



국민대학교
전자정보통신대학
컴퓨터공학부


캡스톤 디자인 I

종합설계 프로젝트

| | |
|--------|-----------------------|
| 프로젝트 명 | 야 너도 키울수 있어 스마트팟 (야스) |
| 팀 명 | 귀농팀 |
| 문서 제목 | 계획서 |

| | |
|---------|-------------|
| Version | 1.4 |
| Date | 2019-APR-11 |

| | |
|----|-----|
| 팀원 | 원정희 |
| | 이우재 |
| | 조현우 |
| | 주연호 |
| | |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |


1.1.1 CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 “야 너도 키울수 있어”를 수행하는 팀 “귀농”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “귀농”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

| | |
|-----------------|------------------------------|
| Filename | 계획서-야 너도 키울수 있어 스마트팟(야스).doc |
| 원안작성자 | 원정희, 이우재, 조현우, 주연호 |
| 수정작업자 | 원정희, 이우재, 조현우, 주연호 |

| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
|------------|-------|----------|----------|---------------|
| 2019-03-11 | 전원 | 1.0 | 최초 작성 | |
| 2019-04-11 | 원정희 | 1.1 | 추가 작성 | 배경 기술 수정 |
| 2019-04-11 | 이우재 | 1.2 | 추가 작성 | 개발 목표 및 내용 수정 |
| 2019-04-11 | 조현우 | 1.3 | 추가 작성 | 개요 수정 |
| 2019-04-11 | 주연호 | 1.4 | 추가 작성 | 개발 목표 및 내용 수정 |
| | | | | |
| | | | | |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

목 차

| | | |
|-------|----------------------------------|----|
| 1 | 개요 | 4 |
| 1.1 | 프로젝트 개요 | 4 |
| 1.2 | 추진 배경 및 필요성 | 4 |
| 2 | 개발 목표 및 내용 | 6 |
| 2.1 | 목표 | 6 |
| 2.2 | 연구/개발 내용 | 6 |
| 2.3 | 개발 결과 | 9 |
| 2.3.1 | 결과물 목록 및 상세 사양 | 9 |
| 2.3.2 | 시스템 기능 및 구조 | 9 |
| 2.4 | 기대효과 및 활용방안 | 11 |
| 3 | 배경 기술 | 12 |
| 3.1 | 기술적 요구사항 | 12 |
| 3.2 | 현실적 제한 요소 및 그 해결 방안 | 13 |
| 3.2.1 | 데이터 오류! 책갈피가 정의되어 있지 않습니다. | |
| 4 | 프로젝트 팀 구성 및 역할 분담 | 12 |
| 5 | 프로젝트 비용 | 13 |
| 6 | 개발 일정 및 자원 관리 | 15 |
| 6.1 | 개발 일정 | 15 |
| 6.2 | 일정별 주요 산출물 | 16 |
| 6.3 | 인력자원 투입계획 | 17 |
| 6.4 | 비 인적자원 투입계획 | 18 |
| 7 | 참고 문헌 | 18 |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

1개요

1.2 프로젝트 개요

요즘 미세먼지나 살충제 등으로 깨끗한 채소, 과일 등을 구하는 것이 점점 어렵게 되어가고 있다. 이러한 문제를 조금이나마 극복하고자 바쁜 현대인들도 집에서 신선하고 안전한 작물을 재배할 수 있는 방법으로 스마트팟을 선정하였다.

스마트팟이란 ICT기술을 비닐하우스, 축사, 과수원 등에 접목해 원격, 자동으로 작물의 생육환경을 적절히 제어할 수 있는 농장이다. 우리는 스마트팟에 있는 기술을 접목하여 집에서도 키울 수 있는 작은 농장을 운영하려고 한다.

이 프로젝트는 작물을 키울 수 있는 스마트팟을 축소한 형태로 기존의 대규모의 스마트팟 보다 축소하여 비용적인 측면을 줄이고 접근성을 향상시킬 수 있다.

작물이 자랄 수 있는 최적의 환경을 구축 할 수 있게 센서들을 사용하여 스마트팟을 구성할 예정이다. 또한 집에 있지 않아도 원격으로 조종하여 관리를 할 수 있다.

