

디지털 트윈을 활용한 IOT 재배환경에 관한 연구

이해영*, 한혜영*, 이종표**

*성결대학교 산업경영공학과

**성결대학교 컴퓨터공학과

gksl2065@naver.com, heyyoungan13@gmail.com, qkrekfthsus@naver.com

HaeYoung-Lee* Hye Young-Han*, JongPyo-Lee*

*Dept. of Computer Science, Sung-kyul University

**Dept. of Computer Engineering, Sung-kyul University

요 약

본 논문에서 디지털 트윈 기술을 비닐하우스 환경 기반으로 하여 스마트팜 IOT 시스템을 구현하였다.

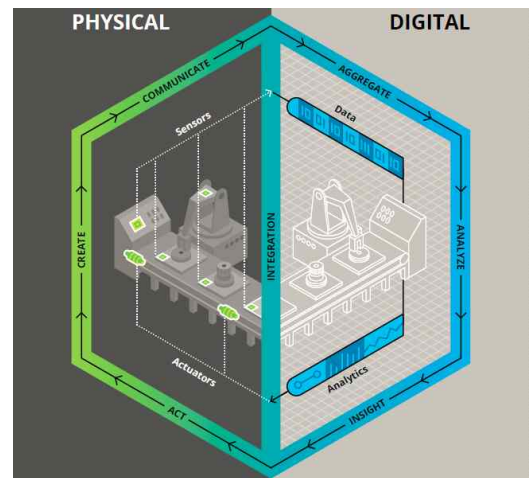
1. 서론

최근 디지털 트윈 개념을 도입한 사업과 연구가 각 분야에서 활발히 연구되고 있다. 현실에서 얻어온 데이터로부터 가상의 “쌍둥이” 객체를 만들어 다양한 시뮬레이션을 할 수 있는 디지털 트윈은 시뮬레이션을 통해 어떤 결과가 나올지에 대해 예측할 수 있어서 제조업이나 산업 등에서 주목받는 기술이다. 현재에는 제품의 불량률 감소, 시뮬레이션을 통한 운영 개선 등 다양한 분야에서 연구되고 있다.

제조업의 생산 과정에서도 주목을 받는 디지털 트윈을 생산 과정에서 피드백을 얻을 수 있다면, 농장에 적용해보는 것은 어떨까? 작물이 자라는 동안에 작물 상태에 대해 피드백을 얻고, 더 좋은 양질의 작물을 수확할 수 있을 것이다. 2018년도의 농가 인구는 231만 5천 명으로 전년 대비 약 10만 7천 명 감소(-4.4%)했으며, 농가 인구 비율은 총인구의 4.5%로 우리나라의 농업 인구가 점점 감소세를 보이고 있다. 농업에 관한 관심이 사라지고 있는 현대에 이러한 디지털 트윈 기술을 이용하여 농장과 도시농업에서 더 좋은 품질의 작물과 관리환경을 조성한다면 작물을 수확할 때, 더욱 만족감과 성취감을 느낄 수 있고, 농업에 관한 관심이 높아질 것으로 예상된다.

2.본론

2.1 디지털 트윈 시스템의 요구사항

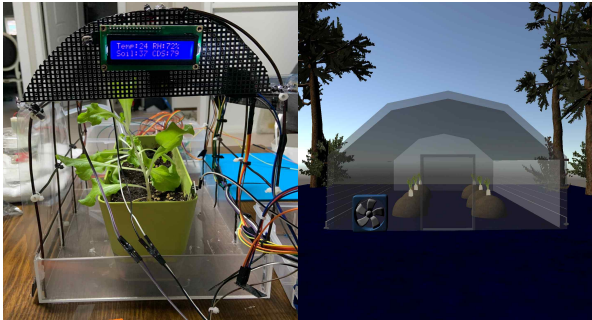


[그림 11] 제조 공정 디지털 트윈 모델

디지털 트윈을 활용하려면 현실에서 수행되는 부분(physical)과 가상에서 수행되는 부분(digital)을 구분해야 한다. 먼저, 현실에서는 센서(sensor) 등을 통해 자료들을 수집하고, 가상 세계에서 데이터를 종합 및 분석하여 시뮬레이션을 시행한다. 시뮬레이션을 통해 얻은 정보들로 다시 현실 세계로 적용하고, 제품을 만든다. 이러한 주기(Cycle)를 통해 IoT 재배환경에 디지털 트윈을 적용할 것이다.

1) Aaron Parrott 외, Industry 4.0 and the digital twin, Deloitte University Press, 5, 2017.

