Manual for ArduCopter, FlyMaple and Quadricopter in this project

Contents

[I. Driver and IDE installation for FlyMaple 1](#_Toc423304744)

[II. Other shields configuration 1](#_Toc423304745)

[A. Wi-Fi shield 1](#_Toc423304746)

[B. GPS shield 2](#_Toc423304747)

[III. Create the firmware with ArduPilot framework 3](#_Toc423304748)

[IV. Hardware setups 6](#_Toc423304749)

[A. Connecting the boards 6](#_Toc423304750)

[B. Wiring 8](#_Toc423304751)

[C. ESCs and FlyMaple 9](#_Toc423304752)

[D. ArduPilot motors position 9](#_Toc423304753)

[V. Working with the original ArduPilot and the modified version 10](#_Toc423304754)

# Driver and IDE installation for FlyMaple

Follow the steps in the Technical Report #1 of BÙI Nhã Đạt.

# Other shields configuration

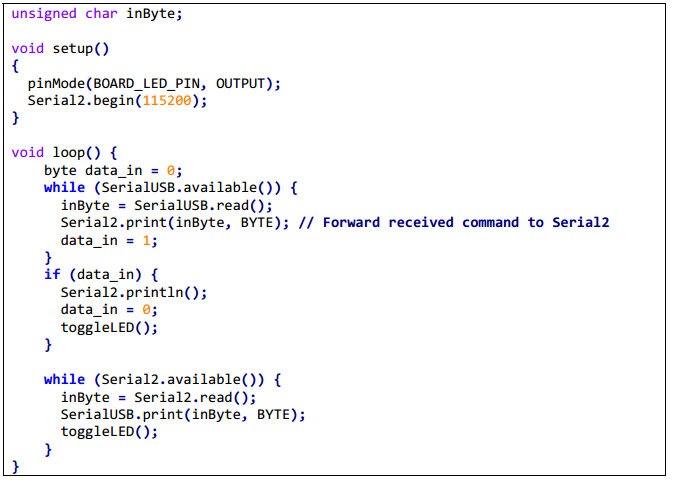
## Wi-Fi shield

* Install a tool to create a Wi-Fi network such as Connectify Hotspot
* Create a network with that tool with specific name and password
* Upload the bridge program (the file in the **additional program** directory) into the FlyMaple board with the MapleIDE then unplug the board.
* Stack the Wi-Fi shield to the FlyMaple board.
* Set the “USB/Arduino” switch to “Arduino” side (this step can be passed as if it is the default settings)
* Set the “Run/Prog” to “Run” side (this step can be passed if it is the default settings)
* Connect the FlyMaple board with Wi-Fi shield to PC
* Send AT commands from the PC via SerialUS comm port (MapleIDE console window)

**AT commands:**



**Bridge program**



## GPS shield

By default, the control shield with the ArduPilot firmware will not understand the messages from the GPS shield using ublox LEAH-5H GPS microchip. However, it just need a simple setting as described in <http://diydrones.com/profiles/blogs/tutorial-programming-your?id=705844%3ABlogPost%3A148030&page=3>

If the red line is on, this shield is powered correctly but the GPS is not fixed, wait for the green light and the “GPS fixed” confirm from the GCS.

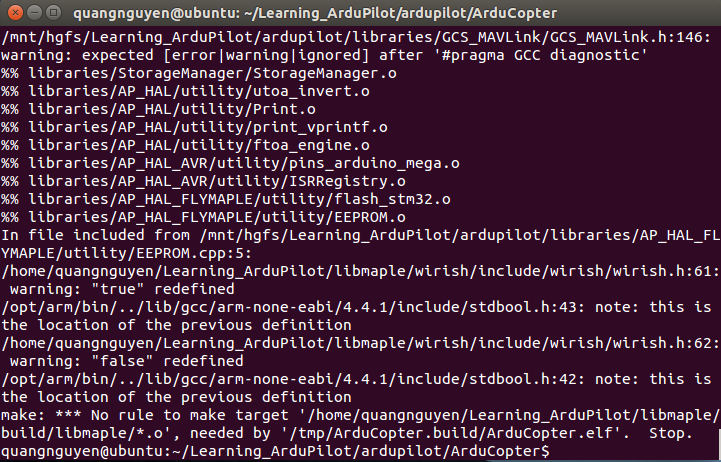
# Create the firmware with ArduPilot framework

One of the ways to create the firmware with ArduPilot is using a virtual machine with Ubuntu or Linux OS.

