Software Requirements Specification

for

Alticraft Flight Data Recorder

Version 3.3

Prepared by Robert Hutter

Dobsinalia International Student Research Group

08.18.2022

Table of Contents

1. Introduction....1
1.1 Purpose....1
1.2 Document Conventions....1
1.3 Intended Audience and Reading Suggestions....1
1.4 Product Scope....1
1.5 References....1
2. Overall Description....2
2.1 Product Perspective....2
2.2 Product Functions....2
2.3 User Classes and Characteristics....3
2.4 Operating Environment....3
2.5 Design and Implementation Constraints....4
2.6 User Documentation....4
2.7 Assumptions and Dependencies....4
3. External Interface Requirements....4
3.1 User Interfaces....4
3.2 Hardware Interfaces....5
3.3 Software Interfaces....5
3.4 Communications Interfaces....5
4. System Features....6
4.2 Configuration....7
4.3 Program preferences....8
4.4 Accelerometer and Gyroscope Data Measurements....11
4.5 Environmental Data Measurements....12
4.6 Basic Flight Process Management....12
4.7 Write Data to SD Card....13
5. Other Nonfunctional Requirements....13
5.1 Performance Requirements....13
5.2 Software versioning....14
5.3 Safety Requirements....14
5.4 Security Requirements....16
5.5 Software Quality Attributes....16
5.6 Business Rules....16

Revision History

Name	Date	Reason for Changes	Version
Alticraft FDR	12.20.2020	Public release of version 3.2	3.2
FDR Update 3.3	08.18.2022	Changes to software testing methods, added section for software versioning.	3.3

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the open-source software used in the Alticraft Flight Data Recorder. It will explain the features and interfaces of the software, what the software will do and the constraints under which it must operate. This document is intended for users of this software and potential developers.

1.2 Document Conventions

This Document was created based on the IEEE template for Software Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

- Partners and/or Customers, who are interested in modifying the Alticraft Flight Data Recorder software
- Typical Users, such as engineers, researchers, and students, who want to use the software for their own projects.
- Programmers who are interested in extending the features of the software.

1.4 Product Scope

Alticraft Flight Data Recorder software is a tool that controls the launch and staging of a rocket or other projectile. Furthermore, the software captures and saves data about the projectile's movement and environment for later research and analysis. It does this by interfacing with various real-time sensors and saves their data to an external drive.

Staging can be controlled with a servo motor, which is activated in a specified manner when the given trigger criteria is reached.

Alticraft Flight Data Recorder software is part of the Alticraft Flight Data Recorder Product and should only be used alongside specified and tested Alticraft FDR hardware.

1.5 References

Alticraft Flight Data Recorder Software's GitHub page:

https://git.dobsinalia.com/RobertHutter/AlticraftFlightDataRecorder

Standard Robotics Library's GitHub page:

https://git.dobsinalia.com/RobertHutter/StandardRoboticsLibrary

IEEE Template for Software Requirement Specification Documents.

https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc

MIT License:

https://opensource.org/licenses/MIT

Arduino's website:

https://arduino.cc

I^{2}c communication protocol:

https://i2c.info/

SPI communication protocol:

https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html

MPU9250 product specification:

https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3 4.pdfhttps://www.hestore.hu/prod_getfile.php?id=9347

MPU9250 Register map and description:

https://invensense.tdk.com/download-pdf/mpu-9250-register-map/

BMP280 product specification:

https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp280-ds001.pdf

Servo motor documentation:

https://www.arduino.cc/en/Reference/Servo

2. Overall Description

2.1 Product Perspective

Alticraft Flight Data Recorder was developed for everyone interested in analyzing moving objects throughout their complete motion, and for those who want to add basic flight controlling features to their airborne systems.

Alticraft Flight Data Recorder software is designed to operate on Arduino Nano V3 compatible 5V microcontrollers. The software is only tested to operate on such hardware (see performance requirements).

2.2 Product Functions

2.2.1 Interface:

- Connect to MPU9250: establish a connection with the MPU9250 board using the I^{2}C communication protocol.
- Connect to BMP280: establish a connection with the BMP280 board using the I^{2}C communication protocol.
- Connect to SD card reader: establish a connection with the SD card reader using the SPI communication protocol.
- Control the color of an RGB LED: control the color of a status RGB using three analog output pins.
- Toggle warning buzzer: toggle a warning buzzer using a digital output pin.
- Listen for push button activation: listen for the activation of a push button and activate set functions on that signal.

