

# 알아서 쑥쑥 (자동화 재배 시스템, 스마트팜)

박규태\*, 오광은\*, 오진섭\*  
\*전북대학교 IT정보공학과

gyu330@jbnu.ac.kr, familye@naver.com, toojs@jbnu.ac.kr

## Grow it self : Automated Cultivation System

Gyu-Tae Park\*, Gwang-Eun O\*, Jin-Seop Oh\*

\*Dept. of Information and Engineering, Jeonbuk National University

### 요 약

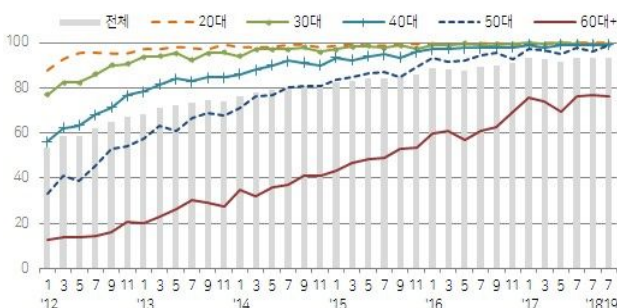
최근 스마트팜이 많이 등장하였지만 기술적인 한계로 인해 많은 제한을 받고 있으며, 이로 인해 사용자의 시각으로 보았을 때 스마트팜을 이용하는 사용자들은 아직까지도 많은 불편함을 겪고 있다. 이에 본 논문에서는 스마트팜에 어플과 웹을 연동하여 스마트팜을 더 편리한 환경에서 효율적으로 관리하고 사용할 수 있게 되기를 기대한다.

### 1. 서론

최근 농가 고령화와 농촌 일손 부족, 4차 산업 혁명 등에 대응해 농업 경쟁력을 높이려면 스마트팜을 활성화해야 한다는 목소리가 높아지고 있다. 하지만 스마트팜 관련 기술은 많은 개선이 필요하다. 첫 번째는 데이터 품질 확보와 표준화고 두 번째는 농민들의 이해도에 관한 문제이다[1]. 스마트팜은 현재 농업이 가지고 있는 많은 문제들을 해결할 수 있는 획기적인 기술이 될 수 있지만 현재는 앞에서 얘기한 것 처럼 많은 개선이 필요하다.

현재 스마트폰 사용률은 매우 높은 상태로 스마트폰을 이용한 시스템 구축은 긍정적인 전망이다. 그림 1을 보면 2019년 7월을 기준으로 스마트폰 전체 사용률은 93.2%이며, 20대에서 50대의 스마트폰 사용률은 100%라고 봐도 무방하고, 60대 이상 스마트폰 사용률은 76%이며 60대 이상 남성의 경우엔 90%이다[2].

© 2012년 1월 이후 스마트폰 사용률 추이 - 월별, 연령별



\*월별 조사 사례수는 최소 3,014명에서 최대 7,831명(표본오차±1.8~1.1%포인트, 95%신뢰수준)  
\*2019년 7월 사례수는 1,002명(표본오차±3.1%포인트, 95%신뢰수준)  
\*출수 월만 제시. 한국갤럽 데일리 오피니언 월별 종합 집계 결과 [www.gallup.co.kr](http://www.gallup.co.kr)

(그림 1. 스마트폰 사용률 추이)

또한 유선 인터넷 방식으로 인터넷에 접속할 수 있는 가구의 비율은 2019년 기준으로 81.6%이며 무선 인터넷 접속률을 합한 개념인 인터넷 접속률은 99.5%에 달한다[3]. 이는 스마트팜에 네트워크를 접속하는 것이 커다란 제약 사항되지 않을 것이라고 할 수 있다.

< OECD 주요 회원국의 가구 인터넷 접속률 (%) >

국가명	한국	네덜란드	아이슬란드	룩셈부르크	덴마크	노르웨이	스웨덴	핀란드
인터넷 접속률	99.5	98.0	99.2	93.0	92.7	96.0	92.1	94.3
	영국	스위스	독일	오스트리아	아일랜드	에스토니아	프랑스	벨기에
	94.8	-	94.4	88.8	89.1	90.5	88.6	87.3

(그림 2. OECD 주요 회원국의 가구 인터넷 접속률)

이에 본 논문에서는 앞에서 제시한 두 가지 문제에 대한 해답을 찾기 위해 아두이노를 통한 자동화 재배 시스템에 어플과 데이터베이스를 융합한 시스템을 설계하고 이를 구현했다. 데이터 품질 확보와 표준화를 위해 아두이노에서 측정한 데이터를 API를 통해 데이터 베이스에 쌓아 해결하고자 했고, 농민들의 편의성과 이해도를 증가시키기 위해 제품의 데이터를 조회하고 스마트팜을 제어하는 기능을 간단한 UI의 어플을 통해 구현함으로써 해결하고자 했다.

