

Transponder and Ground Station Systems for Drones

Ki-Su Kim*, Heon-Seong Ha*, Jong-Chan Lee*

*Student, School of Computer Information Engineering, Kunsan University, Kunsan, Korea

*Student, School of Computer Information Engineering, Kunsan University, Kunsan, Korea

*Professor, Dept. of Computer Information Engineering, Kunsan University, Kunsan, Korea

[Abstract]

In this paper, we propose a case that drone (unmanned aerial vehicle), one of the representative technologies of the 4th Industrial Revolution, threatens airport safety and privacy infringement, and describes a drone control system proposal to solve the problem. Unmanned aerial vehicle (Drone) is creating a serious problem recently, In Korea, on May 21, 19, according to the Jeju Regional Aviation Administration, drones flew over Jeju Jeongseok Airfield twice in the same month, causing problems in aircraft operation. In overseas cases, two drones near the runway of Gatwick International Airport in the UK There has been a disturbance in which the takeoff and landing of the aircraft flies for a while, and various problems have occurred, such as voyeuring the private life of an individual using a drone. This paper is equipped with an Access Point transponder mounted on a drone (unmanned aerial vehicle), and unspecified many who want to receive flight information (coordinates, altitude, and obstacles) of the drone access the drone AP, receive and receive the flight information of the drone, and receive unspecified multiple Drone AP flight information is collected and collected to provide the information of the drone currently floating on one user interface screen. In addition, an AP transponder is proposed to operate a safe drone (unmanned aerial vehicle) and the drone's flight information is transmitted. , To receive and collect and collect data.

▶ **Key words:** ESP8266, WIFI TRANSPONDER, Drone System, Ap Mode Drone, Ground Station for Drone

[요 약]

본 논문에서는 4차 산업혁명의 대표 기술 중 하나인 드론(무인항공)이 공항 안전 및 사생활 침해를 위협하는 사례를 제시 하고 문제를 해결하기 위해 드론 관제 시스템 제안을 기술하였다. 최근 (드론) 무인항공이 심각한 문제를 만들고 있다. 국내의 경우 19년 5월 21일 제주지방항공청에 따르면 같은 달 두 차례에 걸쳐 제주 정석비행장 상공에 드론이 비행하여 항공기 운행에 문제가 발생했고 해외 사례의 경우 영국 개트윅 국제공항 활주로 부근에 출현한 드론 때문에 항공기의 이착륙이 잠시 멈추는 소동이 발생하였으며, 또한 드론을 활용하여 개인의 사생활을 도촬 하는 등 여러 문제가 발생되고 있다. 본 논문은 드론(무인항공)에 Access Point(이하 AP) 트랜스폰더를 장착하여 드론의 비행정보(좌표, 고도, 장애 상태)를 원하는 불특정 다수가 드론 AP에 접근하여 드론의 비행정보를 수신 받고, 수신 받은 불특정 다수의 드론 AP 비행정보를 수집, 취합하여 유저 인터페이스 화면에 현 시점에 운행 중인 드론의 정보를 제공한다. 안전한 드론(무인항공)을 운행하기 위해 AP 트랜스폰더를 제안하고 드론의 비행 정보 송, 수신 및 데이터 수집 및 취합 실험을 수행한다[1-3].

▶ **주제어:** ESP8266 프로세서, 무선 트랜스폰더, 드론 시스템, AP모드 드론, 드론 지상국

- First Author: Ki-Su Kim, Corresponding Author: Jong-Chan Lee
- *Ki-Su Kim (vennomnight1@kunsan.ac.kr), School of Computer Information Engineering, Kunsan University
- *Heon-Seong Ha (ei91081@kunsan.ac.kr), School of Computer Information Engineering, Kunsan University
- *Jong-Chan Lee (chan2000@kunsan.ac.kr), Dept. of Computer Information Engineering, Kunsan University
- Received: 2020. 04. 09, Revised: 2020. 05. 18, Accepted: 2020. 05. 18.