2.2.2 MPU9250:

- Initialize: initialize the connection and setup the device for usage.
- Test connection: test the connection with the device.
- Set gyroscope and accelerometer sensitivity.
- Get accelerometer readings: retrieve the devices acceleration readings on all three axes.
- Get gyroscope readings: retrieve the devices gyroscopic rotational readings on all three axes.

2.2.3 BMP280:

- Initialize: initialize the connection and setup the device for usage. Set power mode and oversampling rates.
- Test connection: test the connection with the device. Read device ID from the module.
- Get barometric pressure readings: retrieve barometric pressure readings from the sensor.
- Get temperature pressure readings: retrieve current temperature readings from the sensor.

2.2.4 SD card reader:

- Initialize: initialize the connection and setup the device for usage. Create workspace folder.
- Test connection: test the connection with the device.
- Create new file: create a new file with a given valid DOS 8.3. filename.
- Write to file: write data to a specific file.

2.2.5 Basic flight process management:

- Toggle launch signal: toggle a launch signal using a digital output pin.
- Control staging servo motor: control the position of a servo motor using an analog output pin.

2.3 User Classes and Characteristics

- Typical Users, such as engineers, researchers, or students, who want to use the software for their own projects.
- Programmers who are interested in working on the project.

2.4 Operating Environment

- Arduino Nano V3 compatible boards which meet memory requirements and operate on 5V

Please note, Alticraft Flight Data Recorder software is designed to run on Arduino Nano V3 compatible boards. Although it may run on other devices, it has only been tested to function properly on Arduino Nano V3 compatible microcontrollers.

2.5 Design and Implementation Constraints

Alticraft Flight Data Recorder is developed in C++ using the Arduino Development Environment. It is important to note, that the Arduino Development Environment does not support all features of the Standard Template Library, the most basic C++ API, however, has added functionality to control the Arduino board. Many components of this software are written into separate source files connected by one main execution file.

2.6 User Documentation

Readme file:
https://git.dobsinalia.com/RobertHutter/AlticraftFlightDataRecorder/src/branch/master/README.md

2.7 Assumptions and Dependencies

Alitcraft Flight Data Recorder is developed to work with Arduino libraries. Therefore, the code will only compile when specified dependencies are resolved, and will only run on the explicitly listed microcontrollers.

The SD card used as the external drive must by formatted to the FAT32 filesystem with sufficient empty storage to save measurements.

3. External Interface Requirements

3.1 User Interfaces

Alticraft Flight Data Recorder does not have any kind of graphical user interface or command line interface. The program executes automatically on startup of the device. Operation of the device can be monitored with a multicolor status led and feedback from the buzzer.

The user can send commands using a push-button. Device feedback is given through the multicolor status led.

3.1.1 Status led color codes:

- **Solid blue:**
  initialization of components in progress.
- **Solid white:**
  - After initialization: an error has occurred during the initialization/test phase.
  - After flight: recording has been stopped, and it is safe to remove power from the device.
- **Solid green:**
  initialization stage has successfully completed, and the device is waiting for the launch signal.
- **Solid red:**
  data recording in progress.
- **Change from green to red:**
  launch signal received, release pushbutton.

- **Flashing red/white:**
  launch countdown in progress.

3.2 Hardware Interfaces

Alticraft Flight Data Recorder uses the I^{2}C and SPI (Serial Peripheral Interface) communication protocols to communicate with its different hardware components.

The MPU9250 and BMP280 devices use the I^{2}C protocol for communication, while the SD card reader uses SPI. The status led, and servo motor are controlled using analog output signals. The onboard buzzer and launch signal are controlled by digital output signals.

3.3 Software Interfaces

Alticraft Flight Data Recorder uses the Standard Robotics Library API developed by Dobsinalia for I^{2}C, and the Arduino SPI API for SPI communication.

To write to the SD card’s memory, the SdFat library from William Greiman is used. Servo motor control is done using built in motor control libraries.

3.4 Communications Interfaces

Alticraft Flight Data Recorder uses the I^{2}C and SPI communication protocol to communicate with its hardware components.

