**SMARTFARM-Kiwoom**



|  |  |
| --- | --- |
| **팀 이름** | 전원일기 |
| **이름** | 이태현 |
| **E-mail** | dlxogus96@naver.com |
| **Github** | https://github.com/CoreanAnt/iot\_project.git |
| **Phone** | 010-2270-8807 |
| **이름** | 김명석 |
| **E-mail** | audtjr37@naver.com |
| **Github** | https://github.com/kms19910502/audtjr37 |
| **Phone** | 010-7626-1154 |
| **이름** | 정준호 |
| **E-mail** | Junho03415@naver.com |
| **Github** |  |
| **Phone** | 010-2528-0159 |
| **한국직업능력교육원 안산** | |

Contents

[**1.** **개요** 1](#_Toc130996632)

[1.1 프로젝트 개요 1](#_Toc130996633)

[1.2 프로젝트 명 1](#_Toc130996634)

[1.3 프로젝트 기간 1](#_Toc130996635)

[1.4 프로젝트 계획 및 제품 흐름도 2](#_Toc130996636)

[1.5 팀원 소개 3](#_Toc130996637)

[**2** **프로젝트 구성** 3](#_Toc130996638)

[2.1 기술 명세서 및 파츠 리스트 3](#_Toc130996639)

[2.2 운영 환경 구성 4](#_Toc130996640)

[**3. 제품 작동 시연** 1](#_Toc130996641)

[**4. 제품 상세 내용** 2](#_Toc130996642)

[4.1토양 습도 및 조도 측정 2](#_Toc130996643)

[4.2대기 온/습도 측정 2](#_Toc130996644)

[4.3 IR 신호 수신 2](#_Toc130996645)

[4.4화재 감지 3](#_Toc130996646)

[4.5수분 공급 3](#_Toc130996647)

[4.6 QT 실시간 센서 값 업데이트 4](#_Toc130996648)

[4.7 웹 서버 스트리밍 4](#_Toc130996649)

[4.7 웹 서버 데이터 제공페이지 5](#_Toc130996650)

[4.8 제품 외형 제작 6](#_Toc130996651)

[4.9 회로 설계 및 제작 6](#_Toc130996652)

[**5. 프로젝트 후기** 7](#_Toc130996653)

[**6. 참고 서적** 8](#_Toc130996654)

# **개요**

## 프로젝트 개요

2023년 현대 사회를 살아가는 사람들은 시간이 부족한 삶을 살아가고 있습니다. 그리고 작물을 키우는 것에는 회의감을 느끼는 사람들도 있습니다. 이에 외로운 삶을 살아가지만, 반려동물을 키우기에는 쉽지 않은 사람들을 위해 반려식물을 제시합니다.

또한, 근본적인 팜의 목적인 식물을 성장시키는 것에도 스마트팜은 분명 역할이 있다고 생각합니다. 최근 들어 한국에서 중국의 미세먼지, 토양오염 등으로 인해 자신이 먹는 작물에 걱정을 가지시는 사람들이 점점 늘고 있습니다. 이에 자기 자신이 직접 보는, 그리고 또 자기 자신이 직접 키우는 환경에서 성심성의껏 키운 작물을 식용할 수 있게 도와주는 자동화 시스템을 만들어서 사용자로 하여금 안전한 먹거리 제공과 노력의 가치를 높여 더욱 보람찬 하루를 제공하고 싶었습니다.

그렇기에 우리는 우리가 만드는 스마트팜은 작은 규모의 시스템을 구축하여 공간적 제약이 있는 환경에서도 작물 재배를 가능하게 하는 것에 중점을 두었고, 자동화 기술을 활용하여 효율적인 작물 관리를 목표로 프로젝트를 진행하게 되었습니다.

