Autopilot & Telemetry Equipment Selection

I have been doing some reading on the autopilot and telemetry system for the project. The following is an attempted summary of the useful info I have found mostly from the ardupilot [wiki](http://ardupilot.org/ardupilot/index.html). Please note that I have not spent much time looking at manufactures for finding the cheapest option. Given the popularity of the system I propose here, I am hoping that lead times are not too long. This will give me some time to look a little deeper and do some further digging (forums/branches of the source etc) and have a discussion before making purchases. Most of this is to catch you up to speed with what we will need for the autopilot and an estimated cost (found at the end of this document).

The NIWA project had success using the [Pixhawk](https://pixhawk.org/choice) to control and navigate their UAV in stable flight. The Pixhawk is very common amongst hobbyists and has been used by UC staff so I thought it would be the best option to investigate.

Pixhawk is the name for the open-source hardware platform designed to host an autopilot. [3DR](https://3dr.com) manufacture the Pixhawk in a convenient ready to use package. The pixhawk can have a number of autopilots installed but Ardupilot (specifically arduplane) is the most commonly used. Arduplane is an open-source project and seemed to be the obvious choice so I decided to have a read through their wiki. Arduplane runs on the pixhawk and takes input from the onboard accelerometers, barometer and magnetometer as well as input from an external GPS and magnetometer module to automatically fly the plane. Arduplane can provide a real-time stream of data to a ground station via a telemetry link. The recommended ground station for general use was [mission planner](http://ardupilot.org/planner/index.html). It is open-source software and many of the examples and capabilities of arduplane were designed to work with mission planner and vice-versa.

The combination of the Pixhawk, Arduplane and mission planner seemed to be the most common configuration meaning that lots of community support will be available. Arduplane and mission planner are also open-source meaning that they are free and give us the best chance to adapt the firmware/software to suit our needs (mainly telemetry of volcanic ash data and launching and returning home from a balloon).

It is probably easiest to order the GPS module from the same manufacturer. 3DR manufacture the [uBlox GPS with compass unit, a buzzer and LED/Arming button.](https://store.3dr.com/t/pixhawk) The Buzzer and arming button are to be connected to the pixhawk upon startup to indicate to the user any issues with the autopilot and to allow the user to arm (enable) the autopilot.



A telemetry radio will be required. 3DR used to manufacture their own implementation of a SiK telemetry radio but they have since ceased providing this. However, the SiK telemetry radio is open-source and runs open-source firmware meaning that other manufacturers can be found. It is designed to work on 433MHz to operate in NZ and specifically designed to communicate via MAVlink, the protocol used for transmitting between ground stations and aerial vehicles. This will integrate easily with the Pixhawk and mission planner. [J-Drones](http://store.jdrones.com/jD_SiK_Radio_Telemetry_radio_p/rfsik20set.htm) is a suitable manufacturer recommended by the ardupilot wiki.

