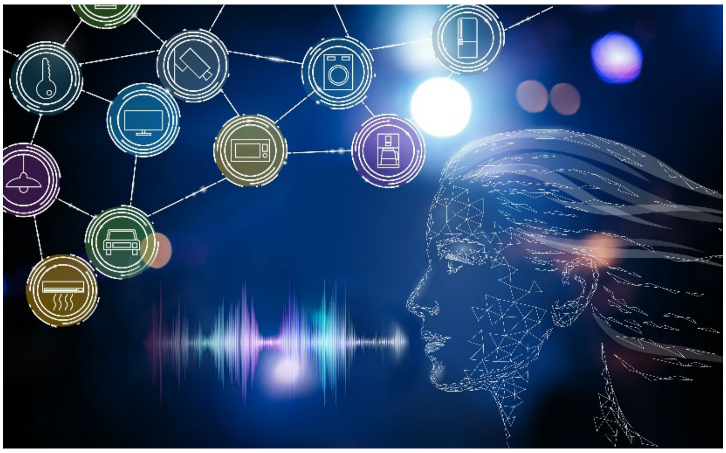
VAD Support on Smart Devices

Evaggelos Katsiantonis, Renesas Electronics

Introduction



Smart devices have become an emerging trend in our days. These can be portable devices, medical, smart home, industry and security related, employed across different domains and able to create a wide range of smart environments. In general, their goal is to support us in carrying out our daily activities, by implementing all those “smart” operations we expect from the environment to make our life easier. What enables that goal to become reality, is the integration of increasingly advancing processing, sensing and communication capabilities, usually combined in small form factors and achieving high power efficiency, allowing for providing full coverage of the environment being monitored and controlled. One of the features though of such system that makes it really smart, is the support of “Voice commands”. This is a functionality that is more and more popular nowadays, essentially enabling a touchless control of the smart devices and consequently of the environment they are deployed in. Talking to your devices becomes finally natural!

To achieve this, embedded devices need to integrate algorithms known as KWS (Keyword Spotting). These algorithms enable the user to interact with Endpoint systems through voice or speech commands. A system running KWS exclusively raises some challenges, mainly in the power management section. It is obvious a wearable, for example, must stay active to “listen” thus affecting the battery autonomy of the device and in general its performance. Here is where a Voice Activity Detection (VAD) technique comes to solve these issues.

VAD description

“Speech detection“ or “Voice Activity Detection”, is a technique that in the microcontroller world is considered as a prerequisite for Voice Command implementation. Why is that exactly? Microcontrollers, that are the heart of a smart device, need to preserve battery life as much as they can, in order to extend continuous device operation till it needs again to be recharged. This can be achieved with the VAD, essentially allowing for the system to enter a sleep mode until enough audible energy has been sensed, triggering an interrupt assertion and consequently waking up the rest of the system in order to execute a power-hungry algorithm (i.e. KWS). This peripheral will let the System on Chip (SoC) stay in sleep mode. Sleep mode, or sometimes referred as “power down mode”, is a condition in which an embedded system enters a low-power mode, thus drastically improving power consumption. Renesas addresses this trend with its DA1470x Family SoC solution which integrates a hardware Voice Activity Detector. An overview of the system is shown in Figure 1.

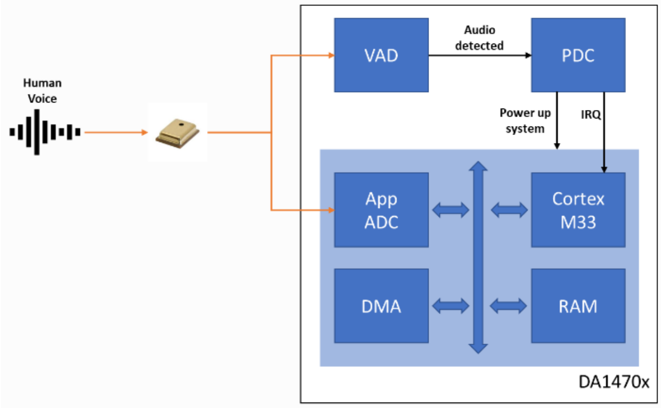


Figure 1: DA1470x System Overview

The DA1470x is a family of multi-core wireless microcontrollers, combining the latest Arm Cortex M33 application processor with floating-point unit, advanced power management functionality, a Graphic Processing Unit (GPU), analog and digital peripherals, a dedicated sensor node controller, and a software configurable protocol engine with a radio that is compliant to the Bluetooth® 5.2 low energy standard.

What are the main blocks involved for KW detection?

