Software Requirements Specification

for

Alticraft Flight Data Recorder

Version 2.2 approved

Prepared by Robert Hutter

Dobsinalia Aerospace Exploration & Smart Crew[2012]

2019.07.24

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Alticraft FDR	2019.05.20	Initial release	1.0
FDR Upgrade 1	2019.05.23	Added new functionality, changed sensors	1.5
FDR Upgrade 2	2019.05.24	Added new functionality, added new hardware	2.0
FDR Review 1	2019.07.18	Reviewed document, clarified items	2.1
FDR Review 2	2019.07.24	Reviewed document, added detail	2.2

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 of the software, the 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

- 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.

1.4 Product Scope

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

1.5 References

Alticraft Flight Data Recorder Software's GitHub page:
https://github.com/Smartheads/AlticraftFlightDataRecorder
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
i2c 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://www.hestore.hu/prod_getfile.php?id=9347
BME280 product specification.

https://www.hestore.hu/prod_getfile.php?id=10549

SD card reader specification:

https://www.vishnumaiea.in/projects/hardware/interfacing-catalex-micro-sd-card-module

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 projectiles.

It was developed to run on Arduino/Geniuno compatible boards but optimized for the Arduino Nano.

2.2 Product Functions

Interface:

- Connect to MPU9250: establish a connection with the MPU9250 board using the i2c communication protocol.
- Connect to BME280: establish a connection with the BME280 board using the i2c 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.

MPU9250:

- Initialize: initialize the connection and setup the board for usage.
- Test connection: test the connection with the board.
- Set accelerometer offsets: set the boards accelerometer offsets, by sending them to the board.
- Get accelerometer readings: get the boards acceleration readings on all three axes.
- Set gyroscope offsets: set the boards gyroscope offsets, by sending them to the board.
- Get gyroscope readings: get the boards gyroscopic rotational readings on all three axes.

BME280:

- Initialize: initialize the connection and setup the board for usage.
- Test connection: test the connection with the board.
- Get barometric pressure readings: get the barometric pressure readings from the sensor.
- Get temperature pressure readings: get the current temperature readings from the sensor.
- Get humidity readings: get the current air humidity readings from the sensor.

SD card reader:

- Initialize: initialize the connection and setup the board for usage.
- Test connection: test the connection with the board.
- Create new file: create a new file with a given name.
- Write to file: write data to a specific file.

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
- Arduino Uno
- Arduino Due
- Arduino Mega

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. All components of the software are written into separate modules connected by one main file.

2.6 User Documentation

Readme file on GitHub:
https://github.com/Smartheads/AlticraftFlightDataRecorder/blob/master/README.md

2.7 Assumptions and Dependencies

Alitcraft Flight Data Recorder is developed in the Arduino Development Environment using the Arduino IDE. The code will only compile in the specified environment and will only run on the explicitly listed microcontrollers.

The used SD card used as the external drive must by using the FAT32 filesystem.

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 runs automatically on startup of the board and uses a multi status light to show that the current status of the system such as, the initialization phase has completed successfully, and the system is recording flight data, or that the launch signal will be sent soon.

The user can send commands using a push-button. Feedback is given through the previously mentioned RGB light emitting diode.

Color codes:

- Blue: initialization/test phase in progress.
- White: an error has occurred during the initialization/test phase. After flight white means, recording has been stopped, and it is safe to remove power from the device.
- Green: all systems nominal, waiting for launch signal.
- Red: recording data. The ignition signal is sent, when the light changes from green to red.

3.2 Hardware Interfaces

Alticraft Flight Data Recorder uses the i2c and SPI communication protocol to communicate with the different hardware components. Two of the three devices use the i2c protocol (the MPU9250 and BME280), while the third uses SPI (SD card reader). The status RGB LED light is toggled using three analog output pins. The launch signal is sent using a digital output pin.

