

Build Embedded Devices on a Solid Foundation: The Unison RTOS

In any project, the sooner you can start adding unique value, the faster you can get a product to market with a greater margin of profit. The fewer defects you have to find and repair, the lower will be your costs. With embedded systems, the way to do this is to start with a solid platform—a matched pair of processor and RTOS—one that has all the capabilities you may need but from which you can select only those you do need. And you need the assurance that these will all work from day one allowing you to address the market with an innovative solution built on a solid foundation.

The Unison RTOS from RoweBots is itself designed to a known standard, POSIX, and is optimized for size, modularity and adaptability to SoCs, MCUs, MPUs, DSPs, FPGAs, and digital signal controllers (DSCs). It is already available in matched processor/RTOS packages that come as development platforms along with an extensive set of I/O modules plus services and protocols for advanced Internet connectivity, wireless communications, USB, security and more. It is based on seven key principles including an optional Cloud capability to immediately allow a developer to add value within the opportunity and seamlessly permit future expansion and enhancement of a product line. These key principles ensure development using Unison RTOS will deliver minimum time to market, minimum risk and minimum total cost of ownership.

Different organizations will usually have an in-house development model that they feel comfortable with. Whatever the development model, starting a design with pre-integrated hardware and software (i.e., RTOS, drivers and service modules) gives that selected development process a head start. And, this results in an immediate leap ahead in time-to-market by avoiding basic integration efforts and eliminating early defects. Fire up the platform and start adding innovation and value.

With so many of today's devices connected to the Internet of Things, it is important to consider the relation of the embedded devices to gateway systems and the Cloud. It is desirable, but not imperative to try to have a common operating system API. The Unison system is built on open standards that make this easier to achieve. This along with its modularity and adaptability give it a number of advantageous characteristics:

Lean

The lean development model is gaining popularity with a number of organizations and the Unison RTOS approach fits with many of that model's goals. The Unison foundation is built to provide standardization, interchangeability of drivers, protocols and service modules and portability of applications. Multidisciplinary teams can use diagrammatic approaches like UML to share ideas and knowledge about system functions, all with a single project manager. Such an environment can be applied to almost any organization's internal project model or design discipline. In fact, the Rowebots' Hurricane Product Development Kits (PDK) created by combining Renesas hardware and RoweBots software, optimize both hardware and software

design into a collection of starter systems. The kits can be adapted to almost any given set of corporate practices, product development approaches and software processes to reduce total time to market and total cost of ownership.

Adaptable

Any organization developing a solution for customer needs must be aware that they are aiming at a moving target. Once a product is in place with a customer, the OEM must be able to quickly react to calls for additional features and expanded functionality—or even a more narrowly focused, lower-cost version of a product. The ability to adapt to meet changing customer needs requires a modular, scalable system that supports various alternatives. The platform-based, model-driven development approach can greatly help if models need to be adapted to new architectures.

This is made easier by a platform that supports a common OS API such as POSIX, which allows the easy introduction of new applications and the porting of applications from one product to the newer version of that product or the use of the same application across a compatible product line. For example, using the RoweBots Unison RTOS on ST STM32 or a Renesas RZ/A1 to run an application could be upgraded to an embedded Linux with RZ/A1 or multicore iMX6 for non-embedded system. This provides great ability to scale in a modular way and provides a broad set of architectural alternatives. Model-based development can be used in both environments.

Such adaptability can get to the hardware layout level when the processor selected belongs to a pin-compatible family. With Unison and its matched processors a developer can drop in the processor that fits perfectly at the end of the design cycle when all the requirements for performance and memory are settled. It can also greatly help in future product enhancement by simply dropping in a higher performance MCU.

