# DESIGN& ELEKTRONIK
## KNOW-HOW FÜR ENTWICKLER

**▼TIANMA**
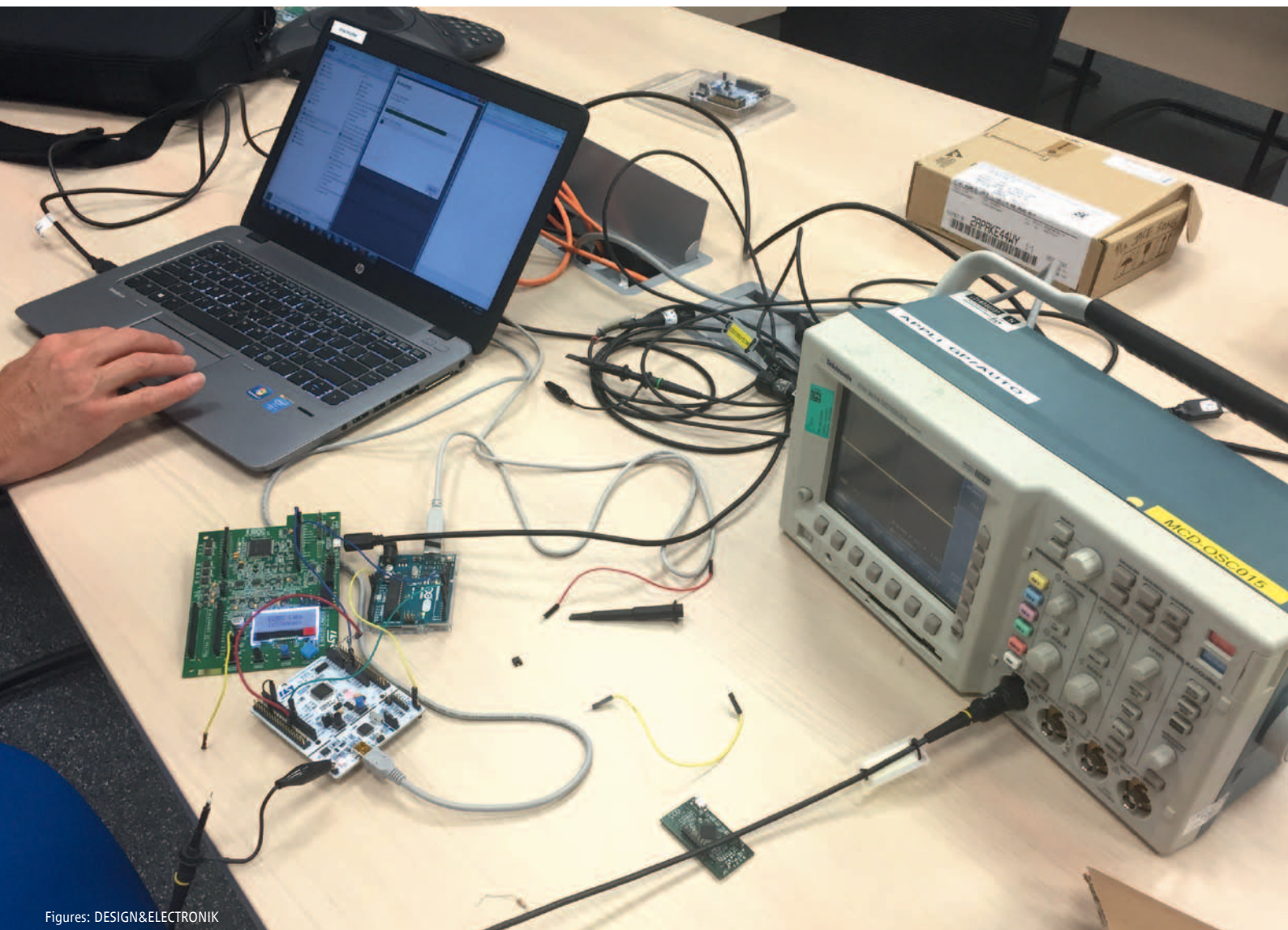
Figures: DESIGN&ELEKTRONIK

# ULPMark goes peripheral

*Measuring the energy efficiency of a microcontroller without its peripheral functions may make sense, or it may not. Nevertheless, ULPBench devised by the EEMBC is not without its worth because you can compare implementation of a memory subsystem and different CPU architectures. But now comes ULPMark, a ULPBench plus that also puts peripheral blocks in the picture.*

.......................................................................

