

농지정보 획득 시스템의 소프트웨어 구현에 관한 연구

이병주*, 김종원**, 곽윤식***

A Study on Software Implementation of Agricultural Land Information Acquisition System

Byungju Lee*, JongWon Kim**, and Yoonsik Kwak***

본 과제(결과물)는 2025년도 교육부 및 충청북도의 재원으로 충북RISE센터의 지원을 받아 수행된 지역혁신중심 대학지원체계(RISE) 글로벌대학30의 결과입니다.(2025-RISE-11-004)

요 약

본 연구는 농지의 위치 및 경계 정보(위도, 경도) 그리고 생육환경 정보(온도, 습도)를 획득하는 디지털화 단말장치, 스마트기기 그리고 서버로 구성되는 농지정보 획득 시스템의 소프트웨어 구현을 목적으로 한다. 이를 위하여 단말 장치의 하드웨어에 대한 초기화(센서 및 통신모듈 등), 통신 프로토콜, 스마트기기와의 연동 그리고 서버 구축에 관한 소프트웨어를 설계, 구현(통신성공률, 98.07% 실거리오차, 2.5CEP)하였다. 구현된 프로토타입의 시스템을 기반으로 실시간 데이터 구축을 위한 모의실험 수행 및 획득 정보의 관리를 위한 DB를 구축하였으며 이를 활용하여 농지 관련 응용 분야(농지 면적 계산, 영농시설 설계, 생육 환경의 개선 등) 및 농산업의 경쟁력 확보를 위한 기초 데이터를 획득할 수 있었다.

Abstract

This study aims to implement software for a farmland information acquisition system consisting of a digitization terminal device, smart device, and server that acquire location and boundary information (latitude and longitude) as well as growing environment information (temperature and humidity) of farmland. To this end, software was designed and implemented for the initialization of the terminal device hardware (sensors, communication modules, etc.), communication protocols, integration with smart devices, and server construction (communication success rate, 98.07% Actual distance error, 2.5CEP). Based on the implemented prototype system, simulations were conducted for real-time data collection, and a database was constructed for managing the acquired information. Utilizing this, it was possible to obtain foundational data for farmland-related application areas (farmland area calculation, agricultural facility design, improvement of growing environments, etc.) and for securing the competitiveness of the agricultural industry.

Keywords

information acquisition system, devices initialization, communication protocol, environmental DB

* (주)보아스에스이 대표이사

- ORCID: <https://orcid.org/0000-0002-9667-3351>

** 국립한국교통대학교 컴퓨터공학과 강사

- ORCID: <https://orcid.org/0000-0001-8396-5075>

*** 국립한국교통대학교 컴퓨터공학과 교수(교신저자)

- ORCID: <https://orcid.org/0000-0002-7407-207X>

• Received: Feb. 24, 2026, Revised: Apr. 27, 2026, Accepted: Apr. 30, 2026

• Corresponding Author: Yoonsik Kwak

Korea National University of Transportation

Tel.: +82-43-841-5345, Email: yskwak@ut.ac.kr

I. 서 론

컴퓨터 기술의 발전과 인터넷 기술의 발전은 모든 산업 분야에 지대한 영향을 미치고 있다. 특히 4차 산업혁명과 더불어 전 산업 분야에서 시너지 효과의 획득을 목적으로 다양한 기술의 융합 형태가 나타나고 있으며 농산업 분야에서도 동일한 기술 발전 동향이 나타나고 있다[1][2].

최근 농산업 기술의 발전 동향으로 기계화, 자동화, 과학화 전략이 채택되어 이를 달성하기 위한 경제적 그리고 정책적인 투자가 국가적 핵심 정책으로 제시, 지원되는 것도 세계적인 국가정책 동향이다. 이는 농산업이 국가의 핵심 전략 산업으로 자리 잡고 있으며 국가의 생존 전략으로 간주하는 반증이다.

