學士學位 論文

가정용 스마트 식물 케어

Smart Plant Care for Home

지도교수 전 병 훈

동국대학교 전자전기공학부

배 윤 호

2020

목 차

1. 서론

1.1 설계의 배경과 필요성

최근, 1인 가구가 증가함에 따라서 젊은 세대를 중심으로 반려 식물 기르기에 대한 관심이 급속도로 확산되고 있다. 또 식물로 심리를 치료하는 경우도 있다. 하지만 바쁜 도시생활로 인해 장시간 집을 비우는 시간이 많은 경우, 식물에 물이 부족해 시들어 버리거나, 햇빛을 많이 받아 시든다거나, 햇빛이 없어 시든다던가 실내에서 키우고 싶지만 빛이 부족해서 못키우는 경우도 있고 해충에게 피해를 받아 대처를 못해서 식물이 죽는 경우가 많이 발생되고 있다. 이러한 문제점들을 해결하고,접근성을 높여 초보자들도 편리하게 식물을 키우기 위하여 집을 비울 때에도 식물을 관리할 수 있는 스마트 화분을 만들기로 하였다.

1.2 설계 주제 및 목적

스마트화분을 제작하는데 있어서 가장 중요하게 생각한 것은, 식물이 잘 자라기 위해서는 식물이 원하는 적절한 상태를 유지해 주는 것이라 생각하였다. 하지만 반응이 적은 식물의 재배 상태를 눈대중으로 유지하기에는 초보자나 시간이 부족한 사람이 제대로 키우기는 힘든 부분이 있다. 장시간 동안 물을 주지 않거나 혹은 물을 너무 많이 줘서 매번 식물을 죽여 화분 키우기를 포기하는 경우가 많았다. 이를 위해 온도나 토양 습도 등 식물의 상태를 확인하기 위한 센서들과 LED 조명을 이용해 필요한 조도를 확보하고, 해충에게서 식물을 보호하기 위해 카메라에 움직임이 잡히면 블로우팬을 이용해 바람을 쐬어서 벌레에게서 보호해 주는 장치를 장착하였다. 또한 물통과 워터펌프가 연결되어 있어 토양 수분 센서를 통해 토양에 수분을 측정하여 적절한 시기에 적절한 양의 물을 화분에 공급하여 식물을 관리할 수 있도록 하였다.

1.3 설계 제약조건

스마트 화분을 설계하면서 가장 중요하게 생각한 부분은, 단순히 식물만 잘 관리하는 것이 목적이 아니라, 사용자가 원하는 식물에 알맞은 값을 조절하여 키울 수 있는 프로그램을 제공하는 것이다. 이는 실제로 사용자가 불편함을 느낄 수 있는 부분이기 때문에 각기 다른 식물에 얼마만큼 물을 줘야 식물을 적절하게 키울 수 있을지 설정해주며, 토양 습도와 조도를 모두 체크하여 각 식물의 필요한 값을 조정하여 관리해준다. 또한 모터 팬을 이용한 해충방지, LED 조명을 이용한 필요한 조도 확보 기능을 추가하였다. 화분에 어떠한 조건을 만족했을 때에만 입력되도록 조절할 수 있고 오동작을 최대한 줄이는 쪽으로 설계하도록 노력하였다.

1.4 관련 자료 및 분석

시중에 나와 있는 제품으로는 아두이노와 라즈베리파이를 기반으로 하는 스마트화분에 여러 센서를 장착하여 수집한 수치에 따라 온도와 습도를 조절하여 화분의 상태를 조절할 수 있는 제품들이 있었다. 하지만, 우리가 고안한 조도를 측정해 줄 수 있는 센서와 해충방지를 위한 방법이 구현되어 있지 않아서, 화분을 키우는데 취약

하고 식물관리에 어려움을 겪을 수도 있을 것이라는 판단을 하였다. 기존에 있는 기능에 여러 가지 기능을 보안하여 더욱더 크게 관심을 기울이지 않아도 화분을 키울 수 있도록 여러 취약점을 보안하려고 노력하였다.

