

# RTOS 기반의 다중센서 LED 가로등 점등제어 시스템 설계

인치국\*, 인치호°

## A Development of Multi-Sensors LED Streetlight Lighting Control System Based on RTOS

Chi-Goog In\*, Chi-Ho Lin°

### 요약

본 논문에서는 LED 가로등의 에너지 효율을 높인 RTOS 기반의 점등제어 시스템을 제안하였다. 제안된 시스템은 RTOS 기반의 점등제어를 위하여 LED 가로등의 제어 모듈을 3개의 태스크로 세분화하여 실시간으로 처리하도록 하였다. 첫 번째 태스크는 조도를 측정하여 LED 점등신호를 전달하고, 두 번째 태스크에서는 모션디텍터를 이용하여 움직임 감지 신호를 전달한다. 세 번째 태스크에서는 앞에서 전달된 신호를 통하여 LED를 점등제어하도록 설계하였다. 시스템의 검증을 위해 ATmega128 MCU에 직접 포팅하여 동작 상태를 검사하였으며, 실제 LED 가로등의 필드 실험을 통하여 조도분포와 동작 상태를 검증하였다. 본 논문에서 제안한 RTOS 기반의 점등제어 시스템은 여러 기능들을 각각의 태스크 모듈로 분리하여 독립성을 보장하기 때문에 시스템의 성능 향상 및 추가 기능들을 가능하게 하고, 지능적인 패턴 점등제어로 인하여 최적의 에너지 절감 효율성을 보였다.

**Key Words** : LED, Streetlight, RTOS. Pattern lighting, Lighting algorithm

### ABSTRACT

In this paper, we proposed a RTOS-based lighting control system to improve energy efficiency. This proposed system, real time process was designed to the specified division of the LED streetlight control module for the RTOS-based lighting control into three different tasks. The first task transmits LED lighting signal by measuring illuminance, and the second task transmits motion detecting signal using motion detector. In the third task, lighting control to LED was designed through passed control signal from other tasks. The execution status was examined ports that are directly to the ATmega128 MCU for the verification of the system, and illuminance distribution and operating conditions were verified through LED street field test. The proposed RTOS-based lighting control system has brought improving system performance and also facilitate an addition of other functions, and it was possible to optimize energy saving by intelligent lighting pattern control.

### I. 서론

임베디드 기술이 발전함에 따라 LED(Light  
Emitted Diode) 조명장치도 단순히 점·소등만 처리하

는 것뿐만 아니라 여러 개의 센서 및 장치들이 결합되  
었고, 이를 구현하기 위한 소프트웨어의 복잡도가 증  
가하게 되었다<sup>1,2)</sup>.

기준에 사용되었던 PLC(Programmable Logic

※ 본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임(I00041).

• 주저자 : 세명대학교 컴퓨터학부, profavor@semyung.ac.kr, 준회원

° 교신저자 : 세명대학교 컴퓨터학부, ich410@semyung.ac.kr, 정회원

논문번호 : KICS2012-07-340, 접수일자 : 2012년 7월 31일, 최종논문접수일자 : 2012년 10월 12일

Controller)<sup>[3]</sup> 기반의 시스템은 각각의 입력에 대하여 처리할 경우 많은 제약을 받으며 모든 과정이 반복 실행되므로 시스템의 성능저하와 하나의 오류에도 시스템의 동작을 다운시킬 수 있는 위험요소를 가지고 있다<sup>[4]</sup>. 그러나 RTOS(Real Time Operation System)<sup>[5]</sup>를 적용하면 정의된 모듈의 동작을 태스크(Task)로 세분화하여 프로그램의 복잡성을 줄일 수 있으며 시스템 동작을 모듈별로 정의하여 동작의 효율성을 높일 수 있다<sup>[6-8]</sup>. LED의 점등은 1초에 수백 번의 점멸을 수행함으로써 커지게 된다. 즉 하나의 프로세스는 지속적인 LED의 점등제어를 수행하여야 한다. 기존의 LED 점등제어 시스템은 하나의 프로세스로 수행하는 PLC 기반의 시스템구조를 가지고 있으며, 타임스케줄에 의해 전체 LED를 점등 및 소등하는 방식이었다. 이러한 시스템에서 여러 개의 센서들을 측정 및 제어 하면서 LED의 점등상태를 유지하려면 인터럽트를 통해서만 가능하였고, 새로운 기능이 추가될수록 인터럽트 주파수를 재조정해야 할 뿐만 아니라 프로그램의 구조도 점점 복잡해지기 때문에 지능형 가로등에 적합하지 않다. 또한 전체 LED의 지속적인 점등으로 인하여 에너지 효율성도 떨어지게 된다<sup>[9,10]</sup>.

