User Manual: Blue Robotics Pressure Temperature Sensor

Spencer Marquardt, Maggie Kosalek

March 25, 2022

Contents

1 Overview

This manual is written as a guide to be used with the Blue Robotics Pressure Temperature instruments (BRPTs). It is organized to show the design and assembly of the BRPTs, then to walk through the steps for preparing the instruments to be deployed in the field. Included are detailed steps for the fabrication of the electronics, step by step instructions for setting up the data logger, battery holder, and pressure housing. There are also details on field deployment methods and the related equipment that is necessary to successfully collect data.

2 Description of the BRPT

The BRPT instruments are versatile and can be deployed in a range of nearshore coastal environments to accurately measure wave pressure and water temperature. The BRPT instruments house the sensing package, batteries, and data logger with RTC. The BRPT is intended for medium length applications (1-15 days) in nearshore applications, water depths of up to 10 m. The BRPTs have been designed as cheap alternatives to much

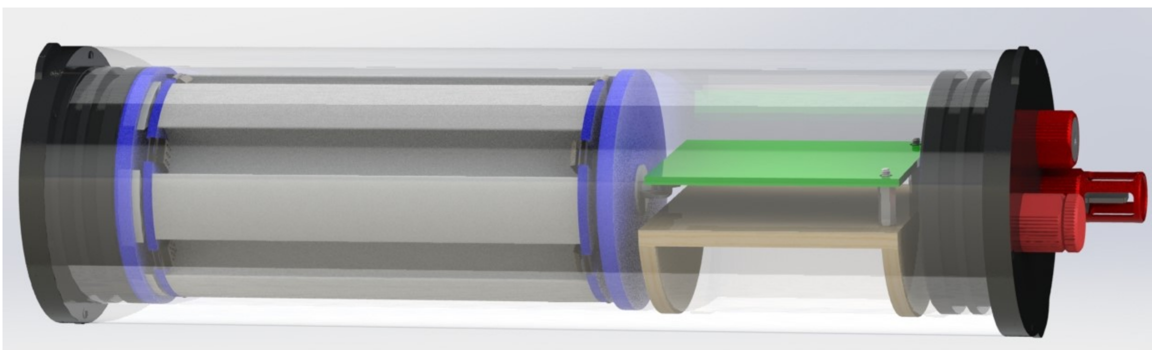


Figure 1: Rendering of the BRPT

more expensive ocean sensing instruments. The name of the instruments is derived from the supplier of most the main components, the California based company, Blue Robotics. The parts were chosen based on their availability, cost, and ease of implementation into a complete sensing package without requiring major modifications. The BRPT is programmed using the Arduino IDE software and the data logger is interfaced using a terminal program such as puttytel or TeraTerm. Sampling intervals can be set programatically by modifying the code in Appendix

3 Specifications

	Bar02 Pressure	Bar02 Temperature	TSYS01 Temperture
Measurement Range	Standard: 300 - 1200 mbar Extended: 1 - 2000 mbar	2-40 °C	-40-125 °C
Accuracy	±2 mbar	±2 °C (from -20-85 °C)	±0.1°C (from -5-50 °C)
Resolution	0.16 mbar	0.1 °C	0.01 °C
Operating depth	0-10 m	0-10 m	0-975 m
Supply voltage	2.5 - 5.5 V	2.5 - 5.5 V	3.3 - 5 V
I2C Logic Voltage	2.5-3.6 V	2.5-3.6 V	3.3 V
Peak Current	1.25 mA	1.25 mA	1.4 mA
Internal Battery pack	Up to 12 rechargable Li-ion batteries		
Memory	SLERJ SSR-LC single channel data logger - 115200 baud		
Storage	8 GB micro SD card		
Sampling Speed	Programable up to 16 Hz		
Housing Detph Rating	150 m		

4 Materials

There are four main categories of materials for the BRPT: the electrical components, batteries and battery holder, pressure housing, and field deployment equipment. Since field deployment sites will vary widely depending on research goals the field equipment list is primarily a suggestion. The list of deployment equipment is best suited for shallow, sandy, coastal areas.

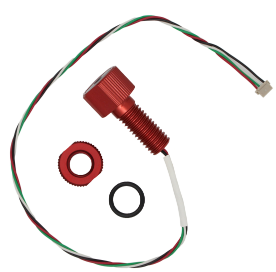
4.1 Electronics

The electrical components in the BRPT consist of primarily off the shelf components that are readily available. The pressure and temperature sensors are both manufactured by TE Connectivity and sold by Blue Robotics, Figure 2. The microcontroller is a Teensy 4.0 from PJRC, Figure 3. Data is stored using the SLERJ SSR-LC single channel data logger, Figure 4. Power is delivered to the board with a 5 V linear voltage regulator. All of the components are integrated using a custom designed PCB, Figure 5. The full parts list is below, where the supplier is noted in parenthesis.

