CATS User Manual

Contents

1 Revision History 3

Glossary 4

2 Introduction 5
- 2.1 Coverage of This Manual 5

3 CATS Vega 6
- 3.1 How to Use 6
  - 3.1.1 Connection to Your Computer 6
  - 3.1.2 Description of the Configurator 6
  - 3.1.3 Mounting 11
  - 3.1.4 Battery, Switch and Actuators 11
  - 3.1.5 Configurable Actions & Finite State Machine 13
  - 3.1.6 Setting up the Minimal Flight Configuration 14
  - 3.1.7 How to Get the Data on Your Computer 16
  - 3.1.8 Generate Plots Using the CATS Plotting Tool 16
  - 3.1.9 Software Updates 17
- 3.2 Hardware 19
  - 3.2.1 Overview 19
  - 3.2.2 Specifications 20

4 Ground Station 21
- 4.1 How to use 21
  - 4.1.1 Explanation of All Menus 21
  - 4.1.2 Telemetry Modes 22
  - 4.1.3 Charging 23
  - 4.1.4 How to get the Data on your computer 23
  - 4.1.5 Software Updates 23
- 4.2 Hardware 24
  - 4.2.1 Overview 24
  - 4.2.2 Specifications 24

5 Example Configuration of the CATS Vega + Ground Station 25
- 5.1 Simple Example Configuration 25
- 5.2 Advanced Example Configuration 27

6 Testing 29
- 6.1 Manual Triggering 29
- 6.2 Sequential Triggering 29

7 Advanced Information 30
- 7.1 Software Overview 30
- 7.2 Telemetry 31
- 7.3 Estimation Algorithms 32
- 7.4 Description of the CLI 34

8 FAQ 36

9 Warranty 37

1 Revision History

Date	Version	Comment
15 December 2022	1.0.0	Initial Release

Table 1: Revision History

Glossary

Apogee The highest point of the rocket during a parabolic flight.. 9, 14, 25, 27, 29

Barometer A device which can measure the barometric pressure of its surroundings. 30

CATS Control and Telemetry Systems. 5–7, 11, 17, 19, 21, 23, 25, 27, 29

CLI Command Line Interface, used to set and get values from the flight computer when not using the configurator. 6, 8, 17, 29, 35

CRC Cyclic redundancy check. 31

DFU Device Firmware Update, A mode used to flash new firmware to an embedded system.. 23, 24

Drogue Chute Parachute which is deployed at apogee. Usually a rather small parachute only destabilizing the rocket for a controlled descent. 14

FHSS Frequency Hopping Spread Spectrum. 31

FreeRTOS An operating system for embedded CPUs, the backbone of the CATS Software. 30

FSM Finite State Machine, a term used to describe the different flight phases and how the changes in the flight phase happens / which changes are allowed to happen. 13

GNSS Global Navigation Satellite System, for example GPS / Galileo. 11, 19, 22

I/O Input/Output, used to group all inputs and outputs of the system in one word. 11, 12, 19, 27, 29

IMU Inertial Measurement Unit, can measure the linear acceleration (i.e. how fast you accelerate straight) and the angular velocity (i.e. how fast you turn). 13, 30

Kalman Filter A filtering technique used to estimate states according to physical laws and measurements. 32

Liftoff An event during flight, when this event happens, the motor of the rocket starts to accelerate and the rocket takes off. 9, 29, 32, 33

Main Chute Parachute which is deployed at a certain height above ground level. Usually a bigger parachute, slowing the rocket down enough to not damage it once it hits the ground. 14, 15

Moving A flight state; when in this state, it is in safe mode; no actions are taken by the flight computer. 29, 32

Patch Antenna A flat antenna with high directivity. The CATS Vega board has a patch antenna, which is used to get the GNSS signal.. 11

Power Supply Needed to power the system, in rockets usually some battery. 5

PWM Pulse Width Modulation, protocol used to actuate most modern servos. 12

Pyro A charge with two wires which can be blown with a high current applied. 9, 11–14, 19, 20, 25, 27, 29

Ready A flight state; when in this state, liftoff can be detected. 29, 32

RF Radio Frequency. 11

Servo A small electrical motor where the position of the motor can be set by applying a PWM signal. 8, 11–15, 20, 27, 29

UART A communication protocol to send data from one location to another over three wires. 19, 20

2 Introduction

Hi there and welcome to the CATS flight computer! In the following pages we will explain the CATS system, composed of the CATS Vega flight computer and the CATS Ground Station, such that you can use it on your rocket. A How To Use section is written for both components that should give you enough information to launch your rocket. The other sections include some advanced information, which is not necessary for the basic usage of the CATS system.
If you have any feedback, suggestions or need further help regarding the CATS system do not hesitate to contact us on our Discord server[1]. This is the first release of the manual and it will be improved based on your feedback!
The whole CATS ecosystem is fully open source. You can find all the code and hardware designs on our GitHub page[2].

2.1 Coverage of This Manual

This manual covers the usage of the CATS Vega flight computer and its ground station. It explains the working principle of the flight computer and how to connect it to a Power Supply and deployment actuators of your choice. It also explains how the flight computer needs to be set up for your particular flight trajectory.
This manual does not cover everything that can be done with the CATS System. In particular, it does not cover how to modify the software or hardware, nor how the software works in detail. For further information about those aspects contact us on our Discord server[1].

[1]https://discord.gg/r7ErmSNvsy
[2]https://github.com/catsystems

3 CATS Vega

This section describes how the flight computer works and how you can set it up for your flight. The How to Use section should give you all the information about the basic features of the flight computer. In case you want more detailed information, Section 7 will give you more insights.

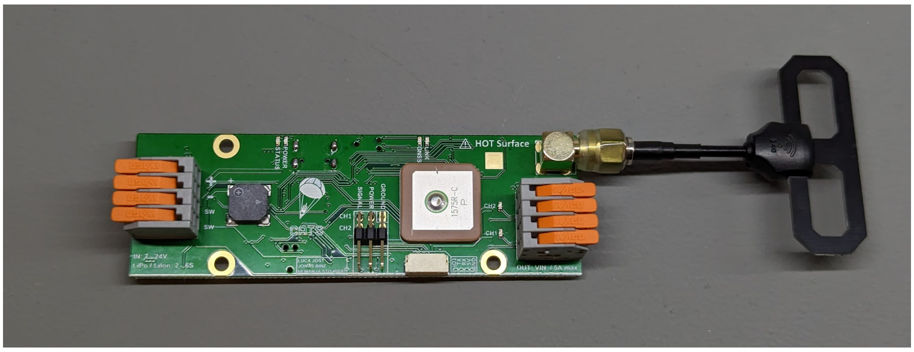


Figure 1: CATS Vega

3.1 How to Use

3.1.1 Connection to Your Computer

Before you connect your CATS Vega to your computer, we recommend you download the configurator from our website. Usually, no drivers are required to get started but if the device is not recognized by your computer please refer to our wiki[3] where we describe some troubleshooting steps.

3.1.2 Description of the Configurator

The configurator allows you to configure the CATS system from your computer. The configurator can be downloaded here[4]. Make sure you download the latest stable version. Release candidates are marked as such and should only be used for testing new features.
Note: This manual describes the current version of the configurator. Currently the configurator does not support telemetry settings, hence those are not explained in detail. It is only shortly mentioned how to set those using the CLI. For further information about this contact us on our discord server. An updated version including telemetry settings will be released beginning 2023 with an update of this manual.

Home Tab

Once the configurator starts, the home tab is shown. If the flight computer is already plugged into your computer just select the correct communication port (label 2 in the Figure 2a) and press connect. If the connection times out, double check that you are attempting to connect with the correct device. If you have multiple flight computers connected make sure you are configuring the desired one. As long as no flight computer is connected, the other tabs are disabled (label 6), and the configurator shows that no board is connected (label 5 and 7).
If no communication port / the wrong communication port is shown in the drop down menu, press refresh (label 1). If you communication port is still not showing up, refer to our wiki for some troubleshooting steps.

[3] https://github.com/catsystems/cats-embedded/wiki/Installation
[4] https://github.com/catsystems/cats-configurator/releases

Main Block	Description
Refresh Button (Label 1)	Used to refresh the shown communication ports in label 2
Com Port Selection (Label 2)	Selection of the communication port to connect to the CATS Vega
Connect Button (Label 3)	Press to connect to the CATS Vega
Home Screen (Label 4)	Some general information about the Configurator
Connection Status (Label 5)	Information box, tells you that the flight computer is currently not connected
Navigation (Label 6)	Greyed out as long as no flight computer is connected. Once it is connected, used to navigate between the tabs.
Connection Status (Label 7)	Shows the connection status

Table 2: Overview of the different Settings for the Home Tab

Configuration Tab

This tab shows you the status of the flight computer and lets you configure various parameters. Figure 2b shows the Configuration tab. In the following Table, the labels are explained in more detail.