I. Introduction

항공기 항법 시스템 중 하나인 트랜스폰더는 지상국에서 송출된 전파가 공중의 비행체에 부딪쳐 되돌아온 반사파(echo)를 레이더 화면에 시각 정보로 전환하는 전자 장비이다. 트랜스폰더가 적절하게 장착된 항공기는 고도정보와 위치정보를 송신하고, 또한 트랜스폰더는 4개의 숫자 코드 조합을 통하여 현재 비행기의 상태를 송신한다. 코드 7700(Emergency), 7600(Lost Communication), 7500(Hijack) 와 같이 조합하여 사용된다[4]. 4차 산업혁명의 대표 기술 중 하나인 드론(무인항공; 이하 드론)에 ACEESS POINT 트랜스폰더[4] 장치를 장착하여 드론의 위치정보와 고도정보 그리고 장애정보 데이터를 송신하여 안전한 드론 비행을 하고자 한다. 트랜스폰더 장치는 GPS 센서를 통하여 고도정보와 위치정보 데이터를 생성하고 WiFi 장치 모듈을 ACEESS POINT 모드로 하여 생성된 데이터를 사용자의 앱을 통하여 드론의 비행정보를 수신 받을 수 있도록 하였다. 본 논문에서는 드론 트랜스폰더 송신장치로 ESP8266 MCU를 사용하고 표준 GPS프로토콜을 수신하는 모듈을 통하여 GPS 센서 정보를 ESP8266 MCU WiFi ACEESS POINT모드를 통하여 데이터를 송신한다. 모바일 앱은 드론으로부터 수신한 위치 및 고도, 장애 상태 정보를 서버로 송신하고, 서버는 수신한 드론 정보를 통합하여 다수의 사용자들에게 제공한다. 본 연구에서는 사용자가 모바일 앱의 지도 UI를 통하여 주변 드론의 위치 및 고도, 장애 상태 정보를 확인할 수 있는 TCP/IP 지상국 서비스 시스템을 구현 하고자 한다. 2장에서 트랜스폰더와 GPS 모듈등 관련 기술을 소개하고, 3장에서 제안된 방법 기술과 실제 실험을 통하여 드론과 서버를 연동하고 모바일로 사용자에게 제공하는 구성을 직접 구현한다. 마지막으로 4장에서 본 논문에 대한 결론을 도출한다.

II. Preliminaries

1. Related works

1.1 ESP8266 WiFi Soc(System on Chip)

ESP8266 모듈은 WiFi 네트워킹을 지원하는 통신 모듈이며, SDK개발환경이 오픈소스로 제공된다. 물리적으로 ESP8266 프로세서의 스펙은 Table 1과 같다.

Table 1. ESP8266 Environment

ESP8266 SPEC	
Voltage	3.3V
Current consumption	10uA - 170mA
Flash memory attachable	16MB max (512K normal)
Processor	Tensilica L106 32 bit
Processor speed	80-160MHz
RAM	32K + 80K
GPIOs	17 (multiplexed with other functions)
ADC	1 input with 1024 step resolution
802.11 support	b/g/n/d/e/i/k/r
Maximum concurrent TCP connections	5

ESP8266 모듈은 3.3V 동작 전압을 가진다. 많은 수의 ESP-xx 모듈이 3.3V에서만 동작하므로 3.3V 전원이 필요하다. UART 통신 라인도 3.3V로 동작하므로 5V 동작전압의 보드들과 연동하려면 Level Converter 모듈을 이용하거나 저항을 이용해서 전압 분배 회로를 구성한다.

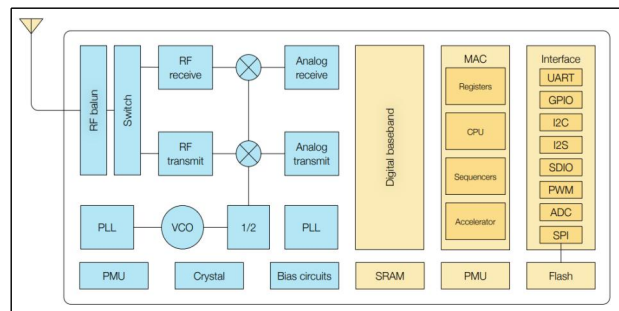


Fig. 1. Functional Block Diagram

Fig.1은 ESP8266의 시스템 다이어그램이다. 인터페이스 블록에 UART, GPIO, I2C, I2S, SDIO, PWM, ADC, SPI 주변장치 모듈이 구성되므로 하드웨어 제어를 유연하게 할 수 있도록 개발 되었다.

