

## I. Overview

The successful application of speech recognition in a product depends heavily on the specification of the product and the proper use of recognition technology. This document provides an overview of the speech recognition technologies and software design guidelines for increasing the reliability of speech recognition in consumer products. Effective use of speech recognition technologies not only enhances recognition accuracy, but also reduces memory requirements and overall product cost.

Speech recognition gives a product the ability to “*listen and understand*”. It is straightforward to create robust speech recognition products using Sensory’s FluentChip™ speech recognition technology. This document will focus primarily on Sensory’s FluentChip™ technology solution for Sensory IC’s.

## II. Speech Recognition: Independent vs. Dependent

Determining the best type of speech recognition to use is a key element in the product’s specification. FluentChip™ technology offers two general classes of speech recognition, plus two derivative classes discussed below.

**1. SI** – In speaker-independent (SI) recognition, the product is *pre-trained* with words and phrases. The product is ready to use “out of the box” and requires no additional training by the user. An example of SI recognition is a voice-operated dimmer lamp with recognition words such as “brighter”, “darker”, “full brightness” and “turn off”. SI commands are typically stored in the on-chip code memory of the Sensory speech recognition IC. In FluentChip™ this technology is called Text-to-SI (T2SI), because SI command sets can be generated from text input of commands in a matter of seconds, using the associated software tool QuickT2SI.

**2. SD** – In speaker-dependent (SD) recognition, the user trains the device to recognize his or her voice by speaking each of the recognition words during a training phase. The product stores a template of the spoken words so it can recognize them when later spoken by the same user. Proper flow design of the training makes it a quick and simple process. Examples of SD recognition include personalized names in a PDA telephone address book, or a list of the user’s favorite or most often watched channels in a TV remote control. SD commands can be stored in on-chip RAM, or in off-chip memory. EEPROM or Flash is a common choice as it offers permanent storage of SD commands during power losses. The choice of storage is a tradeoff between cost and the value of permanent storage.

**3. T2SISD** – Some products require both solutions, where “out of the box” T2SI commands are enhanced with customized SD commands. This allows for simultaneous T2SI and SD recognition. In FluentChip™ this combination technology is called Text-to-SI+SD (T2SISD) recognition. The technology “votes” between the two different recognition types to determine the best match. When invoked, T2SISD offers commands that work “out of the box” and can be later enhanced with custom user-trained SD commands. In contrast, the separate technologies T2SI and SD may exist in the same product, but may not be simultaneously recognized.

**4. SV** – A variant of SD recognition is a biometric password technology called speaker verification (SV) recognition. SV is used to implement voice password solutions. As with SD, it is a speaker-dependent, but it focuses on rejection of imposters. Like SD, the password can be stored in on-chip RAM or off-chip memory, depending on the value placed on permanent storage.

There can be multiple T2SI and/or SD/SV sets in a single product, depending on the product specification and available memory. Any set can be used for recognition as needed under the control of the application program.

### III. Improving Speech Recognition Accuracy

There are six keys to attaining the greatest possible recognition accuracy:

- A. Selecting an appropriate recognition vocabulary.
- B. Using trigger words when appropriate.
- C. Designing speech synthesis prompts to elicit appropriate user response.
- D. Tuning recognition to control false-accept/false-reject errors and confidence scoring.
- E. Managing background noise.
- F. Designing hardware for proper gain and low electrical noise

#### A. Selecting an appropriate recognition vocabulary

The maximum number of SI or SD words in any given set depends on which Sensory IC is chosen. The RSC-4128 has more memory than the RSC-64 and therefore can store and operate on larger vocabularies. The speech recognition success rate depends primarily on the following four parameters:

**1. Limit the number of words in each set** – A recognition set should only contain the list of words and phrases that the user is allowed to speak at recognition time. ALL unnecessary words should be removed from recognition sets whenever possible. Every time a speech recognition product “hears” a word, it compares that word against those in the active recognition set. The more words a recognition set contains, the more likely the recognizer will make *substitution errors* (where the user says one word or phrase in the set, but the recognizer mistakes it for another), especially in noise.

