Software Overview

Year: 2018 **Semester:** Fall **Team:** 6 **Project:** Garbage Collecting Boat
Creation Date: September 03, 2018 **Last Modified:** September 07, 2018

Member 1: Peiyuan Shen **Email:** shen290@purdue.edu
Member 2: Tianjie Jia **Email:** tjia@purdue.edu
Member 3: Qianli Ma **Email:** ma160@purdue.edu
Member 4: Shuang Liu **Email:** liu1579@purdue.edu

Assignment Evaluation:

Item	Score (0-5)	Weight	Points	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 Garbage Collecting Boat consists of one firmware component and one software component. The boat itself is the firmware component. The boat controls two Electronic Speed Controls (ESC), a battery group, several Light Dependent Resistors (LDR), a GPS module and a Wi-Fi module. The microcontroller controls the boat to move, accelerate, brake, receive commands from the user interface and send back information through Wi-Fi communication.

Our major software component is a program that provides a GUI to interact with users. The program can calculate the path of the boat and send all the corresponding commands to the microcontroller on the boat via Wi-Fi module. The commands that the program sends to the microcontroller include the power level of both motors and the operation status of the conveyor belt. The program receives a series of information from the boat, including the battery voltage of each serial of lithium-based battery group, the x and y coordinate of GPS, the x and y degree of the compass, and a filled signal that indicates the storage room is full. In manual control mode, the user presses the direction keys on the user interface, and then the program converts that direction signal into the corresponding rotation speed of two driving motors and transfers them to the microcontroller. In the fix-route mode, the user sets one or more coordinates on the map. Based on the current location and direction of the boat, the computer will go through algorithms introducing below and send the rotation speed of both motors to the microcontroller. When the "storage room is full" signal is received, or the voltage of the battery group is low, the program will go to fix-route mode, turn off the belt, and set the destination coordinates to the initial coordinates the boat recorded.

2.0 Description of Algorithms

1. PWM for controlling ESC to adjust the speed of the motor.
2. Adjusting the heading of the boat and calculate new headings for the target location.

When the user set coordinates of a location and switch to auto mode, the interface program will compute the rotation speed of the two motors to drive the boat to the location. Referring to the Figure 1 below, by measuring the coordinates of the current location and the destination, Angle 2 can be calculated about the north. After receiving the x and y value from the compass, the degree of Angle 1 can be calculated in reference to the north[1], which in this case is a negative degree. Total angle between the current direction and the direction the boat should move is Angle 2 minus Angle 1. If it is a positive value as it is shown below, then the boat should turn right; otherwise, it turns left. Increasing the speed on the left motor to turn the boat right, and the increment is proportional to the absolute value of the total angle because a larger angle means a more deviation from the right track. Every time the program receives the updated GPS and compass signal, it performs the algorithm above and compute the corresponding rotation speed of both motor and send them back to the microcontroller. Thus, the speed of the motor is adjusting from time to time, and the final path of the boat could be like the red line shown below.

The data on the power level of the motors are represented by two duty cycle numbers. The speed of the motors is controlled by changing the duty cycle of the PWM sent from the microcontroller to the ESC module, and the number containing the duty cycle is sent from the user interface. First, the digital data regarding the needed adjusted speed are calculated by the algorithm described above. Once the data is calculated, the user-interface will send these data to the microcontroller on the boat; within the microcontroller, the digital data will be stored in one memory address and then transferred into the Electrical Speed Control module utilizing the TIM peripherals to change the duty of the PWM waveform [2]. There are two memory addresses designed to store the digital data for two motors; and every time there is an update on one of the two motor-speed-control data, the previous data in the memory address gets discarded, and a new one will be stored.

