Low cost high resolution ampere meter for automated power tests for constrained devices

Mario Hoss[1,2], Florian Westmeier[1,3], and Jens-Peter Akelbein[1,4]

[1] University of Applied Sciences, Darmstadt, Germany
[2] Research Assistant, mario.hoss@h-da.de
[3] Student, florian.westmeier@stud.h-da.de
[4] Professor for Computer and Software Engineering, jens-peter.akelbein@h-da.de

Abstract. In recent years, there has been a trend for operating systems to replace more and more conventional baremetal applications, even in constrained class 1 and 2 devices. This leads to new challenges for battery-powered IoT devices, where the battery should not be recharged or replaced regularly. The energy consumption of software must be considered throughout the development. For software that is encapsulated by an operating system and runs on multiple different hardware platforms, this means that power consumption must be determined for each of them. This paper proposes a measuring unit to enable automated regression testing for a variety of different hardware platforms with individual range and resolution requirements. A prototype was created and evaluated in which multiple different shunt resistor measurement modules are switched out by an analog multiplexer, controlled by a HIL platform. The results show that the approach is feasible and offers the possibility of a cheap, scalable, easily customizable measuring unit for the usage in automated power tests.

Keywords: Current measurement · Automatic testing · IoT

1 Introduction

Driven by the exponential growth of the Internet of Things (IoT) it can be observed that a large ratio of these interconnected things are constrained devices [1]. Wearables, smart homes, smart cities and the Industrial IoT introduce embedded devices not only limited in terms of CPU power, but restricted by the available energy. This trend is reflected by terms like ”Set-and-forget”- devices [2] equipped with non-replaceable batteries for ”lifetime-energy-limited” usage, and by ”period-energy-limited” applications replacing batteries periodically [1]. Currently, such devices achieve lifetimes of 5 till 10 years with a typical coin cell battery (like CR2032). Most of the time such devices reside in energy efficient deep sleep modes resulting in an energy consumption less than a factor 10^{-6} compared to active modes. Different to earlier generations of constrained devices, it is no longer common practice deploying hardware specific baremetal applications. With the spread of operation systems (OS) for class 1 and class 2 constrained devices, large parts of the software consists of hardware independent code running on multiple different hardware platforms. To manage the resulting complexity and allow for a safe maintainability of all feasible platforms, it becomes necessary to introduce automated testing, especially in the form of regressions.

For verifying "lifetime energy-limited" devices, tests should not be restricted to finding functional defects only. Test cases should also aim for discovering abnormal energy consumption which needs to be classified as a non-functional defect of a devices behavior. Beyond finding such defects, an automated test infrastructure allows monitoring the code down to individual code changes which may trigger changes of power usage resulting in savings or increases. To determine the energy consumption of a given piece of code, it is either necessary to measure the execution on the target hardware with a power analyzer providing a large enough range and resolution, or to have a sufficient accurate energy model of the target platform. Both approaches are problematic in regards to the scale of the resulting test environment. With the usage of operating systems supporting dozens or hundreds of boards, it becomes necessary to test applications on every OS supported board. As a result, the energy consumption also needs to be predicted for every one of those boards.

This paper proposes a low cost, easily obtainable energy measuring setup for monitoring changes in energy consumption for performing automated regression testing on constrained devices. Furthermore, an evaluation is performed whether the proposed measuring unit is suitable for practical usage in automated testing environments.

2 Related Work

In order to create a model of a devices energy consumption, it is necessary to determine CPU power usage driven by individual instructions as well as the power consumed by peripheral hardware components like transceivers. The first model on instruction level was proposed in [3]. The average power consumption of a system is measured and multiplied by the clock period and the number of clock cycles. Later approaches focus on determining the worst-case energy consumption (WCEC) as the energy useage may vary between instructions with shorter or longer runtime [4]. Strict bounds of WCEC-based analysis were shown lately, as determining data dependent dynamic power consumptions results in an NP-Hard problem, where an approximation cannot be made to an usable degree [5]. Former studies often found the variation in the data dependent dynamic power consumption to be not significant. So it is accounted as a constant value [3,4,6,7]. However, in class 2 constrained devices examined in [5], the variation of the data dependent power consumption amounts to up to 42% of a cores power dissipation. Another study shows cases with even up to 50% [8]. As a result, current approaches focus on predicting the WCEC by methods using statistical analysis or genetic algorithms instead [9].