프로젝트의 최종 목표는 남녀노소 모두가 작동시킬 수 있는 쉬운 시스템을 만드는 것이며, 사용자가 식물 성장을 성공적으로 마칠 수 있게 하는 것이 목표입니다. 사용자는 성공적인 식물성장을 통해 자존감 증진되고, 이는 사용자에게 건강한 하루를 제공할 것으로 생각합니다.

## 프로젝트 명

“키움"이라는 단어는 한국어로 "성장"이라는 뜻을 가지고 있습니다. 이러한 의미는 농업 분야에서도 매우 중요합니다. 농가가 생산하는 작물이 건강하게 자라고 풍성하게 열매를 맺으려면, 적절한 환경과 농작물의 상태를 체계적으로 관리하고 지속적으로 성장을 유지해야 합니다.

또한 "키움"이라는 단어는 스마트 팜에서 사용되는 최신 기술을 활용하여 농작물 생산성과 수익성을 높이는 것과도 관련이 있습니다. 스마트 팜은 센서, 인터넷 연결 기술, 빅데이터 등을 활용하여 농작물의 상태와 생산 환경을 실시간으로 모니터링하고, 그 정보를 기반으로 효율적으로 농작물을 관리할 수 있습니다.

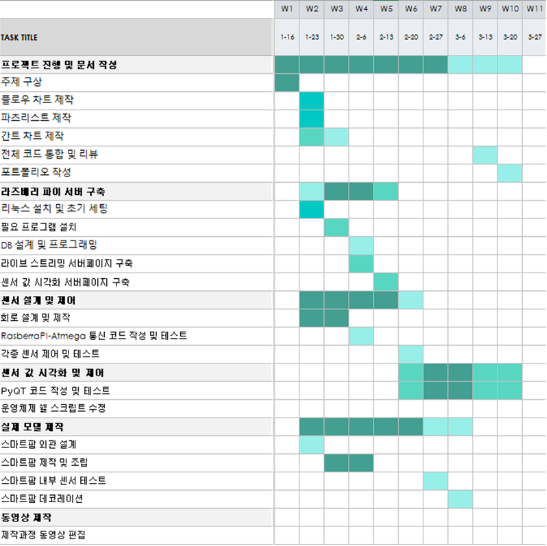
따라서 "키움"이라는 이름은 스마트 팜에서 농작물의 성장과 생산성을 높이는 것을 강조하면서, 동시에 최신 기술을 활용하여 농업을 혁신하는 데에 적합한 이름이라고 생각했습니다.

