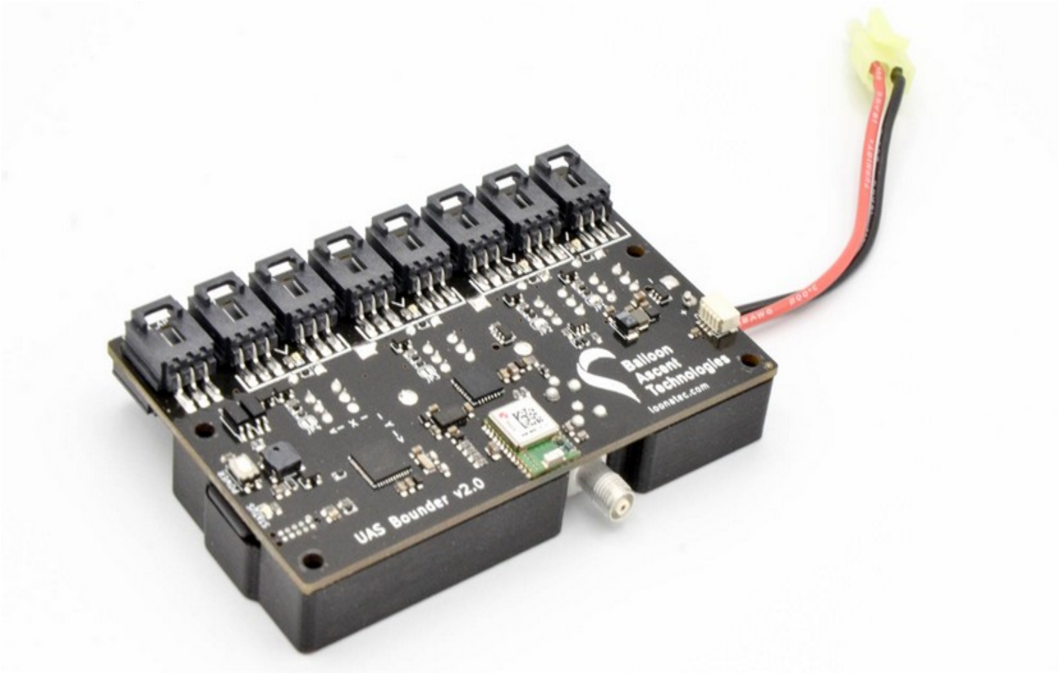
User Guide

UAS BOUNDER

Geofence Enforcer for Drones

Balloon
Ascent
Technologies

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



Overview

The UAS Bounder™ ensures that your unmanned aircraft system (UAS) stays out of trouble and within its authorized flight area. It is completely independent of, and transparent to the autopilot system to ensure reliability. It can be used to eliminate the need for visual line-of-sight operation on authorized test ranges and permit rapid autopilot development by avoiding costly reviews after each change. It can easily be adapted to any UAS that uses standard RC servos and electronic speed controllers (ESC). The UAS Bounder detects range violations and safely terminates the flight when a boundary is breached.

The UAS Bounder is a secondary geofence controller for unmanned aircraft systems. Its features include:

- Multiple activation parameters to choose from:

  GPS geofences, acceleration, magnetometer, and timer limits
- Seven-channel Servo Routing Switch (SRS)
- Eight configurable servo PWM outputs
- Programmed via a micro-SD card prior to launch
- Independent omni-directional GPS receiver
- Separate power that also provides backup power for servo motors after activation
- Ability to log GPS position, acceleration, and other flight characteristics
- Opto-isolated uni-directional autopilot interface for status and position reports
- No RF spectrum approval necessary

