Professor 홍충선

모션 인식을 통한 쉬운 드론 제어 시스템

Leapmotion Drone Controller

2012101978 양정일

2012104136 황진하

2014110449 김태석

2017 May 14

요약

드론은 다양한 산업, 국방, 개인 취미 등 다양한 분야에서 적용 가능성이 높아 사회적으로 많은 관심을 받고 있다. 그러나 이러한 관심에도 불구하고 조종대 이용한 드론 조작법을 습득하기 어려워 초보자들은 쉽게 활용하기가 어렵다. 본 논문에서는 립모션 핸드 모션 인식 기기를 활용해 직관적으로 제어 가능한 드론 컨트롤 시스템을 제안한다. 제안하는 립모션을 드론 제어 시스템은 손을 사용한 쉬운 조작뿐만 아니라 사용자와 시점을 일치시켜 직관적으로 드론을 제어할 수 있다.

목차

[1. 서론 2](#_Toc482468399)

[1.1. 연구배경 2](#_Toc482468400)

[1.2. 연구목표 2](#_Toc482468401)

[2. 관련 연구 2](#_Toc482468402)

[2.1. 드론 2](#_Toc482468403)

[2.2. 드론 제어 장치 4](#_Toc482468404)

[2.3. LeapMotion 4](#_Toc482468405)

[2.4. 드론과 제어장치간 통신 5](#_Toc482468406)

[2.5. 기존 연구의 문제점 및 해결 방안 6](#_Toc482468407)

[3. 프로젝트 내용 6](#_Toc482468408)

[3.1. 시나리오 6](#_Toc482468409)

[3.2. GUI 구성 9](#_Toc482468410)

[3.3. 요구사항 (이 부분 새로 작성) 9](#_Toc482468411)

[3.4. 시스템 설계 10](#_Toc482468412)

[3.5. 구현 14](#_Toc482468413)

[4. 결론 14](#_Toc482468414)

[5. 참고문헌 14](#_Toc482468415)

## 서론

### 연구배경

현재 드론은 국방, 보안, 산업 등 다양한 분야에서 활용하기 위한 연구가 활발히 진행되고 있다. 예를 들어, 물류 분야에서 물류 운송, 농업 분야에서 파종 및 비료 작업, 재난 구조를 위한 공중 촬영을 할 수 있다. 또한 개인의 취미로 레저용, 스포츠용으로 활용하기도 한다

그러나, 위와 같은 드론의 다양한 활용 처 와는 달리 개인이 드론을 소유하고 있는 경우는 많지 않다. 드론이 대중적으로 확산되지 못하는 이유 중 하나는 드론 조종의 불편함이다. 사용자가 드론을 처음 접하면 드론의 시점을 고려하여 드론의 움직임을 제어하는 어려움이 발생한다. 드론을 제어하는 연산인 피치, 요우, 스로틀, 롤을 통해 이동 방향과 시점을 조작하는 방법이 구분 되어있다. 그로 인해 드론 조종 시 사용자가 보는 시선과 일치하지 않는 괴리감으로 인해 조종에 불편함을 느끼게 된다. 드론 조종에 숙련된 경우에도 사람이 계속 조종대 통해 이동할 위치를 정해야 하기 때문에, 컨트롤러에 주기적인 물리적 압력을 가해야 한다. 따라서 이러한 결과로 오랜 시간 드론을 조종하기 어렵다.

따라서 기존의 불편한 조종 컨트롤러 및 방식을 대신해 립모션 핸드 모션 인식 기기를 활용해 직관적으로 제어 가능한 드론 컨트롤 시스템을 제안한다. 해당하는 드론 제어 시스템은 손을 사용한 쉬운 조작뿐만 아니라 사용자와 시점을 일치시켜 직관적으로 드론을 제어할 수 있다.

### 연구목표

본 프로젝트에서는 립모션(Leap Motion) 핸드 모션 인식 기기를 활용해 직관적으로 제어 가능한 드론 컨트롤 시스템을 제안한다. 제안하는 립모션을 드론 제어 시스템은 손을 사용한 쉬운 조작뿐만 아니라 사용자와 시점을 일치시켜 직관적으로 드론을 제어할 수 있다. 이에 해당 하는 연구 목표는 다음과 같다.

첫 번째 목표로 드론을 조종하기 위해서는 드론과 서버를 연결한다. Default State로 대기하는 드론은 Server의 요청으로, 연결을 수행하게 되며, 이후 서버의 명령을 처리한다.