The energy model for different hardware components and the respective power modes often consists of a finite state machine. The target hardware is modeled as different power states and their connecting transitions. For all identified states and transitions, the power consumption is measured. The parameters influencing the power consumption of a state, as well as their behavior, need to be identified. This can be achieved by utilizing regression analysis to create approximation functions for the parameter dependent energy consumption [10]. The first approach to create such a mode used the utilization of a peripheral hardware module as a trigger for a transition [11]. Later work instead correlated power bursts to individual system calls, as the power states of peripheral hardware often do not line up with their utilization in software [12]. A transceiver for example does not immediately go back to its sleep state after sending the data, but will continue to consume power for a short while after [12]. Automatically finding these states and refining the state machines is still an open research question. In practice the creation of state machines for energy models require a lot of repetitive manual interactions [13].

Different to energy models which enable static code analysis, it is also possible to run each test on the target hardware, measure the resulting power consumption and correlate it to the source code. The most common and cheap approach is to measure the voltage drop over a shunt resistor by utilizing an analog digital converter(ADC). Additionally, an Operational Amplifier(OpAmp) is used to amplify the voltage drop [14–16]. Other authors proposed the usage of multiple different shunt resistors or measuring units to reach a higher range and resolution. For the ”Nemo” [17] a microcontroller is switching out five different stages of shunt resistors. In [18] a second shunt resistor OpAmp combination can get added to the circuit, feeding into the same ADC. In the ”Rocketlogger” [19] two additional complete measuring units can be added, with the resulting voltage drop in the circuit being compensated by a differential amplifier.

3 Design Considerations

As it is necessary to determine the power consumption of code being executed on different hardware platforms, the use of power models as well as commercially available measuring units are not practical solutions due to the cost incurred with each hardware target, as well as manufacturer specific interoperability issues. While singular shunt ADC measuring units can generally not provide the needed range and resolution, there are solutions in the field of large scale wireless sensor network (WSN) verification environments (or testbeds), which achieve the necessary range by switching out shunts or adding complete measuring units. However, the solutions are designed to measure the same range and resolution for each identical hardware platform in a testbed.

The presented approach is designed to switch out multiple measuring modules, taking advantage of the availability of breakout boards for low cost power monitor ICs. By doing so, measurement modules can be fitted with different shunt resistors to match individual hardware platforms and their respective energy modes. This way, the proposed measuring setup can be fitted for different ranges and resolutions, but be designed as a mass-producible cheap adapter board. With the ability to calibrate individual measurement modules, a switch occurs as soon as the target hardware changes from any of the multiple sleep modes to active processing. This allows to detect differences in the resulting power consumption of functionalities in different power modes requiring different resolution ranges. Should inaccuracies occur as a result of the switch and mask spikes in power consumption, it is possible to repeat tests without automated range switching, once for each measurement module.

A common approach for automated tests on constrained devices is the use of a hardware-in-the-loop (HIL) platform that flashes a new executable onto the target and sends the results to an automation server. The proposed measuring setup is connected to the HIL platform, a Raspberry Pi, which is triggered by a Jenkins Server to flash the target platform and return the output. For automated power tests, measurements are tagged with timestamps for being correlated with the targets output. This is realized by matching the given time stamp of the measuring unit with one of the target platform. In this way, the measurements are timestamped in a real-time system (RTS) and processed on the more powerful HIL platform

