I. Introduction

Many products, particularly those in the consumer electronics market, use batteries and do not have a continuous non-interrupted source of power like a wall outlet. As such, they need to maximize the time between replacing or recharging the batteries. The Sensory RSC-4x line of microprocessors has been specially designed for use in applications where low overall power consumption is a major consideration. This design guide is intended as a guide for developers who want to make the best use of the RSC-4x's power-down modes. This guide should be used in conjunction with the other Interactive Speech™ design guides to ensure the creation of a high quality speech recognition product.

II. Power-Down and Low-Power Modes

Two supply current power-down modes are available – ***Sleep*** and ***Idle*** modes. For complete details on RSC-4x power and wakeup control, please refer to the RSC-4128 Data Sheet (80-0206) and/or the RSC-464 Data Sheet (80-0282).

A. Power-Down mode

1. Sleep mode – In this mode everything stops as only an I/O event can initiate a wake-up. This is the most efficient power-down mode and can consume as little as 1 micro amp or less.
2. Idle *without* Audio Wakeup mode - In this mode, unlike Sleep mode, OSC2 and Timer2 continue to run. An I/O event or Timer2 interrupt request caused by overflow can generate a wake-up from this mode. This mode can consume as little as 7 microamps.

B. Low-Power mode

1. Idle *with* Audio Wakeup mode – In this mode the Audio Wakeup Logic, external microphone and OSC2 continue to run. This is classified as a low-power mode rather than a power-down mode due to the larger current draw by comparison. This larger current is due to the Audio Wakeup logic and the active microphone. An I/O event, Timer2 interrupt request caused by overflow, or Audio event can generate a wake-up from this mode. The total current consumed in this mode is about 150 microamps.

III. Hardware Considerations

Most of the RSC-4x internal circuitry is automatically turned off when the chip enters power-down mode and does not require any special action by the application program. However, some circuits need to be configured to a specific state in order to minimize current draw.

A. GPIO

The RSC-4x has 16 (RSC-464) or 24 (RSC-4128) general purpose I/O pins, each of which may be configured in one of four modes. Each mode has its own special low power considerations.

1. Weak Pullup Input or Strong Pullup Input – In these modes, the I/O pin is configured as an input with a resistor internally connected to Vdd. If the pin is externally connected to a low voltage then there is a path through the internal pullup resistor to the external low where current will flow. In that case, the pin should be configured as a Hi-Z before entering power-down state.

2. Hi-Z pullup input – In this mode, the I/O pin is configured as an input without any internal resistor. This mode is recommended if the pin is externally tied high or low. However, if the pin is externally left unconnected and floating, then Hi-Z mode is not recommended, since it is possible for the pin to float to an undefined CMOS voltage which will cause current leakage. In that case it is recommended that the pin be configured as a weak or strong pullup input before entering low power state.

© 2006 Sensory Inc.

3. Output – In this mode, the I/O pin is configured as an output and is driven high or low. All RSC-4x output pins are static during low power mode and it is important that they be driven to the state that will draw the least current depending on the external connection. It is also possible to configure an input port as an output for power-down or low-power time, and then restore the original state on wakeup. For example, if a grounding switch is normally connected to a weak or strong input that is NOT used for wakeup, one could place the port line as a LOW output. That way, the switch would not draw current even if accidentally pressed.

B. Analog Front End, Audio Wakeup Circuitry and Microphone

In most RSC-4x applications, the Analog Front End and microphone are turned off in power-down modes to save battery life. However, if the Audio Wakeup feature is desired, then some circuits must be turned on.

1. Audio Wakeup Circuitry – The RSC-4x has an internal circuit used only during Idle with Audio Wakeup mode. In this mode it is necessary to leave the microphone preamp and the Audio Wakeup logic powered on. The RSC-4x's Audio Wakeup Circuitry consumes about 40 microamps.