1.2 UART GPS NEO-6M

NEO-6M 모듈은 위성에서 수신 받은 데이터를 UART 비동기통신으로 송신하는 모듈이다. UART(Universal Asynchronous Receiver/Transmitter; 이하 UART) 통신은 디지털 데이터의 통신 규격이고 1:1통신만 지원한다. 기본 1바이트 단위로 반복 전송하는 형식이다. NEO-6M의 UART통신 속도는 9600Bps를 사용한다.

13	12	1.Reserved
14	11	2.SS_N
15	10	3.TIMEPULSE
16	9	4.EXTINTO
17	8	5.USB_DM
	7	6.USB_DP
	6	7.VDDUSB
	5	8.Reserved
	4	9.VCC_RF
	3	10.GND
18	2	11.RF_IN
19	1	12.GND
20		13.GND
21		14.MOSI/CFG_COM0
22		15.MISO/CFG_COM1
23		16.CFG_GPS0/SCK
24		17.Reserved
		18.SDA2
		19.SCL2
		20.TxD1
		21.RxD1

Fig. 2. NEO-6M Pin Definition

Fig.2는 NEO-6M모듈의 핀의 기능을 나타낸다[5].

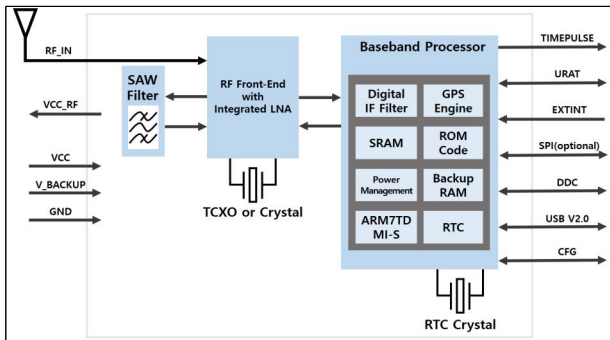


Fig. 3. Block Diagram

Fig.3의 구성도와 같이 UART 인터페이스를 제공하고 옵션으로 SPI통신 USB인터페이스가 제공된다[5].

Table 2. Protocols and interfaces

Prorocol	Type
NMEA	Input/output, ASCII, 0183, 2.3 (compatible to 3.0)
UBX	Input/output, binary, u-blox proprietary
RTCM	Input, 2.3

NEO-6M모듈은 Table 2와 같이 NMEA, UBX, RTCM 프로토콜을 지원한다[5]. 본 논문에서는 NMEA프로토콜을 사용하여 GPS값을 수신한다.

CFG_COM1	CFG_COM0	Protocol	Messages
1	1	NMEA	GSV, RMC, GSA, GGA, GLL, VTG, TXT
1	0	NMEA	GSV, RMC, GSA, GGA, GLL, VTG, TXT
0	1	NMEA	GSV, RMC, GSA, GGA, VTG, TXT
0	0	UBX	NAV-SOL, NAV-STATUS, NAV-SVINFO, NAV-CLOCK, INF, MON-EXCEPT, AID-ALPSERV

Fig. 4. Protocol Messages

CFG_COM1(14번 핀), CFG_COM0(15번 핀)을 HIGH시 그널로 입력하여 NMEA 프로토콜을 사용한다. 수신한 Messages는 GSV, RMC, GSA, GGA, GLL, VTG, TXT를 수신한다[5].

Table 3. Data Received From NEO-6M

\$GPGGA,061117.00,3556.67680,N,12640.98189,E,1,07,1.48,11.3,M,18.8,M,,*6E
\$GPGSA,A,3,01,11,27,08,04,09,16,,,,,2.45,1.48,1.96*04
\$GPGSV,4,1,14,01,17,186,26,04,13,215,27,07,40,309,,08,75,309,36*7C
\$GPGSV,4,2,14,09,18,248,30,10,02,084,,11,54,199,35,16,38,079,39*76
\$GPGSV,4,3,14,20,02,055,,23,13,223,33,26,15,099,25,27,55,035,38*74
\$GPGSV,4,4,14,30,10,318,,50,44,150,31*77
\$GPGLL,3556.67680,N,12640.98189,E,061117.00,A,A*63
\$GPRMC,061118.00,A,3556.67670,N,12640.98184,E,0.444,,010420,,A*74
\$GPVTG,,T,,M,0.444,N,0.823,K,A*2E
\$GPGGA,061118.00,3556.67670,N,12640.98184,E,1,07,1.48,10.8,M,18.8,M,,*69

