

Flying OcPoC Zynq with Ardupilot / ArduCopter

The following documentation will guide you through these steps:

Download the the appropriate files for OcPoC flight controller and ArduPilot's autopilot.

OcPoC, ArduPilot오토파일럿을 작동시키는데 필요한 파일들을 다운로드합니다.

Unpack and install Ubuntu file sytem on the flight controller.

압축을 풀고, 플라이트 컨트롤러에 설치된 우분투 파일 시스템에 인스톨합니다.

Build the ArduCopter executable.

ArduCopter를 빌드합니다.

Load the autopilot onto the OcPoC Zynq.

AutoPilot을 OcPoC에 로드합니다.

Connect the OcPoC Zynq to Mission Planner.

OcPoC Zynq를 Mission 플래너에 연결합니다.

If you get stuck along they there is a link the to the OcPoC Zynq FAQ page at the bottom of every step.

하다가 잘 안되는 것들은 FAQ 페이지를 참조하세요.

If you cannot find the answer to your problem there, 그럼에도 불구하고 적절한 해답을 찾지 못하면

feel free to post in the 'Support and Discussion' section of the User Hub,

and we'll get back to you as soon as we can.

서포트&디스커션에 글을 올려주세요.

Begin by making sure that you have the required tool chains for the installation. Open a terminal and execute the following commands.

설치에 필요한 툴체인이 제대로 설치되어 있는지를 확인하세요. 터미널을 열고 다음 명령어를 입력하세요

Required Tools:

```
sudo apt-get update  
sudo apt-get install git  
sudo apt-get install python-pip  
sudo apt-get install gtkterm  
sudo apt-get install gcc-arm-linux-gnueabi  
sudo apt-get install g++-arm-linux-gnueabi  
sudo apt-get install pkg-config-arm-linux-gnueabi  
pip install future
```

설치되는 툴들

:소스 관리를 위한 git

:파이썬 기반의 패키지 관리 툴

:GTKterm

GTKTerm is a graphical serial line terminal emulator that allows you to take control of your board through the Serial line. With it you can have access to the U-Boot/Linux console of your APF without needing to connect a keyboard/screen to the board.

시리얼 통신을 통하여, 보드를 제어할 수 있도록 해주는 터미널 애플레이터

이 툴을 사용하여 U-Boot와 리눅스 콘솔에 대한 제어를 할 수 있도록 해줌.

gnueabihf의 의미.

hf means Hard Float.

When you use this flag, during compilation, your compiler will compile your program so that it will use the Floating Point Unit once launched.

Your program will run faster but only if you use floating points!

Be careful not to use this flag if your CPU doesn't support it, or it will not start.

the gnueabi stands for the compilation of code for linux

즉, 하드플로트 연산을 지원하는 cpu를 위한 컴파일러를 의미합니다.

또한 리눅스를 위한 컴파일러가 되겠습니다.

Now we will download the necessary files for the file system and the autopilot

오토파일럿 파일 시스템에 필요한 파일들을 다운로드 합니다.

Boot Files: 부팅과 관련된 파일들

```
git clone https://github.com/Aerotenna/OcPoC_Mini_Zynq_Files.git
```

Autopilot Software: 오토파일럿 소프트웨어(제어프로그램)

```
git clone https://github.com/Aerotenna/APM_OcPoC_Zynq.git
```

This will put the directories 'OcPoC_Mini_Zynq_Files' and 'APM_OcPoC_Zynq' on your computer. Make sure you note where these directories are on your system.

컴퓨터 'OcPoC_Mini_Zynq_Files' 와 'APM_OcPoC_Zynq'폴더에 파일들이 저장될 것 입니다.

위 두 폴더(디렉터리)가 제대로 생성되었는지 확인하세요.

Once you have executed the above commands, you are ready to proceed to the next section 'Load Files onto SD Card'.

위 명령어 대로만 제대로 입력하였고, 특별한 이슈가 발생하지 않았다면, "SD 카드에 파일을 라이팅'하는 챕터로 넘어가십시오.

