

다중 재배 환경을 고려한 이동형 스마트팜 모니터링 로봇

김종현*, 채수현*, 전지예*, 김광호*, 윤창환**, 김장원*, 정동원*

*군산대학교 소프트웨어융합공학과

**군산대학교 기계자동차공학과

e-mail: {95chupa, shchae, nayajiyee, ouranos}@kunsan.ac.kr,

kt0654@naver.com, {jwgim, djeong}@kunsan.ac.kr

Mobile Smart-Farm Monitoring Robot for Multi-Crop Cultivation Environments

Jong-Hyeon Kim*, Soo-Hyeon Chae*, Ji-Yee Jeon*, Gwang-Ho Kim*,
Chang-Hwan Yun**, Jangwon Gim*, Dongwon Jeong*

*Dept of Software Convergence Engineering, Kunsan National University

**Dept of Mechanical and automotive Engineering, Kunsan National
University

요 약

이 논문에서는 다중 재배 환경을 고려한 이동형 스마트팜 모니터링 로봇을 제안한다. 기존 스마트 농장 연구는 각 구역마다 고정형 측정 장치를 이용하여 데이터를 수집한다. 또한 고정형 센서 기반 스마트 농장이므로 다중, 다수의 센서가 필요하다는 문제점이 있다. 이 논문에서 제안하는 것은 기존의 스마트 농장과는 달리 스마트팜 모니터링 로봇은 농장의 모든 구역을 이동을 하며 데이터를 측정, 수집한다. 그리고 수집한 데이터를 웹 페이지를 통해 시각화한다. 그 결과 제안한 시스템을 통해 자동으로 이동 및 측정을 하여 다중의 재배 환경에서도 안정적으로 정확한 측정이 가능하다는 장점이 있다.

1. 서론

지구 온난화로 인해 오는 2100년이면 세계 최대 농작물 수출국인 미국의 밀과 옥수수, 콩 등 주요 농작물 생산이 심각한 타격을 입게 될 것이다[1].

매년 농작물이 수요에 비해 공급이 미치지 못한다는 것이 현실이다. 이러한 문제점을 해결하기 위해 적은 힘으로도 농장을 쉽게 관리와 감독할 수 있는 시스템인 스마트 농장 연구가 활발하게 진행되고 있다[2~4].

스마트 농장 연구는 농장의 각 구역마다 센서를 부착하여 환경을 측정 및 제어가 가능하다. 그런데 각 구역마다 센서를 구비해야 하고, 환경에 따른 작물의 성장, 즉 같은 구역이여도 작물이 받는 일조량, 수분공급량들을 고려하지 않는 문제점과 다량의 센서가 필요한 문제점이 있다. 또한 사람이 재배 환경을 제어하는데 체온으로 인한 측정의 부정확함과 높은 위치 측정의 어려움이 있다.

따라서 이 논문에서는 각 재배구역마다 이동형 로봇이 데이터를 측정, 수집해 데이터베이스에 저장하여 작물 성장환경을 측정하고 웹 페이지에 통해 수집한 데이터를 실시간 모니터링 할 수 있는 이동형 스마트팜 모니터링 시스템을 제안한다. 이러한 결과 환경 측정을 위해 구비해야 하는 센서의 수가 감소하고, 사람이 측정을 할 시 측정 위치가 다를 수 있고, 측정 시간이 정확하게 지켜지지 않을

수 있는 단점이 보완된다. 그리고 웹 페이지를 통해 다양한 단말기로 원거리 모니터링이 가능하다.

이 논문의 구성은 다음과 같다. 제2장에서는 스마트 농장의 관련연구에 대해 설명하며, 제3장에서는 전체 시스템 구성과 구현에 필요한 설계도면을, 제4장에서는 구현 결과에 대해 기술한다, 제5장에서는 결론 및 향후 연구에 대해 서술한다.

2. 관련연구

사물인터넷 기반의 스마트 농장관리 연구는 낮은 비용으로 농장을 쉽게 관리, 감독하는 시스템이다. 이 연구는 기상 센서를 이용해 데이터를 수집하고, 수집된 데이터를 액추에이터로 제어하는 시스템이다[2].

