SENSORY®

Speech Recognition Software Design

For Sensory IC's

DESIGN NOTE

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.

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.

A. **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 QuickT2SI™ software tool.

   SI sets fall into two general categories: vocabularies and grammars. A vocabulary is a simple list of words and phrases to be recognized. The voice-operated dimmer lamp example above is a vocabulary. A grammar is a more powerful, flexible and potentially more efficient representation than a list vocabulary. The downside is that it's more complicated to specify and requires learning and adhering to a strict syntax. A grammar comprises of a hierarchical set of phrase definitions, as follows:

```
size       = small | medium | large;
topping    = vegetarian | pepperoni | cheese;
pizza      = ($size | $topping) [*sil%%] pizza;
grammar    = *sil%% (*nota | $pizza) *sil%%;
```

   This grammar will allow 6 phrases, all ending in the word 'pizza', and starting with 'small', 'medium', 'large', 'vegetarian', 'pepperoni' or 'cheese'. In this example, '*sil' represents silence and the special %% symbol informs the recognizer not return the word as part of the result. Also the square parentheses '[]' mean that a word is optional. Finally '*nota' (which stands for "none-of-the-above") is used to detect out-of-vocabulary phrases. That is, when *nota is recognized as the best result, we conclude that the utterance was out-of-vocabulary.

   A recent addition to SI recognition is the ability to create a *flexible* grammar. Such a grammar can recognize multiple pieces of information in one phrase, such as "Bake at 350 for 1 hour". The use of such grammars in products substantially reduces the interactions required to complete a recognition task improve and dramatically improves ease of use. QuickT2SI also supports the creation of custom grammars to meet the developer's specification.

B. **SD** – In speaker-dependent (SD) recognition, the user trains the device to recognize his or her voice by speaking each of the recognition words or phrases during a training phase. The product stores a *template* of the spoken words or phrases 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. An example of SD recognition is a list of the user's favorite or most often watched channels in a TV remote control. SD templates 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 templates during power losses. The choice of storage is a tradeoff between cost and the value of permanent storage.

C. **T2SISD** – Some products require a combination of SI and SD recognition, where "out of the box" T2SI vocabularies and grammars are enhanced with customized SD templates. 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 SI recognition that works "out of the box" and can be later enhanced with custom user-trained SD recognition. In contrast, the separate technologies T2SI and SD may exist in the same product, but may not be simultaneously recognized.

D. **SV** – A variant of SD recognition is a biometric password technology called speaker verification (SV) recognition. SV Is used to implement voice password solutions. Like SD, it is speaker-dependent, but unlike SD, 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 vocabularies and grammars, and/or SD/SV sets in a single product, depending on the product specification and available memory. Any recognition set can be used for recognition as needed under the control of the application program.

Improving Speech Recognition Accuracy

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

A. Selecting an appropriate recognition set
B. Using trigger words when appropriate to alert the product
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 set

The maximum number of SI or SD words in any given recognition set, and the size of any flexible grammar, depends on which Sensory IC is chosen. The NLP-5x has the most memory and processing capability, followed by the RSC-4128 and then by the RSC-464. The more memory and processing power, the larger the recognition set that can be stored and processed. 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 minimum number of words and phrases that the user is expected to speak at recognition time. All unnecessary words should be removed whenever possible. Every time a speech recognition product "hears" an utterance, it compares that utterance against those in the active recognition set. The more words a recognition set contains, the more likely the recognizer will make a substitution error (where the user says one word or phrase in the recognition 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 recognition sets of two words each, and not combined into a single four word recognition 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 short and very long words with caution – Very short words (1 syllable) tend to more easily match with background noise or incidental background conversation, more frequently resulting in false-accept (FA) errors, sometimes known as “false fires”. Using words with multiple syllables reduces the chance of false-accepts in designs where the risk of FA errors must be minimized. Most programs can use normal length words with a few short, unique words without problems.

