CO_2 SOFTWARE

Requirements Specification Document

November 20th, 2017

Sponsor

Dr. Andrew Richardson

Dr. Mariah Carbone

Mentor

Ana Paula Chaves Steinmacher

Team

Sam Beals, James Beasley,
Andrew Greene, Joseph Kelroy

Accepted as baseline requirements for the project:

__________________________ __________________________

Sponsor’s Signature Team Lead’s Signature

Table of Contents

1. Introduction

Global climate change is an issue that will affect every person on the planet. It is widely acknowledged that climate change is driven by rising levels of atmospheric carbon dioxide (CO2) resulting from fossil fuel burning. The importance of research into climate change cannot be understated, but when people think of data associated with climate change they typically think of the carbon footprint of our cities. They often forget that the environment and its carbon cycle hold data that is just as, if not more, valuable. Our goal is to ensure that ecologists can collect data as quickly, efficiently, and as easily as possible from the environment.

The Richardson-Carbone Lab is operated by Doctor Andrew Richardson and Doctor Mariah Carbone at Northern Arizona University. The Richardson-Carbone Lab studies carbon cycling in forest ecosystems. In a nutshell, the research is used to understand the balance between carbon uptake (photosynthesis by plants) and release (respiration both by living and growing plants, as well as by microorganisms decomposing dead organic matter in the soil). Through the use of modern technology, we aim to make the collection of CO_2 data from trees a streamlined process with minimal cost and intuitive user interaction, providing users the ability to collect, visualize, and interpret field readings.

Currently, the Richardson-Carbone Lab has to use a cumbersome set of equipment to conduct their research. This method of data collection is slow, not very portable/durable, and it is not the easiest to view out in the field. The proposed solution will make the entire process fast, portable, and easy to use while also providing important meta data to the researchers in real time. This document will start by explaining the current problem that the Richardson-Carbone lab faces, before then briefly explaining our proposed solution. We will then break down the requirements for that solution into three categories and list them with brief explanations. Finally, we will close out the document with potential risks, our planned development cycle, and a conclusion.

2. Problem Statement

There are many problems with the current method of data collection. The research requires mobility and durability for the outdoors as data collection will occur at many different sites. Currently the process is slow and cumbersome. The stock gas analyzer's software can only be run on a Windows PC as there is no mobile application available that satisfies the requirements of their work. Bringing a laptop to the research site is slow. The laptop must be powered up, connected to the gas analyzer, and then the data collection software must be manually run. This has been unchanged for 15 years. That means that its user interface is outdated by current standards, its feature set is very limited, and it lacks many affordances provided by modern technology. Once the data is collected, the laptop must be shut down and taken to the next site, where this process is repeated. Also when getting data out of current software the naming convention on the files are meaningless and this makes it hard to find correct files. All of these obstacles together slow down the research collection process greatly.

The main problems with the current form of data collection are:

- Collection equipment is bulky and difficult to move
- Collection equipment has to be plugged in and unplugged for set up at each site
- There is no built in method of assigning metadata to readings
- The UI is very basic and shows only a series of changing numbers, with no visual aids or graphs
- There is no built in method of calculating statistics required for the research
- Getting data from the computer is not as easy or useful as it could be

3. Solution Vision

In order to solve these issues for our client, we are developing an application for mobile Android devices that can be used to measure and record data in conjunction with their LI-840A gas analyzer. The application's interface will be designed with a tablet in mind, as the large screen real estate allows for a higher level of detail. The application will use data visualization tools to render a series of graphs that displays the gas analyzer's reported data in real time. The software will also provide several means of statistical analysis so the user can generate the information as needed with the convenience of staying within a single application. Finally the application will be able to transfer data out via email with convenient naming conventions of the files. With all of the tools being readily available on a tablet device, any researcher out in the field will no longer be burdened with having to carry a laptop or any other peripheral devices in order to collect and analyze data.

Features our application will include:

- Data collection via USB-to-Serial connection
- Loading/saving/logging data
- Metadata inputs
- Visualization of data collected in real time
- Adjustable graph displays
- Statistical analysis on collected data sets
- Data set transferring via email

The data our application uses is collected from the gas analyzer itself. The LI-840A will be able to be communicate with the Android device itself by utilizing a direct usb-to-serial connection, and the data is transferred directly to it. The data is delivered in a simple, XML style format, with clear name labels for every measurement the analyzer provides. Originally, the project was envisioned as using a Raspberry Pi as a middleman between the LI-840A and the Android device. However, we decided that the ultimate goal of this project was to make a product that was as compact and mobile as possible, and omitting an additional data transfer point was an obvious decision. The application we are developing will replace the need for such a setup by providing a mobile alternative, while preserving all of the benefits offered by the originally used software.

