CSI 497

Senior Project Proposal

Evan Sutton

Fall 2024

Evan Sutton Senior Project Proposal CSI 497

Table of Contents

1.0 Project Information

1.1 Executive Summary

A problem within the fishing industry is the inability to share and access navigational data from different chartplotter brands. As a tournament angler, I understand and experience this issue every time I am on the water. Having access to all nav data on one graph is crucial to navigational safety and fishing efficiency. This project aims to resolve this issue by converting the data into different file types that can be accepted by all chartplotter brands. Upon completion of the project, I will have created a valuable resource for boaters that wish to share their navigational data among different chartplotter brands. This research will allow me to better understand the electronics that I rely on while on the water.

1.2 Goals, Objectives, and Assumptions

The goal of this project is to successfully reverse engineer GPS data using the available binary code. Depending on the brand of GPS, these waypoint files are saved as .GPX, .HWR, or .USR files. These are proprietary file types with no public information on the encoding-scheme. Once I have found the encoding scheme for each file type, I will create a program to convert the data from each file type.

Reverse engineering will be a new concept. I will make use of a hex editor to decode the binary data. For the program aspect of the project, I will use my knowledge from CSI-230 and CSI-330 as the basis for my code. This program will be very useful for people who need to convert their GPS files to a different file type.

I will have access to all three file types being researched. Under the assumption that each file type contains a distinct and identifiable pattern, I will be able to find the encoding-scheme used in each file. This pattern will apply to each file of the same file type. With this information, manipulation of data content will be possible and the files will be convertible.

I will make use of a hex editor called Hexinator. This program will allow me to quickly decode the binary data to speed up the process of reverse engineering. I will be able to identify patterns within the hex data easier. Once the programming process begins, I will use an online C++ compiler to create my code. To test my converted file types, I will make use of my electronic GPS graphs. By plugging the converted file type into the graph, I can tell if the data was converted successfully or if there is an issue with the code.

1.3 Success Criteria

The reverse engineering will be successful once I have found the encoding-scheme for each file type. I will be able to test my encoding-scheme by manually writing the binary code for a waypoint and checking it with the original waypoint binary data. The conversion of file types will be successful when I can share waypoints between each brand of GPS graphs. The graphs will show an error if there is a mistake in the conversion process.

The core functionality of the project is to convert between .GPX, .HWR, and .USR file types. During this process, all converted waypoints will be stored on an sd card to prevent any loss of data. The user will input one or several different waypoint files, and the system will return and save the files in .GPX, .HWR, and .USR form.

2.0 Execution

2.1.1 Project Timeline

Project Duration	Project Start: August 26, 2024 Project End: April 2025
Initiation Phase Start Date	August 28, 2024
Planning Phase Start Date	September 11, 2024
Execution Phase Start Date	October 2, 2024
Execution Phase	Reverse engineer .HWR files October 2, 2024 Reverse engineer .GPX files October 23, 2024 Reverse engineer .USR files November 13, 2024 Compare encoding-schemes December 4, 2024 Begin file conversion code January 1, 2025 Test file conversion code January 1, 2025
Completion Date	April 1, 2025

2.1.2 Activities In-Scope

1. Creation of sample waypoints to test on.
2. Reverse engineering of file types. Each file type will have its own deliverable for reporting the encoding scheme I have found.
3. Comparison of binary code between each file type.
4. Create a visual representation of the workflow for the program.
5. Write the code that converts the data types.

2.1.3 Activities Out-of-Scope

1. Save initial waypoints on an SD card
2. Organize waypoints by location, icon, or name.
3. Plot waypoints on google earth so the user can have access to the waypoints on their phone.

2.1.4 Project Deliverables

1. .HWR encoding scheme deliverable
2. .GPX encoding scheme deliverable
3. .USR encoding scheme deliverable
4. Elaboration on the similarities and differences between each file type encoding scheme
5. List of waypoint icons and their corresponding binary values
6. Flowchart to show the work flow of program
7. Program that converts .HWR ↔ .GPX
8. Program that converts .HWR ↔ .USR
9. Program that converts .USR ↔ .GPX
10. Final program that converts between all data types

3.0 Risks

The key risk to this project is the possibility of installing corrupt data into the GPS graph. If corrupt data is installed, the graph must be restored to factory default settings. This process wipes all information from the graph and causes the user to lose any unsaved data. Several tests will need to be run after the completion of the coding phase. Each converted waypoint will need to be compared to an original to ensure all data has been converted correctly.

4.0 Bibliography

Humminbird.Johnsonoutdoors.Com, humminbird.johnsonoutdoors.com/us. Accessed 29 Oct. 2024.

“Welcome to Hexinator.” *Hexinator*, hexinator.com/. Accessed 29 Oct. 2024.

“Online Compiler and Debugger for C/C++.” *GDB Online Debugger*, www.onlinegdb.com/online_c++_compiler. Accessed 29 Oct. 2024.