Android device emulator that is part of the SDK. Using the combination of shared change sets and an integrated device emulator, the pairs of developers working on the same feature (UI and Web app logic) can collaborate to work out initial problems that may arise as the result of different understanding about the application details. One of the developers can capture a screenshot of the device emulator using the screen capture tool built into the commercial IDE and share that screen capture with the other developer to show an exact behavioral issue or defect in the code.

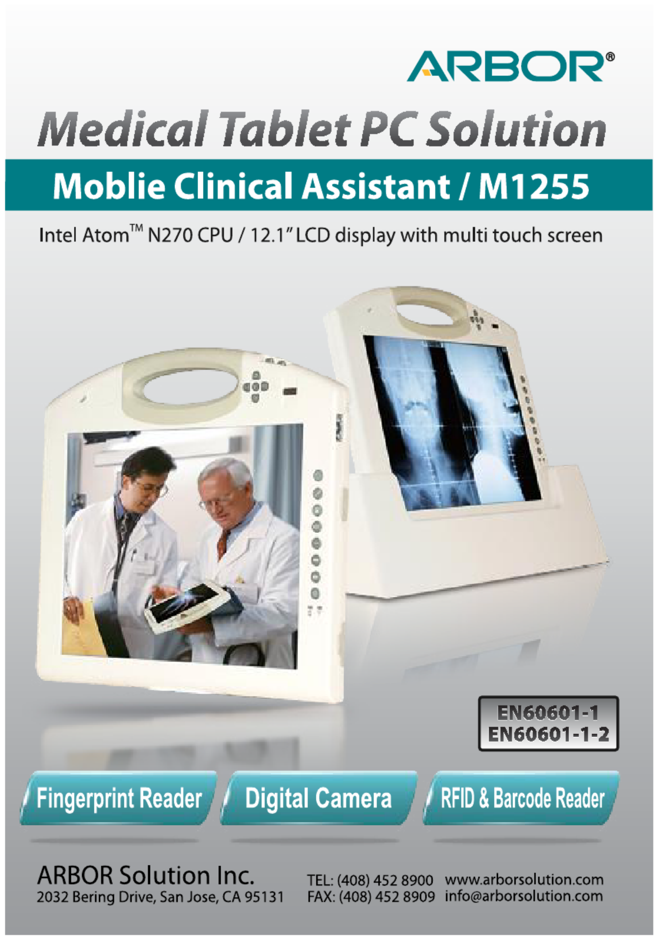
> *Everyone on the team can see how the iteration is progressing and the status of the work items planned for that iteration.*

Agile team collaborative development tools such as RTC allow the definition of multiple short iterations where a small set of application enhancements are to be implemented and validated. A typical agile iteration is two to four weeks long. The team leader can work with the team to map work items from a backlog list into the specific iterations and assign the work items to individual developers. As the developers pick up the work items and begin to make progress on them, their effort is automatically recorded and available for the team leader to track and view. This makes the information about what has been completed, what is being worked on now, and what is still to be done easy to track and view in a dashboard presentation. Everyone on the team can see how the iteration is progressing and the status of the work items planned for that iteration.

When the testers on the team start the functional testing of the application, they can open defects as work items in the shared development project. They can easily grab screen captures of the failed tests and include them in the defect records. The team leader can track these incoming test defects and work with the team to distribute them for resolution.

Products add value to basic SDK

Several Eclipse-based commercial products can be integrated at the same time with the Android SDK to provide



the team with even greater capabilities. For example, the ability to model the device code structure and keep this model in sync with the real source code can be added to the collaborative agile team environment. By integrating a commercial product such as IBM Rational Rhapsody with the combined Android SDK and RTC environment, the team can gain the ability to keep a high-level model of the application in sync with the actual application code. Because the real application structure can be difficult to understand for moderate-sized projects, the ability to generate a model from the source code can prove very valuable for the team.

Commercial static analysis products can be integrated with the Eclipse-based development environment and deliver the ability to analyze the code for quality and security issues. Some of these products can be integrated with the actual change set deliver process so that no code is integrated into the main line code stream unless it has been analyzed for fundamental quality and security issues.

All of the capabilities delivered by the commercial development products extend and enhance the basic SDK supplied by Google. While the Google Android SDK is the fundamental starting point for any project delivering code to be executed on the Android platform, the SDK can be dramatically more effective when integrated with the traditional agile team development features available in other Eclipse-based commercial products becoming more widely available today. **ECD**

ECD in 2D: Ted the Developer shows how IBM's Rational Team Concert helps teams be more collaborative. Use your smartphone, scan this code, watch a video: http://bit.ly/hNfEFw

__Leigh Williamson__ is an IBM Distinguished Engineer who has worked in the Austin, Texas lab since 1988, contributing to IBM's major software projects including OS/2, DB2, AIX, OpenDoc, Java, Component Broker, and WebSphere Application Server. He is currently a member of the IBM Rational Software Chief Technology Officer team, influencing the strategic direction for products in the Rational brand and leading the projects for software development automation and mobile device application development. Leigh holds a BS in Computer Science from Nova University and an MS in Computer Engineering from the University of Texas at Austin.

IBM Rational
leighw@us.ibm.com
www.ibm.com

Challenges of porting Android to non-handset applications

By Vlad Buzov

Initially created for mobile phones, Android is quickly moving into the realm of non-handset embedded devices. Using a multi-OS approach when porting Android to these types of devices can help overcome the adaptation challenges that often lead to additional costs and longer development cycles.

Since Android's first viable commercial release ("Cupcake") less than two years ago, it has become the go-to mobile software platform of today. Because of its flexibility and ability to satisfy a host of technical requirements, Android is quickly finding its way into non-mobile handset devices. You can now find Android in tablets, eBooks, set-top boxes, DVRs and TVs, in-flight entertainment systems, medical instrumentation, and industrial automation.

From a technical perspective, Android is a complete application framework on top of the well-known Linux kernel supporting a variety of architectures and processors. It offers sophisticated functionality and ample opportunities for developers to create interactive user applications with attractive user interfaces.

Because of its roots in the mobile handset, Android's implementation is tailored around mobile phone use cases, starting from user interface constraints and going down to hardware configuration assumptions.

Porting issues to consider

Several challenges are worth noting when porting Android to a non-handset application. Obviously, the challenges listed in this article will vary according to each developer's specific end application and hardware. A few of the key porting issues include extending Android, integrating legacy software, and addressing GNU and bionic runtime environments.

Extending Android

Making a non-handset device in Android might require extending Android to provide new functions depending on the purpose of the embedded product. Simply put, the phone stack must be removed and some other software unique to the product must be implemented or integrated. These steps could be taken on all levels of Android, as depicted in Figure 1.