## 1.3프로젝트 기간

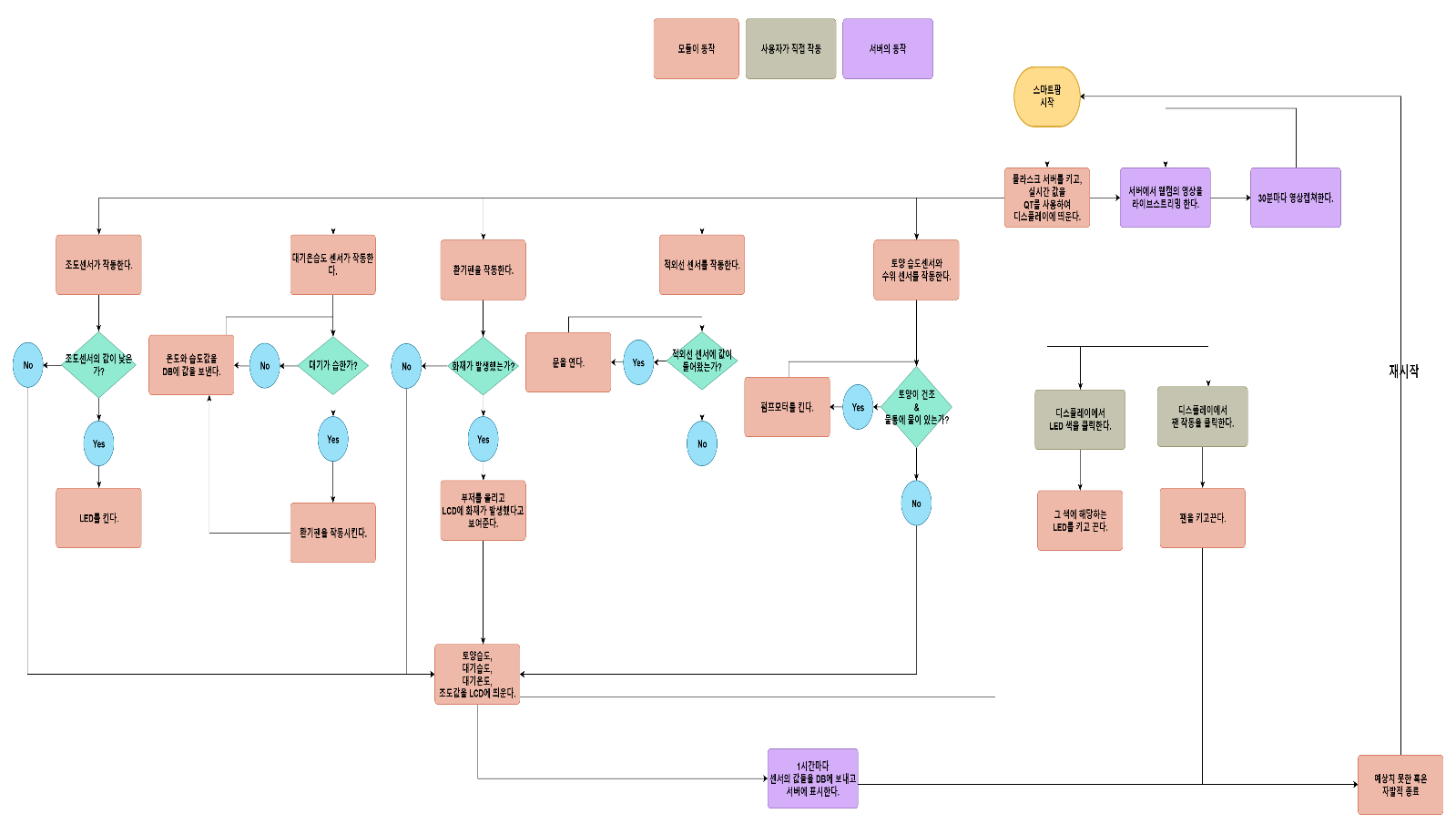
2023년 01월 19일 ~2023년 3월 29일

## 1.4프로젝트 계획 및 제품 흐름도

**- Gantt Chart**



**- Flow Chart**



## 팀원 소개

**- 팀원 별 기술 스펙 및 프로젝트 담당 직무**

|  |  |  |
| --- | --- | --- |
| 성명 | 기술 스펙 | 프로젝트 담당 직무 |
| 이태현 | - OpenCV  - Flask  - Python, C  - SQL 쿼리 구문  - 기구 설계(3D modeling)  - AVR 코드작성 가능 | -기구 설계  -서버 구축 |
| 김명석 | - Python, C  - QT5  - SQL 쿼리 구문  - AVR 코드작성 가능 | -센서 제어  -데이터 시각화 |
| 정준호 | - C  - EasyEDA  - 워드, PPT 문서 작성 | 회로설계  문서작성 및 편집 |

