

지능형 LED 점등시스템을 위한 점등제어 알고리즘 설계

Design of Lighting Control Algorithm for Intelligent LED Lighting System

홍 성 일*, 인 치 호**

Sung-Il Hong*, Chi-Ho Lin**

Abstract

In this paper, we propose the design of lighting control algorithm for intelligent LED lighting system. The proposed lighting control algorithm transmitted to MCU through a data bus the environmental information detected from respectively sensor node. The MCU control software was designed to determine the level maintained to depending on the set control method by comparing the results that calculated the dimming level using a signal value. Also, it was designed to be lighting by cross-performed periodically the rotation and reverse method by created fully symmetrical pattern using the control algorithm to LED lighting device. In this paper, the proposed lighting control algorithm improved the reliability of the data sent by designed the system that can be controlled lighting to stable, and it was maintained the event delivery ratio of 91%. Also, the lighting device was decreased the luminous intensity of 32%, the power consumption of 49%, and heat generation of 32%. As a result, it were could be improved the energy efficiency that the life-cycle of LED has been increased 50%.

요 약

본 논문에서는 지능형 LED 점등시스템을 위한 점등제어 알고리즘 설계를 제안한다. 제안하는 점등제어 알고리즘은 각각의 센서노드에서 감지된 환경정보를 데이터 버스를 통해 MCU에 전송한다. MCU의 제어 소프트웨어는 설정된 제어방식에 따라 신호 값을 이용하여 조광레벨을 계산하고 결과를 비교하여 레벨유지를 결정하도록 설계하였다. 또한 LED 조명기에 점등제어 알고리즘을 이용하여 완전대칭 되는 패턴생성에 의하여 회전 및 반전 기법을 주기적으로 교차 수행하여 점등되도록 설계하였다. 본 논문에서 제안한 점등제어 알고리즘은 안정적으로 점등제어를 하는 시스템을 설계하여 전송 데이터의 신뢰성을 높이고 91%의 이벤트 전달 비율을 유지하였다. 또한, 조명기의 광속은 32% 감소되었고 소비전력은 49% 감소되었으며 발열량은 32% 감소되었다. 결과적으로 LED의 수명은 50% 증가되어 에너지 효율성을 향상할 수 있었다.

Key words : Control circuit, Sensor node, Lighting control, Pattern, MCU

* 世明大學校 컴퓨터學部
(School of Computer, Sungmyung University)

★ 교신저자 (Corresponding author)

※ 감사의 글 (Acknowledgment)

본 연구는 교육과학기술부와 한국연구재단의 지역혁신 인력양성사업과 지식경제부 출연금으로 ETRI SW-SoC융합R&BD센터에서 수행한 시스템반도체 설계인력양성사업의 연구결과입니다.

接受日:2012年 09月 04日, 修正完了日: 2012年 09月 25日
掲載確定日: 2012年 09月 26日

1. 서론

최근 LED의 성능 개선이 빠르게 이루어지면서 응용 제품 개발과 기존 조명기들을 대체하려는 연구와 노력이 많은 부분에서 이루어지고 있다. 특히, 21세기의 조명 설계분야는 지능화 및 자동화 기술 기반으로 발전 되고 있다.^[1-2]

LED 조명은 여러 개의 LED들이 하나의 모듈이 되어 점등되는 것이 일반적이다. LED는 열로 85%소

모되고, 광으로 15%소모된다. LED는 발열이 높을 수록 램프의 수명이 감소하고, 컬러의 변화가 발생하여 조도가 저하되는 등의 많은 문제점이 있다.^[3-5]

기존 조명기기는 센서 동작에 따른 단순 온·오프 기능만을 구현하기 때문에 제어회로 설계가 복잡하지 않았으나 LED 조명기기에서는 인체감지 및 주광감응 센서와 사용자 환경에 따른 설정 등 많은 기능을 통제하고 운영하는 제어 시스템을 사용할 수 있으므로 MCU를 사용한 설계가 필요하다. 문제점 해결을 위하여 본 논문에서는 지능형 LED 점등시스템을 위한 점등제어 알고리즘을 제안한다.

본 논문에서 제안하는 점등제어 알고리즘은 점등 상태에 대한 신뢰성을 높이기 위해 각각의 센서노드가 주변 환경정보 데이터를 원격으로 센싱하고 데이터 버스를 통해 MCU로 전송하도록 한다. MCU의 제어 소프트웨어는 센싱값의 보정을 통하여 미리 설정된 제어방식에 따라 센싱값을 이용하여 조광레벨을 계산하고, 결과를 비교하여 레벨유지 여부를 결정하게 한다. 이와 같이 실시간 이벤트 처리와 조명기기 제어를 할 수 있도록 MCU와 센서노드 사이의 데이터버스를 통하여 데이터를 안정적이고 효율적으로 전송하고, 이벤트 전달 비율을 유지하기 위해 데이터의 전송과 센서노드의 센싱 데이터는 변동률이 기준 값보다 큰 경우에는 이벤트의 타깃 전달율을 만족하기 위하여 독립경로를 사용하여 데이터 전송하고 효율성을 향상시켜 신뢰성을 확보할 수 있다.

