Auto Blind, 중앙 제어장치

### 시스템 소프트웨어

### 최종보고서

### 컴퓨터 정보 공학과

### 1301568 김기영

## 2017. 12. 12

## 

[시스템 소프트웨어 1](#_Toc500805876)

[최종보고서 1](#_Toc500805877)

[컴퓨터 정보 공학과 1](#_Toc500805878)

[1301568 김기영 1](#_Toc500805879)

[2017. 12. 12 1](#_Toc500805880)

[개요 3](#_Toc500805881)

[개발 구조 4](#_Toc500805882)

[Auto Blind 5](#_Toc500805883)

[디자인 5](#_Toc500805884)

[GPIO 구성 7](#_Toc500805885)

[소스 8](#_Toc500805886)

[중앙 제어장치 8](#_Toc500805887)

[디자인 8](#_Toc500805888)

[GPIO 구성 9](#_Toc500805889)

[소스 10](#_Toc500805890)

[특이사항 12](#_Toc500805891)

[포트 포워딩 설정 12](#_Toc500805892)

[부팅 시 자동 실행 설정 13](#_Toc500805893)

[결론 14](#_Toc500805894)

# 개요

사람은 누구나 편한 것을 좋아합니다.

“편하다”는 휴식을 통한 쉼을 의미하지만 어떤 일을 직접하지 않고도 할 수 있는 것을 의미하기도 합니다.

현대 사회에 자동화가 많아지고 있는 현대 사회에서 가정과 같은 소규모 실내 공간에도 자동화가 찾아오고 있습니다.

사용자는 장소에 구별없이 “편함”을 추구하게 됩니다.

Auto는 이런 요구들을 충족시킬 수 있는 자동화 기기를 의미합니다.

따라서 이번 프로젝트를 수행하면서 사람이 보다 편리함을 추구할 수 있는 제품을 만들기 위해서 노력했습니다.

# 개발 구조

최종 제품은 Auto Blind와 중앙제어장치로 구성되어 있다.

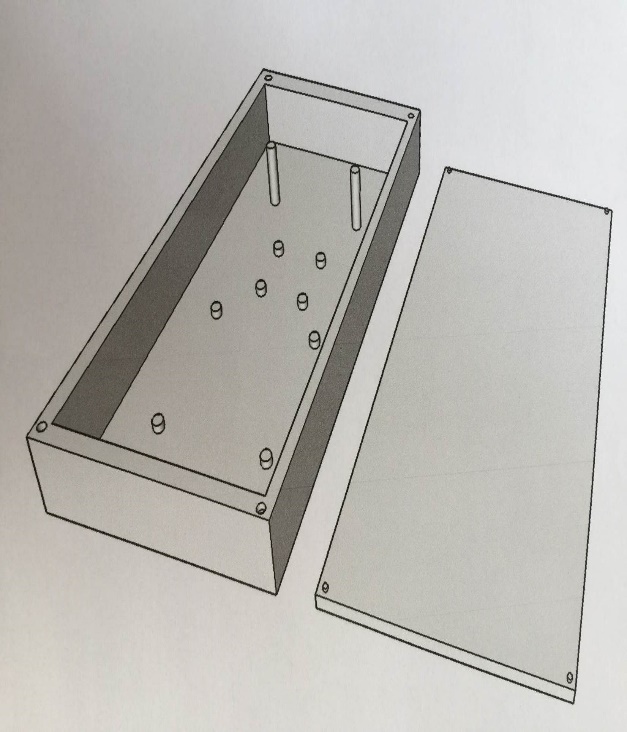
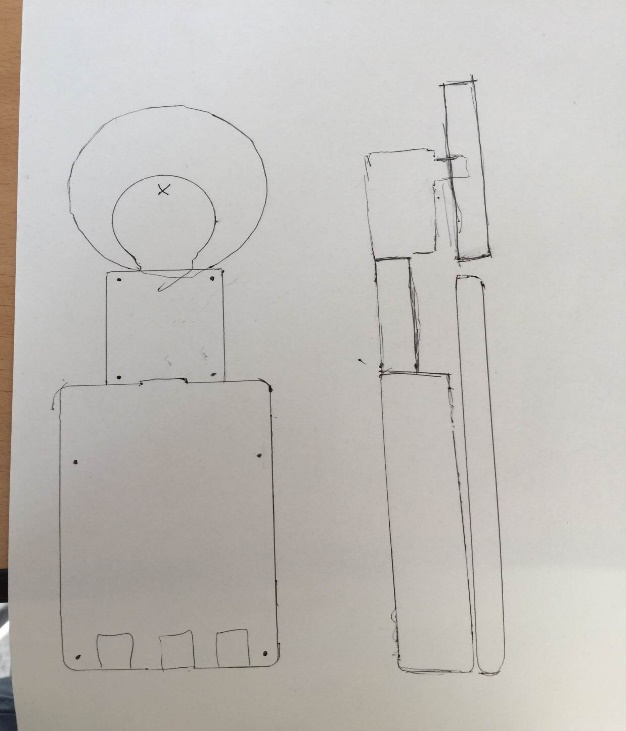
Auto Blind는 라즈베리파이3 1개, GPIO버튼 2개, 모터 컨트롤러 1개, 스탭 모터 1개, 내장 배터리 1개로 이루어져 있다.

GPIO 버튼은 각각 블라인드를 오릴 수 있는 버튼과 내릴 수 있는 버튼으로 쓰인다.