### 2. 관련연구

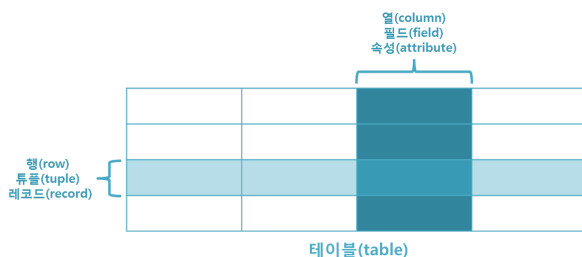
#### 2.1 Node.JS

Node.JS는 Chrome V8 Javascript 기반의 엔진으로 빌드된 런타임이다. 이벤트 기반, 논 블로킹 IO를 기반을 사용해 가볍고 효율적이다. Node.JS 패키지의 생태계는 npm은 세계에서 가장 큰 오픈소스 라이브러리이기도

하다[4].Node.JS는 REPL(Read, Eval, Print, Loop)를 통해 런타임을 제공한다. 하지만 실제 개발에서는 REPL만을 이용해서 개발하기 힘들다. 그래서 V8 엔진을 이용해서 스크립트를 해석해서 실행하는 형태로 보통 구성된다. 본래 Node.JS는 확장성있는 네트워크 애플리케이션을 목표로 나와 서버구현을 위해 주로 사용된다. 하지만 근래에는 VScode와 같은 프로그램들도 Node.JS를 이용해 만들어지는 등 네트워크 애플리케이션 뿐만 아니라 데스크탑 애플리케이션도 제작하게 된다. 본 논문에서는 백엔드 서버 구성에 사용하였다. 센서 데이터 측정과 제어를 담당하는 말단 기기에서 전달하는 데이터 저장 및 조회 기능등을 담당한다.

## 2.2 관계형 데이터베이스

데이터베이스는 여러사람이 공유하고 사용할 목적으로 관리되는 데이터의 집합이다. 논리적으로 연관된 하나 이상의 자료의 모음으로 내용과 구조를 개선하여 검색과 갱신에서 속도와 효율을 낼 수 있게 만든 것이다.[5] 데이터베이스는 실시간 접근성, 지속적인 변화, 동시 공유, 내용에 대한 참조, 데이터 논리적 독립성을 보장한다. 관계형 데이터베이스의 경우 테이블로 이루어져 있고 테이블에서는 Key와 Value 형태로 나누어서 저장하게 된다. 일반 데이터베이스에 비해 관계형의 경우 데이터의 분류, 정렬, 탐색 속도가 무척 빠르다. 신뢰성이 높고 데이터 무결성을 잘 보장한다.[6] 다만 기존 작성된 내용을 수정하기 힘들고 부하의 원인을 찾기 힘들다는 특징이 있다. 위의 기능을 이용하여 여러 기기에서 오는 데이터들을 나누어서 저장한다. 각 기기에 맞는 이름을 부여한 후 이름을 기준으로 데이터를 저장 및 조회가 가능하게 구성하여 빠르게 접근할 수 있게 제공한다.



(그림 3). 관계형 데이터베이스 형태

## 2.3 API(Application Programming Interface)

응용프로그램에서 사용할 수 있도록, 운영체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스이다. 시스템에서는 주로 화면 출력이나 창 제어, 화상처리와 같은 기능들을 사용할 수 있게 제공한다.[7] 근래에는 다양한 회사에서 자신들의 기능을 쉽게 사용하고 적용할 수 있도록 API형태로 제공한다. 앞의 사례들과 같이 본 논문의 시스템에서는 API를 이용해 센서 데이터 저장, 데이터 조회, 각 기기별 시스템 설정 기능을 제공한다. 이를 통해 서로 다른 시스템에서도 빠르게 기능 적용을 쉽게 하여 개발을 용이하게 하였다.

