ISSN 1726-5479

SENSORS & TRANSDUCERS

7 vol. 142 /12

Sensor Networks and Wireless Sensor Networks

International Frequency Sensor Association Publishing

IFSA

Sensors & Transducers

Volume 142, Issue 7,
July 2012
www.sensorsportal.com
ISSN 1726-5479

Editors-in-Chief: professor Sergey Y. Yurish, tel.: +34 696067716, e-mail: editor@sensorsportal.com

Editors for Western Europe
Meijer, Gerard C.M., Delft University of Technology, The Netherlands
Ferrari, Vittorio, Universitá di Brescia, Italy

Editors for North America
Datskos, Panos G., Oak Ridge National Laboratory, USA
Fabien, J. Josse, Marquette University, USA
Katz, Evgeny, Clarkson University, USA

Editor South America
Costa-Felix, Rodrigo, Inmetro, Brazil

Editor for Eastern Europe
Sachenko, Anatoly, Ternopil State Economic University, Ukraine

Editor for Asia
Ohyama, Shinji, Tokyo Institute of Technology, Japan

Editor for Africa
Maki K.Habib, American University in Cairo, Egypt

Editor for Asia-Pacific
Mukhopadhyay, Subhas, Massey University, New Zealand

Editorial Advisory Board

Abdul Rahim, Ruzairi, Universiti Teknologi, Malaysia
Ahmad, Mohd Noor, Nothern University of Engineering, Malaysia
Annamalai, Karthigeyan, National Institute of Advanced Industrial Science and Technology, Japan
Arcega, Francisco, University of Zaragoza, Spain
Arguel, Philippe, CNRS, France
Ahn, Jae-Pyoung, Korea Institute of Science and Technology, Korea
Arndt, Michael, Robert Bosch GmbH, Germany
Ascoli, Giorgio, George Mason University, USA
Atalay, Selcuk, Inonu University, Turkey
Atghiaee, Ahmad, University of Tehran, Iran
Augutis, Vygantas, Kaunas University of Technology, Lithuania
Avachit, Patil Lalchand, North Maharashtra University, India
Ayesh, Aladdin, De Montfort University, UK
Azamimi, Azian binti Abdullah, Universiti Malaysia Perlis, Malaysia
Bahreyni, Behraad, University of Manitoba, Canada
Baliga, Shankar, B., General Monitors Transnational, USA
Baoxian, Ye, Zhengzhou University, China
Barford, Lee, Agilent Laboratories, USA
Barlingay, Ravindra, RF Arrays Systems, India
Basu, Sukumar, Jadavpur University, India
Beck, Stephen, University of Sheffield, UK
Ben Bouzid, Sihem, Institut National de Recherche Scientifique, Tunisia
Benachaiba, Chellali, Universitaire de Bechar, Algeria
Binnie, T. David, Napier University, UK
Bischoff, Gerlinde, Inst. Analytical Chemistry, Germany
Bodas, Dhananjay, IMTEK, Germany
Borges Carval, Nuno, Universidade de Aveiro, Portugal
Bouchikhi, Benachir, University Moulay Ismail, Morocco
Bousbia-Salah, Mounir, University of Annaba, Algeria
Bouvet, Marcel, CNRS – UPMC, France
Brudzewski, Kazimierz, Warsaw University of Technology, Poland
Cai, Chenxin, Nanjing Normal University, China
Cai, Qingyun, Hunan University, China
Calvo-Gallego, Jaime, Universidad de Salamanca, Spain
Campanella, Luigi, University La Sapienza, Italy
Carvalho, Vitor, Minho University, Portugal
Cecelja, Franjo, Brunel University, London, UK
Cerda Belmonte, Judith, Imperial College London, UK
Chakrabarty, Chandan Kumar, Universiti Tenaga Nasional, Malaysia
Chakravorty, Dipankar, Association for the Cultivation of Science, India
Changhai, Ru, Harbin Engineering University, China
Chaudhari, Gajanan, Shri Shivaji Science College, India
Chavali, Murthy, N.I. Center for Higher Education, (N.I. University), India
Chen, Jiming, Zhejiang University, China
Chen, Rongshun, National Tsing Hua University, Taiwan
Cheng, Kuo-Sheng, National Cheng Kung University, Taiwan
Chiang, Jeffrey (Cheng-Ta), Industrial Technol. Research Institute, Taiwan
Chiriac, Horia, National Institute of Research and Development, Romania
Chowdhuri, Arijit, University of Delhi, India
Chung, Wen-Yaw, Chung Yuan Christian University, Taiwan
Corres, Jesus, Universidad Publica de Navarra, Spain
Cortes, Camilo A., Universidad Nacional de Colombia, Colombia
Courtois, Christian, Universite de Valenciennes, France
Cusano, Andrea, University of Sannio, Italy
D'Amico, Arnaldo, Università di Tor Vergata, Italy
De Stefano, Luca, Institute for Microelectronics and Microsystem, Italy
Deshmukh, Kiran, Shri Shivaji Mahavidyalaya, Barshi, India
Dickert, Franz L., Vienna University, Austria
Dieguez, Angel, University of Barcelona, Spain
Dighavkar, C. G., M.G. Vidyamandir's L. V.H. College, India
Dimitropoulos, Panos, University of Thessaly, Greece
Ding, Jianning, Jiangsu Polytechnic University, China
Djordjevich, Alexandar, City University of Hong Kong, Hong Kong
Donato, Nicola, University of Messina, Italy
Donato, Patricio, Universidad de Mar del Plata, Argentina
Dong, Feng, Tianjin University, China
Drljaca, Predrag, Instersema Sensoric SA, Switzerland
Dubey, Venketesh, Bournemouth University, UK
Enderle, Stefan, Univ.of Ulm and KTB Mechatronics GmbH, Germany
Erdem, Gursan K. Arzum, Ege University, Turkey
Erkmen, Aydan M., Middle East Technical University, Turkey
Estelle, Patrice, Insa Rennes, France
Estrada, Horacio, University of North Carolina, USA
Faiz, Adil, INSA Lyon, France
Fericean, Sorin, Balluff GmbH, Germany
Fernandes, Joana M., University of Porto, Portugal
Francioso, Luca, CNR-IMM Institute for Microelectronics and Microsystems, Italy
Francis, Laurent, University Catholique de Louvain, Belgium
Fu, Weiling, South-Western Hospital, Chongqing, China
Gaura, Elena, Coventry University, UK
Geng, Yanfeng, China University of Petroleum, China
Gole, James, Georgia Institute of Technology, USA
Gong, Hao, National University of Singapore, Singapore
Gonzalez de la Rosa, Juan Jose, University of Cadiz, Spain
Granel, Annette, Goteborg University, Sweden
Graff, Mason, The University of Texas at Arlington, USA
Guan, Shan, Eastman Kodak, USA
Guillet, Bruno, University of Caen, France
Guo, Zhen, New Jersey Institute of Technology, USA
Gupta, Narendra Kumar, Napier University, UK
Hadjiloucas, Sillas, The University of Reading, UK
Haider, Mohammad R., Sonoma State University, USA
Hashsham, Syed, Michigan State University, USA
Hasni, Abdelhafid, Bechar University, Algeria
Hernandez, Alvaro, University of Alcala, Spain
Hernandez, Wilmar, Universidad Politecnica de Madrid, Spain
Homentcovschi, Dorel, SUNY Binghamton, USA
Horstman, Tom, U.S. Automation Group, LLC, USA
Hsiai, Tzung (John), University of Southern California, USA
Huang, Jeng-Sheng, Chung Yuan Christian University, Taiwan
Huang, Star, National Tsing Hua University, Taiwan
Huang, Wei, PSG Design Center, USA
Hui, David, University of New Orleans, USA
Jaffrezic-Renault, Nicole, Ecole Centrale de Lyon, France
James, Daniel, Griffith University, Australia
Janting, Jakob, DELTA Danish Electronics, Denmark
Jiang, Liudi, University of Southampton, UK
Jiang, Wei, University of Virginia, USA
Jiao, Zheng, Shanghai University, China
John, Joachim, IMEC, Belgium
Kalach, Andrew, Voronezh Institute of Ministry of Interior, Russia
Kang, Moonho, Sunmoon University, Korea South
Kaniusas, Eugenijus, Vienna University of Technology, Austria
Katake, Anup, Texas A&M University, USA
Kausel, Wilfried, University of Music, Vienna, Austria
Kavasoglu, Nese, Mugla University, Turkey
Ke, Cathy, Tyndall National Institute, Ireland
Khelfaoui, Rachid, Université de Bechar, Algeria
Khan, Asif, Aligarh Muslim University, Aligarh, India
Kim, Min Young, Kyungpook National University, Korea South
Ko, Sang Choon, Electronics. and Telecom. Research Inst., Korea South
Kotulska, Malgorzata, Wroclaw University of Technology, Poland
Kockar, Hakan, Balikesir University, Turkey
Kong, Ing, RMIT University, Australia
Kratz, Henrik, Uppsala University, Sweden

