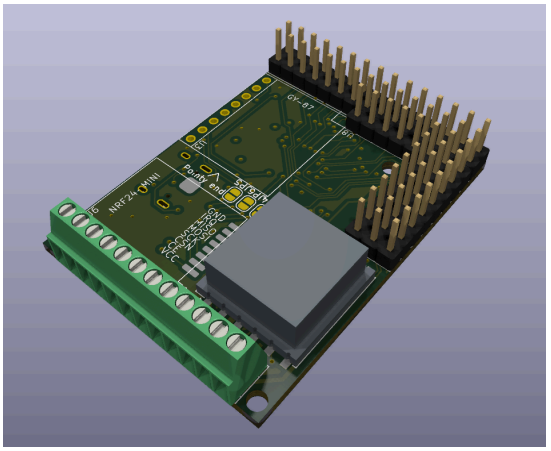
Cansat Flight Computer Datasheet and manual

Hardware features:



Microcontroller: STM32F405RGT6

Sensors (GY87 Module): IMU (MPU6050), Barometer (BMP180)

SD Card slot via SDIO
Radio: 2.4GHz NRF24 (GT24 Module)
GPS: L80

30 GPIOs

System Overview

The product features a large number of peripheral systems and interfaces, this section provides a detailed guide regarding these systems and interfaces from a user's perspective.

