0

|  |
| --- |
|  |
| **캡스톤 연구/개발 노트** |
|  |

|  |  |
| --- | --- |
| **소 속** | ICT 융합학부 |
| **연구과제명** | **무인기 정밀 착륙 유도 장치 개발** |
| **연구기간** | 2024.09.02 ~ 2025.07.01 |
| **연구책임자** | 김교원 |
| **지도 교수** | 박정은 |

프로젝트 개요

❑ **프로젝트명**

|  |
| --- |
| 무인기 정밀 착륙 유도 장치 |

❑ **키워드**

|  |
| --- |
| 드론, 정밀 착륙, 미션 효율성 향상, 부하 방지 |

❑ **서론**

|  |
| --- |
| 현재, 전 세계적으로 무인항공기 사업이 활발히 이루어지고 있으며, 국내에서 해당 분야R&D 예산도 증가하는 추세입니다. (과기부 2023 첨단융합기술 R&D 사업 예산 중 1/3이 무인이동체 분야에 투자됨, 2022에 비해 16% 향상 – 출처: <https://www.e4ds.com/sub_view.asp?ch=16&t=0&idx=16176>)  이렇게 투자되는 연구 자금 덕에, 기술력이 많이 향상되어 국내 사업분야들은 일반적으로 드론 하면 떠오르는 항공 뷰 촬영 및 단순 조종이 아니라 특수 목적 드론에 연구 초점이 많이 맞춰지고 있습니다.  좋은 상황이지만, 특수 목적 드론의 경우 (1)전문 교육을 받은 운용자들도 운용 시 어려움을 느끼는 점(특수 목적 드론 기술의 복잡성, 실전 경험 부족 등), (2)아직까지는 일부 자동화 기능의 성능이 아쉬운 점, (3)일반적으로 고정된 운용 거점을 활용하는 경우가 많아 유동적인 활용이 어려운 점 등의 문제점들이 있습니다.  (위 근거는 특수 목적 드론을 생산하고 특정 제품에서 전투용 적합 판정을 받았었던 중소업체 내부의 운용자와 개발자 시점에서 의견을 바탕으로 작성하였습니다)  따라서 위 문제점들을 상세히 파악하여 해결해보는 방향으로 과제 초점을 잡게 되었습니다.  <문제점 1 설명>  먼저 드론은 비행체이므로 일반 자동차처럼 4방향 조종이 아닌 Yaw,Roll,Pitch 제어가 필요합니다.  다만 이 제어를 위해서 4축 조이스틱이 2개 필요합니다.  드론 조종기 잡는 방법이 따로 있다고요? <사진 1> 조이스틱 2개를 쓰고 있는 일반적인 드론 조종기 사진.  이러한 상황에서 특수 목적 드론의 경우 양 손의 엄지를 조이스틱에 둔 채로 미션 수행을 위해 지정된 버튼 등의 트리거 장치를 활성화해야 합니다. 그렇다 보니 운용자 입장에서는 해당 드론의 기능이 다양할수록 어려움을 느끼게 됩니다.  위 문제가 지속될 경우 운용자에게 아래 문제들을 초래할 수 있습니다.   1. 비행 시간이 누적될수록 사용자의 피로도가 더욱 누적될 수 있음. (어려울수록 집중해서) 2. 여러가지 상황을 보고 그 순간에 맞게 운용해야 하다 보니 정보 과부하가 올 수 있음. 3. 복잡한 조작이나 트리거 사용 중에 조작 실수가 일어날 수 있음.   <문제점 2 설명>  일반적인 드론의 자동화 기능 중에 가장 많이 쓰이고 중요한 것이 특정 위치에 착륙하는 기능이라 생각합니다. 해당 기능은 위급 상황(배터리 부족, 통신 두절 등)에서 사전에 지정된 위치로 복귀하여 사고를 방지할 수 있고 또한 착륙 시 지정된 위치에 착륙하기만 해도 여러가지 사고(충격 완화 재질로 외부가 구성된Station이 아닌 다른 위치에 착륙 시 드론에 손상이 일어남, 군사용 드론일 시 정찰 후 외부에 알려지지 않은 본대로 복귀하지 않을 경우 곤란한 상황이 일어남 등)를 미연에 방지한다고 생각합니다.  다만 일반적인 자동 랜딩 기능의 경우 GPS 좌표 데이터만을 사용하여GPS 값이 정교하지 않거나, 위성이 일정 개수 이상 잡히지 않을 경우 오차율이 심합니다.  이렇게 자동 랜딩 기능이 정교하지 않을 경우, 아래 문제들을 초래합니다.   1. 랜딩 스테이션을 벗어났기에 안정적인 랜딩이 불가능해지고, 랜딩 시 지면에 닿는 스키드, 바디 프레임 등이 데미지를 입어 내구도, 외부 손상으로 인해 드론의 균형이 깨져 이후 비행 시 안정적인 제어가 어려울 수 있음. 2. 완전한 복귀가 아닌, 미션 수행 중 경유지일 경우 목표 지점에 도착하지 않을 시 다음 운용에 어려움이 있음. 3. 단순 랜딩의 문제가 아닌, 해당 좌표로 제대로 갈 수 없음을 의미함.   <문제점 3 설명>  아래 상황에서 문제가 될 수 있습니다.   1. 배터리 교체, 등의 특수 상황에 매번 고정된 거점으로 이동하여야 함. 2. 거점이 고정되어 있으면 경유지가 많은 상황에서 효율적인 동선 설계가 불가능함. 3. 선박에서 운용 시, 선박에 고정되어 있더라도 배가 지속적으로 움직이게 되어   운용 거점이 의도치 않았지만 이동하는 상황이 있음.  결론적으로 위 문제점들을 해결하기 위해 아래 조건들을 충족해야 합니다.   1. 운용자는 자동화 기능이 필요하며, 전부 다 자동화되면 좋겠지만 특정 상황에 트리거를 활성화하는 것보다 드론의 움직임의 자동화를 우선 시 할 시 조작이 보다 용이함. 2. 자동화 기능의 성능이 향상되어야 하며, 그 중 자동 랜딩 기능의 위치가 정밀해야 함.   다만 기존 방식은 GPS 값 만을 쓰기에 보정을 위한 다른 데이터가 필요함.   1. 운용 거점이 이동 가능하여야 하며, 이동된 좌표를 사용자 또는 드론에게 주기적으로 알려주어야 함.   따라서 저는 좌표 데이터만으로 드론이 자동으로 해당 좌표로 이동하며 (1조건 충족)  이동 후, 특정 모듈을 통해 다른 데이터를 받아 랜딩 정밀도를 보정할 수 있으며 (2조건 충족)  이동하여도 착륙에 문제가 없도록 Landing Station이 드론에게 GPS 데이터를 지속적으로 주는(3조건 충족)  무인기 정밀 착륙 유도 장치의 개발이 필요하다 생각하였습니다. |