4 Measurement Unit

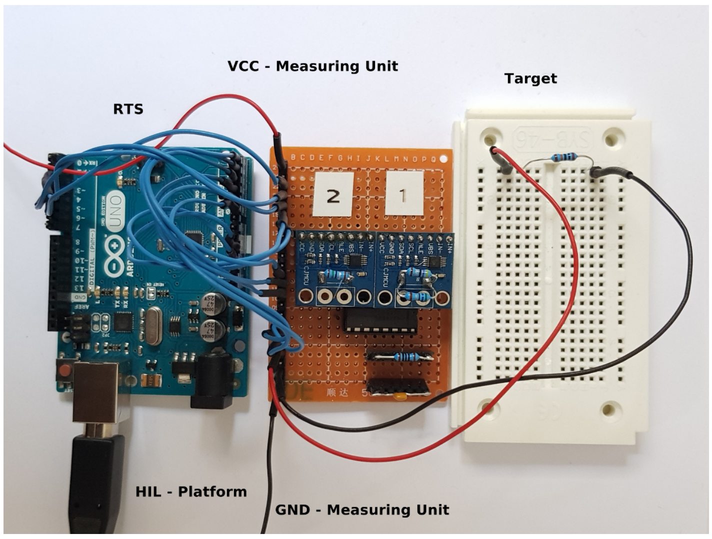

Fig. 1. Measuring Unit

Each measurement module uses a Texas Instruments INA226 power and current monitor IC. Here, the IC is used as a high side current sense amplifier, in which the voltage drop over an external shunt resistor is read by a 16 bit ADC. The shunt is connected in series with the load to be measured, so that the voltage drop can be used to calculate the current through the entire circuit. The module is connected over an I^2C bus also providing internal calibration and the option to read only the average over a series of measurements to correct individual measurement errors.

The INA226 is capable of measuring a voltage drop between -81.9175 mV and 81.92 mV resulting in a resolution of 2.5 µV per bit step of the ADC. So for example, a shunt resistor of 1.5Ω leads to a range of up to 54.61 mA with a resolution of $1.\overline{6}$ µA. Two INA226 breakout boards from CJMCU fitted with different sized shunt resistors are connected in parallel to each other and via an MAX4619 analog CMOS multiplexer in series to the load. To change the measuring range, the multiplexer switches one of the two measurement modules into the circuit. At a supply voltage of 3.3V, the MAX4619 switching process takes a maximum of 25 ns according to its data sheet. A capacitor with series resistor is connected in parallel to guarantee the power supply to the load during the switching process.

The measurement modules and the multiplexer are controlled by an RTS, in this case an Arduino Uno. The INA226 are configured for the shortest conversion time of 140 µs per sample. Each INA226 notifies the RTS about newly available samples via a PIN interrupt. The measured data of the active unit is then read via a dedicated I^2C bus. Once a measurement matches a predefined threshold, the RTS switches the measuring ranges and modules by activating the multiplexer.

The measured values are timestamped and transmitted to the HIL platform via a serial interface. The average function of the INA226 is not used for preserving all raw data as evaluation input.

5 Evaluation

In order to achieve a resolution of the measured current of at least 1 µA, the first module is equipped with two 1% resistors connected in parallel. Both together lead to a resistance of about 2.56 Ω. According to Ohms Law, the circuit offers a resolution of around 0.98 µA. The second measurement module is equipped with a 1.5 Ω resistor for measuring currents of up to 54.61 mA. The capacitor (4,7 nF with a 66 Ω series resistor) is dimensioned to recharge within the ADC conversion time and to not drop below 3V during a switching process in order to measure a SAM R21 Xplained Pro Board without modifications [20]. This allows for a module to switch after each measurement. Before the first measurement takes place the capacitor has to be charged for 31.02 µs.

The time resolution between two new samples currently averages only 4.35 ms at the HIL platform. This is caused by the decision to use an Arduino Uno as RTS. The serial connection from RTS to HIL platform as well as the I^2C connection between INA226 and RTS are both bottlenecks in the current setup. The transmission of unoptimized log text takes a minimum of 3.906 ms including the measured value and the timestamp at 45 byte using a baud rate of 115200 bd. The chosen software I^2C library supports multiple I^2C interfaces where the I^2C bus speed is only 65 kHz. With the standard I^2C library the sample rate is around 470 µs at 400kHz bus speed for requesting and receiving two byte of measurement data. Looking at a logic trace, the transmission of two byte of data at 400 kHz takes around 70 µs. An I^2C command to read from a specific register takes an expected transmission of 5 byte data requiring 190 µs. The library however reads an additional configuration register.

