

**Tilt-Rotor Drone**  
**by**  
**Ben Stone & Zach Crandall**

ELECTRICAL ENGINEERING DEPARTMENT  
California Polytechnic State University  
San Luis Obispo, CA  
2015-2016

## **Abstract**

A drone is defined as “an unmanned aircraft that can fly autonomously [2].” In the year 2016, there exist many applications for small scale drones. Drones, having had primarily military applications, now have use in hobby robotics, photography, and even in surveying. But drones have not yet been perfected and still have room for much improvement. Several problems with drones come in the form of battery life, stability, and size. Large drones can fit many sensors and probes, but are heavy and have less flight time. For autonomous drones, stability can be an issue even though there exist many stable drones. Finally, drones can be large which make them difficult to move around.

The project proposed in this document is a form of drone called a tilt-rotor drone. It uses only two rotors, unlike the popular four rotor design. The goal is maximizing this drone's flight time while minimizing its size and cost.

## **Acknowledgments**

First thanks to Dr. Benson for allowing us to pursue this project. Also thank you to the Cal Poly mechatronics lab for the use of their equipment, and to Zach's father for allowing us to use his tools to create our drone chassis; we appreciate it.

## TABLE OF CONTENTS

Abstract & Acknowledgement.....	1
Introduction.....	3
Requirements and Specifications.....	4
Design.....	6
Testing.....	13
Conclusion and Future Work.....	16
References.....	16
Appendix A.....	18
Appendix B.....	24

## LIST OF TABLES

TABLE I - Requirements and Specifications.....	3
TABLE II - Tilt-Rotor Drone - Level Zero Function Table.....	6
TABLE III - Tilt-Rotor Drone - Level 1 System Table.....	7
TABLE IV - Current Hardware Implementation.....	8
TABLE V - Current Requirements and Expectations Met.....	14

## List of Figures

Figure 1 - Tilt Rotor Drone - Level Zero Block Diagram.....	6
Figure 2 - Tilt Rotor Drone - Level One System Block Diagram.....	7
Figure 3 - Completed Tilt Rotor Drone with Landing Pad.....	10
Figure 4 - PID Model.....	11
Figure 5 - Overall software flow diagram.....	12
Figure 6 - Drone in Hovering Test.....	14

## Introduction

The motivation for the tilt-rotor drone comes from the V-22 Osprey. “In partnership with Boeing, Bell Helicopter Textron built the tiltrotor aircraft that takes off and lands vertically like a helicopter, and once airborne, can convert to a turboprop airplane capable of high-speed and high-altitude flight [3].” This tiltrotor aircraft, called the V-22 Osprey, was designed for different military applications including transportation. But a huge problem with the Osprey is that if one rotor breaks during a flight then there is no failsafe to protect it. If the Osprey is meant to transport people, this presents a serious problem. With a drone, the problem with death can be avoided. And with a lightweight design, it is possible to save the drone with a parachute.

With the V-22 Osprey inspired drone, people can have the full benefits of the tilt-rotor design without worrying about the main problem which is the lack of a failsafe. If the drone fails at low height, then it can be damaged. But there is no gas involved, so the drone cannot not explode. If the drone fails high in the air, then a parachute is be deployed to help protect the drone from the fall. This efficient design can be used in surveying, photography, and anything a drone with more than two rotors can do. This drone design provides decreased power consumption, and may potentially outperform other drone designs by maximizing its flight time.

The Osprey’s tilt-rotor drone’s design has been formulated to meet the problems of flight time and size. Using two rotors instead of four or more results in decrease power used. This gives an advantage to this drone over other drones built with more rotors. The tilt-rotor design also gives the drone a unique ability to both rotors forward, forcing the drone to act as a plane. When the rotors tilt back to their normal positions, the drone then acts like a helicopter. This design gives the advantage of taking of and maneuvering like a helicopter, while flying with the speed of a plane. With these factors in mind, it is a clear choice to design a drone based on the V-22 Osprey.

## Requirements and Specifications

The requirements and specifications for the tilt-rotor drone have been derived from the customer needs. The customers are looking for a low-cost, lightweight, and portable drone. The drone must have a camera and SD card, and it must have a long flight time. Most importantly, the drone must be autonomous with an interface for uploading GPS coordinates that the drone can follow to complete its mission. Table 1 shows a list of engineering specifications and their justifications along with a set of marketing requirements for the drone.

TABLE I: Tilt-rotor drone requirements and specifications

Marketing Requirements	Engineering Specifications	Justification
1,2	The drone must weigh less than 5 lbs.	In order to maximize flight time, the drone must be lightweight. This is because the lighter the drone, the less power the drone needs to consume to create lift.
6,7	The drone must be capable of flying autonomously over a GPS defined flight path.	The customers of this type of drone need the drone to follow a flight path via a given set of GPS coordinates.
3	The drone must have a built on SD card with at least 16 Giga Bytes of memory.	The memory size is large enough to store around 5000 average sized photographs.
2	The drone must fit within a 2'x6"x6" rectangle.	The size constraint is based on portability. The drone must be able to fit in a travel case. This case must be small enough to fit in the trunk of a small car.
5	The price to produce one unit must not cost more than \$150.	This price is less than 3 times my projected selling cost which is \$475.99.
4	The drone must have a camera with at least a 10 Mega pixel camera.	This resolution provides enough resolution for general photographic applications.
8	The drone must have a rechargeable 14V battery that powers the drone in flight for 15 minutes.	The voltage is enough to power all of the motors and sensor on the drone. The batteries must be rechargeable as they are depleted after each use.
4	The drone must be capable of taking photos of the ground.	A good spot for a camera mount is underneath the drone, and ground

		pictures are good for surveying and photography
<b>9</b>	The drone must be capable of flying within 10ft of people without causing any harm.	This is to keep people safe from the drone.
<b>10</b>	The drone must not fly higher than the legal altitude for hobby drones.	This is to keep the drone from breaking any laws.
<b>11</b>	Must be able to communicate via Bluetooth to receive GPS coordinates.	The drone needs to receive GPS coordinates, and a Bluetooth interface provides easier use for users with smart devices.
<b>Marketing Requirements</b> <ol style="list-style-type: none"> <li>1. Lightweight</li> <li>2. Portable</li> <li>3. SD Card</li> <li>4. Camera</li> <li>5. Low-Cost</li> <li>6. Autonomous</li> <li>7. GPS Capabilities</li> <li>8. Rechargeable Batteries</li> <li>9. Safe to Use</li> <li>10. Legal</li> <li>11. Bluetooth Compatible</li> </ol>		

## Design

### Hardware

Our project is composed of a mix of mechanical and electrical hardware. Figure 1 and Figure 2 show the level zero block and level one block for the project respectively, and Table II Table III provide a description of the inputs and outputs along with the functionality of the blocks respectively.