Krishnamoorthy, Ganesh, University of Texas at Austin, USA
Kumar, Arun, University of Delaware, Newark, USA
Kumar, Subodh, National Physical Laboratory, India
Kung, Chih-Hsien, Chang-Jung Christian University, Taiwan
Lacnjevac, Caslav, University of Belgrade, Serbia
Lay-Ekuakille, Aime, University of Lecce, Italy
Lee, Jang Myung, Pusan National University, Korea South
Lee, Jun Su, Amkor Technology, Inc. South Korea
Lei, Hua, National Starch and Chemical Company, USA
Li, Fengyuan (Thomas), Purdue University, USA
Li, Genxi, Nanjing University, China
Li, Hui, Shanghai Jiaotong University, China
Li, Sihua, Agiltron, Inc., USA
Li, Xian-Fang, Central South University, China
Li, Yuefa, Wayne State University, USA
Liang, Yuanchang, University of Washington, USA
Liawruangrath, Saisunee, Chiang Mai University, Thailand
Liew, Kim Meow, City University of Hong Kong, Hong Kong
Lin, Hermann, National Kaohsiung University, Taiwan
Lin, Paul, Cleveland State University, USA
Linderholm, Pontus, EPFL - Microsystems Laboratory, Switzerland
Liu, Aihua, University of Oklahoma, USA
Liu Changgeng, Louisiana State University, USA
Liu, Cheng-Hsien, National Tsing Hua University, Taiwan
Liu, Songqin, Southeast University, China
Lodeiro, Carlos, University of Vigo, Spain
Lorenzo, Maria Encarnacio, Universidad Autonoma de Madrid, Spain
Lukaszewicz, Jerzy Pawel, Nicholas Copernicus University, Poland
Ma, Zhanfang, Northeast Normal University, China
Majstorovic, Vidosav, University of Belgrade, Serbia
Malyshev, V.V., National Research Centre ‘Kurchatov Institute’, Russia
Marquez, Alfredo, Centro de Investigacion en Materiales Avanzados, Mexico
Matay, Ladislav, Slovak Academy of Sciences, Slovakia
Mathur, Prafull, National Physical Laboratory, India
Maurya, D.K., Institute of Materials Research and Engineering, Singapore
Mekid, Samir, University of Manchester, UK
Melnyk, Ivan, Photon Control Inc., Canada
Mendes, Paulo, University of Minho, Portugal
Mennell, Julie, Northumbria University, UK
Mi, Bin, Boston Scientific Corporation, USA
Minas, Graca, University of Minho, Portugal
Mishra, Vivekanand, National Institute of Technology, India
Moghavvemi, Mahmoud, University of Malaya, Malaysia
Mohammadi, Mohammad-Reza, University of Cambridge, UK
Molina Flores, Esteban, Benemérita Universidad Autónoma de Puebla, Mexico
Moradi, Majid, University of Kerman, Iran
Morello, Rosario, University "Mediterranea" of Reggio Calabria, Italy
Mounir, Ben Ali, University of Sousse, Tunisia
Mrad, Nezih, Defence R&D, Canada
Mulla, Imtiaz Sirajuddin, National Chemical Laboratory, Pune, India
Nabok, Aleksey, Sheffield Hallam University, UK
Neelamegam, Periasamy, Sastra Deemed University, India
Neshkova, Milka, Bulgarian Academy of Sciences, Bulgaria
Oberhammer, Joachim, Royal Institute of Technology, Sweden
Ould Lahoucine, Cherif, University of Guelma, Algeria
Pamidighanta, Sayanu, Bharat Electronics Limited (BEL), India
Pan, Jisheng, Institute of Materials Research & Engineering, Singapore
Park, Joon-Shik, Korea Electronics Technology Institute, Korea South
Passaro, Vittorio M. N., Politecnico di Bari, Italy
Penza, Michele, ENEA C.R., Italy
Pereira, Jose Miguel, Instituto Politecnico de Setebal, Portugal
Petsev, Dimiter, University of New Mexico, USA
Pogacnik, Lea, University of Ljubljana, Slovenia
Post, Michael, National Research Council, Canada
Prance, Robert, University of Sussex, UK
Prasad, Ambika, Gulbarga University, India
Prateepasen, Asa, Kingmoungut's University of Technology, Thailand
Pugno, Nicola M., Politecnico di Torino, Italy
Pullini, Daniele, Centro Ricerche FIAT, Italy
Pumera, Martin, National Institute for Materials Science, Japan
Radhakrishnan, S. National Chemical Laboratory, Pune, India
Rajanna, K., Indian Institute of Science, India
Ramadan, Qasem, Institute of Microelectronics, Singapore
Rao, Basuthkar, Tata Inst. of Fundamental Research, India
Raoof, Kosai, Joseph Fourier University of Grenoble, France
Rastogi Shiva, K. University of Idaho, USA
Reig, Candid, University of Valencia, Spain
Restivo, Maria Teresa, University of Porto, Portugal
Robert, Michel, University Henri Poincare, France
Rezazadeh, Ghader, Urmia University, Iran
Royo, Santiago, Universitat Politecnica de Catalunya, Spain
Rodriguez, Angel, Universidad Politecnica de Cataluna, Spain
Rothberg, Steve, Loughborough University, UK
Sadana, Ajit, University of Mississippi, USA
Sadeghian Marnani, Hamed, TU Delft, The Netherlands
Sapozhnikova, Ksenia, D.I.Mendeleyev Institute for Metrology, Russia