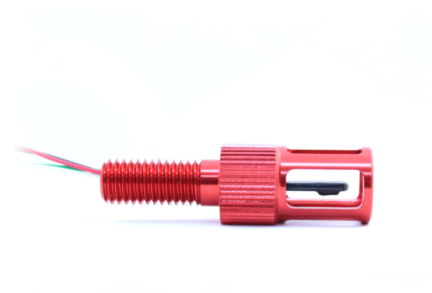
The circuit board is secured onto an electronics shelf. The shelf is comprised of two semicircular disks glued to the flat shelf which creates a platform inside the BRPT's acrylic housing that minimizes the amount of movement of the PCB and components. To accomplish this, a 3" hole saw was used to cut disks from the pallet. The disks were cut in half along their width to make two thinner disks. Similarly, an approximately 6 cm rectangular piece was cut in half to make the shelf. The PCB is attached to the shelf by using $\frac{1}{2}$" hex standoffs with 4-40 threading. The standoffs are attached to the shelf using 4-40 flat head machine screws.

Electrical Components

- 1 x Bar02 Ultra-High Resolution Pressure Temperature Sensor (Blue Robotics)
- 1 x Celsius Temperature Sensor (Blue Robotics)
- 1 x Teensy 4.0 micro controller (PJRC)
- 1 x SSR-LC single channel data logger (SLERJ)
- 1 x CR2032 button battery breakout board (Adafruit)
- 1 x LM1117 5 V fixed linear voltage regulator (Texas Instruments)
- 1 x Custom printed circuit board (JLC PCB)
- 1 x 2 position switch (Adafruit)
- 2 x 10k ohm resistors (Mouser)
- 1 x Jumper wire (Amazon)
- 2 x 4 pin male and female JST connectors (Digikey)
- 3 x 2 pin male JST connector (Digikey)
- 2 x 2 pin female JST connector (Digikey)



(a) Bar02 pressure, temperature sensor



(b) Celsius Temperature sensor

Figure 2: In (a) the MS5837-02BA sensor from TE connectivity is housed in an aluminium M10 penatrator bolt assembly with an O-ring. The Bar02 sensors have an effective measurement range between 1-2000 mbar, with thier highest accuracy in the range of 300-1200 mbar. The Bar02s are effective in up to 10 m depths with an accuracy of ±2 mbar. In (b) the TSYS01 from TE connectivity is housed in a similar M10 penetrator bolt with an aluminium cage to protect the delicate thermistor. The TSYS01 has an accuracy of ±0.1 °C in the range of -40-125 °C. The operating depth of the penetrator bolts is up to 975 m

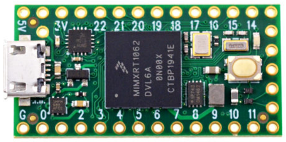


Figure 3: Teensy 4.0 microcontroller. ARM cortex-M7 CPU at 600 Mhz, 1024k RAM, 40 digital IO pins, 3 I2C pins. Programable with Arduino IDE with Teensyduino add-on.

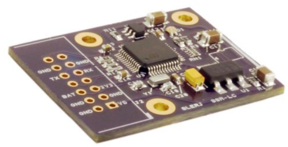


Figure 4: SLERJ SSR-LC data logger. Records and asychronous serial channel at 115200 baud onto a removable microSD card. Communication is over a 5V TTL serial connection. Includes an RTC and connections for a battery to keep time between power cycles

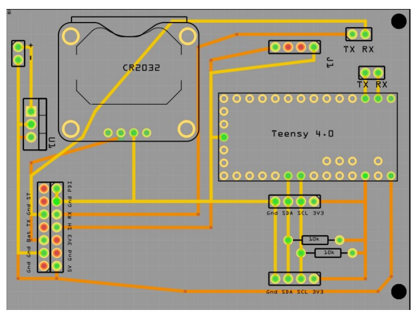


Figure 5: Custom printied circuit board. Integrates all of the electrical components in a small, robust footprint

4.2 Batteries and Battery Holder

The batteries and battery holder make up more than 60% of the internal volume of the BRPT instrument. The battery holder is 3D printed and is designed to hold 12 rechargeable 18650 Li-Ion batteries in a 2S6P configuration, Figure 6. That is two battery cells are in series and 6 cells in parallel. Batteries in series have their voltage added together, while batteries in parallel have their capacities added. The 2S6P configuration doubles the 3.7V output of the batteries and the capacity is multiplied by six. The battery holder assembly includes a battery spacer that can be used if the batteries used are too long for the original battery holder. The battery end caps and spaces have been designed with mounting pegs that help align the battery contacts with the battery channels on the battery holder, Figure 13.