For example, a program that recognizes a set containing “yes” and “no” at one point might later recognize a different set containing “dog” and “cat”. For best accuracy, these should be maintained as two distinct sets of two words each, and not combined into a single four word set containing “yes”, “no”, “dog” and “cat”.

**2. Use phonetically distinct words to reduce substitution errors** – To avoid substitution errors, choose words that do not sound like each other. For example, if the original set is “cat” and “rat,” recognition accuracy is improved if the recognition set is changed to “cat” and “mouse”. It is also better if the words in the command set have differing numbers of syllables, such as in the set, “dog”, “camel”, “elephant” and “rhinoceros”.

**3. Use very long and very short words with caution** – Notwithstanding the suggestion in #2 above, very long words and phrases (more than 4 or 5 syllables) contain a lot of phonetic information to match against. They are therefore less prone to *false-accept errors* (where an unintentional sound is mistaken by the recognizer for a word or phrase in the set), but they are sometimes more prone to *false-reject errors* (where the user says a word or phrase in the set but is not recognized). Very short words (1 syllable) tend to have the exact opposite advantages and disadvantages. Sometimes it is better to have a lot of very long words, for example, in a design where the risk of false-accept errors must be minimized. Or it may be better to have a lot of very short words in a system where the program must be as responsive as possible. Most programs can use normal length words with a few very long and very short words without problems.

**4. Use words with Plosives and Fricatives** – Some sounds are easier to hear in the presence of noise than others. The easiest type of sound to hear is called a plosive (in English, the voiced plosives are “b”, “d”, “g” sounds and the unvoiced plosives are “p”, “t” and “k” sounds). Fricatives (in English the voiced fricatives include “z” and “v” sounds and the unvoiced fricatives include “s”, “sh”, and “f” sounds; the sounds “ch” and “j” are combinations of a plosive and a fricative) are also easy to hear because they are usually higher in frequency than the background noise. Generally a voiced sound is easier to hear than an unvoiced one. The hardest sounds to distinguish from the background noise are the sonorants (in English “l”, “m”, “n”, “r” and “w”) and vowels (in English, the various sounds made by “a”, “e”, “i”, “o”, “u” and sometimes “y”). When there are a lot of sonorants in a row without any plosives or fricatives, it is difficult to clearly recognize all parts of the word. The best recognition words combine alternating fricatives/plosives and sonorants/vowels. For example, the word “goodbye” may be less likely to be falsely rejected in noise compared to the word “hello”.

## B. Using trigger words when appropriate.

Triggers have two uses: they provide a convenient way to get a particular product’s attention, and they provide a way to dramatically decrease the false-accept error rate. There are three key solutions for triggers in FluentChip™ – the T2SI trigger, the SDWS trigger and the T2SISD trigger.

**1. Using a trigger to identify a product** – Sometimes a trigger is used to identify one of several speech recognition products in a room. For example, in a speech recognition multimedia center including a TV, DVD player, stereo and other devices, each should have its own unique trigger name. This allows the user to trigger a single device and say commands to it that won’t be mistakenly acted on by other devices.

**2. Using a trigger to increase accuracy** – Using a trigger substantially reduces the likelihood of false-accept errors in a subsequent command set. False-accept errors occur when background speech or noise unintentionally matches an active command or trigger. For example, a command set with 10 words or phrases will have a random false-accept error rate about 10 times that of a set with only one command or trigger.

Consider a wall clock that announces the time when you say the word “time”. In a noisy environment such as an office or a conference room, it will sometimes announce the time even though the command might not have been intentionally spoken. This occurs when the chip hears a sequence of sounds that is phonetically similar to the command. If the false-accept error rate is too high, the constant announcement of the time will become annoying to the user.

Now consider the same wall clock product that uses a trigger phrase “voice clock”, plus a command “what time is it?” In this case, the false-accept error rate will be substantially reduced for two reasons. First, not one but two phrases must be recognized - in other words, if the hypothetical false-accept error rate of a single trigger (“wall clock”) is 1% and the false-accept error rate of a command is 5%, then the same error rate for a single trigger plus one word command is  $1\% * 5\%$  or 0.05%, reduced by a factor of 20. Second, replacing the short command word “time” with the longer command phrase, “what time is it”, further reduces the error rate.

