사물 인터넷: 안전, 절약을 위한 아두이노, 블루투스 기반의 스마트 멀티탭

한림대학교 스마트IoT학부 20145150 이상욱

요 약

일상생활에 사용되는 전자제품이 인터넷에 연결 되어 다른 기기, 더 나아가 사람과 소통을 가능하게 하는 사물 인터넷(IoT: Internet of Things)의 시장 규모가 커지고 있다. 본 논문에서는 현재 흐름에 맞춰 사물 인터넷 중 하나 인 스마트 멀티탭에 대한 정보를 제공한다. 스마트 멀티탭은 기존 멀티탭에 릴레이 모듈로 전원을 제어하고 릴레이 모듈과 아두이노 키트와 연결한다. 아두이노에 블루투스 통신 모듈을 부착하여 스마트폰을 사용하여 원격으로 각 멀티탭 구의 전원의 ON/OFF를 제어해 안정성을 추구 하였고 한번에 모든 전원을 끌 수 있게 하여 편리함을 추구하였다.

키워드 : Internet of Things(사물 인터넷), Arduino(아두이노), BlueTooth(블루투스)

1. 서 론

1.1 동기

스마트폰으로 인터넷을 통해 다른 사물을 제어할 수 있는 편리함을 추구하면서 기기들 서로 통신을 통해 새로운 기능을 가지게 하는 사물인터넷의 시대가 확산되고 있다. 본 논문은 확산 되고 있는 사물인터넷이 적용 된 많은 가전제품 중 하나인 스마트 멀티탭에 대한 정보를 제공한다.

1.2 관련기술동향

지난 17년 2월 LG유플러스는 스마트폰으로 여러 개의 가전제품의 전원을 동시에 제어할 수 있는 사물인터넷 멀티탭을 출시 하였고 17년 4월 KT는 스마트폰 앱으로 가전제품 여러 개의 전원을 동시에 손 쉽게 제할 수 있는 ‘기가IoT홈 멀티탭’을 출시 했다고 하였다.

이처럼 사물인터넷이 적용 된 멀티탭의 사용 빈도가 높아지고 있다는 것을 알 수 있다.

1.3 개발필요성

2010년 소방 방재청 통계경과에 따르면 화재 발화 요인에 1순위가 부주의 2순위가 전기적 요인으로 조사되었다. 그 중 전기적 요인의 주요 원인은 합선과 과전류에 의한 발화가 대부분을 차지하였다. 이러한 화재의 원인인 부주의 또는 합선과 과전류를 스마트 멀티탭은 스마트 폰을 통한 원격통신으로 멀티탭의 전원을 ON/OFF 할 수 있게 만들어 화재예방을 가능케 하였다.

조사에 따르면 국내 대기전력의 연간 소모량은 206kWh로 이는 대상가구의 소비전력의 10.6%에 해당한다. 특정 시간대에 사용하지 않는 가전기기의 대기 전력을 차단하기 위해 스마트폰의 어플리케이션에 추가적으로 타이머 기능을 추가해 대기전력 차단함으로써 전력소모량을 줄이게 하였다.

1.4 개선된 내용설명

스마트 멀티탭의 구성은 스마트폰과 블루투스 모듈, 릴레이 모듈과 아두이노 키트가 결합 된 멀티탭으로 구성 되어있다. 블루투스 모듈로 스마트폰 과 멀티탭 상호간의 통신하게 설계 되었고 아두이노 키트로 멀티탭을 제어하고 또 아두이노의 릴레이 모듈을 이용하여 멀티탭의 전기를 제어 할 수 있게 설계되었다. 추가적으로 해당 스마트폰의 사용자만 멀티탭을 끄고 킬 수 있게 보안적인 측면에서 설계되었다.

2.본 론

3.1 고객 기술문

아침에 바쁘게 집을 나서다 보면 가전제품의 전원을 차단을 하지 못하고 나가는 경우가 종종 있다. 이는 화재로 이어질 가능성을 높이는데, 이것을 방지하기위해 스마트폰으로 원격으로 멀티탭의 전원을 차단 할 수 있도록 멀티탭을 개조한다.

3.2 문제정의

3.2.1 개발의 필요성

a. 서론에서 화재 발화 요인 1순위가 부주의 2순위가 전기적 요인인 것으로 보았을 때 원격으로 전원이 차단 된다면 1순위 2순위가 함께 해결 될 수 있는 것으로 보아 스마트 멀티탭 제작은 화재율을 낮추는데 큰 역할을 할 것이다.

b. 사람들이 일상생활을 하면서 많이 사용하는 물건에 사물인터넷을 적용 시킴으로써 앞으로 편리하게 쓰일 차세대 가전제품을 연구하고 개발한다.

3.2.2 고객의 요구

사용자 요구사항: 각 구의 전원을 컨트롤 할 수 있게 제작 그리고 모든 전원을 한번에 컨트롤할 수 있게 제작한다.

3.2.3 제약사항

-가격은 3만원대 이하로 제약을 두었다.