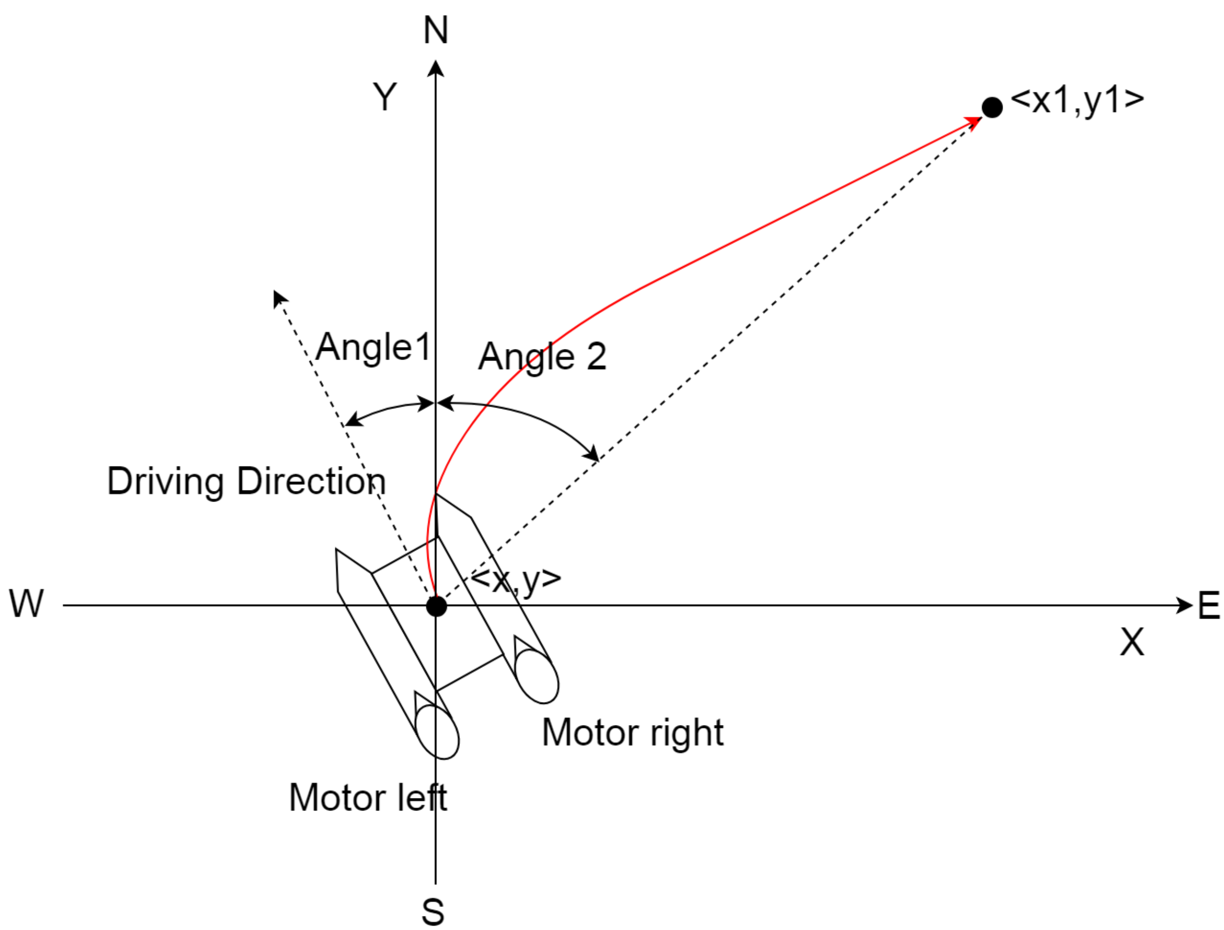


Figure 1. The algorithm of computing the path

3.0 Description of Data Structures

The essential data transmission in our project is the communication between the user interface and the microcontroller.

From the user interface to the microcontroller, three signals will be transferred in serial. The motor power level or rotation of the speed is represented as a duty cycle number. A byte is enough to contain a duty cycle number which typically ranged from 0 to 100. The Conveyor Belt

On/Off signal is a single bit signal, but since the data are transmitted through Wi-Fi byte by byte [3], the first seven bits in Byte3 is reserved for convenience.

On the microcontroller side, the battery voltage is divided first and then monitored through ADC[4] and scaled to one byte. There are four groups of battery, and each can be represented in one byte to monitor the charge left. GPS coordinates and Compass degree are both float number from the device. The filling situation is also a single digit value: 0 for not full, and 1 for full. It is transferred in a byte for convenience. All of those data above are transferred in serial as the image presented below.

From User Interface to the Boat

Byte 1	Byte 2	Byte 3
Right Motor Power Level	Left Motor Power Level	Conveyor Belt On/Off

Figure 2. Data Transmission from the Computer to the Boat

From Boat to the User Interface

Byte 1-4				Byte 5-8	Byte 9-12	Byte 13-16	Byte 17-20	Byte 21-24	Byte 25
BV 1	BV 2	BV 3	BV 4	GPS x coordinate	GPS y coordinate	GPS speed	Compass x degree	Compass y degree	full