본 논문에서는 이러한 문제점을 해결하기 위하여 RTOS 기반의 다중센서 LED 가로등 점등제어 시스템을 설계한다. 제안된 시스템은 PLC기반의 시스템이 가지고 있는 한계점을 극복하고 안정성 및 신뢰성을 보장할 수 있으며, 다중센서를 통해 가로등 주변의 조도와 움직임 감지를 측정하는 기능을 태스크 단위로 분할하여 멀티태스킹을 수행한다. 그리고 다중센서로부터 측정된 정보를 통해 실시간으로 LED의 점·소등 및 점등패턴을 제어하여 저전력 및 높은 에너지 효율을 가지는 LED 점등제어 시스템을 설계한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 본 논문에서 제안하는 RTOS 기반의 LED 점등제어 시스템에 대하여 소개한다. 제 3장에서는 RTOS의 포팅실험과 필드실험에서 제안된 점등제어 시스템을 적용하여 얻은 데이터를 분석한다. 마지막으로 제 4장에서는 분석된 결과를 종합하여 결론을 맺는다.

## II. RTOS 기반의 점등제어 시스템 설계

본 논문에서 제안하는 점등제어 시스템은 각 모듈별 동작을 태스크 단위로 세분화하고 제어신호를 전달하여 실시간으로 점등제어를 할 수 있도록 한다. 그림 1은 제안하는 전체 시스템의 구성을 나타낸 것이다.

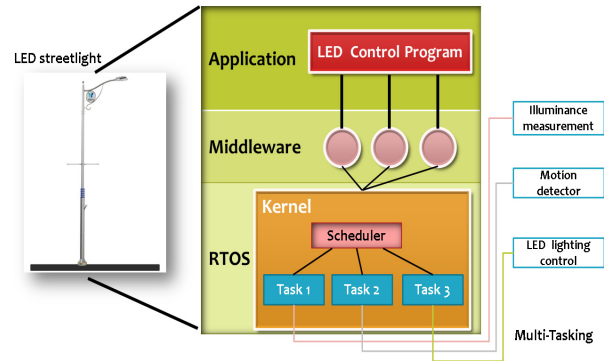


그림 1. 점등제어 시스템의 구성  
Fig. 1. The configuration of the lighting control system

시스템의 최상위 계층인 응용(Application) 계층에서는 LED 가로등을 원격제어 및 모니터링하며, 중간 계층인 미들웨어(Middleware)에서는 그림 2와 같이 지그비(Zigbee) 기반의 무선 네트워크<sup>[11]</sup>를 구성하여 가로등의 조도, 모션 센서들을 포함하는 다중센서 모듈을 통해 실시간으로 센싱된 데이터 값들을 가로등 통제센터에 송수신하게 된다. 이 때 수집된 데이터는 관리자가 원하는 조건에 맞게 점등제어를 할 수 있도록 각 가로등에 신호를 전달하여 원격으로 제어가 가능하도록 하였다.

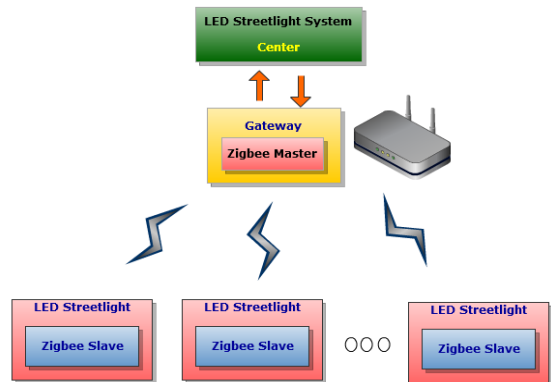


그림 2. 지그비 네트워크 구성도  
Fig. 2. The Zigbee network configuration

마지막으로 RTOS 계층에서는 실제 LED 가로등의 모듈을 제어하는 부분으로서 조도 측정과 움직임 감지, 그리고 LED 점등제어 모듈이 각각의 태스크로 분리되어 있고, 각 태스크는 스케줄러에 의하여 멀티태스킹으로 동작된다. 그림 3은 태스크들의 동작을 블럭 다이어그램으로 표현한 것이다.

