



# CATS User Manual





## Contents

<b>1 Revision History</b>	<b>4</b>
<b>Glossary</b>	<b>5</b>
<b>2 Introduction</b>	<b>7</b>
2.1 Coverage of This Manual . . . . .	7
2.2 Module Overview . . . . .	7
2.3 Basic Functionality . . . . .	7
<b>3 CATS Vega</b>	<b>8</b>
3.1 Hardware . . . . .	8
3.1.1 Specifications . . . . .	8
3.1.2 Hardware Overview . . . . .	9
3.2 Working Principle . . . . .	9
3.2.1 Configurable Actions & Finite State Machine . . . . .	9
3.2.2 Actions . . . . .	11
3.3 How to Use . . . . .	12
3.3.1 Connection to Your Computer . . . . .	12
3.3.2 Description of the Configurator . . . . .	12
3.3.3 Mounting . . . . .	19
3.3.4 Battery, Switch and Actuators . . . . .	19
3.3.5 Setting up the Minimal Flight Configuration . . . . .	22
3.3.6 How to Get the Data on Your Computer . . . . .	23
3.3.7 Visualizing the Flight Data . . . . .	23
3.3.8 Software Updates . . . . .	24
3.4 Beeping Patterns . . . . .	28
<b>4 Ground Station</b>	<b>30</b>
4.1 Hardware . . . . .	30
4.1.1 Overview . . . . .	30
4.1.2 Specifications . . . . .	31
4.2 How to Use . . . . .	31
4.2.1 Explanation of All Menus . . . . .	31
4.2.2 Telemetry Modes . . . . .	33
4.2.3 Charging . . . . .	33
4.2.4 How to Get the Data on Your Computer . . . . .	33
4.2.5 Software Updates . . . . .	34
<b>5 Example Configuration of the CATS Vega + Ground Station</b>	<b>35</b>
5.1 Simple Example Configuration . . . . .	35
5.2 Advanced Example Configuration . . . . .	37
<b>6 Testing</b>	<b>39</b>
6.1 Working Principle . . . . .	39
6.2 Enabling the Testing Mode . . . . .	39
<b>7 Advanced Information</b>	<b>43</b>
7.1 Software Overview . . . . .	43
7.2 Telemetry . . . . .	44
7.3 Estimation Algorithms . . . . .	45
7.4 Using the Python Plotting Tool . . . . .	47
7.5 Description of the CLI . . . . .	48
7.5.1 Get and Set Commands . . . . .	49
<b>8 FAQ</b>	<b>52</b>





## 1 Revision History

Date	Version	Comment
15 December 2022	1.0.0	Initial release
05 March 2023	1.1.0	Addressing first round of feedback; added telemetry explanations

Table 1: Revision History



## Glossary

**Apogee** The highest point of the rocket during a parabolic flight. 7, 10, 11, 16, 17, 22, 35, 37

**Barometer** A sensor which can measure the barometric pressure of its surroundings. 8, 29, 43

**Calibrating** A flight state; when the flight computer is this state, it is in safe mode; no actions are taken by the flight computer. 10, 28, 45

**CATS** Control and Telemetry Systems. 7, 9, 12, 19, 24, 28, 30, 33, 35, 37, 39, 51

**CLI** Command Line Interface, used to set and get values from the flight computer when not using the configurator. 14, 24, 49

**CRC** Cyclic Redundancy Check, used to verify that received data is not corrupted. 44

**DFU** Device Firmware Update, a mode used to flash new firmware to an embedded system. 34

**Drogue Chute** Parachute which is deployed at apogee. Usually a rather small parachute only decelerating the rocket for a controlled descent. 22

**FHSS** Frequency Hopping Spread Spectrum, a protocol for sending data over telemetry by using more than one carrier frequency. 44

**FreeRTOS** An operating system for embedded CPUs, the backbone of the CATS Software. 43

**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. 9

**GNSS** Global Navigation Satellite System, for example GPS / Galileo. 9, 19, 32

**I/O** Input/Output, used to group all inputs and outputs of the system in one word. 7, 9, 19–21, 37

**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). 8, 10, 43

**Kalman Filter** A filtering technique used to estimate states according to physical laws and measurements. 45

**Liftoff** An event during flight, when this event happens, the motor of the rocket starts to accelerate and the rocket takes off. 11, 14, 17, 19, 22, 45, 46

**Main Chute** Parachute which is deployed at a certain height above ground level. Usually a bigger parachute, slowing the rocket down enough to prevent damage once it hits the ground. 14, 22

**Patch Antenna** A flat antenna with high directivity. The CATS Vega board has a patch antenna, which is used to get the GNSS signal. 19

**Power Supply** Needed to power the system, in rockets usually some battery. 7, 20

**PWM** Pulse Width Modulation, protocol used to actuate most modern servos. 20

**Pyro** A charge with two wires which can be ignited with a high current applied. 8–11, 16, 19–21, 35, 37

**Ready** A flight state; when in this state, liftoff can be detected. 10, 15, 19, 23, 28, 45

**RF** Radio Frequency. 19



**Servo** A small electrical motor where the position of the motor can be set by applying a PWM signal.  
8, 10, 11, 14, 19–22, 37

**Touchdown** Final phase of the flight, when the rocket lands on the ground. 11

**UART** A communication protocol to send data from one location to another over three wires. 8, 9



## 2 Introduction

Hi 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<sup>1</sup>. This is the first release of the manual and 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<sup>2</sup>.

### 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<sup>1</sup>.

### 2.2 Module Overview

The CATS system has three main components.

**CATS Vega:** This is the flight computer and the piece of hardware which is placed inside your rocket.

**Ground Station:** This is the receiver which displays all the flight data while your rocket is flying and allows you to track your rocket once it landed.

**Configurator:** This is the software which you can install on your computer and which is used to configure the CATS Vega and to visualize your flights after you've flown.

### 2.3 Basic Functionality

This section gives a very broad overview of the capability of the CATS System to users new to rocketry. Everything described here is explained in much greater detail in the following sections.

The CATS Vega is the heart piece of the system. Its main goal is to eject the main chute at Apogee and the drogue chute at a user-configurable height above ground level. To achieve this goal, barometric and accelerometer data is acquired and fused to estimate the height above ground level and the velocity.

The CATS system defines 5 **events**: *Liftoff*, *Burnout*, *Apogee*, *Main Deployment* and *Touchdown*. Those are triggered automatically by the onboard control system. What is being done, however, during those events is left to the user. That's where actions are configured.

For each event, up to 8 **actions** can be mapped. Those can be delays, pyro and servo channel triggers or low level I/O signals. This is how the user can select how their deployment mechanism shall be actuated. Most commercial recovery mechanisms utilize pyro channels or servo channels, and the CATS Vega has two of each.

Simultaneously, all the important data is transmitted to the ground station. This includes GNSS data, current height, velocity and flight state. The data is also logged onto the onboard flash.

The ground station is also used for the testing mode. In this mode, the user can trigger their defined events to test their separation mechanisms.

---

<sup>1</sup><https://discord.gg/r7ErmSNvsy>

<sup>2</sup><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.

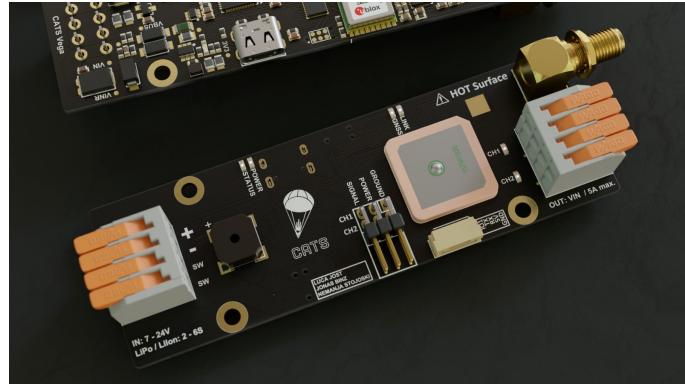


Figure 1: CATS Vega

#### 3.1 Hardware

##### 3.1.1 Specifications

Size	100 x 33 x 21 mm (without the antenna)
Weight	25 g
Input Voltage	<u>7 - 24 V</u>
Power Consumption	100 mA
Number of Pyro channels	2
Number of Servo channels	2
Number of IOs	1
Additional IO	UART
Servo Power	5V / 3 A max.
Microcontroller	STM32F4
Flash Memory	16MB
IMU	LSM6DS032
Barometer	MS5607
Radio Frequency	ISM 2.4GHz
Radio Power	Up to 1W
Radio Range	Tested to 10km @100mW

Table 2: Vega Specifications



### 3.1.2 Hardware Overview

Here, we give a quick overview of the hardware and shows the location of all the ports. These ports are marked in Figure 2, 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. The beeping patterns are explained in section 3.4.
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. Two servos can be connected to this connector.
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 red 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.

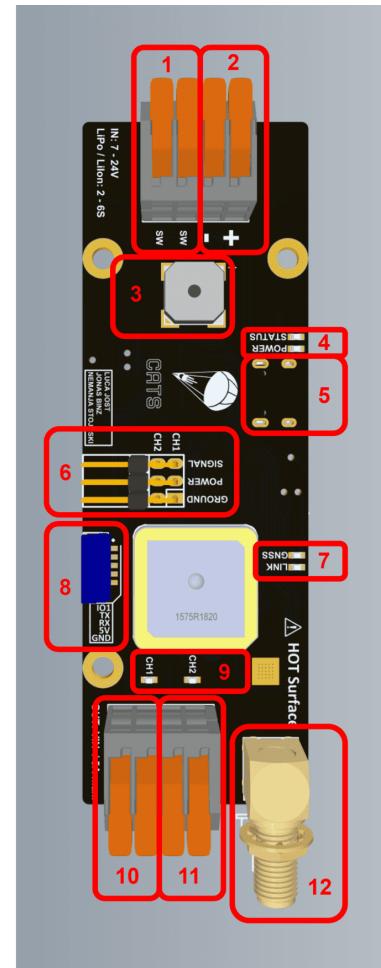


Figure 2: CATS Vega board hardware specifications

## 3.2 Working Principle

In this section the working principle is shortly introduced as some understanding of it is essential to properly understand the configuration options of the Vega flight computer.

### 3.2.1 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 3. When the flight computer is turned on, it is initialized to the Calibration state. Every rocket flight follows the flight states outlined in Figure 3. Whenever a flight transition happens, the associated event is thrown.

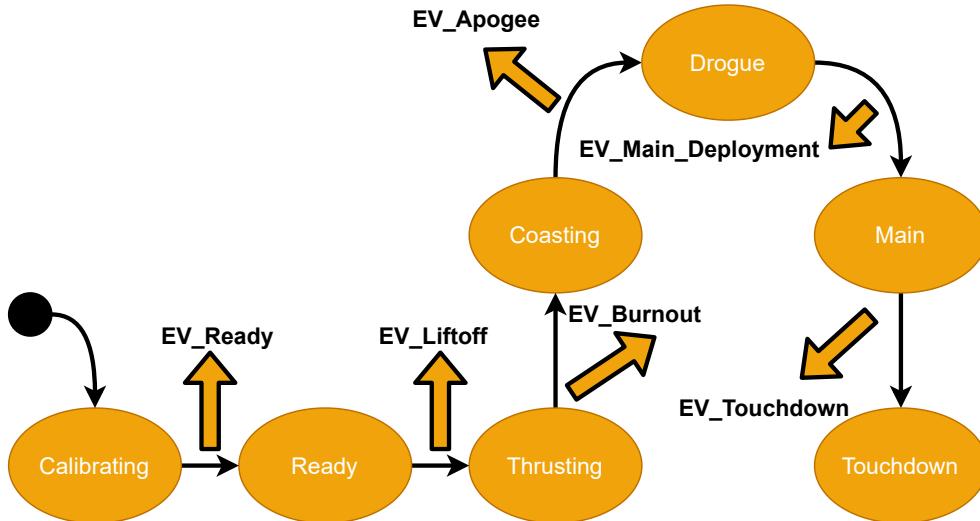


Figure 3: Finite State Machine controlling the CATS software.

