Towards non-CPU activity in low-power MCU-based measurement systems

Ferran Reverter

***Abstract*—This paper evaluates the benefits of having peripheral-triggered peripherals in a microcontroller unit (MCU) intended for low-power sensor applications. In such an architecture, the functionality is moved from the central processing unit (CPU) to the peripherals so that a peripheral is able to trigger another peripheral with non-CPU intervention. For the sensor data logging application under study, both energy consumption and measuring time are reduced by a factor of two with respect to the case of applying an interrupt-based approach that requires the CPU intervention.**

***Index Terms*— Embedded system, energy consumption, microcontroller, sensor interface electronics.**

I. Introduction

A MCU is a low-cost programmable processor-based digital integrated circuit widely used in electronic measurement systems. It has three main blocks embedded: 1) a central processing unit (CPU), which executes instructions sequentially; 2) a memory, which saves the instructions to be executed and the data to be processed; and 3) peripherals, which carry out actions (e.g. timing) in parallel with the CPU activity. Peripherals can be digital (e.g. a digital timer/counter [1],[2]), analog (e.g. an analog comparator [3]) or mixed (e.g. an analog-to-digital converter [4],[5], ADC). In sensor applications [6],[7], the MCU behaves as the mastermind in charge of scheduling and executing different types of tasks: data acquisition, storing to internal or external memory, data processing, communication to other devices, and displaying.

MCUs are devices initially conceived with a clear centralized architecture where the CPU controls any activity inside the chip. Although peripherals can perform some actions simultaneously with the instruction execution done by the CPU, these are completely controlled by the CPU. In other words: the CPU decides when the peripheral starts its action and, once it is finished, the CPU is informed through an interruption [8],[9]; event detection through polling [10] is also possible but not suggested for low-power designs since the CPU is continuously checking the state of the peripherals. Fig. 1 shows in blue solid line an example of how the CPU interacts via interruptions with three peripherals whose actions need to be done sequentially. When the action of peripheral 1 is finished, an interruption is generated and the CPU executes the corresponding interrupt services routine (ISR) which involves, among others, the activation of peripheral 2. Next, this peripheral runs and, once it is done, it interrupts again the CPU which executes another ISR enabling peripheral 3. When the peripherals are running, the CPU can either execute some instructions for processing purposes or be inactive to save power.

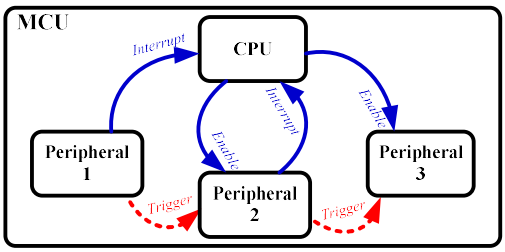

Fig. 1. Peripherals interaction in a centralized (blue solid line) and decentralized (red dashed line) architecture.

More power and functionality have been moved from the CPU to the peripherals in the last generation of MCUs, thus achieving a more decentralized architecture. In this, a peripheral is able to trigger another peripheral with non-CPU intervention. The CPU is employed at the beginning to configure all the peripherals, but afterwards these are able to directly interact each other to carry out the measurement. This is shown, for instance, in Fig. 1 in red dashed line, where the three peripherals are sequentially triggered each other. Once peripheral 1 has finished, it directly issues a trigger signal to activate peripheral 2. Afterwards, when peripheral 2 is done, another trigger signal is issued to directly activate peripheral 3. This peripheral-triggered peripheral (PTP) approach avoids the processing time and the energy consumption of the CPU when executing the ISRs. Therefore, it should reduce the energy consumption of the MCU, but this reduction has not been quantified so far in the literature.

II. Case under study

The case under study is a low-power sensor data logging where the MCU periodically reads an analog input channel with information about a measurand and, then, saves the result to the memory. Three peripherals are involved. 1) A digital timer, with an overflow signal indicating the end of the counting, that defines the periodicity of the measurement (for instance, each second). 2) An ADC, with an end-of-conversion (EoC) signal, that carries out the sampling and conversion of the analog input voltage (V_{in}). 3) A direct memory access

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. This work was supported by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund under project TEC2016-76991-P.

Ferran Reverter is with the Department of Electronic Engineering, Universitat Politècnica de Catalunya – BarcelonaTech, Castelldefels (Barcelona), 08860, Spain (e-mail: ferran.reverter@upc.edu).