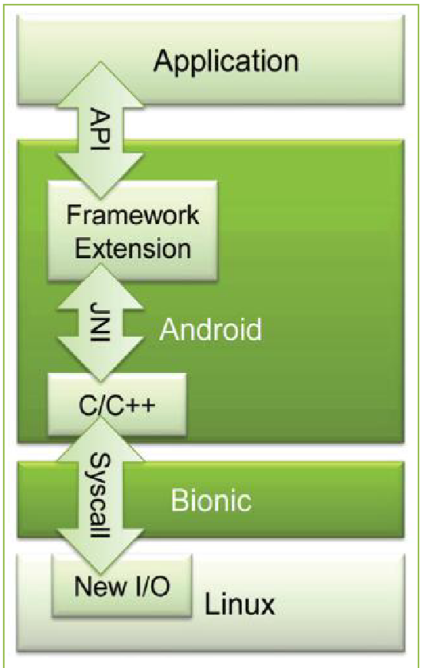


Figure 1 | Android can be extended to provide new functions for a non-handset device.

A new device driver (from the Android perspective) can be added or enabled in the Linux kernel. Some native code must be implemented to interface with the driver over system call interfaces and be accessible to Java framework extension over the Dalvik Virtual Machine Java Native Interface. The framework extension can implement new system services or modify an existing source, such as changing a notification bar design. New intents or Android Interface Definition Language can be added at the framework level, as new permission groups for applications begin to use them. And on top, the Android API can be extended for applications to use new Android functions.

Legacy software integration

If Android is selected for the next generation of an existing embedded product line, the transition would entail migrating from some other Operating System (OS). In this case, the software implementing the necessary functions could already exist and should be integrated into Android.

Even though bionic libc and core libraries are derived from a UNIX-like OS, they provide only the most basic system functions required to run Android. Thus, certain libraries and interfaces used by the software working in a GNU environment might not be available in Android. Several examples include:

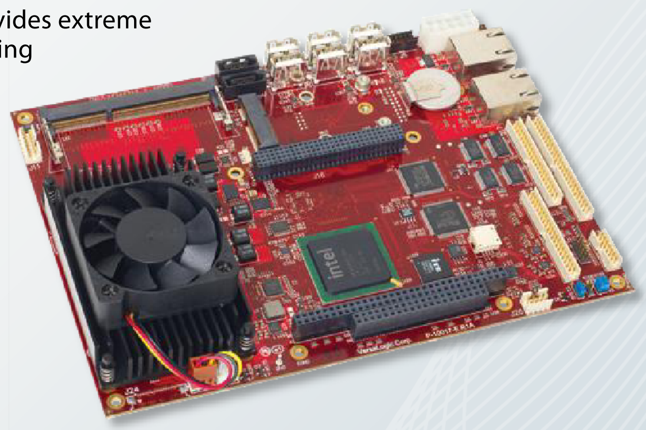
- No SysV InterProcessor Communication (IPC) support
- No Standard Template Library support
- Partial pthreads support
- No Linux headers package
- Minimal "scrubbed" set of headers

This means that compiling GNU-based software could turn into a more difficult task, requiring changes to existing source

Extreme Embedded Performance!

Intel Core 2 Duo processing on standard EBX footprint

VersaLogic's new "Mamba" single board computer provides extreme performance and high reliability for the most demanding embedded applications. It combines a 2.26 GHz Intel® Core™2 Duo processor, high-performance video, and extensive on-board I/O on an industry standard EBX platform.



- 2.26 GHz Intel Core 2 Duo processor
- Up to 8 GB DDR3 RAM
- Dual gigabit Ethernet
- Mid power. 18.5W typical
- PC/104-*Plus* expansion
- Industrial temp. (-40° to +85°C) version
- High-performance video and audio
- Standard EBX format (5.75" x 8")

Mamba is backed by VersaLogic's long-term (5+ year) product availability guarantee and legendary quality. Customization is available even in low volumes. Evaluate one today!

With more than 30 years experience delivering extraordinary support and on-time delivery VersaLogic has perfected the art of service, one customer at a time. Experience it for yourself. Call **800-824-3163** for more information!

Recipient of the VDC Platinum Vendor Award for five years running!

VDC research 2008 PLATINUM VENDOR

VERSALOGIC CORPORATION

800-824-3163 | 541-485-8575 | www.VersaLogic.com/mam

code or bionic. A drastic change such as this might not be acceptable for various reasons, including additional or unexpected maintenance costs. If this is the case, another approach should be considered to make use of legacy software in the Android runtime environment.

> *Android software licenses would reduce the risk of releasing software IP, but might not completely mitigate it.*

Coexisting GNU and bionic environments

One method of using existing GNU-based software is to run both the GNU and bionic runtime environments in parallel (Figure 2). In this scenario, Android communicates with GNU-based software via IPC mechanisms provided by the Linux kernel. One example of a native Android IPC is Binder – if the GNU runtime can be extended to support it. Other examples include conventional mechanisms such as network sockets. The existing software could be extended by an upper-level communication layer, which would be kept separately from original software that stays untouched.

The multi-OS approach

In some situations, the hybrid approach might not be possible. Migration to Android could be performed from a different OS, either commercial or homegrown. Certain system requirements such as performance, boot time, response time, or other real-time characteristics would need to be implemented. Android software licenses would reduce the risk of releasing software IP, but might not completely mitigate it. Certification might be required for certain software, which could be more difficult if it's a part of Android.

While the list of reasons could easily continue, one solution addresses most of these concerns: running Android in parallel with another OS. This scheme can vary depending on features provided by the System-on-Chip (SoC).

Working with a single-core architecture, various virtualization solutions (commercial or open source) can be utilized. For multicore processors, the solution can entail a virtual machine/hypervisor or both OSs running on separate cores with static hardware resource separation.

The communication between OSs (Figure 3) is either provided by the virtualization layer or explicitly implemented for both OSs via shared system resources, such as shared memory and interprocessor interrupts like Multicore Communications API.

Applications for the multi-OS approach comprise embedded products with a user interface driven by Android and core device functions run by a Real-Time OS (RTOS) such as the Mentor Graphics Nucleus OS. Examples of a Linux/RTOS multi-OS application include multifunctional printers, in-vehicle infotainment systems, and medical devices.

Hardware resources

Even with its great design, Android is not a true embedded OS that can fit a minimal system configuration. By using Java, Android achieves high machine independency levels requiring more processor power and memory to run. The rich graphics and multimedia capabilities provide more than ample room for designing innovative user interfaces and applications, but they require more processor power and memory, as well as hardware assistance for graphics operations and audio/video decoding to offload the main processor core.

Modern processors provide the required amount of memory and enough processing power and graphics acceleration features to run Android. However, selecting a higher-level SoC and adding more memory will affect the cost of the final product. Therefore, several questions need to be carefully analyzed before making a final decision about migrating to Android:

- Will the device be display-driven?
- Is the processor powerful enough to run Android?
- Is hardware acceleration for graphics and multimedia available?
- What is the memory budget?