   Very long words or phrases do not tend to have false-accept errors; they tend more to have the opposite problem where the recognizer misses part of the utterance and does not return a match. This results in false-reject (FR) errors. Using shorter words or phrases reduces the chance of false-rejects in designs where the risk of FR errors must be minimized.

   There is an inherent tradeoff between FA and FR errors. It is possible to decrease the chances of one type but usually this increases the chance of the other type.

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 have poor recognition 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 four solutions for triggers in FluentChip™ – the Truly Hands-Free™ trigger, the standard T2SI trigger, the SDWS trigger and the T2SISD trigger. The Truly Hands-Free™ trigger is the most robust in noise.

1) Use triggers 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) Use triggers to increase accuracy – Using a trigger substantially reduces the likelihood of false-accept errors in subsequent recognition sets. False-accept errors occur when background speech or noise unintentionally matches an active recognition set or trigger. ***In products that have recognition sets with multiple words or phrases, the reduction in false-accepts or false fires comes from the focus on recognizing a single utterance. More utterances increase the FA rate over a single utterance on the order of the number of utterances.*** For example, a recognition set with 10 words or phrases will have a random false-accept error rate about 10x that of a set with only 1.

Consider a voice-operated wall clock that announces the time when it hears 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 will occur whenever the chip hears any sequence of sounds that is phonetically similar to the “time” command. If the false-accept error rate is too high, the constant announcement of the time will become annoying.

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 FA 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 FA error rate.

Note that while it is possible to reduce the overall error rate of a speech recognition system to a low level, it is not possible to reduce it to zero. All input systems have an inherent error rate – even manual input devices like pushbuttons and keyboards have a rate of failure (usually operator errors - i.e. pushing the wrong button) 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) Choosing triggers – Even more than for recognition sets, it is important to choose trigger words and phrases carefully. The most common challenge in choosing a robust trigger is choosing a word or phrase that works for the product but is not prone to false-accept or false-reject errors. Triggers which are long (but not too long) and include two or more plosives and fricatives are both recognizable in noise and less prone to false-accept and false-reject errors. An ideal trigger is between four and six syllables long.

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 better trigger candidate. Think about combining the name or brand of the product, the product type, and other phrases such as greetings to make a trigger which works well for your product. These guidelines apply to all trigger types.

Truly Hands-Free™

The Truly Hands-Free™ (THF) trigger is the most robust in noise, providing the lowest possible combination of FA and FR errors. Unlike regular T2SI triggers THF triggers use wordspotting to continuously look though the incoming audio for the trigger and do not require bracketing silence before and after the trigger, Truly Hands-Free™ triggers require custom recordings to tune and test, and require support from Sensory personnel to create. For the most demanding environments and accuracy, such as appliances, this is the best choice. Contact Sensory Sales for more information.

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.

SDWS

The SDWS (speaker-dependent wordspotting) trigger requires training, as do all SD templates. Like THF triggers above, the word-spotting aspect of SDWS allows the recognizer to find the trigger within the audio stream, without being bracketed by relative silence before and after. This makes it robust in noise. SDWS triggers are the best choice if the trigger needs to be customized by the user. The product instructions should stress the attributes of a good trigger (i.e. four syllables, etc)

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. The product instructions should stress the attributes of a good trigger (i.e. four syllables, etc)

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 exists 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 set, 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 *Recording Speech for Synthesis Design Note* (p/n 80-0050).

2) Follow-up Questions for Clarity - The program can also interact with the user through speech synthesis to clarify responses. When recognizing words and phrases (especially a large recognition set) Sensory's T2SI and SD recognition technologies calculate the probability of success. The product can be designed to prompt for additional clarification if the desired confidence hasn't been achieved. For example, if a voice-operated 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

3) Filling missing information in a T2SI flexible grammar – Flexible grammars recognize multiple pieces of information in one phrase, such as "Bake at 350 for 1 hour". This response contains three pieces of information: the cooking method (bake), the cooking temperature (350 degrees) and the cooking time (1 hour). If the user omits one or more pieces of information, the program can re-prompt with "cook at what temperature?" or "cook for how long?" and use recognition to obtain all the needed information to cook the food.