Sandacci, Serghei, Sensor Technology Ltd., UK
Saxena, Vibha, Bhbha Atomic Research Centre, Mumbai, India
Schneider, John K., Ultra-Scan Corporation, USA
Sengupta, Deepak, Advance Bio-Photonics, India
Seif, Selemani, Alabama A & M University, USA
Seifter, Achim, Los Alamos National Laboratory, USA
Shah, Kriyang, La Trobe University, Australia
Sankarraj, Anand, Detector Electronics Corp., USA
Silva Girao, Pedro, Technical University of Lisbon, Portugal
Singh, V. R., National Physical Laboratory, India
Slomovitz, Daniel, UTE, Uruguay
Smith, Martin, Open University, UK
Soleimanpour, Amir Masoud, University of Toledo, USA
Soleymanpour, Ahmad, University of Toledo, USA
Somani, Prakash R., Centre for Materials for Electronics Technol., India
Sridharan, M., Sastra University, India
Srinivas, Talabattula, Indian Institute of Science, Bangalore, India
Srivastava, Arvind K., NanoSonix Inc., USA
Stefan-van Staden, Raluca-Ioana, University of Pretoria, South Africa
Stefanescu, Dan Mihai, Romanian Measurement Society, Romania
Sumriddetchka, Sarun, National Electronics and Comp. Technol. Center, Thailand
Sun, Chengliang, Polytechnic University, Hong-Kong
Sun, Dongming, Jilin University, China
Sun, Junhua, Beijing University of Aeronautics and Astronautics, China
Sun, Zhiqiang, Central South University, China
Suri, C. Raman, Institute of Microbial Technology, India
Sysoev, Victor, Saratov State Technical University, Russia
Szewczyk, Roman, Industr. Research Inst. for Automation and Measurement, Poland
Tan, Ooi Kiang, Nanyang Technological University, Singapore,
Tang, Dianping, Southwest University, China
Tang, Jaw-Luen, National Chung Cheng University, Taiwan
Teker, Kasif, Frostburg State University, USA
Thirunavukkarasu, I., Manipal University Karnataka, India
Thumbavanam Pad, Kartik, Carnegie Mellon University, USA
Tian, Gui Yun, University of Newcastle, UK
Tsiantos, Vassilios, Technological Educational Institute of Kaval, Greece
Tsigara, Anna, National Hellenic Research Foundation, Greece
Twomey, Karen, University College Cork, Ireland
Valente, Antonio, University, Vila Real, - U.T.A.D., Portugal
Vanga, Raghav Rao, Summit Technology Services, Inc., USA
Vaseashta, Ashok, Marshall University, USA
Vazquez, Carmen, Carlos III University in Madrid, Spain
Vieira, Manuela, Instituto Superior de Engenharia de Lisboa, Portugal
Vigna, Benedetto, STMicroelectronics, Italy
Vrba, Radimir, Brno University of Technology, Czech Republic
Wandelt, Barbara, Technical University of Lodz, Poland
Wang, Jiangping, Xi'an Shiyou University, China
Wang, Kedong, Beihang University, China
Wang, Liang, Pacific Northwest National Laboratory, USA
Wang, Mi, University of Leeds, UK
Wang, Shinn-Fwu, Ching Yun University, Taiwan
Wang, Wei-Chih, University of Washington, USA
Wang, Wensheng, University of Pennsylvania, USA
Watson, Steven, Center for NanoSpace Technologies Inc., USA
Weiping, Yan, Dalian University of Technology, China
Wells, Stephen, Southern Company Services, USA
Wolkenberg, Andrzej, Institute of Electron Technology, Poland
Woods, R. Clive, Louisiana State University, USA
Wu, DerHo, National Pingtung Univ. of Science and Technology, Taiwan
Wu, Zhaoyang, Hunan University, China
Xiu Tao, Ge, Chuzhou University, China
Xu, Lisheng, The Chinese University of Hong Kong, Hong Kong
Xu, Sen, Drexel University, USA
Xu, Tao, University of California, Irvine, USA
Yang, Dongfang, National Research Council, Canada
Yang, Shuang-Hua, Loughborough University, UK
Yang, Wuqiang, The University of Manchester, UK
Yang, Xiaoling, University of Georgia, Athens, GA, USA
Yaping Dan, Harvard University, USA
Ymeti, Aurel, University of Twente, Netherland
Yong Zhao, Northeastern University, China
Yu, Haihu, Wuhan University of Technology, China
Yuan, Yong, Massey University, New Zealand
Yufera Garcia, Alberto, Seville University, Spain
Zakaria, Zulkarnay, University Malaysia Perlis, Malaysia
Zagnoni, Michele, University of Southampton, UK
Zamani, Cyrus, Universitat de Barcelona, Spain
Zeni, Luigi, Second University of Naples, Italy
Zhang, Minglong, Shanghai University, China
Zhang, Qintao, University of California at Berkeley, USA
Zhang, Weiping, Shanghai Jiao Tong University, China
Zhang, Wenming, Shanghai Jiao Tong University, China
Zhang, Xueji, World Precision Instruments, Inc., USA
Zhong, Haoxiang, Henan Normal University, China
Zhu, Qing, Fujifilm Dimatix, Inc., USA
Zorzano, Luis, Universidad de La Rioja, Spain
Zourob, Mohammed, University of Cambridge, UK

