Node-Based Range Extending Recon Drone

ECE 445 - Design Document

Team 27
Michael Lally - mlally2
Thomas Korenchan - kornchn2
Dhruv Diddi - diddi2

October 15, 2018

Contents

1. Introduction

1.1 Objective

The utility of drones in reconnaissance roles is now a large and expanding field. Drones are currently used to assess the spread of both urban and forest fires. They have also been used in search and rescue roles in a variety of scenarios; they can fly to and deliver floatation devices to struggling swimmers, locate lost or distressed hikers, and have been tested as a means of locating survivors in building collapses and fires.

However, their utility is still largely limited to exterior action, either operating above the canopy of the forest or in the air around a burning building, as examples. Their ability to operate is limited by where their command signals can reach, making control in locations like building interiors, forests floors, sewers or tunnels, or anywhere dense with obstructions unreliable.

Our objective is to address this issue with a node-based range-extending solution. The principle behind our solution is that as a drone reaches the outer limits of its command space, either due to range limits or interfering obstructions, it can deploy a node. That node, still within the command space of the operator, can pick up commands and retransmit them. This strategy has several benefits. If a node is placed just past a corner or obstruction, or at a nexus of corridors, the drone can continue to operate around that corner, or deeper in those corridors, further than it would be able to while receiving commands from the original operator’s signal. If a node is placed near the outer limit of the operator’s signal, the operator’s linear range can be effectively increased. These benefits can all be achieved on-the-fly, without the need to set up a network.

Our vehicle will take the form of 3 identical segments linked together, with each segment acting as a node. Our vehicle nodes will be capable of receiving and re-transmitting command instructions. The lead segment will be able to jettison following segments while moving. These segments will serve as middlemen, routing instructions to the vehicle and allowing the operator to continue controlling the vehicle as it enters an otherwise unreachable region. Our project ultimately expands on drone control methodologies and allows for operation in previously inoperable environments.

1.2 Background

Our project is not the first foray into drones as a means of reconnaissance and rescue. In the days following the September 11, 2001 attacks in New York City, researchers from the University of South Florida, funded by the National Science Foundation (NSF), were on - site at ground zero using prototype robotic vehicles to search the rubble for survivors and remains. They were limited in their ability, however, as their vehicles were tethered to their operators with a maximum range of 100 feet. We hope to improve on this maximum range with vehicles that can deploy nodes at critical junctions and nexuses in their search space [1]. We aim to forgo the need for a tether to navigate commands from an operator to a vehicle, given that a convoluted or constrained path lies between the two.

It's also not the first exploration into the use of drones as a mesh network to extend connection. In a recent paper from University of Waterloo, researchers discussed the utility of drones as a means of extending vehicular networks, networks that enable self-driving cars to communicate, share information on themselves and their movements, and avoid collisions. Their paper touches on "Line-of-Sight Links", the principle that "drones flying in the sky have a higher probability to connect ground nodes” [2], specifically cars, due to their ability to establish line-of-sight connections over ground-based obstacles like buildings, trees, and other vehicles. Our project aims to expand on and leverage this same principle: by deploying our nodes at locations that provide wide coverage over previously obstructed regions, we can achieve the same effectiveness as a line-of-sight link. Even though there is no direct line-of-sight between the operator and the lead segment, a daisy-chain of retransmitting nodes can establish a comparably effective connection.

1.3 High-Level Requirements

- The lead vehicle segment of our solution will be able to reach an unobstructed range of 500 feet (outdoors) from the controller.
- Each segment of our solution (controller, node segment and lead segment) will maintain a network bandwidth of 64 kbps, which is the minimum data rate required to send 802.15.4 messages.
- Our solution will maintain a minimum operation time of 30 minutes. This requires the network to remain uninterrupted, and each segment to remain functional (does not run out of power or disconnect from network). This is based on the information that average drone operation time is 20-30 minutes [3].

2. Design

2.1 Block Diagrams

Our solution is divided into two main designs, the vehicle (Fig. 1) and the controller (Fig. 2).

The vehicle design is applied to 2 physical vehicles that we will henceforth identify as either:

1) ‘node segment’ (vehicle which communicates directly with the physical controller)
2) ‘lead segment’ (vehicle in the front which communicates with the node segment)

Each segment is further divided into more components: the power supply consists of an alkaline and lithium 9V batteries connected directly to voltage regulators on the PCB to provide constant voltage ratings to the other components of the design, and will allow our vehicles to operate long enough to satisfy the operational time requirement of 30 minutes. The USB-to-UART interface will allow our software program - which governs our data transmission and vehicle movement - to be loaded into the microcontroller flash memory. The microcontroller itself interfaces directly with the wheel and latch mechanism motors to provide control for movement and node-to-lead segment attachment/detachment. It also interfaces with the XBee radio module to receive data from either the physical controller or other vehicle segment.

The controller design is almost identical to the vehicle design, except it does not include a 9V lithium battery, wheel and latch motors, as well as their respective voltage regulators. Also unlike the vehicle design, it includes physical buttons and switches which allow the user to send motion or latch attachment/detachment commands through the XBee module on the controller. It also includes LEDs that signify which segment is currently being controlled, and if the vehicle has disconnected from the network.

