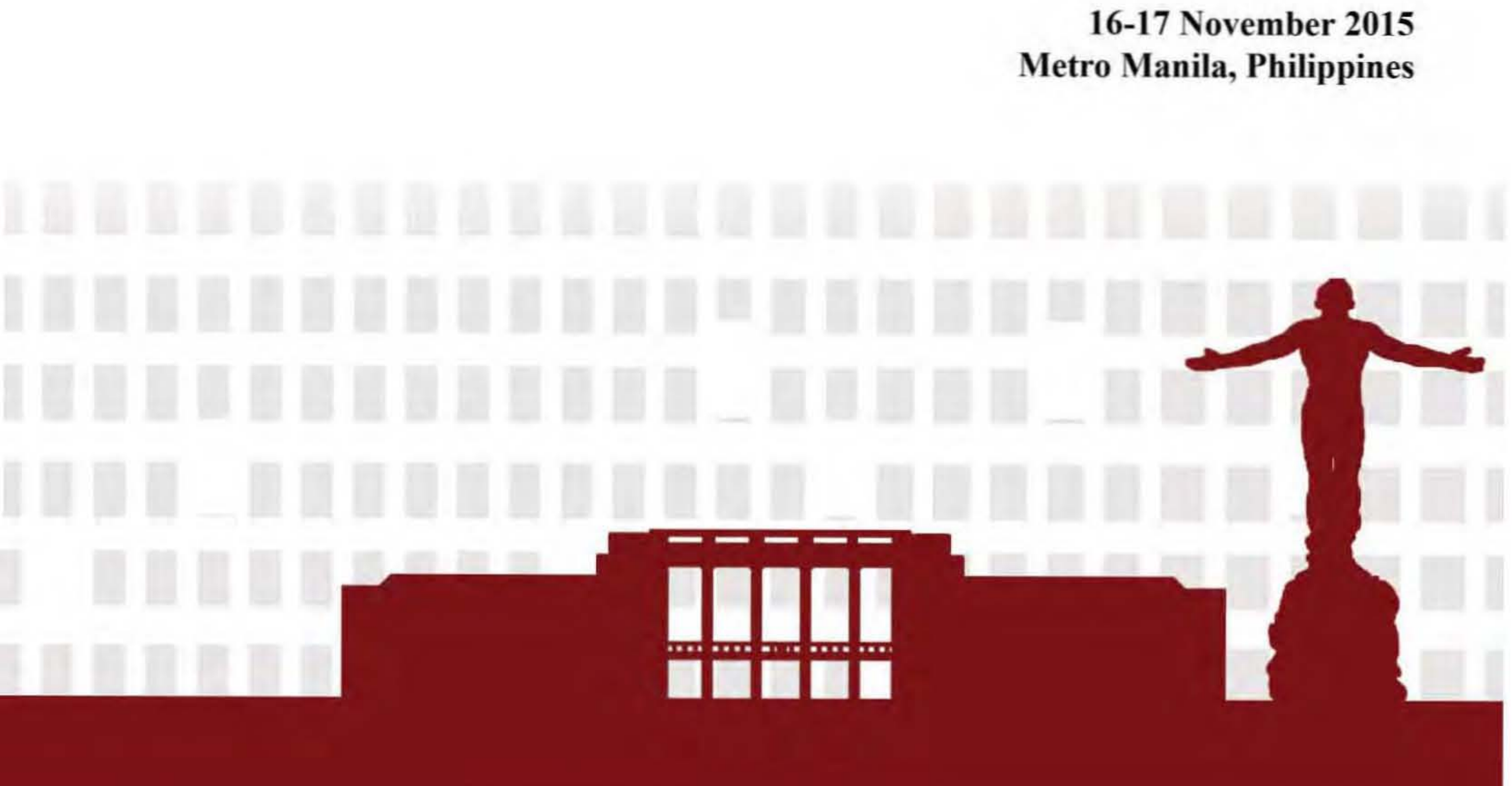
8th AUN/SEED-Net
REGIONAL CONFERENCE ON ELECTRICAL AND ELECTRONICS ENGINEERING

Envision, Enable, and Empower
Smarter and Resilient Societies

co-located with

11th ERDT Conference
on Semiconductor and Electronics, Information and Communications Technology and Energy

16-17 November 2015
Metro Manila, Philippines



Proceedings of the 8th AUN/SEED-Net RCEEE 2015 and 11th ERDT Conference on Semiconductor and Electronics, Information and Communications Technology, and Energy

Editors:
Dr. Joel Joseph S. Marciano Jr.
Dr. Jhoanna Rhodette I. Pedrasa
Dr. Rhandley D. Cajote

