Demo: A High-Performance, Energy-Efficient Node for a Wide Range of WSN Applications

Federico Terraneo
DEIB
Politecnico di Milano
federico.terraneo@polimi.it

Alberto Leva
DEIB
Politecnico di Milano
alberto.leva@polimi.it

William Fornaciari
DEIB
Politecnico di Milano
william.fornaciari@polimi.it

Abstract

We present WandStem, a powerful node for WSN applications, ranging from low-power to high-performance ones. WandStem has a powerful 48MHz 32bit CPU with memory protection; thus, it can run *untrusted* code, to the advantage of security and dependability. It also provides fast current sensing, supporting Coulomb counting and energy profiling. It encompasses hardware timers for packet transmission and reception timestamping allowing accurate time synchronisation and TDMA communication protocols. The node deep-sleep current consumption is around 2.4μA; thus supporting long battery lifetime also in ultra-low power applications.

1 Introduction

Observing the current WSN landscape, both low-power and high-demand applications often struggle with a mix of the energy efficiency, computational power, and development simplicity limitations of current WSN platforms. In the last years, such a scenario has led to achieving ultra-low power by an integrated hardware and software design [6], and has even induced the idea of a “post-mote era” [5].

Roughly speaking, the available WSN platforms lie between two *extrema*. On one side are low-end motes, suitable for low-power applications [7]. On the other side stand high-end, sometimes even Linux-capable nodes [4]. However, no high-end node reached large enough a diffusion to become a standard like the TelosB [7] did in the low-end. A possible explanation can be found in [3], which compares two high-end nodes with the TelosB, using conveniently selected microbenchmarks. The result is that the high-performance nodes tested in the quoted paper are overall still *not* as energy efficient due to their higher deep sleep consumption.

This can be appreciated by observing Figure 1, showing the MIPS and the deep sleep current (entire node with RAM retention and RTC) of some relevant nodes, distinguishing

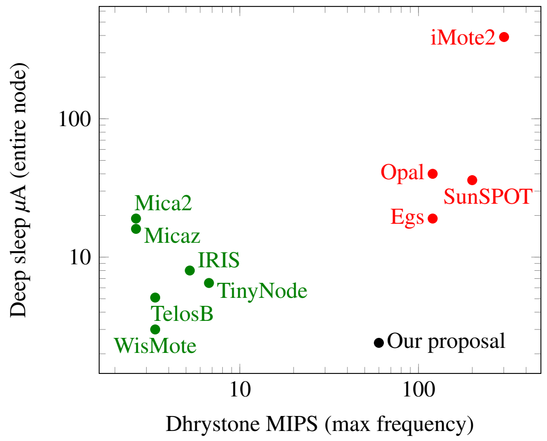

Figure 1. WSN hardware platforms organised by computational power and deep-sleep consumption.

low-power ones (green) and high-performance ones (red). In addition, there is also an unfulfilled need for energy consumption *introspection* along the entire life of a WSN. State of the art solutions are plug-in modules [10], suited for the development phase but less so for use after the WSN deployment. A platform offering the features above, plus precise (sub-μs) timestamping and memory protection, is WandStem, the node that we now come to present in detail.

2 The WandStem node

WandStem is a low-cost WSN node built using a two-layer PCB and COTS components. The node features a low-power efm32gg332f1024 microcontroller based on the ARM Cortex-M3 architecture. This microcontroller, operates at up to 48MHz and provides enough resources (1MB flash, 128KB RAM) to make the abstractions of advanced WSN operating systems affordable. The transceiver is based on the common cc2520, simplifying interoperability with existing WSNs. Sensors are connected to the node through an expansion connector exposing 20 GPIOs as well as ADC channels, SPI and I^{2}C. Finally, WandStem has a USB connector for PC interfacing, and provides a bootloader for an easy programming. The design is released as *open source hardware* under a creative commons license; the schematic and PCB layout can be downloaded at [1]. A working prototype of the node is shown in Figure 2. The remaining part of this section details the WandStem design choices that are most relevant to fulfill the requirements evidenced so far.

International Conference on Embedded Wireless
Systems and Networks (EWSN) 2016
15–17 February, Graz, Austria

ISBN: 978-0-9949886-0-7

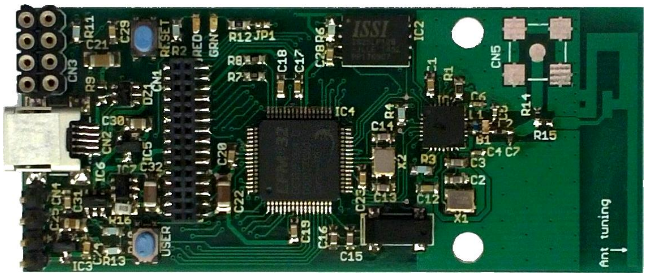
Figure 2. Prototype of the WandStem node

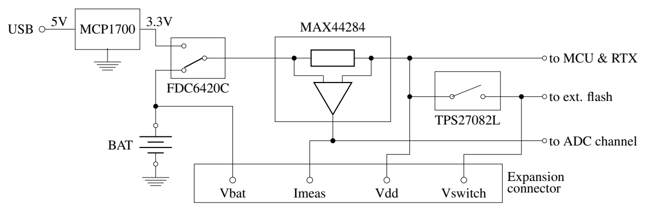

Figure 3. WandStem power tree and current sensor.