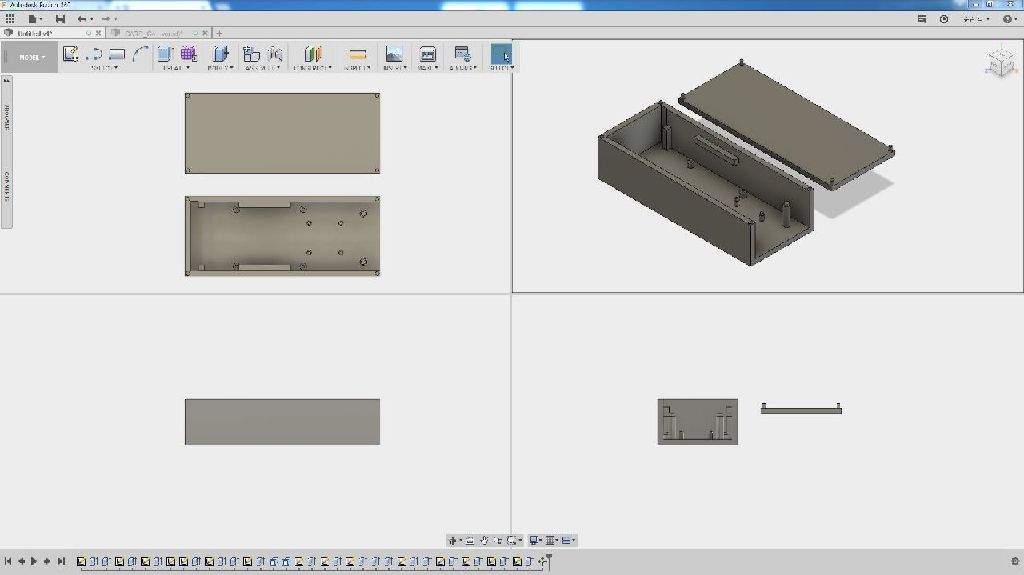
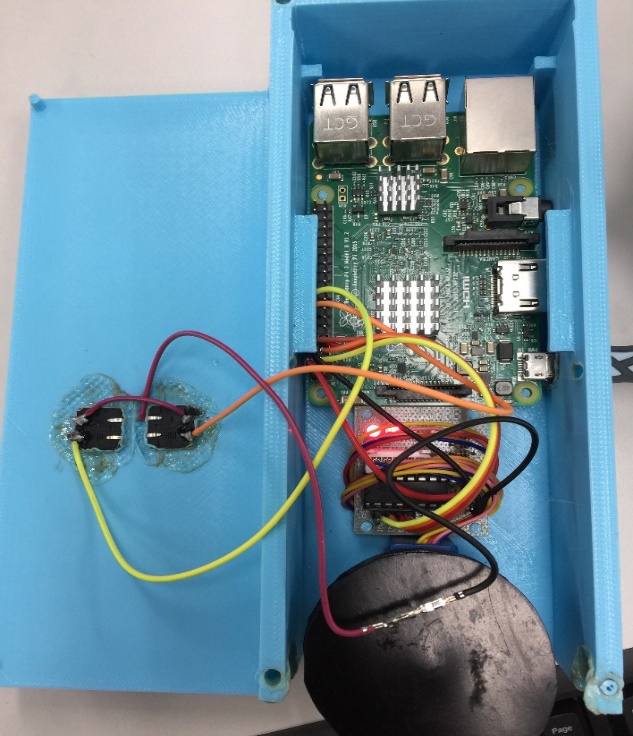
모터 컨트롤러는 스탭모터와 라즈베리파이3과의 GPIO통신으로 모터를 제어할 수 있도록 제작된 전용 컨트롤러이다. 스탭 모터는 4개의 GPIO선을 공급받아 모터를 특정 방향만큼 움직일 수 있다. 라즈베리파이3은 내장 배터리로부터 전원을 공급받아 작동하게 된다.