❑ **기대효과**

|  |
| --- |
| o 운송 드론의 경우 기존 방식보다 더 정확한 랜딩이 가능해져 좋습니다.  o 랜딩 지점의 정확도가 올라가게 되면, 랜딩 이후 후속 작업이 용이합니다. (예를 들자면 운송 드론 랜딩 이후 해당 물품을 자동으로 꺼내주는 장치의 설치 위치, 작업 연계성 등에서 강점이 있다고 생각합니다.)  o 의도되지 않은 착륙을 방지하여 착륙 중 발생할 수 있는 손상이나 사고가 방지됩니다.  o 이 외 GPS 데이터와 마커를 통해 보정하는 방식이 있으나, 그 방식보다는 비용측면에서 절감 가능.  o 도서 간 수송목적의 드론의 경우, 효율적인 waypoint 설정 및 비행이 가능합니다.  o 선박에서의 운용 시 기존 방식보다 효율적인 운용이 가능합니다.  o 이 외, 해당 자동화 기능을 사용하는 상황에서 이점을 볼 수 있습니다. |

❑ **개발 목차**

|  |
| --- |
| 1. 시스템 구성도, 설계, 동작 흐름 2. 각 시스템 파트별 제품 선정 및 임의 견적 3. Git Repository 생성 4. 자재 수급 가능 여부 및 납기일 파악 5. 자재 구매 6. 모듈 단위 검증    1. FC 검증 – MissionPlanner 연동 및 파라미터 세팅    2. GPS 모듈 - NMEA 형식 데이터 수신 확인    3. ESC (모터 구동, PWM 제어, Calibration)    4. Drone Power Board 전력 공급 확인    5. Telemetry 동작 확인 (datasheet 기준 통신 거리 등 검증)    6. 배터리 에이징 테스트 (충, 방전 5회 테스트 진행) 7. 드론 개발    1. 하드웨어 납땜 및 체결       1. FC - MainBoard 체결       2. MainBoard – ESC – Motor 체결       3. MainBoard – Telemetry 체결       4. MainBoard – GPS 체결       5. 모듈 - 드론 프레임 체결       6. Battery 체결    2. F/W 개발       1. FC F/W 개발 또는 기존 오픈소스 활용       2. IMU 센서 Calbration 코드 적용       3. 배터리 잔량 계산 코드 적용 (전압분배 회로 적용)       4. telemetry통해 원격 데이터 송 수신 packet 구조 설계 및 개발       5. GCS로부터 좌표 데이터 수신 기능 확인       6. 좌표 데이터 수신 후 좌표 설정하여 이동하는 기능 개발       7. 랜딩 자동화 기능 개발   7.3  7.4   1. GCS 개발    1. 하드웨어 납땜 및 체결       1. GCS 프레임 (= 기구물)          1. Ctrl Board 연결          2. Joystick 연결          3. Mission Button 연결          4. Telemetry 연결       2. Ctrl Board 배선   8.1.2.1   * + 1. ㄴ     2. ㅇ   1. F/W 개발   8.2.1  8.2.2   1. Station 개발  * Last Update 2024.09.09 |