첫 번째로, IoT 재배환경에 디지털 트윈을 적용하기 위해서는 현실과 가상 세계를 어떤 방식으로 구현을 해야 할지 결정해야 한다. 현실에서는 아두이노(Arduino)를 이용하여 미니 비닐하우스를 제작하였고, 가상 세계는 유니티(Unity3D)를 통해 가상의 비닐하우스 환경을 제작하였다.



[그림 2] 현실과 가상의 객체

두 번째로, 유니티와 아두이노에 식물, 팬, 블라인드 등 작동기(Actuator)도 구현해 현실에서 자료들을 수집한 후, 데이터에 따라 식물의 상태를 나타내거나, 조정해줄 수 있도록 하였다.

2.2 개발환경

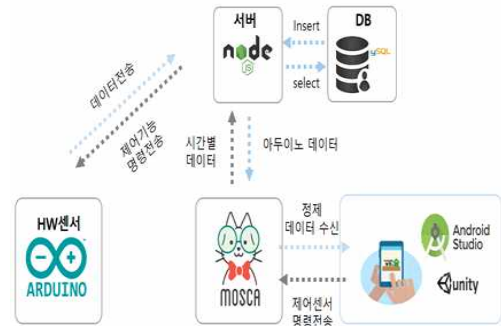
[표 1] Development environment

개발 환경	프로그래밍 언어
Microsoft Visual Studio	JavaScript(Node.js-Express Framework)
SQLyog Community	MySQL
Unity3D	C#
Arduino Sketch	C/C++
Android Studio	Java

IoT 재배환경과 디지털 트윈의 구현은 [표 1]과 같은 시스템을 구성했다.

Arduino와 Android 간 통신과 MQTT 통신, DB를 위해 중개 서버를 만들었고, 가상 환경을 위한 Unity와 Android 어플리케이션을 구현하여 Unity와 연동하여 사용자에게 디지털 트윈을 보여주는 방식이다.

2.3 서비스 및 S/W 구성도



[그림 3] 서비스 구성도

소프트웨어의 구성도는 [그림 2]와 같이, 아두이노 기준으로 송신과 수신 파트로 나뉜다. 아두이노에서 데이터를 송신 시, 센서로 부터 얻은 데이터들을 서버로 전송하고, 서버는 DB에 데이터를 저장하며, MQTT 브로커인 MOSCA와 연동하여 Android와 Unity에 데이터를 송신하여 사용자에게 보여준다.

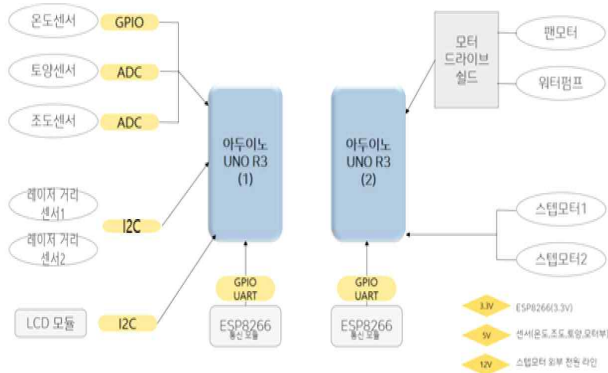
아두이노에서 데이터를 수신할 때에는 Android 앱에서 3자리로 된 명령어 코드를 MOSCA를 통해 서버로 보내고, 명령어 값을 저장한다. Arduino의 제어와이파이모듈(Esp8266)을 폴링방식(polling)으로 적용하여, 서버에 일정 시간마다 접속하여 응답 값을 파싱하여 스마트 재배환경을 제어한다.



[그림 4] S/W 기능 구성도

클라이언트에게 제공되는 S/W의 기능들은 [그림 3]과 같다. 사용자에게 작물에 대한 정보를 제공하고, 재배환경을 원격으로 제어할 수 있는 기능과 디지털 트윈을 통해 작물 상태에 대해 피드백을 받을 수 있다. 또한, 사용자는 농장 일기를 작성하여, 일별 농장에 대한 상태를 기록한다.

2.4 H/W 구성도



[그림 5] H/W 구성도

하드웨어의 구성도는 위의 [그림 5]와 같다. 총 2대의 아두이노와 모듈 총 5가지 구성되어 있다. 모듈 중, 제어 모듈에는 블라인드, 스텝 모터, 워터 펌프와 와이파이 모듈(Esp-01)이 있고, 센서 모듈에는 각종 센서(온습도, 토양습도, 조도, 거리)와 LCD 모니터, 와이파이 모듈이 있다.

