Building the LIDAR/GPS datalogger

This is a building guide intended as supplementary material to the manuscript "Inexpensive aerial photogrammetry for studies of whales and other large marine animals". It is written to allow interested researchers to build the datalogger described in the manuscript. The building task is not difficult for anyone with reasonably good soldering skills.

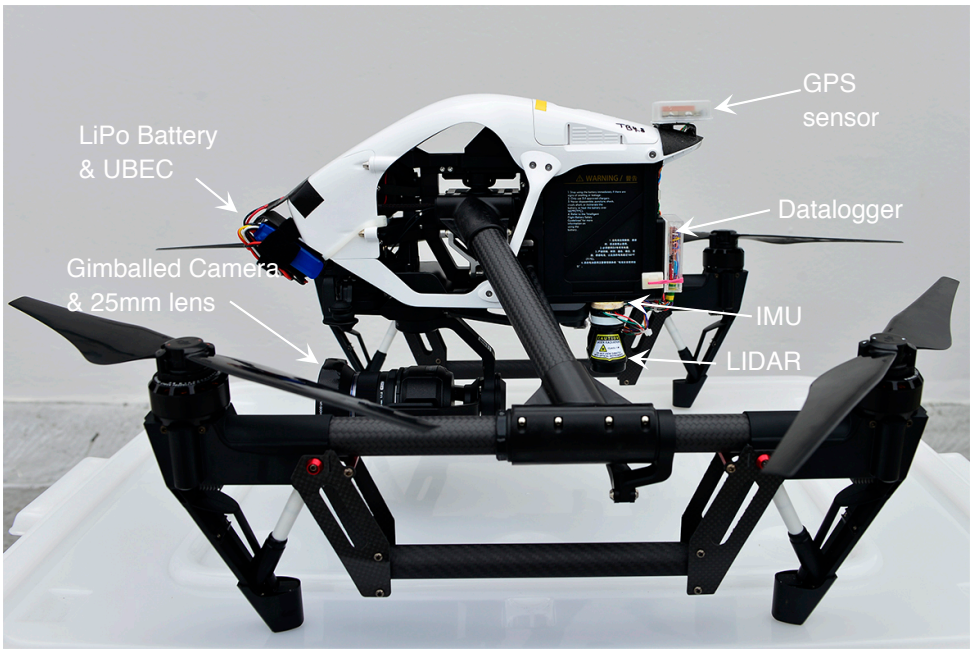
Parts list

Item	Example supplier
Lightware SF11 lidar	www.lightware.co.za
3.3v 8MHz Pro Micro Arduino clone	www.sparkfun.com
3.3v ProMicro MicroSD transflash breakout	www.sparkfun.com
GlobalSat EM506 GPS module	www.sparkfun.com
Pololu MinIMU-9 v5 Gyro, Accel, Compass	www.nicegear.so.nz
DC-DC UBEC 5v 3a voltage regulator	www.mrpositive.co.nz
100µF capacitor	Any electronics store
microSD card	Any electronics store
7.4v 2cell LIPO batteries	Any R/C hobby store

