Autonomous Aquatic Drone

Joe Canfield and Mitchell Overdick

1) Summary

The proposed project is to construct a small solar powered paddle boat capable of navigating itself using GPS waypoints. The boat will be versatile, designed to be flipped 180 degrees in any direction and still function as intended. This involves applying solar panels to both the top and bottom of the boat to ensure the onboard systems have power at all time. In order to implement these specifications, hardware such as GPS, accelerometers, magnetometers, and batteries will be required. Tests will be conducted on small bodies of water such as swimming pools or small lakes like Lake Padden. The end goal will be for the boat to travel Lake Whatcom, launching from Bloedel Donovan Park, circumventing Reveille Island, and finishing in South Bay. Success in navigating Lake Whatcom will demonstrate the capability of a small watercraft to travel a great distance without direct human control.

2) Project Requirements

- The drone must be able to float in freshwater and saltwater, as well as be water tight so it will continue to float for extended periods of time and protect the components inside from water damage.
- The drone must also have sufficient propulsion to overcome average ocean currents (~0.2 m/s) and wind speeds (~10 m/s). The drag due to wind will be assumed to be approximately equivalent to the drag due to ocean current, so a total of 0.4 m/s must be overcome. We want our minimum speed across water to be 0.2 m/s, therefore the drone must be able to travel at 0.6 m/s with no wind or water current.
- The drone must be able to communicate to a satellite through GPS to determine its position on earth.
- The drone must accurately follow the given GPS waypoints to a precision within 10 m. This is a based on a conservative estimate of how accurate our GPS will be.
- The drone must be able to detect drift from current and wind and correct accordingly.
- The drone must communicate wirelessly with the user to transfer the GPS waypoints to and from a host computer. Data rate is not of much importance, since data will only be transferred when the drone is not operating in the water.
- The waypoints shall be transferred to and from the drone as double precision floating point numbers.
- The drone must have enough non-volatile memory to store 65,000 waypoints.
- The unit must have a rechargeable battery with enough capacity to power the drone for 5 hours with no light.
- The operating temperature of the drone must span the temperatures that it would experience. We will consider this range to be between -2 ℃, the freezing temperature of ocean water, and 40 ℃, a high estimate for the temperatures the drone might experience from ocean water and sunlight.
- The size of the drone will be less than 500mm in length and height, restricting our solar panels to be less than 300mm in length and height.
- The drone must have an external method of switching between operating modes. These modes will be off, programming mode, and tracking mode.

