White Paper

Wearable Technology with Unison: For Devices that are Small, Low-Power, Connected and Comfortable

The world of wearable devices is emerging as the newest and the most challenging area of the Internet of Things. Wearable technology is designed to carry out some specific function that necessitates its being worn and therefore must be comfortable and familiar in look and feel. In the best of cases it can be embedded in some familiar everyday article of clothing or jewelry such as a watch or a pair of glasses. On the other hand, it may be a special bracelet or a sensor carried somewhere out of sight on the body. In any event, one common purpose regardless of the specific application is that it must sense, store, interpret and communicate information about the wearer's body or surroundings (***Figure 1***).

Figure 1*: In addition to the latest wireless connectivity options, WearableOS supports all the latest sensors, radios, MCUs and MPUs for designing such devices off the shelf.*

Developers face multiple challenges in wearable technology. In addition to the pressures of cost and time to market, there are the factors of size, and power consumption. Of course, they must also address wearable devices security and safety. All these must be dealt with and solved while the developer must also produce the unique capability the device is intended to provide. The sooner he can start adding unique value, the faster he 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. RoweBots now offers such a platform specifically targeted at developing devices to be worn on the body or concealed in clothing. It is called Unison WearableOS and it combines the best features of a compact yet powerful RTOS combined with the functionality and components tailored specially for the needs of devices that can be worn comfortable and unobtrusively—and sometimes even provide wearable technology fashion.

WearableOS is based on the Unison RTOS from RoweBots, which 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 help the developer get started at the level of adding value and with the opportunity for later expansion and enhancement of a product line.

WearableOS incorporates the Unison RTOS with a special selection of peripheral support, communication protocols, wireless support, HMI, device and Cloud connectivity aimed to get the development project off the ground and adding value as quickly as possible. In addition, of the many microprocessors and microcontrollers supported by Unison, the WearableOS package is targeted as a select number of those most conducive to the specific needs of such systems.

The Unison Foundation

Unison itself offers a platform for very small systems that has a tiny memory footprint in the range of 20K of flash and 8K of RAM. A complete solution looking more like an ultra-tiny embedded Linux configuration might be in the order of 300K of flash using 70K of RAM depending on the configuration without security modules. With security modules, this might be more like 400K of flash and 100K of RAM depending on features selected—still a very small footprint.

Unison is based on a completely native POSIX implementation using nano-kernel technology. As such it does not include inefficient layers or mapping of functions. The Unison RTOS delivers complete POSIX RTOS implementations with file I/O, socket I/O and fully compatible error codes. The RoweBots POSIX RTOS implementation delivers ultra-tiny embedded Linux-compatible solutions that are a single process, multiple thread model with complete I/O and a tiny footprint and other significant improvements over embedded Linux variants that offer multiple process, multiple thread models with complete I/O which require an order of magnitude more resources. Such a small, modular and efficient RTOS environment can be fitted with just the right software modules for peripherals and wireless connectivity to act as the foundation of a truly small and efficient device.

Getting the Right Connectivity

Being properly connected means everything for meeting the challenges in wearable technology. That includes the connections to sensors and as well as wireless connectivity— to sensors, possibly to edge devices, the Cloud and also to other devices carried on the person such as watches and smartphones. It is safe to say that for such devices, wireless support is mandatory. The Unison WearableOS package offers a range of wireless connectivity options depending on the system design. Most often Bluetooth Low Energy (BLE) is used because it provides a direct phone connection with fast connections and low power. Other options such as Bluetooth Classic are required for full audio communications while video requires Wi-Fi and low data rate sensors can use 6loWPAN/802.15.4. Even 2G/3G/4G are available. Unison WearableOS has all the off the shelf wireless support included with the operating system and so these are known to work without having to be adapted.

Another connectivity issue addressed by WearableOS is the human interface. A small, low-power device does not usually have the ability to support a graphical interface or for that matter a number of lights and buttons. It depends on its ability to connect to another worn device such as a watch or a smartphone—or to a remote server—in order to be interactive. The Unison WearableOS package includes simple watch graphics and an Android/iPhone graphics package for use on smartphones. Communicating via Bluetooth classic or BLE lets the user interact graphically with the device. Connection to a phone, for example, also offers a means of communicating to the Cloud or Edge devices without sapping the device's power.

Minimizing Power Consumption

The use of battery power is to no one's surprise a very critical aspect of wearable design. This requires cooperation between the processor hardware and the operating system. The selection of processors supported by the operating system feature both inherent low power consumption as well as power management features that can be controlled by the operating system.

