

2021

추계학술 발표대회

2021.10. 22(금) ~ 24(일)

하이원그랜드호텔컨벤션타워

[논문집]

www.ksii.or.kr

주최 및 주관

www.ksii.or.kr
KOREAN SOCIETY FOR INTERNET INFORMATION
KSII
사단법인 한국인터넷정보학회

후원

kakao

S 쌍용정보통신주식회사
Ssangyong Information & Communications Corp.

SK broadband

KTI
미래케이티아이

SK telecom

길벗 (주)도서출판 길벗

책밥

KGU 경기대학교 대표연구소
범죄예방능동빅데이터연구소

CAU 산업보안 전문인력양성 사업단

NAVER CLOVA

LG유플러스

목 차

A. 정보기술서비스 (좌장 : 우호성 교수 / 한국방송통신대)

- A-1. 주문 시스템 무인화를 위한 매장 시스템 설계
이효상, 김준형, 오암석(동명대)
- A-2. 저자 식별자 연계 기반 연구 실적 통합관리 시스템 구축
유은지, 김태현(KISTI)
- A-3. 국가R&D과제 평가위원 참여정보 활용의 개인정보 이슈에 관한 고찰
김종원, 박진형(KISTI)
- A-4. 사용자 중심의 모바일 비콘 서비스 적용
윤동연, 반민아, 오암석(동명대)
- A-5. 국가R&D 정보 활용도 분석 시스템 구축 방안
신용주, 조우승, 이종원, 김태현(KISTI)
- A-6. 국가R&D에서의 마이데이터 서비스 활용 방안 연구
김종원, 유은지, 신동구(KISTI)

B. 소프트웨어공학 / 인터넷소프트웨어 (좌장 : 김재우 수석연구원 / 국방기술평가원)

- B-1. FTA cut-set 분석을 통한 마코프 모델 변환
김소연, 권기현(경기대)
- B-2. 아동의 색 인지 능력 향상을 위한 AR 어플리케이션 개발
서진호, 이도경, 이성진, 홍민(순천향대)
- B-3. ☆ 무인 자율 물체를 위한 SW코드의 저전력 소비 연구
조재형, 윤예동, 장우성, 김영철(홍익대)
- B-4. 의료정보의 접근허가를 위한 빅데이터 처리기반의 자동인증 지원 의료정보 플
합규성, 강민규, 주수종(원광대)

무인 자율 물체를 위한 SW코드의 저전력 소비 연구

조재형*, 윤예동**, 장우성**, 김영철***
 홍익대학교 소프트웨어융합학과 SE연구실
 e-mail : thenryc*, yedong**, jang**, bob***@selab.hongik.ac.kr

A Study on the Lower Power Consumption of software code for Unmanned Autonomous Objects

Jae Hyeong Cho*, Ye Dong Yoon**, Woo Sung Jang**, R.Young Chul Kim***
 SE Lab, Dept. of Software and Comm. Engineering, Hongik University

요 약

지금까지는 고품질, 안정성, 무결성, 그리고 성능 등에 초점을 두고 소프트웨어를 개발하고 있었다. 그러나 최근들어 무인 자율 로봇 또는 드론과 같은 다양한 형태의 소형무인 시스템들의 기능과 역할이 점점 확대되고 있는 추세이다. 이들 소프트웨어는 한정된 배터리 내에서 시스템의 운영 시간을 늘리고자, 전력 소비에 많은 제한을 두는 특징이 있다. 이를 위해, 저전력 코드 정의와 검증방안이 필요하다. 본 논문에서는 드론에서 주로 사용하는 기본적인 프로그램 블록 구조에서 동일기능을 수행하는 다양한 코드 형태를 대상으로 실험한다. 그리고 비교대상 코드들의 전력 소모량을 측정하여, 저전력 Good Code 개선 방법을 제안한다.

1. 서 론

기존 SW 개발방법은 무인 물체 SW의 소스코드에 최적화되지 않았다. 무인형 자율 드론(초소형 비행체)는 하드웨어 변경 또는 코드 구성의 변화를 통해 전력 최소화할 수 있지만, 소프트웨어 코드 리팩토링 방법을 사용하면 동일구조의 SW로 전력을 감소시킬 수 있다[1]. 본 논문은 저전력 코드 리팩토링 방법을 무인 물체 SW에 적용 및 전력을 측정하여 최적화된 리팩토링 패턴을 찾아낸다.

본 논문구성은 다음과 같다. 2장은 관련 연구를, 3장은 전력 측정방법과 환경구축을 서술한다. 4장은 저전력 코드 패턴을 제안하고, 5장은 실험결과를, 6장은 결론을 서술한다.

2. 저전력 코드 리팩토링 방법[2]

표1의 코드 리팩토링 기법은 전자 장비의 소모전력 줄이기 위한 방법으로 어셈블리 코드를 감소시켜 전력 소모를 줄인다. 기존 방법은 복잡한 임베디드 SW를 위한 방법이다. 무인 물체 SW와 같은 단순 구조에는 부적합하다.