Each event can be used to trigger some actions (i.e., a Pyro channel, a Servo channel, a timer, etc.). The user configures which actions are triggered during which event by the use of the configurator. Configuration of actions is described in section 3.3.2.

Calibrating → Testing	Telemetry input to put it into testing mode. Only works, if the testing mode was activated over the configurator.
Calibrating → Ready	IMU (gyroscope and linear acceleration) readings are constant for 10 seconds.
Ready → Thrusting	The measured acceleration in any direction is larger than the user defined acceleration value for <u>0.1 seconds</u> <b>OR</b> <u>the height above ground level</u> is larger than the user defined value for 1 second.
Thrusting → Coasting	The measured acceleration in the "up" direction is smaller than $0 \text{ m/s}^2$ for 0.1 seconds
Coasting → Drogue	The estimated velocity needs to be smaller than $0 \text{ m/s}$ 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 $[-3, 3] \text{ m/s}$ for 1 second

Table 3: 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. Events are also **unique**; during a flight only one event can be thrown.



**Note:** A software safeguard checks the time between Liftoff and Apogee. If that time is smaller than 1.5 seconds, the flight computer assumes a faulty liftoff detection and no further events or actions are activated. The flight computer then jumps instantly to Touchdown without triggering anything.

### 3.2.2 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	[0-1000]%
Servo 2	[0-1000]%
Low Level I/O	ON/OFF
Delay	[0-15000] ms
Recorder	ON/OFF/PREFILLING

Table 4: Exhaustive List of all possible Actions

How actions can be configured is shown in section 5.



### 3.3 How to Use

Now that an overview of the hardware and software was given, we explain here how to set up your flight computer, how to mount it in your rocket, how to do software updates, how to generate plots of your flight data and much more.

#### 3.3.1 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<sup>3</sup> where we describe some troubleshooting steps.

#### 3.3.2 Description of the Configurator

The configurator allows you to configure the CATS system from your computer. The configurator can be downloaded here<sup>4</sup>. Make sure you download the latest stable version. Release candidates are marked as such and should only be used for testing new features.

##### 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 4) 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 your communication port is still not showing up, refer to our wiki for some troubleshooting steps.

1: Refresh Button	Used to refresh the shown communication ports in label 2
2: Com Port Selection	Selection of the communication port to connect to the CATS Vega
3: Connect Button	Press to connect to the CATS Vega
4: Home Screen	Some general information about the Configurator
5: Connection Status	Information box, tells you that the flight computer is currently not connected
6: Navigation	Greyed out as long as no flight computer is connected. Once it is connected, used to navigate between the tabs.
7: Connection Status	Shows the connection status
8: Flight Log Graph	Load a cats flight log (.cfg) file to plot your flight. More information in Section 3.3.7

Table 5: Overview of the different Settings for the Home Tab

<sup>3</sup><https://github.com/catsystems/cats-embedded/wiki/Installation>

<sup>4</sup><https://github.com/catsystems/cats-configurator/releases>

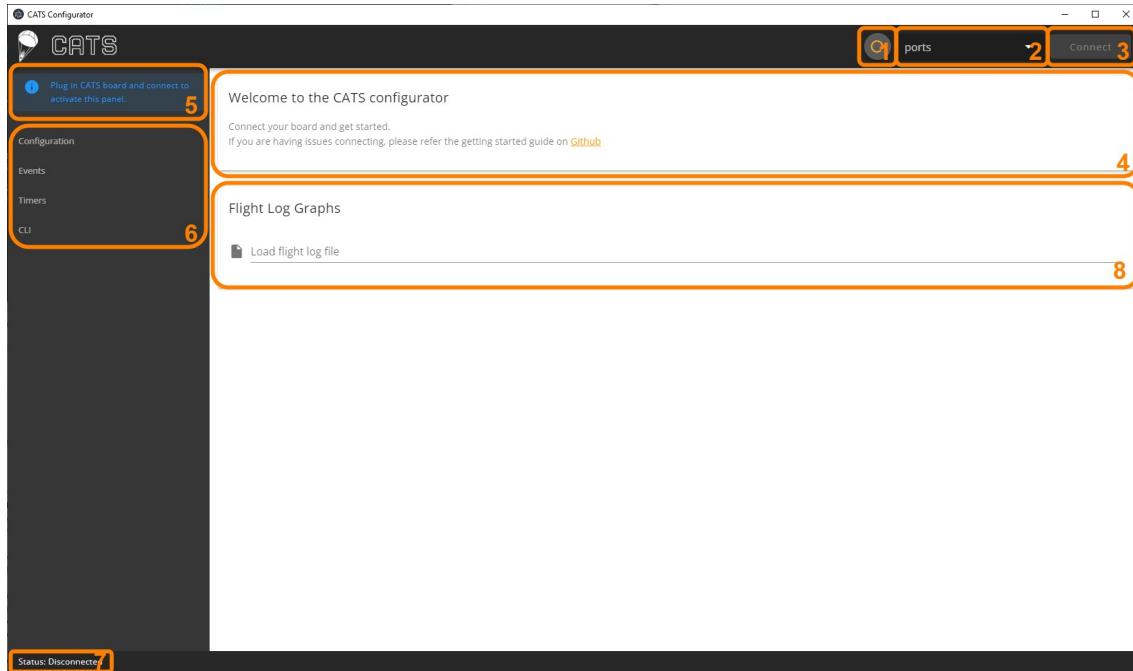


Figure 4: Home Menu.

## Configuration Tab

This tab shows you the status of the flight computer and lets you configure various parameters. Figure 5 shows the Configuration tab. In the following Table, the labels are explained in more detail.

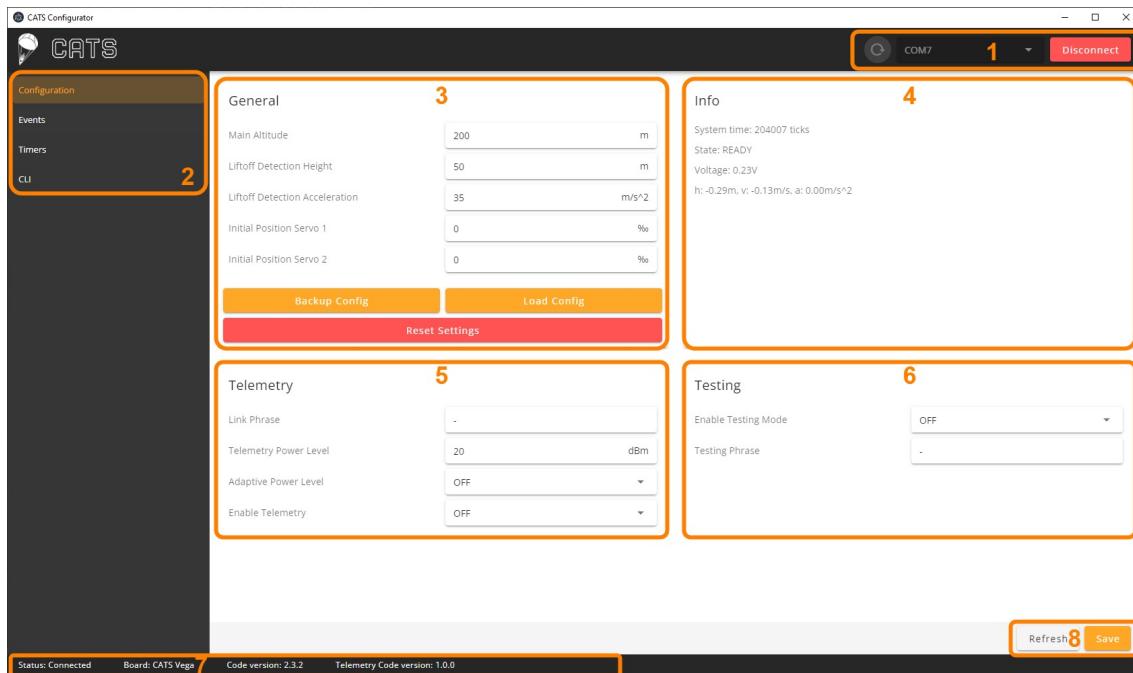


Figure 5: Configuration Menu.



1: Communication Port	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
2: Navigation	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
3: General	Main Altitude	Set the desired height above ground level where the Main Chute should be deployed
	Liftoff Detection Height	Height Threshold above which Liftoff should be detected. Restricted between 50 and 100 m.
	Liftoff Detection Acceleration	Acceleration Threshold above which Liftoff should be detected. We recommend a value of $40 \text{ m/s}^2$ for most flights
	Initial Position Servo 1	Initial Servo angle of the Servo connected to Servo port 1. If you do not use Servos, no settings need to be set here. The flight computer will actively steer to that position at start-up.
	Initial Position Servo 2	Same as for Servo 1 but for the second channel.
	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 configuration, previously backed up on your computer, to the flight computer.
	Reset Settings	Set the default parameters of the flight computer.

Table 6: Overview of the Configuration Tab



4: Info	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 shown)
	State Estimation	The currently estimated height above ground level, velocity and acceleration
5: Telemetry	Link Phrase	4 - 16 character link phrase which needs to be set and needs to match the link phrase set on the ground station for a connection.
	Telemetry Power Level	Power level of the telemetry. The higher the value, the better the connection is. <b>Only set a value above 20 dBm if you have a HAM License!</b>
	Adaptive Power Level	If this is enabled, the configured power level is applied during the Ready state and at liftoff, the max power level is set for optimal reception during flight.
	Enable Telemetry	Set to ON or OFF to enable or disable telemetry.
6: Testing	Enable Testing Mode	If this is set to ON, the next time the flight computer reboots, it enters testing mode. It then needs to be armed by telemetry using the ground station. Make sure to read section 6 thoroughly for more information about testing.
	Testing Phrase	4 - 16 character testing phrase which should be the same than the one configured on the ground station. Make sure to read Section 6 thoroughly for more information about testing.
7: Hardware Info	Status	Connection status of the board
	Board	CATS Board Name
	Code Version	Current code version
	Telemetry Code Version	Current code version on the telemetry chip
8: Save Settings	Save	Save the current settings to the board. <b>Attention:</b> if Save is not pressed, the values are not saved to the board!
	Refresh	Refresh the displayed values. When this button is pressed, the saved values from the flight computer are fetched and shown.

Table 6: Overview of the Configuration Tab (Cont.)