-블루투스 나 WiFi 사용으로만 원격제어

3.2.4 기능설정

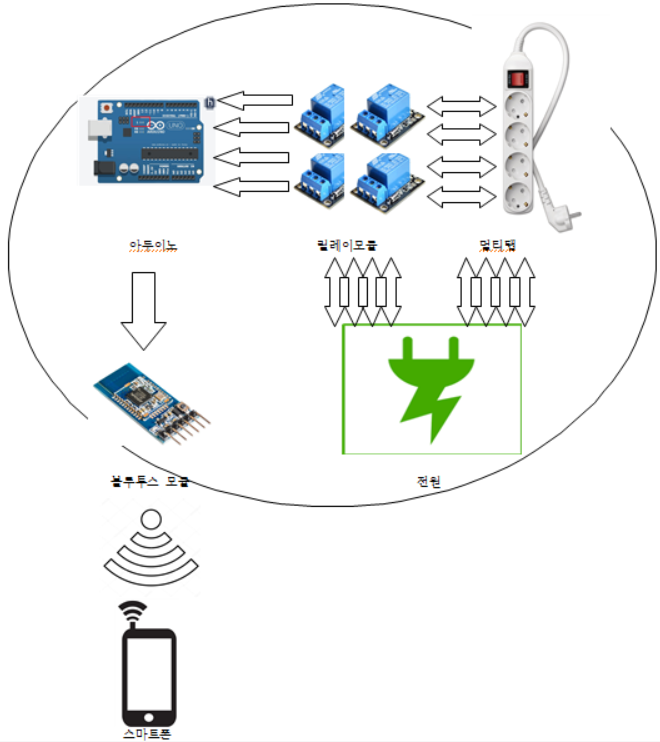


그림1. 사물인터넷 환경의 블루투스 통신을 사용한 스마트 멀티탭의 전체 구조도

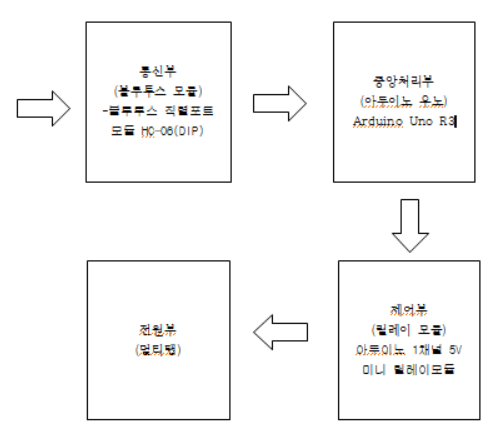
a. 전원 차단 기능 : 아두이노 1채널 5V 미니 릴레이 모듈

b. 통신 기능: 블루투스 직렬포트 모듈 HC-06(DIP)

c. 사용자 환경 기능: 앱인벤터2를 활용한 어플

d. 시스템 제어 기능: 아두이노 (Arduion Uno R3)

그림 2.스마트 멀티탭 시스템의 상위설계도

그림 2의 동작설명은 다음과 같다.

a. 스마트폰에서는 앱인벤터를 통해 전원개수 만큼 ON/OFF 버튼을 만들고 전체 전원 컨트롤 하는 ON/OFF 버튼을 만듭니다. 전원 1의 ON을 누르면 0을 아두이노로 보내주고 OFF를 누르면 1을 보내줍니다. 같은 방법으로 다른 전원의 ON/OFF도 0부터 9까지의 번호를 아두이노로 보내준다.

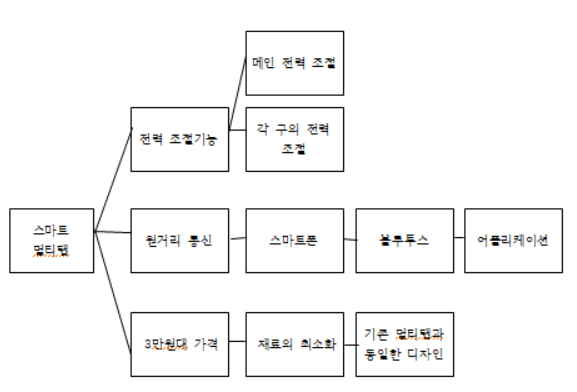
b. 통신부에서는 외부의 스마트폰과 블루투스 모듈로 아두이노와 스마트 폰을 연결한다. 그러면 스마트폰으로 원격제어를 하기 위한 신호가 블루투스 모듈을 통해 중앙처리부인 아두이노 우노로 입력 된다.

c. 중앙처리부에서는 각 구마다 핀번호를 지정해 핀마다 다른 릴레이 모듈을 연결하고 통신부에서 보낸 신호에 맞게 그 전원에 맞는 핀번호의 릴레이 모듈을 동작하게 설계한다.

d. 제어부에서는 아두이노 우노에서 보낸 신호에 맞게 전원에서 멀티탭으로 보내는 전원을 차단,연결 해준다.

e. 전원부에서는 전원에서 바로 오는 전력 과 릴레이 모듈에서 오는 두가지 전력으로 멀티탭의 전기를 사용할 수 있는데 릴레이 모듈을 아두이노로 컨트롤 함으로써 멀티탭의 전원을 조절한다.

