

Fall 2011

## The Propulsive Design Aspects on the World's First Direct Drive Hybrid Airplane

Ankit Nanda  
*Embry-Riddle Aeronautical University - Daytona Beach*

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Aerospace Engineering Commons](#), and the [Aviation Commons](#)

---

### Scholarly Commons Citation

Nanda, Ankit, "The Propulsive Design Aspects on the World's First Direct Drive Hybrid Airplane" (2011).  
*Dissertations and Theses*. 110.  
<https://commons.erau.edu/edt/110>

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of Scholarly Commons. For more information, please contact [commons@erau.edu](mailto:commons@erau.edu).

THE PROPULSIVE DESIGN ASPECTS ON THE WORLD'S FIRST DIRECT DRIVE HYBRID  
AIRPLANE

by

Ankit Nanda

A Thesis Submitted to the  
Graduate Studies Office  
In Partial Fulfillment of the Requirements for the  
Degree of Master of Science in Aerospace Engineering

Embry-Riddle Aeronautical University  
Daytona Beach, Florida  
Fall 2011

Copyright by Ankit Nanda 2011

All Rights Reserved

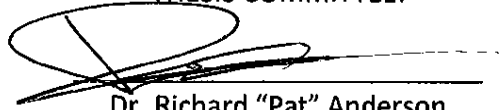
THE PROPULSIVE DESIGN ASPECTS ON THE WORLD'S FIRST DIRECT DRIVE HYBRID  
AIRPLANE

by

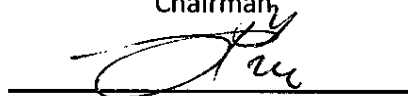
Ankit Nanda

This thesis was prepared under the direction of the candidate's thesis committee chairman, Dr. Richard "Pat" Anderson, Department of Aerospace Engineering, and has been approved by the members of his thesis committee. It was submitted to the Department of Aerospace Engineering and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE:




Dr. Richard "Pat" Anderson  
Chairman



Dr. Jianhua Liu  
Member



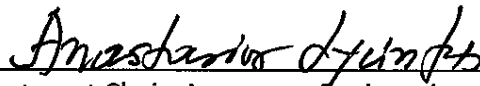
Prof. Charles Eastlake  
Member



Coordinator, MSAE Program

4/6/2012

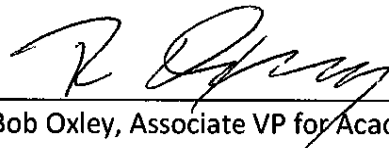
Date



Department Chair, Aerospace Engineering

4/6/12

Date



Dr. Bob Oxley, Associate VP for Academics

4/9/2012

Date

## **ACKNOWLEDGEMENTS**

This thesis has been the culmination of two very long years, with lots of sleepless nights, working on the Eco-Eagle. The project began just on paper, with people questioning if it would ever be a reality and now, not only has that happened, but it is also the first of its kind in the world, thanks to the hard-work and dedication of many students and professors at ERAU.

I would like to begin with thanking Dr. Anderson, who believed in my ability to work on this project. Without his guidance this project would not have been possible. Secondly, I would like to especially thank Mikhael Ponso, for being our brave and trusting test pilot. I have been extremely lucky to be surrounded by professors such as Dr. Liu, Prof. Eastlake and Prof. Griener who were there at the hangar every day assisting us and advising us as to how to proceed in making this dream come true and not letting us back down from any obstacle.

All the people who have been on the Green Flight Challenge team in the past couple of years deserve a special mention, especially for all their volunteer time they put in at the hangar. I want to express my gratitude to Lori Costello, for managing such a huge project and dedicating so much time to the Eco-Eagle. Some of the students I would also like to mention within this team are Donovan Curry, Prateek Jain and Hitesh Patel for being there every day for the long hours alongside me. I would also like to thank Brendon Lyons and Harshad Lalan for being sounding boards for all my ideas. Also, another person who deserves my special thanks is Shirley Koelker. Without her the students at the hangar would have probably not enjoyed working there as much and I personally would have not realized it was ever time to eat food. I would also like to thank all the people and companies that sponsored our project and believed in the ability of the students.

Also, I would like to thank Prof. Carney and her family for reminding me of my goals again and again and being like my family away from my home.

I would like to give a special thanks to my parents and my brother, who have always supported me in every endeavor and have motivated me throughout my life to realize my dreams. Their dedication and hard-work has allowed me to have great role-models in my life and has made me work even harder to achieve my aims in life.

I would like to dedicate this thesis to my family.

## **ABSTRACT**

Author: Ankit Nanda

Title: The Propulsive Design Aspects on the World's First Direct Drive Hybrid Airplane

Institution: Embry-Riddle Aeronautical University

Degree: Master of Science in Aerospace Engineering

Year: 2009-2011

The purpose of this thesis is to design a safe technology demonstrator by implementing a direct drive propulsion system for a gas-electric hybrid aircraft. This system was integrated on the Embry-Riddle Eco-Eagle for the Green Flight Challenge 2011. The aim of the system is to allow the pilot to use the electric motor as an independent power source to fly the aircraft once at cruise altitude, while having a gas engine to allow for higher power capability.

The system was designed to incorporate the motor and the motor control unit provided by Flight Design and Drivetek AG alongside a Rotax 912ULS engine. The hardware is integrated such that the pilot would be able to fly the aircraft with controls similar to conventional general aviation aircraft. This thesis discusses the method of integration of the hybrid powerplant system into a Stemme S-10 and describes the various components of that system.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iv
ABSTRACT.....	v
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
CHAPTER 1: Introduction .....	1
Literature Review.....	2
History of Electric Propulsion in Aviation .....	2
The Green Flight Challenge.....	4
CHAPTER 2: Problem Statement.....	7
Rotax Engine .....	7
Electric Motor .....	13
Electric Motor Performance Characteristics.....	15
Field Oriented Control .....	16
Motor Control Unit .....	17
CAN Bus Communication .....	19
Batteries.....	21
Battery Monitoring Systems .....	22
Communication Boards.....	23
Propeller.....	24
CHAPTER 3: Aircraft System Layout.....	27
The Eco-Eagle Electric Propulsion System .....	27
Battery System Layout .....	31
The Battery Management System .....	36
Pilot Interface System .....	39
Electric Motor Throttle Controller .....	41
Touchscreen Interface .....	42
Graphical User Interface .....	43
CHAPTER 4: Propulsion System Performance.....	49
Propulsion System Efficiency .....	49
Current System Performance.....	50
CHAPTER 5: CONCLUSION AND SUGGESTIONS .....	52

Conclusion.....	52
Future Work and Suggestions.....	53
APPENDIX A: Data Recorded.....	56
APPENDIX B: Electric Propulsion Control System Software.....	60
Battery Management System Communication.....	61
Motor Controller Communication .....	63
CAN Bus Initialization.....	64
Write Function .....	65
Read Function .....	75
CAN Bus Channel Listing .....	79
FPGA Communication .....	86
References .....	88



## LIST OF TABLES

Table 1: GFC Requirements.....	4
Table 2: Rotax 912ULS Specification.....	7
Table 3: Reduced Engine Output RPM.....	10
Table 4: Engine Temperature Limits.....	10
Table 5: Electric Motor Specification.....	14
Table 6: Motor Control Unit Specification.....	18
Table 7: Lithium Iron Phosphate-4 Battery Specification.....	21
Table 8: Propeller Specification.....	24
Table 9: Propeller Performance Table.....	26
Table 10: RPM at Propeller after Reduction.....	30
Table 11: Battery System Components.....	32
Table 12: Electric Motor Throttle Specifications.....	42
Table 13: Touchscreen Interface Specifications.....	43
Table 14: Results from GFC Fuel Efficiency Flight.....	51
Table 15: GFC Results from Speed Flight.....	51
Table 16: List of Data Recorded.....	56
Table 17: Arbitration ID 191 Message.....	67
Table 18: Motor State Definition.....	67
Table 19: Example Transmitted Message for Torque Control.....	73
Table 20: Arbitration ID 610 Message.....	74
Table 21: "Clear Error" Message.....	75
Table 22: CAN Bus Channel Listing.....	79
Table 23: Mode of Operation.....	81
Table 24: Power Module Errors.....	81
Table 25: Hardware Errors.....	83
Table 26: Drive Errors.....	83
Table 27: Drive Warnings.....	84
Table 28: CompactRio Channel Listing.....	87

## LIST OF FIGURES

Figure 1: La France (La France Airship Giclee Print).....	2
Figure 2: Siemens DA36 E-Star Serial Hybrid Aircraft .....	4
Figure 3: Rotax 912ULS .....	7
Figure 4: Rotax 912ULS Engine Performance (Rotax 912ULS DCDI) .....	8
Figure 5: Rotax 912ULS Engine Torque (Rotax 912ULS DCDI) .....	9
Figure 6: Rotax 912ULS Engine Fuel Consumption (Rotax 912ULS DCDI) .....	9
Figure 7: Rotax Engine Cooling System.....	11
Figure 8: Rotax Engine Oil Cooling System (Rotax 912ULS DCDI).....	12
Figure 9: Rotax Cooling System on the Eco-Eagle.....	13
Figure 10: Electric Motor .....	14
Figure 11: Motor Test at Maximum Power Output .....	15
Figure 12: Charging Test with Electric Motor in Generator Mode at 25% Torque .....	16
Figure 13: Motor Control Unit .....	17
Figure 14: MCU Interior Component View .....	19
Figure 15: Lithium Iron Phosphate-4 Cell .....	21
Figure 16: LTC Battery Monitoring Board .....	22
Figure 17: Arduino Mega 2560 .....	23
Figure 18: Experimental MT-Propeller for the Eco-Eagle .....	24
Figure 19: Propulsion System Layout.....	27
Figure 20: Top and Side View of Propulsion System.....	28
Figure 21: Overrunning Clutch (Formsprag Clutch) .....	29
Figure 22: Drivetrain Assembly (Gonitzke, 2010) .....	31
Figure 23: Individual Battery Pack .....	33
Figure 24: Battery Module .....	33
Figure 25: Battery System Connections .....	34
Figure 26: Battery Module Connections .....	35
Figure 27: Wing Cutout for Battery Tray.....	35
Figure 28: Battery Management Commands.....	39
Figure 29: Avionics System .....	39
Figure 30: Electric Motor Control Hardware Layout.....	41
Figure 31: Sliding Potentiometer .....	41
Figure 32: NI TPC-2206 Touchscreen Interface .....	42
Figure 33: Ground Station Motor Test Interface .....	44
Figure 34: Electric Motor Avionics Display .....	46
Figure 35: Rotax Information on the Touchscreen .....	47
Figure 36: Data Recording Tab on Touchscreen .....	48
Figure 37: Eco Eagle in Flight .....	52
Figure 38: LabView Program Block Diagram.....	60
Figure 39: BMS Initialization Sequence.....	61

Figure 40: Closing Main Relays .....	62
Figure 41: BMS Status Indicator.....	63
Figure 42: Enable CAN Bus Communication .....	64
Figure 43: Clear Errors Option .....	66
Figure 44: Enable Motor Switch.....	68
Figure 45: 12V DC to 5V DC Converter.....	69
Figure 46: "Enable Motor" Implementation .....	70
Figure 47: Stay Alive Counter.....	71
Figure 48: Arbitration ID 191 Message Construction .....	72
Figure 49: Throttle Percentage Display.....	73
Figure 50: Clear Errors Case Structure .....	74
Figure 51: Read Data and Update Output Array.....	76
Figure 52: Data Extraction from Output Array .....	77
Figure 53: Data Processing for Channel 390 .....	78
Figure 54: Data Processing for Channel 190 .....	80
Figure 55: Error Handling and Display .....	85
Figure 56: Error Display.....	86

## LIST OF ACRONYMS

ASCII	American Standard Code for Information Interchange
BC	Block Commutated
BLDC	Brushless DC
BMS	Battery Management System
CAFE	Comparative Aircraft Flight Efficiency
CAN	Controller Area Network
EFRC	Eagle Flight Research Center
EMF	Electromotive Force
ERAU	Embry-Riddle Aeronautical University
FOC	Field Oriented Control
FPGA	Field Programmable Gate Array
GFC	Green Flight Challenge
hp	Horsepower
IFSD	In-Flight Shut-Down
LiFePO <sub>4</sub>	Lithium Iron Phosphate
LTC	Linear Technology
MCU	Motor Control Unit
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
mph	Miles per Hour
NASA	National Aeronautics and Space Administration
NiMh	Nickel Metal Hydride
PMSM	Permanent Magnet Synchronous Motor
RTCA	Radio Technical Commission for Aeronautics
SAE	Society of Automotive Engineers

## **CHAPTER 1: Introduction**

The purpose of this thesis is to develop and document the propulsive design aspect for a direct-drive hybrid aircraft. This propulsion system, incorporated in the Eco-Eagle, was designed and developed under the direction of Dr. Richard P. Anderson, at the Eagle Flight Research Center (EFRC) to be Embry-Riddle Aeronautical University's (ERAU) entry into the Green Flight Challenge 2011 (GFC). The challenge required an aircraft to fly a 200 mile course at an average speed of 100 miles per hour (mph) and attain 200 passenger miles per gallon.

The hybrid propulsion system design for the Eco-Eagle allows it to take off powered by a gas engine. At cruise altitude, the gas engine is turned off and level flight is attained by an electric motor.

The Eco-Eagle used a Stemme S-10 airframe as its base. The aircraft's powerplant system comprised of a Rotax 912ULS gas engine and a permanent magnet synchronous motor (PMSM) and a modified propeller in place of the Stemme's regular propulsion system.

Goals for the design of this system included its safe operation while attaining the GFC requirements. Considerations for this included various functioning parameters of the power sources involving maximum power output of each independent source, fuel system location (gas and batteries), system efficiencies, and component integration (example: placement of hardware required for the complete system).

Existing aircraft specifications for the Stemme S-10 resulted in design constraints for modification. The size of the motor and the weight restrictions for the battery system prohibited an electric only propulsion system on this airframe. This hybrid system allows safe flight in the event of an in-flight shut-down (IFSD) of the electric system or requirement of increased power. This also ensures that the output limits on the drive shaft do not exceed the design limits.

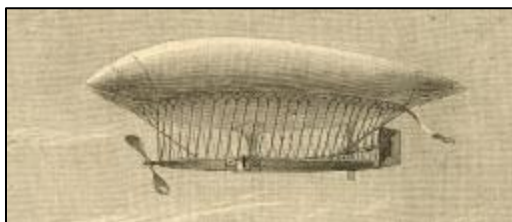
This document shall explain the integration of the components used to develop the direct drive hybrid propulsion system.

## **Literature Review**

### **History of Electric Propulsion in Aviation**

In 1883, the Tissandier Brothers (Albert and Gaston Tissandier) utilized a 1.5 horsepower (hp) battery powered motor to drive a two-bladed propeller for a 37,500 cubic foot airship (Garber, *The Beginnings of the Dirigible*, 2011).

The first airship with the ability to return to its starting point, the LaFrance, flew in 1884 with an electric propulsion system (Garber, *The Beginnings of the Dirigible*, 2011).



**Figure 1: La France (La France Airship Giclee Print)**

Electric propulsion has been implemented in model aircraft since the 1950's (Day, 1983). In the 1990's multiple small aircraft were run on solar power. In 1999, the LF20, created by Lange Flugzeugbau GmbH was flown, powered by nickel-metal hydride (NiMH) cells.

Limitations in battery technology in the early 2000's limited flight distance due to charge capacity and energy density. Advancement in battery technology coupled with better motor designs has since enabled the possibility of longer flights on battery power.

Battery powered aircraft are still limited to the experimental category due to safety concerns related to the system as compared to gas engines. Due to the

generation of high amounts of heat, higher energy density batteries are considered relatively unstable and hence unsafe. In the event of a short-circuit, high energy density batteries have been known to catch fire. A parallel hybrid aircraft allows safety aspects of gas-engines to be combined with the efficiency of battery systems.

In 2003, NASA along with three universities designed and analyzed environmentally friendly propulsion systems for aircraft, utilizing fuel cells. A Cessna 172R was retrofitted with a fuel cell electric motor for the study. It concluded that “an all PEM (Proton Exchange Membrane) power system was not able to directly replace the existing Cessna 172 internal combustion engine with an acceptable level of performance.” The study also determined a hybrid system to be more useful as aircraft require maximum power only for certain flight profiles. The economical aspect of the study showed that fuel cell propulsion systems were not feasible economically as compared to internal combustion engines (Upton, 2006). Due to monetary constraints and energy requirements fuel cells were not used on the Eco-Eagle.

Siemens AG, Diamond Aircraft and EADS collaborated to manufacture a serial hybrid aircraft, which flew in the first half of 2011. A Wankel engine powered an electric motor (70 kW / 93.83 hp) via a generator in addition to a battery power supply, enabling the aircraft to operate the engine at a lower power setting and consuming less fuel (Martin, 2011). A serial hybrid setup requires the motor and the engine to run simultaneously. Shut-down or failure of a single power source results in no power output at the propeller or a lowered supply of output power.



**Figure 2: Siemens DA36 E-Star Serial Hybrid Aircraft**

### **The Green Flight Challenge**

The Green Flight Challenge (GFC) 2011 was sponsored by NASA and Google and organized by the Comparative Aircraft Flight Efficiency (CAFE) foundation at the Charles M. Schulz Sonoma County Airport in Santa Rosa, California. The competition required an aircraft to fly a 200 mile course at an average ground speed of 100 mph attaining, 200 passenger miles per gallon equivalent or better. A total reward of \$1.65 million was offered for the challenge.

The competition required two flights: the first flight measured fuel efficiency, and the second measured average speed.

Listed below are the major requirements for the aircraft design for the competition (The Green Flight Challenge 2011, 2009):

**Table 1: GFC Requirements**

<b>PARAMETER</b>	<b>DESCRIPTION</b>
<b>Range:</b>	200 statute miles, with 30 min. reserve, day VFR at $\geq 4000'$ MSL over non-mountainous, sparsely-populated coastal terrain
<b>Efficiency:</b>	$\geq 200$ Passenger-MPGe energy equivalency



**Speed:**  $\geq 100$  mph average on each of two 200 mile flights

**Minimum Speed:**  $\leq 52$  mph in level flight without stall, power and flaps allowed

**Takeoff Distance:**  $\leq 2000$  feet from brake release to clear a 50 foot obstacle

**Community Noise:**  $\leq 78$  dBA at full power takeoff, measured 250 feet sideways to takeoff brake release

**Passengers** Upright seats with adequate volume for a 6-foot tall, 200 lb adult.

**Wingspan** Must fit inside 44-foot wide hangar for weighing (wingfold is acceptable). Height, length and landing gear footprint limits are defined in Appendix B of the GFC rulebook.

**Vehicle Weight**  $\leq 6500$  lb. with  $\leq 4500$  lb. on main gear and  $\leq 2000$  lb. on nose- wheel or tail-wheel.

**Field of View** Acceptable to FAA licensing authorities and FAA AC25.773-1.

**Control Systems** Must provide dual controls if two or more seats.

**Payload Carried** 200 lbs per seat. Dual pilots if two or more seats.  
200 lbs per seat sandbag ballast in all seats not occupied by pilot/co-pilot.

**Seating Configuration** Tandem seating is allowed, but vehicles with 3 or more seats must place at least 2 seats directly side-by-side. Rapid exit required for all seats.

**Fuel/Energy Use:** Energy consumed: 1 gallon of 87 octane unleaded auto gasoline = 115,000 BTU.

---

<b>Fuels/Energy Allowed</b>	Avgas 100 LL, Jet-A, diesel, unleaded auto gasoline, bio-fuels, H2, synthetics, electricity.
<b>Flightworthiness:</b>	Valid US FAA Airworthiness Certificate for unrestricted day VFR flight in the Continental United States, proof of structural limits (Appendix K), ASTM 2316 compliant vehicle ballistic parachute and all applicable inspections.
<b>Pilot Qualifications</b>	FAA qualified for operating experimental aircraft, with current medical, BFR, 500 flight hours total and 10 flight hours in make & model
<b>Eligibility</b>	Team leader must be a U.S. Citizen or permanent resident

---

## CHAPTER 2: Problem Statement

A combination of the GFC requirements, monetary constraints and pilot safety determined a modified Stemme S-10 to be utilized for the Eco-Eagle. The components for the project were sponsored by various people outside the university. Airworthiness for the aircraft was determined on the basis of the development of a stable system that would not cause any harm to the pilot and allow him to control the aircraft at all times, even in the event of an emergency. The aircraft incorporated a Rotax 912ULS 100 hp gas engine with a 40 hp permanent magnet synchronous motor. The propulsion system was installed in the airplane with minimum modification to the aircraft structure to maintain the original airframe integrity.

Components for the hybrid propulsion system for the Eco-Eagle are defined below:

### Rotax Engine

Take-off for an aircraft is a high power consumption phase in a flight profile. The Eco-Eagle used a 100 hp (75 kW) Rotax 912ULS to meet this performance requirement.



Figure 3: Rotax 912ULS

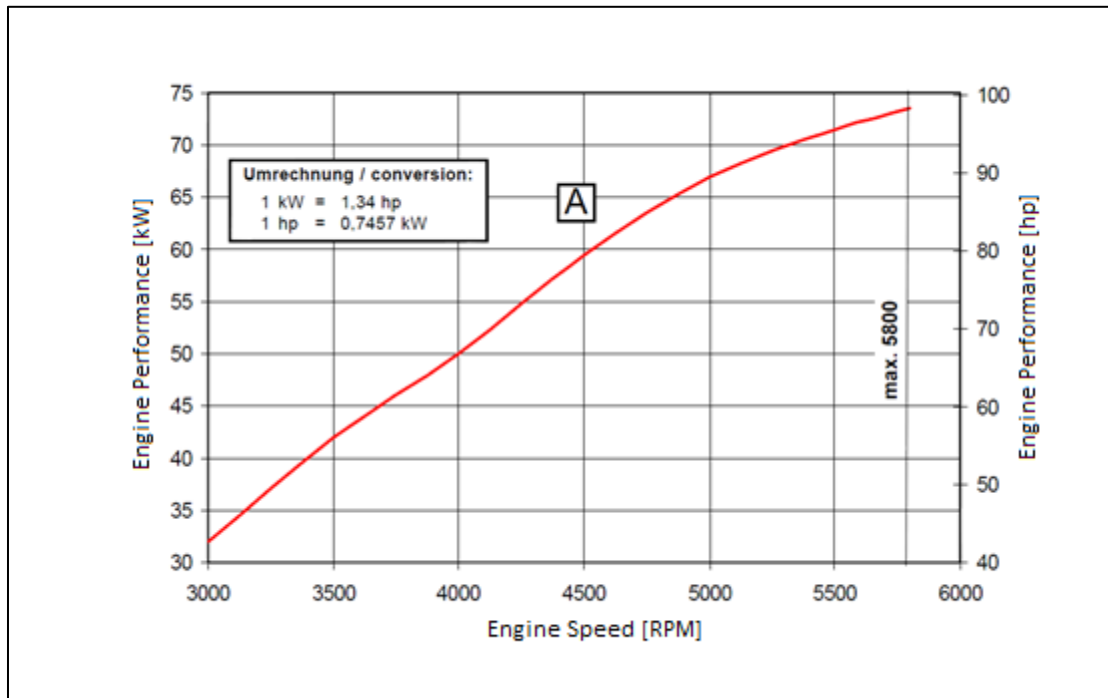
Table 2: Rotax 912ULS Specification

PARAMETER	VALUE
Dry Weight*	134 lb (61 kg)
Maximum Power Output	100 hp

Fuel Type	AVGAS 100LL or minimum of 91 octane automotive fuel
Direction of Rotation	Clockwise from pilot's perspective

\*Note: The weight listed above does not include the weight for the engine accessories, such as the external alternator, the vacuum pump and the overloaded clutch.

