

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	야 너도 키울수 있어 스마트팟 (야스)
팀 명	귀농
문서 제목	중간보고서

Version	1.3
Date	2019-04-30

팀원	원정희 (조장)
	이우재
	조현우
	주연호
지도교수	김인희 교수

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명		
	팀 명		
	Confidential Restricted	Version 1.2	20xx-APR-30

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “야 너도 키울 수 있어 스마트팟(야스)”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	중간보고서-야 너도 키울 수 있어 스마트팟(야스).doc
원안작성자	원정희, 조현우, 이우재, 주연호
수정작업자	원정희, 조현우, 이우재, 주연호

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-04-17	원정희	1.0	최초 작성	목표 및 하드웨어 부분 작성
2019-04-17	조현우	1.1	내용 추가	수정된 연구내용 추가
2019-04-17	이우재	1.2	내용 추가	AWS 부분 작성
2019-04-17	주연호	1.3	내용 추가	안드로이드 부분작성

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명		
	팀 명		
	Confidential Restricted	Version 1.2	20xx-APR-30


목 차

1	프로젝트 목표	4
2	수행 내용 및 중간결과	6
2.1	계획서 상의 연구내용	6
2.2	수행내용	9
3	수정된 연구내용 및 추진 방향	14
3.1	수정사항	14
4	향후 추진계획	15
4.1	향후 계획의 세부 내용	15
5	고충 및 건의사항	17

1 프로젝트 목표

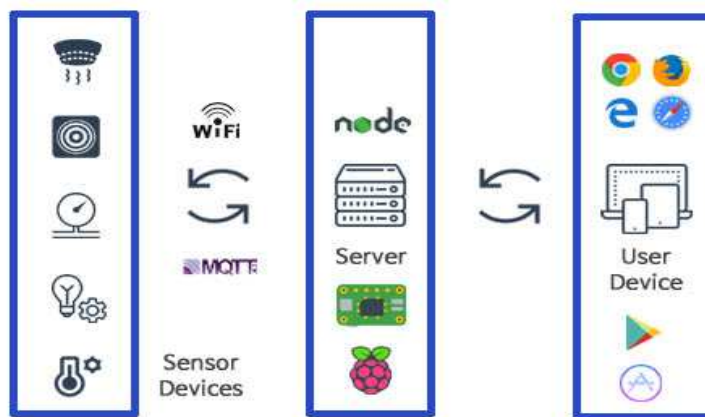


- 본 프로젝트는 스마트팜을 제작하여 서버를 통해 멀리 있어도 스마트폰으로 현재 상황을 체크하고 컨트롤 할 수 있게 컨트롤 하여 식물이 자랄 수 있는게 최적의 상태를 만들어 수확물도 늘린다.
- 현재 문제점이 스마트팜의 데이터 문제를 해결하기 위한 농가뿐만 아니라 가정집에서도 온도와 습도 등 식물이 자라는 환경을 조절할 수 있게 만들어 카메라로 촬영 하여 여러 데이터를 모으는 목표에 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명		
	팀 명		
	Confidential Restricted	Version 1.2	20xx-APR-30

2 수행 내용 및 중간결과

2.1 계획서 상의 연구내용



2.1.1 작동 순서

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

2.1.2 작동 순서 설명

1. 아두이노와 여러가지 센서(온.습도,모터,water pump,Fan,LED, soil moisture,카메라 등 등) 연결한뒤 각각의 센서들로부터 데이터를 받는다. 받은 데이터를 라즈베리 파이로 전송한후 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 아니




라 인증서관련 문제점이 있다. 따라서 아두이노와 aws iot를 바로 연결하기 위해서는 중간에 api gateway와 lambda 기능을 이용해서 인증서와 https 문제점을 해결해야 하는데 이러한 번거로움과 복잡함을 없애기 위해서 중간에 변환기 역할로 라즈베리 파이를 거치게 한다. 꼭 이때문만이 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 컨테이너에서 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.