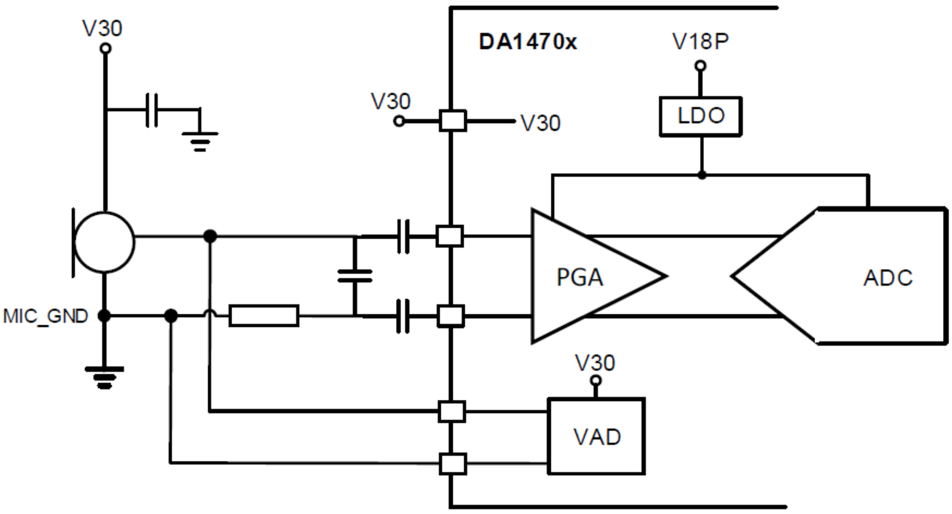


Figure 2: DA1470x, Audio subsystem

The Power Domains Controller (PDC) is a block responsible for acting after waking up or before going to sleep with regard to the power domains of the system. This block allows the system to wait in sleep mode until it is triggered by a GPIO, a timer or other active blocks like VAD. AppADC is a low-power ADC for voice/audio applications. In front of the Application ADC, there is a Programmable Gain Amplifier to adjust the input level of an analog microphone to the ADC input range. The VAD is connected in parallel to the external microphone thus “sensing” audible energy when configured to “Listening” mode. The VAD can trigger the PDC which in turn will wake the system and most important power up the AppADC block. AppADC will begin fetching audio samples and store these into RAM for further processing. Apparently, keeping only a single block powered (VAD) in an embedded system, is an advantage, since power consumption is limited to a single peripheral (<7uA).

How does a VAD operate?

As mentioned, the VAD system relies on the principle of energy-based detection. The VAD computes both the short-term energy (for voice signal) and the long-term energy (for ambient noise) in the voice bandwidth. The detection of a voice event occurs when the relative power between the short-term energy and the long-term energy exceeds a programming threshold (Power Level Sensitivity). These parameters are programmable and are presented in Figure 3.

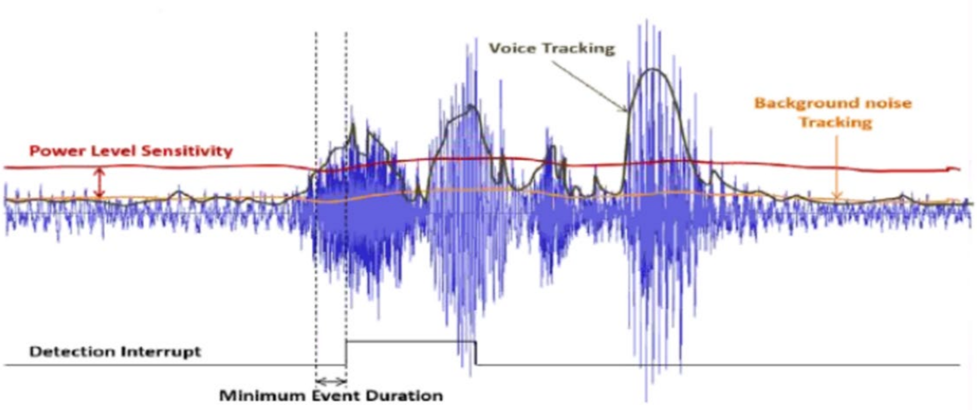


Figure 3: VAD detection related parameters

The parameters mostly responsible for the VAD behavior are:

- VTRACK (Voice Tracking). Adaptation speed of the system depending on the voice input.
- NTRACK (Background Noise). Speed of the system adaptation to the ambient noise.
- PWR (Power Level Sensitivity). Ratio between ambient noise and voice level to be detected.
- MINEVENT (Minimum Event Duration). This parameter allows to set the Minimum vocal signal duration that can be detected by the system.