Note that while it is possible to reduce the error rate of a speech recognition system nearly to zero, it is NOT POSSIBLE to reduce the error rate to exactly zero. All input systems have an inherent error rate – even manual input devices like pushbuttons and keyboards have a rate of failure which is greater than zero. Since it is not possible to completely eliminate the chance of errors, a well-designed product must expect and be able to manage them.

**3. Recognition Trigger Types** – There are three types of triggers: T2SI, SDWS, and a combination type called T2SISD.

### T2SI

The T2SI trigger is quickly and easily created with the QuickT2SI™ tool. There is even a check that helps ensure a good choice for the trigger. This is the best choice if the trigger is a fixed “name” and user-customization is not required.

Even more than for command recognition phrases, it is important to choose T2SI trigger phrases carefully. The most common challenge in choosing a trigger phrase is choosing a phrase that works for the product but that is not prone to false-accept errors.

As described above, command phrases that are long and have a mix of plosives, fricatives, and sonorants are both recognizable in noise and less prone to false-accept errors. An ideal trigger phrase is at least four syllables long and contains one or more fricatives.

For example, “Voice Clock” has a mixture of sounds including fricatives, but is only two syllables. “Sensory Clock” is four syllables and contains fricatives, plosives, and other sounds, so it would be a good candidate for a trigger phrase. Think about combining the name or brand of the product, the product type, and other phrases such as greetings to make a trigger phrase that will work well for your product.

### **SDWS**

The SDWS (speaker dependent word spot) trigger requires training, as do all SD technologies. The word spotting aspect of this SD variant allows it to find words in a stream of speech, without being bracketed by relative silence. This makes it robust in noise. SDWS triggers are the best choice if the trigger is customized.

### **T2SISD**

The T2SISD trigger allows for an SI trigger that can later be enhanced by an SD trigger. This is the best choice if the product must have a name “out-of-the-box”, but must also have the option of user customization.

When using T2SISD for trigger recognition, it is still important to follow the guidelines above in choosing the T2SI trigger phrase.

## **C. Designing speech synthesis prompts to elicit appropriate user response.**

Speech synthesis gives the product the ability to “talk.” It creates products that are user-friendly, fun to use, allow the product to instruct the user what and when to say, and provide feedback to the user. In a well designed product, speech synthesis replaces the need for a written instruction manual by giving real-time help at any point in the program flow. It may be a feature that gives “character” to a product by providing speech with personality and interactive dialogue. Synthesis may require external memory – which increases product cost – but its value in the product can be substantial and may help define the product concept. Music and sound effects can also be used to enhance the user’s audio experience. Speech, music and sound effects are all supported by Sensory’s QuickSynthesis™ tool, a companion tool to FluentChip™.

**1. Phrasing the recognition prompt** - For speech recognition to work successfully, a speech synthesis prompt should be constructed to encourage a user response that is in the active recognition set.

For example, “Where does rain come from: clouds or sunshine?” is a good question because it targets two specific vocabulary items. Both possible answers, “clouds” and “sunshine”, should be part of the recognition vocabulary, so that the product gives the appropriate response to the answer the user chooses. Similarly, the question “What is my favorite food: hamburgers, pizza, or spaghetti?” is an effective question prompt. The user is clearly guided to say one of the choices given in the prompt. When the user is presented with a multiple choice question, not only can the product now recognize the correct answer, but also the incorrect ones. It can provide tailored responses such as, “Rain doesn’t come from sunshine, silly! Try again.”

For more information refer to Sensory Design Note 80-0050-E (Recording Speech for Synthesis).

**2. Follow-up Questions for Clarity** - The program can also interact with the user through speech synthesis to clarify responses. When recognizing commands (especially a large set of commands), Sensory’s T2SI recognition technology calculates the probability of success. The product can be designed to prompt for additional clarification if the desired probability of success hasn’t been achieved. For example, if a recognition doll is told to “walk” and the recognizer matches that with high confidence, it can accept the command and begin

to walk. If it has only medium confidence, the program can prompt the user to confirm with something like, “Did you say walk?” The “yes” or “no” recognition set is only two words long and is more likely to be recognized with high confidence than a larger set. This provides a robust method of getting the right answer. If low confidence is returned, the program can ask, “What did you say?” and begin the recognition task again.