2.1 Power tree and energy metering

The node has two power switches, shown in Figure 3. The first one is used to automatically power the node from the USB whenever it is connected to a PC. The second switch is software-controlled, and provides a way to turn on high-power peripherals, such as external sensors, only when needed. A deep-sleep consumption of just 2.4μA is achieved, despite the considerable processing power. This includes full RAM retention for the CPU and 32KHz RTC.

To support energy introspection, a current sense resistor is added at a convenient place, and connected to a high-side current sense amplifier, whose output is attached to an ADC channel of the microcontroller. The measurement range is up to 208mA with a resolution of 51μA and a maximum error of 3%. The 3dB bandwidth of the amplifier is 3KHz, allowing to track also highly variable current consumption.

To provide accurate Coulomb counting, one has to distinguish two cases. When the node is active, current is sampled by the microcontroller ADC. This is done in hardware not to overload the processor, since the ADC has internal support for measurement accumulation. When the node is in deep sleep, energy is computed based on the sleep time and an off-line characterization of the node deep sleep consumption also taking temperature into account. Figure 4 shows the node consumption in deep sleep compared with the model used for Coulomb counting.

2.2 Timing infrastructure

Timing-related operations in WandStem are meant to be hardware-assisted, allowing precise control of the radio avoiding software-induced jitter. This feature eases the implementation of TDMA protocols, constructive interference schemes [2] and time synchronization [9]. In WandStem, the timer *input capture* feature is used to provide hardware timestamping of events. Three event timestamping sources are provided, including incoming packets from the radio, application-related events, and a third one used to timestamp the 32KHz RTC clock from the high-resolution one, to implement VHT [8]. WandStem also supports hardware-timed event generation, using the *output compare* module to provide hardware-timed packet transmission. The timer software driver extends the hardware 16bit timer to 64 bits, thus allowing a virtually overflow-free time representation with a resolution down to 21ns.

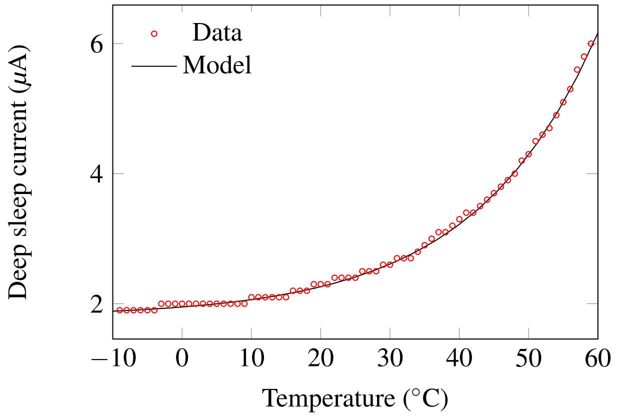

Figure 4. Deep-sleep consumption vs temperature.

2.3 OS support

From the software point of view, Wandstem is supported by the Miosix [1] operating system. The radio drivers include support for the Glossy [2] flooding scheme and FLOPSYNC-2 [9] clock synchronization. The timing infrastructure is based on VHT [8]. At present, we are also in the process of porting TinyOS on WandStem.

3 References

[1] https://miosix.org/wandstem.html.

[2] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with Glossy. IPSN, pages 73–84, 2011.

[3] J. Ko, K. Klues, C. Richter, W. Hofer, B. Kusy, M. Bruenig, T. Schmid, Q. Wang, P. Dutta, and A. Terzis. Low power or high performance? a tradeoff whose time has come (and nearly gone). In G. Picco and W. Heinzelman, editors, *Wireless Sensor Networks*, volume 7158, pages 98–114. 2012.

[4] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling. IMOTE2: serious computation at the edge. In *Proc. International Wireless Communications and Mobile Computing Conference*, pages 1118–1123, Crete Island, Greece, 2008.

[5] P. Pannuto, M. P. Andersen, T. Bauer, B. Campbell, A. Levy, D. Culler, P. Levis, and P. Dutta. A networked embedded system platform for the post-mote era. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, SenSys '14, pages 354–355, 2014.

[6] M. Pasha, S. Derrien, and O. Sentieys. A complete design-flow for the generation of ultra low-power WSN node architectures based on micro-tasking. In *Proc. 47th ACM Design Automation Conference*, pages 693–698, Anaheim, CA, USA, 2010.

[7] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proc. 4th International Symposium on Information Processing in Sensor Networks*, pages 364–369, Los Angeles, CA, USA, 2005.

[8] T. Schmid, P. Dutta, and M. B. Srivastava. High-resolution, low-power time synchronization an oxymoron no more. IPSN, pages 151–161, 2010.

[9] F. Terraneo, L. Rinaldi, M. Maggio, A. Papadopoulos, and A. Leva. FLOPSYNC-2: Efficient monotonic clock synchronisation. In *Proc. 2014 Real-Time Systems Symposium*, pages 11–20, Roma, Italy, 2014.

[10] R. Zhou and G. Xing. Demo abstract - nemo: A high-fidelity noninvasive power meter system for wireless sensor networks. In *Information Processing in Sensor Networks (IPSN), 2013 ACM/IEEE International Conference on*, pages 317–318, 2013.