Building in Ubuntu

Short version:

Assuming you have, say, a SPRACINGF3EVO board:

```
$ git clone git@github.com:cleanflight/cleanflight.git
$ cd cleanflight
$ make clean TARGET=SPRACINGF3EVO # only needed if you have old object files
lying around
$ make TARGET=SPRACINGF3EVO
.
(lots of output)
.
Linking SPRACINGF3EVO
./tools/gcc-arm-none-eabi-6-2017-q2-update/bin/arm-none-eabi-size
./obj/main/cleanflight_SPRACINGF3EVO.elf
   text    data     bss     dec     hex filename
 241598    7724   32896  282218   44e6a
./obj/main/cleanflight_SPRACINGF3EVO.elf
./tools/gcc-arm-none-eabi-6-2017-q2-update/bin/arm-none-eabi-objcopy -O ihex -
-set-start 0x8000000 obj/main/cleanflight_SPRACINGF3EVO.elf
obj/cleanflight_2.1.0_SPRACINGF3EVO.hex
```

If you have another board, just replace the TARGET by your board, e.g. for a Naze, you run `make TARGET=NAZE` instead of `make  TARGET=SPRACINGF3EVO`. If you are unsure what the correct name of your board is, use

```
$ make targets
```

to get a full list of supported targets.

The resulting .hex file can be found in the ./obj/ folder:

```
leco@Hutia:~/repositories/git/cleanflight$ ls -l obj/
-rw-rw-r--  1 leco leco 387709 Okt 28 00:00 cleanflight_SPRACINGF3EVO.hex
```

This is the one you need to supply to the flashing tool (for instructions how to use the Cleanflight Configurator to flash under Ubuntu, see the manual section 'USB Flashing'.

If you later want to build the same target again, it is good practice to clean up the old object files that are still lying around before building again:

```
$ make clean TARGET=SPRACINGF3EVO
```

Short version, part 2:

If you encounter a problem during compiling, it is likely caused by an incompatible ARM cross-compiler that was pre-installed on your Ubuntu. That is indicated by a message like

```
leco@Hutia:~/repositories/git/cleanflight$ make clean TARGET=SPRACINGF3EVO
make/tools.mk:296: *** **ERROR** your arm-none-eabi-gcc is '4.9.3', but '6.3.1'
is expected. Override with 'GCC_REQUIRED_VERSION' in make/local.mk or run 'make
arm_sdk_install' to install the right version automatically in the tools folder
of this repo.  Stop.
```

In this case, un-install the incompatible cross-compiler by

```
sudo apt-get remove gcc-arm-none-eabi
```

Then, install the correct version. This is easiest done by the cleanflight make target 'arm_sdk_install':

```
$ make arm_sdk_install
```

This will download the latest version of the cross-compiler (can take a while). After that is finished, you should be ready to go - just do `make TARGET=SPRACINGF3EVO' (put your correct target board name here, of course).

Long version:

The toolchain that is used for compiling a .hex file - the cross-compiler for ARM microcontrollers, called `arm-none-eabi-gcc`- is constantly maintained by ARM at https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads.

Different Ubuntu versions come with different versions (snapshots) of this cross-compiler, taken from the ARM website at different times. So, for example, the toolchain that comes with Ubuntu 16.04 is version 4.9.3, released in May 2015 while at the time of writing this article (October 2017), the most recent version on the ARM website was 6.3.1.

You can check your version by

```
arm-none-eabi-gcc --version
```

If any version of the ARM crosscompiler ("arm-none-eabi-gcc") is already installed on your system, Cleanflight will try to use it (in old Makefiles, before Cleanflight version 2.0.0) or will notice that it is not the latest version and refuse to work (in all newer Makefiles after Cleanflight version 2.0.0).

If you would like to compile an old version of Cleanflight (before 2.0), the Makefile will not download the current crosscompiler for you. It is easiest to use a Cleanflight version 2.x, and have the Makefile install it for you into the tools/ folder by `make arm_sdk_install`.
Then, checkout - with git - the old version of Cleanflight you want to compile, set your PATH variable to point to the bin/ subdirectory of the gcc-arm-none-eabi.../ folder inside tools/, and make your TARGET again (do not forget to "make clean TARGET=SPRACINGF3EVO" before making the final hex file).