또한 농산업의 자동화, 과학화를 위한 융합기술로 다양한 응용 분야에서 적용되는 농산업 관련 정보의 디지털화를 기반으로 하는 정밀농업 기술이다. 이와 같이 농산업 관련 분야에 관한 정보의 디지털화는 기초적 그리고 우선적으로 확보되어야 할 가장 중요한 인프라 요소로 간주되어 국가 차원의 투자가 진행되고 있으며 성과도 도출되고 있다[3]-[6].

그림 1은 논문에서 제안한 농지 정보 획득 시스템(지리정보 및 생육환경 정보)으로 디지털화 단말장치, 스마트기기 그리고 데이터베이스 구축을 위한 서버 시스템으로 구성되며 실시간의 농지 정보 획득 모형을 나타낸 것이다.



그림 1. 실시간 농지 정보 획득 모형
Fig. 1. Real-time agricultural land information acquisition model

디지털화 단말장치를 기반으로 획득되는 정보 데

이터를 스마트기기와 연동시켜 서버에 실시간의 데이터베이스 구축을 목적으로 한다.

본 논문은 디지털 단말장치, 스마트기기 그리고 서버 시스템으로 구성되는 제안 시스템의 소프트웨어 설계 및 구현에 관한 것이다. 단말장치의 센서 및 모듈들에 대한 초기화 설정 그리고 동작 소프트웨어, 단말장치와 스마트기기와의 연동을 위한 장치 설정 및 연동 시나리오를 설계, 구현하였다. 또한 실시간의 획득 데이터의 저장을 위한 DB설계, 구현하였다.

II. 시스템 설계

본 장에서는 제안하는 소프트웨어 시스템 설계에 관해서 서술한다. 먼저 농지 관련 디지털 정보의 획득 단말기 구성 및 기능을 나타낸 것이 그림 2이다. 단말기의 센서 및 모듈의 종류를 나타내고 있으며 해당 기능을 나타내고 있다. 여기서 첫째, 센서(GPS, 가속도 센서, 온도도 센서)는 각각 농지의 위도 및 경도 정보를 측정하여 농지의 경계정보를 생성, 온도도 정보 측정을 위한 것이다. 둘째, SD 및 WiFi 모듈은 실시간 측정 데이터의 저장 기능과 스마트폰과의 연동 기능을 제공하기 위한 것이다.

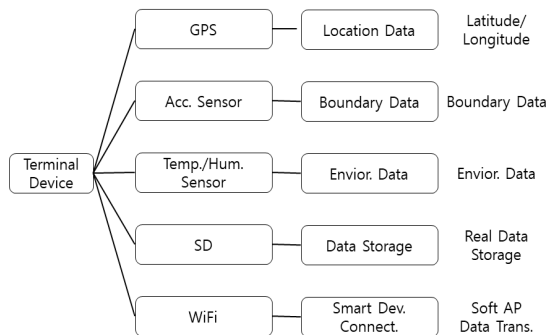


그림 2. 농지정보 획득 단말기의 구성 및 기능
Fig. 2. Structure and functions of agricultural land information acquisition terminal

그림 3은 농지 정보를 수집하는 단말기의 동작 흐름을 나타낸 것으로 단말장치, 스마트기기 그리고 서버 간의 동작 단계를 나타내었다. 그림에 나타난 것과 같이 전원이 인가되면 첫째, 센서 및 모듈의

초기화 과정 둘째, 센서 및 모듈의 실시간 정보 획득 과정 셋째, 획득 데이터 관리 과정으로 구성되며 화살표 방향은 각각의 모듈 및 센서의 단계 과정을 나타낸 것이다. 즉 정보 획득 단말기는 각 센서와 모듈의 초기화를 거쳐 정보를 수집하고, 실시간으로 저장하며, 스마트기기와 연동을 준비한다.

