

아두이노 키트 공모전

프로젝트 명

<나만의 리모컨>

융합특성화자유전공학부

20211806 이소현

2021.11.18

<목 차>

1. 프로젝트 개요	1
2. 프로젝트 계획	1
3. 프로젝트 진행 과정	4
4. 프로젝트 결과	9
5. 참고 문헌	9

1. 프로젝트 개요

1-1 프로젝트 개발 배경

무선통신으로 사물을 제어하는 IoT기술을 모방하여 적외선 센서&리모컨과 같은 센서를 통해 사물을 제어하고자 했다.

1-2 프로젝트 최종 목표

리모컨을 통해 전등 스위치를 제어하고, 사용자에게 알람이 울리길 원하는 시간을 입력 받는다. 알람이 울리기 시작하면 사용자가 일어나 기기 가까이에 와야만 자동적으로 꺼진다. 리모컨을 통한 사물 제어와 알람시계 기능을 구현한다.

2. 프로젝트 계획

2-1 프로젝트 세부사항

① 리모컨

적외선 수신 센서와 리모컨을 통해 사용자가 원하는 기능을 선택하여 작동하게 할 수 있다.

② 전등 스위치 제어

서보모터를 이용해 회전 각도를 조절하여 스위치를 제어한다.

③ 알람 설정 & 리셋

사용자에게 몇 분 후에 알람을 울릴 것인지 분 단위로 입력 받고, 해당 시간이 지나면 알람을 울린다. 숫자를 잘못 입력하였을 경우 리셋 기능을 통해 지금까지의 입력을 지우고 새로 입력을 받는다.

④ LCD

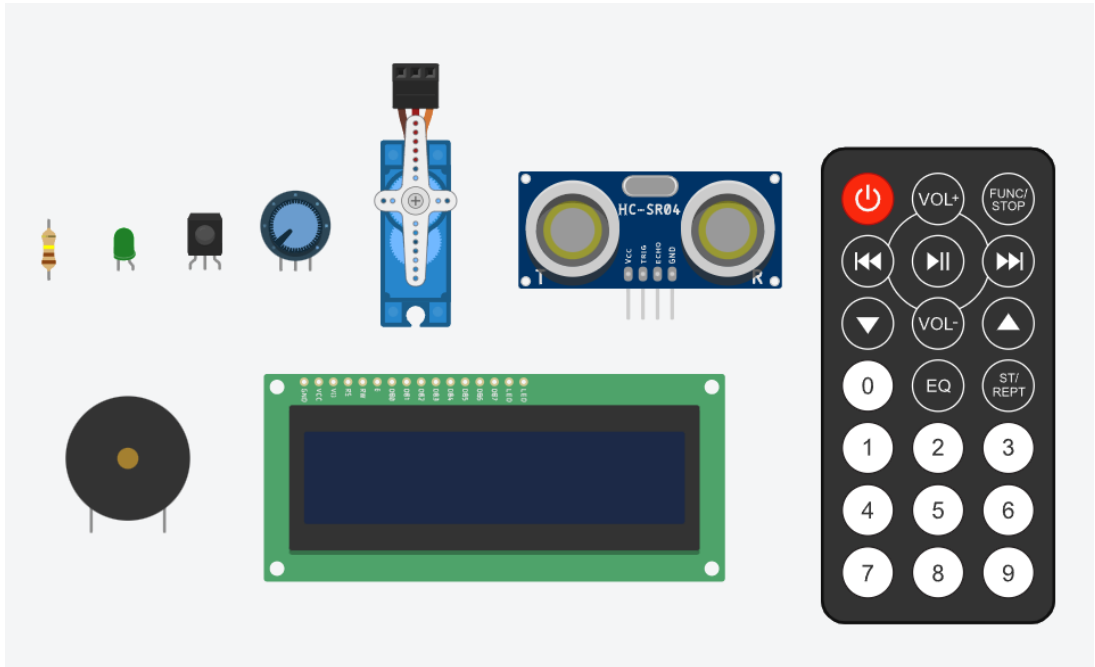
불을 켜면 'LIGHT ON' 과 같은 문자를 출력하여 사용자가 현재 작동되는 기능이 무엇인지 직접 눈으로 확인할 수 있다.