Plus: wire, solder, heatshrink

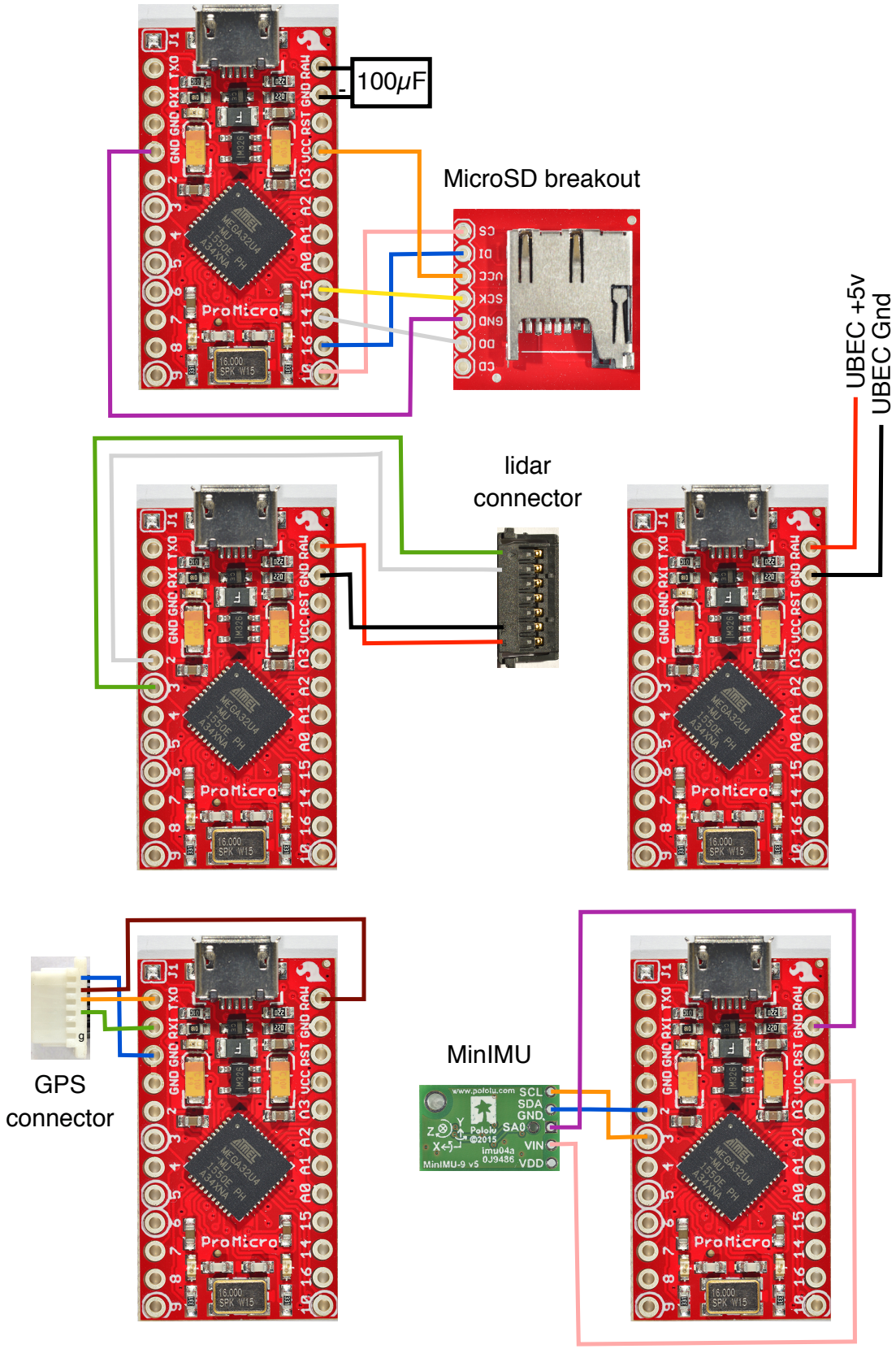
The size and writing speed of the micro SD card are not critical, as the datalogger files are small (c.8kB/min).

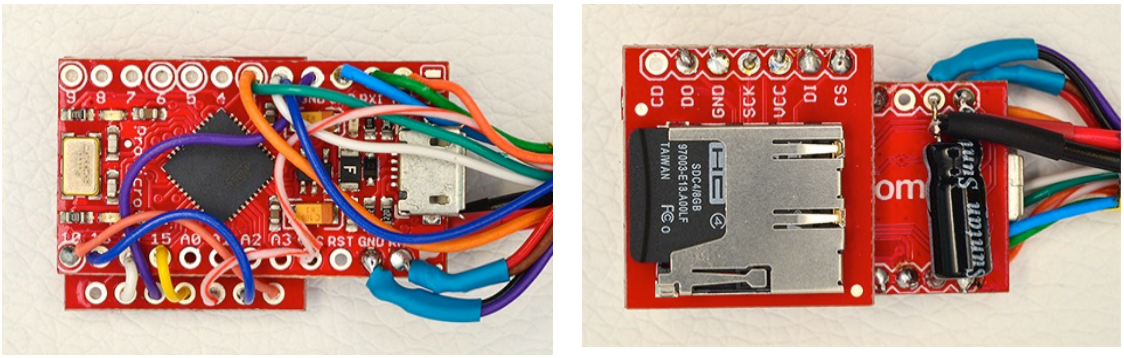
You will need to figure out how long the cabling needs to be to suit your particular UAV and your preferred layout. You'll also need to source appropriate plastic boxes to house the gps antenna and the datalogger. For the datalogger, we've used an Olympus microcassette box. A rubber band provides extra security to hold the lid shut.