2.5 제어 코드

```
try {
    mqttClient.publish(topic, "Cmd", cmd.toString().getBytes(), qos, 0, retained: false);
} catch (MqttException e) {
    e.printStackTrace();
}

else if (topic == "Cmd"){
    cmd = message.toString().split("");
}

});

void parseData(String res){
    // 데이터가 [xxxx]이런식으로 저장되어있기 때문에 "[]"부분과 "[]"부분을 찾아서 파싱
    int code_data = res.indexOf("[");
    if (code_data >= 0){
        String tmp_str = "";
        code = res.substring(res.indexOf(tmp_str) + tmp_str.length(), code_data);
    }
}

void connection(){
    //WiFiEspClient client;
    //client.flush();
    client.stop();
    client.setTimeout(5000);

    //GET Method를 통해 HTTP request
    if (client.connect(server, 3000)) {
        String message = "";
        client.print(String("GET ") + cmd + " HTTP/1.1\r\n" +
            "Host: " + server + "\r\n" +
            "Connection: close\r\n\r\n");

        //HTTP Response
        while(client.available()){
            char c = client.read();
            message += c;
        }
        parseData(message);
        //client.stop();
    }
}
```

[그림 6-1] 명령어 코드 전송

```
connection();
Serial.println(code);

if (code[0] == '1'){
    Serial.println("Go");
    _Go();
} else if (code[0] == '0'){
    Serial.println("Back");
    _Back();
}

if (code[1] == '1'){
    Serial.println("Water_on");
    pumpstart();
} else if (code[1] == '0'){
    Serial.println("Water_off");
    pumpstop();
}

if (code[2] == '1'){
    Serial.println("Fan_start");
    fanstart();
} else if (code[2] == '0'){
    Serial.println("Fan_stop");
    fanstop();
}
```

[그림 6-2] 아두이노 명령어 처리 부분

다음 코드는 안드로이드에서 명령어 코드를 MQTT를 통해 서버로 송신하고, 아두이노는 폴링을 통해 명령어 코드를 수신하여 처리하는 부분이다.

2.6 비닐하우스와 디지털 트윈의 접목

소프트웨어와 하드웨어를 이용하여 현실의 비닐하우스와 가상의 비닐하우스를 접목하기 위해, 아두이노로 수집된 데이터로부터 작동기 제어까지 과정별로 나누어 보았다.

- 데이터 수집 및 분석

Time	Temp	Hum	Soil	Illu
2020-10-05 17:05:25	24	31	4	76
2020-10-05 17:05:27	24	31	5	77
2020-10-05 17:05:29	24	29	4	76
2020-10-05 17:05:31	24	31	4	76
2020-10-05 17:05:33	24	29	4	76
2020-10-05 17:05:35	24	29	5	77
2020-10-05 17:05:37	24	29	4	76
2020-10-05 17:05:39	24	31	5	76
2020-10-05 17:05:41	24	31	5	77
2020-10-05 17:06:01	24	31	4	76
2020-10-05 17:06:05	24	31	4	76
2020-10-05 17:06:07	24	31	6	76
2020-10-05 17:06:09	24	28	5	76



[그림 7] 센서 데이터 수집 및 분석

수집 및 분석 단계에서는 온도, 습도, 토양습도, 조도와 같은 값들은 아두이노에서 다양한 센서를 통해 값들을 DB 서버에 저장하게 된다. [그림 7]을 보았을 때, 사람은 현재의 토양수분이 부족하여 물이 필요한 것으로 판단한다.

- 이해



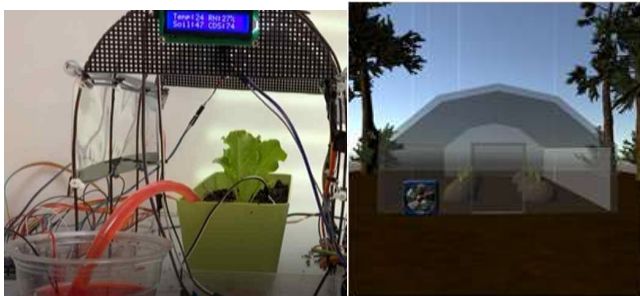
오늘의 농장 소식

● 물이 필요합니다.

[그림 8] 센서 데이터에 따른 유니티 환경 변화

DB 서버에 센서 데이터들을 저장하게 되면, 현실 세계의 데이터를 이용하여 가상 환경을 적용한다. 이해 단계에서는 실제 환경을 이해하고, 이를 사용자에게 보여준다. [그림 8]에서는, 수분이 부족하므로 가상 환경인 유니티에서는 땅의 색깔이 파란색으로 바뀐다.