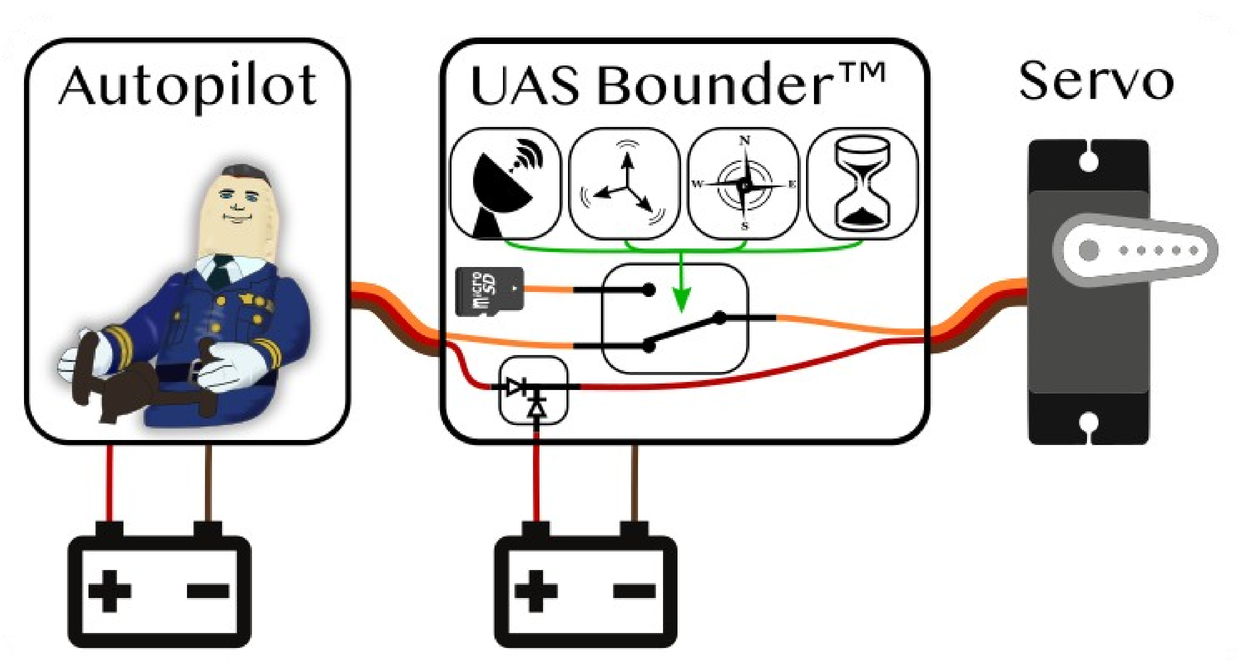


Table of Contents

1 Operating Specifications

Parameter	Min	Typical	Max	Unit
Operating: Temperature	-20	-	60	°C
Operating: Power, No Relays Active(V = 7.5)	-	25	65[1]	mA
Operating: Power, All Relays Active (V = 7.5)	-	105	145[1]	mA
Operating: Power, Current to Motors (total)	0	-	7	A
Operating: Battery Voltage	3.6	-	17	VDC
Trigger: Mission Elapsed Time	1	-	-	seconds
Trigger: GPS Geofence Altitude	-500	-	50,000	meters
Trigger: GPS Geofence Longitude	-180	-	180	degrees
Trigger: GPS Geofence Latitude	-90	-	90	degrees
Trigger: GPS Ascent Rate, Threshold	-500	-	50,000	meters
Trigger: GPS Min Ascent Rate, Rate	-12.5	-	12.5	m/s
Servo: PWM Pulse Width	10	800-2200	9,999	micro-sec
Servo: PWM Pulse Rate	25	50-250	1,500	Hz
Sensor: GPS Altitude	-	-	50,000	meters
Sensor: Accelerometer	-2	-	+2	g’s
Sensor: Magnetometer	-4	-	+4	gauss
Mechanical: Mass (no Enclosure or Batteries)	-	45	-	g
Mechanical: Size without mating connectors	96mm long, 58mm wide, 19 mm heigh			

1 Max current values are due to extra power being consumed to play a tone.

2 Interface

The *UAS Bounder* uses a micro-SD card for both programming and data logging, two opto-isolated serial UART outputs, a single button to turn it on, and an audio buzzer and four indicator LEDs to indicate status.

Programming Overview

The *UAS 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 12.

Data Logging Overview

The *UAS Bounder* creates a new directory each time it is powered on and acquires GPS time. This directory is named in the following format: YYYYMMDD.XXX where XXX is an auto-incrementing value to distinguish multiple flights that occur on the same day.

Inside of this directory are the following files:

- Flight.CSV – Contains comma-separated data recorded at 1 Hz
- Flight.KML – Contains position data in a format that programs like Google Earth use
- Flight.LOG – Contains a time-stamped system log for post-flight analysis

Power-On Button

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

To shutdown the *UAS Bounder* remove its battery(s).

Note: the *UAS Bounder's* logic power remains separate from the autopilot power even when a relay is active. Removing power from the autopilot connections is not necessary to shutdown the *UAS Bounder*.

Note: the Power-on button controls a slow-decaying RC circuit. This ensures that the *UAS Bounder* turns it back on after momentary power glitches.

Audio Tunes

<u>GPS Wait</u>: the *UAS 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.

<u>Launch Tune</u>: Once the *UAS Bounder* has successfully acquired GPS fix and the config file has been read, the intro to Beethoven’s Für Elise will play to indicate that the UAS 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 *UAS Bounder* is unusable while this rapid continuous beeping is sounding.

Servo Relay Activation Tune: A user configurable note can be played multiple times to indicate command activation.

Status LED

The *UAS 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 *UAS Bounder* is unusable until the error is corrected.

Relay LEDs

The *UAS Bounder* has four orange Relay LEDs positioned between the AP_n connectors. They turn on when the associated relay is active, taking autopilot control away from the indicated servo channel(s).

Opto-Isolated Outputs

The *UAS Bounder* has two opto-isolated serial connections that can provide one-way data to the vehicle's autopilot. These two outputs permit the vehicle's autopilot the ability to monitor the *UAS Bounder's* parameters and state, as well as use its GPS as secondary position source.

Hardware

Interface to these two UART signals is via the COMM_AP connector. All three pins are electrically isolated by two opto-isolators to ensure the vehicle's autopilot can not effect the *UAS Bounder's* operation.