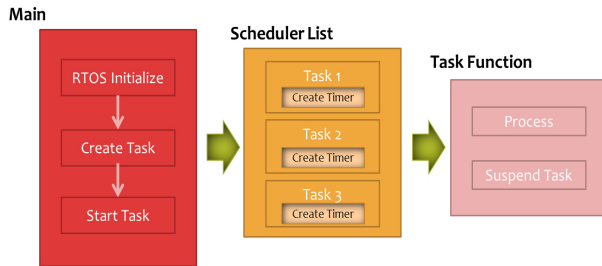


그림 3. 태스크들의 동작 블록 다이어그램  
Fig. 3. The operation block diagram of tasks

멀티태스킹을 수행하기 위해서는 각각의 태스크에 우선순위를 정하여 스케줄러 리스트에 등록을 해야 한다. 먼저 RTOS 시스템의 환경설정을 위해 초기화를 수행하고, 태스크를 생성하여 우선순위에 의해 태스킹을 수행하게 된다. 각각의 태스크에는 타이머를 생성하여 설정된 시간 간격으로 태스크를 재호출하도록 구성하였다.

2.1. LED 점등제어 알고리즘 설계

본 논문에서 제안하는 RTOS 기반의 LED 점등제어는 다중센서를 이용하여 측정된 조도와 움직임 감지하여 효율적인 에너지 절약을 위한 방법이다. 센서 값을 측정할 주변 환경 정보를 바탕으로 어두워졌을 때 자동으로 점등되도록 설계한다. 점등 시 거리에 사람이 없을 경우 마방진을 이용한 완전대칭적인 패턴으로 번갈아가며 점등을 수행하고, 사람이 감지된 경우에는 전체 LED를 점등하여 주변을 환하게 밝혀준다. 그림 4는 본 논문에서 제안하는 점등제어 시스템의 시뮬레이션을 나타낸 것이다.

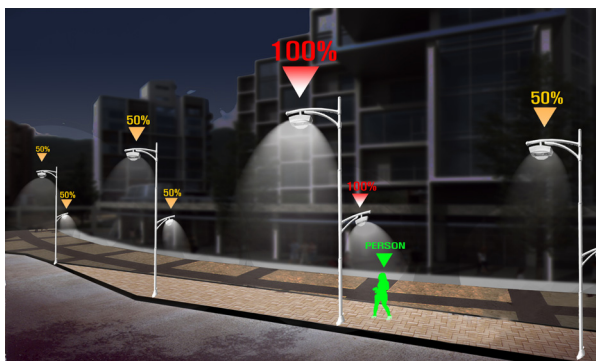


그림 4. 제안하는 점등제어 시스템의 시뮬레이션  
Fig. 4. The simulation of proposed lighting control system

표 1은 조도센서와 모션 디텍터에서 측정된 센싱 값을 10단계 분류에 대한 LED 점등과 움직임 감지상태를 나타낸 것이다.

표 1. 다중센서의 레벨에 의한 LED 제어  
Table 1. The LED control by level of multiple sensors

Illuminance level (max 10 level)	Motion level (max 10 level)	LED lighting	Motion
3 level over	3 level over	LED off	Non-detecting
3 level over	2 level under	LED off	Detecting
3 level under	3 level over	50% pattern lighting	Non-detecting
3 level under	2 level under	100% full lighting	Detecting