❑ **개발 내용 요약**

|  |
| --- |
| 1. 자동 랜딩 기능을 탑재한 드론 개발 2. 드론 운용이 가능한 조종장치 개발 3. 랜딩 (착륙) 을 도와줄 Station 개발 4. 수행 능력 평가 후 실제 효과가 있는지 검증 및 보완 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 프로젝트명 | 무인기 정밀 착륙 유도 장치 개발 | | | 연구기간 | | 2024.09.02~09.09 |
| 제목 | 1. 시스템 구성도, 설계, 동작 흐름 | | | | | |
| **목적**  개발 전, 시스템 구성과 설계, 동작 시나리오 등을 작성하여 이후 과정에서 놓치는 부분 없이 원활히 진행하기 위함.  또한 시스템 구성을 사전에 하게 되면 호환성 검증 등을 하면서 구조 개선이 이루어질 수 있음.  **내용**   1. 시스템 구성도, 설계   동작 흐름 설명  1. 기본적으로 Drone – GCS 간 특정 프로토콜에 기반하여 각각의 데이터를 송수신합니다.  (데이터 종류: 드론 배터리 잔량, 드론 3축 값, 송수신 시간 등. (원격 통신 매개체: RF Module))  2. 사용자가 GCS의 지정된 자동 랜딩 버튼을 물리적으로 누릅니다.  (해당Push lock 버튼이 눌러진 상태에선 자동 랜딩, 한 번 더 눌러 버튼이 올라오면 그 위치에서 호버링)  3. Landing Station은 GCS 상에서 버튼을 누르든 말든 지속적으로 GPS 데이터를 Drone 측에 송신해주며 Beacon이 밝게 빛납니다.  (GPS 데이터 송신, Beacon 동작은 Landing Station에 부속된 버튼에 의해 동작합니다)  4. 드론은 BLE를 통해 해당 GPS 데이터를 수신하여 해당 지점으로 Setpoint 되어 이동합니다.  5. 해당 지점으로 이동할 시 가까워지면 Drone 은 Beacon의 빛을 IR lock 센서를 통해 인식하여 정밀 착륙이 가능하게 됩니다. | | | | | | |
| 기록자 | | 김교원 | 점검자 | | 박정은 | |
| 서명 | | (sign) | 서명 | | (sign) | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 프로젝트명 | 무인기 정밀 착륙 유도 장치 개발 | | | 연구기간 | | 2024.09.09~09.18 |
| 제목 | 1. 각 시스템 파트별 제품 선정 및 임의 견적 | | | | | |
| **목적**  전체 시스템 구성 후, 파트별로 제품 선정을 진행한다.  가격, 성능, 호환성 등을 고려하여 완제품형태의 구성품들을 나열한다.  해당 과정을 통해 얻을 수 있는 이점은 아래와 같습니다.  1. 미리 모듈별 기능, 성능 등을 파악하여 추후 재선정 또는 기능 이슈 등을 방지합니다.  2. 임의 견적을 통해 비용 절감을 할 수 있습니다. (예산 초과 시 대치품 탐색을 하게 됨)  3. 모듈의 오버 스펙을 방지합니다. (전체 구성을 생각하게 되어 다른 모듈과 어울리는 적정 스펙의 제품을 찾게 됨)  [참고 자료]  Capstone\_Design\1.PrecisionLandingModule\1.ResearchNote의 ResearchNoteData.xlsx파일의 “제품선정자료” sheet참조  Capstone\_Design\1.PrecisionLandingModule\1.ResearchNote의 ResearchNoteData.xlsx파일의 “FC 선정 자료” sheet참조 | | | | | | |
| 기록자 | | 김교원 | 점검자 | | 박정은 | |
| 서명 | | (sign) | 서명 | | (sign) | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 프로젝트명 | 무인기 정밀 착륙 유도 장치 개발 | | | 연구기간 | | 2024.09.09~09.18 |