In a terminal navigate to the folder where you cloned 'OcPoC_Mini_Zynq_Files' and execute the following commands.

OcPoC_Mini_Zynq_Files 디렉터리에 가서, 다음 명령어를 실행하세요.

Copy files to SD card: 파일들을 SD카드에 복사합니다.

```
cp ulmage devicetree.dtb uramdisk.image.gz BOOT.bin /media/<your_username>/boot/
```

```
cd Ardupilot_rootfs/
```

```
sudo cp ubuntu-rootfs1.tar.gz ubuntu-rootfs2.tar.gz /media/<your_username>/rootFs/
```

Unpack Ubuntu to SD card:

```
cd /media/<your_username>/rootFs
```

```
sudo tar xvf ubuntu-rootfs1.tar.gz
```

```
sudo tar xvf ubuntu-rootfs2.tar.gz
```

Remove the tar directories:

```
sudo rm ubuntu-rootfs1.tar.gz
```

```
sudo rm ubuntu-rootfs2.tar.gz
```

Make sure decompression was successful:

```
ls
```

You are now ready to compile the autopilot!

오토파일럿 컴파일 준비가 되었네요.

Build ArduPilot Executable(오토파일럿 실행파일 빌드)

Now we have to compile the ArduPilot code for the OcPoc-Mini. In terminal navigate to the where you downloaded the ArduPilot source code and enter the ArduCopter directory. (the OcPoc-Mini)에 작동될 the ArduPilot 코드를 컴파일 해야합니다. 터미널에서 ArduPilot 소스코드를 저장한 디렉터리로 이동 하세요.

Now that you are in the correct directory, first initialize and update the git submodules, then compile the executable with the following commands. 제대로 디렉터리를 찾아 들어오셨으면, 일단 git 서브 모듈들을 초기화하고 업데이트 해주고나서, 실행파일들을 다음 명령어를 입력하셔서 컴파일 하세요.

Building autopilot:

```
git submodule update --init --recursive
```

(서브모듈의 이해 : <http://ohgyun.com/711>)

```
Tools/scripts/install-prereqs-ubuntu.sh -y
```

(우분투 기반의 쉘 스크립트(실행파일이라 이해하면 편함)를 실행.

쉘 스크립트에 대한 이해는 다음 사이트 참조 : <https://help.ubuntu.com/lts/serverguide/backup-shellscripts.html>)

```
./waf configure --board ocpoc_zynq
```

```
./waf --targets bin/arducopter
```

Copying ArduCopter executable to SD card: "실행파일"을SD카드에 저장.

```
cd build/ocpoc_zynq/bin/
```

```
sudo cp arducopter /media/<your_username>/rootFs/root/
```

Note: It is important to get the 'arducopter' file into the the 'root' directory and not anywhere else

Arducopter파일을 root 폴더에 넣어야만 합니다.

Now we have all of our files loaded onto the SD card. You can eject the SD card and insert it into it's port on the OcPoC Mini and continue to Running ArduCopter on OcPoC Zynq Mini

이제 필요한 모든 파일을 sd카드에 저장되었고, SD카드를 제거하고, 그것을 OcPoC에 삽입하고

OcPoC ZYNQ에서 ArduCopter를 실행합니다.

Before You Begin

First make sure you have done the following.

다음의 것들이 제대로 되었는지 먼저 확인해주세요.

SD Card inserted into your OcPoC™ Zynq Mini

SD카드는 제대로 OcPoC ZYNQ에 삽입되었는지

Connect battery power to your OcPoC™ Zynq Mini - see "Powering your OcPoC Board" [Note: this is not necessary for bench testing with OcPoC as the USB connection will supply 5V to the board]

배터리를 제대로 OcPoC에 연결하였는지.-자세한 전원과 관련된 레퍼런스는 Powering your OcPoC Board 문서를 참조.

Connect a MicroUSB cable to the 'USB1' port on your OcPoC™ Zynq Mini, then connect to your computer via USB.