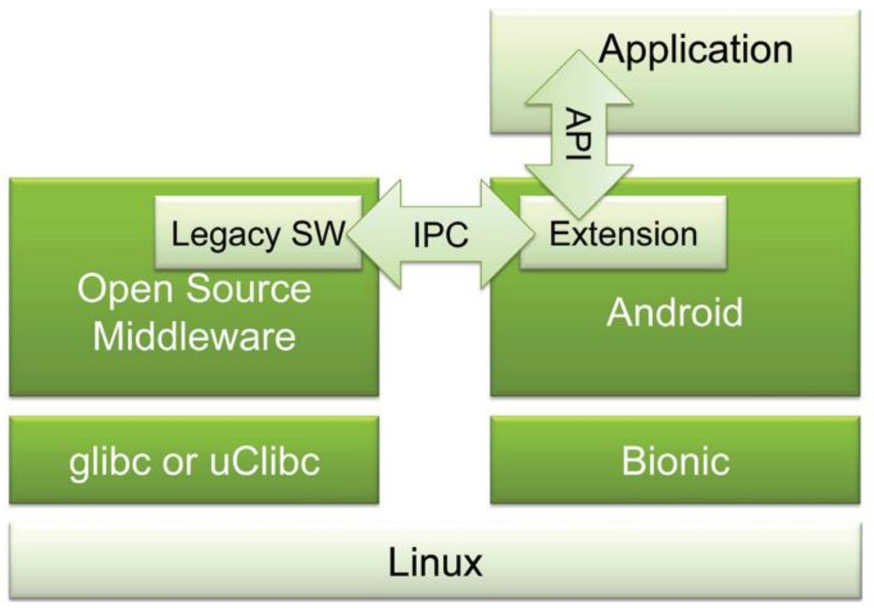


Figure 2 | Both GNU and bionic can run in parallel, allowing Android to communicate with GNU-based software via IPC mechanisms.

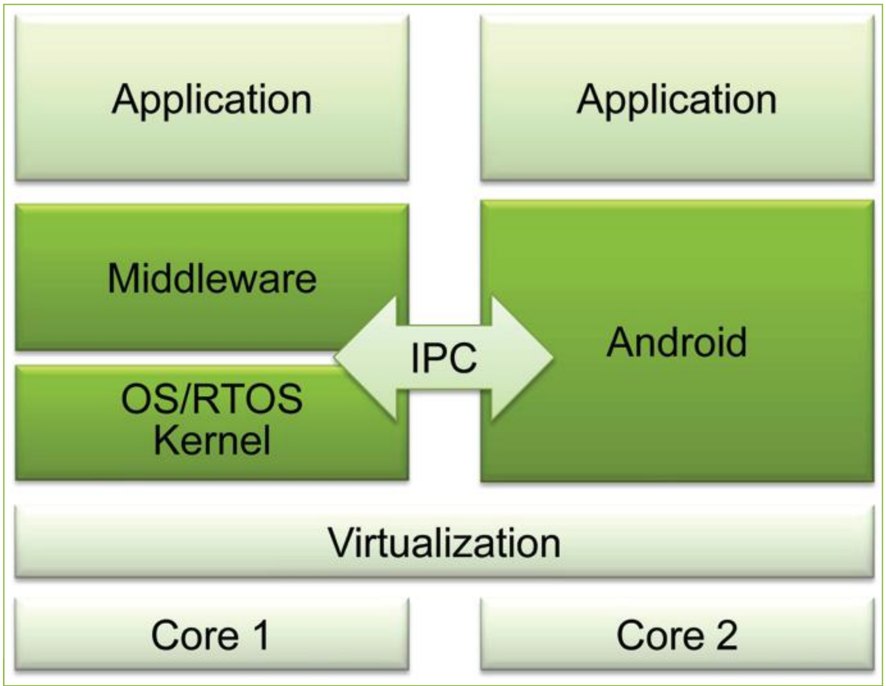


Figure 3 | In a multi-OS approach, Android can run in parallel with another OS and communicate via the virtualization layer or shared system resources.

The answers to these questions depend on product requirements and use cases. Sometimes, having a low-power processor with a reduced feature set is sufficient if there are no multimedia or graphics requirements. If display size is small enough, the device might not need a large amount of memory installed. Having source code integrated into Android rather than running in parallel using some sort of virtualization approach might affect hardware cost, reducing software implementation costs.

Leveraging the Android framework

Android is a rapidly growing software platform. Even though it was originally created for mobile handsets, it is quickly being deployed in many different types of embedded devices. By performing a low-level Android port to an embedded device, developers will have the complete Android framework available for leveraging existing Android applications.

Certain challenges in making an Android-based product might require a deeper understanding of Linux, Java, and system design. However, companies like Mentor Graphics can provide the expertise and services needed to address these problems, significantly shortening the subsequent development cycle for future revisions. **ECD**

Vlad Buzov is a software architect at Mentor Embedded Professional Services. He has more than seven years of experience in Embedded Linux programming, including porting, customizing, and optimizing existing products and developing new custom solutions based on open source components. Prior to Mentor, Vlad served as senior engineer at Embedded Alley and MontaVista Software. Originally from Russia, where he received a Master's degree in Computer Science from St. Petersburg State University, Vlad is now based in San Jose, California.

Mentor Embedded
Vlad_Buzov@mentor.com
www.mentor.com

ECD in 2D: Mentor Graphics demonstrates how Android can be used as the backbone of a car's infotainment system. Use your smartphone, scan this code, watch a video: http://bit.ly/eZ0Ki6

The future of CompactPCI® is serial...

MEN Micro leads the way again:

Save your system.
Migrate cost effectively.
Keep proven mechanics.

CompactPCI® PlusIO PICMG 2.30

- 100% compatible with parallel CompactPCI®
- PCI Express®, Ethernet, SATA, USB
- Support of 4 peripheral slots
- Fast 5 Gb/s connector

CompactPCI® Serial PICMG CPCI-S.0

- Star architecture
- Full Ethernet mesh
- No bridges, no switches
- Support of 8 peripheral slots
- Fast 12 Gb/s connector
- Proposed CPCI-S.0 CompactPCI® Serial specification currently under development

Count on MEN Micro to get you to the future of harsh, mobile and mission-critical embedded technology first!

men micro, inc.

MEN Micro, Inc.
24 North Main Street
Ambler, PA 19002
Tel: 215.542.9575
E-mail: sales@menmicro.com
www.men.de/cpci-serial

www.men.de/cpci-serial........www.men.de/cpci-serial........www.men.de/cpci-serial........www.men.de/cpci-serial........www.men.de

Menu expands at the OS diner

By Maciej Halasz

Android, having captured the bulk of mindshare and ink as the leading mobile/embedded OS, is the blue plate special, but menu choices today also include the MeeGo project – an open source project created jointly by Intel and Nokia – and other embedded operating systems such as Embedded Linux and the Apple operating system for iPhones. The author analyzes the potential impact of MeeGo on embedded device designs, discussing opportunities and threats the MeeGo project faces. Finally, the author recommends some criteria to consider when selecting an OS for your next embedded project.

