User Guide

HAB BOUNDER

Balloon Cut-Down Device

Balloon Ascent Technologies

The sky's not the limit; it's where we start.

Overview

The *HAB Bounder*™ combines four common balloon devices into a single, self-contained unit.

- Pre-programmable cut-down device that can sever a line during the balloon flight. If two are used, then redundant flight termination is assured.
- Flight computer that monitors and records pressure and temperature.
- High-altitude GPS tracker that logs position data in both CSV and KML formats.
- Thermal management device that maintains a constant temperature to ensure proper operation throughout the flight.

Table of Contents

Operating Specifications

Parameter	Min	Typical	Max	Unit
Operating: Temperature	-80	-	60	°C
Operating: Time at cold (heater active)	6	10	18	hours
Operating: Heater Set Point	-60	-10	60	°C
Operating: Battery Type	2x Li/FeS2 AA Batteries			
Trigger: Mission Elapsed Time	1	-	1080	minutes
Trigger: Minimum Atmospheric Pressure	8	-	1200	mBar
Trigger: Pressure Ascent Rate, Threshold	8	-	1200	mBar
Trigger: Pressure Ascent Rate, Rate	-125	-	125	Pa/s
Trigger: GPS Geofence Altitude	1	-	50,000	meters
Trigger: GPS Geofence Longitude	-180	-	180	degrees
Trigger: GPS Geofence Latitude	-90	-	90	degrees
Trigger: GPS Ascent Rate, Threshold	1	-	50,000	meters
Trigger: GPS Min Ascent Rate, Rate	-12.5	-	12.5	m/s
Trigger: External Trigger	-1		5.2	volts
Sensor: Pressure	8	-	1200	mBar
Sensor: Temperature	-60	-	120	°C
Sensor: GPS Altitude	-	-	50,000	meters
Sensor: GPS Altitude Accuracy	±150	±5	-	meters
Sensor: GPS Horizontal/Vertical Velocity	-	-	100	m/s
Mechanical: Mass (no Enclosure or Batteries)	28	30	32	g
Mechanical: Mass (incl. Enclosure & Batteries)	66	68	70	g
Mechanical: Suspended Payload Mass	0	-	5.5	Kg
Mechanical: Size in Foam Enclosure	102mm long, 75mm diameter			
Mechanical: Size without Foam Enclosure	94mm long, 33mm diameter			
Rigging: Dacron Line (break strength)	30	50	80	lb

Checklist

Pre-Flight

1. Program HAB Bounder
    a. Configure HAB Bounder Settings in the config/Flight.cfg file
    b. Insert the micro-SD card
    c. Install Two Batteries
    d. Hold the power-on button for 3+ seconds
    e. Wait for *GPS Wait* tune to start playing
    f. Remove the batteries
    g. Remove the micro-SD card
    h. Verify correct settings are in the PreGPSFix.XXX/Flight.log file
    i. Delete the PreGPSFix.XXX directory after verification
    j. Reinstall the micro-SD card and two batteries

2. Flight Train Rigging
    a. Connect HAB Bounder to Balloon
    b. Attach HAB Bounder to Parachute or Payload
    c. Slide on the radiative battery cover
    d. Check all Flight-Train Connections

Launch

3. Activate *HAB Bounder*
    a. Hold the Power-On Button for 3+ seconds, listen for indicator tones
        i. Beethoven's Für Elise indicates success
        ii. 5 Hz beeping indicates problem—unit inoperable
        iii. Double-beep every 2 seconds indicates GPS trying to acquiring fix
    b. Start Ground Timer, if MET Trigger enabled, when *Toccata and Fugue* plays
    c. Join & Tape Enclosure Closed with Kapton tape

4. Inflate Balloon
5. Launch

Recovery

6. Post-Flight
    a. Remove Batteries

    b. Download and Analyze Data from HAB Bounder

Interface

The *HAB Bounder* uses a micro-SD card for both programming and data logging, a single button to turn it on, and an audio buzzer and LED to indicate status.

Programming Overview

The *HAB Bounder* looks on the micro-SD card for "/config/Flight.cfg" to load the desired settings from. Settings are entered in a json-like format, outlined in the Programming section on pg 7.

Data Logging Overview

The *HAB Bounder* logs data to three different files for each flight. The three files are:

- Flight.CSV – Comma-separated data, recorded at up to 2 Hz
- Flight.KML – Position data in a format that programs like Google Earth use, recorded at 1/10 Hz
- Flight.LOG – Time-stamped system log for post-flight analysis and troubleshooting

Initially the *HAB Bounder* creates a temporary directory when it is powered on, before it acquires GPS time. This directory is named in the following format: `PreGPSFix.XXX`, where `XXX` is an auto-incrementing value to avoid duplication. Any issues with the configuration parameters are immediately recorded to the Flight.log file for fast troubleshooting.