3) System Design Formulation and Specifications

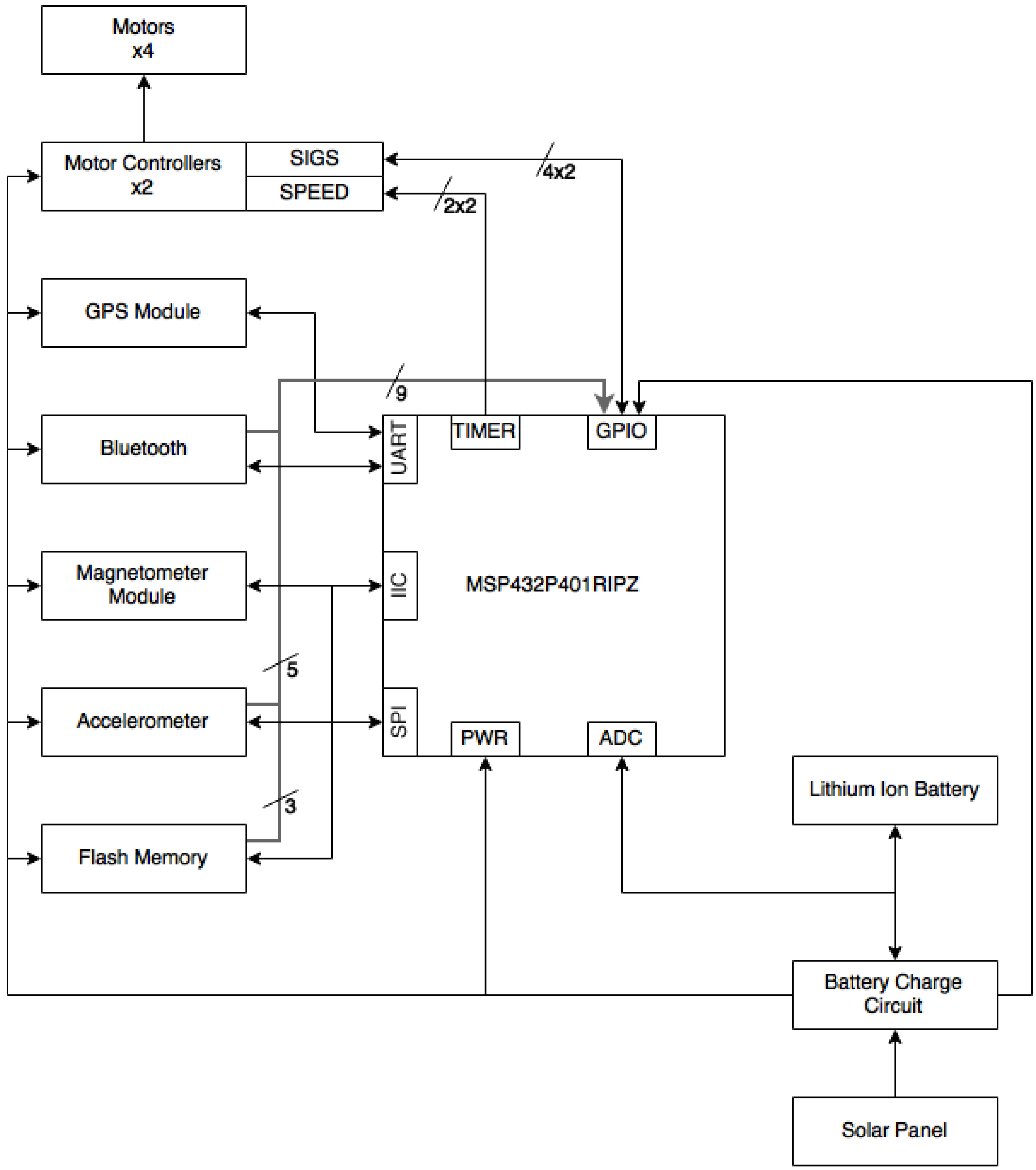

Figure 1: *High Level Block Diagram*

A) MCU, Resources, and Memory.

The ideal microcontroller that will be used for our project is the Texas Instruments MSP432. Since our project will be battery powered, we need an MCU that is capable of entering low power modes while still allowing us to make some significant computations. The processor used by the microcontroller is an ARM® Cortex®M4F core up to 48 MHz.

The specific microcontroller we will be using is the MSP432P401RIPZ. This is an ideal package for us, because it is the only package of the MSP432 that Low Profile Quad Flat (LQFP) package. We are limited to an LQFP package due to our limitations with PCB constructions and our inability to reliably place Ball Grid Array packages. The drawback to this is a larger package size 14mm x 14mm rather than that of the MSP432P401RIZXH which is only 5mm x 5mm and is slightly more ideal. However, with the larger footprint, we have access to the complete capability of the MSP432 chip, allowing us to potentially expand the amount of inputs into the microcontroller. Additional inputs could be ADC inputs that monitor current consumption of each motor, solar panel supply voltage/current among other features that could help make our craft much smarter. The MSP432 has two separate serial communication blocks called Enhanced Universal Serial Communication Interfaces. The first block is capable of up to four eUSCI_A modules which includes UART, IrDA, and SPI. The second block is capable of up to four eUSCI_B modules, which includes I^2C and SPI. The MSP432 also has 48 GPIO pins, 4 Timers capable of PWM, and 24 ADC pins.

Each motor controller needs 2 timer pins to control PWMs for each motor. They also require 4 GPIO pins for controlling motor direction, sleep mode, and fault detection. This means to control all four motors, we will need 4 timer outputs and 8 GPIO outputs. Since the MSP432 has 4 16-bit timers with PWM capability, this perfectly meets our requirements. We will also have 40 leftover GPIO pins after connecting the motor controllers to use for the other devices on our PCB.