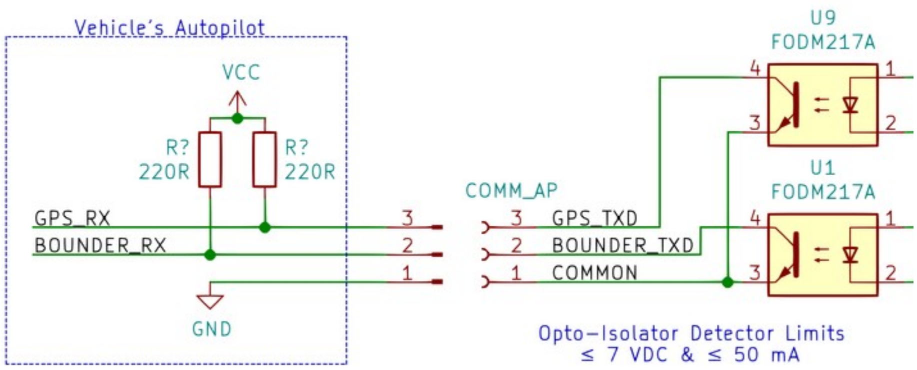


Figure 1: Opto-Isolated Outputs

An external pull-up resistor is required between the receiving microprocessor’s VCC rail and the incoming UART TXD signal. Typically a 220 Ohm resistor can be used on systems where VCC ≈ 3.3 VDC. Higher resistor values can be used for slower baud rates or on systems where VCC is higher..

Real-Time Operating Information

The *UAS Bounder’s* real-time operating information can be transmitted to the vehicle’s autopilot via BOUNDER_TXD. This data is controlled by the “Comm” commands in the Flight.cfg file. See the Configuration Section on pg 12 for details.

GPS Data

The UAS Bounder’s GPS module can directly transmit position data via the GPS_TXD pin. This is controlled by the “GPSComm” command in the Flight.cfg file. See the Configuration Section on pg 12 for details.

Error Mode

The UAS 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
- Sensor Problem—An issue was detected with an onboard sensor
- GPS Problem—An issue was detected with the onboard GPS 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.

3 Servos

Servo Connectors

All of the servo connectors, both incoming from the autopilot (AP_X) as well as outgoing to the servo motors (SERVO_X), utilize 3-pin Molex SL latching connectors. These connectors accept:

- Molex SL latching housings (Molex Part #50-57-9403)
- Standard RC Servo connectors (non-latching)

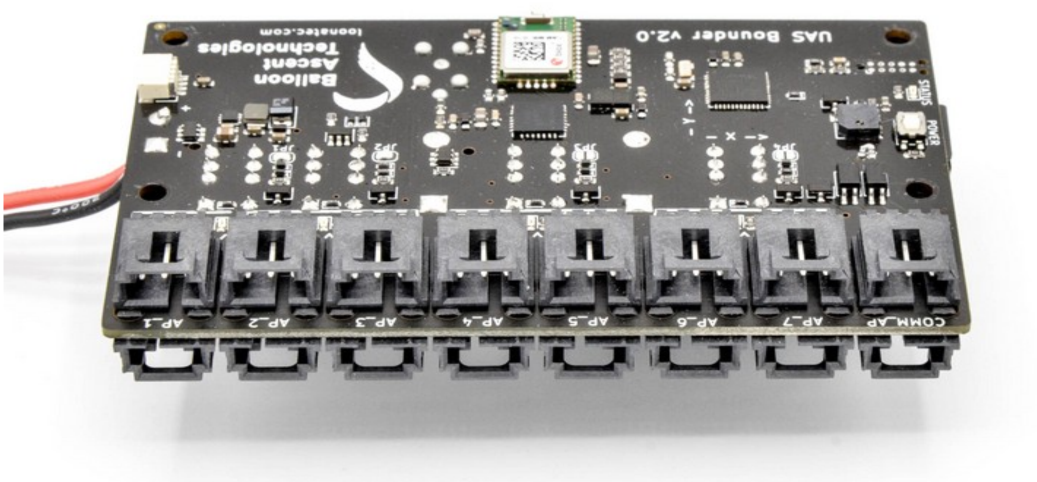


Figure 2: Servo Connectors

Servo Power

Under pre-termination circumstances, power to the servo motors can only come from the servo input connectors from the autopilot. Once a relay is activated, power can come from either any servo input connector or from the *UAS Bounder's* own battery. The power source is determined by a high-power diode-OR circuit to ensure reliable flight termination is achieved (see Figure 4: Servo Channel Diagram)

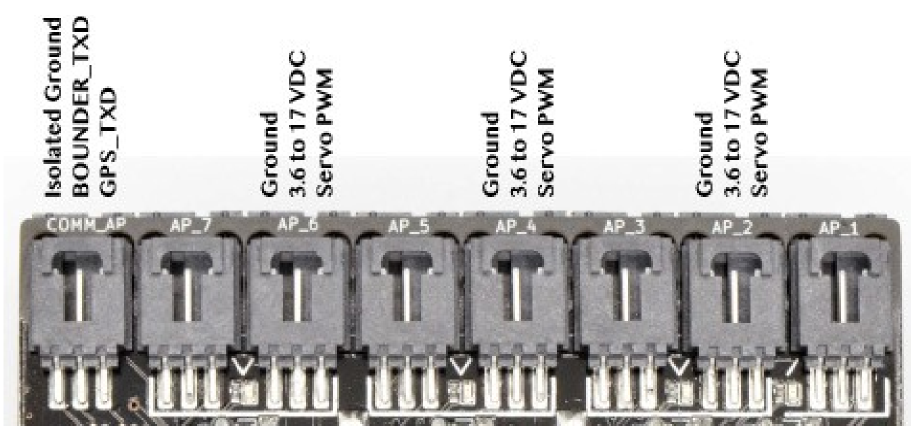


Figure 3: Servo Connector Pinout

Servo Channels

The *UAS Bounder* can switch up to seven independent, autopilot controlled servo channels. There is an eighth servo output (SERVO_8) that the *UAS Bounder* generates itself that does not have a corresponding autopilot input.