3.4.1 Debug interface (via USB serial communication)

Example debug output:

```
-------------------------------------------
ALTICRAFT FLIGHT DATA RECORDER_

COPYRIGHT (C) ROBERT HUTTER 2020

VERSION: 1.0
BUILD DATE: Dec 20 2020
OPERATIONAL MODES:
LAUNCH_LOG_STAGE_
LAUNCH_STABILITY_ABORT_
TOUCHDOWN_DETECTION_
SAR_HELPER_
STAGING TRIGGER MODE: ALTITUDE
TRIGGER VALUE: -200 m
TOUCHDOWN DETECTION ARMED 5000 milliseconds AFTER LAUNCH_
-------------------------------------------
INITIALIZATION PHASE BEGINNING_
Initializing RGB LED...
Initializing MPU9250...
Initializing BMP280...
Initializing buzzer...
Initializing digital output pins...
Initializing servo motor...
Initializing SD module...
CARD TYPE: 3_
```

```
CLUSTERS: 979712_
BLOCKS x CLUSTER: 8_
TOTAL BLOCKS: 7837696_
VOLUME TYPE: FAT32_
VOLUME SIZE: 3918848 kb_
INITIALIZATION PHASE ENDED_
------------------------------------------
TESTING PHASE BEGINNING_
TESTING PHASE ENDED_
ALL SYSTEMS NOMINAL_
ARMING IGNITION SYSTEM_
------------------------------------------
WAITING FOR LAUNCH SIGNAL_
LAUNCH SIGNAL RECIVED_
------------------------------------------
WARNING! IGNITION IN 10 SECONDS_
PRESS ANY BUTTON TO CANCEL_
------------------------------------------
LIFTOFF_
IGNITION AT: 18725228 microseconds_
------------------------------------------
------------------------------------------
STAGING CONDITION MET AT: 3009780500 microseconds_
ACTIVATING STAGING SERVO_
------------------------------------------
TOUCHDOWN DETECTED AT: 5008735080 microseconds_
STOPPING RECORDING_
SETTING LED TO BLUE_
ALERTING SEARCH AND RESCUE_
PRESS AND HOLD BUTTON TO SHUTDOWN_
------------------------------------------
SHUTDOWN COMMAND RECIVED_
SHUTDOWN TIME: 5408682148_
SETTING RGB LED OFF_
SHUTTING DOWN_
```

4. System Features

This section demonstrates Alticraft Flight Data Recorders most prominent features and explains how they can be used and the results they will give back to the user.

Alticraft Flight Data Recorder has three main operational modes. The desired mode must be selected in the source code’s main file under program preferences.

4.1.1 The three core operation modes

- **PASSIVE_LOG:**
  Only logs flight data. Disables launch and staging control.
- **ONLY_LAUNCH_AND_LOG:**
  Enables only launch control and flight data logging.
- **LAUNCH_LOG_STAGE:**
  Enables all features including launch, staging control, and flight data logging.

If a mode is selected which supports staging, a staging trigger must be selected too. This can be done in the source code's main file under the operation mode selector. There are four trigger modes. All trigger modes operate completely independently from each other, which means multiple triggers can be activated at the same time.

4.1.2 The four staging trigger modes

- **ALTITUDE:**
  Activates staging at a specific altitude (m), derived from the barometric pressure readings.
- **PRESSURE:**
  Activates staging at a specific pressure (Pa).
- **TEMPERATURE:**
  Activates staging when the measured temperature leaves a given range (°C).
- **APOAPSIS:**
  Activates staging when a constant decline in altitude is detected.

More information regarding the four staging trigger modes is available at section 4.5. Basic Flight Process Management.

4.1.3 Additional features and modes

- **LAUNCH_STABILITY_ABORT:**
  This feature monitors accelerometer data for anomalies during the launch countdown and aborts launch if a sudden change in attitude is detected. Following a launch abort the system will shut down, meaning a full restart is needed to restart the launch sequence.
- **TOUCHDOWN_DETECTION:**
  This feature watches sensor data and attempts to detect touchdown. If this is detected, recording is finished, and the system lingers in low power mode until it is shut down using the push button.
- **SAR_HELPER:**
  This activates the buzzer upon touchdown to help direct search and rescue teams towards the fallen projectile.

  If solely this mode is activated (without TOUCHDOWN_DETECTION), then the system will still detect touchdown, but will not stop recording and switch to a low power state.

In addition to those above, the user may also to edit key operational parameters.

