

# PIXHAWK와 DRONEKIT을 활용한 인명 구조 시스템 설계 및 구축

현만석\*, 최광훈\*\*, 김재훈\*\*

아주대학교 소프트웨어학과\*

아주대학교 컴퓨터공학과\*\*

e-mail:bws2958, saarc, jaikim@ajou.ac.kr

## Lifesaving System Construction using Pixhawk and Drone-Kit

Manseok Hyun\*, Kwanghoon Choi\*\*, Jai-Hoon Kim\*\*

\*Department of Software and Computer Engineering, Ajou University

\*\*Department of Computer Engineering, Ajou University

### 요 약

재난지역의 영상 촬영 및 물품 조달 등 인간이 직접 수행하기 어려운 임무를 대신하기 위해 드론이 사용된다. 시중에 판매되고 있는 드론들은 비행과 영상촬영 기능을 모두 제공하지만 firmware 및 응용프로그램 소스코드가 제한적으로 공개되어 있어 일반 개발자가 드론의 firmware를 수정하거나 센서 및 카메라를 직접 제어할 수 없다는 단점이 있다. 하지만 개발자의 목적에 맞는 드론의 비행 시스템을 구축하기 위해서는 시스템 내부의 제어코드를 직접 수정할 필요성이 있다. Pixhawk는 firmware 및 관련 application의 개발에 대한 소스 코드가 공개되어 있고, 일반 개발자들의 접근이 상대적으로 용이하다는 장점이 있다. 본 프로젝트에서는 Pixhawk와 3D Robotics 사에서 제공하는 Drone-Kit Platform을 활용하여 조난 상황에 대한 인명 구조 시스템을 설계하고, 응급상황에 대한 드론 제어 시스템을 구현하였다.

Key Words : Pixhawk(픽스호크), Drone-Kit(드론키트), Raspberry pi(라즈베리파이)

### I. 서론

재난지역의 영상 촬영 및 물품 조달 등 인간이 직접 수행하기 어려운 임무를 대신하기 위해 드론이 사용될 수 있다. 시중에 판매되고 있는 DJI, Parrot 사의 상용 드론들은 조종기를 통해 사람이 직접 드론의 비행을 제어하거나 제작사에서 배포하는 어플리케이션에 드론의 이동경로를 설정해주면, 입력된 지역의 GPS 정보를 바탕으로 드론 스스로 자율주행을 하며 영상 촬영이 가능하다. 하지만 상업용 드론의 경우, 영상촬영은 가능하지만 드론의 MCU 내에서 영상처리에 제약이 있으며, 개발자가 특수 상황에서 드론이 스스로 모션을 제어하도록 명령 내리는 데에도 어려움이 있다. 이러한 개발상의 제약을 극복하고, 특정 상황에서 드론을 보다 효율적으로 제어하는 방법을 모색해보고자 한다.

### II. 관련 연구

#### 1) Pixhawk

본 연구에서는 비행 제어 하드웨어로써 3D Robotics사와 AutoPilot Group에서 제작한 Pixhawk를 사용하였다. Pixhawk의 기반 라이브러리는 오픈소스로 공개되어 있어 상업용 드론에 비해 일반 개발자들이 펌웨어에 접근하기 용이하다는 장점이 있다. 뿐만 아니라 microSD slot, UART, CAN, I2C, SPI, ADC 등 다양한 주변 장치들과 연결되기 용이하게 구성되어 있어 다양한 시스템을 구축하기에 용이하다. 그리고 3D Acceleration, Gyro, Magnetic Position sensor가 기본으로 탑재되어 있어, 비행 중 드론의 상태 정

보를 자세하게 받아들 수 있다.[1] 하지만 GPS 모듈은 별도로 구매하여 연결해야 한다.

#### 2) MAVLink(Micro Air Vehicle Link)

Pixhawk는 GCS(Ground Control Station)와 통신하기 위해 MAVLink Protocol을 채택하고 있다. MAVLink Protocol은 작은 사이즈의 UAV와 통신하기 위한 프로토콜로서 8-bit의 packet structure를 갖는다.[2] 송수신기 별로 약간의 차이가 있기는 하지만 지상에서는 1~4km 거리의 통신이 가능하며, 공중에서는 10km 거리까지 통신이 가능하다.