## Auto Blind

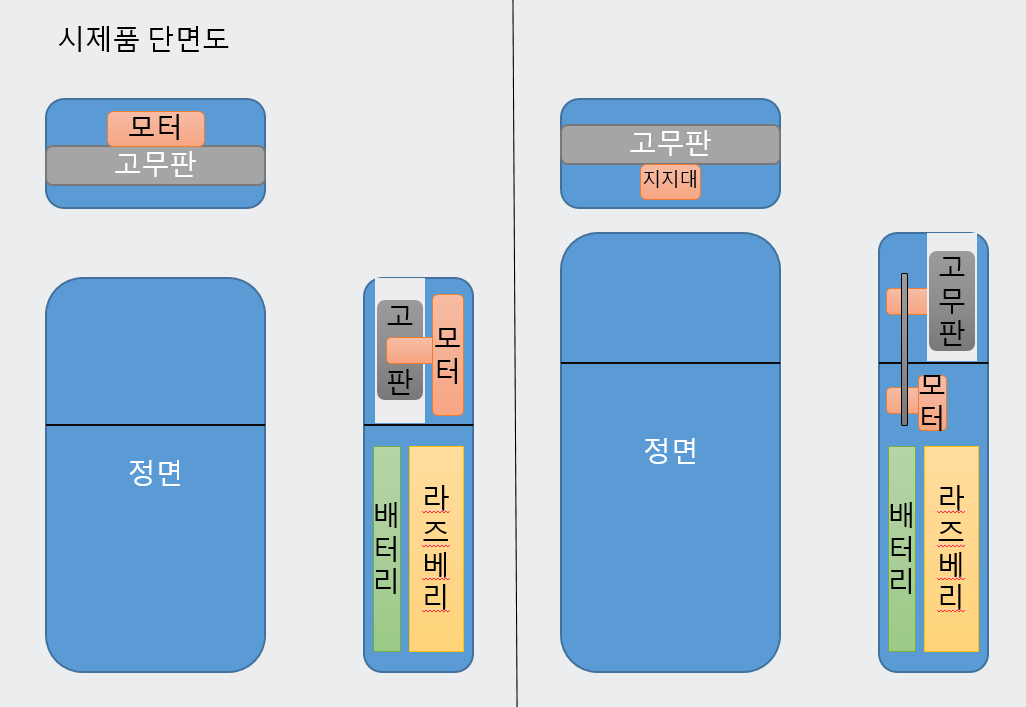
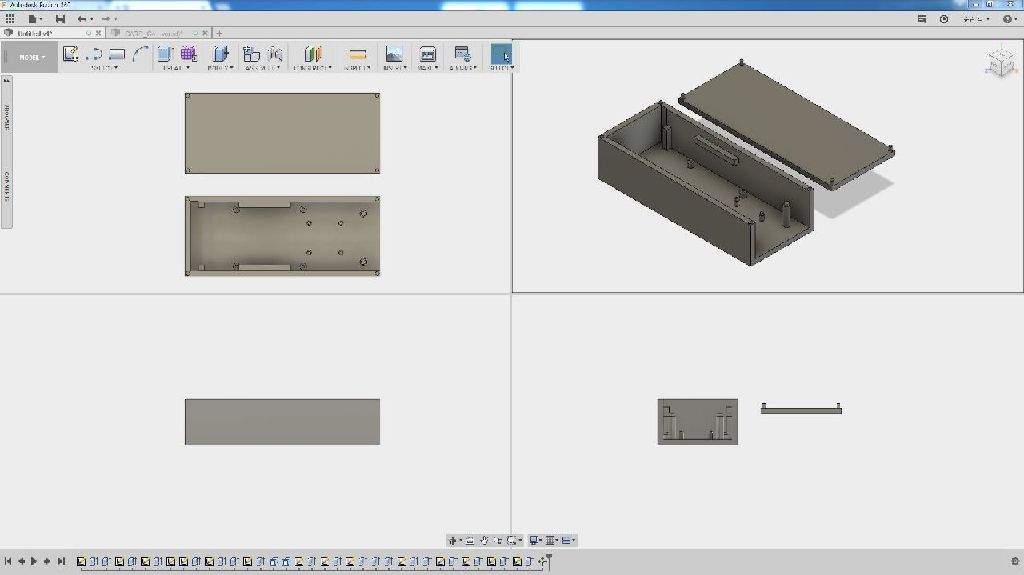
### 디자인



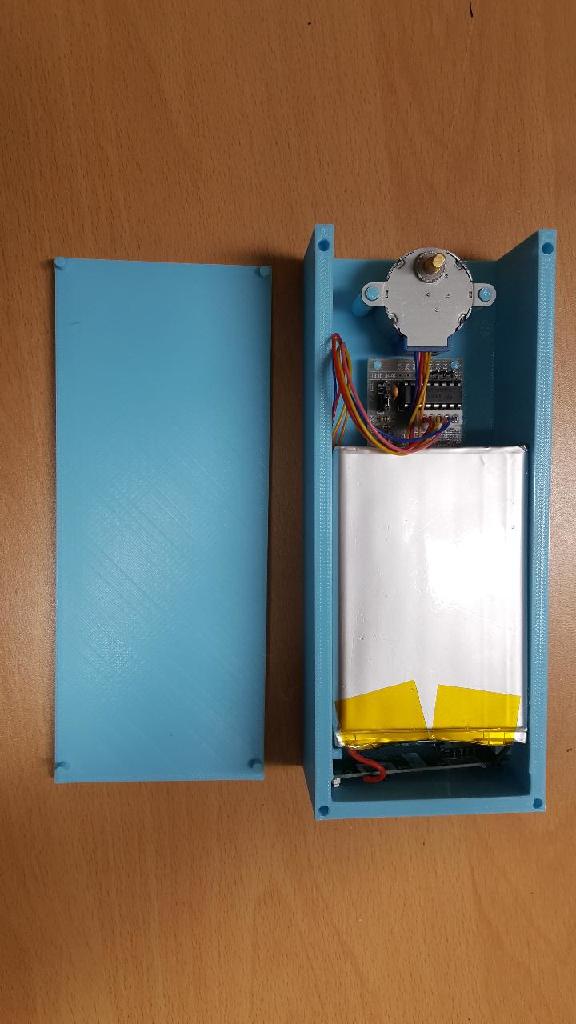
<초기 내부 예상 배치도> <3D 프린팅 예상 그림>



<3D 프린팅 예상 그림> <프로토타입 내부 실 사진>



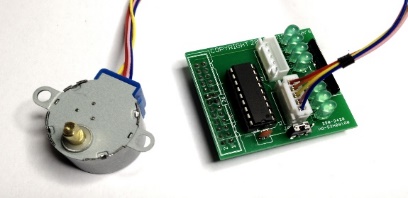
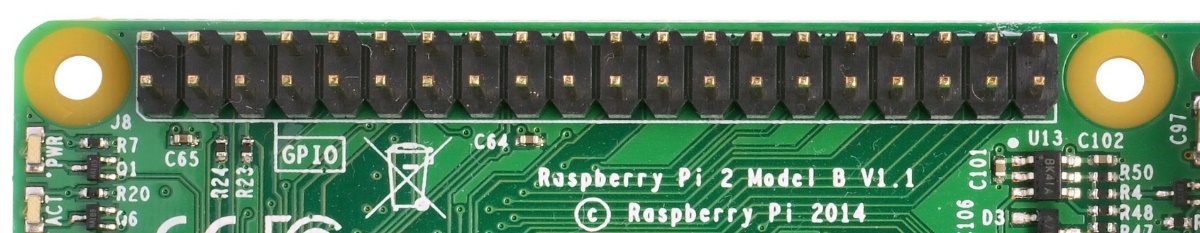
<3D 프린팅 예상 사진> <초기 디자인 2안>