A pin diagram for the final hardware configuration can be found in Appendix 2 at the end of this report.



Figure 1: Tilt-Rotor Drone - Level Zero Function Block Diagram

Table II: Tilt-Rotor Drone - Level Zero Function Table

Module	Tilt-Rotor Drone
Inputs	<ul style="list-style-type: none"> <li>● Power: 14VDC</li> <li>● Bluetooth: Used for inputting GPS coordinates to the drone</li> </ul>
Outputs	<ul style="list-style-type: none"> <li>● Bluetooth: Used to provide user with feedback on its status and whether or not it received the user input command or not</li> </ul>

Functionality	Controls Drone via the Bluetooth input. System takes in Bluetooth and power to control internal functions. System displays status to users through the Bluetooth output
---------------	---

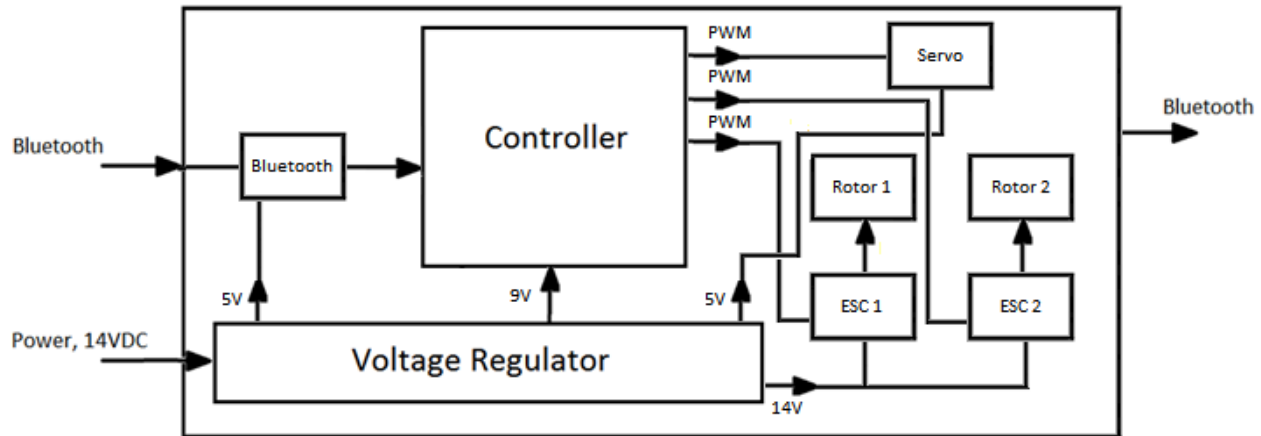


Figure 2: Level 1 System Block Diagram

Table III - Tilt-Rotor Drone Level 1 System Table

Module	Hardware
Inputs	<ul style="list-style-type: none"> <li>Bluetooth - Used for serial communication. Commands sent to drone via bluetooth module.</li> <li>Power - 14V input. Higher voltage used to power rotors.</li> </ul>
Controller	<ul style="list-style-type: none"> <li>ATMega2560 based system. Microcontroller used to send information, such as a PWM waveform, to other modules in the system.</li> </ul>
Bluetooth	<ul style="list-style-type: none"> <li>Breakout module added for wireless communication. Receives commands from bluetooth transmitter and communicates through serial Rx pin on Arduino board.</li> </ul>
Voltage Regulator	<ul style="list-style-type: none"> <li>Divides voltage down from 14V to 9V used for the</li> </ul>
Servo	<ul style="list-style-type: none"> <li>Provides steering and balancing. Controlled with a PWM waveform sent from the controller in order to adjust servo angle.</li> </ul>
ESC's	<ul style="list-style-type: none"> <li>Take PWM waveform from controller and translate it into a three phase signal used by the rotors.</li> </ul>
Rotors	<ul style="list-style-type: none"> <li>Provide lift and thrust capabilities. The rotors cannot be powered by a DC signal, so the ESC's provide three phase power to them.</li> </ul>



Outputs	<ul style="list-style-type: none"> <li>We wrote tests so that bluetooth feedback could help determine servo position and rotor speed.</li> </ul>
Functionality	Hardware block showing connections that provide the functionality needed for drone operation.

Table IV shows a list of hardware used in our project. For reference a photograph of the completed drone is provided in Figure 3 below the table.

TABLE IV: Current Hardware Implementation

Hardware	Description
Arduino Mega (ATMega2560 Chip)	Our chosen board for this project was based on the need for multiple Rx and Tx ports. The Mega2560 provided all the ports needed for the planned interfacing of the bluetooth, GPS module, and internal measurement unit sensor.
Bluetooth Module	The bluetooth module attached allows the drone to receive commands. The wireless communication provided by the bluetooth is essential for an aircraft. Bluetooth also allows for quick testing of other components.
Servo Motor	We use one servo motor on our drone for controlling the tilt of the propellers.
2 Brushless Motors	Two brushless motors are used to propel the drone forward. Motor's are AC three phase, so they require our DC voltage from the battery to be converted (in this case with an ESC).
2 Electronic Speed Controllers (ESC's)	Module that takes a PWM waveform input and converts it into a three phase power based on pulse width. This three phase signal is used to turn the brushless motors.
14.4 V LiPo Battery	The power source for our project supplies 14.4 V. This higher voltage is needed to drive our electronic speed controllers and motors.

Voltage Regulator	We used a voltage regulator to step our 14.4V from the LiPo battery into 9V in order to be used with our Arduino board.
Internal Measurement Unit (IMU)	The internal measurement unit was used to determine by how many degrees our drone had titled, rolled, or turned. The IMU returned a degree value to our board based on these measurements.
Drone Body	The final drone body was comprised of balsa wood reinforced with another thin sheet of wood placed underneath. This provided a more sturdy base than our first plastic body. A mounting hole was also cut in the center of the body to secure the servo.
Motor Mounts	Attached to the sides of the drone body, we cut two mounts for the motors from a solid plastic sheet. These fastned to the motors with screws and were turned with a metal rod attached to the central servo.
Undercarriage	In order to hold all of the components could safely held, and to provide a center of gravity lower than the rotors, we opted for an undercarriage design. We used a cheap
Landing Platform	In order to assure that no flipping issues persisted through testing, we added a landing platform to the bottom of the drone. Wiffle balls attached to small wood beams provided an ideal platform for testing. Velcro was used so that the platform could be easily removed from the undercarriage.