#### 3) 라즈베리파이(Raspberrry Pi)

Pixhawk와 MAVLink로 통신하는 GCS는 여러 가지 방법으로 구성이 가능하다. 그림.3은 GCS를 Lap-top으로 설정하였을 때의 아키텍처 구조도를 나타낸다. Lap-top 컴퓨터에 Mission Planner, Qground Control 등의 GCS software를 설치하고 MAVLink 송/수신기를 연결하여, Lap-top을 GCS로서 활용할 수도 있다. 그러나 이러한 방식은 데이터 사이즈와 전송속도의 문제 때문에 Real-Time으로 영상처리를 하기에 부적합하다. 카메라에 RF(Radio Frequency) 통신 모듈을 부착하여, 드론에서 촬영한 영상을 MAVLink와 별도로 Lap-top으로 받아온다고 하여도 통신 속도에 의한 delay 발생의 문제를 극복하기 어렵다. 따라서 기체와 GCS의 거리가 멀수록 촬영한 영상을 real-time으로 받아오기

힘들며, Data-Loss의 위험성도 증가하게 된다.

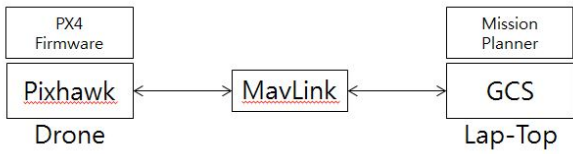


그림.1: Lap-Top을 GCS로 사용했을 때의 시스템 구조

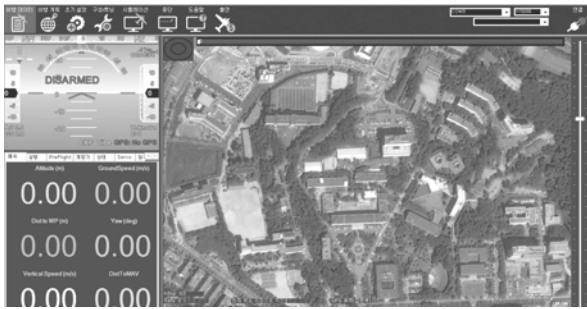


그림.2: GCS Program인 Mission Planner

따라서 본 프로젝트에서는 Real-Time 영상처리가 가능하도록 하기 위하여 Pixhawk와 GCS 역할을 하는 라즈베리파이를 결합하여, Real-Time으로 영상처리를 할 수 있도록 설계하였다. 라즈베리파이에서 MAVproxy라는 GCS 프로그램을 설치하여 Pixhawk와 라즈베리파이 간에 MAVLink 연결을 하게 되는데, 이 경우 그림.3에서 나타나는 Lap-Top과 Pixhawk 간의 Mission Planner를 이용한 MavLink connection이 끊기게 된다.

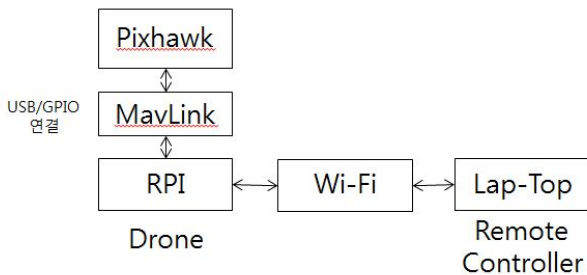


그림.3 : RPi를 GCS로 직접 연결하여 구성한 시스템 구조

#### 4) Drone-Kit

3D Robotics사에서 제공하는 Aerial Platform으로써 MavLink protocol로 연결된 Vehicle에 개발자가 Python, Android API 등을 사용하여, Drone-Kit 응용 프로그램을 만들 수 있도록 기능을 제공한다. Drone-Kit의 주요 목적은 Pixhawk 보드 위에(onboard) 라즈베리파이와 같은 companion computers를 탑재하여 Computer Vision, Path Planning, 3D modelling 등이 가능하게 해준다. 또한 higher latency RF-link에서도 Vehicle과 companion computer 간의 통신이 가능하다.

### III. 본론

기존의 상용 드론은 firmware가 공개되어 있지 않아 개발자의 목적에 맞는 자율적인 기체 제어 프로그램을 개발하기가 어렵다. 하지만 Pixhawk는 firmware가 공개되어 있으며, 3D Robotics 사에서는 드론이 Computer vision을 수행하면서 미션을 수행할 수 있도록 Drone-Kit Platform을 제공한다. 본 프로젝트에서는 Pixhawk와 라즈베리파이, Drone-Kit Platform을 활용한 인명 구조 시스템 설계를 진행하였다.