그림 3. 스마트 멀티탭 시스템의 목적나무



3.2.5 목적나무 설계

0. 스마트 멀티탭

1. 전력 조절 기능

1.1 메인 전력 조절

1.2 각 구의 전력 조절

2.원거리 통신

2.1 스마트폰

2.1.1 블루투스

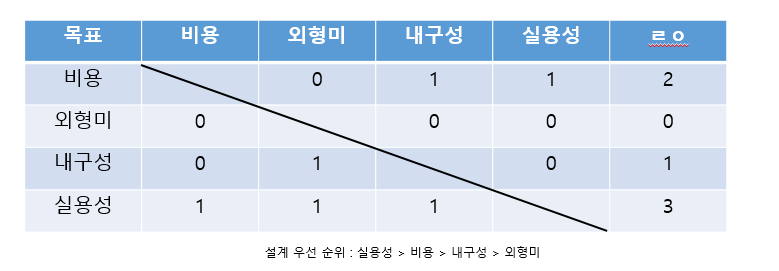
2.1.1.1 어플리케이션

3.3만원대 가격

3.1 재료의 최소화

3.1.1 기존멀티탭과 동일한 디자인

표1. 목표 멀티탭 시스템의 설계 우선순위



설계 우선순위는 실용성을 중점으로 두고 다음은 사람들이 누구나 구입할 수 있게 비용적인 측면을 우선으로 둔다.

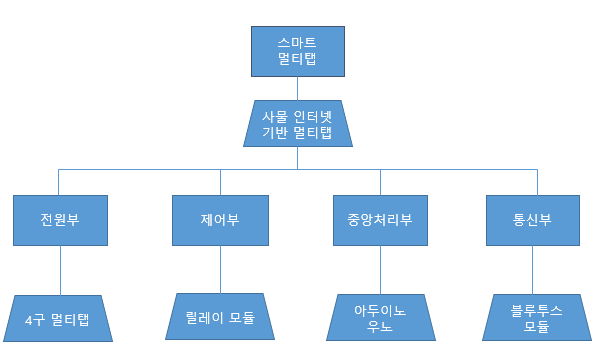


그림 4. 스마트 멀티탭의 기능 수단 나무

3.3 개념 설계

3.3.1 설계 명세

a. 전원부

- 외관은 기본 멀티탭과 동일

- 각 구마다 전원 on/off가 가능하다.

- 전원을 모두 on/off 할 수 있다.

- 콘센트를 기준으로 한 쪽은 릴레이 모듈에, 한 쪽은 외부 전기가 들어 오는 곳에 연결한다.

b. 통신부

- 블루투스 모듈로 아두이노와 스마트 폰을 연결한다.

- 전원은 DC 전압 5V로 아두이노 우 노로부터 공급한다.

- 블루투스 모듈인 HC-06은 아두이노에서 시리얼 통신을 이용하여 데이터 값을 주고 받을 수 있는 모듈이며, 10M 정도까지 무선으로 데이터를 보내거나 받을 수 있다.

c. 중앙처리부

- 범용 온칩 마이크로콘트롤러 개발 보

드 아두이노 우노를 사용한다.

* 전원은 DC 전압 5V로 외부 전력으전부터 아두이노로 공급된다.
* 아두이노(Arduino)는 대중적으로 가장 많이 사용되고 있는 오픈 소스 하드웨어로 다양한 웨어러블 컴퓨터 프로토타입 개발 및 차세대 디지털 기기 발명에 활용되고 있다. 아두이노 우노(Arduino Uno)는 그 중에서도 가장 널리 사용된다.

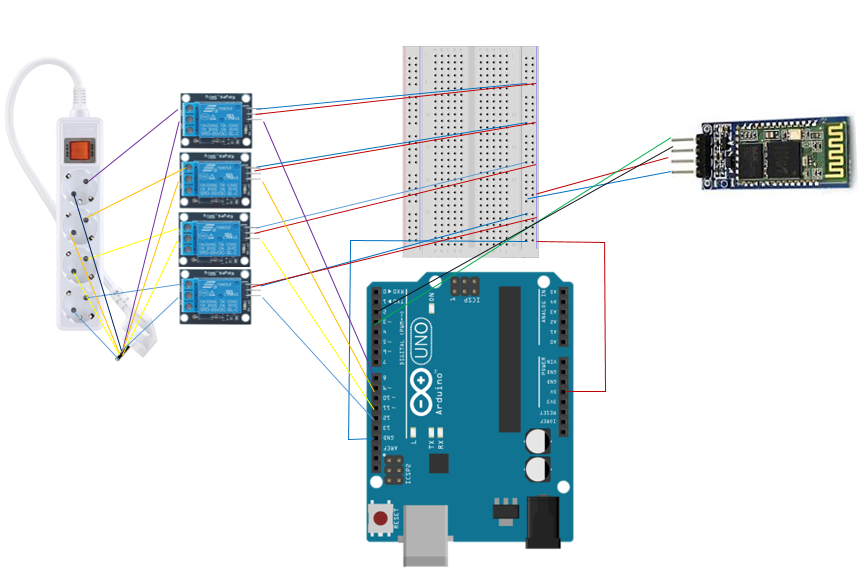
d. 제어부

- 멀티탭과 전원을 차단,연결 하기위해 아 두이노와 멀티탭 사이에 릴레이 모듈을 연결한다.