## Event Tab

This tab shows you the configuration of the individual event. Each event is described more closely in Section 3.2.1. 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 6), the events Liftoff, Apogee, Main and Touchdown have mapped events, meaning that the flight computer only acts when those events are triggered. At Liftoff, the recorder is enabled. At Apogee, Pyro channel 1 is turned on. At Main deployment Pyro 2 is enabled. At Touchdown, the recorder is disabled.

Figure 7 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 taken 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.

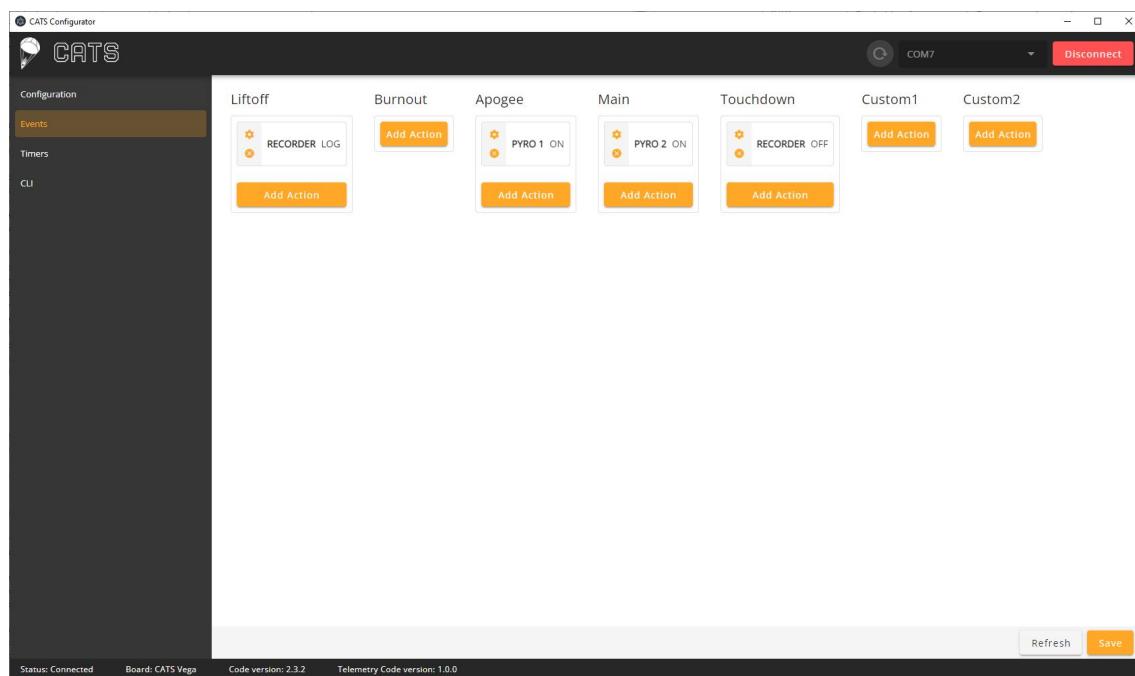


Figure 6: Event Menu.

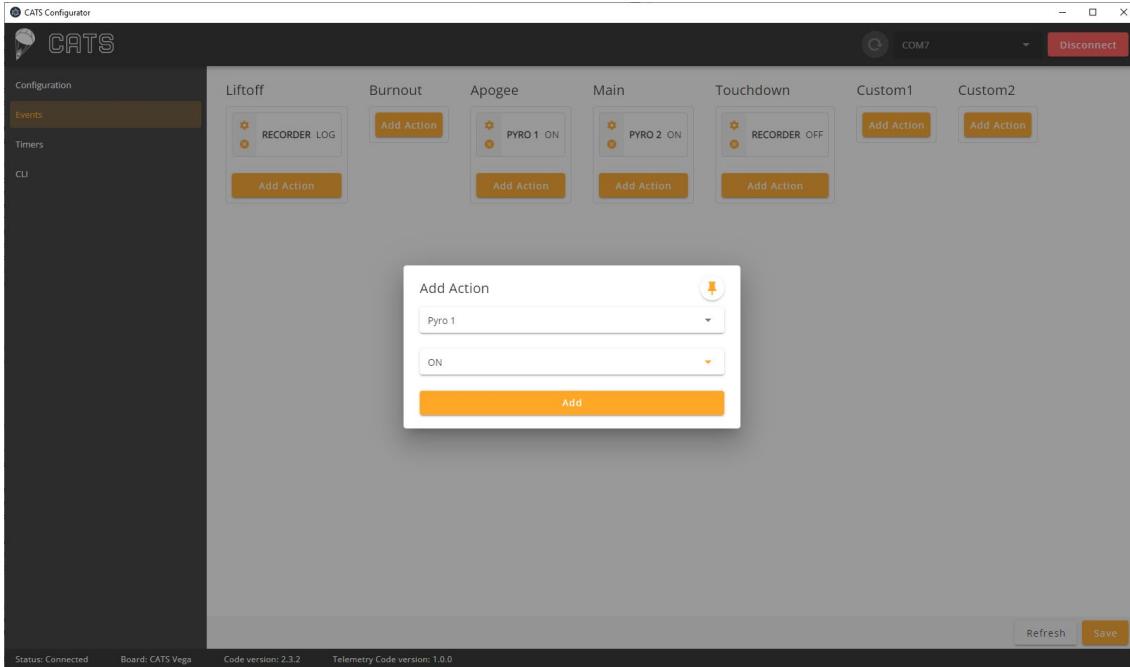


Figure 7: Configuring an Event in the Event Menu.

## 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 8, Timer 1), the start event can be defined, the duration in ms and the end event which shall be triggered. In the example picture 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. This, however does not put the flight computer in the apogee phase! It only triggers that event.

**Note:** Events are unique, meaning that if a timer is used to trigger apogee, only one apogee event will be thrown by the flight computer. It will either be the timer or the event from the estimation, whichever is thrown first.

## Custom Events

As events are unique, two custom events are provided to the user. To activate these custom events, a timer must be started with the end event being the custom event. This allows to apply any actions at an arbitrary point in time.

This can be in particular interesting for payload experiments. If you would like to actuate something 10 seconds after apogee, you can set a timer starting from apogee, running for 10 seconds and activating custom event 1. Then you can map the desired actions to that event.

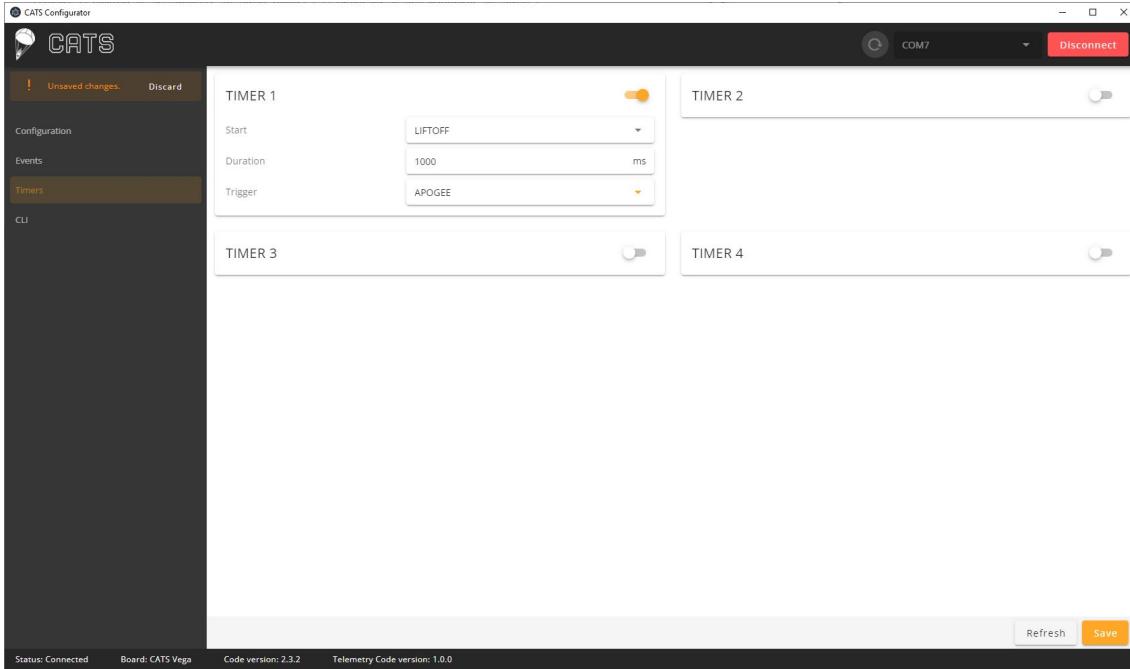


Figure 8: Timer Menu.

## CLI

The CLI tab allows the user to send manual commands to the CATS board. All supported commands are explained in detail in section 7.5. A screenshot of the CLI is shown in Figure 9.

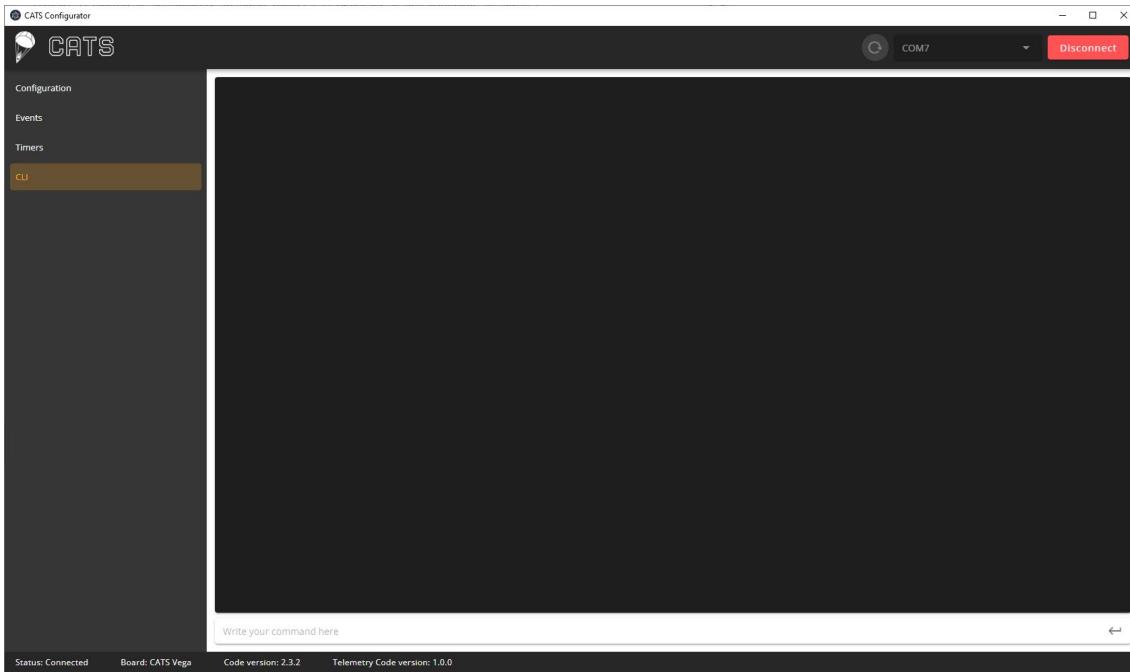


Figure 9: Command Line Interface.