<프로토타입 완성 사진> <내부 배치구성 사진>

### GPIO 구성



02

03

04

14

23

24

5V

GND

모터에 사용되는 GPIO 신호 선 : 2, 3, 4, 14

모터에 사용되는 GND 선

모터에 사용되는 5V 선

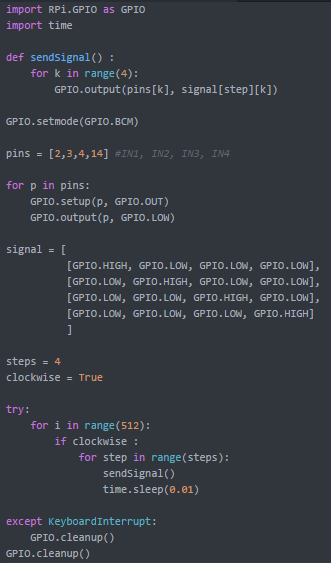
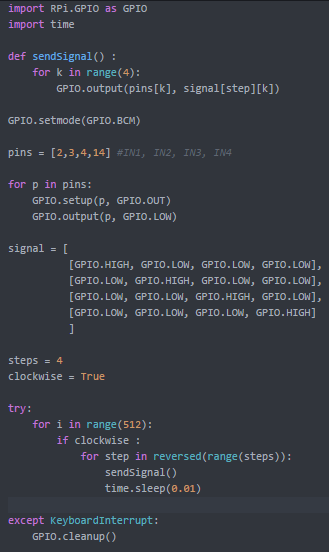
각 버튼에 사용되는 GPIO 신호 선 : 23, 24

각 버튼에 사용되는 GND 선

총 8개의 선을 사용하여 블라인드를 제어할 수 있다.

GND선을 각각 3개선으로 나눌 수 있었지만 블라인더 내부 선 정리를 위해서 하나로 3개를 전부 연결해보았다.

### 소스



파일은 총 3개로 main.py, Auto\_UP.py, Auto\_DOWN.py로 구성되어있다.

소스 내용은 각각 위 소스의 내용과 같다.

main.py는 실질적으로 모터를 구동시키지는 못한다. 버튼 클릭 시 Auto\_UP.py나 Auto\_DOWN.py를 실행시키는 보조 역할을 하게 된다. 하지만 이 main.py는 라즈베리파이가 전원을 입력받을 경우, 전원이 켜지고 나서 항상 켜지도록 설정 되어 화면이나 별도 컨트롤러가 없는 오토 블라인드에 최적화되어 있다.

Auto\_UP.py와 Auto\_DOWN.py는 1부분만 다르고 전체적으로는 일치한다.

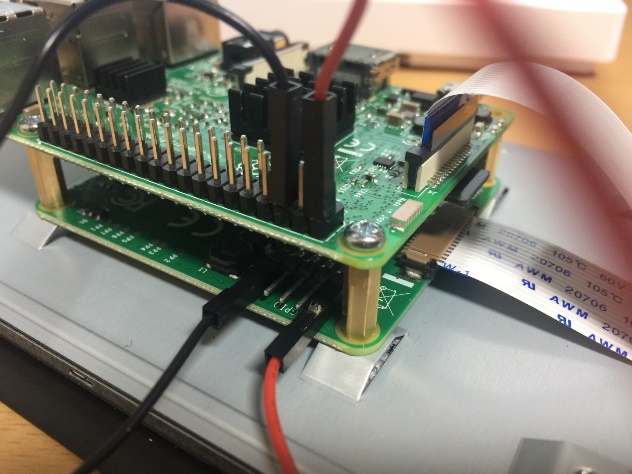
오토블라인드에는 스탭모터가 필요한데 스탭모터에는 4개의 시리얼 통신을 위한 선이 필요하다. [0,0,0,1], [0,0,1,0], [0,1,0,0], [1,0,0,0] 와 같이 순서대로 회전을 지시하거나 역방향으로 회전을 하게 지시 할 수 있다.