The GPS module will only be interfacing with the MCU via UART. We may be including two GPS modules on our boat depending on how much the antenna orientation affects the quality of the GPS signal. The tests have not been conducted yet, but regardless of the results we will have plenty of hardware resources to handle adding a second GPS module oriented in the opposite direction of the other. Assuming worst case, having two GPS modules communicating over GPS, we will need 2 UARTs, bringing our total available UARTS down to 2.

The bluetooth module communicates with the MCU via UART and will require 4 GPIO pins for hardware handshaking features, reset, and active signals. This will bring our available UARTs down to 1, however we will need no more than this. Using the 4 GPIO pins, this will bring our total available GPIOs down to 36.

The magnetometer module communicates with the MCU via I^2C, which will require using 1 of eUSCI_B's I^2C modules, which will be shared with the accelerometer. This is the only interface the magnetometer has with the microcontroller.

The accelerometer communicates with the MCU via I^2C, and will notify the MCU of orientation changes by triggering interrupts on the GPIO. There will be two GPIO connections for triggering 2 different interrupts. This brings our count of available GPIO pins down to 34.

The flash memory communicates with the MCU via SPI, and requires 3 GPIO pins for special control functions. SPI can be implemented on eUSCI_A or eUSCI_B as there is enough remaining resources on both interfaces.

The battery charger circuit has a fault detection feature that will output high or low on a pin in the event of a fault. This feature will require 1 GPIO pin, bringing our available GPIO pins down to 30.

We will also be implementing an ADC to monitor our battery and if the battery voltage drops below a certain value, we can shut our MCU and peripherals down safely before loss of power. This will leave us with 23 extra ADC pins that will allow us to implement more analog monitoring features later on.

The MSP432 has 256KB of flash rom and 64KB of SRAM, which will sufficiently hold our real time kernel and support our tasks/modules ram and memory consumption. We also require a waypoint storage system for storing the proposed route along with a periodic log of the crafts whereabouts that must hold a minimum of 65,000 points. A simple calculation of the space required for this was done by making a binary file with 70,000 waypoints stored in it which resulted in a file greater than 4MB, since we want to be able to store a route and the log files, we calculated our required external flash to be 8MB, this allows us to store both our binary files, with some extra space in case we choose to add more information to each point in our log file such as a timestamp.

3.1a Power Specifications

The solar panel and battery will be used to power the entire system on the boat, which includes the MCU, motors, and all other peripherals. A charger circuit will manage the solar panel charging and discharging of the battery. The operating voltage range of the MSP432 is 1.62V to 3.7V, we will be designing our system to run on 3.3V. The solar panel will output 20V into our battery charging IC, which will regulate that to charge the lithium battery to 3.7V. The battery supply will then pass through a buck converter to regulate the the supply to 3.3V. The battery charging IC has a maximum charge current of 2A, which will give us a maximum power supply of 7.4W. This constraints our circuit components to have a maximum current draw of 2.24A.

B) Description of Software Requirements

The entire system for this project will be implemented using C. We plan to use μC/OS-III as our real time operating system. We chose μC/OS because it allows our platform to expand easily if we need to add more tasks. Another advantage of μC/OS is all of the configurations for porting it are free and available for download online. Aside from driver modules for all hardware peripherals, the two main tasks will be navigation and data acquisition. The navigation task will incorporate major vector algorithms for calculating trajectory and drive power.

4) Development Plan

A) Prioritized List of Features

1. Watertight buoyant chassis
2. Paddle wheel propulsion
3. Solar powered battery charging circuit
4. Waypoint and log file storage system and flash memory
5. Wireless data transfer
6. External power switch
7. GPS for detecting latitude and longitude
8. Heading detection using magnetometer as compass
9. Accelerometer flip detection
10. Fault tolerance
11. Periodic position logging

Our features will be completed in the order that they appear on the prioritized list, with some concurrent development depending on the scale of the task. The watertight chassis is our first task because the rest of the hardware design depends on the dimensions and form factor of the chassis. Once the chassis is completed the propulsion system and battery charging system can be implemented, which allows us to power the boat, the first step to making the boat autonomous.

The next step will be to implement the waypoint storage system and flash memory, and subsequently wireless data transfer.

Once we can wirelessly transmit data to and from the drone, we will also be able to seal the components inside, which requires us to implement the external power switch. The external power switch will be composed of two magnetic relay switches which can be activated from the outside of the drone, but do not require breaching the hull. The user will control the switches by placing small permanent magnets on the hull where the relays are mounted. The two switches will control the three different modes of operation described in section 5.