| 제목 | 1. Git Repository 생성 | | | | | |
| **목적**  단순히 최종 산출물이 아닌, 연구 개발 history를 모두 확인하기 위해.  추후, 자료 공유 시 local보단 원격 저장소를 활용하면 좋기 때문입니다.  Repository에 Folder Tree를 정의해두면 자료 정리하기 용이함.  [참고 자료]  <https://github.com/kimkyowon/Capstone_Design.git> (Repository 경로입니다)  <Folder Tree 이미지, 아래 설명의 (1)은 최상위 폴더부터 하위 폴더로 갈수록 ()안의 숫자가 1씩 늘어남>    (1) Capston\_Design : 가장 상위 폴더입니다.  (2) PrecisionLandingModule : Capstone\_Design의 프로젝트로, Capston\_Design의 하위 폴더입니다.  (3) ResearchNote : 연구 노트와 작성 시 활용된 자료들을 보관하는 곳입니다.  (3) Version\_1.0 : 실제 제작할 제품 관련한 자료들이 들어가는 곳입니다. 초안이기에 Version 1.0으로 지칭하였습니다.  (4) SystemConfiguration : Version1.0제품의 전체 SystemFlowChart등 전체 구성에 관한 자료가 들어갑니다.  (4) Drone : Version 1.0의 Drone에 대한 자료가 들어가는 폴더입니다.  (4) GroundControlStation : Version 1.0의 GroundControlStation (GCS)에 대한 자료가 들어가는 폴더입니다.  (4) LadingStation : Version 1.0의 LadingStation에 대한 자료가 들어가는 폴더입니다.  (5) Src : 개발 시 들어가는 FW src 가 들어가는 공간입니다. (Drone, GroundControlStation 도 동일 구성)  (5) Document : 개발 시 참고해야 할 DataSheet, SpecSheet 등의 자료가 들어가는 공간입니다.  (5) Release(BinaryOrHex) : 개발 후, 최종 FW Binary 또는 Hex 파일이 들어가는 공간입니다. | | | | | | |
| 기록자 | | 김교원 | 점검자 | | 박정은 | |
| 서명 | | (sign) | 서명 | | (sign) | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 프로젝트명 | 무인기 정밀 착륙 유도 장치 개발 | | | 연구기간 | | 2024.09.09~09.18 |
| 제목 | 1. 자재 수급 가능 여부 및 납기일 파악 | | | | | |
| **목적**  선정한 자재들의 수급 가능 여부를 파악하여 의도치 않은 상황을 방지한다.  수급 여부 파악과 동시에 납기일도 알 수 있어 추후 일정 계획을 세울 수 있다.  해당 과정을 통해 얻을 수 있는 이점은 아래와 같습니다.  1. 수급 가능 여부를 파악해두면 추후 나중에 문제가 생겨 교체품 구매시에 수급이 안되는 등의 문제점을 방지할 수 있습니다.  2. 납기일을 파악하여 추후 일정 계획 수립에 도움이 됩니다.  3. 전문 전자부품업체와 의논할 시 대치품 파악에도 도움이 됩니다. (수급이 어려우면 대치품을 찾아주는 업체가 있습니다.)  4. 일반적으로 해외 제품이 많은데, 이의 경우 납기일이 굉장히 김 (통관 절차가 복잡해짐, 수급 업체와 해외 업체의 휴무일을 모두 고려해야 함 등) 따라서 국내 업체에 미리 말해두거나 보유하고 있는 국내 업체를 찾아야 함.  [컨택할 업체 목록]   * ㈜플랜비넷 * DeviceMart * Digikey * 루나볼트 * 가치창조기술   [참고 자료]  Capstone\_Design\1.PrecisionLandingModule\1.ResearchNote의 ResearchNoteData.xlsx파일의 “제품선정자료” sheet참조 | | | | | | |