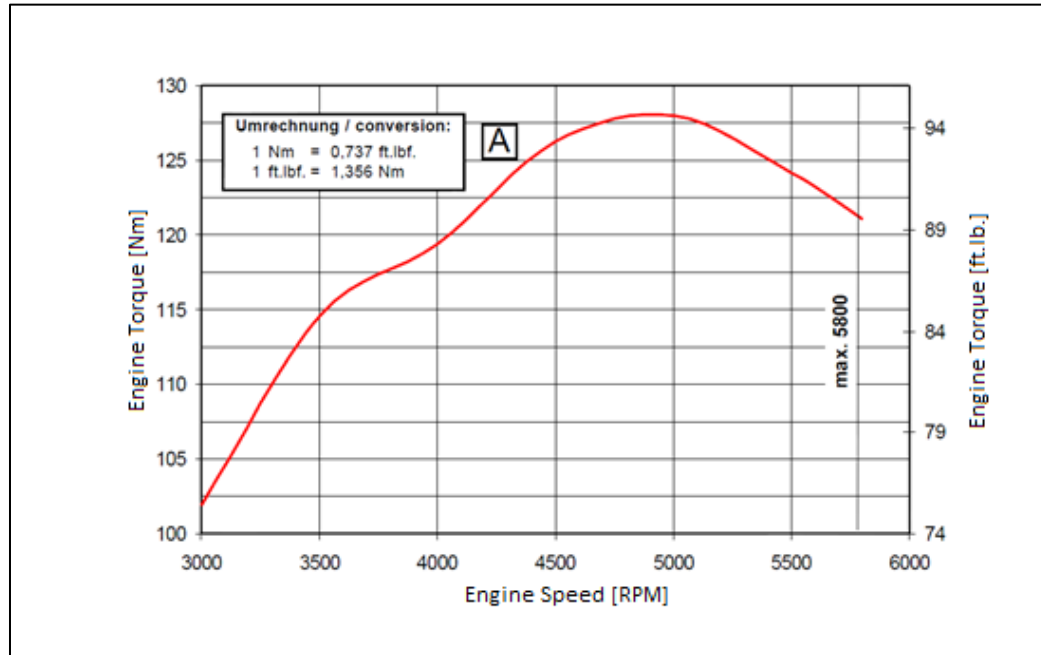
The Rotax 912ULS is an uncertified four-stroke, four horizontally opposed cylinders, air/liquid (50:50 mixture) cooled gas engine developed for recreational aircraft. A minimum of 91 octane automotive fuel is used to operate this engine (Rotax 912ULS Specifications).



**Figure 4: Rotax 912ULS Engine Performance (Rotax 912ULS DCDI)**

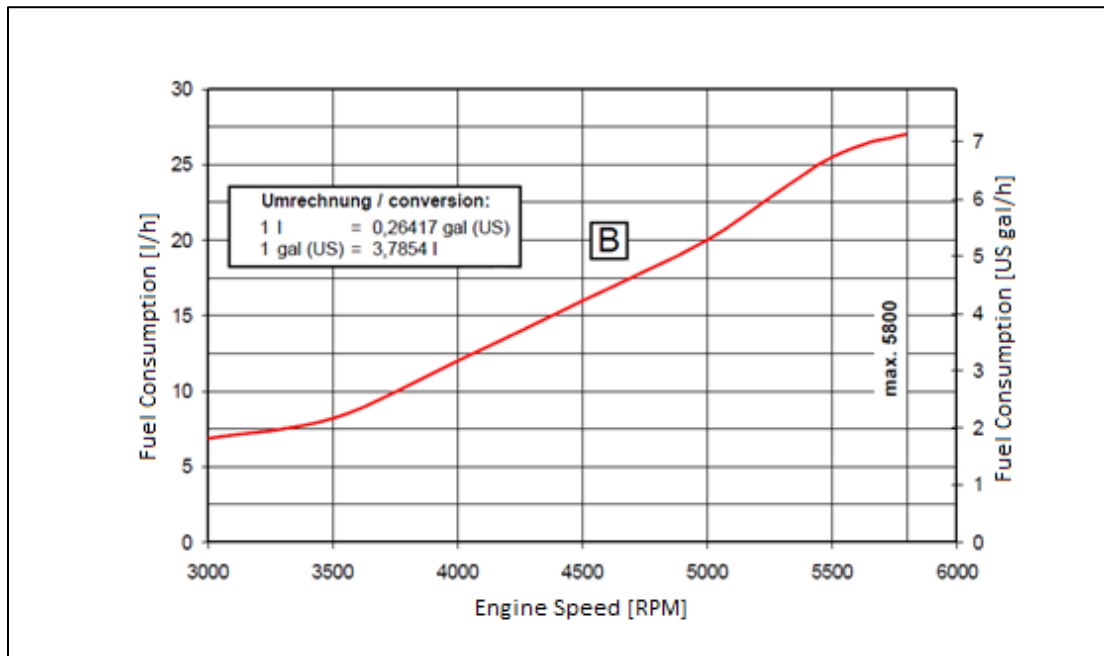
Figure 4 depicts the maximum safe performance rating for the engine. With a tuned exhaust installed, a maximum safe output of 5800 revolutions per minute (RPM) can be achieved for 5 minutes. The maximum sustained power for operation of the engine within its limits, as shown in Figure 4, is 95 hp (69 kW) at 5000 RPM.

The engine has a compression ratio of 10.5:1. A higher compression ratio implies higher mechanical energy output for a given amount of air-fuel mixture. The ULS variant of the Rotax 912 delivers the highest torque in its series due to the highest compression ratio, with a higher fuel burn to provide the extra power.



**Figure 5: Rotax 912ULS Engine Torque (Rotax 912ULS DCDI)**

The engine can produce a maximum output torque of 94 ft-lb or 128 N-m at 5100 RPM, as shown in Figure 5. Above this RPM the torque produced decreases due to restrictions within the intake and exhaust flow for the engine. The amount of heat and friction generated with the higher speeds increases, causing a decrease in the mechanical energy output.



**Figure 6: Rotax 912ULS Engine Fuel Consumption (Rotax 912ULS DCDI)**

From Figure 6, shown above, at 5100 RPM the Rotax 912ULS burns approximately 5.5 gallons of fuel per hour, which exceeded the design constraints for the GFC. An electric motor was added to the propulsion system to alleviate the fuel burn. Compared to a gas engine, an electric motor is more efficient and has lower equivalent fuel consumption. Due to size and weight restrictions for the aircraft propulsion system, an electric motor could not provide enough power alone for take-off.

At 5800 RPM (maximum RPM) the engine burns about 7 gallons of fuel per hour. By tuning the engine the, performance of the engine can be altered by increasing or decreasing the fuel flow and synchronizing the carburetors.

Efficiency of a propeller at these high speeds decreases. The output rotational speed from the engine is reduced via an integrated reduction gearbox, at a ratio of 2.43:1, with a slipper clutch. The output RPM at the different speeds mentioned above varies as shown below in Table 3.

**Table 3: Reduced Engine Output RPM**

ENGINE RPM	OUTPUT RPM
5100	≈ 2098
5500	≈ 2263
5800	≈ 2386

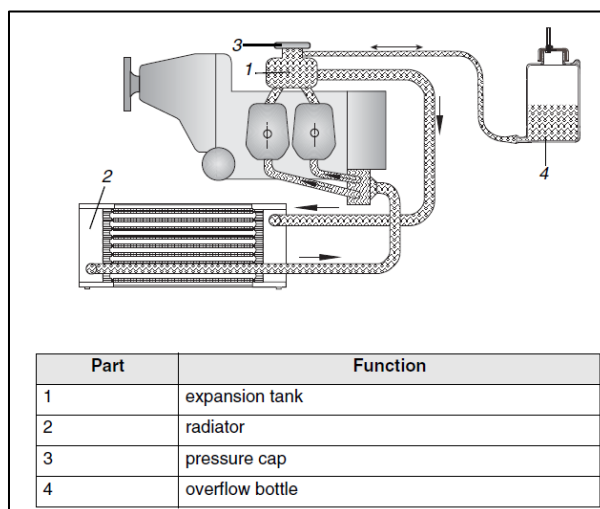
Safe operation of the engine requires the temperature of the cylinder heads and oil to be maintained within a certain limit, as shown below in Table 4.

**Table 4: Engine Temperature Limits**

ENGINE PART	TEMPERATURE
Cylinder Heads	135 <sup>0</sup> C maximum (275 <sup>0</sup> F)
Oil Temperature	Minimum: 50 <sup>0</sup> C (120 <sup>0</sup> F) Maximum: 140 <sup>0</sup> C (285 <sup>0</sup> F) Normal Operation: 90 <sup>0</sup> C – 110 <sup>0</sup> C (190 <sup>0</sup> F - 230 <sup>0</sup> F)

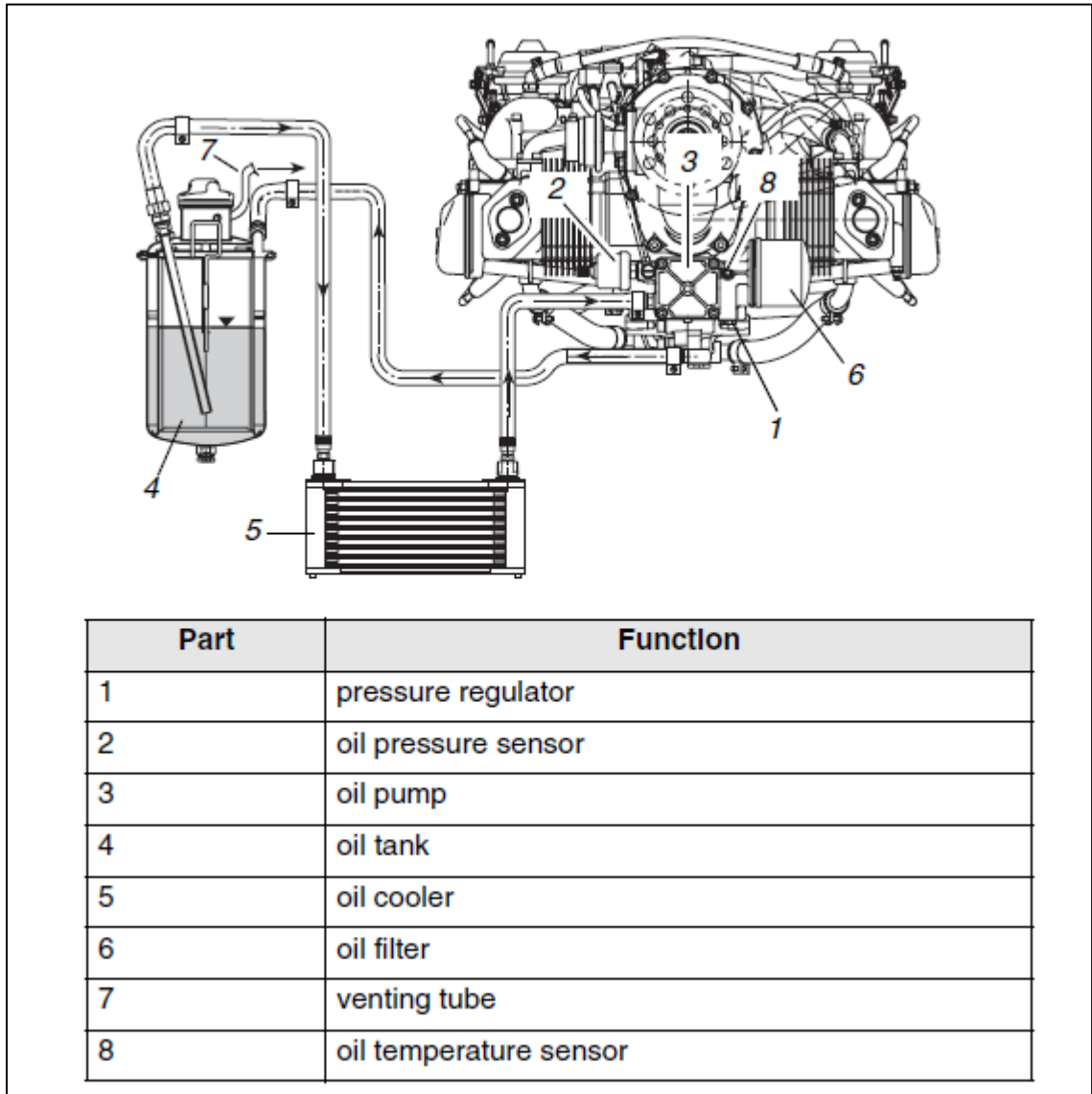
The cylinder head temperature (CHT) and oil temperature affect the performance characteristics of the engine directly. To run the engine at its peak performance, the engine is cooled. An air and liquid cooling system is utilized to cool the cylinder heads. A 50/50 mixture of water and glycol is used as coolant. The temperature is regulated and excess heat is removed by passing the coolant through a radiator before entering the engine. Heat is removed from the cylinder heads by passing the cooler liquid over them during engine operation. Air is forced over the radiator to regulate the temperature.

An expansion tank and overflow bottle collect the extra coolant. For correct flow direction of the coolant the radiator is installed below the engine level with the expansion tank on top.



**Figure 7: Rotax Engine Cooling System**

As specified in the operator's manual from the manufacturer and Table 4, the oil temperature must be regulated for safe operation. Overheating the oil could result in annealed cylinder heads. If the temperature of the oil is below the specified limits, the higher viscosity of the oil hinders the smooth operation of the pistons. An oil cooler is installed (refer Figure 8) to regulate the temperature (Rotax 912ULS DCDI).



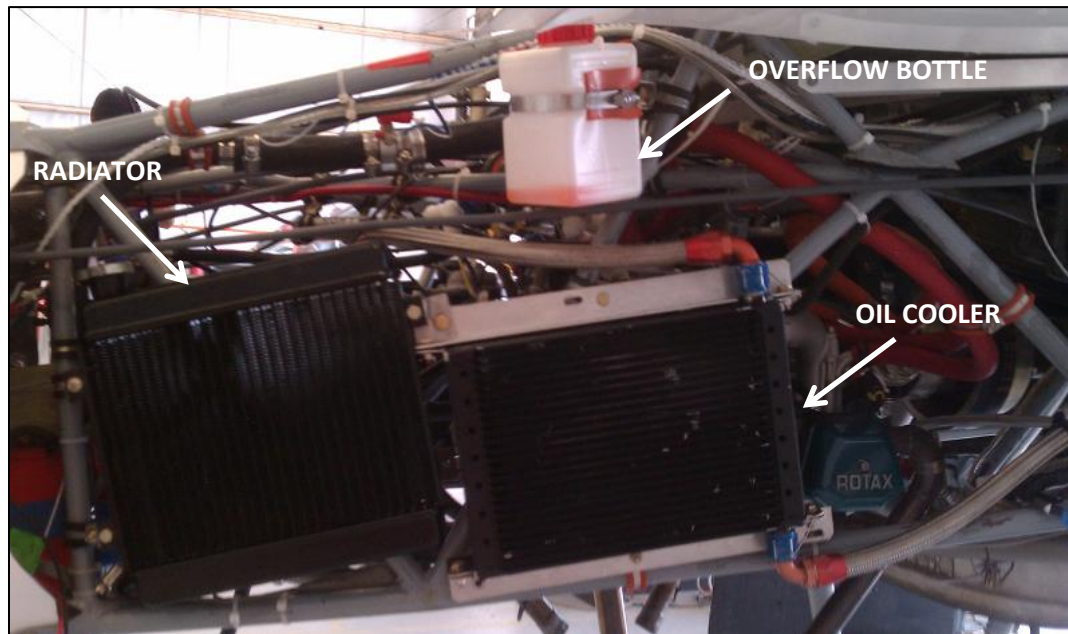
**Figure 8: Rotax Engine Oil Cooling System (Rotax 912ULS DCDI)**

Air is forced through the oil cooler to moderate the oil temperature. This removes some of the heat from the oil before it is sucked into the engine via an oil pump from the oil tank.

Insufficient quantity of oil or coolant causes irreparable damage to the engine's moving parts. For normal operation the, oil pressure must be between 29-73 psi at an RPM greater than 3500.



Permanently open side doors on the fuselage of the Eco-Eagle are utilized to force air over the radiator and oil cooler. Given below is an image of the radiator, oil cooler and overflow bottle as installed on the Eco-Eagle.



**Figure 9: Rotax Cooling System on the Eco-Eagle**

### **Electric Motor**

The electric motor for the direct-drive hybrid propulsion system was a permanent magnet synchronous motor developed by Drivetek AG for Flight Design. A PMSM is a type of motor between an alternating current (AC) induction motor and a brushless direct current (BLDC) motor (Permanent Magnet Synchronous Motors (PMSM) - Motor Drive and Control Solutions). Permanent magnets on the rotor rotate between the windings and cause a sinusoidal flux density within the air gap in the motor. This interaction between the permanent magnets on the rotor and the electromagnets on the stator result in torque output.

The physical construction of the PMSM is similar to a BLDC motor. The input wave for a BLDC is trapezoidal and sinusoidal for a PMSM. A smoother torque is

produced using a PMSM, compared to a BLDC or an AC induction motor. PMSMs cost more than their AC induction counterparts due to their smaller sizes.

Controlled three-phase input current is used to operate a PMSM. For maximum torque output a constant phase difference must be maintained using a motor control unit (MCU), which converts an incoming DC input to the required three-phase output.

To decrease the torque ripple and electromagnetic interference Field Oriented Control (FOC) is implemented. The lack of a rotor coil for mechanical damping limits the motor from utilizing a direct AC power source, this requiring a MCU (Permanent Magnet Synchronous Motors (PMSM) - Motor Drive and Control Solutions).

The 40 hp PMSM in the Eco-Eagle (Figure 10) is rated for a maximum speed of 7000 RPM, 47N-m torque limit with a power consumption of 30 kW.



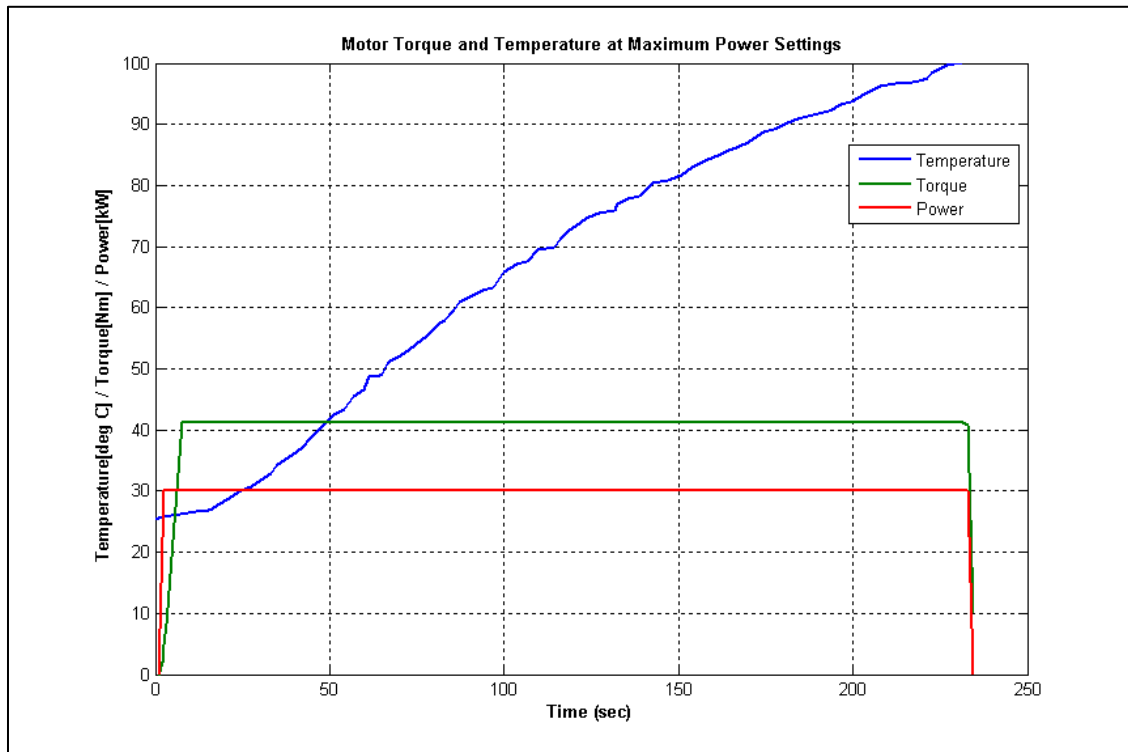
**Figure 10: Electric Motor**

**Table 5: Electric Motor Specification**

PARAMETER	VALUE
Dimensions	Length = 15.57" (395.5mm) Diameter = 5.28" (134mm)
Maximum Power Output	40 hp
Fuel Type	DC Power converted to a 3-Phase Input
Direction of Rotation	Bi-directional (Limited to clockwise from pilot's perspective)

Based upon the charge available, the motor can be utilized as a generator to recharge the system (below 80V).

### Electric Motor Performance Characteristics



**Figure 11: Motor Test at Maximum Power Output**

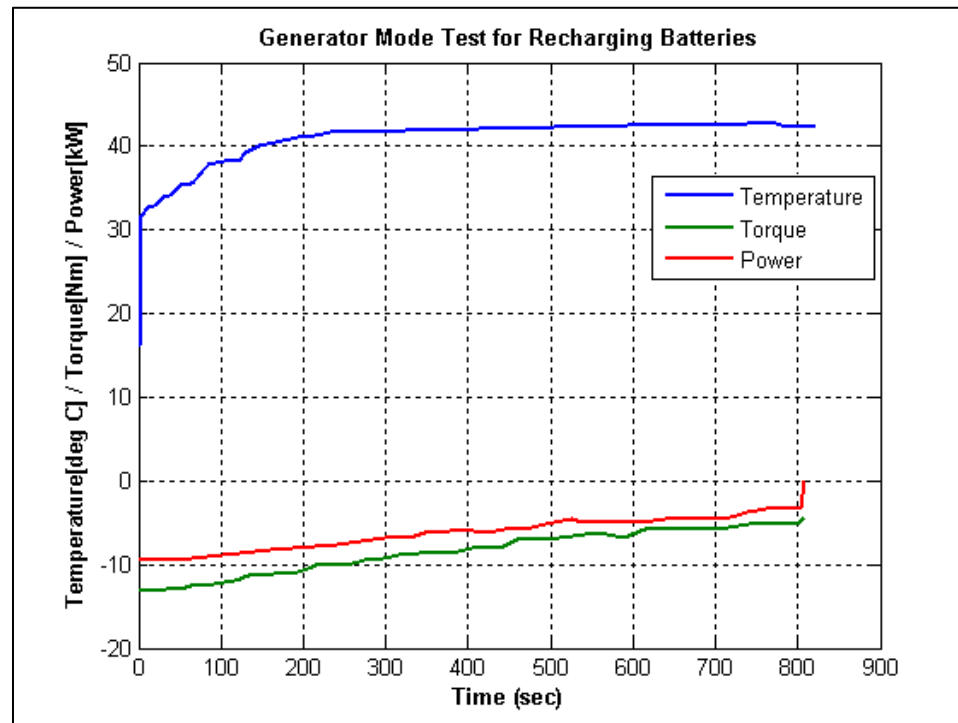
At maximum power output the, electric motor can operate for 180 seconds only, due to the high temperatures, as shown in Figure 11. To avoid component overheating and malfunction, the motor control unit causes the motor to shut down if the upper temperature limit is reached.

