FSVPAD Electronics

Shorting plug: The shorting plug provides power to the Freewave radio and Garmin GPS unit contained within the electronics housing. The construction of the shorting plug, which uses and 8-pin SubConn in-line micro-circular (MCIL) connector, is described in Section II. A wiring diagram of the shorting plug is shown in Figure 1. The pins and wires colors in the figure correspond to the MCIL8F connector and the corresponding bulkhead connector used on the electronics housing cap. A bi-color LED wired potted within the shorting plug blinks red or green in different patterns based on the status of the Arduino microcontroller.

To construct the shorting plug the pigtail of an MCIL connector is trimmed to approximately one-inch. Another half inch of the jacket is removed to reveal the wires. The black and red wires are soldered. A bi-color LED is soldered to the remaining orange and green connections. Here a positive voltage applied to the orange lead results in a green LED. Once soldered use a multimeter and the MICL ping to confirm continuity between the respective pins. Next, a ½" diameter piece of shrink tube is placed around the soldered components such that only a couple of millimeters of the LED are exposed above the top of the tubing. The bottom of the shrink tube where it connects to the bulkhead connector is sealed using a heat gun. Before proceeding, install the lock ring assembly. Scotchcast epoxy is then poured to fill the interior of the shrink tube. Allow the epoxy to set for a few hours or more. The top of the LED can then be ground down using a bench grinder until it is level with the top of the epoxy/heat shrink.

The full circuit diagram for the FSVPAD electronics is shown in Figure 2.

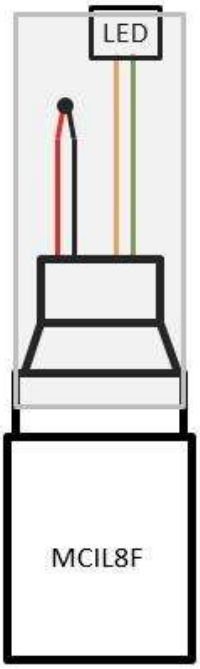

Figure 1: A circuit diagram of the shorting plug, which serves as the power switch for electronics in the FSVPAD housing.