### 3.3.3 Mounting

The CATS Vega **does not** require a specific mounting orientation. The gravity vector 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 github<sup>5</sup>.

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. It will then beep that it is in the Ready state and also show it on the ground station. Once the flight computer is in Ready mode, it is armed and waiting for Liftoff. Do not move the rocket and make sure you follow safety guidelines. Disarming the flight computer at that stage can only be done by turning it off. For more information regarding calibration refer to sections 7.3 and 3.2.1.

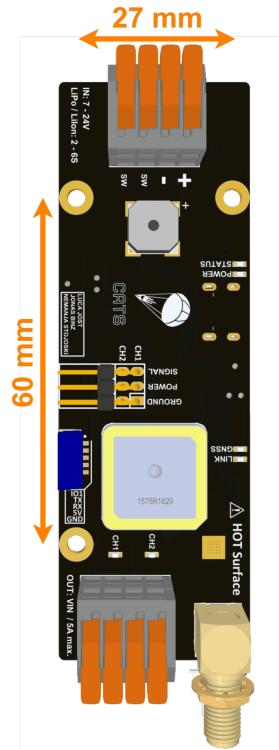


Figure 10: CATS Vega board with the mounting hole and dimensions.

**Warning:** The flight computer should only be powered up once the rocket is upright on the launch pad. The calibration is done only once, as soon as no motion is detected after boot up.

### 3.3.4 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 7 shows an overview of the most important parameters. For further information about the board hardware refer to section 3.1. There, also a picture is shown where each I/O is marked on the board itself (Figure 2)

#### 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.

<sup>5</sup><https://github.com/catsystems/cats-hardware/tree/main/CATS-Vega/3D>



## 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 1V of dropout to the connected circuitry, meaning that the pyro channel will have 1V less than the applied battery voltage. 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 1A 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 below 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!

**Warning:** The pyro channels are short circuit protected with a resettable PTC fuse. If your connected circuit requires more than 1A a solder bridge on the back of the board needs to be closed.

## 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 Specification**

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 7: Overview of the I/Os



### 3.3.5 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 deployment height of the Main Chute deployment
- Time until Main Chute deployment (optional)

With this in hand, the user can configure their flight computer. The timers are optional, as they should only be used as a backup.

1. Connect the flight computer to the user computer.
2. Open the configurator and connect to the board as described in section 3.3.2.
3. In the Configuration tab, set the Liftoff threshold. We recommend using a Liftoff acceleration threshold of  $40 \text{ m/s}^2$ , but make sure that it is around  $20 \text{ m/s}^2$  lower than your maximum expected acceleration.
4. In the Configuration tab, set the main altitude to your desired height. This is the height above ground level 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. Set the link phrase of your CATS Vega by navigating to the configuration tab in the Configurator and setting your link phrase.
7. Make sure that the Testing Mode is disabled.
8. Save the settings.
9. Go to the Events tab.
10. For the apogee event, set your deployment mechanism as described in 3.3.2.
11. For the main deployment event, set your deployment mechanism as described in 3.3.2.
12. Save the settings.
13. (Optional) Go to the Timers tab.
14. (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.
15. (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.
16. Save the settings.
17. Set the same link phrase 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. Mount the flight computer to your rocket.
2. Plug in the switch into the switch port.
3. Plug in the battery into the battery port.
4. Plug in the recovery mechanism for the apogee event.
5. Plug in the recovery mechanism for the main event.
6. Place the rocket on the launch pad.
7. Turn on the flight computer via the switch.
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!

### 3.3.6 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 data onto your desktop.

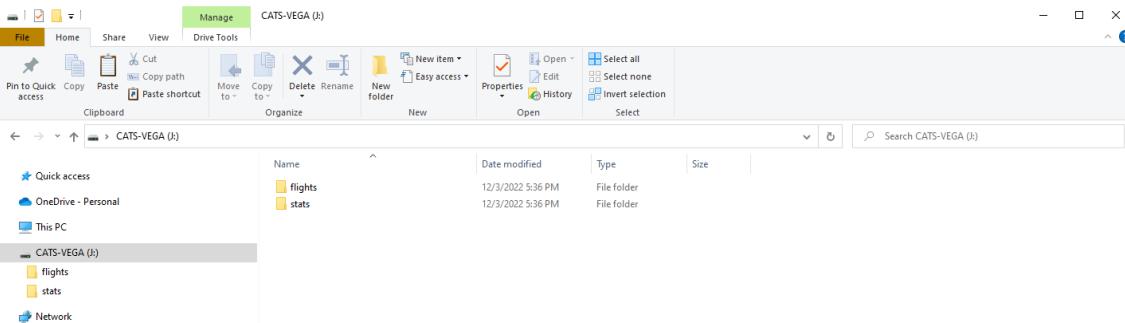


Figure 11: USB drive when the CATS Vega is plugged into the user computer.

### 3.3.7 Visualizing the Flight Data

To visualize your flight data, just open up the Configurator and drag and drop the CATS flight log (.cfg). All important information is then instantly plotted in the Configurator (height, velocity, acceleration, angular velocity (x,y,z), linear accelerations (x,y,z), pressure, state changes and actions). You can then also export the plots as .html files which can conveniently be opened in any browser, keeping their zoom-in functions or as .csv files to do some further processing.

There is also a python plotting tool which you can have a look at as a reference if you want to implement some custom things. This tool is explained briefly in section 7.4.



### 3.3.8 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<sup>6</sup>. To update the software follow those steps:

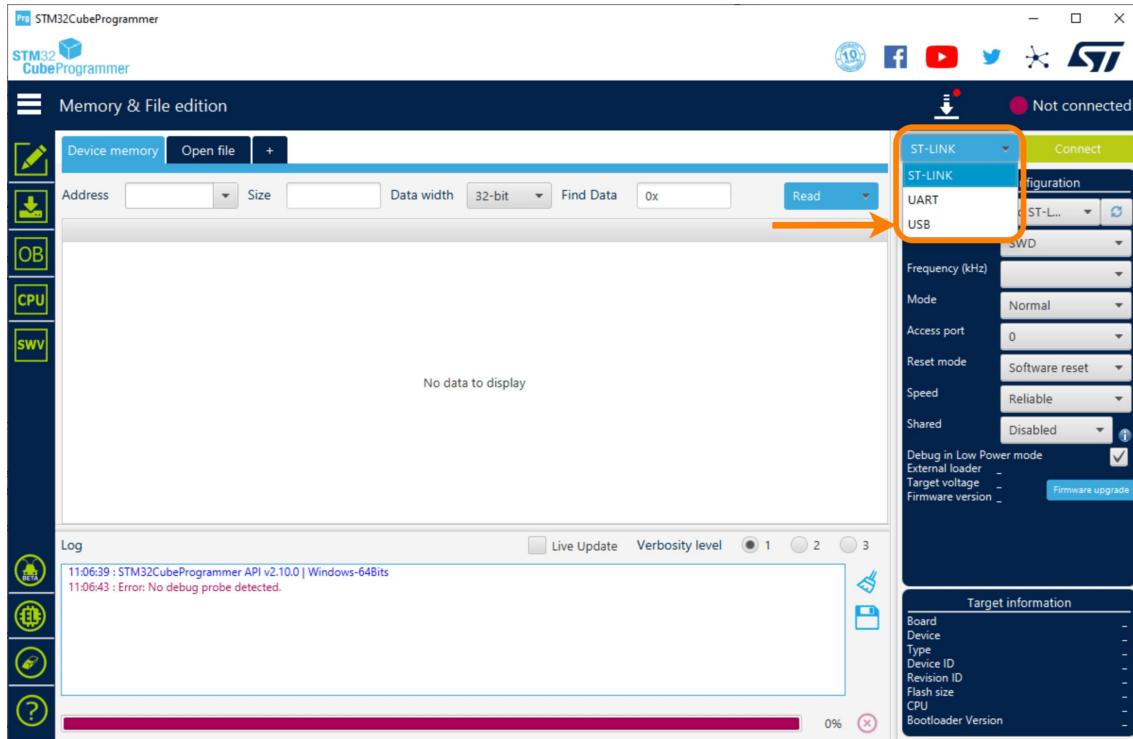
1. Download and install the STM Programmer<sup>7</sup> (STM32CubeProg).
2. Plug in the CATS Vega using a USB-C cable.
3. Start the configurator, connect to the board by selecting the right com port, press the connect button and go to the CLI tab (described in section 3.3.2).
4. In the CLI type `b1` and 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 12a).
8. On the right, select a valid USB port (Figure 12b).
9. Click connect (Figure 12b).
10. Now, on the top right it should be shown that the programmer is connected to the board (Figure 12c).
11. Select Erasing & Programming on the left in the navigation (Figure 12d).
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<sup>8</sup> (Figure 12d).
13. Click start program (Figure 12d).
14. Wait until you see the pop up "File Download Complete" (Figure 12e).
15. Unplug and plug the board back in. Start the configurator and verify that the version number updated.
16. You've successfully updated the software!

---

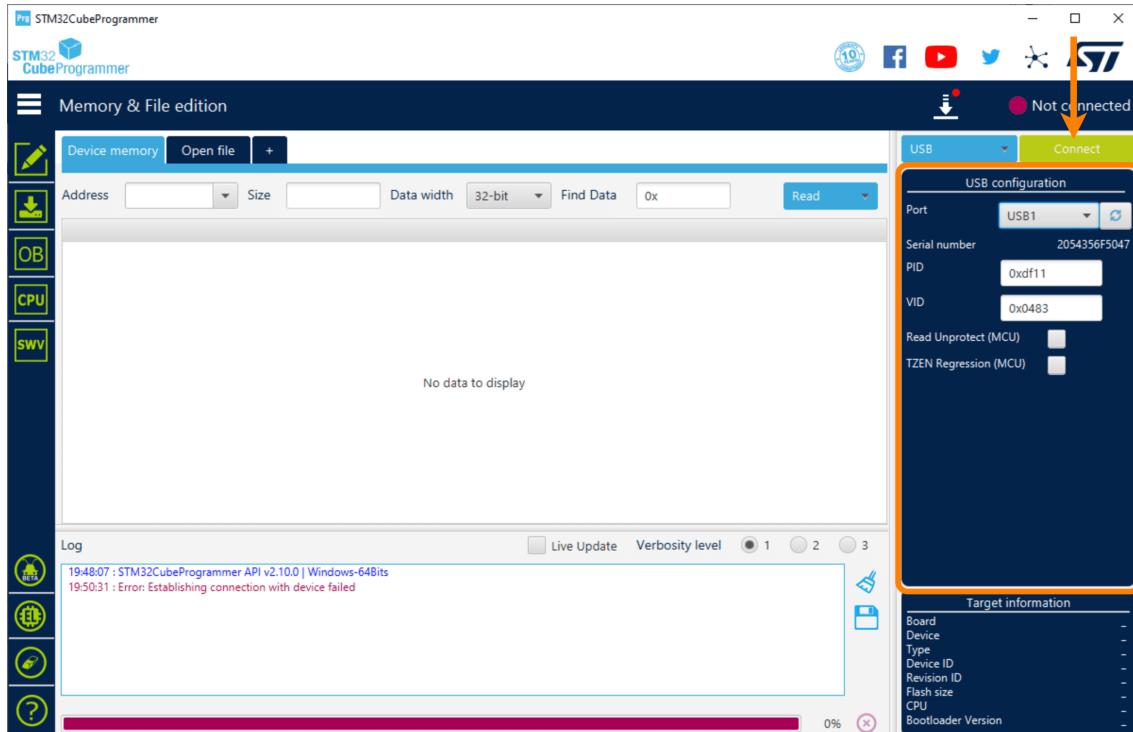
<sup>6</sup><https://discord.gg/r7ErmSNvsy>