스마트 팜 팩토리를 위한 IoT 플랫폼 연구에서는 식물 공장 생산성 향상을 지능적으로 운영하는 시스템이다. 이 연구는 스마트 팜 팩토리라는 재배구역에서 센서를 통해 CO2, 정수, 양액, 광도, pH, 온도, 습도, 기류, 공조 등의 환경을 측정한다[3]. 그러나 다른 재배구역에서 환경을 측정하기 위해선 이러한 다수의 센서간의 데이터 보정 및 센서데이터 품질 관리가 필요한 문제점이 있다.

사물인터넷 기반의 농장물 관리 시스템 구현 연구에서는 농장내의 환경 정보를 데이터를 수집해 실시간으로 제어하고 라즈베리파이 카메라를 장착해 농장의 내부 모습

† 책임저자 : 군산대학교 김장원, 정동원

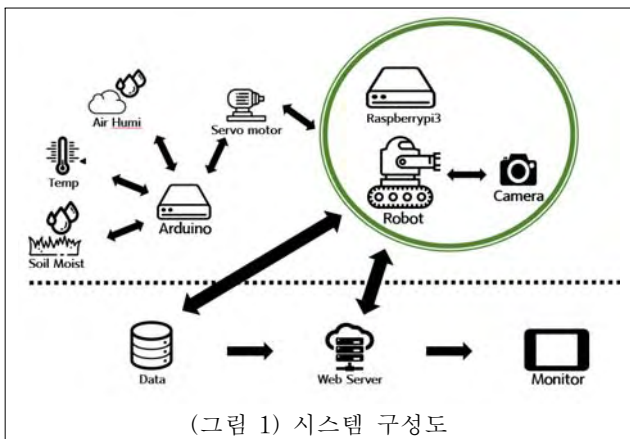
을 wifi를 통해 스마트 기기로 모니터링하는 시스템이다 [4]. 그러나 이런 시스템은 같은 구역이어도 서로 다른 센서를 사용하거나, 측정 방법 및 측정 시간의 차이에 따라 측정 데이터의 값의 차이가 발생하여 센서 데이터 값의 품질 보정 및 정제 이슈가 발생한다.

이 논문은 수평형, 수직형, 다단형 등의 다양한 재배구역에서 발생할 수 있는 이슈들을 해결하기 위한 이동형 스마트팜 로봇 시스템을 제안한다. 제안 시스템에 다양한 센서를 부착하여 재배 배드 사이를 안정적으로 이동할 수 있으며 정해진 시간별로 반복적, 주기적으로 재배구역을 이동하여 사람에 의한 수작업 측정 보다 정확한 측정을 수행할 수 있다.

3. 시스템 설계

3.1 시스템 구성

그림 1은 다중 재배 환경을 고려한 이동형 스마트팜 모니터링 로봇 시스템 구성을 위해 로봇에 라즈베리파이와 아두이노 및 기상 환경 센서를 기반으로 구성되어 있는 시스템 구성도이다.