Main Block	Parameter	Description
Communication Port (Label 1)	Refresh Button	Used to refresh the communication port list
	Com Port	Shows the communication port selected
	Connection Button	Shows the connection status and allows to connect/disconnect
Navigation (Label 2)	Configuration	The current tab
	Events	Event tab, described on next page
	Timers	Timer tab, described on next page
	CLI	access to the CLI, for advanced commands, described on next page
General (Label 3)	Main Altitude	Set the desired height above ground level where the main chute should be deployed
	Liftoff Threshold	Acceleration Threshold for which Liftoff should be detected. We recommend a value of 40 m/s^2 for most flights
	Initial Position Servo 1	Initial Servo angle of the Servo connected to the Servo port 1. If you do not use Servos, no settings need to be set here.
	Initial Position Servo 2	Initial Servo angle of the Servo connected to the Servo port 2. If you do not use Servos, no settings need to be set here.
	Calibrate Accelerometer	Calibration of the accelerometer. If you do this, make sure that the board is placed on an even surface.
	Backup Config	Used to save a configuration file to your computer. This file can then be set on the flight computer by pressing load config.
	Load Config	Load a previously backuped configuration to the flight computer.
	Reset Settings	Set the default parameters of the flight computer.
Info (Label 4)	System Time	The current system time since boot up in ms
	State	The current flight state of the board
	Voltage	The battery voltage (if no battery is applied, a value of 0.48 is schown
	State Estimation	The currently estimated height, velocity and acceleration
Hardware Info (Label 5)	Status	Connection status of the board
	Board	Hardware Revision
	CPU ID	CPU ID and revision on the board
	Code Version	Current code version. Useful to verify that a system updated worked as expected.
Save Settings (Label 6)	Save	Save the current settings to the board. **Careful:** if Save is not pressed, the values are not written to the board!
	Refresh	Refresh the displayed values to read from the board and see what is configured.

Event Tab

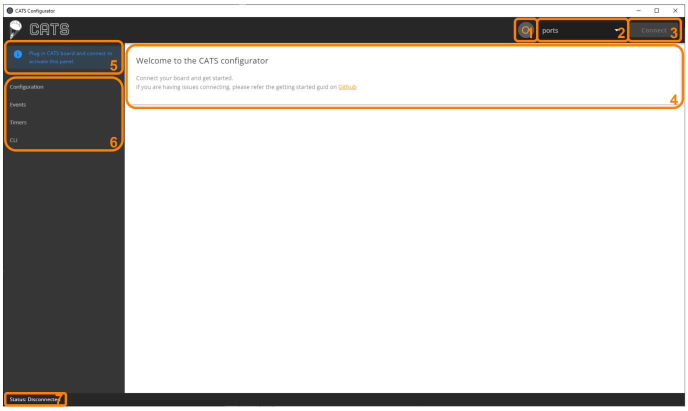
This tab shows you the configuration of the individual event. For each event (Liftoff, Burnout, Apogee, Main Deployment, Touchdown, Custom 1 and Custom 2) up to eight actions can be mapped. On each Event, Add Action can be pressed to add an action to the flight computer. We recommend keeping the recorder log and recorder off event at Liftoff and Touchdown to not miss any data.
In the example (Figure 2c, Pyro channel 1 is turned on, 2 seconds are delayed and then it is turned back off. For the main deployment the same is configured for Pyro 2.
Figure 2d shows the pop up menu which is opened when "Add Action" is pressed. The user then can click on the upper part, which describes what action should be done and then the lower part, which describes what that particular action should do.
To remove events again, press the little cross right of the event you want to delete. Press on the little gear on the left to reconfigure that event.
Don't forget to hit save once you've done some changes!
Note: The custom events can only be triggered as described below in the Timers section.

Timers

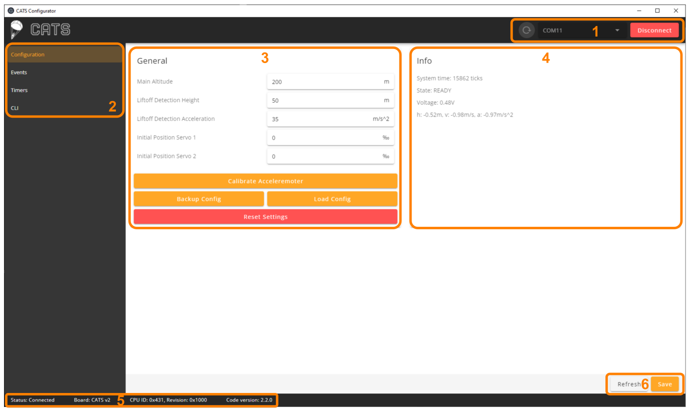
This tab is used to configure the individual timers. There are a total of four timers and each timer can be enabled and disabled through the little yellow button on the top right of each timer field.
Once a timer was enabled (Figure 2e, Timer 1), the start event can be defined, the duration in ms and the end event which shall be triggered. In the example picutre the start event was chosen to be Liftoff, the duration 10000 ms and the end event Apogee. This means that at liftoff, a timer of 10 second is started which triggers the Apogee event at its end.
Timers are the only way to trigger the custom events as described above in the Events section.

CLI

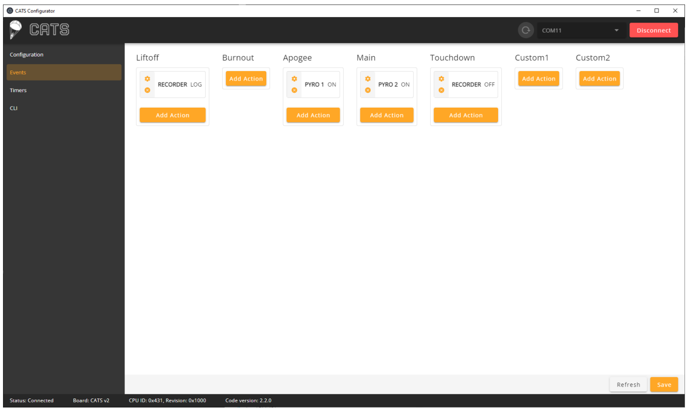
The CLI tab allows the user to send manual commands to the CATS board. All supported commands are explained in detail in section 7.4. A screenshot of the CLI is shown in Figure 2f.



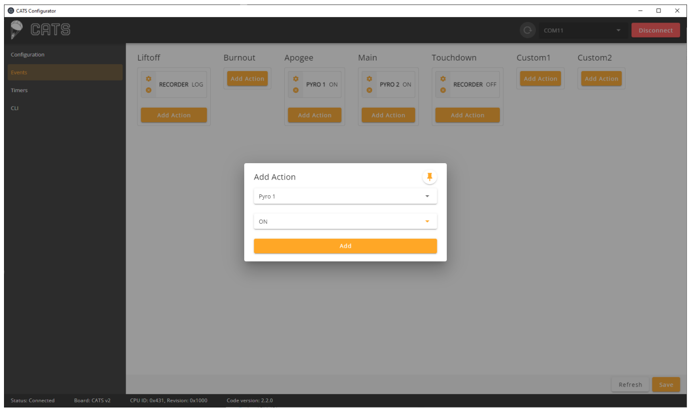
(a) Home Menu.



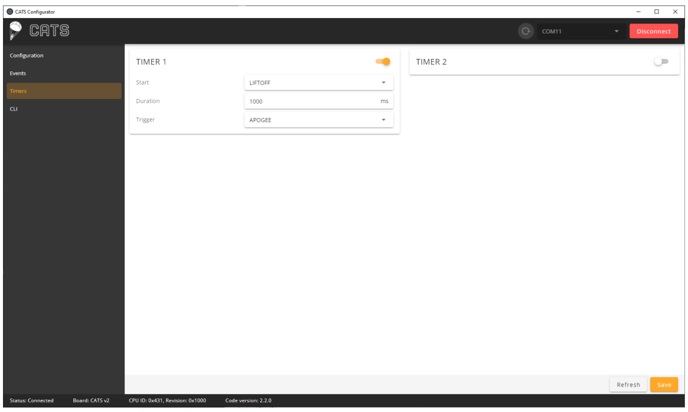
(b) Configuration Menu.



(c) Event Menu.



(d) Configuring an Event in the Event Menu.



(e) Timer Menu.

(f) Command Line Interface.

Figure 2: Configurator Screenshots.

3.1.3 Mounting

The CATS Vega **does not** require a specific mounting orientation. The liftoff direction is automatically detected and used for the internal state estimation. Therefore you can mount it in any way you see fit.
The board has a length of 100 mm, a width of 33 mm and a total height of 15 mm. Three mounting holes are present on the board to secure the system safely to your rocket. The mounting holes are 60 mm by 27 mm apart. The holes are designed for M3 screws and some additional spacers are recommended to keep the electronics from touching your rack. The 3D files of the system can be downloaded from our website.
In order for you to get a good radio reception during the flight, pay close attention to the surroundings of your antennas. Make sure the CATS flight computer is in a radio-transparent section of your rocket (e.g., glass fiber or cardboard). Do not mount your system in a carbon fiber section as it will block all RF signals. Make sure the Patch Antenna on the board has a view of the sky for optimal GNSS reception. Keep your telemetry antenna away from any metallic objects.
When the system is powered up, the up direction is automatically detected as soon as the system is stable. If the system is moved, it will register this movement and disarm itself. Once it is stable again, it will recalibrate itself. Whenever the system arms and disarms itself, a sound is beeped through the buzzer and the state changed is also shown on the ground station. For more information regarding calibration refer to sections 7.3 and 3.1.5.

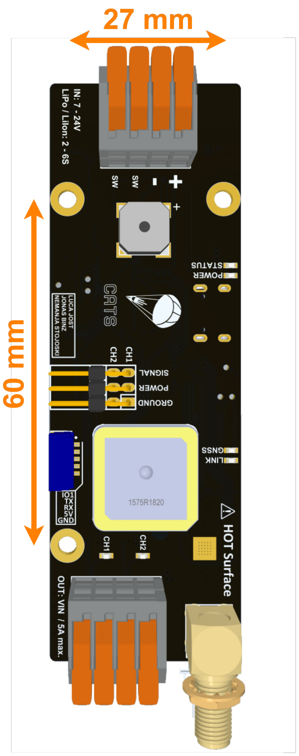


Figure 3: CATS Vega board with the mounting hole and dimensions.

3.1.4 Battery, Switch and Actuators

The CATS Vega has one battery port, one switch port, two Pyro channels, two Servo channels and one low level I/O. In the following, each port is briefly explained. Table 3 shows an overview of the most important parameters. For further information about the board hardware refer to section 3.2. There, also a picture is shown where each I/O is marked on the board itself (Figure 7)

Battery Port

The battery port supports voltages between 7 and 25 Volts. For LiPo and Li-ion batteries this translates to 2 to 6 cell battery packs. The battery port is reverse-polarity protected.