1.3 추진 배경 및 필요성

1.3.1 추진 배경 및 필요성

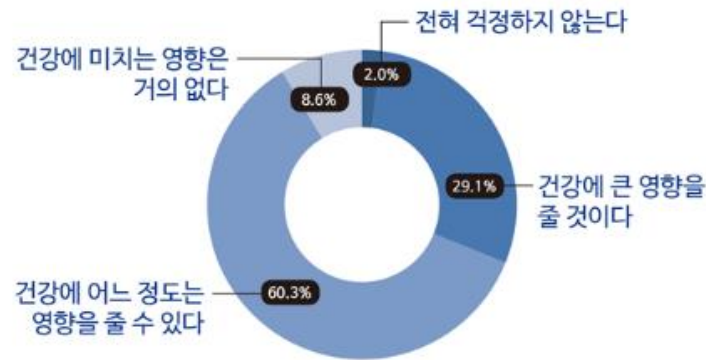
실제로 스마트팟을 하기 위해서 많은 데이터가 필요한데 식물 특성상 자라는 기간이 있기 때문에 데이터를 모으기 쉽지 않다. 그렇기 때문에 가정에 작은 소규모로 채소등 식물을 각각의 케이스 별로 키워 데이터를 모으는데 가장 큰 궁극적 이유이다.

요즘 미세먼지나 살충제 등으로 깨끗한 채소, 과일 등을 구하는 것이 점점 어렵게 되어가고 있다. 최근 전국적으로 고농도 미세먼지가 기승을 부리는 가운데 미세먼지가 식품 안전에 미치는 영향이 매우 크다는 결과가 나왔다. 또한 소비자들에게 노지에서 생산되는 채소가 미세먼지 노출되어 건강에 영향을 미칠 수 있다고 설문조사를 한 결과 다음 그림과 같이 나왔다.

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

노지에서 생산되는 채소가 미세먼지에 노출되어 건강에 영향을 미칠 수 있다고 생각합니까?

농식품소비자 701명 응답 / 2018.9.21~10.6(총 16일간)




'미세먼지가 식품안전에 미치는 영향에 대한 인식' 온라인 공개 설문조사
미세먼지해결시민본부 & 경상대학교 화훼원예학연구소

[그림 1]

경상대학교 원예생산공학실험실은 지난해 9월 3주 동안 공단과 고속도로 옆, 농촌에서 상추와 샷갓, 시금치 등 3가지 채소를 재배하여 중금속 함량을 조사하였다. 조사결과 고속도로 인근 지역에서 재배한 샷갓과 시금치에서 식품안전기준을 초과한 납이 검출 되었다. 위의 결과로 미세먼지가 채소의 중금속 함량에 영향을 주는 것으로 나타났다.

미세먼지와 살충제 등으로부터 우리는 좀 더 신선하고 안전한 채소를 수확하기 위해 스마트팟의 기술을 접목하기로 하였다. 스마트팟이란 ICT기술을 비닐하우스, 축사, 과수원 등에 접목해 원격, 자동으로 작물의 생육환경을 적절히 제어할 수 있는 농장이다. 우리는 이 기술을 이용하여 소규모 채소 및 과일 키울 수 있다.

개인 가정에서 채소 및 과일을 재배하면 신선하고 안전한 채소를 바로 먹을 수 있고, 또한 내가 키우는 채소들이 자라나는 과정을 지켜 보면서 쓸쓸한 행복감과 보람을 느낄 수 있을 것이다. 또한 조그마한 텃밭처럼 꾸며진 곳은 자연적인 공간이기 때문에 아이들의 교육 면에서도 좋으며 휴과 생명의 소중함은 물론이고 수확의 기쁨을 누릴 수 있을 것이다.