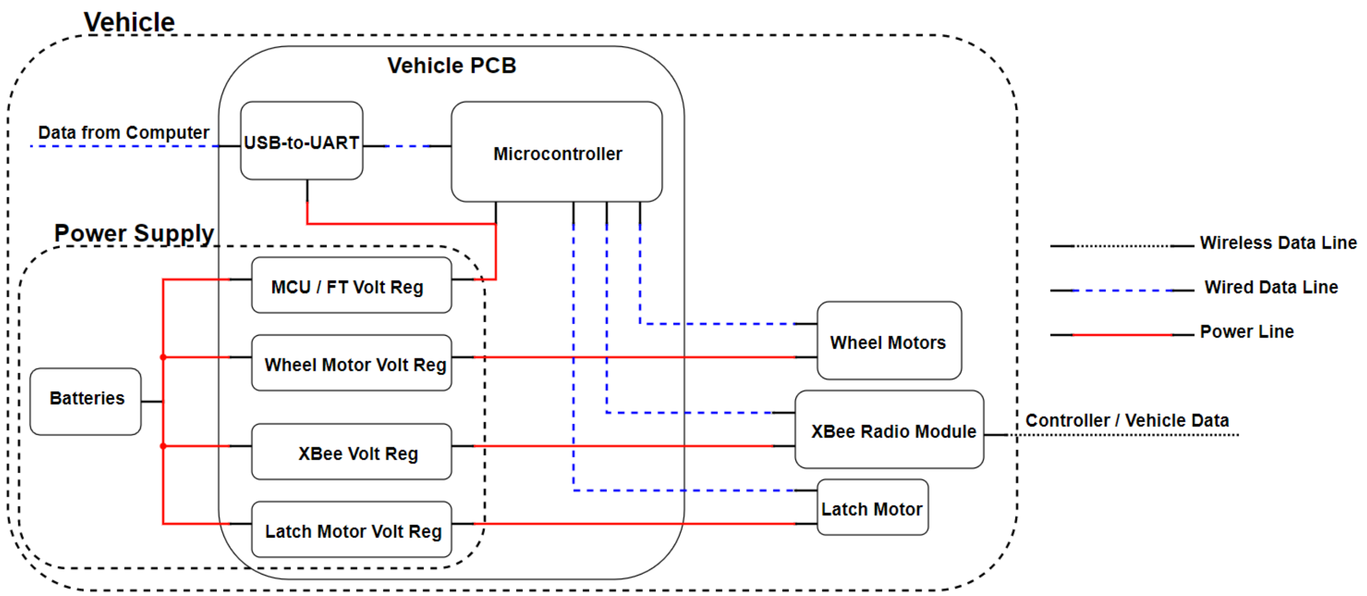

Fig. 1. Vehicle Block Diagram

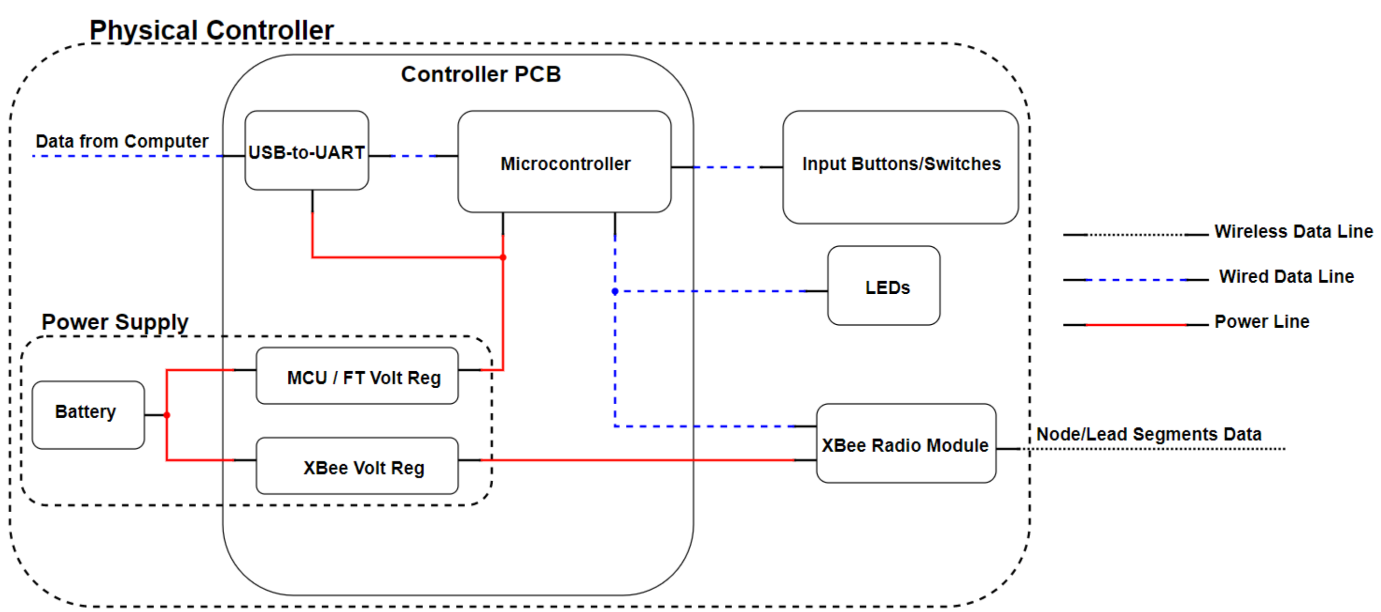

Fig. 2. Controller Block Diagram

2.2 Physical Diagram

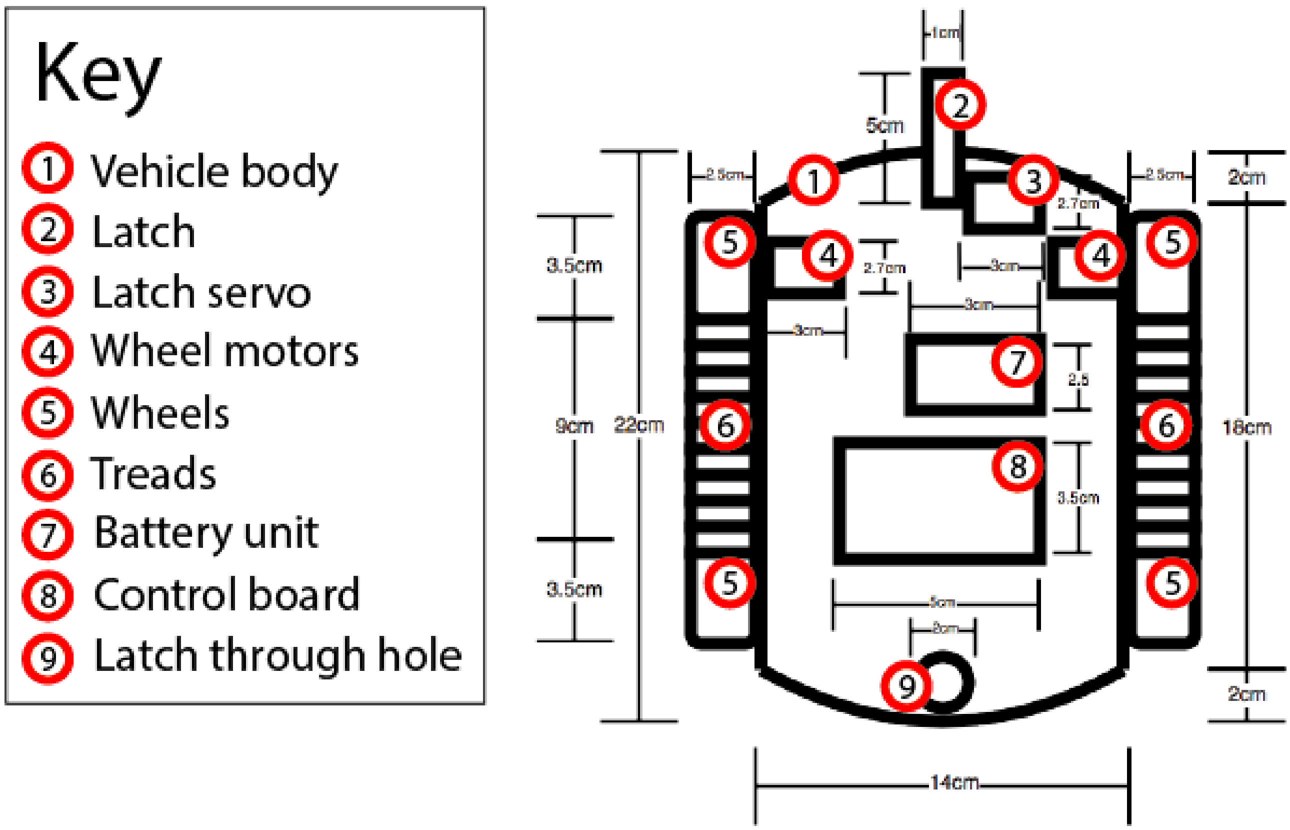

Fig. 3. Vehicle Physical Diagram

2.3 Block Designs

The following sections go into greater detail on the blocks making up the block diagram. The sections are split into two main categories: Vehicle and Controller. The sections under the Vehicle category covers the complex elements making up the vehicle design, functionality, and control scheme. The sections under the Control category describe the design of the controller and its methods for transmitting instructions.

2.3.1 Vehicle Block Designs

2.3.1.2 Power Supply - Battery

For our vehicle design, we have chosen an Energizer 9V Alkaline battery and Energizer 9V Lithium battery as our sources of power for the electrical components in our vehicle design. The alkaline battery will be used to power the microcontroller, USB-to-UART chip, XBee radio module and latch mechanism motor. The lithium battery will be used exclusively to power both wheel motors.

Alkaline Battery power consumption [4]:

1) Current Lower Bound: $10 + 15 + 45 + 100 =$ 170 mA
2) Current Upper Bound: $14 + 15 + 50 + 100 =$ 179 mA
3) Estimated Operating Time: $200\ mAh / 179mA =$ 1.12 hours

Based on our calculations, the microcontroller, USB-to-UART, XBee module and latch motor will easily surpass our timing requirement.

Lithium Battery power consumption [5]:

1) Current Lower Bound: 500 mA
2) Current Upper Bound: 600 mA
3) Estimated Operating Time: $600mAh / (2 * 600mA) =$ 30 minutes

Based on our calculations, the two wheel motors will pass our timing requirement if we provided a current closer to our lower bound.

Requirement	Verification
1. Alkaline Battery must store at least 100 mAh to meet operational time requirement of 30 minutes	1. Charge test a. Connect fully charged battery to a resistor circuit b. Discharge battery using resistor circuit for .5 hours c. Use voltmeter to measure and ensure battery voltage d. Make calculations to ensure at least 100mAh of charge is available
2. Lithium Battery must store at least 500 mAh (for two motors) to meet operational time requirement of 30 minutes	2. Charge test a. Connect fully charged battery to a resistor circuit b. Discharge battery using resistor circuit for .5 hours c. Use voltmeter to measure and ensure battery voltage d. Make calculations to ensure the battery can store 500mAh of charge