Switch Port

The switch port allows the user to add a mechanical switch to the system. If this switch is turned off, the system is fully disconnected from power.

Note: The battery current is routed through the switch. Make sure that the wires and the switch are rated for the currents required.

Pyro Channels

The Pyro channels apply the battery voltage with around 1 V of dropout to the connected circuitry. Usually an electric match is connected to the channel to ignite a black powder charge. However, the channels can also be used to power other devices. For example the Pyro channels can be used to actuate solenoid valves (with an external flyback diode), power cameras or other electronic circuits. By default the maximum current which can be drawn is around 1 A continuous. The reason for that is that the channels are short circuit protected via a resettable PTC fuse. This is more than enough to ignite electric matches before the fuse reduces the current. If the load connected requires more current, the fuse can be bypassed with a solder jumper on the back of the board. In this configuration it is recommended to stay bellow 5 A continuous or 20 A burst. Be very cautious what you do from this point forward as a short on the channel is going to damage the board!

Servo Channels

The Servo channels can be used to actuate PWM Servos. An onboard voltage regulator reduces the battery voltage to 5 V which is used to power your servos. The power rails for the microcontroller is kept completely separate from the 5 V power supply, therefore, even a short circuit on that power rail has no impact on the system. A maximum current of 3 A can be drawn. A PWM signal is always applied to the Servo channel and the end points can be changed in the configurator.

Low Level I/O

The low level I/O can be used to propagate a signal to another system. The voltage level is 3.3 V and the pin is directly connected to the microcontroller therefore the I/O should **only** be used for signal transmission and not to actuate any recovery mechanism!

I/O	Description	Limits
Battery Port	Connect battery	7-24 V
Switch Port	Connect mechanical switch	n.a.
Pyro Channels	Connect up to two Pyros or other devices	Battery voltage / 5 A
Servo Channels	Used for Servo actuation, up to two Servos	5 V / 3 A
Low Level I/O	Only used for signal transmission, no actuation!	3.3 V / 10 mA

Table 3: Overview of the I/Os needed for a flight.

3.1.5 Configurable Actions & Finite State Machine

The system that controls all the different actuations of the VEGA flight computer is the finite state machine (FSM), shown in Figure 4.

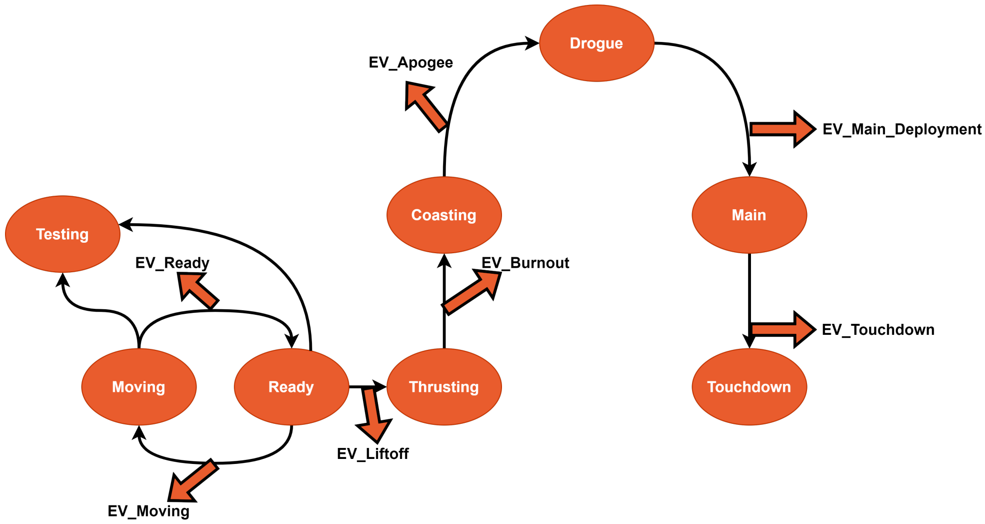


Figure 4: Finite State Machine Controlling the CATS Software.

Each event can be used to trigger some actions (i.e., a Pyro channel, a Servo channel, a timer, etc.) Triggering of actions based on events is described in section 3.1.2.

Moving → Testing	Telemetry input to put it into testing mode
Moving → Ready	IMU (gyroscope and linear acceleration) readings are constant for 10 seconds. Optional: Additionally, a telemetry command must be sent to allow for the transition.
Ready → Moving	Linear acceleration and gyroscope movement above a certain threshold is detected over a range of 10 seconds.
Ready → Thrusting	The measured acceleration in any direction is larger than the user defined acceleration value for 0.1 seconds **OR** the height above ground level is larger than the user defined value for 1 second.
Thrusting → Coasting	The measured acceleration in the "up" direction is smaller than 0 m/s^2 for 0.1 seconds
Coasting → Drogue	The estimated velocity needs to be smaller than 0 m/s for 0.3 seconds
Drogue → Main	The estimated height is smaller than the user defined height for 0.3 seconds
Main → Touchdown	The estimated velocity is in the bound $[-4, 4]\ m/s$ for 1 second

Table 4: FSM Transition Specifications

With this setup for state changes, the flight has a strictly controlled order. The Main event can only be thrown after the Apogee event.
The Apogee event can only be thrown if the acceleration in the "up" direction was smaller than 0 m/s^2 for some time. This gives you the security that hiccups in sensor readings will not influence the reliability of detecting the Apogee and Main events.

Actions

Based on the thrown events, actions can be mapped onto those events. For each event, a total of 8 actions can be mapped. This gives you the option to do many different things for any different application. Examples include:

- Enabling a camera at liftoff using a Pyro channel,
- Actuating a solenoid valve for two seconds only using a Pyro channel,
- Enabling some mechanism at engine burnout,
- Disabling the camera at touchdown using the Pyro channel,
- ...

The full range of actions can be found below

Action	Parameter
Pyro 1	ON/OFF
Pyro 2	ON/OFF
Servo 1	[0-100]%
Servo 2	[0-100]%
Low Level I/O	ON/OFF
Delay	[0-15000] ms
Recorder	ON/OFF/PREFILLING

Table 5: Exhaustive List of all possible Actions

3.1.6 Setting up the Minimal Flight Configuration

For every flight a couple of parameters need to be set for nominal flight performance. For this, some preparation from the user is needed. In particular, the user needs to know:

- Expected maximum acceleration
- Recovery mechanism for the Drogue Chute
- Recovery mechanism for the Main Chute
- Time until Apogee (optional)
- Desired height of the Main Chute deployment

- Time until the Main Chute deployment (optional)

With this in hand, the user can configure their flight computer.

1. Connect the flight computer to the user computer.
2. Open the configurator and connect to the board as described in section 3.1.2.
3. In the Configuration tab, set the liftoff threshold. We recommend using a liftoff acceleration threshold of $40\,m/s^2$, but make sure that it is around $20\,m/s^2$ lower than your maximum expected acceleration.
4. In the Configuration tab, set the main altitude to your desired height. This is where the Main Chute will be deployed.
5. If you use a Servo channel in either of your recovery mechanisms, it is now also the time to set the initial Servo position.
6. Save the settings.
7. Go to the Events tab.
8. For the apogee event, set your deployment mechanism as described in 3.1.2.
9. For the main deployment event, set your deployment mechanism as described in 3.1.2.
10. Save the settings.
11. (Optional) Go to the Timers tab.
12. (Optional) Set the Timer One start event to Liftoff and the Timer One end event to apogee. Set the time until apogee with 1-2 seconds margin.
13. (Optional) Set the Timer Two start event to Liftoff and the Timer Two end event to main deployment. Set the time until apogee with 10-60 seconds margin, depending on the flight time.
14. Save the settings.
15. Set the link phrase of your CATS Vega. This is currently done by sending the command `set tele_link_phrase = "linkphrase"` where the "" are omitted. **After** sending the command, also send the command `save`. **Note:** This will be added to the configurator in a next release.
16. Set the same password on your Ground Station (Navigate to Settings, link phrase and type in the same password).

At this point you are ready to set up the flight computer in your rocket. With this particular flight configuration you would need to do the following steps:

1. Plug in the switch into the switch port.
2. Plug in the battery into the battery port.
3. Plug in the recovery mechanism for the apogee event.
4. Plug in the recovery mechanism for the main event.
5. Mount the flight computer to your rocket.
6. Place the rocket on the launch pad.
7. Flip the switch to power the flight computer on.
8. At this point, you will get data to your ground station.
9. Arm the flight computer over the ground station. The flight state should change to Moving and the buzzer beeped a sound.
10. The flight computer is now armed. Every 6 seconds, the flight computer beeps twice to indicate that it is in the ready state.
11. Launch your rocket!

3.1.7 How to Get the Data on Your Computer

After your flight, just plug your board to a computer with a USB-C cable. The flight computer should act just like a USB drive and you can simply drag and drop your flight onto your desktop.