Sensors & Transducers Journal (ISSN 1726-5479) is a peer review international journal published monthly online by International Frequency Sensor Association (IFSA).
Available in electronic and on CD. Copyright © 2012 by International Frequency Sensor Association. All rights reserved.

Sensors & Transducers Journal

Contents

Volume 142
Issue 7
July 2012

www.sensorsportal.com

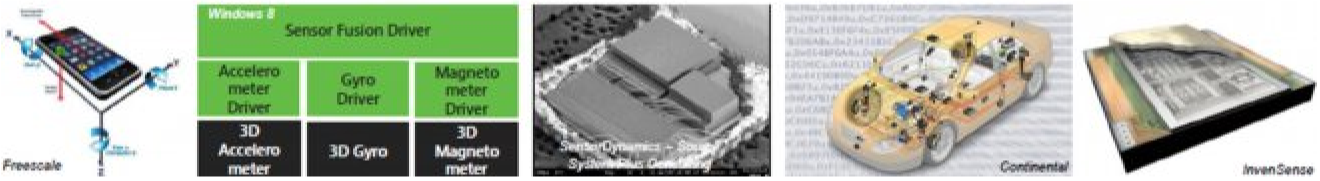
ISSN 1726-5479

Editorial Book Review

Research Articles

Authors are encouraged to submit article in MS Word (doc) and Acrobat (pdf) formats by e-mail: editor@sensorsportal.com
Please visit journal’s webpage with preparation instructions: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm

International Frequency Sensor Association (IFSA).



Sensors & Transducers Journal, Vol. 142, Issue 7, July 2012, pp. 54-60

Sensors & Transducers

ISSN 1726-5479

http://www.sensorsportal.com

Dynamic Analysis of Wireless Sensor Network Programs Based on Binary Tool and Simulator

[1,2] Yuxia Sun, [3] Yong Tang
[1] Department of Computer Science, Sun Yat-sen University, Guangzhou, 510006, China
[2] Department of Computer Science, Jinan University, Guangzhou, 510632, China
Tel.: ++86-20-85228372
[3] College of Computing, South China Normal University, Guangzhou, 510631, China
Tel.: ++86-20-85211353
E-mail: tyxsun@jnu.edu.cn, ytang@scnu.edu.cn

Received: 9 June 2012 /Accepted: 23 July 2012 /Published: 31 July 2012

Abstract: Software quality of sensor network programs has become one of the major concerns during the adoption of WSN technology. Program analysis plays an important role in the software quality assurance. We present a general approach for dynamic program analysis of TinyOS programs before deployment. The approach first utilizes binary dump tool to obtain the disassemble code for the target program under analysis, and then analyzes the dumped file to create associations between concerned instructions and their corresponding memory addresses. Based on those associations, the approach executes the program and analyzes the interested program behaviors at instruction level by utilizing the simulation and analysis infrastructure of Avrora. The examples illustrated in this paper have been implemented in our experiments. *Copyright © 2012 IFSA.*

Keywords: WSN, Program analysis, TinyOS programs, Dump tools, Avrora.

---

1. Introduction

Wireless Sensor Networks (WSNs) attract increasing interest in a number of domains concerned with monitoring of physical environment. However, application development is still one of the main hurdles to a wide adoption of WSN technology [1]. WSN applications are error prone and many software bugs have been reported [2, 3]. It is often difficult or even impossible to debug and fix WSN applications after deployment, for instance, when the sensors are deployed in human-inaccessible places. Therefore, it is very important to develop pre-deployment program analysis and testing techniques for WSN programs.

The de facto standard programming language for sensor nodes is nesC [4] and the underlying operating system is TinyOS [5]. The nesC tool chain (e.g., make system) pre-processes a TinyOS application into a single C file, and then compiles the C code into the target machine code [6]. In this paper, we focus on the analysis of WSN programs over TinyOS operating system and AVR [7] target platform.