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

1.3.2 현재 스마트팟의 문제점

다만 기술이 세대를 지나면서 매우 빠르게 발전해 왔기 때문에 고연령층과 청년층의 기술 이해도와 응용에 있어 다소 차이를 낼 수 있다. 또한 비용측면에서도 많은 부담을 가지고 있다. 물론 규모에 따라 비용이 다르긴 하지만 소규모 농가 기준으로 보면 최소 1억에서 2억정도 든다고 한다. 이런 점들 때문에 스마트팟 적용 증가율은 다소 미비할 수 있다. 이런 세대에서 나오는 차이를 보완하고 비용을 최소화 할 수 있는 요소들이 있는지를 연구하는 것이 중요한 과제이다.

2 개발 목표 및 내용

2.1 목표

- 1) 스마트팟을 제작하여 서버를 통해 멀리 있어도 스마트폰으로 현재 상황을 체크하고 컨트롤 할 수 있게 컨트롤 하여 식물이 자랄수 있는게 최적의 상태를 만들어 수확물도 늘리고 미세 먼지와 살충제 등으로부터 보다 안전하고 신선한 작물을 직접 재배하여 섭취가 가능하게 하는 것이 목표이다.
- 2) 누군가가 직접 심고 키운 것을 사용자가 먹는 것이 아니라 자신이 직접 처음부터 키우기 때문에 자라나는 과정을 지켜볼 수 있어 행복감과 보람을 느낄 수 있다.
- 3) 요즘 아이들은 대부분 스마트폰과 컴퓨터 등을 사용하면서 여가시간을 보내고 있다. 조그마한 텃밭처럼 꾸며진 이 스마트팟을 키우게 된다면, 조금이나마 이 자연적인 공간을 직접 눈으로 관찰 할 수 있어 교육적인 측면에서도 좋을 것이며 휴과 생명의 소중함과 수확의 기쁨을 누릴 수 있을 것이다.
- 4) 실제로 스마트팟을 하기 위해서 많은 데이터가 필요한데 식물 특성상 자라는 기간이 있기 때문에 데이터를 모으기 쉽지 않다. 그렇기 때문에 가정에 작은 소규모로 채소 등 식물을 각각의 케이스별로 키워 데이터를 모으는데 가장 큰 궁극적 목표이다.

2.2 연구/개발 내용

2.2.1 작동 순서

| | | | |
|--|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

1. 아두이노를 통해 얻은 데이터를 라즈베리파이로 보낸후 aws iot로 보낸다.

2. aws iot로 받은 데이터를 lambda 서비스를 이용해 어플리케이션으로 짜놓은 코드를 보낸다. 이때 동시에 iot기능중 하나로 aws dynamo DB에 데이터를 보내 데이터를 관리할 수있다.

3. 어플리케이션으로 사용자는 원하는 조작을 하여 다시 역순으로 데이터 및 신호를 보내서 관리한다.


2.2.2 작동 순서 설명

1. 아두이노와 여러가지 센서(온.습도,모터,water pump,Fan,LED, soil moisture,카메라 등등) 연결한뒤 각각의 센서들로부터 데이터를 받는다. 받은 데이터를 라즈베리 파이로 전송한 후 라즈베리 파이에서 다시 aws iot로 전송한다. 이렇게 중간에 라즈베리 파이를 쓰는 이유는 아두이노와 aws iot와 바로 연결하기엔 여러가지 문제점들이 있다.

우선 aws iot는 https 로 지원을 하는데 아두이노는 http만 지원을 한다. 이뿐만이 아니라 인증서관련 문제점이 있다. 따라서 아두이노와 aws iot를 바로 연결하기위해서는 중간에 api gateway와 lambda 기능을 이용해서 인증서와 https 문제점을 해결해야 하는데 이러한 번거러움과 복잡함을 없애기위해서 중간에 변환기 역할로 라즈베리 파이를 거치게 한다. 꼭 이때뿐만이 아니라 라즈베리 파이와 aws는 호환이 잘되어 있어 aws의 모든 기능들을 사용하기 편하다.

Aws iot에 사물등록을 한후 사물에 필요한 정책(aws iam 서비스)을 정한다. 그 다음 private key, public key, 인증서 이 3가지를 자동생성한후 다운로드 받고 인증서를 라즈베리 파이에 저장한다. 그다음 python 파일을 이용해 인증서와 키를 등록하고 aws iot에 보낼 데이터들을 정하는 코드를 작성한다. 이제 Python 파일을 실행하여 라즈베리 파이와 aws iot와 연동시킨다.