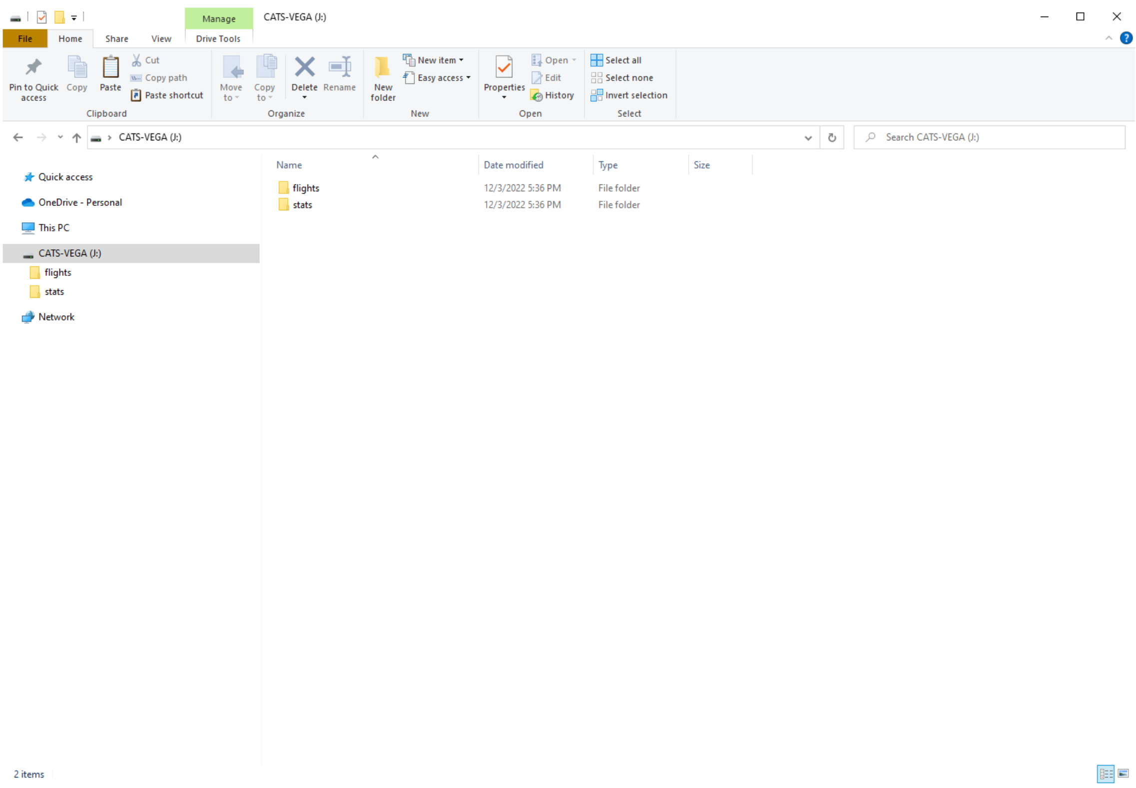


Figure 5: USB drive when the CATS Vega is plugged into the user computer.

3.1.8 Generate Plots Using the CATS Plotting Tool

In order to visualize the logs recorded on the Vega flight computer, you need to clone the cats-logs repository and run the log_visualizer.py script. Note that you need to have git and Python 3 installed on your computer in order to be able to download and run this script.
To visualize your logs, follow these steps:

```
git clone https://github.com/catsystems/cats-logs.git
cd cats-logs/log_parsing
pip install -r requirements.txt
python log_visualizer.py -i <path to input log> -o <path to output directory>
```

The script will parse the log file and generate a self-contained HTML file that contains all the plots. Additionally, the script will also generate raw and processed CSV files for every type of recorded value.
You can get more information on the visualizer script by calling:

```
./log_visualizer.py --help
```

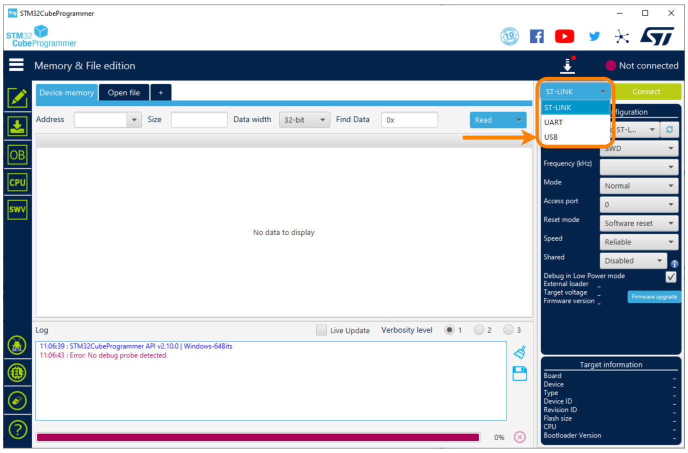
Note: Log visualization will eventually be integrated with the Configurator so that everything is accessible from one place, without needing to install additional tools.

3.1.9 Software Updates

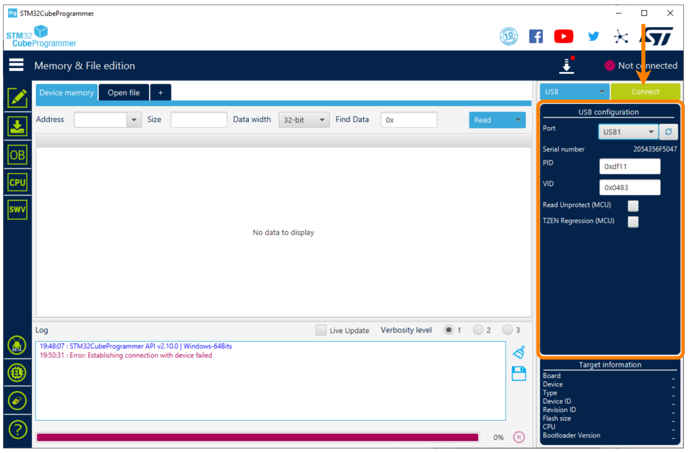
As the software is continuously improved, software updates are a crucial part and should be done whenever a new update is released. Updates are always announced on our Discord server[5]. To update the software follow those steps:

1. Download and install the STM Programmer[6] (STM32CubeProg).
2. Plug in the CATS Vega using a USB-C cable.
3. Start the configurator, connect and go to the CLI tab (described in section 3.1.2).
4. In the CLI type `bl` and send the command.
5. The CATS will disconnect. Close the configurator.
6. Start the STM32 Programmer.
7. In the programmer, on the top right, select USB (Figure 6a).
8. On the right, select a valid USB port (Figure 6b).
9. Click connect (Figure 6b).
10. Now, on the top right it should be shown that the programmer is connected to the board (Figure 6c).
11. Select Erasing & Programming on the left in the navigation (Figure 6d).
12. On the file path, select the program which you want to flash (the file ends with .hex). The latest release of the CATS software is found here[7] (Figure 6d).
13. Click start program (Figure 6d).
14. Unplug and plug the board back in. Start the configurator and verify that the the version number updated.
15. You've successfully updated the software!

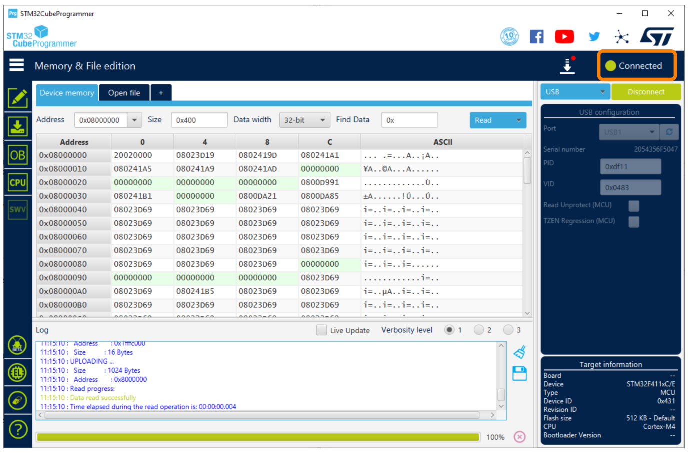
[5]https://discord.gg/r7ErmSNvsy
[6]https://www.st.com/en/development-tools/stm32cubeprog.html
[7]https://github.com/catsystems/cats-embedded/releases



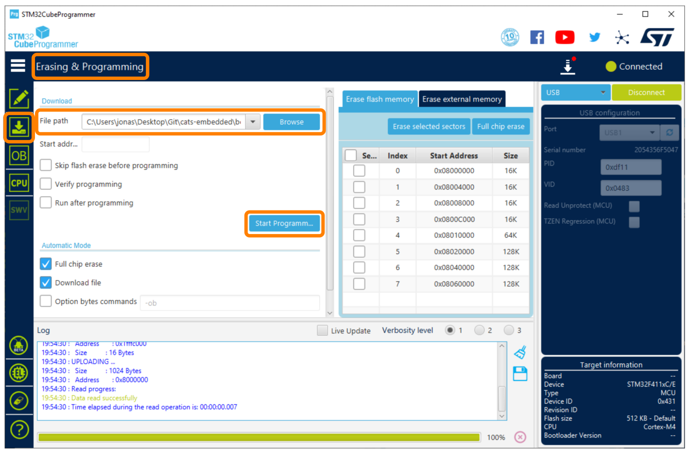
(a) Click on the drop down menu and select USB. This opens the right tab as shown on the picture right.



(b) Select the shown USB port in the "port" setting and click connect.



(c) Make sure the connected tag is shown (top right).



(d) Press on the left on Erasing & Programming. Then select the appropriate .hex file using the browse button. Once chosen, program the board by pressing "Start Program".

Figure 6: Flashing new software to the board.

3.2 Hardware

This section describes the hardware of the Vega flight computer.

3.2.1 Overview

This section gives a quick overview of the hardware and shows the location of all the ports. These ports are marked in Figure 7, matching the numbers in the list.