Table 3은 NEO-6M으로부터 수신받은 GPS데이터이다. 데이터는 “,”로 분류한다[6]. GPGGA(Global Positioning System Fix Data) 데이터의 시작은 061117.00 시간으로 Zulu time(그니리치 표준시) 기준으로 06시11분 17.00초를 뜻한다. 3556.67680은 위도로서 35도 56.67680분을 뜻한다. 도(degree) 단위로 환산 시, 6.67680 / 60 = 0.944613, 대략 35.944613도가 된다. N은 북위를 뜻한다. 적도 남쪽에 있다면 S가 표시된다. 12640.98189는 경도로서 126도 40.98189분을 뜻한다. 도(degree) 단위로 환산 시, 40.98189 / 60 = 0.683031 대략 126.683031도가 된다. E는 동경을 뜻한다. W되면 서경이 된다. ‘1’은 Fix의 종류를 뜻한다.

1. 0: Invalid 잘못된 데이터. 주로 위성이 안 잡힐 경우.
- 1: GPS에서 제공하는 기본 위성으로 계산할 경우.
- 2: DGPS를 이용하여 보정하여 계산할 경우.

‘48’은 계산에 사용한 위성의 개수를 표시한다. 11.3M 해수면 기준의 고도이다. 18.8M은 WGS-84에서 정해놓은 타원체로서 모델링한 지구와 구체로서 모델링된 지구의 고도 차이를 뜻한다. 마지막으로 6E는 Check Sum을 나타낸다. “GPGGA”에 위치정보, 고도정보를 포함하고 있으므로 본 논문에는 “GPGGA”를 사용한다[7].

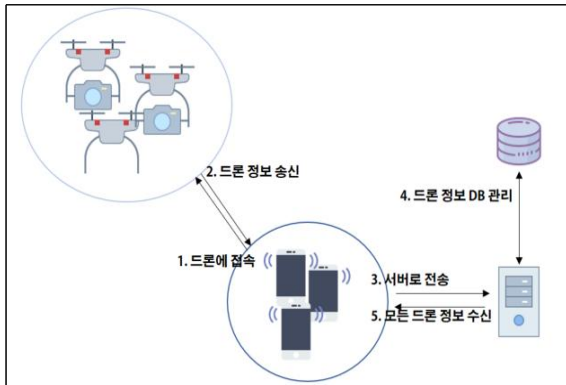


Fig. 5. Ground Station System Architecture

WiFi 모듈을 부착한 드론은 AP역할을 수행하므로 모바일 기기에서 해당 드론으로 접속할 수 있다. 사용자가 드론 AP에 접속하면, 모바일 앱은 HTTP 프로토콜을 통하여 드론으로부터 위치 및 Squawk Code[4]등의 정보를 수신한다. 드론 정보를 수신한 모바일 앱은 여기에 드론의 식별 ID를 추가하여 HTTP프로토콜을 사용하여 수신한 정보를 서버에 전달한다. 서버는 수신한 정보를 식별자, GPGGA 프로토콜[7]을 파싱하여 위도와 경도, 고도 수신 받은 드론의 정보들을 통합하여 최종 업데이트 시간과 함께 DB에 저장 한다. DB에는 이와 같은 방식으로 여러 드론의 정보를 저장하고, 통합된 드론들의 정보는 주변 드론에 위치정보 및 상태정보를 제공한다.

III. The Proposed Scheme

ESP8266의 개발 환경은 Arduino IDE를 사용하여 개발 실험 하였다. Arduino IDE에서 ESP8266칩을 사용하기 위해서는 보드매니저에서 ESP8266보드를 설치해야 한다.