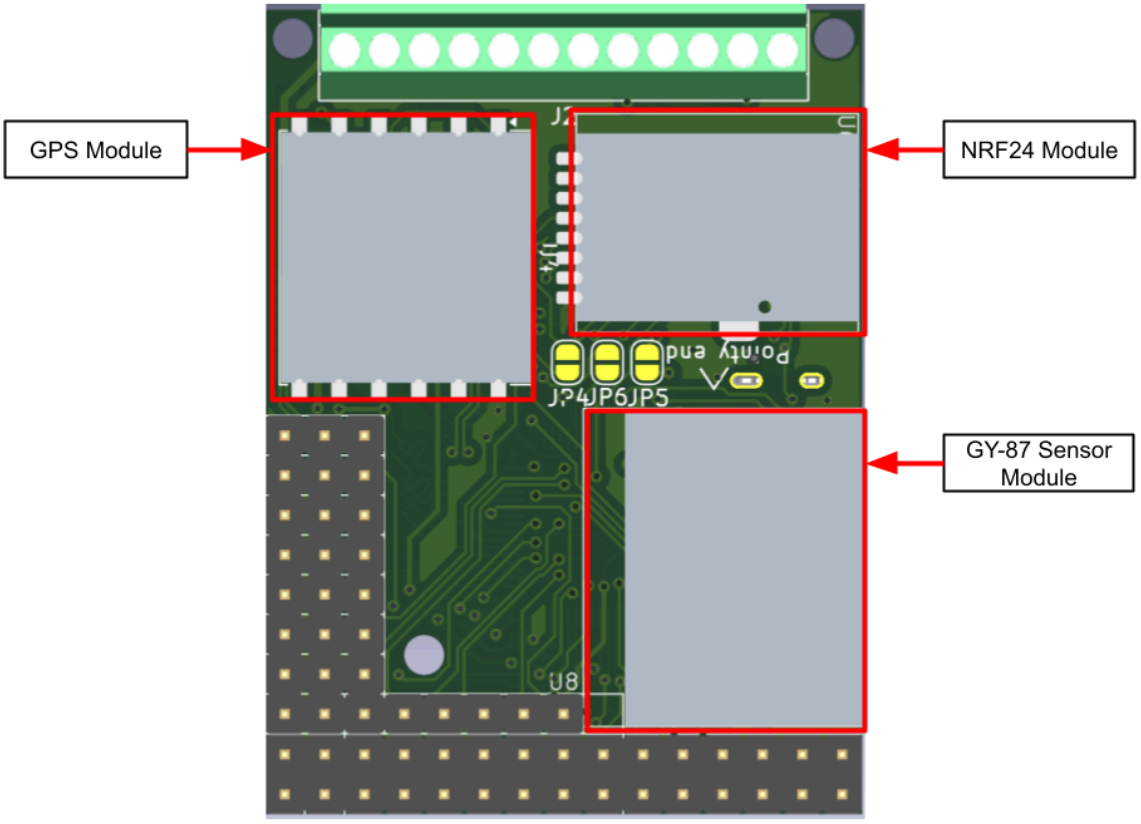


Figure 1: Product peripherals

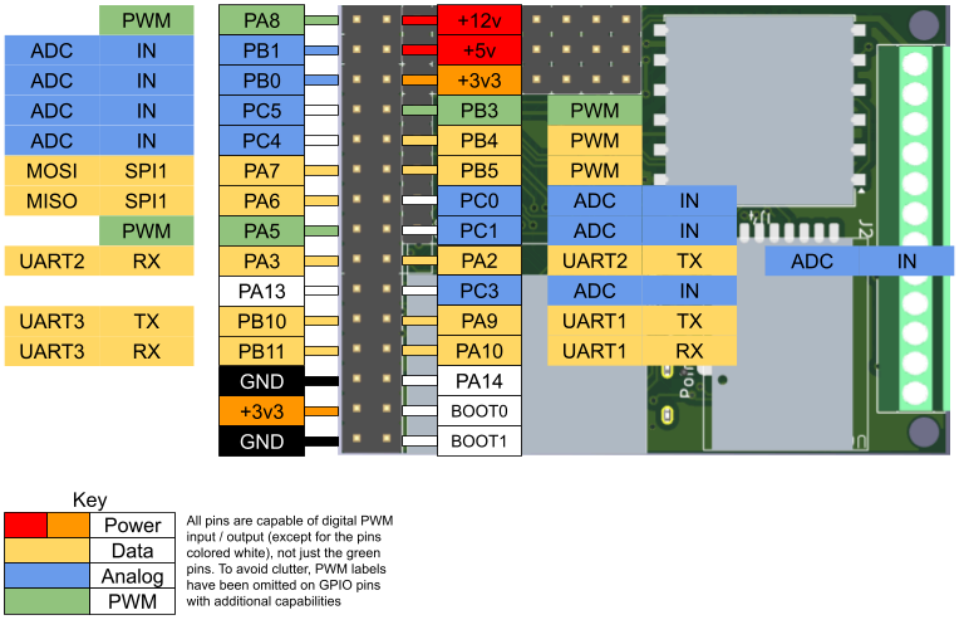


Figure 3: System main GPIO pinout

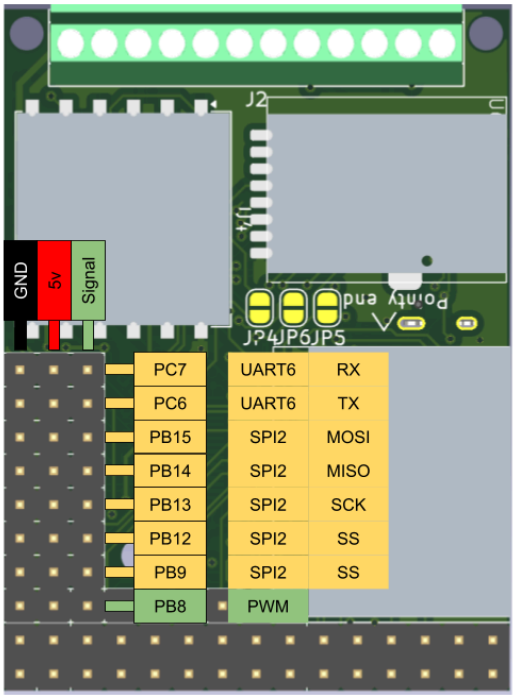


Figure 2: Servo PWM port pinouts

This product features 8 servo PWM, usable for control RC servos. At the same time, these pins can be reprogrammed to be generic GPIO if needed

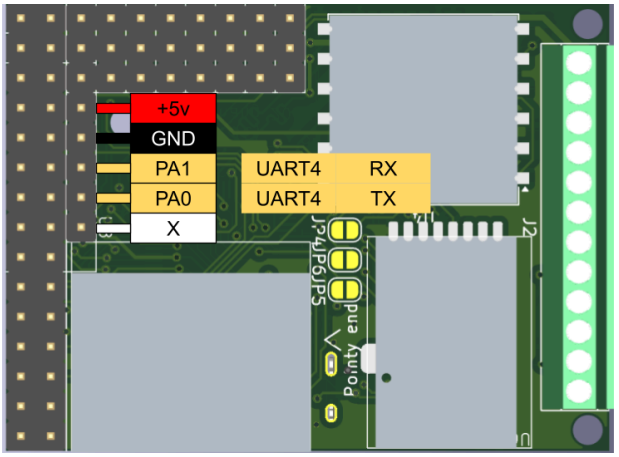


Figure 4: GPS tap off

Due to the complex nature of CANSAT electronics GPS lock can sometimes take an excessive amount of time, thus, the pins of the GPS module has been broken out so the module can be independently debugged.

Under ideal lab conditions, the product can achieve a GPS lock in around 3 min. However some users have reported during extensive SD card write to disk operations, it may interfere with onboard GPS. If that occurs, a 3rd party external GPS installed some distance from the product can resolve the issue.