Frank Riemenschneider

Calling it peripheral profile, the EEMBC has presented a new version of ultra-low-power benchmark ULPBench for embedded microcontrollers. Given its new measurement hardware, the benchmark was renamed ULPMark: Now you have ULPMark-CP (core profile, matching the earlier ULPBench), and ULPMark-PP (peripheral profile). The latter not only takes in A/D converters, timers, SPI and realtime clock. Manufacturers also have the possibility to publish figures for supply voltages below 3.3 V. What started this off was our exclusive report [1] in which our own measurements went down to 1.8 V.

In July already, DESIGN&ELEKTRONIK had an exclusive chance to carry out measurements with the new hardware and software. What is evident is that the hardware (named EnergyMonitor) originally developed by Texas Instruments for ULPBench is not suitable for these measurements. Which is why the EEBMC brought in STMicroelectronics to develop new measurement hardware together with the software.

The socalled Power Shield (Figure 1 bottom, above the earlier EnergyMonitor from TI) is based on an 80 MHz clocked STM32L496VGT6 microcontroller and, theoretically, with an LC display and appropriate configuration buttons can be operated without a PC, unlike the EnergyMonitor.

In this case settings and results are shown on the display (Figure 2). Of course it is more convenient to tie up with a Windows PC over USB, and to install the new STM32 Cube Monitor-Power software (developed by ST), which gives you a graphic display of power consumption over time.

Measurement range in active mode of the MCU is between 100 nA and 50 mA, in power saving mode between 1 nA and 200 mA. Maximum deviation of measured results is supposed to be 2%, which was at least confirmed by our own comparison with the EnergyMonitor.

Taking the three 12-bit A/D converters and a maximum sampling rate of 5 MS/s, a dynamic capture rate of 761 kS/s is achieved.

On the GUI it is very easy to evaluate power consumption for selected time spans, such as for TI's MSP430FR5969 and a total runtime of ULPBench of 10 s that shows 80.317 µJ or 8.0317 µJ/s, producing a ULPMark score of 124.5. As Figure 3 shows, you can also determine that for one ULPBench cycle the MCU takes roughly 2 ms in active mode at 8 MHz clock frequency to work its load, consuming an average 895 µA, meaning a total of 6.475 µJ. In power saving mode, taking about 998 ms, an average 526 nA is needed, and thus 1.575 µJ consumed. So you can say that 80.5% of power is required in active mode, and 19.5% in power saving mode. This is interesting in that this ratio for a 32-bit ARM CPU in the Cortex-M0 is about 60:40%, and in Cortex-M4 even about 40:60%. That makes the IPC of the MSP430 16-bit CPU relatively low.

## ■ The peripheral profile of ULPMark

The problem when comparing the power consumption of peripheral blocks is firstly that different MCUs hold different peripherals, and secondly, even for what are basically comparable functions, that these are implemented differently. So the ULP working group of the EEMBC, in a first step, agreed to a minimum compromise, i.e. to look at blocks found on just about every MCU: an A/D converter, a timer to produce pulse width modulation (PWM), an SPI interface, and a realtime clock (RTC). The benchmark itself consists of ten steps with 1 s spacing:

■ Step 1: x64 bytes data capture by A/D converter with sampling rate of 1 kHz + 20 PWM pulses of 32 kHz and fixed duty factor + RTC active.
■ Step 2: x64 bytes data capture by A/D converter with sampling rate of 1 kHz + 40 PWM pulses of 32 kHz and increasing duty factor + RTC active.
■ Step 3: 1 byte data capture by A/D converter + 40 PWM pulses of 32 kHz and fixed duty factor + RTC active.
■ Step 4: 1 byte data capture by A/D converter + x64 bytes send/receive over SPI interface + 100 PWM pulses of 32 kHz and fixed duty factor + RTC active.
■ Step 5: 1 byte data capture by A/D converter + x64 bytes send/receive over SPI interface and check of data from step 4 + 100 PWM pulses of 32 kHz and fixed duty factor + RTC active.
■ Steps 6-8: like step 5.

■ Step 9: 1 byte data capture by A/D converter + x64 bytes send/receive over SPI interface and check of data from step 8 + 30 PWM pulses of 1 MHz and increasing duty factor + RTC active.
■ Step 10: Check of A/D converter data from steps 3-9 + check of SPI data from step 9 + RTC (check and stop).

DESIGN&ELEKTRONIK was able exclusively, before publication of this new benchmark, to conduct measurements and compare: a number of STM32 derivates, an MSP430FR, and the subthreshold Apollo MCU from Ambiq Micro. Power consumption of the STM32L4 plotted over time is shown in Figure 4. In the core profile, where the CPU and memory system are tested, the MSP430FR could not catch up on the STM32
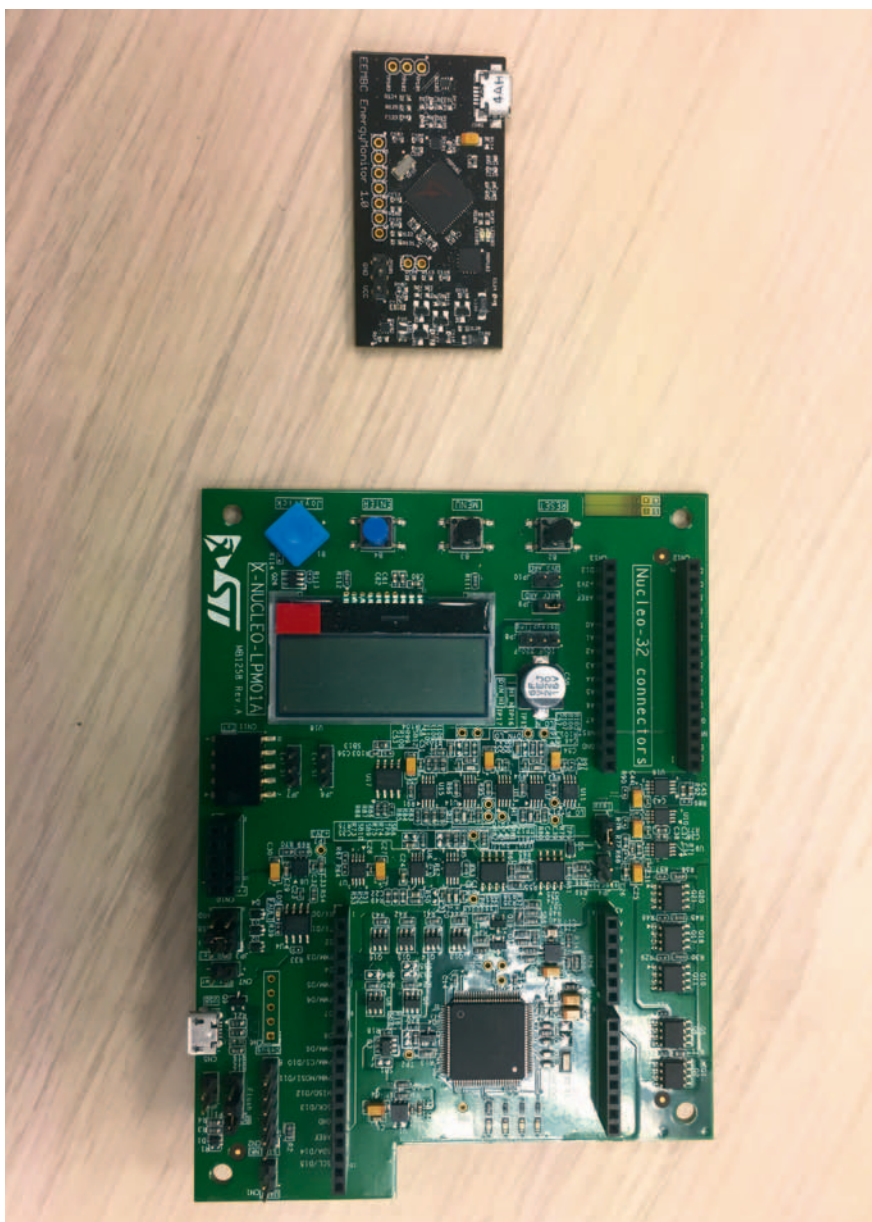