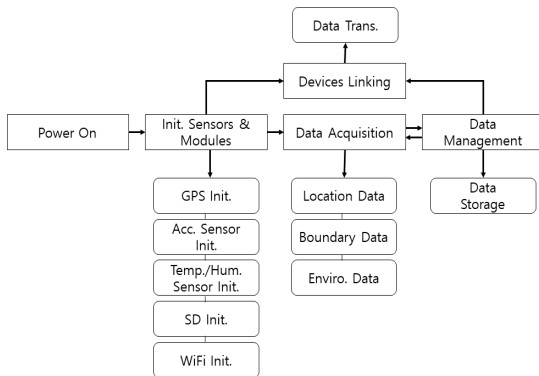


그림 3. 농지 정보 획득 시스템의 동작 흐름도

Fig. 3. Operation flowchart of agricultural land information acquisition system

2.1 위치정보 획득

GPS는 농지의 위치정보 획득 기능을 위한 것으로 농지의 위도와 경도 데이터를 실시간으로 획득하게 된다[9]. 해당 GPS 센서는 시리얼 통신을 통해 GPS 데이터를 NMEA-0183 형식으로 송신하며 해당 포맷의 데이터에서 위도와 경도를 검출하는 소스코드를 개발하였다. GPS 데이터의 GPRMC 포맷은 UTC 시간, 위치유효성(A:유효, V: 무효), 위도, 위도 표시, 경도, 경도표시, 속도, 지면 경로 각도, 날짜, 위치 정확도 정보, 신호상태의 항목으로 구성되며 이 중 위치유효성, 위도, 경도, 날짜를 추출 대상으로 설정하였다. 또한 비정상적인 위도와 경도 데이터를 최소화하기 위해 시리얼 통신 과정에서 일정 시간 동안 데이터를 연속적으로 수집하고, 위치 유효성이 'A'인 데이터만을 선별하여 정상 좌표 여부를 판단한다.

그림 4은 농지 위치정보 획득 순서도를 나타내었으며 그림 5는 GPRMC 위도와 경도 데이터를 검출을 위한 소스코드를 나타내었다.

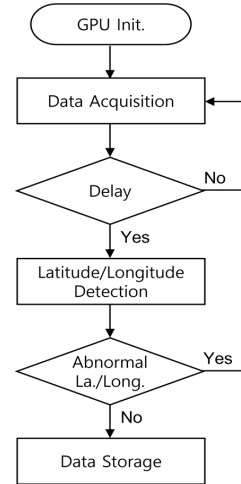


그림 4. 농지 위치정보 획득 흐름도

Fig. 4. Flowchart of agricultural land location information acquisition

```

void readGPS() {
    unsigned long startTime = millis(); // 현재 시간 기록
    while (millis() - startTime < 5000) {
        wdt_reset();
        if (GpsSerial.available()) {
            String inString = GpsSerial.readStringUntil('\n');
            int index = inString.indexOf("GPRMC");
            if(index > 0) {
                //DebugSerial.println(inString);
                if(parsingGPRMC(inString) == fail) continue;

                break;
            }
        }
    }
}

int parsingGPRMC(String sData){
    int result = -1;

    // GPRMC 구성 및 순서
    // GPRMC(0), 시간(1), 위치유효성(2), 위도(3),
    위도표시(4), 경도(5),경도표시(6), 지상속도(7),
    // 지면경로각도(8), 날짜(9), 위치정확도정보(10),
    신호상태(11), 날짜 + 시간값은 UTC

    int nCount=0;
    int nGetIndex=0;

    //임시저장
    String sTemp="";

    //원본복사
    String sCopy=sData;
  