Figure 3 below shows the completed drone. This includes all of the the hardware listed above, and was the final design rendition.



Figure 3: Completed Tilt Rotor Drone with Landing Pad

## Software

In our current implementation, our software consists of PWM code used to drive a servo and motors, as well as bluetooth communication. The electronic speed controllers (or ESC's) also use a PWM waveform in order to adjust speed going to each brushless motor. In this way we were able to use different pulse widths for both or servo and motors, and could adjust them as needed for direction and speed. We also have implemented a software PID controller to help stabilize the drones flight. For our final code we implemented several free libraries not built into the Arduino IDE. The libraries include a set of AHRS libraries provide by Adafruit to get pitch, roll, and yaw data from the IMU as well as a free PID library provided by Arduino.

For the servo, the PWM signal was sent in order to tilt the rotor mounts. The pulse width sent to the servo corresponds to a servo position, which moves the central pipette, which acts to turn the servos. We used the arduino map function in order to set the lower and upper bounds in terms of degrees rather than a 0 to 1024 range. After centering the servo at 22 degrees, we decided to make the minimum and maximum 15 and 30 degrees of tilt respectively.

For the rotors, the electronic speed controls, or ESC's, take a PWM angle and translate that angle into a three phase signal used to drive the motors. The ESC's take the input PWM and then use that signal to turn on each phase of the motors. By varying the duty cycle of the PWM, the RPM of the brushless motors may be increased by increasing the duty cycle.

The bluetooth communication is done via the serial Rx and Tx pins on the arduino and is used to send and transmit data from the drone. Bluetooth is important since communication is needed to set any flight patterns, or even to manually control or test the drone.

To allow the drone to attempt self-stabilizing flight, we also implemented a software PID controller. The simple model for how the PID works is shown in Figure 4.

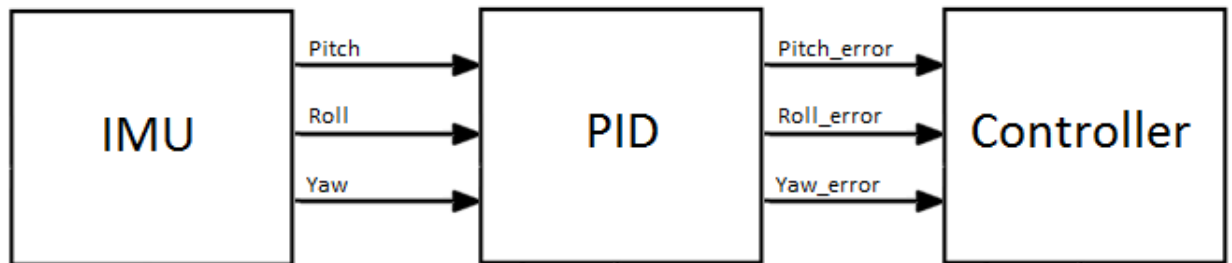


Figure 4: PID Model

The PID worked by taking in pitch, roll, and yaw data from the IMU sensor. We gave a setpoint of zero to the PID for pitch, roll, and yaw values in order to keep the drone normal to gravity. The PID controller then calculated the error based on its proportional, integral, and derivative gains and sent that error to the control block in the code that interpreted the error as servo position and rotor speed.

An overall software flow diagram is given in Figure 5 below. The flow diagram shows the process our drone uses in order to achieve stable flight and to receive Bluetooth commands for increasing altitude and updating orientation manually. First the program starts and initializes all of its needed objects and variables. It then pulls orientation data from the IMU sensor which it then inputs into the software PID controller. Once the PID controller outputs the error, that error is translated into servo angles and rotor speeds. It then checks for a Bluetooth override to see if it should behave differently than by using the PID error values. Finally it updates the speed of the rotors and the servo angle and repeats itself starting by getting another set of orientation values from the IMU.