Secure

Today many connected systems operate in environments that can include danger to human safety and life. If a system is not secure, it is neither reliable nor safe. As is the case with reliability and safety concerns, today's devices must be designed for security from the ground up. And security must remain a concern from system conception to system retirement. This begins with the selection of a secure initial design and extends through communication protocols, strategies such as password, electronic key and physical recognition, the use of secure booting, encryption and many more strategies. However, the judicious selection of the basic system architecture, hardware and software is also a key requirement. The Unison RTOS includes a rich set of security features that can be put in place at the foundation to build a secure system and application set. They include:

- TLS/SSL for secure connections between applications
- IPsec for encrypted VPNs.
- Filtering or firewall features

- Secure mail or secure SMTP along with secure web pages or HTTPS for implementing layered security strategies.
- Secure shell (SSH) and secure file transfer (SFTP), both of which are completely new protocols: SSH for secure configuration of target systems and SFTP for secure file transfer to/from target systems.
- Secure management - SNMP v3, to secure both authentication and transmission of data between the management station and the SNMP agent.
- A Secure Bootloader, which requires the files which are downloaded for reflashing the system to be encrypted and checked before loading. And finally,
- An Encrypted File System

Safe

Did we already mention safety? Well it's that important. Embedded devices whose failure can threaten human lives include those used in medical equipment, industrial control, transportation and military equipment. These industries have adopted standards that define different levels of risk that must be addressed to be considered safe and that are often required for certification.

The Unison RTOS was developed with safety in mind and offers two main features that help ensure safety in systems. Determinism guarantees quick response to threatening conditions and makes the operation of the system predictable so that it can be reliably tested to meet strict timing requirements. Emergency stop with zero boot time means that a device can be halted instantly and restarted with zero boot time if required. Thus an unsafe condition can be halted immediately and brought back to a safe condition or the device diverted to an emergency action for remedy.

Connected

The era of the Internet of Things has opened up a new set of demands and criteria for connected embedded devices. Where in the past, a device could rely on Wi-Fi and Internet along with USB to cover connectivity needs, today's devices must accommodate a very broad range of sensors, both wired and wireless. Wired connections include USB 2.0, QSPI and MMC for storage, UART (with COM 1 and 2) and GPIO with and without interrupts.

In addition, a greater variety of wireless protocols must be seamlessly supported. In addition to the various flavors of Wi-Fi (802.11a/b/g/n) many must also support Bluetooth Smart/Smart Ready, 802.15.4 with 6LoWPAN along with LoRa, 3G, 4G and UHF. The increasing variety of radios used for gateways makes ever more demands for protocols with security. An overview of the TCP related protocols supplied with Unison is shown in *Figure 1*.