Below is a diagram depicting relays 1 & 2, and servo channels 1-3. Relays 3 & 4, and their associated channels 4-7, are not shown because their configuration is identical to relay 2.

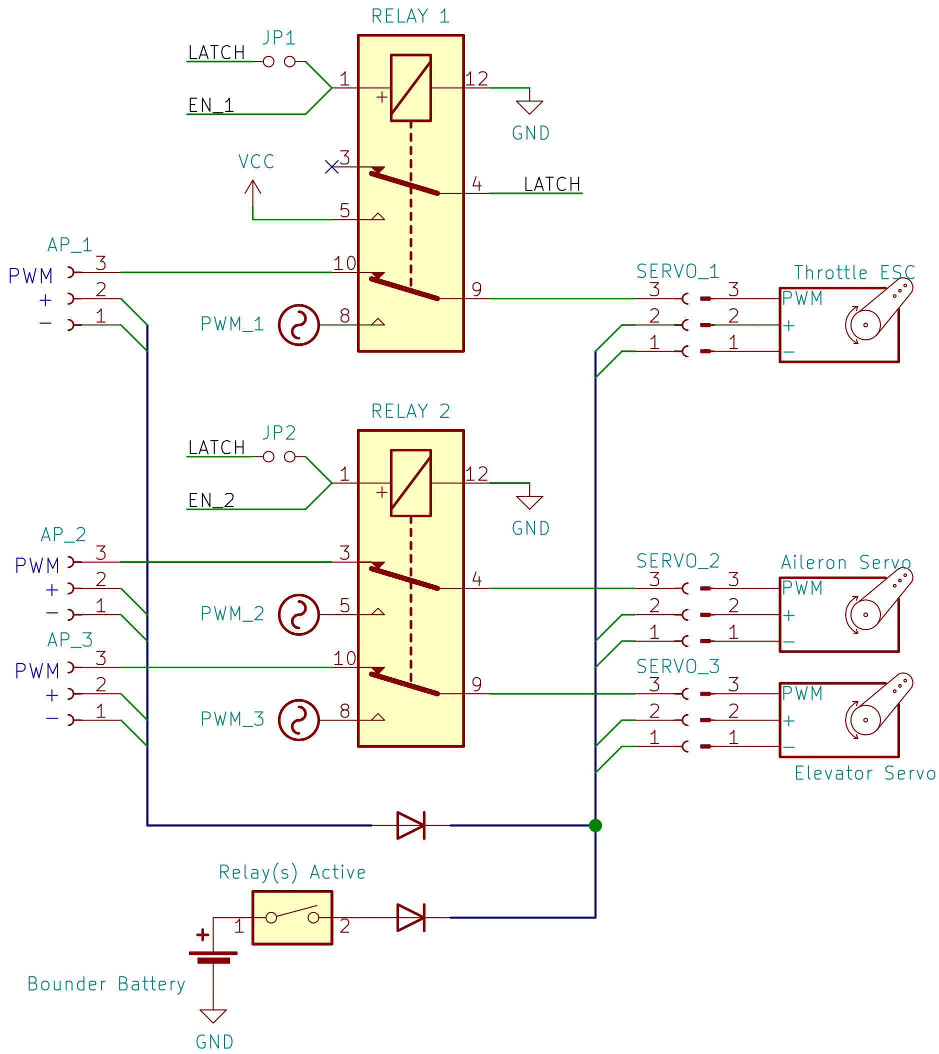


Figure 4: Servo Channel Diagram

Mechanical Relays

There are four mechanical relays on the *UAS Bounder* that switch servo channels 1-7 between being controlled by autopilot input (AP_x) and preconfigured, internally generated PWM signals (PWM_x).

Each relay switches two servo channels, except relay 1 which uses one of its outputs to generate the latching logic (LATCH).

Relay	Output A	Output B
1	Chanel 1	Latching logic
2	Chanel 2	Chanel 3
3	Chanel 4	Chanel 5
4	Chanel 6	Chanel 7

Relay Activation — SetServoRelays

Each relay can be activated with the *SetServoRelay* command. The command parameters are *true*/*false*, listed in sequential order.

The example line of code below activates relay 2. This would take control of two servo channels (SERVO_2 & SERVO_3) away from autopilot control (AP_2 & AP_3) and feed them internally generated PWM signals (PWM_2 & PWM_3). It leaves the other relays inactive, so their associated servo channels will remain under autopilot control.

```
SetServoRelays: false, true, false, false, false
```

Relay Activation — Latching Logic

The *UAS Bounder* is equipped with an option to permit the electrical latching of the relays. If this latching feature is enabled, then it ensures that once a relay is activated it remains activated until power is removed, regardless of other *Flight.cfg* commands.

Relays 1-4 can be individually configured to use the latching logic signal generated by relay 1. The latching logic signal is controlled with the four solder jumpers JP 1-4.

Relay	Jumper	Note
1	JP1	Normally latching, JP1 is shorted by a cuttable trace
2	JP2	Normally non-latching, JP2 is open
3	JP3	Normally non-latching, JP3 is open
4	JP4	Normally non-latching, JP4 is open

Example: on the above diagram, if JP1 & JP2 were both shorted then two modes of operation are possible for channels 2 & 3.