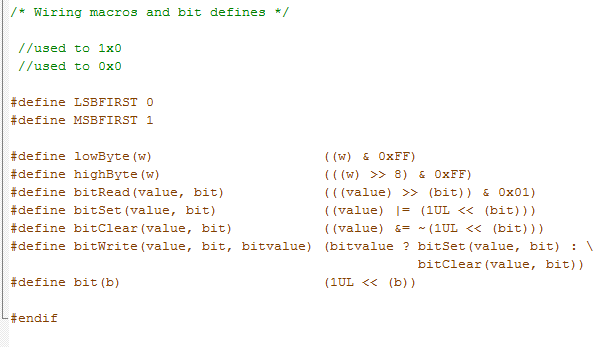
* Install a program to create a virtual machine with Ubuntu OS (In case of your laptop/computer is using Ubuntu OS, create the virtual machine with Windows OS). Please refer the setting here

<http://dev.ardupilot.com/wiki/simulation-2/sitl-simulator-software-in-the-loop/setting-up-sitl-on-windows/>

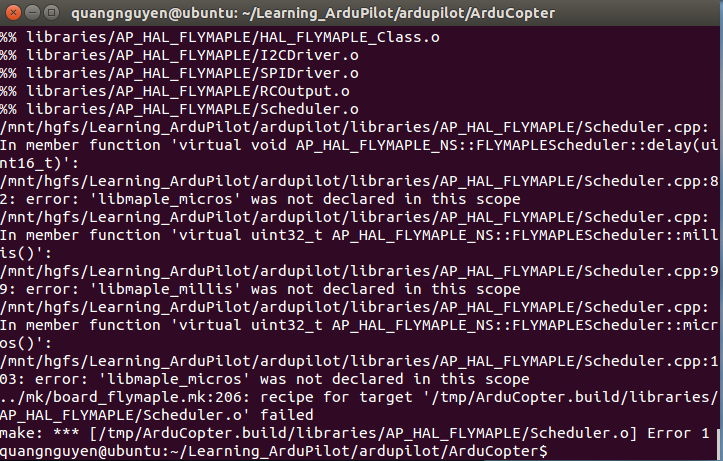
* In the Ubuntu machine, following the steps in <http://dev.ardupilot.com/wiki/building-the-code/building-apm-for-flymaple/> to get the required files and test the FlyMaple. After finish this step, there should be the ArduPilot and the libmaple in the machine using Ubuntu.
* Try to build the firmware for the first time using (**make clean** and **make flymaple-quad**, if you get this error, then you are good.



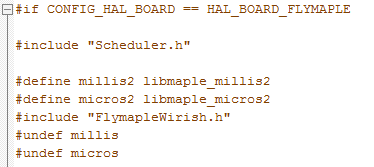
* In order to fix that error, modify the file **libmaple/wirish/include/wirish/wirish.h** as shown below

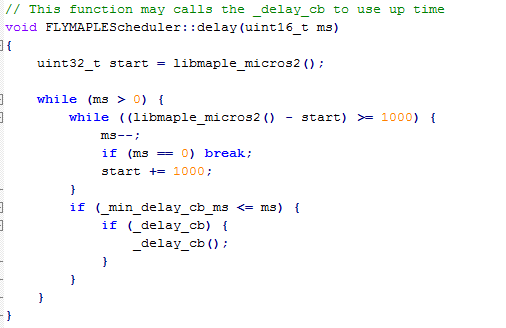


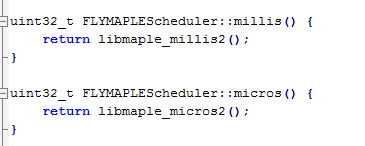
* Now try the make again, another error should happen like this