마이크로 USB 케이블을 OcPoC의 USB1 포트에 연결하고, USB를 컴퓨터와 연결합니다.

Connecting to your OcPoC™ for the first time

Once you have a USB to Micro-USB connected from your PC to the specified USB UART(USB1) and at least 5 volts DC supplied to the board. Open a terminal on your PC and execute the following command

USB케이블을 통해, OcPoC보드와 컴퓨터가 연결되고 나면, 터미널을 열고 다음 명령어를 입력하세요.

Note: See FAQ: OcPoC™ Zynq Mini for connecting to OcPoC from Windows

Open GTK:

```
sudo gtkterm
```

Inside GTK Term:

Ctrl + Shift + S

A 'Configuration' window will pop up. 설정 창이 뜰 것임.. Under the 'Port:' dropdown box select the USB port your OcPoc Zynq Mini is connected to(usually the last on list). 포트 밑, 드롭다운 박스에서 OcPoC Zynq가 연결된 컴퓨터 USB 포트를 선택하세요.

On the 'Baud Rate:' drop down select the option '115200' and click the 'ok' button. :

보레이트 드롭박스에서 115200을 선택하세요.

At this point we recommend reboot your board so that you may see the boot-up process. Simply disconnect then reconnect the board power. 이쯤에서 보드를 재부팅 시키는 것을 권장합니다. 그냥 전원 코드를 뽑고 다시 꽂으세요.

You should now see the boot process scrolling across your GTK Terminal

이제 GTK 터미널에서 스크롤 되며 쓰여져 나가는 부트프로세스(부트 로그)를 확인할 수 있을 것입니다.

Simply provide 'root' as the login in credentials.

Login:

root

Now we will execute the auto pilot. Depending on how you will connect to your Ground Station(GS) software you will pass a different argument. The most common method is to use wireless telemetry.

이제 오토파일럿 프로그램을 실행하고, 그라운드 스테이션을 어떻게 연결했느냐에 따라 다른 아규먼트들을 확인하게 될 것입니다. 가장 흔한 메소드는 무선 텔레메트리를 쓰는 것이지요.

Connect to GS via Wireless Telemetry:

```
./arducopter -C /dev/ttyPS1 -B /dev/ttyS3
```

Connect to GS via USB:

```
./arducopter -A /dev/ttyPS0 -B /dev/ttyS3
```

To enable other peripherals, use the following startup options:

Peripheral Device/Function	Startup Argument
----------------------------	------------------

Telemetry via micro-USB	
-------------------------	--

-A /dev/ttyPS0	
----------------	--

Wireless Telemetry	
--------------------	--

-C /dev/ttyPS1	
----------------	--

GPS/Compass	
-------------	--

-B /dev/ttyS3	
---------------	--

uLanding	
----------	--

-E /dev/ttyS6	
---------------	--

uSharp

-F /dev/ttyS0

(Optional) Enabling ArduCopter Auto Execution

If you would like ArduPilot to start automatically during the boot up process of the OcPoC™, without connecting via USB and logging in over gtkterm, you will first need to create a blank text file named 'ardupilot' in the SD card's 'rootFs/etc/init.d/' directory. With the SD card connected to your PC, execute the following commands:

팁: ArduPilot 프로그램이 시스템 부트 이후에 자동적으로 실행하길 원하면, 'rootFs/etc/init.d/' 디렉터리에 ardupilot 라는 빈 파일을 만들고, 그 내부에 다음 기술하는 명령어를 넣으면 됩니다.

Create ardupilot Script:

```
cd /media/<user_name>/rootFs/etc/init.d/
```

```
sudo touch ardupilot
```

Open the 'ardupilot' file in your preferred editor (be sure to open as a super user, i.e. 'sudo gedit ardupilot'), and copy the following text into the file: ardupilot파일을 적당히 사용자가 원하는 에디터(예: vi)로 열어서

다음 글자들을 복사해 붙여넣으면 됩니다.