1. If relay 2 is activated via the *SetServoRelays* command then the autopilot only looses control of SERVO_2 & SERVO_3 while that command is active or until superseded by another command.
2. If relay 1 is activated then the LATCH signal is energized, ensuring that both relay 1 & 2 will activate and remain on until the *UAS Bounder*'s power is removed. Note, relay 2 does not have to be activated by the *SetServoRelays* command to activate it this case.

SERVO_8 & Relay 5

There is an additional eighth servo channel output (SERVO_8) with no associated autopilot input. Both ground and power pins on this connector are switched by power MOSFETs (max 3 Amps).

This output can be used to control a servo not connected to the autopilot or provide switched power to a wide variety of devices such as locator beacons, parachutes, release mechanisms, etc.

Servo Status in Data File

Each Servo Channel has a column in the *Flight.csv* data file (S_xx). Each entry indicates one of the following states:

- "_" → Relay controlling this channel is not active, autopilot in control of this servo channel
- "OFF" → Relay controlling this channel is active, servo output is continuously low
- "ON" → Relay controlling this channel is active, servo output is continuously high
- XXX → Relay controlling this channel is active, servo output is indicated by the displayed number (PWM delay in microseconds)

4 Programming

The *UAS Bounder* is programmed via the "Config/Flight.cfg" file. The following is an explanation of the various options in that file.

Configuration Section

This section is used to configure the UAS Bounder and its subsystems. The following is an example of the default settings for a v2 UAS Bounder. See the Data Messages section on pg 26 for additional details.

```
Version: 1
SDCardData: Timestamp, System, Position, AccMag
SDCardRate: 1
CommData: Timestamp, System, Position, AccMag, GPS
CommBinary: false
CommHeader: true
CommDevice: Serial, 1, 57600, 1
ServoMux: 25000000, 250
GPSComm: 57600, 7
```

- Version → A user configurable version number for tracking *Flight.cfg* changes in the *Flight.log* files.
- SDCardData → The collection of data containers to be logged to the SD card's *Flight.csv* file. Arguments:
  - **Timestamp** → GPS timestamp and *UAS Bounder's* MET
  - **System** → System data such as Battery Voltage, Status flags, and ServoMux info
  - **Position** → Longitude, Latitude, Altitude, Heading, Ground Speed, Vertical Speed
  - **AccMag** → Acceleration in XYZ, Aggregate acceleration, Magnetometer in XYZ, and RPM
  - **GPS** → Either the NMEA GGA (ASCII) or the UBX-NAV-PVT message (binary)
- SDCardRate → Record every x sensor cycles
  A value of 1 results in 2 Hz data, while a value of 2 results in 1 Hz data.
- CommData → The collection of data containers to be transmitted on the opto-isolated serial link, same options as with DataLoggerSDCard
- CommBinary → Whether to transmit binary or ASCII data over the opto-isolated serial connection
  - Arguments: true or false

- CommHeader → Whether to transmit an ASCII header at power-on to aid in decoding the subsequent binary or ASCII data [true/false]
- CommDevice → The opto-isolated serial connection (BOUNDER_TXD) settings. Arguments:
  - **Serial** → Name of the port type [not user selectable]
  - **1** → Number of the port [not user selectable]
  - **57600** → Baud rate [9600 to 115200 are supported]
  - **10** → Output interval in seconds, user selectable (1 to 600 are common)
- ServoMux → Servo PWM generation settings. Arguments:
  - **25000000** → Frequency of the PWM generator in Hz. This can be tuned for very precise PWM generation if needed
    Typical: 25000000, normal range: 23000000 to 28000000
  - **250** → Servo PWM output frequency in Hz. 250 Hz is typical for a digital servo, 50 Hz is typical for an analog servo. Note: all servo channels will use the same output frequency rate so combining digital and analog servos may not work well.
- GPSComm → The opto-isolated serial connection (GPS_TXD) settings. Arguments:
  - **57600** → Baud rate [9600 to 115200 are supported]
  - **7** → Decimal value of the UBX-NAV message desired (7 is for UBX-NAV-PVT)
    See section 32.17 of the u-blox M8 Receiver Description for other valid message options.

Command Section

The *UAS Bounder* can be configured to have a reasonable number of commands. The actual allowable number of commands is dependent on their complexity—RAM usage. The UAS Bounder will enter into the *Error Mode* and report the issue in *Flight.log* if too many commands are attempted.

Each of these commands is acted upon independently by the conditions set and results in the various 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) Turn on Relays 2 & 4, leaving Relays 1, 3, & 5 deactivated
3) Configure all of the servo channels to output 1000 microsecond PWM signals
4) Play a G5 note twice of duration one second, separated by one second delays
5) Configure all of the servo channels to output 2000 microsecond PWM signals
6) Play a A5 note twice of duration one second, separated by one second delays
7) Configure all of the servo channels to output 1500 microsecond PWM signals
8) Play a B5 note twice of duration one second, separated by one second delays
9) Pause for 6 seconds
10) Deactivate all of the Relays (test runs for a total of 18 seconds)

```
Command: PwrOnTest
{
  Conditions:
  {
    Trigger:
    {
       Compare: Timestamp.MET   = 30.000000
    }
  }
  Actions:
  {
    SetServoRelays: false, true, false, true, false
    SetServoPWMs:  1000, 1000, 1000, 1000, 1000, 1000
    PlayTone: 784, 1.00, 1.00, 2
    SetServoPWMs:  ON, 2000, 2000, 2000, 2000, 1000
    PlayTone: 880, 1.00, 1.00, 2
    SetServoPWMs:  OFF, 1500, 1500, 1500, 2000, 1000
    PlayTone: 988, 1.00, 1.00, 2
    Pause: 6.00
    SetServoRelays: false, false, false, false, false
    PlayTone: 1047, 1.00, 0.00, 1
  }
}
```

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.