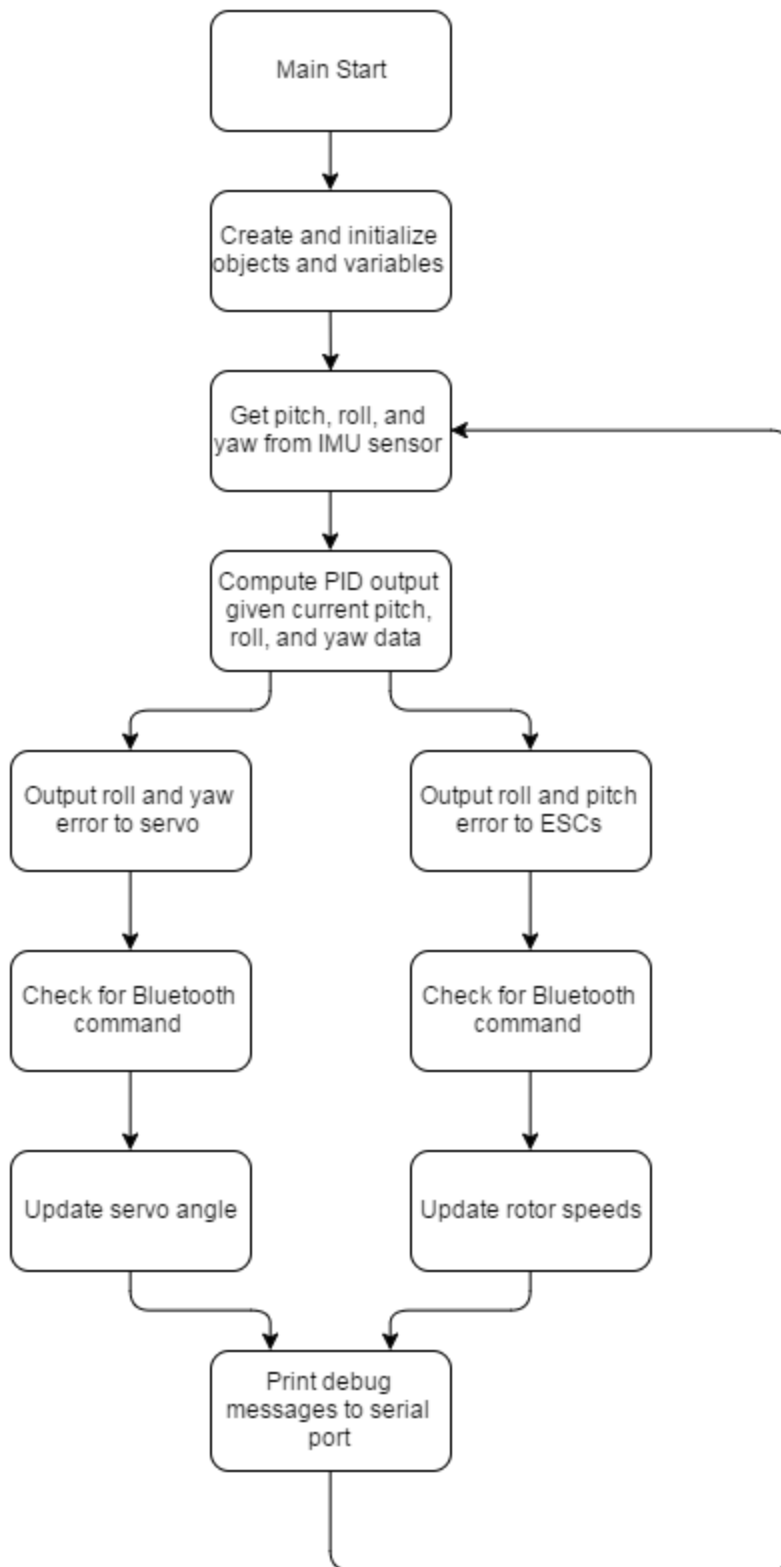


Figure 5: Overall software flow diagram

## Testing

To test our drone we decided that it would be best to test the individual components before integrating them into the final design. Below are tests of different modules of our system. Also included are the engineering specifications that each test covered or indirectly tested. Below the tests, Table V lists our marketing requirements and their corresponding engineering specifications, and if those corresponding specifications were met.

1. **Servo Positioning Test** - The position of the servo in our design dictates where the propellers face and thus where our drone will fly. In order to test and calibrate the servo to the correct position we used a potentiometer connected to the servo and read back the angle in the Arduino sketch terminal. This allowed for easy angle adjustment and calibration. This code is included in the appendix of this report.

### Engineering Specifications Tested:

- Battery: Battery was able to power the servo and arduino through all tests.
- Bluetooth: Servo responded to direct commands through bluetooth.

2. **Motor and ESC Test** - To test our motors and ESC's we hooked up a bench power supply to the input of the ESC's. We then had to determine the direction each motor was spinning. This could have gone either way as the ESC's three output wires are not labeled. The wire combinations switch the rotor direction, so after manual testing we labeled the wires for correct clockwise (first) and counterclockwise (second) operation of each motor.

### Engineering Specifications Tested:

- Battery: Our battery was able to provide over 15 minutes of power to motors.
- Bluetooth: Motors were primarily tested using direct control from bluetooth module.

3. **Bluetooth Test** - To test our bluetooth module we wrote data to our Arduino from a computer terminal and echoed the data back to the same window. Later on we integrated the bluetooth into the whole system, and tested by moving the servo and giving the motors power.

### Engineering Specifications Tested:

- Battery: The battery was able to power our board and the bluetooth module.
- Bluetooth Control: While not able to receive GPS commands, our bluetooth module was able to receive manual commands given to the drone; tested over 30 feet.

4. **Hover Test** - Our most intensive test before the Beta demo was our hover test. For this the drone was required to leave the ground of its own power. To do this we used the bluetooth module to first straighten the servo and then gave the motors enough power to leave the ground. We were able to achieve lift, however the drone cannot be controlled well enough manually to sustain hovering for long. A picture we were able to take while the drone was hovering is included in Figure 6 below.

**Engineering Specifications Tested:**

-Drone weighs less than 5 pounds. Takeoff possible at current weight of 2.5 lbs.



Figure 6: Drone in Hovering Test

5. **Tests Not Performed** - Some of the original engineering specifications still needed to be added and tested.
  - a. The GPS location test. While we were able to achieve control over bluetooth, the GPS functionality was never added.
  - b. Added memory card and camera functionality. We ran out of time before implementing a camera into the drone undercarriage. However, given the power of the motors, we believe that this would have been possible.
  - c. Flying within 10 feet of people without danger. Due to the strength of the motors, we were unable to command enough control of the drone to safely pilot around bystanders. With a more powerful servo or having two servos total may solve this.

- 6. No Test Needed Engineering Specifications** - These specifications did not need a direct test to determine if they had been met.
- Size - Drone did fit within a 2'x6'x6' box.
  - Cost - Our drone was produced for around \$300, which was above our ideal cost of \$150, however, it was still under our sell price of \$475.99
  - Legal Flight Height - Due to bluetooth range and control constraints, our drone could only fly at most 5 feet high and 40 feet from the user.

TABLE V: Current Requirements and Expectations Met

Marketing Requirements	Engineering Specifications	Met?
1,2	The drone must weigh less than 5 lbs.	YES
6,7	The drone must be capable of flying autonomously over a GPS defined flight path.	NO
3	The drone must have a built on SD card with at least 16 Giga Bytes of memory.	NO
2	The drone must fit within a 2'x6"x6" rectangle.	YES
5	The price to produce one unit must not cost more than \$150.	YES
4	The drone must have a camera with at least a 10 Mega pixel camera.	NO
8	The drone must have a rechargeable 14V battery that powers the drone in flight for 15 minutes.	YES
4	The drone must be capable of taking photos of the ground.	NO
9	The drone must be capable of flying within 10ft of people without causing any harm.	NO
10	The drone must not fly higher than the legal altitude for hobby drones.	YES
11	Must be able to communicate via Bluetooth to receive GPS coordinates.	YES
<b>Marketing Requirements</b>		
1. Lightweight		



2. Portable
3. SD Card
4. Camera
5. Low-Cost
6. Autonomous
7. GPS Capabilities
8. Rechargeable Batteries
9. Safe to Use
10. Legal
11. Bluetooth Compatible

### **Conclusion and Future Work**

In terms of efficiency our design worked as intended. We were able to achieve lift with our current design weight and also now can remotely control the drone via bluetooth. Power to the servo and motors all work as intended, and our LiPo battery worked during testing.