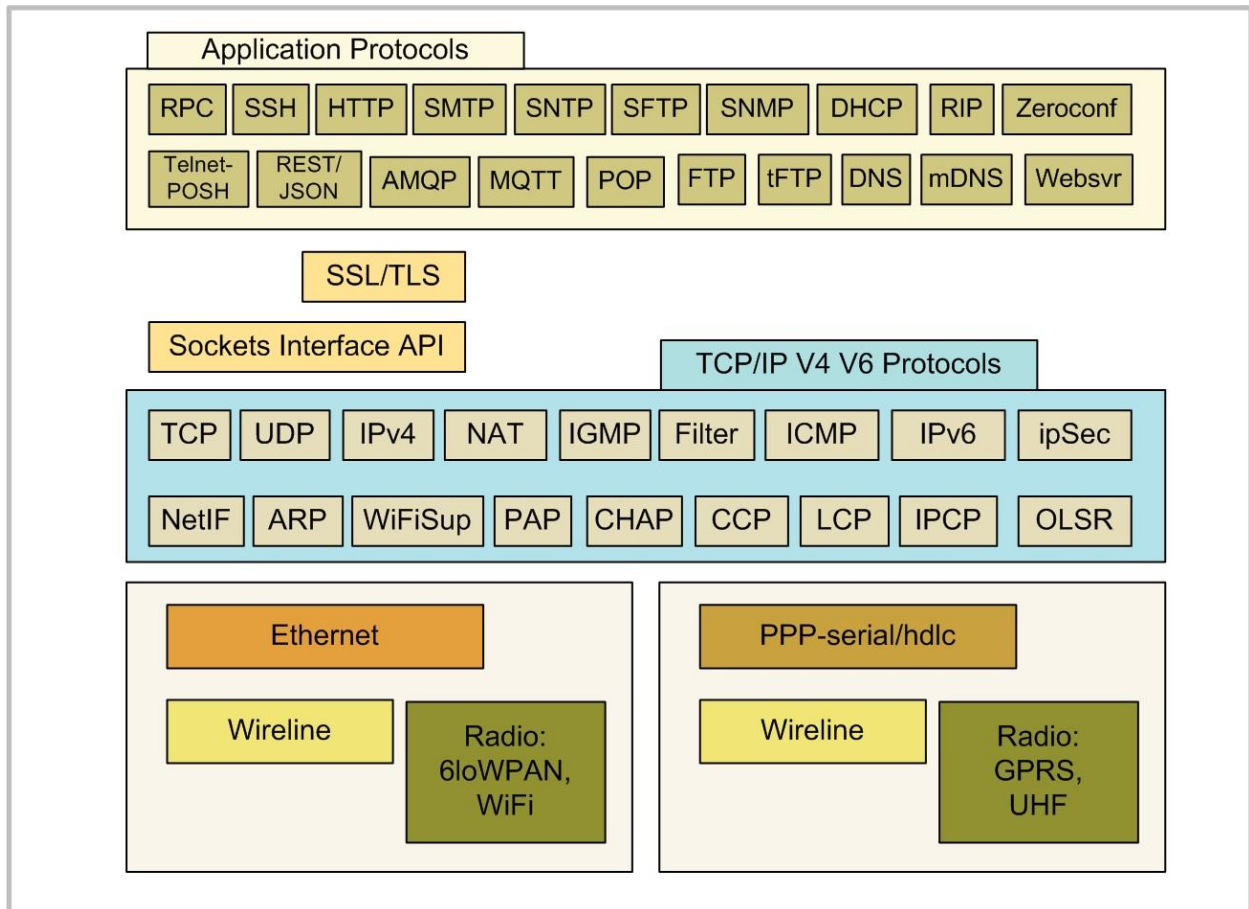


Figure 1: Application and TCP/IP protocols supplied with Unison are tested and proven so that they can be selected and integrated with the RTOS build as required by the application.

Unison will deliver virtually any connectivity solution that can be selected and integrated into the RTOS design off the shelf. That means that for the broad range of wired and wireless connectivity options, the Unison RTOS will be fully connected and ready to use.

Complete

Connected devices are connected to a wide variety of things in this Internet of Things. To be adaptable to the many possibilities while minimizing time to market requires access to a sufficient selection of other protocols as well. These connect such things as memory, mechanical, display, camera and other sensor systems. For example, mechanical devices are connected using a variety of technologies including PWM, A/D and D/A, and encoders for such things a step motors and more. Storage devices use a variety of SPI Flash, NOR and NAND Flash, RAM and MMC interfaces in addition to SATA. Then there is USB, which is gaining popularity for a large number and variety of devices. For cameras and displays, interfaces include USB, HDMI, LDC buses and touch sensors. Unison has it all off the shelf and ready to become a part of the system's RTOS.

Power management is also a critical factor for embedded and sensing devices, which are often placed in spots that are difficult to access. Many newer processors have special power saving modes of operation for such things as memory and on-board peripherals. Unison supports such features allowing users to minimize consumption by strategically taking advantage of special modes of operation that offer minimal power usage.

Cloud

The Internet of Things necessarily includes the Cloud where that data is gathered and evaluated, where high-level decisions are made. The Cloud is the source of program changes and updates to the connected devices, where everything is pulled together. Integrating with the Cloud involves selecting the Cloud platforms that are used along with their operating systems and applications, which should be compatible with those used on the connected devices. While compatible does not mean identical, compatible connection is needed for acquiring and storing data, and performing analysis without cumbersome conversion efforts. Unison RTOS supplies a broad set of protocols that can be used on the Cloud side to easily and securely connect and transfer data and process commands.

The Internet of Things is growing and evolving rapidly and is subtly becoming a greater part of everyday life. Keeping up with its development requires the latest components and tools to deal with it and its newer applications. In the Unison RTOS world this means the ability to work with other leading edge applications like Microsoft Azure. It also means the use of the best tools such as a wide selection among the IDE offerings along with advanced Unison support tools that allow use and tracking of Unison memory and objects such as dials, gauges charts for variable displays plus timing tools and displays to understand scheduling, interrupt processing and more.