Introduction

In recent years, we all have seen new multimedia/multipurpose devices from RIM, Motorola, Nokia, and Apple. Technology advancements in both hardware and software have resulted in faster-than-ever-before expansion of the embedded space in a variety of markets. It is the largest market segment, the consumer electronics market, that drives the most innovation.

A new OS, MeeGo, found its way to the consumer electronics market OS menu (where the iPhone OS and Android items tend to jump out) a few months ago (Figure 1).

The goal behind many of these OSs (or software stacks) is an application framework that allows developers to write software once and to then run it on many different platforms enabled with the same OS. Graphics frameworks such as the Nokia-owned Qt and the Android user interface API facilitate such binary compatibility.

With a number of alternatives available to build the next generation of embedded devices, it is critical that developers understand operating systems' strengths and weaknesses and the trade-offs each requires.

Figure 1 | Timeline of first software releases of major software stacks used in mobile market

What is MeeGo?

Announced jointly by Intel and Nokia, MeeGo is a Linux-based open source software stack for mobile devices. Hosted by the Linux Foundation, the goal of MeeGo as an open source project is to support creation of multimedia based devices in markets such as handsets, netbooks, In-Vehicle Infotainment (IVI), smart TV, and more.

The MeeGo project is based on Intel's Moblin project and Nokia's Maemo. Since its inception we have seen many other vendors joining the MeeGo project, including AMD.

MeeGo's modular approach includes MeeGo core frameworks that are shared by all profiles and different user experience layers for each type of device, with one example being In-Vehicle Infotainment (see Figure 2).

The core OS of MeeGo uses a Linux kernel and a set of packages from Nokia's Debian-based Maemo and Intel's Fedora-based Moblin projects. As shown in Figure 2, these APIs support a variety of frameworks, such as Communications, Multimedia, or Location, in the MeeGo stack.

The Nokia-Intel Alliance produced a key outcome, the adoption of the Qt framework as a default application development framework. This framework allows companies to develop applications for a variety of devices and put into practice a "develop once, run on many" philosophy. It's a standardization approach that other OSs such as Android and iPhone OS also use, and it dictates APIs and frameworks against which to develop value add software.

MeeGo alternatives – Android, iPhone OS, Linux

In embedded markets such as automation, industrial, consumer, and medical, products shipping today are often Linux-based.

Linux

The conventional Linux approach offers engineers the most flexibility in designing a complete solution for an embedded system. It allows engineers to design the smallest and fastest software for a product. It also requires more effort, as engineers have to design and build core features that application engineers can develop against. The process of designing with Linux has dramatically improved over the years and today offers access to many tools and to support and knowledge from open source as well as commercial vendors. With a proven Qt framework that abstracts application API from OS, applications written for one device will run on others, although recompilation of the code might be required.

Android

While the Android OS relies on Linux and its device drivers to access various peripherals, such as a USB or an LCD display, the connection to Linux ends pretty much there. Most of the software that's running on an Android device is written in Java. Google wrote and controls all the core APIs and the stacks that application engineers develop against. Any modifications to core APIs introduced by Google ensure backwards compatibility with previous Android releases.

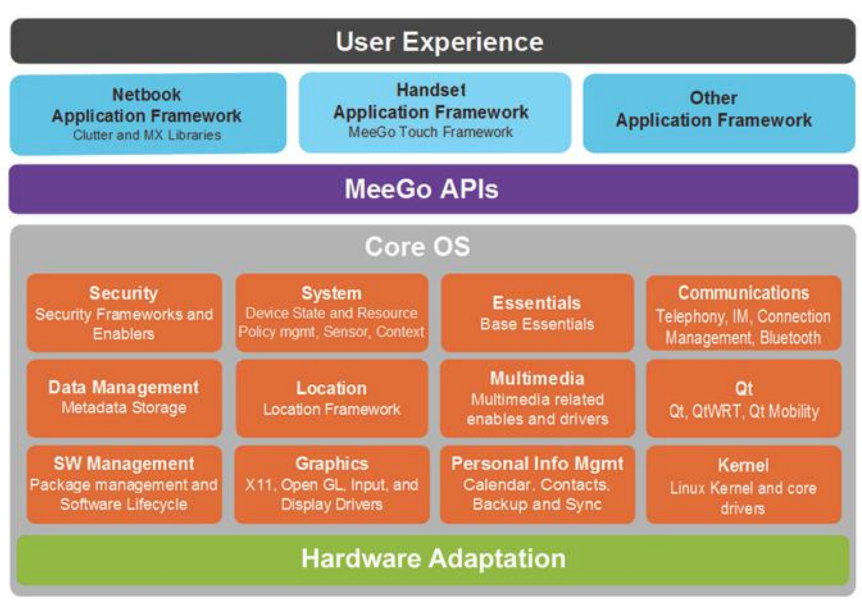


Figure 2 | MeeGo architecture (Source: MeeGo.com)

This approach defines a common interface for application developers and is a reason why the same application from an Android market can run on different Android enabled devices. This approach, however, does not give developers complete control over the core frameworks. Engineers can contribute by submitting changes to the Android core API, which is open source, but it is up to Google maintainers to decide if code gets accepted or not.

MeeGo

Like the Android OS, MeeGo also aims to simplify embedded Linux device development. The benefit of using MeeGo is that it is all Linux. Similarly to Android and iPhone OS, an engineer gets a complete reference platform and can focus on writing value-add software. It is fairly easy to adapt existing Linux applications to run under the MeeGo framework. Companies with products at market can upgrade their designs by leveraging MeeGo's many features and frameworks. They can offer modern user interfaces and still provide a customized user experience.

MeeGo is completely open sourced and allows engineers to easily contribute changes both upstream and to MeeGo. Assuming MeeGo adoption happens and given the number of Qt and Linux engineers in the embedded space, the number of applications for MeeGo should grow rapidly.

iPhone OS

The iPhone OS is Apple proprietary, limiting engineers to developing applications and allowing them only to develop for Apple enabled devices.

When comparing alternatives, we should look at not only the functionality an OS offers, but we should also support these alternative options for application development.

As shown in Table 1, all of the OS options listed offer a Software Development Kit (SDK), which can be used by engineers to develop value add software/applications. The SDKs include an appropriate cross environment to compile code for a target device. They also come with a nice set of tools to assist with application design, compilation, and testing. MeeGo offers tools such as QtCreator to facilitate a WYSIWYG design approach. Apple has its proprietary Xcode, and Android comes with the Eclipse Integrated Development Environment (IDE), which includes Android Development Toolkit (ADT) plug-ins. The Eclipse framework is the most popular IDE for Linux-based projects and is used as the basis for many commercial Linux development tools.