⑤ LED

LCD가 안 보일만큼 멀리서도 리모컨 입력이 정상적으로 되었는지 사용자가 알게끔 한다.

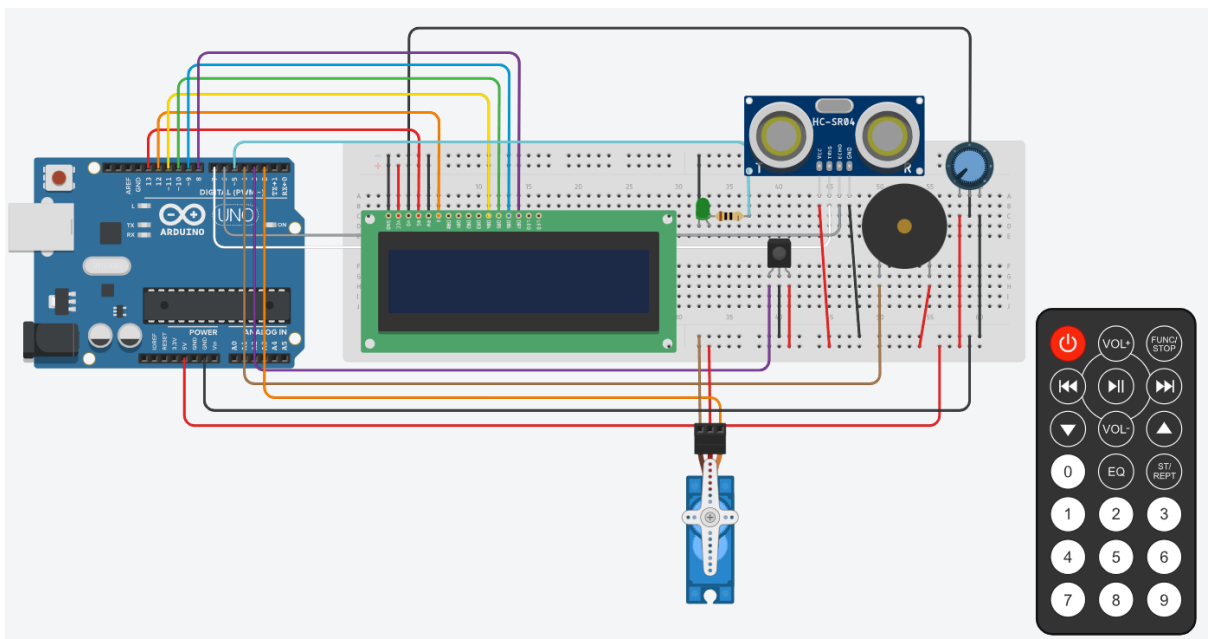
2-2 시스템 구조도

준비물



100~220Ω 저항, LED, IR센서&리모컨, 가변저항, 서보모터, 초음파 센서, 수동 부저, LCD

회로도



2-3 모듈별 기능 명세서

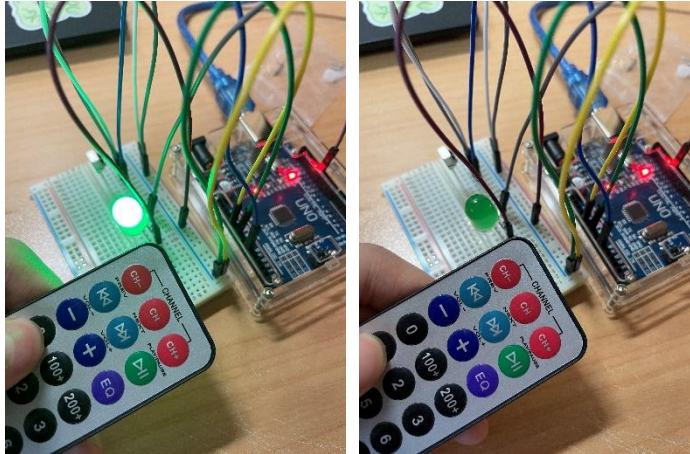
모듈명	기능
서보모터	각도 조절을 통해 전등 스위치를 제어한다.
초음파 센서	사용자가 가까이 있는지 확인한다.
적외선 수신 센서 & 리모컨	리모컨을 이용해 원하는 버튼을 누르면 적외선 센서를 통해 해당 입력을 수신한다.
LCD(+가변저항)	LCD를 통하여 현재 어떠한 기능이 동작되고 있고, 알람이 울리기까지 시간은 얼마나 남았는지 사용자에게 알려준다. 가변저항을 통해 LCD의 밝기를 조절한다.
LED	리모컨의 입력이 있을 때, 정상적으로 입력되었다면 LED를 깜빡인다.
수동 부저	해당 부저를 통해 알람을 울린다. 계이름을 직접 입력하여 멜로디를 바꿀 수 있으며, 해당 프로젝트에서는 '학교 중' 노래를 이용하였다.