Processors supported by WearableOS include:

- **MCU**: STM32L1xx *(ST Microelectronics)*
- **MCU**: STM32L4xx *(ST Microelectronics)*
- **MCU**: STM32F303 *(ST Microelectronics)*
- **MCU**: RX111 *(Renesas)*
- **MCU**: RX231 *(Renesas)*
- **MCU**: TM4C1294NCPDT *(Texas Instruments)*
- **MCU**: TM4C123GH6PMI *(Texas Instruments)*
- **MCU**: MKL26Z128VLH4 *(NXP)*
- **BLE + CPU**: DA14581 *(Dialog Semiconductor)*
- **BLE + CPU**: nRF51822 *(Nordic)*
- **MCU**: The Apollo1 Cortex M4F-based *(Ambiq Micro)*

The new RX231 Group of MCUs from Renesas, for example, achieves power consumption levels as low as 0.2 mA/MHz when only the CPU is operating and the peripheral functions are halted. When operating in software standby mode, in which the contents of the on-chip RAM and registers are maintained but other functions are halted, the RX231 devices achieve an ultra-low level of 2 μA, contributing to better system power efficiency and longer battery life. The energy-saving architecture in the NXP MKL26Z128VLH4 is optimized for low power with 90nm TFS technology, clock and power gating techniques, and zero wait state flash memory controller. A range of power management functions is provided by this device as well. The operating system's power management features are tailored to the management functions of the respective processors.

Peripheral and Sensor Support

Given the size and power restrictions, a wearable device cannot be a multipurpose "utility knife" type system, but must focus on doing one or two things very well and very efficiently. However, that focused set of capabilities can vary widely and so it is important that a goodly selection of sensors be available to match the intended specific purpose of a device. RoweBots has identified a selection of sensors and peripheral devices, such as GPS chips, that can be quickly integrated into the design. These selected devices include:

From ***ST Microelectronics*** :

- **MEMS: LPS331AP** - high-resolution pressure sensor
- **MEMS: MPU-9150** - Nine-axis (Gyro + Accelerometer + Compass) MotionTracking
- **MEMS: LIS3DH** - ultra low-power high performance 3-axes "nano" accelerometer
- **MEMS: LIS2DM** - ultra low-power high performance 3-axis "femto" accelerometer
- **MEMS: H3LIS331DL** - low-power high-g 3-axis digitalaccelerometer
- **MEMS: LPS331AP** high-resolution pressure sensor Gyroscope: L3GD20

From ***Bosh :***

- **Pressure Sensor: BMP180**

From ***Texas Instruments :***

- Temp Sensor: **TMP006**
- Bluetooth:**CC2564MODN**
- Pwr Management: **BQ24075 USB** - Friendly Li-Ion Battery Charger
- Battery Fuel Gauge: BQ27510-g2
- Buck-Boost Converter (Regulator): **TPS63002**
- Haptic Driver: **DRV2605**
- PulseOximeter: **AFE4403**
- Battery Charger: BQ25101H
- WiFi + BT + BTLE: **L1837MOD**

From ***Intersil :***

- Light Sensor: **ISL29023**

From ***Cypress Semiconductor :***

- FLASH: **S25FL116**
- SPI FLASH: **S25FL512S**
- NAND FLASH: **S34ML02G**

From ***Panasonic :***

- Bluetooth: **PAN1326**
- Bluetooth: **PAN1315**

From ***Microchip Technology :***

- 6LoWPAN: **MRF24J40**
- LCD controller: **SSD2805**
- LCD: **1.54 MIPI Display**

From ***Lumex :***

- RGB LED: **SML-LX0404SIUOGUSB**

From ***u-blox :***

- GPS: MAX-7C
- GSM: LISA-U2

From ***RedPine :***

- WiFi + BT + BTLE: **RS9113**

RoweBots has the driver source code for all these devices. For any devices a customer selects, RoweBots can quickly compile and port the drivers for the specific MCU. In addition to these supported devices, new processors, wireless options and sensors are being added all the time so developers should consult with RoweBots to get the list of the latest available devices and find out what is on the roadmap.