또한, LED 조명기기의 내부발열 상승으로 인한 컬러변화와 전력소모의 증가 등의 문제점 해결을 위하여 전력소모가 적은 고휘도 백색 LED의 특성을 이용한다. 전체 LED 조명기기의 점등은 패턴을 적용하고 제어하여 LED의 발열량을 감소시키며 모듈별 램프 수명을 극대화시키고 동작 LED 대비 최대의 광속을 늘릴 수 있도록 설계한다. 패턴에 의한 점등제어를 하게 되면 LED 조명기기의 휘도 변화가 발생할 수 있으나 고휘도 백색 LED 모듈을 사용하면 간단히 해결할 수 있다. 따라서 본 논문에서 제안하는 점등제어 알고리즘은 안정적으로 점등제어를 하는 지능형 LED 점등시스템을 설계할 수 있었고, 전송 데이터의 신뢰성 향상을 위해 독립경로를 사용하여 평균 91%의 이벤트 전달 비율을 유지 하였다. 결과적으로 조명기기의 광속은 32%정도 감소되었으나 고휘도 백색 LED를 사용하여 문제점을 해결하였으며 소비전력은 49% 감소되었으며, 발열량은 32% 감소되었다. 또한, LED의 수명은 50% 증가되어 에너지 효율성을 향상할 수 있었다.

II. 지능형 LED 점등 시스템

본 논문에서 제안하는 지능형 LED 점등 시스템을 위한 점등제어 알고리즘의 장점은 다수의 센서를 이용하여 지능형으로 점·소등 제어가 가능하게 하여 에너지 절감에 효과적이며, 프로그램 변경에 의해 원격 감시 및 제어기능 추가가 쉽기 때문에 점등제어 시스템 설계 및 구축에 따른 비용절감 등의 효과를 얻을 수 있다. 또한, 환경기반 정보들을 실시간 이벤트 처리 및 제어를 하여 전송 데이터의 신뢰성을 높이고 이벤트 전달비율을 유지할 수 있다.

그림 1은 점등제어 알고리즘을 활용하여 조명기기를 제어하기 위한 지능형 LED 점등 시스템의 구성을 나타내었다. 지능형 LED 점등 시스템에서 사용하는 구동회로는 저가격의 다이내믹 구동 방식을 사용, 다중 센서의 센싱 데이터를 AD 컨버터를 통해 디지털 신호로 변환하고, 조도센서의 센싱 데이터를 이용하여 밝기에 따른 패턴제어를 하여, 불필요한 에너지 소모를 방지하고 LED 조명기기의 효율을 높인다.

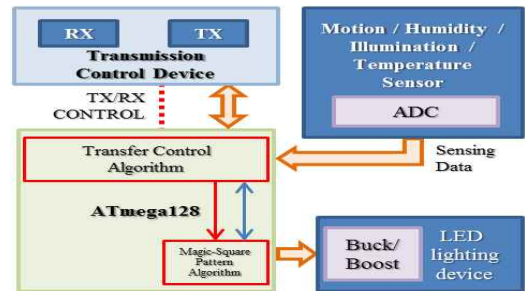


Fig. 1. Intelligent LED lighting system structure

그림 1. 지능형 LED 점등시스템 구성도

센서를 통하여 센싱된 아날로그 데이터를 MCU의 입력으로 전송하여 디지털 데이터로 변환 후, 알고리즘에 의하여 기존 데이터와 비교하고, 점·소등을 위한 데이터 전송제어와 조명기기 점등패턴을 제어한다. 이 과정에서 센싱 데이터에 대한 인터럽트 발생시, 알고리즘에 의해 조명기기의 밝기와 패턴에 변화를 주게 한다. 그림 2는 지능형 LED 점등 시스템에서 점등제어 알고리즘을 이용한 점등제어 처리과정을 나타내었다.

점등제어를 위한 데이터의 처리는 사용할 포트와 플래그, 타이머를 초기화하고, 점등패턴의 초기값을 설정하여 Y축 스캐닝 라인을 지정한다. 그리고 조도 변화에 의한 외부 인터럽트의 발생 여부를 판단하여, 인터럽트 발생시 Y축 스캐닝라인 지정 값을 초기화

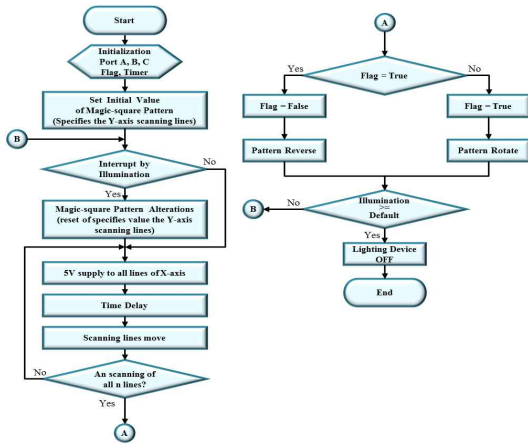


Fig. 2. Data transmit control algorithm
그림 2. 데이터 전송제어 알고리즘

하고 점등패턴을 변경한다.