The test results in Figure 11 were for the motor operation at 100% torque with a power supply of 132V at 30 kW. It shows that the motor cannot be operated at 7000 RPM for a sustained period of time. Optimum torque and speed settings for sustained flight can be determined via flight testing.

Cooling the electric motor can allow for better performance at higher RPM, using an in-built system with a 50/50 liquid coolant. The motor is cooled with a separate

cooling system (radiator) than the gas engine as the engine runs at much higher temperatures than the motor's operational limits.

Given below are the results for the motor operating in its generator mode:



**Figure 12: Charging Test with Electric Motor in Generator Mode at 25% Torque**

At 25% torque (approximately 42 N-m) the generator can recharge a small battery pack without overheating. The Eco-Eagle did not utilize this mode. The fuel burnt to overcome the frictional losses to rotate the shaft cause the system to be counter-productive and not feasible.

### Field Oriented Control

Field oriented control or vector control is a method of motor control where the rotor is used as the origin for an axis system. The stator phase currents are measured and transformed to the rotor axis. This method is computationally intensive, but allows for more accurate control while being able to obtain the optimal torque output (Atmel Corporation, 2009). To be able to provide a command to the motor using this system,

the immediate speed and position of the rotor must be known, using hall-effect sensors. Hall-effect sensors produce a voltage due to the presence of a magnetic field. This form of control allows for the maximum torque while minimizing the torque ripple.

To obtain the desired speeds and torque output, a particular rotor current must be obtained by varying the stator currents as the rotor currents cannot be measured directly. The stator currents must be varied such that the addition of the three phases results in the currents cancelling out each other. This three phase system is then converted to a two-axis time-varying system rotated to the rotor flux direction. For steady state conditions, the values of the current in this new two-axis system must be constant, thus allowing for a transformation angle to be determined. The transformation angle guides the position for the next voltage vector for the motor. The angle relates to the commutation angle for the stator current in the form of a vectorial command (Zambada). To accomplish this a motor control unit is required to assist in these computations.

### **Motor Control Unit**

The motor control unit (MCU) is a three-phase inverter manufactured by Semikron. The MCU (Figure 13) contains a firmware that allows communication between the MCU and the electric motor. It also provides access to send commands to the electric motor by form of CAN bus messages in hexadecimal format sent to the MCU from a separate computer.



**Figure 13: Motor Control Unit**

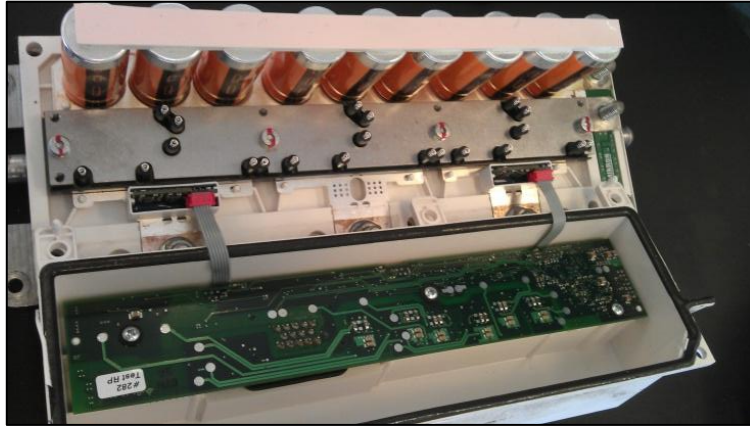
**Table 6: Motor Control Unit Specification**

<b>PARAMETER</b>	<b>VALUE</b>
Manufacturer	Semikron
Model	SKAI 3201MD20 – 1450W
Operating Power	8V to 12V at 0.5A to 1 A
Communication Protocol	CAN Bus
Communication Connector Type	AMPSEAL 16 Connector System
Weight	6.6 lb (3.0 kg)
Dimensions	13.58" x 4.48" x 3.76" (345mm x 113.8mm x 95.4mm)
Cooling Required	50/50 water/glycol solution at 10 l/min

The firmware contains the pre-defined list of CAN channels. Various safety parameters for running the motor such as temperature limits and speed limits are set using this firmware. These channels also contain data regarding the performance of the electric motor. The firmware was developed by Drivetek AG for Flight Design.

The system contains a digital signal processing controller manufactured by Texas Instruments: TMS320LF2406A DSP Controller. The controller operates at a speed of 40 MHz and requires 3.3V to function. It utilizes the Controller Area Network (CAN) 2.0B communication protocol and has an internal on-board memory of up to 32K words and 16 bits (4 sectors) of flash electronically erasable and programmable read-only memory (EEPROM) (Texas Instruments).

The motor control unit provides a 15-pin AMPSEAL connector to interface with the processor for powering the processor and enabling CAN communication. Internally the motor control unit contains nine capacitors (Figure 14), which hold the charge provided by the main battery system. The maximum charge capacity for this system is 160V.



**Figure 14: MCU Interior Component View**

The power stored in the capacitors is converted, using information from the FOC control executed on the DSP controller, to provide the appropriate 3-phase power supply for running the electric motor, supplied in the correct phase. A metal-oxide semiconductor field effect transistor (MOSFET) junction that opens/closes as required creating the three phase power supply to ensure that the power generated is in the correct phase.

A MOSFET or a metal oxide semiconductor field effect transistor is utilized to amplify or switch incoming signals. The MOSFETs act as electronic switches. The MOSFET junction allows the motor controller to behave as a power inverter to assist in the conversion from an incoming DC power input to a three-phase power output.

### **CAN Bus Communication**

CAN bus or Controller-Area-Network is a commonly used standard of communication in the automotive industry to allow various micro-controllers to communicate serially with each other without the requirement of a centralized host computer. Data in this architecture is parsed in the form of serial messages with a header, also known as an arbitration ID. The arbitration ID also allows the system to determine the priority of the messages. A smaller value implies a higher priority for message transmission and vice versa. Each message has a maximum length of 8 bytes. It allows for data speeds of up to 1 Mbps.

CAN architecture allows for various methods of implementation including CANOpen, DeviceNet, J1939 and LIN.

CANOpen allows multiple microcontrollers to communicate with one another. It allows for the configuration of each node in a network via a single configuration tool or master system (ESAcademy). A CANOpen device contains an 'object-dictionary' where each entry is a certain configuration parameter based upon the system. A configuration entry can be made mandatory or optional via this system as well. This mode is implemented on the MCU from Drivetek AG.

DeviceNet protocol uses the CAN bus architecture to allow the exchange of information between various devices. It allows for communication between the lower and higher level devices within a system. DeviceNet offers connectivity between only 64 networked nodes unlike the 127 possible nodes in CANOpen (ESAcademy).

J1939 was developed by the Society of Automotive Engineers (SAE). It limits the bitrate to 250 kbps unlike CANOpen which allows bitrates within 10-1000 kbps. The types of connectors for this specification are severely limited in comparison to the CANOpen system. The maximum number of nodes that can be incorporated within this is 30 (TK Engineering Oy, 2009).

LIN or Local Interface Network is a cheaper method of implementing the CAN bus architecture. It was developed by the LIN Consortium to be utilized where the "the bandwidth and versatility of CAN are not needed" (Vector Informatik GmbH, 2011).

Data in the CAN bus architecture is transmitted in the form of frames. There are four frame types (Robert Bosch GmbH, 1991):

- a) **Data Frame**: This contains data from transmitters. This is the only frame that contains any actual data from the nodes. This frame can contain up to 8 bytes of data.



- b) **Remote Frame**: This is transmitted to request the transmission of a data frame with the same arbitration ID.
- c) **Error Frame**: This is transmitted by a unit if a CAN bus error is detected.
- d) **Overload Frame**: This frame is utilized to provide an extra delay between the data frame and the consecutive remote frame.

### Batteries

The Eco-Eagle utilized lithium iron phosphate (LiFePO<sub>4</sub>) batteries. These batteries are the second generation of lithium-iron cells and are safer than their predecessors due to an altered chemical composition making them more stable system, with decreased energy density and capacity.



**Figure 15: Lithium Iron Phosphate-4 Cell**

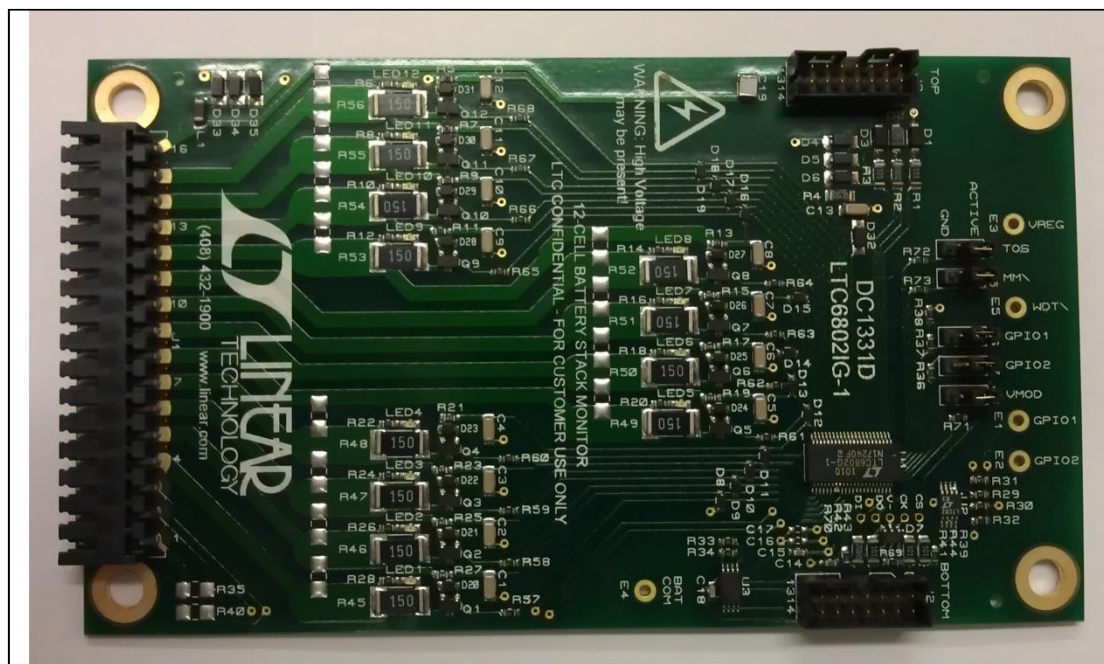
**Table 7: Lithium Iron Phosphate-4 Battery Specification**

BATTERY PARAMETER	VALUE
Model	100122200-2C
Nominal Capacity	20 Ah, 3.2V
Energy Density	125Wh/kg or 56.7 Wh/lb
Weight per cell	1lb 1.92 Oz
Dimensions	7.9" x 4.8" x 0.4"
Maximum Charging Rate	20A (1.0C rate)
Maximum Discharging Rate (Continuous)	40A (2.0C rate)
Safe Voltage Lower Limit	2.3V

To ensure the safety of the system, these batteries were safely tested by video-recording a single cell being shorted from a distance. From the footage for the test, it was determined that the cells would not catch on fire if a short-circuit occurred as the tabs on the batteries melted off within a few seconds. This stops all current flow through the system immediately, before the possibility of a catastrophic failure.

### Battery Monitoring Systems

The manufacturer recommends a BMS to be incorporated for added safety of the battery system. Within this various parameters for the battery system, such as voltage, temperature and current are monitored constantly. This is accomplished by the integration of commercially available printed circuit boards (PCB) with an on-board processor to collect these values, for example: the LTC6802IG-1, manufactured by Linear Technology (refer Figure 16).



**Figure 16: LTC Battery Monitoring Board**

The LTC board allows measurement of up to 12 Li-Iron cells (60 V maximum). Multiple boards can be attached together to monitor a system up to 1000V.

The board also assists in the charging process by providing a cell-balancing system, within which if the cells are in charging mode, it cuts-off and restarts charging in a manner to ensure that all cells have the same voltage. This increases the life of the batteries as their capacities remain the same and thus allow for a more even discharge each time. It does so via the incorporation of a MOSFET switch associated with each cell input to discharge the overcharged cells. This chip also contains a built-in noise filter and an on-board temperature sensor. It utilizes a serial interface for communication purposes (Linear Technology, 2011).

### **Communication Boards**

The battery monitoring systems as well as the relays need commands to be transmitted and received for operation. To accomplish this task multiple Arduino communication boards were utilized.

Arduino boards (Figure 17) are microcontrollers that provide an open-source prototyping platform for electronics. It can receive signals from various sensors as well as commands electronic controllers. Several different types of Arduino boards can be easily purchased based upon user-requirements for processing ability. The boards have an on-boards Atmel AVR processor (Arduino). The boards are relatively inexpensive and easy to program. They can be used across various operating platforms due to an open-source base.



**Figure 17: Arduino Mega 2560**

## Propeller

The Stemme S-10 is a German motor-glider, which consists of a folding propeller and a retractable nose-cone section, thus allowing it to take off without requiring a tow-plane. The original propeller diameter was 1.63m (5 ft 4.25”).

To maximize its efficiency, a bigger and more efficient propeller was utilized. To A custom experimental propeller was designed by MT-Propeller Entwicklung GmbH.



**Figure 18: Experimental MT-Propeller for the Eco-Eagle**

**Table 8: Propeller Specification**

PARAMETER	VALUE
Model	MTV – 1 – A / 184 – 51
Serial Number	100837
Type Certificate Number	P25B0
Propeller Diameter	184 cm
Number of Blades	2
Approximate Weight	23 lbs
Blade Pitch	Minimum: 5° Maximum: 50°

The model number can be broken down to describe the propeller in further detail:

- MTV: Specifies that it is a variable pitch propeller.
- A: Specifies that it has a special flange type meant for smaller engines, with a bolt size of 7/16" – 20UNF, with a bolt circle diameter of 80mm.
- No alphabet at this position implies that this propeller has no counterweights producing a moment towards a low pitch position.
- 184: This is the diameter of the propeller in centimeters (6.04 ft). As there is no alphabet preceding this number, it implies a right hand sense of rotation.

The propeller blade is built with high compressed epoxy impregnated wood in the root area with the rest of the blade made of lightweight wood covered by epoxy fiberglass (MT-Propeller, 2008).

The propeller data (power curves, efficiency curves) for this particular variant of the blade are not available. The performance for this particular blade is similar to a blade within the same family: MTV – 1 – A / 180-51. The latter propeller is 4cm shorter than the one installed on the Eco-Eagle. Given below in Table 9, is a listing of the efficiency for the propeller as a function of the power coefficient ( $C_p$ ) and the advance ratio ( $J$ ).

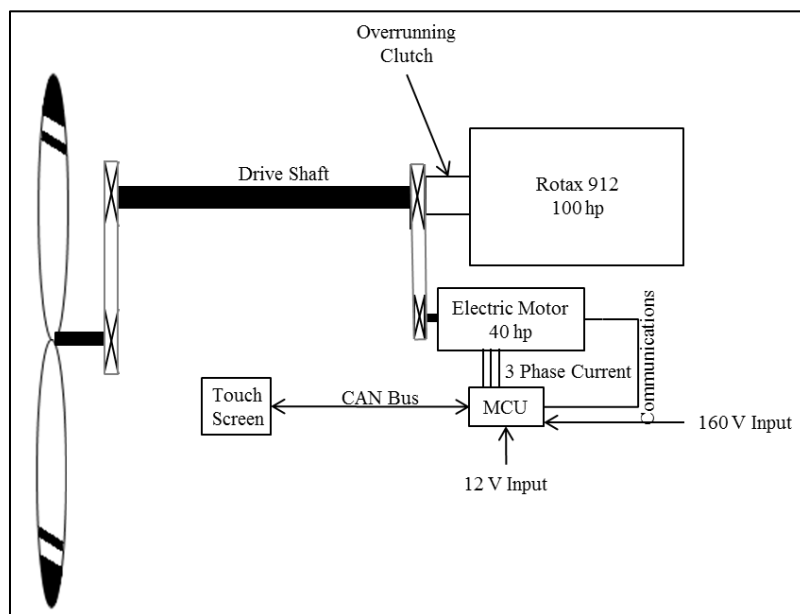
Table 9: Propeller Performance Table

>HELIX SYSTEM< *CP*		MTV-1-A/180-51		R0295M20 24.01.06							
Flughöhe	0 m	*Adv.R. J vo	0.20 -	entsp. V vo	54.0 km/h						
entspr.	0 ft		bi 1.60 -		bi 432.0 km/h						
			(= 20 Intervalle)								
P-Drehzahl	2500 1/min	*Cp*E-2 vo	2.0 -	>> P_eff vo	37.2 %						
Dehm.Prop.	1.80 m		bi 16.0 -		bi 297.7 %						
			(= 20 Intervalle)								
Copyright: MT-Propeller Entwicklung GmbH & Co. KG, Postfach 0720, D-94307 Straubing											
<b>Wirkungsgrad etaP in Abhängigkeit von J und Cp</b>											
	Cp*E-2 =										
J =	2.00	2.70	3.40	4.10	4.80	5.50	6.20	6.90	7.60	8.30	9.00
0.200	0.500	0.477	0.448	0.416	0.380	0.345	0.307	0.270	0.235	0.205	0.184
0.270	0.600	0.590	0.564	0.536	0.504	0.466	0.427	0.382	0.334	0.291	0.259
0.340	0.667	0.671	0.653	0.631	0.605	0.575	0.537	0.496	0.445	0.388	0.339
0.410	0.712	0.728	0.718	0.701	0.682	0.660	0.632	0.597	0.556	0.499	0.432
0.480	0.741	0.769	0.765	0.753	0.738	0.721	0.702	0.679	0.649	0.609	0.547
0.550	0.762	0.797	0.798	0.790	0.779	0.766	0.752	0.735	0.714	0.692	0.661
0.620	0.776	0.816	0.823	0.818	0.810	0.800	0.789	0.776	0.760	0.742	0.723
0.690	0.786	0.830	0.840	0.838	0.833	0.825	0.817	0.807	0.795	0.780	0.764
0.760	0.793	0.838	0.852	0.853	0.850	0.844	0.838	0.831	0.822	0.810	0.796
0.830	0.796	0.844	0.860	0.863	0.862	0.859	0.854	0.848	0.842	0.833	0.822
0.900	0.797	0.846	0.865	0.870	0.871	0.869	0.866	0.862	0.857	0.850	0.842
0.970	0.796	0.847	0.868	0.875	0.877	0.877	0.875	0.872	0.868	0.863	0.857
1.040	0.792	0.846	0.868	0.878	0.881	0.882	0.881	0.879	0.876	0.873	0.868
1.110	0.786	0.842	0.867	0.878	0.883	0.886	0.886	0.885	0.883	0.880	0.876
1.180	0.778	0.837	0.865	0.877	0.884	0.887	0.889	0.889	0.888	0.886	0.883
1.250	0.765	0.831	0.860	0.875	0.883	0.888	0.890	0.891	0.891	0.890	0.888
1.320	0.752	0.821	0.855	0.872	0.882	0.887	0.891	0.893	0.893	0.893	0.891
1.390	0.736	0.812	0.848	0.868	0.879	0.886	0.890	0.893	0.894	0.894	0.893
1.460	0.719	0.800	0.841	0.863	0.875	0.883	0.889	0.892	0.894	0.895	0.895
1.530	0.698	0.787	0.832	0.857	0.871	0.880	0.886	0.891	0.894	0.895	0.895
1.600	1.204	0.771	0.821	0.849	0.866	0.876	0.884	0.888	0.892	0.894	0.895
	Cp*E-2 =										
J =	9.00	9.70	10.40	11.10	11.80	12.50	13.20	13.90	14.60	15.30	16.00
0.200	0.184	0.167	0.153	0.140	0.129	0.119	0.111	0.103	0.096	0.090	0.085
0.270	0.259	0.234	0.214	0.196	0.181	0.167	0.155	0.144	0.135	0.126	0.118
0.340	0.339	0.305	0.278	0.255	0.235	0.217	0.201	0.187	0.174	0.163	0.153
0.410	0.432	0.382	0.347	0.317	0.292	0.269	0.250	0.232	0.216	0.202	0.189
0.480	0.547	0.470	0.420	0.384	0.353	0.325	0.301	0.279	0.260	0.243	0.227
0.550	0.661	0.596	0.504	0.457	0.419	0.386	0.357	0.331	0.307	0.286	0.267
0.620	0.723	0.701	0.659	0.540	0.492	0.453	0.418	0.387	0.359	0.334	0.311
0.690	0.764	0.747	0.728	0.705	0.575	0.526	0.485	0.448	0.416	0.386	0.360
0.760	0.796	0.782	0.766	0.750	0.731	0.702	0.560	0.516	0.478	0.444	0.413
0.830	0.822	0.809	0.796	0.782	0.767	0.751	0.732	0.596	0.550	0.509	0.472
0.900	0.842	0.832	0.820	0.808	0.795	0.782	0.767	0.751	0.715	0.585	0.542
0.970	0.857	0.849	0.839	0.828	0.817	0.805	0.793	0.781	0.766	0.749	0.623
1.040	0.868	0.862	0.854	0.845	0.835	0.825	0.814	0.803	0.792	0.779	0.764
1.110	0.876	0.872	0.866	0.859	0.850	0.841	0.832	0.822	0.812	0.801	0.789
1.180	0.883	0.879	0.874	0.869	0.862	0.854	0.846	0.837	0.828	0.818	0.809
1.250	0.888	0.885	0.881	0.876	0.871	0.864	0.857	0.849	0.841	0.833	0.824
1.320	0.891	0.889	0.886	0.882	0.877	0.872	0.866	0.859	0.852	0.844	0.836
1.390	0.893	0.892	0.890	0.886	0.883	0.878	0.873	0.867	0.860	0.854	0.847
1.460	0.895	0.894	0.892	0.890	0.886	0.882	0.878	0.873	0.867	0.861	0.855
1.530	0.895	0.895	0.894	0.892	0.889	0.886	0.882	0.878	0.873	0.867	0.862
1.600	0.895	0.895	0.895	0.893	0.891	0.888	0.885	0.881	0.877	0.872	0.867
Cp = P / (rho*n^3*D^5)			J = v / (n*D)				Ct = Cp*etaP/J				

## CHAPTER 3: Aircraft System Layout

### The Eco-Eagle Electric Propulsion System

A parallel hybrid / direct drive powerplant system allows the Eco-Eagle to fly using either the electric motor or the gas engine individually.