Battery Holder Components

- 1 x 3D printed battery holder
- 1 x 3D printed battery spacer, if required
- 2 x 3D printed battery holder end caps
- 12 x 209 Battery contacts (Digikey)
- 12 x 2600 mAh 18650 rechargeable Li-Ion batteries (Digikey)
- 1 x 7" 8-32 threaded rod (McMaster-Carr)
- 2 x 8-32 flanged wingnuts (McMaser-Carr)
- 1 x 2 pin female JST connector (Digikey)
- 18 gauge Striveday^{TM} stranded wire (Amazon)

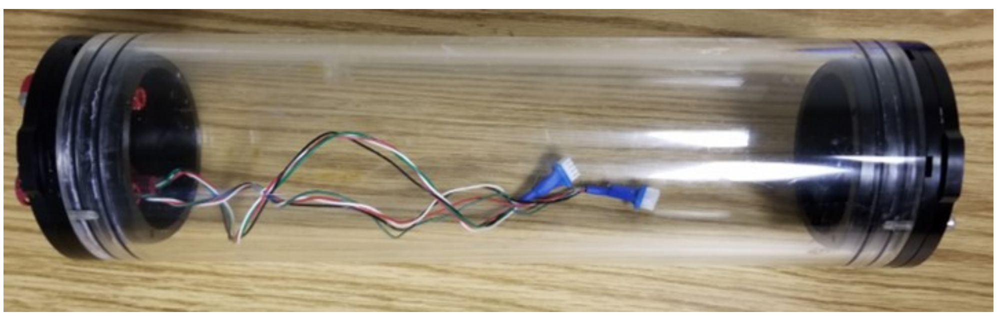
Figure 6: 3D prited battery holder

4.3 Pressure Housing

The pressure housing is also sourced from Blue Robotics. They have several options for pressure housing sizes, and material. The BRPT utilizes the 3" series components with the 11" long acrylic tube acting as the main body. The endcaps and O-ring flanges are machined aluminum parts. The endcaps include three O-rings each, with a face seal between the endcap and O-ring flange and two piston seals between the O-ring flange and the acrylic tube. Two modifications to the parts from Blue Robotics are required for the BRPT. First, one endcap needs to have three 10 mm thru holes cut for the sensors and a vent assembly. The vent is used to prevent an air lock when the endcaps are installed and for depressurizing the housing before opening. The second modification is for securing the endcap assembly to the acrylic housing. Two holes are drilled into the acrylic that align with two opposing holes on the endcap and 4-40 helical inserts are secured in the holes. These holes align with two opposing holes on the endcaps, where in the original design the M2x12 screws are replaced with longer 3/4" 4-40 socket head screws.

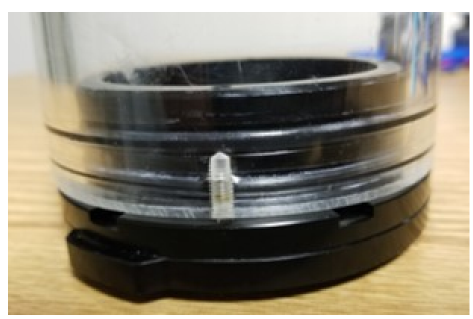
Pressure Housing Components

- 1 x 298 mm long 3" series acrylic tube (Blue Robotics)
- 2 x aluminum O-ring flange (Blue Robotics)
- 2 x aluminum or acrylic endcap (Blue Robotics)
- 1 x vent assembly (Blue Robotics)
- 4 x dash number 230 NBR O-ring
- 2 x dash number 148 NBR O-ring
- 8 x M3x12 screws
- 4 x 4-40 3/4" socket head screws (McMaster-Carr)
- 4 x 4-40 helical insert (McMaster-Carr)



(a) Pressure housing

(b) Top endcap



(c) O-ring flange

Figure 7: Pressure housing assembly. In (a) the housing is shown with both endcaps installed. In (b) the sensors and vent are shown. (c) is a detailed view of the tapped hole with helical insert.

4.4 Deployment Equipment

The equipment discussed in this section is intended to be used for field deployments of the BRPT in shallow, sandy environments. A ground anchor, commonly used for securing large tents can be used to secure the BRPT in place. The BRPT is held to the screw anchor using two or more hose clamps. Silicone sheeting can be used to protect the acrylic. Care should be taken not to over tighten the hose clamps, as too much force could result in deformation and/or cracking of the acrylic tube. If the pressure housing is deformed it could result in leaks that cause damage to the electronics.