3. 프로젝트 진행 과정

한번에 여러 부품을 테스트하는 것이 아니라 파트 별로 나눠 제품이 어떻게 동작하는지 이해하고, 그에 맞춰 코드를 구성함.

① 적외선 센서 입력 확인

0을 누르면 LED가 켜지고, 1을 누르면 꺼진다.



1) 리모컨을 통한 적외선 송수신 라이브러리 <IRremote.h> 사용법 숙지

2) switch문을 통하여 버튼에 따라 다른 동작을 하게끔 설정

코드

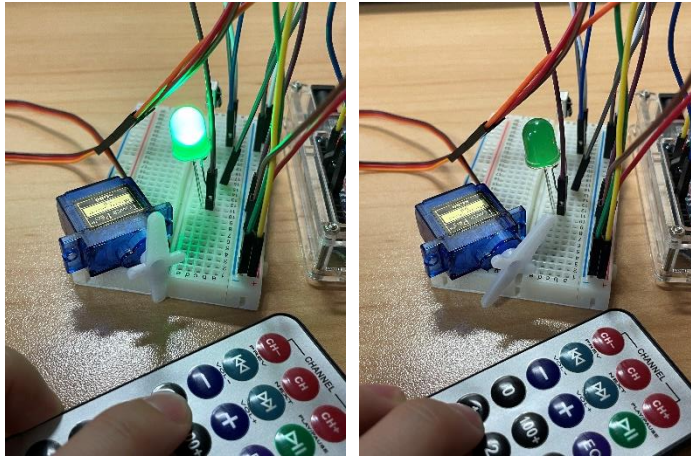
에듀이노 예제 코드¹ 참조

```
#include <IRremote.h>
int RECV_PIN = A0; // 입력값 핀 설정
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // 수신 시작
  pinMode(12, OUTPUT); // LED핀 설정
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);

    switch (results.value) {
      case 0xFF6897: digitalWrite(12, HIGH); break; // 0버튼이 눌리면 LED ON
      case 0xFF30CF: digitalWrite(12, LOW); break; // 1버튼이 눌리면 LED OFF
    }
    irrecv.resume(); // 다음 출력값 수신 준비
  }
}
```

② 적외선 센서를 통해 서보모터 제어

0을 누르면 서보모터 각도 90°, 1을 누르면 45°



1) 서보모터 제어 라이브러리 <Servo.h> 사용법 숙지

코드

```
#include <IRremote.h>
#include <Servo.h>

Servo servo; // servo라는 객체 선언

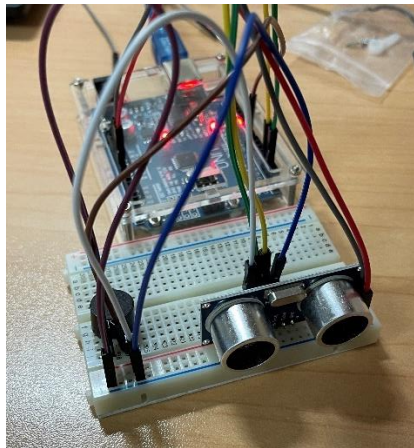
int RECV_PIN = A0; // 입력값 핀 설정
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
    Serial.begin(9600);
    irrecv.enableIRIn(); // 수신 시작
    pinMode(12, OUTPUT); // LED 핀 설정
    servo.attach(7); // 서보모터 핀 설정
}
void loop() {
    if (irrecv.decode(&results)) {
        Serial.println(results.value, HEX);

        switch (results.value) {
            case 0xFF6897: // 0 버튼
                digitalWrite(12, HIGH);
                servo.write(90); // 모터 각도 90도
                break; // 0버튼이 눌리면 LED ON

            case 0xFF30CF: // 1 버튼
                digitalWrite(12, LOW);
                servo.write(45); // 모터 각도 45도
                break;
        }
        irrecv.resume(); // 다음 출력값 수신 준비
    }
}
```