VAD considerations

Although some default values are available for these parameters it is obvious these will not perform well in either quiet or noisy background environment. It is the responsibility of the product manufacturer to perform measurements in both situations, address these issues and fine tune the system to achieve its full capabilities.

A common challenge that end-product vendors need to tackle in most of the available VAD systems, is “False Alarms”. There are many cases where the system wakes up due to background noise and not actual voice. To minimize this effect and improve VAD performance the MINEVENT parameter needs to be set at high values.

But what about latency? It is obvious that high latency values are responsible for poor “Word spotting” efficiency. Latency during wakeup greatly affects the “Word spotting” algorithm, because it will truncate phonemes of the spoken words that are used for detection, resulting in corresponding increased miss rates. Configuration of different parameters to tackle this challenge has proved to be a must. In DA1470x, all VAD parameters can be configured.

What are the effects on running a KWS locally on an embedded system?

Pros

- No need internet connection in order to run “spotting” algorithm remotely

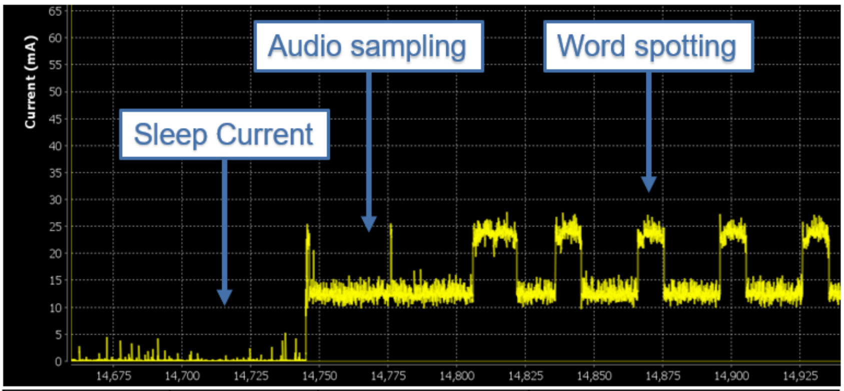
- Detection speed. Runs locally, thus response is better.

Cons

- Limited number of keywords available due to system memory constraints.
- Need to update keywords manually.

Renesas Low power Keyword Detection

With the DA1470x, Renesas provides a demo-solution on “Voice command” functionality combining the VAD and a “Word spotting” algorithm. This demo presents a simple voice command scenario where the user can turn On or Off lights in a room using the DA1470x devkit. The command output is presented on an LCD, displaying the status of a light bulb. On system startup, the DA1470x is set in sleep mode and remains so, until enough energy is detected from the analogue microphone. In this state the microprocessor will use approximately 75uA with the microphone powered on and VAD set in “Listening mode”. During wakeup the power consumption increases, since the AppADC begins to capture audio samples and M33 executes the “Spotting” algorithm. Till that point the advantages of using a VAD technique become apparent. The system does not consume 7.5mA with the processor awake, but only 75uA, thus having a significant positive impact on battery life.



VAD peripheral	MIC powered	M33 awake & sampling audio	M33 awake & sampling audio & running Spotter lib
+5uA	+48uA	+7.5mA	+10mA

Figure 4: DA1470x Power consumption

Furthermore, in this demo the “Spotting” algorithm adds an extra safety feature. A false trigger can be quickly detected, and the system can return in sleep mode to save energy. The DA1470x family has an important feature that can help minimize “False triggers”. The VAD in “listening mode” can output NFI (Noise Floor Information) status. This essentially is the ambient noise reference level that is constantly calculated at a rate that is programmable. NFI can be used to optimize the sensitivity threshold level of the VAD during its usage in real condition.

As mentioned, latency in voice command detection is an important challenge that most voice command solutions are facing. The VAD in this design example adds only 10ms detection latency. An additional latency is introduced from the “Spotting” algorithm during initialization. The DA1470x family can operate at 160MHz, thus keyword processing time is dramatically reduced. Still the total latency introduced is approximately 40ms which inevitably will truncate at least 1 phoneme. To overcome this issue the model that encapsulates the “Command” word needs to support more than one version of the command.

Support of voice commands increasingly becomes a “must” feature of the smart devices nowadays. Renesas offers a means to address this trend with its DA1470x SoC solution.

The design example on the VAD is available on the Renesas products site under the Bluetooth DA1470x family: https://www.renesas.com/eu/en/products/interface-connectivity/wireless-communications/bluetooth-low-energy/da14701-highly-integrated-advanced-bluetooth-52-soc

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

https://www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

https://www.renesas.com/contact-us

Trademarks

Doc Number: R01WP0020EU0100