	중간보고서		
	국민대학교	프로젝트 명	
	컴퓨터공학부	팀 명	
	캡스톤 디자인 I	Confidential Restricted	Version 1.2 20xx-APR-30

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.

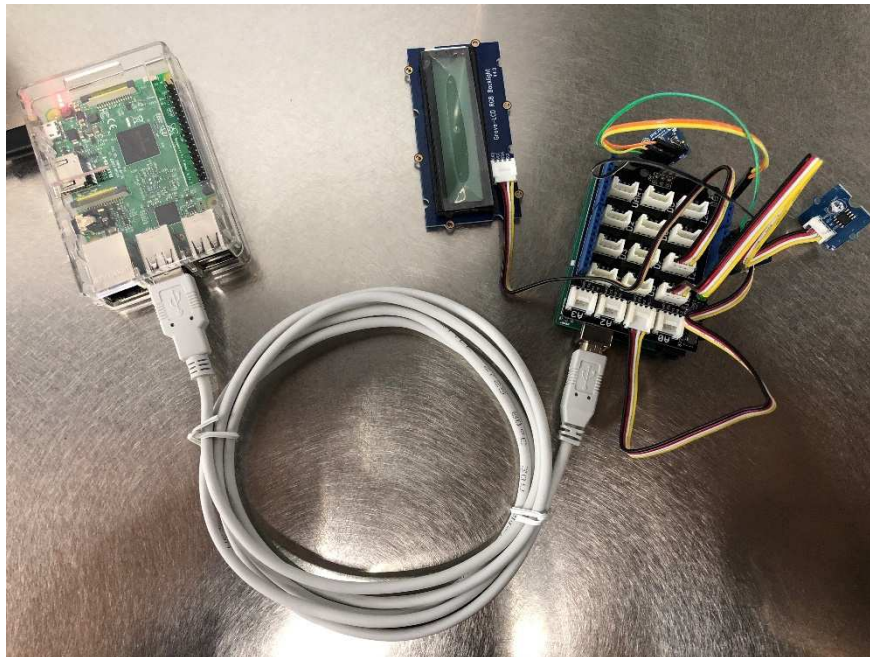
이 정보들을 네트워크 서버를 통해 받아서 모바일 어플리케이션에 보여질 수 있도록 하는 부분과 그것들을 원격으로 제어 할 수 있는 것과 사용자가 편하게 이용할 수 있게 하는 UI를 설계하고 개발하는 것이 앱 개발부분의 목표이다.

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.



2.2 수행내용

Hardware



-하드웨어로는 raspberry pi, arduino ,각종 센서로 이루어져 있으며 raspberry pi에 arduino를 연결하고 arduino에 센서들을 연결하여 기능들을 구현하고 있다.

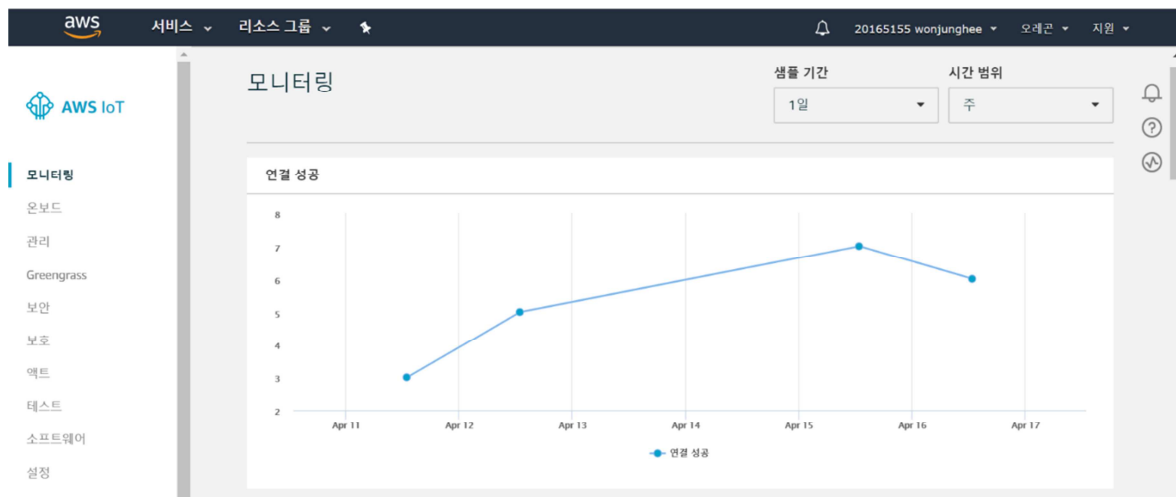
- Node.js 언어를 이용하여 raspberry pi에서 코딩하여 기능 구현하였다.