The Unison WearableOS Package

Unison WearableOS comes with the Unison release for the appropriate MCU and the standard development board from the MCU manufacturer. In addition, there is a set of Unison-compatible software components that are selected to address the specific needs of wearable technology. Among these are the Unison kernel with serial I/O and a power-safe and wear-leveling file system for fixed media. This latter is important for Flash memory that will offer low-power nonvolatile storage on the person. Stored data may periodically be transferred wirelessly to an edge server or to the Cloud but local storage will definitely be a requirement in many cases.

Since the data involved is often of a very personal nature, it must be protected from eavesdropping and tampering. WearableOS includes the Transport Layer Security (TLS) protocol, which has succeeded the Secure Sockets layer (SSL) protocol. TLS works by using a handshake session between client and server in which the client requests identification and is sent a certification that includes the server's name and its public encryption key. TLS forms the foundation of other security protocols and strategies. With TLS as the basis for secure transport, WearableOS provides a host of ready-to-use wireless protocols as discussed in the section on connectivity.

And finally, in addition to the graphics and touch panel functionality mentioned earlier, WearableOS supports camera connectivity, which are devices that require fairly high bandwidth and wide area connections. So beyond the Wi-Fi support there is also support for cellular connectivity in the form of 2G/3G/LTE and now 4G/LTE-A. The proviso for developers, of course, is that such devices must be used with keen awareness of power consumption in applications using wearable technology.

TI Tiva Watch Kit

The Unison WearableOS also comes with a demonstration package in the form of a low-power watch kit based on WearableOS and the Texas Instruments Launchpad series of development boards. The watch includes such features as a 240 x 240 color display and a BLE connection to an Android smartphone. I has an optional sensor pack for environmental and vital signs monitoring, the ability to display accelerometer data on Android and an optional fuel gauge and battery. It allows control of the watch from the phone and the ability to monitor events from the phone.

In addition to the WearableOS operating system, the kit includes graphics for displaying time, weather and other data, a command interpreter for watch control, a simple Android control program and an Android accelerometer display. The kit also comes with the Code Composer Studio IDE and a rich complement of software modules to implement a watch. Among these are the core kernel, I/O model, serial I/O debugging interface and the file system. Bluetooth modules include BLE, BT Classic and BLE command processing. There is accelerometer control, timer and watch setup software and more.

The kit lets you get started quickly learning about WearableOS by developing a device using an open source hardware kit and the supplied software (***Figure 2***). The hardware comes with all gerber files and schematics that make it easy to install selected sensors. Then, taking the known hardware components, you can layout your own watch and design a case, and you will have a fully functional smart watch. The only things that are not free are the hardware components ($100 approximately) and the Code Composer Studio IDE, which may also be free depending on the application size. The Unison OS and all the software for development are free for prototyping.

Figure 2: The Tiva watch kit lets developers get hands-on experience working with the Unison WearableOS.

A Sample Design

A further design example for WearableOS can be used either to develop end user applications or as a trial development platform to modify for alternative hardware designs using components already supported by WearableOS. Delivered as a ready-to-program device is a bracelet that incorporates a selection of supported peripherals along with APIs for developing both Android and iPhone apps. The overarching concern is to provide high functionality at absolute minimum power consumption. That works out to a minimum of 16 hours with 25% on time. In addition, all the major components fit into a space of 10mm x 10mm ***(Figure 3)***.

The processor is an ARM Cortex-M0-based device from Dialog Semiconductor, the DA14581. It offers up to 8 connections with fully integrated Bluetooth low energy. It also supports a flexible memory architecture for Bluetooth profiles and application code that can be updated over the air.

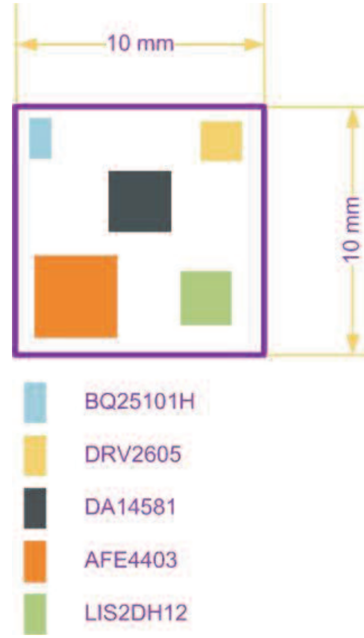

Figure 3: Size comparison of major bracelet components.

It interfaces directly to the antenna for ultra-compact configuration. The battery charger is a tiny 1.6mm x 0.9mm package that uses a micro USB connection and can charge overnight. It feeds a 20 mAh Li-ion battery that itself is only 3x10x10mm.