모든 X축의 라인에 5V를 인가하여 점등을 시키고 초기화된 타이머 시간만큼의 시간지연을 하여 스캐닝 라인을 이동한다. 이때, 모든 라인을 스캐닝 할 때까지 라인 스캔을 반복한다. 라인 스캐닝이 완료되면 플래그 값을 검사하여 true면 false로 변경하여 패턴을 반전시키고, false면 true로 변경하여 패턴을 회전시킨다. 조도센서의 센싱값이 점등을 위한 기본 값 이상이 될 때 까지 인터럽트 발생여부 판단 단계부터 반복하고 이상이 되면 LED 조명 기기를 소등한다.

센서 네트워크에서 발생하는 데이터는 센싱 데이터의 변화량이 시간변화에 비하여 크지 않고, 인접한 노드들의 데이터 값은 유사한 값을 가지게 된다. 또한, 이벤트가 발생할 경우에 데이터의 변동률이 크기 때문에 점등제어 알고리즘의 사용이 필요하다.

점등제어 알고리즘은 조도센서와 연동하여 아날로그 신호를 디지털 신호로 변환하고, 빛의 밝기 변화에 따른 패턴제어를 한다. 변환된 주변의 빛이 기본 값 이하일 때까지 LED는 계속 점등을 하게 되고, 이상이 되면 조명은 자동 소등되어 모든 프로세스를 종료하고 불필요한 에너지 소모를 방지하는 LED 조명 기기의 에너지 효율을 높인 방법이다. 그림 3의 점등제어 알고리즘은 미리 설정해 놓은 타임 스케줄에 따라서 시스템의 상태를 결정하고 동작한다.

가로등에 설치한 조도센서로 광량을 측정하고, 동작감지 센서를 통해 감지를 시작하여 데이터 버스를 통해 MCU로 전달한다. MCU에 있는 제어 시스템은 보정을 통해 미리 설정된 제어방식에 따라 신호 값을 이용하여 조광레벨을 계산하고, 결과를 비교하여 레벨유지 여부를 결정한다. 이때, 광량 감지와 동작감지

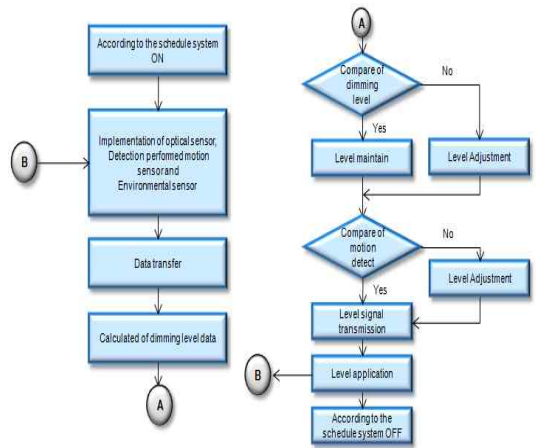


Fig. 3. Lighting control algorithm
그림 3. 점등제어 알고리즘

에 따른 데이터를 기존 측정값과 비교하여 레벨 조정 여부를 결정하게 된다. 따라서 효율적으로 데이터와 제어신호 송수신을 위해, 센서 데이터 처리를 위한 점등제어 알고리즘을 적용한다.

그림 4는 점등패턴제어 알고리즘을 나타내었다. 주변의 광량에 의한 인터럽트 발생 여부를 확인하여 기준 이하면 LED의 개수를 패턴을 적용하여 50%, 70% 단위로 점등하고, 광량이 거의 없으면 100%의 LED를 점등한다. 이때, 지정된 플래그 값을 다시 지정하여 점등패턴을 반전 및 회전을 시키고 측정 조도가 기준 이상이면 조명 기기를 소등한다.

점등패턴의 초기값 설정에 따른 포트 A의 출력 값은 달링톤 어레이(소스)를 거쳐 점등을 위한 Y축 라인의 값들을 받고, 조도에 의한 인터럽트 발생 여부를 확인하여 변화가 있을 경우, 기준 값과 비교하여 패턴을 50%, 70%, 100%로 변경하고, 패턴을 재설정하여 Y축 스캐닝 라인을 설정한다. 조명기기의 LED 모듈 선택을 위한 포트 C의 출력 값을 디코더를 통해 변환하여 달링톤 어레이(싱크)를 거쳐 X축 라인의 선택 값들을 받는다. 그리고 초기 설정된 시간만큼 지연하고 포트 C의 출력 값을 변화시켜 Y축에 대한 스캐닝 라인을 이동한다. 포트 A의 출력 값에 대한 Y축의 모든 라인을 스캐닝 할 때까지 포트 C의 출력 값을 변화시켜 LED 모듈 선택부터 반복한다. 이때, 현재 플래그 상태를 검사하여 참이면 값을 반전시켜 재설정하고 패턴의 반전에 의한 패턴 값을 재설정한다. 거짓이면 플래그 값을 참으로 변경하고 패턴 회전에 의한 값을 재설정 한다. 포트 F를 통한 조도 측