Once GPS time is acquired, this folder is renamed to the following format: `YYYYMMDD.XXX` where `XXX` is an auto-incrementing value to distinguish multiple flights that occur on the same day.

The temporary `PreGPSFix.XXX` folder is removed after the files are successfully moved to the time-stamped folder. Only temporary folders from flights aborted before GPS fix will remain.

Power-On Button

Pressing this button will turn the *HAB Bounder* on. Repeated pressing or constant pressing will have no impact on the *HAB Bounder's* operation. It can not turn off the *HAB Bounder*

To shut down the *HAB Bounder* remove one of its batteries.

The Indicator LED flashes are visible through this button.

Audio Tunes

GPS Wait: the *HAB Bounder* will play a 2-tone beep briefly once every two seconds to indicate that GPS position fix has not yet been obtained. It should not be flown until GPS fix is acquired.

GPS Time Tune: Once the *HAB Bounder* has successfully acquired GPS time and the config file has been read, the intro to Beethoven's Für Elise.

Launch Tune: Once the *HAB Bounder* has successfully acquired GPS fix the intro to Bach's *Toccata and Fugue in D Minor* will play to indicate that the *HAB* Bounder is ready to be flown.

ERROR Tune: If an error is detected during the power-on sequence, a continuous rapid 2-tone sound will be played. The *HAB Bounder* is unusable while this rapid continuous beeping is sounding.

Burn-Wire Activation Tune: A user configurable note can be played multiple times to indicate command activation.

Status LED

The *HAB Bounder* has one green Status LED. Possible blink patterns are:

- While acquiring GPS lock it will double-flash at 1/2 Hz with the GPS Wait tune.
- In normal operations the status LED will flash at 4 Hz (with varying flash durations).

If there is an error, the Status LED will flash rapidly, indicating that the *HAB Bounder* is unusable until the error is corrected.

Error Mode

The *HAB Bounder* will enter Error Mode if any of the following problems are encountered:

- micro-SD Card Problem—No card present, incorrect formatting, or no available space
- Blank SD Card—it will write a default "/config/Flight.cfg" file to the card to edit
- Config file error—check the Flight.LOG file for an indication of what the issue is
- Broken Burn-Wire—One or more of the burn-wires was determined to be broken
- Sensor Problem—An issue was detected with an onboard sensor

Check the Flight.LOG file for an indication of which subsystem failed. Note: if the Flight.LOG does not exist then it is likely that there is an issue with the micro-SD card itself.

Programming

The *HAB Bounder* is programmed via the "/config/Flight.cfg" file. It is formatted in a quasi-JSON style.

If a blank micro-SD card is inserted into the HAB Bounder then: 1) a "/config/" directory will be created, 2) a default sample "Flight.cfg" file will be added to that folder, and finally 3) the HAB Bounder will enter its Error Mode to signal the config file is now ready for customization.

The following is an explanation of the various options in that file.

Configuration Section

This section is used to configure the *HAB* Bounder and its subsystems. The following is an example of the default settings for a v5 *HAB* Bounder.

```
Version: 1
#EnableDebugInfo:
HeaterEnableTemp: -20
SDCardData: Timestamp, System, Position, Ambient, Inputs
SDCardRate: 1.0
CutterControl:
#GPSLowPwr:
#ExtTrigger: 1
```

- Version → A user configurable version number for tracking *Flight.cfg* changes in the *Flight.log* files. This is optional and can be omitted if not needed.
- # → Any line that begins with a '#' character is considered a comment and ignored. The lines are colored green for ease of identification in this document but may not be in your text editor.
- EnableDebugInfo → Enable to record extra information in the "Flight.log" file
  - SD Card information
  - Memory used per Command selected
  - Available free memory
  - Time to GPS fix
- HeaterEnableTemp → This is the temperature at which the heater turns on. The heater ensures that the batteries and sensors remain in an operable temperature range. For proper operation, it is recommended to operate with this set-point at -20 °C when using L91 type batteries.
- SDCardData → The collection of data sets to be logged to the SD card's *Flight.csv* file. Data sets can be omitted if the logged data is not desired. Arguments:
  - **Timestamp** → GPS timestamp and *HAB Bounder's* MET

- **System** → System data such as Battery Voltage, Status flags, and Burn-Wire info
- **Position** → Longitude, Latitude, Altitude, Heading, Ground Speed, Vertical Speed
- **Ambient** → Pressure, Pressure Rate, Humidity
- **Inputs** → External Trigger state