The accuracy of the measured values was performed using resistors with a tolerance of 1 %. For the lower end of the measuring range with a 470 kΩ and in the upper end with a 100 Ω resistor. The expected measured value for the lower measuring range at 3.3 V is between 6.95 µA and 7.09 µA, due to the tolerance of the resistors. The actual measured value varies between 5.01 µA and 8.93 µA. The average of several measurements results in a value of 6.94 µA. Thus the relative error is only 0.2%. For the upper measuring range, the expected value is between 32.67 mA and $33.\overline{3}$ mA. The actual measured current fluctuated between 30.50 mA and 30.51 mA. The mean value is 30.50 µA, the relative error 6.5%.

Both measurement modules operate continuously. Since only one measurement module in the circuit is flown by a current at a time, the other one can not measure a voltage drop at the shunt. Switching the multiplexer to the other measurement module during a running measurement results in an erroneous measurement. For preventing the creation of measurement values, the first sample after a changeover is ignored by the RTS. The first valid sample after this switching process is delivered after a maximal two times of the time resolution.

6 Conclusion and future work

A cheap, scaleable, easily customizable and digitally controllable measuring unit was developed to enable automated power test for a wide array of constrained devices. While the choice of RTS did act as a bottleneck for the sample rate and, as a result, the switching delay, it can be fixed by replacing the RTS. With the INA226 supporting I^2C high-speed mode and the option to connect the RTS to the HIL platform via JTAG, measurements at a sample rate of close to the 140µs conversion time of the INA226 should be feasible. Measuring inaccuracies of the INA226 in the setup are largely limited to one resolution step, with only around 2% of the measured values for the 470k resistor being off by more then one step. This can be adjusted by choosing a shunt resistor resulting in double the resolution and half the range. Such tradeoff would be reasonable for measuring the sleep modes. However, moving from a breadboard prototype with 1% resistors as a shunt to a PBC and a lower tolerance shunt resistor should further improve the measurements accuracy.

To cover the error class of extremely short wake-ups from sleep modes, the use of an analog multiplexer allows more modules to be added and selected on a per test basis. So for example an additional module with a high sample rate ADC and a low range shunt resistor could be added.

A future version of the measuring unit is planned using a RTS supporting high-speed I^2C communication as well as a JTAG interface. Such improvement allows a higher sample throughput and automated firmware changes for the measuring units. For example a ATSAMD21 based RTS can support multiple high-speed I^2C interfaces via their SERCOM architecture.