The last steps in completing the required features will be the GPS and magnetometer. The GPS and magnetometer allow the drone to know where it is and calculate its current heading and required heading to the next waypoint. Once this is completed the drone should be able to complete a voyage as described by the project summary. The next steps will add troubleshooting capability and robustness to the design, though are not critical to the function of the drone.

We will add the functionality of the accelerometer so that the drone will detect when it has been flipped and can reverse the direction of the motors and switch which solar panel is active. Fault tolerance is a feature that will detect when the drone is about to lose power, and safely power down. Periodic position logging will allow us to analyze the path of the drone on a voyage and compare that with the path designated by the waypoints.

B) Task Allocation

Mitchell will be in charge of the flash memory and bluetooth modules, because those two modules are closely related. If time allows, Mitchell will also make the accelerometer module. Joseph will be in charge of the motor controller circuit, as well as the GPS and magnetometer modules, because the GPS and magnetometer are very closely related. The battery charging circuit and navigation algorithm will be worked on cooperatively because those modules are critical to the operation of the drone and may require more attention.

Task	Mitchell Overdick	Joseph Canfield
Motor Controller Circuit		X
Bluetooth Module	X	
GPS Module		X
Magnetometer Module		X
Accelerometer Module	X	
Battery Charging Circuit	X	X
Flash Memory	X	
Navigation Algorithm	X	X

Schedule

Below is a tentative development schedule for the next two quarters, we have attempted to be conservative in our estimates, including "Catch Up" weeks that can be used to work on whichever aspect of the project we are falling behind on.

Week of (Starting Mon)	Task
Jan 2 (Winter quarter starts Wed Jan 4)	Microcontroller initialization
Jan 9	Solar Panel Testing/Charging Circuit Design
Jan 16	Module Design (Motor Control/Bluetooth)
Jan 23	Module Design (GPS/Flash)
Jan 30	Module Design (Magnetometer/Accelerometer)
Feb 6	Hardware Design Review
Feb 13	Module Testing (Motor Control/Bluetooth)
Feb 20	Module Testing (GPS/Flash)

Feb 27	Module Testing (Magnetometer/Accelerometer)
Mar 6	PCB/Chassis Design
Mar 13	PCB/Chassis Design
Mar 20 (Spring Break)	Catch Up Time
Mar 27 (Spring quarter starts Tues Mar 28)	Catch Up
Apr 3	Motor Controller Code (PCB Assembled and Tested)
Apr 10	Flash Memory Code (Software System Design Reviews Start)
Apr 17	Bluetooth Code
Apr 24	GPS/Magnetometer Code
May 1	Navigation/Data Logging Code
May 8	Accelerometer Code
May 15	Catch Up Time/Low Power Optimization
May 22	Catch Up Time/Low Power Optimization (Code Reviews)
May 29	Final Construction/Field Testing
June 5 (Finals Week)	Final Construction/Demonstration

C) Required Development Tools

To complete this project we will require specific tools to develop our software and hardware. A full list of our development tools is below:

- MSP432 LaunchPad development board - Purchased
- Code Composer Studio v6 - Provided
- Altium - Provided
- 3D Printer - Provided
- Mixed Domain Oscilloscope - Provided
- Power Supply - Provided
- Computers - Provided

D) Description of Prototype

The prototype for demonstration will be a minimalistic watercraft capable of being deployed, but perhaps only to limited scope. Due to the constraints of our development time, our chassis will be most likely composed of 3D printed components, using the main PCB for structure.

5) User Interface Requirements

There will be two user interfaces with this project; one on the craft, and one on a host PC. The interface on the craft will be two magnet-activated switches, one being a power switch and the other being a programming switch. The switches will be normally closed and the placement of a magnet on them will open them. Having no magnets required on the craft for tracking mode prevents the risk of them falling off and deactivating the craft. A special mounting surface will be constructed to hold the magnets in place over their respective sensors. They will follow the following order, where a “1” denotes a magnet on, a “0” denotes a magnet off, and an “X” denotes don't care.