The interface of the app will involve multiple windows, the primary one being the data logging screen. Using the GraphView library, we will be able to read in the data stream and graph it onto the screen in real time. It is a built-in functionality in GraphView to allow the graphs that are displayed in the app to be manipulated in size and visibility by the user using simple touch commands. This fulfills the requirements of both having real time data display as well as having that display be adjustable by the user.

Any subset data that is logged in the application will be immediately saved after the the user stops the logging. The user can access these data logs in a window accessible from the data logging window, choosing them from a list. In order to avoid visual clutter and confusion for the user, there will be metadata elements associated with every subset of data. The initial values for metadata include things such as a name, the location, an image, and GPS coordinates. This allows for the user to save all of the data they record with our app, and then be able to easily identify and access it later.

It should be noted that this software is not exclusive to just researchers aiming to shrink the size of their research kit. Any researcher that finds our application’s interface accessible or its statical analysis tools to be of use can employ our product in their work. It could potentially increase the productivity of countless researchers across the entire field.

4. Project requirements

We have divided the project requirements into three groups: functional requirements, performance requirements, and environmental requirements. In this section, we will explain each of these groups and describe what it means to be that type of requirement, as well as which requirements fall under it.

4.1. Functional Requirements

Functional requirements can be thought of as features. Here, we will list all the features that the software will offer. This section will not describe how the features will perform, or within which limitations the features have to operate, as these details will later be explained in the performance requirements and environmental requirements sections.

1. Data Streaming

   The software must be able to establish a connection via a direct usb-to-serial port, and read data from the LI-840A gas analyzer.

2. Data Visualization

   The software feature the ability to display the data collected from the gas analyzer in the form of a line graph in real time as it is reported from the LI-840A gas analyzer. This data includes measurements of CO_2, water vapor, and other variables. The graph must have the ability to display each of the values reported from the LI-840A one at a time, allowing the user to click which value they currently wish to see. The graph will also feature scaling to be able to pick inputs for the x and y values.

It will also have zooming ability to display the graph closer or further out through direct user manipulation.

3. Specifying Data Subsets

The software will feature the ability to specify subsets of collected data to show the parts of the data that the user wants to see. This will be done by allowing the user to choose when to start the data subset, and when to end that data subset while viewing the live graph readout.

4. Saving Data Subsets

The software must feature the ability to save subsets specified by the user onto the device after collection. These subsets must also have to ability to be loaded, and have their values and graphs reexamined.

5. Subset Statistical Analysis

The software must feature the ability to perform statistical analysis on collected subsets and have the calculated values displayed to the user after collection. These statistics will only be calculated for data subsets, and not for the live graph readout. The required statistics are slope, standard error, and R^2.

6. Subset Metadata

Each collected subset must have the option for user defined metadata. The required metadata fields are site name, plot number, operator name, description, time of collection (automatically filed in), temperature, location image, and GPS coordinates. All of these metadata fields will be represented using text, except for location image which will be represented using an picture taken on the device, if the device has a camera available.

7. Subset Emailing

   The software must be able to email collected subsets to a recipient specified by the user. The collected subsets will be compressed into a single file, and then be sent using integration with the device's default email client. The compressed file will contain the raw collected data and any metadata fields input by the user.

4.2. Performance Requirements

A performance requirement describes how certain features are expected to perform. This can relate to the speed at which a feature will operate, how many clicks it would take to access a certain feature from the home screen.

1. Real Time Data Analysis

   The data streaming will be done with a live readout of a graph and four number readouts. This is specific to functional requirements 1 and 2, Data Streaming and Data Visualization. The LI-840A has the ability to output data points twice per second. The requirement of our app will be to match the live readout speed and plot the graph in real time. This means that graph will plot in real time and the data readouts; water, carbon dioxide, temperature, and pressure, will also update every second. The numerical outputs of the readings will all be aligned with the graph. Depending on how fast the the data collection and transferring is in the application, there may be a timed buffer. The buffer could be between one and five seconds. This means that the data being displayed will be a few seconds behind "real time".

2. User Interaction

This performance requirements is specific to the functional requirements 3 and 4, Specifying Data Subsets and Saving Subsets. The process of data collection through the mobile application itself will require minimum effort or time expense from the user. The pathway to start a new subset will only take a matter of four actions of the user. Those steps would be, starting the app, starting a new subset, filling out the Metadata, and then starting the collection of data. This way, the process of starting a new subset is convenient while in the field.

3. Mobility

The chosen device for the application to run on is an 8 inch Android Tablet. As a performance requirement, the device must be suitable for field work. As stated in the Problem Statement, the field work can not be done on a laptop. The device must be light and portable. The size of the tablet was chosen because can be held with one hand, making it very convenient when moving to site to site and collecting data.