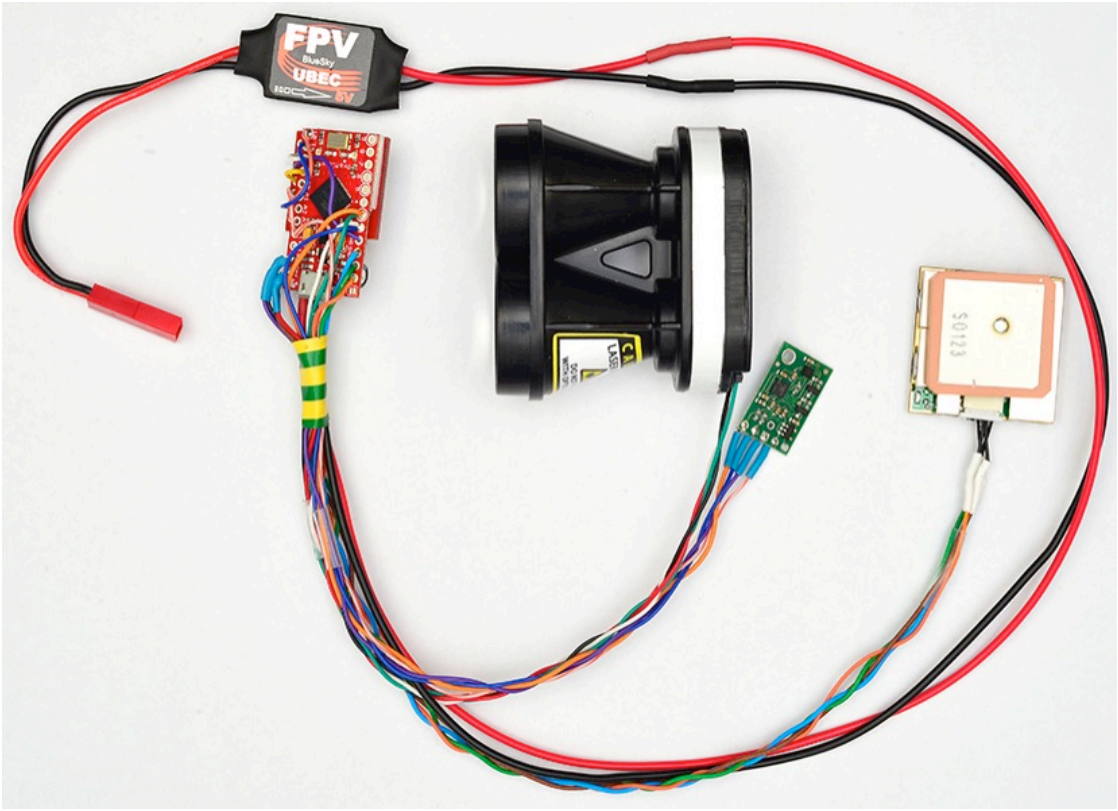
Layout of the datalogger components on the DJI Inspire 1Pro

Datalogger wiring diagram





Completed wiring of ProMicro board and breakout



Completed assembly of component parts before attachment to UAV

Programming and setup of LiDAR + Datalogger

LiDAR setup (SF11C Laser Altimeter from Lightware Optoelectronics)
This needs to be set up using a micro-USB cable plugged in between your computer and the LiDAR unit. You will need a terminal program to do this, examples are the vendor-supplied Lightware Terminal on MS Windows, Zterm on Mac OS, or GTKTerm on Linux.

http://www.dalverson.com/zterm/
https://sourceforge.net/projects/gtkterm/

You will need to connect to the appropriate serial/COM port as created by the USB connection. If in doubt try several from the list. If using Z-Term, connect the cable to the lidar and plug it in to your computer before you start Z-term, Under Settings/Modem Preferences

choose the “usbserial….” port, and under Settings/Connection set baud rate to 115200 (8 bits 1 stop bit). You should see a live stream of data. If not, try changing the baud rate.