- Event-Driven, 비동기식 프로그래밍을 지원하는 Johnny-five를 이용하여 현재 raspberry pi에 온도와 습도를 Event와 상관없이 데이터를 가져오게 구현하였다.



```
pi@raspberrypi: ~/my_farm
File Edit Tabs Help
pi@raspberrypi:~ $ cd my_farm/
pi@raspberrypi:~/my_farm $ node my_farm.js
1555505686414 Available /dev/ttyACM0
1555505686527 Connected /dev/ttyACM0
aws connect
1555505690402 Repl Initialized
>> celsius : 21.4
fahrenheit : 70.52
kelvin : 294.55
-----
Barometer
pressure : 99.834
-----
Hygrometer
humidity : 34.606
-----
Altimeter
feet : 0
meters : 0
-----
celsius : 21.4
fahrenheit : 70.52
kelvin : 294.55
-----
```

- 현재 aws iot에서 구축한 서버에 접속했다는 의미로 console.log를 이용하여 aws connect를 보여준다.
- 센서를 이용해 섭씨, 화씨, 절대 온도, 습도, 기압, 고도를 순간순간 데이터를 가져오는 것을 확인할 수 있다. 하지만 우리는 섭씨만을 사용할 것이다.



- 연결에 성공 하면 aws iot 서비스의 모니터링에 가면 보이는 것 처럼 그래프를 통해 확인할 수 있다.

AWS 역할

AWS IoT : 현재 AWS IoT에 사물 등록 완료하였고 인증서 생성 및 정책 설정 완료 하였다.

AWS IAM 정책(AWS에서 제공하는 서비스들의 접근 권한 설정)을 사물에 등록성공하였고 장비(라즈베리파이)에서 required 하는부분(물리 사물이 보고한 마지막 형태)과 앱(app)부분에서 desired 하는 부분(물리 사물의 상태를 변화시키려고 하는 방향)을 IoT Shadow 라는 service가 담당하는데 이 기능은 사물을 고려했을 때 24시간 항시대기상태일수가 없기 때문에(배터리 문제등)AWS IoT부분에서 사용자가 명령한 부분들을 담고 있다가 사물이 sleep상태에서 ON이 되면 자동으로 명령한 데이터를 보내는 기능을 한다. 즉, 일명 보드(board)역할을 한다고 볼수있다.각 사물이 할일들을 보드(board)에 붙이면 그 역할을 수행할 사물이 각자 자기 역할에 맞게 할일을 떼가는 형식이라고 보면 된다. 이러한 기능을 테스트 할 수 있는 기능이 있다. 이 test에서 데이터를 required하고 desired하는 부분까지 성공 하였다. 라즈베리파이 쪽에서는 라즈베리파이 라이브러리에 Javascript용 AWS IoT디바이스 SDK를 설치완료했다.

주제 게시

\$aws/things/AirSensor2016... ✕

\$aws/things/AirSensor20165... ✕

\$aws/things/AirSensor20165... ✕

게시
QoS of 0으로 게시할 주제와 메시지를 지정합니다.