두 번째 목표는 모션인식 기능을 수행하는 장치인 립모션을 통해, Server에 모션 인식 Data를 입력하고, 이 Data를 드론을 제어하기 위한 Data로 처리하는 것이다.

세 번째 목표는 드론 기능을 제어하고, 이를 제어함에 있어 편리하게 하는 것이다. 처음 드론을 사용하는 사람도 간단한 설명을 통해 드론을 제어할 수 있도록, 핸드 제스쳐를 통해 드론을 제어할 것이며, 이에 대한 기능을 추가하여 활용성을 높일 수 있는 방향으로 구현하는 것이다.

네 번째 목표는 드론과 사용자와의 시점을 일치시켜 직관적으로 드론을 제어하는 것이다. 시점의 일치를 통해 드론 상시로 사람을 기준으로 움직여 직관적인 제어가 가능하도록 한다.

## 관련 연구

### 드론



* 드론은 기체에 사람이 타지 않고 지상에서 무선 전파유도에 의해서 원격조정이 사전에 입력된 프로그램에 따라 비행이나 조정이 가능한 무인 항공기를 말한다. 초기의 목적은 표적, 정찰, 감시 등의 군사용으로 개발되었다가 점차 활용 목적이 민간분야로 확대되어가고 있으며 화산지역, 자연재해 지역, 원자력 발전소 사고지역 등의 인간이 접근하기 어려운 지역으로 드론을 투입하여 운용하기도 한다. 요즘엔 카메라를 장착한 드론인 헬리캠으로 다양한 각도의 생동감 있는 영상을 촬영하여 방송, 산업, 개인의 취미활동 등에 사용되어 일반인들도 드론을 많이 이용하고 있다.

### 드론 제어 장치



* 드론을 제어하는데 있어서 기본적으로 두 가지 방법이 있다. 가장 일반적인 방법으로 많이 쓰이는 조이스틱을 통한 조종과 최근에 많이 쓰이는 스마트폰 앱을 통한 조종이 있다.
* 일반 조이스틱 장치는 보다 섬세하게 드론을 조종하며 인식거리도 길다는 장점이 있지만 초보자들이 사용하긴 쉬지 않다는 단점이 있는 반면, 스마트폰 앱에 경우 조이스틱 장치보단 쉽게 조종을 할 수 있다는 장점이 있다. 하지만 인식거리가 짧은 편이며 스마트폰 성능에 따라 반응속도가 달라질 수 있으므로 최신형 스마트폰이 아닐 경우 불편함을 겪을 수 있다.

### LeapMotion



* 립 모션(Leap Motion)은 센프란시스코에 있는 벤처회사이자, 매우 정교한 손 동작 인식 센서를 지닌 유저 인터페이스를 제공하는 장치를 말한다. 작은 아이팟 크기의 장치로 8입방 피트(cubic feet)의 3차원 공간을 매우 정확하게 읽어낸다.
* 립모션의 기술은 MS의 키넥트(Kinect)와 비슷한 원리다. (키넥트는 색상을 구분하는 RGB 카메라와 깊이를 구분하는 IR 카메라로 구성됨) 하지만, 키넥트보다 200배 높은 감도를 가지며, 약 100분의 1밀리미터의 움직임까지도 김지한다. 키넥트가 모니터의 정면에 위치하여 동작을 인식한다면, 립모션은 화면 아래에 위치해 동작을 인식한다.

### 드론과 제어장치간 통신

* 드론은 원격 또는 자동 조종을 통해 임무를 수행할 때 무선통신 기술을 사용한다. 이러한 통신 중 위성통신은 보통 군용 및 방송용으로 사용하고, 민간인들이 드론제어에 사용 가능한 무선 통신엔 블루투스, WI-F, 셀룰러 시스템 등이 있다.

#### 블루투스

* 블루투스는 컴퓨터, 스마트폰 음향기기 등과 같은 기기들을 서로 연결하여 정보를 교환할 수 있는 근거리 무선 기술 표준이다. 주로 단거리에서 저전력 무선통신이 필요할 때 사용된다.
* 일반적으로 블루투스는 전송속도가 Wi-Fi에 비해 비교적 느린 대신 저전력 통신을 제공해주는 기술이기 때문에 많은 데이터전송이 필요하지 않은 드론 제어에 있어서는 적합하다. 하지만 임무수행에 있어 사진, 혹은 동영상 등의 고용량 자료전송이 힘들다는 단점이 있다.

