Transforming Endpoints with Artificial Intelligence

Chris Shore, Automotive & IoT Business

White Paper

There has never been so much compute power available in IoT devices - now is the time to be designing and deploying true intelligence in IoT endpoints. Devices which are aware of their environment are capable of adding huge value to the data they collect and the services which they enable. This paper explores how new and emerging technology can enable your devices to see, hear and feel what is going on, interpret what they sense, take autonomous actions and generate real, trusted, valued data for the cloud. Recent developments in hardware and software will enable a step change in the capability of resource-constrained endpoint platforms, allowing developers to build vision, voice and vibration capability, coupled with genuine intelligence at the endpoint.

I. In the Beginning

Erasmus said, many years ago, "In the land of the blind, the one-eyed man is king." The power of sight is one of the greatest advantages in the world. It is one of the ways in which sighted animals have the ability to sense what is around them. But the creatures which inhabit planet Earth have not always had such powers.

In his book “In the Blink of an Eye”, Andrew Parker puts forward the theory that the explosion in the diversity of life which we call the Cambrian Explosion, was triggered by the emergence of animals possessing the gift of sight. Those creatures possessed an immediate and powerful advantage making them much more effective predators. In a very short period of time (in geological terms), other creatures developed features such as armor and the ability to move and react quickly.

In 2016, at Arm’s TechCon event in Santa Clara CA, Softbank President Masayoshi Son related this to the emergence in the electronic world of systems which can sense their environment and make decisions based on what they see, hear and feel. The development of truly intelligent sensor technology and systems which can interpret their environment in real-time, is driving an explosion in the Internet of Things (IoT).

This leads to the realization that the value of a device is defined first by the type and amount of data which it can collect and, second, by the quality of the analysis which it can carry out on that data.

II. Smart is the Future

A. Not all Sensors are the Same

Smart devices will, undeniably, own the future of the IoT. It is useful to look at some of the ways in which a smart sensor differs from its “dumb” predecessor. Fig. 1 shows two steam gauges – one clearly much, much older than the other.

Fig. 1.
Two steam gauges

We would obviously say that the gauge on the left is “dumb” while the one on the right is “smart”. This distinction manifests itself in many aspects of the way in which the two devices behave and are used. At a basic technological level, one is analog, the other digital; beyond that, one has a fixed function and is non-upgradeable, the other is reprogrammable and can be upgraded and redeployed for different use cases; one can be read openly and presents only raw data, the other presents analyzed data and can do so over secure communication links; and so on.

B. Vision, Voice & Vibration on the Device

The ability to detect light, vibration or sound is useful. But is limited by the level of analysis which can be applied to the data. The addition of "intelligence" i.e. the ability to characterize, analyze and interpret data transforms simple sensors into devices which can hear, see and feel, an analog of the animal senses which we characterize as "voice", "vision" and "vibration".

It has been, and remains to be, relatively easy to implement sophisticated analysis functionality in the cloud. In the data center at the heart of the network, there is easy access to large amounts of processing power. However, increasingly there is a drive to locate as much as possible of the intelligence associated with data analysis at the edge of the network, ideally in the endpoint device itself. There are many advantages, including greater power efficiency, lower latency, lower data bandwidth costs, increased security and privacy. Implementing useful intelligence on small, cost-sensitive and constrained devices represents a much greater challenge but is necessary if the IoT is to scale to the many billions of devices predicted by many analysts.

The process of implementing voice, vision or vibration capability involves several processing stages. Fig. 2 shows how signal conditioning, feature extraction and decision algorithms form a chain of processing capable of extracting decision actions from raw data. The first stage, signal conditioning, is regarded as a traditional Digital Signal

Processor (DSP) problem, the final stage, decision algorithms, as a Machine Learning (ML) problem. The middle stage, feature extraction, lies somewhere between the two.

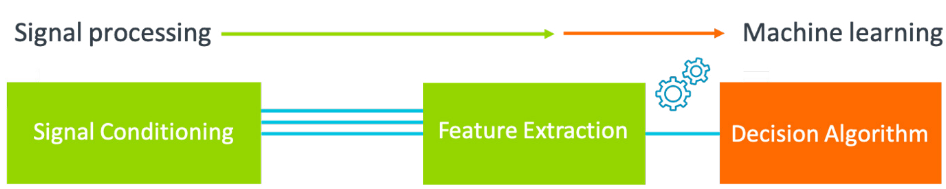


Fig. 2.
Signal Processing & Machine Learning

III. Devices that can Hear, Feel & See

A. Voice

Connecting a microphone to a processing system of some kind is simple. It is easy to create a dumb device capable only of storing and replaying sound. The addition of sufficient processing capability transforms this into a smart device that can do all that but also enhance, characterize, analyze and act on what it hears. As explained above, this often requires different kinds of compute capability.

