RBR*cervata*
DATA CONTROLLER

INSTRUMENT GUIDE

rbr-global.com

Table of contents

1 Getting started

> This instruction guide is for the RBR*cervata* generation 2025. For previous models, please see RBR*cervata* FW ≤ 1.17.

RBR recommends new users start by reading General overview, which summarizes the features and use cases of the RBR*cervata*, then complete the bench testing steps outlined in Quick start guide.

Software overview outlines diagnostic commands, configuration behaviour, and deployment behaviour. Hardware overview provides a reference for hardware features. Maintenance guide provides instructions for routine maintenance tasks.

1.1 General overview

The RBR*cervata* data controller acts as a removable underwater data storage and power device to enable long-duration subsurface deployments of RBR data loggers, such as the RBR*concerto*^{3}, RBR*maestro*^{3}, RBR*quartz*^{3} BPR, and APT instruments. It supplies power to wired instruments and can archive logged data from RBR instruments over RS-232 and/or an inductive modem.

Its features include

- Backing up data for centralized retrieval by continuously downloading from instruments connected directly to it via RS-232 and/or indirectly via an RBR inductive mooring line modem (MLM) system.
- Powering directly-connected data loggers to extend their deployment lifetime.
- Reconfiguring data loggers to match a specified configuration.
- Facilitating bulk data retrieval by connecting it to a desktop computer via USB for fast data transfer.

Upon connection to the data/power port of a logger, the RBR*cervata* provides external power and continuously downloads from the data logger internal storage to the SDXC card internal to the data controller. Based on its own configuration, the data controller may also reconfigure the data logger as necessary. To facilitate bulk data retrieval, the data controller can be connected to a desktop computer over USB.

An instrument can be connected to or disconnected from the RBR*cervata* at any time. Power delivery to an attached instrument is constant, regardless of the state of the data controller, including deep sleep.

Fig. 1. As shipped with support kit, MLM test loop cable, and USB cable

1.1.1 Data collection

The RBR*cervata* will periodically download data from the attached instrument. To avoid the instrument running out of memory, the data controller can be configured to automatically clear the instrument memory and restart the deployment to minimize downtime.

Datasets are stored on an internal SD card in an RSK format compatible with Ruskin, and may be downloaded onto a connected desktop computer when the RBR*cervata* is mounted as a USB storage device. Data within these RSKs is only minimally processed, and each dataset must be opened with Ruskin at least once before it can be opened with RSKtools.

1.1.2 Energy conservation

The RBR*cervata* will enter a deep sleep mode whenever it is not downloading from the connected instrument. At a configurable interval, it will wake, download any new data from the instrument, then return to sleep. The sleep duration can be configured to balance the trade-off between power consumption and the recency of data upon retrieval of the data controller.

1.1.3 Deployment management

The RBR*cervata* USB port exposes a diagnostic terminal providing the user with a set of scripts to monitor, control, and troubleshoot the data controller behaviour (e.g. starting/stopping deployment, downloading data, checking memory usage, updating the controller configuration). The RBR*cervata* is able to alter instrument deployment parameters according to its internal configuration (e.g., sampling rate, bursting parameters). In the event that an instrument configuration doesn't match that which the data controller expects, it will stop, reconfigure, and restart the instrument.

For complete instructions see Quick start guide.

1.1.4 Status LEDs

To view the status LEDs, open the RBR*cervata* following the instructions in Opening the RBR*cervata*.

Green LED	Red LED	Meaning
Off	Off	The SD card is not in use, the system is powered down, or the system is in deep sleep.
On (single blink)	On (single blink)	Immediately upon power-on, both LEDs blink once.
Off	Blinking (short)	The data controller is writing to the SD card as it downloads from a connected instrument, or as it collects internal diagnostic information.
Off	Blinking (long)	The system is applying a configuration or firmware update.
Off	Blinking (SOS; i.e., short-short-short / long-long-long / short-short-short)	A configuration or firmware update has failed.
On (solid)	Off	A configuration or firmware update has succeeded. It should reboot shortly after.

