ISAAC 2019
&
ISAAC 2019

PROGRAM

Advanced and Applied Convergence & Adavanced Culture Technology

7th International Symposium, ISAAC 2019 in conjunction with ICACT 2019

November 7 - 10 2019 Jeju, Korea
Revised Selected Papers

Hosted by :

IIBC

Co-hosted by :

IPACT

제주한라대학교 CHEJU HALLA UNIVERSITY

SMT 사단법인 한국모바일학회 Society of Mobile Technolgy

DONG NAI TECHNOLOGY UNIVERSITY

Co-sponsored by :

Ministry of Science and ICT

KOREA COMMUNICATIONS COMMISSION

KOFST

NRF National Research Foundation of Korea

KISA KOREA INTERNET & SECURITY AGENCY

IIBC The Institute of Internet, Broadcasting and Communication

IPACT The International Promotion Agency of Culture Technology

The International Symposium on Advanced and Applied Convergence (ISAAC 2019) & The International Conference on Advanced Culture Technology (ICACT 2019)

Hosted by	The Institute of Internet, Broadcasting and Communication (IIBC) & The International Promotion Agency of Culture Technology (IPACT)
Co-hosted by	Cheju Halla University The Society of Mobile Technology Dong Nai Technology University
Date	November 7~10, 2019
Venue	Halla Convention Center, Cheju Halla University, JeJu, Korea

[1st day] November 7 (Thursday)

Time	Program	Remarks
10:00 ~ 18:00	Preparation for Conference	Office
18:00 ~ 21:00	Committee Meeting for ISAAC 2019 & ICACT 2019 and Welcoming Dinner	Chairs

[2nd day] November 8 (Friday)

Time	Program Main Auditorium Halla Convention B Hall	Remarks
12:00 ~	**Registration Open**	Office
13:30 ~ 14:30	**International Conference (Poster)**	Session Chairs
14:30 ~ 15:50	**Oral – A**	Session Chairs
15:50 ~ 16:10	**Break Time**	

Time	Program Main Auditorium Halla Convention B Hall	Remarks
16:10 ~ 17:30	**Oral – B**	Session Chairs
17:30 ~ 18:00	**Welcoming and Awards Ceremony** • Opening Remarks: JeongJin Kang, President of IIBC • Welcoming Speech: Sung Hun Kim, President of Cheju Halla University • Congratulatory Speech: Heeryong Won, Governor of Jeju Island • Congratulatory Speech: Phan Ngoc Son, President of Dong Nai Technology University, Vietnam • Awards Ceremony • Group Photos	Session Chairs
18:00 ~ 20:00	• **Welcoming Conference Dinner (Banquet)**	Session Chair

[3rd day] November 9 (Saturday)

Time	Program Main Auditorium	Remarks
09:00 ~ 09:30	• Break with Snack and Drink	Session Chairs
09:30 ~ 12:00	• Round table on Scientific and Educational Collaboration with Cheju Halla University in Jeju	Session Chairs
12:00 ~ 13:00	**Lunch Time**	Office
13:00 ~ 17:00	• Industrial Field Tour • Culture Tour	Office

[4th day] November 10 (Sunday)

Time	Program (Seminar Room)	Remarks
09:00 ~ 09:30	• Break with Snack and Drink	Office
09:30 ~ 12:00	**Committee Meeting For ISAAC 2020 & ICACT 2020 Conference (Closed Meeting)**	Chairs

Good Code Style of Low Power Consumption for ECO Environment of Smart City in 4th Industrial Revolution

*Md Ibrahim Khalil[*1], Jae Hyeoung Cho[*2], Won Young Lee[*3], Hyun-Sik An[*4], Jae Sung Hong[*5], Young Sik Park[*6], Woo Sung Jang[*7], Bo Kyung Park[*8], R.Young Chul Kim[*9], Dr. Sheak Rashed Haider Noori[10**]*

SE Lab., Hongik University, Sejong Campus, 30016, Republic of Korea
{ibrahim[1], cho[2], wylee[3], ahn[4], hong[5], park12160422[6], jang[7], park[8], bob[9] }@selab.hongik.ac.kr[*]
Dept. of Computer Science and Engineering, Daffodil International University, 1207, Bangladesh
*drnoori@daffodilvarsity.edu.bd[10**]*

Abstract

In near 4th Industrial revolution environment, there will come out plenty of automatic and smart software in diverse fields such as VR/AR, Autonomous Vehicles, AI, Big data, Blockchain software, and so on. Specially in Smart city, they also need to use many smart software in diverse IoT devices. In future's smart city, there will be very critical issue of Ecoenvironments. That is why we propose good code styles to reduce power consumption. So we analyze procedural language C classified with statements, branches, loops, modulation (communications), and recursive paradigm. With Keil MCU Eval Board, we measure them to find good styles of power consumption in C language. We expect to guide good code stylesof low power to developers in embed systems.

