

Iot 5조

-플랜팅 프로젝트-

2018125020 류호원 | 2017125018 김현석 | 2018125021맹성주

●프로젝트 배경

⊙반려식물의 필요성

-코로나로 인한 실내 인구의 증가

-환경 파괴로 인한 대기오염

⊙반려식물의 효과

-우울감 해소

-인테리어

-미세먼지 개선

⊙반려식물의 어려움

-수분, 온도 조절

●플렌팅 프로젝트

- 반려 식물은 살아가는데 있어서 많은 도움을 주지만 바쁜 현실에서 반려식물의 수분, 온도 등을 조절하며 관리하기는 어려운 일입니다.

저희는 그런 바쁜 현대인에게 맞추어 자동을 수분을 조절하고 식물의 상태를 확인하고 식물을 관리하는데 유용한 정보를 제공합니다.

또한 미세먼지 개선도, 온 습도 변화량 통계, 식물의 상태에 따른 디스플레이까지 상용화되지 않고 반려식물의 효과를 더 증진시킬 수 있는 기능들을 추가로 제공하여 사용자의 만족도를 더 높이고자 하였습니다!

●주요 기능 및 요구 사항

⊙온/습도 알림 및 확인

- 아두이노를 통해 시간별, 요청별 데이터를 수집

⊙자동 물주기

- 연결된 펌프를 이용하여 자동으로 물을 주는 기능

⊙미세먼지 확인

- 미세먼지 센서를 이용하여 미세먼지를 수치를 수집

⊙식물의 상태 디스플레이

- 식물의 상태가 좋은 지/ 잘 관리 되고 있는지는 식물만 보고는 알 수 없음으로 디스플레이로 온/습도에 따라 식물의 관리 상태를 표현

⊙온도, 습도, 미세먼지 통계 제공

- 온도, 습도, 미세먼지 수치를 수집하고 android에 그 수치와 통계를 제공함으로써 사용자가 각각의 정보를 확인 가능

⊙라이브 데이터 요청

- 정해진 시간이 아니더라도 사용자가 원할 때 언제든 식물의 상태를 제공 받을 수 있는 기능

⊙고온, 과습, 저온, 건조 상황에 대한 알림

- 날씨등의 외부요인으로 인해서 식물의 관리 상태가 안 좋아질 경우 적절한 조치를 알림

●구성원의 역할 및 개발 추진 실적

⊙2018125021맹성주

-

⊙2018125020 류호원

- 데이터 저장 람다함수 구성

- 아두이노로부터 받은 시간별 데이터를 람다함수를 이용하여 dynamodb에 저장

- 안드로이드 전송 람다함수 구성(시간 데이터 추가)

- dynamodb에 저장된 데이터를 람다함수를 이용하여 android로 전송

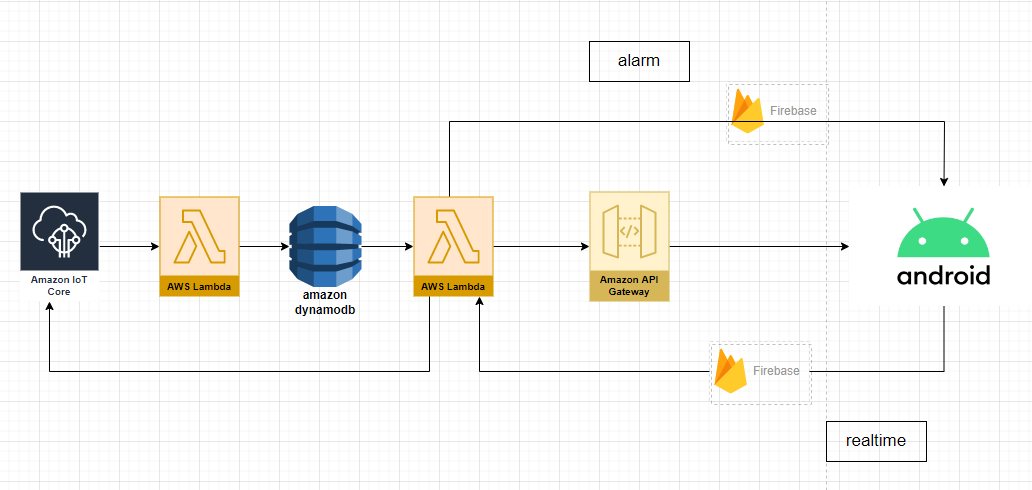
- 아두이노로부터 받은 라이브 데이터를 람다함수를 이용하여 android로 전송