**Figure 1. The Power Shield (bottom) developed by STMicroelectronics is the new measurement hardware for the EEMBC benchmark ULPBench. It can work independently of a PC and costs 70 dollars.**
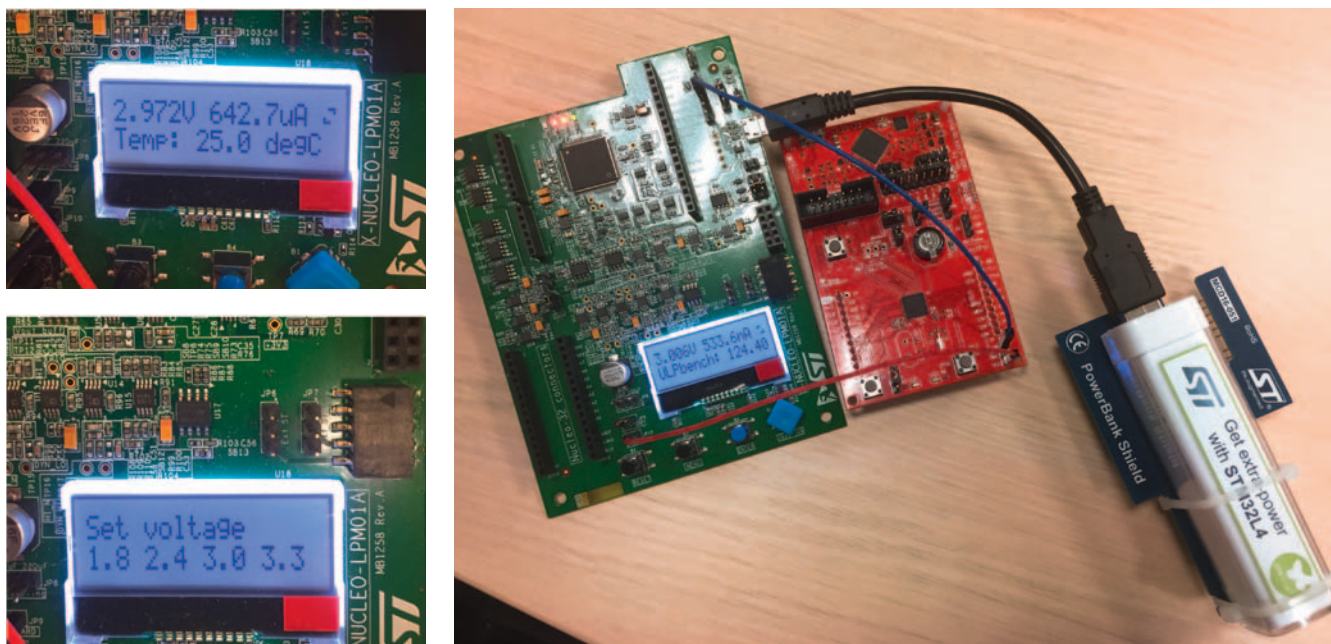
**Figure 2. Measurement data can be read on an LC display (top left), settings made like supply voltage (bottom left), and benchmark results shown (right).**

[1]. But the peripheral profile, that was quite clear even before the measurements, is virtually tailor-made for the MSP430. And why? Because it is the only ultra-low-power MCU with an A/D converter that can be operated at the defined sampling rate of 32 kHz in stop mode, what TI calls LPM3, without at all involving the CPU.