#### 셀룰러 시스템

* 셀룰러 시스템은 이동 무선 통신에서 기지국이 넓은 영역을 셀이라 불리는 구역으로 나누어 통신 서비스를 제공하는 것이다.
* 드론에서 셀룰러 시스템을 사용한다면 제조사 입장에서는 통신사와 연계를 해야 한다는 제도적 문제점이 있고 사용자입장에서는 매달 통신료가 청구된다는 문제점이 있다. 반면 국내의 경우 촘촘한 망으로 인해 어느 장소라도, 통신이 끊기지 않는다는 장점이 있다. 특히 드론처럼 이동성이 높은 기기에는 유리한 장점이다. 하지만 공중에 셀룰러 망이 개설되어 있지 않기 때문에 망을 사용하기 위해서는 고도에 제한을 받는 등 드론 무선통신에 적용하기에는 어려움이 있다.

#### Wi-Fi

* Wi-Fi란 Hi-Fi(High Fidelity)에 무선 기술을 접목한 것으로 근거리 컴퓨터 네트워크 방식인 랜(LAN, Local Area Net-work)을 무선화 한 것이다. Wi-Fi는 고속의 데이터 전송이 가능하여 하나의 채널로 제어 신호와 함께 실시간으로 영상 전송이 가능하다는 장점이 있다. 또한 추가적인 수신기 없이 일반적으로 사용하는 노트북 혹은 스마트폰과 직접 연결이 가능하지만 드론에 사용하기에는 몇가지 문제점이 있다. 첫 번째로 Wi-Fi모듈들은 출력이 제한되기 때문에 제한된 출력으로 드론을 제어하기에는 통신범위의 제약이 존재한다. 두 번째로 만약 출력제한이 존재하지 않는다 하더라도 Wi-Fi는 비 허가대역인 ISM대역을 사용하기 때문에 통신범위가 넓어지면 같은 채널을 사용하는 기기들 간의 간섭문제가 발생한다. 이러한 한계점들로 인해 Wi-Fi는 특수 목적용 드론보다는 레저용 드론에 주로 사용된다.

#### 위성통신

* 위성통신이란 인공위성이 중계소 역할을 담당하는 장거리 통신방법으로 대기권 밖의 상공에 쏘아 올린 인공위성을 통해 통신 신호를 중계한다. 인공위성은 용도에 따라 군사위성, 기상위성, 과학위성, 통신위성 등으로 분류되는데 그 중 통신위성이란 통신 신호를 중계할 목적으로 지구를 돌고 있는 인공위성을 의미한다.
* 위성통신의 경우 대부분 지상에 망이 구축되는 셀룰러 시스템, Wi-Fi 환경과 다르게 재해 혹은 전시와 같이 지상의 기반 통신시설이 붕괴된 상황에서도 사용이 가능하다. 특히 드론의 경우 군사용으로 많이 사용된다는 것을 감안하면 이는 다른 통신 기술들에 비해 큰 장점이 될 수도 있다. 하지만 비용 및 주파수의 포화, 지상과 교신 시 시간지연 등 여러 가지 제한사항들로 인해 일반적인 사용에는 어려움이 있다.

### 기존 연구의 문제점 및 해결 방안

* 기존의 드론은 위에서 언급한 것처럼 기본 조이스틱과 스마트폰 앱을 통한 조종이 대부분이었다. 하지만 이러한 장치들은 여전히 초보자들에겐 어렵고 섬세한 기술들이기에 사고가 날 수 있는 위험이 높다.
* 이러한 문제들을 해결하기 위해선 위에서 언급한 Leap Motion, App을 통해 어려운 조종방식을 배워야 하는 조종기가 아닌 직관적으로 모든 사용자들이 다룰 수 있게 간단한 손동작 하나로 드론을 제어 할 수 있는 사용자 환경을 제공되어야 한다.

## 프로젝트 내용

### 시나리오

#### 서버 연결



Default State인 Drone을 제어하기 위해, Server와 연결을 한다.

연결이 성공적으로 완료 되면 Drone은 Connected State가 된다.