중앙 제어장치

### 디자인

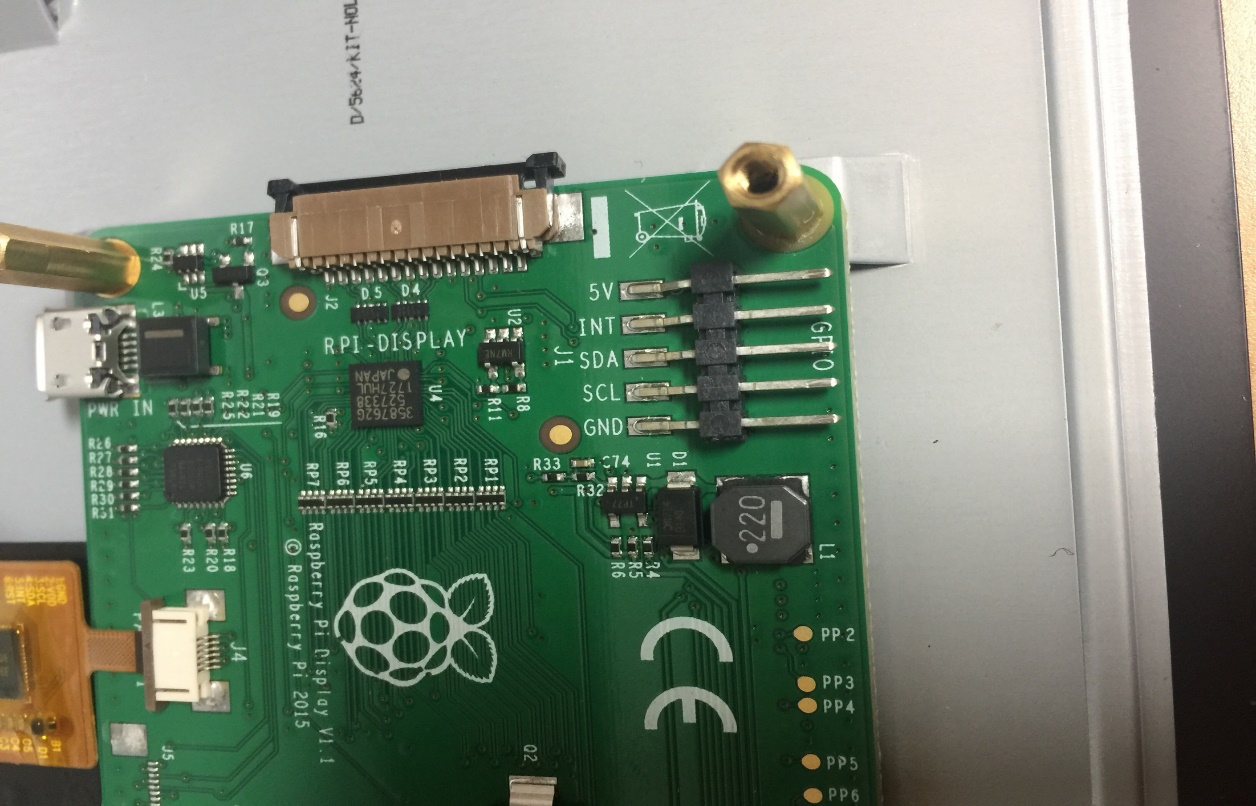
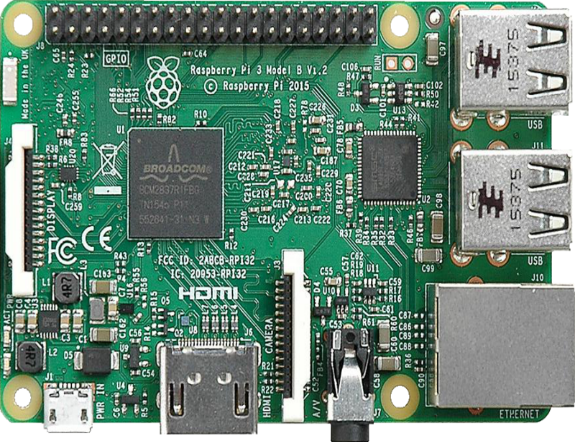


<메인 함수 실행 화면> <본체 후면 사진>



<LCD 후면부 사진> <GPIO 배치 사진>

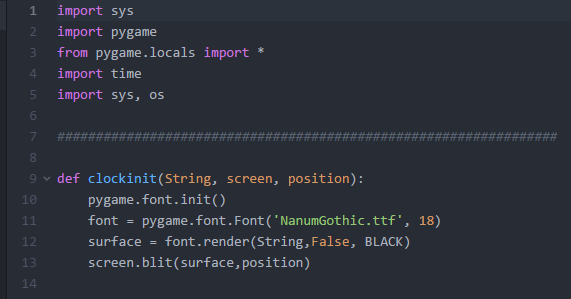
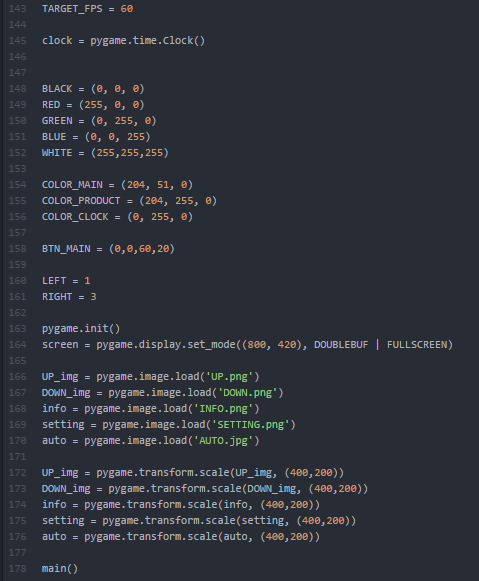
### GPIO 구성



LCD 장치에 필요한 GPIO 선들은 5V 전원선과 GND 선이다.

각 라인에 맞게 연결해주고, 라즈베리파이에서 지원하는 LCD 시리얼 전용선으로 LCD 패널과 연결해주면 GPIO 관련 설정은 끝난다.