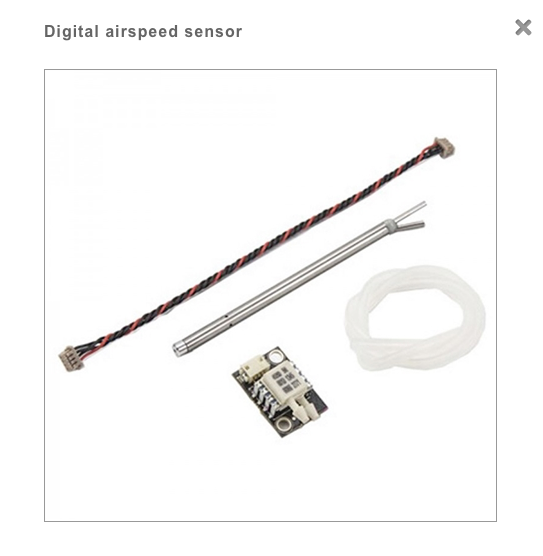


The link as-is will only likely give us a range of less than a few kilometers but without the requirement for a [license in NZ](http://www.rsm.govt.nz) using the [433MHz band](https://www.caa.govt.nz/rpas/). The greater range required for a full scale system is likely to require consideration of different antenna design and likely to require a radio license to enable us to reach the power we need. I can see the range being one of the bigger (more expensive) issues of the project. However, given that we are currently aiming for drone operation of less than 400ft for now, this link will be a good starting point for testing all the systems we need for the project at a low altitude. Perhaps we can consider the need for greater range when we reach that milestone. Note that ardupilot and mission planner provide a directional [antenna tracking](http://ardupilot.org/antennatracker/index.html) feature that could prove useful.

The autopilot makes use of an air speed sensor. This is primarily used for achieving a desired speed or altitude and can increase the range of the UAV. The recommended sensor is from [j Drones.](http://store.jdrones.com/AirSpeed_sensor_MPXV7002_p/senair01kit.htm)



The Pixhawk website recommends that use of the new digital airspeed sensor which costs a little more but could improve the accuracy of readings. I still need to contact JDrones to check on the availability of [these sensors](http://store.jdrones.com/digital_airspeed_sensor_p/senair02kit.htm).



The airspeed sensors can plug in directly to the pixhawk by sharing the I2C port with the GPS module.

Other electronics will include the servo’s for the control surfaces but we can look into these next week or whenever it comes time to select the airframe. At that stage we can have a look at power management and power budgeting to determine what batteries and supply circuitry we need for flight.

One problem we are going to run into is the need to test fly and tune our glider at low altitude. This means we will need to take off from the ground (requiring a powered system) and will need manual control (needing and RC controller and associated antenna). These will bump up the costs for the project even though they may not be used in the final solution. However, it is important that our plane is correctly tuned especially at higher altitudes and wind speeds. We can make use of arduplane’s autotune feature that involves manually flying the plane through a number of maneuvers while the software tracks the response of the plane. Further manual tuning is then recommended for best results.

The need for powered and RC controlled flight may be avoided if we choose a glider and place our system within it in such a way that we can use predetermined control parameters found from another project. However, in order to test the system at any altitude we would need some kind of tethered balloon system in an open field. This seems like it would be more hassle than it’s worth. For that reason, I recommend getting an RC controller and associated receiver. This will enable us to progressively set-up and test the system in easy steps.

The very highly recommended RC controller (and the one used in online examples, and also in the NIWA project) is the Taranis X9D manufactured by FrSky. After initial searching I found a [retailer](http://www.multiwiicopter.com/products/frsky-taranis-plus-australia-new-zealand) for the system.



Any other sensors (namely our volcanic ash sensors) can send data to the pixhawk via the I2C line. Perhaps an arduino or similar micro can be used as a slave device and act as an abstraction layer for any other sensors we may need. Because both ardupilot and mission planner are open source, we will be able to modify the code in such a way that it polls the additional sensors for data and compiles this and stores it alongside other data in the onboard SD card and sends it via the telemetry link. Initial searches show examples [one](http://diydrones.com/forum/topics/custom-sensors-and-real-time-logging?id=705844%3ATopic%3A1716302&page=1#comments) and [two](http://diydrones.com/forum/topics/incporating-sensor-data-into-the-apm-log-files?xg_source=activity) indicate that this is possible.

|  |  |  |
| --- | --- | --- |
| **Equipment** | **Manufacturer/supplier** | **Price** |
| Pixhawk | 3DR | 200.00 |
| GPS Module | 3DR | 90.00 |
| Buzzer/arming button | 3DR | 16.00 |
| Telemetry Set | jDrones | 120.00 |
| Digital Airspeed Sensor | jDrones | 65.00 |
| RC Controller | FrSky/multiwiicopter | 232.00 |
| **Total** |  | **723.00** |

Additional Costs:

* Battery
* Power circuitry
* Air frame
* Servos/motors
* Ash sensing system