

Xilinx Zynq FPGA, TI DSP, MCU 프로그래밍 및 회로 설계 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

Radar Based Quadcopter GSD

Introduction

이 문서는 비행을 시작하기 위한 최소한의 설정 과정을 안내한다.

Bill of Materials(BOM – 구성 부품)

- 1 Airframe(기체 프레임)
- 4 Landing Gear Legs(랜딩 기어)
- 1 RC 조종기
- 4 프로펠러
 - CCW 2개
 - CW 2개
 - 16 개의 육각 볼트(설치용 2 개, 프로펠러 백업용 2 개)
- 1 CAN Hub(Airframe 에 부착)
- 1 USB Ground-Station Telemetry Unit(USB 타입의 지상 관제탑과의 통신 유닛)
- 1 Antenna for Airframe Telemetry(기체 자체에 통신을 위한 안테나)
- 2 Velcro Straps(벨크로 스트랩 – 찝찝이)
- 1 Self-adhesive Velcro Section(벨크로 고정형)
- 1 Allen Wrench(앨런 렌치)
- 1 SD Card

Assembly

1. 상자에서 Airframe 을 꺼내고 접혀 있는 4 개의 팔을 편다.
이들은 단단히 눌러 제자리에 고정해야한다.
2. 고정시킬 다리, 통신용 모듈 및 기타 악세사리를 박스에서 꺼낸다.
3. 랜딩 기어 다리를 기체의 모터가 올 위치 아래 박고 고정한다.
4. GPS 스탠드를 올리고 브라켓을 조여서 고정한다.

참고: GPS 의 화살표가 항공기의 전면부를 나타내며 아래를 향한 레이더는 뒤를 나타낸다.

5. 통신용 안테나를 기체 후면에있는 기체 원격 측정 모듈에 연결한다.
안테나를 가장 가까운 보강재와 같은 방향으로 각을 잡도록 한다.
6. ESC 와 모터에 대한 연결을 수행한 이후에 프로펠러 팩을 분리하고 프로펠러를 아래와 같은 방법으로 조인다.

Motor Position	Orientation
Left – Front	CW
Right – Front	CCW
Left – Back	CCW
Right – Back	CW

7. 자가 접착식 벨크로를 배터리 윗면에 붙이고 이것을 항공기 밑면의 벨크로에 단단히 붙인다.
8. 나머지 벨크로 끈을 사용하여 배터리를 항공기에 단단히 고정시켜야 한다.
9. 마이크로 SD 카드를 OcPoC 의 마이크로 SD 카드 슬롯에 넣고 딸깍 소리가 날 때까지 누른다.
10. 배터리의 XT60 커넥터를 기체에 연결한다.
11. 조종기 throttle (왼쪽 기본 스틱)을 아래쪽 위치로 놓고 조종기를 구동한다.
12. 이제 항공기는 지상 기지국에 연결할 준비가 되었다.

참고: 배터리가 연결되면 Ardupilot 이 자동으로 시작된다.

조종기가 조종사의 손에 있는지, throttle 이 아래쪽에 있는지, 어린이, 애완동물, 귀중품, 나무가 무인 항공기 근처에 있는지 확인해야 한다.

























Connection to Ground Station

참고: 초기 캘리브레이션이 완료된 후, 조종기가 켜져있을 때 무인 항공기가 이륙 할 수 있으므로 프로펠러에서 멀리 떨어져 있어야한다!

지상의 기지국 SW 를 실행한다.

Mission Planner 에 연결하기 위해:

1. 지상 기지국 전용 통신 모듈을 지상 기지국 USB 포트에 연결한다.
2. Windows 라면 장치 관리자를 열도록 한다.
3. 포트(COM & LPT) 섹션을 연다.
4. Silicon Labs CP210x USB to UART Bridge 를 찾아 COM 번호를 기록한다.
5. Mission Planner 에서 Baud 에 57600 과 앞서 기록한 COM 번호를 입력한다.
6. Connect 버튼을 누른다.

참고: QGroundControl 을 사용하는 경우 SW 가 원격 측정 모듈을 자동으로 감지 할 수 있다.

Calibration & Controller Setup

자동 조종 장치가 작동하기 전에 Mission Planner 의 초기 설정 메뉴에서
관성 측정 장치 (IMU), 자이로, 자력계, RC 송신기 및 비행 모드를 보정하는 단계를 따르도록 한다.

참고: 컨트롤러 설정 중에 스위치를 지정하여 충돌 회피를 활성화 할 수 있다

Flight

비행을 시작하려면 무인 항공기가 준비되어 있어야 한다:

1. Throttle 을 아래로 잡고 오른쪽으로 Yaw 를 넣는다.
2. 3 초를 기다린다.
3. 이제 Yaw 를 멈추면 비행 준비가 완료 된 것이다.

동작을 중단하려면:

참고: 드론이 지상에 있는지 확인한다!

1. Throttle 을 아래로 잡고 왼쪽으로 Yaw 를 넣는다.
2. 3 초간 기다리면 드론의 동작이 중단 됨에 따라 프로펠러가 느려지기 시작한다.

Missions

임무와 관련된 비행을 수행 하려면 PX4 또는 Ardupilot

및 해당 Ground Control Station (예 : QGroundControl 또는 Mission Planner) 문서를 참조하여 자율 비행 임무를 준비하라.

Developing for OcPoC(User Hub)

Aerotenna User Hub 는 비행 소프트웨어 개발, OcPoC 구성, 자체 Linux 커널 생성, ROS (Robot Operating System) 설치 등과 같은 OcPoC 을 처음부터 시작하는 절차에 대한 깊이있는 정보를 제공한다. User Hub 는 지속적으로 업데이트되므로 자주 확인 하라.
Aerotenna User Hub 는 다음에서 찾을 수 있다:

<http://aerotenna.readme.io/>

Flight Software

OcPoC Flight Controller 는 Ubuntu 14.04 와 함께 Linux 4.0.0 의 수정 된 버전을 구동한다.
PX4 및 ArduPilot 모두 개발 키트에 포함 된 SD 카드에서 구동 할 수 있다.

- ArduPilot
ArduPilot 은 Aerotenna 가 지원하는 두 가지 오픈 소스 비행 스택 중 하나다.
이것은 GPL-3 라이선스와 많은 코드 개발자들의 커뮤니티다.
아래 링크에서 Ardupilot 커뮤니티에 접근 할 수 있다:

www.ardupilot.org

- PX4
PX4 는 Aerotenna 가 지원하는 다른 오픈 소스 비행 스택이다.
이것은 관대한 BSD 라이선스를 가지고 있다.
아래 링크에서 PX4 커뮤니티에 접근 할 수 있다:

www.px4.io

Configuring the OcPoC

- Requirements

OcPoC 를 구성하려면 아래 사양이 필요하다:

- Vivado 2017.1 이상의 버전들

- Petalinux SDK

- 아래 Git Repo 의 클론들:

- Linux-Xlnx: <https://github.com/Xilinx/linux-xlnx>

- U-Boot-Xlnx: <https://github.com/Xilinx/u-boot-xlnx>

- Device-Tree-Xlnx: <https://github.com/Xilinx/device-tree-xlnx>

OcPoC 프로젝트를 처음부터 만드는 경우에는 Xilinx Vivado 의 기본 사항을 알고 Linux 커널 빌드 시스템을 잘 이해해야 한다.
더 자세한 정보는 Aerotenna User Hub 에서 찾을 수 있다.