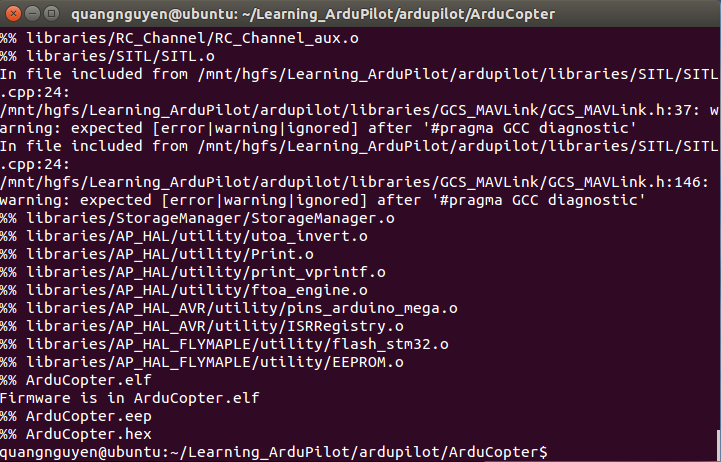
* Fix the file **ardupilot\libraries\AP\_HAL\_FLYMAPLE\Scheduler.cpp** like this







* Try the make again, this time it should work



* Upload the firmware into the board (**make upload**).

**IMPORTANT NOTES:**

* At the time of this manual, the **libmaple** and the **ardupilot** from github have some kind of problems. With the ardupilot, it is the problem in the library with the inav\_inertial. with libmaple, the new version is not compatible with ardupilot.
* The best option is using the libmaple in the zip file and wait for someone fix the problem in the ArduPilot source code
* Remember to fix the path in the mk file.

# Hardware setups

## Connecting the boards

The boards should be connected as shown below

Table ‑: Powering the embedded system

|  |  |  |  |
| --- | --- | --- | --- |
| **Board** | **Vin** | **Powering Pin** | **Receiving power from** |
| **FlyMaple** | 5V | Middle leg of any Pin connecting with the ESCs (D28, D27, D11, D12) (one red wire only) | ESC red wire |
| **Wi-Fi shield** | 5V | +5V | Pin 5V of FlyMaple |
| **GPS shield** | 5V | Vin | Pin 5V of Wi-Fi shield |

From the table above, there must be a jumping wire connecting Pin 5V of Wi-Fi shield and Vin of GPS shield. **Figure 9-1** and **figure 9-2** gives the overview of the completed embedded system.

