A Novel Suggestion on Informatics Framework of Smart Devices involving Hardware,Firmware,Software & Communication Protocol/s Verification in the context of IoT based on E Theorem Prover.

N.T.Kumar*,Thiago R,Sandro B,Vinicius R,Lisane V,Vanderlei P
Tech 4 People/Versor Inovação,Santo André – SP,Brazil.
email id* : tejdnk@gmail.com

Abstract :

We present in this short communication **-** *"E Theorem Prover and Verification of Smart Devices & Protocols Based on IoT Environments – A Novel Suggestion on Informatics Framework of Smart Devices involving Hardware,Firmware & Software Verification."*

keywords: E Theorem Prover/Smart Devices/Hardware/Firmware/Software/IoT

Introduction & Inspiration :

Based on the information contained in the published literature it is very much inspirational to use novel methodologies to probe hardware/firmware/software in an innovative way.E Theorem Prover is an excellent high performance tool to design develop,implement and test novel methodologies.We wish to highlight the importance of E Theorem Prover in the context of "Smart Devices" and their testing.Since E Theorem Prover is developed in C it is useful to test and verify "Smart Devices".Refs[1-9]

"E is a high performance theorem prover for full first-order logic with equality.It is based on the equational superposition calculus and uses a purely equational paradigm. It has been integrated into other theorem provers and it has been among the best-placed systems in several theorem proving competitions. E is developed by Stephan Schulz, originally in the *Automated Reasoning Group* at TU Munich." **Source :** https://en.wikipedia.org/wiki/E_theorem_prover

Informatics Framework & Implementation :

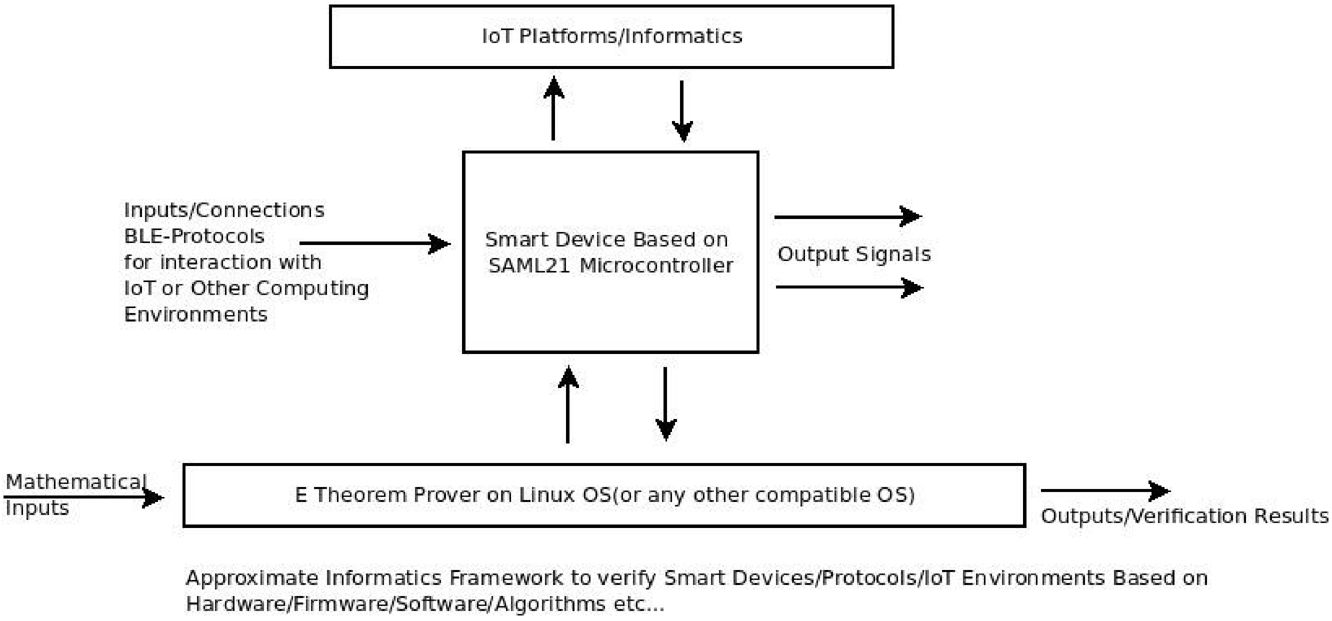
IoT Platforms/Informatics
Inputs/Connections
BLE-Protocols
for interaction with
IoT or Other Computing
Environments
Smart Device Based on
SAML21 Microcontroller
Output Signals
Mathematical
Inputs
E Theorem Prover on Linux OS(or any other compatible OS)
Outputs/Verification Results
Approximate Informatics Framework to verify Smart Devices/Protocols/IoT Environments Based on
Hardware/Firmware/Software/Algorithms etc...

Figure I. Approximate Informatics Framework to probe Smart Devices/Protocols/IoT/Algorithms [Based on Refs[1-9] && Additional Information on Software Used]

Please Note : Figure I & Figure II -Actual Connections/Interfacing with the concerned computing environments or testing methodologies will vary to some extent. Readers are advised to check the requirements before using them.This is an attempt to inspire others to use Theorem Provers in Verification.

For Example : We would like to suggest "Heuristics Approach" in designing,implementing and testing of verification methodologies involving hardware/firmware/software/IoT concepts.For more information please refer to additional information links and the references mentioned in this paper.