### 소스

시스템 콘솔 명령어 실행

GUI 환경 개발 툴

화면 동기화 함수

FPS 60으로 고정

시스템 시간 받아오기

색상표 초기화

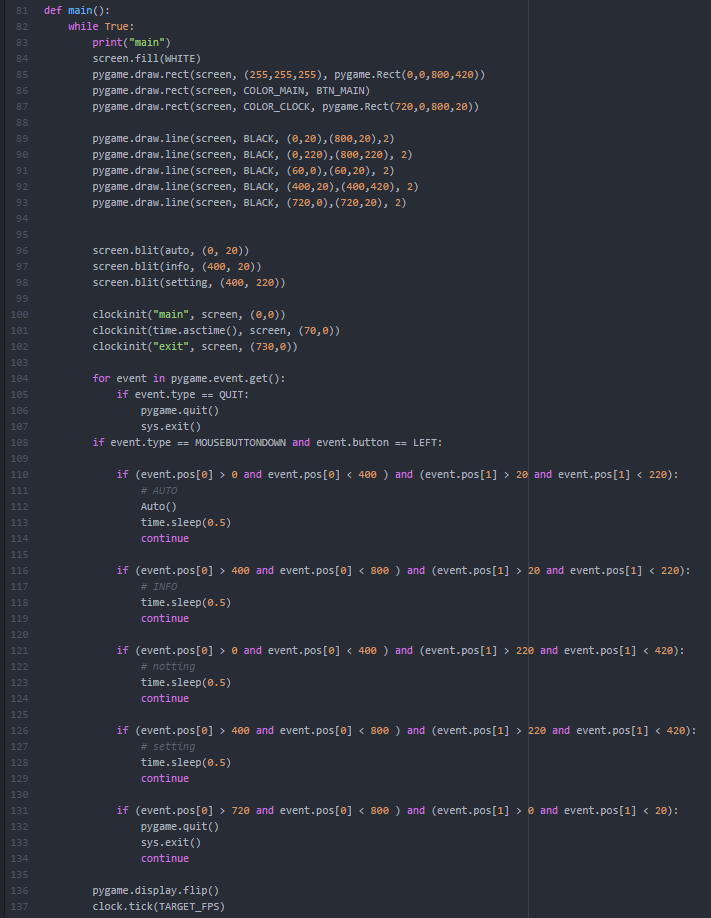
키 입력 초기화

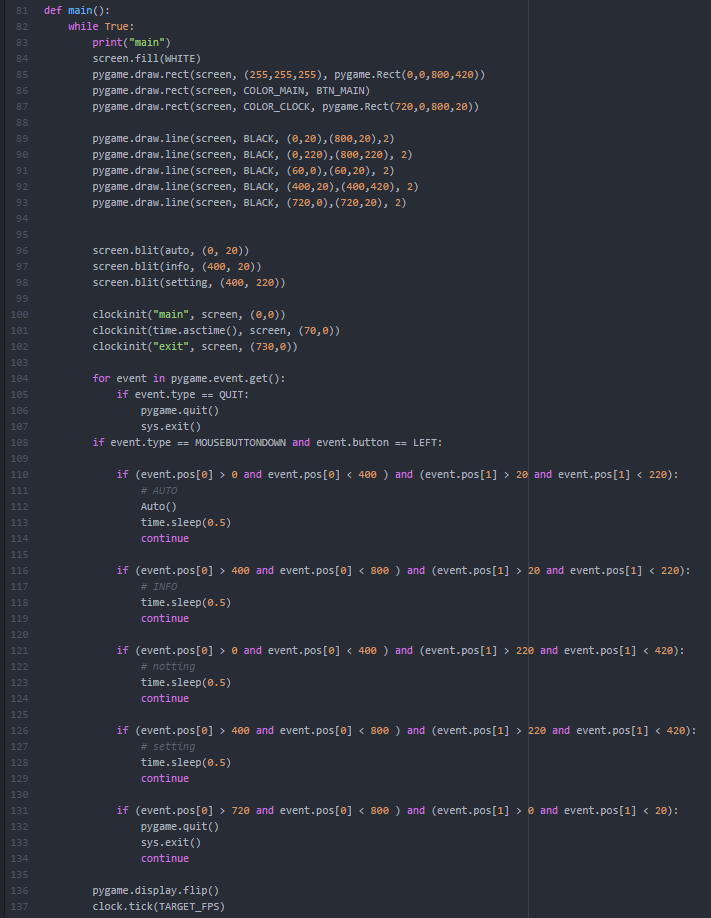
화면 출력, 더블 버퍼, 전체화면

따로 준비해둔 이미지파일 불러오기

이미지를 화면에 출력하기

메인함수 호출



사각형 그리기

선 그리기

그림 넣기

글자 넣기

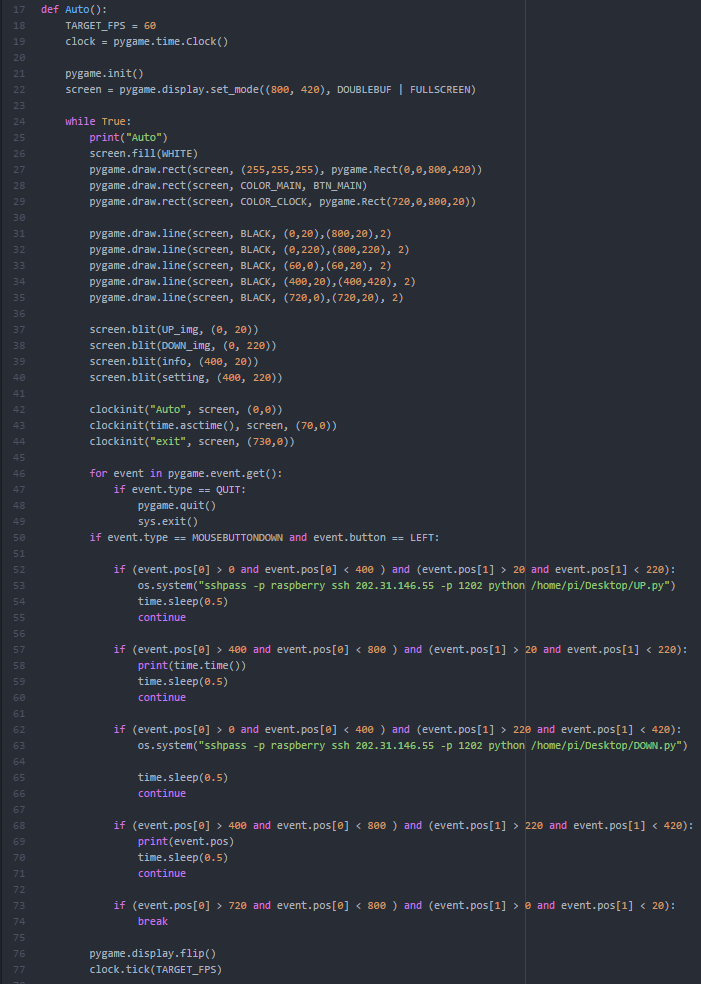
이벤트 처리 부분

상단 바에서 ‘x’를 눌렀을 경우

마우스 버튼 클릭다운이면서 왼쪽 버튼인 경우

포인트 위치가 각 범위 내인 경우

화면 갱신, FPS 조절



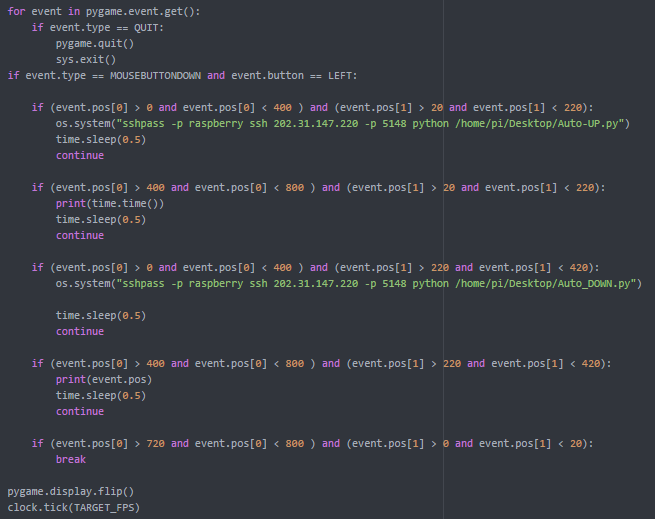
사각형 그리기

선 그리기

그림 넣기

글자 넣기

이벤트 처리 부분

상단 바에서 ‘x’를 눌렀을 경우

마우스 버튼 클릭다운이면서 왼쪽 버튼인 경우

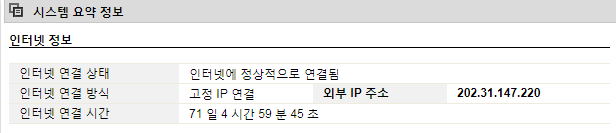
포인트 위치가 각 범위 내인 경우