- 28v 에 240v AC 또는 DC 전원 다른 전기 부품의 모든 종류를 제어한 다. MCU는 타이밍 제어 스위치의 용도로 사용 할 수 있다.

- 각 구마다 전선으로 릴레이 모듈과 연결하고 각 릴레이 모듈마다 외 부 전원과 연결한다. 콘센트 두 구 멍 중 한쪽만 차단하여 전력을 차 차단하는 방식이다.

3.3.1.1 그림5 .내부 배치도



3.3.1.2 그림6. 시스템 외형도

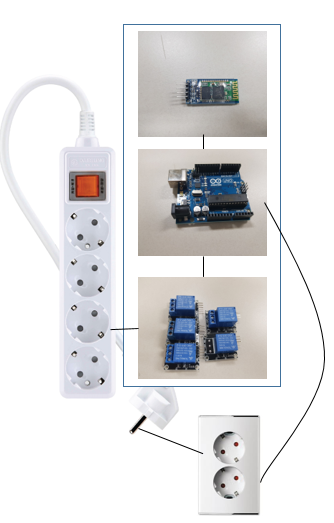
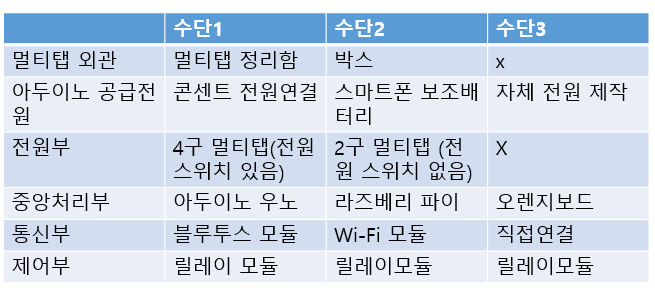


그림 5에서 내부 배치도는 각 장치끼리의 연결을 나타낸다. 우선 멀티탭을 벽 콘센트에 꽂고 멀티탭에 전력을 받는 선 두개 중 하나는 각 구마다 연결한다. 나머지 하나는 릴레이 모듈과 연결하고 각 구와 릴레이 모듈끼리 모두 연결해준다. 그리고 릴레이 모듈을 각 핀번호에 맞춰 아두이노 우노와 연결해준다. 아두이노 우노와 블루투스 모듈를 연결한다.

그림 6에서는 내부배치도와 각 장치끼리의 연결은 같지만 하드웨어 측면에서 볼 수 있다. 멀티탭의 오른쪽에 구멍을 뚫어서 릴레이 모듈과 연결하는 공간을 만들고 나머지 연결 장치들은 박스안에 넣어 멀티탭과 고정시킨다. 아두이노 전원 공급을 위해 전원 공급선을 빼서 멀티탭의 전원과 같이 콘센트에 꽂아 작동한다.

3.3.2 설계 대안



3.4 예비설계 단계

3.4.1 개념설계 구체화

a. 중앙처리부

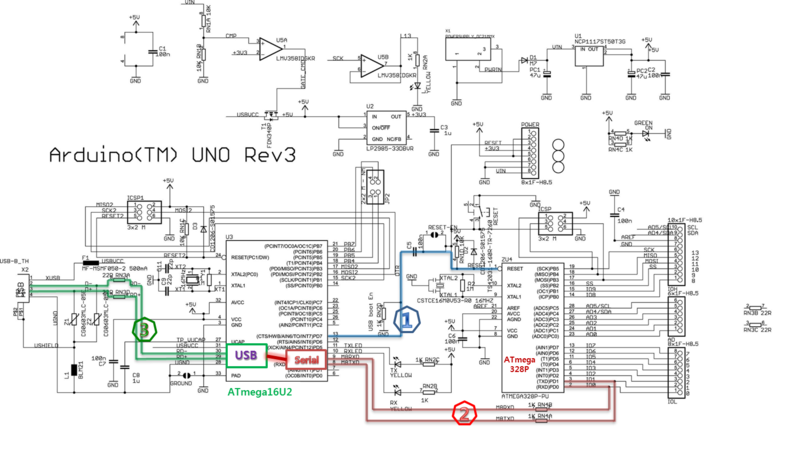


그림 7. 아두이노 우노 보드 상세 설계도 (중앙처리부)

b. 통신부

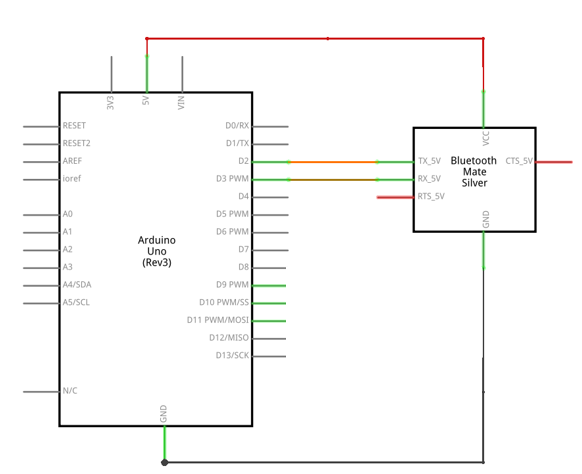


그림 8. 아두이노 우노 와 블루투스 모듈 연결 설계도 (통신부)