To discover WSN program problems before deploying it onto the target sensor node, we can utilize WSN simulator to run the program and analyze its behaviors. Avrora is a WSN simulator with cycle accurate execution times. Avrora provides a framework for simulating and analyzing assembly codes written for AVR microcontrollers on sensor nodes [8, 9]. The framework allows users to add monitoring code to the simulation to collect information on program execution, which information can be used for program analysis. Because the collection is at instruction level, the information can be very precise and detailed. Although Avrora has provided a few profiling utilities that can be directly used to study program behaviors, more profiling tools have to be developed to achieve various profiling goals.

This paper presents a general program analysis approach for TinyOS applications. The approach first identifies the features of user-concerned instructions in the disassemble code. Next, based on the identified features, the approach observes and analyzes the instructions' executions over Avrora. In the first phase, the approach uses GNU Binutils [10] package to obtain the disassemble information of object files, which are performed over AVR target platform. The package contains such binary tools as avr-objdump for manipulating object files.

The rest of this paper is organized as follows: Section 2 outlines the dynamic analysis approach. Section 3 illustrates how to analyze dumped files. Section 4 shows how to investigate the execution of the program under analysis. The final section concludes our argument.

2. Outline of Approach

We propose a general program analysis approach based on the binary dump tool, avr-objdump [11], and the instruction-level simulator, Avrora. By utilizing avr-objdump, we can generate the assembler mnemonics for the machine instructions from the object file of a TinyOS program. By utilizing Avrora, we can run the assembler code and analyze the program behaviors at instruction-level. The approach consists of three steps as below.

(1) Use AVR-objdump tool to obtain the disassemble code for the target program under analysis.

The target binary code of the TinyOS program under analysis can be generated using nesC make system or using avr-gcc. The binary code is executable on the AVR target platform including the simulated ones. However, binary code is difficult for us to understand and to analyze. To inspect the contents of binary-code files, disassemblers can be invaluable. The simulator Avrora allows us to run and analyze the object program with disassembly information. Therefore, the first step of our program analysis is to generate the disassemble code from the target binary code of the TinyOS program under analysis. Because we focus on the AVR target platform, we utilize the binary tool, avr-objdump, to achieve this goal.

The file extension of the standard ELF object file format is .elf. The avr-objdump tool can be applied to ELF files. Assume that the object file is demo.elf, we can use the avr-objdump tool with the option –D to generate a file named demo.od. The file demo.od is executable over Avrora and contains the disassembly information dumped from the object file demo.elf.

(2) Analyze the dumped file (.od file) to identify the user-concerned instructions, and then create association between those instructions and their corresponding memory addresses.

Because a user is always concerned with multiple types of instructions, the identified information for each instruction involves both instruction type and such particular instruction information as instruction name.

(3) Investigate the execution of the dumped file.

Inspect the execution of the concerned instructions to perform program analysis, by utilizing Avrora instrumentation framework (API).

Next, we will present the approach with details and examples. Because first step above is easy to implement, we will dwell on the last two steps.

3. Analyze Dumped File

3.1. Identify User-concerned Instructions

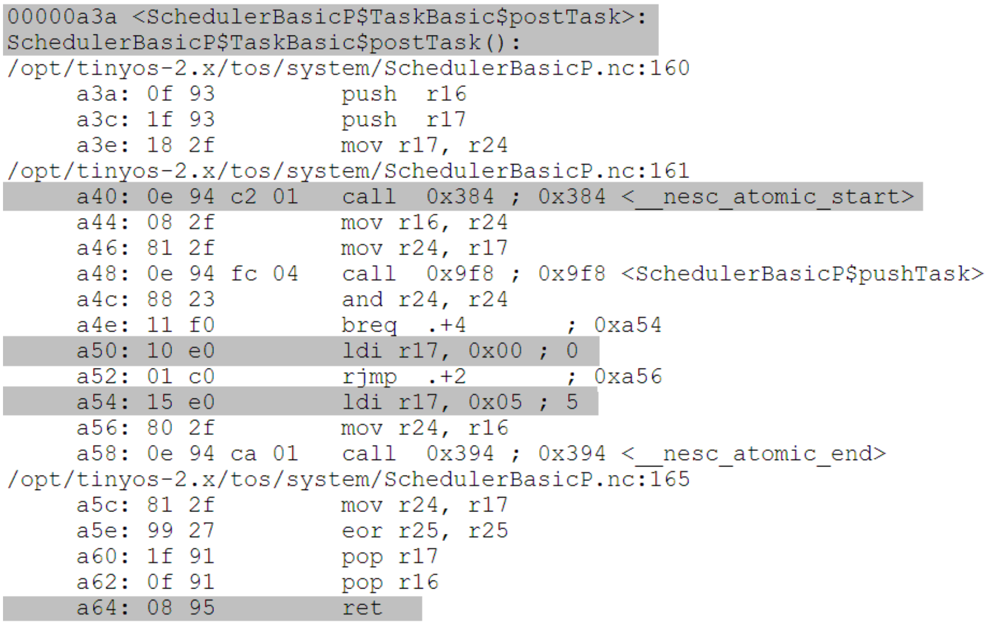
As in a standard ELF object file, in a dumped .od file, .text section contains the machine code of the compiled program. To identify all the user-concerned instructions, we need scan the whole .text section.

In the .text section of the dumped file, instructions are grouped in functions or in interrupt handlers. Because the code segments of two functions (or interrupt handlers) can never be overlapped, the segment of a function (or an interrupt handler) can be identified by finding its initial line and its final line. The lines mentioned above have the following characteristics, which enable us to make them out when scanning the dumped file.