**Figure 19: Propulsion System Layout**

A system of pulleys and an overrunning clutch was incorporated to accomplish this, as shown above in Figure 19. The clutch system allows only one of the two power sources to turn the propeller. This limits the maximum power output, as seen by the drive shaft, to 100 hp. This guarantees safe operation of the powerplant with the existing drive shaft, as the original engine rating for the aircraft, before modifications, was for a 100 hp engine.

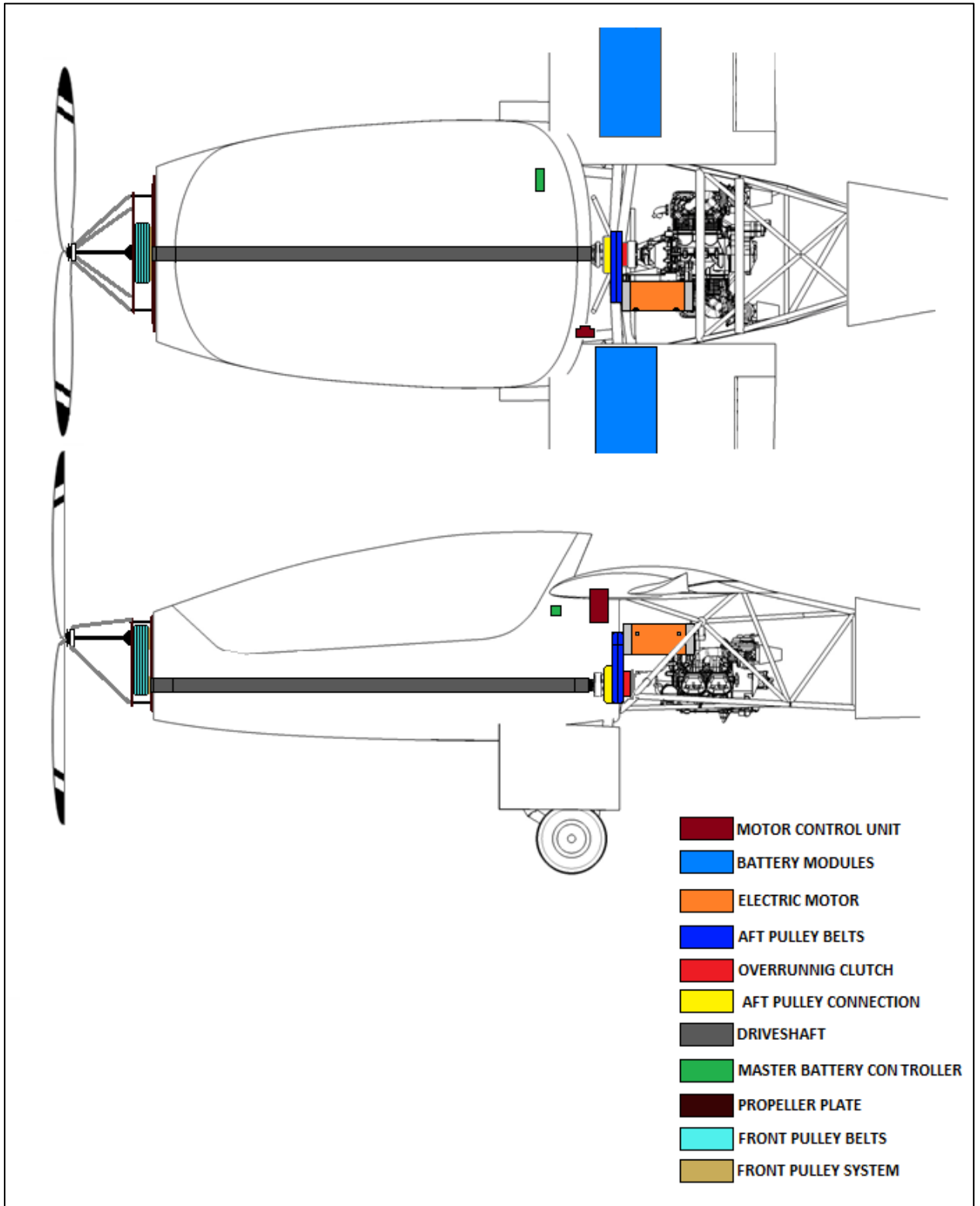
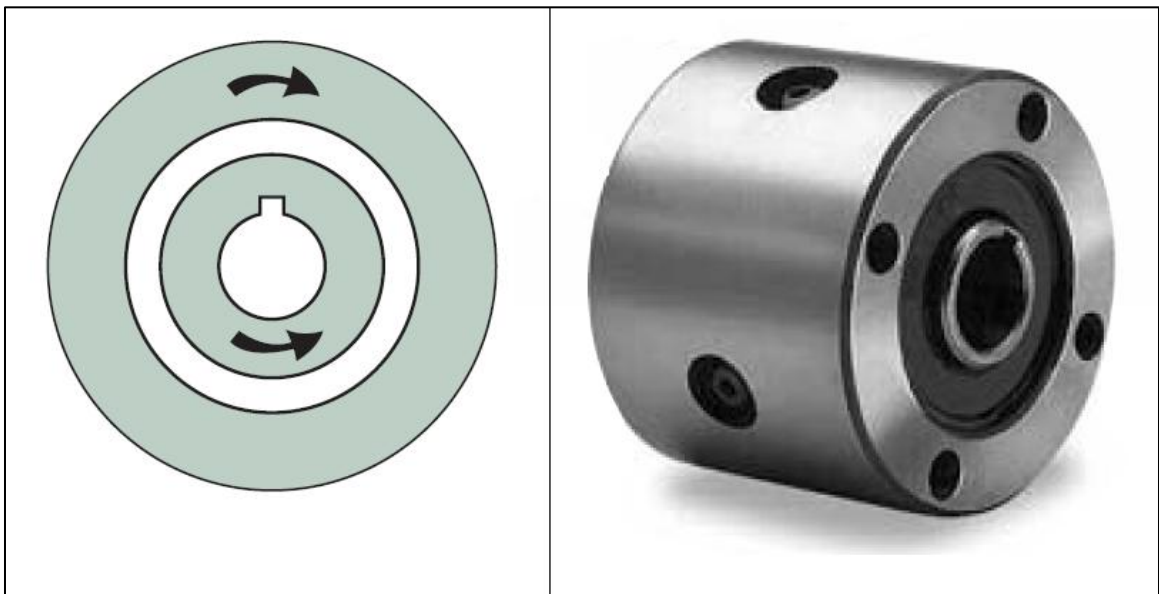


Figure 20: Top and Side View of Propulsion System



An in-house design, picking up the existing engine mounts from the aircraft truss was implemented to integrate the hybrid propulsion system. To allow proper ground clearance with the new and longer propeller, the plates in the front of the aircraft for holding the propeller in place and transferring the torque from the drive shaft to the propeller were modified.

A pulley and clutch system was incorporated between the electric motor, gas engine and the drive shaft to transfer torque to the propeller from the powerplant. An overrunning clutch (Figure 21) was utilized to allow running a single powerplant to provide the torque to the driveshaft at a given time.



**Figure 21: Overrunning Clutch (Formsprag Clutch)**

The clutch ensures that the direction of rotation does not change by locking in the counterclockwise direction. The overrunning clutch has two races, inner and outer, which can either spin together at the same time or spin individually (Formsprag Clutch). By attaching the gas engine directly to the outer race only and the electric motor to the inner race via a pulley system, it can be ensured that the electric motor does not turn the gas engine, thus separating the two powerplants as two individual operations. To

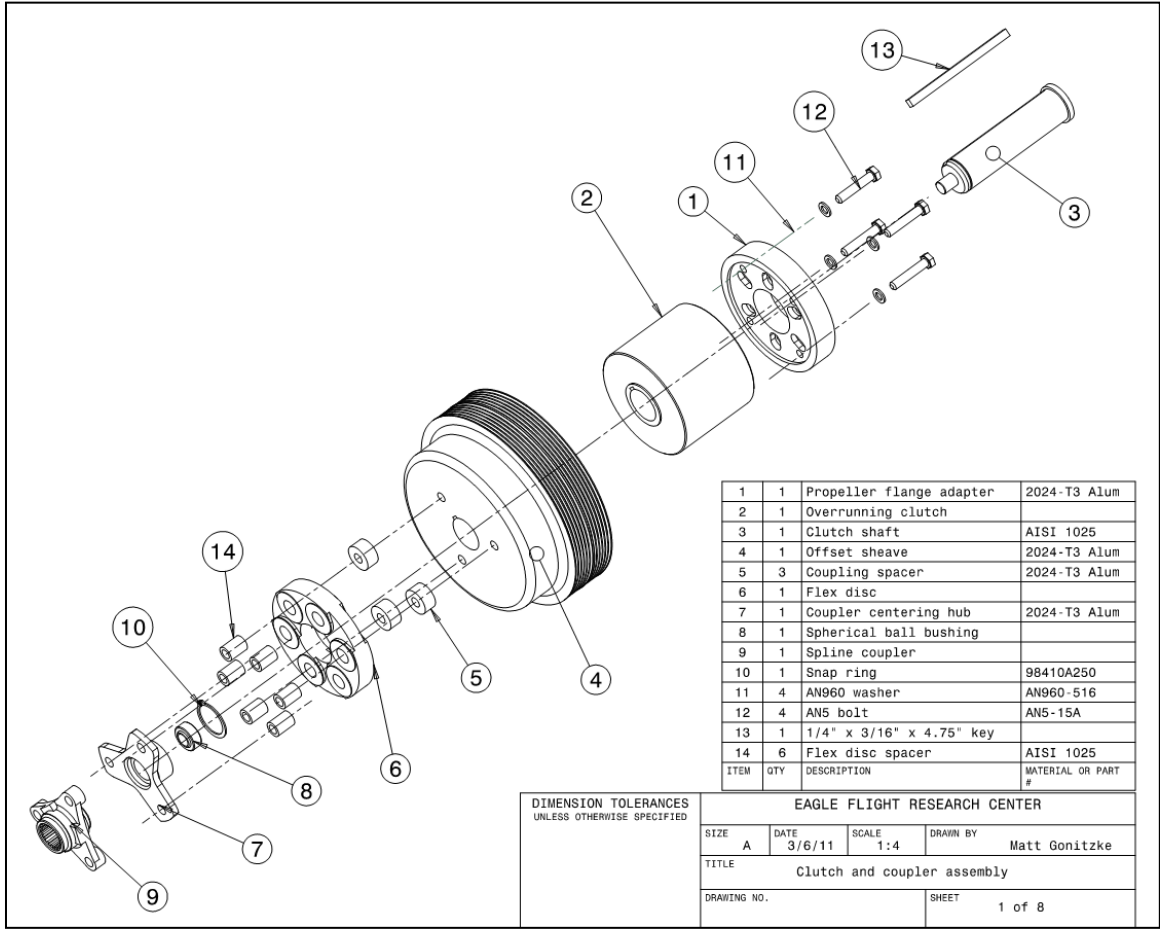
ensure that only a single system is forcing the propeller to rotate, the gas engine is turned off by the pilot manually before the electric motor is started.

The ratio for the pulley from the electric motor to the driveshaft is 1:3.1. This reduced the RPM to a value more suitable for the propeller. The maximum output from the electric motor is 7000 RPM. With this ratio, the speed drops down to roughly 2258 RPM at the driveshaft. This speed is further reduced by the presence of a set of pulleys in the front of the aircraft, where the ratio is 1.18:1. Thus, the RPM from the engine and the motor now drop down to the values shown below in the table.

**Table 10: RPM at Propeller after Reduction**

RPM AT ENGINE/MOTOR	RPM AT PROPELLER
Engine	
• 5800	• 2022 RPM
• 5500	• 1918 RPM
• 5000	• 1743 RPM
Motor	
• 7000	• 1913 RPM
• 5800	• 1585 RPM

The following image depicts the assembly for the drive train mentioned above:



**Figure 22: Drivetrain Assembly (Gonitzke, 2010)**

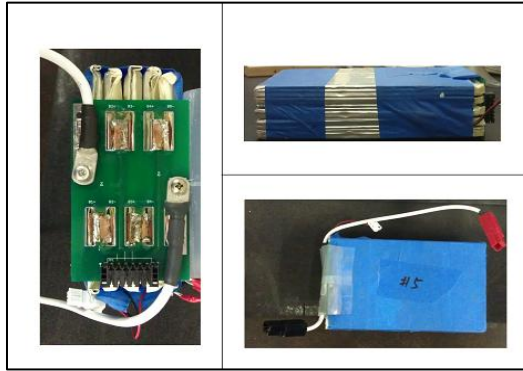
**Battery System Layout**

The Eco-Eagle utilized lithium iron phosphate (LiFePO<sub>4</sub>) batteries (Figure 15) with custom-built battery controllers that communicate with the touchscreen display system to allow the pilot to keep track of the battery system in-flight. The following is a list of all the components that build up the battery system:

**Table 11: Battery System Components**

<b>COMPONENT</b>	<b>NUMBER INSTALLED</b>
LiFePO <sub>4</sub> Cells	<ul style="list-style-type: none"> <li>• 5 per battery pack</li> <li>• 40 per battery module (8 packs = 1 module)</li> <li>• 160 cells overall (4 battery modules installed)</li> </ul>
Relays	<ul style="list-style-type: none"> <li>• 1 Master Relay</li> <li>• 1 Transitional Master Relay</li> <li>• 4 Module Main Relays (1 per battery module)</li> <li>• 4 Module Transitional Relays (1 per battery module)</li> </ul>
Battery Monitoring Boards (LTC Board)	16 (4 per battery module)
Temperature Sensors	32 (1 per battery pack)
Arduino Microcontroller	<ul style="list-style-type: none"> <li>• 4 slave boards (1 per module)</li> <li>• 1 master controller board</li> </ul>
Circuit Protection Fuses	<ul style="list-style-type: none"> <li>• 4 (100 Amp, 300 V<sub>DC</sub>) in the battery modules(1 per battery module)</li> <li>• 1 main fuse (175 Amp, 300 V<sub>DC</sub>)</li> </ul>
Current Sensors	4 (1 per battery module)

The maximum voltage on the system was approximately 150 V with a nominal voltage of 132V. The airplane integrated four battery modules (two in each wing), with the possibility of a fifth module being incorporated in the empennage of the aircraft. Five individual cells were connected in series to form a battery pack (Figure 23). Each battery module consisted of eight battery packs.



**Figure 23: Individual Battery Pack**

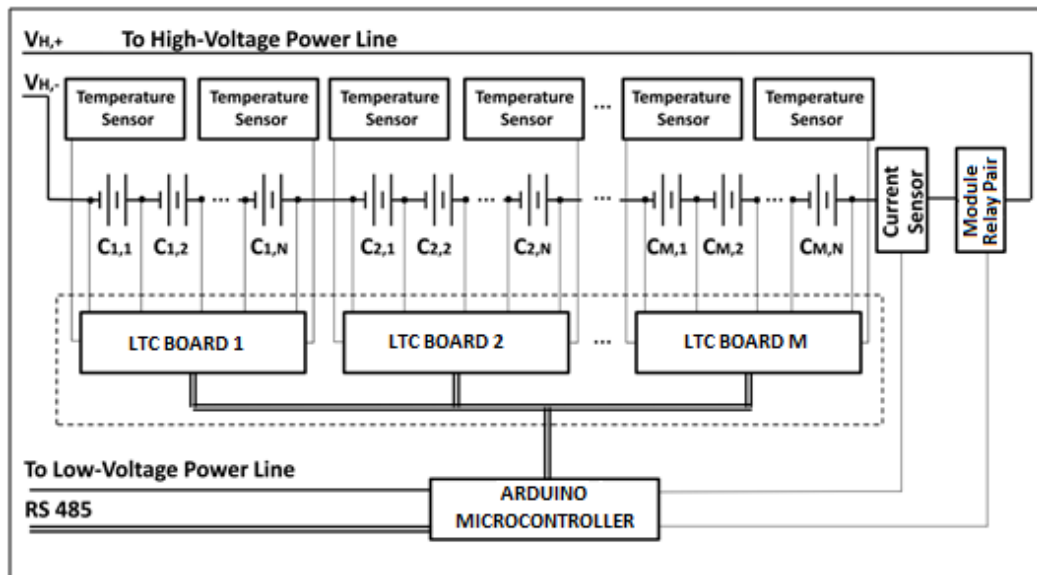
The battery packs were connected in series to build up the voltage of the system to the required amount. Every battery module (Figure 24) formed by this was a 20 amp-hr module, which implies that a module could provide a steady 20 amp current for one hour continuously. The battery modules were combined in parallel to each other to build up the current of the system to a resulting 80 amp-hr with the four modules in the wings. With the addition of the fifth module, the output could be increased to 100 amp-hr.



**Figure 24: Battery Module**

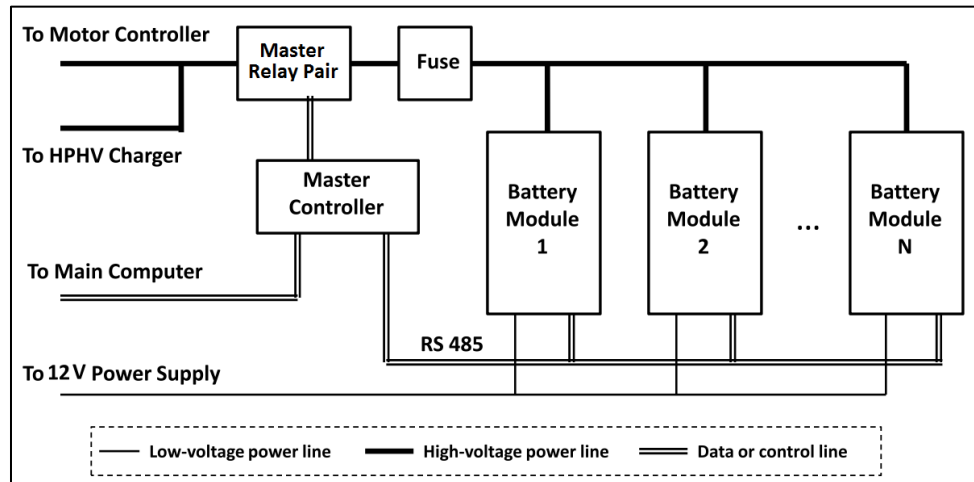
Multiple circuit protection systems were implemented to increase system safety. This was accomplished in the form of a health monitoring software, called the “Battery Management System” (BMS) as well as physical circuit protection provided by a combination of fuses and relays, as suggested in “Chapter 11: Aircraft Electrical Systems” of “AC 43.13-1B: Acceptable Methods, Techniques and Practices – Aircraft Inspection and Repair” (AIRCRAFT ELECTRICAL SYSTEMS, 1998).

The individual battery modules were separated from each other using a relay and fuse system. These relays were termed “Module Main Relays.” Along with these main relays were the “Module Transitional Relays.” The module main relays allow the battery modules to be connected to each other and the module transitional relays ensure that the overall voltage of the modules connecting to each other are relatively close, by balancing each other if needed, to avoid a huge current flow as soon as the modules get connected to the main power lines (Figure 25).



**Figure 25: Battery System Connections**

Power is provided to the electric motor controller by closing the “Master Relay” and the “Transitional Master Relay” after the module main relay and transitional main relays are closed, as shown in Figure 26. This allows for a centralized control system for the battery management process.



**Figure 26: Battery Module Connections**

The batteries were installed in the aircraft using a tray which slid into the in-board wing section of the aircraft, one on each side, as shown below.



**Figure 27: Wing Cutout for Battery Tray**

To accomplish this, a rectangular hole was cut into the wing to remove the existing wing-tanks. The trays are not permanently installed into the wings and can be removed for ease of maintenance. Limited space within the trays for the batteries and wires ensured that the batteries did not move about freely in the wing.

The wiring for providing the power to the motor as well as the communication cables for operating the battery system pass through the inboard wing section and are passed behind the pilot. Only the communication cables are allowed to pass through

into the cockpit, where the master controller board is installed. The power cables are connected together in the lower left side of the fuselage, just above the left landing gear, near the motor control unit to pass on the power in a safe manner via a fuse and relay system.

### **The Battery Management System**

The software to perform health monitoring and controlling the batteries along with the above mentioned controller boards form the battery management system (BMS). The software allows various modes to be commanded based upon the state of charge for the batteries. It operates at a baud rate of 57600. There are two switches associated with the functioning of this system:

- a) **Kill Switch:** This allows the pilot to provide power to the battery management system, including the controller boards and the relays. This power is provided by a separate 12V aircraft battery.
- b) **Enable Switch:** This switch sends a message to the master controller to close the master transitional relay and the master relay. The master transitional relay ensures that the capacitors on the MCU are charged to the same amount as the battery system to avoid a high amount of current to be drawn in a short period.

The four main modes for the battery management system are:

- a) Stand-By Mode
- b) Relay Mode
- c) Charging Mode
- d) Discharging Mode

The modes can be commanded individually by sending a single character to the master battery controller. Depending upon the mode in which the battery system is



being used, one of the four modes must be enabled to allow for the safe usage of the system via the “Battery Management System.”

Within these modes, other commands (single characters) can be sent to the system to allow the main relays and the transitional main relays to be controlled as well as monitor the voltage, temperature and current drawn through the system. Health monitoring is accomplished with LTC (Linear Technology) boards that measure the voltages of the cells and temperature of the packs. Current is measured via current sensors. This data is sent to the slave (Arduino) controller boards where the appropriate action is decided based upon certain presets.

- a) **Stand-By Mode ('s')**: This is the default start-up mode for the controller. This mode allows the various modes to be invoked from a starting point. It allows the monitoring of the states for the battery modules (online or offline) as well as the state of all the relays (main and master). This mode is normally used only for debugging purposes on the ground. It can display any open connections within the system also.
- b) **Relay Mode ('r')**: This mode enables the main relays and transitional main relays to be opened ('o') or closed ('c') as well as verify status of these relays ('d'). Within this mode, voltage, temperature and current cannot be monitored.

When the command to close the relays is sent, the master controller sends a signal to the slave controllers (Figure 17) to first close the module transitional relays and then the module main relays to allow the battery modules to provide power to the motor via the MCU.

- c) **Charging Mode ('+')**: This mode must be enabled when the battery system is being charged on the ground using an external power source. This allows the system to monitor itself to ensure that the appropriate main relays are opened if a cell within a module reaches its maximum safe voltage limit (3.75 V). If no power is being provided to battery modules, while in this mode, batteries would

start to balance each other. This means that the cells that have a higher voltage than the ones in the associated battery pack will provide charge for other cells, until they equalize in voltage. This increases the battery life for the system. If the cells are closer to each other in voltage, this mode decreases the chances of a module cutting off for protecting just a single cell from its lower voltage limit even if the others hold enough charge to run the system. The voltage ('v') of the cells, temperature of the batteries packs ('t') and current drawn by each module ('i') can be viewed within this mode via a command. For safety, as the system is being charged on the ground, even though the software is set-up to self-monitor the system for any exceedences, voltage and current for the system should be monitored manually through this software. In order to charge the system, the main relays and the master relay must be closed first. The system can be charged without being in this mode but is highly unsafe and must be avoided. Currently, there is no in-built feature to automatically command this mode during charging and hence must be commanded manually via the master board. A reason to force the user to do this is to have the operator constantly monitor the system as it is being charged, for safety.