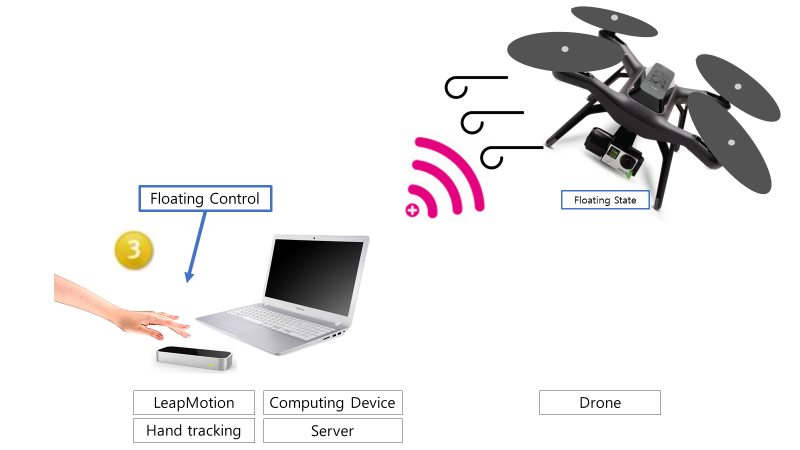
#### 이륙 명령 입력



Connected State인 Drone을 이륙하기 위해 Floating Motion을 LeapMotion을 통해 입력한다.

입력 된 Motion Data는 Server Application을 통해 Drone에 float 명령을 전송하게 된다.

#### Hand Tracking을 통한 드론 기능 제어



Floating State인 Drone의 움직임을 제어하기 위해, LeapMotion에 Drone control motion을 입력한다.

Drone은 Server로부터 받은 명령에 따라, 정해진 비행을 하거나, 사진을 촬영하는 등, 해당 기능을 수행하게 된다.

#### 착륙 명령 입력 / 연결 해지



Floating State인 Drone을 Landing 시키기 위해, 해당 Motion을 입력한다.

해당 Server는 드론의 안전한 착륙을 보조하며, Drone은 Landing 기능을 수행한다.

이후, 해당 기능을 마친 Drone과 Server의 연결을 해지한다.

### GUI 구성



구현된 GUI 환경은 다음과 같다.

작성한 프로그램을 크롬 웹 브라우저로 실행 시킨 모습이다.

좌측 패널에는 드론이 촬영한 영상을 실시간으로 전송되는 화면을 표시한다.

우측 패널에는 드론 컨트롤러의 현재 입력 상태와, 하단에 상태창을 표시한다.

### 요구사항

#### 환경에 대한 요구사항

* 해당 프로젝트에 사용하는 Parrot Bebop Drone은 기기의 특성 상, 바람에 대한 영향을 많이 받기에 장비를 운용하는 날씨가 바람이 강하게 불지않는 날씨여야 한다.
* Drone의 특성 상, Drone이 일정공간을 자유롭게 비행하여도, 충돌이 일어나지 않도록, 어느정도 공간이 충분한 상태여야한다.
* Server와 Drone의 통신 환경이 일정하게 유지되기 위해, 전파의 방해를 받지않는 곳에서 운용하여야 한다.

#### Drone에 대한 요구사항

* Drone에 대한 제어 명령을 전송하기 위해, Server와 통신하기 위한, 통신 장치가 요구된다.
* Drone에 대한 제어를 위해, Drone의 현재 상태를 기록하는 Sensor가 자체적으로 내장 되어있어야 한다.
* Drone에 대한 제어 오류로 생기는 사고를 발생하기 위하여, 자체적으로 안정 장치가 설치 되어있어야 한다.

#### LeapMotion에 대한 요구 사항

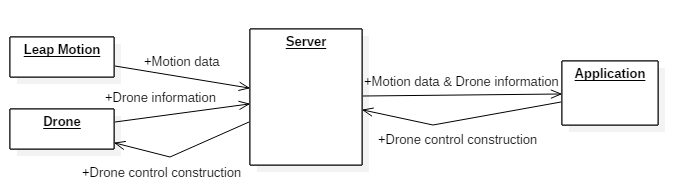
* 해당 프로젝트의 안전을 위해, Drone을 제어 할 수 있는, User의 Motion을 정확하게 인식하여야 한다.
* LeapMotion의 Motion Data를 처리 할 수 있는 별도의 Application을 필요로 한다.

