# IMPROVEMENT OF TINYOS IMPLEMENTATION FOR SMALL MEMORY FPGA SYSTEM

*Ieryung Park, Hosoon Shin, Jihan Park, Eungu Jung and Dongsoo Har*

Department of Information and Communications, Gwangju Institute of Science and Technology, Republic of Korea.
{pleastop, hosoon, iceflame, egjung, hardon}@gist.ac.kr

## ABSTRACT

We used FPGA for developing prototypes of novel sensor network system because of its short turn-around-time and flexibility. For FPGA-based sensor network system, we have to decrease the size of operating system because of small memory. Scalability is another important feature because if logic circuit in the FPGA is changed, the operating system also has to reflect this modification. In this paper, we discuss a way to implement TinyOS as an operating system in FPGA based sensor system with module-integration and discarding unused codes of TinyOS. This implementation can ensure scalability of TinyOS and drop down the code-size.

## 1. INTRODUCTION

Wireless sensor networks are composed of large numbers of tiny sensor systems which called mote; the sensor network node consists of a small and low-power micro controller unit (MCU), a radio frequency (RF) transceiver for wireless networking and several kinds of sensors.

FPGA based mote used for development of sensor network system because of FPGA's short turn-around-time and flexibility [1]. For FPGA-based motes, we have to decrease the size of TinyOS because FPGA chip in the mote doesn't have enough memory for application programs. Moreover, scalability is another important feature because if we change logic circuits in FPGA, the operating system also has to reflect changes of the system also.

First, we implemented all modules of TinyOS for our novel FPGA-based mote. Secondly, we integrated the modules which are related to RF and serial communication because they are too large for memory of FPGA and their hierarchy is very complex to configure TinyOS for continuous modification of behavior of FPGA-based MCU.

Finally, we discarded unused codes in RF and serial communication parts. Consequently, we could drop down the code size of the TinyOS and ensure scalability for configurable MCU.

## 2. RELATED WORKS

U.C. Berkeley developed a mote; mica as Smartdust project. Mica series [2] uses Atmega128L of Atmel, has 128KB ROM and they include CC1000 (Mica1, 2) or CC2420 (MicaZ) of Chipcon Inc. for RF communication. U.C. Berkeley also have developed novel motes; Telos series [3], based on MSP430F1611 of Texas Instruments. Telos series uses CC2420 of Chipcon Inc. as RF transceiver.

TinyOS [4]; component-based and event-driven operating system was developed by U.C. Berkelery in 2000. TinyOS is written in nesC; extended version of C language. Its module-typed kernel and libraries are compiled together with the applications, which are also written in nesC, to make a single image file for loading on target board.

C. Lynch and F. O'Reilly implemented TinyOS for PIC16F877-based mote [5]. They also developed a program which converts a intermediate codes of TinyOS for compiling TinyOS to their PIC-based mote.

## 3. SYSTEM ARCHITECTURE

A. Hardware Architecture

Our novel mote is based on Spartan3 XC3S400 of Xilinx Inc. It consists of 400 K gates, sixteen hardware multipliers and 36 KB block RAM. We implemented a core of a MCU which has compatibility with MSP430-core and several peripherals like 16-bit timer, synchronous peripheral interface (SPI) for communication with RF modem, UART for communication with RS232 chip using Verilog hardware description language on this FPGA. Other parts of mote consist of CC2420, RF transceiver of Chipcon Inc., RS232 chip, reset button, two user buttons and three LEDs. Fig. 1 shows the hardware architecture of the mote.

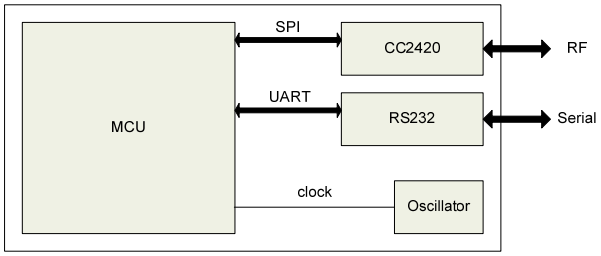


Fig. 1: Hardware Architecture.