2.3.1.3 Power Supply - Voltage Regulators

The voltage regulators supply a constant voltage based on component-specific requirements to the respective components that require them. Each of our regulators will convert 9V from our alkaline battery power supply to the specified voltages below:

- MCU / USB-to-UART: 5V
- XBee Module: 3.3V
- Wheel Motors: 9V
- Latch Motor: 5V

Requirement	Verification
1. Provide component-specific output voltage (see above) reliably from a 9V source	1. Reliable voltage regulation test a. Setup voltage supply of 9V from either a voltage generator or a battery input b. Measure output voltage and current using a multimeter to be 9V and 50mAh respectively

2.3.1.4 USB-to-UART IC

The FT232RL is a USB to serial UART interface with single chip USB to asynchronous serial data transfer interface. It operates at ~5V with a current draw of 15 mA for normal operation [6]. The entire USB protocol is handled on the chip. We will use the chip to interface and program the MCU which is going to control and process signals from the XBee module and motor components of the project. The chip requires no USB-specific firmware programming.

Requirement	Verification
1. Transmit data between UART and USB at 9.6 kbps	1. USB-UART testing a. Use a socket connection to connect a computer to the MCU b. Use Arduino online interface to upload processing code to the MCU bootloader c. Test connection reliability for different data transfer rates d. Ensure data rate of at least 9.6kbps is achievable

2.3.1.5 Microcontroller Unit (MCU)

For our microcontroller, we chose the ATmega2560 since it is low-power and high performance. Specifically, it will receive a regulated voltage of 5V at 10 - 14mA, which allows the clock to run at ~8 MHz [7]. For our vehicle design, it communicates with the FT232RL chip to translate our software program from USB to UART through the Serial Peripheral Interface (SPI) protocol, then load it into the MCU's programmable Flash memory. It also communicates with the XBee radio module to send/receive data packets wirelessly to and from the physical controller. It interprets the encoded packets and accordingly communicates with the wheel motors and latch motor. It controls the wheel and latch mechanism motors, using its Pulse Width Modulation (PWM) output pin.