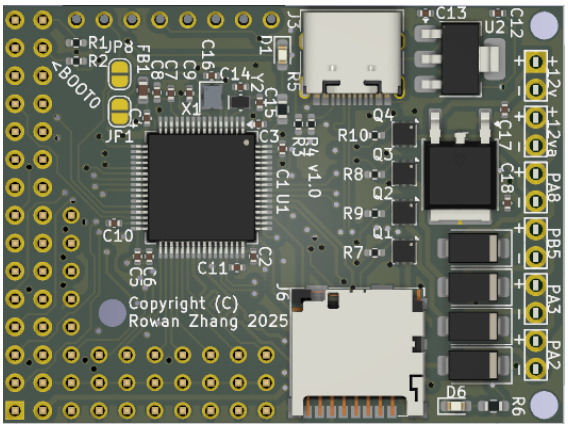
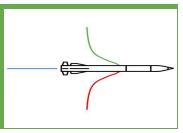


Figure 5: Back of PCB

The screw terminals located on the left of the above figure is the high current DC terminals, which includes both the battery terminals, as well as the nichrome wire terminals.

This product has 4 high current DC channels that can be used to power nichrome wires to sever plastic lines for deployment of recovery mechanisms and other systems.



The nichrome wire needs to be at least 50mm long, and it is not recommended to turn on the wire for more than a few seconds. The wire can be turned on by applying a high signal to the microcontroller pin whose name is next to the port within which the nichrome wire is connected.

The 12va terminal powers the nichrome wire channels. While the 12v terminal powers all other systems on the product. Despite the label of “12v”, the system can safely run with voltage as low as 8v, however 9v - 12v is the recommended voltage range, and 9v batteries and 3s LiPo batteries are recommended power sources. It is highly recommended to use 2 separate batteries, one for nichrome wire and one for everything else to avoid a microcontroller “brown out”

Developers' guide

The Cansat Flight computer is an open source product that the customer can develop their in-house firmware for. Onboard LED is at pin PC13.

Design source files: https://github.com/Elven-Aerospace-Industries/cansat

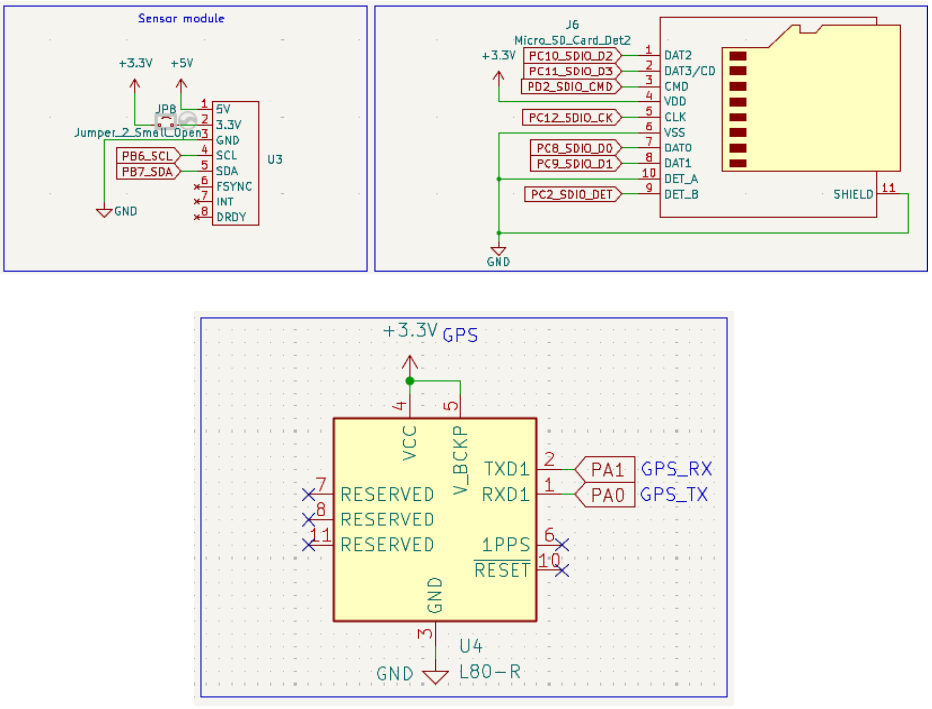


Figure 6: Connections of internal peripherals

Recommended 3rd party libraries

https://github.com/nRF24/RF24

https://bitbucket.org/christandlg/bmp180mi/src/master/

https://github.com/rfetick/MPU6050_light

https://github.com/mikalhart/TinyGPSPlus/tree/master

The SD card slot is connected to the SDIO pins, so the STM32SD library is needed, it uses the same function names as the regular Arduino® SD Library https://github.com/stm32duino/STM32SD DET pin: PC2

The automated test scripts used for the product can be found at https://github.com/Elven-Aerospace-Industries/cansat/releases/download/v1.0/flight-computer-tests.zip which can be used as a reference for firmware development, it also includes a copy of the recommended libraries