Example: The following example causes a triangular geofence boundary to be active, from one minute until ten minutes after GPS fix acquisition.

```
Conditions:
{
  Arm:
  {
    Compare: Timestamp.MET = 60
  }
  Trigger:
  {
        GeoFence:
        {
          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.BatV
- AccMag.AccN
- AccMag.MagN
- AccMag.RPM
- Position.Longitude
- Position.Latitude
- Position.Altitude
- Position.GroundSpeed
- Position.VerticalSpeed

GeoFence:

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

```
GeoFence:
{
  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:

- **Remain** → "inside" triggers if the UAS Bounder goes outside of the geofence region, while "outside" will trigger if the UAS Bounder enters into the geofence region.
- **Altitude** → Minimum and maximum altitudes permitted over the entire geofence.
- **Waypoint** → Each GeoFence needs at least three and not more than twelve waypoints.

Actions

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

Options include:

- SetServoRelays → Takes a sequential string of up to five relay activation states.
  - Arguments: true or false, x5
  - If less than five arguments are supplied, then remainder will be set to false.
  - Note: Servo Relays remain in the last state set, unless SetServoRelays is called again by either the same Command's Actions or by a different Command
  - Relay Order:
    - Relay 1 → Controls SERVO_1 channel & LATCH signal
    - Relay 2 → Controls SERVO_2 & SERVO_3 channels
    - Relay 3 → Controls SERVO_4 & SERVO_5 channels
    - Relay 4 → Controls SERVO_6 & SERVO_7 channels
    - Relay 5 → Controls SERVO_8 channel
- SetServoPWMs → Takes a sequential string of up to eight servo channel PWM values.
  - Arguments: Servo PWM duration in microseconds (800 to 2200 typical), x8
  - If less than eight are supplied, then the remainder will be populated from either the UAS Bounder's default or previous values if set since power-on
  - Servo channel order is: 1, 2, 3, 4, 5, 6, 7, 8
- 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

Example

Below is a complete example that both tests the servos briefly at power-on and then ensures that the *UAS Bounder* stays within a geofence region.

- It configures the COMM_AP opto-isolated serial port for ASCII output at 57,600 baud, and at 2 second intervals. It sends all of the possible data to both this port and to the SD card (except GPS). The SD card will log data 2 Hz.
- It configures the GPS module to transmit the UBX-NAV-PVT message at 57600 baud on the GPS_TXD opto-isolated pin on the COMM_AP connector.
- It runs a brief (18 second) test 30 seconds after GPS fix is acquired.
- It also sets a geofence that triggers if the *UAS Bounder* is ever taken outside of a small area. If this geofence command triggers all of the relays are enabled, a 30 second tune is played, and the relays remain on until power is removed.

```
Version: 1
SDCardData: Timestamp, System, Position, AccMag
SDCardRate: 1
CommData: Timestamp, System, Position, AccMag
CommBinary: false
CommHeader: true
CommDevice: Serial, 1, 57600, 2
ServoMux: 25000000, 250
GPSComm: 57600, 7
Command: PwrOnTest
{
  Conditions:
  {
    Trigger:
    {
       Compare: Timestamp.MET   = 30.000000
    }
  }
  Actions:
  {
    SetServoRelays: false, true, false, true, true
    SetServoPWMs:  1000, 1000, 1000, 1000, 1000, 1000
    PlayTone: 784, 1.00, 1.00, 2
    SetServoPWMs:  ON, 2000, 2000, 2000, 2000, 1000
    PlayTone: 880, 1.00, 1.00, 2
    SetServoPWMs:  OFF, 1500, 1500, 1500, 2000, 1000
    PlayTone: 988, 1.00, 1.00, 2
    Pause: 6.00
    SetServoRelays: false, false, false, false, false
    PlayTone: 1047, 1.00, 0.00, 1
  }
}
Command: GeoTest1
{
  Conditions:
```

```
  {
    Trigger:
    {
        GeoFence:
        {
        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
        }
    }
  }
  Actions:
  {
    SetServoPWMs:  1400, 1500, 1600, 1700, 2000, 1000, 2000, 1000
    SetServoRelays: true, true, true, true, true
    PlayTone: 784, 1.00, 1.00, 15
  }
}
```

5 Log Files

The *Flight.csv* file logs data at up to 2 Hz, and is written at 1 Hz. 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
- Servo Status
    - S_01 Servo Channel 1 Status
    - S_02 Servo Channel 2 Status
    - S_03 Servo Channel 3 Status
    - S_04 Servo Channel 4 Status
    - S_05 Servo Channel 5 Status
    - S_06 Servo Channel 6 Status
    - S_07 Servo Channel 7 Status
    - S_08 Servo Channel 8 Status
- Measurements
    - BatV The battery voltage
    - NumSat Number of GPS Satellites currently used by the GPS module
    - Lon /Lat GPS longitude in degree decimal
    - Alt GPS altitude in meters above Ellipsoid
    - GndS GPS ground speed in m/s
    - VerS GPS vertical rate in m/s
    - HDG GPS heading in degrees

- AccX/Y/Z Accelerometer readings in indicated axis, in g's
- AccAgg Max Aggregate Acceleration of all 3 axis, in g's, since last report
- MagX/Y/Z Magnetometer readings in indicated axis, in degrees
- RPM Integrated Aggregate Magnetometer, in degrees per minute

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 *UAS 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 *UAS 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.