Requirement	Verification
1. Can receive and transmit data from Serial interface at a minimum rate of 256 kbps	1. SPI testing a. Connect MCU to Serial interface using FT232 chip to a terminal interface b. Start timer and send data packet of specified size c. Echo data back from the MCU d. Stop timer and calculate data transfer rate and ensure at least 256 kbps
2. Can output PWM of duty cycle 50% signals to control motors for to navigation and	2. PWM testing a. Connect MCU general output pin to breadboard b. Make breadboard connection to oscilloscope c. Note PWM duty cycle range to be at least functioning at 50%

2.3.1.6 XBee Radio Frequency Module

We chose the XBee Series 1 RF module to establish wireless communication between the physical controller, node segment and lead segment vehicles. The modules we have chosen operate on 2.8 - 3.4V at a current of 45-50 mA. They run at a 2.4GHz frequency and their method of network and data transmission is based on the 802.15.4 standard [8]. It supports point-to-point and multipoint networks. On the same configured network, two XBees in the same area they will automatically 'sync' and pass serial data back and forth using TTL serial interface. XBee also allows us to configure baud rate, sleep modes and power modes, which are relevant to creating a scalable design for the project.

Requirement	Verification
1. Maintain baud rate of at least 100kbps	1. Baud rate testing a. Connect XBee module to XBee Explorer and open X-CTU interface b. Use X-CTU interface to configure network and send random data packets of different data sizes c. Use another XBee Explorer to collect data from network and calculate baud rate d. Test for different baud rates for both receiver and transmitter and ensure measurement of at least 100kbps
2. Each module should maintain a transmission range of at least 60 ft in outdoor settings	2. Range testing a. Connect Xbee module to XBee Explorer and open X-CTU interface b. Establish connection with another XBee module to receive data packets c. Measure at distances of 100, 200, 300, 400 and 500 feet (and at least 60 feet is functional)

2.3.1.7 Vehicle Software

Software running on the on our node segment and vehicle segment PCBs will be used to handle their functional operations. It will manage unpacking instructions sent from the controller using those instructions to drive vehicle operations.

Requirement	Verification
1. Receive and acknowledge instructions received through the radio module	1. Connection testing a. LEDs on the controller will be used to give a visual indication that a node is receiving and acknowledging instructions
2. Can correctly interpret instructions	2. Instruction testing a. Establish an acknowledged connection between the controller and a node b. Send instructions to change the configuration of the latch and to drive the wheels in the following ways: i. Forward, straight ii. Forward, left iii. Forward, right iv. Backward, straight v. Backward, left vi. Backward, right
3. Can facilitate the re-transmission of instructions from node to node	3. Daisy-chain software testing a. LEDs on the controller will be used as a visual indication that a node is receiving and acknowledging instructions b. One node N1 will be placed inside the range of the controller, another N2 outside the range of the controller c. If the indicator shows that N2 is receiving and acknowledging instructions, this will show that N1's software is successfully facilitating a re-transmission