sshpass는 ssh 접속을 위해 페스워드를 매번 입력해야하는 번거로움을 줄여준다. (-p) ssh를 이용해서 원격지의 IP, PORT(-p)를 설정해주고 파일을 실행시킬 수 있다.

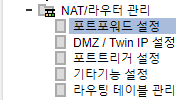
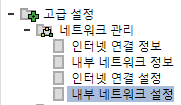
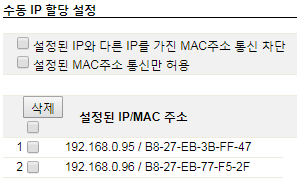
화면 갱신, FPS 조절

## 특이사항

### 포트 포워딩 설정



202.31.147.220:8080



소스 상에서는 외부에서 접근하기 위해 공유기의 [외부 IP 주소 : 포트번호] 조합으로 접근을 시도한다.

이 접근은 유효화하기 위해서 포트포워딩을 설정해주어야 한다.

첫번째로 공유기 내부에서 내부IP 주소가 섞이거나 재설정되지 않도록 수동 IP 할당을 해주어야한다. 이 작업이 없다면 당장은 문제없이 작동을 하지만, 추후에 내부 IP 주소가 바뀌는 경우가 생기는 등 연결에 문제가 생기게 되므로 꼭 해주어야 한다.

두번째로는 포트 포워딩을 설정해주어야 한다.

외부 포트로 접근을 하면 내부 네트워크의 특정 포트번호로 스위칭해주는 기능이다.

Auto Blind에서는 SSH 프로토콜만을 이용하여 원격 접속을 하기 때문에 XRDP 같이 3389번이 아니라 22번 전용 포트번호를 개방해준다.