	MeeGo	Android	Linux	iPhone OS
Custom Hardware Adoption	✓	✓	✓	✗
Predefined Software Frameworks	✓	✓	✗	✓
Developer Contributions to Core Software Framework	✓ (Controlled by open source maintainers)	✓ (Controlled mostly by Google employees)	✓ (Controlled by open source maintainers)	✗
Custom Software Framework Extensions	✓	✓	✓	✗
Application Market	✓ (Multiple)	✓	✗	✓
Licensing	Open Source	Open Source	Open Source	Proprietary
Application Development Environment (SDK)	✓	✓	✓	✓
Emulations	✓ (QEMU)	✓ (QEMU)	✓ (QEMU/ Native)	✓ (Proprietary)

Table 1 | MeeGo alternatives comparison

Weighing the choices

There are many reasons why Android is such a popular option today. It has been on the market for more than three years, and it has a rich community of developers building apps that are available through Android Marketplace.

Many users are familiar with the Android interface through its widespread adoption in the mobile market and marketing campaigns on mainstream media. With consumers on a first-name basis with the Android, it is easy for other devices to leverage Android's popularity and reach a wider customer base.

Despite the Android familiarity advantage, a number of reasons speak to MeeGo as a viable choice:

1. Customization – Certain levels of customizations, for example, adding support for another hardware device, are difficult in Android, and iPhone OS doesn't allow it. MeeGo and Linux provide a well-defined process and APIs to support it.
2. Linux open source alignment – Similarly to mainstream Linux designs, MeeGo uses many upstream packages. Android uses its own open source APIs, and iPhoneOS uses its proprietary stacks.
3. Easy migration of existing Linux-based products and applications.
4. Performance/feature optimizations – One of the benefits over standard Linux, which often is not optimized for specific features/performance.
5. Feature completeness – MeeGo offers all features and frameworks you would expect to see in an embedded OS, similar to Android and iPhone OS.

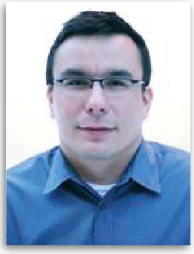
If having compatibility between devices for your applications is not an issue, open source frameworks such as MeeGo or Linux can be customized, allowing you to pull into your product design only the frameworks you care about. If you modify the MeeGo Core set of packages, you will not be able to claim MeeGo compliance, but the MeeGo project will make your task of building a custom media framework easy. The Qt-based graphics stack used by MeeGo will provide you with a complete development framework and the latest graphics UI, which you can leverage in your product.

If your next product is a mobile phone, Android is a very good choice as it offers access to a large pool of applications. If you are designing a custom embedded device such as a smart TV, MeeGo or commercially supported Linux is a good option as both offer a good development environment and greater flexibility.

When it's time to order…

Although MeeGo has entered the embedded space late compared to the well established Android and iPhone OS, it has the potential to become a major player in many market segments. MeeGo success will correlate heavily with that of the Intel Atom processor in the embedded market. However, given MeeGo alignment with different market segments, such as those for handsets and In-Vehicle Infotainment, it is just a matter of time before MeeGo will be available for more than a handful of architectures and processors. Given its features and openness, the MeeGo project can become a serious alternative to Android and other embedded OSs if it continues on its current path. ECD

ECD in 2D: MeeGo can do a variety of multimedia tasks as shown by this demo on the Freedom Jump Set-top-Box. Use your smartphone, scan this code, watch a video: http://bit.ly/eSXn5Y



Maciej Halasz *is director of product management at Timesys Corporation, provider of award-winning, proven embedded Linux solutions. Maciej has over 10 years of experience in embedded computing with a focus on embedded Linux and real-time systems. He holds a master's degree in Computer Science from ENST de Bretagne, France, a master's degree in Telecommunication Systems from Poznan University of Technology, Poland, and an MBA from the University of Pittsburgh Joseph M. Katz Graduate School of Business.*

Timesys Corporation
maciej.halasz@timesys.com
www.timesys.com

The return of the "talking car"

By Ed Chrumka

Ed discusses the prevalence of connected, embedded services and in-car experiences that are being made safer and easier to use through voice technology. The article highlights the technical advances in speech and automotive platforms that will continue to make this possible, such as those from the ng Connect Program and Nuance, and discusses the types of connected offerings soon to hit the streets. This article also speaks to the onslaught of mobile apps that are being designed for the connected car, and how voice capabilities will continue to be integrated to ensure a safer, smarter engagement behind the wheel.

In 1983, Chrysler Corporation introduced the "talking car" to America. Called Electronic Voice Alert (EVA), the car provided drivers and passengers with in-vehicle messages such as "please fasten your seat belt," "don't forget your keys," and "your washer fluid is low" in English, French, and Spanish. This technology was pretty advanced stuff at the time, but unfortunately, owners were soon searching for EVA's off switch or pulling fuses after enduring "a door is ajar" notification one too many times.

It took almost 10 years before OEMs began to seriously engineer speech technology back into their vehicles.

A regulatory trigger

In 1994, the Federal Communications Commission established requirements to provide location information for emergency calls originating from wireless phones. This seemingly simple mandate to provide the 911 caller's location set off great debate about costs and privacy. It also jump-started critical thinking on how location information might be exploited for commercial purposes, a concept embraced by multiple industries, including the automotive sector.

Thus, the word "telematics" was born, a term with European roots that incorporates the study of human-computer interfaces.

Vehicle telematics is driven by a network of wireless carriers, telematics service providers, content suppliers and aggregators, and automotive manufacturers all working together (see Figure 1). They are continuously evolving systems and services to fulfill their individual business models and deliver an ever-broadening range of safety, communication, navigation, and infotainment products to consumers through powerful mobile devices and embedded in-vehicle systems.

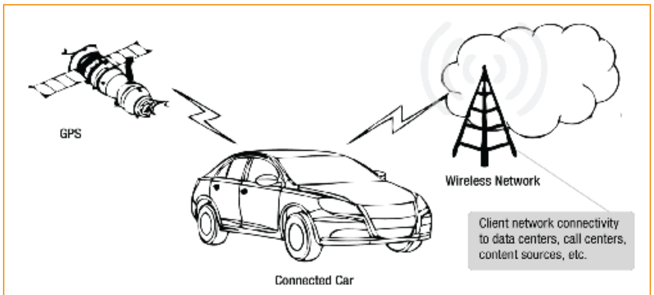


Figure 1 | The telematics network includes wireless carriers, telematics service providers, content suppliers and aggregators, and automakers working together to deliver communication, navigation, and infotainment services in the connected car.