| 기록자 | | 김교원 | 점검자 | | 박정은 | |
| 서명 | | (sign) | 서명 | | (sign) | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 프로젝트명 | 무인기 정밀 착륙 유도 장치 개발 | | | 연구기간 | | 2024.09.09~09.18 |
| 제목 | 6-1. 모듈 단위 검증 – FC 검증 | | | | | |
| **목적**  드론의 핵심 부품인 FC를 중점으로 여러가지 모듈이 장착됩니다.  따라서 다른 모듈을 검증하기 전, FC의 검증이 선수되어야 하며  기본 동작이 검증된 시점에 initializing을 위한 파라미터 세팅을 진행합니다.   1. Mission Planner 연동  * 해당 PC앱을 연동함으로서, FC의 각종 모듈 값을 모니터링 할 수 있습니다.   (통신 속도, FC 내부의imu 등 센서 값 등)  V1.3.82 앱 다운로드 경로 (<https://ardupilot.org/planner/docs/mission-planner-installation.html>)   * 통신 방식은 RF 모듈을 활용해야 하지만, 아직 오지 않은 관계로 유선(Serial)방식을 사용해 검증하였습니다.     (PC와 FC를 유선 연걸 후, 좌측 빨간 박스에서 port 선택 후, 우측 연결 버튼을 누르면 연결됩니다)   * 주요 항목 간략한 소개      1. [데이터]: 지금 보이는 화면을 통해 드론의 데이터를 수집할 수 있습니다. 2. [계획]: 경로 지정을 통해 항로 설정이 가능합니다. 3. [설정]: 드론에 부착된 모듈등의 하드웨어를 보정하거나 수정할 수 있습니다. 4. [구성]: 이곳에서 FC의 설정값을 수정할 수 있습니다. (Initializing을 위한 Parameter 세팅 가능) 5. [모의 시험]: 실제 드론 없이 모의 비행이 가능합니다. 6. [비행화면]: FC에 부착된 IMU센서를 통해 기체 각도를 표현한 이미지, 실시간으로 변화합니다. 7. [개요]: 기본적인 비행 데이터를 제공해줍니다 (고도, 위도Yaw,등등)   (3) [명령]: 제공되는 API에 한해, 각종 모듈에 Command를 전송할 수 있습니다.  (3) [메시지]: PC앱과 FC간의 데이터 송수신을 모니터링 할 수 있습니다.  (3) [사전 점검]: 비행 전 상태를 알 수 있도록 사용자 지정 값 이상일 경우 비행이 가능합니다.  추후 모듈들이 도착하면 해당 기능을 통해 검증 예정입니다.     1. FC 데이터 모니터링     FC 자체에 IMU센서가 있는데, 해당 센서는 초기 부팅 시 FC의 수평 각에 맞춰 0으로 됩니다.  따라서 초기 부팅 시, 책상위에 수평을 유지한 채로 Mission planner에 연동 시 오차 없이 수평으로 나오면 정상입니다.  이후, 직접 FC를 기울여보며 테스트해보시면 됩니다.  테스트 영상은 아래 경로에 업로드 하였습니다. “Capstone\_Design\1.PrecisionLandingModule\2.Version\_1.0\1.Drone\2.Document\2.DataSheet\_SpecSheet\1.FC”   1. PWM 테스트   PWM는 Pulse width modulation의 약자로, 쉽게 말해서 신호의 길이를 조절하는 것입니다.  해당 OrangeCube FC는 해당 PWM출력 포트를 통해, 사용자가 mission trigger를 쓸 수 있도록 구성되어 있지만  사실 PWM 신호 자체는 정확하지 않기 때문에 (신호 길이가 ms단위로 딱 맞기도 어려울 뿐더러 일반적인 신호는 on,off 처럼 이분법적으로 나뉘어야 하지만 적게는 1ms ~ 2000ms처럼 나올 수 있는 신호 범위가 크기 때문)  일반적으로 선호하는 방식은 아니긴 합니다.  