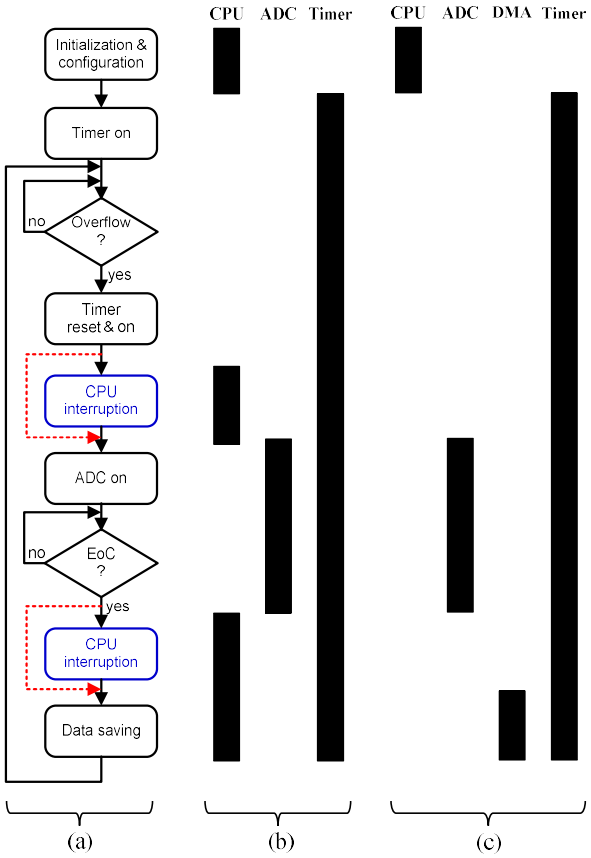

Fig. 2. (a) Flowchart of the measurement process; differences between ISR and PTP techniques are highlighted in blue solid line and red dashed line, respectively. Time intervals in which the embedded resources are activated when applying the (b) ISR, and (c) PTP techniques.

(DMA) controller that moves data from the ADC to memory without CPU intervention; the use of the DMA only makes sense when applying the PTP approach.

Fig. 2(a) shows a flowchart of the measurement process for the case under study. At the beginning, the CPU carries out an initialization and configuration of the peripherals. Afterwards, the timer starts counting. When there is an overflow, the timer is automatically reset and starts counting again. The overflow also generates either a CPU interruption that subsequently activates the ADC, or a direct trigger of the ADC when applying the PTP approach; this is highlighted in blue solid line and red dashed line in Fig. 2(a), respectively. Once the conversion is done, again two options are possible: either an interruption to the CPU that involves saving the conversion result to memory, or a trigger of the DMA that directly moves the data to memory. After a certain time interval, the digital timer overflows again and the previous steps are repeated.

Figs. 2(b) and (c) show how the different resources embedded into the MCU are activated when applying the ISR and PTP techniques, respectively. In Fig. 2(b), the CPU becomes active to attend the ISRs related to both timer and ADC. However, in Fig. 2(c), the CPU continuously remains inactive after the initial configuration. A timer-triggered ADC together with a ADC-triggered DMA enable the sensor data logging with a 0% CPU-load.

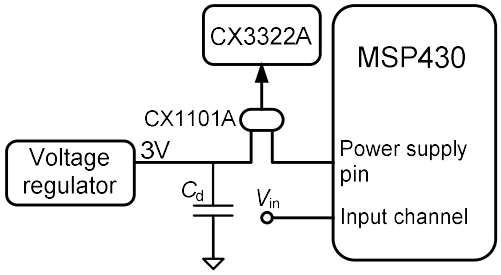

Fig. 3. Measurement setup employed to evaluate a MSP430 operating in both approaches shown in Fig. 2; C_d is a decoupling capacitor.

III. Results

The case explained before has been experimentally evaluated using a commercial low-power 16-bit MCU (MSP430FR5969 from Texas Instruments) powered at 3 V and operating at 1 MHz. This MCU enables us to test both approaches shown in Fig. 2. In addition, this MCU has several low-power modes (LPM) that automatically adapt to the needs, for instance: 1) LPM1 in which the CPU is disabled but the peripherals operate at high frequency, and 2) LPM3, the same as in LPM1 but the peripherals run at low frequency. An embedded 16-bit TA0 timer running at 32 kHz in LPM3 was set to periodically activate the measurement. An embedded 12-bit ADC running at 1 MHz in LPM1 was employed to digitize the signal connected to A2 input channel set in single-ended mode. During the tests, the ADC was set to do conversions with a 10-bit resolution and the input signal was connected to ground. In order to quantify the energy consumption required for both approaches represented in Fig. 2, the current consumption was monitored by a current sensor (CX1101A) connected to a current waveform analyzer (CX3322A from Keysight Technologies), as shown in Fig. 3.