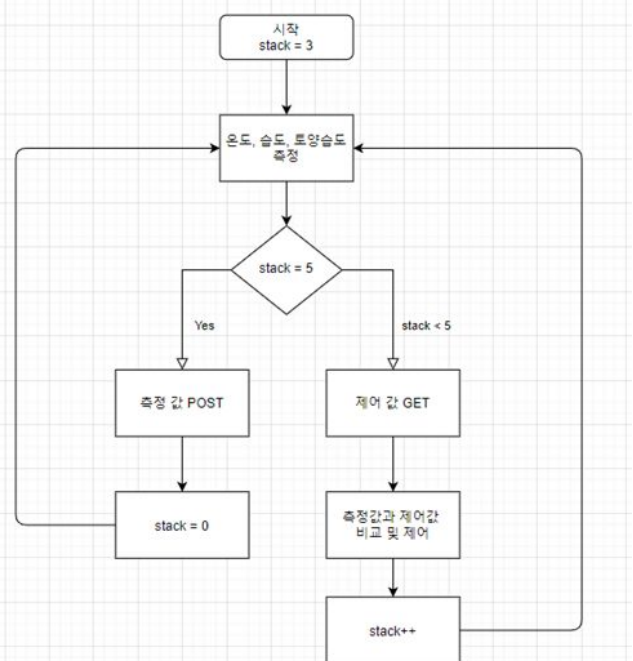
## 2.4 식물공장의 LED 인공광

식물 공장의 경우 태양과이용형 식물공장과 완전 제어형 식물공장으로 나눌 수 있다. 태양광이용형 식물공장은 온실을 근간으로 태양광을 기본광으로 이용해 재배하는 시스템이다. 반면 완전제어형 식물공장의 경우 완전 밀폐된 공간에서 인공광을 이용해 식물을 재배하는 것을 기본으로 하고 있다. 식물재배 인공광으로 LED사용이 시도 되고 있다. 다층재배의 경우 태양광으로는 충분한 광량 확보가 힘들다. 이러한 이유로 식물재배에 충분한 광을 확보하기 위해 인공광을 이용하는데 고압나트륨 조명의 경우 광량은 많지만 열방출이 많아 상단에 있는 식물은 고온 피해를 받기 쉽다. 그래서 광량은 많지만 열이 발생이 적은 LED조명을 사용할 경우 이러한 문제를 해결할 수 있다. 또한 파장 폭이 작고 단색광이므로 식물재배에 쉽게 사용할 수 있다. 광합성유효방사가 존재한다. 식물의 광합성에 이용되는 파장범위는 400nm ~ 700nm로 가시광선 범위인 380nm ~ 760nm와 유사한 파장 범위를 보인다. 식물분야에서는 가시광이라고 하지 않고 광합성유효방사라고 한다.[8] 일반적으로 청색, 녹색, 적색광이 유사한 파장을 보인다. 본 논문에서 언급된 기기에서도 적색광과 청색광을 사용한 인공광을 사용하였다.

## 3. 설계 및 구현

### 3.1 시스템 구조도

그림 3은 본 논문의 자동 재배 시스템의 기능 처리 순서이다.