The connected car

With the availability of pervasive and high-speed wireless broadband communications technology like 4G and Long-Term Evolution (LTE), consumers can exchange ever-increasing amounts of data and information to maintain their connected lifestyles, even when driving their vehicles. Furthermore, digital lifestyles are no longer associated with defined generational demographic models. Our global society as a whole is moving rapidly to a connected existence, with products and services within this arena following suit.

Given the abundance of data and information exchanged and processed while driving a vehicle, automotive manufacturers are building features and technical capabilities into vehicles to safely deliver telematics services to their customers and subscribers (see Figure 2).

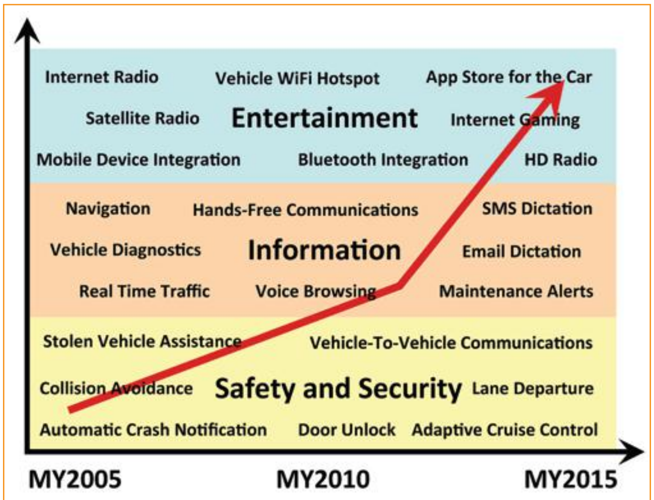


Figure 2 | Automotive manufacturers are incorporating an array of connectivity services and technical capabilities into their vehicles.

Today, connecting in-vehicle devices to the automotive subsystem is typically accomplished using Bluetooth technology. For a vehicle to be connected, it must incorporate an embedded telematics module connected to the vehicle's network and either a modem connected to the wide area network or a gateway capability that enables a mobile device such as a smartphone to integrate with vehicle subsystems, acquire a GPS signal, and connect to the voice and data mobile network.

Emerging trends in mobile social networking

With the emergence of mobile social networking, new uses for wireless technology are developing constantly. At the forefront of emerging trends are real-time and location-based capabilities. Real-time features allow users to contribute content and broadcast it as it is being uploaded. Location-enabled social networking web applications share a user's location, enabling them to check in to places as they arrive and automatically associate them with local events.

This flood of social networking capabilities and services is responsible for significant increases in wireless network utilization. Most companies today utilize social networking sites and build groups to enhance their brand image and advertise products and services to businesses and consumers. Savvy businesses promote these interactions, driving traffic to their own sites and encouraging feedback and discussions on how to continuously improve and/or create new products and services.

The commercial impact of social media and the content that is exchanged has driven device manufacturers and the automotive industry to make specific technical accommodations to support these services safely through multilingual speech recognition, speech-to-text, and text-to-speech capabilities. It is becoming increasingly important that access to social media sites is speech-enabled and can safely occur when driving a vehicle.

Using connected speech for connected services

Connected services enhance the user experience by combining the power of the client device (smartphone or embedded telematics unit) with powerful data center application platforms and content servers. Connected speech technology follows a similar hybrid approach that is robust to vehicle noise and results in superior recognition performance. Table 1 indicates a variety of connected services enabled by connected speech.

Domain	Connected Speech	Connected Service
Phone & Messaging	• Voice-activated dialing • Reading SMS messages • Reading e-mail messages	• SMS message dictation • E-mail message dictation • Social networking updates
Navigation & Location-Based Services	• Command & Control • Voice Destination Entry • POI search on local database • TMC message readout	• Local search/extended POI using network database • Driving-related information services, for example real-time traffic & weather info, fuel pricing
Entertainment	• Music selection for in-vehicle music players	• Music and video on demand with connection to online stores • Internet Radio
Full Internet Access	• Search for predefined topics	• Open Voice Search (say anything)

Table 1 | Implementing connected speech technology can tailor a variety of connected services specifically for in-vehicle use.

Connectivity should not come at the expense of safety and convenience. A simple, easy-to-use driver interface that minimizes driver distraction is critical to the ongoing success of connected solutions. Connected speech recognition can help address key safety concerns by enabling hands-free, eyes-free access to and control of in-vehicle systems and services, reducing visual/manual tasks needed to switch back and forth between various in-vehicle interfaces.

Reducing driver distractions

In the science of human factors, what we do behind the wheel is generally referred to as the "driving task." The driving task requires a baseline level of skill and mental concentration, as well as a visual focus on the road and maintaining hands on the wheel to quickly respond to changing traffic or road conditions. However, drivers face a number of distractions that run the gamut from eating a cheeseburger to listening to an audio book to engaging in the activity garnering the most national attention of late – texting while driving.

The U.S. Department of Transportation along with policy and law makers have spent the past year looking at ways to minimize distracted driving. Much of this effort seeks to promote tougher state laws banning texting while driving, bolstered by industry and federal educational campaigns, public service announcements, and the endorsement of national celebrities encouraging drivers to avoid inappropriate device use behind the wheel.

However, driver distractions go far beyond simply texting while driving, given the number of devices brought into vehicles, including navigation devices, MP3 players, and built-in infotainment systems. Because these systems and devices have become commonplace in our mobile environment, completely locking out features or banning them is difficult. Instead, consumers should be empowered to use technology in a safer, smarter way.

Speech recognition has great potential to alleviate driver distractions in our connected world. Consumer acceptance of in-vehicle speech recognition is continuing to improve. A 2009 Automotive Voice Interface User Survey indicated the following:

- Eight out of nine respondents who own speech-enabled in-vehicle systems and navigation devices regularly use the voice recognition capabilities.
- 73 percent of users have a high degree of satisfaction that would lead them to recommend the technology to friends and family, as well as plan to repurchase automobiles with speech-enabled functions.
- 83 percent of users always or frequently place phone calls using voice commands.
- Navigation users entered an address by voice 76 percent of the time.

The correct application of speech recognition results in decreased glances away from the roadway, and some research has shown that glance patterns when using voice activation are similar to that of normal driving. Measures of driving performance such as lane and speed deviation are better with voice than with manual interfaces. Recent studies indicate that drivers improve their ability to maintain the ideal car position by 19 percent using speech recognition as compared to manual dialing. On average, voice input helped drivers keep their eyes on the road 200 to 300 percent better than manual input.

The road ahead

Connected speech applications are emerging on a daily basis as more and more developers realize that speech recognition is a smart way to go. Applications to search the web, dictate messages and e-mail, update social networking sites, and acquire destination street addresses or points of interest can already be downloaded from application stores on a variety of smartphone and mobile device platforms. The onslaught of these applications will continue to hook new users and be considered valuable features while driving, so they must be designed to operate safely and on a hands-free basis.