- d) **Discharging Mode ('-')**: The discharging mode, similar to the charging mode, allows the system to self-monitor the batteries, while current is being drawn by the electric motor. Within this mode as well, voltages, current drawn and temperatures are monitored and can be viewed simultaneously. This ensures that the batteries are being discharged safely and that the modules are automatically cut-off, by opening the associated main relays, if the voltage for a cell drops below a certain limit (2.8 V) or if there is a short in the system and the current spikes. The system can be discharged once all the relays are closed, even if this mode is disabled, but for the purpose of safety, it is imperative that this mode first be enabled. This ensures that the batteries do not overheat or go below their safe voltage limit during discharge. If any error is read by the master

controller, the associated module is switched-off by opening the battery module relay to allow for the continued safe operation with the other battery modules.

If an incorrect command is sent to the controller, within a serial window, it displays all the possible commands, as shown in the screenshot Figure 28 below:

```
Available commands:"
+'---Charging mode; '-'---Discharging mode:"
'v'---read cell voltages;"
't'---read temperature;"
'i'---read current;"
'a'---read everything."
'r'---Relay operation mode:"
'd'---read relay status;"
'c'---close the main relay;"
'o'---open the main relay."
's'---Standby mode:"
'w'---check and report openwire;"
'm'---check the size of free RAM."
'e'---check the number of CRC errors."
'l'---calibrate the current sensor."
```

**Figure 28: Battery Management Commands**

### **Pilot Interface System**

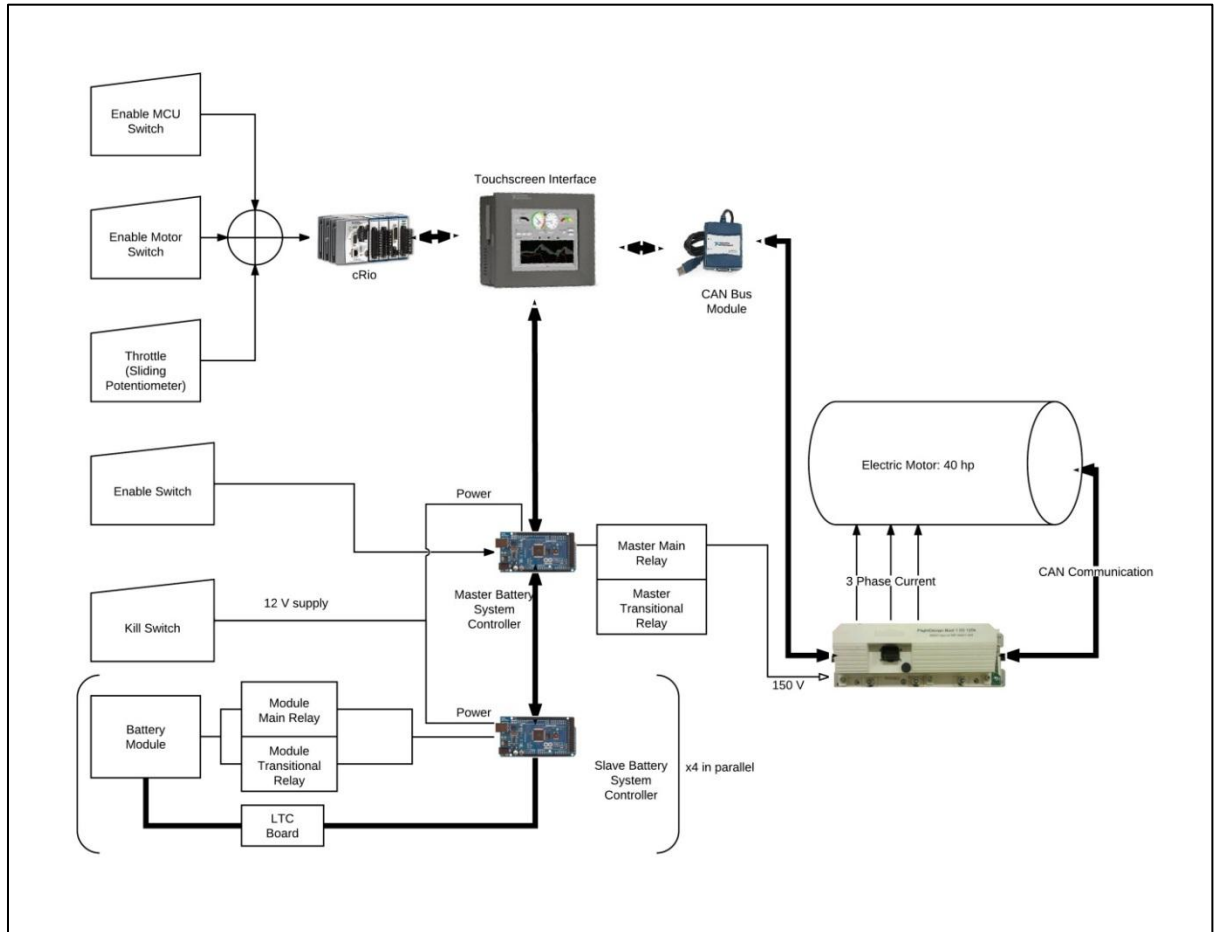
The pilot interface to operate the aircraft was modified to allow the pilot to command the electric motor as well as the gas engine with ease. Controls for operating the Rotax engine were not modified, thus allowing the pilot to operate it as though it were a regular gas-engine aircraft.



**Figure 29: Avionics System**

The use of a PMSM (motor) requires the use of a single electronic motor controller as mentioned earlier. The electronic interface was used to transfer and receive signals to and from the motor. This allowed the pilot to command a torque from the motor by increasing the amount of current being drawn from the battery system. In certain cases, the pilot could also command a particular speed from the electric motor while keeping the overall torque constant. In the case of the Eco-Eagle, the pilot commands only a torque from the motor as the constant speed case is irrelevant due to a constant speed, variable pitch propeller being utilized.

The system was integrated into the existing aircraft system such that the pilot could adapt to the new system with ease. To accomplish this multiple switches and a new throttle quadrant was added to the airplane along with a touchscreen display system for pilot interaction with the motor controller as shown in Figure 30 below.



**Figure 30: Electric Motor Control Hardware Layout**

### Electric Motor Throttle Controller

To provide a control that would be easy to adapt to for the pilot, a sliding potentiometer was used to allow the command of torque from the electric motor. The physical functioning of the unit allowed it to be used in a manner similar to a push/pull throttle quadrant.



**Figure 31: Sliding Potentiometer**

**Table 12: Electric Motor Throttle Specifications**

Manufacturer	Celesco Transducer Products, Inc.
Model	MLP-50
Range of Motion	2 inches
Linearity	$\pm 1\%$
Life Expectancy	> 25 million cycles
Vibration	up to 10 G's to 2000Hz maximum

### Touchscreen Interface

A touchscreen interface by National Instruments was embedded in the avionics panel to comply with the space constraints, using minimal space while providing maximum information from various sources and communicating with the MCU. This particular model was chosen as it fit the space requirements, as well as provided connectivity for a CAN-bus module via USB and Ethernet connectivity for the real-time data acquisition unit (Compact Rio).

**Figure 32: NI TPC-2206 Touchscreen Interface**

**Table 13: Touchscreen Interface Specifications**

<b>PARAMTER</b>	<b>VALUE</b>
Manufacturer	National Instruments
Model	TPC-2206
Connectivity	2 Ethernet ports, 2 USB 2.0 ports, 2 serial ports
Interface Size	5.7 inches diagonal
Processor	1.33GHz Intel Atom
Operating System	Windows XP Embedded Standard 7

Using this interface, multiple data screens for various modes of aircraft operation were displayed in tabbed windows. For example, the default tab contained information for the operation of the electric motor, with the following tab containing information for the Rotax engine. Consequently, more tabs could be added if further data needs to be displayed, while retaining the physical size of the unit constant.

### **Graphical User Interface**

One of the key factors for any avionics system apart from handling and processing information is the display of the data in a sensible manner for pilot. If the data is displayed in a proper format, it reduces the stress on the pilot, thus allowing him to make better decisions in the event of an emergency.

The graphical user interface for the Eco-Eagle was designed to allow the pilot to obtain a quick overview of the key performance factors for the electric motor and obtain emergency notifications. At the same time, for overall performance information, certain parameters of the Rotax engine were monitored alongside the electric motor and displayed on the same screen. This was done due to the limited space in the cockpit, as seen in Figure 29. The ability to have multiple screens within the same digital

console allowed all parameters to be displayed in a single location rather than various independent gauges within the cockpit.

The original version of the interface was developed for ground testing the aircraft and captured all possible parameters that were involved with the motor's operation, as shown below in Figure 33.



**Figure 33: Ground Station Motor Test Interface**

The on-board version was modified to incorporate human factors, such as color coding warnings and cautions for the pilot. The distributed display on the pilot interface allowed to pilot to obtain important information about system operation at a glance.

The interface was split into three separate display tabs:

- a) Electric Motor
- b) Rotax Engine
- c) Record Data

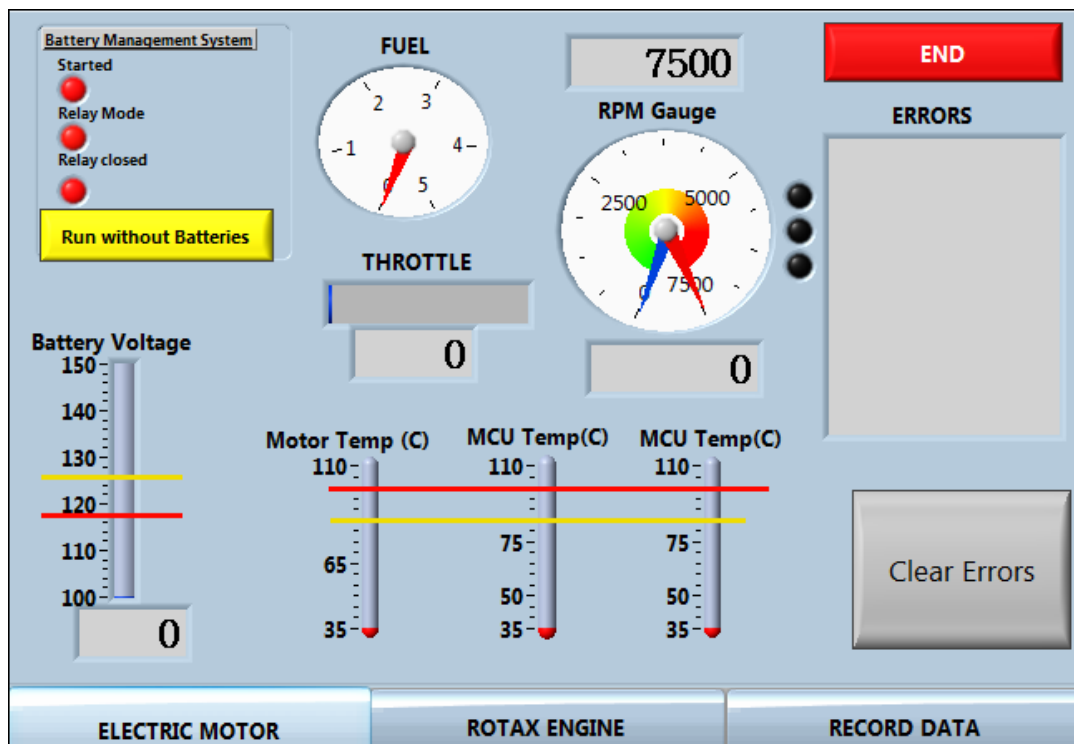


This allowed maximum data to be displayed on a single screen without overwhelming the pilot with information.

The default display for the GUI was the electric motor display. It contained all the information that the pilot would require to operate the electric motor in a safe manner. The screen indicated if the BMS was ready for use or not.

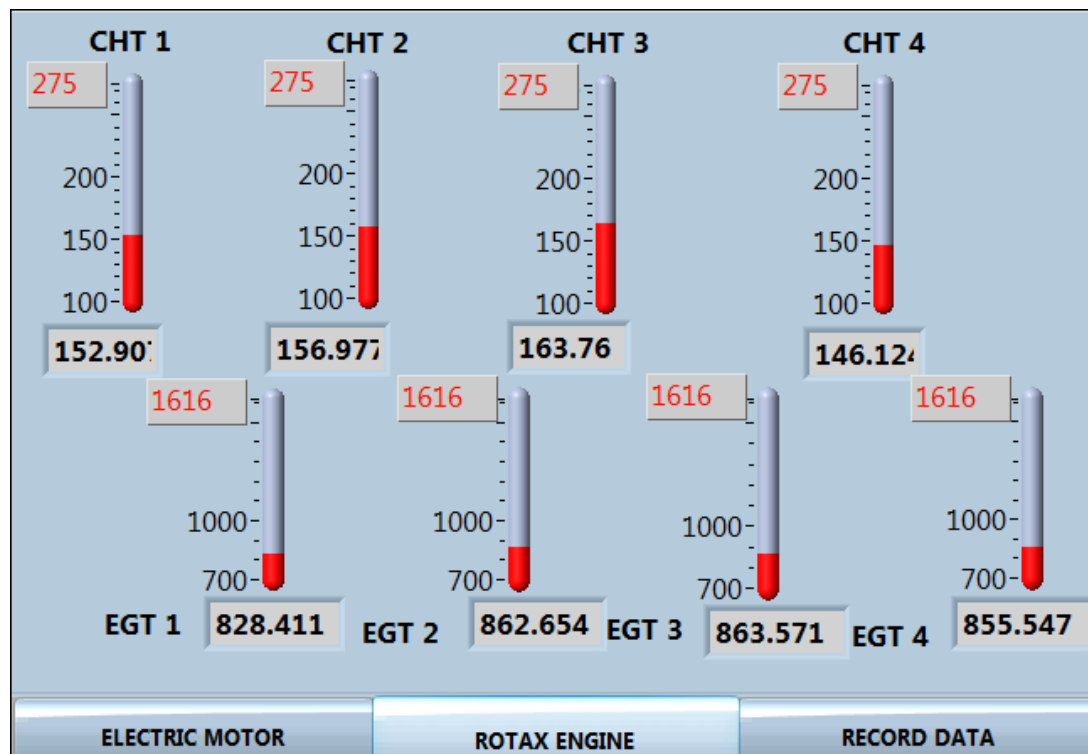
The temperature of the motor and the motor controller, overall voltage of the battery system for the electric motor and RPM for the motor were also display on the same screen. As this was the main screen for aircraft operations, the fuel level indication for the Rotax engine was also presented here.

All the data represented on the screen was color coded to allow the pilot to make quick judgments rather than just relying on memorizing actions associated with numerical values. As is common practice, any action that would require the system to be shut-off (either automatically or by pilot action) was indicated in red. For example, the lowest safe operation voltage for the battery system was 118 V and indicated by the presence of a red line through that point on the voltage indicator. Similarly, any caution warnings to the pilot were indicated in yellow. This implied that the system would not shut-off immediately, but the system was close to its maximum operational limit and appropriate considerations must be taken while running it. For example, the motor temperature caution line was drawn at 90 °C. The action associated with this in the pilot operation handbook was to decrease the demanded torque, until the temperature reduced to a safer point or switch over to the gas engine, if the temperature approached the red line limit as shown in Figure 34 below.



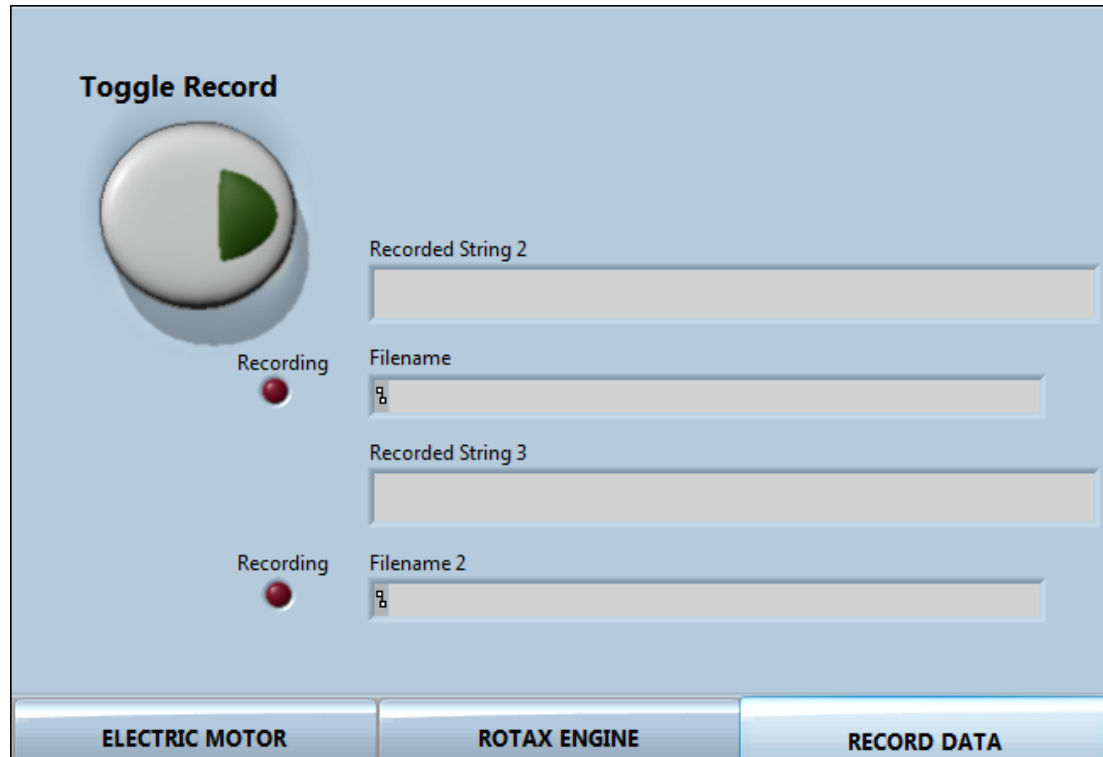
**Figure 34: Electric Motor Avionics Display**

The "Rotax Engine" tab contained information regarding the cylinder head temperatures (CHT) and the exhaust gas temperatures (EGT) for the engine, with the upper limit values displayed in red, as shown below in Figure 35. This tab was meant to encapsulate data that was not otherwise captured by the independent gauges in the cockpit for the gas engine. As such, these values are not constantly monitored by the pilot as a good indication for these can be obtained from the stand-alone coolant exit temperature gauge. This data was monitored on the screen to ensure that the presence of the electric motor was not hampering the functionality of the engine in any form.



**Figure 35: Rotax Information on the Touchscreen**

The last tab on the interface was to allow the pilot to record the data captured from the electric motor and the motor controller and the temperatures from the engine in an Excel file, as shown in Figure 36. The software could possibly be modified such that the data is recorded from the start of the program, but this could possibly cause the memory on the touchscreen to run-out before all the required data was captured. Thus, via a programming decision, an optional toggle button was provided to the pilot to record the data. This would split the data into two files, with the first one recording all the parameters and the second one recording only the data from the CAN bus communication with the motor controller. This was done as the data rates varied for the electric motor data and the gas engine data, with each file being recorded at the appropriate rates.



**Figure 36: Data Recording Tab on Touchscreen**

The data captured from this option allows the possibility to better understand the performance of the electric motor.

The data from this software is stored in a comma separated file with the name of the file being date-stamped. The channel listing for the file is listed in Appendix A.

The software to operate the electric motor is described in detail in Appendix B.

## CHAPTER 4: Propulsion System Performance

### Propulsion System Efficiency

The efficiency of the aircraft system due to presence of the parallel propulsion system is no longer dependent just on the engine and the propeller, but is now dependent on all the following systems working together:

- Electric motor
  - Freewheeling ( $\eta_e$ )
  - Switched on ( $\eta_o$ )
- Gas engine ( $\eta_R$ )
- Battery System ( $\eta_B$ )
- Drive train ( $\eta_D$ )
- Propeller ( $\eta_P$ )

The goal though is not to attain an aircraft that is just fuel efficient based upon its flight alone, but rather how much power it requires on the ground and pulls from the “grid” along with the amount of AVGAS it burns. Thus, to calculate the overall efficiency now, the following components must be considered as well:

- Motor control unit ( $\eta_M$ )
- Charging unit ( $\eta_C$ )

The efficiency equation for the propulsion system on the Eco-Eagle can now be set up as follows for the case where the gas engine is enabled:

$$\eta = \eta_e * \eta_R * \eta_D * \eta_P \quad \text{Equation 1}$$

where the drivetrain efficiency ( $\eta_D$ ) is dependent on the following components:

- Clutch ( $\eta_L$ )
- Drive shaft ( $\eta_S$ )
- Pulleys in the front ( $\eta_{pf}$ )

Thus,

$$\eta_D = \eta_L * \eta_S * \eta_{pf} \quad \text{Equation 2}$$

In the case of the power being supplied by the electric motor, the gas engine is no longer part of the system providing propulsive output. Thus, Equation 1 now changes as follows:

$$\eta = \eta_o * \eta_B * \eta_D * \eta_P \quad \text{Equation 3}$$

But, in this case, the drive train now includes a set of pulleys in the aft ( $\eta_{pa}$ ) as well, to help deliver the power output from the motor to the drive shaft and also, the clutch is engaged and hence is no longer rotating as part of the system. This causes the equation for the drivetrain to vary as shown below:

$$\eta_D = \eta_{pa} * \eta_S * \eta_{pf} \quad \text{Equation 4}$$

### Current System Performance

The battery system efficiency is based upon the efficiency of all the components working together, including the battery monitoring system ( $\eta_x$ ), relays ( $\eta_r$ ) and fuses ( $\eta_y$ ). Heat loss occurs through all these components, thus reducing the overall power conversion from the electrical energy from the cells to the mechanical energy output at the motor shaft. Along with this, there is also a power loss due to the heat generated as the power is inverted from a single-phase DC input to a three phase output for the electric motor from the motor control unit that must be accounted for ( $\eta_z$ ).

$$\eta_B = (\eta_x)^{\# \text{ of BMS boards}} * (\eta_r)^{\# \text{ of relays}} * (\eta_y)^{\# \text{ of fuses}} * \eta_z \quad \text{Equation 5}$$

Unfortunately, at this time, the only efficiencies known are those for the gas engine ( $\eta_R = 34\%$  maximum), and the propeller ( $\eta_P = 87\%$  maximum). The other values

must be determined by individual tests to determine the overall efficiency of the propulsion system.

The only information that is currently available for the efficiency and performance of the Eco-Eagle is from the results from the Green Flight Challenge 2011, as shown below in Table 14 and Table 15.