As a result, in steps 1-8 and 10 power consumption is much below that of a

Cortex-M4 CPU like the STM32L4 (Figure 5), while in step 9, where the sampling rate was increased to 1 MHz, this advantage no longer has any effect.

In the overall result — measured at 3.0 V supply voltage — the STM32L433 (256 kB flash memory) scoring 65 can just overtake the MSP430FR with 63.7, while the STM32L0 with a Cortex-M0 CPU and its power-guzzling A/D converter only mana-

ges 57.5 points.

The STM32L452 with 512-kB memory scores 61.7. These are not official scores from the manufacturers by the way, and they might manage somewhat higher figures by one or the other optimization of code.

## ◼ *Ambiq Micro flops*

The same applies of course to the Apollo MCU, which — and this is not a typing error — comes to a catastrophic score of 32 on 3.0 V supply voltage. Naturally you ask yourself how this could happen seeing as the subthreshold MCU put up such an excellent showing on the core profile. There are two main reasons: Firstly, Ambiq Micro has not implemented DMA transfer. For the A/D converter you only find an 8-byte FIFO buffer.

This means that after eight data captures the CPUs are woken up, and must transfer data into SRAM. In the STM32 this procedure is found in the operating modes LPSleep (A/D converter operation) and LPRUN (evaluate data), so that in step 1 of the benchmark already 16.9 µJ is consumed (STM32L433) but 28.6 µJ (Apollo). In step 9, where the PWM duty factor is increased, the STM32L4 can remain in LPRUN mode, while the Apollo again has to be woken up into active mode every time. Eventually there is 58.5 µJ (STM32L433) compared to 210 µJ (Apollo). The second reason is an abysmal SPI-IP. This consumes on average two to four times as much power as
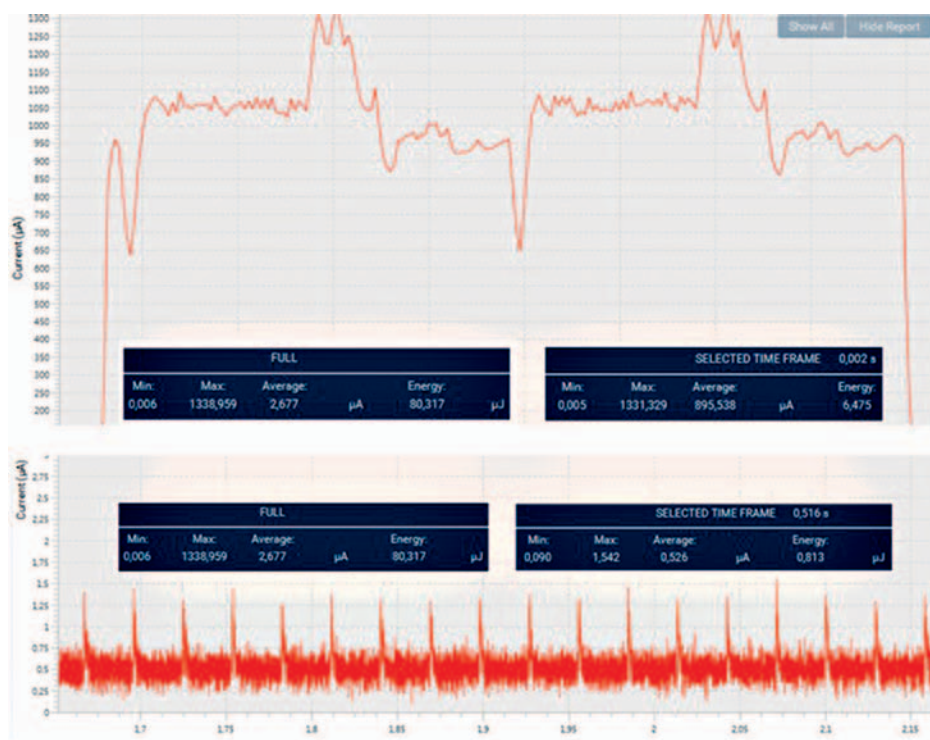