Deployment Equipment

- 48" ground anchor
- silicone rubber sheeting
- marker buoy
- hose clamps

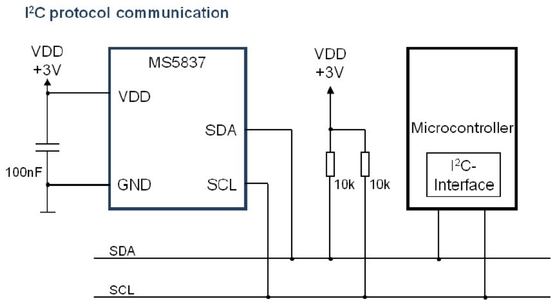
Figure 8: Deployment equipment for shallow water sandy environments

5 BRPT Design and Assembly

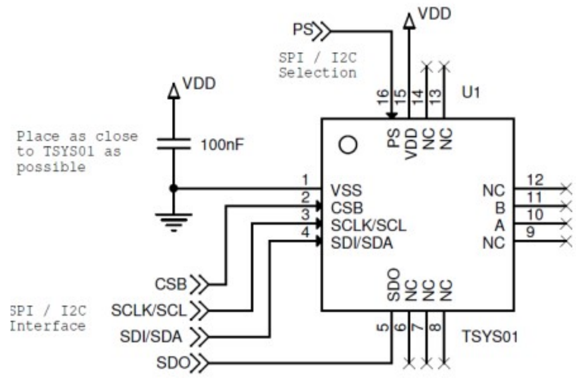
This section will provide details about the design elements of the BRPT and includes instructions for assembling the components discussed in Section 4. Details of the assembly of the electrical components are covered in Section 5.1. The battery holder is covered in Section 5.2.

5.1 Circuit Assembly

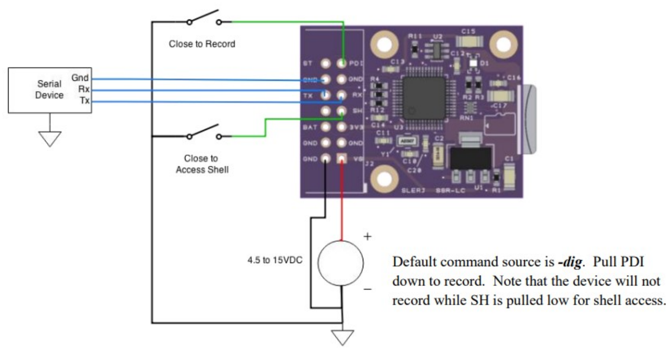
All of the electrical components listed in Section 4.1 are soldered to a custom printed circuit board (PCB). The PCB was designed using Fritzing, a circuit design software, the result is shown in Figure 5. The circuit diagrams shown in Figure 9 were all integrated into the PCB design. The input output diagram for the Teensy 4.0 is shown Figure 10, it shows the pins that are used in the BRPT circuit.



(a) Circuit diagram for MS5837-02BA sensor



(b) Circuit diagram for TSYS01 sensor



(c) Circuit diagram for SSR-LC data logger

Figure 9: circuit diagrams for the components on the PCB

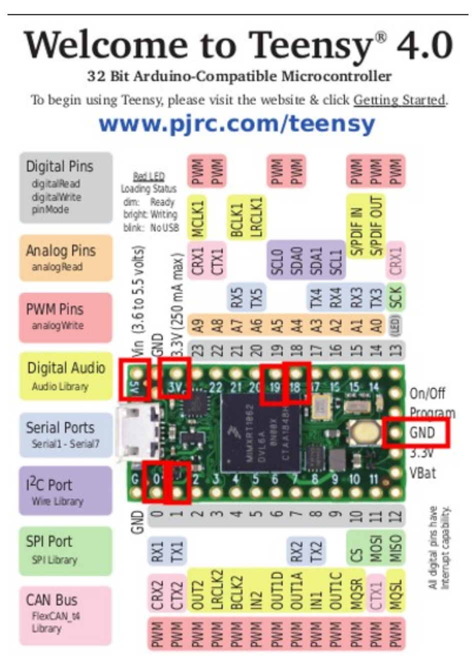
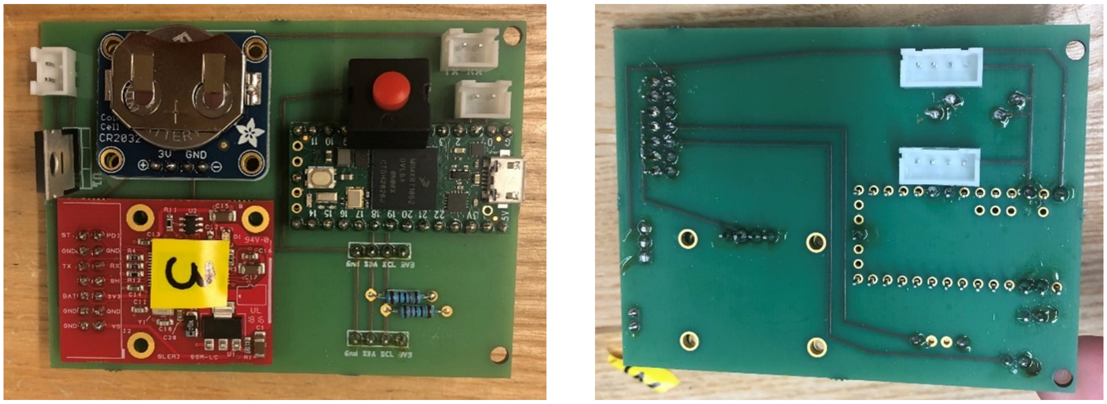