2. 설계 개념

2.1 설계 목표

화분에 설치된 여러 센서를 부착시켜 자동으로 식물 및 토양의 상태를 측정하여 측정값을 토대로 모터 팬 그리고 LED 생장등을 동작시키는 기능을 가져 자동적으로 식물을 관리해주는 시스템을 설계하였다. 부착하는 센서로는 토양습도센서, 조도센서 이를 토대로 동작하는 기기로는 수분 공급 워터 펌프, 모터팬, LED생장등이 있다. 화분의 각 상황에 맞는 기능을 자동으로 구현하여 관리하기 때문에 화분을 직접 관리하지 않아도 화분에 가장 적절한 상태를 유지할 수 있고, 또한 카메라를 이용해 해충의 움직임을 포착하여 해충피해를 방지할 수 있으므로 사전에 피해를 방지하여 화분을 더 효과적으로 관리할 수 있다.

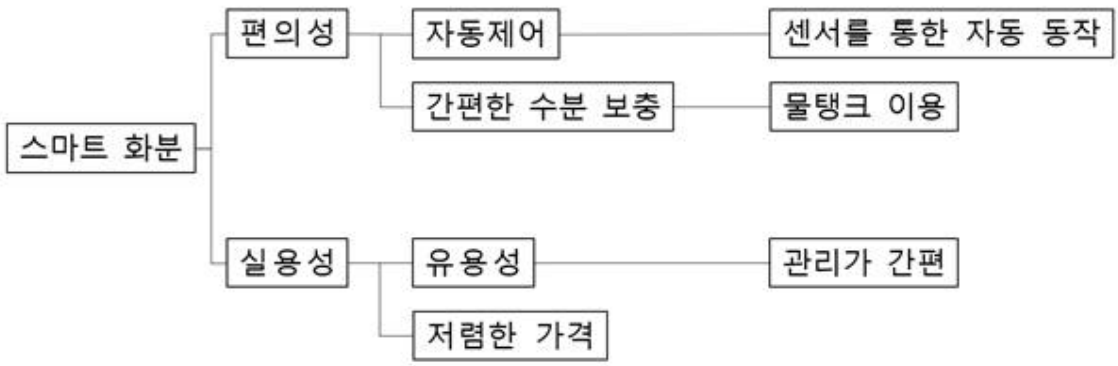

Fig. 1

2.2 설계안

(블록도)

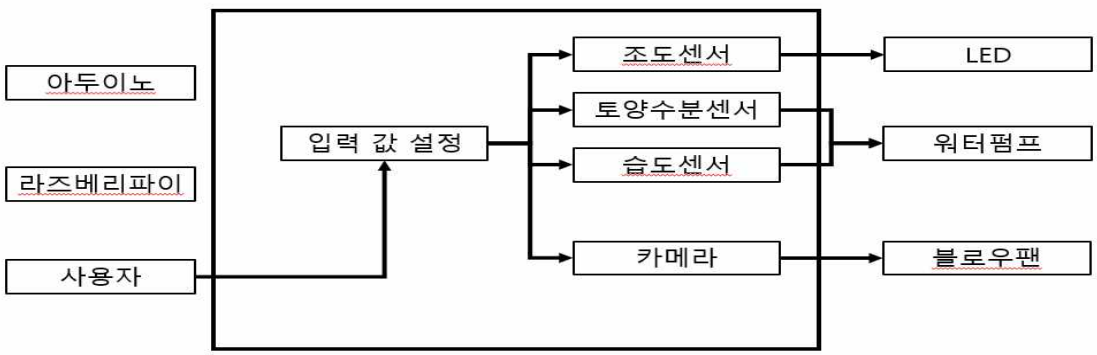

Fig. 2

2.3 평가항목과 평가방법

평가항목	평가방법	평가기준
토양습도센서가 제대로 동작하는지 확인한다.	시리얼모니터를 통해 측정되는지 확인한다.	현재 상태는 적정한가. 0~1023으로 환산한 수치가 나온다. 900을 기준으로 센서값이 기준 이상 시 수분이 공급
워터펌프가 동작하는지 확인한다.	워터펌프를 통해 수분이 공급되는지 확인한다.	1초간 수분 공급. 5초 뒤에 습도가 900미만이면 1초간 수분 공급
화분에 주어지는 밝기를 측정한다.	LED를 통해 식물에 빛을 쐬어준다.	빛의 밝기는 적절한가. 화분에 적정하게 비추고 있는지 확인. 320(일반공간 조도)을 8단계로 세분화하여 점등.
카메라를 통해 해충 등의 움직임을 측정	움직임이 포착되면 블로우팬을 통해 바람이 나오는지 확인.	실험적으로 측정 경계값을 조정하여 평가. 측정 불필요한 움직임을 배제하여 작동하는지 확인.