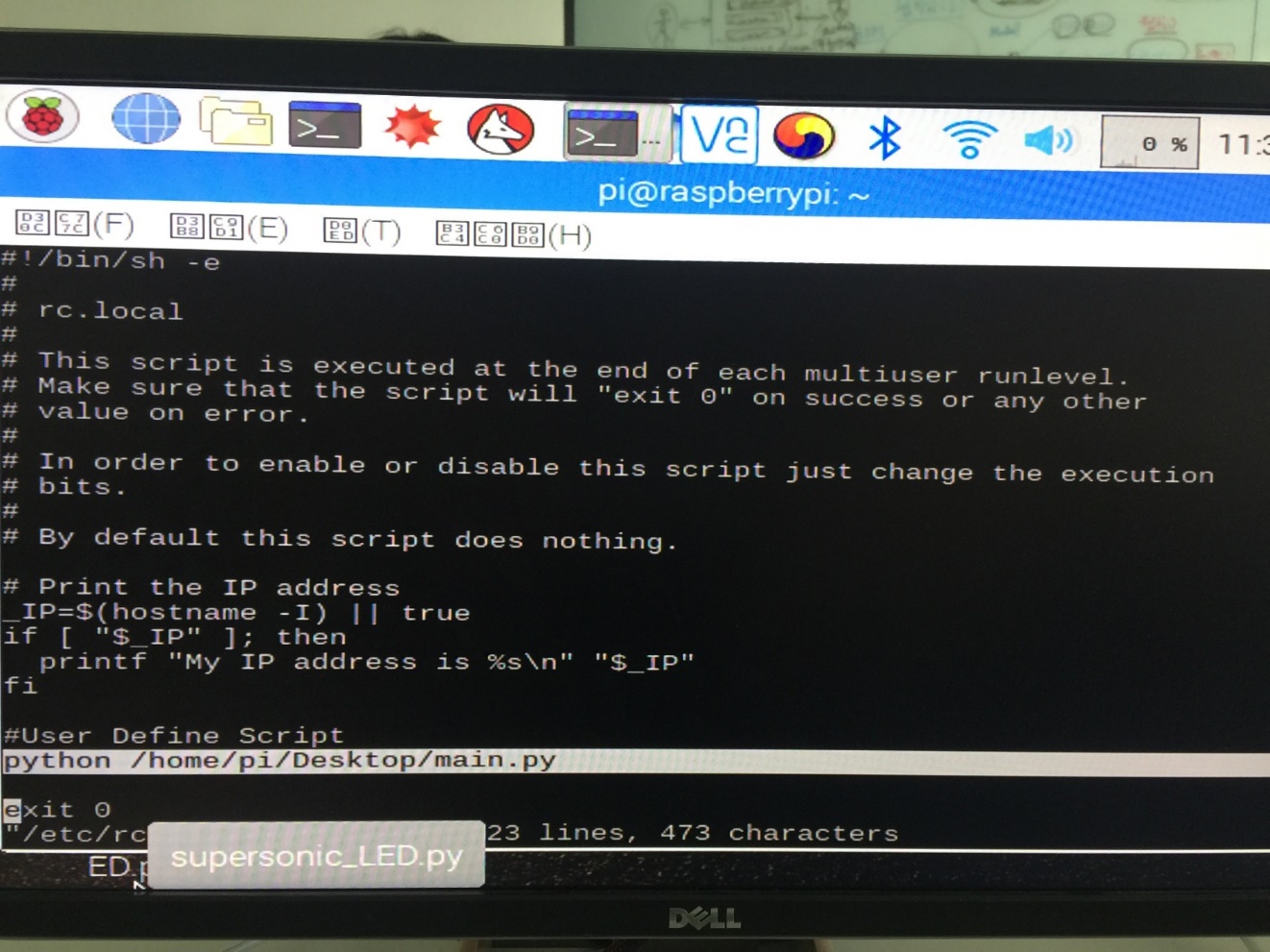
여기까지 하면 공유기 설정은 끝이 난다.

### 부팅 시 자동 실행 설정

리눅스 계열에서 부팅 후 특정 명령어를 실행시킬려면 /etc/rc.local 파일을 수정함으로써 프로그램을 자동 실행시킬 수 있다.

vi나 nano같은 에디터를 사용하여 rc.local파일을 수정한다.

sudo 권한이 필요할 수 있다.



# 결론

Auto Blind를 만들게 된 계기가 연구실 창문의 블라인드의 줄 부분이 손에 닿기 어려운 위치에 있어서 이를 쉽게 조종할 수 있도록 하기 위해서 만들기 시작했다.

실생활에 필요한, 실용적인 제품을 만들려고 하였고, 번뜩이는 아이디어가 아니라 가장 기본적인 아이디어로 시작하였다.

이번 기말고사 작품에서는 SSH와 포트포워딩, Pygame을 이용하여 간단하면서도 복잡한 스마트 홈을 구축해보았다.

중앙 제어장치는 IPTIME 공유기 내부망의 포트포워딩 기능을 통해서 외부에서도 원격 제어가 가능하도록 제작되었다. 또한 Pygame 라이브러리를 활용하여 GUI 환경으로 원격지 기기의 기능을 활용할 수 있게 하였다.

GUI환경의 인터페이스나 기기를 추가할 경우 등에 대해서는 추가적으로 개발이 필요한 부분이다. 단기간에 오류를 잡아내고 외적으로 좋게 만들기에는 아쉬웠다.

Auto Blind는 블라인드를 자동으로 올리고 내릴 수 있도록 하는 보조장치이다.

그렇기 때문에 기기 자체에서도 중앙제어장치의 제어가 없어도 기능이 실행되어야 했다. 기기 자체에 버튼을 추가하여 중앙 제어장치가 없어도 Auto Blind의 기능을 잃지 않도록 제작되었고, 배터리를 내장해 벽에 거치하는 형태로 소형화에 힘을 썼다.

내부 배터리를 추가할 슬롯이 없다는 점, 3d 프린팅의 설계 오류때문에 모터의 위치가 중앙에서 벗어난 점이 아쉬웠다.

기말고사를 준비하는 기간이 길지도 짧지도 않았지만 이 시간동안 Auto Blind를 기획하고 제작하면서 스마트 홈과 주변장치간의 통신 방식과 어떻게 하면 더 좋은 제품을 만들 수 있는지, 시제품이 만들어지고난 후에 추가적으로 더 할 일에 대해서 깊은 생각을 할 수 있는 시간이였다.