# **프로젝트 구성**

## 2.1 기술 명세서 및 파츠 리스트

**- 기술 명세서**

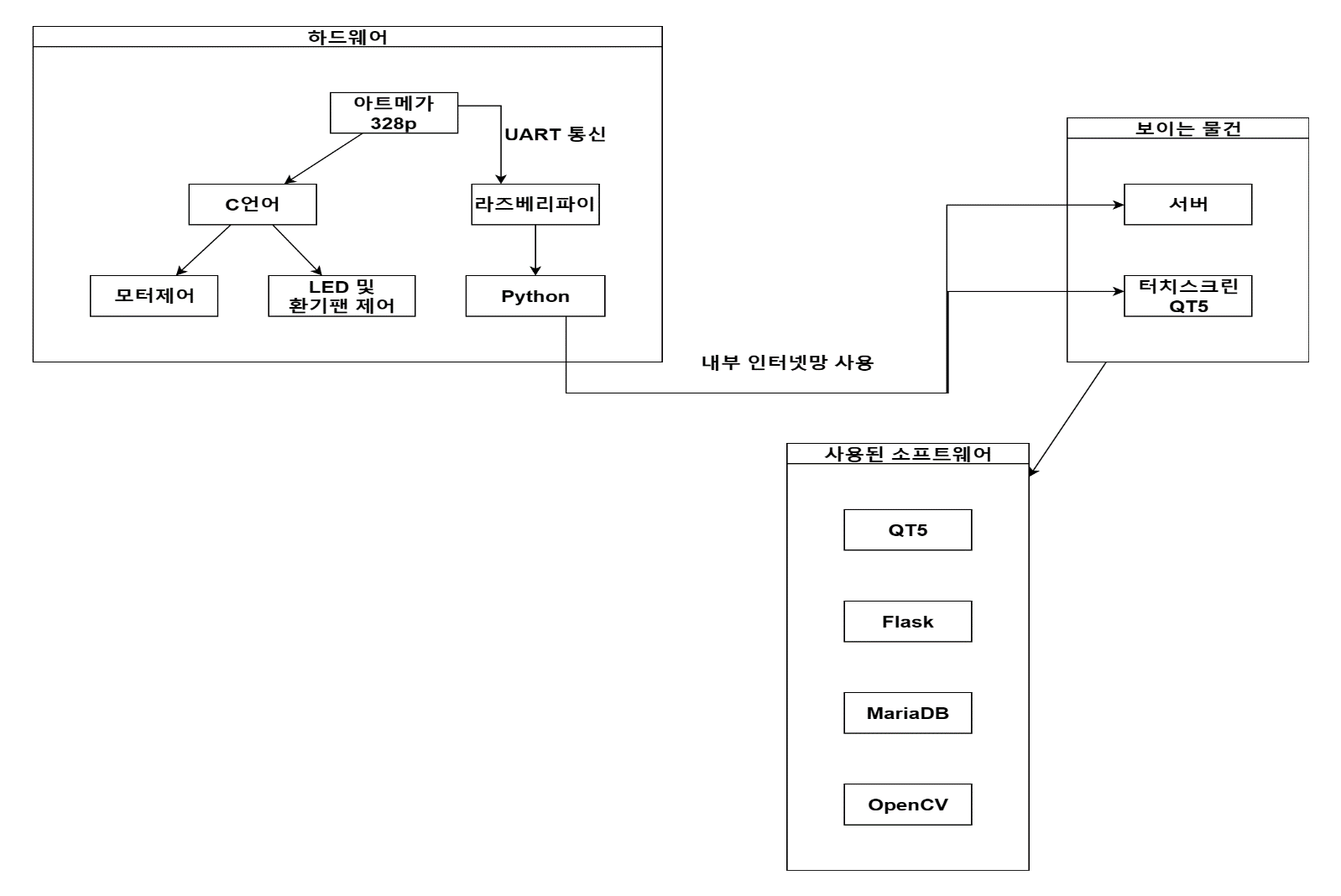
|  |  |  |
| --- | --- | --- |
| 사용 프로그램 | 버전 | 사용 목적 |
| Python | 3.9.2 | Atmega328 UART 통신, LED및 환기 팬 제어 |
| Atmel studio7 | 7.0.2542 | UART통신및 온/습도, 토양 습도, 조도, 불꽃 감지,  수위 센서와 수중 펌프 제어 |
| QT5 | 5.9 | 데이터의 시각화 및 led및 환기 팬 수동 조작 |
| OpenCV | 4.5.5 | 실시간 영상 수신 및 30분 마다 캡쳐 |
| mariaDB | 15.1 | 대기 온/습도, 토양 습도의 측정값 저장 |
| Flask | 1.1.2 | 웹서버 구축 |
| blender | 3.4 | 3D 기구 설계 |
| easyEDA |  | 회로 설계 |