<sup>7</sup><https://www.st.com/en/development-tools/stm32cubeprog.html>

<sup>8</sup><https://github.com/catsystems/cats-embedded/releases>

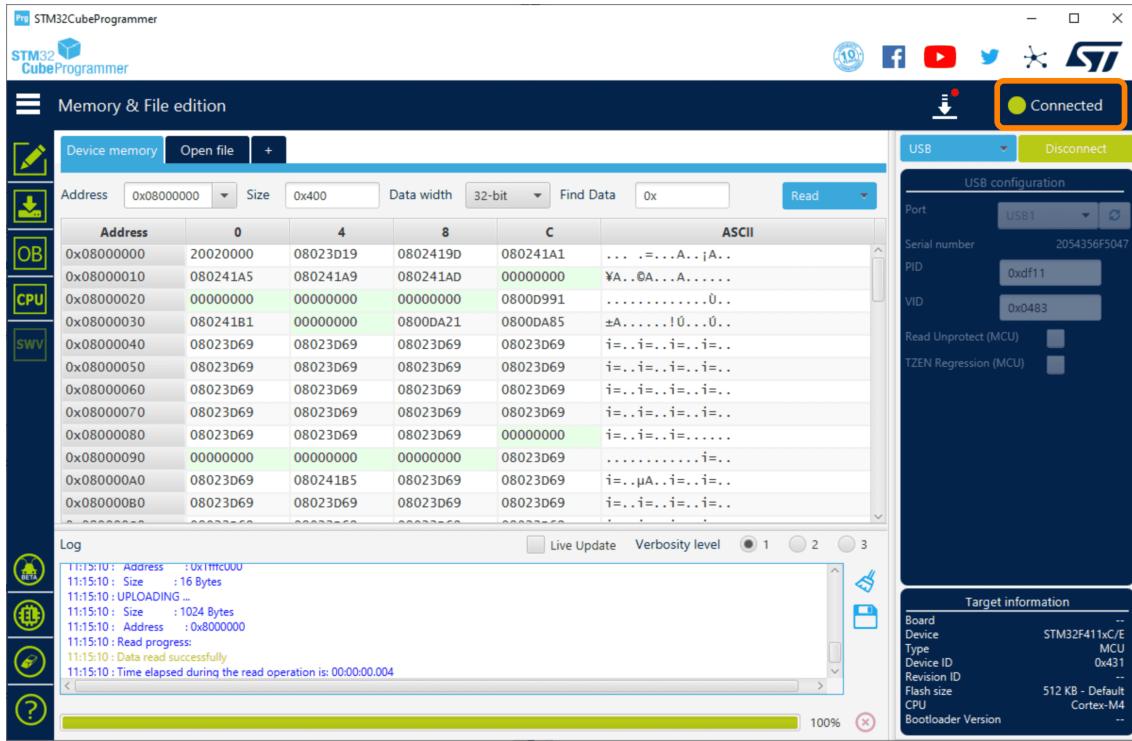


(a) Click on the drop down menu and select USB. This opens the right tab as shown on the picture right.

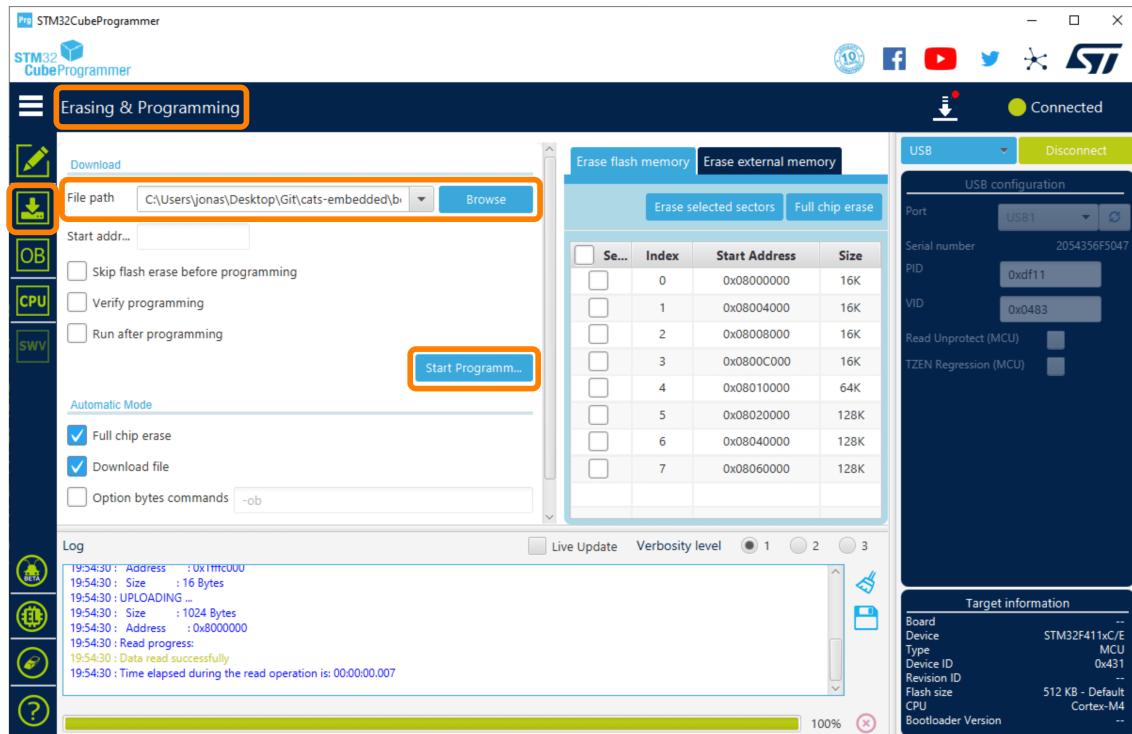


(b) Select the shown USB port in the "port" setting and click connect.

Figure 12: Flashing new software to the board.

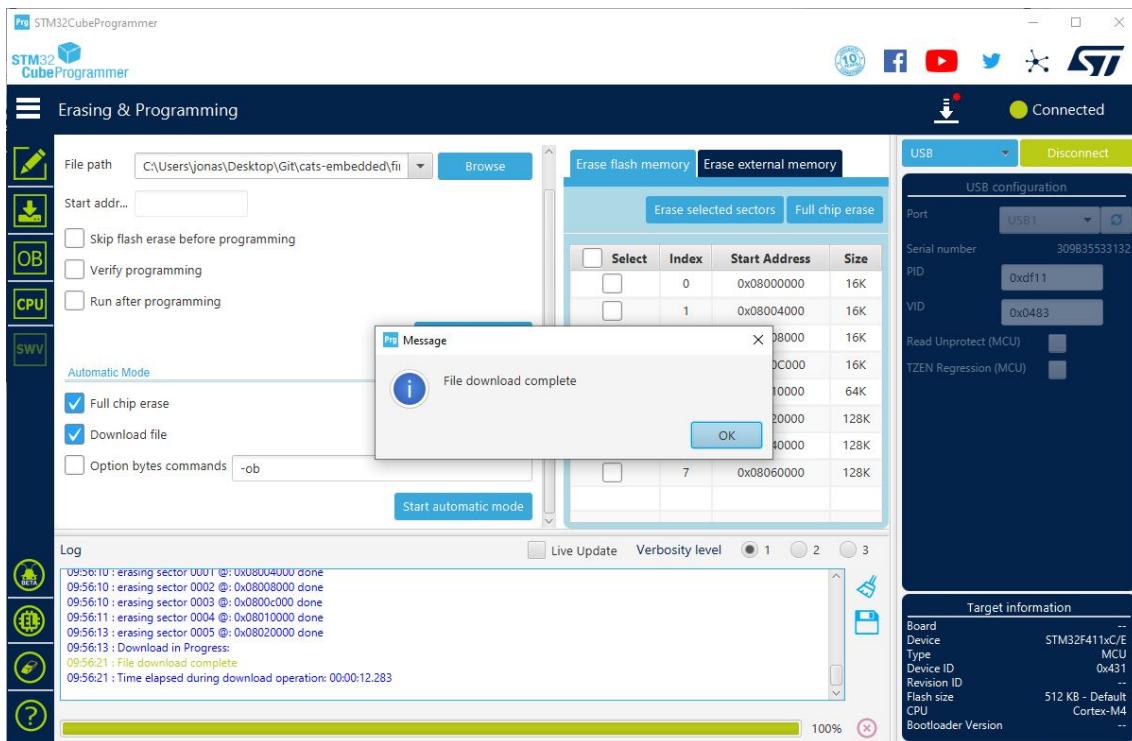


(c) Make sure the connected tag is shown (top right).



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

Figure 12: Flashing new software to the board (cont.).



(e) This is what you should see, if the program was flashed properly.

Figure 12: Flashing new software to the board (cont.).



### 3.4 Beeping Patterns

The CATS Vega flight computer has many beeping patterns that inform the user about the current state of the system or about potential errors. The exhaustive list below should be enough to know if an error is present or which state the flight computer is in.

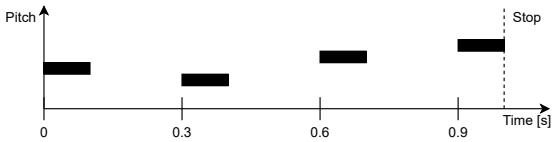
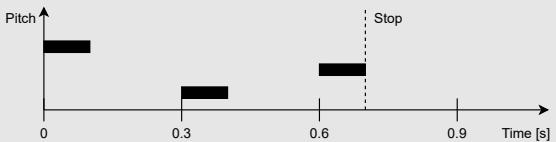
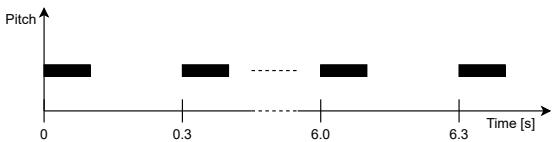
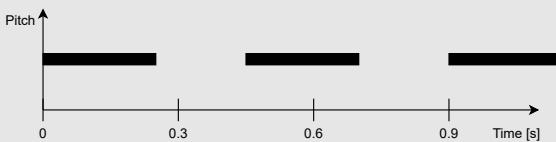
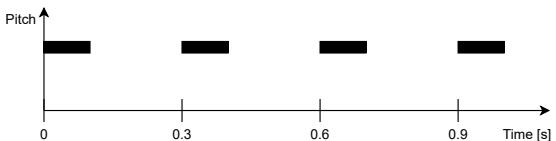
State	Description	Pattern
Bootup	Flight computer has booted up.	
Calibrating → Ready	Flight computer switched from Calibrating state to Ready state.	
Ready	Calibration was successful and flight computer is in Ready state.	
Testing	Flight computer is in testing mode. Only beeped after the computer is rebooted.	
Testing Armed	Flight computer is in armed testing state.	