(1) The initial line of a function (or an interrupt handler) starts with *00* and ends with *a pair of angle brackets*. The function name (or the interrupt number) is available both in the brackets and in the second line of the function (or the interrupt handler). For example, Fig. 1 illustrates the disassembly segment of the TinyOS postTask function in a dumped file (e.g. demo.od). The complete function name is *SchedulerBasicP.TaskBasic.postTask(),* where *SchedulerBasicP*, *TaskBasic* and *postTask* are TinyOS system component name, interface name and function name, respectively. Recall that when a nesC function is compiled to a C function, a function name and its corresponding component name and interface name are separated by the *$* character, as shown in the first line in Fig. 1. The OS postTask function attempts to put (i.e., post) a task into the system task queue. Fig. 2 shows the nesC source code of the OS postTask function.

(2) The last line of a function (or an interrupt handler) is actually its last instruction. For a function, the last instruction is *ret*, as the last line shows in Fig. 1. For an interrupt handler, the last instruction is *reti*.

In the .text section, each line containing machine code represents an instruction's information. Each such line is in the form of '*address: machine_instruction assembly_instruction ; comment*', where *address* is the memory address of the instruction, *machine_instruction* is the machine code in the form of a sequence of bytes, *assembly_instruction* is the text representation of the instruction generated by the disassemble tool (i.e., avr-objdump with option -D), and comment makes the disassemble code even more human-readable.



```
00000a3a <SchedulerBasicP$TaskBasic$postTask>:
SchedulerBasicP$TaskBasic$postTask():
/opt/tinyos-2.x/tos/system/SchedulerBasicP.nc:160
     a3a: 0f 93          push  r16
     a3c: 1f 93          push  r17
     a3e: 18 2f          mov r17, r24
/opt/tinyos-2.x/tos/system/SchedulerBasicP.nc:161
     a40: 0e 94 c2 01    call  0x384 ; 0x384 <__nesc_atomic_start>
     a44: 08 2f          mov r16, r24
     a46: 81 2f          mov r24, r17
     a48: 0e 94 fc 04    call  0x9f8 ; 0x9f8 <SchedulerBasicP$pushTask>
     a4c: 88 23          and r24, r24
     a4e: 11 f0          breq  .+4        ; 0xa54
     a50: 10 e0          ldi r17, 0x00 ; 0
     a52: 01 c0          rjmp  .+2        ; 0xa56
     a54: 15 e0          ldi r17, 0x05 ; 5
     a56: 80 2f          mov r24, r16
     a58: 0e 94 ca 01    call  0x394 ; 0x394 <__nesc_atomic_end>
/opt/tinyos-2.x/tos/system/SchedulerBasicP.nc:165
     a5c: 81 2f          mov r24, r17
     a5e: 99 27          eor r25, r25
     a60: 1f 91          pop r17
     a62: 0f 91          pop r16
     a64: 08 95          ret
```

Fig. 1. Disassembly code of TinyOS postTask function.

```
/*Return SUCCESS (0) if the post succeeded, EBUSY (5) if it was already posted. */
async command error_t TaskBasic.postTask[uint8_t id]() {
atomic {
return pushTask(id) ? SUCCESS : EBUSY;
}
}
```

Fig. 2. NesC source code of TinyOS postTask function.

The given goal of a user's program-analysis determines which instructions are concerned by the user. Therefore, the user-concerned instructions are different in various program-analyses. Sometimes, it is easy to identify the instructions, while in other cases, it is not.

(1) Some instructions are easy to identify because the disassembly information associates the instruction to its corresponding source explicitly. For example, in the third highlighted line of Fig. 1, the disassembled text string is '*call 0x384; 0x384 <__nesc_atomic_start>*'. It is easy to parse this string and get the following results: the instruction mentioned above is a call instruction and the invoked function's name is *__nesc_atomic_start()*.

(2) However, not all the disassembly information can directly associate an instruction in the object file to its exact source, especially when a source statement corresponds to multiple lines of object instructions. In this case, program-analysis requires more in-depth investigation of both the source code and the disassembly information. Next, we will illustrate it with an example.

Recall that a TinyOS program posts a task via a *post* statement. After compilation, the statement is converted into a C function *taskName$postTask()*, which calls the OS postTask function mentioned above. During program analysis, a user may concern whether or not a task-posting is successful. However, we cannot always get such information by tracing the return value of the function *taskName$postTask()*, because it is also valid to use *post()* without a return value. Our solution is to identify the key instructions in the OS postTask function and trace their executions, which instructions indicate a successful posting or a failed one. As Fig. 2 shows, OS postTask function attempts to push the task id to OS task queue by calling *pushTask()*: the return value of the invocation is *SUCCESS* (i.e., *0*) if the posting is successful, and *BUSY* (i.e., *5*) otherwise. By studying the disassembly code of the OS postTask function, we found the key instructions that can distinguish those two cases, namely the fourth and the fifth highlighted lines, respectively, shown in Fig. 1. Both key instructions above have fixed relative offsets in the OS postTask function. As shown in Fig. 1, within the OS postTask function, the relative offset of the instruction that denotes a successful posting is *0x16*, and the relative offset of the one that denotes a failed posting is *0x1a*. Therefore, based on identifying the entry of the OS postTask function, we can identify the key instructions that indicate whether or not a posting is successful.

3.2. Create Instruction-Address Association

When a user-concerned instruction is identified, the association between the instruction and its memory address need to be established. During the dynamic program-analysis, the program is executed instruction by instruction, and we monitor each instruction's execution. By utilizing the created association mentioned above, we can retrieve the current instruction's information from the current instruction's address. Because a user is always concerned with multiple types of instructions, the instruction's information retrieved contains both instruction type and particular information such as instruction name. For example, in Fig. 1, line 4 is a function-entry instruction, the instruction's information always involves the instruction type (e.g., *FuncEntry*) and the function name (i.e., *SchedulerBasicP.TaskBasic.postTask*).