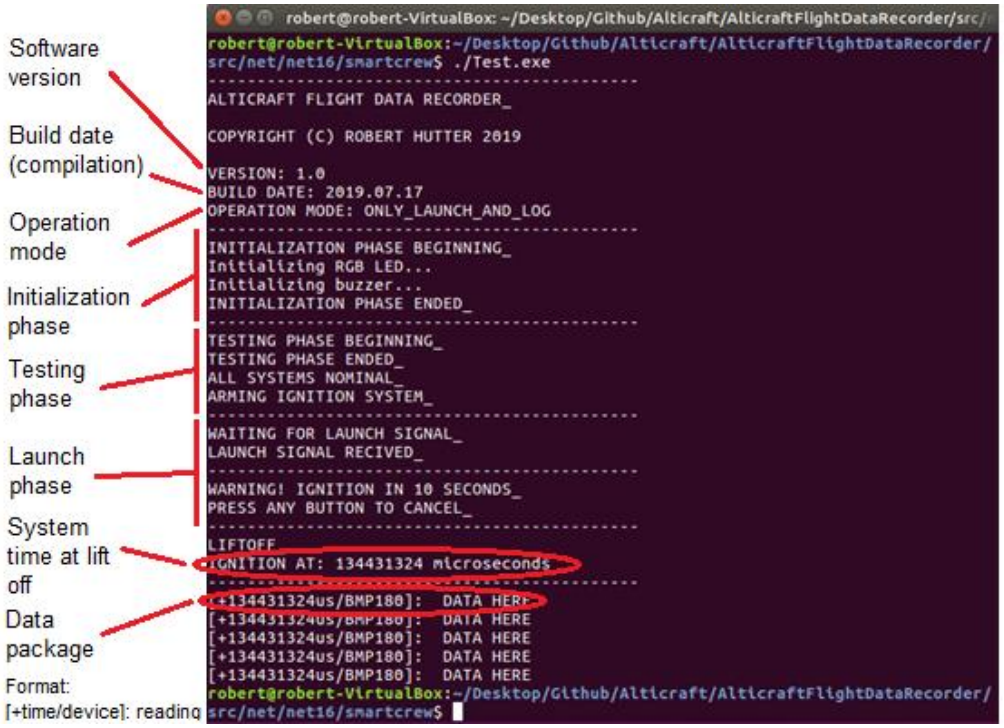
3.3 Software Interfaces

Alticraft Flight Data Recorder uses the I2Cdevlib API developed by Jeff Rowberg to communicate in the i2c protocol, and the built in SPI API for the other protocol. In order 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 i2c and SPI communication protocol to communicate with connected devices. All data sent to the external drive is mirrored to the USB serial port.

Overview of messages during operation:



4. System Features

This section demonstrates Alticraft Flight Data Recorder's 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.

The three core operation modes:

- PASSIVE_LOG: Only log 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 trigger must be selected too. This can be done in the source code's main file under the operation mode selector. There are two trigger modes.

The three staging trigger modes:

- PRESSURE: Activates staging at a specific altitude (m), derived from the barometric pressure readings.
- ALTITUDE: Activates staging at a specific pressure (kPa).

Other then setting the operation mode, the preferences allow the user to edit some key operational parameters.

Editable operation parameters and their default settings:

- DEBUG mode on/off. Default value ON.
- BAUD_RATE for serial communication. Default 115200
- NEWFILE_INTERVAL_TIME: The interval (in microseconds) at which to create a new log file and write to that. This feature is designed to avoid file corruption. Default value: 3000000 (3 seconds).

4.1 Accelerometer and gyroscope data measurements

4.1.1 Description and Priority

Read accelerometer and gyroscope data from the MPU9250 and save it to a file.

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®).

4.2 Environmental data measurements

4.1.1 Description and Priority

Read data about the environment from the BME280 and save it to a file, including barometric pressure, temperature and humidity.

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 manor that makes post flight analysis as simple as possible (easily importable into Microsoft Access® and Microsoft Excel®).

4.3 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 the apoapsis height.

4.1.3 Functional Requirements

A push button must be installed, and the barometric pressure sensor must be operating nominally.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Alticraft Flight Data Recorder should run comfortably on all specified devices (see 2.4).

5.2 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.

5.3 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.

5.4 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 until commanded.

5.5 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.
- Gyroscope: a device used to measure the rotation of an object at a given moment.
- Arduino: a small microcontroller with a size that fits in the palm of a hand.
- RGB LED: a light emitting diode with variable red, green and blue light components, which allows for the user to prescribe operation color.