The following phrases might be included in recording sessions for all products using synthesis. These phrases may help improve overall product accuracy by notifying the user of possible problems:

- ▶ Did you say... [each recognition item]?
- ▶ What did you say?
- ▶ Talk louder.

## D. Tuning recognition to control false-accept/false-reject errors and confidence scoring

There are two methods to controlling recognition. One method controls whether a word is recognized or not, and the other method controls the confidence with which the word is recognized.

**1. False-accept and false-reject errors** - There is an inherent, inversely proportional relationship between false-accept and false-reject errors in speech recognition. The goal of any speech recognition system is to reduce the chance of both as much as possible, but in some products it may be desirable to decrease the relative chance of one type at the risk of increasing the other type. For example, in a critical system that seeks to minimize false-accept errors, the program can be configured to use a more strict recognition threshold. Some appliances fit this model. Likewise, in a system where responsiveness is demanded, an occasional false-accept error may be acceptable as an unanticipated product response can be designed to appear as a “spontaneous outburst”. Some toy characters are better served by this model.

**2. Confidence scoring** – Any time a word is recognized, a confidence score is generated. The program can react differently depending on the confidence level. For example, in a critical system, a medium confidence score can be used to prompt the user to repeat one more time, as in “Did you say X?”

Different FluentChip™ recognition technologies implement false-accept/false-reject errors and confidence scoring in different ways.

### 1. T2SI

During SI recognition, the balance between false-accept and false-reject errors is controlled by the “Out-of-Vocabulary Sensitivity” parameter in the Quick T2SI tool’s “Settings” tab. There are five possible settings: “Reject most utterances”, “Reject more utterances”, “Normal”, “Reject fewer utterances” and “Reject fewest utterances”. The default setting is “normal” which attempts to strike a balance between false-accept and false-reject errors. Moving to “Reject more” or “Reject most” decreases false-accept errors, but may increase false-reject errors. Again, this may be appropriate for some appliance applications. Likewise, moving to “Reject fewer” or “Reject fewest” decreases false-reject errors, but may increase false-accept errors. Toy characters may be best suited for this kind of tuning.

In the FluentChip™ API call for T2SI and \_T2SI(), there is a parameter called *knob* which controls the confidence threshold level. There are five possible settings, numbered 0 to 4. Lower numbers result in more error returns of “OK” (high confidence) results and fewer medium and low confidence error return results. Higher numbers result in fewer high confidence results and more medium and low confidence results. Note the T2SI *knob* parameter is ignored for triggers – it is only valid for command sets.

Sensory recommends using the default setting of “2” for *knob* and **strongly** recommends accepting high and medium confidence error codes as a successful recognition. For applications which cannot tolerate false-accept errors (one which favors “Reject More” or “Reject Most” in Quick T2SI), it may be best to accept only high

confidence error codes. This may be the case with some appliances. For toy characters, where responsiveness is valued, one may favor a Quick T2SI setting of “Normal” or even “Reject Fewer”, and accept medium confidence error codes as success.

## 2. SD

In SD, the recognition sets are created by the user during run-time and there is no separate software tool for creating them. So the *knob* setting is used in the various FluentChip™ API SD calls to balance between false-accept and false-reject errors. There are five possible settings, numbered 1 to 5. The default setting is 3 which attempts to strike a balance between false-accept and false-reject errors. Higher numbers decrease false-accept errors, but may increase false-reject errors. Lower numbers decrease false-reject errors, but may increase false-accept errors.

## E. Managing background noise.

The bane of any speech recognition system is background noise. Noise hurts speech recognition in two ways. First by making it harder to recognize valid words and phrases (false-reject and substitution errors) and second, by creating random data patterns that can be mistaken for words and phrases in the recognition sets (false-accept errors). While it is impossible to eliminate background noise, it is possible to minimize its effects.

**1. Recognize only when necessary** – One way to manage background noise is to limit the time when recognition happens. For example, a program might use speech prompts (also referred to as speech synthesis) like “what is three plus four?” to tell the user when and what to speak, and to signal when the speech recognition chip is expecting a response. Command recognition should use a timeout of 3-5 seconds to limit the window of time in which the recognizer is listening. This improves recognition accuracy AND limits the possibility of noise triggering false-accept errors.