Development Tools

The Unison approach to development tools is to provide an environment that is inexpensive yet versatile and expandable, giving the developer the opportunity to put together just the right environment for their projects. This means a strong emphasis on Eclipse-compatible integrated development environments (IDEs), which are available from multiple vendors for the range of processors supported by Unison. Since they are Eclipse compatible, these tool suites from multiple vendors fit easily into the Eclipse-based Remedy IDE for Unison, giving the developer a minimum of an editor, a compiler tool chain and a debugger in addition to any other tools included in that vendor's IDE.

RoweBots also offer the GNU C/C++ compiler (or a substitute compiler) for all supported processors. Embedded C/C++ developers have been migrating towards the GNU C/C++ compiler as their compiler of choice for some time. It is well known to offer reliable operation on a broad set of hardware targets with relatively low costs. The plug-in nature of the Eclipse IDE makes it easy for developers to switch out or add in Eclipse-compatible tools as needed.

In addition to the basics, there are two important tools supplied at no additional cost with the

Unison RTOS. These are the Remedy RTOS Viewer and the Remedy RED Analyzer, which are specifically designed for developing with Unison. The Remedy RTOS Viewer is integrated along with the IDE to support the tool suites from Keil, Mentor Graphics, Texas Instruments, Microsemi and more. Since it is integrated with the IDE, it is able to view all registered objects and internal kernel structures in the Unison environment, including, semaphores and their count, message queues mutexes, as well as threads and their status including stack usage and current state and memory pool status. Updates are made automatically to the information each time a breakpoint is hit.

The Remedy RED Analyzer features three types of analysis: **Remote control** features to set target variables and control dynamic event tracing, **Event time-based triggering** and displays, and **Data collection**, transfer and display. Using these features and their graphic displays allows a developer to visualize the timing and resource usage inside the Unison operating system and tune it to exactly fit the application's needs. Remedy RED consists of a host-based viewer and a remote control server and data logger on the target system (**Figure 2**). The logged data is sent to the viewer where the user can examine it in different views for detailed analysis. The user can also set triggering and logging options, collect and view the data and zoom in on problems.