2. Microphone – Virtually all RSC-4x applications have a microphone of some kind, usually an external electret microphone. An electret microphone acts as a current source, but also requires about 100 microamps to operate. If a design does not use Idle with Audio Wakeup mode, the external microphone power should be turned off. Refer to figure 1 for a sample schematic on how an application that uses some combination of Sleep and Idle modes might look. In this circuit, P2.5 is used to control the microphone power. During normal operation or Idle with Audio Wakeup mode, P2.5 is configured as a high output, thus providing power to the microphone. During Sleep mode or Idle without Audio Wakeup modes, P2.5 is configured as a low output, thus turning off the microphone and saving battery life.

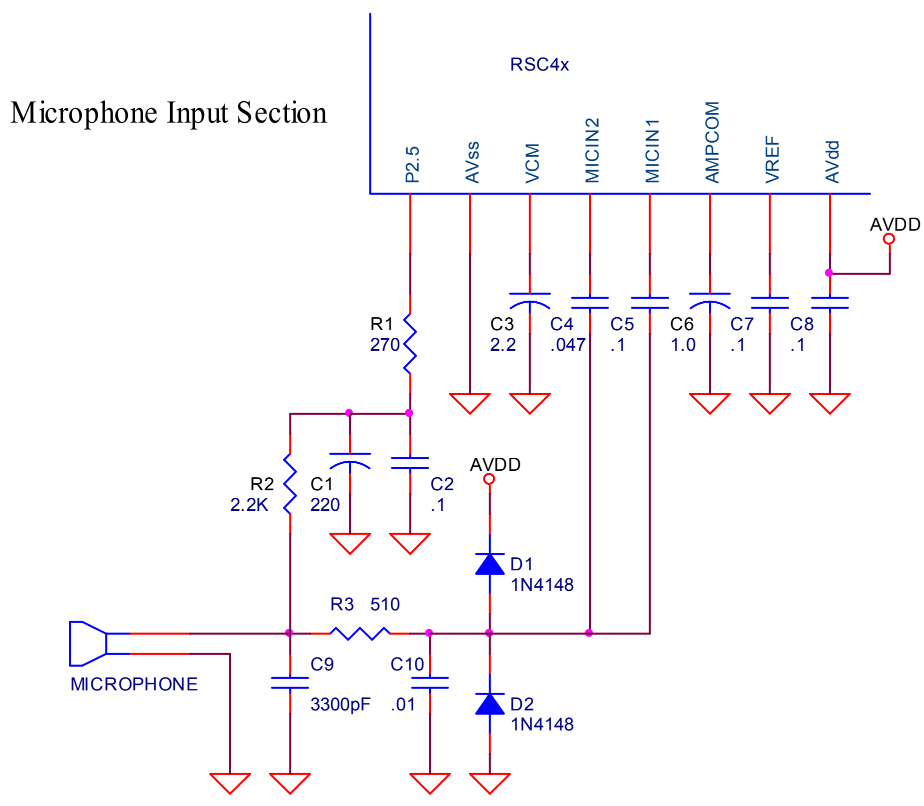

Figure 1

© 2006 Sensory Inc.

C. PWM and DAC

1. The PWM and DAC should be turned off and the outputs driven to a state that will draw the least power. The FluentChip macros SleepIO and GoIdle both take care of this automatically.

D. Clocks

There are two clocks in the RSC-4x, called OSC1 and OSC2. The main difference between Sleep and Idle modes is that all clocks are turned off in Sleep mode, while OSC2 is left turned on in Idle mode.

1. OSC1 – The 3.58MHz oscillator is the main clock for the RSC-4x and it is required to run at this speed for voice recognition and speech output. This clock is always turned off during low power mode. OSC1 takes 10mS to power up and stabilize – this should be taken into consideration when designing a low power system.