```

```

while(true)
{
    //구분자
    nGetIndex=sCopy.indexOf(",");

    if(-1 !=nGetIndex)
    {
        sTemp=sCopy.substring(0, nGetIndex);
        if(nCount==1{

gpsTime=sTemp.substring(0,2)+"-"+sTemp.substring(2,4)+
"-:"+sTemp.substring(4,6);
        } else if(nCount==2{
            valid = sTemp;    // A or V
        } else if(nCount==3{

latitude=sTemp.substring(0,2).toFloat()+sTemp.substring(2)
.toFloat()/60.0000000;
        } else if(nCount==5{

longitude=sTemp.substring(0,3).toFloat()+sTemp.substring(
3).toFloat()/60.0000000;
        } else if(nCount==9{

gpsYearMonthDay="20"+sTemp.substring(4,6)+"-"+sTemp
p.substring(2,4)+"-"+sTemp.substring(0,2);
        }

        sCopy=sCopy.substring(nGetIndex+1);
    }
    else
    {
        break;
    }

    ++nCount;
}

if(valid.indexOf("A") > -1) result = 1;

return result;
}

```

그림 5. GNRMC 위도와 경도자료 검출을 위한 소스코드

Fig. 5. Source code for detecting lat./long. data from GNRMC

2.2 경계정보 획득

농지(노지)의 경계 정보를 실시간으로 추출하기 위하여 가속도 센서를 활용하였다. 가속도 센서의 오차를 최소화 및 최적화를 위해 초기화 시, 가속도 계와 자이로스코프, 자기계 센서에 대한 보정 과정을 수행하였다. 그림 6은 가속도 센서를 이용한 농지 경계 정보 획득을 위한 흐름도 그리고 그림 7에서는 가속도 센서의 보정 소스코드를 나타내었다.

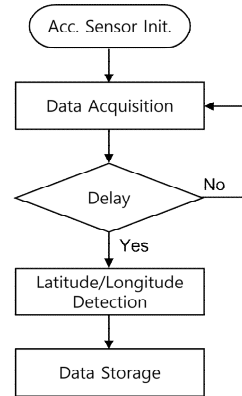


그림 6. 농지 경계 정보 획득 흐름도

Fig. 6. Flowchart of agricultural land boundary information acquisition

```

#include "MPU9250.h"
MPU9250 accelgyro;
MPU9250 mpu;
float yaw, pitch, roll;
float magX, magY, magZ;
float linearAccX, linearAccY, linearAccZ;
float accX, accY, accZ;
float gyroX, gyroY, gyroZ;

void initMpu9250() {
    // Serial.println(F("Initialization MPU9250..."));

    Wire.begin();
    delay(2000);

    if(!mpu.setup(0x68)){ //change to your own address
        while(1){
            Serial.println("MPU connection failed. Please
            check your connection~~~.");
            delay(5000);
        }
    }

    //calibrate anytime you want to
    //Serial.println("Accel Gyro calibration will start in
    5sec");
    //Serial.println("Please leave the device still on the
    flat plane.");
    mpu.verbose(true);
    delay(300);
    mpu.calibrateAccelGyro();

    //Serial.println("Mag calibration will start in 5sec");
    //Serial.println("Please Wave device in a figure eight
    until done.");
    delay(3000);
    mpu.calibrateMag();

    mpu.verbose(false);
}

```

그림 7. 가속도 센서 보정을 위한 소스코드

Fig. 7. Source code for accelerometer calibration

2.3 데이터 관리

데이터 관리는 실시간 획득 데이터의 저장 기능에 관한 것으로 SD 카드 모듈을 활용하였다. 즉 센서와 모듈로부터 실시간으로 획득한 데이터 값을 파일로 저장하도록 설계하였다. 데이터 형식은 시간, X축 가속도, Y축 가속도, Z축 가속도, 습도(%), 온도(°C)로 설계하였으며 추후 활용에 도움이 될 수 있도록 파일은 일 단위로 분리하여 저장하도록 설계하였다. 그림 8은 이를 위한 소스코드를 나타낸 것이다.

```
void writeFile(String fileName){
    File myFile=SD.open(fileName, FILE_WRITE);

    if(myFile){
        myFile.print(gpsYearMonthDay); myFile.print(" ");
        myFile.print(gpsTime); myFile.print("\t");

        myFile.print(accX); myFile.print("\t");
        myFile.print(accY); myFile.print("\t");
        myFile.print(accZ); myFile.print("\t");

        myFile.print(h); myFile.print("\t");
        myFile.print(t); myFile.print("\t");

        myFile.print(latitude,6); myFile.print("\t");
        myFile.print(longitude,6);
    }
    myFile.close();
}
```