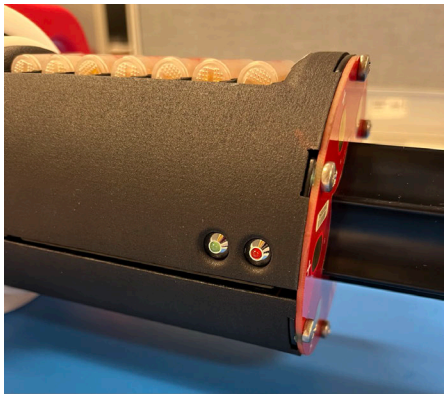


Fig. 2. Status LEDs

1.1.5 Support kit

The RBR*cervata* ships with a support kit that includes

- 1x O-ring removal tool
- 1x Silicone grease
- 5x O-ring 2-243, RBR #0002408
- 5x O-ring M2.0x112 N70, RBR #0010762
- 7x Desiccant canisters 12.4mm o.d. x 35mm, RBR #006170

Fig. 3. Support kit contents

1.1.6 Cables

In addition to the support kit, the RBRcervata ships with

- 1x 2m USB MCIL-6-FS-FLS to USB-C cable, RBR #0011088
- 1x Inductive test loop cable, RBR #0008512

Fig. 4. MCBH-6FS inductive test loop cable

1.2 Quick start guide

Tools needed

- USB to MCIL-6-FS cable (provided)
- MLM test loop (provided)
- Computer with a USB port
- Serial terminal

The deep-sleep feature will be disabled while the RBR*cervata* is connected to the computer USB port. If you connect the USB cable while the data controller is in a deep-sleep state, it can take up to 15s to boot and register on your computer.

1.2.1 Configuration

RBR configures each RBR*cervata* to work with a specific instrument. For details particular to your instrument or to change the configuration, please contact RBR.

To estimate power and telemetry requirements, use the RBR*cervello* Deployment Calculator.

Use your instruments or sub-surface modems (SSM) only with a properly configured data controller. Failure to do so will result in invalid data.

1.2.2 Before a deployment

1. Open the RBR*cervata* following the steps in Opening the RBR*cervata*.
2. Install fresh batteries (48 x D-cells) and desiccant following the steps in Hardware overview.
3. Connect the RBR*cervata* USB MCBH port to a computer via the provided USB to MCIL-6-FS cable.
4. Open a terminal on the corresponding communications port following the instructions in Using the USB port.
5. Press **Enter** and wait until you see the `RBRcervello>` prompt.
6. Type `disable` and press **Enter**. See disable for more information.
7. (*Optional but recommended*) Apply any pending software updates by following the instructions in Updating the instrument.
8. (*Optional*) Apply any pending configuration updates by following the instructions in Updating the instrument.

Fig. 5. USB interface port

⚠ Update the firmware and configuration separately otherwise the configuration update may not be applied.

9. Test the MLM loop by following the instructions in MLM troubleshooting.
10. Type `scrub-instrument` and press **Enter** to erase any data that might be stored in the instrument(s). See scrub-instrument for more information.
11. Type `scrub-controller` and press **Enter** to erase any data that might be stored in the RBR*cervata*. See scrub-controller for more information.
12. Type `set-cervello-clock YYYY-MM-DDTHH:MM:SS` with the appropriate time in UTC and press **Enter** to synchronise the clock on the RBR*cervata.* See set-cervello-clock for more information.

ℹ If the RBR*cervata* establishes communications with an instrument within 24h of the RBR*cervata* clock being set, the RBR*cervata* will set the instrument clock to its own clock. This allows the RBR*cervata* to update the clocks of instruments left logging while the RBR*cervata* is retrieved along with their data.

13. Type `status` and press **Enter** to verify that the instruments and the controller have the `disabled` status. See status for more information.
14. (*Optional*) Type `enable` and press **Enter** to begin the deployment and enable the instruments immediately. See enable for more information.
15. (*Optional*) Type `status` and press **Enter** to verify that the instruments and the controller have the `logging` status.
16. Disconnect the USB to MCIL cable from the RBR*cervata*.
17. Close the RBR*cervata* following the instructions in Closing the RBR*cervata*.