③ 초음파 센서+알람

알람이 울리다가 일정 거리 내에 물체가 있을 경우 알람 종료



```
02:44:04.822 -> 147cm
02:44:04.963 -> 148cm
02:44:05.055 -> 111cm
02:44:05.194 -> 146cm
02:44:05.288 -> 49cm
02:44:05.380 -> 95cm
02:44:05.475 -> 43cm
02:44:05.615 -> 41cm
02:44:05.709 -> 39
```

☒ 자동 스크롤 ☒ 타임스탬프 표시

- 1) pitches.h 라이브러리를 통해 음계별 주파수 불러오고, 사용
- 2) 초음파 센서를 이용해 사물이 얼마정도 떨어져있는지 알아보기

코드

```
#include "pitches.h" //음계별 주파수 저장됨

#define C 262 //도를 뜻하는 주파수 262를 c로 정의

int trig=2;
int echo=3;
const int spk=9;

void setup() {
  Serial.begin(9600);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
}

void loop() {
  tone(spk,C); //스피커를 통해 c값인 262 즉, 도의 음을 내라
  digitalWrite(trig,HIGH);
  delay(1);
  digitalWrite(trig,LOW);
  int dis=pulseIn(echo,HIGH)*340/2/10000;
  // 초음파 발생 시간을 계산해 거리 (cm) 단위로 표현
  Serial.print(dis);
  Serial.println("cm");
  if (dis<40){ //40cm 이내에 물체가 있으면 알람 멈춤
    noTone(spk);
    exit(0);
  }
  delay(100);
}
```


코드 2

배열을 통해 <학교 중> 멜로디 구현

```
#include "pitches.h"

#define C 262
#define D 294
#define E 330
#define G 392
#define A 440

int alarm[] = {
    G, G, A, A, G, G, E, 0,
    G, G, E, E, D, 0,
    G, G, A, A, G, G, E, 0,
    G, E, D, E, C
}; //학교 중 멜로디 배열

int alarmlen = sizeof(alarm) / sizeof(int); //총 출력해야할 음 길이

void alert() {
    while (1) {
        for (int i = 0; i < alarmlen; i++)
        {
            tone(spk, alarm[i], 500); //alarm에 들어있는 음 하나씩 출력
            delay(500);
        }
    }
}
```

④ 타이머 기능

- 1) 리모컨 버튼이 수신한 값==버튼이 의도하는 값은 아니기 때문에 정상적으로 활용 할 수 있도록 수정이 필요했다.

ex) 리모컨에서 1을 눌렀지만 수신되는 값 자체는 16진수로 0xFF30CF를 나타냄

```
String Timer = "";
String Time = "";
if (irrecv.decode(&results)) {
    if (results.value != 0xFFC23D) {
        switch (results.value)
        {
            case 0xFF6897:
                Time = "0";
                break;

            case 0xFF30CF:
                Time = "1";
                break;

            case 0xFF18E7:
                Time = "2";
                break;

            case 0xFF7A85:
                Time = "3";
                break;
```

switch 문을 이용하여 입력에 따라 Time 값 지정

2) 알람 기능 구현

```
Timer = Timer + Time;
int data = 0;
data = Timer.toInt();

for (int i = 0; i < data; i++) {
    delay(60000); //1분
}
alert(); //알람 기능 함수
```

앞서 선언한 Timer와 Time 모두 String형으로 선언한 이유는 120분 알람을 맞추기 위해선 1, 2, 0 총 세번의 입력이 필요한데 String형을 이용해 더하면 따로 세 번 입력했다고 하더라도 Timer에 120이라고 저장되기 때문이다.

