Software Overview

Year: 2020 **Semester:** Fall **Team:** 6 **Project:** Snow-weAR Goggles
Creation Date: September 7, 2020 **Last Modified:** September 10, 2020
Author: Mitchell Ciupak **Email**: ciupak@purdue.edu

Assignment Evaluation:

Item	Score (0-5)	Weight	Point s	Notes
Assignment-Specific Items				
Software Overview		x2		
Description of Algorithms		x2		
Description of Data Structures		x2		
Program Flowcharts		x3		
State Machine Diagrams		x3		
Writing-Specific Items				
Spelling and Grammar		x2		
Formatting and Citations		x1		
Figures and Graphs		x2		
Technical Writing Style		x3		
Total Score				

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:
Relevant overall comments about the paper will be included here

1.0 Software Overview

The Sown-weAR Goggle device requires four software components on the main microcontroller that interface with the firmware of five exterior components (Global Positioning System, Inertial Measurement Unit, TOLED Display, ADC, and radio transceiver). On startup, the power monitor begins and then the main/home component powers on the TOLED Display, calls to start the radio component, and awaits users input to start the session component.

The radio component is responsible for transmitting and receiving GPS data asynchronously from other components. The radio component waits for user input to request the coordinates of other users with radio capability activated. The radio component also can receive a request from another headset to send coordinates. The radio component can be shut off via interrupt by the battery monitor or manually by the user.

The session component is called by user input in the main or home state. Once this session is started, the component initializes the IMU and GPS if it is not already on. The session component updates the Display and records data. When a user marks a session over, the user will receive a summary of their session and all measurements will be turned off except for the radio and GPS.

When the device is powered off properly the system will turn off devices, clear unnecessary data and configuration, and power off. If power is cut or the battery dies without the battery monitor component anticipating it, the system will make checks when the device powers on again.

2.0 Description of Algorithms

The first algorithm needed is one to interpret battery statistics to monitor the state of the battery. In a low-power mode, this algorithm will run an interrupt to shut down the radio transceiver. In an extremely low-power mode, the battery monitor component will begin to shut down all aspects of the device and run garbage collection.

During a session, an Inertial Measurement Unit (IMU) and GPS interpretation algorithm is needed to parse the data returned by the IMU into practical metrics: degree of incline, altitude, and speed. Once these calculations are complete, the algorithm will call the Display class to update the Display accordingly.

The next set of algorithms work together under the same class. The “request location through radio” is an algorithm that is triggered by user input and interfaces with the radio transceiver to send a signal to other users requesting their GPS data. Then, the “send location through radio” algorithm starts up the transceiver and sends its location via radio.

The last set of algorithms work under the same class as well. The start session algorithm is tasked with initializing the IMU, running the Display, and updating the Display through the IMU interpretation algorithm mentioned earlier. The end session algorithm is triggered by user input while in the start session algorithm and is tasked with shutting the IMU down, calculating averages on the data, and creating a session report summary. Each of these algorithms play their role in delivering the functionality of the Snow-weAR Goggles.

3.0 Description of Data Structures

Our software will make use of arrays, linked lists and variables that are subject to members of classes. A separate class will be instantiated for each device, the IMU, GPS, radio, ADC, and Display. Data structures to store and interpret information from each device will be implemented and include relevant data and a timestamp for each metric. A buffer will monitor data relevance and continuously delete unnecessary information to free RAM for data processing and display.

Appendix 2: State Machine Diagrams

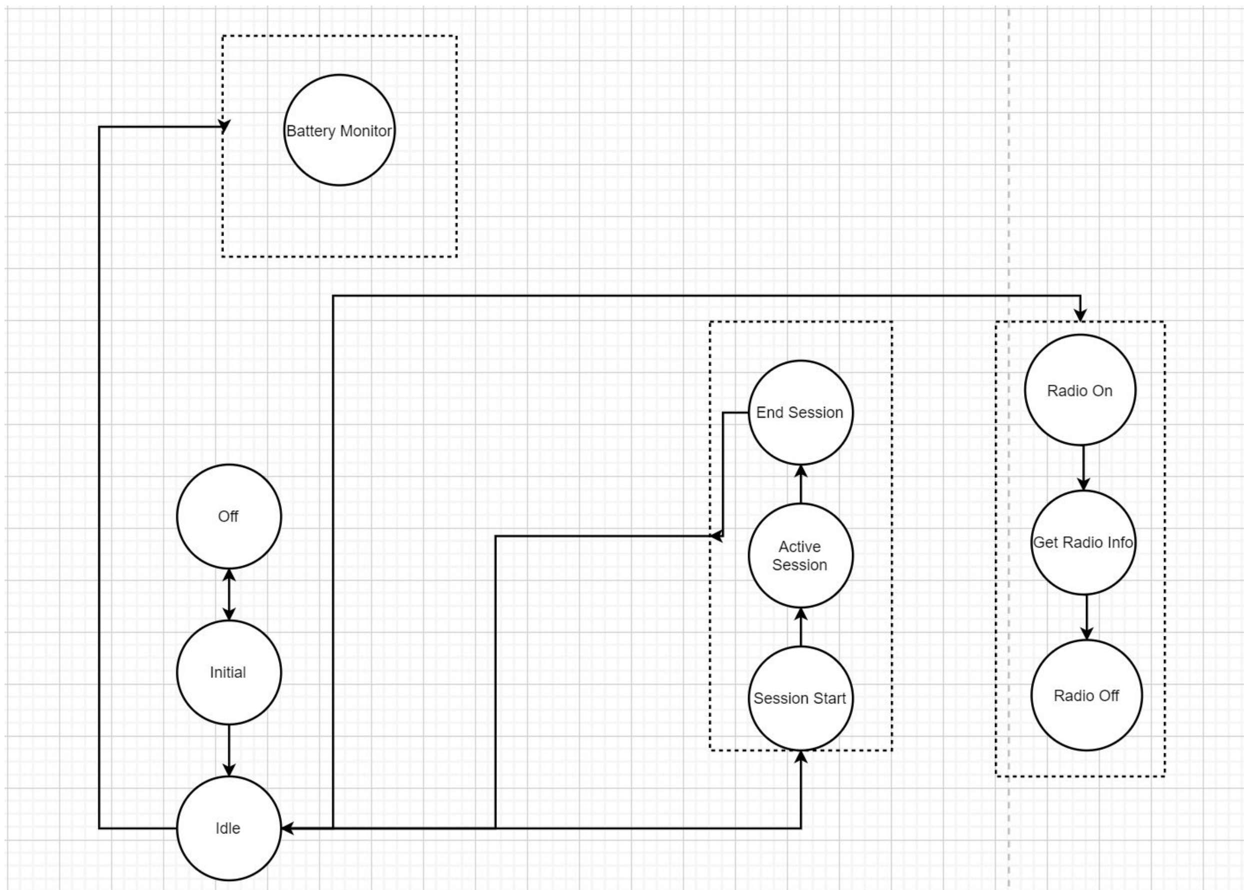

Figure 1 - Hierarchical State Machine Diagram