BV: Battery Voltage full: storage room full

Figure 3. Data Transmission from the Boat to the Computer

4.0 Sources Cited:

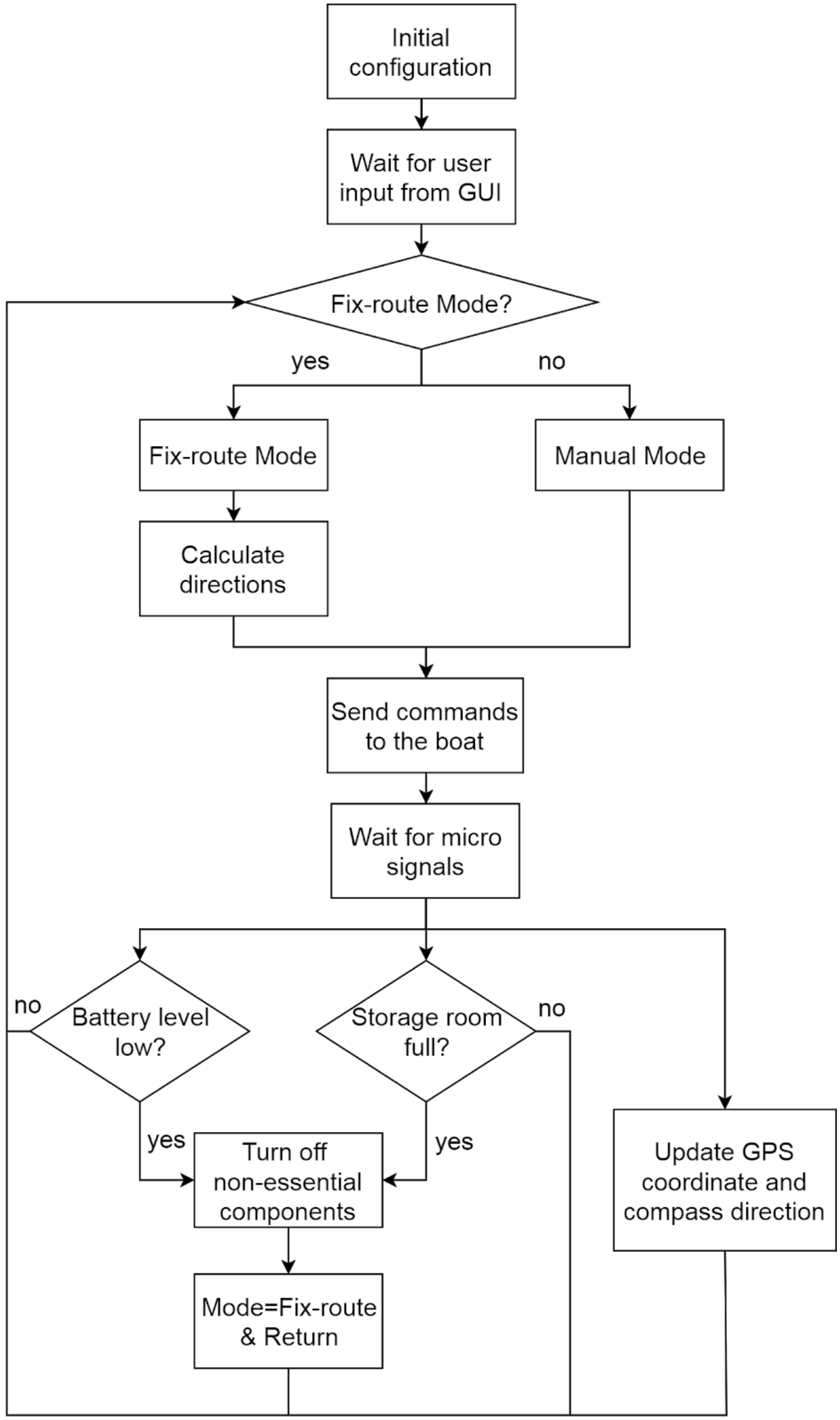
[1] Honeywell. *Compass Heading Using Magnetometers.* [Online]. Available:https://www.st.com/resource/en/datasheet/stm32l496ag.pdf. [Accessed: 06-Sep-2018]

[2] STMicroelectronics. (2018) *STM32F4 Reference Manual.* [Online]. Available:https://www.st.com/content/ccc/resource/technical/document/reference_manual/group0/81/ea/88/1f/97/9e/4a/d0/DM00305666/files/DM00305666.pdf/jcr:content/translations/en.DM00305666.pdf. [Accessed: 06-Sep-2018]

[3] NURDs. *ESP8266.* [Online]. Available:https://nurdspace.nl/ESP8266. [Accessed: 06-Sep-2018]

[4] RICHTEK. (2014) *Battery Discharge Curve.* [Online]. Available:https://www.richtek.com/battery-management/en/designing-liion.html. [Accessed: 06-Sep-2018]

Appendix 1: Program Flowcharts



Appendix 2: State Machine Diagrams