2. 연결된 aws iot를 통해 라즈베리파이에서 전송해온 데이터를 각 분야별로 이용할 수 있다. 우선 aws lambda기능이 있는데 lambda 기능이란, 보통의 물리적 서버 또는 가상 서버 기반의 일반적인 접근방법과 다르다. 여러 함수로 구성된 논리적인 구성과 이들 함수를 실행하는 서비스만 신경 쓸 수 있게끔 만들어놓은 서비스이다. 각 함수는 컨테이너

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

에서 실행한다. 각 컨테이너는 운영체제의 커널이 격리된 환경을 구현하는 서버 가상화 방식이기에 이러한 인프라를 관리할 필요가 없어진다. 이러한 접근 방식을 서버리스(serverless)라 부른다. 이 lambda 서비스를 이용해서 어플리케이션에 함수를 전송하여 사용자가 어플리케이션을 통해 aws iot를 관리 할 수 있다.

또한, lambda 서비스에서 따로 추가하여 aws에서 제공하는 Dynamo DB를 동시에 관리할 수 있다. Dynamo DB는 noSQL 기반의 aws(amazon web service)에서 제공하는 DB이다. 여기서 aws에 강력한 기능이 나오는데 한곳에서 데이터를 가지고 와서 다른 aws기능에서 쉽게 가져와 이용하고 관리를 할 수 있다. 이렇게 lambda 서비스로 사용자가 사용할 어플리케이션에 정보를 제공한다.

3. 이렇게 받은 데이터를 AWS서버를 통해 DB로 저장을 한 뒤 그 데이터를 모바일 어플리케이션을 통해 사용자에게 필요한 정보로 다듬어 실시간으로 사용자에게 볼 수 있게 한다. 이 것은 네트워크를 통해 정보를 전달하기 때문에 장소에 구애 받지 않고 정보를 이용할 수 있다.

모바일 어플리케이션에 보여질 정보들은 센서를 통해 받은 실시간 온도,습도 등의 현재 환경정보와 또, 물을 준 시간, 일정시간 동안의 온도, 습도의 평균, 급변화량과 같은 환경에 필요한 정보들 그리고 카메라를 통한 실시간 영상자료등이다.

이 정보들을 네트워크 서버를 통해 받아서 모바일 어플리케이션에 보여질 수 있도록 하는 부분과 그것들을 원격으로 제어 할 수 있는 것과 사용자가 편하게 이용할 수 있게 하는 UI를 설계하고 개발하는 것이 앱 개발부분의 목표이다.

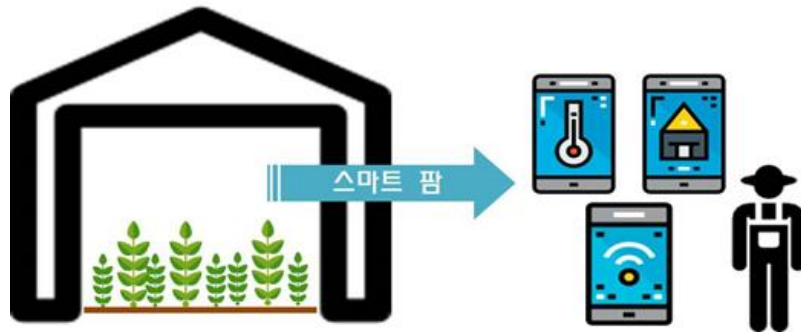
모바일 어플리케이션의 개발 환경은 Android Studio이고 개발언어는 Java다.