**Table 14: Results from GFC Fuel Efficiency Flight**

Fuel used	3.82	Gallons 100LL
Energy used	3.8	kWh
Equivalent fuel used	4.1	Gallons (auto fuel)
Flight time (for speed)	2:00:48	Time
Flight time (for mileage)	2:04:07	Time
Distance (for speed)	142.5	Miles
Distance (for mileage)	148.1	Miles
Mileage	72.2	ePMPG
Speed	70.7	MPH

**Table 15: GFC Results from Speed Flight**

Fuel used	4.19	gallons 100LL
Energy used	3	kWh
Equivalent fuel used	4.47	gallons (auto fuel)
Flight time (for speed)	1:43:21	Time
Flight time (for mileage)	1:44:53	Time
Distance (for speed)	143.9	Miles
Distance (for mileage)	146.2	Miles
Mileage	65.5	ePMPG
Speed	83.5	MPH

## CHAPTER 5: CONCLUSION AND SUGGESTIONS

### Conclusion

The system described above was implemented on the Eco-Eagle and flown (Figure 37) in the Green Flight Challenge after a couple of weeks of flight testing at the Eagle Flight Research Center at ERAU in the fall of 2011.

Initial runs of the electric motor on the ground proved to be promising for the flight profile selected for the aircraft, but the results could not be recreated in flight. Via troubleshooting the system, it was determined that the firmware installed on the MCU at the time would not allow the motor to be enabled if the shaft RPM exceeded 600 RPM. This issue was fixed by installing an upgraded firmware, (created by Drivetek AG) which allowed the motor to be enabled as long as the RPM on the shaft was below 2500 RPM.



**Figure 37: Eco Eagle in Flight**



The drive system created for the aircraft did not work as predicted under the current configuration, due to limitations within the motor controller causing the motor to spin up to 3000RPM only.

### **Future Work and Suggestions**

Upon installation of the new firmware, the motor started in flight, but unfortunately, the pilot was unable to obtain the maximum possible torque output from the motor. The motor would stop each time at roughly 30% demanded torque and the RPM would stagnate at this point, even if the demanded torque was increased. It was verified with the diagnostic software that this issue was not the result of the LabView interface created. At this RPM and torque, the aircraft currently has a 100 foot per minute descent.

In order to overcome the issue of the limited torque, Drivetek AG had been contacted to verify the firmware for the MCU. For future variations of this project, once this issue has been fixed, the aircraft should be thoroughly flight tested to gather data based upon the motor performance. This would provide a good basis for understanding the possibility of an all-electric aircraft. To fix the issue of the RPM and torque being limited, via troubleshooting it was determined, at the time, that the cause for this was related to the current phase being produced as a result of the field oriented control system in the electric motor. The software included for diagnostic purposes with the motor controller can be utilized to assist with this along with the recommendations from Drivetek AG. Using the software, the "Index Angle Offset" for the motor must be modified such that the current phase for the motor is such that the voltage difference measured for the phases after the mathematical conversion to the axis along the rotor is zero. This can be viewed in the motor controller diagnostic software as " $U_{dqAbs}$ ". This offset angle allows the motor to run in the correct direction at the required torque while keeping the three phase current within limits for safe operation. As the values for the offset can vary greatly, a starting point to ensure the correct value is listed on a label on the electric motor. Also, if the index offset value is incorrect, it would cause the motor to spin in the wrong direction at certain times. This issue was encountered on the

ground during testing in the form of an intermittent spin direction change upon system startup. Thus, correcting the offset index value should alleviate the two issues simultaneously.

To allow for a more reliable system for running the electric motor, a ground initialization procedure could be programmed within the current LabView interface software, within which the safety system values for the motor including parameters such as the offset index, spin direction, temperature, current and voltage limits and their associated gains could be sent to the motor control unit. This would ensure that the motor always starts with the same initial conditions, thus removing any intermittent startup issues associated with the internal safety system for the motor. As this channel can be setup in LabView as a single time, “run-at-startup”, initialization command, following the battery monitoring system initialization, the safety values would remain constant throughout a run of the motor without allowing the pilot to alter them by error.

Along with this, it would also be highly recommended to re-create the firmware for the MCU in-house, thus making it easier to make further modifications to the operation of the system. The method for communication between the motor controller and an interface should be kept as a CAN bus architecture, as this robust architecture allows for easy modifications as well as reliable communication between the systems. The CAN bus system is programmable in various programming languages. In the case of the Eco-Eagle, it could also be programmed via LabView, thus providing a more robust system where the communication between the various real-time systems can be better controlled. The hardware installed in the aircraft currently would allow for such a modification to the communication system without the requirement of new parts.

The cooling system on the aircraft needs to be improved as well, once further performance data is collected, as the cooling of the system is currently inconsistent and it may overheat on a hotter day. The current cooling system does not allow the pilot to control the amount of air intake thorough the side doors. The possibility of closed doors

in flight while the electric motor is running would reduce the drag on the aircraft drastically and would improve the overall efficiency for the airplane.

The belt system for the pulleys in the front needs to be verified as the belts have on multiple occasions slipped in flight. This can be attributed to the belts being slightly loose as a fixed torque value was not used to tighten the belts. Also, as part of the changes in the propeller system, the pulleys were moved from their original positions on the Stemme S-10, thus, re-sizing the pulley may be another option to overcome the same barrier.

Along with this the battery system needs to be made more robust and sturdy, as the current configuration causes certain communication cables to come loose during installation of the battery trays into the wings. By changing the type of connectors and providing a more slack with the connections, this issue can be tackled.

To allow improvement on the current design, tests must be conducted to verify the efficiencies of all the components as mentioned earlier. The efficiency of the electric propulsion aspect of the aircraft can be verified by measuring the charge on the batteries before a flight and then measuring the amount of current drawn from the grid by the charger after a flight to charge the batteries to the same point as that before the flight. This would result in the true fuel consumption for the electric motor.

Thus, with the modifications mentioned above, the propulsion system could be improved upon without modifications to the physical system itself with the current direct-drive layout.

## APPENDIX A: Data Recorded

Table 16: List of Data Recorded

COLUMN NUMBER	PARAMETER RECORDED	UNITS
1	Timestamp	Milliseconds Since Program Start
2	CHT 2	<sup>0</sup> C
3	CHT 3	<sup>0</sup> C
4	Unused	-
5	Unused	-
6	CHT 1	<sup>0</sup> C
7	CHT 4	<sup>0</sup> C
8	EGT 1	<sup>0</sup> C
9	EGT 2	<sup>0</sup> C
10	EGT 3	<sup>0</sup> C
11	EGT 4	<sup>0</sup> C
12	Unused	-
13	Enable Motor Switch	Boolean (On / Off)
14	Electric Motor RPM	RPM
15	Life Counter	-

16	Theta Electric Motor	Degrees
17	Processed Current $I_d$	Amps
18	Actual Torque	N-m
19	Absolute Current, $I_{abs}$	Amps
20	Battery System Voltage	Volts
21	Processed Current $I_q$	Amps
22	Motor Temperature	$^{\circ}\text{C}$
23	MCU Temperature 1	$^{\circ}\text{C}$
24	MCU Temperature 2	$^{\circ}\text{C}$
25	Hallsector Position	-
26	DESAT Error	Boolean (True / False)
27	5V Supply too Low Error	Boolean (True / False)
28	Over Temperature Error	Boolean (True / False)
29	Battery Voltage too Low Error	Boolean (True / False)
30	Over Current I1 Error	Boolean (True / False)
31	Over Current I2 Error	Boolean (True / False)
32	Over Current I3 Error	Boolean (True / False)
33	Bus DC Link Voltage too High Error	Boolean (True / False)
34	MCU Temperature Sensor 1 Fault	Boolean (True / False)

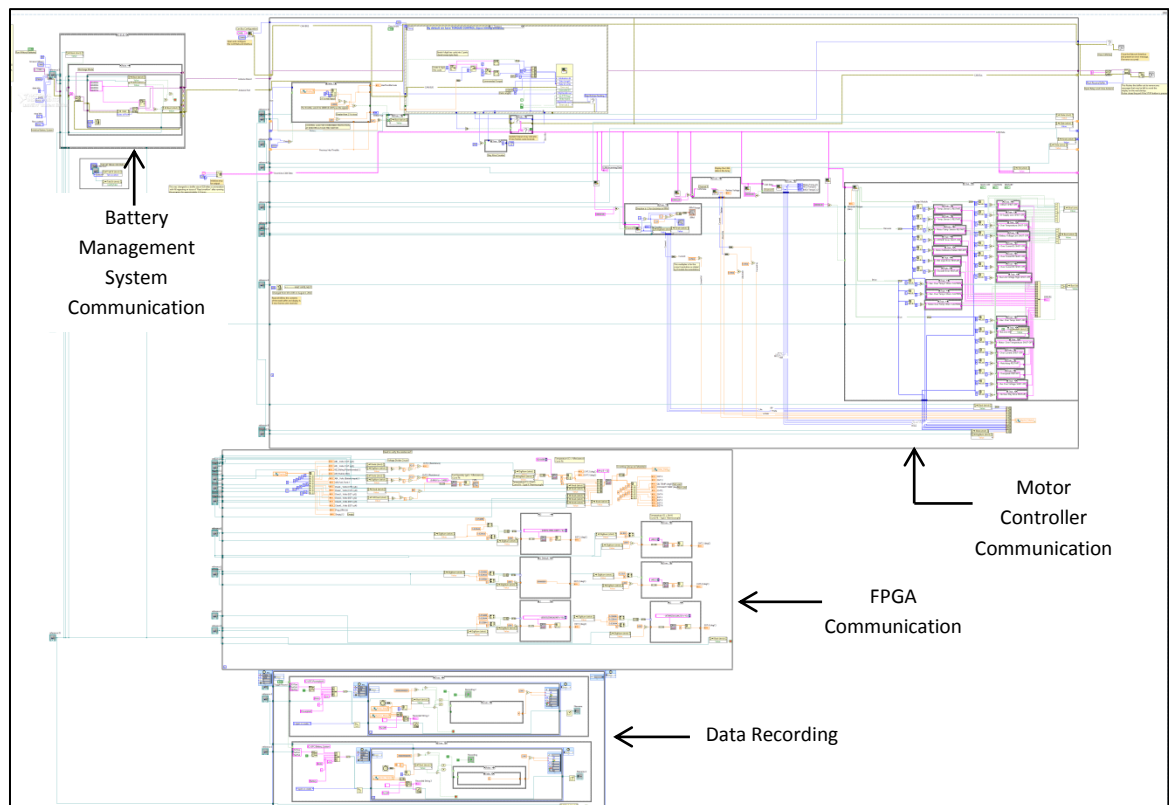
35	MCU Temperature Sensor 2 Fault	Boolean (True / False)
36	Motor Temperature Sensor Fault	Boolean (True / False)
37	MOSFET Error	Boolean (True / False)
38	Motor (Position) Referencing Failed	Boolean (True / False)
39	INX Input in Error State	Boolean (True / False)
40	Encoder Error	Boolean (True / False)
41	Unused	-
42	MCU Over Temperature Warning 1	Boolean (True / False)
43	MCU Over Temperature Warning 2	Boolean (True / False)
44	Motor Over Temperature Warning	Boolean (True / False)
45	Unused	-
46	Unused	-
47	Unused	-
48	Unused	-
49	Unused	-
50	MCU Over Temperature Error	Boolean (True / False)
51	Bus Under Voltage Error	Boolean (True / False)

52	Motor Over Temperature Error	Boolean (True / False)
53	Over Current Error	Boolean (True / False)
54	Watchdog Error	Boolean (True / False)
55	Over Speed Error	Boolean (True / False)
56	Bus Over Voltage Error	Boolean (True / False)
57	No New Stay Alive Message Error	Boolean (True / False)
58	Throttle Position	0 – 100 %
59	Nominal Torque	N-m

## APPENDIX B: Electric Propulsion Control System Software

The control system software was implemented in a LabView environment, utilizing FPGA (Field Programmable Gate Arrays) programming and a CAN (Communication Area Network) bus system. The FPGA sub-section of the program was meant only for the conversion of analog to digital signals. This assisted in the reading of the position of the switch to enable the electric motor as well as obtain a digital value for the position of the torque controller for the electric motor. The software was divided into four major parts:

- a) Battery Management System Communication
- b) Motor Controller Communication
- c) FPGA Communication
- d) Data Recording

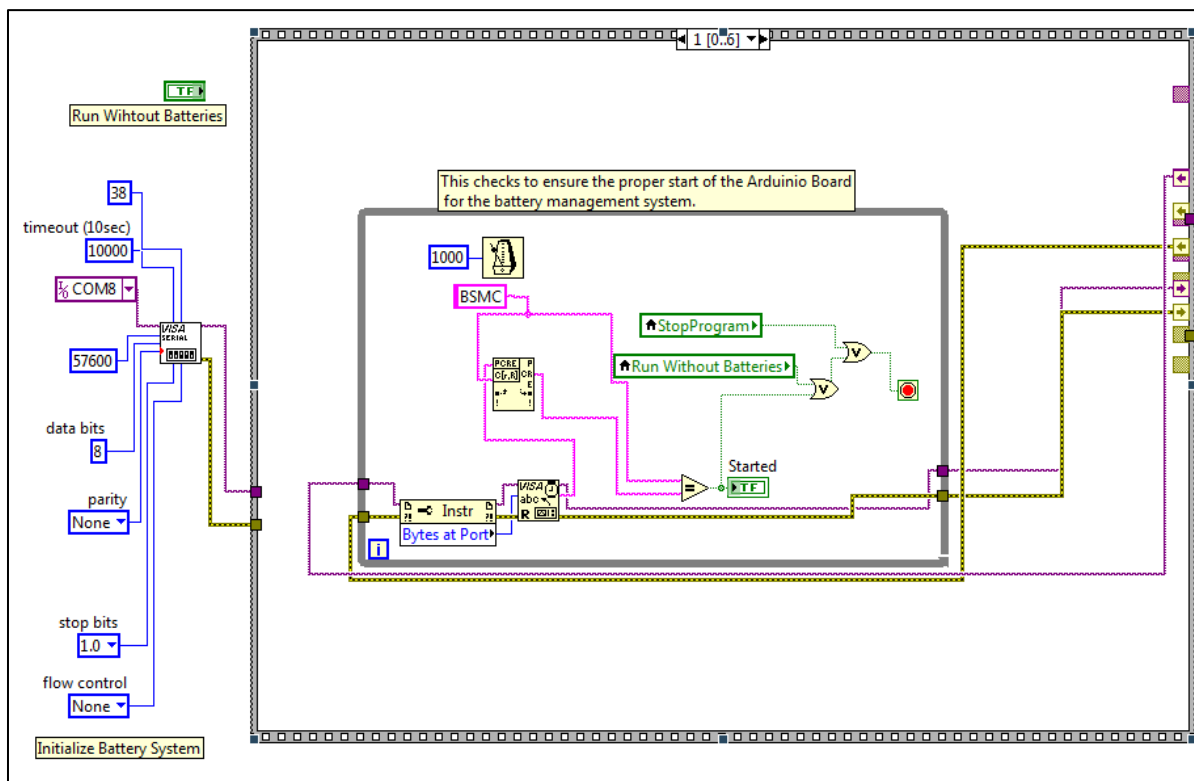


**Figure 38: LabView Program Block Diagram**



## Battery Management System Communication

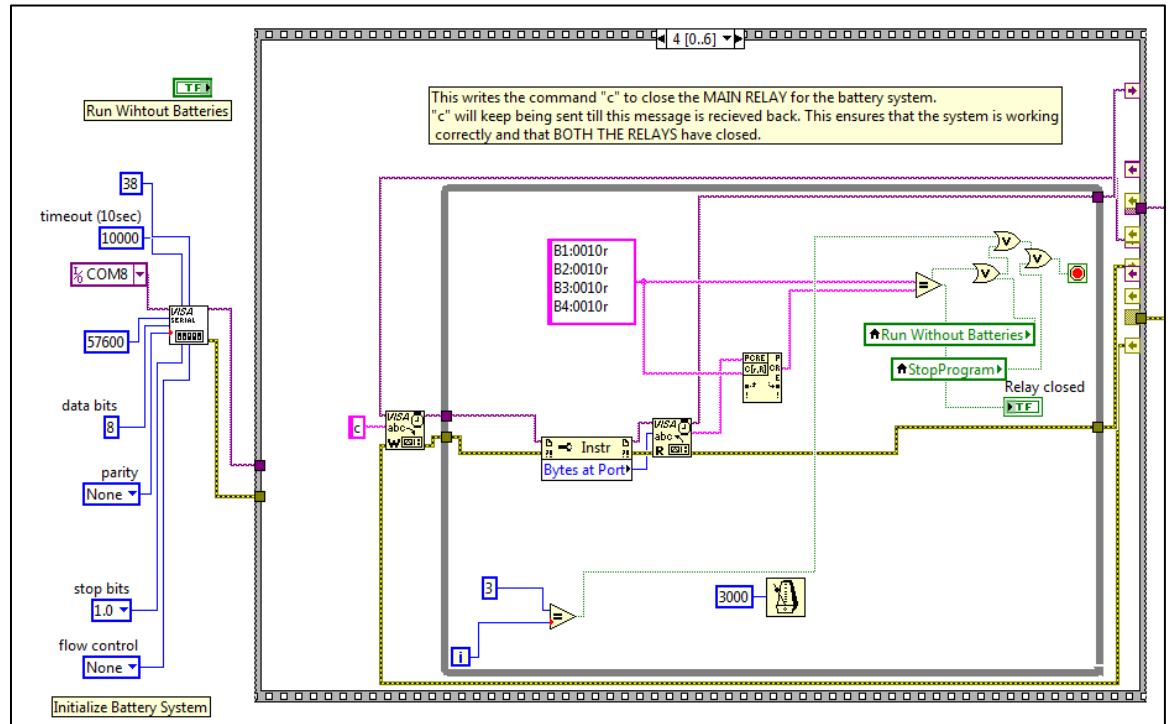
Once the pilot runs the program on the touchscreen, the first section to execute is the battery management system communication protocol. This step consists of a sequence of steps being executed starting with the verification of the correct start of the BMS, as shown in Figure 39. This set of commands for communication with the BMS is accomplished through the master controller board at a baud rate of 57600.



**Figure 39: BMS Initialization Sequence**

If the controller is initialized, it moves onto enabling the “Relay Mode” for the BMS, at which point a message is passed to close the main relays. As the relays may take a different amount of time, each time the code is executed due to the physical parameters involved within the electronic system, a loop is executed to verify the state of these relays. Using a basic string comparison, if within three read loops of the incoming data the relays are not found to be closed, a message is sent to detect their

status. This method ensures that the correct message is read, if the relays are closed, as shown in Figure 40.



**Figure 40: Closing Main Relays**

The message returned for the proper closing of the relays is as shown below:

B1:0010r

B2:0010r

B3:0010r

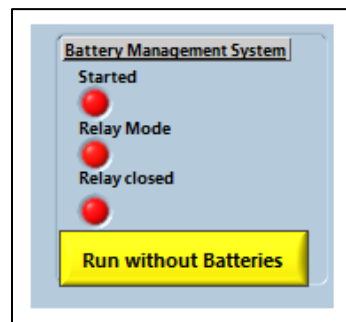
B4:0010r

This shows the status of the relays for the four battery modules. The message can be decrypted as follows:

The numbers following the “B” are a binary code to indicate the open (0) or close (1) position, except for the third position digit, which is a non-binary value:

- Master relay position indication.
- Error reason- this includes error codes related to the possible issue with a relay not closing upon command or a short existing in the system or an open wire in the loop.
- Indication for position of module main relay.
- Indication of position on module transitional relay.

If the relays close, a message is transmitted to enable the discharging mode for the battery system. The success for these steps is indicated by three corresponding lights (Figure 41) that turn green.



**Figure 41: BMS Status Indicator**

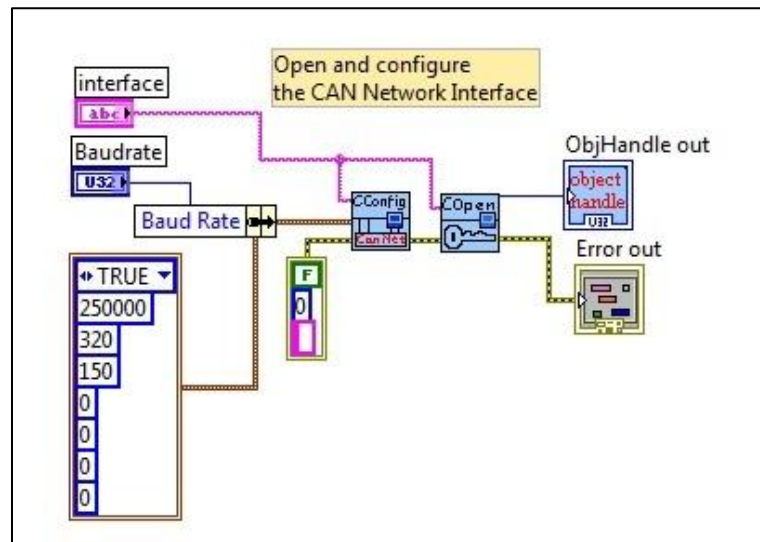
In the event that the BMS does not function correctly or the system needs to be run without the batteries functioning, a facility is provided to allow the sequence for the battery management system to be skipped. This can be accomplished by pressing the button marked "Run Without Batteries" (Figure 41). This stops all the loops within the sequences and exits from the battery management initialization phase without taking any action. This also provides the ability to run the system with an external power source, while the aircraft is on the ground with the same software.

### **Motor Controller Communication**

Upon successful execution of the initialization and activation of the BMS or if the system is run without batteries, the communication with the motor control unit is enabled. The MCU communicates with the interface program in 100ms intervals.

## CAN Bus Initialization

The MCU interacts with the touch screen system and the electric motor via a CAN bus system. In order to allow a communication path to be set between the MCU and the touch screen, the NI CAN bus module has to be initialized as shown in Figure 42 below.



**Figure 42: Enable CAN Bus Communication**

The data speed for interfacing with the firmware on the MCU is set at 250 kbps.

LabView allows for multiple CAN bus modules to be utilized within the same program, and hence, it is extremely crucial to set and select the right interface module while initializing the CAN bus communication system. The interface selection was locked by not allowing the end-user of the software to have the ability to select the interface number. This value was set within the background within the code.

If the system is not initialized correctly due to a hardware or software related error, the program halts and must be restarted once the issue has been rectified. The most common error encountered within this has been the CAN bus module being unplugged from the touchscreen.

If no error occurs, the program starts to ping the motor control unit. As the CAN bus protocol utilizes serial communication cables, the read and write functions are performed in a continuous loop consecutively.

### **Write Function**

The interface program developed for the touchscreen system allows the pilot to command the electric motor by demanding a torque using a sliding potentiometer. As mentioned earlier, no command can be sent using this graphical user interface to edit the security parameters for the motor.

The arbitration IDs (channel IDs) utilized to demand torque from the motor and clear any possible errors that may exist for the MCU are channel 191 and channel 610. The MCU has an in-built safety feature within which if a constant updating message stream is not written/transmitted to the MCU from computer interface, it assumes that the motor is offline and stops transmitting any messages to the electric motor.