\$aws/things/AirSensor20165157/shadow/update

주제 게시

```

3  "desired": {
4    "welcome": null
5  },
6  "reported": {
7    "welcome": null,
8    "moisture": "low"
9  }
10 }
11

```

\$aws/things/AirSensor20165157/shadow/update 2019. 4. 17. 오후 8:09:59

내보내기 숨기기

```

{
  "state": {
    "desired": {
      "welcome": null
    },
    "reported": {
      "welcome": null,
      "moisture": "low"
    }
  }
}

```



\$aws/things/AirSensor20165... ✕

\$aws/things/AirSensor2016... ✕

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

\$aws/things/AirSensor20165157/shadow/get/a... 2019. 4. 17. 오후 8:10:55

내보내기 숨기기

```
{  
  "state": {  
    "reported": {  
      "temperature": 18,  
      "humidity": 35,  
      "pm025": 10,  
      "pm100": 30,  
      "moisture": "low"  
    }  
  },  
  "metadata": {  
    "reported": {  
      "temperature": {  
        "timestamp": 1555413642
```

주제 목록

주제 게시

\$aws/things/AirSensor20165... ✕

\$aws/things/AirSensor20165... ✕

\$aws/things/AirSensor2016... ✕

게시

QoS of 0으로 게시할 주제와 메시지를 지정합니다.

\$aws/things/AirSensor20165157/shadow/get/accepted

주제 게시

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

\$aws/things/AirSensor20165157/shadow/get/a... 2019. 4. 17. 오후 8:39:30

내보내기 숨기기


```
{  
  "state": {  
    "reported": {  
      "temperature": 18,  
      "humidity": 35,  
      "pm025": 10,  
      "pm100": 30,  
      "moisture": "high"  
    }  
  },  
  "metadata": {  
    "reported": {  
      "temperature": {  
        "timestamp": 1555413642
```

	중간보고서		
	프로젝트 명		
	팀 명		
	Confidential Restricted	Version 1.2	20xx-APR-30

Android app (어플)



현재까지 APP UI 디자인을 완성했고 안드로이드 스튜디오에서 바로 깃허브로 변경된 코드를 커밋할 수 있게 하였다. 이렇게 되는 과정에서, 초기에는 안드로이드 스튜디오에서 만든 뒤 소스코드를 바로 깃허브에다가 올리면 되겠다고 생각을 했는데 git에 대한 사용숙지 미달과 스튜디오에서 체크아웃하는 과정에서 계속된 오류가 나와 시간을 많이 소요하였다. 당시 스튜디오 내에서 체크아웃을 하게 되면 프로젝트 디렉토리내에서 capstone 21폴더가 만들어지게 되고 리모트에 있는 소스코드를 다 가지고 오게 되는데 거기 있는 소스코드와 여러 번 부딪히고 있었다. 그래서 두가지 방향으로 잡게 되었다. UI DESIGN을 따로 만든 뒤 깃허브에다 올리는 것이 충돌이 덜할 수 있겠다 생각을 했고, 아니면 깃허브에서 문제는 없고 원래 안드로이드 스튜디오를 하게 되면 여러 이슈들이 잡히는가에 대한 가능성도 있겠다 생각했다. 현재도 그 오류를 잡아내기 위해 노력하고 있고 명확하게 잡지는 못하였다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서	
		프로젝트 명	
		팀 명	
		Confidential Restricted	Version 1.2 20xx-APR-30

3 수정된 연구내용 및 추진 방향


3.1 수정사항

초기 계획서에서 작성한 목표가 구체적으로 잘 드러나있지 않아 목표점을 분명히 하기 위해 목표를 기존에 스마트팟 구현에서, 가정에서도 이용할 수 있는 작은 스마트팟(pot)으로 변경하였다. 가정에서 식물을 키우는 집이 많기 때문에, 식물을 키운다는 것에 초점을 두고 개인가정에 보급할 수 있다면 수요도 있을 것으로 판단된다. 앞으로 어린이들에게 식물을 지켜보는 것과 동시에 ICT에 대한 교육도 할 수 있어 추가적인 확장을 할 수 있게 하는 요소들도 많이 가지고 있다.