c. 제어부

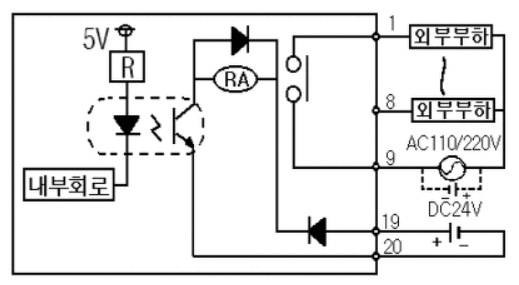


그림 9. 릴레이 모듈 상세 설계도 (제어부)

d. 전원부

멀티탭 전원 과 아두이노 우노 전원은 외부 전원 이용, 릴레이 모듈과 블루투스 모듈은 5V 전원을 사용

3.4.2 원형개발 (시스템 하드웨어)

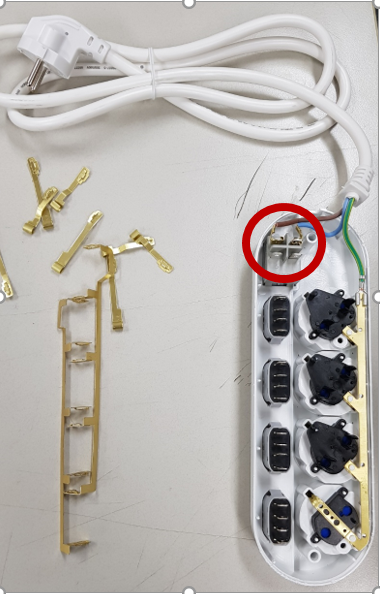


그림 10. 멀티탭 분해 모습 그림10에서 보이는 것과 같이 멀티탭을 분해해 빨간색 동그라미 친 부분(외부 전원에 멀티탭을 연결한 부분)을 한 쪽을 릴레이 모듈과 연결한다. 나머지 한쪽은 멀티탭의 각 구에 연결한다. 또 릴레이 모듈과 각 구에 한쪽을 연결한다. 연결 할 때 절연전선 한 개로는 전압이 약해 전선 두개를 사용해 연결 하다. 릴레이 모듈에 연결할 때 외부 전원에서 오는 전력은 릴레이 모듈에 COM에 연결하고 멀티탭에서 오는 전력은 릴레이 모듈에 ON부분에 연결한다. 각 구 마다 구분 할 수 있게 같은색(보라,주황,노랑,파랑)으로 연결하고 선 끼리 얽히지 않게 선 두개를 절연 테이프로 붙였다. 완성 한 모습은 그림 11이다.

