Hazelnut - An Energy Efficient Base IoT Module for Wide Variety of Sensing Applications

Branimir Pervan*
Filip Turcinovic
branimir.pervan@fer.hr
filip.turcinovic@fer.hr
University of Zagreb, Faculty of Electrical Engineering and Computing
Zagreb, Croatia

Emanuel Guberovic*
eguberovic@green-light.agency
Green Light Technologies Ltd.
Zagreb, Croatia

ABSTRACT

With the everlasting expansion of Internet of Things (IoT), backed by the increased availability of cheap hardware making it available for home users, more and more modules are finding their usage in everyday situations. To maximize the pervasiveness of IoT modules, human intervention has to be taken to a bare minimum. This primarily addresses the need for relatively frequent battery charging, due to the fact that cheap microcontrollers for IoT modules available to the masses are still significantly energy inefficient. This renders such modules hardly usable in amateur and semi-professional environments.

As a solution to the problem of cheap but energy inefficient modules, this paper introduces a concept of designing an IoT module of two microcontrollers: one powerful and peripheral-rich with networking capabilities, and one extremely energy-saving, used to wake up the first one from a deep sleep mode when more intensive tasks are processed and functionalities needed. The system is described from a top-view along with its implementation in a case study. Power consumption is evaluated practically through a well-defined measurement methodology and mutual comparison of the results. We show that by adding an additional cheap and energy-saving microcontroller, one can achieve significant energy savings with an insignificant rise in the overall price of the system.

CCS CONCEPTS

• **Computer systems organization → Embedded hardware**; • **Hardware** → *Impact on the environment*.

KEYWORDS

internet of things, energy saving, energy efficiency, embedded systems

*Both authors contributed equally to this research.

ECBS '19, September 02–03, 2019, Bucharest, Romania

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Branimir Pervan, Filip Turcinovic, and Emanuel Guberovic. 2019. Hazelnut - An Energy Efficient Base IoT Module for Wide Variety of Sensing Applications. In *Proceedings of ECBS '19: 6th Conference on the Engineering of Computer Based Systems (ECBS '19).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The Internet of things (IoT) is based on the network of devices communicating with each other, getting information from the environment and storing them on the server for further processing. The idea of having such devices existed since the 1990s but was not reachable mainly due to wireless and battery technologies from that time. Since today's technologies made it possible, IoT is now a rapidly growing network of heterogeneous devices which are mostly used in automatization of everyday processes. That concept of wireless sharing information and automatization lead towards the smart home idea which is to equip a home with sensors and actuators for convenience, security and better management of energy usage. Since a considerable amount of the IoT devices use batteries, one of the biggest problems developers encounter is energy consumption due to network connectivity and performing complex algorithms when needed.

There have been a number of research efforts regarding energy consumption in battery-powered devices. Most of them are based on optimizing algorithms, using more economic sensors or memory and battery technologies. For instance, in [5] authors, after listing challenges for efficient energy supply, introduced emerging memory technologies as a solution. For battery-powered wearable, since many of them belong to domains of data mining, multimedia, and digital processing, authors proposed approximate computing as a design paradigm that can leverage intrinsic resilience of applications to execute computations approximately, resulting in energy improvement. In [1] authors proposed energy efficient algorithm devoted to scheduling the duty cycle of different sensors and appliances. Authors in [2] presented a solution to prevent the battery from surge current or high-rate charge and discharge that might reduce the lifespan of a battery using Model Predictive Control (MPC), a model based on energy management method achieving hybridization. Results in [7] showed that wake-up energy is the main contributor to the overall battery lifetime. Authors presented that a wake-up frequency of less than once per 30 seconds will enable better battery lifetime.

To overcome the problem of relatively low energy-efficiency in advanced IoT application, this paper proposes a low-cost solution based on combining two microcontrollers: ESP8266 and ATtiny85. Both of the microcontrollers have large community and good vendor support and can be acquired for less than $3 combined. The experimental system based on mentioned microcontrollers combined resulted in significant energy savings while retaining the same functionality.