조도와 모션감지 레벨을 10단계로 분류하여 조도 레벨이 3단계 미만일 경우에만 LED를 점등시킨다. LED 점등 시 레벨이 3단계 이상일 경우에는 LED를 마방진 응용 패턴으로 부분점등하고, 2단계 이하일 경우에는 전체 LED를 점등하도록 설계하였다. 그리고 조도 센서의 오차를 줄이기 위하여 그림 5와 같이 조도 레벨 분류 알고리즘을 설계한다. 조도 센서는 광량에 따라 저항 값이 변화되게 됨으로 MCU의 ADC 포트에 연결하여 사용한다. ATmega128 MCU의 ADC의 경우 10비트로 분해되기 때문에 0에서 1023의 값을 가지게 된다. 조도레벨 분류 방법은 10회의 조도를 측정하여 조도데이터 변수에 누적을 하고, 그에 대한 평균값을 구한다. 이러한 결과 값을 기준으로 102로 나누어 0에서 9단계로 레벨을 반환하도록 설계하였다.

```

int cds_adc_f() {
    unsigned char ADC_avr=0;
    unsigned char illuminance=0;
    unsigned int illuminance_level=0;
    PORTA=0x01; // cds adc selection
    cDATA(0); delay_us(10);
    sDATA(0); delay_us(10);

    // 10 loop sum of illuminance data
    for(i=0 ; i<10; i++)
        illuminance += ADC_Read(1);
    ADC_avr=illuminance/10;

    // 10 level divide to measured illuminance data
    illuminance_level = ADC_avr / 102;
    return illuminance_level;
}
    
```

그림 5. 측정 조도 레벨 분류 알고리즘  
Fig. 5. The algorithm of measuring illuminance level

본 논문에서 점등제어를 위한 태스크는 3개로 구성되어 있다. 먼저 첫 번째 태스크는 조도 센서를 통하여 측정된 값을 10단계로 나누어 분류하고, 3단계 이하일 경우에만 점등되도록 제어신호를 전달한다. 조도 측정은 30초 간격으로 타이머에 의하여 재호출 된다. 그림 6은 태스크 1에서 설계한 알고리즘을 나타낸 것이다.

```

void task1_exe(void *p_arg)
{
    // create timer (30 seconds)
    rtos_tmr_create(task1_exe, AA_SIG_ILLUMINANCE , 0,
    (30*1000));
    for(;;) {
        // check signal
        rtos_sigs_t sigs = Rtos_wait(AA_SIG_ILLUMINANCE,
        _RTOS_FALSE);
        // if receive to illuminance signal
        if(sigs & AA_SIG_ILLUMINANCE) {
            // save to illuminance level
            illuminance = cds_ads_f();
            // if illuminance level less than 3 level,
            // send to LED lighting signal
            if(illuminance < 3 ) ledPower = 1;
            // Otherwise, sending to LED off signal
            else ledPower = 0;
        }
    }
}
    
```

그림 6. 조도 측정을 위한 태스크 1의 알고리즘  
 Fig. 6. A Task 1 Algorithm for measuring illuminance

두 번째 태스크에서는 모션 디텍터를 통하여 움직임을 감지한다. 움직임 감지 레벨도 조도 레벨과 같이 10단계로 분류하여 움직임 감도의 조절이 가능하다. 본 논문에서는 움직임 감지레벨이 3단계 미만일 경우에만 감지 신호를 보내도록 설계한다. 그림 7은 태스크 2에서 설계한 알고리즘을 나타낸 것이다.

```

void task2_exe(void *p_arg)
{
    // create timer (1 second)
    rtos_tmr_create(task2_exe, AA_SIG_MOTIONDETECT, 0,
    (1*1000));
    for(;;) {
        // check signal
        rtos_sigs_t sigs= Rtos_wait(AA_SIG_MOTIONDETECT,
        _RTOS_FALSE);
        // if receive to motion detecting signal
        if(sigs & AA_SIG_MOTIONDETECT) {
            // save to motion detecting level
            motionDetector = motion_adc_f();
            // if motion detecting level less than 3 level,
            // send to LED lighting signal
            if(motionDetector < 3 ) motionOn = 1;
            else motionOn = 0;
        }
    }
}
    
```

그림 7. 움직임 감지를 위한 태스크 2의 알고리즘  
 Fig. 7. A Task 2 algorithm for motion detecting

세 번째 태스크에서는 상위 태스크들의 다중 센서에 대한 제어신호를 통하여 LED 매트릭스(Matrix)의 점등 및 패턴변화를 하게 된다. 패턴변화를 위하여 타