1. **Switch port**; A manual switch needs to be connected between the two connectors.
2. **Battery port**; A battery needs to be connected between these two ports. Pay attention to the right polarity.
3. **Buzzer**; Beeps the readiness of the flight computer.
4. **Status LEDs**; If there is power, the POWER LED is enabled. The STATUS LED blinks when everything is going well.
5. **USB Connector**; The connector is on the other side of the board.
6. **Servo Connector**; This connector fits the standard servo connectors.
7. **Telemetry LEDs**; The GNSS LED blinks every time GNSS Coordinates have been received. The LINK LED blinks when a connection to the Ground station was established.
8. **Low Level I/O and UART connector**; External hardware can be connected to the CATS board to transmit data over this connector;
9. **Pyro LEDs**; These LEDs are turned on when continuity of the Pyro channel is detected.
10. **Pyro Channel 1**; A Pyro charge can be connected to this connector.
11. **Pyro Channel 2**; A Pyro charge can be connected to this connector.
12. **Antenna Connector**; An antenna needs to be connected to this connector to send the data to the ground station from the CATS Vega.

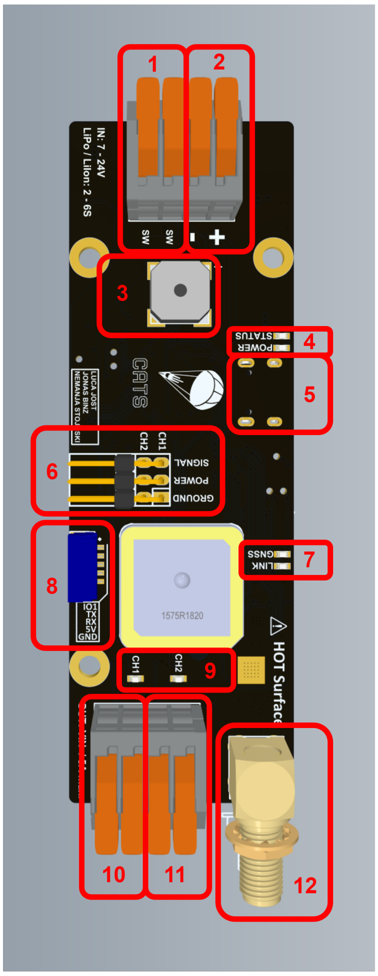


Figure 7: CATS Vega board hardware specifications

3.2.2 Specifications

Size	100 x 33 x 21 mm (without the antenna)
Weight	25 g
Input Voltage	7 - 24 V
Power Consumption	100 mA
Number of Pyro channels	2
Number of Servo channels	2
Number of IOs	1
Additional IO	UART
Servo Power	3 A max.
Microcontroller	STM32F4
Flash Memory	16MB
IMU	LSM6DSO32
Barometer	MS5607
Radio Frequency	ISM 2.4GHz
Radio Power	Up to 1W
Radio Range	Tested to 10km @100mW

Table 6: Vega Specifications

4 Ground Station

The ground station is the counterpart to the CATS Vega. It receives the data from the board and can send commands to it. It allows you to track the position and velocity of your rocket in real time as well as the health and other important information about the rocket. In the following chapters we will explain in detail how to use it and also provide more details on its working principle.

Figure 8: Ground Station

4.1 How to use

This section should be sufficient for the basic usage of the ground station. For more advanced information refer to the sections further below.

4.1.1 Explanation of All Menus

To navigate on the display, use the joystick to move left, right, up and down. Use the A button to enter a menu or select an option and the B button the go back.

Live Data

The Live data screen shows all the data received from the Vega computer as well as information about the link.
On top of the screen the current status of the flight computer is shown. Information about the rocket altitude, vertical velocity, GNSS coordinates, battery voltage and pyro continuity is shown below.
At the bottom, information about the telemetry link is displayed:

- **AGE** - The package age in seconds. If no package is received for 5 seconds, the link disconnects.
- **SNR** - Signal to noise ratio in dB. The link can be kept alive down to a SNR of -15 dB. If the SNR is low, lots of radio interference is present at your location.
- **LQ** - Link Quality in percent, this is the ratio of packages received over the last 3 seconds.
- **RSSI** - Received signal strength indication in dBm. The link can be kept alive down to a RSSI of -110 dBm. As a rule of thumb the RSSI reduces by 6 dB with every doubling of distance.

Recovery

The recovery window helps you track down your rocket once it is on the ground. The last known GNSS location of the rocket (or the current if connection is still established) is used to track your rocket. The ground stations onboard sensor suite is used to calculate the distance and direction to the rocket. For the direction to work, make sure you calibrate the devices compass outdoors in the general vicinity of the launch site with no large metallic objects close by. The calibration procedure will be explained in a later version of this manual.
The GNSS coordinates of the ground station and the rocket are shown in the recovery window, as well as the distance to the rocket. To find your rocket, simply follow the arrow!

Settings

The ground station settings are built in a way that it should be mostly self explanatory. Tool tips are shown for all options. The current Software version support the following settings:

- **Timezone**: Chose the timezone which you are operating in for correct time display.
- **Link Phrase**: Link phrase which needs to match the link phrase configured on the CATS Vega for connection.
- **Telemetry Mode**: Dual or diversity mode, described in section 4.1.2.

4.1.2 Telemetry Modes

Under the telemetry settings on the ground station you will find an option called *mode*. Since the ground station has two receivers, they can be used in two modes.
In **dual mode** the ground station can track two Vega computers. This can be useful if you separate a section of your rocket and want to keep track of it as well as the main body.
In **diversity mode**, the ground station tracks just one Vega computer. Packages are fused from both antennas making it possible to receive more data then with just one receiver. We recommend you use a directional as well as an omnidirectional antenna in diversity mode to get the best performance out of it.

4.1.3 Charging

The ground station is powered by a Li-Ion 18650 battery. With a fully charged battery the system can run for more than 10 hours. Charging the battery can be done through the USB port. With a charging current of 1 A it can take up to four hours for it to fully charge up. While the battery is charging, the LED next to the USB port will light up and turn off once it is fully charged. The internal battery can also be replaced by removing the battery cover on the back of the ground station. If the battery is replaced with one of other specification, the remaining battery estimate displayed on the screen can differ from the actual battery percentage.

4.1.4 How to get the Data on your computer

Just like the CATS Vega, the ground station can be connected to any computer and is recognized as a mass storage device. A folder will open up on your computer and you can just drag and drop the recorded logs to your preferred location. The logs from the ground station are stored in a .csv file format.

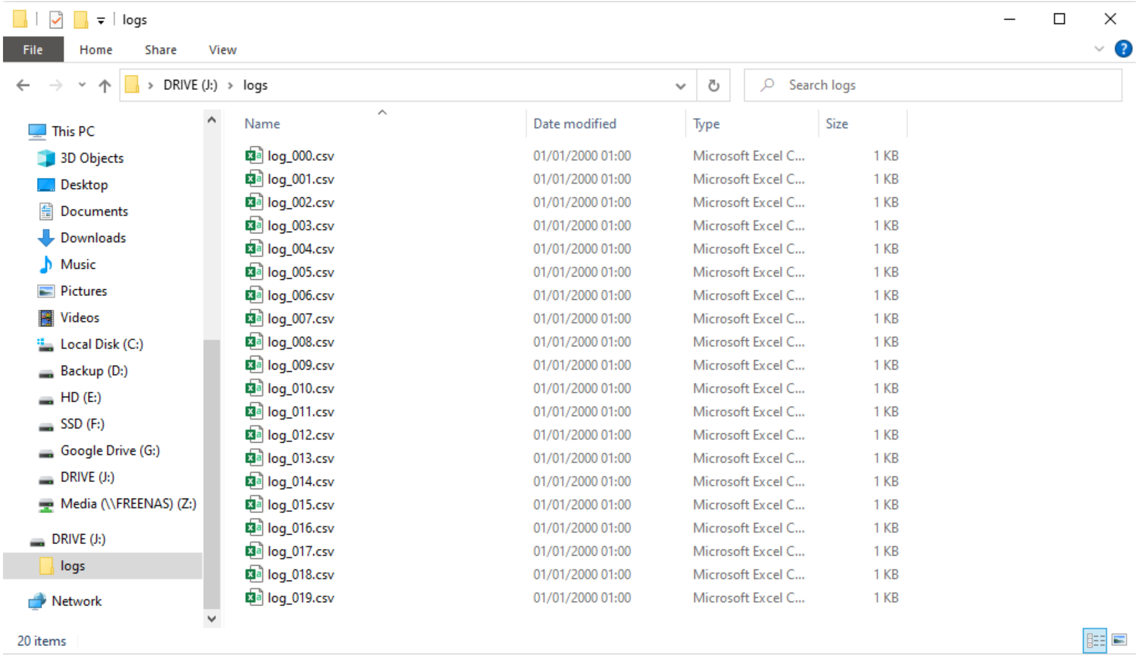


Figure 9: Ground Station Data when connecting the Ground Station to the user computer.

4.1.5 Software Updates

To do software updates on the Ground Station, DFU mode needs to be entered. First connect the ground station with your computer. On the back of the board, press and release the Reset button, followed by a press and release of the Boot button. This procedure needs to be done quite quickly for it to work. If the device successfully entered DFU mode a new mass storage device will appear on your computer with the name SAOLA1RBOOT. If the storage device does not appear repeat the procedure a few times. Once you successfully entered the firmware update mode, simply drag and drop the firmware file into the folder.
Firmware files have the file ending .UF2 and the newest version of the ground station can always be downloaded from our repository.