표 1

3. 분석 방법

3.1 아두이노1(토양수분센서+ 워터펌프 모터)

```
int Sensor_pin = A1;


int in3 = 7;
int in4 = 6;


int limit_val = 900;
int motor_speed = 127;


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); //센서와 보드간 시리얼 통신
```

```
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  //int data = analogRead(Sensor_pin); //0~1023 범위로 환산된 저항값으로 나옴 (0일수록 습함, 1023일수록 건조)

  Serial.print("습도값은 ");
  Serial.print(data); //시리얼모니터에서 토양수분 정도 숫자 확인

  if(data>limit_val)
  {
    Serial.println("물 들어갑니다");
    start_pump();
    delay(1000);
    stop_pump();
  }
  else
  {
    Serial.println("충분해요");
  }

  delay(5000);
}
```

```
void start_pump() //펌프 동작
{
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void stop_pump() //펌프 정지
{
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
```

3.2 아두이노2(조도센서+LED점등)

```
#include <Adafruit_NeoPixel.h>
#include <Wire.h>
#include <BH1750.h>

BH1750 lightmeter;

int led = 2;
int NUM_LEDS = 8;

Adafruit_NeoPixel stick_led = Adafruit_NeoPixel(NUM_LEDS, led, NEO_GRB + NEO_KHZ800);
```

```
void setup() {
  Serial.begin(9600);
  Wire.begin();
  lightmeter.begin();

  stick_led.begin();
  stick_led.show();
}

void loop() {
  int lux = lightmeter.readLightLevel();

  Serial.print("현재 조도값은 ");
  Serial.print(lux);
  Serial.println(" lx");

  int off_num =lux/40;

  if (off_num >= 8 )
  {
    led_on(0);
    Serial.println("0개 LED 켜짐");
  }
  else
  {
   led_on(NUM_LEDS-off_num);
   Serial.print(NUM_LEDS-off_num);
```

```
    Serial.println("개 LED 켜짐");
  }
}

void led_on(int on_num)
{
  for (int i=0; i<on_num; i++)
  {
    stick_led.setPixelColor(i,10,10,10);
  }

  for (int i=NUM_LEDS-1; i>=on_num; i--)
  {
    stick_led.setPixelColor(i,0,0,0);
  }
  stick_led.show();
}
```

3.3 라즈베리파이(카메라 + 팬 모터)

```
import cv2
import numpy as np
import time
import RPi.GPIO as GPIO
import threading

GPIO.setmode(GPIO.BCM)     #Pin 설정
GPIO.setup(16, GPIO.OUT)
GPIO.setup(20, GPIO.OUT)
```

```
GPIO.setup(19, GPIO.OUT)
GPIO.setup(26, GPIO.OUT)

GPIO.output(16,False)
GPIO.output(20,False)
GPIO.output(19,False)
GPIO.output(26,False)

flag_exit = False

def diffimage(i):  #이미지를 비교하여 연산
    diff0 = cv2.absdiff(i[0], i[1])
    diff1 = cv2.absdiff(i[1], i[2])
    return cv2.bitwise_and(diff0,diff1)

def getgrayimage(cam):  #color 이미지를 gray로 바꿈
    image = cam.read()[1]
    gray_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    return gray_image

def updateimage(cam, i):  #이미지 비교대상을 바꿈
    i[0] = i[1]
    i[1] = i[2]
    i[2] = getgrayimage(cam)

def camera_f():  #카메라에 움직임을 포착하여 cnt값을 설정
    global cnt
    thresh = 64
```

```
    cam = cv2.VideoCapture(0)

    i = [None, None, None]
    for n in range(3):
        i[n] = getgrayimage(cam)

    while True:
        diff = diffimage(i)
        ret, thrimg = cv2.threshold(diff, thresh, 1, cv2.THRESH_BINARY)
        count = cv2.countNonZero(thrimg)

        if(count > 10):
            cnt = 1
        else:
            cnt = 0

        updateimage(cam, i)

        if flag_exit: break; #키보드 인터럽트 시 종료


def detect_f():  #설정된 cnt값으로 움직임이 감지되면 팬 모터 동작
    global cnt

    while True:
        if(cnt == 1):
            print("detect!!!")
            GPIO.output(16, False)
            GPIO.output(20, True)
```

```
            GPIO.output(19, False)
            GPIO.output(26, True)
        else:
            print("0")
            GPIO.output(16,False)
            GPIO.output(20,False)
            GPIO.output(19,False)
            GPIO.output(26,False)

        time.sleep(0.05);

        if flag_exit: break;  #키보드 인터럽트 시 종료


if __name__ == "__main__":
    cnt = 0
    #효율적 동작을 위해 멀티스레드 이용
    th1 = threading.Thread(target = camera_f)
    th1.start()
    th2 = threading.Thread(target = detect_f)
    th2.start()

    try:
        while True:
            time.sleep(2.0);
    except KeyboardInterrupt:
        pass

    flag_exit = True
```