이머를 호출하여 10초 간격으로 플래그(flag) 변수의 상태 비트를 변환시킨다. 점등신호만 할당이 된 경우에는 마방진을 이용한 패턴으로 점등이 되며, 일정시간 간격으로 패턴을 90도 회전 또는 반전시켜 점등을 하게 된다. 그러나 점등신호와 움직임 감지 신호가 모두 할당된 경우에는 모든 LED를 점등 시켜 주변을 밝게 비추게 된다. 그림 8은 점등제어를 처리하는 태스크 3의 모듈을 설계한 알고리즘을 나타낸 것이다.

```

void task3_exe(void *p_arg)
{
    // create timer (10 seconds)
    rtos_tmr_create(task3_exe, AA_SIG_FLAG , 0,
    (10*1000));
    // create timer (1 second)
    rtos_tmr_create(task3_exe, AA_SIG_LEDOn , 0, 1000);
    for(;;) {
        rtos_sigs_t sigs = Rtos_wait(AA_SIG_FLAG,
        AA_SIG_LEDOn, _RTOS_FALSE);
        // check to FLAG signal
        if(sigs & AA_SIG_FLAG) {
            //control signal for pattern rotation at 90 degrees
            if(flag==0) flag = 1;
            //control signal for pattern reversion
            else if(flag == 1) flag = 0;
        }
        // check to LED control signal
        if(sigs & AA_SIG_LEDOn) {
            if(ledPower == 1 & motionOn == 0) {
                // reverse and rotate to pattern by flag signal
                for(int i=0; i<1000; i++)
                    LEDON(patternLighting(flag));
            }else if(ledPower == 1 & motionOn == 1)
                // full LED lighting
                for(int i=0; i<1000 i++)
                    LEDON(fullLighting);
        }
    }
}
    
```

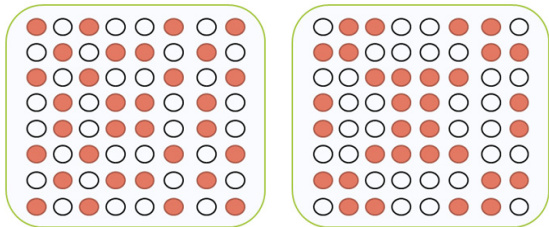
그림 8. 점등제어 처리를 위한 태스크 3의 알고리즘  
 Fig. 8. A Task 3 algorithm for lighting control process

### 2.2. 마방진을 이용한 LED Matrix 패턴

마방진은 1에서  $n^2$ 까지의 양의 정수를 정사각형으로 배열하여 가로, 세로, 대각선상의 합계가 모두 같도록 만든 것을 말한다. 마방진 패턴은 영역을 크게 4사분면으로 나누었을 경우 이웃하는 사분면 사이에 패턴의 대칭성이 있고, 이러한 특성을 이용하면 상하 좌우 대칭적인 패턴 생성이 가능하다. 패턴이 완전히 대칭이 된다면 패턴을 반전시키거나 회전시켜도 조도를 골고루 분포할 수 있다. 조도 분포는 평면적 또는 공간적인 조도의 확산이나 변화를 나타낸 것으로 평균 조도가 동일하여도 분포 상태에 따라 달라 보이기 때문에 조도의 분포상태는 매우 중요하다. 마방진 패턴은 4사분면 중에 한 영역에 각 행과 열 방향으로 같



은 개수씩의 배열을 0과 1로 구분한 후, 이를 다른 사분면 배열에 대칭시킴으로써 생성된다. 그림 9는 마방진을 이용하여 얻을 수 있는 패턴을 나타낸 것이다.



(a) Magic square pattern A (b) Magic square pattern B  
 그림 9. 짝수마방진의 응용패턴  
 Fig. 9. The applied pattern of an even magic square

### 2.3. 패턴변화 점등 설계

마방진을 이용한 패턴점등을 하기 때문에 모든 LED를 골고루 사용하기 위한 패턴변화 알고리즘이 필요하다. 따라서 그림 10과 같이 LED Matrix를 일정시간 간격으로 90도로 회전하고 반전시켜 점등을 하게 된다. LED를 번갈아가며 점등하고, 인접하여 점등되는 LED가 줄어들기 때문에 LED의 내부발열 온도를 저하시킨다. 이러한 점은 LED의 성능을 향상시킴과 동시에 효율적인 에너지 절약이 가능하다.