Figure 10: Input output guide for Teensy 4.0 microcontroller. The pins used in the BRPT circuit are outlined in the red boxes



(a) top (b) bottom

Figure 11: Assembled circuit on PCB. The power connector is on the top left side of the board and the TX/RX jumper connections are on the top right side. The shell access button is in the top middle. The RTC battery and SSR-LC data logger are on the left, and the Teensy 4.0 is on the right, with the two pullup resistors on the bottom right.

All components are soldered on to the top side of the PCB as shown in Figure 11a except for the two four pin male JST connectors, which are soldered on the bottom side of the PCB (Figure 11b) .

The jumper wires are included to provide access to the SLERJ SSR-LC shell interface, described in Section 6. The jumper consists of two wires with the each end terminated with a two pin female JST connector. Note: the jumper connects the TX and RX channels between the SSR-LC Teensy where TX goes to RX in each direction.

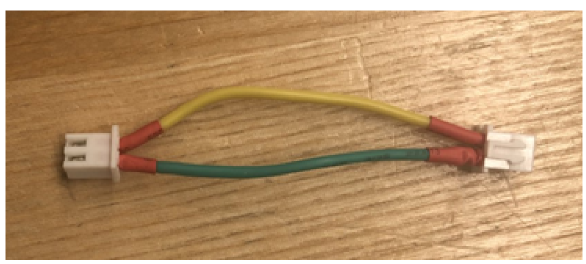


Figure 12: Jumper wire connects the TX and RX channels between the SSR-LC datalogger and the Teensy. TX connects to RX and RX connects to TX.

5.2 Battery Holder Assembly

The battery holder assembly consists of the parts listed in Section 4.2. The main body of the battery holder, the two endcaps, and the spacer are 3D printed parts. The endcaps and spacer have a series of pegs and matching holes to align the six battery channels. The endcaps are held on the main body by a 7” threaded rod and two wing nuts. The battery holder has six channels that fit two 18650 Li-Ion rechargeable batteries in series. Not all 18650 batteries are the same length, therefore a spacer was designed to extend the length of the battery holder to accept various brands of battery. Figure 13 shows the whole assembly.

Each endcap has six battery contacts that are soldered together using short sections, $\approx$ 3 cm, of stranded wire. The battery contacts are then slid into the six channels on the endcap. The second endcap is identical except a longer length of wire runs through the

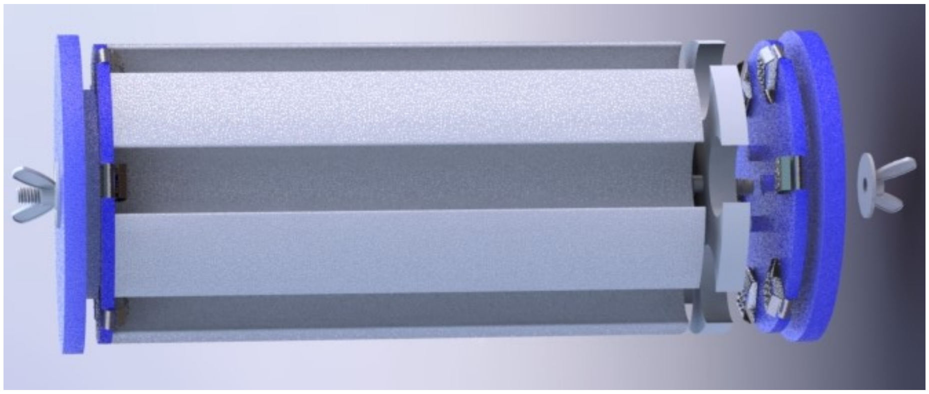
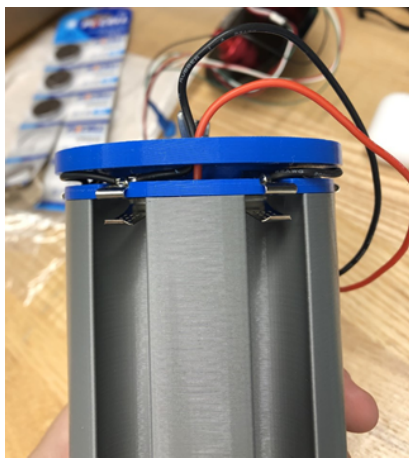


Figure 13: 3D printed battery holder with spacer shown on the left.

length of the battery holder to the top. Both ends have a length of wire that runs out of the top side of the battery holder which terminate in a two pin female JST connector. The connector is attached such that the bottom of the battery holder is the positive end, while the top is the negative end, therefore batteries should be inserted accordingly, see Figure 14.