Fig. 4(a) shows the profile of the current consumption and the charge (obtained by integrating the former) when the MCU applied the ISR-based approach represented in Fig. 2(b) after the initial configuration. Six stages can be distinguished. 1) The timer is operating in LPM3 with a very small current consumption (less than 1 μA). 2) Due to the overflow of the timer, the MCU wakes up from LPM3 to active mode, thus requiring seven clock cycles and 12 nC. 3) The CPU executes the timer-related ISR, which involves the activation of the ADC. 4) The ADC carries out the sampling and conversion of the input signal operating in LPM1. 5) The EoC wakes up the MCU from LPM1 to active mode, needing five clock cycles and 10 nC. 6) The CPU executes the ADC-related ISR, which saves the result of the conversion to memory. Note that the end of stage 3 overlaps with the beginning of stage 4 since the ADC is activated inside the ISR and, therefore, the CPU still needs some time to return from the ISR. Fig. 4(a) also shows a zoom of the current profile during stage 4. Here, two sub-phases can be seen [11]: 4a) synchronization and sampling, and 4b) conversion, that require nine and twelve clock cycles, respectively. Overall, the process in Fig. 4(a) needs a charge of 31 nC, which corresponds to an energy of 93 nJ, and a time interval of 78 μs. The wake-up stages (2 and 5 in Fig. 4(a)) cause a current peak of almost 15 mA and require 71% of the total charge.

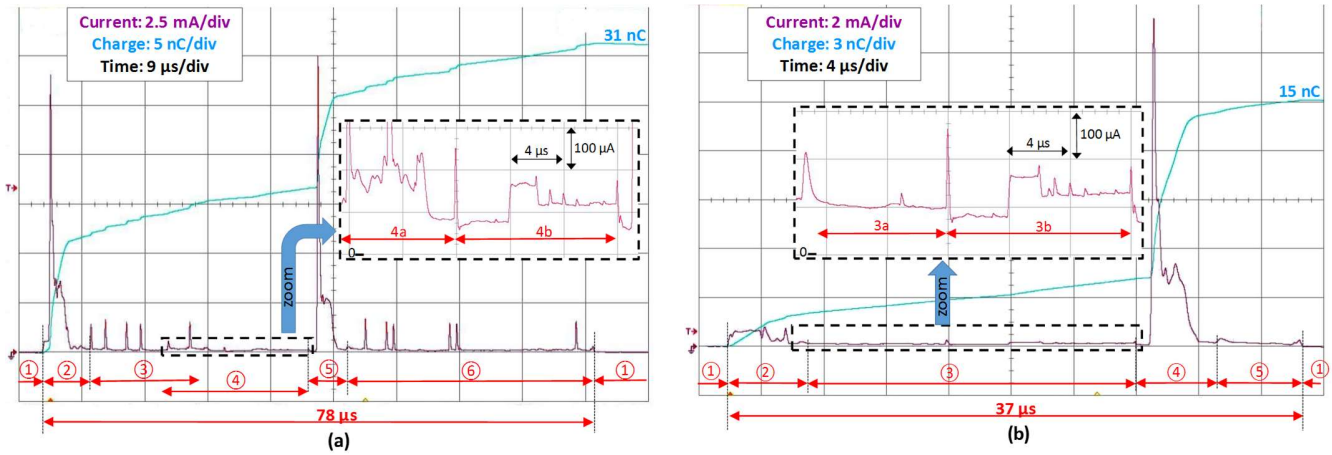

Fig. 4. Profile of the current consumption and the resulting charge –obtained by integrating the former– when the MCU applied (a) the ISR-based approach represented in Fig. 2(b), and (b) the PTP approach represented in Fig. 2(c).