- SDCardRate → Record every x.x seconds in 0.5 second intervals
  - Example: a setting of "0.5" results in 2 Hz *Flight.csv* data
- CutterControl → Enable the CutterControl subsystem for the burn-wires, no arguments
  - Required to use the CutLine *Action* in a *Command*
- GPSLowPwr → Enable low power mode on the uBlox GPS module
  Ideal for long-duration flights where battery life is important
  - Reduces HAB Bounder power consumption by about 20%
  - Limits GPS to 1 Hz solution instead of the normal 2 Hz
  - Disables SBAS, which reduces accuracy and removes an integrity check
  - Can result in lost GPS position fixes in the dataset for a couple of seconds at a time
- ExtTrigger → Enable the External Trigger input
  Required to log or compare *Inputs.ExtTrig* operational value
  - Parameter of "1" triggers on a falling edge (enables a pull-up resistor)
  - Parameter of "2" triggers on a rising edge (enables a pull-down resistor)
  - The "Inputs.ExtTrig" variable changes from 0 to 1 when the desired voltage change is detected on the input pins

Flight.cfg Formatting Notes

- Lines commented out, those that start with a '#', can not have any characters before the '#'. This includes spaces.
- Blank lines between Commands are permitted, as long as there are not any characters on the blank line. This includes spaces.
- The Flight.LOG appends a line number to the beginning of each configuration line to aid with debugging. These are ignored by the HAB Bounder so it is possible to directly copy an old configuration from a Flight.LOG file into a new Flight.cfg file. Example of a valid line:

```
[   1] Version: 1
```

Command Section

The *HAB Bounder* can be configured to have a reasonable number of commands. The actual allowable number of commands is dependent on their complexity. For instance, the *HAB Bounder* supports at least seven 128 waypoint geofence commands or 30 simple ones using a Compare trigger.

The *HAB* Bounder will enter into the *Error Mode* and report the issue in *Flight.log* if too many commands are attempted. For a better idea of memory usage, the limiting factor, add `EnableDebugInfo:` to the configuration section of the Flight.cfg file and review the Flight.log file after powering on the *HAB Bounder* for ~30 seconds.

Each of the commands is acted upon independently by the conditions set and results in the desired actions being executed. The name, *PwrOnTest* in the below example, is user configurable and used to identify the command activation in the *Flight.log* and *Flight.kml* files.

The PwrOnTest command, shown below, will result in the following:

1) Activate once, 30 seconds after GPS fix is acquired
2) Play a B6 note 10 times of duration 1/4 second, separated by 1/4 second delays
3) Pause for 5 seconds
4) Activate burn-wire #0 for 10 seconds at normal power
5) Pause for 5 seconds
6) Play a C6 note 10 times of duration 1/4 second, separated by 1/4 second delays
7) Pause for 5 seconds
8) Activate burn-wire #1 for 10 seconds at normal power

```
Command: PwrOnTest
{
  Conditions:
  {
    Trigger:
    {
       Compare: Timestamp.MET = 30.000000
    }
  }
  Actions:
  {
    PlayTone: 988, 0.25, 0.25, 10
    Pause: 5.00
    CutLine: 0, 0, 1.2, 10.00, false
    Pause: 5.00
    PlayTone: 1047, 0.25, 0.25, 10
    Pause: 5.00
    CutLine: 1, 1, 1.2, 10.00, false
  }
}
```

Conditions

This is a list of conditions that must be met for the command to be acted upon. Activate options include:

Arm → Conditions that must be met before the Trigger conditions are compared. Can permit a Command to be activated multiple times instead of the normal single activation. Not required.

Trigger → Conditions that must be met before the actions can be executed.

Disarm → Conditions that disable the command. Not required.

AndCollection → Group of conditions that are grouped together with a logical AND operator

OrCollection → Group of conditions that are grouped together with a logical OR operator

Example: The following example causes a triangular geofence boundary to be active, from one minute until ten minutes after GPS fix acquisition as long as the pressure is less than 800 hPa for 5 seconds.

```
Conditions:
{
  Arm:
  {
     AndCollection:
     {
        Compare: Pressure.Pressure < 800, 5
        Compare: Timestamp.MET = 60
     }
  }
  Trigger:
  {
        GeoFence:
        {
          Remain: inside
          Altitude: -500, 2233
          Waypoint: -70.636642, 41.585285
          Waypoint: -70.624188, 41.579674
          Waypoint: -70.625283, 41.589149
        }
  }
  Disarm:
  {
    Compare: Timestamp.MET >= 600
  }
}
```