구체적인 목표 변경과 여러 시행착오를 통해 수정되어야 할 부분들이 보이게 되었다. 그것들은 첫째로, 물리적으로 스마트팟을 어떻게 구현할것인가에 관한 부분과 데모들을 실패했을 때 감수해야될 비용과 다시 회수해야될 자원들에 대해 구체적인 계획이 없었으며 앞으로 설정해야된다는 것이다. 두번째로는, 서버연동하는 부분과 아두이노와 라즈베리 파이를 연결하는 부분이 많은 시행착오를 겪고 있고 우리팀이 예상했던 것과는 다르게 많은 시간을 소요하고 있다. 그렇기에 연동하는 모든 부분들에 대해 항상 시간이 많이 소요될 것이라 여기고 그것에 관련한 시간을 더 길게 배치할것이다.

안드로이드 개발하는 과정에서 계속해서 시간을 잡는 이슈들과 직면했기 때문에 생각대로 되지 않는다는 것을 항상 염두에 두고 시간을 길게 잡고 구체적인 계획을 짤 것이다.

계획서상 개발내용에 명시적으로 기술들이 들어가있지 않고, 방향이 잘 정해져 있지 않아 다소 모호했던 것들을 변경된 수행계획서에 좀더 구체적으로 기술했다. 기술한 내용에는 우리가 현재 처해있는 문제들과 무엇을 개발할지가 적혀있다. 중복될 수도 있지만 전 순서와 되도록 내용의 일치와 일관성을 유지하기 위해 내용들을 좀 더 길게 풀고 전 순서에 있던 내용들을 다시 언급하는 부분이 많이 늘었다.

	국민대학교		중간보고서	
	컴퓨터공학부		프로젝트 명	
	캡스톤 디자인 I		팀 명	
			Confidential Restricted	Version 1.2 20xx-APR-30

4 향후 추진계획

4.1 향후 계획의 세부 내용

하드웨어 기능

아직 구입하지 못한 센서를 구매하여 그에 맞는 기능들을 추가 하고 아직 aws iot 와 연동은 하였지만 데이터를 주고 event를 받는 기능은 구현하지 않아 그 부분을 구현 하여 어플리케이션에서 데이터를 확인하고 조작을 할 수 있게 만들 예정이다.

각 센서들을 특정 온도, 습도에 맞게 커튼 열기 , 물주기 등 식물이 자랄 수 있는 최적의 환경을 만들 수 있게 데이터를 확인하여 제작 할 예정이다.

또한 성장 과정에 따라 물 주기, 광합성 등 환경에 변화를 주는 코드를 제작할 예정이다.

계획에는 없었지만 여유가 된다면 카메라 기능으로 식물을 체크 할 수 있는 기능과 미세먼지 센서를 추가하여 그날의 미세먼지 지수도 체크할 예정이다.

AWS 기능

AWS IoT Shadow service를 라즈베리파이와 앱(App)간의 required와 desired를 우선적으로 해결할 생각이다.그래야 데이터를 주고받고 할 수 있고 AWS IoT에 데이터가 들어와 나머지 service를 사용하는데 필요하다. 그 이후 AWS lambda service를 이용해 AWS에서 제공하는 Dynamo DB에 데이터를 보내는것과 여건만 된다면 SNS 보내는 service도 생각중이다.

Android app 기능

안드로이드 스튜디오를 통해 만들어진 어플리케이션을 aws s아를 이용하여 aws iot 서버와 연동시킬 것이다.

	중간보고서		
	국민대학교 컴퓨터공학부 캡스톤 디자인 I	프로젝트 명	
		팀 명	
		Confidential Restricted	Version 1.2 20xx-APR-30

서버와 연동 후 데이터를 어플로 가져오는 것이 여기서는 스튜디오내에 있는 gradle을 이용해 dynamoDB와 연결할 것이다.

어플리케이션 화면에는 dynamoDB에서 데이터를 가져와 현재 온도, 습도, 토양 습도 등 데이터를 눈으로 볼 수 있게 제작하고 버튼을 만들어 서버에 연동된 raspberry pi에 event를 전달하여 센서들을 컨트롤 할 수 있게 만들 예정이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명		
	팀 명		
	Confidential Restricted	Version 1.2	20xx-APR-30

5 고충 및 건의사항