2. OSC2 – The 32.768KHz oscillator is optional and is not required unless the application implements timekeeping or uses Idle with Audio Wakeup mode. OSC2 can be driven from an external crystal, or internal RC oscillator. If the internal RC oscillator is used, the RC mode MUST be explicitly set by the application program. If not, the OSC2 will draw excess current, in both normal operation, and in Idle modes. Of course, the oscillator will also not reliably oscillate, or operate at a predictable frequency. It is also possible to run the RSC-4x's processor clock from OSC2, which can save current in some low power systems.

IV. Software Considerations

There are a number of FluentChip macros/functions available for programmers to use in their applications. Complete source code for all listed functions is available in the source\ subfolder of the FluentChip install folder.:

A. Functions

1. SleepIO, GoSleep – Use these functions for Sleep mode.
2. IdleT2 – Obsolete function reserved for legacy purposes. Use GoIdle instead.
3. GoIdle – Use this function for Idle without Audio Wakeup mode and Idle with Audio Wakeup mode.
4. SetupI/O, SetupAudio – Use these functions to configure wakeup conditions.
5. StartTimer2, StartTimer2NoDelay – Use these functions to start OSC2 and Timer2. Typically, these only need to be called once at the start of your program.
6. AfeOn, AfeOff – Use these functions to manually turn the Analog Front End (AFE) on or off. In most cases this is handled automatically by the SleepIO and GoIdle.
7. SwitchToOsc1, SwitchToOsc2 – These functions switch the RSC-4x processor clock to run from OSC1 or OSC2. Please note that OSC2 should be previously turned on in order to run from it.

B. Clock Configuration On Wakeup

If developers intend to use sleep or idle mode, they should always use the "GoSleep", "SleepIO", "GoIdle" or "IdleT2" functions and macros provided in the Sensory FluentChip library to ensure proper clock configuration when coming out of sleep or idle mode. Failure to do so may result in some initial instructions being improperly executed after wakeup.

V. Batteries

A. Capacity

There are several commonly used types and sizes of batteries. A representative table is shown here. Please note that this table is intended only to be representative and there are large variations among manufacturers.

	Carbon zinc (mAH)	Alkaline (mAH)	Rechargable NiMH (mAH)
9V	400	565	150
AAA	540	1190	750
AA	1100	2700	2100
C	3800	7900	2200
D	8000	17000	2500

Table 1

B. Voltage

Batteries drop in voltage as they are used. For example, while AA batteries are nominally rated at 1.5V, a brand new battery will usually start out its life at about 1.65V, then drop to 1.5V as it is used. As it is further used, it's voltage will continue to drop until it is completely depleted, at which point the measured voltage will be close to 0V. A representative battery depletion chart is shown in Figure 1.

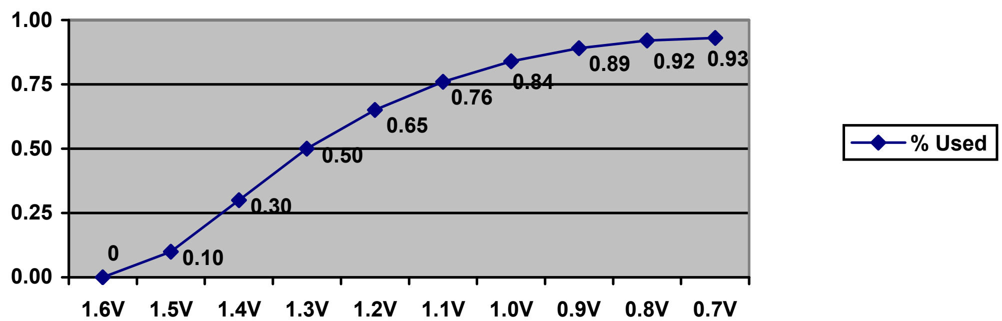

Figure 2

C. Lifespan

While the above table indicates the theoretical capacity of a battery if it could be completely drained, in practice this is never possible in an RSC-4x product. The RSC-4x has an operating range of 3.6V down to 2.4V and when the voltage at the RSC drops below 2.4V, it stops working and the batteries must be replaced, even if they still have some energy left.