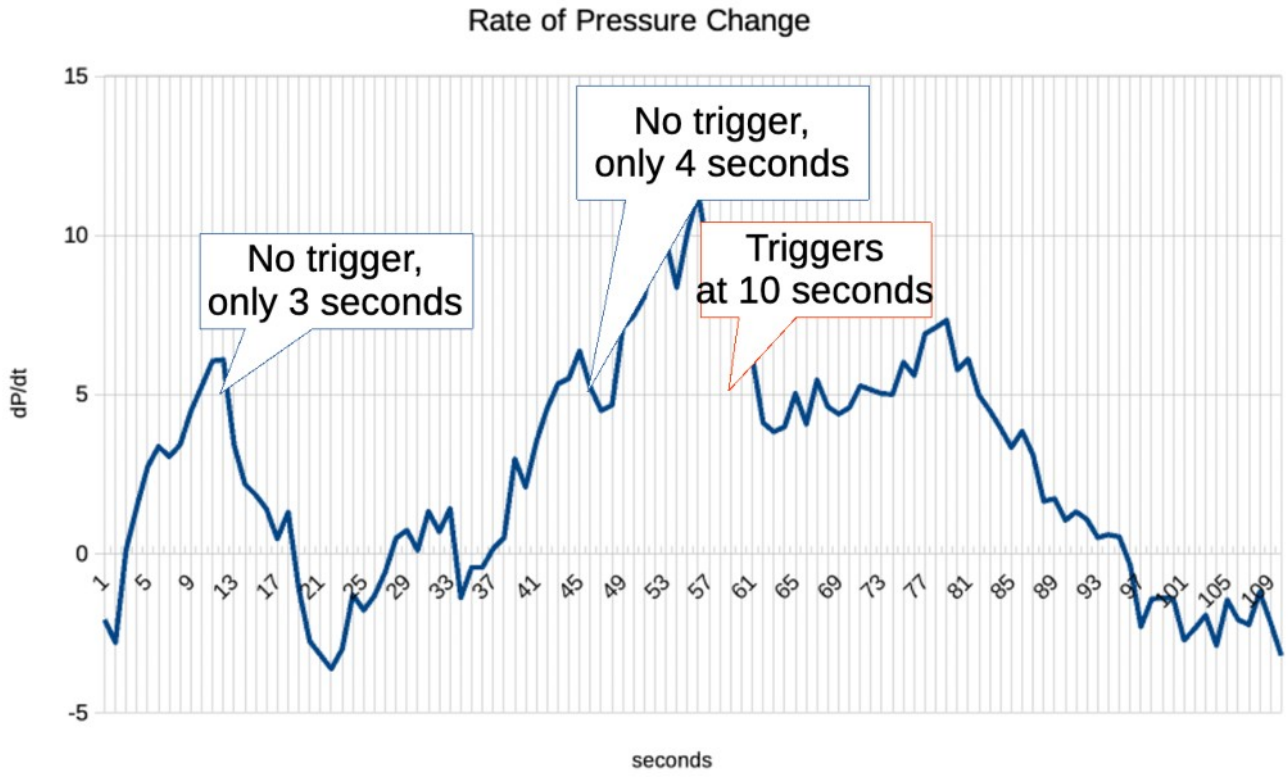
The following is a list of possible logical conditions for Arm, Trigger, and Disarm:

Compare:

This command compares an operational value against a preset value using “<”, “>”, “=”, “<=”, “>=”. Some possible operational values:

- Timestamp.MET
- System.BoardTemperature
- System.BatteryVoltage
- System.SystemStatus
- Pressure.Pressure
- Pressure.PressureRate
- Pressure.Humidity
- Position.Longitude
- Position.Latitude
- Position.Altitude
- Position.Heading
- Position.GroundSpeed
- Position.VerticalSpeed
- Inputs.ExtTrig

A debounce parameter can be added to the Compare command. This parameter will require the comparison to be valid for a specified number of seconds, tested at 2 Hz. Example: when Pressure Rate goes above 5 Pa/s for at least 10 seconds: `Compare: Pressure.PressureRate > 5, 10`



GeoFence:

This condition compares the *HAB Bounder's* current position against a polygon geofence.

```
GeoFence: 7
{
  Remain: inside
  Altitude: -500, 2233
  Waypoint: -70.636642, 41.585285
  Waypoint: -70.624188, 41.579674
  Waypoint: -70.625283, 41.589149
  Waypoint: -70.615079, 41.586535
  Waypoint: -70.613184, 41.591485
  Waypoint: -70.629393, 41.596487
  Waypoint: -70.638358, 41.595474
}
```

Arguments include:

- GeoFence: **7** → Max number of waypoints expected for this Geofence.

  If no number is supplied then the *HAB Bounder* will accept up to 12 waypoints. If this is set below the actual number of waypoints provided in the Flight.cfg file then an error will be generated in the Flight.log.

