

Environmental lapse rate as a wildfire tool: an Arduino based RPAS data logger

Brandon MacKinnon, M. Sc.

Options for remotely piloted aircraft implementation

Remotely piloted aircraft (RPA) have become easy to fly, cheap, and mobile and current regulations would allow suitable atmospheric sampling within wildfire controlled airspace up to 2000 ft AGL, which would allow an assessment of stability in the lower atmosphere. Sensors can be carried on consumer level RPA such as a DJI Phantom 4 but would require a special mounting bracket be made to attach the sensor package. Larger commercial sized RPA would still need a custom bracket, though weight would be less of a priority since they have much larger payload capacities. A light weight bracket could be designed and 3D printed to suit any drone that can lift it.

The sensor package could be manufactured around an Arduino or Raspberry Pi as the processors. In this application the capacity of a Raspberry Pi is not required and an Arduino with several components should suffice. An Arduino would need to be connected to the following sensors and peripherals at minimum:

1. Real Time Clock (RTC)
2. SD module
3. Temperature sensor
4. Pressure Sensor
5. Power source

The real time clock (RTC) serves to keep track of time in a very precise manner. The SD module stores the data while the RPA is in the air and allows the user to download the data upon landing. The temperature and pressure sensors allow collection of the desired data. Finally, a power supply that is separate from the RPA will allow for greater flexibility when choosing an RPA platform and assist in simplifying the design. Suggested sensors and models for each component can be seen below in Table 1.

Table 1: Lapse rate RPA sensor components

Sensor	Model
RTC	DS3231 (I2C) or DS3234 (SPI and 256B SRAM)
SD module	SparkFun DEV-13743 or other
Temperature	Adafruit TMP117 (more accurate) or MCP9808
Pressure	DPS310 (more accurate) or BMP388
Battery	16340, 18350, or AAA
Boost module	Convert 3.3v-5v from battery to 5v for powering electronics

Sensors used for the prototype

A prototype was created using the recommended sensors and functioning code was written that appropriately captures and logs the necessary data. The sensors and costs of the items used for the prototype are as follows:

- 1) Arduino Nano Every with pin headers (\$18)
- 2) Adafruit TMP117 (\$16)
- 3) Adafruit DPS310 (\$10)
- 4) Adafruit DS3231 RTC (\$19)
- 5) DFRobot Micro SD module (\$7)
- 6) Adafruit 5V buck-boost (\$14)
- 7) Battery (~\$10)

Total cost for the prototype not including a 3D printed case at current prices is approximately \$84. A 3D printed case would not significantly increase this cost.

The sensors are connected to the Arduino Nano Every using I2C protocol so that many sensors can be attached to the same pin headers. Each sensor has a unique address that allows for individual transfer of data to the Arduino. The RTC maintains time with its internal battery and only needs to be reset if the sensor loses power.

Arduino code

When the Arduino is powered on it creates a new file labelled based on the hour, minute, and second the unit was powered on. The Arduino then sets up all of the sensors and begins a loop that records data every second. The data is added to the file created on startup. Each time the unit is started up a new unique logging file is created. The data is logged in .csv format and can be loaded directly into excel for processing or run through the R script for automatic processing of the data. The code is available on request.

R Script

Usage of the R script requires a general working knowledge of R. However, beyond setup of the program the user only needs to change some specifics in the code and run it. The code is simple, utilizes the R package “aiRthermo” and only the working directory and file name need to be modified. However, the user should double check that the axis limits are applicable to their captured data. The figure will be saved in PDF format but the user can change that if desired in the code, an example created by the code can be seen below in Figure 1. The code is available on request.

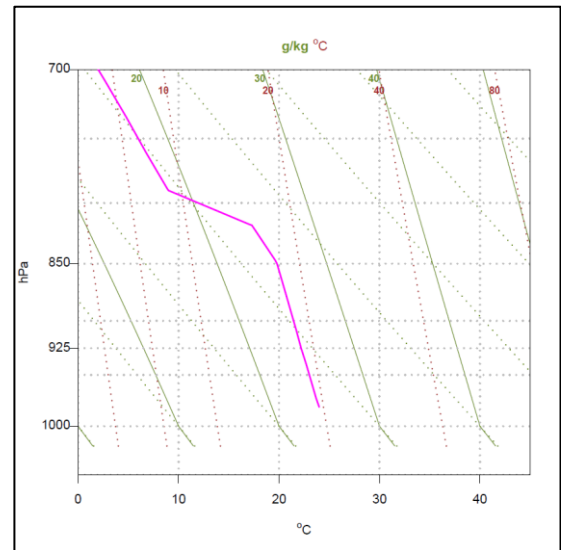


Figure 1: Example of Stueve diagram created by the the R code.

Optional expansions

Many stability indices require a metric for relative humidity. A Relative humidity sensor such as the Adafruit AHT20 can provide data accurate to +/- 2% RH.

Several of the sensors have the ability to report more than a single metric. The DPS310 can also report temperature; however it is not as accurate as the TMP117. Therefore, future iterations could reduce the number of sensors required by accepting less potential accuracy.

For more information

Brandon MacKinnon | (403) 312-1425
Brandon.MacKinnon@fpinnovations.ca

Michael Benson
Michael.Benson@fpinnovations.ca

Follow us



fpinnovations.ca