4.2 Configuration

The Alticraft Flight Data Recorder can be configured based on the information in this document by modifying the source code. All configurable settings are in the header file named AlticraftFDRConfig.h. The other header file named AlticrafFDR.h contains device/hardware specific preferences like accelerometer and gyroscope offset values, or calibration data for the BMP280. These values are to be set once and not to be modified later in use.

Program preferences located in the AlticraftFDRConfig.h header file may be edited manually or using the Alticraft Config Wizard. This CLI program helps configure the settings by following all configuration rules specified in this document (i.e., required parameters, conflicting modes, etc...). For this reason, it is highly recommended to use the Alticraft Config Wizard when modifying program parameters.

4.3 Program preferences

Editable operation parameters and their default settings (in alphabetical order):

- **ACCEL_SENSITIVITY:**
  MPU9250 accelerometer sensitivity setting. Default value is 0. Table containing sensitivity data and corresponding settings below.

FDR Setting Value	Full-Scale Range	Sensitivity Scale Factor
0	±2 g	16384 LSB/g
1	±4 g	8192 LSB/g
2	±8 g	4096 LSB/g
3	±16 g	2048 LSB/g

- **ACCEL_X_OFFSET:**
  x-axis accelerometer offset value. Default value is 0. Please set this to your devices own specific value.
- **ACCEL_Y_OFFSET:**
  y-axis accelerometer offset value. Default value is 0. Please set this to your devices own specific value.
- **ACCEL_Z_OFFSET:**
  z-axis accelerometer offset value. Default value is 0. Please set this to your devices own specific value.
- **BAD_MEAS_ABORT_THRESHOLD:**
  Maximum amount of sequential false readings allowed during launch countdown before the system aborts launch due to a suspected hardware sensory fault. Default value: 100
  Note: actual number of readings which can be false depends on the measurement interval (empty readings, created during off times also count towards being false measurements).
- **BAD_WRITE_ABORT_THRESHOLD:**
  Maximum number of times an error can sequentially occur during writing to the SD card during the launch countdown before the system aborts launch due to a suspected fault. Default value: 100
  Note: actual number of write failures allowed depends on the measurement interval (empty loop iteration during off times also count towards being write exceptions).
- **BASE_PRESSURE:**
  Barometric base pressure. Height will be calculated based on this base value. Unit Pa. Default value: 100325 (standard barometric pressure)
- **BAUD_RATE:**
  Data transfer rate for serial USB communication. Default 9600 b/s
- **BMP280_DIG_T1 through T3:**
  BMP280 factory calibration values for temperature measurement. Default is 0. Please set these to your devices own specific values.
- **BMP280_DIG_P1 through P9:**
  BMP280 factory calibration values for pressure measurement. Default value: 0. Please set these to your devices own specific values.
- **DEBUG:**
  Debug mode on/off. Default value ON.

- **GYRO_SENSITIVITY:**
  MPU9250 gyroscope sensitivity setting. Default value is 0. Table containing sensitivity data and corresponding settings below.

FDR Setting Value	Full-Scale Range	Sensitivity Scale Factor
0	±250 °/s	131 LSB/(°/s)
1	±500 °/s	65.5 LSB/(°/s)
2	±1000 °/s	32.8 LSB/(°/s)
3	±2000 °/s	16.4 LSB/(°/s)

- **GYRO_X_OFFSET:**
  x-axis gyroscope offset value. Default value is 0. Please set this to your devices own specific value.

- **GYRO_Y_OFFSET:**
  y-axis gyroscope offset value. Default value is 0. Please set this to your devices own specific value.

- **GYRO_Z_OFFSET:**
  z-axis gyroscope offset value. Default value is 0. Please set this to your devices own specific value.

- **LAUNCH_ABORT_SENSITIVITY:**
  Maximum deviation in accelerometer data allowed during launch countdown before the launch stability system aborts launch (if LAUNCH_STABILITY_ABORT mode active). Default value: 0.2 (Δg between two measurements (defined by MEASUREMENT_RATE))

- **LAUNCH_ARM_TIME:**
  Time required to start launch sequence (in milliseconds). The user will have to press and hold the push button for this amount of time to initiate the countdown. Default value: 3000.