The bracelet incorporates two ways of alerting the wearer. One is an RGB LED from Lumex that offers options for configuring signals based on combinations of the three colors. The other is the DRV2605 haptic driver device from Texas Instruments. In this application, it receives notice of the phone ringing via the Bluetooth in the processor and vibrates to alert the wearer to an incoming call. This can be especially useful to women, who carry their phones in a purse.

Two other devices lend themselves to the development of creative apps. One is the Texas Instruments AFE 4403, and analog front end specifically suited for pulse oximeter and heart rate monitor applications. It features very precise timing characteristics and can be used in any number of fitness applications. The second is the LIS2DH12 3-axis accelerometer from ST Microelectronics. This is a very flexible device. It can be set for acceleration rates from 2g to 16g and can be set to generate interrupts based on various events. It also provides a fully configurable timing controller. It communicates with the processor via the SPI interface.

The Rowebots bracelet thus represents two opportunities for the developer. On the one hand it is a ready-made platform with a variety of creative apps using the supplied APIs. On the other hand, it provides an example of the sort of hardware configurations and designs that are possible using the wide variety of peripherals and protocols that are supported out of the box by Unison WearableOS.

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 tools 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 offers 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 and 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: **R**emote control features to set target variables and control dynamic event tracing, **E**vent time-based triggering and displays, and **D**ata 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 4)***. 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.

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

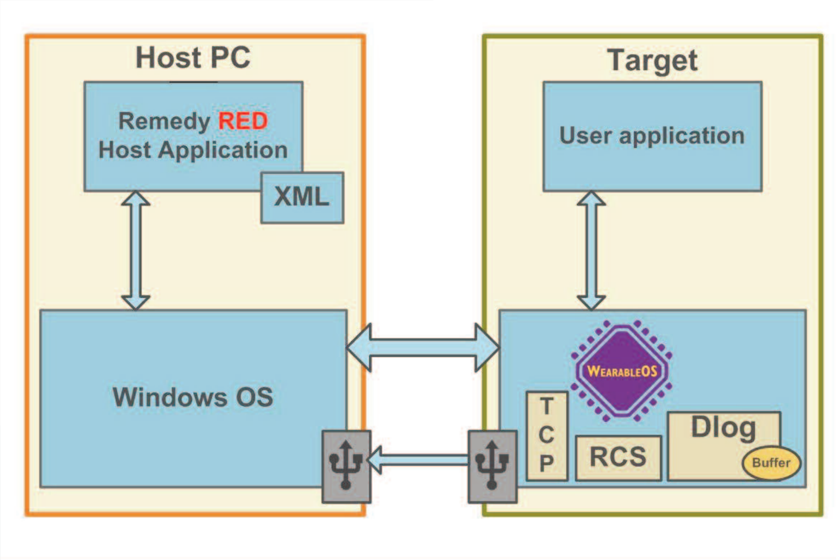

Figure 4. The Remedy RED Analyzer gives the developer full insight into the timing and resource utilization and behavior of the Unison RTOS.

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.

Building on a solid platform starts with the Unison RTOS, which is itself lean, built on standards like the POSIX API for interchangeable modules, protocols and drivers and adaptable, so applications can be easily enhanced, ported and customized for user demands. It includes a rich set of *security* features that can be put in place at the foundation to build a secure system and application set. The ability to build in safety is supported by determinism and zero boot time, which make it able to respond to an unsafe condition can be halted immediately and brought back to a safe condition or the device diverted to an emergency action for remedy.

Unison also provides completeness and connectivity through its rich set of available wireless and other communication modules, its support of memory, mechanical, display, camera, touch sensor and other sensor systems along with more universal connectivity via USB. And, importantly, Unison provides Cloud access with a broad set of protocols that can be used on the Cloud side to easily and securely connect and transfer data and process commands. Keeping up with developments in the Internet of Things 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. That includes the ability to make use of the latest tools and IDEs as discussed above.

Starting from the Unison platform, Unison WearableOS takes that solid set of characteristics and focuses them to the needs of those developing wearable technology. The core components of the Unison RTOS, the wireless communication protocols, the ready to go support of a rich set of sensors and peripherals, tiny graphics packages and Android connectivity, power management, a selection of MCUs and more put a set of components and tools into the developer's hand to get started adding value from day one.

Contact: sales@rowebots.com
+1 519 279 46 00

RoweBots
Limited