여튼, 이를 검증하기 위해 이전에 작성하여둔 MyPwmFilter 코드를 활용해 FC가 출력하는 PWM값이 정확한지 측정하였습니다.  (아래는 직접 작성한Pwm Filter library)    Capstone\_Design\1.PrecisionLandingModule\2.Version\_1.0\1.Drone\2.Document\2.DataSheet\_SpecSheet\1.FC 경로에서 “PwmTestCode\_Arduino.zip” 을 확인해보시면 됩니다.  (아래는 테스트 영상의 일부 사진)    테스트 코드 설명: “**우측 사진”**을 보면 “저, 중, 고”, “전환”, “숫자”, “pwm최소값”, “pwm최대값” 형태인데.  각 파트를 설명하자면 아래와 같습니다.   1. “저, 중, 고”는 Pwm의 최소, 중간, 최대 값 출력버튼을 의미합니다. 2. “전환”은 자동으로 “저, 중, 고”를 변환해줍니다. 3. “숫자”는 FC의 포트 번호입니다. 여기서 9는 “AUX OUT”포트의 1번입니다. 4. “pwm 최소값”은 말 그대로 pwm의 최소값입니다. 사진처럼 1000이 최소값이며 “저”를 누르게 되면 1000이 출력됩니다. 5. “pwm 최대값”은 말 그대로 pwm의 최대값입니다. 사진처럼 2000이 최대값이며 “고”를 누르게 되면 2000이 출력됩니다.   동작에 대한 설명은 아래와 같습니다.   1. 최상단 사진인 VS code를 보시면, 핀 할당 표가 있습니다. 말 그대로 arduino의 pin 할당 표입니다. 2. pin\_switch가 1000이면 pin\_a로 들어오는 pwm수치를 파악해 변화하면 pin\_a의 pwm의 값을 출력해줍니다. 3. pin\_switch가 1500이면 pin\_b로 들어오는 pwm수치를 파악해 변화하면 pin\_b의 pwm의 값을 출력해줍니다. 4. pin\_switch가 2000이면 pin\_fc로 들어오는 pwm수치를 파악해 변화하면 pin\_fc의 pwm의 값을 출력해줍니다. 5. Pwm 수치가 기존과 다를 시, 좌 하단 사진의 Arduino Serial Monitor에 출력되도록 하였으며, 데이터 형태는 아래와 같습니다.     [시간] -> [핀번호] [이전값] ===> [현재값] [날값]  **시간은** 말 그대로 serial print가 출력된 시간입니다.  **핀 번호는** 변화된 pwm값이 들어온 번호입니다. (예시 : sw값이 변하면 sw값인 0이)  **이전 값은** 말 그대로 이전 pwm값입니다.  **현재 값은** 말 그대로 현재 변화되어 들어온 pwm값입니다.  **날 값은** 앞서 설명드린 것 처럼, pwm값은 딱 맞게 들어오지 않기 때문에, 보정되지 않은 값을 의미합니다.  (현재 코드에서 보정 범위는 100ms입니다)  테스트 영상은 아래 경로의 “TestVideo.zip”폴더에 존재합니다.  Capstone\_Design\1.PrecisionLandingModule\2.Version\_1.0\1.Drone\2.Document\2.DataSheet\_SpecSheet\1.FC  테스트 과정이 모두 기재되진 않았지만, 위 테스트를 통해 얻은 결론은 아래와 같습니다.   1. 현재 보유중인 FC는 AUX OUT의 1,2,3,4만 동작함. 2. PWM Filtering을 위한 오차범위는 100이면 충분함. 3. 장시간의 테스트에도 FC의 PWM출력은 정상적으로 동작함.   [참고 자료]  Capstone\_Design\1.PrecisionLandingModule\2.Version\_1.0\1.Drone\2.Document\2.DataSheet\_SpecSheet\1.FC 경로의  “TestVideo.zip” 과 “PwmTestCode\_Arduino.zip” | | | | | | |
| 기록자 | | 김교원 | 점검자 | | 박정은 | |
| 서명 | | (sign) | 서명 | | (sign) | |