6 Mechanical

Four 3.2 mm (1/8 inch) holes for M3 (#4) screws are provided for mounting.

The GPS module (indicated by U5 below) has a fairly omni-directional antenna. It works best if the antenna is on the side or top when placed in the UAS. The orientation depicted below (GPS facing the ground) would be the worst possible flight orientation; both for the antenna orientation and because the servo wiring could partially obscure its view of the sky.

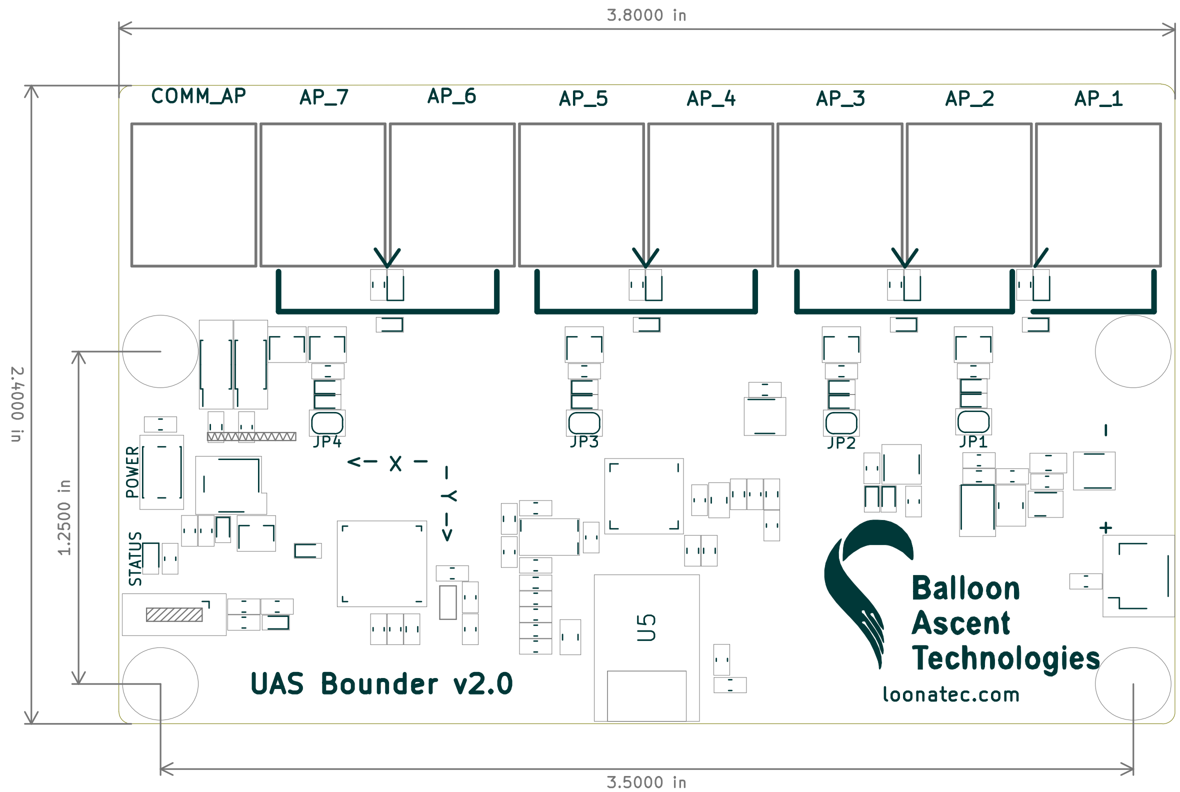


Figure 5: Mechanical Diagram

External GPS Antenna

There is an SMA connector that permits the connection of an external GPS antennas. If one is not connected, then the *UAS Bounder's* GPS will use the on-board chip antenna. The following external antenna options have been tested and worked:

- Active GPS Antenna: Mini GPS Antenna (ANT-GPSC-SMA)
- Passive GPS Antenna: Advantech 1750005865
- Passive GPS Antenna: ABRACON AEACAC054010-S915

7 Troubleshooting

Device Will Not Turn On

The UAS Bounder should turn on (Status LED illuminated) within two 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
- Check the battery orientation
- Install new fully charged batteries

Device Cannot Acquire GPS Fix

The UAS Bounder normally acquires GPS fix in less than 5 minutes. If it is not able to acquire a GPS fix within 20 minutes then please:

- Try power-cycling it as the GPS module sometimes seems to lock onto spurious signals at power-on
- Check that it has a clear and unobstructed view of most of the sky down to the horizon
- Check that there are GPS satellites overhead and visible to the UAS Bounder. https://in-the-sky.org/satmap_worldmap.php

Updating the Firmware

1. Place the UPDATE.bin file in the root directory of the micro-SD card
2. Insert the card
3. Press and hold the power-on button (5-7 seconds) until the tone starts playing
4. Let the UAS Bounder acquire GPS fix, then wait for an additional two minutes
5. Power off, remove the micro-SD card
6. 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