그림 8. 실시간 데이터 저장을 위한 소스코드
Fig. 8. Source code for real-time data storage

2.4 Soft AP모드(WiFi) 구성

Soft AP(Access Point) 모드는 단말기가 공유기(AP) 역할을 수행하도록 설정하는 기능으로 시스템은 전원 인가 후 Wi-Fi 인터페이스 초기화 및 AP 모드(SSID, 암호, 채널, 최대 접속 수 등)를 설정한 뒤 Soft AP모드로 동작한다. 이후 스마트폰 등의 외부 장치는 설정된 SSID를 통해 단말기에 직접 접속할 수 있으며, 내부적으로 웹서버를 구동하여 센서 데이터 확인의 서비스 요청을 처리한다.

그림 9는 Soft AP의 초기화 및 구성 흐름도를 나타내며, 그림 10은 Soft AP 초기화 및 웹서버 등록 소스코드이다.

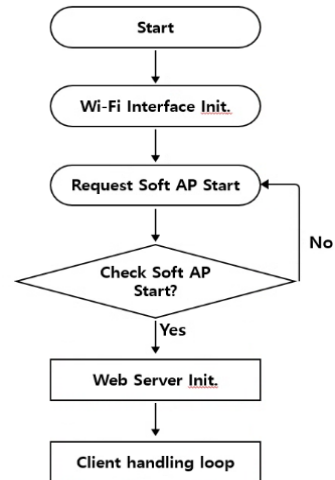


그림 9. Soft AP(Wi-Fi) 구성 흐름도
Fig. 9. Flowchart of soft AP(Wi-Fi) configuration

```
bool initSoftAP() {
    Serial.println("Soft AP 초기화 시작");
    WiFi.mode(WIFI_AP);

    bool result = WiFi.softAP(ap_ssid,
    ap_password, ap_channel, 0, ap_max_connection);

    if (result) {
        Serial.println("Soft AP 시작 성공");
        Serial.print("  AP IP: ");
        Serial.println(WiFi.softAPIP());
        Serial.print("  SSID: ");
        Serial.println(ap_ssid);
        Serial.print("  클라이언트 최대 수: ");
        Serial.println(ap_max_connection);
    } else {
        Serial.println("Soft AP 시작 실패");
    }

    return result;
}
```

```
void setupWebServerRoutes() {
    Serial.println("웹서버 라우트 등록");
    server.on("/", HTTP_GET, handleRootPage);
    server.on("/data", HTTP_GET, handleJsonData);
    server.on("/config", HTTP_POST,
    handleConfigUpdate);
    server.on("/status", HTTP_GET, handleStatus);
    server.onNotFound(handleNotFound);

    server.begin();
    Serial.println("웹서버 시작 (포트 80)");
    Serial.println("192.168.4.1 접속하세요");
}
```

그림 10. Soft AP 초기화 / 웹서버 등록 코드
Fig. 10. Source code for soft AP initi. & web server registration

2.5 스마트기기 연동

본 제안 시스템은 서버를 중심으로 제안 단말장치와 스마트기기를 연동하여 서버로의 실시간 데이터 전송을 수행하도록 설계하였다. 단말장치에 내장된 WiFi 모듈을 사용해 단말기를 AP로 설정하여 스마트기와 연동시켰다. 먼저 WiFi 초기화 시, 네트워크에 연결된 기기에 IP 주소와 기타 네트워크 설정을 자동으로 할당하는 프로토콜인 DHCP 및 AP로 동작할 수 있도록 설정하였다. 그리고 실시간 데이터 전송을 위한 UDP 통신을 이용하여 스마트기기로 요청을 처리할 수 있도록 설계하였다.