Table 8: Overview of state beeping patterns.

**Note:** The pitch axis is only showing relative changes and is unitless.



Error	Description	Pattern
Filter Error	A Barometer or accelerometer error is present. If this error persists, <b>do not fly as the functionality of the flight computer is compromised!!!</b>	TBD
Pyro Error	A pyro that is configured to trigger is not detected. If this error persists, <b>do not fly as the recovery mechanism you configured will not work!!!</b>	TBD
Log Full	The flash chip is full. The flight will not be recorded if you decide to still fly.	TBD
Telemetry Hot	The telemetry chip reached a temperature of 60 ° Celsius. Damage to the telemetry could happen.	TBD

Table 9: Overview of error beeping patterns.

**Note:** Errors are currently not beeped out. They are only shown over telemetry on the ground station. Beeping of errors will be added in a future patch.



## 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 13: Ground Station

### 4.1 Hardware

#### 4.1.1 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.



#### 4.1.2 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 10: Ground Station Specifications

## 4.2 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.

#### 4.2.1 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 to go back.

#### Live Data

The Live data screen shows all the data received from the Vega flight computer as well as information about the data link quality.



Figure 14: Live Data of the Ground Station.

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, pyro continuity and errors 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 vs the expected number of packages 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!

## Testing

The testing window is used to do some manual tests. Refer to section 6 for a thorough explanation of the testing mode and how to use it.



## Data

Coming Soon

## Sensors

Coming Soon

## 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.
- **Testing Phrase:** Test phrase which needs to match the test phrase configured on the CATS Vega. This test phrase is crucial to use the testing mode of the system explained further in section 6.
- **Telemetry Mode:** Dual or diversity mode, described in section 4.2.2.

### 4.2.2 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.

### 4.2.3 Charging

The ground station is powered by a Li-Ion 18650 battery. With a fully charged battery the system can run for more than 8 hours. Charging the battery can be done through the USB port. With a charging current of 500 mA it can take up to 6 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.

### 4.2.4 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.

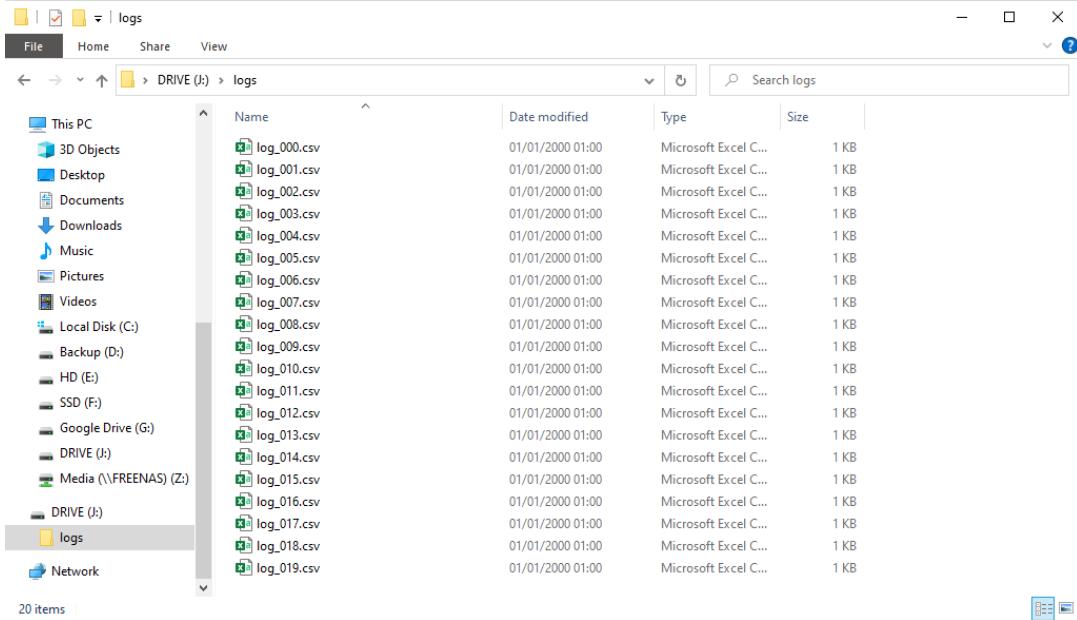


Figure 15: Ground station data when connecting the ground station to the user computer.

#### 4.2.5 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. The old file is then overwritten.

Firmware files have the file ending .UF2 and the newest version of the ground station can always be downloaded from our repository.

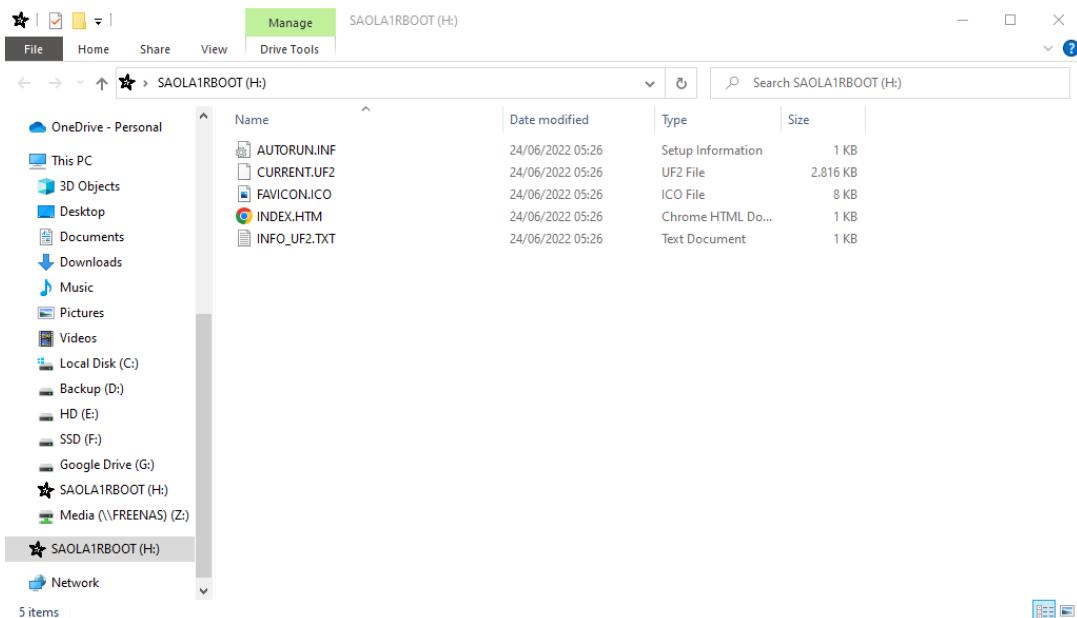


Figure 16: SAOLA1RBOOT mass storage device when successfully changing to the DFU Mode



## 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 and main 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 desired configuration:

1. The drogue is deployed by a pyro, connected to Pyro Channel 1 that needs to be turned on (Fig. 18b).
2. Time to Apogee is 15 seconds and should be used as a backup timer (Fig. 18c).
3. The main is deployed by a pyro, connected to Pyro Channel 2 that needs to be turned on (Fig. 18b). No timer is configured for the main.
4. The liftoff acceleration is simulated to be  $70 \text{ m/s}^2$  (Fig. 18a).

This needs to be configured to the flight computer. Then the accessories need to be added to the CATS Vega. This includes the battery, switch and Pyro Channel 1 and 2. Once those are added to the system, the CATS should look as shown in Fig. 17.

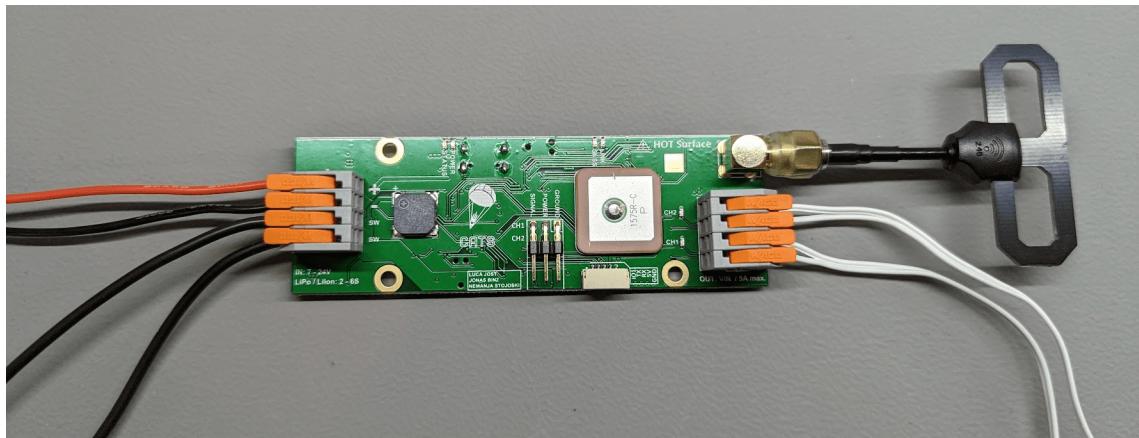
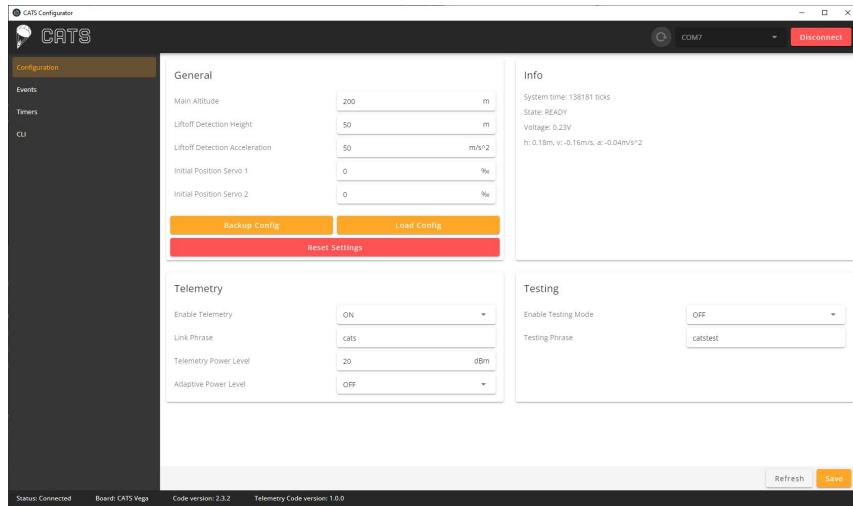


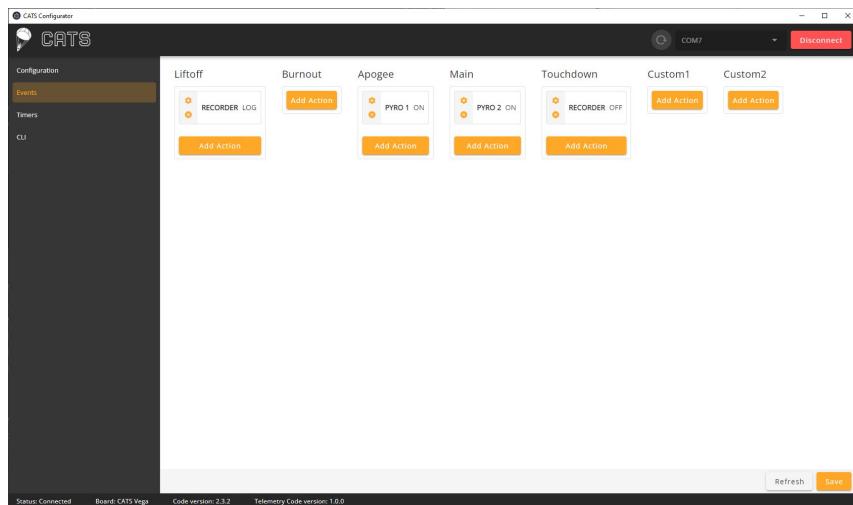
Figure 17: CATS Vega with a battery, switch, Pyro 1 and Pyro 2 connected.