The rest of this paper is organized as follows: Section 2 gives a short overview of the proposed model, Section 3 describes the implementation and both the results of the power consumption measurements and the results acquisition methodology. The paper ends with the conclusion and future work in Section 4.

2 SYSTEM OVERVIEW

The main motivation for combining two microcontrollers into one base module was the fact that, for most top-level applications, a vast majority of modules sleeps (or should sleep) until triggered or until a predefined period of time runs up. Two different types of modules provide the flexibility needed in IoT: the system can decide in runtime if it has to wake up the more powerful microcontroller. Additionally, the system retains the ability to wake itself up periodically, if needed. For this particular paper, we combined ESP8266 [3], a relatively small but powerful microcontroller by Espressif with a rich set of peripherals, including networking module, and ATtiny85 [6], a low-power 8-bit AVR RISC-based microcontroller by Microchip.

Although ESP8266 has three different modes of sleep, not one of them fully achieves energy saving while retaining versatility. In *Modem-sleep* mode, ESP8266 turns Wi-Fi module off consuming $15mA$ of substrate current. In *Light-sleep* mode, situation somehow improves to still unacceptable $0.4mA$ of substrate current consumed with Wi-Fi and System clock being shut down, and CPU put to *Pending* mode. *Deep-sleep* mode enables ESP8266 to turn off everything but its *Real-Time clock* (RTC), consuming approximately 20 µA of substrate current, but with a price of waking up only by de facto hard resetting itself. The reset can be carried out by the aforementioned RTC, preprogrammed with a relatively short and fixed amount of time, or by an external impulse.

3 IMPLEMENTATION AND EVALUATION

To implement the proposed model, we used ESP8266 on Wemos D1 mini development board and TSOP packaged ATtiny85. Attiny85 consumes around 4 µA in power-off sleep mode with watchdog timer turned on. On the other hand, in active state, it consumes around $0.7mA$ when supplied voltage is around $3.3V$ and internal $1Mhz$ oscillator is used. With the same supply voltage ESP8266 draws a supply current of approximately 20 µA in deep sleep mode, whilst its peak supply current for the Wi-Fi module goes up to $170mA$ combined with CPU supply current of $20mA$.

3.1 Methodology

ESP8266 is a widely used micro-controller with a wide variety of possible uses, differing from simply reading a current sensor value to doing complex calculations and sending data over wireless networks. Sometimes these tasks are very simple and their processing takes few seconds or less so micro-controller mostly remain idle, other times they might take more to complete the assigned tasks and CPU and other peripherals work at full load. To analyze how our proposed system handles all of these cases, we evaluated power consumption for different values of active time.

When Hazelnut system is idling, ESP8266 is in deep sleep and ATtiny85 is in power-down mode. ATtiny85 uses a value obtained from watchdog timer and watchdog interrupt counter to decide whether it is time to wake up or put ESP8266 to idle. To simulate the real-case scenario, during the active time, ESP8266 calculates 1000 multiply and accumulate (MAC) operations and sends HTTP GET request with a calculated value in an infinite loop until ATtiny85 sends sleep signal. ATtiny85's watchdog timer is set at 8 seconds, and a complete test period is set at 10 watchdog timer interrupts. Evaluation is executed for different values of active time, varying from 0 to 10 watchdog interrupts. Idle time in every iteration of the evaluation is the difference between complete test period and active time.

Power consumption is sampled with a period of 0.1 seconds, using the RaspberryPI and INA219 [4] I2C module which powers the Hazelnut test system. Measuring system has a resolution of $0.1mA$ or $0.33mW$ with an estimated possible error of 1%. Measured results are compared to the power consumption of single Wemos D1 device working at full load.