Future implementations would include the IMU, or inertial measurement unit, to provide autonomous stabilization, and also the implementation of the GPS module. These two modules would allow the drone function independently of user control, as well as perform via bluetooth commands.

### **REFERENCES**

[1] R.M. Ford and C.S. Coulston, *Design for Electrical and Computer Engineers*, McGraw-Hill, 2008.

[2] J. Villasenor, "What Is a Drone, Anyway?", *Scientific American Blog Network*, 2016. [Online]. Available: <http://blogs.scientificamerican.com/guest-blog/what-is-a-drone-anyway>. [Accessed: 14- Mar- 2016].

[3] BOEING, "Boeing: Historical Snapshot: V-22 Osprey Tiltrotor", *Boeing.com*, 2016. [Online]. Available: <http://www.boeing.com/history/products/v-22-osprey.page>. [Accessed: 14- Mar- 2016].

[4] R. L. Wilson, "Ethical issues with use of Drone aircraft," *Ethics in Science, Technology and Engineering, 2014 IEEE International Symposium on*, Chicago, IL, 2014, pp. 1-4.

- [5] A. Harrington, "Who controls the drones?", *Engineering & Technology*, vol. 10, no. 2, pp. 80-83, 2015.
- [6] P. Ross, "Open-source drones for fun and profit", *IEEE Spectr.*, vol. 51, no. 3, pp. 54-59, 2014.
- [7] T. Diaz, "Lights, drone... action", *IEEE Spectr.*, vol. 52, no. 7, pp. 36-41, 2015.
- [8] R. Sparrow, "'Just Say No' to Drones", *IEEE Technol. Soc. Mag.*, vol. 31, no. 1, pp. 56-63, 2012.
- [9] D. Schneider, "Open season on drones?", *IEEE Spectr.*, vol. 51, no. 1, pp. 32-33, 2014.
- [10] "Autopilot and TV-equipped aerial drone", *Electr. Eng.*, vol. 75, no. 8, pp. 1956.

## Appendix A

### Arduino Sketch Code - Servo Calibration

```

#include <Servo.h>
Servo centralServ;
int throttlePin = 0;
void setup()
{
  centralServ.attach(11);
  Serial.begin(9600);
}
void loop()
{
  int throttle = analogRead(throttlePin);
  int throttle_centralServ = map(throttle, 0, 1023, 15, 30);

  if(throttle_centralServ < 15 || throttle_centralServ > 30)
  {
    centralServ.write(25);
  }
  else
    centralServ.write(throttle_centralServ);

  Serial.print(throttle_centralServ);
  Serial.print("\n");
}

```

### Arduino Sketch Code - Final Code

```

#include <PID_v1.h>
#include <SoftwareSerial.h>
#include <Servo.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_LSM303_U.h>
#include <Adafruit_BMP085_U.h>
#include <Adafruit_Simple_AHRS.h>

// Create sensor instances.
Adafruit_LSM303_Accel_Unified accel(30301);
Adafruit_LSM303_Mag_Unified mag(30302);
Adafruit_BMP085_Unified bmp(18001);

```

```
// Create simple AHRS algorithm using the above sensors.
Adafruit_Simple_AHRS    ahrs(&accel, &mag);

// Update this with the correct SLP for accurate altitude measurements
float seaLevelPressure = SENSORS_PRESSURE_SEALEVELHPA;

//Define Variables we'll be connecting to the PID object
double Setpoint, Input, Output, val;

// orientation object for AHRS
sensors_vec_t orientation;

// Create PID object
PID myPID(&Input, &Output, &Setpoint, 100, 0, 0, DIRECT);

// Servo to control rotor position
Servo centralServ;

// Electronic Speed Controllers(ESC) for controlling rotor speed
Servo esc1;
Servo esc2;

// Variables used to change ESC speed and servo position
int throttle_esc1 = 0;
int throttle_esc2 = 0;
int throttle_centralServ = 25;

// Software Serial Port for Bluetooth interface
SoftwareSerial mySerial(12, 13); // RX, TX

// Byte for reading characters on the serial port
char myByte;

void setup()
{
  // Initialize the sensors.
  accel.begin();
  mag.begin();
  bmp.begin();

  // Attach the ESCs and servos to Arduino pins as well as initialize values.
  esc1.attach(6);
  esc1.write(0);
```

```
esc2.attach(7);
esc2.write(0);
centralServ.attach(4);
centralServ.write(25);

// Open serial communications and wait for port to open:
Serial.begin(115200);
while (!Serial) {
  ;
}

// set the data rate for the SoftwareSerial port
mySerial.begin(115200);

// Initialize AHRS object for initial position data
ahrs.getOrientation(&orientation);

// Use initial heading data for the PID setpoint
Setpoint = orientation.heading;

// Set the saturation values of the PID object
myPID.SetOutputLimits(-255, 255);

// Set PID to Automatic and set Kp, Ki, and Kd values
myPID.SetMode(AUTOMATIC);
myPID.SetTunings(10, 1, 0);
}

void loop() // run over and over
{
  // Compute PID output
  myPID.Compute();

  // Map val to which can be between -255 to 255 to go between values of 13 and 29
  val = map(Output, -255, 255, 13, 29);

  //dif(val <= 18 || val >= 24)
  //{
  // throttle_centralServ = (int)val;
  //}

  //else{
  // throttle_centralServ = 22;
```

```

//}

// Debug prints
Serial.print((int)val);
Serial.print(" ");
Serial.print(orientation.heading);
Serial.print(" ");
Serial.print(Setpoint);
Serial.println("");

// Get orientation using AHRS object
if (ahrs.getOrientation(&orientation))
{
  // Set input of PID to current heading
  Input = (double)orientation.heading;
  /* 'orientation' should have valid .roll and .pitch fields */
  // Serial.print(F("Orientation: "));
  // Serial.print(orientation.roll);
  // Serial.print(F(" "));
  // Serial.print(orientation.pitch);
  // Serial.print(F(" "));
  // Serial.print(orientation.heading);
  // Serial.println(F(""));
}

// Calculate the altitude using the barometric pressure sensor
sensors_event_t bmp_event;
bmp.getEvent(&bmp_event);
if (bmp_event.pressure)
{
  /* Get ambient temperature in C */
  float temperature;
  bmp.getTemperature(&temperature);
}

// Conditionals allowing for Bluetooth control of rotor speed and servo position
if (mySerial.available())
{
  myByte = mySerial.read();
  if(myByte == 'w')
  {
    mySerial.write('w');
    throttle_centralServ--;
  }
}