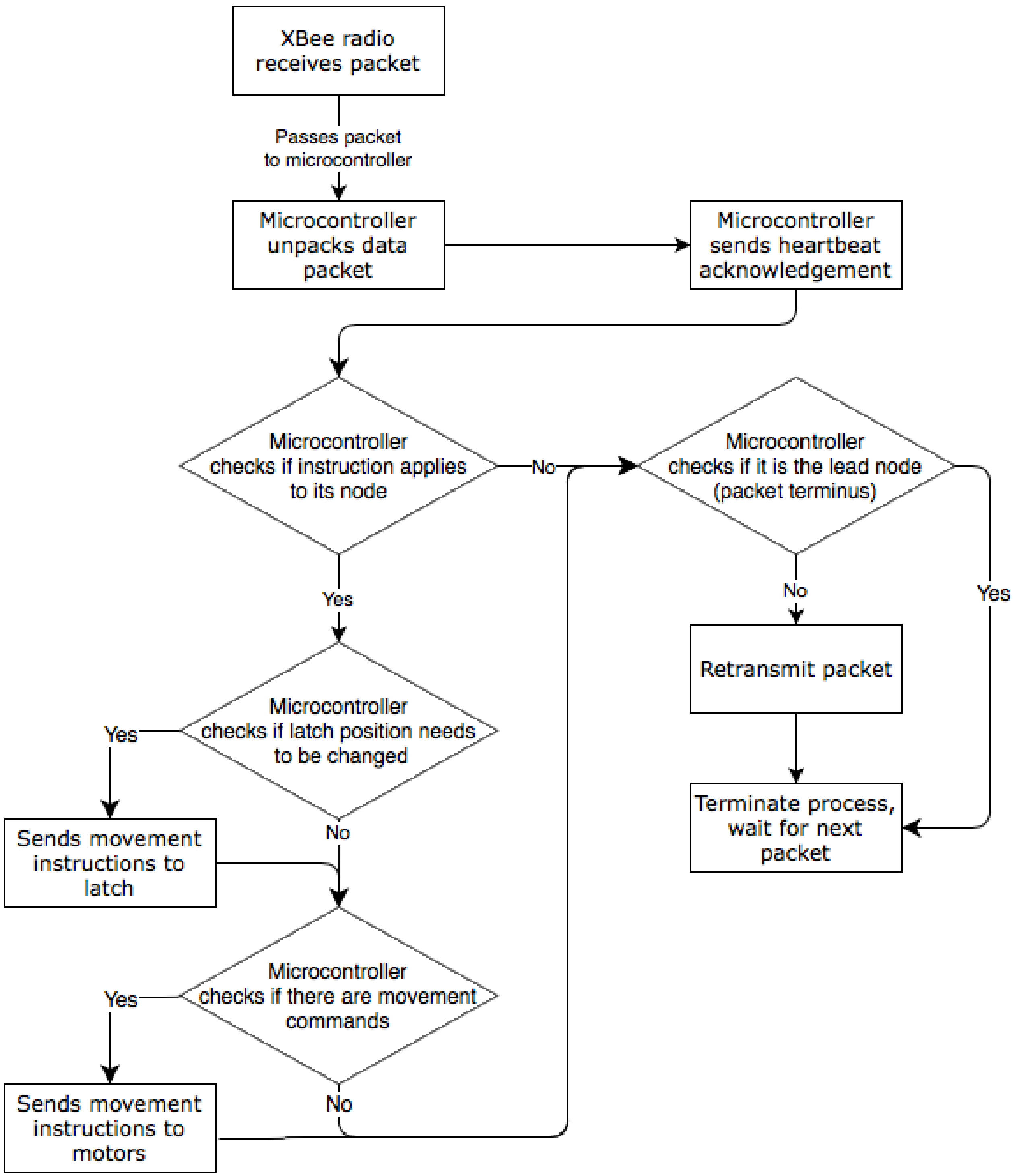

Fig. 4. Packet Handling Algorithm

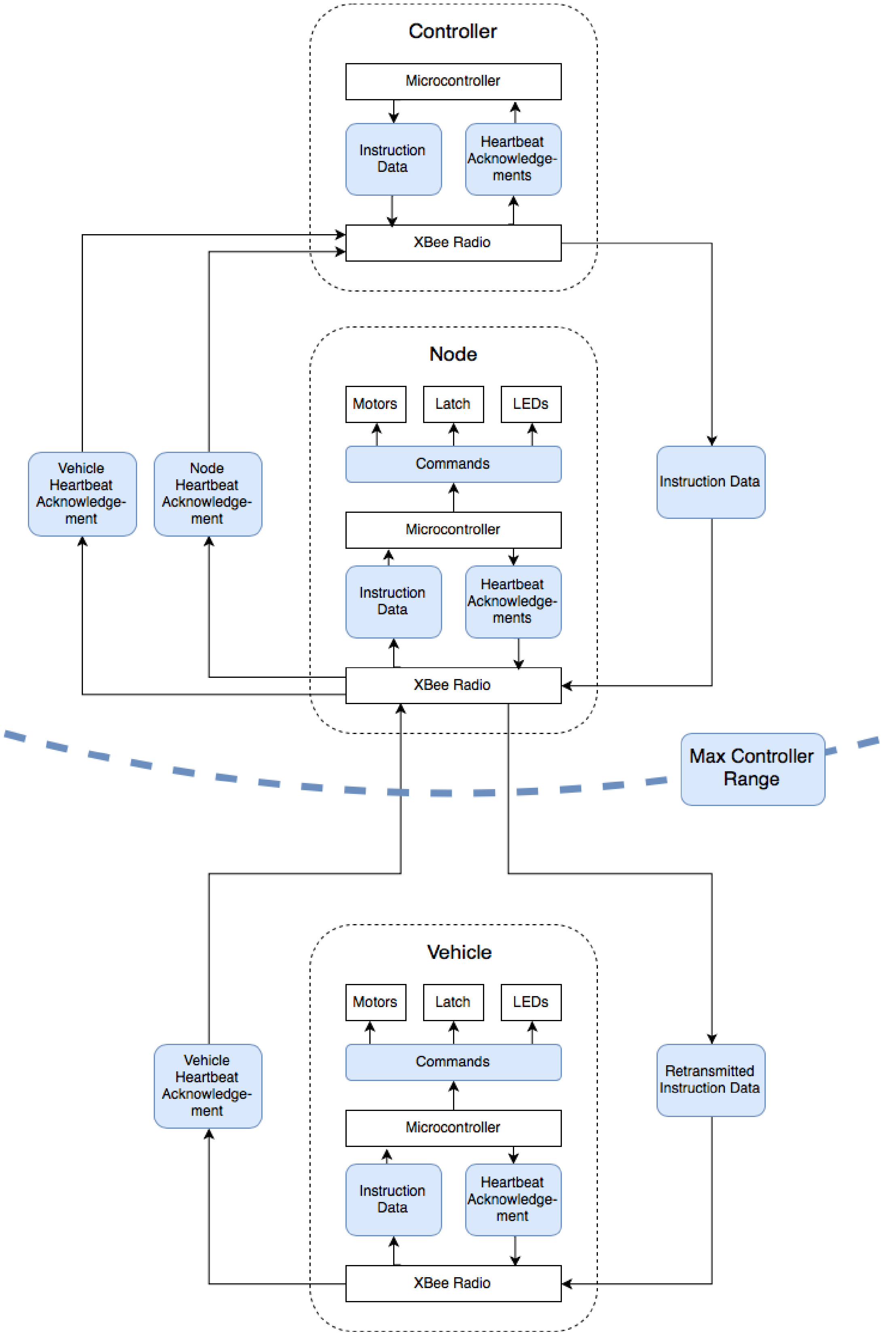

Fig. 5. Data Transmission Between Nodes

2.3.1.8 Wheel Motors

We chose the UAV Brushless Motor A2212/13 1000kv motors to propel the node segment and the lead segment. We will run the motors at close to 9V at 500 - 600 mA [9]. Given the density of PLA plastic material for 3D printing at 1.25 grams/cm³ [10], we will have an estimated vehicle weight as calculated below:

$$18cm * 14cm * 2cm * 1.25g/cm^3 = 630\text{g}$$

Based on this calculation and given that the vehicles will be driven by 2 motors at an RPM/V constant of 1000kv, we estimate that we will have more than enough torque to move our vehicle segments [9].

Requirement	Verification
1. The torque requirements for the latch servo is at least a minimum of 1kg/cm to propel the node segment	1. Torque requirements a. Connect motors to voltage supply a. Attach drive gear to the end of the motor b. Connect latch sprockets and latch physical design to the gear c. Observe torque and latch acceleration for different input voltages/current for the motor and to ensure we have torque of at least 1kg/cm
2. Temperature of the motors does not exceed 60°C in persistent (>2 minute) duty cycle changes.	2. Temperature test d. Connect servo motors to MCU output pin e. Create different voltage inputs using serial signal library for the MCU f. Use IR thermometer to measure temperature of the motor casing and ensure measurement doesnt go above 60°C

2.3.1.9 Latch Motor

The motor we have chosen for our latch mechanism is a 180° rotation servo motor. It operates at 4.8 - 6.6V and a current of 100mA [11]. It also provides a stall torque of 9.4kg/cm, which is sufficient to raise a latch mechanism which, given the density of PLA plastic material for 3D printing at 1.25 grams/cm³ [10], will have an estimated weight of:

$$5\ cm * 3\ cm * 1\ cm * 1.25\ g/cm^3 = 18.75\ g$$

The physical latch mechanism will be powered by this motor in such a way that when the mechanism is connecting the node and lead segments together, if a user sends the latch input command from the physical controller, the motor will either raise or lower the mechanism to respectively detach or attach the two segments.