(표1) 기존 저전력 코드 리팩토링 패턴의 일부

Energy Bad Smell	리팩토링 패턴
Loop Structure	do-while문 사용
Common sub-expressions	증가하는 패턴을 사용
parameter passing	Call by Reference를 사용하는 패턴

3. 무인 물체 SW를 위한 저전력 코드 측정 방법 설계

본 논문은 기존 리팩토링 방법[3]의 개발을 위해 시도했던 이론을 모두 적용하여 전력 측정한다. 자세한 방법은 표 2와 같다.

3.1 무인 물체 SW 전력 측정 환경 설정

무인 물체 SW 전력 측정 환경은 ARM사의 ULINK Plus 모듈과 keil사의 MCBSTM2F400보드를 사용한다. 보드는 CPU 전력 측정 기능을 제공한다. ULINK Plus는 소스 코드 구간별 전력을 측정해 그래프로 가시화 한다[4].

(표 2) 저전력 패턴을 찾기 위한 코드 리팩토링 방법

Energy Bad Smell	측정 방법
Double Loop Structure	<ul style="list-style-type: none"> 2중 while 전역 측정 2중 do-while 전역 측정 2중 for 전역 측정
Common sub-expressions	<ul style="list-style-type: none"> ++ 증가식 전역 측정 -- 감소식 전역 측정
Parameter passing	<ul style="list-style-type: none"> Call by Value 전역 측정 Call by Reference 전역 측정 다중 매개변수 전달 배열형 매개변수 전달
...	...

3.2 무인 물체 SW 전력 측정 프로세스 설계

표2에 전력 측정을 위한 순서는 다음과 같다.
 ① MCBSTM2F400보드, ULINK Plus, PC를 연결하고, PC에서 Keil5 개발 환경을 실행한다.
 ② Keil5 에 무인 물체 SW 소스코드를 입력 후 디버깅 구간을 설정하고 디버그모드로 프로그램을 실행한다.
 ③ 출력된 전력 그래프를 확인 및 비교한다.
 ④ 전력감소 패턴을 찾았다면 측정을 종료한다.

4. 무인 물체 SW를 위한 저전력 코드 측정

4.1 Double Loop Structure(다중 조건문) 전력 측정

표3의 Double Loop Structure의 실험은 3가지 형태의 반복문을 포함하고, 각 반복문은 소비 전력이 다르다. 본 논문은 모터 제어와 관련된 코드를 삭제한 후 실험한다[5].

(표 3) 다중 Loop Structure를 여러가지 패턴으로 변경한 드론코드

2중for문	2중do-while문	2중while문
<pre>for(uint8_tk=0;k<length;k+=min((int)length,BUFFER_LENGTH)) { for(true&&(timeout=0 millis()-t1<timeout);count++){ data[count]=1; Serial.print(data[count],HEX); }while(true&&(timeout=0 millis()-t1<timeout)); k+=min((int)length,BUFFER_LENGTH); }while(k<length); }</pre>	<pre>do{ do{ Serial.print(data[count],HEX); if(count+1<length)Serial.print(""); count++; }while(true&&(timeout=0 millis()-t1<timeout)); k+=min((int)length,BUFFER_LENGTH); }while(k<length); }</pre>	<pre>while(k<length){ while(true&&(timeout=0 millis()-t1<timeout)){ data[count]=1; Serial.print(data[count],HEX); if(count+1<length)Serial.print(""); count++; } k+=min((int)length,BUFFER_LENGTH); }</pre>
16.8239mAs	15.8465mAs	15.8045mAs

실험은 정확도를 높이기 위하여 각 20회 씩 반복 수행하여 산술평균하였다. 결과적으로 기존 코드의 for 반복문을 while 반복문으로 리팩토링 한다면, 전력의 소모를 줄일 수 있다.

4.2 Common sub-expressions 전력 측정

Common sub-expressions 은 ARM 컴파일러로 측정했을 때 감소하는 코드가 적합하다고 예상한다. 표4의 측정결과 같이 실제 증가하는 패턴을 감소 시켰을 때 어셈블리 라인수(LOC) 및 소비전력이 더 낮은 것을 볼 수 있다.

(표 4) Common sub-expressions을 사용해 드론코드를 전력측정 한 결과

Language	FOR 문의 증가식	FOR 문의 감소식
C Language	<pre>for(int i=0; i<len-1; i++) { q[i-1] = q[i]; } q[0] = data;</pre>	<pre>for(int i=len-1; i>=0; i--) { q[i-1] = q[i]; } q[0] = data;</pre>
Assembly	<pre>1E60 SUBS r0,r4,#1 E009 B 0x08001996 4625 MOV r5,r4 E015 B 0x080019D0</pre>	<pre>1E60 SUBS r0,r4,#1 E009 B 0x08001996 4625 MOV r5,r4 E015 B 0x080019D0</pre>
LOC	15	14
평균값	16.322(mAs)	16.321(mAs)

4.3 Parameter passing전력 측정

Parameter passing에서 전역변수는 사용 할 때 외부함수나 메모리를 사용하기 때문에 표5와 표6과 같이 지역변수를 사용할 때 보다 전력이 더 소모되는 것으로 볼 수 있다.