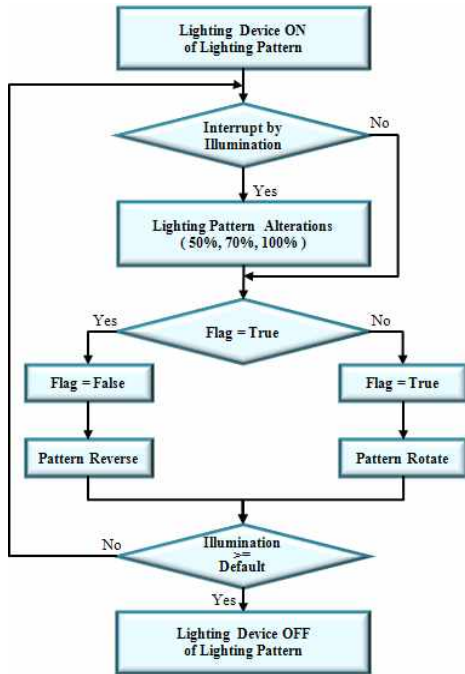


Fig. 4. Pattern control algorithm
 그림 4. 패턴제어 알고리즘

정 변환 값을 점등을 위한 기본 설정 조도와 비교해서 기본 값 이하이면 인터럽트 발생 여부 확인 단계로 이동하여 반복 수행하고, 기본 값 이상이면 조명기기를 소등시키고 프로세스를 종료한다.

패턴제어 알고리즘을 적용하여 지능형 LED 조명기기를 효율적으로 제어하기 위한 데이터 흐름을 그림 5에 나타내었다.

그림 6은 LED 조명기기를 지능형으로 제어하기 위해 ADC 채널이 다중 센서 플랫폼의 센서로부터 데이터를 획득하는 프로그램을 나타내었다. 다중 센서

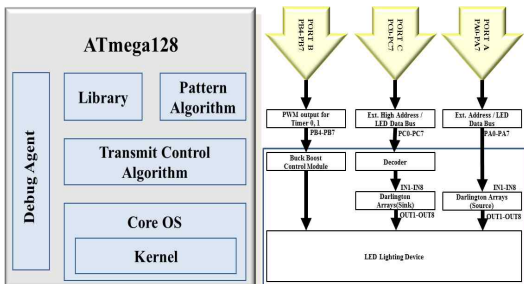


Fig. 5. LED Lighting device dataflow
 그림 5. LED 조명기기 데이터 흐름

의 센서 노드로 부터 입력되는 센싱값의 변화에 따라서 각각 센싱값을 입력받고 ADC를 통해 디지털 데이터로 변환한다. 프로그램 일부를 변경하여 온도 및 다중센서를 이용한 제어도 가능하다. sel5 selection에서는 ATmega128과의 인터페이스에서 센싱값을 추출하기 위한 제어단자를 선택하고 래치를 활성화시켜 입력되는 센싱값을 저장한다. 이때 래치가 비활성화 상태면 이전의 센싱값을 유지시킨다. adc selection에서는 제어단자로부터 입력되는 센싱값을 사용하기 위해 포트 A의 데이터버스를 선택한다. 데이터버스를 선택하기 위한 AD2부터 AD0의 값은 0x01이면 ADC0은 cds_adc 값을 추출하고, 0x02이면 lm35_adc 값을 추출한다. 또한 0x03과 0x00이면 motion_adc와 voice_adc값을 추출한다. adc_value

```

void cds_adc_f(void) {
    unsigned char i=0;
    int k,j,m,n;
    // sel5 selection
    // cds adc selection
    // adc_Value
}

void lm35_adc_f(void) {
    // sel5 selection
    // lm35 adc selection
    // adc_Value
}

void motion_adc_f(void) {
    // sel5 selection
    // motion adc selection
    // adc_Value
}

void voice_adc_f(void) {
    // sel5 selection
    // music adc selection
    // adc_Value
}

int main(void) {
    unsigned int i;
    mcu_init();
    main_set();
    while (1) {
        cds_adc_f();
        lm35_adc_f();
        motion_adc_f();
        voice_adc_f();
    }
}
    
```

Fig. 6. Sensing data conversion & control program
 그림 6. 센싱 데이터 변환 및 제어 프로그램

에서는 각각의 센서로부터 측정된 센싱값을 디지털 값으로 변환하여 LED 조명기기 점등제어에 사용하고 디스플레이에 활용할 수 있도록 처리한다.

