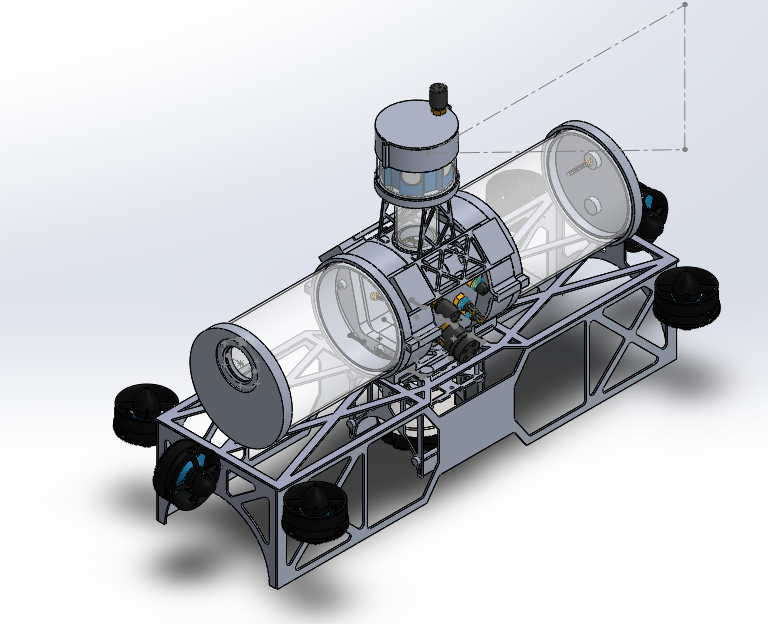
User Manual

Ocean’s 7



*Hassan Alahmed, Abby Caballero, Kyle Harlow,*

*Daniel Henderson, Yuvin Kokuhennadige, Cassandra Noice*

**Table of Contents**

**Problems 3**

**User Base 4**

**System Functionality 5**

**Power 5**

**Controls System 5**

**Backplane 5**

**System Usage 6**

**Limitations 7**

# Problems

In previous years the University of Colorado RoboSub has had a variety of electronics related issues. Two different vehicles had runs ended by devices being plugged in with improper polarity, and the team has had limited controls power. Usually the control algorithms were simple PID loops run on the main computer (CPU). There have also been problems with wire management.