As 4G and LTE networks emerge, hybrid connected speech capabilities will continue to follow a Moore's Law behavior. With higher-speed connectivity, additional client processing power can be exploited and more data can be exchanged with cloud-based recognition servers to complete complex transactions and push natural language capabilities into more devices. Additional speech-enabled reference implementations will range beyond only telematics and continue to exploit the capacity and performance of these emerging networks.

Multi-industry consortia, such as the ng Connect Program founded by Alcatel-Lucent, are creating services that utilize emerging technology coupled with next-generation networks. Through the ng Connect Program, the LTE Connected Car was created to showcase how high-speed networks can transform the in-vehicle experience, both as a passenger and a driver. The car features Nuance Voice Control and Vocalizer TTS for Automotive, which enable future connected services such as SMS dictation, voice destination entry for navigation systems, voice access to music libraries stored on in-vehicle entertainment systems, and voice search of the Internet, delivering context-oriented results.

Nuance Communications, a leading provider of speech and imaging solutions around the world and member of the ng Connect Program, aims to show different markets and constituencies the benefits of advanced speech recognition technology and LTE networks. The power of the ecosystem created in the program will help accelerate the adoption of this promising field of technology. ECD

ECD in 2D: See how Nuance's voice technology lets Audi A8 drivers give verbal commands to navigation and infotainment systems. Use your smartphone, scan this code, watch a video: http://bit.ly/eU8R8P

Ed Chrumka *recently joined Nuance as Senior Product Manager – Connected Car Services. Prior experience includes nearly 14 years at OnStar, where he led numerous product development programs. As Chief Technologist, he invented the OnStar Virtual Advisor and incorporated speech-enabled subscriber self-service applications in the OnStar call centers. He led engineering teams within OnStar's Advanced Systems Development group authoring hardware specifications for new services and HMI designs, and was awarded several patents for his inventions. In 2007, he progressed into the subscriber services organization and led the business team in the development of the next-generation OnStar call center application; culminating in the launch of OnStar China in 2009. Before joining OnStar, he held a variety of engineering, operations, and international business positions at Ameritech and National Steel Corporation. Ed has a BSEE and MSIE from The University of Michigan. He is married with two children. Outside of professional interests, he enjoys reading, travel, organic farming, and restoring classic cars and trucks.*

Nuance Communications
edward.chrumka@nuance.com
www.nuance.com

TechChannels

Up-to-the minute, in-depth, focused info

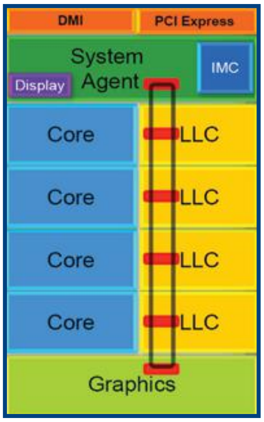
OpenSystems Media is continuing innovation with TechChannels:

Microsites that explore topics in more detail. Combine contributed articles from industry experts with the best in staff-written material, and mix in instant information from:

- Latest news and videos
- Blogs
- White Papers
- E-casts
- Twitter
- Facebook
- LinkedIn

tech.opensystemsmedia.com

Editor's Choice



Multicore processors target embedded applications

Whether you design or purchase the CPU board to power your next embedded project, the processor architecture and integrated features are important elements in the selection process. This architecture is at the heart of any embedded system and will influence system performance, component count, future updates, I/O configuration, and overall power dissipation.

With news that may affect your decision, Intel recently announced its second generation Core family including seven processors that support extended life cycle embedded applications. These Core i3/i5/i7 processors combine either two or four CPU cores, an integrated graphics processor, Last Level Cache (LLC), and a system agent/memory controller that all communicate using a scalable on-die ring interconnect system. All of the CPU cores, including the graphics core, support Intel Turbo Boost Technology, allowing clock frequencies to scale up temporarily to handle intense workloads. Prices for the Intel Core processors with seven-year life cycle support range from $138 to $378 in quantities of 1,000.

Intel | www.intel.com | www.embedded-computing.com/p47163

Rugged XMC delivers 10 Gigabit Ethernet

Plug-in mezzanine cards allow embedded designers to easily upgrade performance as a system matures or to completely change functionality to meet new requirements. XMC, as defined by VITA 42, is a popular embedded computing standard that combines the familiar PCI Mezzanine Card (PMC) with the latest in serial fabric technology. The XPort3300, a new XMC Module from Extreme Engineering Solutions (X-ES) offers designers dual 10 Gigabit Ethernet interfaces with front-panel or rear I/O support. In addition, a x8 PCI Express 2.0 port provides a high-speed interface between the XPort3300 and the host module via the P15 connector. The module is available in both conduction- and air-cooled configurations and targets multiple embedded computing applications such as remote sensor interfacing, traffic aggregation, storage, and data offloading.

Extreme Engineering Solutions | www.xes-inc.com | www.embedded-computing.com/p47162

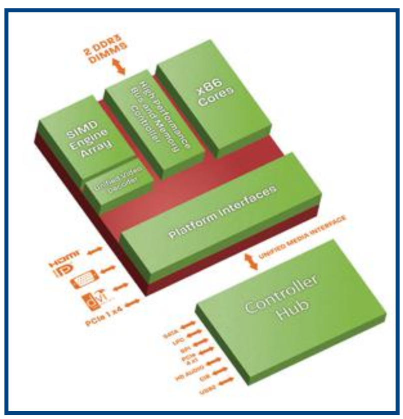
Modbus gateways expand remote connectivity

Serving the need for unattended monitoring and control in industrial applications, Modbus has become an important communications protocol for distributed automation systems. Modbus is an open and royalty-free standard for serial communications between a central controller and scattered Remote Terminal Units (RTUs). In spite of widespread Modbus deployment, many newer controllers and interface devices come equipped with Ethernet as the default communications standard.

To simplify the integration of these standards, Advantech has announced two new Modbus Gateways with built-in Ethernet switches. The EKI-1221D and EKI-1222D are bi-directional gateways for integrating new and existing Modbus serial devices to TCP/IP networked units. The Modbus/Ethernet interface allows remote management of devices without network connectivity. The EKI-1200 series gateways come with Windows and Web-based utilities for simplified user configuration.