- 데이터가 고온, 과습, 저온, 건조 인 경우에 각 상황에 맞는 안드로이드 알림 요청 및 구성

⊙2017125018 김현석

-

●Framework 구성



1. Iot core에서 data값을 받아옵니다

2. 람다 함수를 이용하여 받아온 데이터를 dynamo db에 저장합니다

3. dynamo db에저장된 데이터를 람다함수를 이용하여 android에 전송합니다

4. android에서는 받아온 데이터를 평균을 내어 사용자에게 유용한 정보를 제공합니다

-데이터 보기

-미세먼지 량 보기

-실시간 데이터 보기

5. 실시간 데이터를 요청하는 경우 firebase를 경유하여 iot core에 실시간 데이터를 요청하는 람다 함수 호출

6. 데이터가 고온, 과습, 저온, 건조 상황에는 람다와 firebase를이용하여 안드로이드에 적절한 알림

●기능 구현(esp 32-data인식 및 aws로 전송)

코드

~~코드코드코드

●기능구현(aws- data인식, 저장, android로 전송)

⊙Rules를 이용하여 esp32로부터 받아드린 data을 lambda로 전송

텍스트이(가) 표시된 사진

자동 생성된 설명

- flag5는 라이브데이터를 확인하는 flag로 라이브 데이터가 아닌 시간별 전송 값을 전송합니다

⊙ lambda를 이용하여 data를 dynamodb에 저장

텍스트이(가) 표시된 사진

자동 생성된 설명

- dynamodb에 맞는 형식으로 변경, 시간별 변화 추이를 확인하기 위하여 timezone함수를 이용하여 localtime을 받아오고, 한국시간에 맞게 변환 실행합니다

그후 dynamodb.put\_item으로 data저장합니다

⊙ dynamodb에 저장된 data

테이블이(가) 표시된 사진

자동 생성된 설명

⊙ data\_db에 저장된 data들을 api게이트웨이를 이용하여 android로 전송합니다

텍스트이(가) 표시된 사진

자동 생성된 설명

-Id 값을 0로 고정하여, id를 기준으로 데이터를 android로 전송합니다

⊙ 라이브 데이터 요청

텍스트이(가) 표시된 사진

자동 생성된 설명

- android로부터 현제 데이터 확인 요청이 오면, esp 32보드로 ask: livedata요청을 보내게 됩니다

텍스트이(가) 표시된 사진

자동 생성된 설명

- esp32에서 flag5로 현제 데이터를 받는다면 android\_alram 람다를 호출합니다

텍스트이(가) 표시된 사진

자동 생성된 설명

- 호출된 android\_alram함수에서는 flag들을 기준으로 고온, 과습, 저온, 저습 등의 식물 상태를 구분하여 android에 적절한 알람을 전송합니다

●기능구현(안드로이드)

-android에서 구현한 기능은 4가지가 있다. 첫번째는 ESP32가 주는 온도, 습도 데이터를 aws dynamoDB에 전달하는데 이 DB에 있는 데이터를 android가 받아와서 그래프와 텍스트로 표현한 것 두번째는 미세먼지도 마찬가지로 dynamoDB에 저장된 것을 24개씩 (하루를 기준) 받아 평균을 내어 그래프로 표현한 것 세번째는 실시간으로 esp32 데이터를 받는 기능인데 android <–> firebase <–> aws <–> esp32 경로를 통해 실시간 데이터 요청을 하면 데이터를 계속 받다가 해당 active를 벗어나면 데이터를 받지 않는 형식으로 구현하였다. 네번째는 물통에 물이 없거나 온습도와 미세먼지가 지나치게 높으면 알람을 주는 형식을 만들었다.

⊙ 첫번째(데이터 보기)

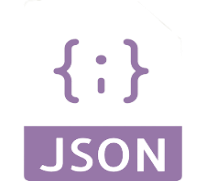
-MainActivity에서 데이터 보기 Button을 클릭하면 PostDataActivity로 이동한다. 가장 상단에 있는 EditText에 esp32에 해당하는 id(0,1,2)를 입력하고 검색 Button를 누른다. 그러면 REST API 방식으로 AWS GateWay에서 준 API(아래주소)를 이용하여 해당 입력했던 데이터를 JSON방식으로 AWS에게 전달하게 된다.