(a) The top portion of the battery holder. The negative end of the batteries should face the top

(b) The bottom portion of the battery holder. The positve end of the batteries should face the bottom

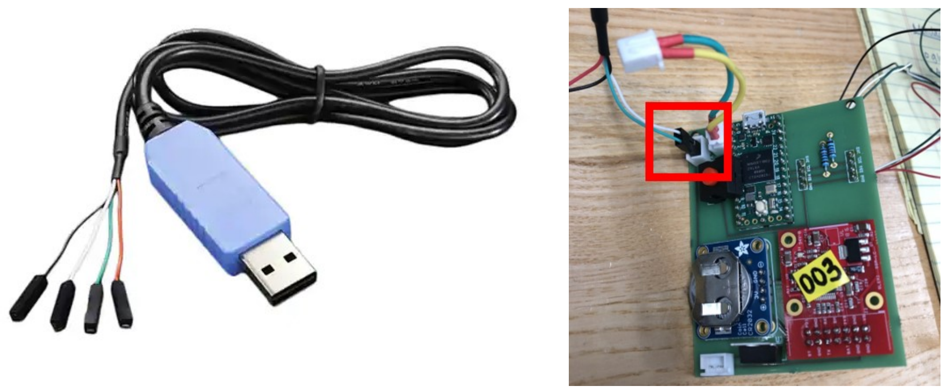
Figure 14: Top and bottom sections of the battery holder.

6 Data Logger Shell Access

The shell interface of the SSR-LC data logger allows the user to set up the logging parameters including the time on the RTC, the format of the output file, and the relative file path of the output file. The SSR-LC data logger uses a 5 V compatible TTL serial connection which requires a USB to TTL serial cable. A terminal program must also be installed on the computer being used for setup. To access the shell interface the following steps are required

Accessing the Shell Interface

1. Disconnect the jumper wire from the outermost 2-pin connector and connect a USB to TTL communication cable to the TX and RX output pins from the PCB as shown in Figure 15



(a) USB to TLL serial cable

(b) PCB connection

Figure 15: The TTL cable is connected to the TX and RX channels on the SSR-LC data logger from the headers on the PCB

2. Connect the USB into a computer with a terminal software such as Puttytel or TeraTerm installed. The following instructions are for Puttytel

3. Open Puttytel and select the options as follows: Serial Line → COMXX, Speed → 115200 baud, Connection type → Serial. Note: the COM port will vary, to find on a windows machine go to the device manager to determine the port number. These options are shown in Figure 16.

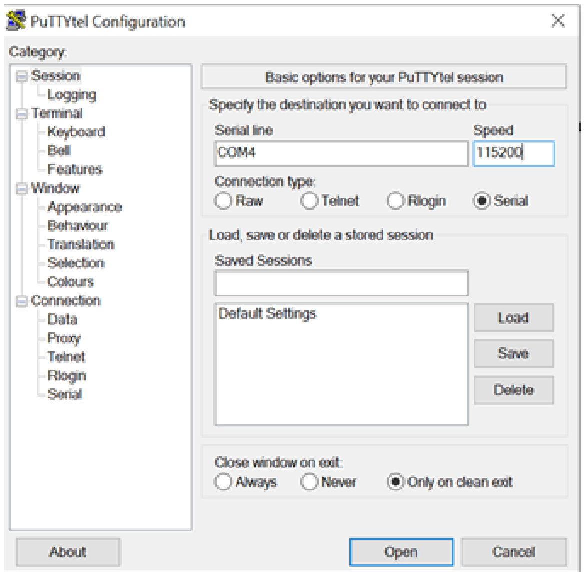


Figure 16: Puttytel console screen with the options for accessing the SSR-LC data logger shell

4. Click Open

5. Push the shell access button on the PCB. This pulls the shell pin to ground on the SSR-LC as seen in Figure 9c. When the shell is active a green LED will blink slowly on the SSR-LC.

6. Connect the power and ground pins on the TTL cable to the PCB as shown in Figure 17. As soon as power is applied text should appear in the Puttytel terminal window as seen in Figure 18.

   Troubleshooting note: If text does not immediately appear, try pushing the shell button again, then unplugging the power and plug it back in. The board can also be powered via the micro USB cable on the Teensy. The shell can only be accessed when power is first applied to the board.