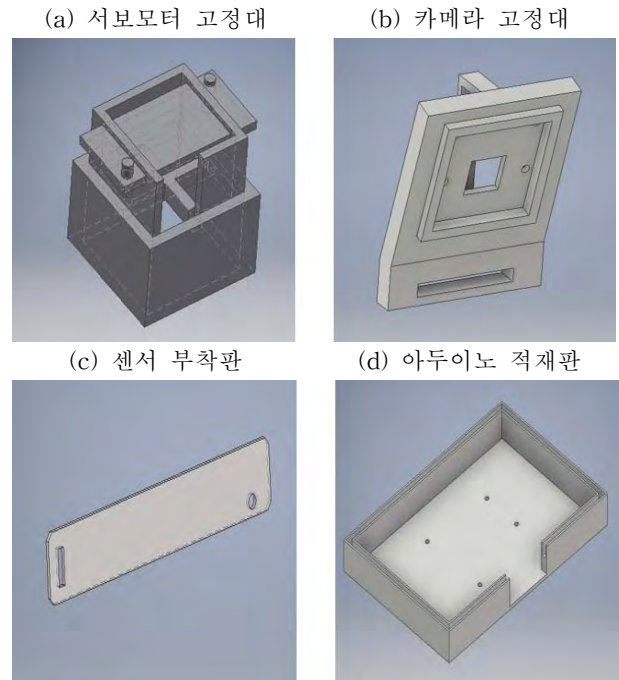


로봇은 아두이노와 라즈베리파이가 메인으로 구성된다. 라즈베리파이에는 MySQL를 이용하여 데이터베이스를 구축하였고, 웹 페이지를 연동하기 위하여 Nodejs로 서버를 구축한다. 라즈베리파이 카메라를 통해 측정 당시의 작물의 환경 사진도 촬영한다. 아두이노에 서보모터를 장착해 온습도, 토양습도 센서들이 로봇의 팔 형태로 제어하여 데이터 값을 측정한다. 측정한 후 데이터베이스에 데이터 값을 누적, 적재한다. 누적된 데이터 값을 웹 페이지에 테이블로 출력하고, 그래프를 통해 데이터 값들을 시각화한다. 또한 웹 페이지를 통해 라즈베리파이 카메라로 촬영한 작물의 사진도 출력한다.

3.2 설계도면

그림 2는 3D프린터로 다중 재배 환경을 고려한 이동형 스마트팜 모니터링 로봇 로봇을 제작하기 위해 설계한 도

면이다. AUTODESK INVENTOR 라는 소프트웨어를 이용하여 도면을 설계 하였다.



(그림 2) 프로토타입 제작을 위한 3D 도면

그림 2(a)는 서보모터를 고정하는 역할과 동시에 서보모터가 원활하게 움직이도록 설계하였고, 그림 2(b)는 라즈베리파이 카메라를 고정시키기 위한 고정대로 로봇의 본체에 탈, 부착이 가능하게 설계하였다. 그림 2(c)에는 센서를 부착하여 위아래로 움직일 수 있는 일종의 로봇 팔의 모습으로 설계하여 서보모터와 연결이 가능하고, 그림 2(d)는 라즈베리파이와 아두이노를 적재하기 위한 설계도면을 보여준다.

4. 구현

그림 3은 다중 재배 환경을 고려한 이동형 스마트팜 모니터링 로봇의 구현 내용이다.

그림 3(a)는 Node.js를 통하여 라즈베리파이 안에 서버를 생성 후 데이터베이스와 연결을 하고 웹 서버를 생성 한 후 URL주소를 통해 웹 페이지에 접속이 가능한 소스코드 중의 일부이다.

그림 3(b)는 이동형 스마트팜 모니터링 로봇이 자동으로 트래킹하기 위한 소스코드 일부와 아두이노에 부착된 서보모터를 제어하고 온습도, 토양습도 센서들을 통해 데이터 값을 측정하기 위한 소스코드이다.

그림 3(c)는 아두이노 센서들을 통해 측정된 데이터 값을 MySQL을 통해 연동한 데이터베이스에 누적하여 날짜별로 테이블에 출력한다.

(a) Node.js 소스코드

```

var app = express();
http.createServer(app).listen(8000,function(){
  console.log("Server running at http://127.0.0.1:8000");
});
app.use(express.static(path.join(__dirname, 'public')));
// http://localhost:8000/ ---- list.html
app.get('/list',function(req,res)
{
  fs.readFile('list.html','utf8',function(error,data)
  {
    if(error)
    {
      console.log("readFile Error");
    }
    else
    {
      mySqlClient.query("select * from sensor;", function(error,results)
      {
        if(error)
        {
          console.log("error: ", error.message);
        }
        else
        {
          res.send(ejs.render(data,{ prodList:results }));
        }
      });
    }
  });
});

```

(b) 아두이노 소스코드

```

void servoinit() {
  digitalWrite(SERVO_PIN, LOW);
  pinMode(SERVO_PIN, OUTPUT);

  TCCR2A = (1 << WGM21);
  TCCR2B = (1 << CS21);

  TCNT2 = 0;
  OCR2A = 255;

  TIMSK2 |= (1 << OCIE2A);
  sei();
}

void servoSetPosition(uint16_t highTimeMicroseconds) {
  TIMSK2 &= ~(1 << OCIE2A);
  servoHighTime = highTimeMicroseconds * 2;
  TIMSK2 |= (1 << OCIE2A);
}

float getHumi() {
  float h = dht.readHumidity();
  return h;
}

float getTemp() {
  float t = dht.readTemperature();
  return t;
}

int getMoist() {
  int m = analogRead(MOIST_PIN);
  return m * 100 / 1023;
}