  Note: there is no need to manually close the polygon. If the first waypoint is not the last then the *HAB Bounder* will automatically include it to close the polygon. This repeated waypoint does not need to be included in the expected number of waypoints as long as it isn't listed as a user-entered waypoint.

- **Remain** → "inside" triggers if the *HAB Bounder* goes outside of the geofence region, while "outside" will trigger if the *HAB Bounder* enters into the geofence region.
  Defaults to "inside" if this argument is missing.

- **Altitude** → Minimum and maximum altitudes permitted over the entire geofence.
  Defaults to -1,000 to 50,000 meters if this argument is missing.

- **Waypoint** → Each GeoFence needs at least three and not more than 128 waypoints. The *HAB Bounder* will repeat the first waypoint to close the polygon automatically if needed.

Actions

This section holds the commands that the *HAB Bounder* will execute when the *Conditions* are satisfied. The commands are executed sequentially.

Options include:

- CutLine → Activates a cutter output. Requires the `CutterControl:` subsystem to be enabled in the Configuration Section. The five arguments:
  - Cutter number, 0 (inner burn-wire) or 1 (outer burn-wire)
  - Cutter Condition: only activate if the selected cutter number passes its continuity test, otherwise skips activation and moves on to the next action
  - Power level, 0.1 to 5.0 watts, 1.2 W is normal
  - Duration, 0.1 to 32,000 seconds
  - Remove from continuity test after activation, true or false
- PlayTone → Causes an audible tone to be played. The *Command's Action* execution will pause while the tone is played. The arguments:
  - Frequency, in Hz
  - Duration on in seconds
  - Duration off in seconds
  - Number of iterations through the on/off cycle
- Pause → Causes the Action execution to be paused
  - Argument: Seconds to delay

Geofence Example

Below is a complete example that both has a test command briefly after power-on and then ensures that the *HAB Bounder* stays within a geofence region.

- It runs a brief (5 second) test 30 seconds after GPS fix is acquired.
- It sets a geofence that triggers if the *HAB Bounder* is ever taken outside of a small area. If this geofence command triggers then the burn-wires are activated in a standard sequence.
- It disables the geofence once the *HAB Bounder* starts rapidly falling for three seconds after balloon burst. This ensures the burn-wires do not activate post-burst even if it drifts out of the geofence area while descending under parachute.

`[example on following page]`

```
HeaterEnableTemp: -20
SDCardData: Timestamp, System, Position, Ambient
SDCardRate: 1.0
CutterControl:
Command: PwrOnTest
{
  Conditions:
  {
    Trigger:
    {
       Compare: Timestamp.MET   = 30.000000
    }
  }
  Actions:
  {
    PlayTone: 784, 0.5, 0.5, 10
  }
}
Command: GeoTest1
{
  Conditions:
  {
    Trigger:
    {
        GeoFence: 8
        {
        Remain: inside
        Altitude: -500, 2233
        Waypoint: -70.636642, 41.585285
        Waypoint: -70.624188, 41.579674
        Waypoint: -70.625283, 41.589149
        Waypoint: -70.615079, 41.586535
        Waypoint: -70.613184, 41.591485
        Waypoint: -70.629393, 41.596487
        Waypoint: -70.638358, 41.595474
        Waypoint: -70.636642, 41.585285
        }
    }
    Disarm:
    {
      Compare: Position.AscentRate  < -15, 3
    }
  }
  Actions:
  {
    PlayTone: 988, 0.25, 0.25, 10
    Pause: 5.00
    CutLine: 0, 0, 1.2, 10.0, false
    Pause: 1.00
    PlayTone: 1047, 0.25, 0.25, 10
    Pause: 5.00
    CutLine: 1, 1, 1.2, 10.0, false
  }
}
```

ExtTrig Example

Below is a complete example that relies on an external trigger input to activate the burn-wires.

- The external trigger input is configured to activate on a falling edge. The *HAB Bounder* connects a 10 Kohm pull-up resistor on the external trigger line to 3.3 VDC. The external trigger source then only needs to sink less than 1 mA of current to activate it.
- Once the external trigger is held low for at least one second the *ExtRadio* command is armed. Repeated activations are possible with this configuration since the `Inputs.ExtTrig` is used as the Arming condition instead of as just a Trigger condition.
- To guard against a spurious external trigger command being sent early in the flight and not being detected until the pressure falls below 800 hPa an `AndCollection` is used. This ensures the ExtTrig input is still being held low when the *ExtRadio* command is activated.
- To test the external trigger source on the ground, the Trigger's
  `Compare: Pressure.Pressure < 800`
  can be replaced with a different test that doesn't require a flight or pressure chamber such as:
  `Compare: Timestamp.MET > 30`

```
HeaterEnableTemp: -20
SDCardData: Timestamp, System, Position, Ambient, Inputs
CutterControl:
ExtTrigger: 1
Command: ExtRadio
{
  Conditions:
  {
    Arm:
    {
        Compare: Inputs.ExtTrig = 1, 1
    }
    Trigger:
    {
       AndCollection:
       {
          Compare: Inputs.ExtTrig = 1
          Compare: Pressure.Pressure < 800
       }
    }
  }
  Actions:
  {
    PlayTone: 988, 0.25, 0.25, 10
    Pause: 1.00
    CutLine: 0, 0, 1.2, 10.0, false
    Pause: 5.00
    CutLine: 1, 1, 1.2, 10.0, false
  }
}
```