DSP processing can:

- Filter out background noise
- Detect "interesting" sound in a noisy environment
- Use "beam-forming" techniques to focus on a particular sound from a particular direction
- Identify particular types of sound
- Transform the input signal into a series of extracted "features"

ML processing can then:

- Extract individual words
- Identify the speaker
- Classify the sound (e.g. is it breaking glass? Or an animal noise?)
- Transform sequences of words into meaningful commands or questions

The techniques involved are well understood and the subject of extensive research. When implementing functions like this on a constrained device, it is important to choose the right-sized compute components. Fig. 3 shows how the processing chain can be mapped to processors in the Arm Cortex and Ethos families.

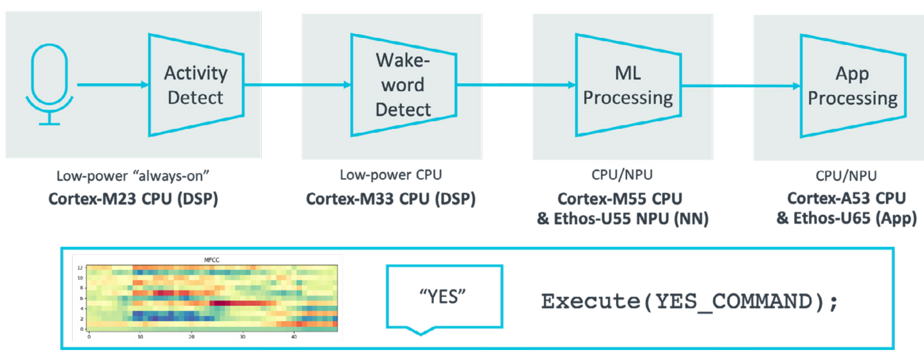


Fig. 3.
A Voice Processing Chain

Mapping the earlier, and simpler, elements in the chain to a highly power-efficient processor like the Cortex-M23 allows for great power saving when the device is essentially idle with no detectable voice input. Progressively more powerful processing elements can then be powered up when required.

The chain above introduces two relatively recent processors from Arm - the Cortex-M55 and Ethos-U55. Cortex-M55 is an evolution of the Cortex-M family which adds the Helium M-profile Vector Extension (MVE) technology for accelerating DSP workloads; Ethos-U55 is a dedicated Neural Processing Unit (NPU) which is optimized for ML workloads. Both individually and in combination, these processors provide a significant performance uplift over earlier options, while retaining the real-time characteristics and ease of programming associated with the Cortex-M family. Fig. 4 and Fig. 5 show the improvements in speed to inference and in energy efficiency.

Fig. 4.
Speed to Inference in Voice Processing

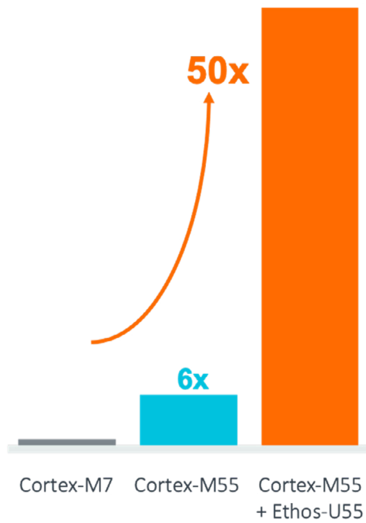
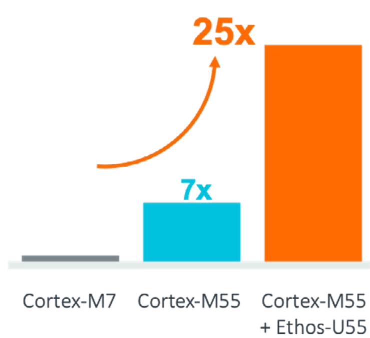


Fig. 5.
Energy Efficiency in Voice Processing



B. Vision

Vision is an increasingly common use case. Typically, it requires significantly greater processing capability than voice. However, research into software optimization, neural network compression and quantization is making it possible to deploy increasingly sophisticated functions on very constrained devices.

A basic camera with a built-in processor will be capable of detecting the presence of an object in the field of view. Such a device would be able to say e.g. “There is something here.” This would be well within the capability of a processor like the Cortex-M23.

Adding more processing power, such as Cortex-M55 and Ethos-U55, would allow the camera to classify detected objects into known categories, effectively enabling it to make statements like “There is a person here.”

We can continue the progression in complexity and compute requirement, enhancing the camera further to be able to identify specific objects. It might then be able to say, for example, “Fred is here.” Processing solutions such as Cortex-A55, coupled with Ethos-N77 would be suitable for this.

C. Vibration

Vibration is a little harder to classify. We use the term to mean the measurement, tracking and analysis of changing signals. This could be physical vibration or movement. Equally, it might be electric current, temperature, magnetic flux and so on. A system can use these signals, either singly or in combination, to build up a "signature" of how a particular system behaves in normal operation. DSP and/or ML techniques can then be used to watch for and act upon anomalies in the data over time. This allows for a number of potential applications, the most obvious of which is predictive maintenance. The ability to detect, in advance of failure, that a mechanical or electrical system is behaving in an abnormal way can give advance warning of potential failure and trigger the need for maintenance of prophylactic replacement in advance of damage being caused.

Typically, DSP algorithms running on a microcontroller such as the Cortex-M23 or Cortex-M33 can be used to perform preliminary analysis of the data in real-time. More complex systems, or those consisting of a multiplicity of sensors, might be sufficiently complex to warrant the increased performance of a processor like the Cortex-M55. The "cleaned up" data can then be further analyzed using ML algorithms to enable decision making.

The "Bob Assistant" shown in Fig. 6 is one example of such a product. This simple device can be attached (retro-fitted) to any piece of moving machinery. When first attached, it enters a learning mode during which it determines what normal behavior feels like. On completion of the learning phase, it enters a long-term monitoring phase, during which it continually looks for anomalies to departures from normal behavior. It can signal such anomalies, using a wireless communications link, to a local gateway.

Fig. 6.
The Bob Assistant

Bob is based around a Cortex-M4 processor using Mbed OS.

Arm has been involved in the development of a similar system for monitoring high-voltage transmission lines, developed for Izoelektro by Slovenian company Irnas. More information can be found at https://www.ram-center.com/.

IV. Sensor Fusion

A single sensor system is useful but can be limited in certain ways. Systems which employ more than one sensor, either more than one of the same type or multiple sensors of different types, can provide better functionality in a number of areas.

- **Elimination of Deprivation**
  A single sensor system is vulnerable to the failure of that sensor. Providing more than one sensor allows for continued operation in the event of failure of one or more sensors.

- **Improved Spatial Coverage**
  A single sensor can only provide data from its "field of view". Adding more sensors can increase the physical area over which data can be collected.

- **Improved Temporal Coverage**
  Many sensors cannot operate with a 100% duty cycle. Even those which can do so will need periodic maintenance during which they will need to be taken off-line. Additional sensors will provide coverage for these inevitable periods when data would otherwise be lost.

- **Improved Precision**
  All sensors have a physical limit on the accuracy of the data which they provide. Adding additional sensors can allow multiple data points to be combined and correlated to increase the accuracy of the resulting measurement.

- **Reduction of Uncertainty**
  All sensors may, from time to time, produce data which seems unusual or anomalous. Very often, this is precisely the data which we are looking for (e.g. in a predictive maintenance application). However, it is always possible that the anomalous data may be a result of failure of the sensor itself. Adding additional sensors provides a cross-check which can reduce the possibility of taking action on erroneous data.

The increased compute capability now available in endpoint devices makes complex sensor fusion solutions more powerful than ever. The possibility of deploying processors such as Cortex-M55 and NPU engines such as Ethos-U55 make the necessary computing power available in even the most power sensitive use cases.

V. Recent Innovations

Arm offers compute technologies coupled with software and tools to help companies streamline the design, development, and support for AI-based IoT applications. Current processing solutions from Arm range from Cortex-M0 to Cortex-M7. These devices have shipped in excess of 70B units since their introduction. The Cortex-M7 is the current leader in raw performance, enabling a diversity of DSP and ML algorithms to be executed, while the Cortex-M0 is the smallest processor in the range.

The recent introduction of Cortex-M55 enables a 5x increase in DSP performance and a 15x increase on typical ML workloads. The parallel launch of the Ethos-U55 microNPU (Neural Processing Unit) enables an additional 32x performance increase for ML workloads The combination of these leads to a theoretical performance increase in excess of 400 times.

For IoT and embedded developers building applications on Cortex-M hardware, they need software to 'just work'. Arm's platform for building and deploying endpoint AI and IoT devices enable software developers to unlock the full potential of Arm-based hardware by supporting any workload, any device, any software stack, and any cloud. Our next-generation development tool, Keil Studio will help the millions of developers get from prototype to production as quickly as possible. Keil Studio is the combination of Mbed Studio with the professional support of Keil tooling that allows IoT and embedded engineers to easily move through development using a single interface, within a single environment, while working with a wide range of microcontrollers and development boards.

Another recent innovation to be aware of is the Arm Custom Instructions technology. This allows, for the first time, chip designers to introduce custom instructions to processors to add unique application-specific features. Currently, this is limited to Cortex-M33 but Arm has committed to supporting Custom Instructions on all future Cortex-M products.

For more information please visit Arm's Endpoint AI for IoT solutions page.

Acknowledgments

The Bob Assistant is developed by éolane, using Mbed OS from Arm and ML software from Cartesiam.