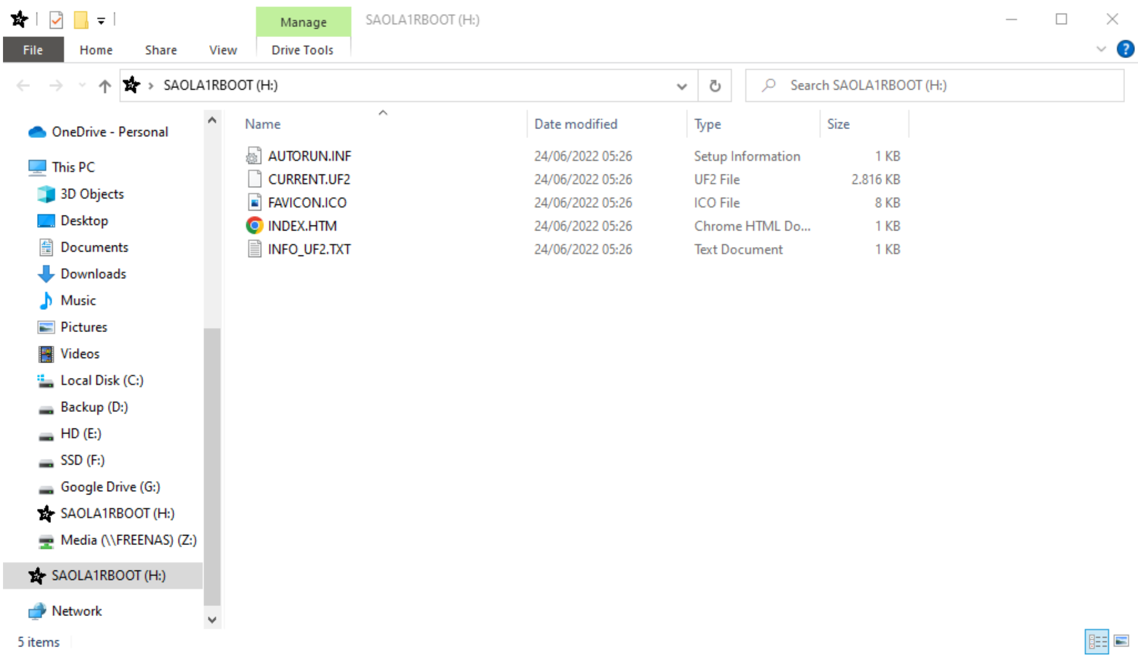


Figure 10: SAOLA1RBOOT mass storage device when successfully changing to the DFU Mode

4.2 Hardware

4.2.1 Overview

The ground station is based around an ESP32S2 microcontroller and features a beautiful transflective display making it very readable even in bright sunlight. The onboard flash can store up to 1 MB of data, enough to track over an hour of flight data.

4.2.2 Specifications

Microcontroller	ESP32-S2
Flash Memory	4MB
Battery	Li-Ion 18650
Power Consumption	60mA
Charging Current	500mA
Screen	LS027B7DH01
Radio	2x SX1280
Radio Range	Tested to 10km @100mW
GNSS	ATGM336H-5N

Table 7: Ground Station Specifications

5 Example Configuration of the CATS Vega + Ground Station

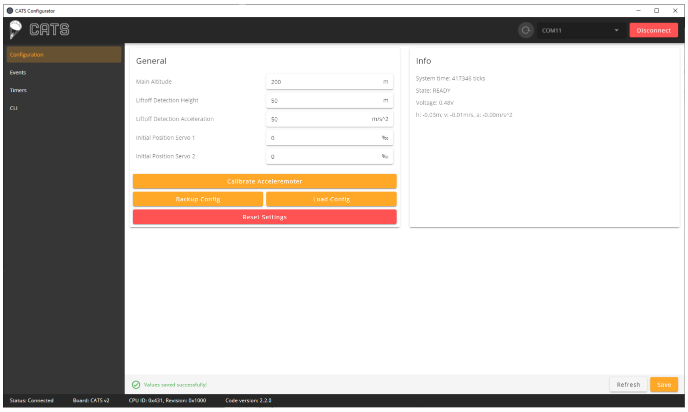
In this chapter, two example configurations are shown with the relevant configuration in the GUI and the actual setup on the hardware.
The first example configuration is simple with only a drogue chute. The second example is for more advanced rockets with also some customized events.

5.1 Simple Example Configuration

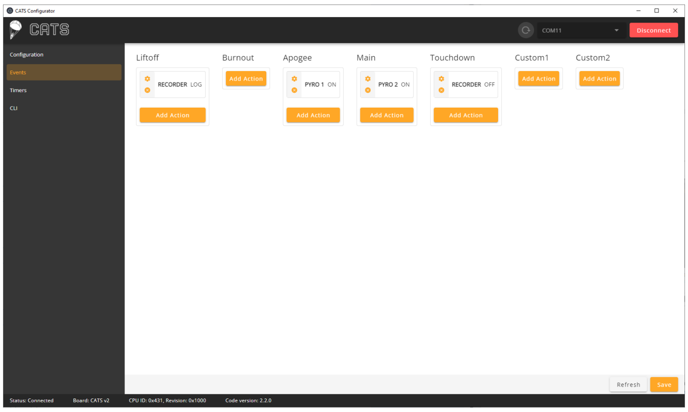
For the simple configuration let us assume the following configuration:

1. The drogue is deployed by a pyro, connected to Pyro Channel 1 that needs to be turned on (Fig. 11b).
2. Time to Apogee is 15 seconds and should be used as a backup timer (Fig. 11c).
3. The main is deployed by a pyro, connected to Pyro Channel 2 that needs to be turned on (Fig. 11b). No timer is configured for the main.
4. The liftoff acceleration is simulated to be 70 m/s^2 (Fig. 11a).

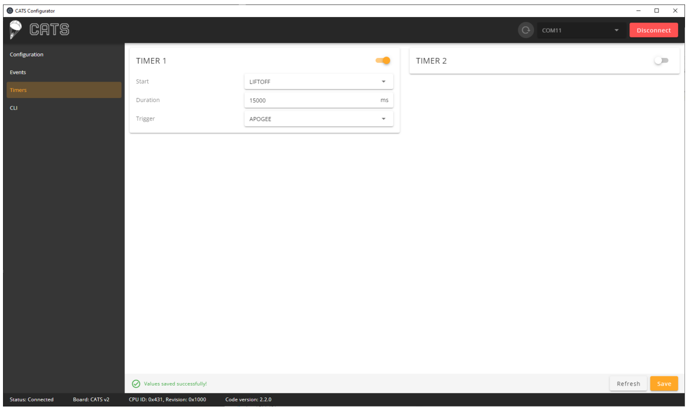
Then, the required accessories need to be added to the CATS Vega. This includes the battery, switch and Pyro Channel 1. Once those are added to the system, the CATS should look as shown in Fig. 12.



(a) Configuration Tab (simple example).



(b) Event Tab (simple example).



(c) Timer Tab (simple example).

Figure 11: Configurator screenshots for the simple example configuration.

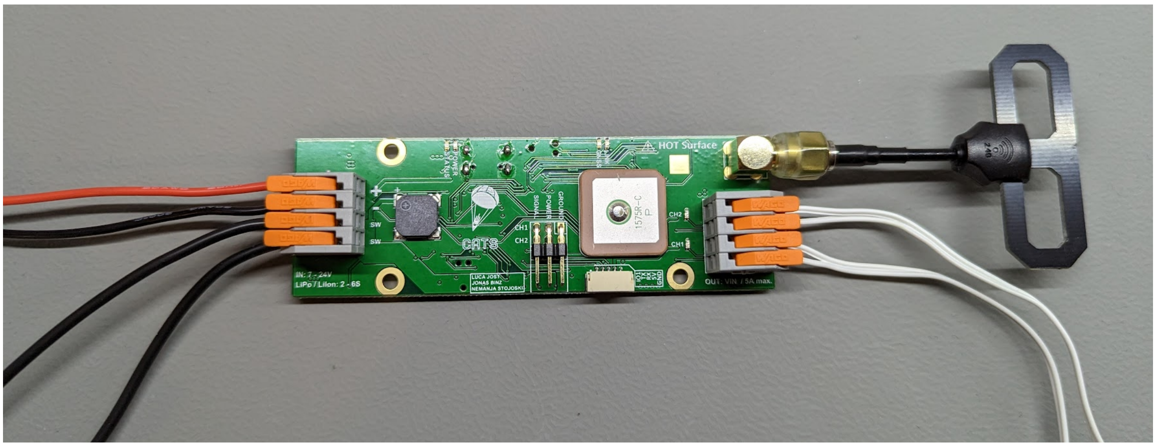


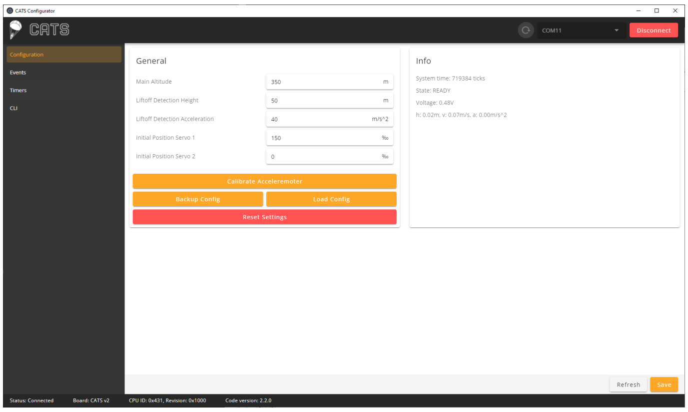
Figure 12: CATS Vega with a battery, switch and Pyro 1 connected.