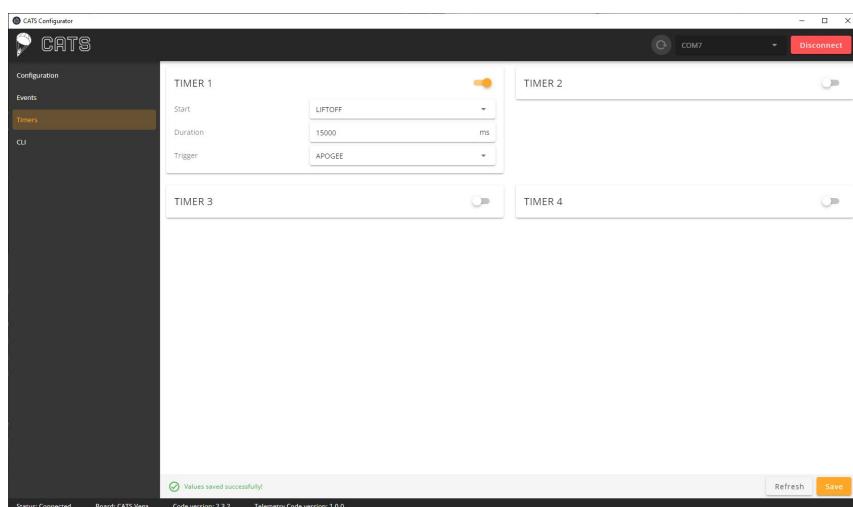
## Control And Telemetry Systems



(a) Configuration Tab (simple example).



(b) Event Tab (simple example).



(c) Timer Tab (simple example).

Figure 18: Configurator screenshots for the simple example configuration.



## 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. 20b).
2. Time to Apogee is 45 seconds and should be used as a backup timer (Fig. 20c).
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. 20a, Fig. 20b).
4. No main parachute backup timer is used (Fig. 20c).
5. The main parachute should be opened at 350 m (Fig. 20a).
6. The liftoff acceleration is simulated to be  $60 \text{ m/s}^2$  (Fig. 20a).
7. A camera is connected to Pyro Channel 2 which should turn on at liftoff and turn off at touch-down (Fig. 20b).
8. The low level I/O should be turned on at burnout (Fig. 20b).

This needs to be configured to the flight computer. Then the 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. 19.

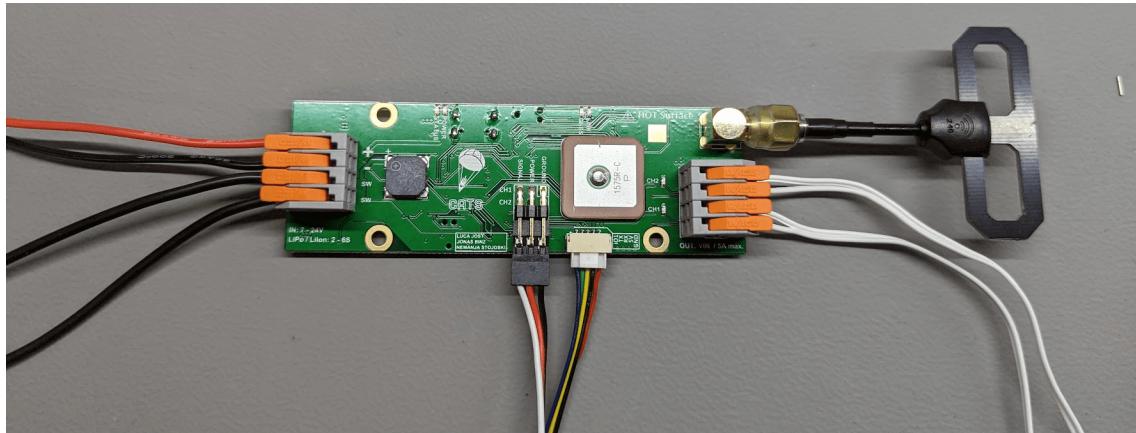
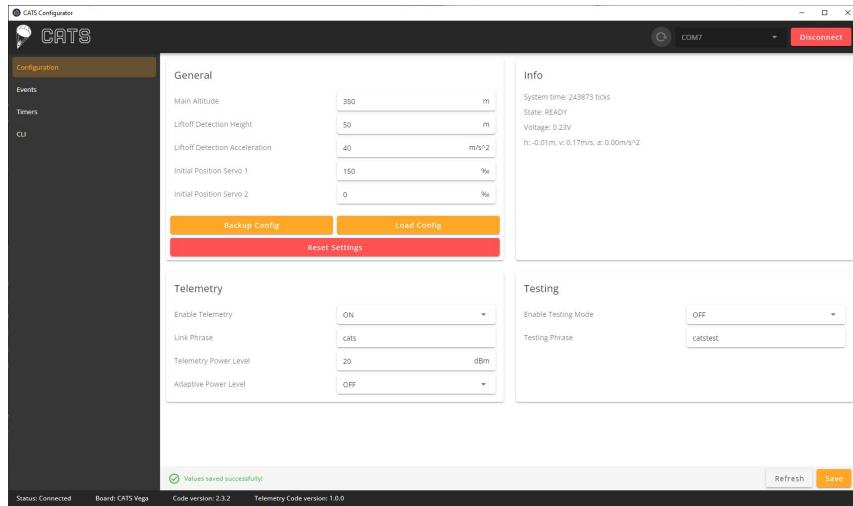


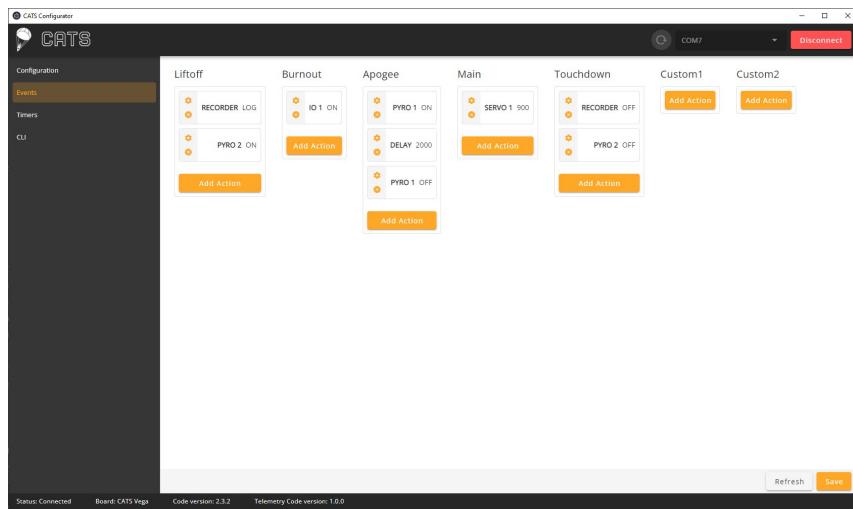
Figure 19: CATS Vega with a battery, switch, solenoid valve, camera and servo connected.



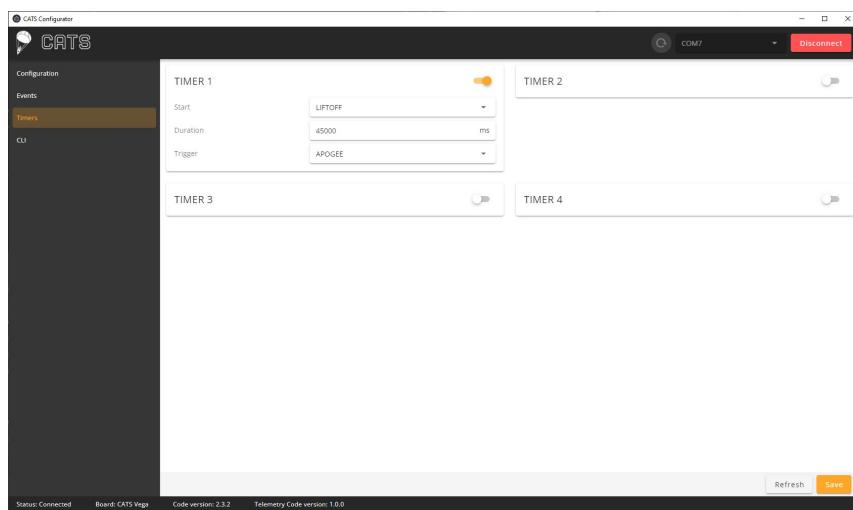
## Control And Telemetry Systems



(a) Configuration Tab (advanced example).



(b) Event Tab (advanced example).



(c) Timer Tab (advanced example).

Figure 20: Configurator screenshots for the advanced example configuration.



## 6 Testing

This section explains how to run tests with the CATS system.

### 6.1 Working Principle

The testing mode of the Vega flight computer works only with a ground station. Once the flight computer is in this mode all processing of information is disabled. This means that if the flight computer is flown while in testing mode, make sure you bring a shovel with you since you will need it to recover your rocket.

Once the flight computer is in testing mode, a beeping sound is emitted, signaling that the computer is indeed in testing. A command from the ground station needs then to be set to arm the flight computer. When the flight computer is armed, events can be executed from the ground station.

**Note:** Unlike during flight, events can be triggered several times without needing to reboot the system.

By design, the testing mode only enables triggering events and not actions so that the user can verify that they added the correct configuration to the flight computer.

### 6.2 Enabling the Testing Mode

To enable the testing mode, follow these steps:

1. Connect your CATS Vega to your computer, start the configurator and connect to the flight computer.
2. Navigate on the home screen on the configurator and select testing mode.
3. Set a testing password.
4. Reboot the flight computer. If the flight computer is not restarted, the testing mode is not activated.
5. The flight computer should beep with the "Testing" Beeping Pattern. See section 3.4 for more information about beeping.
6. Turn on your ground station.
7. Make sure that the link phrase and the testing password match the ones on the flight computer.
8. Navigate to the testing menu on the ground station.



Figure 21: Ground Station Main Menu.

9. Read the disclaimer carefully and arm the testing mode. **Attention:** After this step executing events will trigger the connected mechanisms! Make sure to follow safety guidelines.



Figure 22: Arming the test mode.

10. A screen will pop up that shows that testing mode is being activated. Wait until this screen disappears.



Figure 23: Waiting that testing mode is being started.

11. The flight computer should beep with the "Armed Testing" Beeping Pattern. See section 3.4 for more information about beeping.
12. Select the event which you want to trigger.



Figure 24: Selection of the Event to be triggered.

13. Click on the event and confirm.



Figure 25: Confirming to trigger the desired Event.