th1.join()

th2.join()

4. 최종 설계

4.1 설계도

4.1.1 아두이노 - 토양수분센서 & 워터펌프

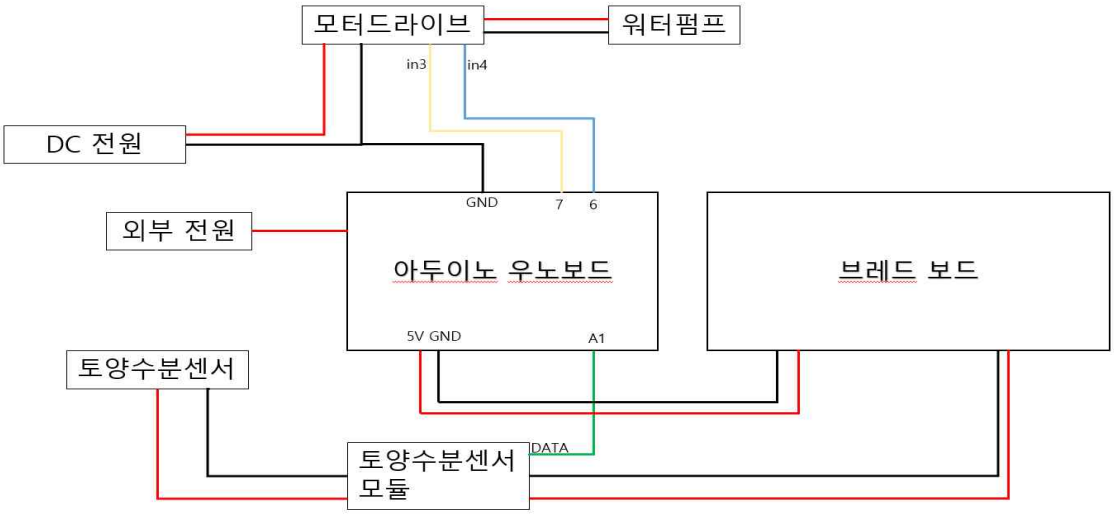

Fig. 3

4.1.2 아두이노 - 조도센서 & LED

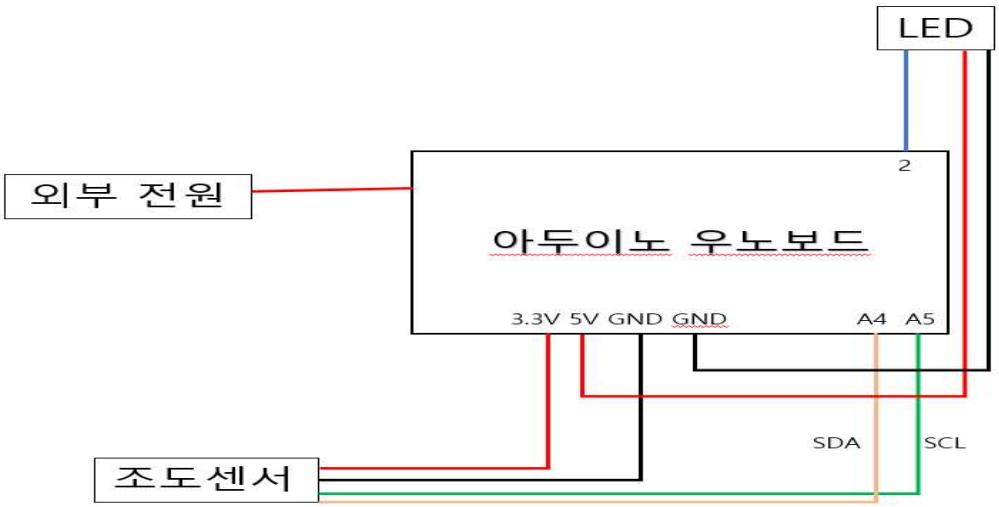

Fig. 4

4.1.3 라즈베리파이 – 카메라 & 팬모터

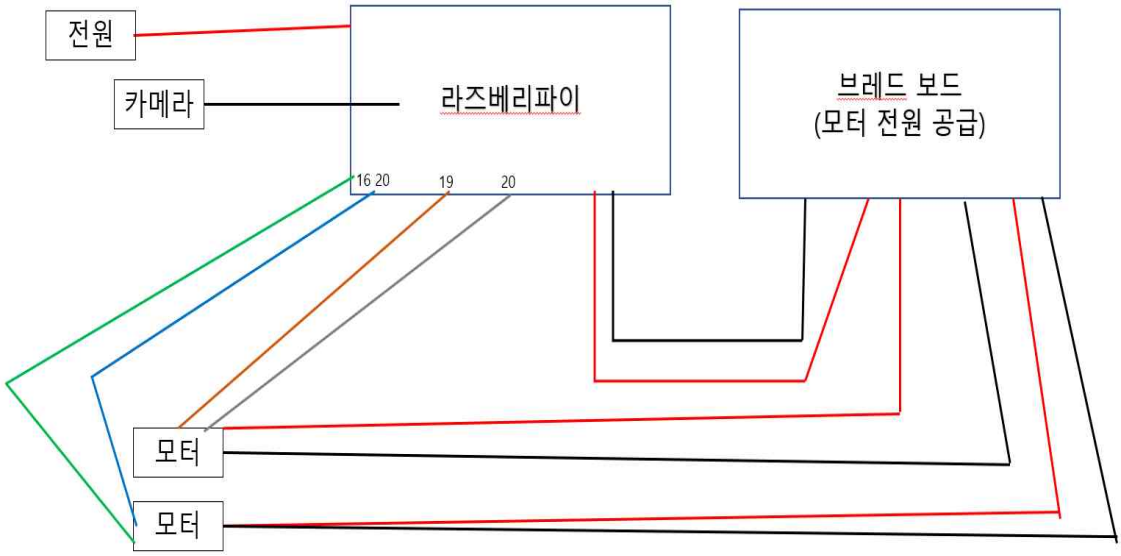

Fig. 5

4.1.4 외형 설계 – Rhinoceros 툴 사용

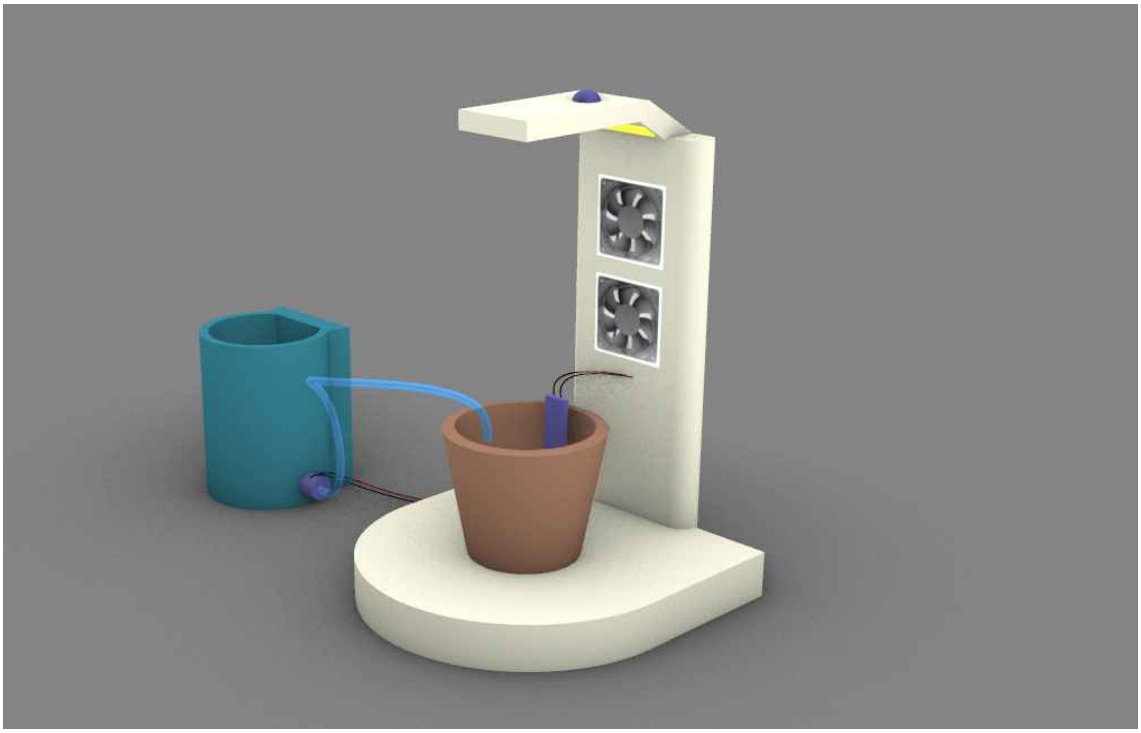

Fig. 6