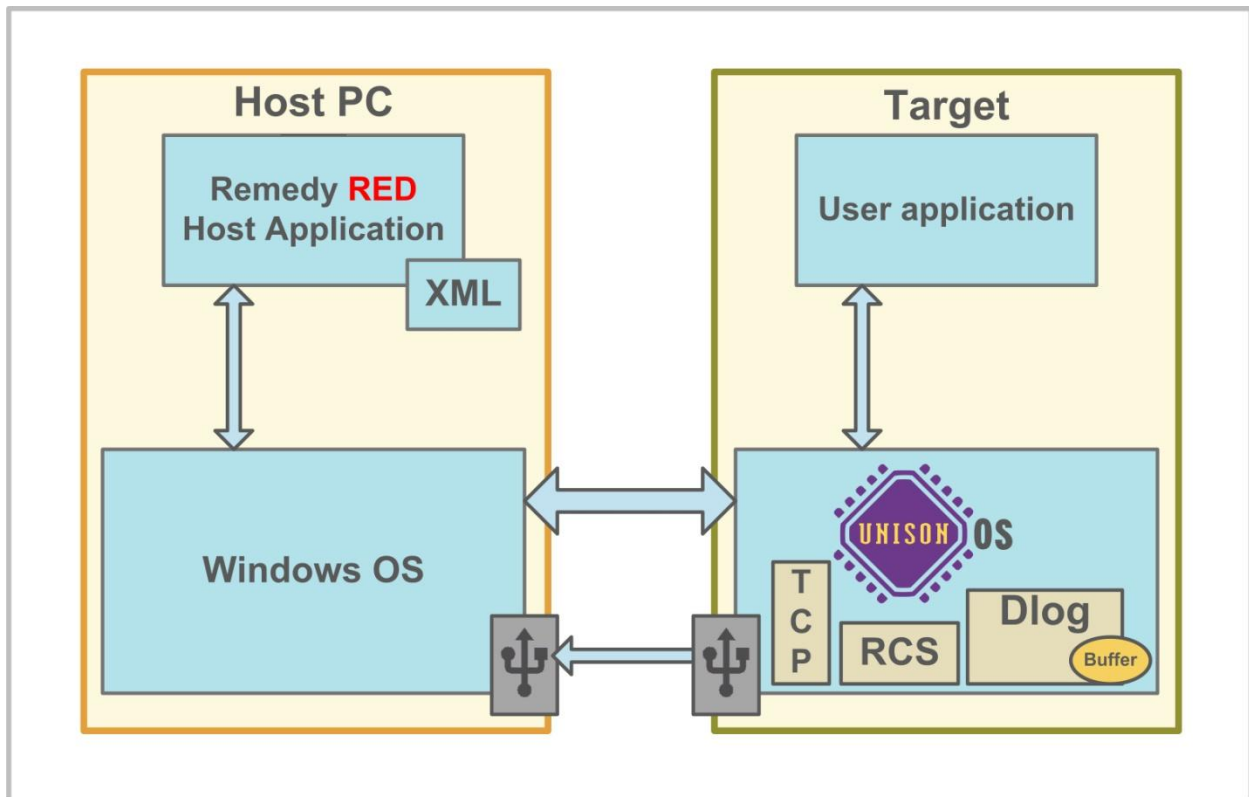


Figure 2: The Remedy RED Analyzer gives the developer full insight into the timing and resource utilization and behavior of the Unison RTOS.

One example of a Remedy RED Analyzer application is analyzing complex timing issues. For example, a system that has a broad set of I/O may rarely encounter a condition that leads to a system crash. The crash seems random and data collection is difficult. Using the Remedy RED Analyzer, the problem can be traced and a diagnosis made by setting up a trace on the suspected I/O and a trigger event to capture the conditions leading up to the crash. Examination of the crash data can then turn up clues to the failure. The user can expand the scope of the data to achieve a complete analysis and solution to the problem. Many other cases involving multithreaded timing behavior can be quickly captured and analyzed.

One other important tool supplied with Unison is the Remedy Bootloader. The bootloader allows a new program to be installed on the target system, which is useful both during development and for field upgrades since it can be done remotely. The bootloader also had an encryption/decryption option, which is important for doing secure remote updates. The reboot phase will flash or copy the program into execution memory and transfer control to the new program. If the new program fails, then the program will generally revert to the backup program and program this image. This approach is used to ensure that field service or product return is avoided at all costs.

The Open Standards Platform

Having a small, efficient, modular platform based on open standards like POSIX and Linux can offer truly significant savings in integration and testing, maintenance, time to market and the ability to adapt to growing needs and market opportunities. Among other things, it allows developers to take advantage of a wide variety of development tools and the world of open source software. The question might immediately come up, “Well can’t I get an open source kernel and save even more money?” The answer is, “Yes, you can get an open source kernel (for free), but it will cost you more money and time, which is money, than you would care to invest.” For one thing, you’ll initially be dealing with an unknown and how do you establish that it will even work for you let alone determine its actual performance?

Does it have all the components you’ll need for your project? If not, will the ones you do find work with this kernel? You’d have to test and at what cost in time and cash? The same if you write your own. What is the real memory footprint going to be? What is the interrupt latency? What are the security measures that come with it (if any)? What is the boot time? Can it work from flash? If you run into trouble, where do you turn for support? It turns out that “free” really isn’t free, especially at the foundation level of a development project. A Unison license comes with all these questions and more answered along with support and a maintenance program that will give you confidence along with a solid platform as a starting point for developing and adding your own unique value.

The use of open source can certainly have value at the application level where it will also need to be tested, verified and debugged in the context of the larger application. But doing that on the basis of a proven platform will bring home the true advantages of open source because lean and

modular go together. The solid, adaptable platform is already available in several preconfigured versions that include ConsumerOS, MedicalOS, WearableOS, and VehicleOS. From these, the development team can continue to customize by substituting in various combinations of proven software components such as networking protocols, protocols for security, wireless, file system and more.

Support and maintenance programs are provided at different levels according to user need and are sold in blocks of ten hour rather than as yearly contracts to make sure you pay for only the help you need. At all levels, email support and binary upgrades are included. The level of support and response increases from Silver to Platinum levels—up to and including source code updates and real-time phone support. This can greatly help reduce development costs, especially if we consider the cost of maintenance after the fact.

