Technical Article

Smart Sensing with Ultra-low-power MCUs – Part 5: Smart Microphones

Zhaohong Zhang

In the fifth installment of this six-part smart sensing series, I'd like to discuss another new trend in user interfaces – smart microphones, as demonstrated in popular automated home products used by consumers today such as Google Home and Amazon Echo.

The smart microphone integrates a traditional microphone with ultra-low-power digital signal processing technology for extracting command information from a human voice. When the application only needs a small set of commands, the digital signal processing capability within the smart microphone can do all the computing locally without waking up the main system or connecting to a server. This capability is the key to interfacing with the human voice interface in smart appliances/machinery, TVs, robots and other digital assistants for both consumer and industrial applications.

So, how does a smart microphone work? Figure 1 below shows a conceptual block diagram of the technology required.

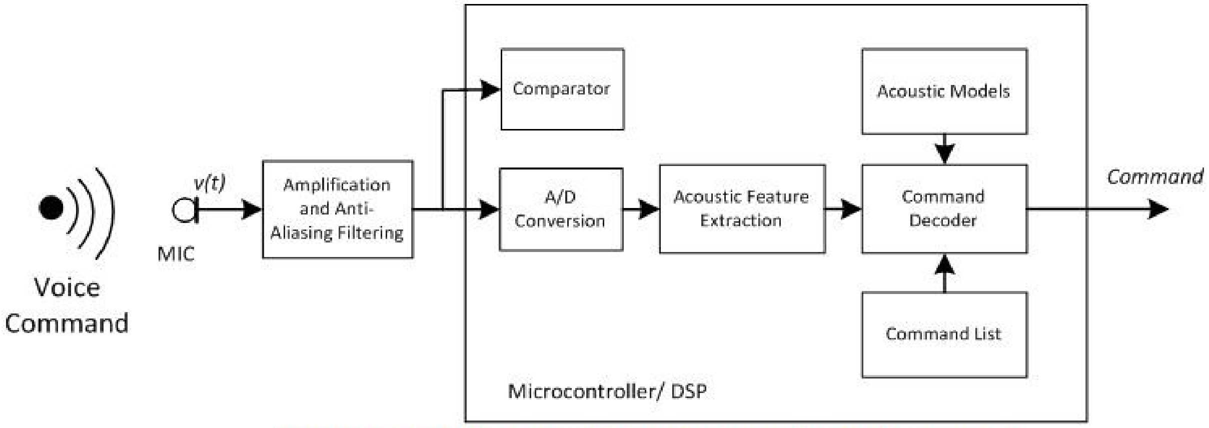

Figure 1: Smart microphone functional block diagram

In Figure 1, a microphone converts a sound wave into an electric signal. The electric signal is amplified and low pass-filtered by the analog amplifier for proper sampling by an analog-to-digital converter. The acoustic feature extraction block pulls the acoustic feature vectors from the microphone data and presents the results to the command decoder block (we'll delve into this further later). In order to recognize short commands, the command decoder block checks the accumulated acoustic feature vector (maybe for 1second) against information from acoustic models and the command list table to see if there is a valid command. If a valid command is found, the smart microphone then sends it to the main system.

Now, to fully understand the acoustic feature extraction, let's examine the processing involved.

First, the acoustic feature extraction removes the linguistic component in the voice and discards unrelated components like background noise. This feature is often the most computationally intensive part of speech recognition. The most accurate and widely accepted feature for voice signature extraction is called mel-frequency cepstral coefficient (MFCC). To compute MFCC from raw voice data you must:

1. Frame the signal into short frames.
2. Calculate the power spectrum of each frame.
3. Apply the mel filter bank to the power spectrum and sum the power in each filter.
4. Take the logarithm of all mel filter-bank powers.
5. Take the discrete cosign transform (DCT) of the log filter bank powers.
6. Typically, keep DCT coefficients 2-13 and discard others.

(For more on MFCC – read this tutorial.)

Voice is a time-variant signal. The sort-time power spectrum computed by the fast Fourier transform (FFT) algorithm is a common technique used to analyze time-variant signals. The voice signal is first organized into short frames so that the voice in the frame can be considered “statistically stationary.” The frame also needs to be large enough to cover the lowest frequency of interest. A typical frame size for voice recognition is 20-40ms.

To properly represent the voice signals, the frames need to overlap, typically by 75 percent. Assuming a 40ms frame, MFCC calculations will update every 10ms based on data collected in the last 10ms, plus data acquired in the previous 30ms. All MFCC processing needs to be completed within 10ms to keep up with incoming data.

Since the human ear is less capable of distinguishing high-frequency components in voices, the original FFT power spectrum is grouped to mel filter banks to match the human ear. The power in each mel filter bank bin is the summation of the FFT bins in the mel filter bank bin. The size of the mel filter bank is small at low frequencies and becomes larger at higher frequencies. In general, 20 to 30 mel filter bank bins are calculated. The power of the mel spectrum is converted to logarithmic values in order to match human hearing sensitivity to sound intensity. Discrete cosine transform (DCT) is then performed to further compress the data.

Let’s take a look at an example of computing the MFCCs. Assume the sample rate of12.8KHz and a frame size of 40ms. The mel algorithm would take an ARM® Cortex-M4F central processing unit (CPU) about 4ms to complete the MFCC algorithm when running at 48MHz. A good choice for this sort of system is the MSP430FR5994 MCU with a low energy accelerator (LEA). A rough estimate shows that the MSP430FR5994 (with the LEA module’s help) can complete the MFCC algorithm in about 4ms – with 16MHz clocking. The MSP430FR5994 also only consumes about 33 percent of the power compared to the ARM Cortex-M4F, which is very important in battery-operated systems.

The LEA module is a hardware accelerator especially designed for vector math operations such as FIR filtering and FFT. LEA supports the following calculations required by the MFCC algorithm:

- **Data filtering and windowing:** You can use the LEA module’s finite impulse response (FIR) function for filtering when needed. A Hamming windowing operation is typically applied to the input data before the FFT algorithm to reduce side lobes in the spectrum. This task can be easily performed using the LEA module’s vector multiply function.
- **FFT and power spectrum**: The LEA module directly supports the FFT algorithm. Its vector multiply and addition functions can make power-spectrum calculations easy and fast.
- **Logarithmic conversion mel spectrum**: The Taylor series is typically used in calculating Logarithmic conversion. It is very time-consuming to perform such an operation on a vector. The LEA module offers a polynomial function to perform this calculation on a vector.
- **DCT**: The LEA module’s FFT function can also calculate the DCT.

After a pre-determined time interval, the command decoder block checks the accumulated MFCC features against information in the database to determine if there is a valid command. If 12 MFCC features are generated every 10ms, for example, 1,200 MFCC features will accumulate over a period of 1second. The user can choose the number of MFCC features to compare. More MFCC features generate a more accurate output at the cost of more computing power.

The LEA module also offers maximum and minimum search functionality. The value in the vectors can be signed or unsigned. Combined with vector addition and subtraction functions, the vector maximum and minimum search function will make database searches more efficient.

How would you integrate a smart microphone into your application design?

Additional Resources

- Download the white paper, “Setting a new standard for MCU performance while minimizing energy consumption.”
- Learn how to use the LEA module in smart fault indicators in the white paper, “Smart fault indicators with ultra-low power microcontrollers.”
- “Mel Frequency Cepstral Coefficient (MFCC) tutorial” talks more about voice feature extraction for speech recognition.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265