만들어진 어플리케이션은 사용자에게 로그인 정보를 물은 뒤 로그인이 완료되면, 자신의 키우고 있는 팜들을 선택하게 하고 위에서 언급한 농장의 정보들을 보여준다. 또, 자동으로 환경들이 조절이 되는 팜이지만, 특수상황을 대비해 원격으로 제어 할 수 있는 버튼들을 제시해 자신이 직접 통제할 수 있게 한다.

| | | | |
|--|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

2.3 개발 결과

2.3.1 시스템 기능 요구사항




1. 아두이노가 스마트팜 주변 환경에 대한 데이터들을 습득하고 라즈베리파이로 전송한다..
2. 라즈베리 파이에서 AWS IoT 서비스로 데이터를 전송한후 lambda 서비스를 이용하여 사용자가 사용하는 어플리케이션으로 함수를 보낸다.
3. 함수를 받은 어플리케이션은 이를 실행하고 사용자에게 UI를 제공한다.
4. 사용자는 제공받은 UI를 통해 원하는 작업을 한후 다시 어플리케이션을 통해 실행 함수를 lambda 서비스로 보낸다.
5. 이를 받은 lambda 서비스는 AWS IoT 서비스를 이용해 다시 라즈베리 파이로 보내 라즈베리는 아두이노로 보내 최종적으로 스마트팜에 설치되어 있는 각종 sensor들을 작동시킨다.

2.3.2 시스템 비기능(품질) 요구사항

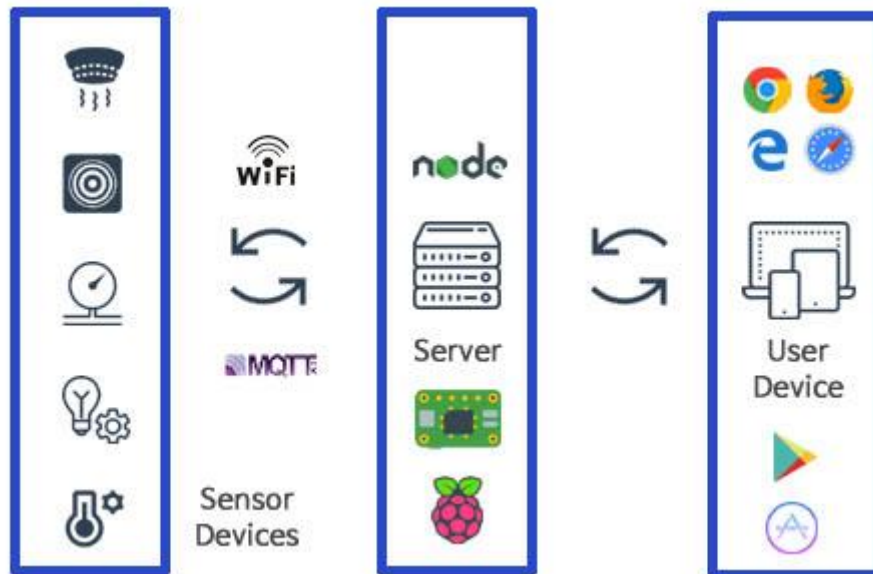
AWS에서 지원하는 서버를 이용하기 때문에 고성능의 컴퓨터가 필요하진 않지만 AWS를 이용하기 위한 별도의 요금이 필요하다.

AWS 서버로 데이터를 보낼때 사용하는 하드웨어 장비들이 AWS와 연동이 가능한지 확인이 필요하며, 어떠한 설정이 필요한지 확인해야한다.

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

이 프로젝트는 하드웨어적인 부분 보다는 기능적인 면이 더 중요시 여겨진다. AWS에서 어떠한 정보를 받고 어떠한 정보를 저장할것이며, 어떠한 정보를 제공할것인지 확인 해야한다.

2.3.3 시스템 구조




각종 sensor(온.습도,모터,water pump,Fan,LED, soil moisture,카메라)에서 스마트팟 주변환경 데이터를 받아드려와 아두이노로 전달, 아두이노로 전달된 데이터를 라즈베리를 거쳐 AWS IoT로 전송된다. AWS IoT에서 lambda와 Dynamo DB에 데이터를 전송한다. Dynamo DB같은 경우 IoT에 전달된 온.습도 정보와 soil moisture 정보를 각 쿼리에 맞춰 저장한다.