References

1. Bormann, C., Ersue, M., Keranen, A.: Terminology for constrained-node networks. RFC 7228, RFC Editor (May 2014), http://www.rfc-editor.org/rfc/rfc7228.txt
2. Jayakumar, H., Lee, K., Lee, W.S., Raha, A., Kim, Y., Raghunathan, V.: Powering the internet of things. In: Proceedings of the 2014 International Symposium on Low Power Electronics and Design. pp. 375–380. ISLPED '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2627369.2631644
3. Tiwari, V., Malik, S., Wolfe, A., Lee, M.T..: Instruction level power analysis and optimization of software. In: Proceedings of 9th International Conference on VLSI Design. pp. 326–328 (Jan 1996). https://doi.org/10.1109/ICVD.1996.489624
4. Jayaseelan, R., Mitra, T., Li, X.: Estimating the worst-case energy consumption of embedded software. In: 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06). pp. 81–90 (April 2006). https://doi.org/10.1109/RTAS.2006.17
5. Morse, J., Kerrison, S., Eder, K.: On the limitations of analyzing worst-case dynamic energy of processing. ACM Trans. Embed. Comput. Syst. **17**(3), 59:1–59:22 (Feb 2018). https://doi.org/10.1145/3173042
6. Wgemann, P., Distler, T., Hnig, T., Janker, H., Kapitza, R., Schrder-Preikschat, W.: Worst-case energy consumption analysis for energy-constrained embedded systems. In: 27th Euromicro Conference on Real-Time Systems. pp. 105–114 (July 2015). https://doi.org/10.1109/ECRTS.2015.17
7. Georgiou, K., Kerrison, S., Chamski, Z., Eder, K.: Energy transparency for deeply embedded programs. ACM Trans. Archit. Code Optim. **14**(1), 8:1–8:26 (Mar 2017). https://doi.org/10.1145/3046679
8. Ascia, G., Catania, V., Palesi, M., Sarta, D.: An instruction-level power analysis model with data dependency. VLSI Design **12**(2), 245–273 (2001). https://doi.org/10.1155/2001/82129
9. Pallister, J., Kerrison, S., Morse, J., Eder, K.: Data dependent energy modeling for worst case energy consumption analysis. In: Proceedings of the 20th International Workshop on Software and Compilers for Embedded Systems. pp. 51–59. SCOPES '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3078659.3078666
10. Friesel, D., Buschhoff, M., Spinczyk, O.: Parameter-aware energy models for embedded-system peripherals. In: 2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES). pp. 1–4 (June 2018). https://doi.org/10.1109/SIES.2018.8442096
11. Shnayder, V., Hempstead, M., Chen, B.r., Allen, G.W., Welsh, M.: Simulating the power consumption of large-scale sensor network applications. In: Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems. pp. 188–200. SenSys '04, ACM, New York, NY, USA (2004). https://doi.org/10.1145/1031495.1031518
12. Pathak, A., Hu, Y.C., Zhang, M., Bahl, P., Wang, Y.M.: Fine-grained power modeling for smartphones using system call tracing. In: Proceedings of the Sixth Conference on Computer Systems. pp. 153–168. EuroSys '11, ACM, New York, NY, USA (2011). https://doi.org/10.1145/1966445.1966460

13. Buschhoff, M., Friesel, D., Spinczyk, O.: Energy models in the loop. Procedia Computer Science **130**, 1063 – 1068 (2018). https://doi.org/10.1016/j.procs.2018.04.154, the 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops
14. Haratcherev, I., Halkes, G., Parker, T., Visser, O., Langendoen, K.: PowerBench: A Scalable Testbed Infrastructure for Benchmarking Power Consumption, pp. 37–44. s.n. (2008), haratcherev:2008
15. Hartung, R., Kulau, U., Wolf, L.: Distributed energy measurement in wsns for outdoor applications. In: 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). pp. 1–9 (June 2016). https://doi.org/10.1109/SAHCN.2016.7732983
16. Lim, R., Ferrari, F., Zimmerling, M., Walser, C., Sommer, P., Beutel, J.: Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In: 2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). pp. 153–165 (April 2013). https://doi.org/10.1145/2461381.2461402
17. Zhou, R., Xing, G.: Nemo: A high-fidelity noninvasive power meter system for wireless sensor networks. In: 2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). pp. 141–152 (April 2013). https://doi.org/10.1145/2461381.2461426
18. Ptsch, A., Berger, A., Springer, A.: Efficient analysis of power consumption behaviour of embedded wireless iot systems. In: 2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). pp. 1–6 (May 2017). https://doi.org/10.1109/I2MTC.2017.7969658
19. Sigrist, L., Gomez, A., Lim, R., Lippuner, S., Leubin, M., Thiele, L.: Measurement and validation of energy harvesting iot devices. In: Design, Automation Test in Europe Conference Exhibition (DATE), 2017. pp. 1159–1164 (March 2017). https://doi.org/10.23919/DATE.2017.7927164
20. Atmel Corporation: SAM R21 Xplained Pro User Guide (Apr 2016), http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42243-SAMR21-Xplained-Pro_User-Guide.pdf, rev.: Atmel-42243D-SAM-R21-Xplained-Pro_User Guide-04/2016