#### Computing Device에 대한 요구 사항

* Motion Data를 처리하고, Drone에 연결되는 서버 역할을 할 수 있는 정도의 사양이 필요하다.
* Server로 운용하기 위해, Drone와 연결 할 수 있는 통신 기능이 있어야 한다.

### 시스템 설계

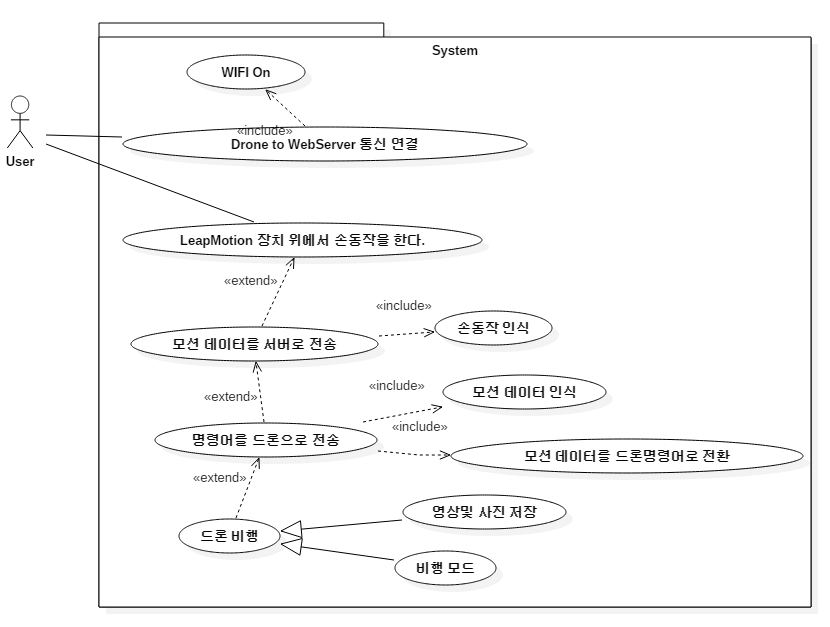
#### 시스템 구성도



시스템은 서버와 어플리케이션으로 구성되며, 입력 장치로는 Leap Motion을 사용해 손의 움직임을 감지해 Motion data를 입력한다. 해당 Motion Data는 Application을 거쳐, Drone의 Control Construction으로 구성이 되며, 해당 Command를 통하여, 최종적으로 Drone을 Control 한다.