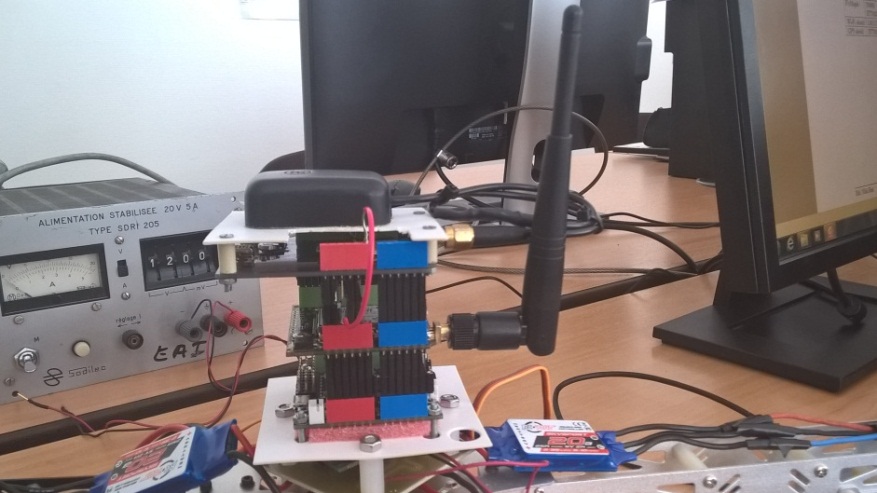


Figure ‑: Completed control system

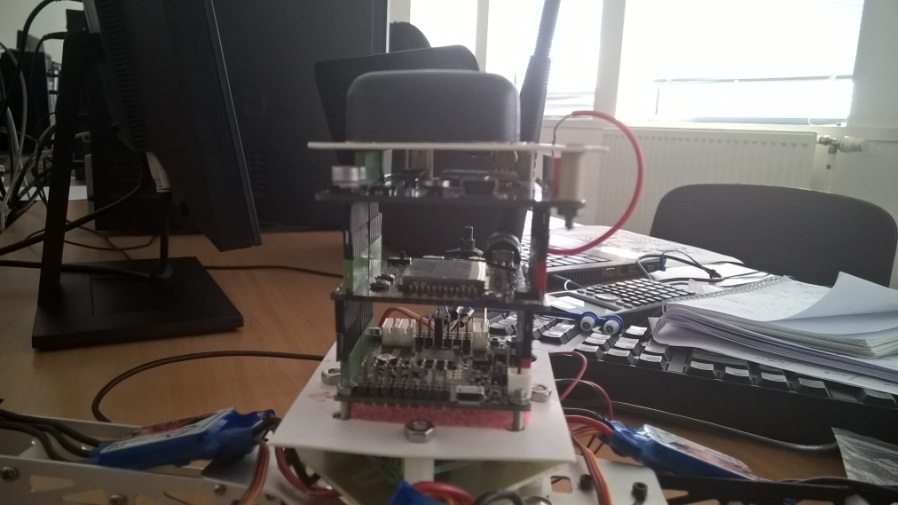
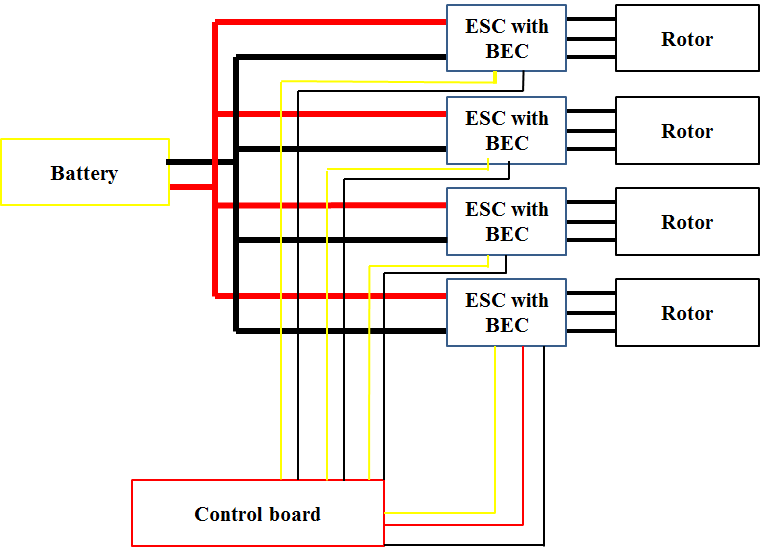


Figure ‑: Completed control system (2)

## Wiring

The wiring between the ESCs and the FlyMaple should be as shown below



**NOTE:** We only need one red wire, do not used more than one unless the required current is more than 2A (which cannot happen in this project)