The write/command function for the interface software is divided into two parts:

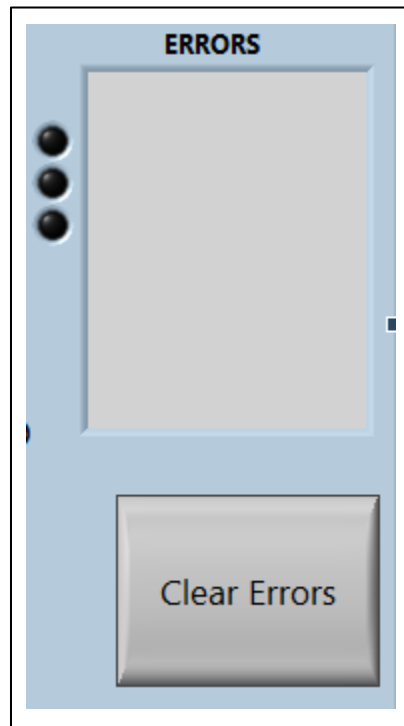
- a) Enable Torque Control
- b) Clear Errors

The two parts are implemented in the form of a “CASE Structure”, with the default case executing a loop for the sending a torque command message to the electric motor for the arbitration ID 191.

LabView provides the option to execute a case structure with a default mode known as “Timeout.” If no case is true, then the timeout loop is executed. Due to the requirement of a constant message being sent to the motor controller to keep its communication with the electric motor switched on, the default mode was setup to transmit such a message at 100ms intervals.

The case structure also allows an instance to be executed, just once, if a certain value changes, in this case, if the Clear Errors button is engaged, as shown in Figure 43,

below. Once this case is executed it goes back to the default timeout mode, unless another case is true.



**Figure 43: Clear Errors Option**

### ***Enable Torque Control***

The electric motor can be commanded to run two different modes:

- a) **Torque Control:** Within this control mode a certain torque value can be demanded from the electric motor with a certain speed value acting as a limiting agent. The maximum torque that can be produced with the given motor is 50 N-m. This value can also be limited by varying the associated safety parameters within the diagnostic software.
- b) **Speed Control:** This mode allows a certain speed to be commanded with the torque value acting as a limiter. Based upon ground tests run with the motor, this mode did not produce results any different from the torque

control mode and it was determined that even within this mode the end user would only be able to command a torque rather than speed. This was attributed to the firmware that was pre-loaded on the MCU as the results were the same with running the system using the diagnostic software and the software created for the touchscreen interface.

As mentioned earlier, a CAN bus message for an arbitration ID is transmitted as an 8 byte message. The message is sent as a combination of hexadecimal values for the MCU with the various byte positions signifying different commands. For the arbitration ID 191 the message is constituted of the following:

**Table 17: Arbitration ID 191 Message**

BYTE POSITION	FORMAT	CONTENTS
0	8 bit Unsigned Integer	State
1	8 bit Unsigned Integer	Stay Alive Counter
2	8 bit Unsigned Integer	Unused
3	8 bit Unsigned Integer	Unused
4 – 5	16 bit Integer	Torque Value
6- 7	16 bit Integer	Speed Value

#### **Deconstruction of the Message in Arbitration ID 191**

The byte position 0, “State” contains further data, regarding the mode of operation for the electric motor:

**Table 18: Motor State Definition**

BIT POSITION	CONTENTS
0	0 = Disable motor
	1 = Enable Motor
1	0 = Torque Control

	1 = Speed Control
2	1 = Set Torque to Zero
3 - 4	00 = Auto Mode
	01 = FOC Control
	11 = BC Control
5	Reference Motor
6	Switch Main Relay On
7	Unused

The first bit position (position 0) allows the motor to be enabled or disabled. To allow the pilot to command this, a physical “Enable Motor” toggle switch (Figure 44) was incorporated.



**Figure 44: Enable Motor Switch**

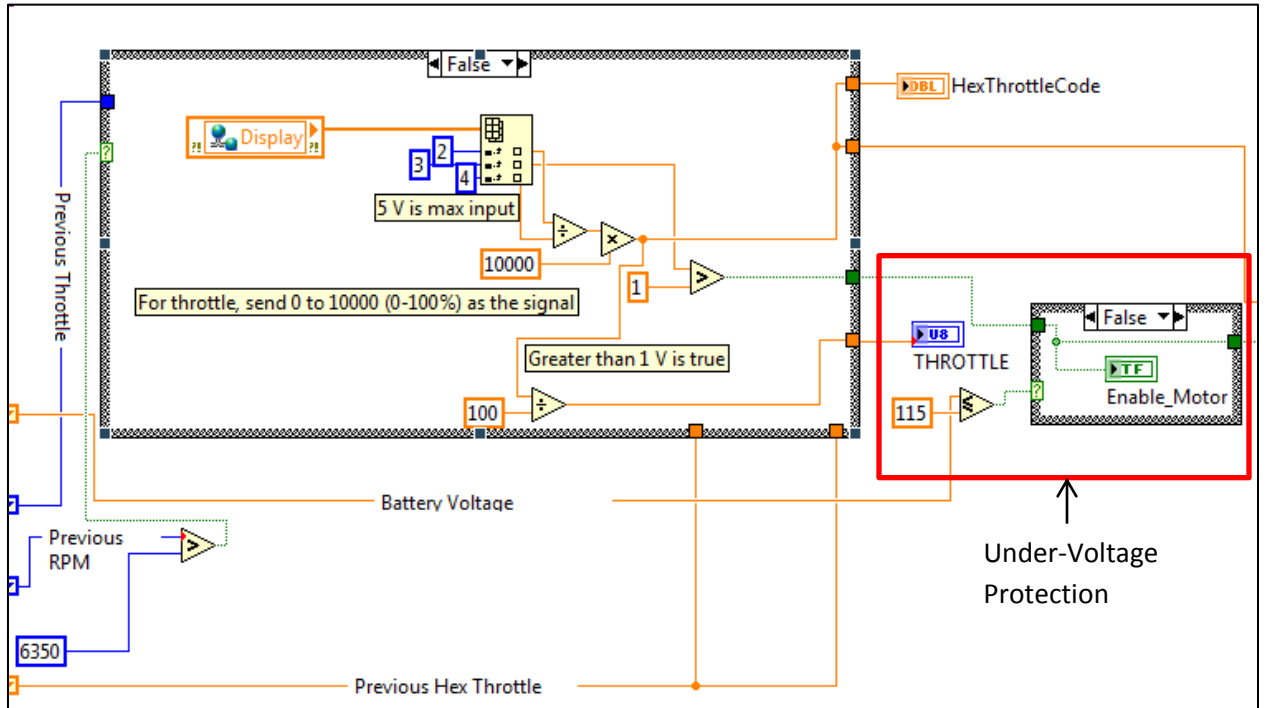
The incoming 12V signal was transformed to a 5V signal using a step-down power converter (Figure 45) before reaching the “Enable Motor” switch.





**Figure 45: 12V DC to 5V DC Converter**

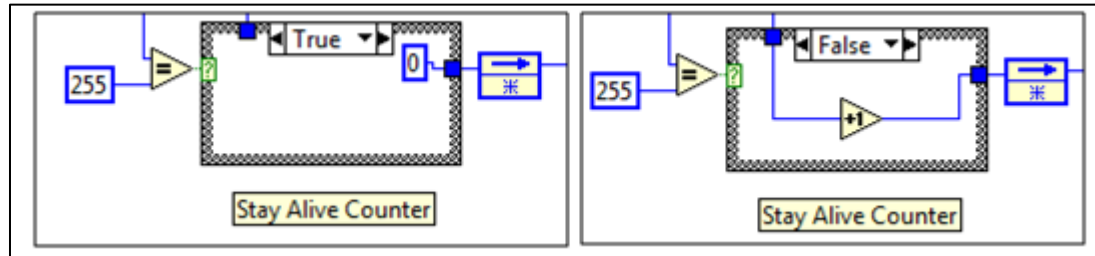
This signal is read using an analog to digital converter module on the cRio, and then stored in a variable (Channel 3). If the value for the variable is greater than 1V, the system is considered enabled, and the appropriate message (0 or 1) is then sent to construct the message for channel 191. In order to increase the safety of the system, before the motor is enabled, a check is performed to verify the last known battery system voltage. This is done so that in the event of a failure of the health-monitoring of the battery modules, if the voltage reaches the safe lower limit (115V), the system is automatically cut-off by disabling the electric motor, as shown in Figure 46 below.



**Figure 46: "Enable Motor" Implementation**

The variable "Display" in the figure above is a network-shared variable that contains the digital signals from the cRio. Channels 2, 3 and 4 contain the signals for the overall reference 5V signal, enable motor switch position voltage and the sliding potentiometer voltage (torque controller) respectively.

The second byte position refers to the "Stay Alive Counter Value," which as mentioned earlier, must be updated constantly to keep the communication system alive with the MCU and the electric motor. In order to accomplish this, the counter is sent as part of the message, where it is updated in every loop, from 0 to 255, as shown below in Figure 47, using a feedback loop with the counter value as the feedback variable for comparison.



**Figure 47: Stay Alive Counter**

The message is limited from 0 to 255 as the MCU accepts only hexadecimal messages, thus limiting the range of messages. If this counter is not updated every second, the MCU will stop transmitting messages to the electric motor. This is done so that the motor does not run while no torque or speed is being commanded. This time interval can be changed using the diagnostic software.

The following two byte positions are currently unused and could potentially be used if the firmware is modified to carry data within these bytes of channel 191. This implies that if the firmware software was to be modified, other data could also be sent/received within the unused bits of this channel.

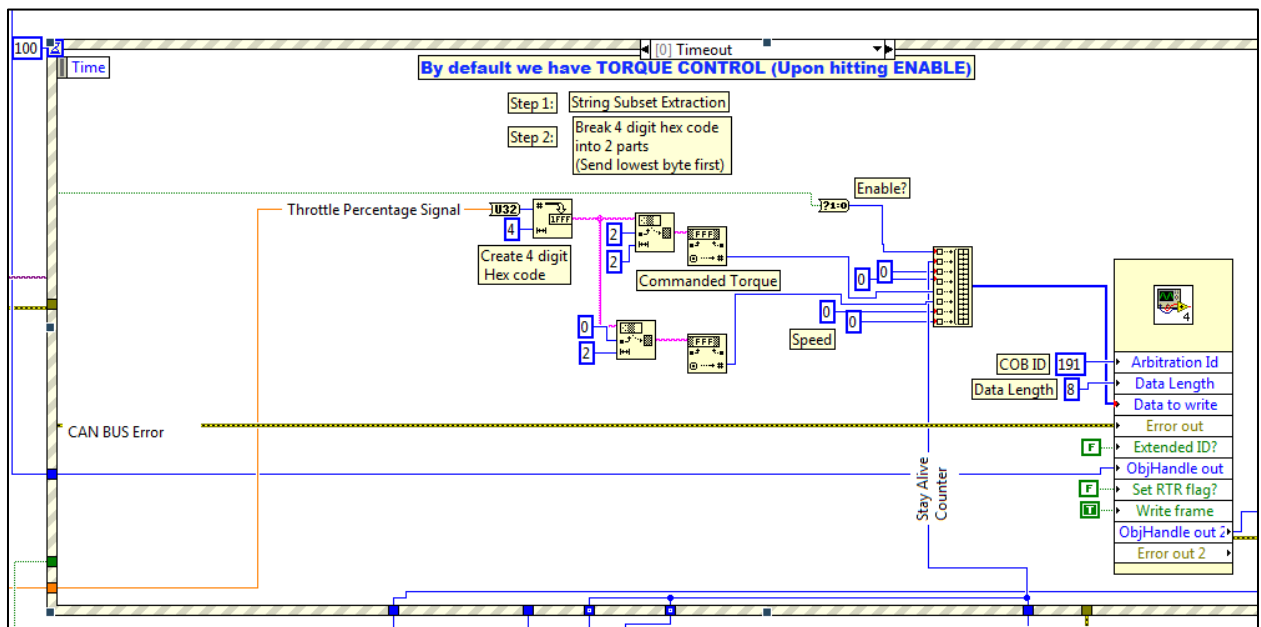
The torque command message is sent as a percentage of the maximum allowable torque (50 N-m). As the value varies from 0% to 100% and the system has a resolution of 0.01%, the commanded percentage is scaled by 10000 as a decimal value cannot be sent using a hexadecimal system. The value is converted to a 4-digit hexadecimal equivalent before being sent to construct the message.

The last two bytes, containing the speed setting for the electric motor have to be sent as a combined 4-digit hexadecimal as well. The resolution for the speed on the electric motor is 1 revolution per minute.

Based upon the CAN Open standard, the multi-byte data message is created with data objects (bytes) joined together as a sequence of consecutive bytes, starting with the least significant byte. For interacting with the firmware on the MCU, if a message

consists of two consecutive bytes forming a single 4-digit hexadecimal value, the message is sent with the bytes inverted. Example:

- Integer Value: 2500
- Equivalent Hexadecimal Value = 9C4
- Equivalent 4-digit hexadecimal value = 09C4
- Message sent in 2 consecutive bytes:
  - First Byte Position = C4
  - Second Byte Position: 09



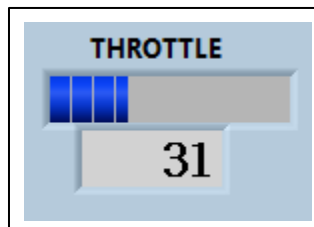
**Figure 48: Arbitration ID 191 Message Construction**

Thus, for example, a message for arbitration ID 191 to enable the motor with torque control, with automatic FOC mode and a demanded torque of 25% and 7000 (hexadecimal equivalent = 1B58) RPM as the speed limit would be as follows:

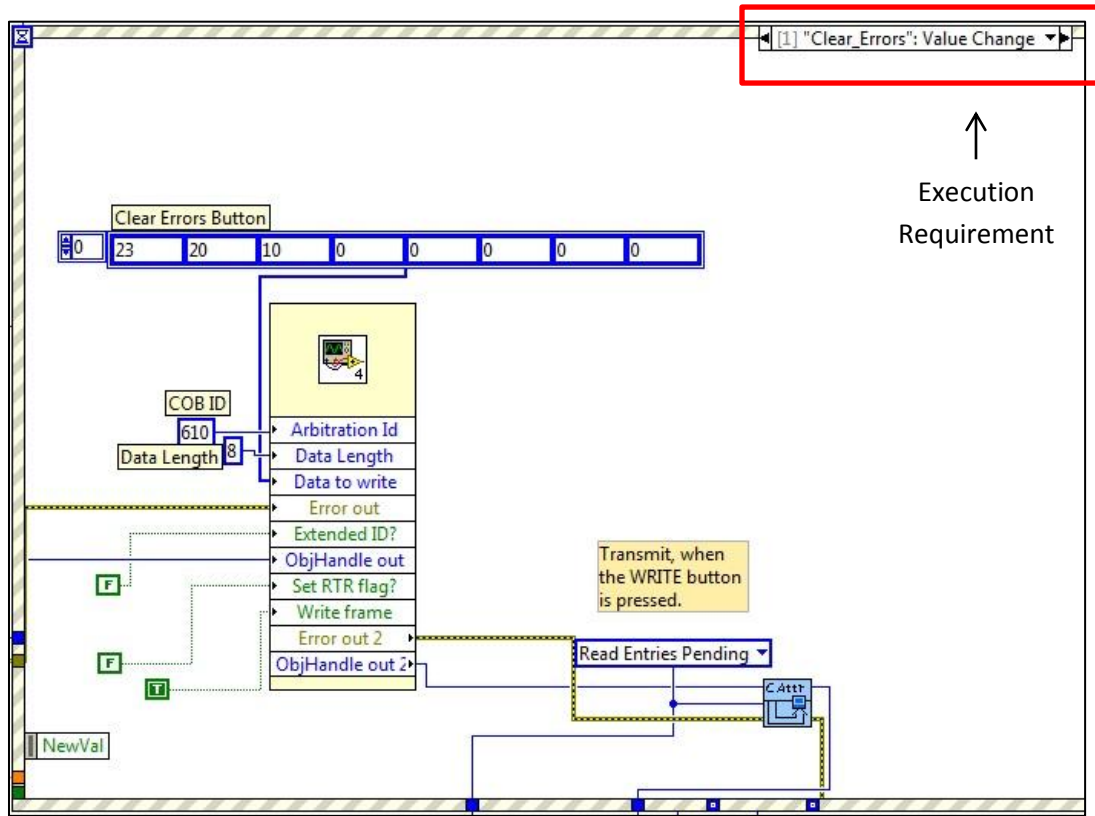
**Table 19: Example Transmitted Message for Torque Control**

BYTE POSITION	CONTENTS	VALUE
0	State	01
1	Stay Alive Counter	0 – 255 (Incrementally, in hexadecimal)
2	Unused	00
3	Unused	00
4	Torque Value Position 1	C4
5	Torque Value Position 2	09
6	Speed Value Position 1	58
7	Speed Value Position 2	1B

To assist the pilot, the percentage throttle commanded is displayed on the graphical user interface, as shown in Figure 49.

**Figure 49: Throttle Percentage Display*****Clear Errors***

The motor controller continuously monitors various performance parameters for the electric motor. This allows the system to take appropriate actions based upon how the motor is running, which allows for its safe operation. This mode is executed, as mentioned earlier, only when the “Clear Errors” button is engaged. This is accomplished using the option to execute a case structure upon a value change occurring in the boolean variable “Clear\_Errors”, as shown in Figure 50.



**Figure 50: Clear Errors Case Structure**

The “Clear Errors” option transmits a message to the motor controller arbitration ID 610 with 8 bytes of data. The arbitration ID 610 is utilized for sending parameters to the motor controller to allow the motor controller settings to be modified. This includes all the parameters defined in the object dictionary for this motor controllers. Index 1020 in the arbitration ID 610 allows a message to be transmitted for clearing any internal errors that may be present on the motor controller. The message for this channel is defined in Table 20 below:

**Table 20: Arbitration ID 610 Message**

BYTE	CONTENTS
0	Command
1 – 2	Index (Object Number)
3	Subindex

4 – 7	Maximum 4 bytes of data
-------	-------------------------

As only the option for clearing the errors was utilized within the interface program and the object numbering and definition is proprietary information, none of the other possible modes have been defined in this document. The message shown in Figure 50 for clearing the errors can be decomposed as follows:

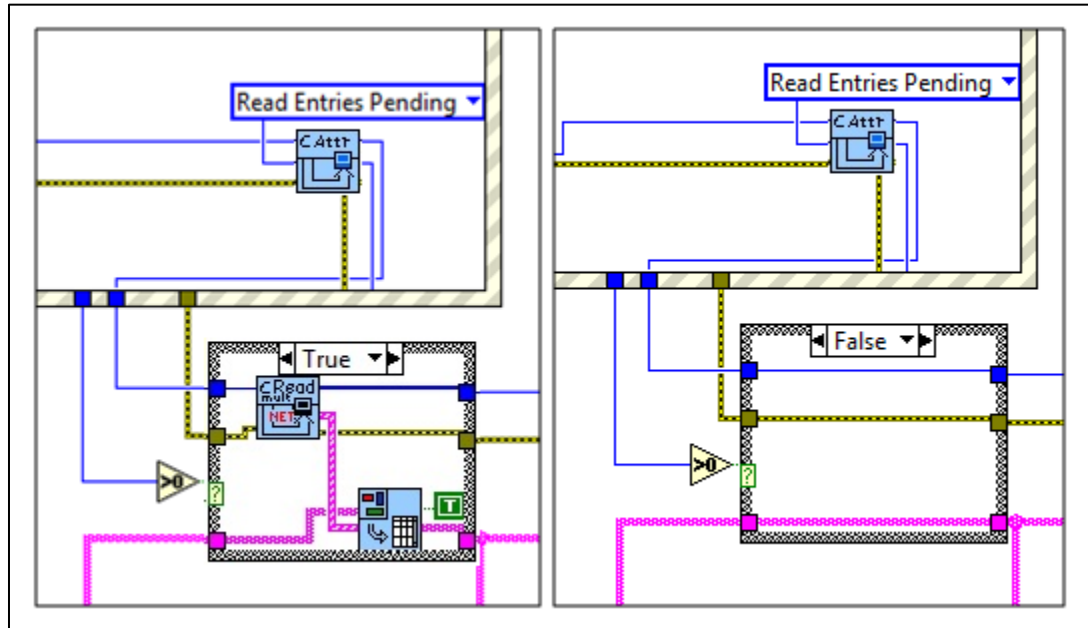
**Table 21: "Clear Error" Message**

BYTE	CONTENTS
0	23
1	20
2	10
3 – 7	0

The byte position 0 defines the command to be executed once the message is received by the MCU. In this case, the hexadecimal value '23' implies a download of the command be requested to the motor controller. The following two bytes refer to the object numbers corresponding to the objects defined for the MCU. Similar to the messages sent for enabling the torque control, the 4-digit hexadecimal values must be broken into two parts and transmitted with the least significant byte first. Thus, bytes 1 and 2 refer to the object number 1020. This object clears any errors that may exist on the MCU at that point.

### **Read Function**

Upon completion of the "write function," any data that may be in the CAN buffer is read and stored in a string output array after the incoming frames are converted to a 2D string array, as shown in Figure 51.



**Figure 51: Read Data and Update Output Array**

The function shown above reads any entries that may be in the buffer and update the output array. In the absence of any new data in the buffer, it updates the output array with the same values from the last known good value in order to provide a hold on the data to view the correct information rather than the data jumping between loops.

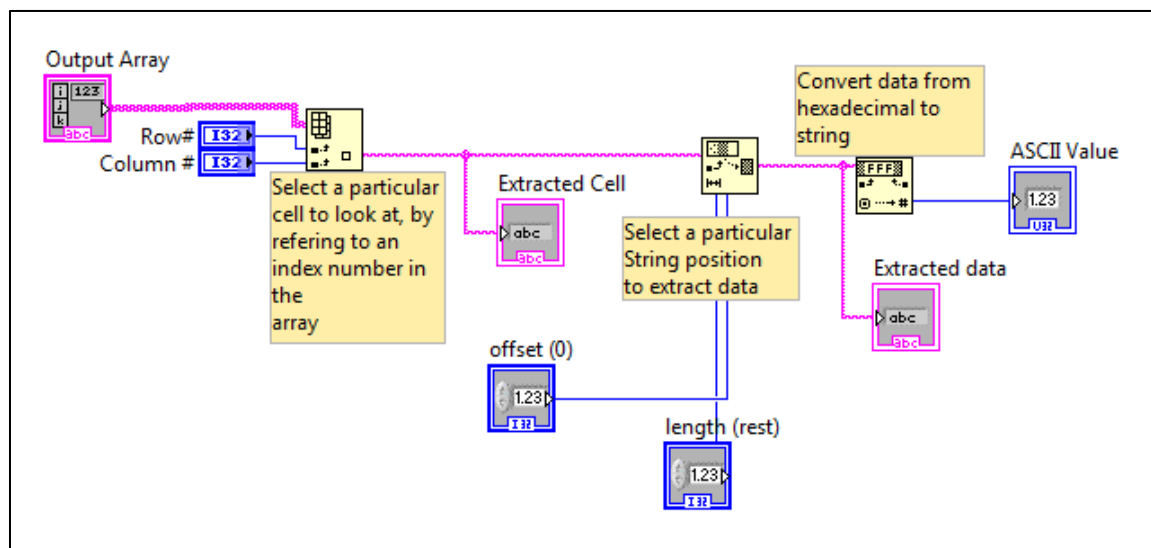
The incoming data stream is divided into four channels, with each containing 8 bytes of information in hexadecimal format. The data from these channels is processed and the required information from these is then displayed on the graphical user interface. The four channels are:

- a) 00000390
- b) 00000290
- c) 00000490
- d) 00000190



The information in the string array is read in a certain manner based upon the channel ID. This is accomplished by performing a basic string comparison for the channel number before stripping and processing the data.