- 실행 및 최적화



[그림 9] 사용자의 유니티 환경 제어

실행 및 최적화 단계에서는 사용자가 직접 가상 환경의 피드백을 통해 식물이 자라는 환경을 개선한다. [그림 9]에서는 [그림 8]에서 토양수분이 부족하여, 수중 펌프를 제어하는 모습을 실제 환경과 가상 환경으로 보여준다. 물이 부족한 상태뿐만 아니라, 냉각팬을 작동시켜 온도를 낮추거나 블라인드를 제어할 수도 있다.

□	Time	Temp	Hum	Soil	Illu
□	2020-10-05 17:05:35	24	29	5	77
□	2020-10-05 17:05:37	24	29	4	76
□	2020-10-05 17:05:39	24	31	5	76
□	2020-10-05 17:05:41	24	31	5	77
□	2020-10-05 17:06:01	24	31	4	76
□	2020-10-05 17:06:05	24	31	4	76
□	2020-10-05 17:06:07	24	31	6	76
□	2020-10-05 17:06:09	24	28	5	76
□	2020-10-05 17:06:11	24	28	10	77
□	2020-10-05 17:06:13	24	30	17	75
□	2020-10-05 17:06:15	23	30	28	76
□	2020-10-05 17:06:17	23	30	40	77
□	2020-10-05 17:06:19	23	31	61	77
□	2020-10-05 17:06:21	23	31	75	76

[그림 10] 환경 제어를 통한 데이터 변화

[그림 10]과 같이, 사용자가 수중 펌프를 제어 후 데이터가 변화하여, 가상 환경도 변화한 모습이다.

3. 결론

신기술인 디지털 트윈을 접목하면서, 신기술이 식물 재배 관리의 적합성과 영농 편의성에 대해서 어떤 영향을 미치는가에 대해 연구를 진행하였다.

첫 번째로, 편리한 관리 시스템 환경 구축이다. 식물에 대한 기본 정보를 입력하고 실시간 데이터를 어플리케이션을 전송받으면, 실시간으로 식물의 한

경을 제어하여, 실제 농장을 방문하지 않아도 식물의 상태를 가상세계(유니티)로 확인할 수 있다.

두 번째, 질 높은 식물 환경 및 재배율 증가이다.

신기술 접목 전 상추 재배에 대한 환경을 만드는 과정과 접목 후의 식물 생장은 각각 다른 변화를 나타내고 있었다.

여름철 재배환경에 동일한 조건으로 상추를 재배했으며, 상추가 잘 자라는 환경 온도는 22~25℃ 사이이다.

접목 전, 상추의 환경 관리는 매번 식물의 상태를 육안으로 확인하며 물과 그늘을 배치하는 방식이었다. 그러한 이유로 토양습도의 정확성이 떨어지는 어려움을 겪었다.

접목 후, 상추의 환경 관리는 이전과 다른 양상을 띠었다. 실시간 데이터를 보여주며, 농장 공간이 있지 않아도 관리가 쉬웠으며, 데이터에 따른 토양의 습도와 내부 온도를 조절해주어 접목 전보다 5일 정도 빠르게 수확할 수 있었다.

본 논문은 스마트팜 모니터링을 기반으로 디지털트윈을 활용한 스마트 IoT 재배 관리 환경을 구현하면서, 신기술이 재배 관리의 적합성과 영농 편의성을 제공한다는 연구결과를 도출했다.

디지털 트윈의 기술은 스마트 시티와 제조 분야뿐만 아니라 농업에도 큰 영향을 미칠 수 있다.

농업이 가지는 고질적인 노동력 부족 문제를 해결할 수 있으며, 도시 농업의 증가로 필요한 다양한 편의 시스템이 요구되는 상황에서 디지털 트윈의 적용은 도시 농업에 커다란 확장을 가져다줄 것이라고 예상한다.

더 나아가 머신러닝 기반의 식물 생육 패턴을 분석하여 사용자에게 적절한 재배 시기와 자동 예측 시스템을 도입된다면 차세대 기술로 디지털 트윈 시대가 열릴 것이다.

이 논문은 2020년도 미래창조과학부 ICT인재 양성 사업 한이음의 재원으로 지원을 받아 수행된 기초연구사업이다.

참고문헌

-IITP, 글로벌 과학기술정책정보서비스, 주요국의 디지털 트윈 추진 동향과 시사점, 이슈 분석 133호,

2020-02-14
https://now.k2base.re.kr/portal/issue/ovsealssued/view.do?polisuid=ISUE_000000000000931&menuNo=200046&pageIndex=1 참고)