**Figure 3. Plot of power consumption in active mode (top) and power saving mode (bottom).**
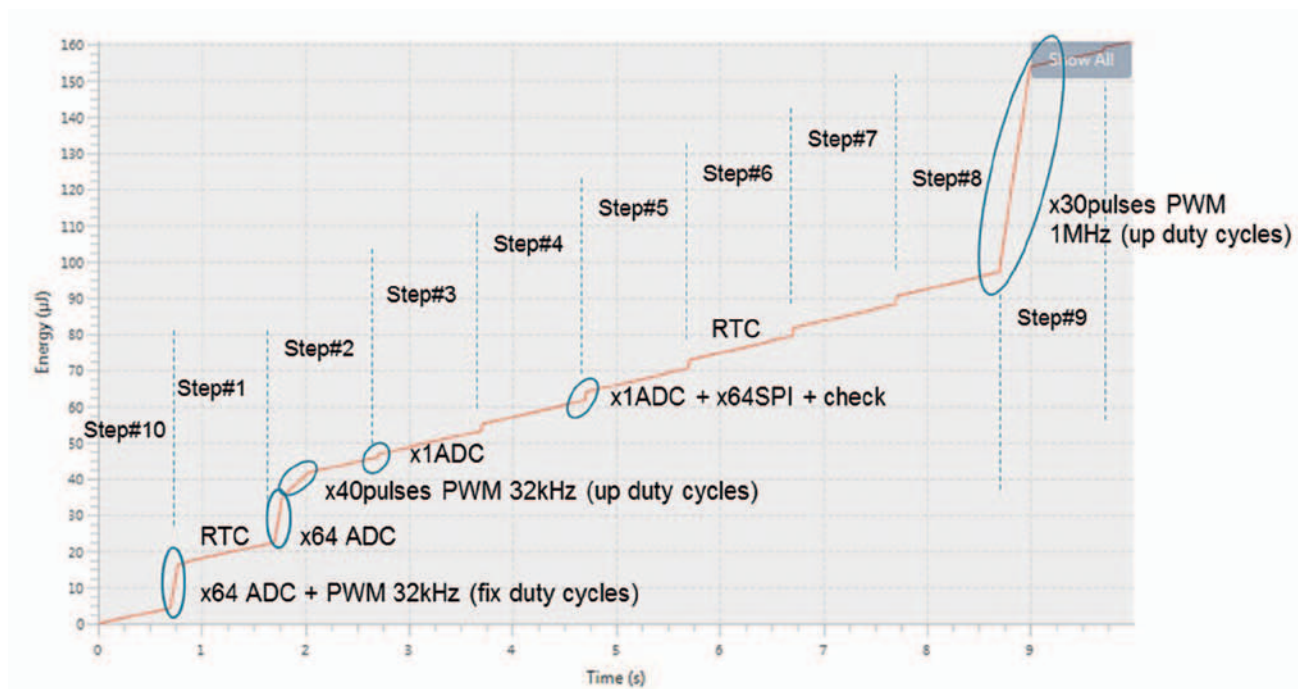
Figure 4. Pattern of power consumption in peripheral profile over ten test steps for STM32L4.

the STM32L433. PWM too is more power-devouring than in the ST. Extremely energy-efficient on the other hand is Ambiq's real-time clock, which registered the high score for the MCU in the core profile.