그림 7은 제안한 점등제어 알고리즘을 통해 센싱 데이터를 전송하기 위한 프로그램을 나타내었다. 센싱 데이터의 변동률이 지정된 값보다 작을 때, 이벤트의 발생이 없는 것으로 처리한다. 이때, 이벤트가 반드시 싱크 노드에 전송되는 것은 아니므로 단일경로를 유지하여 데이터가 전송 된다. 하지만 데이터 변동률이 기준 값보다 큰 경우에는 이벤트의 타깃 전달율을 만족하기 위해 h개의 독립경로를 통해 데이터를 전송하게 된다.

```

procedure DATA_TRANS()
  h ← node_id;
  data_n ← sensing events;
  repeat
    data_c(h) ← receive(node_c(h));
    data_n ← data of aggregation(data, data_c(h));
    update data variation(dvarh) at nodeh;
    check threshold conditions;
    update Touth;
  until Touth has expired
  if (event occur) then
    send data_n to each of the h parents node;
  else
    send data_n to the primary parent node;
  endif
  sleep until the next period;
end DATA_TRANS
    
```

Fig. 7. The transmit control algorithm
그림 7. 전송제어 알고리즘

그림 8은 제안한 점등제어 알고리즘의 검증을 위해 점등시스템에 적용하여 작성한 패턴에 의한 점등제어 프로그램을 나타내었다.

본 논문에서는 제안한 알고리즘을 적용하고 LED 점등 시스템을 패턴을 이용하여 점등제어를 하기 위해 가로·세로 $n \times n$ 의 1부터 n^2 까지 정수를 한번 씩만 사용하여 정사각형 모양으로 나열하고 가로·세로·대각선의 합이 모두 동일해 지도록 만든 순열 알고리즘을 이용한 방진(magic square)을 사용하였다. 표 1은 8*8형태의 방진을 적용한 배열의 형태를 나타내었고 각각의 행과 열의 합이 260으로 동일하다.

점등제어를 위한 패턴에 사용한 방진은 2차원 배열형태 값에서 가로, 세로, 대각선 중 어느 곳으로 더해도 일정한 값이 나오게 하는 방법을 이용하여 일반형이 $4 \times n + 2$ 형태를 유지하도록 하였다.

```

void Pattern_control_Algorithm {
  init_port(); // I/O port initialization
  init_adc(); // AD converter initialization
  init_flag(); // Flag initialization
  init_timer(); // Timer initialization
  PORTC = StartOfLed(); // LED on
  while (std_intensity < def_intensity) {
    if (int_light > std_light) {
      alt_pattern();
    }
    if(flag=1) {
      flag = 0;
      Time_delay(); // Time delay
      LED_Reverse();
    }
    else {
      flag = 1;
      Time_delay(); // Time delay
      LED_Rotate();
    }
  }
  PORTC = EndOfLed(); // LED off
}
    
```

Fig. 8. Pattern lighting control program
그림 8. 패턴 점등제어 프로그램

Table 1. Two-dimensional array of 8*8 magic square
표 1. 8*8 방진의 2차원 배열

	1열	2열	3열	4열	5열	6열	7열	8열
1행	63	62	4	5	59	58	8	1
2행	10	11	53	52	14	15	49	56
3행	18	19	45	44	22	23	41	48
4행	39	38	28	29	35	34	32	25
5행	31	30	36	37	27	26	40	33
6행	42	43	21	20	46	47	17	24
7행	50	51	13	12	54	55	9	16
8행	7	6	60	61	3	2	64	57

III. 실험 결과

점등제어를 위한 전송제어 프로그램의 실험환경은 네트워크 시뮬레이션 도구 NS-2를 사용하여 실험하였다. 그림 9는 제안한 지능형 LED 점등시스템을 위한 점등제어 알고리즘을 적용하여 이벤트 발생에 따른 데이터 전송률 측정을 위한 트리기반 센서노드의 실험 환경을 나타내었다. 다중센서 노드의 수의 변화로 네트워크 규모를 조절하였고, 노드와 노드 사이의 범위는 20m로 제한하였다. 주기적으로 싱크노드에 센싱 데이터를 전송하도록 설정하고 센서노드의 임의지

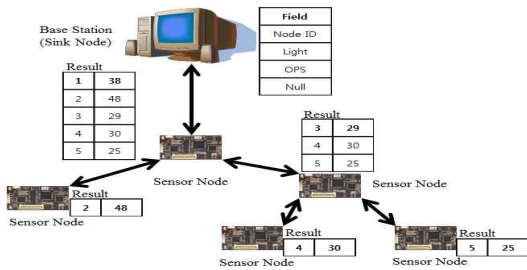


Fig. 9. Sensor node topology

그림 9. 센서노드 구성

역에서 중요 이벤트가 발생하도록 설정하였다.

