Proof of Concept Test Demo Sheet

**TEAM NAME: Oceans 7 Test Date & Time:**

**Test # and Name (from PoC plan) 2B: Communications stress Test Test Type: Feasability**

1. **Purpose of Test and How it Relates to Project (Brief, concise, but complete, description):**

Critical Functionality Demonstrated: UART and I2C communication and PWM signals will function in an environment with the high levels of EM noise that will be generated by the eight motors drawing upwards of 200A.

Relation to Overall System: Communications testing will ensure that the various submodules of the CU RoboSub are able to reliably communicate with each other. If digital communications are untenable in the noisy environment that is expected then alternate designs will have to be pursued for communications and or power systems.

1. **Test Setup, Pre-conditions, and Procedure (Brief, concise, but complete, description):**

Set Up and Preconditions

- Voltage source set to 5V with a max current draw of 1A

- Voltage Source set to 14.8V with 5A

- Two microcontroller dev boards (the STM32 Nucleo-144 and an Arduino Uno, in place of the sensors that use I2C comms) with firmware loaded to support sending and receiving data via I2C and UART communication protocols at 9600 Baud

- A desktop computer with a serial terminal to communicate with the microcontroller

- Electromagnet, or power drill to simulate electromagnetic noise

- Backplane test board

- BlueRobotics ESC 30A (electronic speed controller)

- Blue Robotics T100 Thruster

Note: The *control set* of strings is defined as {‘UART’, ‘I2C’, ‘motor-on’, ‘motor-off’}. These are used throughout the test.

Testing Procedures:

1. connect the 5V and ground lines of the power supply to the 5V and ground of the back plane
2. Connect the 5V and and ground pins of the Nucleo board to the 5V and Ground of the back plane

C. Connect the TX pin of the Nucleo to the RX pin on the backplane test board

D. Connect the I2C SCL pin of the Nucleo to the I2C SCL pin on the backplane test board

E. Connect the 5V and ground lines on the opposite end of the backplane test board to the power and ground pins of the Arduino.

F. Connect the TX pin of the Arduino to the RX pin on the backplane test board

G. Connect the I2C SCL pin of the Arduino to the I2C SCL pin on the backplane test board

H. Connect the voltage source’s high and low voltage lines to the backplane test board input and ground pins.

I. Connect the Nucleo to the desktop computer via USB

J. Turn the voltage supply on

K. Send the string ‘I2C’ through the serial terminal on the desktop computer. This sets the microcontroller to use the I2C communication protocol

L. Send strings (not in the control set) to the Nucleo using the terminal

M. Record the information returned to the terminal

N. Send the string ‘UART’ through the terminal on the desktop computer. This sets the microcontroller to use the UART communication protocol

O. Send “Good Night Moon” one hundred times

P. Record the information returned to the UART terminal

Q. Connect the 14.8V line from the voltage supply to the 14.8V pin on the backplane

Q. Connect the ESC PWM header to the male ESC header on the backplane

R. Using jumper wires, connect pin PX.XX of the Nucleo to the PWM pin of the female ESC header on the backplane, and connect the other 2 pins on the ESC header to 5V and Ground (they are labeled)

S. Connect the power and ground cables of the ESC to the connectors labeled as ESC power and ESC ground

T. Plug the 3-phase cables into the motor

U. Send the string ‘motor-on’ to turn on motor, then place it near the backplane test board (note: ‘motor-off’ will turn off the motor)

V. Repeat steps K through P recording the data separately

W. Turn off the voltage source, disconnect the microcontrollers, and remove the power systems test board from the backplane test board

X. Compare the sample data with the recorded data

1. **Success Criteria (feasibility) / Selection Criteria (alternatives) / Design Info (gathering): (Concise, complete, quantitative):**

Fidelity of data sent via I2C is 99% or higher

Fidelity of data sent via UART is 99% or higher

Fidelity of data sent via I2C with the motor running is >95%

Fidelity of data sent via UART with the motor running is >95%

1. **Instructional Team Notes:**
2. **Test outcome and what was learned (toward finalizing design):**