Requirement	Verification
1. The torque requirements for the latch servo is at least a minimum of 1kg/cm to support the weight of the 3D printed latch.	1. Torque requirements a. Connect servo motors to voltage supply b. Attach drive gear to the servo end of the motor c. Connect latch sprockets and latch physical design to the gear d. Observe torque and latch acceleration for different input voltages/current for the motor
2. Temperature of the servo motors does not exceed 60°C in persistent (>2 minute) duty cycle changes.	2. Servo temperature test g. Connect servo motors to MCU PWM pin h. Create different duty cycle using serial signal library for the MCU i. Use IR thermometer to measure temperature of the motor casing

2.3.2 Controller Block Designs

2.3.2.1 Power Supply - Battery

The total power supply for our controller is only a single alkaline 9V battery, since we do not have any motors to power (which require higher current draw). Since we are only powering an MCU, USB-to-UART chip and and XBee module, our estimated power consumption is:

1) Current Lower Bound: $10 + 15 + 45 =$ 70 mA
2) Current Upper Bound: $14 + 15 + 50 =$ 79 mA
3) Estimated Operating Time: $200\ mAh / 79mA =$ 2.53 hours

Based on our calculations, an alkaline 9V battery will have no trouble powering the controller for at least 30 minutes to meet our timing requirement.

Requirement	Verification
1. Alkaline Battery must store at least 50 mAh to meet operational time requirement of 30 minutes	1. Charge test a. Connect fully charged battery to a resistor circuit b. Discharge battery using resistor circuit for .5 hours c. Use voltmeter to measure and ensure battery voltage and ensure charge of at least 50mAh

2.3.2.2 Power Supply - Voltage Regulators

The voltage regulators in our controller design are identical to the ones we use in our vehicle design, except we do not need regulators for the wheel and latch motors. See Section **2.3.1.3**.

2.3.2.3 USB-to-UART IC

The functionality of this chip on the physical controller's PCB is identical to that of the same chip on each of the vehicle segment PCB's. See Section **2.3.1.4**.

2.3.2.4 Microcontroller Unit (MCU)

The functionality of the ATmega2560 chip for our physical controller is nearly identical to the chip in the vehicle design, minus the communication with wheel and latch mechanism motors (See Section **2.3.1.5**). In place of the motors, it instead communicates with physical buttons, switches and LEDs for vehicle movement and latch attach/detach input commands (Fig. 6).

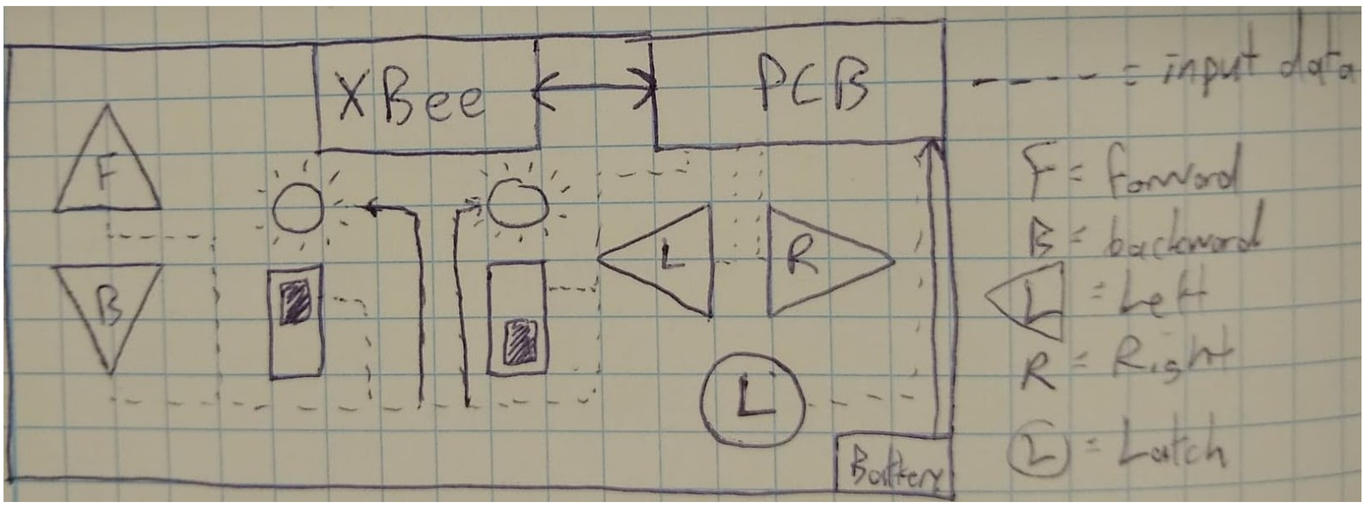

Fig. 6. Sketch illustrating input buttons, switches and LEDs

2.3.2.5 XBee Radio Module

The XBee radio modules provide identical functionality as they do in our vehicle design. See Section **2.3.1.6**.

2.3.2.6 Controller Software

The software on the controller's microcontroller is responsible for creating instruction packets and sending them to the controller's XBee radio to be transmitted. Figure 7 details the software flow diagram which outlines this process.