Power Switch	**Programming Switch**	**Mode**
1	X	OFF
0	1	Programming Mode
0	0	Tracking Mode

When the craft is in programming mode, the bluetooth module will be enabled and discoverable by a host PC, all other modules will be disabled. When the host PC connects, the craft will immediately try to transmit a log file if one exists, if not no transfer to the host PC will commence. The host PC will then be able to transmit a specially formatted binary file to the craft which holds all the waypoints for the next voyage. Once the transfer is complete, the files will be stored in nonvolatile memory and the craft can be powered down.

Two simple GNU Octave programs will be written for formatting the waypoints. One will take a Google Earth .kml file as input and output a binary .dat file that can be uploaded to the craft. The second will take the log file sent from the craft and convert it into a .CSV and .KML file for viewing on Google Earth and processing the data at each log point.

When the craft is in tracking mode, the bluetooth module will be disabled and the craft will begin to seek GPS waypoints. It will continue to seek waypoints until it reaches the final one where it will idle there waiting to be picked up.

Appendix A - Preliminary Parts List

The following is a list of the preliminary parts we will be using in our design. The parts shown are the major components of our design, and passive parts have ignored.

Part Number	Description	Cost	Lead Time	Power
MSP432P401RIPZ	MCU	$8.00	immediate	3.8mA
MMA8653FC	Accelerometer	$1.00	immediate	26µA
GPS-HMC5883L	GPS/Magnetometer	$13.00	immediate	67.1mA
NRF8001-R2Q32-T	Bluetooth	$4.25	immediate	14.6mA
W25Q64FVSSIG	Flash (8MB)	$1.10	immediate	25mA
LT3652EDD#PBF	Solar Charger IC	$7.00	immediate	85µA
DRV8834RGER	Motor Driver	$3.00	immediate	1mA
RF500TB	Motor	$5.00	immediate	85mA
N/A	Solar Panels	$6.00	immediate	N/A
Turnigy 1S 20C 2200mAh Lipoly	Battery	$5.00	immediate	N/A
PAM2305AAB330	Buck Regulator	$0.58	immediate	90% min eff. multiply total current draw by 1.11
N/A*	Crystal Oscillators	~$0.65	immediate	Negligible
N/A*	Resistors	~$0.15	immediate	~200µA
N/A*	Capacitors	~$0.20	immediate	Negligible
N/A*	Inductors	~$0.15	immediate	Negligible
N/A*	Schottky Diodes	~$0.75	immediate	Negligible
NCP15XH103J03RC	Thermistor	$0.11	immediate	Negligible

*Part numbers not shown, price and power consumption general conservative estimate.

Appendix B - Cost Breakdown

Part	**Cost**	**Quantity**	**Total**
MCU	$8.00	1	$8.00
Accelerometer	$1.00	1	$1.00
GPS/Magnetometer	$13.00	2	$26.00
Bluetooth	$4.25	1	$4.25
Flash (8MB)	$1.00	1	$1.00
Solar Charger IC	$7.00	1	$7.00
Motor Driver	$3.00	2	$6.00
Motor	$5.00	4 (2 at a time)	$20.00
Solar Panel	$6.00	4	$24.00
Battery	$5.00	1	$5.00
Buck Regulator	$0.58	1	$0.58
Crystal Oscillator	~$0.65	1	$0.65
Resistors	~$0.15	16	$2.40
Capacitors	~$0.20	29	$5.80
Inductors	~$0.15	7	$1.05
Schottky Diodes	~$0.75	4	$3.00
Thermistor	$0.11	1	$0.11
Sum			$115.84

In addition to this price, our PCB, which we are using for the chassis, will cost around $17, making our total cost around $140.

Appendix C - Power Budget Calculations

Our power budget is based on the current consumption per component and how many of each component are needed.

Part	**Power**	**Quantity**	**Total**
MCU	3.8mA	1	3.8mA
Accelerometer	26µA	1	26µA
GPS/Magnetometer	67.1mA	2	134.2mA
Bluetooth	14.6mA	1	14.6mA
Flash (8MB)	25mA	1	25mA
Solar Charger IC	85µA	1	85µA
Motor Driver	1mA	2	2mA
Motor	85mA	4 (2 at a time)	170mA
Resistors		16	200µA
Buck Regulator	90% efficient	1	Factor of 1.11
Sum			388.4mA