VI. Common Battery Powered RSC-4x Designs

A. Without Regulation

One of the most common application designs is to use two 1.5V batteries in series, as shown in figure 2. In this application, Vcc will start out at about 3.3V (2 x 1.65V), and drop to 2.4V (2 x 1.2V), at which point the RSC-4x stops working and the batteries must be replaced. This application uses about 2/3rds of the total energy in the batteries.

© 2006 Sensory Inc.

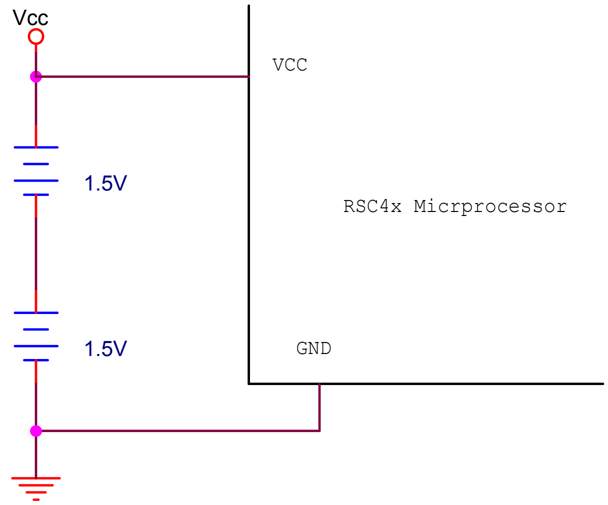

Figure 2

B. With Regulation

If a regulator can be used in the design, as in Figure 3, then it is possible to substantially increase the time required between battery changes. In this application, Vbatt will start out at about 4.95V (5 x 1.65V), but is regulated down to 3.3V at Vcc. The voltage can drop all the way down to 2.4V (3 x 0.8V), at which point the voltage goes out of regulation, the RSC-4x stops working and the batteries must be replaced. This application uses about 90% of the total energy in the batteries.

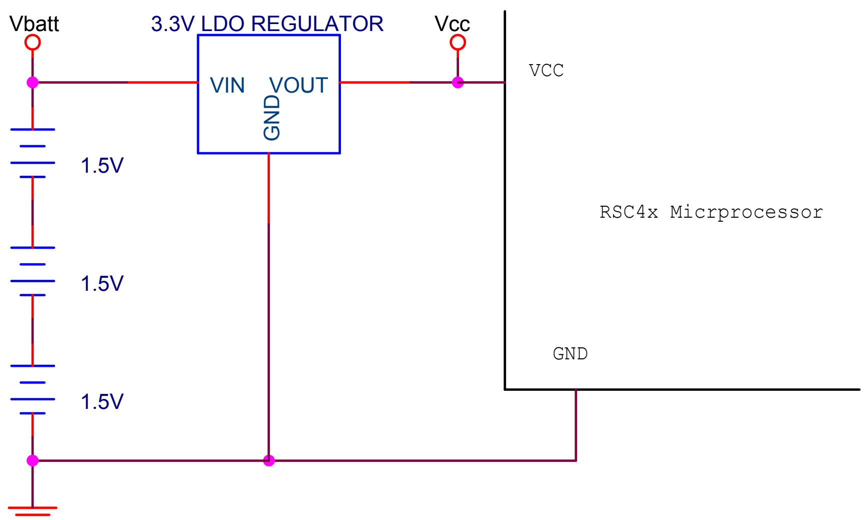

Figure 3

VII. Calculating Low Power Mode Battery Life

The formula for determining the theoretical maximum lifespan of a system on a change of batteries is:

```
Lifetime =  Ibatt * PCTused / Idraw
Where:      Ibatt = Total battery capacity
            PCTused = the percentage of total capacity that can be used
            before the voltage drops below the RSC-4x minimum threshold
            Idraw = the average current draw of the product.
```