그림 10은 그림 9의 센서노드 4에서 발생한 조도에 대한 이벤트 측정값을 그래프로 나타내었다. 노드 5의 측정 데이터는 노드 3을 통하여 노드 1로 전달되고, 노드 2의 측정 데이터는 노드 1로 바로 전달되었다. 노드 4에서 이벤트 발생 시, 센싱 데이터가 노드 3과 1을 통하여 노드 1로 전송되는 것을 측정하였다. 결과적으로 이벤트가 발생하여도 이웃한 센서 노드를 통하여 전달되기 때문에 센싱 데이터의 변동률이 기준 값보다 큰 경우에는 제한한 점등제어 알고리즘



Fig. 10. Sensor node event measurement

그림 10. 센서 노드 이벤트 측정

Table 2. Data transfer rate

표 2. 데이터 전송률

	50 (ea)	100 (ea)	150 (ea)	200 (ea)	Average (%)
single path	54	53	50	46	51
multi path	94	90	87	86	89
path repair	82	80	82	80	81
independent path	92	91	91	90	91

을 통해 정상적인 데이터의 전송이 이루어지는 것을 확인 할 수 있었다.

표 2는 데이터 전송 경로 별 센서노드 수는 50개에서 200개까지 증가시켜 평균 전송률을 비교하여 나타내었다.

단일경로와 다중경로의 경우는 전송률이 54%에서 46%, 94%에서 86%로 전송률이 각각 8%가 감소하였고, 경로복구와 독립경로의 경우는 82%에서 80%, 92%에서 90%로 각각 2% 감소하였다. 결론은 독립경로의 경우에 데이터 전송률이 가장 높게 측정되었다.

그림 11은 데이터 전송 경로 별 센서 네트워크의 임의의 지역에서 발생한 이벤트 전달율의 측정결과를 그래프로 나타내었다.

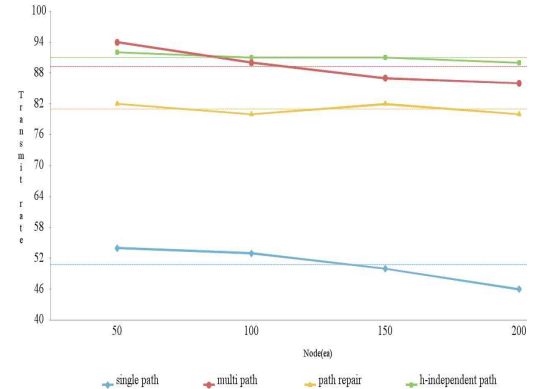


Fig. 11. Important event transmit rate comparison

그림 11. 중요 이벤트 전달율 비교

조도 분포는 평면적·공간적 조도의 확산이나 변화를 의미하는 것으로 패턴 변화에 따른 평균 조도가 동일해도 분포상태가 다르게 보이는 문제를 고려하여 프로그램을 적용하고 실험 하였다.

그림 12는 패턴제어 프로그램을 적용한 LED 모듈 점등상태의 결과를 나타내었다.

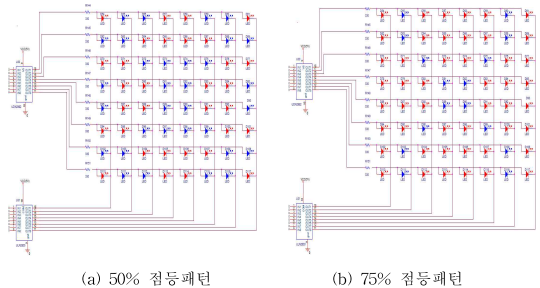


Fig. 12. The lighting pattern control

그림 12. 점등패턴 제어

주변의 조도 값과 기준 수치의 비교를 통해 얻어진 결과를 인터럽트로 처리하여 종류에 따른 점등패턴을 조명기기에 적용하여 LED 모듈의 점등 개수를 조절하였고 패턴을 변화시켰다. 패턴제어 프로그램 실행 결과로 얻어진 패턴으로 LED 조명기기를 점등시켜 상하·좌우 대칭이 완전하게 이루어지도록 하였고, 패턴을 제어하여도 조도를 균일하게 분포시키는 결과를 얻을 수 있었다. 점등패턴의 형태가 완전하게 대칭을 이룰 수 있는 것은 표 1의 8*8방진을 이용하여 LED 모듈별로 가상의 번호를 부여하여 50% 점등패턴이면 홀수 또는 짝수에 해당하는 모듈만 점등하였기 때문이다. 점등패턴의 제어는 패턴의 변화를 주어 디밍 제어가 가능하도록 구현이 가능하다.

그림 13은 주변의 광량과 태양광의 밝기에 따른 LED 동작 개수를 점등제어 알고리즘을 통해 제어하여 LED 조명기기에 적용한 결과를 나타내었다.