이것을 정말 시간이 가는 것처럼 이용하기 위해서는 문자열형을 정수형으로 변환해야 하기 때문에 data라는 변수를 이용하였다.

for문을 이용해 입력된 만큼 분당 delay를 거친 후, 알람이 울린다.

3) 알람 리셋

```
case 0xFF629D: //ch버튼
    Time = "";
    Timer = "";
    break;
```

ch버튼을 누르면 이때까지 저장되었던 Time와 Timer모두 초기화되어 알람을 새롭게 설정할 수 있다.

4) LCD를 통해 알람까지 남은 시간 출력

```
for (int i = 0; i < data; i++) {
    int Clock = data - i; //몇 분 남았는지 Clock에 저장
    lcd.clear();
    lcd.write("Alarm...");
    lcd.setCursor(0, 1); //두번째 줄로 위치 이동
    lcd.print(Clock);
    lcd.write(" minute left");

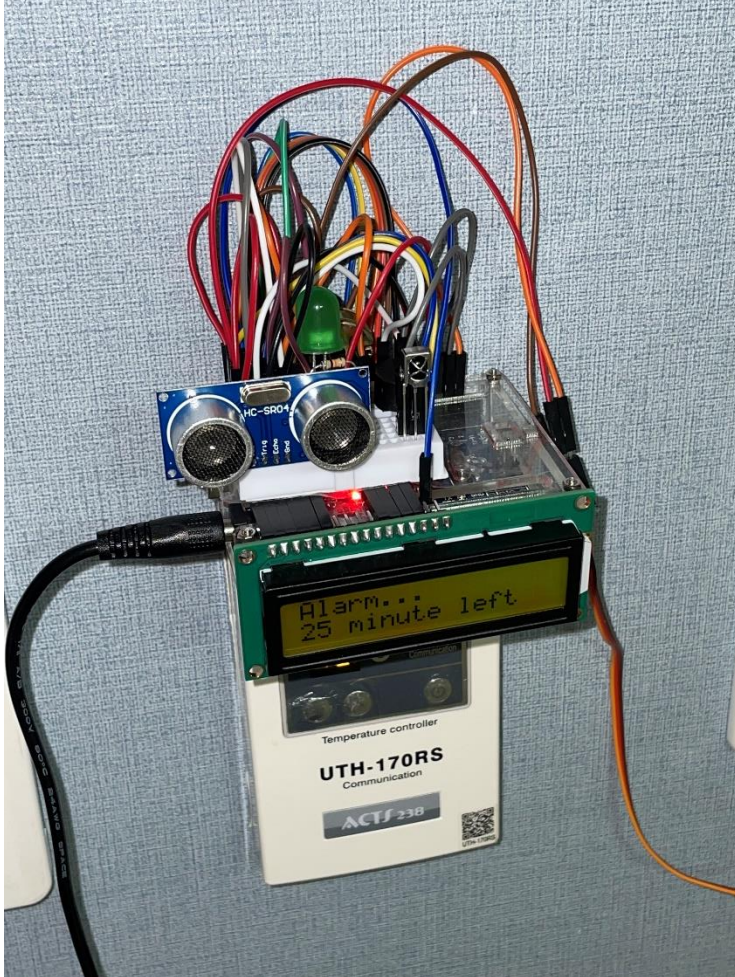
    delay(60000); //1분
}
```



만약 26분으로 설정했다면 1분 후 "25 minute left"를 출력한다.

4. 프로젝트 결과

<https://www.youtube.com/watch?v=uvN3YUE7riE>



리모컨을 이용하여 스위치를 제어하고, 알람을 설정한다.

알람이 울릴 때 사용자가 기기에 가까이 다가가면 자동적으로 꺼진다.

5. 참고 문헌

¹ https://www.eduino.kr/product/detail.html?product_no=83&cate_no=55&display_group=1