5.2 Advanced Example Configuration

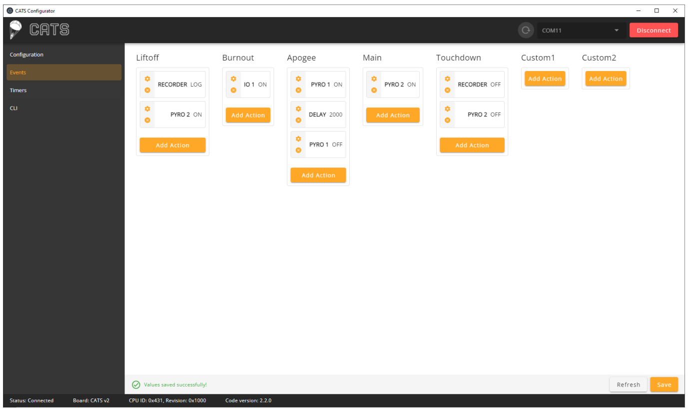
For the advanced configuration, let us assume the following:

1. The drogue is deployed by a solenoid valve, connected to Pyro Channel 1 that needs to be on for two seconds and then turned off again (Fig. 13b).
2. Time to Apogee is 45 seconds and should be used as a backup timer (Fig. 13c).
3. The main parachute is deployed by Servo Channel 1 that needs to be at 15 degrees initially, and deploys when it is at 90 degrees (Fig. 13a, Fig. 13b).
4. No main parachute backup timer is used (Fig. 13c).
5. The main parachute should be opened at 350 m (Fig. 13a).
6. The liftoff acceleration is simulated to be 60 m/s^2 (Fig. 13a).
7. A camera is connected to Pyro Channel 2 which should turn on at liftoff and turn off at touchdown (Fig. 13b).
8. The low level I/O should be turned on at burnout (Fig. 13b).

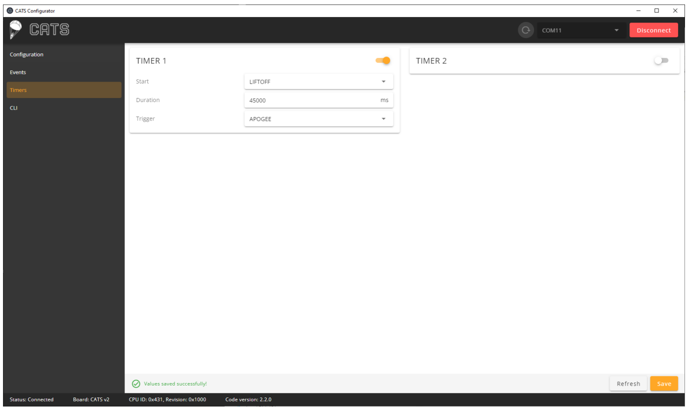
Then, the required accessories need to be added to the CATS Vega. This includes the battery, switch, solenoid valve on Pyro Channel 1, camera on Pyro Channel 2 and the Servo on Servo Channel 1. Once those are added to the system, the CATS should look as shown in Fig. 14.



(a) Configuration Tab (advanced example).



(b) Event Tab (advanced example).



(c) Timer Tab (advanced example).

Figure 13: Configurator screenshots for the advanced example configuration.

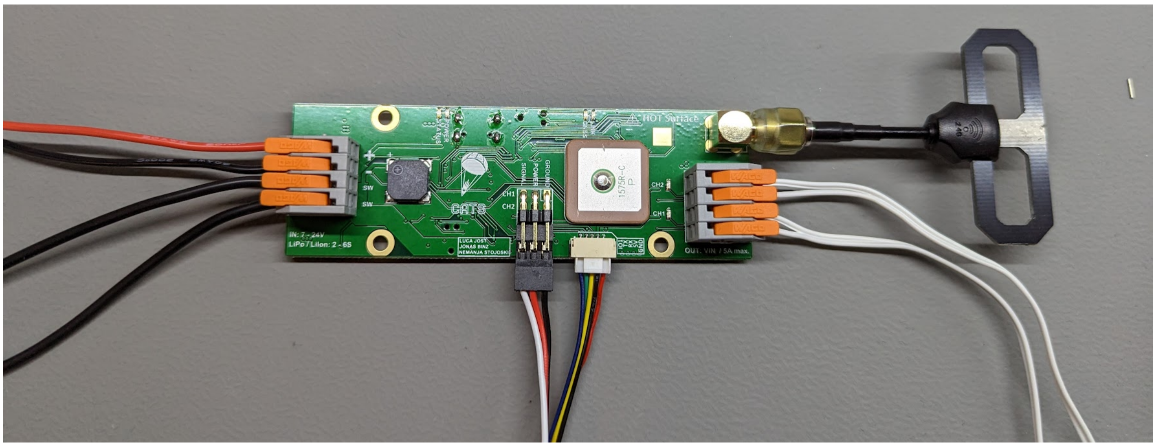


Figure 14: CATS Vega with a battery, switch, solenoid valve, camera and servo connected.

6 Testing

This section briefly explains how to run some tests with the CATS system. There are two testing modes. The first mode is to manually trigger the events with the ground station. The second mode is to have a controlled sequential triggering of all the events. The two different modes are explained below.
For both modes, first a command has to be sent to set the CATS board in the testing mode over telemetry. This can only be done while the board is in the Moving or Ready state. You also **need** to enable the testing mode over the CLI using the set command. Only then the testing mode can be enabled.

6.1 Manual Triggering

Once the board is in the testing mode, on the ground station there is a menu to actuate the different events.

Note: This feature is under development.

6.2 Sequential Triggering

Again, first the board has to be set into testing mode. Then, sequential testing can be selected. With this, the board will sequentially go through all the events, triggering one event every ten seconds. This means that only actions that have been configured beforehand are being executed.
As an example, it was configured that the Apogee event enables Pyro 1 and that the main deployment event sets Servo 2 to 90 degrees. Additionally, a timer was configured that starts at Liftoff and triggers Apogee after 15 seconds. Then, when the flight computer is set to testing, the following events will happen:

1. 10 seconds of wait
2. Ready Event → no action
3. 10 seconds of wait
4. Liftoff Event → timer is started
5. 10 seconds of wait
6. Burnout Event → no action
7. 5 seconds of wait (due to the timer)
8. Apogee Event is thrown by timer → Pyro 1 is turned on
9. 5 seconds of wait
10. Apogee Event → Pyro 1 is turned on
11. 10 seconds of wait
12. Main Deployment Event → Servo 2 turns 90 degrees
13. 10 seconds of wait
14. Touchdown Event → nothing happens

This allows you to test all the I/Os.

The testing mode is only enabled if it was enabled through the configurator. If testing is disabled, the testing command over telemetry will be ignored.
Note: Further information about the testing and the steps to take to do testing will follow in a next release of the manual.

7 Advanced Information

In this section, the core pieces of the CATS Vega board are explained in more detail. This part of the manual is written for advanced users.

7.1 Software Overview

In this section we briefly touch on how the software is set up. This section is for advanced users with some coding experience and in no way does the user need to understand this section to be able to use the flight computer.
The software is implemented in C++. The backbone of our implementation is FreeRTOS. All hardware is first initialized and at the end of the initialization the tasks are started. An overview of the running tasks is shown in Figure 15.

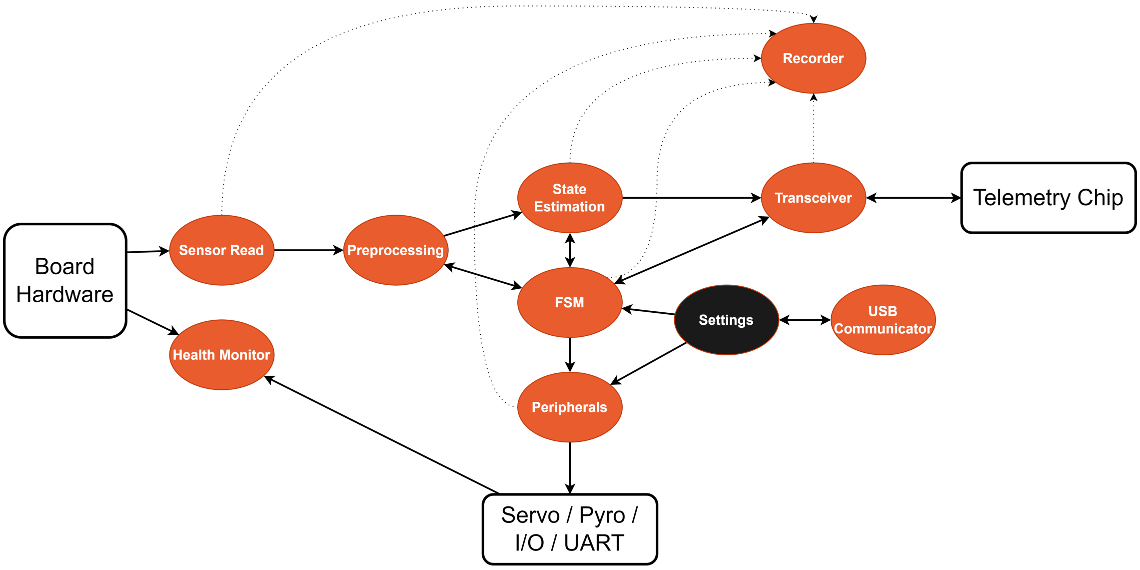


Figure 15: Illustration of the different FreeRTOS tasks interacting with each other and the hardware. The black circle 'Settings' is just a memory region which is being accessed by different tasks.

A brief explanation of every task is given here.

- **Sensor Read** reads out the Barometer and the IMU and writes those values to a globally shared variable.
- **Preprocessing** takes this raw data and transforms barometric pressure to height above ground level and linear acceleration to acceleration in the 'up' direction. It then writes this data again to a shared memory region.
- **State Estimation** estimates the current height and velocity based on the Preprocessing task's output and the FSM state.
- **FSM** computes the current flight phase based on the estimate from the State Estimation and Preprocessing tasks.
- **Transceiver** handles the communication between the main chip and the telemetry chip.
- **USB Communicator** handles the interface between the flight computer and the user computer. If no USB is connected, this task is not started.
- **Peripherals** triggers all user-defined events.
- **Health Monitor** checks values like the buzzer and the battery voltage.
- **Recorder** records all the data using a queue to the flash chip.