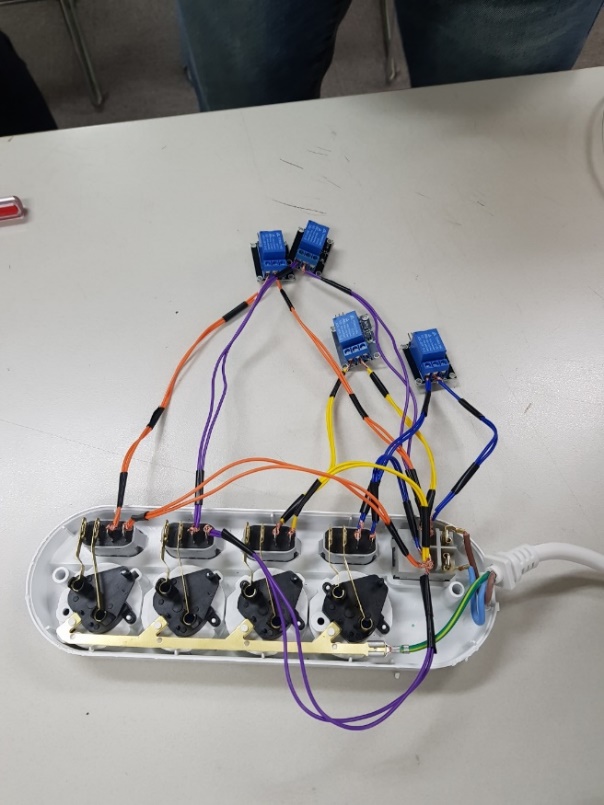


그림 11. 릴레이 모듈과 멀티탭을 연결한 모습

다음으로는 절연전선을 사용하지 않고 범퍼케이블로만 연결한다. 각 릴레이 모듈을 아두이노 우노에 연결한다. 릴레이 모듈이 4개 이기 때문에 아두이노에 연결 공간이 부족하다. 브레드 보드를 사용하여 공간을 확보 후에 연결한다. 브레드 보드의 파란색 선 부분(-부분)에 GND를 연결하여 그 줄을 모두 GND 줄로 만들고 빨간색 선 부분(+부분)에 5V를 연결하여 그 줄 모두 5V줄로 바꾼 후 릴레이 릴레이 모듈의 GND와 VCC(5V)를 연결해 준다. 그리고 아두이노에서 설정할 핀번호 맞게 SIG를 연결한다.(8,9,10,11 핀번호 사용) 이때 선이 헷갈리지 않게 GND는 검정색 VCC는 빨간색 SIG는 릴레이 모듈과 멀티탭을 연결한 색으로 연결하였다. 아두이노는 외부전원과 전원을 연결한다. 그림 12는 아두이노를 브레드보드를 사용하여 릴레이 모듈 4개와 연결한 사진이다.

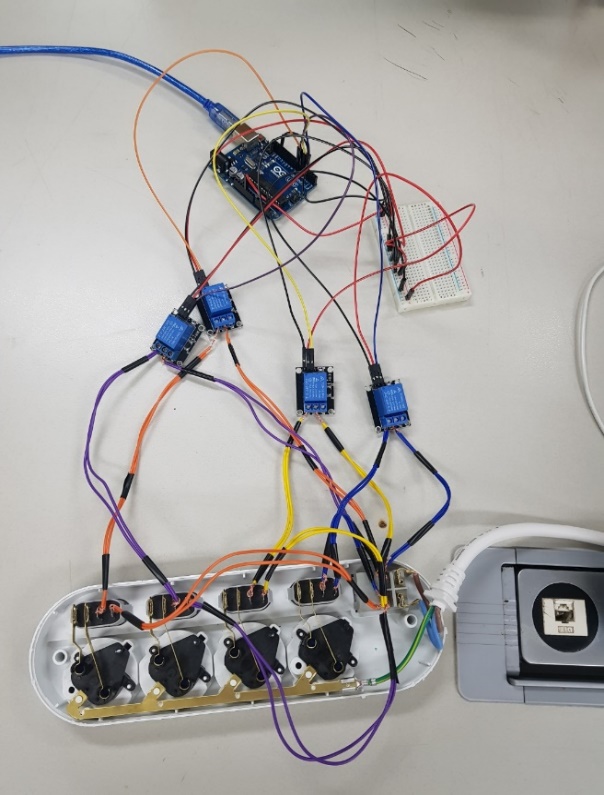


그림 12. 아두이노를 브레드 보드를 사용하여 릴레이 모듈과 연결한 모습

아두이노 우노에 블루투스 모듈을 연결한다. 블루투스 모듈의 GND와 VCC 는 브레드 보드의 GND와 VCC에 연결한다. 그리고 RXD와 TXD가 있는데 RXD는 아두이노 3번 핀에 TXD는 아두이노 2번 핀에 연결한다. 그림 13이 하드웨어 완성품이다.

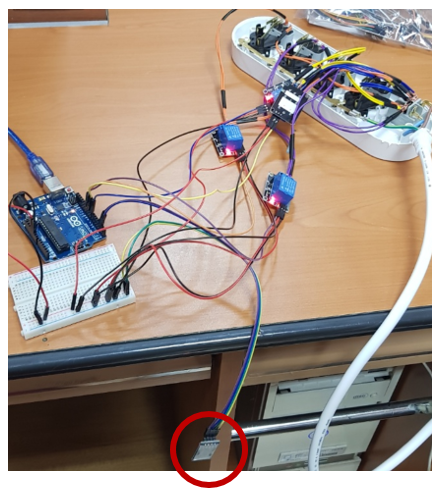


그림 13. 블루투스 모듈을 아두이노와 연결한 사진

3.4.3 원형개발 (소프트웨어)

아두이노 사용을 위하여 아두이노를 아두이노 스케치 프로그램을 사용하여 코딩한다. 2번,3번 핀을 블루투스의 TXD,RXD로 설정한다. 블루투스와 아두이노를 연결 하기 위한 코드를 적고 블루투스를 통해 받을 데이터를 0~9로 정한다.



표 3 데이터에 따른 멀티탭의 ON/OFF

그리고 표3과 같이 아두이노에 입력되는 데이터 마다 릴레이 모듈을 컨트롤해서 각 구의 전력을 차단, 연결하게 설정해준다.

그리고 스마트 폰에서는 앱인벤터2를 사용해 아두이노를 조작할 어플을 만들어준다.