그림 11은 스마트기기 연동 흐름도를 나타내며 그림 12은 WiFi모듈 초기화 소스코드를 나타냈다. 데이터 프로토콜은 표 1과 같이 구성되며, 센서 데이터를 전송하기 위해 20바이트 고정길이 프레임を使用한다. 수신 장치는 STX_1, STX_2 패턴을 탐색해 프레임 시작을 동기화하며, Checksum을 이용해 프레임 오류 여부를 판단한다.

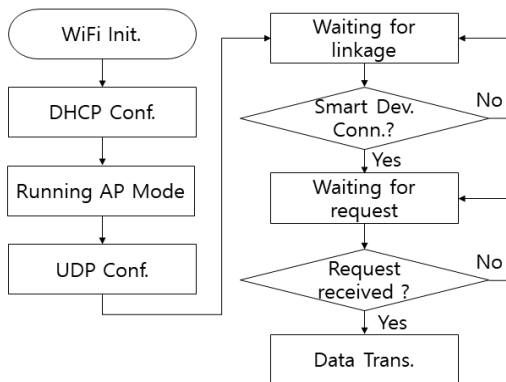


그림 11. 스마트기기 연동 흐름도

Fig. 11. Flowchart of smart device integration

```

void transmit() {
    debugSerial.listen();
    debugSerial.println("Transmit Ready.");

    byte stx[] = {0x2B, 0x5A};
    byte crc = 0x00;

    //STX
    data_[0] = stx[0];
    data_[1] = stx[1];

    //장비 ID
    data_[2] = SERIAL_ID[0];
    data_[3] = SERIAL_ID[1];
    data_[4] = SERIAL_ID[2];
    data_[5] = SERIAL_ID[3];
    data_[6] = SERIAL_ID[4];
    data_[7] = SERIAL_ID[5];

    //Checksum
    for (int i = 0; i < (transmitSize - 2); i++) {
        crc += data_[i];
    }

    data_[transmitSize - 2] = crc;

    softSerial.write(data_, sizeof(data_));
}
  
```

```

void read() {
    int count = 0;

    while(LoraSerial.available() > 0 && ++count <
receivedBufferSize) {
        byte newData = LoraSerial.read();
        receivedLoRaData[dataSaveIndex++] = newData;

        if (dataSaveIndex >= receivedBufferSize) {
            dataSaveIndex = dataSaveIndex -
receivedBufferSize;
            processLoRaData();
            break;
        }
    }
}
  
```

그림 12. 데이터 송신부 / 수신부 소스코드

Fig. 12. Source code for data transmitter / receiver

표 1. 데이터 프로토콜

Table. 1. Data protocol

	STX 1	STX 2	Device ID						Reserved	
byte	00	01	02	03	04	05	06	07	08	09
Data	0x28	0x5A	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x0C
	Temperature		Humidity		Reserved			Sequence ID (SID)	Checksum	#r
byte	10	11	12	13	14	15	16	17	18	19
Data	0x00	0x70	0x01	0x6C	0x00	0x00	0x00	0x00	0x42	0x0D

III. 모의실험

시스템의 동작 및 자료 획득을 위한 모의실험을 제안 시스템을 기반으로 수행하였다.

샘플 3대를 12시간 연속해서 다양한 온도 및 습도 환경 그리고 노지에서 데이터 획득 및 신뢰성 확인하는 과정을 분리하여 수행하였다.

시험 항목으로

- GPS, 로그 데이터의 확인
- 온도 범위, 10°C ~ 55°C 범위에서
12시간 이상 동작 확인
- 데이터 수집, 로그 데이터의 확인
- 샘플 3대, 온도, 습도 센서의 데이터 획득 확인
- 온도 설정, -10°C/10°C/20°C/30°C/50°C/55°C
- 습도 설정, 30%/50%/80%

를 설정하고 실험 과정을 반복하여 데이터 수집 신뢰성 확인(로그데이터 검증, 통신신뢰도 추정, 현장 테스트 2회)을 수행하였다.