Keywords:*Good code, Low Power consumption, ARM, ULINK plus.*

1. INTRODUCTION

Nowadays, in 4th Industrial Revolution, it may continue to increasethe huge size of the software and also become very important issue ofthe power consumption [1]. In particular, reducing energy consumption is one of the important factors to better software for sustainable services based on limited resource, that is, Eco Environment in the smart city. For this reason, we measure an experiment consisting in the execution of several code patterns on the embedded devices such as ARM ULINK plus module and Keil MCU Eval Board. So we recommend good code styles to enhance the quality of software with low power consumption. The rest of the paper is organized as follows. In chapter 2, we describe the concept of ARM's ULINK plus module and Keil MCU Eval Broad as power measurement experiments. In chapter 3, we recommend good code styles. In chapter3 we mention results and discussion. Finally, in chapter 5, conclusion and future research this paper.

2. Experimental Power Measurement Setup

The Keil MCB Eval Board enables to create and test working programs based on the STMicroelectronicsSTM32 Connectivity Line of ARM [4]. This board comes with two USB cables. Both cables must beconnected to the computer. The first USB cable attaches to the USB connector, and provides power to theboard. By connecting ULINK plus [5] model to Keil MCU Eval Board, software development is possible with Keil tool and the result of code can be checked on LCD screen.

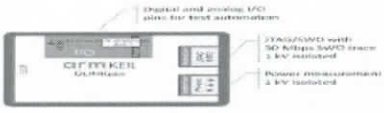

Figure 1. ULINK plus

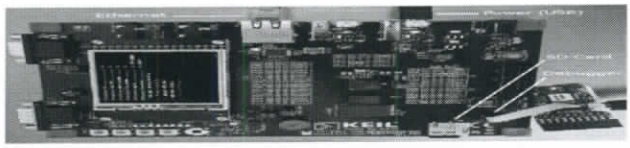

Figure 2. Keil MCB Eval Board

3 Good Code styles for Low Power Consumption

As a result, we recommend*code power patterns* in the procedural language (C) paradigm, which analyze to measure the power of thesoftware code. Each pattern are measured for power consumption at least 50 times.Power measurement is P = V x I wherethe average voltage (V) is set at 3.3V and the current value (I).

4. Power Consumption Measurement

- Loop statements

[Table 1] Comparisonsof For, While, Do- While, While Unrolling, and Do- While Unrolling

	Increasing for statement	while arr[100]times	do-while arr[100]times
Loop	61: for(i=0; i<=100; i++){ 0x08001970 BF00 NOP 0x08001972 E005 B 0x08001980 0x0800197E 1C64 ADDS r4,r4,#1 0x08001980 2C64 CMP r4,#0x64 0x08001982 DDF7 BLE 0x08001974 Power : 5.287359mW	60: int arr[100]; 61: int i = 0; 62: while(i < 100) { 0x08001876 E002 B 0x0800187E 63: arr[i] = i; 64: i++; 65: } power : 5.728602mW	60: int arr[100]; 0x0800186E 4806 LDR r0,[pc,#24] ; @0x08001888 61: int i = 0; 62: do { 63: arr[i] = i; 64: i++; 65: } while (i < 100); power: 5.7421188mW
	Decreasing argument of FOR	while unrolling arr[100]times	do-while unrolling arr[100]times
	61: for(i=100; i>=0; i--){ 0x08001970 2464 MOV5 r4,#0x64 0x08001972 E005 B 0x08001980 0x0800197E 1E64 SUBS r4,r4,#1 0x08001980 2C00 CMP r4,#0x00 0x08001982 DAF7 BGE 0x08001974 Power : 5.267592mW	60: int arr[100]; 61: int i = 0; 62: while(i < 100) { 0x08001876 E006 B 0x08001886 63: arr[i] = i; 64: arr[i + 1] = i + 1; 0x0800187E 1C61 ADDS r1,r4,#1 0x08001880 F84D0021 STR r0,[sp,r1,LSL #2] 65: i += 2; 66: } power : 5.7331758mW	60: int arr[100]; 0x0800186E 4808 LDR r0,[pc,#32] ; @0x08001890 61: int i = 0; 62: do { 63: arr[i] = i; 0x08001876 F84D4024 STR r4,[sp,r4,LSL #2] 64: arr[i + 1] = i + 1; 0x0800187E 1C61 ADDS r1,r4,#1 65: i += 2; 0x08001884 1CA4 ADDS r4,r4,#2 66: } while (i < 100); power: 5.73746514mW
	"**For deccreasing statement**" is lower power consumption than while, do while, while and do while unrollling		