- **LAUNCH_COUNTDOWN:**
  Length of countdown (in milliseconds) before launch Time period between launch sequence activated and actual launch. Default value: 10000

- **NEWFILE_INTERVAL_TIME:**
  The interval (in milliseconds) at which to create a new log file and write to that. This feature is designed to avoid file corruption. Default value: 2000 (2 seconds).

- **MEASUREMENT_RATE:**
  Number of measurements to be taken within 1 second. Default value: 50.
  Note: the actual measurement frequency may vary from this setting due to hardware limitations.
  To disable measurement rate limiting, comment the line defining this parameter.

- **PRESSURE_OVERSAMPLING:**
  Pressure oversampling value. Default is 3. Higher oversampling setting provides for more accurate reading at the expense of time (measurement takes longer). Table containing pressure oversampling data below.

FDR Setting Value	Pressure Oversampling	Typical Pressure Resolution	Recommended Temperature Oversampling
1	x1	16 bit / 2.62 Pa	x1
2	x2	17 bit / 1.31 Pa	x1
3	x4	18 bit / 0.66 Pa	x1
4	x8	19 bit / 0.33 Pa	x1
5	x16	20 bit / 0.16 Pa	x2

- **SERVO_STAGE_ANGLE:**
  Angle (in degrees) of staging servo after staging is activated. Default value: 90. Value must be between the following range: 0-180

- **SERVO_START_ANGLE:**
  Starting angle (in degrees) of stage servo. Default value: 0. Value must be between the following range: 0-180.

- **SHUTDOWN_TIME:**
  Amount of time (in milliseconds), that the push button must be pressed to shut down the system. Default value: 3000.

- **STAGE_TRIGGER_TIME:**
  Amount of time (in milliseconds), in which the staging trigger must be met to activate actual staging. This delay helps filter out erroneous readings. Default value: 1000.

- **TEMPERATURE_OVERSAMPLING:**
  Temperature oversampling value. Default is 1. Table containing temperature oversampling data below.

FDR Setting Value	Temperature Oversampling	Typical Temperature Resolution
1	x1	16 bit / 0.0050 °C
2	x2	17 bit / 0.0025 °C
3	x4	18 bit / 0.0012 °C
4	x8	19 bit / 0.0006 °C
5	x16	20 bit / 0.0003 °C

- **TOUCHDOWN_DETECTION_FROM:**
  Time (in milliseconds), counted from launch, after which the touchdown detection system is armed. This delay helps prevent touchdown detection from activating on the launch pad. Default value: 5000.

- **TOUCHDOWN_DETECTION_TIME:**
  Amount of time (in milliseconds), in which touchdown detection criteria must be met before activating touchdown detection. Default value: 10000.

For staging trigger specific settings refer to Trigger criteria.

4.4 Accelerometer and Gyroscope Data Measurements

4.1.1 Description and Priority

Read accelerometer and gyroscope data from the MPU9250 and save it to the SD card.

4.1.2 Stimulus/Response Sequences

The reading and saving process happens on a set time interval determined by the specific application of the system and the available hardware specifications.

4.1.3 Functional Requirements

Data should be written to the file in a manner that makes post flight analysis as simple as possible (easily importable into Microsoft Access® and Microsoft Excel®).

4.5 Environmental Data Measurements

4.1.1 Description and Priority

Read data about the environment from the BMP280 and save it the SD card, including barometric pressure and temperature.

4.6 Basic Flight Process Management

4.1.1 Description and Priority

Manage basic flight processes by controlling a digital launch signal and by controlling the position of a staging servo motor. Activate a warning buzzer before sending the launch signal.

4.1.2 Stimulus/Response Sequences

The launch signal is activated by a push button, while operation of the staging motor is triggered by reaching a specified launch trigger altitude/temperature/pressure or apoapsis height.

All staging trigger modes operate completely independently from each other, which means multiple triggers can be activated at the same time.

4.6.1.1 Trigger criteria

Below is a table, which explains the exact trigger criteria for each individual staging trigger mode. For the staging to be activated the specified trigger condition must stay true for at least the time specified in the STAGE_TRIGGER_TIME option.