Example 1: A TV remote control application consists of a RSC-4x microprocessor in a single-chip solution. Two Alkaline AAA batteries power the system. A microphone is used, but is turned off when not in use. The RSC stays in SleepIO mode most of the time but is run at full power for about 12 minutes/day. (= 0.5 minutes/hour). The system draws 10 microamps in SleepIO mode and 10 milliamps in normal (full power) mode. The estimated Lifetime of the batteries in this system is:

```
            Ibatt = 2 AAA cells = 1190 mAH (Section 5-A)
            PCTused = .65 (Sections 5-B, 6-A)
            Idraw(sleep) = 10 uA (=.01 mA)
            Idraw(wake) = 10 mA
            Idraw = ([Idraw(sleep) * 59.5] + [Idraw(wake) * 0.5]) / 60
                  = ([.01 * 59.5]) + [10 * 0.5]) / 60
                  = (0.595 + 5) / 60
                  = 0.09325 mAH/hour
Lifetime =  1190 mAH * .65 / .09325 = 8295 hours (= 345 days)
```

Example 2: The same system is devised as in Example 1, except that Audio Wakeup mode is used to allow the user to wake up the remote by a double hand clap. Also, Alkaline AA type batteries are used to compensate for the increased current needs of the system. The RSC stays in Audio Wakeup mode most of the time but is run at full power for about 12 minutes/day. (= 0.5 minutes/hour). The system draws 200 microamps in Audio Wakeup mode and 10 milliamps in normal (full power) mode. The estimated Lifetime of the batteries in this system is:

```
            Ibatt = 2 AAA cells = 2700 mAH (Section 5-A)
            PCTused = .65 (Sections 5-B, 6-A)
            Idraw(awmode) = 200 uA (=.2 mA)
            Idraw(wake) = 10 mA
            Idraw = ([Idraw(awmode) * 59.5] + [Idraw(wake) * 0.5]) / 60
                  = ([.2 * 59.5]) + [10 * 0.5]) / 60
                  = (11.9 + 5) / 60
                  = 0.28 mAH/hour
Lifetime =  2700 mAH * .65 / .28 = 6268 hours (= 261 days)
```

VIII. GoIdle Considerations

Special consideration must be given when the program calls GoIdle with T2Wake set to wake up once every second, as when a Real-Time Clock (RTC) must be kept.

A. Fast and Slow Clocks

The RSC-4x has two oscillators, OSC1 which runs at 3.58 MHz and OSC2 which runs at 32.768 KHz. Either one may be selected as the main processor clock. While the RSC-4x draws much less current when running from OSC2, it is important to remember that instructions take much longer to execute. In other words, there is little if any net power savings from simply switching the processor clock to OSC2.

B. OSC1 Power-on Time

There are times when it is beneficial for RSC-4x applications to select OSC2 as the processor clock and turn off OSC1 before calling GoIdle. The reason is because when the RSC-4x wakes up every second, and if OSC1 is selected as the processor clock, there is a 10-20 millisecond delay before OSC1 fully powers up and stabilizes and can run any instructions. During this turn-on time the RSC-4x consumes much more current than if OSC2 had been selected as the processor clock and OSC1 had been turned off.

C. T2 Interrupt Time

In some cases, it is less costly in battery current to use OSC1 as the processor clock even after wakeup from calls to GoIdle, under the following conditions:

 © 2006 Sensory Inc.

1. If the RSC-4x is running from an external code memory, and/or the application has a lot of peripheral devices that will consume current when the RSC-4x is running in,
2. **AND,** If the T2 interrupt service routine is very long,
3. **THEN**, it can turn out to be more costly in current to run from the slow clock.

The reason is because most external memory and peripheral devices consume the same amount of power whether they are used at fast or slow clock speeds. The programmer must consider whether it is more costly in current to have a slow running T2 interrupt in a system with lots of peripheral devices drawing current, or a fast running T2 interrupt with a 10-20 millisecond turn-on time and choose the option which will draw the least overall current.