7.2 Telemetry

The telemetry system is based on 2.4 GHz LoRa and utilizes FHSS for the transmission. A key benefit of FHSS is that it makes transmissions much more resistant to interference and more difficult to intercept. Allowing more devices on the same frequency band with little or no impact on the link quality.

Hopping Pattern

The hopping pattern is defined with a link phrase. This link phrase is hashed using a CRC32 algorithm, and the resulting CRC is then used as the seed for a pseudo-random number generator. The generator is run 20 times, defining the hopping pattern. With this method, a link phrase always generates the same hopping pattern. Both the transmitter and receiver require the same link phrase for the transmission to work.

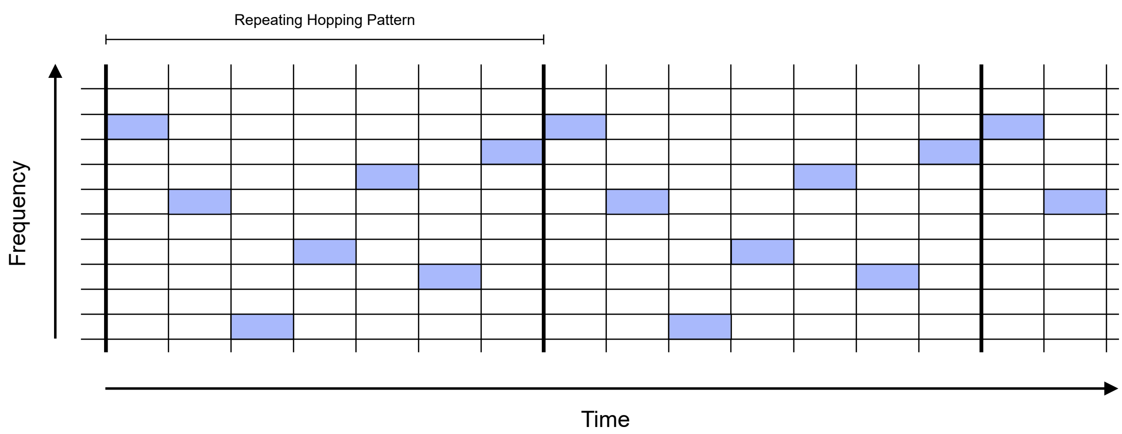


Figure 16: FHSS transmission example

Synchronization

The receiver waits on the first frequency until a sync packet is received. The sync packet contains the link CRC to identify the transmission source. If the remote CRC matches the local CRC, the receiver hops to the subsequent frequency, waiting for data. Each data package contains a checksum to validate the contents. The time between packages is measured and used to jump to the next frequency when no package was received in the estimated time frame. A total of 30 hops can be performed without receiving a package before the synchronization is lost. On connection loss, the receiver returns to the first frequency.

7.3 Estimation Algorithms

The state estimation is used to estimate the velocity and the height of the rocket. For this, it uses the barometric pressure data and the linear acceleration in the z direction.

Calibration of Sensors

As soon as the system enters the Ready state, the linear acceleration is calibrated. This allows you to mount the flight computer in any direction.
During Moving or Ready, the current height above sea level is always estimated. The height above ground level, which is the actual important value is always calculated based on the height above sea level. To calculate the height above sea level we assume that the barometric pressure is very slowly varying. At the moment when Liftoff is detected, the height above sea level is locked in place and only the height above ground level is updated. With this approach, we can make sure that no accidental Liftoff is detected as a slow pressure decrease would lead to a changing height above sea level, never triggering the Liftoff event based on pressure.

Kalman Filter

To estimate the height and velocity based on the calibrated values a Kalman Filter is implemented. Here the derivation of the filter is described. We assume the state and the noise to be

$$x(t) = \begin{pmatrix} h(t) \\ v(t) \\ a_o(t) \end{pmatrix} \qquad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

where h is the height above ground level, v is the vertical velocity and a_0 is the estimated offset of the measured vertical acceleration. v_1 is the noise applied on the vertical acceleration and v_2 is the noise variable used to estimate the linear acceleration offset.
Additionally, we need to define the input to the system, which, in this derivation is $u(t)$ being the linear acceleration measured in z direction.

$$\dot{x}(t) = \begin{pmatrix} v(t) \\ a(t) \\ \dot{a_0}(t) \end{pmatrix} = A \cdot x(t) + B \cdot u(t) + G \cdot v(t) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} h(t) \\ v(t) \\ a_o(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \cdot u(t) + \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix}$$

this is discretized using a first order approximation

$$A_d = e^{AT_s} \qquad B_d = \int_0^{Ts} e^{A_d t} \delta t \cdot B \qquad G_d = \int_0^{Ts} e^{A_d t} \delta t \cdot G$$

giving the system

$$x(k+1) = A_d \cdot x(k) + B_d \cdot u(k) + G_d \cdot v(k)$$

The process measurement noise matrix becomes

$$Q(k) = \begin{pmatrix} Q_{acc}(k) & 0 \\ 0 & Q_{acc_0}(k) \end{pmatrix}$$

The measurement step assumes, that the height is already calculated from the barometric pressure. For this we use the standard barometric pressure formula.

$$z(k) = h_{meas}(k) + \omega(k) = \left(\frac{p(k)}{p_0}^{\frac{1}{5.257}} - 1\right) \cdot \frac{T_0 + 273.15}{L} - h_0 + \omega(k)$$

with $L = -0.0065$, $T_0 = 15$ C, $p_0 = 101250$ Pa, $h_0 =$ calibrated altitude above sea level and ω being the measurement noise.
the measurement function becomes, very easily

$$z(k) = H \cdot x(k) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} h(k) \\ v(k) \\ a_o(k) \end{pmatrix}$$

the measurement noise matrix becomes a scalar.

$$R(k) = R_{height}$$

And then, the standard Kalman filter equations can be used to propagate the state.

Gain-Scheduling

The thought behind gain scheduling this algorithm is that we want to reduce the trust on the barometer during high velocity phases. We do this because in the transonic regime the barometer behaves unpredictably. With the gain scheduling, this unpredictable behaviour is ignored during high velocity phases.
What we want is that at Liftoff and while the speed of the rocket is large, the accelerometer is given much more weight in the estimation of the height and the velocity. Once we get to smaller velocities, we want to trust the barometric pressure more as accelerometer drift impacts the estimate. Additionally, the rocket arcs over, further decreasing the quality of accelerometer measurement.
We have two variables that affect the trust of the two sensors, Q_{acc} and R_{height}. In the algorithm, Q_{acc} is kept constant and only R_{height} is changed during flight.
Further below the conditions are given for the changes in R_{height}.

$$R_{height} = \begin{cases} R_{initial}, & \text{for state = MOVING or IDLE} \\ R_{max}, & \text{for state = LIFTOFF} \\ R_{max} \cdot f(v), & \text{for state = COASTING} \\ R_{initial}, & \text{otherwise} \end{cases}$$

With this gain scheduling, unexpected barometric measurements during high velocities are effectively filtered out.

7.4 Description of the CLI

This section describes all commands available on the CLI. To access the CLI, plug in the board, connect to the configurator and navigate to the CLI tab.
In the list below, square brackets [] mean that the argument is optional and <>specify the parameter name.
Whenever changes to the configuration are done using the CLI, we heavily encourage to recheck the configuration using the `config` command.

`bl`	Put the board into DFU mode	needed for software updates (refer to section 3.1.9)
`cd`	Change the current working directory	
`config`	Print the flight config in a human-readable format	
`defaults [--no-outputs]`	Reset to default settings	The default configuration triggers pyro channels. If –no-outputs is passed the pyro triggering will not be set.
`dump`	Print configurable settings in a paste-able format	
`flash_erase`	Erase all sectors on the flash chip, this might take a while	
`flash_test`	Test writing and reading from the flash	Only for testing purposes, should not be used!
`flash_start_write`	Start writing to flash	Only for testing purposes, should not be used!
`flash_stop_write`	Stop writing to flash	Only for testing purposes, should not be used!
`flight_dump <flight_id>`	Print a specific flight in binary format	
`flight_parse <flight_id>`	Print a specific flight in a human-readable format	
`get [<variable>]`	Get a variable value	
`help [<command name>]`	Display all commands with a description	
`lfs_format`	Reformat flash file system	
`log_enable`	Enable log output on the terminal	
`ls [<path>]`	List all files in current working directory	

`reboot`	Reboot the computer	This command does not save the changed settings by itself
`rec_info`	Get information about flash usage	
`rm [<path>]`	Remove a file	
`save`	Save flight configuration	
`set [<variable>=<value>]`	Set a variable	
`sim`	Simulate a flight	Experimental!
`stats <flight_id>`	Print flight stats	
`status`	Show current sensor data, flight phase and other important information	
`version`	Show the firmware version	

Table 9: Exhaustive List of CLI Commands

8 FAQ

To be extended with user feedback...

9 Warranty

The use of the CATS System is at your own risk. It is recommended that the CATS System is always used in conjunction with a second safety system that operates in a different manner, such as a motor ejection or another electronic system, to ensure maximum safety.
The manufacturer is not liable for any damages that may occur as a result of using the CATS System, and will not be held responsible for any damages inflicted on third parties.
The manufacturer cannot be held liable for any program errors or malfunctions in the software.
The hardware warranty covers manufacturing errors and defects in workmanship and materials for a period of two years from the date of purchase. This warranty does not cover damage caused by improper handling, accidents, or other external factors. In the event of a covered manufacturing error or defect, the manufacturer will repair or replace the hardware at no cost to the user. To make a claim under this warranty, the user must provide proof of purchase and report the issue to the manufacturer within the warranty period. The manufacturer reserves the right to inspect the hardware and determine the cause of any reported issue before providing a repair or replacement. This warranty is non-transferable and only applies to the original purchaser of the hardware.