Log Files

The *Flight.csv* file logs data at up to 2 Hz, and is written to the SD card every 10 seconds. The *Flight.kml* file logs and writes data at 1/10 Hz.

The Flight.CSV Data File

Data

- DateTime — GPS date in YYYY/MM/DD HH:MM:SS format
- MET — Internal second timer, starts counting from 0 after initial GPS acquisition
- Status — A bit-mapped data flag
  - Bit-0 — Reserved
  - Bit-1 — Reserved
  - Bit-2 — GPS Power Save Mode active
- LoCFS — INOP on HAB Bounder
- CutV — Voltage across burn-wires
- CutC — Cutter Continuity
  - Bit-0 — Burn-Wire #0, 0=good, 1=fail
  - Bit-1 — Burn-Wire #1, 0=good, 1=fail
- Measurements
  - BatV — The battery voltage
  - IntT — Internal temperature in degrees Celsius
  - NumSat — Number of GPS Satellites currently used by the GPS module
  - Lon /Lat — GPS longitude in degree decimal
  - Altitude — GPS altitude in meters above Ellipsoid
  - GroundSpeed — GPS ground speed in m/s
  - VerticalSpeed — GPS vertical rate in m/s
  - Pressure — Ambient pressure, in hecta-Pascal (hPa)
  - PressureRate — Pressure rate of change, in Pascal per second (Pa/s)
  - Humidity — Ambient humidity, in percent (%)
  - ExtTrig — External trigger state, 0 or 1

Note: unless otherwise noted, each data value is updated for every entry.
Note: the GPS position information has a max data rate of 1 Hz.

The Flight.LOG File

This file provides a log of the system operations, commands sent and received, as well as any issues encountered and errors generated.

Device Section

Basic information to help identify the *HAB Bounder* for troubleshooting

Sensors Section

Displays the power-on status for each subsystem

Configuration Section

Provides a copy of the *Flight.cfg* file that was used for this flight

Flight Section

Displays when *Commands* were executed as well as any error messages that arose during the flight

The Flight.KML Data File

This file allows easy plotting of the geofences, events, and flight path in applications such as Google Earth.

Geofence boundaries are shown by a red (remain outside) or green (remain inside) box. Because KML does not support floating boxes (the *HAB Bounder's* geofence minimum altitude can be any altitude less than 50 km), the min/max altitudes are included as a note that shows up if the box is clicked.

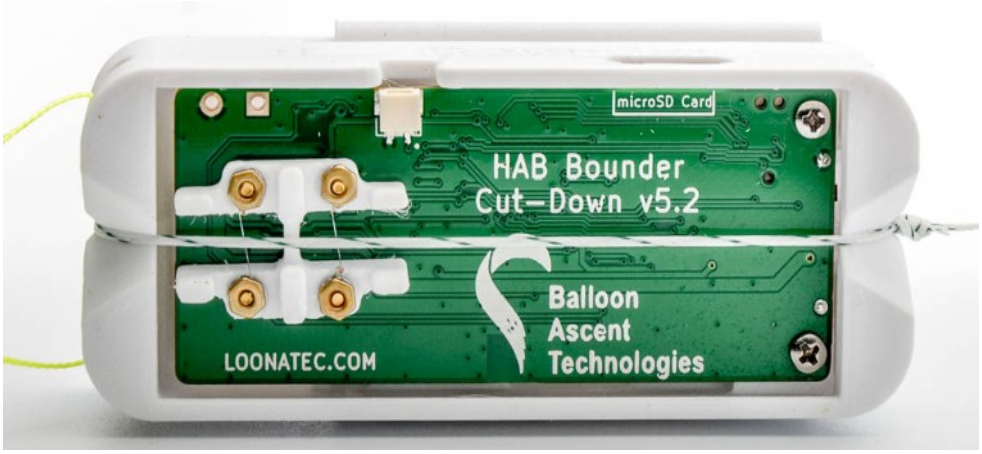
Each Command event position is included as a KML <Point> to show where it occurred.