The Interactive Speech™ Product Line

Sensory's **Interactive Speech™** product line makes consumer electronics more intelligent by enabling them to talk and hear with speech synthesis, voice recognition, and other advanced audio and interactive technologies. It is designed for integration into cost-sensitive consumer electronic applications such as home electronics, smart toys, music players and personal communication devices. The hardware line includes the award-winning RSC-4x family of mixed signal processors, the *VR Stamp*™ 40-pin DIP module, and the SC-691 music and speech synthesis slave processor. Embedded software options include our FluentSoft™ Recognizer, which enables speech recognition on non-Sensory processors and DSPs. Sensory's BlueGenie™ Voice User Interface, the first Voice Recognition and Synthesis option for BlueTooth enabled devices, offers user friendly control of headsets, music players and other BT devices requiring hands-free operation.

RSC Microcontrollers and Tools

The RSC product family contains low-cost 8-bit speech-optimized microcontrollers that are fully integrated and include A/D, pre-amplifier, D/A, RAM, and ROM circuitry. With Sensory's FluentChip™ firmware, the RSC family offers speech recognition, speaker verification, speech and music synthesis, voice recording and playback, and an entire suite of interactive robotic and sonic networking technologies. The family is supported by a complete suite of evaluation and development toolkits that include the ability to quickly create speaker independent recognition sets in many languages.

Speech Recognition Modules and Tools

The VR Stamp™ is a complete speech recognition module based on the RSC-4x and is ideal for fast design and easy production. A low-noise audio channel and standardized 40-pin DIP footprint allow rapid prototyping, less debugging, and shorter time to market. The *VR Stamp Toolkit* includes everything needed to get started today, including VR Stamps, Module Programming Board, sample applications, and a complete set of development tools featuring the Phyton IDE and limited-life C compiler, QuickSynthesis™ 4 and Quick T2SI-Lite™ speech tools.

SC6 Slave Processor and Tools

The SC-691 is a standard slave synthesizer that accepts compressed speech data from other microprocessors or microcontrollers and converts it to speech. The chip operates up to 12.32 MIPS, and provides high-quality, low data-rate speech compression and MIDI music synthesis, with unlimited speech duration using external memory. Sensory offers hardware and software tools for analyzing speech files, editing speech data and generating coded speech.

FluentSoft™ Recognizer

The FluentSoft™ Recognizer is the engine powering the FluentSoft**™** SDK. It provides a noise-robust, large-vocabulary, speaker-independent solution with continuous digit recognition and word-spotting capabilities. This small-footprint software recognizes thousands of words and runs on non-Sensory processors including Intel XScale, TI OMAP, and ARM9, and supports operating systems such as MS Windows, Linux, and Symbian.

BlueGenie™ Voice User Interface

The BlueGenie Voice Interface software suite runs on CSR's BC-5 MM Kalimba DSP, and enables manufacturers of Bluetooth products to integrate full voice control and synthetic speech output without the need for visual displays or complex user interfacing. It frees designers to pack functionality onto small form factor Bluetooth devices and answers consumer demand for a truly hands-free experience.

Important notices:
Sensory Incorporated (Sensory, Inc.) reserves the right to make changes, without notice, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Sensory, Inc. assumes no responsibility nor liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask-work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask-work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustrative purposes only. Sensory, Inc. makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Safety Policy:
Sensory, Inc. products are not designed for use in any systems where malfunction of a Sensory, Inc. product can reasonably be expected to result in a personal injury, including but not limited to life support appliances and devices. Sensory, Inc. customers using or selling Sensory Incorporated products for use in such applications do so at their own risk and agree to fully indemnify Sensory, Inc. for any damages resulting from such improper use or sale.

SENSORY®

575 N. Pastoria Ave, Sunnyvale, CA 94085
Tel: **(408) 625-3300** Fax: **(408) 625-3350**

© 2006 SENSORY, INC. ALL RIGHTS RESERVED.
Sensory is registered by the U.S. Patent and Trademark Office.
All other trademarks or registered trademarks are the property of their respective owners.