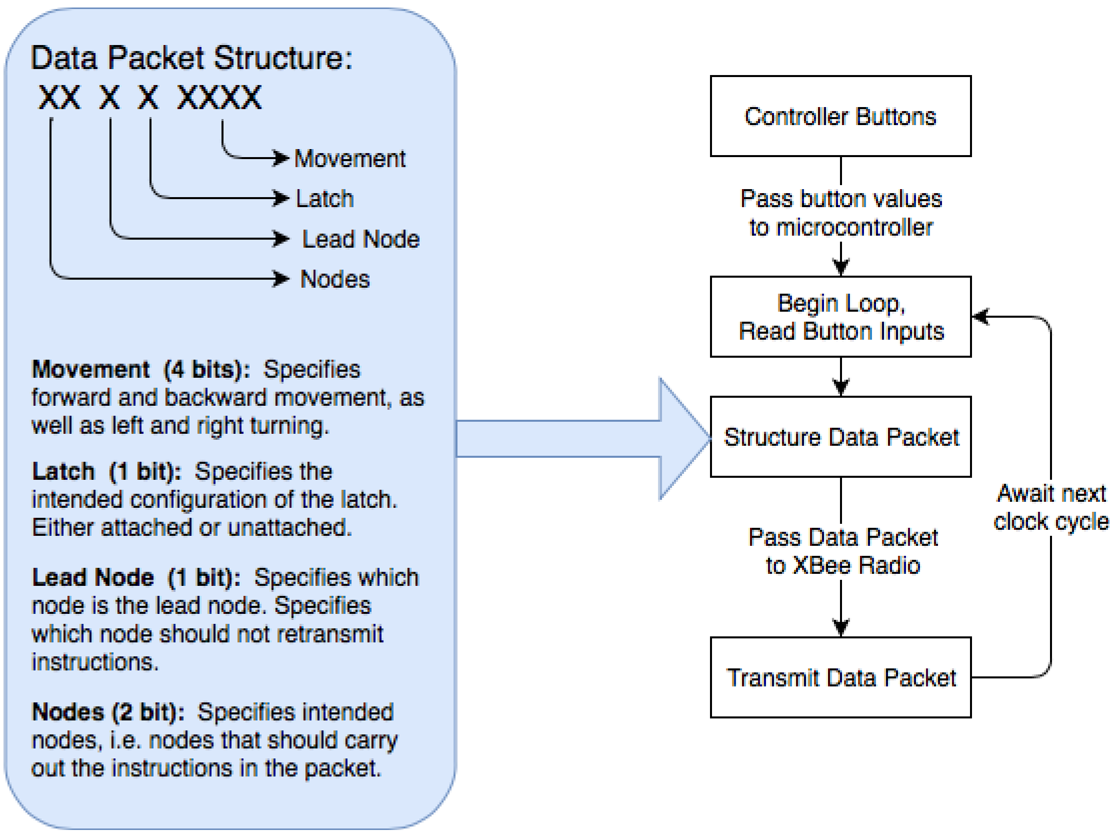

Fig. 7. Instruction Packet Creation in the Controller

Requirement	Verification
1. Software must be able to transmit package instructions correctly based on button input	1. Packet fidelity testing a. Establish a connection between the controller-mounted XBee radio and a radio module inserted in a computer's serial port b. Motor packets received and ensure bits properly align with what should be transmitted

2.3.2.7 Input Buttons / Switches

Input buttons and switches are an important component of the project. The controller uses the input buttons to control the segment motors and movement as well as the latch mechanism. The switches are used to set the segment to be in “Active” i.e. in a state to receive controller data packets and operate on the commands.

Requirement	Verification
1. Buttons must be easily pressable	1. Button Test a. Press button and ensure ease in pressing down
2. Switches must reliably switch between on and off	2. Switch Test a. Connect switches to a breadboard b. Attach voltage supply across the switches c. Measure current using a multimeter in the on-off states of the switch

2.3.2.8 LEDs

The LEDs are going to be used to indicate which segment is actively receiving commands. The LEDs turn on if the switch from the controller is switched on and the segment is actively responding with an XBee ack signal. We can use these to know if the segment is operational.

Requirement	Verification
1. Must be visible from 1 meter away with a drive current of 10mA	1. LED visibility test a. Supply 10mA input voltage using voltage supply b. Test visibility from 1m away

2.4 Tolerance Analysis

We are using XBee Series 1 RF modules which use PAN ID based WSN networks to communicate with nodes. The reliability of such wireless networks is affected by in-band radio frequency interference, multipath distortion due to reflection of RF waves and anomalies in the node sensors and receivers. By measuring the reduction in throughput of by external interference, we can establish a good baseline for errors and fault tolerance. We can use received signal strength and signal quality as indications as well as record time to recover from faults. We can perform these using the XBee Evaluation Kit components [12]. The RF Interference metrics and test protocols address the following goals:

1. Measure radiated power output and received signal strength of the RF Physical layer
2. Measure Packet Loss and Data Throughput rate at the MAC layer nominally and in the presence of multipath interference
3. Measure Packet Loss and Data Throughput rate with active WLANs operating within WSN channel allocation [12].

The XBee Evaluation Kit has spectrum analysis tools (including a raspberry pi MCU) for the 802.15.4 signal analysis [13].

3. Cost and Schedule

3.1 Cost Analysis

The following sections outline the development costs of this project. This analysis accounts for cost of labor for a team of three UIUC ECE new graduate engineers, cost of components making up the design, and the cost of necessary specialty services.

3.1.1 Parts List

Part	Cost (prototype)	Cost (bulk)
Microcontroller * 3 (Microchip, ATMEGA2560V-8AU)	$35.55	$25.83
XBee Radio chip * 3 (Digi-Key, 602-1892-ND)	$75.00	$51.00
USB to UART * 3 (Mouser, 895-FT232RL)	$13.50	$7.05
Wheel motors * 4 (Amazon, YoungRC A2212 1000KV Brushless Motor)	$59.94	$59.94
Latch motors * 2 (Amazon, KOOKYE 1PCS Servo Motor Metal Gear 180 Degree Rotation)	$19.98	$19.98
Batteries * 3 (Energizer, 522 9 Volt Alkaline Battery)	$9.00	$7.50
Batteries * 2 (Energizer, 522 9 Volt Lithium Battery)	$20.00	$8.00
PCBs (PCBWay)	$5.00	$1.00
Assorted circuit components (resistors, capacitors, ICs, etc) (Digikey; est.)	$10.00	$0.40
3D printer filament 1kg * 2 (Hatchbox, 3D PLA-1KG1.75-BLK)	$39.98	$17.99
Total	**$287.95**	**$190.69**

Additionally, we include a cost of $2 per hour of 3D printer time over 30 hours. We incur an additional cost of $60. All included, our prototype cost is $347.95.

3.1.2 Labor

Our labor costs estimate a $45/hour salary per engineer. They also estimate a 10 hour per week work schedule per engineer, over the 15 week duration of ECE 445. 3 engineers are working on our team.

$$\frac{\$45}{1\ hour} * \frac{10\ hours}{1\ week} * \frac{15\ weeks}{} * 3 * 2.5 = \$50,625$$

Out estimated cost of labor is: $50,625.