8 Consumables

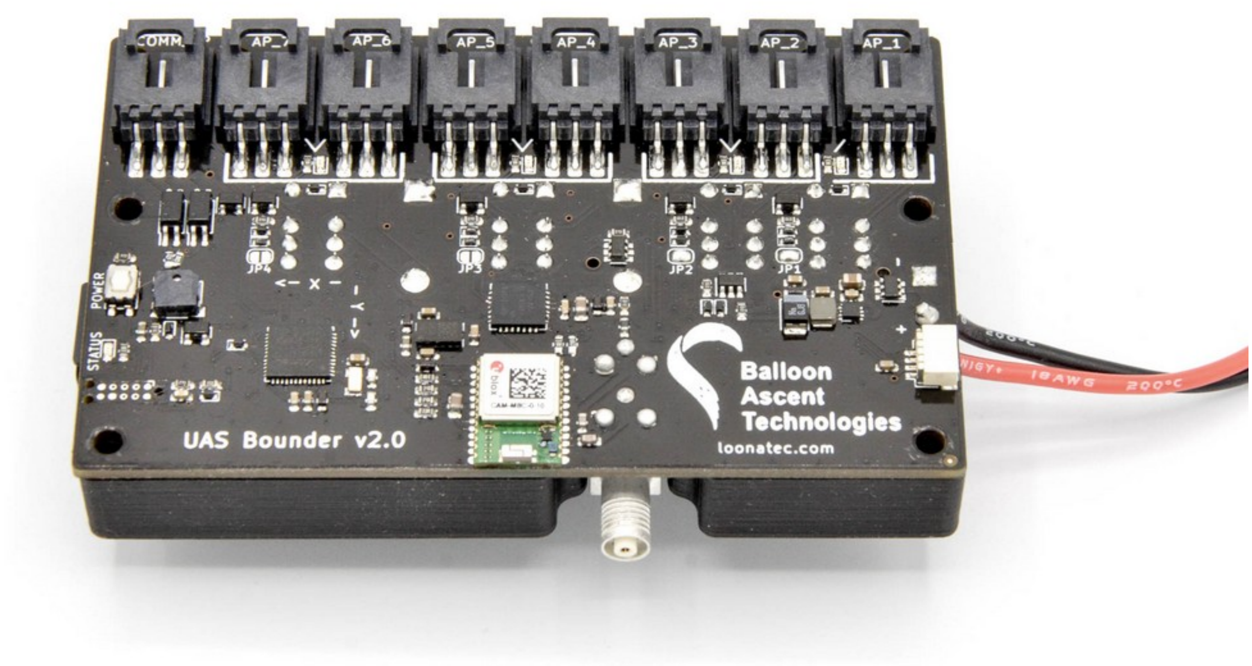
Batteries

The *UAS Bounder* has been designed to use a wide range of batteries. The important parameters are:

- Voltage range must stay within 3.6 to 17 volts DC
- Current capacity must be at least as great as the attached servos plus 100 mA for the *UAS Bounder's* operations

The *UAS Bounder* comes with a Mini-Tamiya connector with a polarity to match Titan Power's brick batteries. Other connectors can be configured upon request.

- Titan Power's 3 Ah 7.4V Brick Battery



Memory Card

Any SD/SDHD micro-SD card should work with the *UAS 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.

9 Data Messages

Timestamp Data Message

ASCII

YYYY/MM/DD HH:MM:SS, MET

Binary

Byte Offset	Number Format	Scaling	Decimal Places	Name	Unit	Description
0	U4			MET	s	**M**ission **E**lapsed **T**ime since GPS acquisition

---

System Data Message

ASCII

```
BatV, Status, S_01, S_02, S_03, S_04, S_05, S_06, S_07, S08,
```

Binary

Byte Offset	Number Format	Scaling	Decimal Places	Name	Unit	Description
0	I2	1e-2	1	BatV	V	Battery Voltage
2	X1			Status		System status flags (see graphic below)

Status Bitfield

7	6	5	4	3	2	1	0
			PowerSaveGPS		Iridium Active		

Position Data Message

ASCII

```
Longitude, Latitude, Altitude, GroundSpeed, VerticalSpeed, NumSat
```

Binary

Byte Offset	Number Format	Scaling	Decimal Places	Name	Unit	Description
0	I4	1e-7	5	LON	°	Longitude
4	I4	1e-7	5	LAT	°	Latitude
8	I4	1e-3	1	ALT	m	Altitude
12	I4	1e-5	2	HDG	°	Compass Heading
16	I4	1e-3	1	SPD	m/s	Ground Speed
20	I4	1e-3	1	AST	m/s	Ascent Rate

AccMag Data Message

ASCII

```
AccX, AccY, AccZ, AccAgg, MagX, MagY, MagZ, RPM
```

Binary

Byte Offset	Number Format	Scaling	Decimal Places	Name	Unit	Description
0	U2	1e-3	2	AccAgg	g	Max aggregate acceleration since last report

---

GPS Data Message

ASCII

NMEA GGA Sentence

Binary

uBlox UBX-NAV-PVT Message

10 Safety Precautions and Recommendations

Regulatory

Always adhere to the regulations governing Unmanned Aircraft Systems applicable in the country of operation.