**Warning:** Testing mode enables manual triggering of events and corresponding actions associated with them. This feature should only be used for testing purposes and never during flight or other non-controlled activities. CATS GmbH is not responsible for any potential injuries or material damage caused by manual operation of the CATS System.



## 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 26.

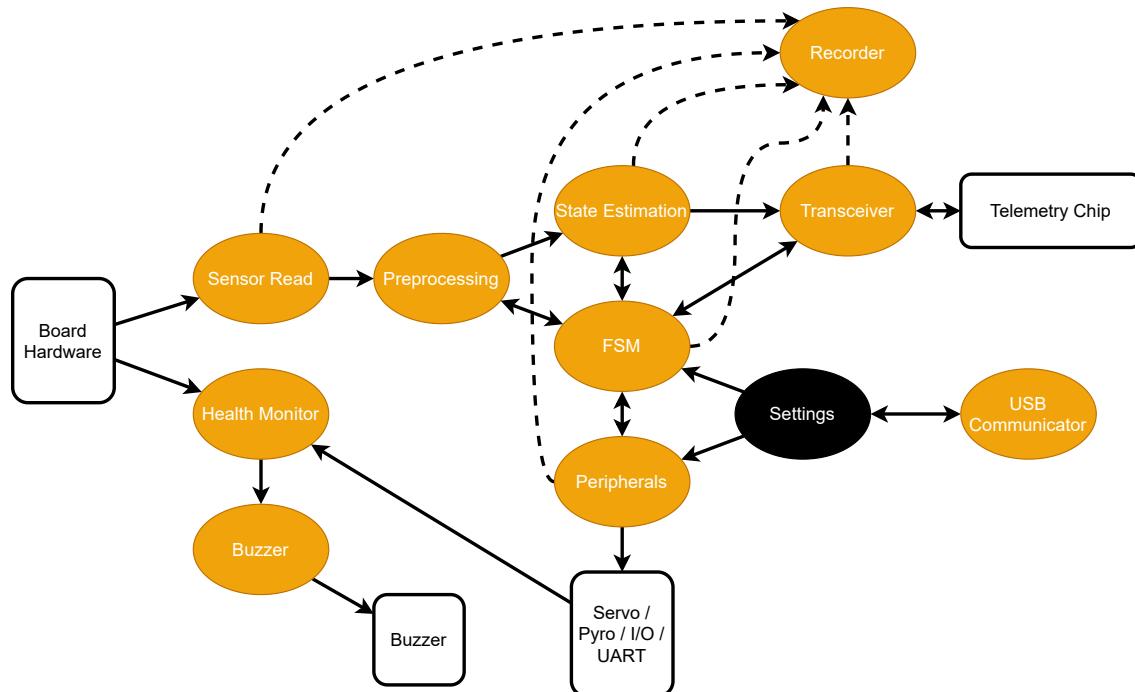


Figure 26: Illustration of the different FreeRTOS tasks interacting with each other and the hardware. The black circle 'Settings' is just a memory region that is being accessed by different tasks.

A brief explanation of every task is given here.

- **Sensor Read** reads out the Barometer and the IMU and provides them to the preprocessing task.
- **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 provides this data to the FSM and state estimation task.
- **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** is a collection of three tasks, handling 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.
- **Buzzer** actuates the buzzer.

## 7.2 Telemetry

The telemetry system is based on 2.4 GHz LoRa and utilizes FHSS (Frequency-Hopping Spread Spectrum) for transmission. A key benefit of FHSS is that it makes transmissions much more resistant to interference and more difficult to intercept. Additionally, it allows 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 hash value 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.

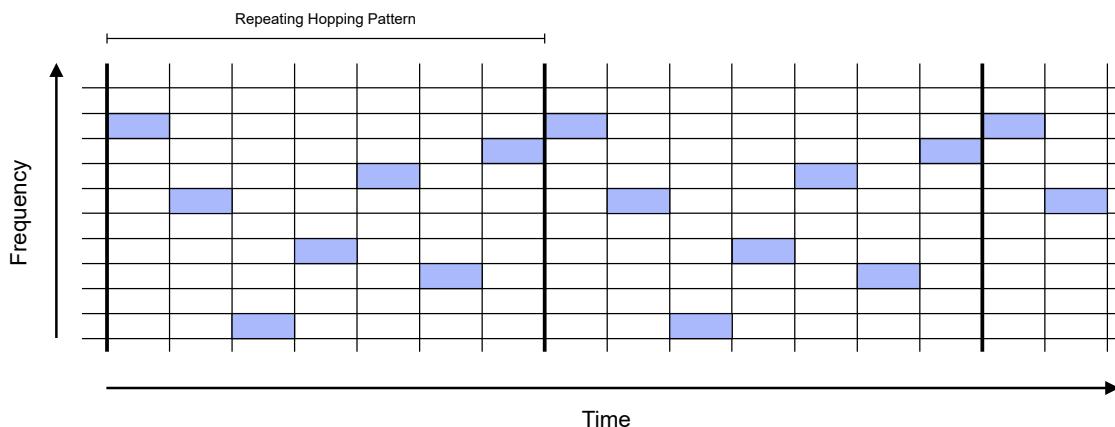


Figure 27: FHSS transmission example

### 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. What is being done here, is that based on the gravity vector, the up direction is computed which is then used throughout the flight.

**Warning:** The flight computer should only be powered up once the rocket is upright on the launch pad. Since we want to prevent the flight computer going in and out of the Ready state, the calibration is done only once, as soon as no motion is detected after boot up.

During Calibrating 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

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

where  $h$  is the height above ground level,  $v$  is the vertical velocity and  $a_o$  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.

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

this is discretized using a first order approximation

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

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  $\omega$  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 idea behind gain scheduling this algorithm is that we want to reduce the trust in 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 back 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.

The conditions for the changes in  $R_{height}$  are given below.

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

Where  $f(v)$  is a function dependent on the current velocity.

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



## 7.4 Using the Python Plotting Tool

If you would like to have some more control over the flight data and access it easily, you can have a look at our old python plotting tool. With this you can do everything that the Configurator can do, but it should be easier for you to do changes.

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:

```
1 git clone https://github.com/catsystems/cats-logs.git
2 cd cats-logs/log_parsing
3 pip install -r requirements.txt
4 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:

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



## 7.5 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 angle brackets <> 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.

<code>bl</code>	Put the board into DFU mode	needed for software updates (refer to section 3.3.8)
<code>cd</code>	Change the current working directory	
<code>config</code>	Print the flight config in a human-readable format	
<code>defaults [--no-outputs]</code>	Reset to default settings	The default configuration triggers pyro channels. If <code>--no-outputs</code> is passed the pyro triggering will not be set.
<code>dump</code>	Print configurable settings in a readable format	
<code>flash_erase</code>	Erase everything on the flash chip; this might take a while	
<code>flash_test</code>	Test writing and reading from the flash	Only for testing purposes, should not be used!
<code>flash_start_write</code>	Start writing to flash	Only for testing purposes, should not be used!
<code>flash_stop_write</code>	Stop writing to flash	Only for testing purposes, should not be used!
<code>flight_dump &lt;flight_id&gt;</code>	Print a specific flight in binary format	
<code>flight_parse &lt;flight_id&gt;</code>	Print a specific flight in a human-readable format	
<code>get [&lt;variable&gt;]</code>	Get a variable value, described in Table 14	
<code>help [&lt;command name&gt;]</code>	Display all commands with a description	
<code>lfs_format</code>	Reformat flash file system	
<code>log_enable</code>	Enable log output on the terminal	



<code>ls [&lt;path&gt;]</code>	List all files in current working directory	
<code>reboot</code>	Reboot the computer	This command does not save the changed settings by itself
<code>rec_info</code>	Get information about flash usage	
<code>rm [&lt;path&gt;]</code>	Remove a file	
<code>save</code>	Save flight configuration	
<code>set [&lt;variable&gt;=&lt;value&gt;]</code>	Set a variable, described in Table 14	
<code>stats &lt;flight_id&gt;</code>	Print flight stats	
<code>status</code>	Show current sensor data, flight phase and other important information	
<code>version</code>	Show the firmware version	

Table 12: Exhaustive List of CLI Commands

### 7.5.1 Get and Set Commands

The variables described here can be read by the get command or written by the set command. Those variables are then saved to the configuration on the flight computer only if save is also commanded.

<code>acc_threshold</code>	Acceleration threshold above which liftoff is detected	
<code>liftoff_detection_agl</code>	Altitude above ground level threshold above which liftoff is detected	
<code>main_altitude</code>	Altitude above ground level under which main deployment is triggered	
<code>timer1_start</code>	Event which triggers timer 1	
<code>timer1_trigger</code>	Event which is triggered once timer 1 elapses	
<code>timer1_duration</code>	Timer 1 duration	
<code>timer2_start</code>	Event which triggers timer 2	
<code>timer2_trigger</code>	Event which is triggered once timer 2 elapses	



<code>timer2.duration</code>	Timer 2 duration	
<code>timer3.start</code>	Event which triggers timer 3	
<code>timer3.trigger</code>	Event which is triggered once timer 3 elapses	
<code>timer3.duration</code>	Timer 3 duration	
<code>timer4.start</code>	Event which triggers timer 4	
<code>timer4.trigger</code>	Event which is triggered once timer 4 elapses	
<code>timer4.duration</code>	Timer 4 duration	
<code>ev_calibrate</code>	Set the actions associated to the calibration event	Do not use!
<code>ev_ready</code>	Set the actions associated to the ready event	Do not use!
<code>ev_burnout</code>	Set the actions associated to the burnout event	Do not use!
<code>ev_apogee</code>	Set the actions associated to the apogee event	Do not use!
<code>ev_main_deployment</code>	Set the actions associated to the main deployment event	Do not use!
<code>ev_touchdown</code>	Set the actions associated to the touchdwon event	Do not use!
<code>ev_custom1</code>	Set the actions associated to the custom 1 event	Do not use!
<code>ev_custom2</code>	Set the actions associated to the custom 2 event	Do not use!
<code>servo1_init_pos</code>	Set the initial value of the servo 1	
<code>servo2_init_pos</code>	Set the initial value of the servo 2	
<code>tele_link_phrase</code>	Set the telemetry link phrase	
<code>tele_test_phrase</code>	Set the testing pass phrase	
<code>tele_power_level</code>	Set the telemetry power level	
<code>tele_adaptive_power</code>	Enable or disable adaptive power for the telemetry power level	
<code>buzzer_volume</code>	Set the buzzer volume	



<code>battery_type</code>	Set the battery type used with the CATS Vega	
<code>rec_elements</code>	Set the desired recorded elements	
<code>rec_speed</code>	Set the desired sampling period for recording	

Table 14: Exhaustive List of parameters used in the get and set Commands



## 8 FAQ

### **My testing mode doesn't work**

To enable testing mode, make sure that you enabled the testing mode of the flight computer first with the configurator, and that a valid testing password is set. Otherwise, the flight computer will start in the default mode and not accept any testing commands sent by the ground station.

### **The Configurator sometimes doesn't accept what I configure**

Try bypassing any sort of USB hub if you see issues with the Configurator. If that doesn't help, write to us on Discord or submit an issue on GitHub.

### **To be extended with user feedback...**



## 9 Disclaimer

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.