(표 5) Parameter passing 패턴을 측정한 결과1

Language	Call by Value	Call by Reference
C Language	<pre>void setSpeed(int speed) { if(speed < 0) { speed = 0; } else if (speed > 100) (speed = 100;) }</pre>	<pre>void setSpeed(int* speed){ if(*speed < 0) { *speed = 0;} else if (*speed > 100) { *speed = 100; } }</pre>
Assembly	<pre>CMP r0,#0x00 BGE 0x080019A0</pre>	<pre>6801 LDR r1,[r0,#0x00] 2900 CMP r1,#0x00 DA02 BGE 0x080019A8</pre>
LOC	8	12
평균값	39.7738(mAs)	39.8363(mAs)

(표 6) Parameter passing패턴을 측정한 결과2

Language	Multiple Parameters	Parameters in Array
C Language	<pre>void setRawData(int ax, int ay, int az, int gx, int gy, int gz) { int m_ax,m_ay,m_az,m_gx,m_gy, y,m_gz,scale_factor=1; m_ax = ((double)-ax / 256.0) * scale_factor; m_ay = ((double)-ay / 256.0) * scale_factor; ... }</pre>	<pre>void setRawData(int arr[]) { int m_ax,m_ay,m_az,m_gx,m_gy, m_gz,scale_factor=1; m_ax = ((double)-arr[0] / 256.0) * scale_factor; m_ay = ((double)-arr[1] / 256.0) * scale_factor; ... }</pre>
Assembly	<pre>..... E92D4FF0 PUSH</pre>	<pre>..... E92D5FF0 PUSH</pre>

	<pre>(r4-r11,r) B085 SUB sp,sp,#0x14 4604 MOV r4,r0 460D MOV r5,r1 4616 MOV r6,r2</pre>	<pre>(r4-r12,r) ED2D8B08 VPUSH.64 (d8-d11) 4604 MOV r4,r0</pre>
LOC	81	89
평균값	15.5186	16.1533

5. 실험 결과

무인 물체 SW를 위한 Energy Bad Smell 리팩토링 패턴은 표7과 같다.

(표 7) 무인 물체 SW를 위한 리팩토링 패턴

Energy Bad Smell	무인 물체 SW를 위한 리팩토링 패턴
Loop Structure	while을 사용하는 패턴
Common sub-expressions	인수가 감소하는 패턴
Parameter passing	지역변수를 사용하는 패턴

전체 무인 물체 SW에 제안한 리팩토링 방법을 적용한 결과는 표8과 같다. 단, 모터 구동 함수는 제거하고, 순수 연산 코드만 실행하여 전력을 측정하였다.

(표 8) 무인 물체 SW를 위한 리팩토링 적용 결과

	적용 전	적용 후
전력소모(mAs)	285.4185(mAs)	280.3884(mAs)

6. 결론

본 연구는 무인 자율 물체 소프트웨어의 코드를 이용하여 효율적으로 배터리의 전력을 줄일 수 있는 방법을 제안한다. SW코드를 전력소모가 낮은 다른 패턴으로 대체하기 위해 다중 Loop Structure, Common sub-expressions, Parameter passing의 패턴을 사용 해 전력을 측정하여 불필요하게 사용되는 소프트웨어의 전력량을 개선 했다. 향후 연구로는 이번 연구에서 줄이지 못한 코드 호출 관계에 사용되는 코드들의 전력을 개선하는 연구를 진행 할 것이다.

Acknowledgement

본 논문은 교육부 및 한국연구재단의 4단계 두뇌한국21 사업 (F21YY8102068)과 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임.(No. 2021R111A305040711)

참고 문헌

- [1] Jonathan L. Herlocker, Lee R. Gordon, Jhon Riedl, "GroupLens : Applying Collaborative Filtering to Usenet News", Communication of ACM. Vol.40 No.3, pp.87-97, 1997.
- [2]An, Hyun Sik, et al. "성능 저하 식별을 통한 저전력 개선용 코드 가시화 방법." 정보처리학회논문지:컴퓨터 및 통신 시스템, vol. 9, no. 10, 한국정보처리학회, Oct. 2020, pp. 213-220, doi:10.3745/KTCCS.2020.9.10.213.
- [3]안현식, et all. "고급 프로그래밍 코드 내 전력 소비 측정 통한 저전력 코드 패턴 메카니즘 식별 가이드." (2019).
- [4]HyunSik Ahn, WonYoung Lee, and R. Young Chul Kim, "Guideline of extracting Low Power-Consumed Code Mechanism with Power Consumption in High-Level Code,"
- [5]https://github.com/MichaelD33/AIO-Quadcopter-Flight-Contro


kakao

 **쌍용정보통신주식회사**
SsangYong Information & Communications Corp.

 **SK broadband**


[주]넷케이티아이

 **SK telecom**

 **길벗** (주)도서출판 길벗

 **채밥**

 경기대학교 대표연구소
범죄예방능동빅데이터연구소

 **산업보안 전문인력양성 사업단**

NAVER CLOVA

 **LG유플러스**