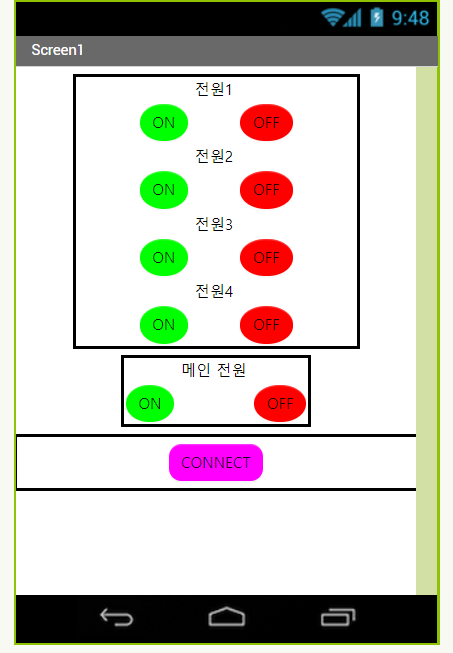


그림 14. 아두이노를 조작할 리모컨 어플

그림 14와 같이 아두이노를 조작할 리모컨 어플을 만든다. CONNECT버튼을 눌러 블루투스 모듈과 연결하고 각 전원의 ON/OFF버튼으로 전력을 차단,공급한다. 메인전원의 ON/OFF로 모든 구의 전력을 차단, 공급하게 코딩한다. 버튼마다 아두이노로 보내는 데이터는 표3과 동일하게 코딩한다. 그림15는 앱인벤터2로 리모컨을 코딩한 사진이다.

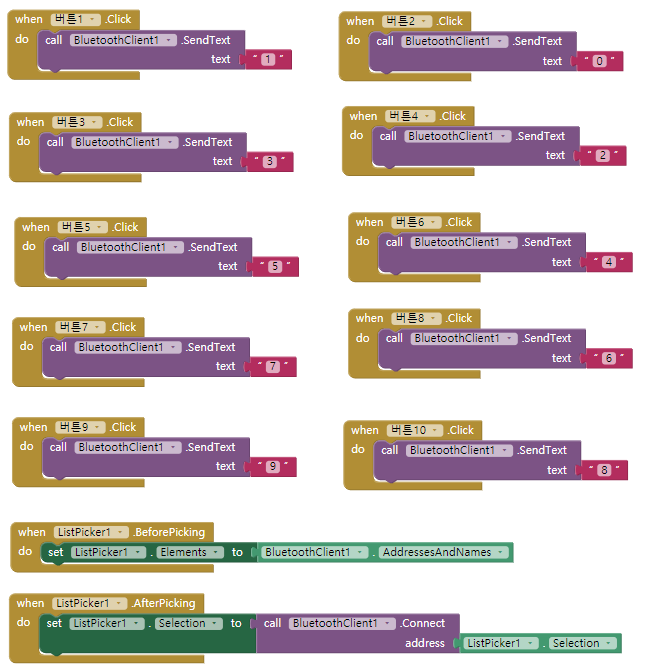


그림 15.앱인벤터2로 리모컨을 코딩한

사진

코딩을 완료한 후 MIT AI2 Companion 어플로 만든 어플의 QR코드를 읽어와 스마트 폰에 설치한다. 어플로 블루투스를 연결할 때 스마트폰 자체의 블루투스가 켜져있고 블루투스 모듈을 읽어 올 수 있어야지만 어플로도 연결을 할 수 있다.

3.4.4 개념검증시험

스마트 멀티탭에서 나온 릴레이 모듈, 아두이노 우노, 블루투스 모듈을 박스에 정리해서 넣고 박스를 멀티탭에 절연테이프로 고정시켜 그림 16과 같이 개발을 완료 시켰다. 제대로 작동을 하기 위한 실험을 위해 멀티탭의 전원과 아두이노의 전원을 외부 콘센트에 꽂아 전원을 공급 시키고 스마트폰으로 블루투스를 켜 블루투스 모듈과 연결시킨다. 앱인벤터2로 만든 어플을 들어가 CONNECT를 눌러 블루투스 모듈과 어플을 연결하고 전원을 하나씩 눌러

멀티탭의 전원을 차단,공급이 제대로 작동 되는지 확인한다. 어플에서 전원1 ON을 누르면 1을 블루투스 모듈을 통해 아두이노로 보낸다. 아두이노에서는 1을 받으면

전원1에 연결된 릴레이 모듈에 전원을 연결하라는 명령을 보낸다. 그러면 전원1에 연결된 릴레이 모듈에 불이 들어오면서 멀티탭의 구 전원 버튼에 불이 들어온다. 멀티탭 전원 콘센트에 코드를 꽂아서 전원이 잘 통하는지 실험한다. 모든 전원의 버튼을 누르고 차단,공급이 되는지 실험하고 마지막으로 모든전원 ON/OFF버튼을 눌러 모든 전원이 켜지고 꺼지는지 실험한다.



그림 16. 개발 완료 된 스마트 멀티탭을 실험하는 사진

4.결론

IoT 시대에 맞게 낭비되는 전력을 효율적으로 제어하기 위하여 실시간으로 전원을 제어 할 수 있는 스마트 멀티탭을 제작하였다. 이를 통해 전력소모를 최소화함으로써, 전력을 효율적으로 사용 할 수 있는 시스템의 방식을 제안하였다. 제안한 시스템은 스마트 멀티탭의 블루투스와 스마트폰과의 통신을 통해 정보를 공유하고 멀티탭을 제어하는 시스템이다. 스마트 멀티탭은 스마트폰에서 변경된 정보를 1초 안으로 스마트 멀티탭에 전달하여 멀티탭 전원을 관리하여 화재에 대한 예방을 할 수 있다는 것을 확인하였다. 또한 멀티탭을 제어할 수 있는 사용자는 어플이 설치되어 인가된 사용자만 제어할 수 있게 설계되어 인가되지 않은 사용자의 무분별한 사용을 제한하였다.

