The microcontroller (MCU) has been the backbone of embedded systems for longer than most can remember. MCUs are the preferred processor for many applications across a myriad of markets. Both industrial and consumer products depend on microcontrollers. MCU’s small size, low power consumption, and compact memories are ideally suited for many applications. Integrated peripherals and memory along with low prices are highly-attractive propositions for system developers.

A 2013 report by IC Insights details how 4- and 8-bit MCU deliveries are expected to grow 6%, to 6.7 billion units, shipped in 2013. By 2017 these chips are predicted to take 23% of the MCU market in terms of revenues, and a 28% share of units shipped. While 32-bit MCUs are sharing part of the MCU market, 8-bit devices still retain a significant portion of revenue.

The AVR® architecture is one of the leading 8-bit architectures. It has become synonymous with ease-of-use since its inception in the mid-1990s by Alf-Egil Bogen and Vegard Wollan, two students from the Norwegian University of Science and Technology. The AVR is the most attractive device for students and enthusiasts to learn about computer architectures and embedded programming. The AVR processor core was one of the first MCUs to use on-chip flash for program memory storage. Plus, the AVR architecture allows in-system reprogramming. Over the past 15 years the AVR core’s ease-of-use has made it the preferred processing engine in a wide variety of applications. It’s also used in a multitude of embedded projects at universities around the world. The choice of using AVR in the well-known Arduino open-source development platform provides testament to the success of the architecture.
AVR single-cycle RISC engine

The AVR CPU core is a RISC (reduced instruction set computer) engine. RISC does not mean a reduced amount of instructions, but rather, reduced complexity of instructions. A RISC CPU normally has more instructions than a CISC CPU. This rich instruction set has upsides for a compiler as it can find the optimal single-cycle instruction for its need. The resulting machine code increases speed and reduces memory usage in both SRAM and Flash. The AVR architecture combines this rich instruction set with 32 general-purpose 8-bit working registers directly connected to the ALU (Arithmetic Logic Unit). AVR uses the Harvard architecture, which separates memories and buses for program and data (Figure 1). It has a two-stage single-level pipeline. While one instruction is being executed, the next instruction is pre-fetched from program memory. Two arbitrary registers feed the ALU to perform the requested operation and write back the result. Most instructions are executed within one clock cycle. Due to the this single-cycle execution of arithmetic and logic operations, AVR based MCUs deliver close to 1 MIPS (Million Instructions Per Second) per megahertz of clock frequency.

Fig 1. The AVR’s Harvard architecture separates the program memory from the data memory.

Designed for high-level C code development, the AVR instruction set and CPU design are tuned for minimum code size and maximum execution speed. This combination of single-cycle instruction execution and
high code density delivers computing performance and low power consumption that still leads the industry. Right from the very beginning another aspect has underpinned the success of the AVR architecture. It is the availability of free design and development software, now including tools such as the AVR GCC C/C++ compiler and the Atmel Studio 6 IDE (Integrated Development Environment). This combination of industry-leading 8-bit performance and free development tools has resulted in AVR retaining a significant loyalty among engineers and programmers.

Three Families
Atmel has used the AVR CPU in three high-performance and power efficient 8-bit MCU families: the entry-level tinyAVR®, the mid-range megaAVR®, up to the most recent family, the AVR XMEGA®.

The tinyAVR devices are optimized for applications that require performance, power-efficiency and ease-of-use in a small package. Capable of operating at just 0.7V, the devices integrate an ADC, Flash, EEPROM and a brown-out detector. In addition the chips provide for hardware debug for fast, secure, and cost-effective in-circuit firmware troubleshooting.

The mid-range megaAVR is more suited to applications requiring larger amounts of code. Offering performance up to 20 MIPS, the range offers a wide selection in terms of memories, pin counts and peripherals (Figure 2). There are specialized parts with USB, LCD controllers, CAN, LIN, and power-stage controllers.

Fig 2. The megaAVR CPU and peripheral units provide power and flexibility.
The AVR XMEGA MCUs are composed of various fundamental blocks including the AVR CPU, SRAM, Flash, EEPROM, and a number of peripherals. The AVR XMEGA instruction set also supports 16-bit register access and 32-bit arithmetic. A key feature of the family is the use of power-saving peripherals, achieved via the device’s highly innovative Event System. This is a set of features that allows peripherals to interact without intervention from the CPU. It allows peripherals to send signals directly to other peripherals, ensuring a short and 100% predictable response time. When using the capabilities of the event system, it is possible to configure the chip to do complex operations with very little intervention from the CPU. This saves both valuable program memory and execution time.

All the 8-bit AVR devices use the company’s picoPower® technology. This includes an optimized balance of high-performance and low-leakage transistors, low-voltage operation, various low-power and sleep modes with fast wake up, and the use of hardware DMA. Plus the AVR XMEGA features the Event System that offloads work from the CPU to save power and provide consistent interrupt reaction times. Key performance characteristics include operation over 1.8 to 5.5V, consumption of 200uA per MHz in active mode, 0.1uA in power-down mode with full RAM retention, 0.6uA in power-save mode (with a 32kHz crystal oscillator running), and less than 1us wake-up time.

Applications and the IoT

The 8-bit AVR MCUs have seen a multitude of design wins in virtually all market sectors. In addition to the general-purpose MCUs described previously, there is a wide range of application-specific AVR based devices suitable for lighting, smart battery, industrial and automotive applications among many others. A huge and fast growing application area is wireless. In fact, RF is fast becoming a standard peripheral in MCUs of all sizes and bandwidths. The opportunities in wireless are certainly many – and none more so than in the Internet of things (IoT).

Worldwide wireless infrastructure makers have predicted that by 2020 there will be 50 billion mobile wireless devices connected to the Internet. However, the major wireless players will not entirely dominate this growing arena. Jim Tully, Research Director at market analyst Gartner Group, notes: “Our research says that by 2018, 50% of the Internet of things solutions will be provided by startups which are less than three years old. We can estimate what the internet of things will be like now. But we know that most of the things that will exist in 2018 we can’t even conceive of because they haven’t been invented yet.”

While there is little doubt that low-power 32-bit MCUs will take a share of the market, the opportunity is still highly significant for 8-bit MCUs with low-power demands for the multitude of embedded RF devices that will be required. A primary example of an AVR based device that is well placed to take advantage of these applications is the ATmega256RF (Figure 3). This IC is an IEEE 802.15.4-compliant single-chip wireless solution suitable for ZigBee® RF4CE, IPv6 / 6LoWPAN and ISM (industrial, scientific, medical) wireless applications.
Fig 3. The ATmega256RF combines AVR ease-of-use with a high-performance RF link.

The device combines an AVR MCU and a best-in-class high-sensitivity 2.4GHz RF transceiver, which offers data rates from 250-kbit/s up to 2-Mbit/s. It offers 64K to 256K of Flash and consumes up to 50% less current in some operating modes. It allows for 16MHz operation at only 1.8V supply voltage and gives a fast wake-up time from sleep mode to active modes. In addition to this, it implements various power-down modes, such as a wake-on-radio feature, which keeps the RF transceiver active while the microcontroller sleeps, further enhancing efficiency.

Conclusion

The ease-of-use, high performance, power efficiency and tight code density capabilities of the AVR architecture make it a very serious proposition for design engineers working in a wide range of applications.
and markets. They are ideally suited for wireless and the vast opportunities opening up with the Internet of things. There is a vast breadth and range of 8-bit AVR MCUs available. The last 15 years have certainly been eventful for the AVR, and it the next 15 will be even more so.