```

(c) 수집된 센서 데이터 (MySQL)

```

Database changed
mysql> select * from sensor;
+-----+-----+-----+-----+-----+
| date       | temp  | humi  | moist | indexkey |
+-----+-----+-----+-----+-----+
| 20170728053321 | 25.8  | 54.3  | 0     | 20170728 |
| 20170728053323 | 25.4  | 56.4  | 0     | 20170728 |
| 20170728053325 | 25.4  | 56.7  | 0     | 20170728 |
| 20170728053331 | 25.4  | 56.8  | 0     | 20170728 |
| 20170728053334 | 25.4  | 57.1  | 0     | 20170728 |
| 20170728053336 | 25.4  | 57.4  | 0     | 20170728 |
| 20170728053343 | 25.4  | 57.7  | 0     | 20170728 |
| 20170728053346 | 25.5  | 57.9  | 0     | 20170728 |
| 20170728053348 | 25.5  | 58.1  | 0     | 20170728 |
| 20170728053354 | 25.5  | 58.2  | 0     | 20170728 |
| 20170728053356 | 25.5  | 58.1  | 0     | 20170728 |
| 20170728053358 | 25.5  | 58     | 0     | 20170728 |
| 20170728083706 | 23.9  | 62.9  | 0     | 20170728 |
| 20170728085000 | 25.1  | 55.4  | 4     | 20170728 |
| 20170728085227 | 25.2  | 56.2  | 1     | 20170728 |
| 20170728085229 | 25.6  | 55.1  | 2     | 20170728 |
+-----+-----+-----+-----+-----+

```

(그림 3) 구현 소스코드 및 수집된 데이터

그림 4는 측정한 데이터를 웹 페이지로 출력한 화면이다. 그림 4(a)는 구현한 웹사이트의 메인화면이고, 그림 4(b)는 측정한 데이터 값을 실시간으로 출력하는 화면이다. 그림 4(c)는 재배구역을 자동으로 트래킹하여 측정할 시 라즈베리파이 카메라를 통해 촬영한 작물의 사진을 출력한 화면이다.

(a) 웹 페이지 메인화면



(b) 실시간 센싱데이터 출력화면

date	temp	humi	moist	indexkey
20170728053321	25.8	54.3	0	20170728
20170728053323	25.4	56.4	0	20170728
20170728053325	25.4	56.7	0	20170728
20170728053331	25.4	56.8	0	20170728
20170728053334	25.4	57.1	0	20170728
20170728053336	25.4	57.4	0	20170728
20170728053343	25.4	57.7	0	20170728
20170728053346	25.5	57.9	0	20170728
20170728053348	25.5	58.1	0	20170728
20170728053354	25.5	58.2	0	20170728
20170728053356	25.5	58.1	0	20170728
20170728053358	25.5	58	0	20170728
20170728083706	23.9	62.9	0	20170728
20170728085000	25.1	55.4	4	20170728
20170728085227	25.2	56.2	1	20170728
20170728085229	25.6	55.1	2	20170728
20170728085309	25.4	55.9	0	20170728

(c) 작물의 사진출력화면



(그림 4) 웹 페이지 구현

그림 5는 측정한 데이터를 그래프로 시각화한 화면이다. 그림 5(a)는 날짜 별 온도의 값을, 그림 5(b)는 날짜 별 습도의 값을, 그림 5(c)는 날짜 별 토양수분 데이터 값을 시각화 한 사진으로 수분이 없을 시 0 수분이 많을시 3으로 값이 표시된다.