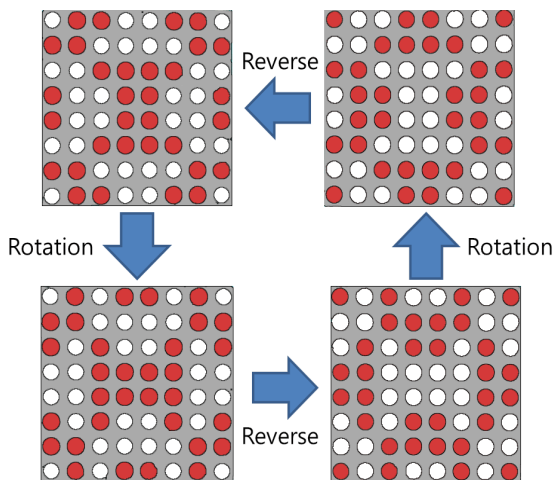


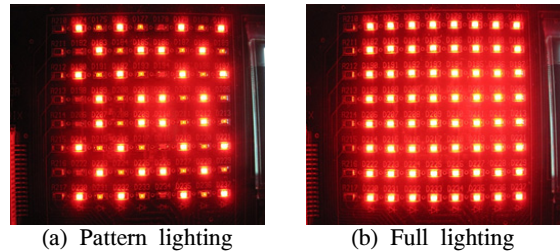
그림 10. 패턴변화 알고리즘  
 Fig. 10. The algorithm of pattern changes

## III. 실험 내용

### 3.1. 테스트 실험장치의 구성 및 실험

본 논문에서는 제안된 점등제어 시스템을 검증하기 위하여 AVR 계열의 ATmega128 MCU와 조도 센서, 모션 디텍터, 그리고 LED 매트릭스가 탑재된 “H-LED Training Kit”라는 제품을 이용하였다. 실험

환경 구축을 위해 PC에 컴파일러 도구인 WinAVR과 통합개발 프로그램인 AVR Studio 4를 설치하고 JTAG를 통해 앞서 설계한 RTOS 기반의 점등제어 프로그램을 장치에 포팅하였다. 컴파일 결과 23.0 KB의 실행파일을 얻을 수 있었고, 그림 11과 같이 프로그램이 정상적으로 동작함을 확인할 수 있었다.



(a) Pattern lighting (b) Full lighting  
 그림 11. 테스트 프로그램의 동작 검증  
 Fig. 11. The activity validation of the test program

기존의 디밍 제어를 통하여 휘도를 조절하여 에너지 절약을 했던 방법과는 달리 마방진을 이용한 패턴점등 알고리즘은 LED를 부분 점등하는 방법이다. 따라서 별도의 디밍 회로 구성이 불필요하기 때문에 개발비용 절감효과를 얻을 수 있다. 그리고 인접하여 점등되는 LED의 수가 줄어들기 때문에 내부발열 온도가 저하되며 온도상승으로 인해 발생하는 수명, 밝기, 색상 변화 등에 대한 성능저하 요인의 문제점 해결이 가능하였다.

### 3.2. LED 가로등의 필드 실험

실제 LED 가로등의 실험을 위해 본교 전산정보학관 앞에 설치된 LED 가로등 테스트 모델에서 실험을 진행하였다. 패턴에 따른 조명도를 분석하기 위하여 그림 12와 같이 다중센서 플랫폼이 결합되어 있는 Hmote2420의 싱크노드와 싱크노드로부터 데이터를 수집하는 베이스노드 그리고 수집된 데이터를 화면에 보여줄 수 있는 노트북을 이용하여 조도를 측정하였다.

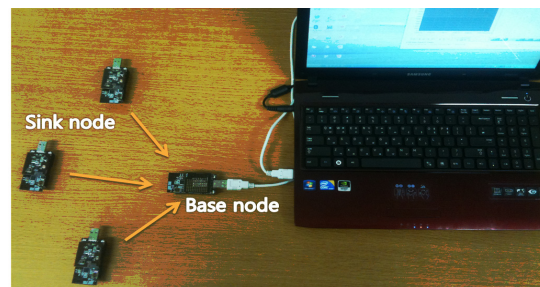


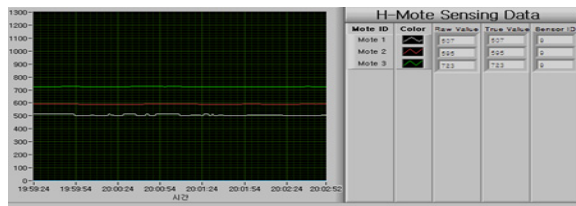
그림 12. 조도 측정을 위한 장치 구성  
 Fig. 12. The device configuration to measuring illuminance