**- 파츠 리스트**

|  |  |  |
| --- | --- | --- |
| 하드웨어 | 모델명 | 사용 목적 |
| 조도 센서 | SEN030101 | 스마트 팜 내 광량을 확인하기 위한 센서 |
| 대기온습도센서 | DHT11 | 대기 중 온/습도를 분석을 통해 식물 성장에 도움을 주기 위함 |
| 토양습도센서 | Keystudio soil sensor | 토양 내 물기를 측정하여 물을 주기 위함 |
| 적외선 센서 | Arduino IR Sensor | 스마트 팜의 개폐를 위해 리모컨 작동 |
| 수위감지센서 | MLDU01A | 물통에 물이 없을 때 모터의 공 회전 방지를 위한 센서 |
| 환기 팬 | NF-A12x15 FLX | 식물 성장을 위한 통풍을 해주기 위함 |
| 부저 | Arduino Passive Buzzer | 불이 났을 때 소리가 울려 사용자에게 알려주기 위함 |
| 스태핑모터 | 28BYJ-48 | 리모컨 적외선이 감지되면 문 개폐를 위한 모터 |
| 펌프 모터 | SZH-GNP155 | 식물에 물에 물을 주기 위한 모터 |
| LED | 슬림 LED바 | 어두울 때 식물에 빛을 주기 위함 |
| Camera | TB189 | 실시간으로 식물의 성장을 관찰하기 위함 |
| 터치스크린 | Raspberry-Pi Touch Display | 실시간 데이터 시각화 및 간단한 작동을 위함 |

## 2.2 운영 환경 구성

**- 구성도**



**- 구성 내용**

|  |  |  |
| --- | --- | --- |
| 구성 | | 구성 내용 |
| RaspberryPi | QT5 | 센서 값의 시각화 및 led, 환기 팬의 수동 제어 |
| flask | 웹 서버 구축 |
| mariaDB | 센서 값의 저장 |
| opencv | 실시간 영상 스트리밍 및 30분 단위 캡쳐 |
| atmega | C언어 | 대기 온/습도 및 조도, 토양 습도, 불꽃 감지, 수위 센서 값을 측정 및  토양습도값의 따른 수중 펌프 모터 제어 |

# **3. 제품 작동 시연**

- 제품 시연 영상 Github : https://github.com/CoreanAnt/iot\_project.git

- 제품 사진

|  |  |
| --- | --- |
| 제품 상태 | 실행 상태 |
| 1. 정면      1. 측면      1. 후면 | 1. QT를 이용한 데이터 시각화      1. 플라스크를 이용한 웹 서버 실시간 스트리밍      1. 플라스크를 이용한 웹 서버 DB 데이터 열람 |

# **4. 제품 상세 내용**

## 4.1 토양 습도 및 조도 측정

//조도 측정

unsigned short ReadADC1(void)

{

ADMUX = 0x43; // ADC3을 사용할 것이다.

ADCSRA |= (1 << ADSC); // 변환 시작

while (ADCSRA & (1 << ADSC));

return ADC;

}

//토양 습도 측정

/\* ADC channel 2값을 읽어서 리턴 하는 함수 \*/

unsigned short ReadADC2(void)

{

ADMUX = 0x44; // ADC4를 사용할 것이다.