(a) 50% lighting (b) 70% lighting (c)100% lighting

Fig. 13. LED lighting device pattern control
그림 13. LED 조명기기 패턴제어

본 논문에서 제안하는 점등제어 알고리즘은 일반적으로 LED 모듈 전체를 점등하는 것이 아니라 점등상태를 회전 및 반전시키기 때문에 소비전력의 절감과 LED 모듈 수명증가, 발열량을 감소시킬 수 있는 결과를 얻었다. 점등제어 알고리즘을 적용한 결과를 패턴에 따른 모듈별 사용 빈도와 동작 시간을 프로그램과 수식(1)을 이용하여 분석하였다.

$$\begin{aligned}
 P_r &= P_m \times P_c \\
 P_t &= P_m \times P_c^2 \\
 T_t &= 8_{hr} \times 60_{min} \times 60_{sec} \\
 T_{pr} &= T_t \div (P_m \times P_c) \\
 L_t &= (P_m \times P_c^2) \times (T_t \div (P_m \times P_c))
 \end{aligned}
 \tag{1}$$

LED 모듈을 매일 8시간동안 점등 시키면 시간 (T_t)은 $28,800_{sec}$ 가 되고, 점등패턴에 의한 LED 모듈의 패턴점등시간(P_m)을 40_{sec} 로 설정하여 분석하였다. 패턴 반복시간(P_r)은 LED 모듈의 패턴점등시간과 패턴이 1회 반복했을 때 패턴변환횟수(P_c)의 승산을 하고, 패턴당 점등시간(P_t)은 패턴변환횟수(P_c)의 제곱에 패턴점등시간을 승산하였다. 전체 패턴의 반복횟수(T_{pr})는 전체 점등시간(T_t)을 패턴점등시간과 패턴이 1회 반복했을 때 패턴변환횟수를 승산하여 결과를 얻었다. 또한, 개별 LED의 점등시간(L_t)은 패턴당 점등시간과 총 패턴의 반복횟수(T_{pr})를 승산하여 결과를 얻었다.

각 LED당 사용빈도는 패턴변화에 따른 LED별 점등 횟수를 개별 증가시켜 분석하였으며, 표 3과 같이 모든 램프를 균등하게 점등하여 동일한 수명을 가지는 결과를 얻을 수 있었다.

Table 3. Frequency in use of a rotation
표 3. 1회전시 사용빈도

	1열	2열	3열	4열	5열	6열	7열	8열
1행	2	2	2	2	2	2	2	2
2행	2	2	2	2	2	2	2	2
3행	2	2	2	2	2	2	2	2
4행	2	2	2	2	2	2	2	2
5행	2	2	2	2	2	2	2	2
6행	2	2	2	2	2	2	2	2
7행	2	2	2	2	2	2	2	2
8행	2	2	2	2	2	2	2	2

표 4와 같이 모든 LED의 점등시간(L_t)이 전체 LED의 점등방식보다 50%로 점등시간이 감소된 결과와 LED 조명기기의 모든 LED의 사용빈도가 동일하여 점등시간도 동일하다는 결과를 얻었다.

Table 4. Day lighting time
표 4. 1일 점등시간

	1열	2열	3열	4열	5열	6열	7열	8열
1행	240	240	240	240	240	240	240	240
2행	240	240	240	240	240	240	240	240
3행	240	240	240	240	240	240	240	240
4행	240	240	240	240	240	240	240	240
5행	240	240	240	240	240	240	240	240
6행	240	240	240	240	240	240	240	240
7행	240	240	240	240	240	240	240	240
8행	240	240	240	240	240	240	240	240

전체 점등 방식을 사용한 기존의 LED 조명기기는 모든 LED 모듈을 점등하기 때문에 60(lm/w)의 광속과 50℃의 발열량, 75W의 전력을 소비하였지만 점등 제어 알고리즘을 적용하여 패턴에 의한 점등제어를 하는 LED 조명기기는 41-60(lm/w)의 광속과 34-43℃의 발열량, 38-75W의 전력을 소비하는 결과를 얻었다. 따라서 전체 점등 방식과 비교하였을 때, 광속과 발열량은 최대 32%, 소비전력은 최대 49%가 각각 감소하였다. LED 모듈 동작 개수가 최대 전체의 50%로 감소된 만큼 광속이 줄었지만, 발열량 감소에 따른 LED 모듈의 수명상승과 에너지 효율 상승 결과를 얻을 수 있었다. 표 5는 64W 기준의 LED 조명기기에서 전체 점등방식과 점등제어 알고리즘을 적용한 점등방식의 비교결과를 나타내었다.

Table 5. A comparison of full lighting in pattern control 표 5. 전체 점등 방식과 패턴제어 점등 방식의 비교

동작패턴	전체 점등제어	패턴 점등제어	증감률(%)
총 LED(ea)	64	64	0 %
동작 LED(ea)	64	32-64	0-50 %
소비전력(W)	75	38-75	0-49 %
발열량(℃)	50	34-43	14-32 %
광속(lm/W)	60	41-60	0-32 %
수명(hr)	50,000	50,000 -75,000	0-50 %

IV. 결론

본 논문에서는 지능형 LED 점등 시스템 구현을 위한 점등제어 알고리즘 설계를 제안했다.

제안된 지능형 LED 점등시스템 구현을 위한 점등 제어 알고리즘은 각각의 센서노드가 환경정보 기반의 중요 이벤트를 감지하여 점등제어를 하고, 데이터 버스를 통해 MCU로 전달하며, MCU의 제어 소프트웨어는 설정된 제어방식에 따라 신호 값을 이용하여 조광레벨을 계산하여 결과를 비교하고, 레벨유지 여부를 결정하도록 하면서 회전 및 반전 기법을 주기적으로 교차 수행하여 패턴을 이용한 점등제어가 되도록 설계 하였다.