To implement association between the concerned instructions and their corresponding addresses, a direct way is to utilize a list structure as below: to represent each concerned instruction, create a structure-type which consists of several fields for denoting instruction information and one field for instruction address; create a list-type whose elements are of this structure-type. During the identification process, as presented in Section 3.1, we put each identified instruction's information and address to the above list. To retrieve the instruction's information from an instruction's address, we need search the list mentioned above. This way to create and use associations is traditional and direct.

A more concise and more efficient way to the implementation is to utilize the *map* type. In a map, the *key-value* is generally used to uniquely identify the element, where the *mapped value* is some sort of value associated to this *key*. Therefore, the association between an instruction's address and its information can be created by mapping the instruction information (i.e., value) to the instruction address (i.e., key). Many popular programming languages, such as Java and C++, support the map type [12, 13]. For instance, in Java, we can utilize the class of *java.util.HashMap* for the implementation: use the method *put*(*instructionAdr, instructionInfo*) to associate the specified instruction information with the specified instruction address in a map, and use the method *get*(*instructionAdr*) to retrieve the instruction information to which the specified address is mapped.

4. Investigate the Execution

By utilizing the simulator Avrora, we can run a TinyOS program and investigate the execution at instruction-level. Although Avrora has provided several profiling options for basic profiling purpose, in most cases, we need develop our own tool for various program analyses, which are always complicated due to the particular user concerns. To serve this role, we need utilize the Avrora infrastructure [9] to develop our own tool to monitor and profile the execution of the program. The development of such a tool consists of two steps:

(1) Create a monitor and add it to the Avrora simulation. With the infrastructure of Avrora [9], we can write a subclass of the *avrora.monitors.MonitorFactory* class and Avrora can attach instrumentations to the Monitor.

(2) Add instrumentation to the program to collect user-concerned information during the simulation run and generate a resultant report. We can utilize such Avrora mechanisms as probes and watches to implement the instrumentation: Probes are used to observe specific instructions when they are fired; Watches are used to monitor specific data memory-addresses when they are accessed [9].

Suppose that we want to trace the execution of some types of instructions (e.g., function entry, function exit, successful posting and failed posting) and extract some concerned information from the instruction (e.g., the function name, the task name, and the instruction type), we can override the method *fireBefore()* or *fireAfter()* of the interface *Probe* exposed by Avrora. We can add a probe before (or after) the concerned types of instructions. The probe will fire before (or after) this instruction's execution and collect the concerned information from the instruction. If we want to trace the executions of several types of instructions at the same time, we can also insert a probe before (or after) every instruction, and perform the analysis as below:

(1) Get the instruction's type by utilizing the mapping from the instruction's address, which mapping has been created during the analysis of the dumped file, as described in Section 3.

(2) If the instruction type is not user-interested, we do nothing; otherwise, we get the concerned instruction information from the instruction's address, also by utilizing the previously-created mapping, as presented in Section 3.

5. Conclusions

We present a general framework for dynamic analysis of TinyOS programs. Based on a binary dump tool, we get the disassembly code of the program under analysis. By analyzing the dumped code, we create the associations between the concerned instructions and their memory addresses. By utilizing the instruction-level simulator Avrora, we run the program and study the program behaviors. The observation and analysis of the program execution is based on the associations created previously. The framework presented in this paper is promising to implement many program-analysis tools for sensor network programs.

Acknowledgements

This research is supported by Major Program of NSFC (Grant #60736020), NSFC (Grant #60970044), Distinguished Youth Researcher Nurturing Program in Guangdong Universities (Grant #LYM08017), and Fundamental Research Funds for the Central Universities (Grant #51208035).

References

[1]. L. Mottola, G. Picco, Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art, *ACM Computing Surveys*, Vol. 43, Issue 3, 2011, pp. 19:1–19:51.

[2]. Barrenetxea G., Ingelrest F., Schaefer G., Vetterli M., The Hitchhiker's Guide to Successful Wireless Sensor Network Deployments. SenSys, in *Proceedings of the 6th ACM Conf. on Embedded Network Sensor Systems (SenSys'08),* New York, NY, USA, 2008, pp. 43–56.

[3]. Werner-Allen G., Lorincz K., Johnson J., Lees J., and Welsh M. Fidelity and Yield in a Volcano Monitoring Sensor Network, in *Proceedings of 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI'06)*, Seattle, WA, November 2006, pp. 27-27.

[4]. D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, D. Culler. The nesC language: A holistic approach to networked embedded systems, in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'03)*, San Diego, June 2003, pp. 1-11.

[5]. P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, TinyOS: An operating system for wireless sensor networks, in Ambient Intelligence, *Springer-Verlag*, 2005.

[6]. Creating a New Platform for TinyOS 2.x (http://www.tinyos.net/tinyos-2.x/doc/html/tep131.html).

[7]. Atmel AVR (http://en.wikipedia.org/wiki/Atmel_AVR).

[8]. B. L. Titzer, D. K. Lee, J. Palsberg. Avrora: Scalable Sensor Network Simulation with Precise Timing, in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN'05)*, Los Angeles, California, April 2005, pp. 477-482.