조도측정은 조도분포를 분석하기 위하여 가로등의 정면으로부터 1미터 간격으로 3개의 싱크노드를 설치하여 측정하였다. 측정된 결과, 패턴 점등시의 조도는 최고 723 럭스(lux)로 은은한 밝기를 보였으며, 패턴 점등을 할 경우에도 조도의 변화폭이 크지 않았다. 그리고 모션 디텍터를 통해 움직임을 감지할 경우에는 전체 LED가 점등이 되면서 최대 1207 럭스로 밝기가 40%이상 상승되었다. 그림 13과 그림 14는 필드 실험을 통해 정상적으로 동작되는 모습과 패턴 점등시와 전체 점등시의 조도를 LabView를 이용하여 측정된 결과를 나타낸 것이다.

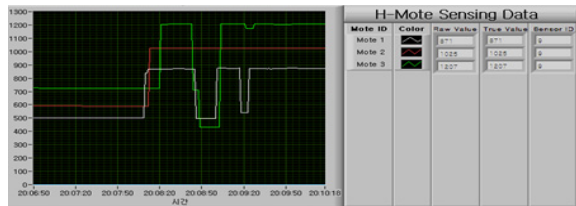


(a) The image while pattern lighting (b) The image while full lighting

그림 13. LED 가로등의 필드 실험 모습  
Fig. 13. The appearance of LED streetlight in the field



(a) The measuring illuminance while pattern lighting



(b) The measuring illuminance when motion detecting

그림 14. 점등방식에 대한 조도측정  
Fig. 14. The measuring illuminance about lighting way

본 논문에서는 유동인구에 따라서 에너지 절감 효율이 달라지기 때문에 설치된 가로등을 기준으로 일몰 후 40분, 일출 전 40분 사이의 유동인구를 측

정 하였다. 그 결과, 일몰 후에는 유동 인구가 많았으며, 일출 전에는 유동 인구가 거의 없는 것으로 측정되었다. 그 외의 야간 시간대에는 보행자의 교통량이 매우 미미한 것으로 나타났다. 따라서 본 논문에서 제안하는 점등제어 시스템을 적용한다면 약 40% 이상의 에너지 절감 효과를 얻을 수 있었음을 확인하였다.

### 3.3. 제안된 점등제어 시스템의 비교

기존에 설치된 LED 가로등은 타임스케줄에 의하여 일괄 점등되는 단순한 기능으로 항상 최대 출력으로 주변을 비추었기 때문에 소비전력이 높고, LED의 내부온도의 상승으로 인하여 LED의 수명을 저하시켰다. 그러나 제안된 점등 시스템은 조도 센서에 의한 실시간 점·소등하여 에너지를 절약하고, 패턴점등 제어를 통한 LED의 동작 개수가 50%로 감소로 인하여 소비전력이 46% 감소되었을 뿐만 아니라 LED의 발열온도도 19% 감소되었다. 또한 LED의 적은 사용률로 인하여 수명이 50%이상 향상 되었고, 사람 유무 인지 기술을 통해 효율적인 조명효과를 줄 수 있었다. 표 2는 기존의 LED 점등시스템과 제안된 점등시스템을 비교하여 나타낸 것이다.

표 2. 기존시스템과 제안된 시스템의 차이점  
Table 2. The differences between previous system and proposed system

Item	The existing lighting system	The proposed lighting system	Function evaluation
Operating LED	64	32	50.0% reducing
Power consumption	69.6	37.6	46.0% reducing
Heat(℃)	57	46	19.3% reducing
Luminous efficiency	73	45	39.4% reducing
Lifetime	10000-50000	15000-80000	50.0% improving
Lighting method	Full lighting	Pattern lighting	Energy saving
Design method	PLC-based design	RTOS-based design	Improved availability
Energy saving method	Power control by time schedule	Realtime power control using illuminance sensor	Realtime control
Motion detect	Install to additional devices	Detected by the motion detector	Efficiency lighting control

## IV. 결 론

본 논문에서는 기존의 PLC 기반의 시스템의 단점을 극복하고, 에너지의 효율적인 절약이 가능한 RTOS 기반의 점등제어 시스템을 제안하였다.