Trigger Mode	Condition Description
Pressure	Trigger condition is met, when the ambient pressure is LOWER than the trigger pressure set (pressure decreases upwards).
Temperature	Trigger condition is met when the ambient temperature leaves a given range (bound by min and max). The range can be set with two temperature values. (TRIGGER_TEMPERATURE_MIN and TRIGGER_TEMPERATURE_MAX) To invert the trigger range, enter the max temperature value as the min one, and the min one as the max temperature value.
Altitude	Trigger condition is met when the measured altitude is HIGHER than the trigger altitude set.
Apoapsis	Trigger condition is met, when the measured altitude is constantly decreasing for given time set in the STAGE_TRIGGER_TIME option.

4.1.3 Functional Requirements

The staging servo motor must be installed and configured properly for staging to work.

4.7 Write Data to SD Card

4.1.1 Description and Priority

During flight, the FDR takes many measurements from the environment, which must be saved to the SD Card. This is done with interfacing with an SD card using the SPI communication protocol.

Data must be saved in a format, which enables simple data analysis and evaluation. The simplest format, in which the data could be stored is the comma separated value format (CSV).

Debug information is stored to a file called “DEBUG.TXT” if debug mode is active.

The workspace (folder) naming convention is very simple (since it must comply with the DOS 8.3 filename requirements), it has the format “*FLTXXX*”, where XXX is the number of already existing workspaces on the device (to avoid name collision). The counter starts at 0, so the first folder is named “*FLT0*” (second “*FLT1*”.

4.1.2 Stimulus/Response Sequences

Data recording begins during launch countdown and continues until manual discontinuation or touchdown is detected. During this period data is saved to the SD card in log files, which are generated every few seconds, as defined by the NEWFILE_INTERVAL_TIME preference value.

4.1.3 Functional Requirements

The data is saved in CSV (comma separated values) format as follows.

Note: Temperature data is saved in a fixed point number format. This means, that a datapoint with the value of “*2805*”, should be interpreted as “*28.05*”.

Timestamp	Pressure	Altitude	Temp	Accel X	Accel Y	Accel Z	Gyro X	Gyro Y	Gyro Z
µs	Pa	m	°C	g	g	g	°/s	°/s	°/s

timestamp,pressure,altitude,temp,accelX,accelY,accelZ,GyroX,GyroY,GyroZ

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Alticraft Flight Data Recorder software is designed to run on Arduino Nano V3 compatible devices. Due to the Arduino Nano’s hardware limitations, Alticraft FDR must adhere to the following memory constraints:

Program Storage Space	Dynamic Memory
30720 bytes	2048 bytes

5.2 Software versioning

Software versioning is done based on the Semantic Versioning Specification (SemVer). In compliance with the specification all published versions are assigned a three digit version number in the following format: x.y.z. In addition, development versions/pre-release versions may be assigned additional version numbers in the following format: x.y.z-alpha.n.

The following table contains all released software versions and the corresponding Software Requirements Specification document version they are based on:

Software version	SRS version	Additional information
1.0.0-alpha.x	-	Development version
1.0.0-beta.1.0	3.3	Virtually tested to meet all requirements of the SRS. Physical testing is required.
1.0.0	3.3	Tested virtually and physically to meet all requirements of the SRS.

5.3 Safety Requirements

Due to the operating nature (projectile flight control) of Alticraft Flight Data Recorder, it is of highest importance, that the software runs stabile and reliably. Due to this reason, Alticraft Flight Data Recorder software must undergo rigorous testing, based on a predetermined schedule.

Testing and validating the software is done using the Alticraft Runtime Simulator. This Windows C++ program emulates the actual Arduino runtime environment and allows for testing with predefined flight profiles without physically needing to conduct a test. Flight profiles for different test cases can be generated using the Alticraft Flight Data Generator. The following section lists all different system configurations and flight profiles which require testing.

5.3.1 Software Testing Schedule

Below is a table containing all software configurations, in which Alticraft FDR must be tested to operate. All possible configurations must be tested virtually using the Alticraft Runtime Simulator before releasing a beta-version software release. Popularly used configurations must be tested physically before releasing a production software release. Testing results for each version shall be documented in a software testing log document.

All operational modes below must be tested in both DEBUG and normal operational modes.