Future Direction To Implement Verification Methodologies :
(in extending our Testing/Verification Framework Using Python based Environments)

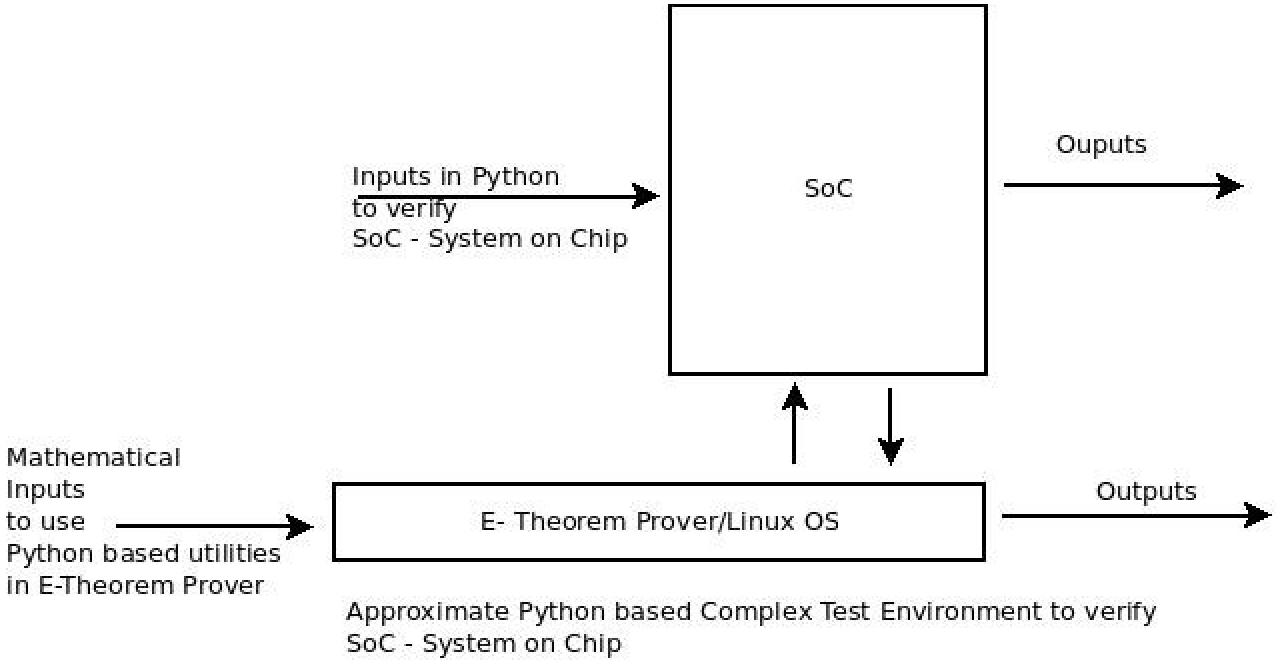

Figure II. Approximate Informatics Framework based on Python /E Prover to implement "Complex Test Environment" for large scale projects.

Source of inspiration : https://www.design-reuse.com/articles/15886/a-phyton-based-soc-validation-and-test-environment.html

Conclusion :

Finally to sum up the idea presented in this communication – it is possible to develop a novel informatics framework as mentioned in the title to perform "E Theorem Prover" based verification of Hardware/Firmware/Software/Communication Protocols towards designing/prototyping/implementing better standards in the context of next generation Smart Devices and IoT Environments using mathematical concepts.

Acknowledgement/s:

No competing financial interest/s is/are declared in preparing this manuscript. This manuscript is meant to inspire others to develop more advanced Hardware/firmware/Software verification and its applications in this demanding area of sleep studies using novel smart devices. The Authors strictly abide by all copyright agreements in using open source software or other such technologies used in this paper. Special thanks to all who made this happen. We thank FAPESP R&D funding via Versor Innovations/Tech 4 People Project,Santo Andre,SP,Brazil for generously supporting our research work.

Additional Information on Software Used:

[a] : https://en.wikipedia.org/wiki/E_theorem_prover

[b] : http://wwwlehre.dhbw-stuttgart.de/~sschulz/E/Technology.html

[c] : Navarro Pérez, Juan Antonio & Rybalchenko, Andrey. (2011). Separation logic + superposition calculus = heap theorem prover. Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI). 46. 556-566.
DOI : 10.1145/1993498.1993563.

[d] : http://www4.informatik.tu-muenchen.de/~schulz/E/E.html

References :

[1] : http://karennicolejohnson.com/wp-content/uploads/2012/11/KNJohnson-2012-heuristics-mnemonics.pdf

[2] : http://wwwlehre.dhbw-stuttgart.de/~sschulz/E/E.html

[3] : http://wwwlehre.dhbw-stuttgart.de/~sschulz/bibliography.html

[4] : https://www.infoq.com/news/2009/10/testing_heuristics

[5] : https://en.wikipedia.org/wiki/Smart_device

[6] : Weiser, Mark (1991). "The Computer for the Twenty-First Century". Scientific American. **265** (3): 94–104. doi:10.1038/scientificamerican0991-94.

[7] : Poslad, Stefan (2009). Ubiquitous Computing Smart Devices, Smart Environments and Smart Interaction. Wiley. ISBN 978-0-470-03560-3.

[8] : https://en.wikipedia.org/wiki/Sensor_node

[9] : https://www.techbriefs.com/component/content/article/tb/techbriefs/electronics-and-computers/2227