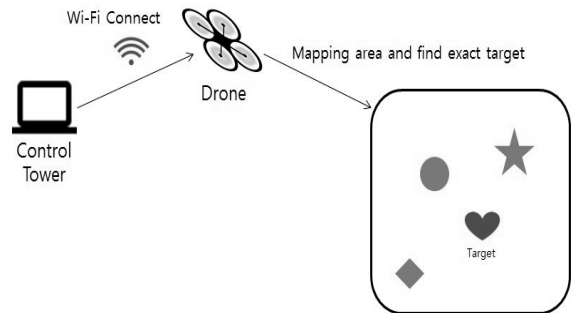


그림.4 : 시스템 설계도

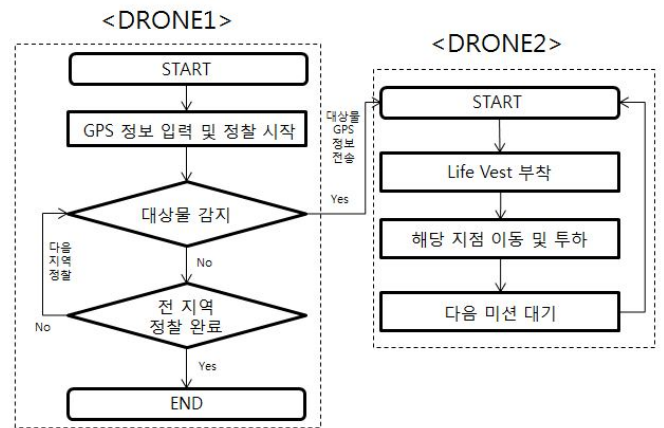


그림.5 : 인명 구조 시뮬레이션 흐름도

기본적인 기체 제작 이후, Control Tower에서 드론에 제어 명령을 내리기 위해 Python으로 Drone-Kit 응용 프로그램을 제작한 후, Wi-Fi로 PC와 라즈베리파이 간의 network를 구성하였으며, Putty를 활용한 SSH 통신으로 원격 제어 하였다.

인명 구조 시뮬레이션은 다음과 같다. 그림.4에 표시된 영역을 정찰하기 위해 라즈베리파이에서 정찰 영역 전체에 대한 GPS 정보들을 저장한다. 드론은 입력된 GPS 정보를 바탕으로 자율 주행하면서, 기체 하단에 부착된 Pi-Camera와 OpenCV를 활용하여 사전에 저장된 하트 모양의 대상물을 감지하게 된다. 드론이 대상물을 감지하면 Life Vest를 떨어뜨리고, 대상물 위치의 좌표 정보를 저장한다.

1) 하드웨어 제작

Pixhawk와 드론 제작에 필요한 프로펠러, 모터 등을 구입하여 그림.5처럼 기본적인 하드웨어를 제작한다. 인명 구조를 위한 구멍조끼의 운반을 위해 Servo Motor 추가 및 집게를 추가적으로 제작하였고, 드론 제어를 위해 기체 하부에 라즈베리파이 및 파이카메라를 연결한다.

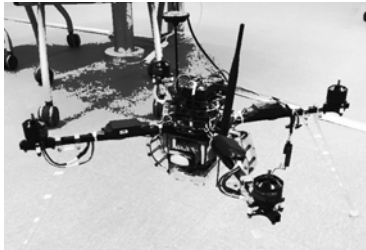


그림.6 : Pixhawk와 RPI를 결합한 드론

이후 Mission Planner와 Pixhawk를 USB 연결하여, 드론에 Firmware를 설치한 후, 기체의 안정적인 비행을 위한 Calibration 설정을 한다.

2) 드론 비행 및 영상 촬영

그림.6, 그림.7은 드론을 각각 10m, 15m 높이에서 촬영한 사진이다. 인명구조를 위해서는 촬영한 사진 정보를 바탕으로 색상 및 모양 인식을 통해 대상물을 탐지할 수 있어야 한다. 대상물의 탐지에는 openCV를 활용한 확대, 축소와 같은 기술적인 분석도 필요하지만 무엇보다 피사체에 대한 정확도 높은 영상정보가 필요하기 때문에 드론의 비행 높이를 달리하며 대상물을 촬영하였다.

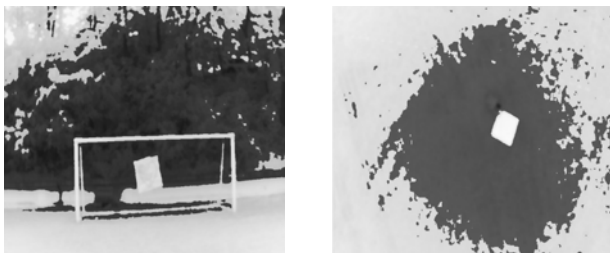


그림.7 : 10m(좌), 15m(우) 높이에서 촬영한 사진

10m 높이에서 촬영한 이미지는 드론이 자율주행하면서, 하트모양에 대한 색상인식이 가능하였으나, 15m에서 촬영한 이미지는 색상인식이 불가능하였다. 영상에 대한 해상도는 721 × 480이었으며 24KB의 크기를 가졌다. 이후 720 × 960으로 해상도를 좀 더 올렸으나, 해상도를 올릴수록 영상의 크기가 커져 라즈베리파이의 영상처리 시간이 지연되는 문제가 발생하였다.