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 T2SI recognition, the balance between false-accept and false-reject errors is controlled by the “Out-of-Vocabulary Sensitivity” parameter in the QuickT2SI 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; In FluentChip™ v3 API the range is 0 to 4 and in FluentChip™ v5 API the range is 1 to 5. 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 mid-range setting for knob (knob=2 in Fluentchip™ v3, and knob=3 in FluentChip™ v5) and strongly recommends accepting high and medium confidence results as successful recognition. For applications which cannot tolerate false-accept errors (one which favors “Reject More” or “Reject Most” in QuickT2SI™), it may be best to accept only high confidence results. This may be the case with some appliances. For toy characters, where responsiveness is valued, one may favor a QuickT2SI™ setting of “Normal” or even “Reject Fewer”, and accept medium confidence results as successful.

2) **SD** – In SD, recognition sets are created by the user at run-time and there is no separate software tool like QuickT2SI™ for creating them. So the knob setting is used in FluentChip™ API SD calls to balance between false-accept and false-reject errors. In both FluentChip v3 and FluentChip v5, the range of knob in SD calls is 1 to 5. The default mid-range setting for knob is always knob=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. Non-trigger recognition sets 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 or phrases – If the product must continuously listen for audio input (for example, a lamp), then using a single trigger word or phrase helps to manage the effect of background noise by only listening for a single word or phrase instead of multiple words or phrases in a recognition set all at once. With a single trigger word or phrase, 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. See section B for trigger options.

3) 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). A headset microphone can also be used to provide a good SNR on some products, for example a bluetooth transceiver.

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 the Sensory *RSC Hardware Design Guide* (p/n 80-0073) or Sensory *NLP-5x Hardware Design Guide* (p/n 80-0332).

The Interactive Speech™ Product Line

Sensory's **Interactive Speech™** product line makes consumer electronics more intelligent by enabling them to talk, hear, move and interact with the external world using naturally sounding spoken commands-all without training and even in noisy environments! Sensory offers both chip and software solutions that offer advanced speech recognition with hands-free functionality, biometric speaker verification, text-to-speech (TTS) synthesis, high quality stereo music and sound effects, robotics and LCD controls, and interactive sensing capabilities. These technologies are designed for integration into cost-sensitive consumer electronic applications such as home appliances, smart toys, music players and personal communication devices. The hardware line includes the NLP-5x Natural Language Processor, the RSC-4x family of mixed signal processors, and the SC-691 music and speech synthesis slave processor. Embedded software options include the FluentSoft™ Recognizer, which offers speech recognition technologies for non-Sensory processors and DSPs. Sensory's BlueGenie™ Voice Interface, the first speech recognition, TTS and synthesis option for *BlueTooth*® enabled devices, offers hands-free control of headsets, music players and other *BlueTooth*® devices.

NLP-5x Natural Language Processor and Development Tools

The NLP-5x features a high-performance 80MHz 16-bit DSP with on-chip ADC, hi-fidelity stereo DAC, microphone preamplifiers, RAM, OTP code and constant memory, and many kinds of peripheral interfaces and control blocks. With Sensory's FluentChip™ 5 firmware, it provides a single chip solution capable of accurate speech recognition; text-to-speech (TTS) synthesis with morphing; compressed speech; high fidelity music; motor and LCD control; and man-machine interfaces (MMI) with interactive sensors. Sensory offers a complete suite of evaluation and development tools that include the ability to create complex grammars with a natural language interface in multiple languages.

RSC-4x Family of Microcontrollers and Developer Tools

The RSC-4x (**Recognition, Synthesis and Control**) 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.

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 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:

Safety Policy:

SENSORY®

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