제안한 점등제어 알고리즘을 LED 조명기기에 적용한 결과, 안정적으로 점등제어를 하는 지능형 LED 점등시스템을 설계할 수 있었고, 전송 데이터의 신뢰성 향상을 위해 독립경로를 사용하여 평균 91%의 이벤트 전달 비율을 유지하였다. 조명기기의 광속은 32%정도 감소되었으나 고휘도 백색 LED를 사용하여 문제점을 해결하였다. 소비전력과 LED의 내부발열량

은 각각 49%와 32% 감소시켰으며 LED의 수명을 50% 향상시켜 에너지 효율성을 높일 수 있었다. 또한, LED를 각 모듈별로 완전대칭 형태의 패턴으로 점등제어를 하여 균일한 광량과 조도분포를 유지하였다. 결과적으로 조명관련 산업의 지능형 LED 점등 시스템 구현을 위한 실시간 모니터링 및 센싱 기술력 향상과 데이터 전송제어를 통한 안정성 및 신뢰도 향상에 기여할 수 있음을 입증하였다.

제안한 점등제어 알고리즘을 적용한 점등제어 시스템은 MCU를 사용하여 점등제어를 하기 때문에 지그비 통신 방식을 이용한 센서 네트워크를 구성하여 조명기구 주변의 차량 및 보행자의 통행량, 온도, 습도, 조도, 환경정보 등과 같은 다수의 요소를 실시간으로 모니터링을 하여 상황에 따른 점등제어를 할 수 있도록 활용이 가능하다.

향후에는 지능형 LED 점등 시스템의 활용도를 높이기 위해 유비쿼터스 환경 기반의 네트워크를 이용하여 관리자가 간편하게 언제 어디서나 LED 조명기기의 주변상황에 적극 대처할 수 있는 점등제어 방식과 네트워크의 활용으로 조명기기에 지그비 통신 방식을 이용한 다수의 센서노드를 추가하여 조명기기의 주변도로의 이용 빈도에 따른 점등제어 알고리즘과 지형 및 지역 특성을 고려한 점등제어 알고리즘을 개발하여 효율적인 LED 조명기기 관리체계를 확립하는 연구가 수행되어야 한다.

참고문헌

- [1]. 정광성, 김창범, 문철홍, “Xilinx Spartan3 FPGA를 이용한 8bit Processor LED 조명 Board 설계”, *CICS '09 논문집*, pp.331-332, 2009.
- [2]. 지철근, *최신 조명공학*, 문운당, 서울, 2000.
- [3]. 서울반도체, *Z-POWER LED Series Technical Data sheet for W4218X*, www.zled.com, 2007.
- [4]. 손상빈, 여인선, “온도 특성을 고려한 LED 전구의 방열 및 회로 설계”, *전기학회논문지*, 제56권 제7호, pp.1261-1267, 2007.
- [5]. 이미영, 이우희, 이준하, 이홍주, “열화특성을 고려한 LED 조명의 구동회로 설계”, *전력전자학회 2005년도 전력전자학술대회 논문집*, pp.81-83, July. 2005.
- [6]. J. Baek, Y. Nam, and D. Seo, “ATS-DA :Adaptive Timeout Scheduling for Data Aggregation in Wireless Sensor Networks,” *In Proc. Int. Conf Information Networking*, Jan. 2007.

- [7]. C. Karlof, Y. Li, and J. Polastre, "ARRIVE: Algorithm for Robust Routing in Volatile Environments," *ReportNo. UCB/CSD-03-1233*, Mar. 2003
- [8]. M. Shur and R. Zukauskas, "Solid - state lighting : toward superior illumination", *Proceedings of the IEEE*, 98(10), pp.1691-1703, 2005.
- [9]. LU Jiaying, WU Xiaobo, "A PWM Controller IC for LED Driver Used to Multiple DC-DC Topologies", *Power and Energy Engineering Conference*, pp.27-31, 2009.

저 자 소 개

홍 성 일 (학생회원)

2007년 2월 : 세명대학교 컴퓨터
과학과 이학사
2009년 8월 : 세명대학교 교육대학원
교육학석사
2012년 8월 : 세명대학교 대학원
전산정보학과 박사과정 수료 (컴퓨터학
전공)

<주관심분야> SOC CAD, ASIC 설계, CAD 알고리즘,
SOC 설계, RTOS 및 내장형 시스템

인 치 호 (정회원)

1985년 : 한양대학교 공과대학 전자
공학과 공학사
1987년 : 한양대학교 대학원 공학석
사(CAD 전공)
1996년 : 한양대학교 대학원 공학박
사(CAD 전공)
1992년 ~ 현재 : 세명대학교 컴퓨터
학부 교수

<주관심분야> SOC CAD, ASIC 설계, CAD 알고리즘,
SOC 설계, RTOS 및 내장형 시스템