```

```
}  
if(myByte == 'i')  
{  
  mySerial.write('i');  
  throttle_centralServ++;  
}  
  
if(myByte == 'a')  
{  
  throttle_esc1-=3;  
}  
  
if(myByte == 'd')  
{  
  throttle_esc1+=3;  
}  
  
if(myByte == 's')  
{  
  throttle_esc1 = 0;  
}  
  
if(myByte == 'j')  
{  
  throttle_esc2-=3;  
}  
  
if(myByte == 'l')  
{  
  throttle_esc2+=3;  
}  
  
if(myByte == 'k')  
{  
  throttle_esc2 = 0;  
}  
}  
  
//int throttle = analogRead(throttlePin);  
//int throttle_centralServ = map(throttle, 0, 1023, 15, 30);  
  
if(throttle_centralServ < 12)  
{
```

```
    centralServ.write(12);
}
else if(throttle_centralServ > 30)
{
    centralServ.write(30);
}
else
{
    centralServ.write(throttle_centralServ);
}

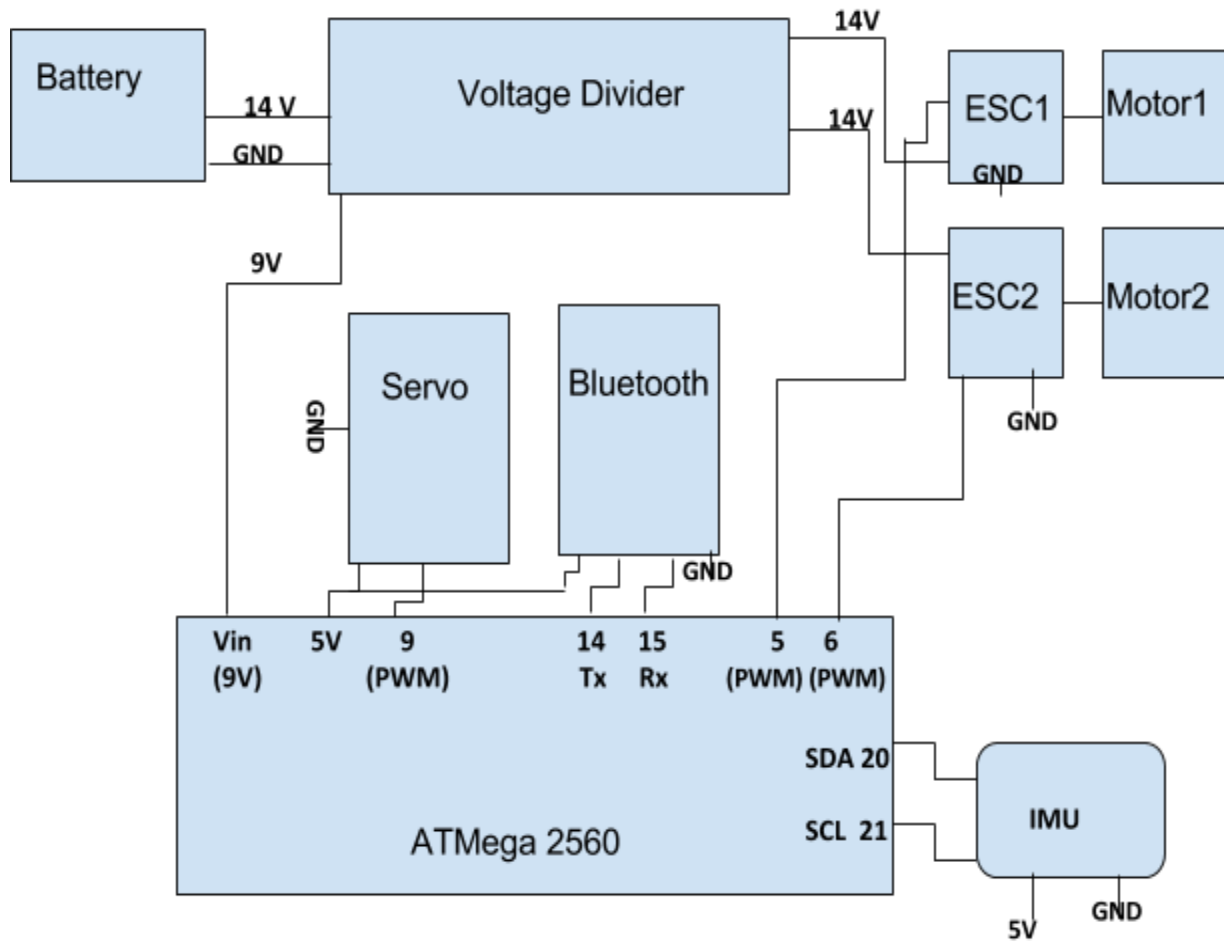
if(throttle_esc1 < 0)
{
    esc1.write(0);
}
else if(throttle_esc1 > 180)
{
    esc1.write(180);
}
else
{
    esc1.write(throttle_esc1);
}

if(throttle_esc2 < 0)
{
    esc2.write(0);
}
else if(throttle_esc2 > 180)
{
    esc2.write(180);
}
else
{
    esc2.write(throttle_esc2);
}

    delay(100);
}
```



## Appendix B: Hardware Pin Diagram



## APPENDIX C — ANALYSIS OF SENIOR PROJECT DESIGN

Please provide the following information regarding your Senior Project and submit to your advisor along with your final report.

Attach additional sheets, for your response to the questions below.

**Project Title: Tilt-Rotor Drone**

**Student's Name: Benjamin Stone**

**Student's Signature: BS**

**Student's Name: Zachary Crandall**

**Student's Signature: ZC**

**Advisor's Name: Bridget Benson**

**Advisor's Initials: BB Date: 06/09/15**

### • Summary of Functional Requirements

*Describe the overall capabilities or functions of your project or design. Describe what your project does. (Do not describe how you designed it).*

Our project is a two rotor drone capable of receiving commands over bluetooth in order to fly based on user inputs. The project has an onboard internal measurement unit that calculates the pitch, roll, and heading of the drone. This allows for the drone to correct its orientation autonomously. The tilt-rotor drone must have a long lasting battery, and be lightweight and portable. For the general customer, the drone must be affordable with a design to maximize the power used while in flight.