4. Availability of the Mobile Application

The mobile application will start running before it is connected to the gas analyzer. Once the gas analyzer is connected, there will be a screen displaying that connection was successful. Once connected, the data collection can begin. The app will need to be able to run for up to 20 minutes. The mobile application will have no issues for running this long. Connection of the gas analyzer is consistent, and the app will not need to be restarted.

4.3. Environmental Requirements

As described we need to make an application to make reading measurements in the field easy to use and portable. There is a lot of leeway on how we need to get this application to work but not on what we need to get it to do. The only given piece of equipment for this is the LI-840A gas analyzer. The environmental requirements are as follows:

1. Connection to the Gas Analyzer

   We need to have a connection between the gas analyzer and a device that can run an application on it to get the data from it. This connection needs to be a sturdy connection that will not easily disconnect when moving in the field.

2. Portable device to run application

   This device must:
   - Be Durable
   - Have a sizable screen to navigate and view easily
   - Have wifi to transfer data out of the application
   - Have a camera for metadata pictures
   - Have GPS for metadata

.

5. Potential Risks

As far as potential risks go in our project, there are not many that are concerning to our prospects. If any were to arise, we would be able to circumvent their effects, albeit with additional work and effort on our part and potential costs to our client.

The most obvious of the risks is the direct USB-to-Serial connection between the Android device and the LI-840A gas analyzer ultimately being non-functional or abhorrently suboptimal. One of the most paramount requirements of our solution is mobility, and having to omit the direct connection would mean including, at minimum, one additional device. Such an addition would lead to our solution being more cumbersome for our clients, and would serve to defeat the purpose of our solution. This scenario would manifest only if the direct connection should prove to be too slow to be practical in our client's' research or if the connection simply does not work with any Android tablet device.

In order to visualize the data for the user, we are implementing the Android GraphView Library, which allows for real-time graphing of data. It was indicated by our clients that real-time data evaluation is an integral part of their research, and our software failing to include it would be unacceptable. A risk involving this software would be a lack of compatibility with Android Studio and the Android tablet itself. If this were the case, then we could be forced to switch to a different data visualization API, potentially one that does not support real-time graph updates. If this situation were to arise, then we would be failing to meet one of the most essential aspects of the solution, that being real-time data presentation.

5.1 Risk Mitigation

In regards to the risk involving the direct connection between the Android device and the gas analyzer, we have already undergone multiple tests to show that there is little to no risk involved with this approach. We used a usb connection between the LI-840A and an Android smartphone running an app that displays live data input from a usb device. Using this setup, we achieved a live data transfer in several trials. There were not any issues present in the trials and the app ran without any performance issues. The tablet will be running on the Android OS and has the exact same usb port as the smartphone, meaning the connection to the analyzer will function properly. At this point, we have no reason to believe that the connection between the tablet and the LI-840A will not work.

If the device is unable to take in the data and also graph without losing performance speed, it will be very easy to work around this issue. The application could be simply be designed to be taking in all of the data as it comes in, and then displays it with a two-second buffer. This would still provide the user with a live readout of the data, and its visualization would not be compromised in any way. It should be noted that the application being designed is not going to be CPU-intensive, so there is very little chance that this contingency will need to utilized.

6. Project Plan

Our development process has begun and will end April 30th, 2018. This gives up a total of fifteen weeks to develop our software. The following gantt chart details a breakdown of which tasks we will be working on each week. Tasks were ordered based on their dependency to other features, as well as which tasks could be worked on in parallel. For example, live graph readout, metadata implementation, and statistical computation can all be worked on in parallel, but only after we have completed data streaming and formatting.

We will start our development with streaming data from the instrument into the application. This is important to do first so we will know what kind of data will be coming in. Next we will be implementing the basic application screens and navigation. This is a crucial second step, as it lays the foundation for where features will be located in the application, as well as creating an organized template of where our code will be placed. This will be an ongoing process. As the application progresses, more app screens will be created. Next we will move to parsing it into a format that can be easily read and manipulated. Once the data is prepared, we can work on many features in parallel, such as the live graph readout, statistical computations, and addition of metadata to readings. We have allotted three weeks to these tasks, as we expect it to be the bulk of our codebase. Following that, we start to implement features related to saving and loading sets of data as well as transferring it out of the application. Our development cycle ends with a two week span dedicated to testing and debugging the software, ensuring that our final product is completely functional and polished.

7. Conclusion

The goal of our software is to aid in the gathering of field data for ecological research at Northern Arizona University. The set of requirements in this documents were chosen in order to address the concerns our sponsors, with the key requirements being system portability, compatibility with existing gas analyzing hardware, and simple data visualization, and secondary requirements of intuitive navigation, the addition of metadata to readings, and the transfer of data sets out of the application. Through the specification of these requirements and a well defined development schedule, we believe we have minimized any potential risks as well as laid the groundwork for a smooth production cycle.