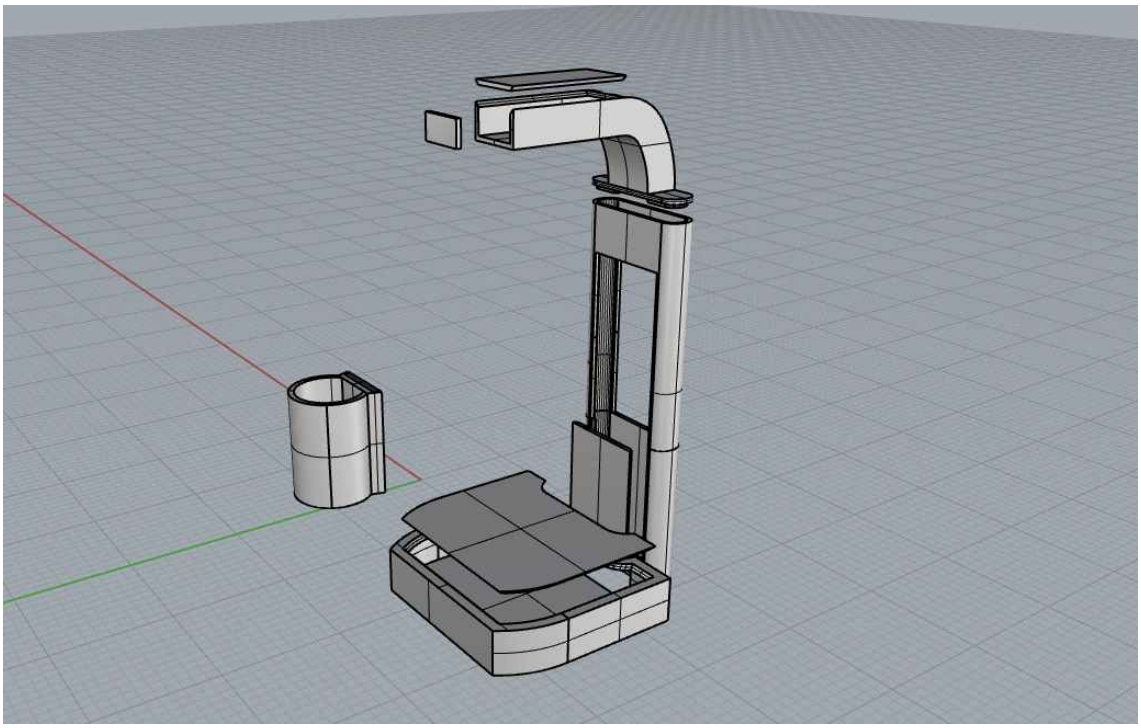

Fig. 7

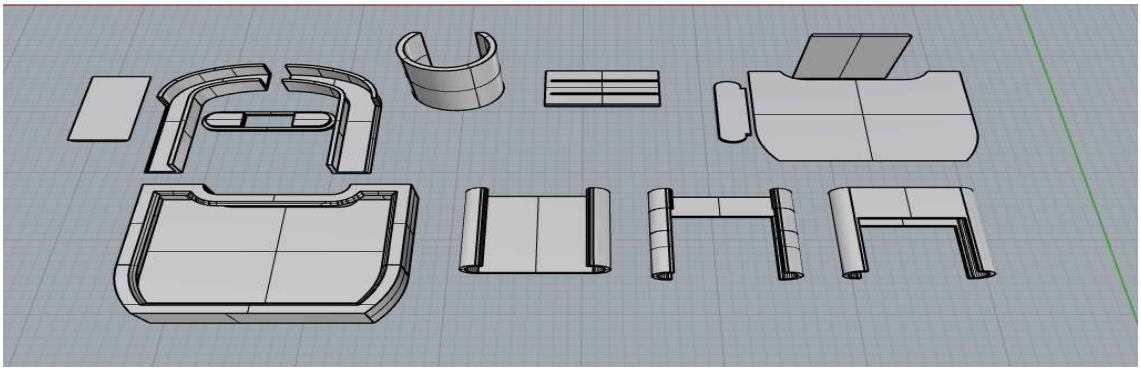

Fig. 8

- 외형 기본 설계
- 부품 조립 및 배선 작업을 위한 세부 설계 & 수치 적용
- 3D 프린팅을 위한 파츠 분할

4.2 부품목록 및 재료비 명세

품종	규격	단가(원)	수량	금액(원)
라즈베리파이 800만화소 카메라		28,000	1	28,000
아두이노 AA 배터리 케이스 6구 홀더		1,200	1	1,200
모터드라이브		1,500	1	1,500
아두이노 우노 보드		6,100	1	6,100
토양수분 센서		500	1	500
조도 센서		5,800	1	5,800