ADCSRA |= (1 << ADSC); // 변환 시작

while (ADCSRA & (1 << ADSC));

return ADC;

}

## 4.2 대기 온/습도 측정

//read the data

for (j=0; j<5; j++)

{

*uint8\_t* result=0;

for(i=0; i<8; i++)

{

while(!(DHT11\_PIN & (1<<DHT11\_INPUTPIN)));

*\_delay\_us*(30);

if(DHT11\_PIN & (1<<DHT11\_INPUTPIN))

result |= (1<<(7-i));

while(DHT11\_PIN & (1<<DHT11\_INPUTPIN));

}

bits[j] = result;

}

## 4.3 IR 신호 수신

uint8\_t read\_ir() //IR신호 수신

{

uint8\_t pulse\_width = 0;

int timeout = 100;

while(PINC & (1<<IR\_IN\_PIN) && timeout) // Wait for IR signal to start

{

\_delay\_us(5000);

timeout--;

}

while(!(PINC & (1<<IR\_IN\_PIN))) // Measure IR signal pulse width

{

\_delay\_us(5000);

pulse\_width++;

}

return pulse\_width;

}

## 4.4 화재 감지

// 불꽃 감지 센서 모듈에서 신호를 감지했는지 확인

if (PINC & (1 << SENSOR2\_PIN))

{

*sprintf*( p, "fire : 0 \n" ); //불꽃 미 감지

Puts0( p );

// 불꽃이 감지되지 않았다면 부저 끄기

PORTD &= ~(1 << BUZZER\_PIN);

DDRD &= ~(1 << BUZZER\_PIN);

*\_delay\_ms*(1000);

}

else

{

*sprintf*( p, "fire : 1 \n" ); //불꽃감지!

Puts0( p );

// 부저 핀을 출력으로 설정

init\_pwm();

for(int r=1;r<=5;r++)

{

// 불꽃이 감지되었다면 피에조 부저 울리기

play\_tone(500, 10000);

}

}

## 4.5 수분 공급

if (PINC & (1 << SENSOR1\_PIN)) // 수위 센서 핀이 LOW인 경우

{

sprintf( p, "water : 0 \n" ); //물이 부족합니다.

Puts0( p );

\_delay\_ms( 500 );

PORTD |= (1 << RELAY4\_PIN);

}

else

{

sprintf( p, "water : 1 \n" );

Puts0( p );

\_delay\_ms( 500 );

if(ReadADC2()<440 )//수중 펌프 동작 관련

{

PORTD &= ~(1 << RELAY4\_PIN);

\_delay\_ms( 10000 );

PORTD |= (1 << RELAY4\_PIN);

}

else

{

PORTD |= (1 << RELAY4\_PIN);

}

}

## 4.6 QT 실시간 센서 값 업데이트

def update\_data(self):

data = self.ser.readline().decode()

now = datetime.datetime.now()

if "temperature:" in data:

temperature = int(data.split(":")[1])

temperatureValue = temperature

self.temperature\_label.setText("Temperature : " + str(temperatureValue))

self.temperature\_label.setStyleSheet("color: blue;"

"background-color: #87CEFA;"

"border-style: dashed;"

"border-width: 3px;"

"border-color: #1E90FF")

self.temperature\_label.adjustSize()

if now.minute == 0 and now.second < 10:

dC.insertSensor1(temperatureValue)

(생략)

if \_\_name\_\_ == "\_\_main\_\_":

app = QApplication(sys.argv)

window = SmartFarm()

while True:

window.update\_data()

sys.exit(app.exec\_())

## 4.7 웹 서버 스트리밍

CAPTURES\_DIR = os.path.join('static', 'captures')

if not os.path.exists(CAPTURES\_DIR): # 캡쳐된 이미지 저장경로

os.makedirs(CAPTURES\_DIR)

fourcc = cv2.VideoWriter\_fourcc(\*'XVID') # 인코딩 포맷

def video\_stream():

cam = cv2.VideoCapture(0)

cam.set(cv2.CAP\_PROP\_FRAME\_WIDTH, 1920)

cam.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, 1080) # 해상도

capture\_interval = 1800 # 캡쳐 간격 (초)

last\_capture\_time = time.time()

while True:

ret, frame = cam.read() # 매 초마다 스트리밍을 수행

if not ret:

break

# 캡쳐 간격이 되면 프레임을 저장

if time.time() - last\_capture\_time >= capture\_interval:

last\_capture\_time = time.time()