참고 문헌

[1] 김홍석, 나재환, 박소현, 곽수영, ‘원격 모니터링 및 제어가 가능한 와이파이 스마트 콘센트,2014

[2] 이정혁, 김상현, 오창세, 서민석, 김영돈, 박현주. ‘Zigbee 통신 기반 스마트 멀티탭 시스템 구현’, 2014

[3] 전정우, 이미라, ‘대기전력 감소를 위한 와이파이 기반의 스마트 멀티탭 시스템, 2017

[4] 왕승민, 김요셉, 오세인, 한병호, 허헌, 이양희, 장홍수, ‘멀티탭을 활용한 스마트 미터링 시스템 개발’, 2011

[5] 허태성, 이석원, 이도영, ‘스마트 원격제어 멀티탭 시스템’, 2017

[6] 이세훈, 이강복, 민창경, 김재승, 이윤수, ‘클라우드 웹 서비스 기반의 전기절약을 위한 스마트 멀티탭 개발’, 2017

[7] 민주성, 이소현, 송인우, 김옥주, 최윤걸, 정요한, 안종석, ‘홈 네트워크상의 홈 오토메이션을 위한 스마트 멀티탭’, 2012

부록

1. 상위설계도 : 그림 2로 대체함
2. 상세 설계도 : 상위 설계도의 각 모듈에 대한 상세 회로도, 그림 7,8,9로 대체함
3. 배치도: 인쇄 회로기반(PCB)상의 부품 배치도, 외형물상의 커넥터 배치도, 그림 5,6으로 대체함
4. 조립도 : 모듈상의 연결도, 그림 5로 대체함
5. 소프트웨어 프로그램 : 아두이노 우도 구동 프로그램

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); //블루투스의 Tx, Rx핀을 2번 3번핀으로 설정

int one = 8;

int two = 9;

int three = 11;

int four = 10;

int input;

void setup() {

pinMode(one,OUTPUT);

pinMode(two,OUTPUT);

pinMode(three,OUTPUT);

pinMode(four,OUTPUT);

// 시리얼 통신의 속도를 9600으로 설정

Serial.begin(9600);

while (!Serial) {

; //시리얼통신이 연결되지 않았다면 코드 실행을 멈추고 무한 반복

}

Serial.println("Hello World!");

//블루투스와 아두이노의 통신속도를 9600으로 설정

mySerial.begin(9600);

}

void loop() { //코드를 무한반복합니다.

if (mySerial.available()) { //블루투스에서 넘어온 데이터가 있다면

input = mySerial.read();

}

if (Serial.available()) { //시리얼모니터에 입력된 데이터가 있다면

mySerial.write(Serial.read()); //블루투스를 통해 입력된 데이터 전달

}

if(input == '0') {

digitalWrite(one,LOW);

}else if(input == '1') {

digitalWrite(one, HIGH);

}else if(input == '2') {

digitalWrite(two, LOW);

}else if(input == '3') {

digitalWrite(two, HIGH);

}else if(input == '4') {

digitalWrite(three, LOW);

}else if(input == '5') {

digitalWrite(three, HIGH);

}else if(input == '6') {

digitalWrite(four, LOW);

}else if(input == '7') {

digitalWrite(four, HIGH);

}else if(input == '8') {

digitalWrite(one,LOW);

digitalWrite(two,LOW);

digitalWrite(three,LOW);

digitalWrite(four,LOW);

}else if(input == '9') {

digitalWrite(one, HIGH);

digitalWrite(two, HIGH);

digitalWrite(three, HIGH);

digitalWrite(four, HIGH);

}

}

1. 부품목록

|  |  |
| --- | --- |
| 품목명 | 모델명 |
| 아두이노 | Arduion Uno R3 |
| 블루투스 모듈 | 블루투스 직렬포트 모듈 HC-06(DIP) |
| 멀티탭 | SAFE 멀티탭 4구 개별 접지 1.5m |
| 릴레이 모듈 | 아두이노 1채널 5V 미니 릴레이모듈 |
| 브레드 보드 | WANJIE 브레드보드 801 |
| 절연전선 | Any Vendor 10색 절연전선-1M |
| 절연테이프 | Any Vendor 전기용 절연테이프 19mm\*10m(검정) |
| 점퍼 케이블 M/F | 소켓 점퍼 케이블 40P (칼라) (M/F) |
| 점퍼 케이블 M/M | SZH 테스트[CH254] 소켓 점퍼 케이블 40P (칼라) (M/M) 20cm |

1. 주요부품 데이터시트

-아두이노 우노 :

http://www.fecegypt.com/uploads/dataSheet/1522237550\_arduino%20uno%20r3.pdf

-릴레이 모듈

http://www.fecegypt.com/uploads/dataSheet/1480848003\_2\_channel\_5v\_10a\_relay\_module.pdf

-블루투스 모듈

https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf