Flight controller user manual

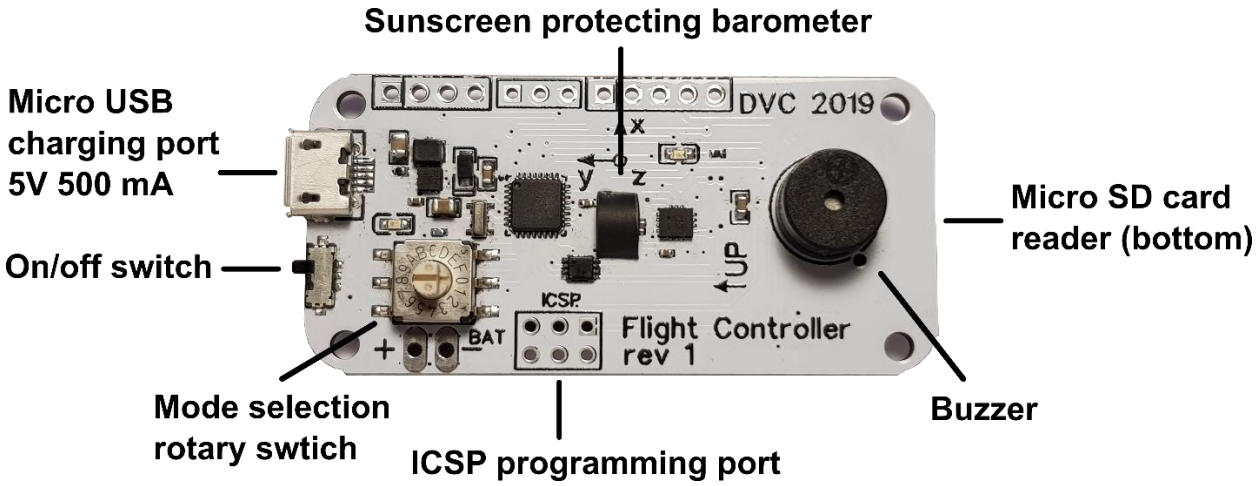
REV 1.1

DanInvents

Disclaimer:

In no respect shall DanInvents be accountable for any liabilities, claims, demands, damages or suits resulting from the use of the flight controller and/or its related software. By using this circuit, you assume all risks associated with this product and its associated features. While the circuitry and software have been tested, they should be considered experimental and handled with caution.

Flight controller layout:



Technical specifications:

Dimensions (length, width, height) in mm	58.6, 27.6, 0.90
Mass	8 g
Minimum input voltage (without protection)	2.90 (2.50) V
Maximum input voltage (without protection)	4.25 (4.50) V
Current consumption while idle*	33 mA
Current consumption while logging*	35 mA (66 mA while beeping)
Sampling rate	125 Hz
Altimeter range	-500 m to 9000 m
Accelerometer range (adjustable)	± 16 g (± 2g, ± 4g, ± 8g)
Gyro range (adjustable)	$\pm 500\ °/s$ ($\pm 125\ °/s, \pm 250\ °/s, \pm 1000\ °/s, \pm 2000\ °/s$)
Temperature range	-40°C to 85°C
Maximum number of files	100

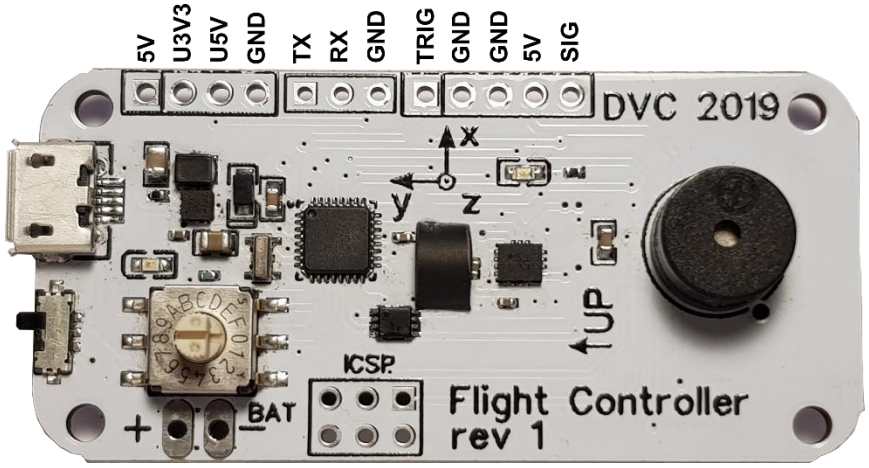
*Current draw without a motor. A SG90 servo with no load draws between 150 to 200 mA while moving. Under load it can draw as much current as 500 mA. Therefore, a battery capable of sourcing over 500 mA is required for smooth operation.

The flight controller features battery charging. The li-ion or li-polymer battery will be charged with a maximum current of 500 mA. To do so, connect a 5V DC source such as a mobile phone charger or a power bank with a micro USB cable to the board. The battery will start charging and once the process is complete the green LED will turn on.

In addition to an integrated battery charger, the board features overcharge, overdischarge and overcurrent protection. These protection features can be disabled by soldering the jumper located on the bottom side of the board.

The board can output up to 12 W of power from the 5V pin. Do not drive the board at 12 W for extended periods of time without cooling, this will destroy the electronics.

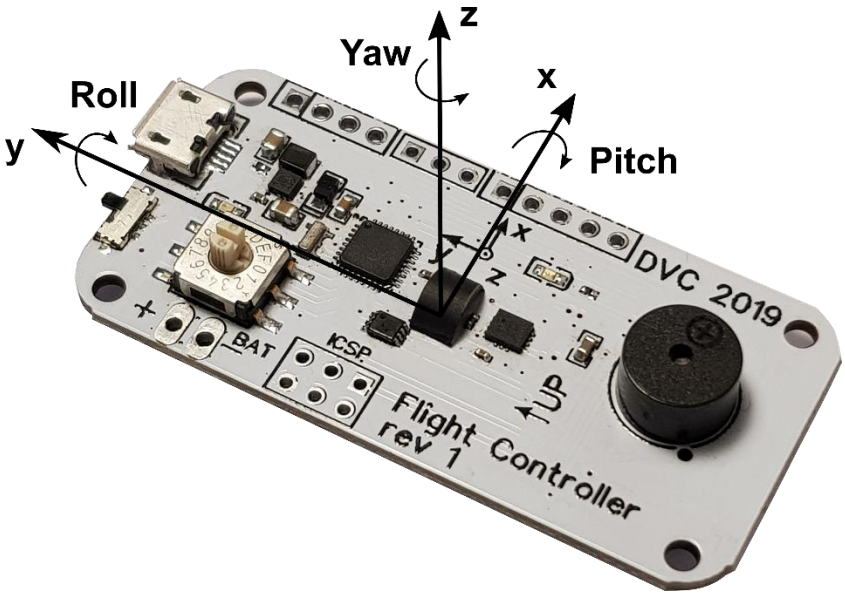
Flight controller pinout:



U3V3 = utility pin 3.3 V
U5V = utility pin 5 V

The purpose of the utility pin is connectivity with other devices. When the flight controller activates the servo motor, the U3V3 and U5V pins source 3.3 and 5 V respectively. This is useful to send signals to other electronics such a microcontroller or a camera. These pins cannot output much current (about 40 mA) and should not be used to drive heavy loads.

Coordinate system:



Acceleration is measured along the x, y and z axes. This acceleration is positive (negative) when it's pointing in the same (opposite) direction as the main axes. The pitch, yaw and roll angles grow positive (negative) when the rotation is anticlockwise about their axis. For example, if the roll angle increases to 180 and then decreases to 90 degrees it means that the controller rotated 180 degrees clockwise and then 90 degrees anticlockwise.

Operational Modes:

The mode selection rotary switch allows toggling between the operational modes. To change from one mode to another, turn off the device and rotate the switch to the desired position. Then turn the board on again.

A) Automatic mode: The flight controller detects apogee automatically and activates the servo motor after a set time (see mode C).
B) Timer mode: The flight controller activates the servo motor after a set time (see mode D).

C) Configure mode: Adjustment of deploy time in the automatic mode. Every step (starting from 0) corresponds to 0.6 seconds increase in the deployment time.
D) Deploy mode: Adjust the deploy time in the timer mode. Every step (starting from 0) corresponds to 1 second increase in deployment time.
E) Servo start position: Adjust the start position of the servo motor.
F) Servo final position: Adjust the final position of the servo motor.

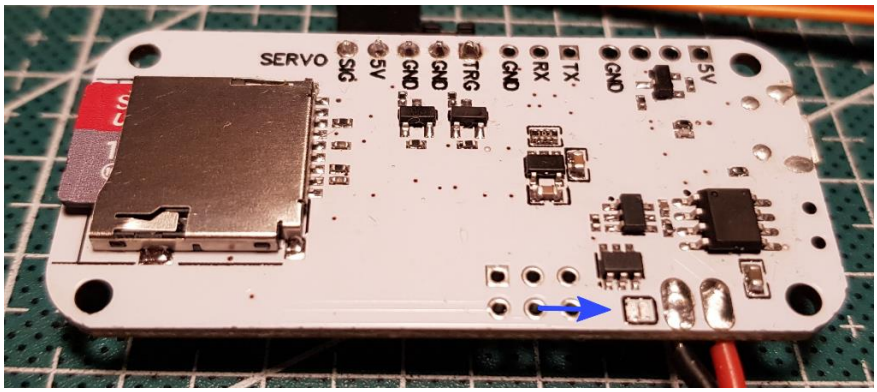
0 to 9) Used for setting the motor position and the deploy time.

Any other switch position will start recording data automatically and will not activate the servo motor.

How to use:

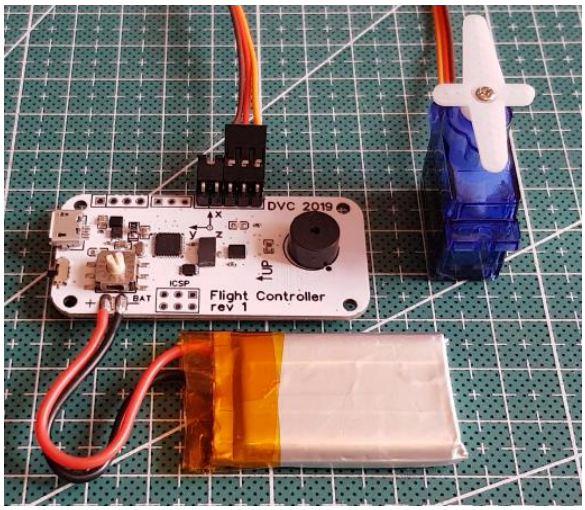
For a detailed video on how to use this board go to https://www.youtube.com/watch?v=Vj6iDYC5_sM. Here is a summary of how it works.

First time use: Solder the battery wires to the pads as well as the pin header to the servo port. Connect the positive battery voltage to + and the battery ground to –. If you are using a lithium-ion or lithium-polymer battery that features protection, solder the jumper located on the bottom side of the flight controller.



It might be that once you switch on the board it will not work. This sometimes happens when a battery is installed, and the protection IC cuts off power. In this case, switch it off and connect a micro USB cable to a 5V power supply, and the other end to the charging port. This will reset the protection IC allowing power to flow from the battery.

Insert an SD card to your flight controller. For best results, use an SD card class 10. If the card cannot be read, if it is full or there are over 100 files of the same type (e.g. log0.txt,…,log100.txt) the blue LED will turn on and data will not be recorded.



To adjust the start (end) position of the servo motor attach the servo motor to the corresponding pins on the board. Move the rotary switch to E (F) and turn on the device. Once the blue LED is blinking slowly rotate the shaft to the desired motor position and turn off the power. The deployment time for the automatic (timer) mode is adjusted by rotating the switch to C (D). Once the LED is blinking rotate the shaft to 0 and then select the desired deployment time. In automatic (timer) mode every step increases the time delay by 0.6 (1) seconds. Therefore, if 2 is selected in automatic (timer) mode the set time is 1.2 (2) seconds.

Once the board is configured, place it in upright position on your application. Then, insert a jumper between the trigger pin and ground. When this jumper is disconnected the board will interpret it as a trigger and it will commence logging data in the automatic and timer modes. Alternatively, one can leave the pin header connected to the trigger pin and the board will detect launch once there is positive acceleration along the y axis.

At the time of deploy the flight controller will move the servo motor to the stored “final” position, the utility pin will flip HIGH and the blue LED will turn on.

Recommendations:

To ensure accurate altitude readings, do not cover the vent hole in the barometer. If you place your board in a closed space, make sure that there are vent holes so that the inner and outer pressure equalize. The barometer is protected with a sunscreen, since sun light shining on the sensor cause it to malfunction. If you remove this cover, make sure to place the board in a dark environment for accurate readings.

After you have logged 100 files, transfer them to a computer and erase them from the SD card or use another SD card to continue logging.

Programming the flight controller

The flight controller is already programmed, should you want to modify the software, follow these steps:

Install the latest Arduino IDE software in your computer. Download the modified Adafruit libraries as well as the EmotiBit BMI160 libraries and place them in your Arduino libraries folder. You can find the code for the Flight Controller as well as the modified Adafruit libraries from this link https://github.com/DanInvents/Flight-Controller.

From the examples tab, upload Arduino ISP to an Arduino Uno. Now you can use your microcontroller as a programming board. Next, connect the wires from a 6-pin flat ribbon cable so that the ISP lines from the Arduino match the ISP lines from the flight controller connector. Gently press the contacts against the ICSP header and from the Sketch>Upload Using Programmer upload the code. For more information and connection diagrams refer to https://www.arduino.cc/en/tutorial/arduinoISP.

Uploading code to the flight controller will erase its memory. Therefore, any stored motor positions and deployment times will be erased.