Firmware development

The STM32f405RGT6 is Arduino® compatible using STM32duino
https://github.com/stm32duino/Arduino_Core_STM32

Development environment setup:

1. Download Arduino IDE (arduino.cc, or https://github.com/arduino/arduino-ide/releases/tag/2.3.10)
2. Click File -> preferences -> additional board manager URLs
3. Paste in the URL https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json
4. Search for STM32 in the board manager, and install the official core from STMicroelectronics

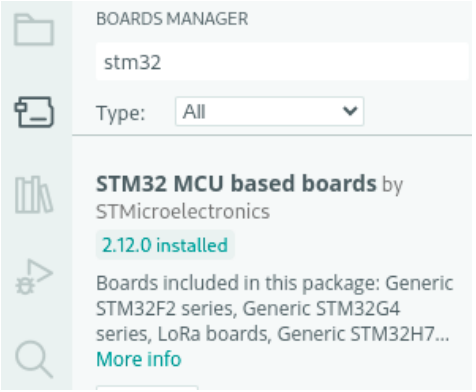


Figure 7: Core installation

5. Install STM32cubeprogrammer

Note:

STM32duino uses the STM32 microcontroller pin names instead of the integer pin numbers of the Arduino ® boards.

So, to control a pin, please substitute the number for the pin name. E.g. to turn the LED on, use

```
digitalWrite(PC13, HIGH);
```

instead of

```
digitalWrite(13, HIGH);
```

Uploading firmware

1. Click on the board dropdown, and select the STM32F4

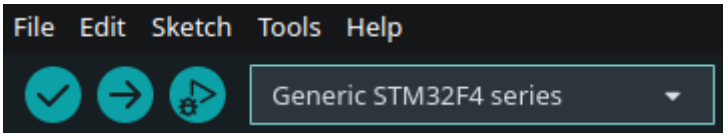


Figure 8: Board selection

2. Click on Tools, and set the following parameters

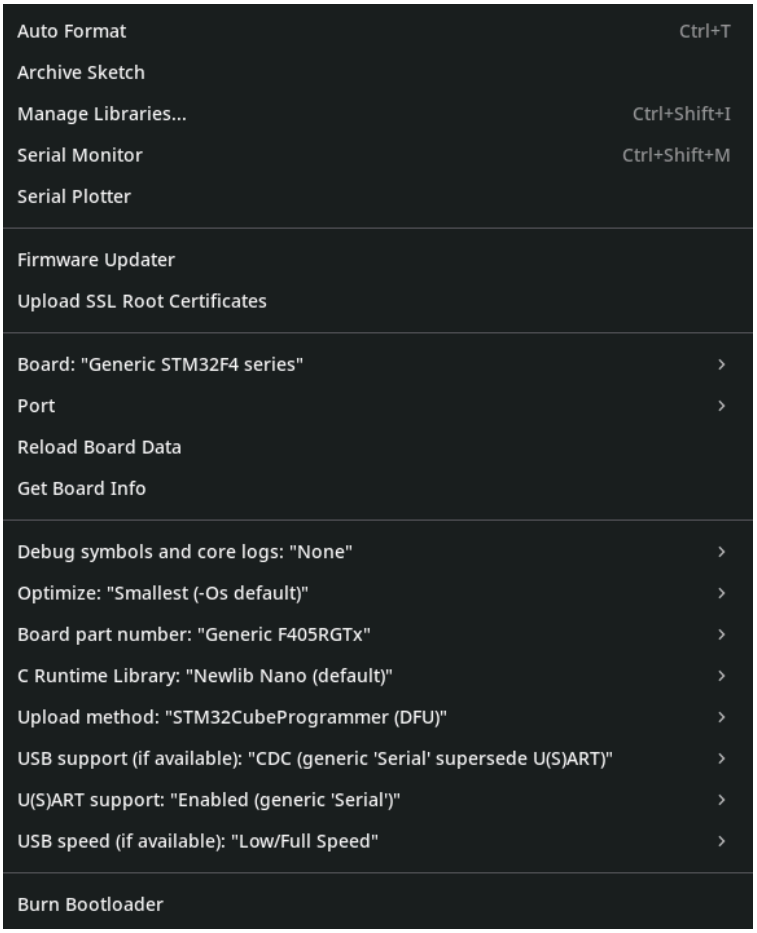


Figure 9: Upload configuration

3. Short the BOOT0 jumper located near the corner to +3v3, then connect the product to a workstation via USB. Lastly, click the upload button in Arduino® IDE to upload the firmware

Note: The Arduino ® IDE, STM32duino, and STM32cubeprogrammer are 3rd party software, and are not products of Elven Aerospace Industries