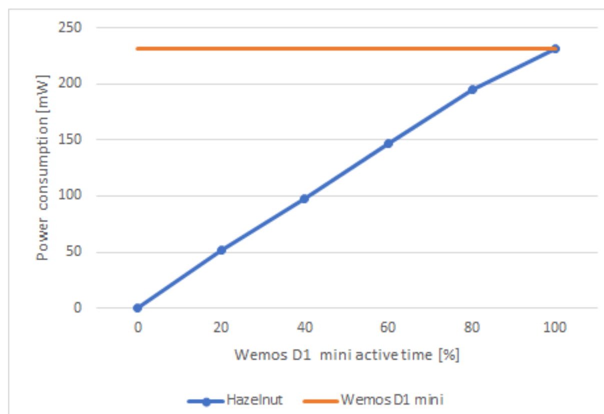

Figure 2: Hazelnut power consumption for different overall active time

3.2 Results

While Hazelnut is in the active state, power consumption is around $231mW$ on average, and it gets down to $0.65mW$ in the idle state. Since power consumption of Wemos D1 device is approximately $231mW$, we can notice that power consumption of the whole system is almost completely determined by it. This due to the fact that ATtiny85, generally a low powered device, spends even less energy when kept in power-off mode. Its power-off current with watchdog timer ON is 6 µA at $3.3V$ of the supply voltage.

Figure 3 depicts exemplary power consumption when Hazelnut is in active state 60% of the time. In that evaluation, iteration active time lasts 48 seconds and idle time lasts 32 seconds. Average power

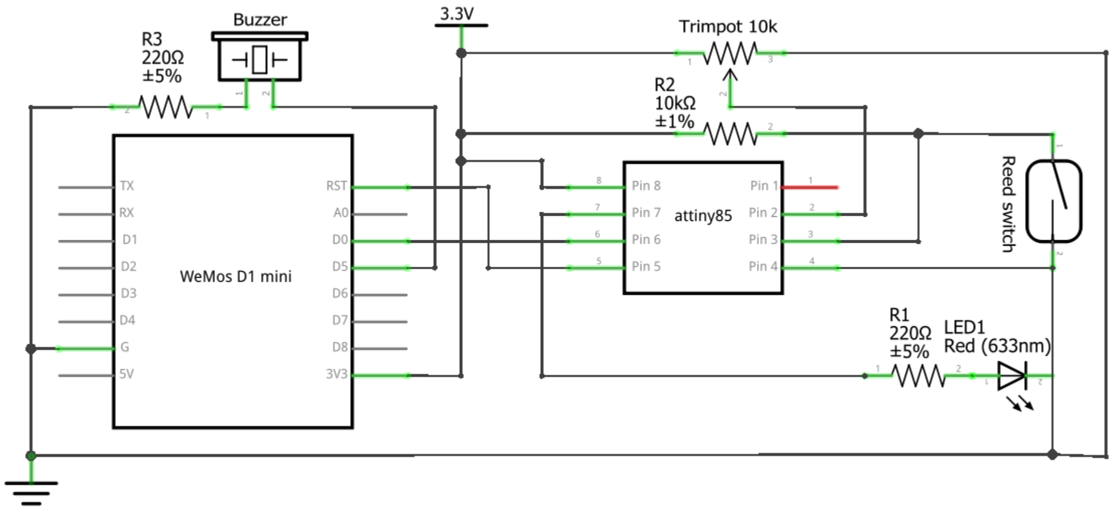

Figure 1: Block scheme of the Hazelnut prototype

in run time of nine iterations is $146.4mW$, which is around 63% of the power consumption of the active Wemos D1 device.

Power consumption scales analogously to the duration of an active state in other evaluation iterations as well:

- $0.65mW$ at 0% active time,
- $51.92mW$ at 20% active time,
- $98.1mW$ at 40% active time,
- $146.4mW$ at 60% active time,
- $195.05mW$ at 80% active time,
- $231.00mW$ at 100% active time.

This results also displayed on figure 2, depict the efficiency proposed system has in reducing power consumption in uses where active time is below 100%, yet quite clearly assert the efficiency in use where active time varies between 0% and 100% of the overall time.

Even at 80% active time, where Wemos D1 is woken from a deep sleep every 16 seconds, power consumption is reduced by 15.57% in comparison to single Wemos D1 device working at full load.