ISBN: 978-616-406-075-3

Published by: ASEAN University Network / Southeast Asia Engineering Education Development Network (AUN/SEED-Net) JICA Project
Faculty of Engineering, Bldg. 2
Chulalongkorn University, Bangkok
Thailand 10330

Printed in the Philippines by: ERZALAN PRINTING PRESS
45 Cotabato Street, Luzviminda Village, Batasan Hills, Quezon City, Philippines

8th AUN/SEED-Net Regional Conference
on Electrical and Electronics Engineering 2015

co-located with

11th ERDT Conference
on Semiconductor and Electronics, Information and Communications Technology, and Energy

Envision, Enable and Empower Smarter and Resilient Societies

Published by: ASEAN University Network / Southeast Asia Engineering Education Development Network (AUN/SEED-Net) in partnership with Engineering Research and Development for Technology (ERDT) and University of the Philippines Diliman.

ISBN: 978-616-406-075-3

MODIFIED IEEE 754 UNSIGNED HALF-PRECISION FLOATING POINT LIBRARY FOR CONSTRAINED MEMORY SYSTEMS

Kym Derrick P. Uy[*], Carl C. Dizon and Nestor Michael C. Tiglao
Ubiquitous Computing Laboratory, Electrical and Electronics Engineering Institute
University of the Philippines Diliman, PHILIPPINES.
[*]E-mail: kpuy@up.edu.ph

ABSTRACT
Memory-constrained systems have a limit to the size a numerical representation. Almost all systems catering to this kind do not have a floating-point module. Instead, most resort to integer arithmetic with the assumption that most calculations will not overflow. However, there is a need for a floating-point module for accurate calculations, such as modeling of battery and temperature data of wireless sensor networks (WSNs). Also, battery capacities and energy consumption of WSN motes may improve in the near future that these parameters itself cannot be expressed integrally.

It is crucial for WSNs to be able to calculate numbers to some extent. Some parameters of a WSN node, like battery level and temperature, require some calculations to be meaningful. As technology progresses, these parameters may extend large amounts that it cannot be accurately calculated solely by the node and its limited numerical range. For our energy-harvesting battery simulations with Contiki OS, the battery capacity is large compared its energy consumption. We also intend to simulate a supercapacitor battery model containing exponential functions. Accurate operations with such numbers are difficult using only integral values and a maximum magnitude of 65535 offered by the OS.

A solution this study presents is to implement an unsigned half-precision floating point representation of a number based on the IEEE 754 standard. This was made with the fact that our battery simulations only deal with positive numbers. On a 16-bit number representation, the 10 most significant bits define the mantissa and other 6 bits the exponent. With a +31 exponent bias and no sign bit, this representation can accommodate numbers from 10-12 to 1012. The length of the mantissa offered floating point numbers accurate enough for our purposes. All arithmetic operations except division, which accept two numbers of this format, were implemented in our library. A function to convert a number to this format was also implemented for variable numbers. On the other hand, conversion of constants to this format, and vice versa, were done manually to reduce the number of operations, and subsequently the memory footprint, of the simulation code. For our purposes, the code is written in C and is included as a module in Contiki OS.

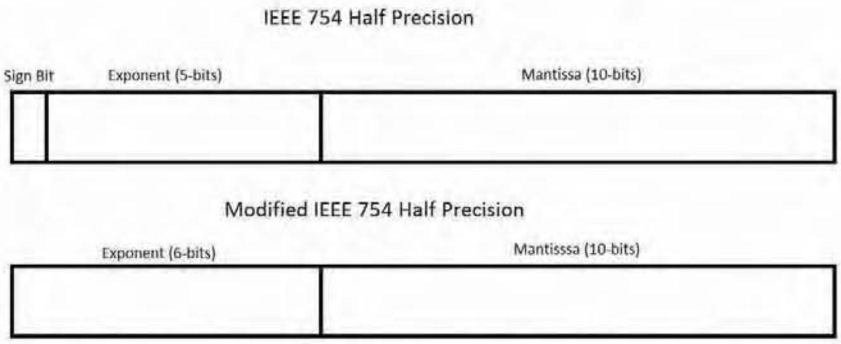


Figure 1. Bit allocation comparison between IEEE 754 standard and our modified version

Keywords: Contiki OS, floating point, IEEE 754

Acknowledgment: This study is part of the SmartWire Project funded by the Department of Science and Technology.

References
[1] K. Kuladinithi, O. Bergmann *et al*, "Implementation of CoAP and its application in transport logistics", *Proc. IP+SN*, Chicago IL, USA, 2011.
[2] W. Kahan, "IEEE Standard 754 for Binary Floating-Point Arithmetic", UC Berkeley, Berkeley CA, USA, 1997.