Advantech | www.advantech-eautomation.com | www.embedded-computing.com/p47153

Embedded designs benefit with x86 architecture



Embedded system designers can significantly lower development costs and schedules by adopting hardware and software compatible with the widely supported x86 personal computer architecture. Multiple applications such as digital signage, set-top boxes, information kiosks, point-of-sale machines, and gaming platforms can take advantage of some of the same low-cost, off-the-shelf hardware components and software tools used for desktop development. AMD, a notable x86 stalwart, recently upgraded its offerings to combine a low-power CPU and a discrete-level GPU into a single Accelerated Processing Unit (APU) for embedded applications. The AMD Embedded G-Series processor combines an x86 core with a graphics processor and DirectX 11 support. APU configurations are available with single- or dual-processor cores, at 9 W or 18 W Thermal Design Power (TDP), and two levels of graphics and video performance. Each APU supports single or dual high-resolution displays with hardware decode support for H.264, VC-1, MPEG2, WMV, DivX, and Adobe Flash.

AMD | www.amd.com | www.embedded-computing.com/p47165

kontron

» Looking for the latest flexible processor technology? «

Kontron brings next generation Intel® processors, like the Intel® Atom™ processor E6x5C and E6xx series, to market on high performing, highly flexible embedded platforms.

» The next generation Intel® Atom™ processor is a cost effective building block for future designs
» Open standard interfaces deliver unprecedented levels of I/O flexibility
» For use in extended temperature ranges of -40°C to +85°C
» Increased software services including consultation, design, porting and validation

INTEL® ATOM™ PROCESSOR-BASED SCALABLE PLATFORMS

nanoETXexpress-TT
Computer-On-Module

» Intel® Atom™ processor E6xx series
» Ready for use in harsh environments
» Optimized integrated resources
» Custom I/O via carrier boards

MICROSPACE® MSMST
PCIe/104™ SBC

Learn more at
kontron.com/NextGen

» Intel® Atom™ processor E6x5C series
» Flexible I/O functionality with FPGA core
» HSMC for custom interfaces
» Ideal for small, low-power devices

CRITICAL QUESTIONS ... ANSWERED

Call, Email or Visit today.

Call: 1-888-294-4558
Email: info@us.kontron.com
Visit: kontron.com/NextGen

intel
Embedded
Alliance
Premier

Copyright © 2011 Kontron AG. All rights reserved. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Rev. # G109

If it's embedded, it's Kontron.

By Don Dingee

ddingee@opensystemsmedia.com

BLOG

Stories reshaping embedded in 2011

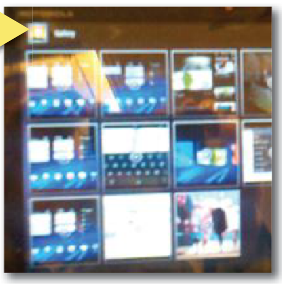
Read all 20 stories at http://bit.ly/storiesembedded2011

#4 These ARE the Droids you're looking for …


twitter

Motorola Mobility tripled at #CES. Atrix 4G huge. Droid Bionic huge. XOOM in bit of Android 3.0 and 4G rundown, but may score.

dondingee
Don Dingee



"We have just begun to see how the creativity around Android is going to push all kinds of devices to the forefront …" How about a new Android 3.0 tablet? We caught a glimpse of the upcoming Motorola XOOM at CES in its unfinished state.

Consumer Embedded Show

BLOG The afterblog from CES

"GE, after a 30 year hiatus, returned to the show to lay out a vision of digital energy from generation to homes to electric vehicles (Chevy Volt and GE Wattstation pictured together – yes, that's a revelation in itself that the plug is in front of the driver door, no driving off with the plug in), and its booth was mobbed every time I passed it."


twitter

A one-button cellphone cool? @LifecommLLC device (in charger) means mobility for loved one http://yfrog.com/h7tnumj

l2myowndevices
Don Dingee

Combine a one-button Qualcomm-based GPS speakerphone with the Hughes Network Systems' telematics network and American Medical Alert's expertise in personal emergency response service, and you get the new Lifecomm. See more at http://lifecomm.com

BLOG

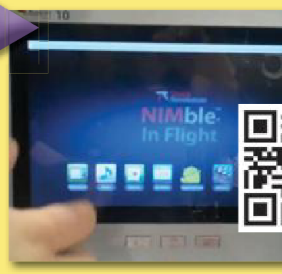
What's really changed is how the XOOM is being pre-marketed via social networks. See blog: "The new marketing of XOOM" http://bit.ly/newmarketingxoom

For the 2nd year running, Don is your recognized Smart Mobile Device Pundit! http://freescale.com/pundit

Left2MyOwnDevices channel at http://youtube.com/l2myowndevices

BL Healthcare

GE Appliances



Touch Revolution

2nd Generation Intel® Core™ i7 Processor Solutions

Optimized For Embedded Computing Applications.

- 6U VPX
- 3U VPX
- 6U cPCI
- 3U cPCI
- XMC
- VME

X-ES 2nd Generation Intel® Core™ i7 Processor Solutions: Delivering Innovation

In 2010, Extreme Engineering Solutions, Inc. (X-ES) developed more Intel® Core™ i7 processor products based on VPX, CompactPCI, VME, CompactPCI Express, and XMC form factors than anyone in the industry. This year, X-ES has added solutions based on the 2nd generation Intel Core i7 processor. Providing products customers want, when they want them – that truly is innovation that performs.

X-ES offers an extensive product portfolio that includes commercial and ruggedized single board computers, high-performance processor modules, multipurpose I/O modules, storage, backplanes, enclosures, and fully integrated systems.

2nd generation Intel Core i7 processor solutions available in a variety of form factors.
Call or visit our website today.

X-ES
Extreme Engineering Solutions

608-833-1155 • www.x-es.com

Annapolis Micro Systems

The FPGA Systems Performance Leader

High Performance Signal and Data Processing in Scalable FPGA Computing Fabric

GEOINT, Ground Stations, SDR, Radar, Sigint, COMINT, ELINT, DSP, Network Analysis, Encryption, Image Processing, Pattern Matching, Oil & Gas Exploration, Financial Algorithms, Genomic Algorithms

Direct Seamless Connections with no Data Reduction
Between External Sensors and FPGAs
Between FPGAs and Processors over IB or 10GE
Between FPGAs and Standard Output Modules
Between FPGAs and Storage Arrays

Annapolis
Micro Systems

Ultimate Modularity
From 1 to 8 Virtex 4, 5 or 6 FPGA/Memory Modules
Input/Output Modules Include:
Quad 130 MSPS thru Quad 550 MSPS A/D
1.5 GSps thru 5.0 GSps A/D, Quad 600 MSps D/A,
Dual 1.5 GSps thru 4.0 GSps D/A
Infiniband, 10G, 40G or 100G Ethernet or SFPDP

VME/VXS/VPX, IBM Blade, PCI-X/PCI Express, PMC/XMC, MicroTCA

No Other FPGA Board Vendor Streams This Volume of Data Real Time Straight Into the Heart of the Processing Elements and Then Straight Back Out Again

190 Admiral Cochrane Drive, Suite 130, Annapolis, Maryland USA 21401
wfinfo@annapmicro.com USA (410) 841-2514 www.annapmicro.com