#### UML Diagram을 통한 시스템 모델링

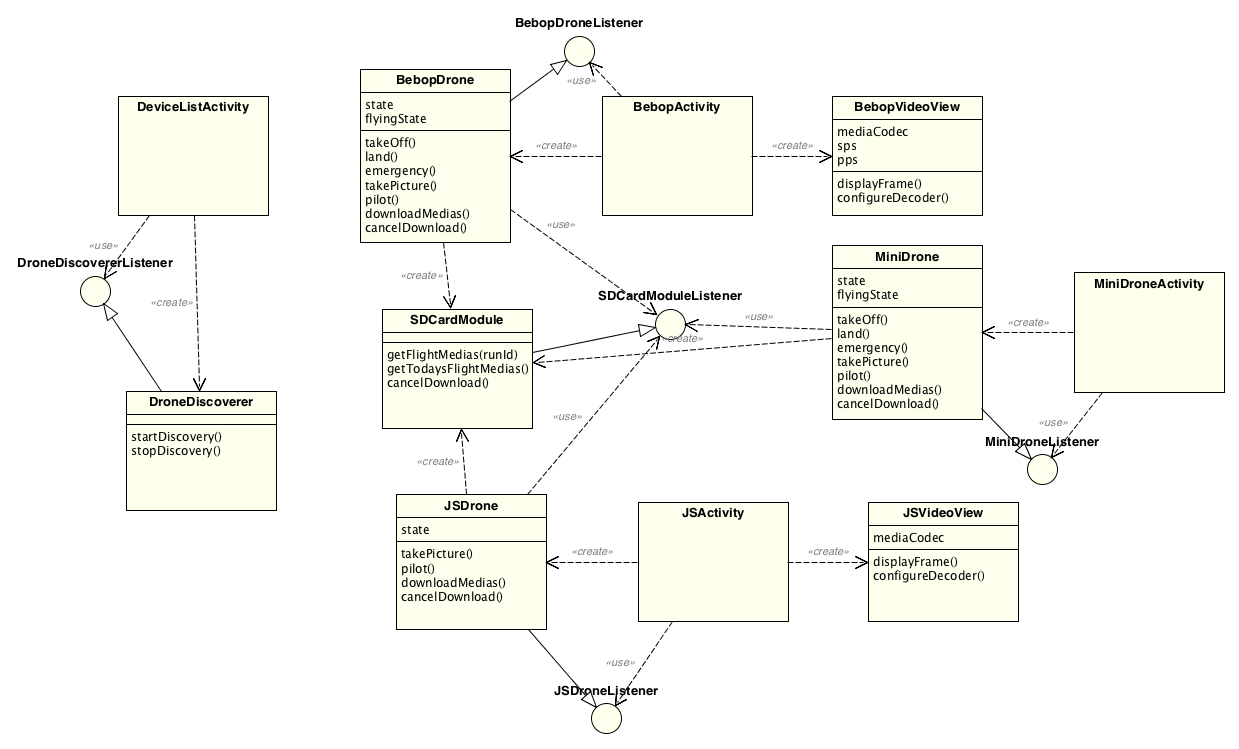
##### Use Case Diagram



현 프로젝트 내 Actor로써 User만 존재한다. 해당 User Actor는 Device와 Server간의 연결을 수행 할 수 있고, LeapMotion Device내에 Motion Data를 입력 할 수 있다.

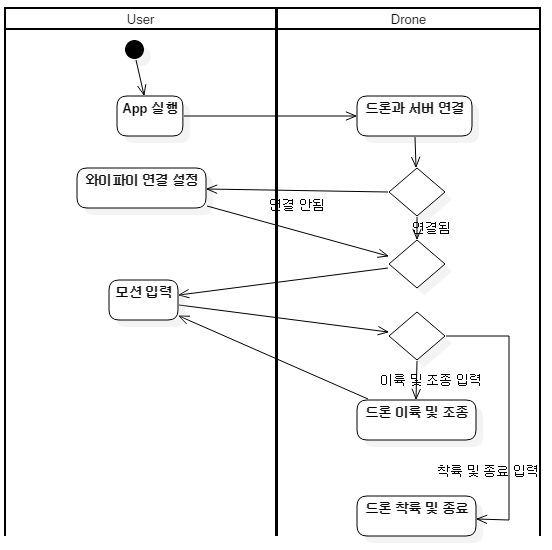
User는 Drone과 Server를 Wi-Fi를 이용하여 통신한다. User가 LeapMotion 장치 위에 손을 올리고 모션을 입력하면, 기기가 모션을 인식하고 모션 데이터 값을 Server로 전송한다. 전송된 모션 데이터는 Server에서 드론 명령어로 전환되며 해당 값은 드론으로 전송된다. 명령어를 입력 받은 드론은 현재 전송 값에 따라 동작 한다.

##### Class Diagram



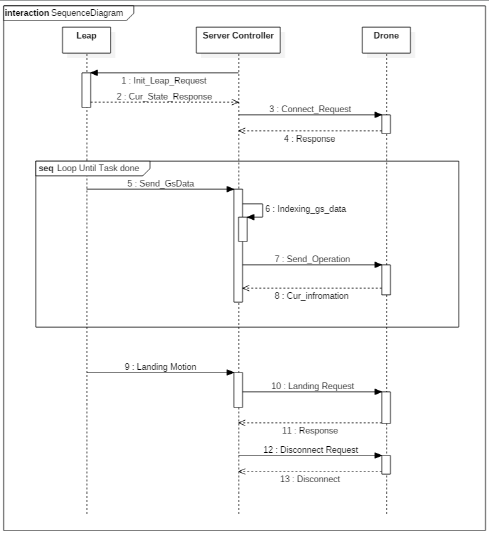
클래스는 크게 Bebop, JS, Mini 부분으로 이루어져 있다. Mini에서 사용자의 입력 값을 통해 드론에게 전달할 명령 값을 처리하여 전송한다. 전송 받은 Bebop은 전송 받은 명령 값을 통해 드론이 조종된다. 비행 중 상태를 JS에 반환한다. JS에 반환된 정보는 드론에 대한 정보와 영상이 있다. JS는 Mini와 Bebop에 의해 전송된 조종 정보와 영상 등을 출력한다.