The profile of the current consumption and the resulting charge when the MCU applied the PTP approach in Fig. 2(c) is shown in Fig. 4(b). Now, five stages can be observed. 1) The timer runs operating in LPM3. 2) The overflow of the timer wakes up the MCU from LPM3 to LPM1, necessitating five clock cycles and 2 nC, but the CPU is inactive. 3) The ADC is triggered and does the sampling and conversion of the input signal operating in LPM1. 4) The EoC wakes up the MCU from LPM1 to active mode, needing five clock cycles and 10 nC. 5) The DMA controller is triggered and saves the result of the conversion to memory in five clock cycles; the DMA transfers the data in active mode, but the CPU remains inactive. Fig. 4(b) also shows a zoom of the current profile during the sampling and conversion. Comparing the zoom of both figures, one realizes that the sub-phase a) is quite different, but b) is very similar. The sub-phase 3a in Fig. 4(b) shows a current profile cleaner than that in Fig. 4(a) since this is not affected by any ISR return executed by the CPU. Removing the CPU activity during the sampling of the input signal also provides benefits in terms of variability of the conversion result, as also suggested in [7], [12]. In summary, the process in Fig. 4(b) needs a charge/energy of 15 nC/45 nJ, and a time interval of 37 µs, which are less than half of the numbers indicated in Fig. 4(a). Here, the wake-up charge in stages 2 and 4 represents 80% of the total required charge.

IV. Conclusions

The use of smart peripherals performing operations directly between them without CPU intervention has clear benefits in terms of energy consumption and measuring time. This has been proven herein in a data logging application where three embedded peripherals (i.e. timer, ADC, and DMA controller) periodically record the analog input data with a 0% CPU-load. In comparison with the classical ISR-based approach, this PTP configuration requires, for the case under study, less than half of the energy and time, which is very attractive for autonomous sensors applied to smart cities, wearable devices, internet of things, among others.

Although MCU manufacturers are making a big effort to move power and functionality from the CPU to the peripherals, some of them still cannot be directly triggered, but they need the intervention of the CPU. For example: an embedded timer cannot directly trigger another timer in the commercial MCU under test. Moreover, the wake-up energy has a predominant contribution. For this reason, the energy optimization in the next generation of MCUs should focus more on the wake-up rather than on the peripherals.

References

[1] A. U. Khan, T. Islam, and J. Akhtar, "An oscillator-based active bridge circuit for interfacing capacitive sensors with microcontroller compatibility," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 11, pp. 2560–2568, Nov. 2016.

[2] E. Sifuentes, R. Gonzalez-Landaeta, J. Cota-Ruiz, and F. Reverter, "Seat occupancy detection based on a low-power microcontroller and a single FSR," *Sensors*, vol. 19, no. 3, 699 (11pp), Feb. 2019.

[3] A. Asif, A. Ali, and M. Z. U. Abdin, "Resolution enhancement in directly interfaced system for inductive sensors," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 10, pp. 4104–4111, Oct. 2019.

[4] A. U. Khan, T. Islam, B. George, and M. Rehman, "An efficient interface circuit for lossy capacitive sensors," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 3, pp. 829–836, Mar. 2019.

[5] S. Li, A. Pandharipande, and F. M. J. Willems, "Daylight sensing LED lighting system," *IEEE Sens. J.*, vol. 16, no. 9, pp. 3216–3223, May 2016.

[6] L. Dutta, A. Hazarika, and M. Bhuyan, "Nonlinearity compensation of DIC-based multi-sensor measurement," *Measurement*, vol. 126, pp. 13–21, Oct. 2018.

[7] H. Ganesan, B. George, S. Aniruddhan, and S. Haneefa, "A dual slope LVDT-to-digital converter," *IEEE Sens. J.*, vol. 19, no. 3, pp. 868–876, Feb. 2019.

[8] P. R. Nagarajan, B. George, and V. J. Kumar, "Improved single-element resistive sensor-to-microcontroller interface," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 10, pp. 2736-2744, Oct. 2017.

[9] Z. Czaja, "Time-domain measurement methods for R, L and C sensors based on a versatile direct sensor-to-microcontroller interface circuit," *Sens. Actuators A Phys.*, vol. 274, pp. 199–210, May 2018.

[10] L. Bengtsson, "Direct analog-to-microcontroller interfacing," *Sens. Actuators A Phys.*, vol. 179, pp. 105-113, 2012.

[11] F. Reverter and M. Gasulla, "Experimental characterization of the energy consumption of ADC embedded into microcontrollers operating in low power," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, Auckland, New Zealand, 20-23 May 2019.

[12] F. Reverter and R. Pallàs-Areny, "Uncertainty reduction techniques in microcontroller-based time measurements," *Sens. Actuators A Phys.*, vol. 127, no. 1, pp. 74–79, Feb. 2006.