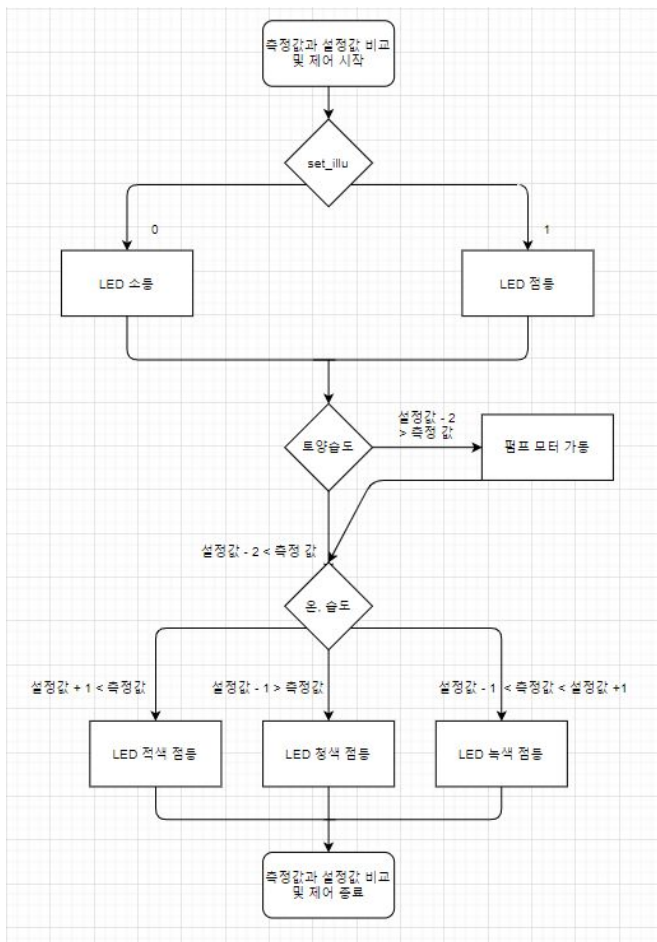
(그림 4) 기능 흐름도

시스템은 온도, 습도, 토양습도를 측정하고 이를 이용하여 재배환경을 일정하게 유지시켜 준다.

이과정에서 시스템은 stack의 값을 3으로 가지고 시작하며 온습도, 토양습도를 측정한 후 stack값이 증가한다. 측정한 값과 설정한 제어값을 비교하여 펌프, 온도 조절 장치를 on/off 하여 환경을 일정하게 유지한다. 증가하는 stack 값이 5가 되면 이 때의 측정된 값들을 서버에 전송하게 된다. 5:1의 비율로 제어와 기록을 실행하게 된다.

### 3.2 알고리즘 명세서

그림 4는 본 논문의 시스템에서 재배환경 유지에 사용되는 알고리즘이다.



(그림 5) 알고리즘 명세서

GET 방식으로 가져온 제어 값들을 센서에서 측정한 값들과 비교하고 제어하게 된다. 처음으로 set\_illu 값에 따른 LED 점등제어 수행한다. 다음으로 토양습도의 제어 값과 측정값 비교에 따른 신호로 모터의 동작유무를 결정한다. 이 때 수치 2의 여유를 두어 설정한 값에 원활히 도달할 수 있도록 한다. 온 습도의 제어 값과 측정값 비교 후 각 상황에 맞는 LED 점등한다. LED 점등의 신호를 온도 조절 장치의 on/off로 이용할 수 있다.

### 3.3 구현

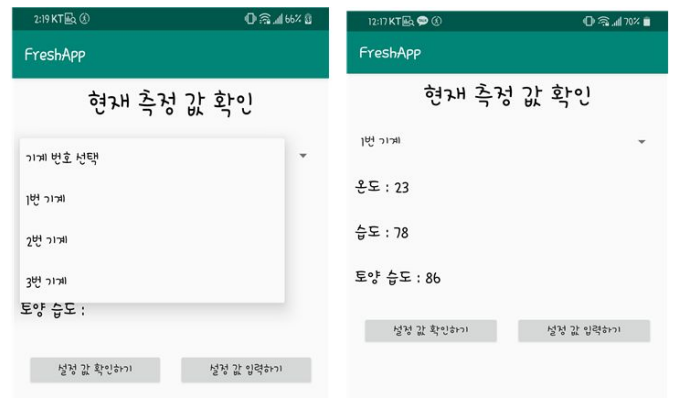
그림 5은 조립된 재배기이다.



(그림 6) 조립된 재배기

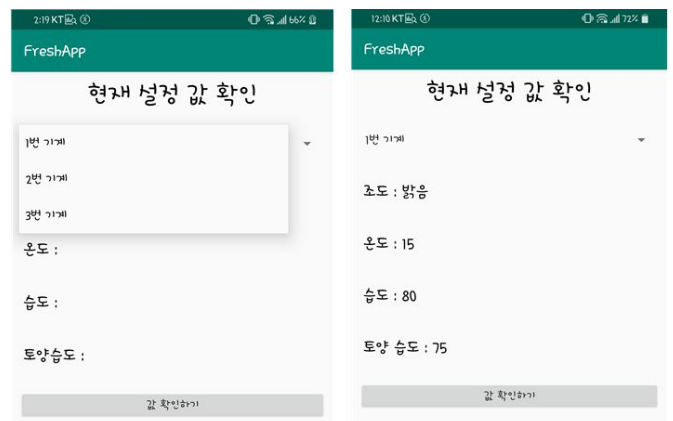
재배기에서 측정된 값들은 서버에 저장되기 때문에 어플리케이션을 통해 확인 할 수 있습니다.