##### Activity Diagram



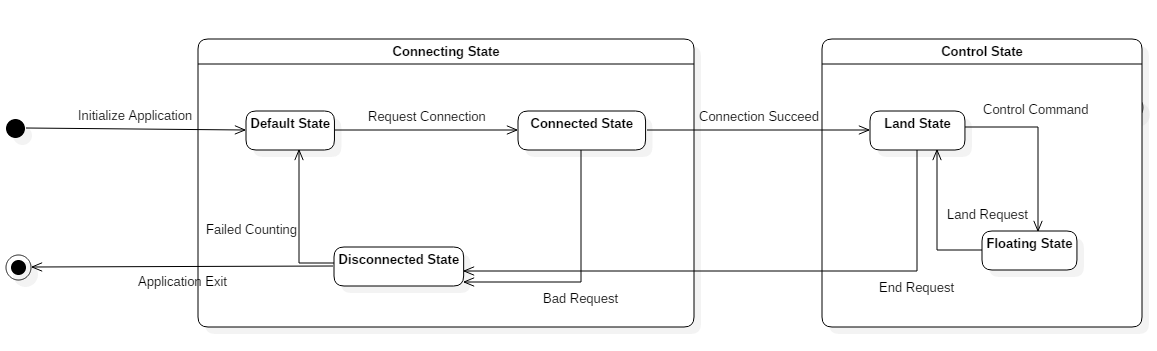
Leap을 통해 서버와 드론에 연결을 요청한다. 통신이 설정하기 전까지 반복하여 연결을 설정한다. 연결이 설정되면 leap에서 모션을 입력하여 server에 전송한다. server에서는 모션 데이터를 처리하여 Drone에 명령 값을 전송한다. 전송 받은 값에 따라 Drone은 이륙 및 비행을 하고 이 수행을 착륙 및 종료 명령 값이 전송될 때까지 무한히 반복한다. 착륙 및 종료 모션을 입력하여 서버에 전달하면 데이터를 처리하여 Drone에 착륙 및 종료 값을 전달하고 통신 종료를 수행한다.

##### Sequence Diagram



위 Sequence Diagram는 전체적인 흐름을 보여주고 있다. 초기에 Leap으로부터 연결을 요청하고 드론은 연결 요청에 대해 정보를 반환하여 통신을 설정한다. 설정이 된 후에는 Leap으로부터 조종에 대한 명령 값을 전달한다. Server는 data를 처리하여 드론에게 명령 값을 전달한다. 착륙 명령 값이 전달되기 전까지 무한히 반복한다. Landing 명령 값이 전송되면 server는 명령 값을 처리하여 드론에게 전송한다. 착륙 명령 값이 전송된 후에 통신 종료 요청을 통해 연결이 종료된다.

##### State Diagram



State는 Drone과 Control Server 사이의 Connecting 상태 와, Control 상태의 2가지로 나뉘어 진다.

컨트롤러 실행 시, Control Server에 연결을 위한 Initializing을 하며, Default State로 돌입한다.

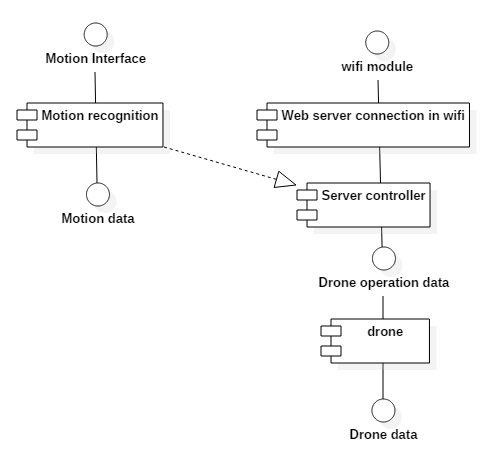
이후, 드론의 IP주소로 접속해, 연결 요청을 수행하게 되며, 해당 연결의 Response에 따라, Connected State와 Disconnected State로 나눠진다.

정상적인 연결이 수행되면, 드론의 움직임을 조종하는 Control State로 돌입한다.