\_, jpeg = cv2.imencode('.jpg', frame) # JPG 이미지로 인코딩

now = datetime.datetime.now()

filename = now.strftime('%Y%m%d\_%H%M%S.jpg') # 이름 설정

filepath = os.path.join(CAPTURES\_DIR, filename)

with open(filepath, 'wb') as f:

f.write(jpeg.tobytes())

# 이미지 스트리밍

\_, jpeg = cv2.imencode('.jpg', frame)

yield (b'--frame\r\n'

b'Content-Type: image/jpeg\r\n\r\n' + jpeg.tobytes() + b'\r\n')

# frame은 이미지 받고 다음 응답 받는데 사용

Content-Type 데이터 저장 및 구분

+ jpeg.tobytes() + b'\r\n은 이미지 데이터 전송완료 구문

@app.route('/video')

def video():

return Response(video\_stream(), mimetype='multipart/x-mixed-replace; boundary=frame')

multipart/x-mixed-replace -> 이미지 실시간 업데이트 구문

@app.route('/live')

def live():

flash('you can watch live plant')

return render\_template("live.html")

## 4.7 웹 서버 데이터 제공페이지

@app.route('/data')

def data():

flash('목록을 눌러 데이터를 확인하실 수 있습니다!')

return render\_template('data.html')

@app.route('/data1')

def data1():

conn = pymysql.connect(user='root',

passwd='1234',

host='localhost',

db='test',

charset='utf8')

cursor = conn.cursor()

cursor.execute("SELECT \* FROM temperature")

data\_list = cursor.fetchall()

conn.close()

return render\_template('data1.html',data\_list=data\_list)

## 4.8 제품 외형 제작

- 하단에 볼 캐스터를 부착하여 이동이 쉽도록 하였고 스텝 모터를 사용하여 문을 자동 개폐 할 수 있도록 설계하였습니다. 또한 터치스크린을 부착하여 실시간으로 데이터를 볼 수 있으며 수동으로 조작 하기 편리하도록 하였습니다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 도면 | | | | |
| 전체 |  | |  | |
| 기계실 |  | 스텝모터  커버 | |  |

## 4.9 회로 설계 및 제작

|  |
| --- |
|  |

- 크리스탈 추가(PB6, PB7) : 원활한 명령 처리를 위해 설치하고, 발진 오차 값에서 발생되는 노이즈를 해결하기 위하여 세라믹 캐패시터(22pF) 사용

**- easyEDA 을 이용하여 회로도 작성**

|  |
| --- |
|  |
|

# **5. 프로젝트 후기**

|  |  |
| --- | --- |
| 이태현 | 프로젝트를 제작하며 저는 “우리가 등한시하는 곳에 대한 새로운 시각”에 대한 생각을 해볼 수 있었습니다. 이 과정을 배우기 전에는 기계가 자동으로 작동하는 것에 대해 별 큰 감흥이 존재하지 않았지만, 이 프로젝트를 통해 타인과는 다른 시각으로 물건들을 바라볼 수 있는 경험을 얻어 보람찼습니다.  또한 스마트 팜 제작을 통해 제가 어느 부분에서 부족한 지 깨달을 수 있어 좋았습니다. |
| 김명석 | 삶의 전환을 위해 입문한 IOT의 길에서 이 프로젝트를 시작하면서 제가 정말 이걸 해낼 수 있을지 불안했으면서도 오랜만에 동심으로 돌아가 정말 즐겁게 무언가를 만들어 본 것 같습니다. 열심히 공부했던 지식들로 구상했던 것들이 하나 하나씩 이루어져 갈 때마다 불안함은 자신감으로 바뀌어 갔고 프로젝트를 완성했을 때는 무척 큰 성취감을 느낄 수 있었습니다. 비록 지식의 한계로 어려움에 부딪혔던 때도 있었지만 팀원들과 함께 극복해 가면서 IOT에 입문하길 정말 잘했다고 생각이 들었습니다. |
| 정준호 | 스마트 팜 프로젝트를 제작하면서 많은 것을 배우고 유익한 경험을 했습니다..  프로젝트를 통해 농작물 재배와 IoT 기술에 대한 이해도가 높아졌고, 하드웨어와 소프트웨어를 결합해 실제로 동작하는 시스템을 만드는 경험도 해볼 수 있었습니다.  프로젝트 제작과정에서는 하드웨어와 소프트웨어 모두에 대한 전문적인 지식이 필요했지만, 다양한 온라인 자료와 서적들을 활용하여 문제를 해결할 수 있었습니다. 또한, 프로젝트를 진행하면서 실패와 문제 해결 과정에서 많은 것을 배울 수 있었고, 자신의 아이디어를 실제로 구현해보는 경험은 매우 유익했습니다. |