3.1.3 Grand Total

By adding up out cost of labor and our parts and services costs, we get a total cost of producing a prototype of:

$$50,625.00 + 347.95 = \$50,972.95$$

3.2 Schedule

Week	Task	Delegation
9/30	Prepare for Mock Design Review, work on Design Document	Dhruv
	Prepare for Mock Design Review, work on Design Document	Michael
	Prepare for Mock Design Review, work on Design Document	Thomas
10/7	Finalize controller/vehicle PCB designs	Dhruv
	Design software flow diagrams	Michael
	Design wheel, treads, and chassis models for 3D printing	Thomas
10/14	Order components	Dhruv
	Finalize software flow diagrams/code, order components	Michael
	Print version 1 wheels, treads, and chassis	Thomas

10/21	Begin version 2 PCBs	Dhruv
	Test node connections (controller: latch attach/detach input)	Michael
	Test node connections (controller: movement inputs)	Thomas
10/28	Finalize version 2 PCBs and order	Dhruv
	Debug node connections (daisy-chained commands)	Michael
	Debug node connections (daisy-chained commands)	Thomas
11/4	Assemble components, finalize controller design	Dhruv
	Assemble components, test node handoffs	Michael
	Assemble components, test node handoffs	Thomas
11/11	Test drive logic with final controller	Dhruv
	Verify module integration for vehicles	Michael
	Debug node handoffs	Thomas
11/18	Design Test	Dhruv
	Design Test	Michael
	Design Test	Thomas
11/25	Design Test	Dhruv
	Design Test	Michael
	Design Test	Thomas
12/2	General bug fixes/improvements	Dhruv
	General bug fixes/improvements	Michael
	3D print shell/housing for nodes and controller	Thomas
12/9	Final Presentations, finish final paper	Dhruv
	Final Presentations, finish final paper	Michael
	Final Presentations, finish final paper	Thomas

4. Ethics and Safety

Safe and ethical practices are also a concern in the completion of this project, and there are multiple components of the vehicle that require special consideration in this area. The first is the radio module. All communications between the controller and the vehicle will be via radio. As much of the project's development will take place in the ECEB, and there are many signal processing student projects, both in senior design and in the rest of the ECE curriculum, being worked on in the building it is important that we do not interfere with their development.[14] We will fully adhere to FCC standards for amateur radio communication while developing our project, so as to avoid harmful interference with other entities. As one of our group members is a certified amateur HAM radio operator, the FCC regulations are well documented, and our development does not call for transmitting without a specific goal in mind we judge this to be a low level, manageable risk. We will also ensure that the channels of communication for the RF module are not susceptible to attacks from proxy controllers.[14] XBee RF series 1 modules allow encrypted transmission from the modules and the PAN ID based network access model ensures only network configured XBee devices are able to communicate reliable and freely in the network.

The second area in need of consideration is the battery. We plan on using alkaline and lithium batteries to power our vehicle segments, which can be dangerous if handled incorrectly. To mitigate this risk, we are following common sense battery handling procedures. We will only transport our batteries - and our vehicle segments if the batteries are installed - in a protective, padded case to reduce the risk of puncturing them if dropped. We will also ensure that we are not storing our batteries, or testing our vehicle in any wet or damp environment. We will only test dry, controlled environments where fire-suppression tools are easily accessible. Through these practices and through constant awareness, we believe this risk will also be low.

We will ensure all the circuits are unexposed and use good engineering practices to ensure all circuitry is properly soldered and tested for temperature conditions with the battery loads. The motors from the circuit are also susceptible to overheating and damage from running at improper voltage inputs. All components of the project have voltage regulator which ensure that the connected devices receive proper voltage and current inputs to maintain good working state for the motors. PLA has a glass transition temperature 60–65 °C, a melting temperature 173–178 °C. With proper mounting we will ensure that the 3D printed design is not affected by the heat from the circuitry or the motors.[14]

5. Citations

[1] nsf.gov. (2001). *At WTC Search, Graduate Students Deploy Shoebox-Sized Robots*. [online] Available: https://www.nsf.gov/od/lpa/news/press/01/pr0178.htm [Accessed 20 Sep. 2018].

[2] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang and X. Shen, "Drone Assisted Vehicular Networks: Architecture, Challenges and Opportunities," in *IEEE Network*, vol. 32, no. 3, pp. 130-137, May/June 2018.

[3] Wondershare Inc., “Top 10 Drones with Longest Flight Time for 2018”, 2018 [Online]. Available: https://filmora.wondershare.com/drones/drones-with-longest-flight-time.html [Accessed 4 October 2018]

[4] Energizer, “Energizer 522”, 9V Alkaline Batterydatasheet, 2018. [Online]. Available: http://data.energizer.com/pdfs/522.pdf [Accessed 4 October 2018].

[5] Energizer, “Energizer L522”, 9V Lithium Battery datasheet, 2018. [Online]. Available: http://data.energizer.com/pdfs/l522.pdf [Accessed 4 October 2018].

[6] Future Technology Devices International Ltd., “FT232R USB UART IC Datasheet”, FT232RL datasheet, Aug. 2005. [Revised Jun. 2018]. [Online]. Available: https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf [Accessed 4 October 2018].

[7] Atmel, “8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash”, ATmega2560 datasheet, Feb. 2014. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf [Accessed 4 October 2018].

[8] Digi International Inc., “XBee/XBee-PRO RF Modules”, XBee Series 1 RF module datasheet, 2009. [Online]. Available: https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf [Accessed 4 October 2018]

[9] RobotShop, “UAV Brushless Motor”, January 2018. [Online]. Available: https://www.robotshop.com/en/uav-brushless-motor-a2212-13-1000kv.html#description [Accessed 4 October 2018].

[10] Jason Bowling, “Estimating Cost of 3D Prints in PLA”, *blogspot.com*, Feb. 2, 2014. [Online]. Available:http://www.shortcircuitsandinfiniteloops.blogspot.com/2014/02/estimating-cost-of-3d-prints-in-pla.html [Accessed 4 October 2018].

[11] KOOKYE, “KOOKYE Servo Motor Metal Gear 180 Degree Rotation”, *kookye.com*, Jan. 1, 2016. [Online]. Available: http://www.kookye.com/2016/01/01/kookye-servo-motor-metal-gear-180-degree-rotation/. [Accessed 4 October 2018].

[12] Richard Alena, Ray Gilstrap, Jarren Baldwin, Thom Stone, and Pete Wilson**, “**Fault Tolerance in ZigBee Wireless Sensor Networks”, *2011 IEEE Aerospace Conference, Big Sky, MT*, March 2011. [Abstract]. Available: https://ti.arc.nasa.gov/publications/2724/download/ [Accessed 4 October 2018].

[13] National Instruments, “The Basics of ZigBee Transmitter Testing”, March 2011. [Online]. Available: ftp.ni.com/evaluation/rf/The_Basics_of_ZigBee_Transmitter_Testing.pdf [Accessed 4 October 2018].

[14] IEEE, “IEEE Code of Ethics”, January 2018. [Online]. Avaiable: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed 4 October 2018].