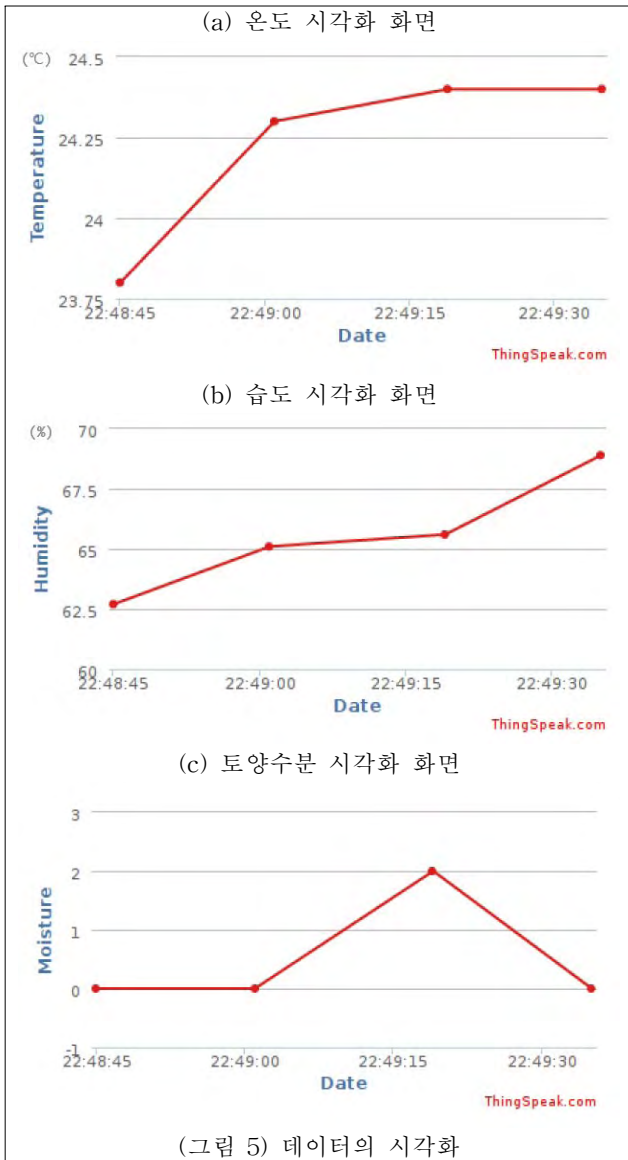
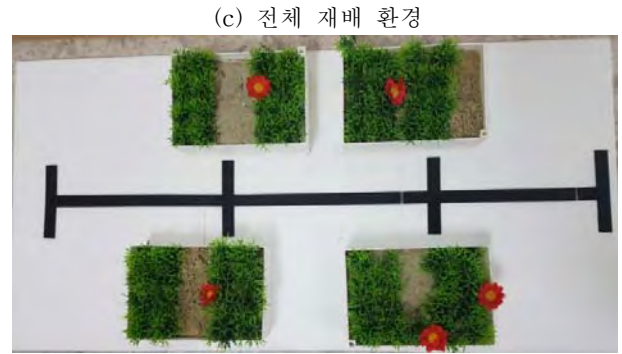


그림 6는 데이터를 측정하기 위한 로봇이다. 그림 6(a)는 기본적으로 아두이노와 라즈베리파이를 적재한 로봇의 프로토타입이고, 그림 6(b)는 로봇이 직접 환경을 측정하여 데이터를 측정하는 모습이다.



(그림 6) 모니터링 로봇 프로토타입

5. 결론

이 논문에서는 아두이노 및 다양한 기상 센서와 라즈베리파이를 이용한 이동형 스마트팜 재배 환경 측정 로봇을 제안하였다. 제안 로봇은 재배 구역을 움직이며 데이터를 정해진 시간마다 자동으로 수집하고 실시간으로 데이터베이스에 저장한다. 또한 라즈베리파이에 탑재된 웹 서버를 통한 수집된 데이터의 실시간 분석이 가능한 웹 인터페이스를 제공한다.

이 시스템은 무인으로 정해진 경로를 이동할 수 있으며, 정확한 시간에 재배 환경을 측정하고 동일한 센서를 이용한 안정적인 측정이 가능하다.

향후 연구에서는 측정된 데이터를 바탕으로 가정용 텃밭 관리 로봇과 같이 작물에게 직접 수분을 공급하고 공기를 환기 시켜주는 것 등의 환경 제어 기능을 추가하는 연구를 진행할 것이다.

참고문헌

- [1] 한국과학기술자협회, “기후변화, 식탁을 위협한다, 농작물 수확량 '반토막'”, <https://goo.gl/Ut3Er2>, 2017.
- [2] 유영선, 김백기, “사물인터넷 기반의 스마트 농장관리 시스템” 한국정보기술학회 한국디지털콘텐츠학회 하계공동학술대회 논문집, pp. 427~429, 2017.
- [3] 황성일, 주성용, 주종문, “스마트 팜 팩토리를 위한 IoT 플랫폼 연구” 한국통신학회 2016년도 동계종합학술발표회, pp. 520~521, 2016.
- [4] 김경아, 정연만, 박동영, “사물인터넷 기반의 농장물 관리 시스템 구현” 한국통신학회 2016년도 동계종합학술발표회, pp. 366~367, 2016.