Table 1 depicts minimal, maximal, average and median power consumption for different active time values. Max values are dependent on the Wemos Wi-Fi functionalities used and vary between $472.26mW$ to $592.27mW$ independently of active time percentage. Average values, as previously stated, are analogous to active time percentage. Median values are close to $0mW$ when the active time is smaller than idle time and around $230mW$ when the active time is bigger than idle time.

As stated before, ESP8266 devices are often used to periodically make a single reading or calculation and send those results over Wi-Fi followed by another sleep period. On figure 4 average power consumption of Hazelnut system is shown for different values of wake up time. As in previous evaluations, ESP8266 is calculating 1000 MAC operations and sending data over Wi-Fi network, but this time only one iteration is done every time it is woken up from the idle state. Since the aforementioned iteration of operations takes between 5 and 10 seconds to complete, we have chosen to test wake up periods from 16 to 80 seconds. Here we observe that average power consumption drops dramatically when waking up time changes from 16 to 32 seconds, yet even at 16 seconds wake up time is less than 50% of a full load ESP6288 device.

	Power consumption [mW]			
Active time [%]	Min	Max	Average	Median
0	0.62	1.0	0.65	0.62
20	0.62	472.26	51.92	0.62
40	0.62	592.27	98.1	1.0
60	0.62	506.27	146.4	219.50
80	0.62	477.76	195.05	231.88
100	73.37	475.76	231.0	230.26
100 (Wemos D1 only)	213.26	403.01	231.92	230.88

Table 1: Hazelnut power consumption values for different values of active time

3.3 Practical use-case implementation

To test the implementation, we used a real-life case study, deploying the module to a home refrigerator. To generate a waking-up interrupt signal on ATtiny85, a Reed switch was used. An LED diode was used for a heartbeat signal when the interrupt was generated, and a Buzzer to generate periodic sound if the door stayed open for an amount of time specified by the potentiometer. The schematic of the module is depicted in figure 1. The module was powered by a set of 3 AAA batteries, rated at $1.2V$ and $950mAh$. In that setting, the module worked continuously for 19 days.

4 CONCLUSION AND FUTURE WORK

In this paper, we presented a solution for energy consumption in highly pervasive battery-powered IoT modules. The solution is based on combining two microcontrollers of different types: one with higher processing power and a rich set of peripherals (ESP8266) and other power-saving (ATtiny85) used to intercept interrupts