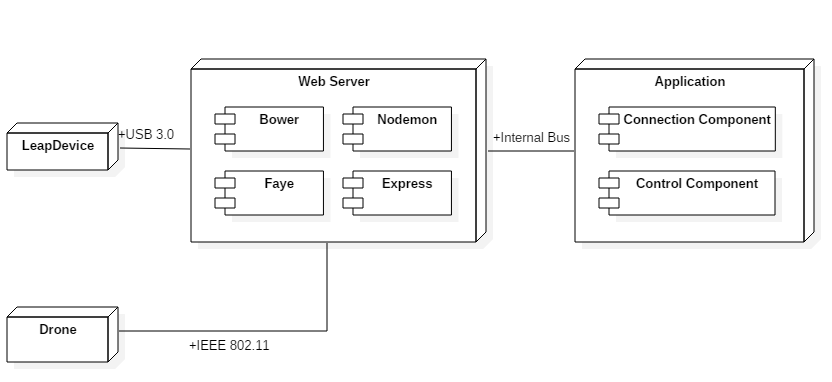
Control State는 드론이 땅에 Landing 되어있는 Land State와 드론의 활강 상태를 조종하는 Floating State롤 구분이 된다.

해당 하는 프로그램을 마치면, 연결을 해지에 Disconnected State에 돌입. 정상적으로 프로그램을 종료한다.

##### Component Diagram



##### Deployment Diagram



Deployment Diagram은 총 4개의 Node로 구성된다.

입력 장치에 해당하는 LeapDevice와 Output command를 수행할 Drone이 각각의 I/O Node로 존재한다.

해당하는 Device의 연결을 실행하고, User에게 View를 제공하기 위한 Web Server가 존재하며, 이 Web Server는 4개의 Component Module을 통해 구성된다. Motion Data의 해석과 Control에 대한 명령어 생성은 Application Node에서 구현이 되며, 프로그램이 실제로 계산이 되는 부분이다.

각각의 Node간의 연결은 위의 그림을 통해 연결 방식을 확인 할 수 있다.

##### Event Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Easy Drone Control System using Motion Recognition Event Table** | | | | | |
| **Event** | **Trigger** | **Source** | **Use Case** | **Response** | **Dest** |
| **Server 연결** | **WiFi 연결 성공** | **사용자** | **연결 입력 확인** | **연결 성공 출력** | **Server** |
| **LeapMotion기기 처음 연결** | **기기간 연결 성공** | **사용자** | **연결 입력 확인** | **연결 성공 출력** | **LeapMotion** |
| **Drone기기 처음 연결** | **기기간 연결 성공** | **사용자** | **연결 입력 확인** | **연결 성공 출력** | **Drone기기** |
| **사용자가 LeapMotion기기 위에서 손동작** | **LeapMotion 기기 연결 성공** | **사용자** | **손동작 인식 확인** | **모션 데이터 출력** | **LeapMotion** |
| **LeapMotion Data 생성** | **손동작 인식 성공** | **사용자** | **모션 데이터화** | **모션 데이터를 서버에 전송** | **LeapMotion** |
| **모션데이터를 드론 명령어 변환** | **모션 데이터 인식 성공** | **사용자** | **데이터 변환 확인** | **명령어를 드론으로 전송** | **Server** |
| **드론 비행** | **드론 명령어 인식 성공** | **사용자** | **드론 명령어 확인** | **명령어 수행** | **Drone 기기** |
| **기기간 연결 해제** | **해제 성공** | **사용자** | **연결 해제 시도** | **연결 해제 출력** | **모든 기기** |

현재 연구 중인 프로젝트는 Database를 사용하지 않으므로, Entity-Relation Diagram은 생략한다.

### 구현

구현에 대한 코드는 다음의 Khuhub를 통해 확인 할 수 있다.

Khuhub URL : <http://khuhub.khu.ac.kr/2017-1-creative-design/LeapMotionDroneControl>

## 결론

본 프로젝트를 통해, 모션 인식을 이용하여 손의 움직임으로 드론을 제어한다. 또한 사용자와 드론의 시선을 일치시키는 솔루션을 통해 기존의 드론의 시점과 사용자의 시점이 일치하지 않아 생기는 문제는 해결한다. 따라서 해당 연구를 통해, 드론 사용에 익숙치 않거나 드론을 처음 조종하는 조종자에게 더 쉬운 조종 환경을 제공한다.

## 참고문헌

[1] 장성기, 『드론 새로운 세상을 만나다』, 크라운(2016)

[2] 우지윤, 『스크래치 아두이노 립모션』, 디지털북스(2016)

[3] Google, "3DR Solo Development", https://dev.3dr.com/, (2017.3.27)

[4] Google, “Leap Motion Developer”, https://developer.leapmotion.com/, (2017.3.27)