그 후 람다(Android\_AWS\_GetData)에게 데이터가 들어가 DynamoDB에서 Temp와 Humid 가져온다. 다시 이전처럼 AWS Gateway를 통하여 android 에게 JSON형식으로 데이터를 넘겨준다. 받은 데이터를 가지고 RecyclerView를 통해 각 데이터를 text로 LineChart를 이용하여 그래프로 표현한다.

https://vn23hf5ug1.execute-api.ap-northeast 2.amazonaws.com/default/Andorid\_AWS\_GetData(AWS API 주소)

⊙ 두번째(미세먼지량보기)

-MainActivity에서 미세먼지량 보기 Button을 클릭하면 FindChangeActivity로 이동한다. 상단에 esp32에 해당하는 id를 입력하고 검색 Button를 누르면 데이터 보기때의 과정을 거처 DynamoDB에서 데이터를 하루 기준 24개(1시간에 한번 데이터 저장)의 데이터 평균을 그래프로 표현한다.



⊙ 세번째(실시간보기)

- MainActivity에서 데이터 보기 Button을 클릭하면 RealTimeActivity로 이동한다. 가장 상단에 있는 EditText에 esp32에 해당하는 id(0,1,2)를 입력하고 검색 Button를 누른다. Android에서 AWS Gateway를 통해 람다(Android\_AWS\_Communicatrion)에 데이터(esp32 id ,0)를 JSON 형식으로 보내게 된다. 람다에서는 해당되는 esp32 shadow에 MQTT통신인 publish를 이용하여 실시간으로 데이터를 통신을 요청한다. Esp32는 브로커를 통해 해당 메시지를 subscribe 하여 실시간데이터를 shadow를 통해 다른 요청이 있을 때까지 계속 보낸다. 이 shadow에서 데이터를 받으면 AWS IOT Action인 규칙을 통해 람다(android\_alram)으로 보내진다. 여기서 flag를 통해 실시간 데이터 관련인지 파악한 후 조건에 맞는 publish를 하는데 이번에는 AWS가 아닌 firebase와 통신을 한다. 이는 람다에 Firebase\_admin 외부라이브러리를 import하여 사용하였다. 이 데이터를 받은 firebase는 Notification을 설정한 android 앱에 지속적으로 센서값을 Toast형식으로 띄우다가 이용자가 해당 RealTimeActivity를 나간다면 데이터(id,###)을 이전과 같이 esp32에 mqtt통신을 통해 데이터 전송을 멈추는 요청을 한다.

⊙ 네번째(알람)

-알람은 위와 같이 android – firebase – aws -esp32의 과정과 똑같고 다른부분은 esp32에서 shadow의 flag 값의 변화에 따라 다른 topic를 publish 하여 realtime 과 다른 알람과 구분하여 notification을 하여 유저에게 이상치에 대하여 알려준다.

●영상

●문제점 및 해결 내용

⊙ 아두이노의 다양한 센서와 케이블 사용으로 인한 심미적요소 저해

-

⊙aws-firebase 연동 이슈

-파이썬 람다에 외부 모듈 import(firebas\_admin)

⊙restfull api의 사용 및 이해

- restfull api 의 https관련 전송 방식 이해 후, json object를 활용하여 string변환하여 android 앱 활용

⊙ 미세먼지 센서의 오작동

- 미세먼지 센서에서 간혹 오작동으로 4200000의 오류값을 리턴

=> 센서의 문제라고 판단하여 해당 오류값이 리턴된다면 그 오류값은 삭제하고 통계적 이용을 위하여 보통 대기의 평균값(60) 으로 정보 제공

⊙ currentdata 요청 시 subscribe 실패

- livedata 요청 중단 메세지인 currentdata를 요청하였을때, aws shadow에는 currentdata가 subscribe가 되지만 아두이노 에서는 subscribe가 안되는 문제 발생

=> flag를 이동하여 확인해본 결과 livedata요청과 겹쳐져서 currentdata가 중간중간 구독이 안되는 현상 발생.

currentdata신호를 반복적으로 요청하여 다음에 바로 livedata가 덮이더라도 또currentdata가 들어가도록 수정하여 해결

●개인 기여내용 및 배운점

⊙2018125021맹성주

-

⊙2018125020 류호원

-

⊙2017125018 김현석