Another approach is to start recognition manually by pushing a button before speaking, or to identify a natural equivalent event in the existing product flow (for example, opening and closing the door on a microwave). This is an alternate way of opening a listening window which lasts a few seconds. Clever use of such events in the product design may avoid the need for triggers and greatly reduce the chance of a false-accept error.

**2. Use single trigger words/phrases** – If the product must continuously listen for commands (for example, a lamp), then using a single trigger word helps to manage the effect of background noise by only listening for a single word instead of several words at once. With a single trigger word, the chance of a false-accept error is much smaller. This trigger then leads to a command set, with a listening window open only for a few seconds to minimize the chance of a false-accept error from noise.

**3. Use Wordspotting (SD only)** – The most noise robust recognition technology Sensory offers is speaker-dependent wordspotting (SDWS). It should be used any time no more than four phrases (for the RSC-4128) or two phrases (for the RSC-464) are needed and custom command training is a feature of the application.

**4. Increase the signal-to-noise ratio (SNR)** – If the product is meant to be used in a noisy environment, care must be taken to manage the noise. For example, consider speech recognition being used in a TV remote control. Either the TV should be muted when the user is expected to talk, or the remote control should be physically close to the user to improve signal-to-noise-ratio (SNR). For an electronic game with loud noises and/or music, the user should be provided with headphones for listening to the sounds. A headset microphone can also be used to provide a good SNR. For an automotive Hands-Free-Kit, muting during recognition or a directional microphone pointed at the driver should always be used.

## F. Designing hardware for proper gain and low electrical noise

Software cannot run without a hardware platform on which to run. A well designed hardware platform should have proper microphone input gain and low levels of electrical noise. Sensory provides a helpful design guide for those seeking to design their own circuits. For more information refer to Sensory Design Note 80-0073 (Speech Recognition Hardware Design).

## The Interactive Speech™ Product Line

The Interactive Speech line of ICs and software was developed to “bring life to products” through advanced speech recognition and audio technologies. It is designed for cost-sensitive consumer-electronic applications such as home electronics, home automation, toys, and personal communication. The product line includes the award-winning RSC-4x general-purpose microcontrollers and tools, the *VR Stamp™* 40 pin DIP module and tools, the SC series of speech and music synthesis microcontrollers. Our suite of software development kits are designed to run on non-Sensory processors and DSP's, and support most popular operating systems.

### **RSC Microcontrollers and Tools**

The RSC product family contains low-cost 8-bit speech-optimized microcontrollers designed for use in consumer electronics. All members of the RSC family are fully integrated and include A/D, pre-amplifier, D/A, ROM, and RAM circuitry. The RSC family can perform a full range of speech/audio functions including speech recognition, speaker verification, speech and music synthesis, and voice recording/playback. The family is supported by a complete suite of evaluation and development toolkits.

### **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 Python IDE and limited-life C compiler, QuickSynthesis™ 4 and Quick T2SI-Lite™ speech tools.

### **SC Microcontrollers and Tools**

The SC-6x product family features the highest quality speech synthesis ICs at the lowest data rate in the industry. The line includes a 12.32 MIPS processor for high-quality, low data-rate speech compression and MIDI music synthesis, with plenty of power left over for other processing and control functions. Members of the SC-6x line can store as much as 37 minutes of speech on-chip and include as many as 64 I/O pins for external interfacing. Integrating this broad range of features into a single chip enables developers to create products with high quality, long duration speech at very competitive price points.

### **FluentSoft™ Technology**

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 up to 5,000 words; runs on non-Sensory processors including Intel XScale, TI OMAP, and ARM9 platforms; and supports operating systems such as MS Windows, Linux, and Symbian.

### **3Dmsg™ Technology**

3Dmsg's ([www.3Dmsg.com](http://www.3Dmsg.com)) Animated Speech technology offers animated avatars with advanced speech recognition and synthesis capabilities for use in smartphones, language trainers, and kiosk applications. Facial expressions can be configured to show emotions and lip synchronization can be automatically driven from voice or text data.

### **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 or 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.



S E N S O R Y®

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

© 2007-2009 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.