Note: This text is specifically for the Ubuntu rootFs setup for OcPoC™, and will run ArduCopter. If you're running ArduPlane, set 'ARDUPILOT' equal to 'arduplane' as opposed to 'arducopter'. You can also add any other peripheral options in the 'STARTCMD' variable (i.e. you could change to STARTCMD="/root/\$ARDUPILOT -C /dev/ttyPS1 -B /dev/ttyS5 -F /dev/ttyS0"). 이 부분부터는 OcPoC에 우분투 루트 파일 시스템을 설치하기 위한 작업입니다.

혹시 ArduPlane을 실행하고 있다면, ARDUPILOT을 arduplane과 같도록 설정해야합니다.

또한 다른 페리페럴 옵션들을 STARTCMD 변수에 추가해주어야 합니다.

Contents of ardupilot Script:

```
#!/bin/sh
```

```
#### BEGIN INIT INFO
```

```
# Provides:          ArduCopter
```

```
# Required-Start:
```

```
# Required-Stop:
```

```
# Default-Start:     2 3 4 5
```

```
# Default-Stop:      0 1 6
```


Short-Description: ArduCopter Autopilot Daemon

Description: starts ArduCopter Autopilot

END INIT INFO

PIDFILE=/var/run/ardupilot.pid

피드 파일이란????????????????????

ARDUPILOT="arducopter"

STARTCMD="/root/\$ARDUPILOT -C /dev/ttyPS1 -B /dev/ttyS3"

각종 변수 선언 및 초기화=각각의 명령어들을 저장하는 것.

start(){

/var/run/ardupilot.pid과 "arducopter"같으면 에러 처리.

if [-f \$PIDFILE] && pgrep \$ARDUPILOT; then

echo 'Service already running' >&2

return 1

fi

echo 'Starting service...' >&2

local CMD="\$STARTCMD &" 다시한번 명령어 Parse하는 부분..

&은 쓰레드 추가해서 실행시키는 것..

su -c "\$CMD" root > "\$PIDFILE" 실행하는 부분이 여기!

즉 su -c "/root/\$ARDUPILOT -C /dev/ttyPS1 -B /dev/ttyS3 root > pidfile.. 과 같겠음.

su는 알겠는데, -c ??????????

아무튼

echo 'Service started' >&2

}

stop(){

피드파일 내용이 없거나, ARDUPILOT이 없으면..

if [! -f \$PIDFILE] || ! pgrep \$ARDUPILOT; then

```

        echo 'Service not running' >&2

        return 1

    fi

    echo 'Stopping service...' >&2

프로세스를 킬건데, 어디 정의된 프로세스냐면
$ARDUPILOT이고 , 파일 삭제할건데, $PIDFILE임..

    pkill $ARDUPILOT && rm -f "$PIDFILE"

    echo 'Service stopped' >&2

}

```

Actions

```

case "$1" in
    start)

        start

        ;;

    stop)

        stop

        ;;

    restart)

        stop

        start

        ;;

    esac

exit 0

```

Save and close the 'ardupilot' file, and make sure it is executable with the following command:

Ardupilot 파일을 저장하고, 닫고, 그 다음 이 파일이 다음 명령어를 통해 실행권한을 가지고있는지 확인해야 합니다.

Set Permissions for ardupilot Script:

```
sudo chmod 755 ardupilot
```

Next, with the SD card back in the OcPoC™, power on the board and login via gtkterm, then run the following command: 다시 SD 카드 이야기로 돌아와서... 보드에 전원을 넣고, GTKTERM을 통해 로그인하고 다음 명령어를 실행합니다.

Install ardupilot Script on OcPoC™:

```
update-rc.d ardupilot defaults
```

```
reboot
```

Now during the normal bootup process of the OcPoC™, ardupilot will automatically start without the need to connect via USB. You can also now run the following commands from the OcPoC™ terminal:

이제부터는 시스템이 부트하면서 ardupilot 프로그램이 자동적으로 실행될 것입니다.

New OcPoC™ Terminal Commands:

```
/etc/init.d/ardupilot stop
```

```
/etc/init.d/ardupilot start
```

```
/etc/init.d/ardupilot restart
```