### • Primary Constraints

*Describe significant challenges or difficulties associated with your project or implementation. For example, what were limiting factors, or other issues that impacted your approach? What made your project difficult? What parameters or specifications limited your options or directed your approach?*

Our main challenge was balancing the drone in mid-air. Early on in hover testing we encountered a large problem with spin. This was caused by a severe limiting factor in the strength of our onboard servo. This equipment limitation made the project extremely difficult in that we could not correct the drone from over spinning after the rotors tilted to a certain point. This would cause our drone to lose control and oftentimes crash, leading to broken propeller wings that had to be replaced.

The project parameter that limited us the most would be in the low-cost of the system, which prevented the purchasing of better equipment. We set a desired cost of \$150.00 for the entire drone, however we ended up spending over \$250.00 for the final product. This cost parameter directed us to try and save on parts, however if we had a more powerful servo, we may not have encountered as many issues.

Other challenges that we faced with this project include: design, interface, and user-friendliness. The design challenges mostly came in the form of the customer whom we wanted to serve with this project. We had to decide whether we should design the drone for someone trying to use it for business purposes, or for the general hobbyists that wants an entry level drone to experiment with. For interface, we needed to come up with a way to meet the 21<sup>st</sup> century smartphone users while keeping in mind the older generation that may need a different kind of interface when operating the system. Finally, we wanted the design to be easy to use, lightweight, and to be small/portable. We anticipate running into

challenges trying to keep the drone light and efficient while trying to keep the costs down. My final decisions on customers and specifications can be seen in Chapter II.

- **Economic**

- *What economic impacts result? Consider:*

*Human Capital – What people do.*

*Financial Capital – Monetary instruments.*

*Manufactured or Real Capital – Made by people and their tools.*

*Natural Capital – The Earth's resources and bio-capacity.*

From an economic standpoint, if the project were sold it would be done so primarily as a hobbyist drone.

- *When and where do costs and benefits accrue throughout the project's lifecycle?*

The cost of the project is immediate, as the parts are bought beforehand and a total product is completed quickly. The benefit is also realized at the beginning of the project lifecycle, as the product is sold with the total paid upfront. This makes the project viable as long as the price outweighs the cost.

- *What inputs does the experiment require? How much does the project cost? Who pays?*

*Original estimated cost of component parts (as of the start of your project).*

*\$150.00*

*Actual final cost of component parts (at the end of your project)*

*\$270.00*

*Attach a final bill of materials for all components.*

*Attached at conclusion of report.*

*Additional equipment costs (any equipment needed for development?)*

We had one battery Li-Po battery charger, that was needed for testing, however that was only a cost of \$25.00

- *How much does the project earn? Who profits?*

The project may earn the price and cost difference discussed below (\$175.99) for each unit sold. This profit would be then based on the number of unit the product creators were able to sell. The product creators profit.

- *Timing*

*When do products emerge? How long do products exist? What maintenance or operation costs exist?*

*Original estimated development time (as of the start of your project), as Gantt or Pert chart*

*Actual development time (at the end of your project), as Gantt or Pert chart.  
What happens after the project ends?*

With regard to economic impacts, the tilt-rotor drone affects human capital, financial capital, and natural capital. It affects human capital in that it adds to humans learning how to use and maintain drones. Though the drone is autonomous, the user can develop an understanding of how the drone operates and may gain a respect for the convenience of drone technology. For financial capital, people can use this system to generate income for themselves. This comes in the form of land surveying, photography, or any other legal applications that someone can come up with for a small business using drones. And for natural capital, the drone uses power which costs natural resources. The batteries, being rechargeable, have a solar charging option which can make use of the wasted energy that we receive from the sun.

Over the product's lifetime, we plan on it generating between \$500,000 and \$600,000 in revenue. We plan on selling most of the products during the first 2 to 3 years, and we estimate the product dying out by the 5<sup>th</sup> year. Each year we plan on the product raising \$120,000 in sales. And more money is made by individuals and businesses wanting the rights to do things with the product. Overall, we expect the cost of the project to be \$300,000; leaving about \$200,000 in personal profit.

Most of the costs associated with the product arise during the 1<sup>st</sup> year of the product's lifetime. This comes in the form of returns and repairs. Users pay for the inputs of the project, which are power. Users who make use of their 1-year warranty, sending back their products for repairs also add to costs of the project.

Once the project finishes, we plan on starting another business venture.

**Gantt Chart of Project Development**

Task Name	Start	Finish	Hours of Labor [hrs]	Duration [days]	Feb., 2016	Mar., 2016	Apr., 2016	May., 2016	Jun., 2016
1: Decide on Design	2/15/2016	3/2/2016	34	17					
1.1: Generate Bill of Materials	2/15/2016	2/19/2016	10	5					
1.2: Order Parts	2/20/2016	2/23/2016	8	4					
1.3: Test Individual Parts	2/24/2016	3/2/2016	16	8					
2: Build First Prototype	3/3/2016	3/26/2016	48	24					
2.1: Write Sensor Code	3/3/2016	3/10/2016	16	8					
2.2: Construct System	3/11/2016	3/18/2016	16	8					
2.3: Test System	3/19/2016	3/28/2016	16	8					
3: Build Final System	3/27/2016	4/30/2016	70	35					
3.1: Redesign Based on Prototype Failures	3/27/2016	4/1/2016	12	6					
3.2: Update Bill of Materials	4/2/2016	4/4/2016	6	3					
3.3: Order Additional Parts	4/5/2016	4/7/2016	6	3					
3.4: Test new parts	4/8/2016	4/11/2016	8	4					
3.5: Write additional code	4/12/2016	4/18/2016	10	5					
3.6: Construct System	4/17/2016	4/21/2016	10	5					
3.7: Test System	4/22/2016	4/30/2016	18	9					
4: Refinement and Report	5/1/2016	6/3/2016	72	36					
4.1: Compile Test Results	5/1/2016	5/11/2016	22	11					
4.2: Write Report	5/11/2016	6/3/2016	50	25					
<b>Total Hours</b>	<b>224</b>								
<b>Total Days</b>	<b>112</b>								

**• If manufactured on a commercial basis:**

- *Estimated number of devices sold per year*
- *Estimated manufacturing cost for each device*
- *Estimated purchase price for each device*
- *Estimated profit per year*

- *Estimated cost for user to operate device, per unit time (specify time interval)*

If the tilt rotor drone was manufactured on a commercial basis, we estimate selling 1,000 units per year for five years. The cost per device should be near \$475, and the manufacturing cost should stay at or near \$300. After start up and maintenance costs, we expect to make \$175,000 in the first year of production and \$190,000 per year in the years following. To operate the device, the user spends near \$100 in utilities per year.