Flight Tracks are time-stamped in a KML <gx:Track> to permit reviewing the flight progression through time. Tracks are marked in blue.

Note: Command event points and Flight Tracks may be shown 'below' the ground due to GPS or map inaccuracies. Google Earth allows each point and track to be raised independently for viewing by the "Get Info" dialog -> "Altitude" tab.

Rigging

This section covers how to connect the *HAB Bounder* into your flight-train to ensure proper termination or cut-down when programmed.



A few notes of caution:

- The burn-wires on the *HAB Bounder* are thin and fragile. Excessive direct pressure can cause them to break. Never press down on them directly with your finger.
- Trim any excess line after the knots. Excess line can tangle and not allow the Balloon Line to separate from the Device Line. This is especially important for lighter payloads.

Line to be Cut

Typically the *HAB Bounder* is rigged to cut the flight-train line attaching the balloon. Cutting the line immediately under the balloon is recommended to ensure it can't tangle during ascent with any other lines, preventing it from releasing after being cut. It also helps ensure that the *HAB Bounder* doesn't interfere with the parachute during descent, which is usually tied below the *HAB Bounder* by a 2+ meter length of line.

The line to be cut, Balloon Line (pictured as black below), is attached to the *HAB Bounder* with a San Diego Jam Knot and passes directly over the burn-wires. The knot allows the loop around the *HAB Bounder* to tighten, but not loosen, thus ensuring that there is always good contact between the line and the burn-wires.

The retention line (pictured as yellow above) is tied through the provided hole to ensure the *HAB Bounder* isn't lost after burn-wire activation.

Insulation

Foam Enclosure

The *HAB Bounder* ships with a two-part, light-weight extruded polypropylene foam enclosure. This enclosure both insulates the *HAB Bounder* during flight and protects it during landing.

Figure 1: HAB Bounder sitting in the bottom half of the foam enclosure

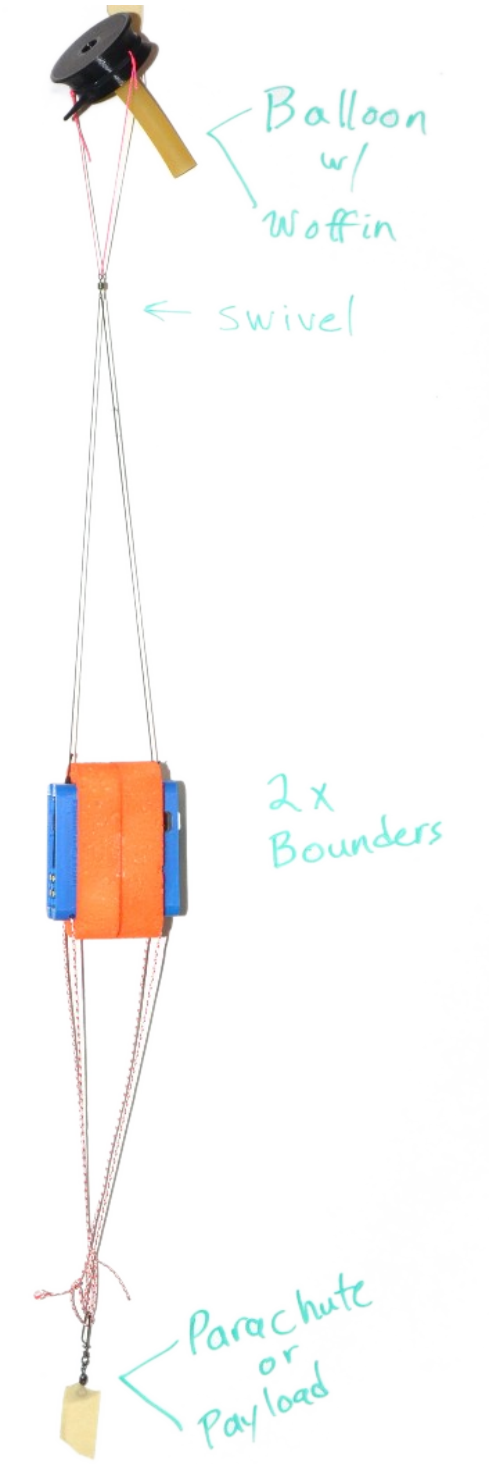
The foam enclosure has a small notch at one end. This is to ensure the Balloon Line can pass freely through the enclosure after it is cut.

Radiative Shield

The *HAB Bounder* ships with a radiative insulation device. This cover slides over the batteries and helps ensure that they stay warm at high altitudes. Please ensure the batteries are not touching the radiative shield after installation, as shown in the photo below.