그림 6은 현재 측정 값을 확인 할 수 있는 화면 입니다.



(그림 7) 현재 측정값 확인 화면

그림 7은 각 재배기마다 설정된 값을 확인할 수 있다.



(그림 8) 현재 설정값 확인 화면

다음 아래의 그림 8은 설정 값 입력 화면으로 이곳에서 밝기, 온도, 습도, 토양습도를 각 재배기 별로 설정 할 수 있다.

(그림 9) 재배기 설정 값 입력 화면

#### 4. 결과

본 논문에서는 스마트팜을 효율적으로 관리하기 위해 아두이노의 데이터를 조회하고 아두이노를 제어할 수 있는 어플과 이 데이터를 의미있게 사용할 수 있도록 측정된 데이터를 저장해 놓을 수 있는 데이터베이스를 연동하는 시스템을 구현했다. 본 논문에서 구현한 스마트팜은 기존의 단층 구조의 농사 형식에서 다층의 건물과 같은 형식을 적용해 공간의 효율성을 증진시키고 환경에 영향받지 않아 고정적이며 높은 품질의 생산물을 안정적으로 생산해 낼 것을 기대하였다. 본 논문에서 구현한 시스템은 기존의 사용자가 스마트팜을 효율적이고 편리하게 사용할 수 있게 만들어 줄 것이라고 예상된다.

향후 목표로는 본 논문의 시스템에서 사용자가 직접 작물의 광량과 목표 습도등을 직접 지정하여 사용하는 형태로 제작되었지만 비전처리를 이용해 작물의 현재 상태를 평가하고 최적의 결과값을 산출해서 자동으로 조절되는 시스템을 만들어 보는 것이다. 이를 위해서는 비전처리에 관한 지식과 작물의 이미지에 따른 작물 상태를 알아낼 수 있는 연구가 필요하다. 뿐만 아니라 작물 상태에 따라 최적값을 맞춰주기 위한 연구 데이터도 필요하다. 현재 측정되고 있는 값들을 이용해 추후 목표를 이루기 위한 데이터로 사용할 예정이다.

[본 논문은 과학기술정보통신부  
정보통신망의인재양성사업의 지원을 통해 수행한  
ICT멘토링 프로젝트 결과물입니다]

#### 참고문헌

- [1] 함규원 기자, “스마트팜 활성화로 농업 경쟁력 높여야... 정책적 뒷받침 필요”, 농민신문, 2020년9월14일, <https://www.nongmin.com/news/NEWS/POL/ETC/326761/view>
- [2] 한국갤럽, “2012-2019 스마트폰 사용률, 브랜드, 스마트워치, 손목시계에 대한 조사”, 한국갤럽, 2019년 8월 29일, <https://www.gallup.co.kr/gallupdb/reportContent.asp?seqNo=1041>
- [3] 과학기술정보통신부, “가구 인터넷 보급률 및 컴퓨터 보유율”, e-나라지표, 2020년 3월 26일, <https://www.index.go.kr/potal/main/EachDtlPageDetail.do?idxcd=1345>
- [4] Open JS Foundation, “Node.js에 대해서”, 공식문서, <https://nodejs.org/ko/about/>
- [5] wikipedia.org, “데이터베이스”, wikipedia, 2020년 9월 24일, <https://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%84%B0%EB%B2%A0%EC%9D%B4%EC%8A%A4>
- [6] tcpschool.com, “관계형 데이터베이스”, tcpschool, [http://www.tcpschool.com/mysql/mysql\\_intro\\_relationalDB](http://www.tcpschool.com/mysql/mysql_intro_relationalDB)
- [7] wikipedia.org, “API”, wikipedia, 2020년 9월 24일, <https://ko.wikipedia.org/wiki/API>
- [8] 이상우, “식물공장과 LED 인공광 이용한 식물재배”, Optical science and Technology, 121, 2020. 01