그다음은 lambda service를 이용해서 사용자 어플리케이션으로 정보를 전달, 사용자는 전달된 내용을 어플리케이션을 통해 눈으로 모니터링 하고 각종 sensor들을 control 한다.

2.3.4 결과물 목록 및 상세 사양

| 대분류 | 소분류 | 기능 | 형식 | 비고 |
|------|-------|--------------------|----|----|
| 어플 | 어플 재생 | 어플에서 기능을 선택한다. | | |
| 하드웨어 | 센서 | 어플에서 선택한 기능을 실행한다. | | |
| DB | 데이터 | 데이터를 저장한다. | | |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

2.4 기대효과 및 활용방안

가정에서 손쉽게 스마트팟을 이용해 원하는 작물을 맞춤형으로 주변 환경을 구성할수 있다는 장점이 있어 가정에서 사용자가 원하는 깨끗한 채소 및 과일을 얻을수 있다. 또한 바쁜 직장인들은 가정에서 식물들을 관리하기가 편리해진다. 또한 이러한 것을 확장하여 스마트팟 렌탈 시스템을 도입도 생각 할 수 있다. 각 가정에 정수기나 비데 같은 렌탈 서비스를 차용하여 스마트팟도 각 가정에 렌탈 해주고 스마트팟에 필요한 서버 및 시스템을 관리하고 데이터를 수집하여 좀더 빠르게 빅데이터를 수집할 수 있을거라 기대한다.

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

3 배경 기술

3.1 기술적 요구사항

-운영체제 : 라즈비안 os, ubuntu 16.4.4 , window10

-데이터 베이스 : DynamoDB

-서버 : AWS

-언어 : Node.js , Java 기반 Android Studio 언어

-개발환경 : intelliJ, Andorid Studio, aws , Node.js

3.1.1 활용 정보

*Aws 서버에서의 활용 내용

- Aws lam
- Aws lamba
- Aws DynamoDB
- Aws lot

*하드웨어 개발 활용 내용

nodejs : Event-Driven Programming, Asynchronous I/O Programming

· Johnny-five : Node.js-Based Open Source IoT/Robotics Programming Framework, Event-Driven, 비동기식 프로그래밍을 지원한다. 다양한 Arduino 호환 보드에서 테스트 되어, 동일한 API를 제공하여 하나의 플랫폼에서 작성된 코드를 다른 플랫폼에서 수정 없이 사용할 수 있다. 지원하지 않는 API에 대하여 사용자가 Javascript 라이브러리를 제작, 확장 및 실험할 수 있다.

3.1.2 프로젝트 테스트 환경

1. 안드로이드 스튜디오로 제작한 어플리케이션으로 동작 작동
2. 라즈베리파이와 아두이노에 연결된 센서 정상작동 확인
3. 센서 작동에 따른 DB를 DynameDB에 저장한 데이터를 어플리케이션에서 확인

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

3.2 현실적 제한 요소 및 그 해결 방안

3.2.1 데이터

현실적 제한 요소 : 농산물을 최적의 상태로 키우기 위해 데이터가 있어야 되는데 현재 축적된 데이터와 가지고 있는 지식이 없다.

해결 방안 : 기상자료개방포털 사이트에서 데이터를 가져온다.

3.2.2 기타

현실적 제한 요소 : aws 서버를 사용한 적이 없어 공부하는데 많은 시간과 착오가 많다.


해결 방안 : KMU Crowd Sensor Cloud 모임에서 강의를 듣는다.

4 프로젝트 팀 구성 및 역할 분담

| 이름 | 역할 |
|-----|---|
| 원정희 | - Software Project Leader - 하드웨어 소스 개발 |
| 이우재 | - Server 개발 및 관리 |
| 조현우 | - DB 설계 및 DB Query 시스템 개발 |
| 주연호 | - Android 어플 제작 |

5 프로젝트 비용.

| 항목 | 예상치 (MD) |
|----|----------|
|----|----------|

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

| | |
|---------------|----|
| 소형 비닐하우스 제작 | 3 |
| 센서테스트 및 연동 관리 | 20 |
| 서버 연결 | 3 |
| 데이터 베이스 모델링 | 3 |
| 서버 관리 | 5 |
| 계획서 제작 및 보고 | 20 |
| | |
| 합 | 54 |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