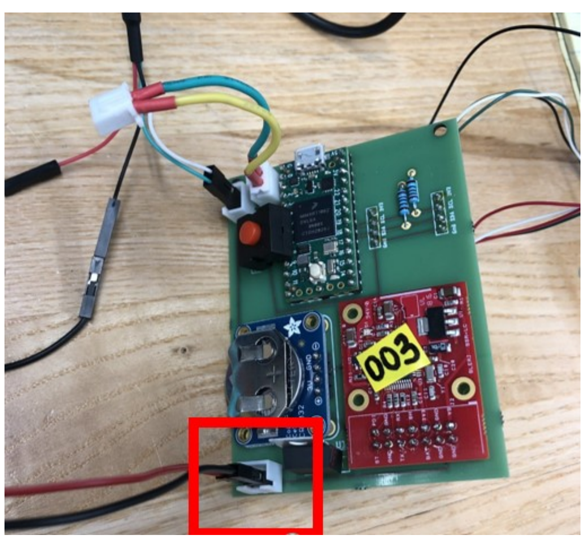


Figure 17: Power and ground pins from TTL cable connected to the PCB

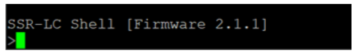


Figure 18: Shell access confirmation message in Puttytel terminal

7. Type “config” and hit enter. This will bring up the crrent configuration settings of the data logger (Figure 19). All commands can be found in the SSR-LC user manual. The following are the commands to configure the data logger.
   - To view the current date type “date” and hit enter
   - To view the current time type “time” and hit enter
   - To set the date, type “date *yyyymmdd*” to input the year, month, and day, then hit enter. The date is formatted as indicated by the itlaicized string.
   - To set the time, type “time *hhmmss*” to input the hour, minute, and second. To start the clock hit enter. The time is in 24 hour format.

     Note: the SSR-LC cannot be automatically set with the computer clock, therefore, when inputting the seconds set it ahead of the current time then hit enter to synchronize with the desired clock source.
   - To set the file type to time-tagged, type “config file type tt” and hit enter.
   - To set the file path, type “config file path *filepath*”. Replace *filepath* with the desired file structure

     e.g. inputting “BRPT3/st[YMDhms]” will save a new file appended with the date and time the file was created in the BRPT3 folder.
   - To save changes to the configuration settings, type “config save”.
8. To confirm the settings on the data logger type “config” again.

Figure 19: Puttytel terminal output showing the SSR-LC configuration status.

9. Disconnect the TTL cable from the computer, turn off shell access with the button on the PCB, and recconnect the jumper wire.

7 Preparing the BRPT for Deployment

This section describes the steps to prepare the BRPT for data collection and serves as a pre-deployment checklist. The setup process requires a micro USB cable, USB to TTL serial cable, Arduino IDE with the Teensyduino add on installed, and a terminal program for accessing the data logger shell interface.

1. Insert an SD card into the SSR-LC data logger
2. Confirm the date and time on the data logger is correct by accessing the shell interface. Confirm the file path and output format are as desired. Follow the steps in Section **??**

3. Connect the micro USB cable to the Teensy and upload the desired code. A sample code is provided in Appendix A.

4. Connect the Bar02 and Celsius sensors using the connections on the bottom side of the the PCB. Figure 11b. The connectors on the sensors are too large to fit through the holes in the endcap, so the pressure housing endcap must be removed for this step.

   Note: If the code in Appendix A is used, when the sensors are connected and intialize within their expected operation range, the built in LED on the Teensy will flash 3 times to indicate successful initialization and begin sending data to the data logger. The built in LED will blink at the sampling rate.

5. Confirm the sensor output in the Arduino IDE serial monitor.

6. Prepare the battery holders

   - Remove one wing nut from the battery holder assembly to remove onr of the terminals.
   - Insert the batteries in the same orientaion, with the negative poles all facing the top end of the battery holder (Figure 14a).
   - Replace the endcap, align the pegs with the holes on the main body, then secure with the wingnut.

     Important: If the batteries are inserted improperly the circuit will be damaged. There is no protection circuitry to prevent a short circuit. Batteries that are inserted backwards will be damaged and could result in fire.