그림 13는 서버 상에 실시간으로 저장된 데이터를 나타낸 것이다. 각 행은 특정 시점의 측정값을 나타내며 데이터는 일자(시간) X·Y·Z축 가속도, 습도, 온도, 위도, 경도 순으로 구성되어 있다.

06-26 05:06:54	-1.83	3.17	-0.49	52.00	23.10	36.630847	127.330337
06-26 05:06:59	-2.14	2.75	-0.98	52.10	23.10	36.630847	127.330337
06-26 05:07:04	-1.71	3.23	0.12	52.10	23.20	36.630847	127.330337
06-26 05:07:09	-2.01	3.11	-0.31	52.10	23.20	36.630847	127.330337
06-26 05:10:31	-1.89	2.93	-0.24	54.90	23.60	36.630439	127.330184
06-26 05:10:36	-2.01	3.11	-1.28	54.80	23.60	36.630420	127.330230
06-26 05:10:41	-1.89	2.99	-0.31	54.70	23.60	36.630401	127.330238
06-26 05:10:46	-1.77	2.87	-0.55	54.60	23.60	36.630420	127.330223
06-26 05:10:51	-1.65	2.99	-0.31	54.70	23.60	36.630378	127.330245
06-26 05:10:56	-1.83	3.05	-0.55	54.60	23.60	36.630405	127.330238
06-26 05:11:01	-1.89	3.11	-0.55	54.60	23.60	36.630355	127.330276

그림 13. 수집된 데이터의 실시간 저장
Fig. 13. Real-time storage of collected data

이와 더불어, 그림 14과 같이 제안 시스템의 무선 통신 기능을 검증하기 위해 Soft AP를 구성하여 웹서버를 구동하고 외부 단말의 접속 여부를 확인하는 시험을 추가로 수행하였다. 해당 시험에서는 단말장치가 Soft AP 모드로 동작하면서 설정된 SSID와 보안 키를 통해 스마트폰 및 노트북 등의 외부 단말이 접속 가능 여부를 검증하였다.

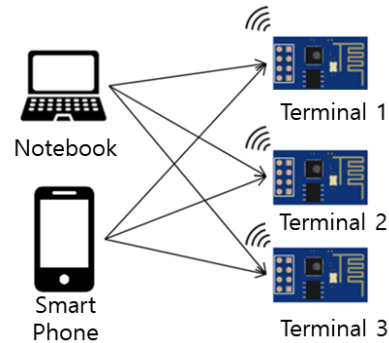


그림 14. 무선 통신 시험 구성
Fig. 14. Wireless communication test configuration

IV. 결 론

본 논문에서는 정밀농업 관련 농지정보(지리정보-위치정보(경도, 위도), 생육 환경정보-온도, 습도)의 디지털화를 위한 장치를 기반으로 하는 농지 정보 획득 시스템의 소프트웨어를 설계, 구현하였다. 단말장치, 스마트기기, 서버로 구성된 농지 정보 획득 시스템의 하드웨어 초기화, 운영 프로토콜, 데이터 저장 DB를 위한 소프트웨어를 설계, 구현하였으며 모의실험 결과(1차 35분, 2차 51분, 10초 간격으로 데이터 송수신 실험, 통신성공률, 98.07% 실거리오차, 2.5CEP), 획득 정보의 관리를 위한 DB를 기반으로 목적하는 농지 정보를 효과적으로 실시간 획득이 가능하였다. 이를 활용하여 농지 관련 응용 분야(농지 면적 계산, 영농시설 설계, 생육 환경의 개선 등) 및 농산업의 경쟁력 확보를 위한 기반 기술을 얻을 수 있었다.

추가적으로 제안 시스템의 제품인증 과정 및 국가 공공망과의 연동을 수행하여 공공기관의 활용성을 확보하는 것이 연구과제로 남아 있다.