점퍼선 케이블 MM (수수)		900	1	900
점퍼선 케이블 MF (암수)		850	1	850
점퍼선 케이블 FF (암암)		50	1	850
RGB 네오픽셀 LED		1,980	1	1,980
온·습도센서		3,900	1	3,900
미니 모터펌프		1,000	1	1,000
1X4 건전지 케이스		2,200	2	2,200
소형 브레드보드 400핀		1,300	2	1,300
모터 팬모듈		8,600	2	8,600
워터펌프 실리콘 호수		2,400	1	2,400
라즈베리파이4B		79,000	1	79,000
SD카드		18,450	1	18,450
라즈베리파이 4 C타입 아답터		6,360	1	11,360
아두이노 우노 보드		6,100	1	6,100
라즈베리파이 500만화소 카메라		9,600	1	9,600
모터 팬 모듈		4,300	2	8,600
순간접착제, 스테인리스 직자		15,300	1	15,300
사포, 봉투		950	1	950
배송비				5000
총 계 (원)				221,440

< 표 2 >

5. 결과분석

기존 스마트 화분에서의 기능보다 조금 더 발전된 형태의 스마트 화분으로써, 기존의 틀을 벗어나 새로운 기능을 더 추가하고자 하였다. 움직임을 포착하여 모터 팬을 작동시키는 방법은 다른 스마트화분과 차별성이 있다. 하지만 기존에 유사한 제품이 나와있기 때문에 좀 더 차별성을 두기 위해선 다른 기능을 추가하거나 앞으로 발전에 대해서 좀 더 생각해 봐야한다.