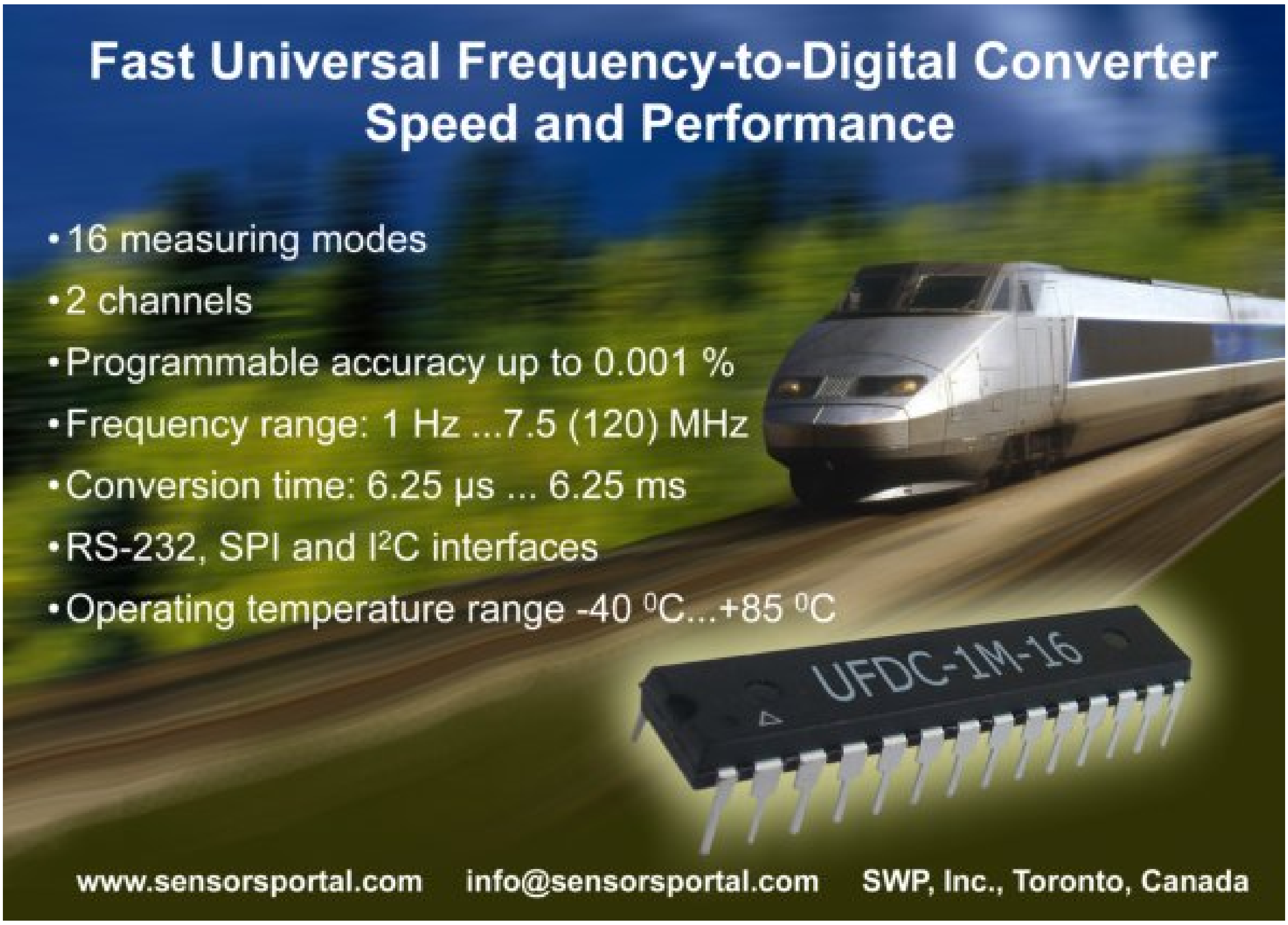
[9]. Avrora: the AVR Simulation and Analysis Framework (http://compilers.cs.ucla.edu/avrora/).

[10]. GNU Binutils for the AVR target (http://www.nongnu.org/avr-libc/user-manual/overview.html).

[11]. Avr-objdump (http://ccrma.stanford.edu/planetccrma/man/man1/avr-objdump.1.html).

[12]. J. Gosling, B. Joy, G. Steele, G. Bracha, The Java Language Specification, 3rd Edition, *Addison-Wesley*, 2005.

[13]. Reference of the C++ Language Library (http://www.cplusplus.com/reference/).



Sensors & Transducers Journal

Guide for Contributors

Aims and Scope

Sensors & Transducers Journal (ISSN 1726-5479) provides an advanced forum for the science and technology of physical, chemical sensors and biosensors. It publishes state-of-the-art reviews, regular research and application specific papers, short notes, letters to Editor and sensors related books reviews as well as academic, practical and commercial information of interest to its readership. Because of it is a peer reviewed international journal, papers rapidly published in *Sensors & Transducers Journal* will receive a very high publicity. The journal is published monthly as twelve issues per year by International Frequency Sensor Association (IFSA). In additional, some special sponsored and conference issues published annually. *Sensors & Transducers Journal* is indexed and abstracted very quickly by Chemical Abstracts, IndexCopernicus Journals Master List, Open J-Gate, Google Scholar, etc. Since 2011 the journal is covered and indexed (including a Scopus, Embase, Engineering Village and Reaxys) in Elsevier products.

Topics Covered

Contributions are invited on all aspects of research, development and application of the science and technology of sensors, transducers and sensor instrumentations. Topics include, but are not restricted to:

- Physical, chemical and biosensors;
- Digital, frequency, period, duty-cycle, time interval, PWM, pulse number output sensors and transducers;
- Theory, principles, effects, design, standardization and modeling;
- Smart sensors and systems;
- Sensor instrumentation;
- Virtual instruments;
- Sensors interfaces, buses and networks;
- Signal processing;
- Frequency (period, duty-cycle)-to-digital converters, ADC;
- Technologies and materials;
- Nanosensors;
- Microsystems;
- Applications.

Submission of papers

Articles should be written in English. Authors are invited to submit by e-mail editor@sensorsportal.com 8-14 pages article (including abstract, illustrations (color or grayscale), photos and references) in both: MS Word (doc) and Acrobat (pdf) formats. Detailed preparation instructions, paper example and template of manuscript are available from the journal's webpage: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm Authors must follow the instructions strictly when submitting their manuscripts.

Advertising Information

Advertising orders and enquires may be sent to sales@sensorsportal.com Please download also our media kit: http://www.sensorsportal.com/DOWNLOADS/Media_Kit_2012.pdf


Sergey Y. Yurish
Digital Sensors and Sensor Systems: Practical Design
IFSA
Digital Sensors and Sensor Systems: Practical Design
Sergey Y. Yurish