References

- [1] M. H. Choi and J. Y. Park, "Trends of Technologies and Standards for Digital Agriculture and Future Standardization Initiative", Precision Agriculture Science and Technology, Vol. 4, No. 4, pp. 171-193, Dec. 2022. <https://doi.org/10.12972/pastj.20220015>.

- [2] U. Yeo, I. Lee, K. Kwon, T. Ha, S. Park, R. Kim, and S. Lee, "Analysis of Research Trend and Core Technologies Based on ICT to Materialize Smart-farm", Protected Horticulture and Plant Factory, Vol. 25, No. 1, pp. 30-41, Mar. 2016. <http://dx.doi.org/10.12791/KSBEC.2016.25.1.30>.
- [3] Y. Kwak, J. Lee, M. Cho, H. Kim, H. Shin, and S. Song, "A Study on the Quality Measurement and Environment Monitoring for Farm Products based on Sensor Network", Journal of KIIT, Vol. 13, No. 11, pp. 159-164, Dec. 2013. <http://dx.doi.org/10.14801/kiitr.2013.11.3.159>.
- [4] S. W. Kim, "Design and Implementation of Smartfarm Integrated Platform", The Journal of Korean Institute of Communications and Information Sciences, Vol. 46 No. 12, pp. 2403-2410, Dec. 2021. <https://doi.org/10.7840/kics.2021.46.12.2403>.
- [5] Y. Kim and E. Kang, "Smart Multi-Sensor and Environment Monitoring System for Agriculture Growth Management", Journal of The Korea Contents Association, Vol. 17 No. 1, pp. 138-147, Dec. 2017. <https://doi.org/10.5392/JKCA.2017.17.12.138>.
- [6] J.-Y. Lee and B.-M. Kim, "Growth Environment Monitoring System for Open-Field Smart Farms Using Big Data Model", Journal of JKIIICE, Vol. 28, No. 1 pp. 47-53, Jan. 2024. <http://doi.org/10.6109/jkiice.2024.28.1.47>.
- [7] S.-S. Wee, W.-S. Lee, and N.-S. Jung, "Farm-map Application Strategy for Agri-Environmental Resources Management", Journal of the Korean Society of Agricultural Engineers, Vol. 64, No. 3, pp. 1-8, May 2022. <https://doi.org/10.5389/KSAE.2022.64.3.001>.
- [8] F. Karim, F. Karim, and A. frihida, "Monitoring system using web of things in precision agriculture", Procedia Computer Science, Vol. 110, pp. 402-409, 2017. <https://doi.org/10.1016/j.procs.2017.06.083>.

- [9] Datasheet, PRF-G191 GPS G-Mouse Specifications, <https://www.devicemart.co.kr>. [accessed: Apr. 18, 2026]

저자소개

이 병 주 (Byungju Lee)



2004년 8월 : 충북대학교
학과(공학석사)
2025년 2월 ~ 2006년 6월 :
삼성전자 주임연구원
2026년 7월 ~ 2009년 10월 :
충북대학교 산학협력단
선임연구원

2010년 9월 ~ 현재 : (주)보아스에스이 대표이사
관심분야 : 모니터링시스템

김 종 원 (JongWon Kim)



1994년 2월 : 청주대학교 대학원
전자공학과(공학석사)
2021년 2월 : 안양대학교
컴퓨터공학과(박사수료)
2023년 3월 ~ 현재 :
국립한국교통대학교
컴퓨터공학과 강사

관심분야 : 무인기 비행제어시스템, ASIC, 임베디드
시스템

곽 윤 식 (Yoonsik Kwak)



1986년 9월 : 경희대학교
전자공학과(공학석사)
1994년 3월 : 경희대학교
전자공학과(공학박사)
1991년 5월 ~ 현재 :
국립한국교통대학교
컴퓨터공학과 교수

관심분야 : 센서네트워크, 인터넷 통신, 행정시스템