6 개발 일정 및 자원 관리

6.1 개발 일정

| 항목 | 세부내용 | 1 월 | 2 월 | 3 월 | 4 월 | 5 월 | 비고 |
|--------|-------------|-----|-----|-----|-----|-----|----|
| 요구사항분석 | 요구 분석 | | | | | | |
| | SRS 작성 | | | | | | |
| 관련분야연구 | 주요 기술 연구 | | | | | | |
| | 관련 시스템 분석 | | | | | | |
| 설계 | 시스템 설계 | | | | | | |
| 구현 | 코딩 및 모듈 테스트 | | | | | | |
| 테스트 | 시스템 테스트 | | | | | | |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

6.2 일정별 주요 산출물

| 마일스톤 | 개요 | 시작일 | 종료일 |
|-----------|--|------------|------------|
| 계획서 발표 | 개발 환경 완성 (GCC 설치, 기본 응용 작성 및 테스트 완료) 산출물 : 1. 프로젝트 수행 계획서 2. 프로젝트 기능 일람표 | 2019-03-04 | 2019-03-14 |
| 설계 완료 | 시스템 설계 완료 산출물 : 1. 시스템 설계 사양서 | 2019-03-14 | 2019-03-20 |
| 1 차 중간 보고 | 각 센서 구현 산출물 : 1. 프로젝트 1 차 중간 보고서 2. 프로젝트 진도 점검표 3. 1 차분 구현 소스 코드 | 2019-03-21 | 2019-04-15 |
| 2 차 중간 보고 | 어플리케이션과 서버 연동 후 센서 구현 산출물 : 1. 프로젝트 2 차 중간 보고서 | 2019-04-29 | 2019-05-17 |
| 구현 완료 | 시스템 구현 완료 산출물: 스마트팟 모형 제작 | 2019-05-18 | 2019-05-20 |
| 테스트 | 시스템 통합 테스트 산출물: 스마트팟 전체 테스트 | 2019-05-20 | 2019-05-24 |
| 최종 보고서 | 최종 보고 산출물: 1.결과물 제출 2.최종 계획서 제출 | 2019-05-24 | 2019-05-31 |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

6.3 인력자원 투입계획

| 이름 | 개발항목 | 시작일 | 종료일 | 총개발일(MD) |
|-----|--------------------|----------|-----------|----------|
| 원정희 | 라즈베리파이, 아두이노 센서 코드 | 2019-3-4 | 2019-5-31 | |
| 이우재 | 서버관리 | 2019-3-4 | 2019-5-31 | |
| 주연호 | 안드로이드 어플 제작 | 2019-3-4 | 2019-5-31 | |
| 조현우 | DB 관리 | 2019-3-4 | 2019-5-31 | |
| | | | | |
| | | | | |
| | | | | |

| | | | |
|---|-------------------------|-------------|-------------|
|  국민대학교 컴퓨터공학부 캡스톤 디자인 I | 계획서 | | |
| | 프로젝트 명 | | |
| | 팀 명 | | |
| | Confidential Restricted | Version 1.2 | 20xx-MAR-25 |

6.4 비 인적자원 투입계획

| 항목 | Provider | 시작일 | 종료일 | Required Options |
|---------------|--------------|-----|-----|------------------|
| 컴파일러 | Microsoft | | | |
| 개발용 PC 4 대 | Dell | | | |
| 임베디드 보드 | Raspberry pi | | | |
| 임베디드 보드 | adruino | | | |
| 온도센서 | | | | |
| 습도센서 | | | | |
| 토양수분센서 | | | | |
| 모터센서 | | | | |
| FAN 센서 | | | | |
| 카메라 센서 | | | | |
| Water Pumb 센서 | | | | |

7 참고 문헌

| 번호 | 종류 | 제목 | 출처 | 발행년도 | 저자 | 기타 |
|----|----|----|----|------|----|----|
| | 서적 | | | | | |
| | 기사 | | | | | |