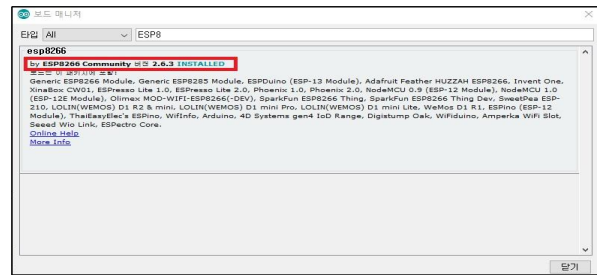


Fig. 6. Board Manager ESP8266

```

ESP8266WebServer server(80);
SoftwareSerial ss(Rx,Tx);
void setup() {
    Serial.begin(115200);
    WiFi.softAP(ssid, password);
    IPAddress myIP = WiFi.softAPIP();
    server.on("/", handleRoot);
    server.begin();
    ss.begin(9600);
}
    
```

Fig. 7. Initialize Code

Fig. 7은 ESP8266 WIFI AP모드 및 UART통신 초기화 코드이다[5]. server(80) 초기화를 통하여 포트 80번으로 HTTP프로토콜서버 초기화를 실시하고, ss(Rx,Tx) UART 9600bps로 통신 속도를 설정하여 GPS모듈과 통신 가능하도록 초기화한다[7].

WiFi.softAP(ssid,password)의 파라미터는 ssid="DRONE_AP", password="123456789"로 AP의 SSID, PW를 초기화 하였고, server.on 메소드에 함수 handleRoot를 인자로 넘겨 server가 수행될 시 인자로 넘어온 함수가 수행되어지도록 초기화 작업을 수행한다. 마지막으로 WiFi.softAPIP() 함수를 호출하여 최종 AP모드로 초기화한다.



Fig. 8. Able to Connect to AP Mode Drone

초기화 코드를 만들고 컴파일 후 업로드를 하면 Fig. 8과 같이 WiFi SSID DRONE_AP에 접속이 가능하다. 본 논문의 실험 환경은 안드로이드 버전 10으로 진행되었다.

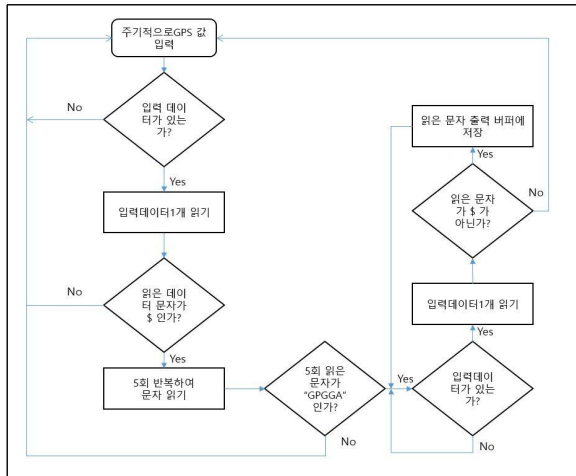


Fig. 9. Logic FlowChart

Fig. 9은 GPGGA 데이터를 ESP8266에서 파싱하는 로직 순서도이다. 먼저 GPS에서 데이터가 입력되면 ESP8266내부 메모리에 저장된다. GPS 값 입력 로직이 ESP8266내부 메모리에 데이터가 입력이 되었는지 확인한다. 첫 번째 입력 데이터를 읽어 “\$” 문자인지 확인한다. “\$”이 문자는 NMEA 프로토콜의 시작을 알린다. 그 후 “GPGGA”의 데이터의 수신을 확인하기 위하여 5번의 입력 데이터가 “GPGGA”인지 확인한 후 텍스트의 시작인 “\$”문자가 입력 될 때 까지 데이터를 입력 받아 AP모드에서 사용하는 출력버퍼로 “GPGGA”데이터를 이동한다.

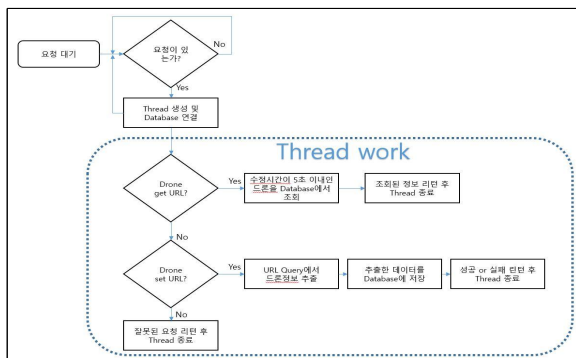


Fig. 10. Server FlowChart

Fig. 10에서 서버는 HTTP 서비스를 제공하고 클라이언트는 URL로 서버에 요청 한다[8]. 서버는 요청이 들어오면 쓰레드를 생성하고 메인 쓰레드는 계속 요청에 응답하기 위하여 대기한다[9]. 생성된 쓰레드는 접속한 URL이 드론정보를 가져오는 요청일 경우는 DB를 조회하여 최신 수정된 시간이 5초 이내인 정보들만 모아서 응답해주고 쓰레드를 종료한다. 드론의 정보(위치, 고도정보, 상태정보 이하 정보라 칭함)가 DB에 저장되어 있어도 최신 수정된 드론정보가 아닌 경우