Given the fact that the test scenario in the EEMBC working group was proposed by its leader, a TI person, you naturally have to surmise how realistic it is, and whether the sampling rates of 32 kHz are not set too low. When you look at the typical target applications of an ultra-low-power MCU, where it is usually a matter of capturing and further processing sensor data, we cannot recognize this, very many are found in ranges of 10 Hz to maximally 4 kHz, as in current meters. Furthermore, comrades-in-arms in the working group, like NXP, Silicon Labs, STMicroelectronics, Renesas and

ARM, would have had an opportunity to protest and put forward counter-proposals.

What we also found interesting in our examinations was the question of why TI in its MSP432 has not implemented the low-energy accelerator (LEA) from the MSP430FR. Even if this functionality is partly created in the MSP432 by the integrated Cortex FPU, the LEA, which works without involving the CPU and signals completion of calculations by an interrupt, has the edge in terms of power consumption for numerous functions.

What spoke against integration the way TI saw it was not so much the technology as the little acceptance of a proprietary solution on a standard MCU with ARM CPU.

## Summary

In the peripheral profile the EEMBC has taken the correct approach to producing a more realistic measurement scenario for ultra-low-power MCUs. Having said that, the integration of a UART would have been interesting for example, plus the question of how MCUs respond if more than the few variables of the ULPMark have to be kept in SRAM in power saving mode. Read about measurements looking into this in an upcoming issue of DESIGN&ELEKTRONIK. (fr)

**REFERENCE**

[1] Riemenschneider, F.: Benchmark — ULPBench debunked. Part 1. DESIGN&ELEKTRONIK 2016, no. 8, p. 9 et seq.
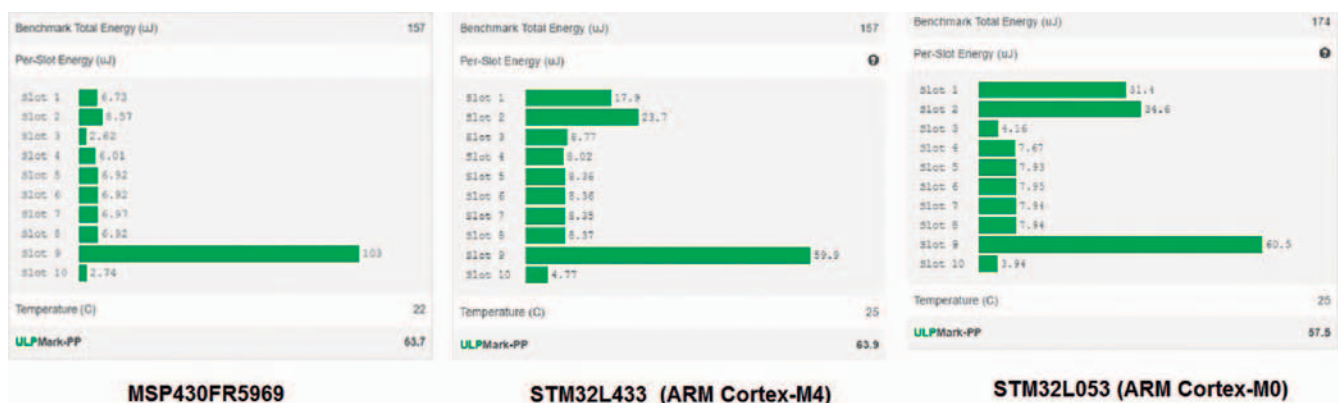


Figure 5. Comparison of measurement results of the peripheral profile for STM32L0, STM32L4 and MSP430FR. Only the TI MCU can execute A/D conversion at a 32 kHz capture rate in stop mode, for which reason power consumption in the particular steps is substantially below that of the ARM controller. If the relatively weak 16-bit CPU has to be activated (step 9) because of the high capture rate, the result is immediately much poorer than for the faster 32-bit Cortex CPUs.