Finding and fixing defects increases drastically that later it goes in the development process and after shipment to a customer, it can be devastating (**Figure 3**). Finding and fixing defects early also greatly affects time to market. This, combined with the use of a pre-qualified development platform can speed that time and result in cost savings up to 1000x.

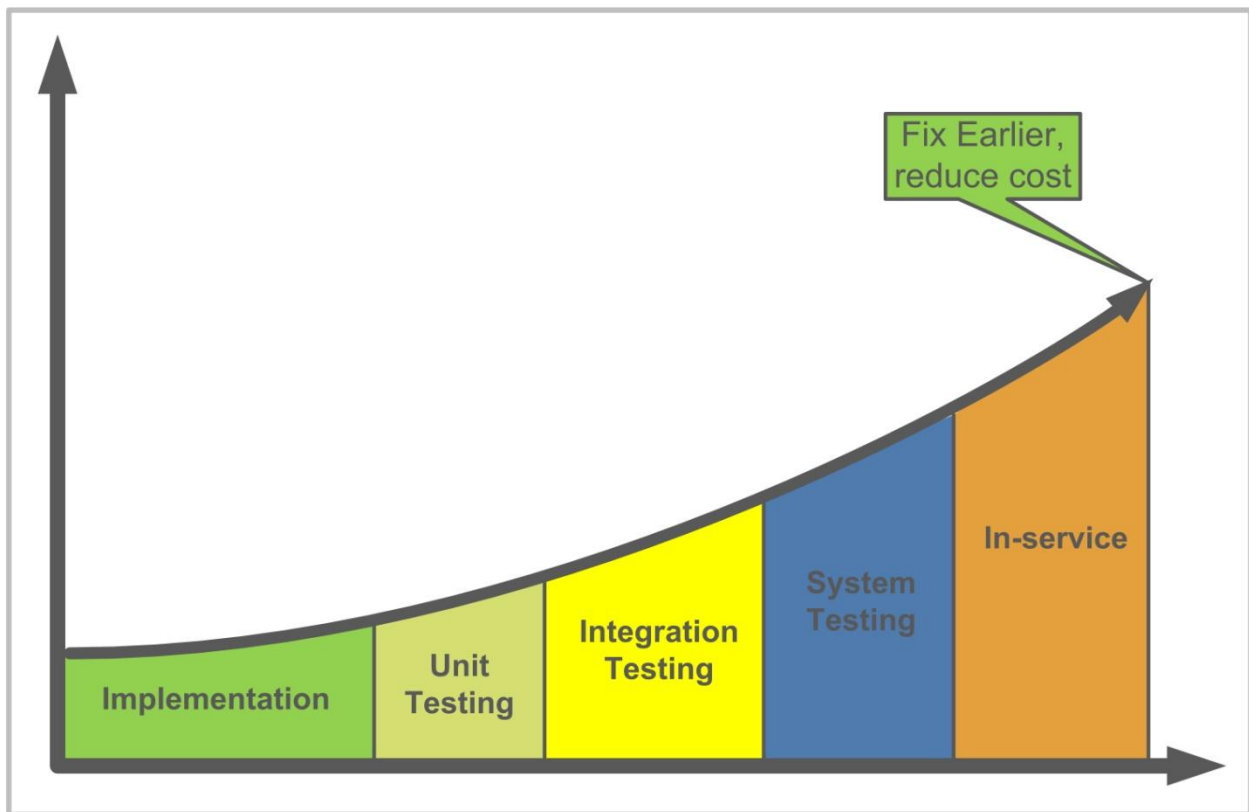


Figure 3: The more that can be implemented in the foundation with proven code and the sooner any defects introduced during development can be remedied, the lower the cost of doing so.

Within an adaptable hardware/software framework, new protocols and connectivity options are easy to add. This framework should be complete, providing all the components that the design requires. Additionally, security should be intrinsic to the framework and not "bolted on" as an afterthought. Typically, high quality framework platforms will also have open standards based APIs along with integration into a variety of Integrated Development Environments (IDEs).

Unison's advantage is that it provides all these options in a selection of proven modules and protocols for a wide variety of processors. These can be selected and utilized to produce the specific RTOS foundation needed to address the exact needs of a proposed application. With Unison, you can immediately start building your innovative product addressing your customer's needs while assuring forward compatibility to more advanced or specialized products.