Operational Mode	Stage Trigger (if applicable)	Trigger value	Other Test Requirement
PASSIVE LOG	n/a	n/a	n/a
ONLY_LAUNCH AND_LOG	n/a	n/a	Touchdown at positive altitude
	n/a	n/a	Touchdown at negative altitude
LAUNCH_LOG_STAGE	Pressure	Positive number	n/a
	Altitude	Positive number	n/a
		Negative number	n/a
	Temperature	Positive number	n/a

		Negative number	n/a
	Apoapsis	n/a	Apoapsis at positive altitude
		n/a	Apoapsis at negative altitude
TOUCHDOWN_DETECTION	n/a	n/a	n/a
TOUCHDOWN_DETECTION and SAR_HELPER	n/a	n/a	n/a
SAR_HELPER	n/a	n/a	n/a
LAUNCH_STABILITY_ABORT	n/a	n/a	n/a

In addition to those above the following features and settings must be tested:

- Accelerometer and gyroscope sensitivity (0 through 3)
- Pressure and temperature oversampling (1 through 5)
- Measurement frequency limiting set and disabled

Abnormal operational configurations and events, which also must be tested:

Abnormal event/configuration	**Operational mode**	**Expected behavior**
BMP280 disconnects during launch countdown	LAUNCH_LOG_STAGE or ONLY_LAUNCH_AND_LOG	System detects abnormality and aborts launch.
BMP280 disconnects during flight	LAUNCH_LOG_STAGE	False readings are disregarded during staging and touchdown detection.
MPU9250 disconnects during launch countdown	LAUNCH_LOG_STAGE or ONLY_LAUNCH_AND_LOG	System detects abnormality and aborts launch.
MPU9250 disconnects during flight	LAUNCH_LOG_STAGE	False readings are disregarded during staging and touchdown detection.
SD card disconnects during launch countdown	LAUNCH_LOG_STAGE or ONLY_LAUNCH_AND_LOG	System detects abnormality and aborts launch.
SD card disconnects during flight	n/a	SD card failure has no effect on other subsystems (i.e., staging, touchdown detection)

5.3.2 Other Safety Requirements

To prevent data loss during recording due to corrupted files, Alticraft Flight Data Recorder saves data into separate files based on timestamp. The interval at which to create a new log file can be set in the program preferences. (See section 4. System Features)

Upon activating the push button, before the launch signal is sent, a warning buzzer is activated, alongside with visible warning signs from the status RGB LED.

An additional launch safety feature is the LAUNCH_STABILITY_ABORT feature, which when activated will abort launch if a change in attitude is detected during the launch countdown sequence (see Additional features and modes).

To increase data security and system safety, the system will automatically abort launch if any of Alticraft Flight Data Recorders components fail during launch countdown. Component features are compartmentalized, meaning all failures are contained within each subsystem. For example, the failure of the SD card has no effect on staging, touchdown detection, and SAR helper functions.

5.4 Security Requirements

Alticraft Flight Data Recorder does not have any security requirements and thus any type of user can use it without any additional privileges.

5.5 Software Quality Attributes

Alticraft Flight Data Recorder is designed for the highest level of stability to ensure that no data is lost, and that the measurements do not stop unexpectedly.

The software goes through a rigorous testing program (see Software Testing Schedule) before release, which ensures software quality and function reliability.

5.6 Business Rules

Alticraft Flight Data Recorder is open-source software and can be used by anybody at any time for any purpose.

Disclaimer:

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

Appendix A: Glossary

- **Accelerometer:** a device used to measure the acceleration of an object at a given moment.
- **Accelerometer/gyro offset**: a constant value used to zero accelerometer values, when the device is stationary. Offset values can be calculated using accelerometer/gyroscope calibration software.
- **API**: Auxiliary Programming Interface. Part of software which can be accessed and used by other programs.
- **DOS 8.3. filename:** Filename convention originally used by Microsoft on older versions of DOS and Microsoft Windows.

- **FAT32:** File Allocation Table 32. A filesystem originally developed for floppy disks, widely used as the filesystem of hard drives and SD cards-
- **Gyroscope:** a device used to measure the rotation of an object at a given moment.
- **I^{2}C:** Inter-Integrated Circuit. A 2 wire serial communication protocol.
- **IEEE:** Institute of Electrical and Electronics Engineers
- **LSB:** least significant bit. Used when discussing sensitivity settings of accelerometer and gyroscope. The LSB number specifies the sensitivity of a given measurement range.
- **SD card:** Secure Digital card. Proprietary non-volatile memory card.
- **SPI:** Serial Peripheral Interface. A 4 wire serial communication protocol.
- **RGB LED:** a light emitting diode with variable red, green and blue light components, which allows for the user to prescribe operation color.