본 논문에서는 여러 기능들을 각각의 태스크 모듈로 분리하여 독립성을 보장하기 때문에 시스템의 성능 향상을 가져올 뿐만 아니라 다른 기능들의 추가가 용이하였고, 지능적인 패턴 점등제어로 인하여 최적의 에너지 절감이 가능하였다. 패턴 점등할 경우에는 조도의 변화폭이 크지 않았으며, 소비전력이 46% 절감되었다. 또한 발열온도가 19% 감소되었기 때문에 LED의 수명을 극대화 시킬 수 있었다.

따라서 본 논문에서 제안된 RTOS 기반의 점등 제어 시스템을 적용할 경우 안정적이고 높은 에너지 효율을 얻을 수 있었다.

### Acknowledgement

본 연구는 지식경제부 출연금으로 ETRI SW-SoC 융합R&BD센터에서 수행한 시스템반도체 설계인력양성사업의 연구결과입니다.

### References

[1] Denardin and G.W., “An Intelligent System for Street Lighting Control and Measurement”, *IAS 2009*, pp.1-5, Oct. 2009.

[2] Lubbers, E., “ReconOS: An RTOS Supporting Hard-and Software Threads”, *FPL 2007 International Conference*, pp. 441-446, Aug. 2007.

[3] AS Tasu, “Programmable logic controller”, *Rom. Journ. Phys.*, 51(1-2), pp. 305-310, Dec. 2006.

[4] Je Wung Kim and Sung-Soo Lim, “System Model-driven Conversion from PLC-based Systems to RTOS-based Systems”, *The Korean Society Of Computer And Information*, 14(33), pp. 13-26, Mar. 2009.

[5] JA Stankovic and R Rajkumar, “Real-Time Operating Systems”, *Real-Time Systems*, 28(2-3), pp.237-253, Nov. 2004.

[6] Parisa Razaghi and Andreas Gerstlauer, “Host-compiled multicore RTOS simulator for embedded real-time software development”, *DATE(2011)*, pp. 1-6, Mar. 2011.

[7] S. Hong, “Coping with Embedded Software Crisis using Real-time Operating Systems

and Embedded Middleware“, *Invited for presentation at IEEE Asian Pacific ASIC (AP-ASIC) Conference*, Jeju Island, Korea, Aug. 2000.

[8] A. Gerstlauer, H. Yu and D.D. Gajski., “RTOS Modeling for System Level Design”, *DATE of Conference 2003*, pp. 130-135, Dec. 2003.

[9] King-Tong Lau, W. S. Yerazunis, R. L. Shepherd and D. Diamond, “Quantitative colorimetric analysis of dye mixtures using an optical photometer based on LED array”, *Sensors and Actuator B: Chemical*, 114(2), pp. 819-825, Apr. 2006.

[10] M.S. Wu, H.H. Huang, B.J. Huang, C.W. Tang, and C.W. Cheng, “Economic Feasibility of Solar-powered LED Roadway Lighting”, *ISESCO Science and Technology Vision*, 4(6), pp. 43-47, Aug. 2009.

[11] Jae-han Ju, Tae-hoon Kim, “The Design of Leakage Current Control System using ZigBee Communication”, *KICS*, 34(12), pp. 330-339, Dec. 2009.

#### 인 치 국 (Chi-Goog In)



2011년 2월 세명대학교 컴퓨터 학부 졸업(이학사)  
 2011년 3월~현재 세명대학교 컴퓨터학과 석사과정  
 <관심분야> SOC CAD, ASIC 설계, CAD 알고리즘, SOC 설계, RTOS 및 내장형 시스템

#### 인 치 호 (Chi-Ho Lin)



1985년 한양대학교 공과대학 공학사  
 1987년 한양대학교 대학원 공학석사(CAD 전공)  
 1996년 한양대학교 대학원 공학박사(CAD 전공)  
 1992년~현재 세명대학교 컴퓨

터학부 교수  
 <관심분야> SOC CAD, ASIC 설계, CAD 알고리즘, SOC 설계, RTOS 및 내장형 시스템