Table1: Memory usages.

| Application | Hardware | Codes (byte) | Datas (byte) |
|---|---|---|---|
| Blink | MSP430 | 2582 | 40 |
| | FPGA | 2310 | 38 |
| CntToRfm | MSP430 | 11628 | 371 |
| | FPGA | 6002 | 44 |

B. Software Architecture

Most of TinyOS is written by nesC which is an advanced programming language of C language for component-based programming. Each module of TinyOS has a defined *interface* and other modules only interact with it using methods of the interface. Moreover, modules can be bound by *wiring*, which is an action that connects two modules through interfaces.

## 4. IMPLEMENTATION

TinyOS (Mica and Telos version) has modules for UART, SPI and RF transceiver for RF communication. There are unnecessary codes for our FPGA-based mote and these codes increase size of the image file. There is not enough memory for application because usually there is only 36 KB sized block RAM in FPGA.

Hence, we integrated these modules to one module, SimpleRF and discarded unnecessary codes of UART, SPI and RF transceiver for decreasing memory usage. These changes are shown in Fig. 2.

Memory usages for two simple applications are represented on Table1. *Blink* is an application that flashes a LED by timer interrupt and *CntToRfm* is an application that sends a packet through RF transceiver. The two memory usages for Blink application are almost the same. In CntToRfm application that uses integrated RF module, however, Code-usage is dropped down from 11628 bytes to 6002 bytes although our mote has compatibility with Telos which is based on MSP430-core.

Moreover, integration of these modules increases the retagetability for changing architecture of the MCU and sensor board because we can modify sources easily.

## 5. CONCLUSIONS

In this paper, we discussed a way to implement TinyOS for FPGA-based sensor system. We implemented TinyOS with module-integration and discarding unused codes. Thus, the improve implementation satisfied FPGA memory requirement and it can reflect logic circuit changes in FPGA easily because of scalable implementation also. Consequently, our implementation has smaller code-size and more scalability than original implementation.

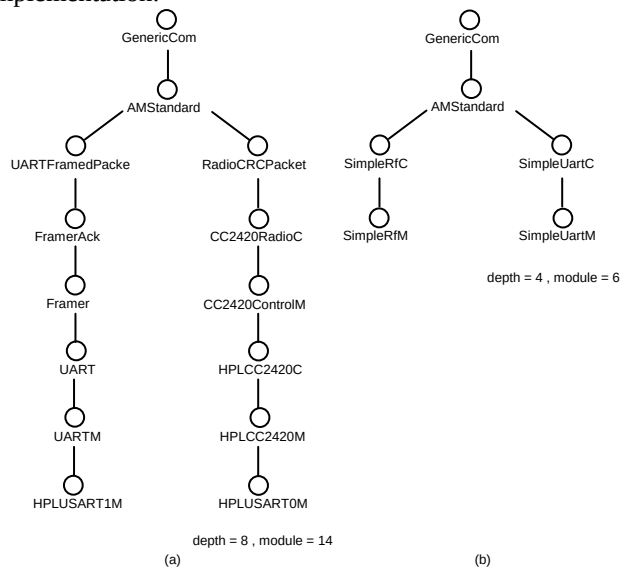


Fig. 2: Original implementation and our implementation.

## REFERENCES


[1] P. S. Zuchowski, C.B. Reynolds, R.J. Grupp, S.G. Davis, B. Cremen, B. Troxel, "A Hybrid ASIC and FPGA Architecture". Computer Aided Design, 2002. ICCAD 2002. IEEE/ACM International Conference on 10-14 Nov. 2002 Page(s):187-194, 2002.

[2] J. Hill and D. Culler, "Mica: a wireless platform for deeply embedded networks," IEEE Micro, vol. 22, no. 6, pp. 12-24, November/December 2002.

[3] J. Polastre, R. Szewczyk, and D. Culler, "Telos: emabling ultra-low power wireless research," Information Processing in Sensor Networks (IPSN'05), April 2005.

[4] 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. New York, NY:Springer-Verlag, To Appear.

[5] C. Lynch and F. O'Reilly, "PIC-based TinyOS implementation," Wireless Sensor Networks. Proceedings of the 2nd European Workshop on Sensor Networks, January/Febuary 2005.