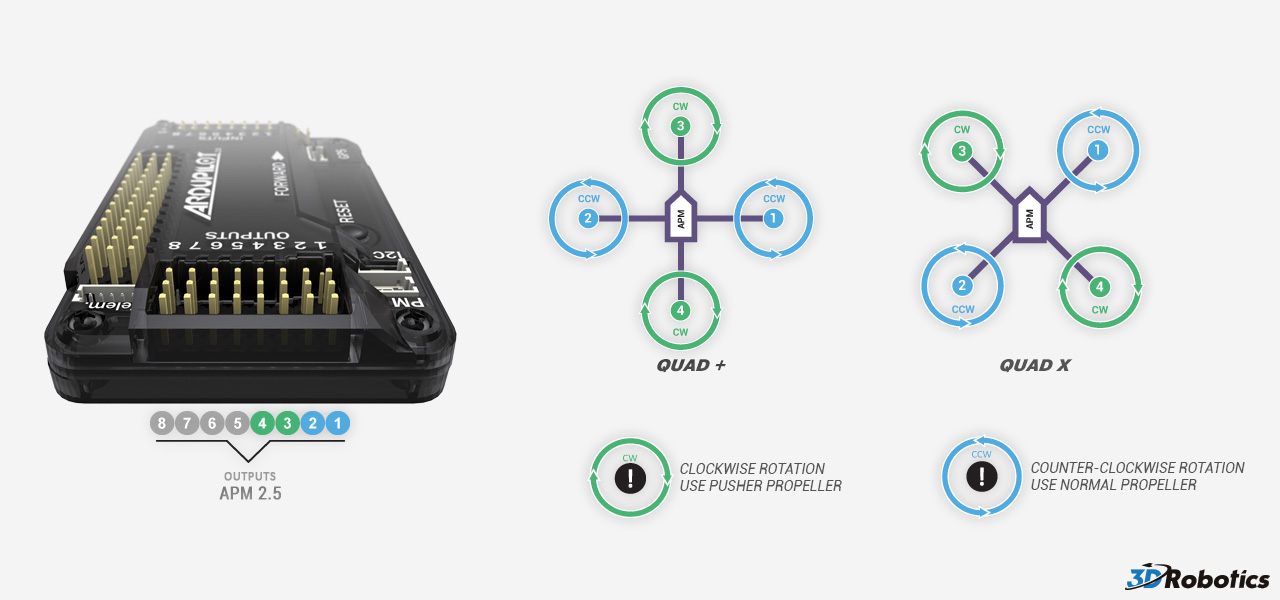
## ESCs and FlyMaple

The connection between the ESCs and the PIN of FlyMaple is in the table below

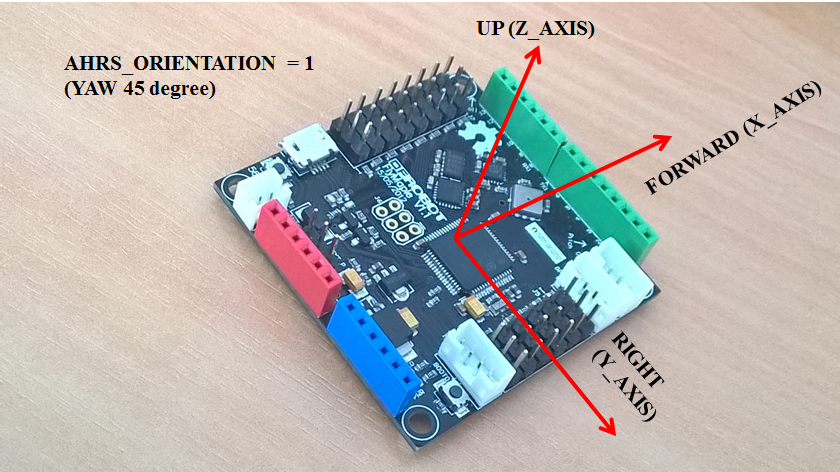
|  |  |
| --- | --- |
| **ESC** | **PIN** |
| 1 | 28 |
| 2 | 27 |
| 3 | 11 |
| 4 | 12 |

## ArduPilot motors position

The calculation in this project belongs to a quadricopter plus-frame. ArduPilot numbers the motors as shown below



The current orientation of the board (AHRS\_ORIENTATION) is 1 (YAW 45) as can be seen below



This orientation is suitable for the current quadricopter, however, this value can be change easily by Mission Planner or modified manually the value of **AHRS\_ORIENTATION** in the code before building the firmware.

# Put the Controller-integrating-module into the original ArduPilot

Following the instruction in the **Modified\_code** directory.

# Put new controller into the module

You should do this step after finishing the section above. Please real the manual in the Completed\_Code directory to have a more detail instruction.

# Working with the completed versions

## Calibrate and program the ESC with the specific ArduPilot version

## Use the completed version

The completed version is based on ArduCopter v3.3. This version includes the new module to integrate new controller into the framework. It also includes two new controller : **New\_PID\_Controller** and **Simple\_IB\_Controller**. Resutls relate to these controllers in simulation can be found in the **Full report.**

# Important sites

## ArduPilot and ArduCopter

<http://dev.ardupilot.com/wiki/learning-the-ardupilot-codebase/learning-ardupilot-introduction/>

<http://dev.ardupilot.com/wiki/apmcopter-code-overview/>

## Building and modifying the code

<http://dev.ardupilot.com/wiki/building-the-code/>

<http://dev.ardupilot.com/wiki/editing-the-code/>

## MAVLink, Mission Planner and other GCS

<http://planner.ardupilot.com/>

<http://copter.ardupilot.com/wiki/common-optional-hardware/flying-with-a-joystickgamepad-instead-of-rc-controller/>

<http://planner.ardupilot.com/wiki/other-project-and-common-topics/common-choosing-a-ground-station/>

<http://qgroundcontrol.org/dev/mavlink_arduino_integration_tutorial>

<https://pixhawk.ethz.ch/mavlink/>

## Software In The Loop (SITL) simulation

<http://dev.ardupilot.com/wiki/simulation-2/sitl-simulator-software-in-the-loop/setting-up-sitl-on-windows/>

<http://dev.ardupilot.com/wiki/simulation-2/sitl-simulator-software-in-the-loop/setting-up-sitl-on-linux/>

## Helpful forum

<http://www.instructables.com>

<http://diydrones.com>