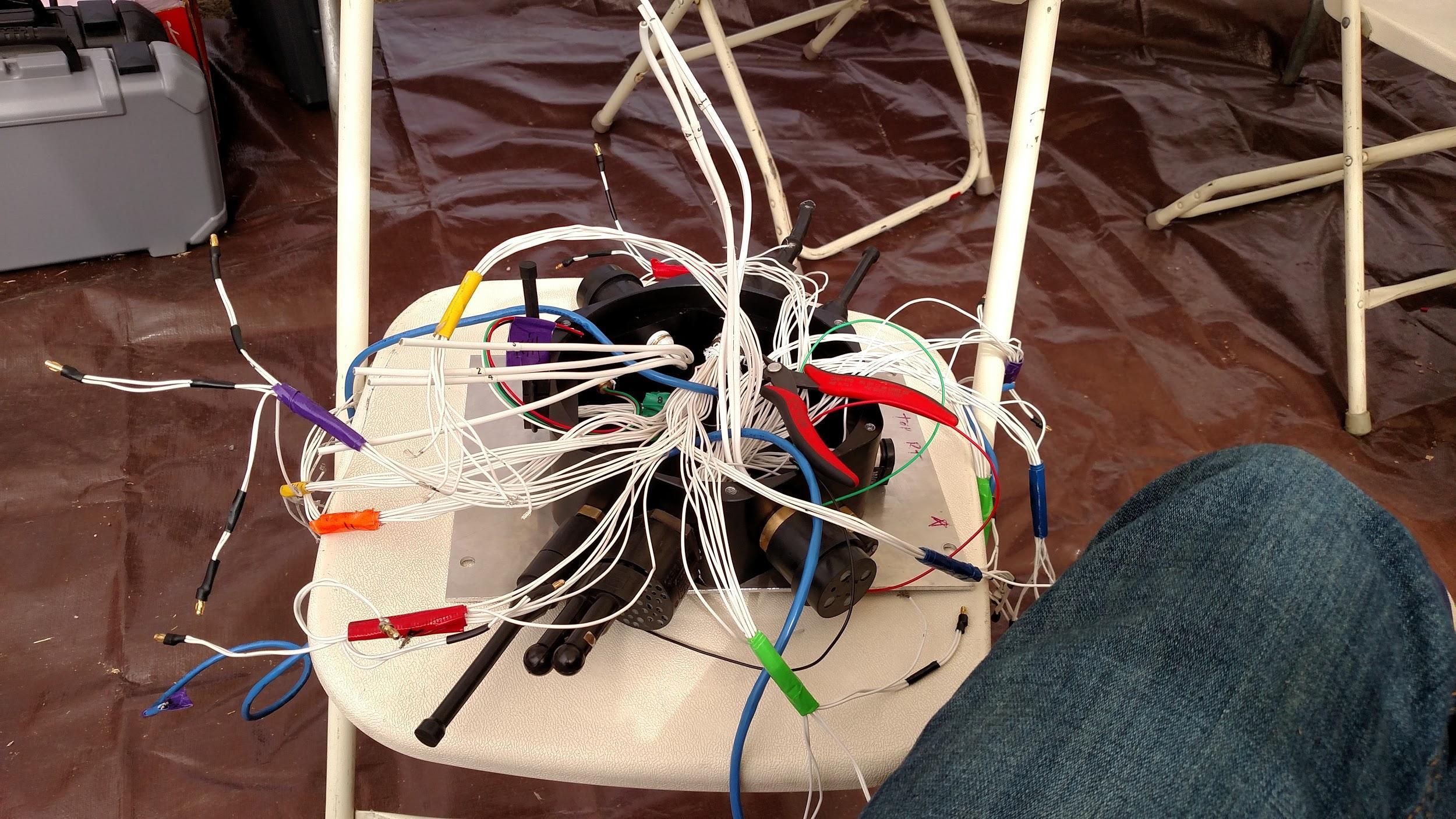


Figure 1.1: Doctor Octopus, the external to internal electrical connections required by Caligula the 2016 Vehicle

Finally, power distribution across the large variety of voltages and currents required by the vehicle has caused problems as well. Large motor currents have caused excessive heating inside the vehicle, and data transfer disruptions.

In order to solve this problem, Ocean’s 7 is designing the following three integrated systems:

1. A power system
2. A controls system
3. A backplane

The power system seeks to efficiently merge and convert power delivered from our two 14.8V, 10C, 10Amp-Hour batteries. The controls system will execute controls algorithms to increase the stability of the vehicle while offloading that information from the main CPU. The backplane will distribute power throughout the vehicle with slots for our own conversion, merge, and controls boards, as well as additional boards possibly created by the RoboSub team.

# User Base

There are three distinct users for our product. They will be delineated the RoboSub Team (RoboSub), the Autonomous Underwater Vehicle (AUV), and the divers. RoboSub is the team who has designed vehicles in the last three years. The team is primarily composed of engineers, many of whom will be able to determine functionality from schematics and algorithms. They will also be the group integrating the system into the AUV. The AUV is the RoboSub team's vehicle. The vehicle functions autonomously, and will be using our system to distribute power throughout the vehicle. The AUV will communicate primarily with the CPU, which will be sending updated vectors for the control algorithm to head towards. The AUV will receive data from a variety of sensors, including the hydrophones array from HydroDynamics. Finally, the divers will interact with our system via the kill switch. Divers are the primary link between the RoboSub team and the AUV during competition runs and practice. They will press the kill switch to kill motors if a dangerous situation, either to personnel or the AUV arises, or to restart a run.

# System Functionality

## Power

The power system will be split into two major components. The first will be a merge circuit. The merge circuit will draw 100A from each battery at 14.8V and will merge the current into a single 200A trace. From there the trace will be distributed to different motors as well as a 12A trace running to the power conversion board. The power conversion board is the second major component of the power system. The power conversion board will take the 14.8V trace and convert it into 5V-1A, 12V-1A, 19V-4A, and 48V-1.5A traces to power the internal electronics of the vehicle.