Rather than using `make arm_sdk_install`, it is also possible to download the latest cross-compiler directly from the ARM website https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads. Extract the copy into the tools/ folder of your cleanflight/ folder. The resulting folder structure should look like this:

```
leco@Hutia:~/repositories/git/cleanflight$ ls tools/
gcc-arm-none-eabi-6-2017-q2-update
leco@Hutia:~/repositories/git/cleanflight$ ls tools/gcc-arm-none-eabi-6-2017-
q2-update/
arm-none-eabi  bin  lib  share
```

Old version of the document

The following is outdated and kept only for historical reasons and in the vague hope that it can still be of help for someone.

We suggest that you take an alternative PPA from Terry Guo, found here:

https://launchpad.net/~terry.guo/+archive/ubuntu/gcc-arm-embedded

This PPA has several compiler versions and platforms available. For many hardware platforms (notably Naze)
the 4.9.3 compiler will work fine. For some, older compiler 4.8 (notably Sparky) is more appropriate. We
suggest you build with 4.9.3 first, and try to see if you can connect to the CLI or run the Configurator.
If you cannot, please see the section below for further hints on what you might do.

Adjust the ARM_SDK_DIR variable in make/tools.mk file with the correct "gcc-arm-none-eabi-xxx" version.

Setup GNU ARM Toolchain

Note specifically the last paragraph of Terry's PPA documentation -- Ubuntu carries its own package for
`gcc-arm-none-eabi`, so you'll have to remove it, and then pin the one from the PPA.
For your release, you should first remove any older pacakges (from Debian or Ubuntu directly), introduce
Terry's PPA, and update:

```
sudo apt-get remove binutils-arm-none-eabi gcc-arm-none-eabi
sudo add-apt-repository ppa:team-gcc-arm-embedded/ppa
sudo apt-get update
```

For Ubuntu 14.10 (current release, called Utopic Unicorn), you should pin:

```
sudo apt-get install gcc-arm-none-eabi=4.9.3.2014q4-0utopic12
```

For Ubuntu 14.04 (an LTS as of Q1'2015, called Trusty Tahr), you should pin:

```
sudo apt-get install gcc-arm-none-eabi=4.9.3.2014q4-0trusty12
```

For Ubuntu 12.04 (previous LTS, called Precise Penguin), you should pin:

```
sudo apt-get install gcc-arm-none-eabi=4.9.3.2014q4-0precise12
```

Building on Ubuntu

After the ARM toolchain from Terry is installed, you should be able to build from source.

```
cd src
git clone git@github.com:cleanflight/cleanflight.git
cd cleanflight
make TARGET=NAZE
```

You'll see a set of files being compiled, and finally linked, yielding both an ELF and then a HEX:

```
...
arm-none-eabi-size ./obj/main/cleanflight_NAZE.elf
   text    data     bss     dec     hex filename
  97164     320   11080  108564   1a814 ./obj/main/cleanflight_NAZE.elf
arm-none-eabi-objcopy -O ihex --set-start 0x8000000
obj/main/cleanflight_NAZE.elf obj/cleanflight_NAZE.hex
$ ls -la obj/cleanflight_NAZE.hex
-rw-rw-r-- 1 pim pim 274258 Jan 12 21:45 obj/cleanflight_NAZE.hex
```

You can use the Cleanflight-Configurator to flash the `obj/cleanflight_NAZE.hex` file.

Bricked/Bad build?

Users have reported that the 4.9.3 compiler for ARM produces bad builds, for example on the Sparky hardware platform.
It is very likely that using an older compiler would be fine -- Terry happens to have also a 4.8 2014q2 build in his
PPA - to install this, you can fetch the `.deb` directly:
http://ppa.launchpad.net/terry.guo/gcc-arm-embedded/ubuntu/pool/main/g/gcc-arm-none-eabi/

and use dpkg to install:

```
sudo dpkg -i gcc-arm-none-eabi_4-8-2014q2-0saucy9_amd64.deb
```

Make sure to remove `obj/` and `make clean`, before building again.

Updating and rebuilding

Navigate to the local cleanflight repository and use the following steps to pull the latest changes and rebuild your version of cleanflight:

```
cd src/cleanflight
git reset --hard
git pull
make clean TARGET=NAZE
make
```

Credit goes to K.C. Budd, AKfreak for testing, and pulsar for doing the long legwork that yielded this very short document.