The first step requires the output data to be converted from hexadecimal format to its ASCII equivalent, as shown in Figure 52.



**Figure 52: Data Extraction from Output Array**

Depending upon the channel and the byte position, different actions are performed for obtaining the required data as shown below in Figure 53. This also includes the channel number as one of the bytes of information.

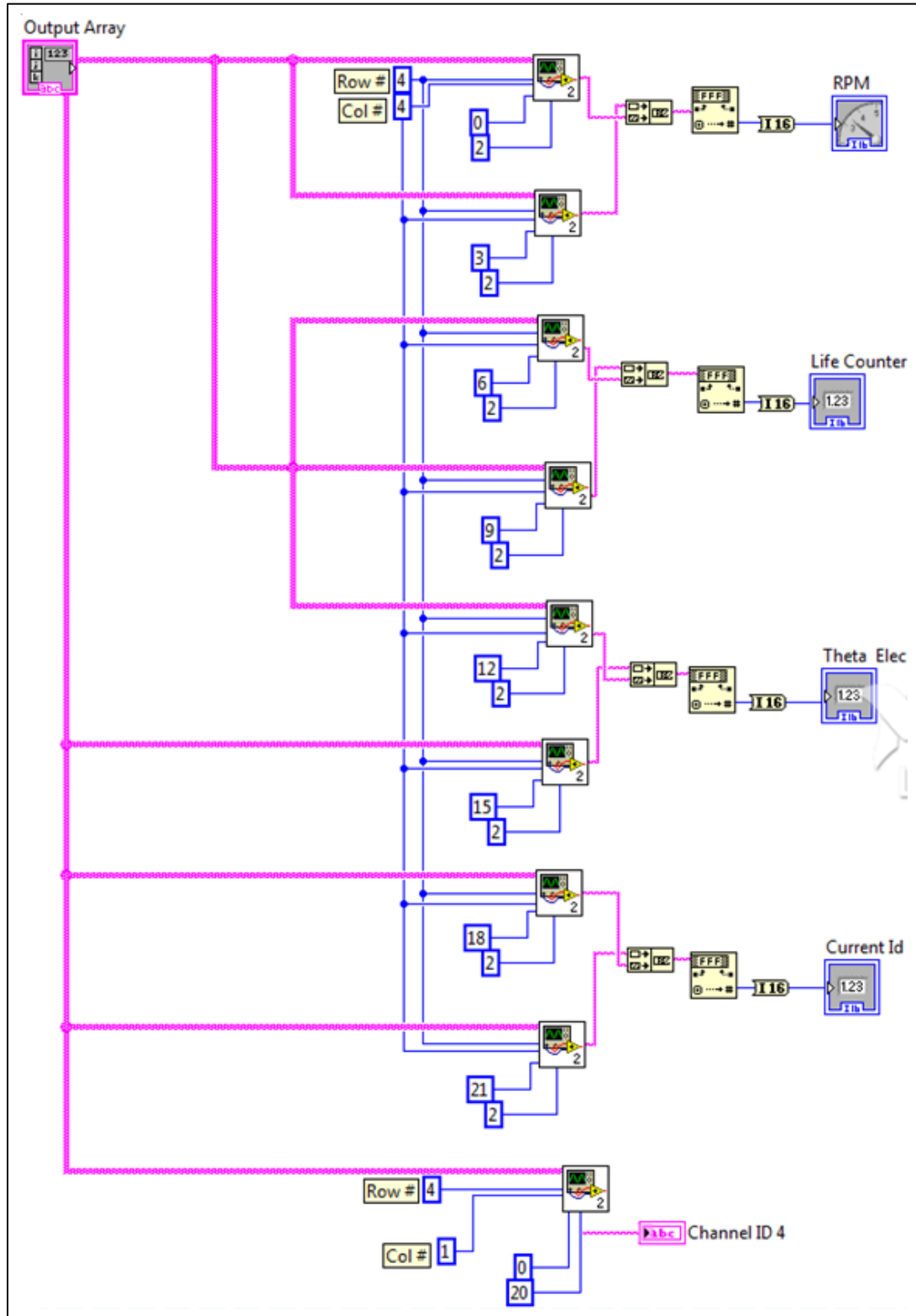


Figure 53: Data Processing for Channel 390

### CAN Bus Channel Listing

Each channel contains 8 bytes of data, with each byte carrying different data. As each byte can carry only 256 bits of data (8 bytes =  $2^8$  bits), if certain bytes are grouped together data can be represented correctly by increasing the limit of the values sent.

**Table 22: CAN Bus Channel Listing**

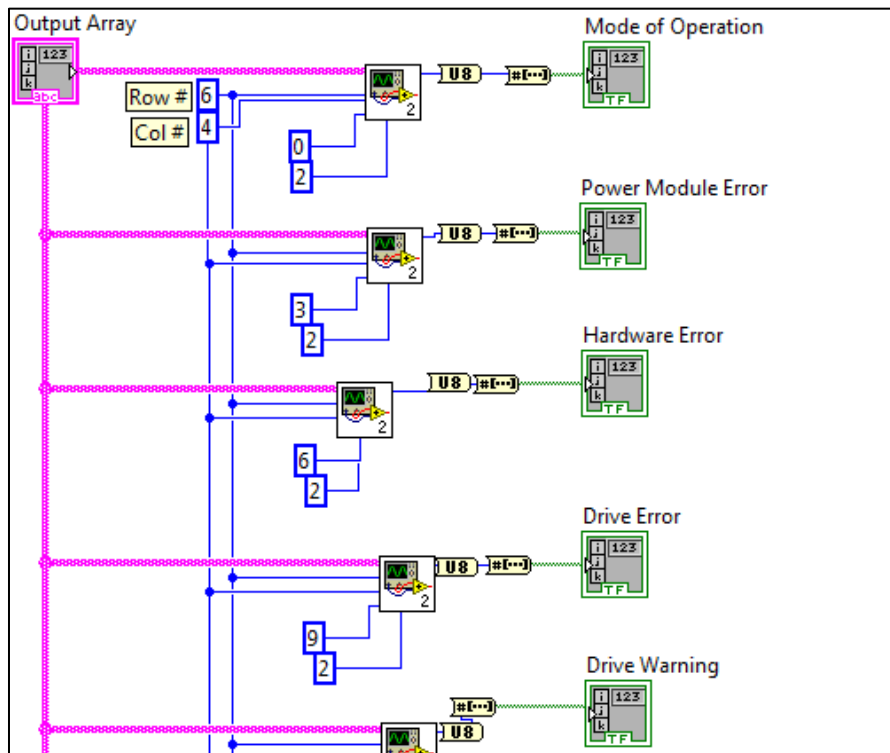
CHANNEL ID	BYTE	RESOLUTION	CONTENTS
00000390	0 – 1	1 RPM	Motor Speed (>0 = clockwise)
	2 – 3	1 unit	Life Time Counter
	4 – 5	1 unit	Theta Electric
	6 - 7	0.0504 A	Current $I_d$
00000290	0 – 1	0.01 %	Actual Torque
	2 – 3	0.0504 A	Absolute Current ( $I_{abs}$ )
	4 – 5	0.1 V	Battery Voltage
	6 - 7	0.0504 A	Current $I_q$
00000490	0 – 1	1° C	Motor Temperature
	2 – 3	1° C	MCU Temperature 1
	4 – 5	1° C	MCU Temperature 2
	6	1 unit	Hallsector position (1 thru 6) 254 = Motor Referencing ON 255 = Index from Encoder Detected
	7	-	Unused
00000190	0	-	Mode of Operation
	1	-	Power Module Errors
	2	-	Hardware Errors
	3	-	Drive Errors
	4	-	Drive Warnings
	5	-	Unused

	6 – 7	0.1 N-m	Nominal Torque
--	-------	---------	----------------

Hexadecimal data has a limited resolution. Once the data has been converted to the decimal (ASCII) format, it must be scaled using the resolution mentioned in Table 22.

### ***Interpretation of Data in CAN Bus Channel 190***

The data carried by the first five bytes of channel 00000190 are further decomposed into a bit format to indicate if an error has occurred via a 0 (false) or a 1 (true). Thus, the incoming data for these bytes is type-casted as boolean arrays as shown in Figure 54, below.



**Figure 54: Data Processing for Channel 190**

### Mode of Operation

This byte contains information regarding the commanded mode of operation for the motor.

**Table 23: Mode of Operation**

BIT	CONTENTS
0 – 2	Drive Mode:
	00 = Off
	01 = Torque Control
	11 = Speed Control
3	Unused
4	Control Mode (FOC or BC):
	0 = Automatic
	1 = Force Control Mode
5	Force Control Mode (If Bit 4 = 1)
	0 = FOC Control Mode
	1 = BC Control Mode
6 – 7	Unused

### Power Module Errors

If any error occurs due to any issues regarding the power unit on the motor control unit, they can be viewed within this byte.

**Table 24: Power Module Errors**

BIT	CONTENTS
0	DESAT
1	5V Supply too Low
2	Over Temperature
3	Battery Voltage too Low

4	Over Current I1
5	Over Current I2
6	Over Current I3
7	Bus DC Link Voltage too High

These errors refer to the power module that assists in the conversion of the incoming DC current to the 3 phase current required by the electric motor.

- Bit 0: The error in bit 0, DESAT, refers to the MOSFET chip not functioning in the saturation region (w.r.t the linear charging region of bipolar transistors). This could cause the system to function incorrectly and hence shut-off.
- Bit 1: This bit refers to the power supply required to run the motor controller and provide power for the CAN communications to take place.
- Bit 2: Bit 2 changes to a true value if the temperature for the power module exceeds its internal safe limit. This limit is dependent on the manufacturer and cannot be edited.
- Bit 3 and Bit 7: Bits 3 and 7 refer to the 12V voltage supply required to power the MCU.
- Bit 4 thru Bit 6: Bits 4 thru 6 refer to the 3-phase current being sent from the MCU to the electric motor after the incoming DC power from the battery modules is converted.

### Hardware Errors

The electric motor and the MCU have internal temperature sensors in order to ensure that the system does not overheat. To assist with this, the motor system is liquid cooled using a 50/50 water/glycol solution. The MCU internally monitors the functioning of the MOSFET junction. A MOSFET or a metal oxide semiconductor field effect transistor is utilized to amplify or switch incoming signals. If the gates for the MOSFET are not functioning correctly, an error occurs and the MCU shuts-off all signals to the motor and switches it off. Similarly, the MCU also monitors the position and state of the

encoder to ensure the proper and safe operation of the motor within this byte. This data is represented within this byte as shown below in Table 25.

**Table 25: Hardware Errors**

<b>BIT</b>	<b>CONTENTS</b>
0	MCU Temperature Sensor 1 Fault
1	MCU Temperature Sensor 2 Fault
2	Motor Temperature Sensor Fault
3	MOSFET Error
4	Motor (Position) Referencing Failed
5	INX Input in Error State
6	Encoder Error
7	Unused

### **Drive Errors**

This byte contains error information regarding the physical performance of the electric motor.

**Table 26: Drive Errors**

<b>BIT</b>	<b>CONTENTS</b>
0	MCU Over Temperature
1	Bus Under Voltage
2	Motor Over Temperature
3	Over Current
4	Watchdog Error
5	Over Speed
6	Bus Over Voltage
7	No New Stay Alive Message

Bit 0 or bit 2 turn to 1 (true) if the temperature for the MCU or the motor goes beyond the safe limit. The safe limit for the temperature can be edited using the diagnostic software with the ground station. For the Eco Eagle, this limit was set to 100 °C for all three sensors.

Bit 1 and Bit 6 refer to the main DC supply from the battery modules to operate the electric motor. The maximum safe limit for the motor is 170V and the minimum, being limited by the current battery module system, is 115V. The current supplied to the MCU from these modules is monitored as well with the error bit 3.

The watchdog error in bit 4 occurs if there are any internal communication errors within the MCU system.

As mentioned earlier, a constant updating message has to be transmitted from to the MCU (“Stay Alive Counter”) or else an error flag is set in bit 7, which causes the motor to stop functioning.

### **Drive Warnings**

In order to avoid the motor from stopping due to parameters such as the temperature, a warning flag can be set to inform the pilot that he/she is approaching the upper limit for safe operation and thus must take action appropriately by either reducing the demanded power output from the motor or switching it off till the temperature reduces to a safer limit.

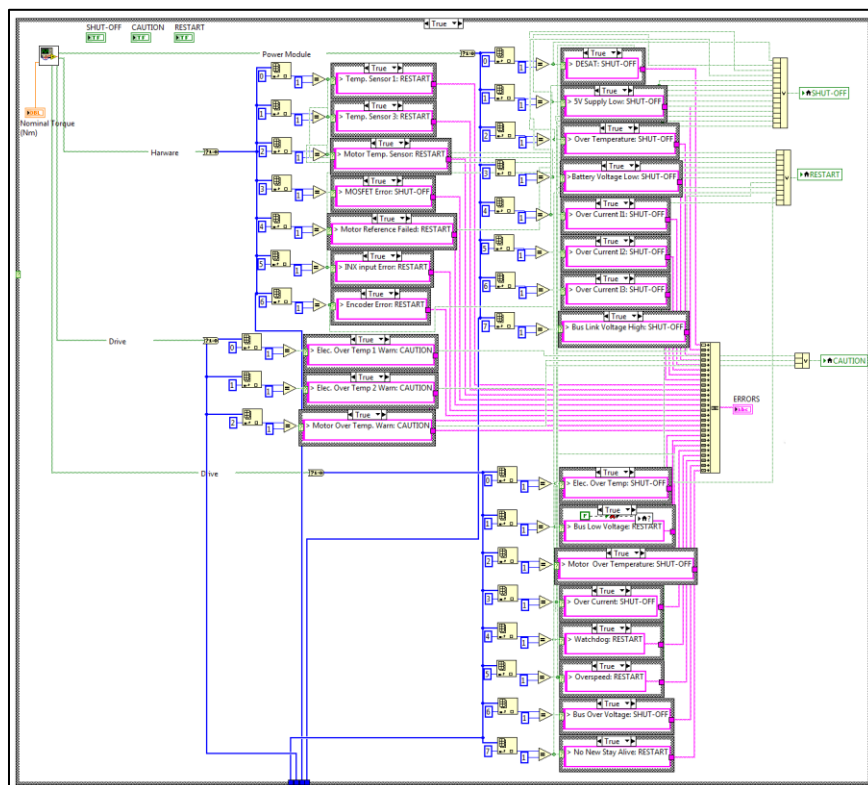
**Table 27: Drive Warnings**

<b>BIT</b>	<b>CONTENTS</b>
0	MCU Over Temperature Warning 1
1	MCU Over Temperature Warning 2
2	Motor Over Temperature Warning
3 – 7	Unused



Thus, using the information described above, a self-monitoring system ensures the safe operation of the motor. To assist the pilot with this and in order to avoid adding to the burden of operation, a graphical user interface was designed accordingly.

If an error occurs, the associated bit turns true and the data is processed accordingly to display the correct message and action required on the screen. This was accomplished as shown below in Figure 55.



**Figure 55: Error Handling and Display**

The errors listed above were categorized according to the severity of the issue occurring into three categories, with each category defining the kind of action required to remove the error:

- a) Shut-Off
- b) Restart

### c) Caution

Each category was also color coded, so that a LED indicator on the screen would assist the pilot in quickly identifying the kind of issue occurring, as shown in Figure 56. Along with the LED, text would also be displayed to identify the exact error along with the category of the error.



**Figure 56: Error Display**

### Nominal Torque

The last byte in this channel (00000190) contains the nominal torque for the electric motor. This refers to the maximum possible torque that the motor can generate. This value can be edited via the diagnostic software with a maximum safe limit of 50 N-m.

### FPGA Communication

The FPGA communication module was created by two students at the Eagle Flight Research Center as part of a related project to read temperature information from the Rotax engine on a ground test station. The module was then incorporated into the current interface software to allow for other analog signals to be read.

Within this, two analog to digital converters were utilized with the CompactRio to allow for voltage readings. Given below is the channel listing for the various incoming signals from these:

**Table 28: CompactRio Channel Listing**

<b>CHANNEL ID</b>	<b>MODULE NUMBER</b>	<b>INCOMING SIGNAL</b>
AI0	Mod 1	CHT 2 Voltage
AI1	Mod 1	CHT3 Voltage
AI2	Mod 1	Sliding Potentiometer Voltage
AI3	Mod 1	“Enable Motor” Switch
AI4	Mod 1	5 V Input Reference Voltage
AI5	Mod 1	Fuel Tank Reading
AI6	Mod 1	CHT 1 Voltage
AI7	Mod 1	CHT 4 Voltage
AI0	Mod 2	EGT 1
AI1	Mod 2	EGT 2
AI2	Mod 2	EGT 3
AI3	Mod 2	EGT 4
AI4	Mod 2	Unused
AI5	Mod 2	Unused

## References

- (1998). AIRCRAFT ELECTRICAL SYSTEMS. In *AC 43.13-1B - Acceptable Methods, Techniques, and Practices - Aircraft Inspection and Repair*. Federal Aviation Administration.
- The CAN and CANOpen, Plus DeviceNet Technology*. (2000). Retrieved from Thaiio.com: PC Interface Hardware: <http://www.thaiio.com/CANinfo.html>
- The Green Flight Challenge 2011*. (2009). (CAFE Foundation) Retrieved 2011, from [http://cafefoundation.org/v2/gfc\\_main.php](http://cafefoundation.org/v2/gfc_main.php)
- About RTCA*. (n.d.). (RTCA, Inc) Retrieved 2011, from <http://www.rtca.org/aboutrtca.asp>
- Arduino. (n.d.). *Introduction to Arduino*. Retrieved from arduino.cc: <http://www.arduino.cc/en/Guide/Introduction>
- Atmel Corporation. (2009). AVR32723: Sensor Field Oriented Control for Brushless DC motors with AT32UC3B0256. Retrieved from Atmel Field Oriented Control for Brushless DC Motors: [http://atmel.com/dyn/resources/prod\\_documents/doc32126.pdf](http://atmel.com/dyn/resources/prod_documents/doc32126.pdf)
- Day, D. (1983). *Electric Aircraft*. Argus Books.
- Design Science. (n.d.). *Brushless DC Motors: AC Motors*. Retrieved from All About Circuits : [http://www.allaboutcircuits.com/vol\\_2/chpt\\_13/6.html](http://www.allaboutcircuits.com/vol_2/chpt_13/6.html)
- ESAcademy, T. o. (n.d.). *CAN Open USA*. Retrieved from CANOpen.us: [www.canopen.us](http://www.canopen.us)
- Formsprag Clutch. (n.d.). *Overrunning Clutches Application Manual*. Warren: Formsprag Clutch.
- Garber, S. (2009). *U.S. Centennial of Flight Commission*. (NASA) Retrieved 2011, from Centennial of Flight: [http://www.centennialofflight.gov/essay/Lighter\\_than\\_air/Beginning\\_of\\_the\\_Dirigible/LTA6.htm](http://www.centennialofflight.gov/essay/Lighter_than_air/Beginning_of_the_Dirigible/LTA6.htm)
- Garber, S. (2011, March 11). *The Beginnings of the Dirigible*. Retrieved from U.S. Centennial of Flight Commission: [http://www.centennialofflight.gov/essay/Lighter\\_than\\_air/Beginning\\_of\\_the\\_Dirigible/LTA6.htm](http://www.centennialofflight.gov/essay/Lighter_than_air/Beginning_of_the_Dirigible/LTA6.htm)
- Gonitzke, M. (2010). *Overrunning Clutch Assembly Manual*. Eagle Flight Research Center.
- La France Airship Giclee Print*. (n.d.). Retrieved from [www.art.com](http://www.art.com): <http://www.art.com/products/p4122042215-sa-i4751347/la-france-airship.htm>
- Linear Technology. (2011). *LTC6802-1 Multicell Battery Stack Monitor*. Retrieved from Linear Technology: <http://www.linear.com/product/LTC6802-1>

- Martin, F. (2011, June 20). *Press Releases - Siemens Global Website*. Retrieved 2011, from [http://www.siemens.com/press/en/pressrelease/?press=/en/pressrelease/2011/corporate\\_communication/axx20110666.htm](http://www.siemens.com/press/en/pressrelease/?press=/en/pressrelease/2011/corporate_communication/axx20110666.htm)
- Microchip Technology Inc. (2002). *Brushless DC Motor Control Made Easy*.
- MT-Propeller. (2008, October 8). *Operation and Installation Manual*.
- National Instruments Corporation. (2010). *What is LabView?* Retrieved from LabView: <http://www.ni.com/labview/whatis/>
- Permanent Magnet Synchronous Motors (PMSM) - Motor Drive and Control Solutions*. (n.d.). Retrieved from Texas Instruments: [http://www.ti.com/ww/en/motor\\_drive\\_and\\_control\\_solutions/motor\\_control\\_type\\_permanent\\_magnet\\_PMSM.htm](http://www.ti.com/ww/en/motor_drive_and_control_solutions/motor_control_type_permanent_magnet_PMSM.htm)
- Robert Bosch GmbH. (1991). *CAN Specification Version 2.0*. Stuttgart: Bosch.
- Rotax 912ULS DCDI*. (n.d.). Retrieved from Rotax Service: <http://www.rotaxservice.com/documents/912Sperf.pdf>
- Rotax 912ULS Specifications*. (n.d.). Retrieved from AeroPropulsion Technologies, Rotax Service: [http://www.rotaxservice.com/rotax\\_engines/rotax\\_912ULSs.htm](http://www.rotaxservice.com/rotax_engines/rotax_912ULSs.htm)
- Texas Instruments. (n.d.). *TI DSP Controllers*. Retrieved from Texas Instruments Digital Signal Processing Controllers: <http://www.ti.com/lit/ds/sprs145l/sprs145l.pdf>
- TK Engineering Oy. (2009). *CANopen vs. J1939*.
- Upton, D. S. (2006). *Towards Electric Aircraft: Progress Under the NASA URETI for Aero propulsion and Power Technology*. *SAE Technical Paper Series*. New Orleans: SAE International.
- Vector Informatik GmbH. (2011). *Local Interconnect Network*. Retrieved from [www.vector.com](http://www.vector.com): [http://www.vector.com/vi\\_local\\_interconnect\\_network\\_en.html?gclid=CITJsKisv6wCFYFT7AodTUcGsQ&et\\_rp=1](http://www.vector.com/vi_local_interconnect_network_en.html?gclid=CITJsKisv6wCFYFT7AodTUcGsQ&et_rp=1)
- Zambada, J. (n.d.). *Sensorless Field Oriented Control of PMSM Motors*. Retrieved from *Sensorless Field Oriented Control of PMSM Motors*: <http://ww1.microchip.com/downloads/en/AppNotes/01078A.pdf>