- **Environmental**

- *Describe any environmental impacts associated with manufacturing or use, explain where they occur and quantify.*

As with any project with plastic parts and batteries, our project would produce a small amount of waste at the end of its life cycle, however during use no feweffects are produced.

- *Which natural resources and ecosystem services does the project use directly and indirectly?*

*The project uses a small amount of electricity, so no major ecological impact with use.*

- *Which natural resources and ecosystem services does the project improve or harm?*

*N/A (too small of a scale to impact)*

- *How does the project impact other species?*

*No.*

For environmental impacts, it is important to examine that the drone is primarily constructed from plastic. This means that the environment experiences the burning of fossil fuels to produce the system. If the system is thrown away, then the landfills are impacted with more petroleum based waste. Indirectly, this product uses Indium in the electronic circuitry. This resource is depleting, so ideally a better material should be used in the future. This system can, however, make use of the sun's energy by being charged via a solar panel. This helps in making use of some of the sun's wasted energy and in promoting the use of solar panels which is a more environmentally friendly option than nuclear reactors.

- **Manufacturability**

- *Describe any issues or challenges associated with manufacturing.*

The biggest challenge associated with manufacturing is the drone's chassis. The chassis needs to be constructed using a strong, yet lightweight design. This means using either a sturdy plastic or possibly Aluminum which require specialized equipment (laser cutters, CNC mills, etc.) to manufacture on a large scale. Another challenge is sensors. Currently my sensors come from different locations, which means

that we must pay shipping fees to several different companies. This increases the manufacturing cost which impacts the cost of the final product. Finally, the drone is made of multiple units that need to come together at some point. This presents an assemble problem that can be solved using robots or people, but this adds another expense to the production. Also using robots which is the more efficient way, causes a need for maintenance and technicians which also add production costs.

#### • Sustainability

- *Describe any issues or challenges associated with maintaining the completed device, or system.*
- *Describe how the project impacts the sustainable use of resources.*
- *Describe any upgrades that would improve the design of the project.*
- *Describe any issues or challenges associated with upgrading the design.*

To make this system sustainable, a rechargeable batteries are required. Disposable batteries would add to the landfill problem, and would promote waste. To upgrade the design to be more sustainable, we could implement solar panels for power and try to create a better controller for managing power more efficiently. The challenge for implementing a solar panel is that the battery for the drone could take up to a day to charge.

#### • Ethical

- *Describe ethical implications relating to the design, manufacture, use, or misuse of the project.*

Following the Utilitarian ethical framework, one could argue on both sides for whether this system is ethical or not. On one side, one could argue that it could be used for small business owners. This adds to the economy and makes the business owner happy as well as everyone affected by the revenue created by the business. On the other side, one could argue the opposite that the small business isn't useful and takes up more space than it is worth. The drone also shows the advancement of technology and encourages hobbyists to make improvements to them and inspires them. But the other argument is that too many drones are being made and are clogging the airspace. With no needed license to operate a drone, this could potentially lead to damage from drone accidents. If this drone were to be mass produced, then supply for this type of drone would go up which would bring demand down. This would make it less expensive for individuals to start working on drones as a hobby and maybe to start a business. But mass producing drones causes many environmental issues including pollution and depletion of natural resources.

Using the IEEE code of ethics, this system meets the full list of requirements. Most importantly, it meets the first requirement in which we accept responsibility for consequences associated with this system. We devote my energy to ensure that the system is "consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment [1]." Also, since we am not an expert in drones, we "seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others [1]."

#### • Health and Safety

The main safety concern for this system is if it were to crash into something or someone. This drone could potentially cause damage to people or property if it were to malfunction. The design attempts to prevent this, but there is no way to avoid all potential accidents. We definitely need a lawyer to write up a document that voids me of responsibility for the misuse of the drone. We have written specifications to comply with these health and safety concerns in Chapter II of this report.

#### • Social and Political

In design and use, this system indirectly adds to the political issue of capital. Since this system can be used to start a small business, the design must take into account the laws regarding such a system. The stakeholders in capitalism are any person living in the US whether or not they subscribe to capitalism. Another stakeholder would be government agencies who decide whether or not to keep capitalism running. In whether or not we should continue making plastic products also comes into play. The stakeholders in this issue are consumers where the product impacts them directly, and small business owners who are affected indirectly. Since this system can be used to start a business, purchasing these drones creates competition which could cause other companies to create even better drones. This system can also benefit certain governmental or educational stakeholders in that it is adding more drones to the world and making people more aware of the usefulness and dangers of drones. Inequities come in the form of benefitting the consumer and negatively affecting the government agencies concerned with the dangers of having unregulated drones in the sky.

#### • Development

For developing this system, we had to research the different kinds of sensors used in this type of application. We researched different kinds of inertial measurement sensors. We also had to do research on drones, as we are not experts in this area. For designing the chassis, we researched different CAD software to find which one best met my needs in terms of features and cost. Finally, we had to do some research into projects already done that are similar to my system. We performed research on both their failures and their successes in order to make my system the best that it can

### Appendix A - Bill of Materials

\*Parts for final system design.

\*\*Amounts rounded to the nearest dollar for simplicity.

Part	Price (in USD \$)	Number Ordered	Total
Arduino Mega	65.00	1	65.00
Chassis (wood)	20	1	20.00
Fittings	varies	varies	10.00 (estimate)
ESC's	25.00	2	50.00
Motors	23.00	2	46.00

<b>Servo</b>	<b>18.00</b>	<b>1</b>	<b>18.00</b>
<b>IMU Module</b>	<b>16.00</b>	<b>1</b>	<b>16.00</b>
<b>Bluetooth Module</b>	<b>8.00</b>	<b>1</b>	<b>8.00</b>
<b>Battery</b>	<b>37.00</b>	<b>1</b>	<b>37.00</b>
		<b>Total System Cost</b>	<b>270.00</b>