Technical Article

A Primer in Voice Interface Technology

Munan Xu

Voice interfaces have emerged as an exciting new spin on how users interact with computers. How do these new systems work? What are the hardware requirements for creating such a device? As voice-controlled interfaces become more and more ubiquitous, I've dug in a little deeper to provide some insight into the technology that makes these devices tick.

What Is a Voice Interface?

Speech recognition has been around in some form since the 1950s, when engineers at Bell Labs built a system to recognize individual digits. Speech recognition is only one aspect of a complete voice interface, however. A voice interface incorporates all aspects of a traditional user interface: it should be able to present information and provide a way for users to control it. In a voice interface, control – and sometimes the information presented – happens through speech. A voice interface may also be an additional option alongside more traditional user interfaces such as buttons or a display.

Your first encounter with a device that had a voice interface was probably a cellphone, or a very basic speech-to-text program on your PC. Those systems were slow, inaccurate and often had a limited vocabulary.

What took voice recognition from afterthought to the darling of the computing world? First, there came significant improvements in both computing power and algorithm performance (bonus points if you know what a hidden Markov model is). Then, the ability to leverage cloud technologies and big data analytics improved speech-recognition algorithm training and increased the speed and accuracy of recognition.

Adding Voice Recognition to Your Device

Several people have asked me for advice about how to add some kind of voice interface to their projects. Texas Instruments actually offers several different products, including the Sitara™ family of ARM® processors and the C5000™ DSP family, that are capable of speech processing. Both families have distinct strengths that make them better suited for particular use cases.

A key factor when choosing between a DSP and an ARM solution is how (or if) the device will leverage a cloud-based speech platform. There are three types of scenarios: offline, where all of the processing occurs locally on the device; online, through cloud-based voice processing in devices like Amazon Alexa, Google Assistant or IBM Watson; or a hybrid of both.

Offline: In-car Voice Control

Although it may seem like everything under the sun needs to connect to the internet these days, there are still applications where internet connectivity just might not make sense, whether from a cost perspective or even the lack of a reliable internet connection. Many infotainment systems in modern cars are a great example of an offline voice-interface system. These systems typically only use a limited set of commands such as “Call [name here],” “play “Holland Road”, by Mumford and Sons,” and “volume up/down.” Although there has been great progress in voice-recognition algorithms for traditional processors, their performance can leave something to be desired. In such cases, a DSP like the C55xx will offer the best performance for your system.

Online: Smart Home Hub

Much of the buzz around voice interfaces centers around connected devices such as Google Home and Amazon Alexa. Amazon in particular has generated a lot of attention, since it allows third parties to integrate into its voice-processing ecosystem with Alexa Voice Services. There are also other cloud services such as Microsoft Azure that provide speech-recognition services and function in a similar fashion. For these devices, it is critical to note that all voice processing happens in the cloud.

Whether or not the easy integration is worth offering up data to one of these voice service providers is up to you to decide. However, with a cloud provider doing the heavy lifting, the device side of things becomes much simpler, and in fact, a minimally featured Alexa enabled device really only needs to be able to play and record audio files, as the speech synthesis aspect of the interface also occurs in the cloud. Since no special signal processing functionality is needed, an ARM processor is sufficient to handle the interface duties. This means that if your device already has an ARM processor, you can probably integrate a cloud-based voice interface.

It’s important to note what services like Alexa are not. Alexa does not implement any sort of device control or cloud integration directly. Much of the “smarts” that drive Alexa are cloud-based functions provided by developers that leverage the speech-processing capabilities of Alexa to drive input into their existing cloud applications. For example, if you tell Alexa to order you a pizza, your favorite pizza shop must have programmed a “skill” for Alexa. A skill is code that defines what should happen when you order a pizza. Alexa invokes the skill every time you ask for a pizza. This skill ties into an online ordering system that places the order for you. Similarly, smart home device makers must implement a skill that defines how Alexa interacts with their device and online services. Amazon provides several skills itself and third-party developers have supplied many more, so even without developing any skills yourself, Alexa devices can be very useful.

Hybrid: Connected Thermostat

Sometimes, it is necessary to ensure some base functionality even without an internet connection. For example, it would be really problematic if your thermostat refused to change temperature if the internet went down. To prepare for this eventuality, a good product designer would design some of the voice processing locally so that there is no functionality gap. To enable this, a system might have a DSP such as the C55xx for local speech processing and an ARM processor to implement the connected interface to the cloud.

But What about Voice Triggering?

You may have noticed that up until this point, I have not mentioned the truly magical aspects of the new generation of voice assistants: the always-listening “trigger word.” How can they follow your voice anywhere across a room, and how do these devices still hear you even when they are playing audio? Unfortunately, there is no magic under the hood – just some very clever software. This software is independent of cloud-based voice interfaces and can be implemented for offline systems as well.

The easiest aspect of this is the "wake word." A wake word is a lightweight local speech-recognition routine continuously sampling the incoming audio signal looking for a single word. Since most voice services will happily accept audio without a wake word in it, the word does not necessarily have to be specific to any particular speech platform. For this type of functionality, since the requirements are fairly low, it is possible to accomplish on an ARM processor using an open-source library such as Sphinx or KITT.AI.

In order to hear you from across the room, voice-recognition devices use a process called beamforming. Essentially, these devices determine where a sound is coming from by comparing the arrival time and phase differences between different microphones. Once it determines the location of the targeted sound, the device uses audio-processing techniques such as spatial filtering to further reduce noise and enhance signal quality. Beamforming depends on microphone geometry, and true 360-degree recognition requires a nonlinear microphone array (often a circle). For wall-mounted devices, even two microphones can enable 180 degrees of spatial discrimination.

The final trick voice assistants employ is automatic echo cancellation (AEC). AEC works somewhat like noise-canceling headphones, but in reverse. The algorithm takes advantage of the fact that the output audio signal such as music is known. While a noise-canceling headphone uses this knowledge to cancel out external noise, AEC cancels out the effect of the output signal on the input microphone signal. Your device can ignore the audio that it puts out itself and still hear the speaker, regardless of what might be playing. AEC is computationally intensive and best implemented in a DSP.

To implement all of the features discussed: wake word recognition, beamforming, and AEC requires an ARM processor and DSP working together: the DSP powers all signal-processing functionality, and the ARM processor controls the device logic and interface. Leveraging a DSP plays to its strengths in performing operations on pipelines of incoming data deterministically, thus minimizing processing delays and enabling a better user experience. The ARM is free to run a high-level operating system such as Linux to control the rest of the device. Such advanced functionality all occurs locally; a cloud service, if used, only receives the end result of this processing as a single audio file.

Conclusion

Voice interfaces seem to have attained significant popularity and will likely be with us in some form for a long time to come. Although several different processing options exist to enable voice-interface technologies, TI has an offering to fit whatever your application may need.

Additional Information:

- Jump-start your audio design with the new Audio Pre-Processing System Reference Design for Voice-Based Applications Using 66AK2G02.
- Download our newest white paper "Voice as the user interface – a new era in speech processing."

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