# **6. 참고 서적**

-C 언어 입문

저자: 김성현

출판사 이지스퍼블리싱 ([싸니까 믿으니까 인터파크도서 - Do it! C언어 입문 (interpark.com)](https://book.interpark.com/product/BookDisplay.do?_method=detail&sc.prdNo=263638762&gclid=CjwKCAjwzuqgBhAcEiwAdj5dRtZCRlz6kEX4ZZ0jGGoojokeLDXdiMm4KXrn5F3XIHmMRdLtMzAq_RoC8ssQAvD_BwE))

-혼자 공부하는 파이썬

저자: 윤인성

출판사: 한빛 미디어 ([싸니까 믿으니까 인터파크도서 - 혼자 공부하는 파이썬 (interpark.com)](https://book.interpark.com/product/BookDisplay.do?_method=detail&sc.prdNo=354751685&gclid=CjwKCAjwzuqgBhAcEiwAdj5dRiXcAJ9SNCsLGtlG_IPUxp5rTjIpAg8KGpXJMDydpIs9iir5XghVrBoCSUgQAvD_BwE))

-알기 쉬운 파이썬 SQL 코딩하기

저자: 정현희

출판사: 크라운출판사([싸니까 믿으니까 인터파크도서 (interpark.com)](https://book.interpark.com/product/BookDisplay.do?_method=detail&sc.prdNo=354924550&gclid=CjwKCAjwzuqgBhAcEiwAdj5dRsrXiQ8QBwfaTdG0qKDDKtRvttupkeziYcsBgH7B6L5YPzwy6gocZBoC3MEQAvD_BwE))

-리눅스와 함께 하는 라즈베리 파이

저자: 피터 멤브리&데이비드 하우스

출산사: 제이펍 ([리눅스와 함께하는 라즈베리 파이 - YES24](http://www.yes24.com/Product/Goods/12212242))

-점프 투 플라스크

저자: 박응용

출판사: 이지스 퍼블리싱([Do it! 점프 투 플라스크 - YES24](http://www.yes24.com/Product/Goods/95751850))

-Atmega 328 프로그래밍

저자: 허경용

출판사: 제이펍([ATmega328 프로그래밍 : 네이버 도서 (naver.com)](https://search.shopping.naver.com/book/catalog/32467251777?query=328&NaPm=ct%3Dlfjhfh74%7Cci%3Dffa16d20060d7dafb4adf4e532b903a6b11608fe%7Ctr%3Dboksl%7Csn%3D95694%7Chk%3Ddb2474a36620328d4bedc81b4d40301057ad2e65))

-파이썬으로 만드는 나만의 GUI프로그램

저자: 김민휘

E-Book ([알라딘: [전자책] PyQt5 Tutorial - 파이썬으로 만드는 나만의 GUI 프로그램 (aladin.co.kr)](https://www.aladin.co.kr/shop/wproduct.aspx?ItemId=235430963))