## Controls System

The control system will calculate motor outputs to maneuver to the desired vector delivered by the CPU. In addition, the CPU will provide data processed from two inertial measurement units (IMU)s and one Doppler Velocity Logger (DVL). From there it will control the ESC’s to maneuver properly.

## Backplane

The backplane will take a 12A, 14.8V trace from the merge board and pass that to the power conversion board. From there it will take outputs from the power conversion board delivering traces of 5V, and 12V at 1A, and 19V at 4A, and 48V at 1A and 0.5A. This power will be delivered to a variety of internal components. It will also deliver PWM signals from the control board to the ESCs. Finally it will deliver a 5V trace to the control board.

# System Usage

The RoboSub team will conduct the final integration of the system. They will be responsible for plugging the boards into the main hull of the AUV. Each board will be plugged into a labeled slot. Additional similar slots will be designed to fit multiple types of boards. Each board will also only be designed to fit in a single direction.

The manual kill switch will visible on the outside of the vehicle. The divers, in order to stop a run in a safe manner, will use the kill switch. It will shut down all motors upon activation.

The AUV will use the controls board by providing desired heading and speed vectors to the board. From there the control board will adjust the speeds of each motor in order to execute that desired heading and speed.

## Controls Software

The controls board features the STM32F767ZI ARM M7 microcontroller, as well as an FT232 UART/USB device. The software provides an interface for communicating with any PC via a serial port or terminal. This can be achieved via a ROS node, python script, serial terminal emulator, or command line. The means of communication is variable length ASCII packets, terminated in semicolons (the microcontroller makes use of the *character match flag interrupt* when it receives the character “;”, 0x3B). Instructions for interfacing with the microcontroller are shown below, and the microcontroller code is commented to serve as a manual for any changes required.

### Serial Communication Settings

|  |  |
| --- | --- |
| **Baud Rate** | 57600 |
| **Data Bits** | 8 |
| **Stop Bits** | 1 |
| **Parity** | None |

### 

### 

### Packet Types

Serial communication between the PC and the controls board is accomplished using variable length ASCII strings. All packets begin with a single header character, followed by a message body, and terminated with a semicolon, “;”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Header** | **Body** | **Example** |
| **Update**  *Sends all values needed for controls algorithms* | “u” | comma separated floating-point values | “u10.0,3.14e-2,-6;” |
| **Control** | “c” | word from set of control strings | “cmotor-on;” |

Note // update packets should always send all 18 values in the following order:

,,,,,,,,,,,,,,,,,

where “” denotes a first derivative and “~” denotes a desired value. If any of those values are unavailable, zeros should be sent instead.

**Control Strings Set:** {“motor-on”, “motor-off”, “reset”}

### Response

Upon receipt of a packet, the microcontroller will reply with an error message of length 3 bytes, including a newline character.

|  |  |
| --- | --- |
| **Error Code** |  |
| “en” | Error-None: packet was parsed/command was executed successfully |
| “eh” | Error-Header: header character not recognized |
| “eb” | Error-Body: message body is not recognized as a command |
| “ev” | Error-Values: in an update packet, values are malformed and cannot be interpreted as floating point numbers |
| “ec” | Error-Count: an incorrect number of values were received in an update packet |
| “eo” | Error-Other: some undefined error occurred |

# Limitations

The controls system is limited by the motor layout and output. Four T100 motors are aligned vertically and used to control pitch, roll, and depth. Two Video Ray Pro 4 motors are aligned down the length of the robot and centered with the center of mass, and will be used to control velocity. Two T100 motors are aligned horizontally to the vehicle and will be used to control strafing. The vehicle has a defined depth limit based on the mechanical hardware. The seals are rated to 100 ft. depth. In addition our power system will not be able to deliver maximum power to the AUV for an extended duration of time. The vehicle has a total of 20amp hours between the two batteries. Exceeding the maximum current draw of 200 amps between the two batteries could cause them to overheat and fail.

# Change Log

* Added Table of Contents
* Updated System functionality
  + Change power specifications to match CDR, adjusting current and voltage levels
  + Changed Backplane specifications to match CDR, dropping communication, and updating trace values
* Updated Limitations to include more concrete examples