As shown in Table 1, we measured the power of consumption using statement, while, dowhile,while and do while unrolling. We say "**for decreasing statement**" consumed less power thanothers

-Branch statements

[Table 2] Comparison table of*if than else* and *switch-case statement*

Branch	if than else Power: 5.971284mW	Int a=80 A=80 Yes * No A=70 Yes ! No A=90 Yes @ No A=60 Yes #	switch-case power : 5.546343mW	Int a=80 case=80 Yes * No case=70 Yes ! No case=90 Yes @ No case=60 Yes # No default
	"**Switch case**" consume less power than "**if than else**"			

Table 2 shows "**Switch Case**"consumes less power than "**if than else**".

-Call by Value and Call by Reference /Global and local variable

[Table 3] Comparison of the Call by Value and Call by Reference parameters

Call by Value And Call by Reference	Call by Value parameter1 by 1 Power : 5.532912mW	Call by Reference parameter 1 by 1 Power: 5.532417mW	Call by Value 4 power: 5.6034198mW	Call by Reference 4 power: 5.5768218mW
	"**Call by reference**" is less power consume than "**call by value**".			

There are parameters a call by value and call by reference. We went down to 68 line of call by value and 67 line of call by reference. Call by reference is less power consume than call by value.

[Table 4] Comparison of Global variable and Local variable(Double,Float)

Global variable and Local variable	Global variables power: 5.7421188mW	Local variables power: 5.7421188mW
	"**Global variables**" consume the less power than "**Local variables**"	

Table 4 compares the use of global and local variables. The assembly language of BL.W command is used to declare all global variables. Power consumption of global variable is less consume than local variable.

5.RESULT AND DISCUSSION

Table 5 shows the result of calculating the average of the above example codes after performing 10 times experiment of each code. We marked the bold character of results which are the low power consumption.

Table 5: Experimental Results

Syntax	Power measurement	Syntax	Power measurement
Statement 1 Statement 2 Statement 3	**5.469255mW**	If than else Switch case	5.971284mW **5.546343mW**
Increasing argument of FOR	5.287359mW	Call by value(4args)	5.592313mW
Decreasing argument of FOR	**5.267592mW**	Call by reference(4args)	5.573106mW
While	5.745003mW	4 Calls by value(1arg)	5.532912mW
Do while	5.720649mW	4 Calls by reference(1arg)	**5.532417mW**
While unrolling	5.772261mW	Global variable	**5.2852338mW**
Do while unrolling	5.771480mW	Local variable	5.2814652mW

6. CONCLUSION AND FUTURE RESEARCH

In this paper, we examinebetter code stylesfor software quality of low power consumption. We apply some good code using different syntax to develop software. However, in the future, we believe that we will need to further experiment on more complex and larger code to provide more accurate power dissipation with significant power difference.

ACKNOWLEDGEMENT

This research was supported by the Ministry of Trade, Industry and Energy(MOTIE), KOREA, through the Education Program for Creative and Industrial Convergence(Grant Number N0000717) and this research was supported by Basic Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2017R1D1A3B03035421).

References

[1] Qiaing Tong, Ken Choi, Jun Dung Cho, "A Review on System Level Low Power Techniques" 2014

[2] YoungBea Kim, Qaing Tong and Ken Choi,eunchongLee, Sung-Joon Jang and Byeong-Ho Choi " System Level Power Reduction for YOLO2 Sub- modules for Object Detection of future Autonomous Vehicles"

[3] Bo Kyung Park, Hyun Seung SON, and R. Young Chul Kim, "Improvement for effective Combination Cost of solar Energy Integrated Monitoring System based on Long Range, low power Wireless Platform (LoRa Technology) 17 august 2019, PP 51-53.

[4] Hyun Sik An, Je seoung Hong, JinHyub Lee, Ji Hoon Park, Eun Young Byun, R. Young Chul Kim, "Monitoring ArtikPlatfrom Based Electrical Power Consumption on the Solar Power Monitoring System" 20 December 2017