6. 향후일정

캡스톤 디자인 프로젝트를 통해 제작품을 완성하였지만, 여기에 그치지 않고 좀 더 효과적인 식물케어의 방법이 있을지 연구해봐야겠다고 생각했다. 이번에는 구현하지 못한 온,습도 센서를 이용해 환경을 개선하는 방안에 대해서도 생각을 해볼 것이고, 무조건 기능을 추가한다고 좋은 것이 아니라, 기존에 있는 기능을 보완하여 효과적인 방법으로 구현해야겠다고 생각한다. 기능 추가를 한다면 모듈식으로 추가로 센서나 장비를 부착하는 형태로 구성하고 싶다.

7. 파급효과

실내활동이 엄청나게 늘어난 지금 반려 식물이라는 말이 있을 정도로 가정에서 식물 재배가 늘어나고 있고 마음의 안정을 얻는 사람들도 늘어나고 있다고 생각해서 가정에서 손쉽게 그리고 바빠서 신경을 쓸수 없는 시간에도 반려 식물을 케어해주는 시스템으로 접근성이 늘어날 것이라 생각한다. 가정용 스마트 식물 케어이지만, 더 나아가서 하나의 화분을 관리하는 것부터 규모 있는 스마트 팜 같은 분야까지 확장할 수 있는 가능성이 있다고 생각한다. IOT를 결합한다면 좀 더 편리한 구성을 만들 수 있을 것이다.

8. 결론

지금까지 배우고 느꼈던 것들을 머릿속으로만 생각하는 것이 아니라 실제로 적용해보는 의미 있는 시간이었다고 생각한다. 사회 현상이나 문제를 찾아 거기서 개선할 점이나 새로운 아이디어를 찾아내서 지금까지 배웠던 공학 지식을 이용해서 처음부터 끝까지 아이디어를 구상하고 제작 할 수 있는 기회는 드물기 때문에 이번 기회를 통해서 한 단계 성장할 수 있는 기회였던 것 같다. 아이디어 구상단계에서부터 시중에서 나와 있는 제품과 다른 새로운 아이디어를 생각한다는 것, 그리고 사용자의 needs를 맞추는 것 자체가 쉽지 않은 일이었다. 혼자 하는 일이라면 힘들었겠지만 팀원들과 함께 구상하고 각자 잘하는 분야를 나눠서 연구함으로써 효율적인 진행이 가능했고 좋은 결과를 낸 것 같아 감사한 생각이 들었다. 이론적으로만 끝났던 학습이 이번 기회를 통해 앞으로 필드에서 산업을 바라보는 관점과 문제점을 생각하고 해결할 수 있는 함량을 기른 것 같다.

참고 문헌

1. 김성우, “사물인터넷을 품은 라즈베리 파이(개정판)”, 제이펍, 380p 381p, 2020.08.12

2. 서민우, “진짜 코딩하며 배우는 라즈베리파이4”, 앤써북, 142p ~ 152p, 2020.08.25.

ABSTRACT

Smart Plant Care for Home

Bae Yoon Ho
Dept. of Electronic Eng.
Undergraduate School
Dongguk University
SEOUL KOREA

Recently, as the number of single-person households has increased, interest in raising pet plants is rapidly spreading around the younger generation. There are also cases where plants treat psychology. However, busy urban life, if you spend a lot of time away from home for a long time, plants wither due to run out of water or get high sunlight or get low sunlight. In addition, there are many cases where people want to raise indoors but cannot raise them due to lack of light, and where plants die due to damage from pests. In order to solve these problems and increase accessibility, even beginners will be able to make smart pots that can manage plants when they are away from home.

The most important thing in making smart pots was to maintain the proper conditions that the plants wanted to grow well. However, it is difficult for beginners or people who do not have enough time to keep the low-response plant cultivation in the eye. For this purpose, sensors and LED lights are used to check the conditions of plants such as temperature and soil humidity to secure necessary illumination. To protect plants from pests, the camera is equipped with a device to protect them from insects by using a blowpan when the camera is in motion. In addition, water canisters and water pumps are connected so that soil moisture can be measured through soil moisture sensors and an appropriate amount of water is supplied to the pot at an appropriate time to manage the plant. At this time, two control devices were used: water supply and light were controlled by Arduino and the camera was controlled by Raspberry Pi. Furthermore, if it is partially supplemented, I think there is a possibility that it can expanded from managing a single pot to areas such as a smart farm with large scale. If IOT is combined, we can create a more convenient configuration.