7. Prepare the pressure housing. See Figure 20 for and exploded view of the pressure housing components.

   - Remove and inspect all O-rings for damage. Clean the O-rings and remove any debris from the grooves
   - Re-grease the O-rings and replace them in the grooves
   - Ensure the four M3x12 screws (shorter) are securely threaded into the O-ring flange and the longer 4-40 3/4" screws can be inserted into their respective holes on the acrylic tube.
   - Install the bottom endcap

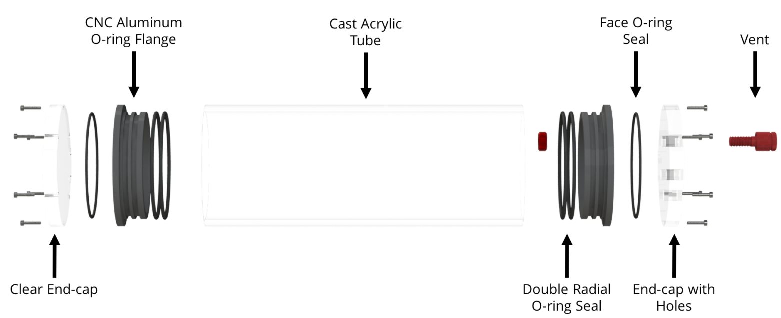


Figure 20: Exploded view of pressure housing parts. Note that the end with the vent cap also has the pressure and temperature sensors attached

8. Slide the battery holder into the pressure housing with the battery connector facing the open end.

9. Connect the battery connector to the PCB. The initialization sequence should begin, confirm that the sensors initialize and the data logger is recording. The LED on the Teensy will blink at the sampling frequency and the data logger will show a solid red LED (confirming the SD card is mounted) and the green light will blink rapidly (indicating the receipt of data)

10. Insert the PCB and shelf assembly into the pressure housing

11. Install the top endcap and secure with the 4-40 3/4” screws

Figure 21: Fully assembled BRPT

Figure 22: BRPT in the field

A Sample Code

The code below is included as a reference. The library for the TSYS01 and MS5837-02BA sensor have both been modified for the purpose of increasing the sensor output resolution. The unmodified version of these libraries is available on giHub via links on the Blue Robotics website.

```
// Include Libraries
#include <Wire.h>
#include <TSYS01.h> // modified TSYS01 library
#include <MS5837.h> // modified MS5837 library

// Define pin for built in LED control
#define LEDpin 13
#define HWSERIAL Serial1 // Turn on hardware serial for TX1/RX1 on Teensy for logging to SL
// Set I2C address for each sensor
MS5837 Psensor;
TSYS01 Tsensor;

String TSYS;
String TBAR02;
String PBAR02;

void setup() {
  Wire.begin();

  Serial.begin(115200); // Initialize serial monitor
  Serial1.begin(115200); // Initialize TX1/RX1
  pinMode(LEDpin, OUTPUT); // Set pinMode for LED control

  // Initialize pressure sensor
  // Returns true if the initialization was succesful
  // We can't continue with the rest of the program unless the sensor works properly
  Psensor.init();
  while (!Psensor.init()) {
    Serial.println("Psensor init FAIL!");
    Serial.print("\n\n\n");
    delay(5000);
  }
  Psensor.setModel(MS5837::MS5837_02BA);
  Psensor.setFluidDensity(1029);
  Psensor.read();
  Serial.println("Pressure sensor init SUCCESS!");
  Serial.print("P = "); Serial.print(Psensor.pressure() / 100.); Serial.println(" mbar \n")
  delay(1000);

  // Initialize temperature sensor
  // Returns true if initialization was successful
  // We can't continue with the rest of the program unless we can initialize the sensor
  Tsensor.init();
  Tsensor.read();
```

```
 while ((Tsensor.temperature() < 0) || (Tsensor.temperature() > 100)) {
    Serial.print("Tsensor init FAIL!");
    Tsensor.init();
    Tsensor.read();
    Serial.println("\n\n\n");
    delay(5000);
  }

  Serial.println("Temperature sensor init SUCCESS");
  Serial.print("T = "); Serial.print(Tsensor.temperature()); Serial.println(" C \n");
  delay(1000);

  // Blink 3 times to indicate successful initialization of both sensors
  for (int n = 0; n < 3; n++) {
    for (int r = 0; r <= 255; r++) {
      analogWrite(LEDpin, r);
      delay(1);
    }
    for (int r = 255; r >= 0 ; r--) {
      analogWrite(LEDpin, r);
      delay(1);
    }
  }
  Serial.println("TSYS01 [*C],TBar02 [*C], PBar02 [mbar]");
  Serial1.println("TSYS01 [*C],TBar02 [*C], PBar02 [mbar]");
}

void loop() {
  analogWrite(LEDpin, 255);
  Tsensor.read();
  Psensor.read();
  String TSYS = Tsensor.temperature();
  String TBAR02 = Psensor.temperature() / 100.;
  String PBAR02 = Psensor.pressure() / 100.;
  Serial.print(F("TSYS = "));
  Serial.print(TSYS);
  Serial.println(" *C");

  Serial.print(F("PBar02 = "));
  Serial.print(PBAR02);
  Serial.println(" mbar \n");

  Serial1.println(TSYS + "," + TBAR02 + "," + PBAR02);
//  delay(500);
  analogWrite(LEDpin, 0);
  delay(10);
```