드론이 현재 운행 중이지 않다고 판단하여 응답에서 제외된다. 접속한 URL이 드론정보를 입력하는 URL일 경우 URL에 포함된 드론정보 중에서 위치 및 고도, 상태 정보를 DB에 저장하고, 성공여부를 응답해준 후 쓰레드를 종료한다.

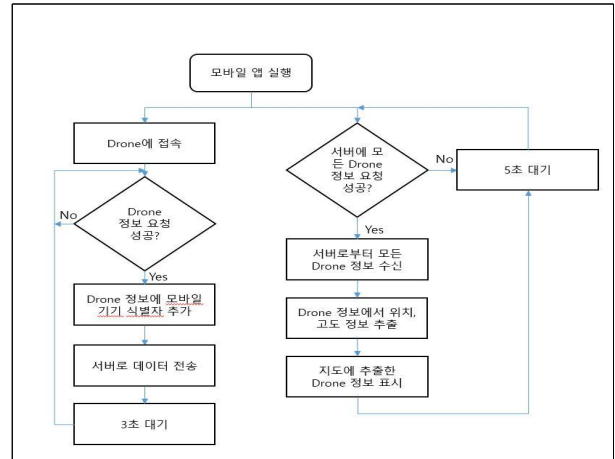


Fig. 11. Android App FlowChart

Fig. 11과 같이 모바일 앱이 실행되면 두개의 쓰레드가 생성되어 처리된다[10]. 첫 번째 쓰레드는 드론으로부터 정보를 수신한 쓰레드이다. WiFi 프로토콜로 드론에 접속한 후 드론으로부터 드론의 위치, 고도, 상태의 정보를 3초마다 요청하고 수신한 드론정보 데이터를 서버로 전송한다. 두 번째 쓰레드는 5초마다 서버로부터 연결되어 있는 모든 드론들의 정보를 수신한다. 수신한 드론들의 위치, 고도, 상태정보를 모바일 앱 지도 화면에 표시한다[11].

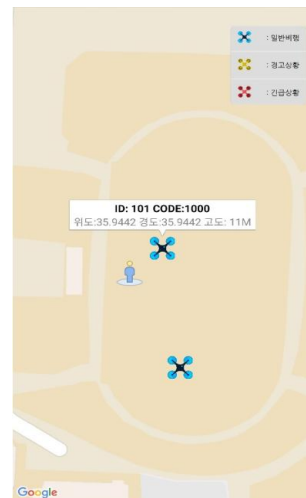


Fig. 12. Drone location on Google Maps

Fig.12과 같이 안드로이드 앱 상에 연결되어 있는 드론의 위치와 고도, 드론의 상태를 지도 위에 표시한다.



Fig. 13. Emergency state of the drone on Google Maps

Fig. 13은 드론의 현 상태를 관제하기 위하여, 드론 고도정보와 드론의 모터 혹은 프로펠러 고장 등으로 기기 결함이 발생하였을 경우 드론 상태 코드 7700(Emergency) 긴급 상황을 사용자의 모바일 앱에 전송하고, 모바일 앱은 현재의 상태를 Fig. 13 같이 지도에 표시한다. Fig. 14는 드론의 상태 코드 7600(Communication failure) 통신두절 상태를 표시한다.