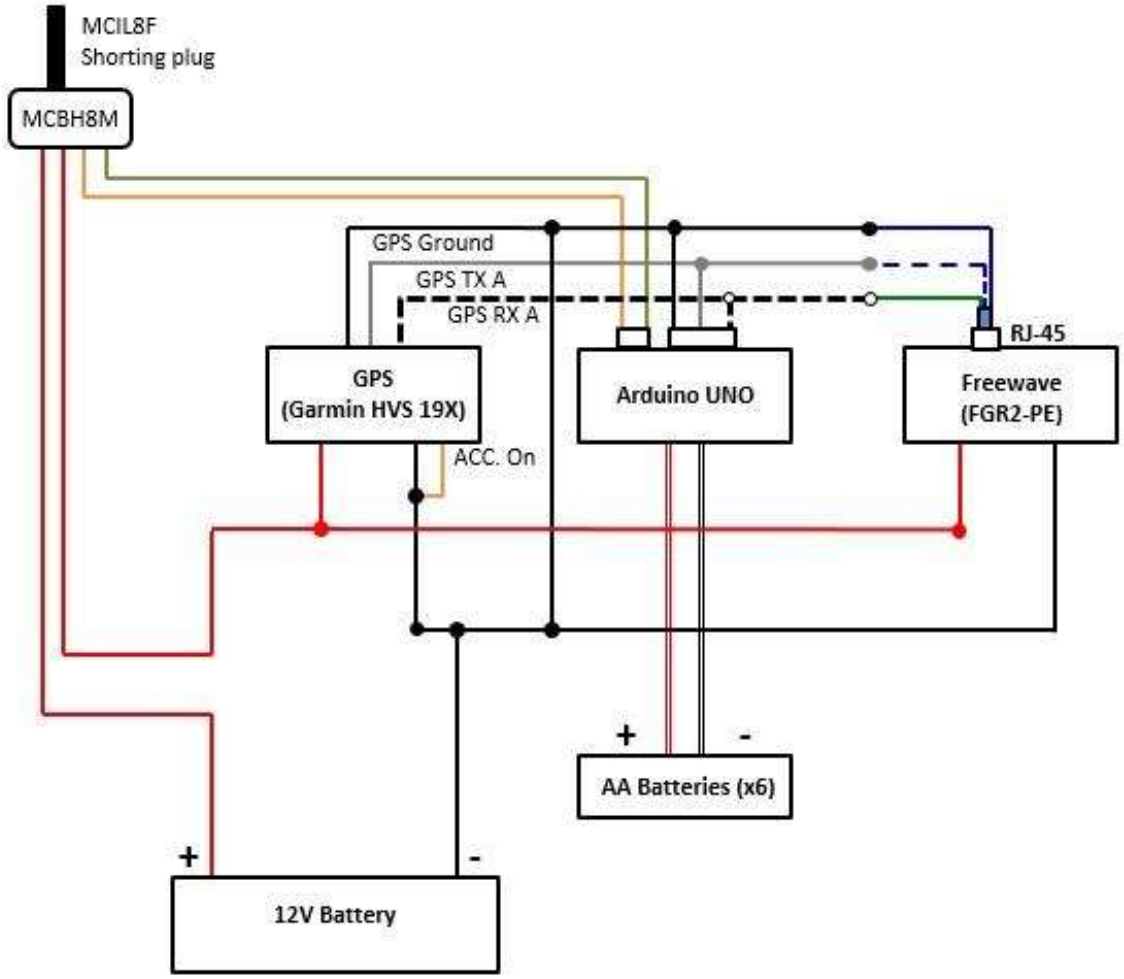

Figure 2: Diagram of drifter electronics. Note that the Black, dashed GPS RX A line is a white wire and the gray GPS TX A wire from the Garmin unit connects to the blue and white wire within the Ethernet cable. The ACC On wire included a resistor that changes the default sampling rate of the unit from 10 Hz to 1 Hz. Circuit information specific to the individual components are included below.

A.I Arduino Uno

Real-time GPS data acquisition and data logging in addition to the control of the bulkhead switch LED is accomplished using an Arduino Uno microcontroller with the Adafruit SD Logger board driven by a custom script. The bi-color (red/green) LED is controlled using two I/O pins on the Arduino board and changes in color are achieved by reversing the

polarity to the I/O pins. The current hardware receives GPS data strings on digital pins on the Arduino board without native serial support. Therefore, a MAX232 integrated circuit has been added to allow for serial communication through other digital pins. The wiring diagram for the MAX232, complete with its connections to the GPS unit, are shown in Figure 3 and described in the caption. GPS strings output by the Garmin GPS unit are spliced such that both the Freewave radio and the micro-controller receive the same NMEA strings.

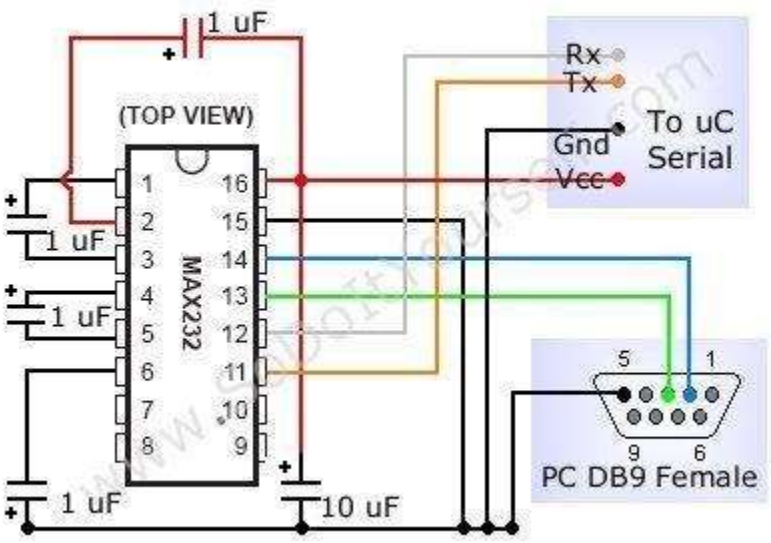

Figure 3: The MAX232 integrated circuit with wiring diagram for software serial communications. For the drifter the DB9 connector has been removed; wires associated with Pin 2 (TX GPS), Pin 3 (RX GPS), and Pin 5 (ground) are replaced with the default write colors for the Garmin GPS unit. These wire colors are gray (RX; DB9 Pin 2), white (TX; DB9 Pin 3), and black (ground; DB9 pin 5). Pins 11 and 12 from the MAX232 are connected to the pins programmed for software serial communications on the Arduino board (Pins 2 and 3). The power and ground pins for the circuit go to the ground and 5V pings on the Arduino board.