Pressing the space bar stops the live stream, and brings up the LiDAR setup menu. Input the following settings. Press the appropriate letter to expand the menu.

```
      ***  SF11/C  Rev 1.21  ***

-----------------------------------
a: Hide system menu
   b: Zero datum offset ........ 0.00 m
   c: Measuring mode ........... long range
-----------------------------------
d: Hide communication menu
   e: Serial port baud rate..... 19200
   f: Serial output mode........ on demand
   g: I2C bus address........... 0x55
-----------------------------------
h: Show analog and alarm menu
-----------------------------------
m: Hide filter menu
   n: Output on lost signal..... 130.00 m
      p: Confirm lost signal.... 1.0 seconds
   q: Median filter............. on
      r: Median filter size..... 16 results
-----------------------------------
```

For the median filter we use a window size of 16. As the device samples at 16 Hz this corresponds to a 1 second filter window. This setting makes the lidar much less sensitive to momentary interruptions of the beam, and produces far fewer outliers.

Serial baud rate (d, e) is not relevant. The I2C protocol (d, g) is crucial.

Arduino software
We used the Arduino IDE software version 1.6.12 with a SparkFun 3.3v ProMicro Arduino board

In the Tools menu go to: Tools -> Board -> Board manager...

Find "SparkFun AVR Boards" and click install.

After that finished, in the Tools menu go to:
Tools -> Board -> SparkFun Pro Micro
Tools -> Processor -> ATmega32U4 (3.3V, 8 MHz)
Tools -> Port -> (select appropriate value once it is plugged in)

It is important that the correct values are set for the above.

You will need to install the TinyGPS++ processing library from
http://arduiniana.org/libraries/tinygpsplus/
Follow the instructions on the website to download the library andinstall it into your Arduino "libraries" folder.

For inclinometer support using the Pololu MinIMU-9 board, install the LSM6 library from https://github.com/pololu/lsm6-arduino
Follow the instructions on the website to download the library and install it into your Arduino "libraries" folder.

With the drone_GPS_and_LiDAR_logger_with_accel.ino and support_fns.ino files installed in a folder named "drone_GPS_and_LiDAR_logger_with_accel" within your Arduino "sketchbook" folder, select File -> Open in the Arduino IDE and then run Sketch -> Verify/Compile. An error will occur if the wrong family of Arduino board is selected or if it can't find the support libraries where it expects to find them on the hard drive.

Finally, with the Arduino connected and the appropriate serial Port selected from the Tools menu, select Sketch -> Upload. This will re-compile the code and upload it via the USB cable. Lights on the Arduino will blink during the process. When it is complete the words "Done uploading." will appear near the bottom of the Arduino software window.

To test, select Tools -> Serial Monitor from the menu. New data will only be output when a valid GPS fix is in place.

Before Installation on the UAV we glue the flat side of IMU to the rear of the lidar, and make up a lightweight spacer to glue the LIDAR underneath the UAV. We attempt to do this as near to the centre of balance as we can.

Installation on the UAV
We made carbon fibre brackets for the GPS sensor and to hold the battery and UBEC on the I1P. Something less sophisticated should suffice.

Wires can be held out of the way with adhesive cable clips, and/or tucked inside the body work of the UAV.

Running the datalogger
Connecting the battery starts the the datalogger. It needs to have a GPS fix in order to write data. A red LED will light up on the GPS and on the datalogger board. Once it gets a fix, you'll see red flashes on the GPS as it sends a data string. The datalogger flashes green once a second when it's writing a new line of data, and flashes amber when it flushes the data to the microSD card.

A 500 mAh battery will last several hours. We change or charge after 4 hours or so. We disconnect it if not flying for more than a few minutes.

The datalogger files are .csv files. The easiest way to open them is to open a blank Excel sheet, and use File/Import. Note that the times are GMT. To delete these files off the chip, you must delete, and empty the trash. The file header is self explanatory.

Happy flying!

Steve Dawson[1] & Hamish Bowman[2]
Departments of Marine Science[1] and Geology[2]
University of Otago, Dunedin New Zealand