Redundant Termination

It is possible to use two *HAB Bounders*, configured as pictured to the right, to provide redundant termination for a latex balloon flight.



Explanation from top to bottom:

- The latex balloon (not shown) is attached to the Woffin. Once the line below is cut by a *HAB Bounder*, the balloon will rapidly ascend until it bursts.
- A large swivel is used so that Balloon Line can be shared between two *HAB Bounders*. Once one HAB Bounder™ cuts, the free end will pass through the swivel, releasing the balloon.
- Two *HAB Bounders* are attached together with a wrap of tape around their foam enclosures. Only one half of each HAB Bounder's™ two-part foam enclosure is pictured for clarity.
- The Retention Line is looped through both *HAB Bounders* and attached to either the parachute or payload—depending on how the flight-train is laid out. If attached to the parachute, additional line between the parachute and the *HAB Bounders*' Retention Line is recommended so that during descent the *HAB Bounders* do not interfere with the parachute's deployment.

Note: Shadows from the camera flash during photographing make lines appear doubled.

Troubleshooting

Device Will Not Turn On

The device should turn on within 3 seconds of the power-on button being pressed. If it doesn't, check the following:

- The power-on button not fully being depressed to actuate the switch and held for 3 seconds
- Check the battery orientation
- Install new batteries. The Energizer L91 batteries should not be stored in hot conditions for extended periods of time (months). In that condition they self-discharge more quickly.

Error Indicated

A 5-flash-per-second by the Indicator LED and rapid beeping indicates a malfunction. The *HAB Bounder* will not operate correctly when this pattern is flashing. Do not fly. Possible causes include:

Immediate error after power-on

- Issue with the micro-SD Card: Check that a card is inserted and has a good connection (re-seat card)

Error after GPS time is acquired

- Error in the /config/Flight.cfg file: Check the Flight.log file for insight on line that is causing issues
- Broken Burn-Wire: Check the Flight.log file's ___ section to see if a wire is broken
- Sensor Malfunction: Check the Flight.log file's Sensor section for which sensor failed, contact Balloon Ascent Technologies for repair options

Updating the Firmware

- Place the UPDATE.bin file in the root directory of the micro-SD card
- Insert the card
- Press and hold the power-on button (5-7 seconds) until the tone starts playing
- Let the *HAB Bounder* acquire GPS fix, then wait for an additional two minutes
- Power off, remove the micro-SD card
- Look at the latest *Flight.log* file to ensure that the Device section's firmware line indicates the date associated with the UPDATE.bin file you just used

Consumables

Batteries

The *HAB Bounder* has been designed to use primary lithium batteries in the 'AA' size. The Lithium/Iron Disulfide (Li/FeS2) chemistry has proven to work well, even at the low temperatures experienced on balloon flights. Possible suppliers of these batteries include:

- Energizer Ultimate Lithium L91 AA Battery
- Varta Ultra Lithium 6106 (FR14505) AA Battery
    - Note: the 6106 is rated at 18% less capacity than the L91 battery (2.9Ah vs 3.5Ah)

While the *HAB Bounder* will operate on standard alkaline batteries, it is strongly recommend against using them due to their poor performance at cold.

Memory Card

Any SD/SDHD micro-SD card should work with the *HAB Bounder*. We recommend using the SD Memory Card Formatter (https://www.sdcard.org/downloads/formatter/) from the SD Association to format new cards. Using the OS formatting utility (Mac OSX or Windows) can cause problems.

Note: some cards can operate slower than expected. If this happens they will generate a "`WARNING - RunLoop Timing Overflow`" message in the Flight.LOG file. This is not a problem as long as it only happens occasionally, once a minute or less frequently.

Rigging Line

Below are some of the rigging line options that we have used and liked. The Green-Spot line is our preferred line due to it being widely available and how well the burn-wires cut it. The Nano cord holds knots well and comes in a wide variety of colors. The Twisted Nylon line is super stretchy and can help dampen jerks and bounces on long flight-trains during the flight.

Green-Spot Dacron Fishing Line. Some manufacturers include: Izorline or Cortland

Atwood Rope Nano Cord: https://atwoodrope.com/collections/nano-cord

Twisted Nylon Mason Line: Home Depot, Amazon, Erin, etc.

Safety Precautions and Recommendations

Regulatory

Always adhere to the regulations governing high-altitude ballooning applicable in the country of operation. In the United States, high-altitude balloon flights are governed by “FAA Part 101”.

Touch Hazard

The HAB Bounder’s two burn-wires are fragile and produce high temperatures when activated. Never press-down on or touch the burn-wires.

Falling Payload Hazard

Always clear the launch area prior to launch. While unlikely, balloon or line failure after release may result in a falling payload impacting the ground before the parachute fully deploys.