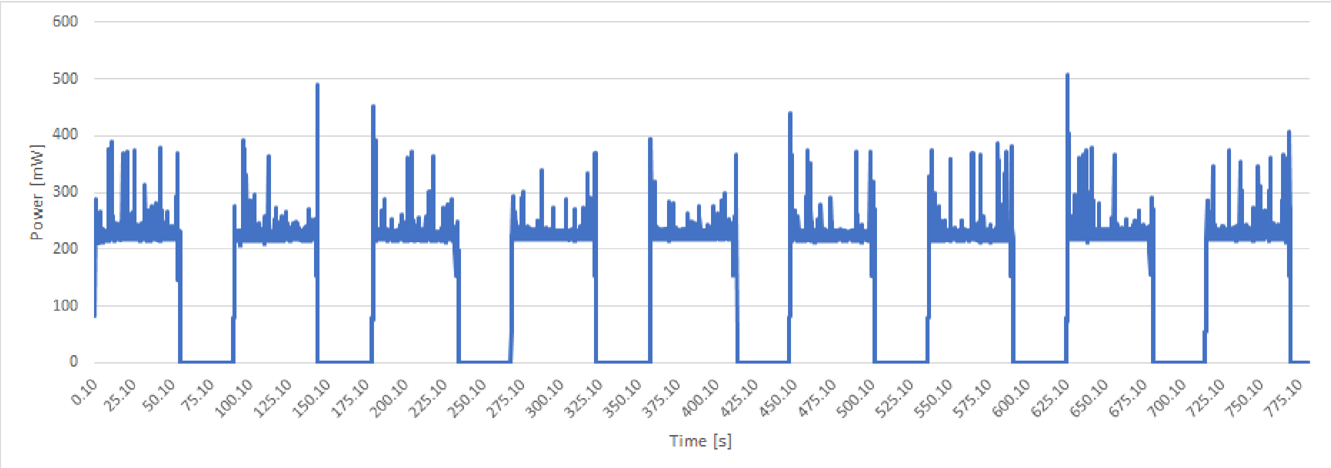

Figure 3: Hazelnut power consumption when Wemos D1 is active 60% of the time

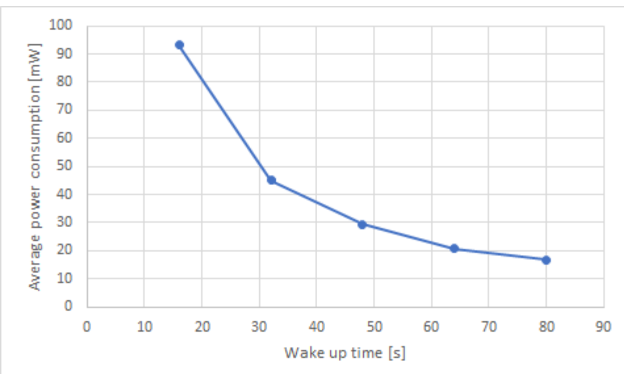

Figure 4: Hazelnut power consumption for different wake up time

to the module and implement the waking-up logic for the first microcontroller. We discussed the advantages of such an approach in terms of gaining flexibility and increasing energy-efficiency while retaining a relatively low price of the overall module.

The experimental system is developed and the results have shown that energy consumption in active state of the system is almost completely determined by the energy consumption of the ESP8266 microcontroller which proves ATtiny85's low energy usage. It is shown that ESP8266 microcontroller in the deep-sleep mode saves $99,7\%$ of the energy which, combined with the ATtiny85's low consumption, enables developed system to save $77,6\%$ of the energy when its active time is 20% and $15,57\%$ when the active time is 80%. The results have also shown that median values of power consumption are around $0mW$ when the active time is smaller than idle time and around $230mW$ when the active time is bigger than idle time.

Future work could focus on replacing a power-saving microcontroller with an, i. e., RC switch, although this would inevitably lead to a significant loss in flexibility.

ACKNOWLEDGMENTS

The source code for every component described in this paper is available at https://github.com/project-hazelnut

REFERENCES

[1] S. F. Abedin, M. G. R. Alam, R. Haw, and C. S. Hong. 2015. A system model for energy efficient green-IoT network. In *2015 International Conference on Information Networking (ICOIN)*. 177–182. https://doi.org/10.1109/ICOIN.2015.7057878
[2] B. A. Aderemi, A. T. Puati Zau, S. Daniel Chowdhury, T. O. Olwal, and A. M. Abu-Mahfouz. 2018. Hybrid Battery Technologies with Battery Management System in Power and Energy Sectors. In *2018 IEEE PES/IAS PowerAfrica*. 716–721. https://doi.org/10.1109/PowerAfrica.2018.8521028
[3] Espressif. 2019. ESP8266. Retrieved March 4, 2019 from https://www.espressif.com/en/products/hardware/esp8266ex/overview
[4] Texas Instruments. 2019. INA219 ZerÃÿ-Drift, Bidirectional Current / Power Monitor With I2C Interface. Retrieved March 8, 2019 from http://www.ti.com/lit/ds/symlink/ina219.pdf
[5] H. Jayakumar, A. Raha, Y. Kim, S. Sutar, W. S. Lee, and V. Raghunathan. 2016. Energy-efficient system design for IoT devices. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. 298–301. https://doi.org/10.1109/ASPDAC.2016.7428027
[6] Microchip. 2019. Attiny85. Retrieved March 4, 2019 from https://www.microchip.com/wwwproducts/en/ATtiny85
[7] S. Tozlu and M. Senel. 2012. Battery lifetime performance of Wi-Fi enabled sensors. In *2012 IEEE Consumer Communications and Networking Conference (CCNC)*. 429–433. https://doi.org/10.1109/CCNC.2012.6181000