3) GPS 정보를 활용한 지형 정찰

라즈베리파이에 드론이 정찰해야할 그림.4에 표시된 영역의 GPS 정보를 텍스트 파일로 입력하면 라즈베리파이는 사진

에 제작된 응용프로그램을 바탕으로 드론이 자율 주행 비행을 하도록 명령을 내린다. 그와 동시에 라즈베리파이에 연결된 Pi-Camera를 통해 드론이 저장된 GPS 좌표들을 지날 때마다 지형 정찰 및 사진 촬영을 하게 된다.



그림. 8: GPS 정보를 활용한 지형 정찰

4) openCV를 활용한 대상물 인식

드론이 지형 정찰 임무를 수행하다가 HSV 색상 값 인식 기술을 바탕으로 대상물을 감지하게 하였다. 감지 후, 드론은 해당 지점에 대한 영상을 촬영하고 Drone-Kit을 활용하여 촬영 지점의 GPS 정보를 취득 및 파일에 저장하도록 하였다. 관련 연구로 열화상 카메라를 활용하여 사람을 인식하는 경우도 있었으나[3], 화재나 조난으로 인한 인명 구조의 경우 열화상 카메라만으로는 구조해야할 대상물에 대한 정보가 흑백 명암의 차이만으로는 구별하기 어렵기 때문에 본 연구에서는 HSV 모델을 활용한 색상 검출 방법을 활용하였다. HSV 모델을 사용한 이유는 Red, Green, Blue로 모든 색상을 표시하는 RGB 모델과 달리, Hue(색상), Saturation(채도), Value(밝기)를 사용하기 때문에, 야외 환경 촬영에 있어, 인간이 판단하는 컬러 값과 카메라를 통해 컴퓨터가 인식한 컬러 값의 차이가 최소화될 수 있을 것이라 판단하였기 때문이다.

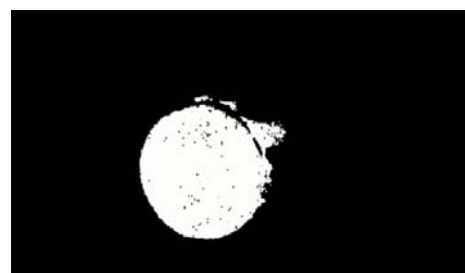


그림. 9: openCV의 색상 인식 기능을 활용한 대상물 감지

그림. 9는 openCV를 활용하여 카메라 화면상에 나타난 붉은 색을 인식하여, 해당 부분의 색상 값을 하얀색으로 그 이외의 색상 값은 검정으로 처리한 것을 나타낸다.

5) 물품 수송을 위한 집게 부착

재난 지역에 대한 영상 정보 추출과 실질적인 구조가 이루어지게 하기 위하여, 라즈베리파이에 서보모터를 연결하여, 물품을 자유롭게 낙하할 수 있도록 집게를 제작하였다.



그림.10: 물품 수송을 위한 집게 제작

#### V 참고문헌

- [1] Pixhawk, <https://pixhawk.org/>
- [2] MavLink, <http://qgroundcontrol.org/mavlink/start>
- [3] Ji Hoon Joung, Wonpil Yu, 히스토그램 평활화를 이용한 열화상에서의 보행자 검출을 향상 방법. 대한전자공학회 하계종합학술대회 제34권 1호, 2011년.
- [4] Meier, L., Tanskanen, P., Fraundorfer, F., Pollefeys, M.: Pixhawk: A system for autonomous flight using onboard computer vision. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 2992-2997 (2011)

이러한 기능들을 바탕으로 재난 상황에 대처하는 드론 제어 시스템을 구축하고 다음의 시뮬레이션을 진행하였다. 드론이 그림.4의 영역을 정찰하다 하트 모양의 대상을 감지하게 되면, 라즈베리파이가 Servo Motor를 움직여 Life Vest를 떨어뜨리게 된다. 실제 비행에서는 모터의 출력이 부족하여, 가벼운 물체로 대신하였다.

#### IV. 결론

Pixhawk과 Drone-Kit을 활용하면 개발자가 자신의 목적에 맞게 드론을 직접 제작 및 제어가 가능하다. 본 프로젝트는 이러한 기술을 바탕으로 인명 구조 시스템을 설계 및 테스트하였다. 이후 개발로는 드론과 GCS 간의 통신 방식 개선과 실제 실험을 통한 색상 인식 성능 측정 및 개선, 효율적인 지형 Mapping 방법에 대한 연구가 필요하다.

이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2015R1D1A1A01060034)