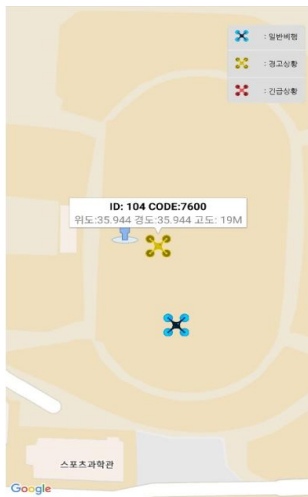


Fig. 14. Communication failure state of the drone on Google Maps

IV. Conclusions

4차 산업혁명 기술 중 하나인 드론의 안전 비행을 위한 연구로서, 실제 비행기에서 사용되는 트랜스폰더장치를 드론에 장착하고 이를 사용하는 방법을 제시하였다. 현재의

위치 정보 및 고도, 상태를 서버에서 수집하고 드론들의 정보를 통합하여 앱을 으로 드론의 정보를 실시간 확인 가능한 시스템을 구성하고 실험을 통하여 결과를 확인하였다. 이를 통하여, 주변 드론의 관제를 가능하게 함으로서 드론으로 인한 피해를 줄이고 악용을 예방하는 드론 안전 비행 기초 시스템을 구성하였다. 앞으로 드론의 항적 표시, heading, 속도 정보를 추가하여 heading과 항적, 속도 데이터를 활용함으로써 드론 방향을 예측하고 추락 지점의 예측 로직을 추가함으로써 안전한 드론 비행을 위한 연구를 지속적으로 진행할 계획이다.

ACKNOWLEDGEMENT

This research was supported by X-mind Corps program of National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (No. 2019H1D8A110981411).

REFERENCES

- [1] Yonhapnews, <https://m.post.naver.com/viewer/postView.nhn?volumeNo=8963968&memberNo=481955>
- [2] NEWSIS, <http://https://http://www.donga.com/news/article/all/20190521/95619915/1>
- [3] NEWSIS, <https://news.joins.com/article/23229976>
- [4] Ministry of Land, Infrastructure and Transport "Air Navigation," pp.50-108, 2018.
- [5] Espressif Systems, https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- [6] UBLOX, <https://www.waveshare.com/w/upload/2/2c/NEO-6-DataSheet.pdf>
- [7] Wikipedia, <https://ko.wikipedia.org/wiki/NMEA>
- [8] Yong-Tae Kim, "A Design of Web Server Architecture Environment for Reliability Enhancement and Secure Web Services", The journal of the Korea Institute of Maritime Information & Communication Sciences, Vol. 14 No. 2, pp. 343-350, Feb. 2010. DOI: 10.6109/jkiice.2010.14.2.343
- [9] Dongmin Kim, "OpenFx = An Open Source Serverless Platform by API Call Unit Applications Supported Fast Response : OpenFx", KIISE Transactions on Computing Practices (KTCP), Vol. 25 No. 11, pp. 543-551, Nov. 2019. DOI: 10.5626/KTCP.2019.25.11.543
- [10] ByeongDo Kang, "A Software Development Process for Mobile

Applications", Journal of Internet Computing and Services, Vol. 15 No. 4, pp. 135-140, Apr. 2014. DOI: 10.7472/jksii.2014.15.4.135

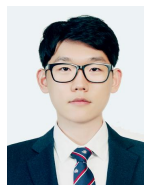
- [11] MinJeong Koo, "Design and Implementation of the Specialized Business-Work Management System(WMS) Using Mobile App and GPS", Journal of Digital Convergence, Vol. 11 No. 8, pp. 359-365, Aug. 2013. DOI: 10.14400/JDPM.2013.11.8.359

Authors



Ki-Su Kim received the B.S., M.S. degrees in Computer Information Engineering from Kunsan University, Korea, in 2016, 2017, respectively. Ms. Kim is currently a Ph.d Course in the Department of Computer

Information Engineering, Kunsan University. He is interested in Embedded Systems, PCB Design and Machine Learning, and Control System.



Heon-Seong Ha received the B.S. degrees in Computer Information Engineering from Kunsan University, Korea, in 2018 respectively. Ms. Ha is currently a M.S. Course in the Department of Computer

Information Engineering, Kunsan University. He is interested in Web Programming, Block Chain System



Jong-Chan Lee received the M.S. and Ph.D. degrees in computer science and engineering from Soongsil University, Korea, in 1996 and 2000 respectively. Dr. Lee was a senior member of engineering staff in Mobile

Telecommunication Research Laboratory, Electronics and Telecommunications Research Institute (ETRI) from 2000 to 2005. Since 2005, he has worked in the Department of Computer Information Engineering, Kunsan National University as an professor. His current research interests are in the areas of data analysis and deep learning