Fig. 6. MLM port

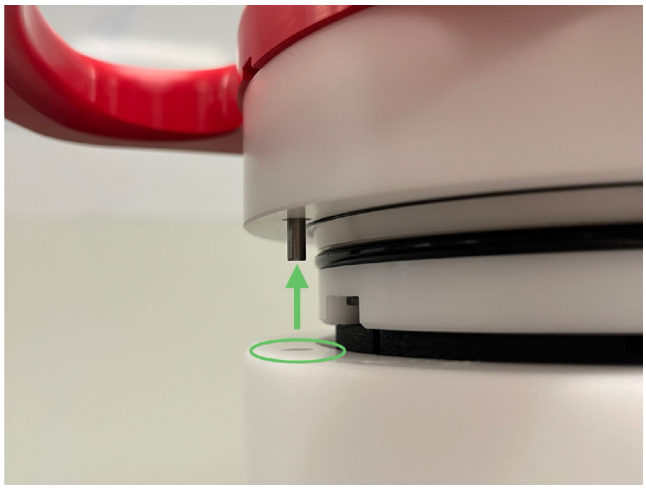


Fig. 7. Endcap alignment pin

1.2.3 After a deployment

After retrieving the RBR*cervata*

1. Connect the RBR*cervata* USB MCBH port to a computer via the provided USB to MCIL-6-FS cable.
2. Apply power using the USB to MCIL-6-FS cable as the battery pack might be depleted.
3. Open a terminal on the corresponding communications port following the instructions in Using the USB port.

> Baud rate, the number of data/parity/stop bits, and flow control settings do not matter for USB communication devices.

4. Press **Enter** and wait until you see the `RBRcervello>` prompt.
5. Type `disable` and press **Enter**. See disable for more information.
6. Type `set-cervello-clock YYYY-MM-DDTHH:MM:SS` with the appropriate time in UTC and press **Enter**. This will allow for timestamp correction after retrieving the data. See set-cervello-clock for more information.

> If the RBR*cervata* establishes communications with an instrument within 24h of the RBR*cervata* clock being set, the RBR*cervata* will set the instrument clock to its own clock.
> This allows the RBR*cervata* to update the clocks of instruments left logging while the RBR*cervata* is retrieved along with their data.

7. Type `status` and press **Enter** to verify that the instruments (if any) have the `stopped` state and the controller has the `disabled` state.
8. Retrieve data via the USB port following the steps in Retrieving data.

1.2.4 Retrieving data

The RBR*cervata* data controller supports *USB On-The-Go*: allowing stored data to be retrieved via its USB port as if it were a standard USB storage device.

> ⚠ As of the 2025 hardware revision of the RBR*cervata*, the internal SD card is no longer removable. If you experience issues retrieving data from the data controller, please contact support@rbr-global.com.

> ℹ **Tools needed**
> - USB to MCIL-6-FS cable (provided)
> - Computer with a USB port
> - Serial terminal

To retrieve data via the external USB port

1. Power on the data controller.
2. Close Ruskin.

> ⚠ Always close Ruskin before connecting the controller to your computer.

3. Connect the provided USB patch cable to the controller and computer.
4. Open the serial terminal as described in Using the USB.
5. Disable the controller using the `disable` command. See disable for more information.
6. Check that all the instruments are disabled using the `status` command. See status for more information.
7. Use the `mount-storage` command to expose the data to the host computer. See mount-storage for more information.

```
RBRcervello> mount-storage
Checking deployment status...
Stopping services...
Mounting data partition as read-only
```

> ℹ For the drives to be detected in MacOS, you may need to disconnect and reconnect the USB cable to the computer.

8. Copy the data from the **RBRcervello** drive that is shown on the computer.
9. Unmount or eject the **RBRcervello** and **RBR Update** drives.
10. Use the `unmount-storage` command to cleanly disconnect the drives. See unmount-storage for more information.

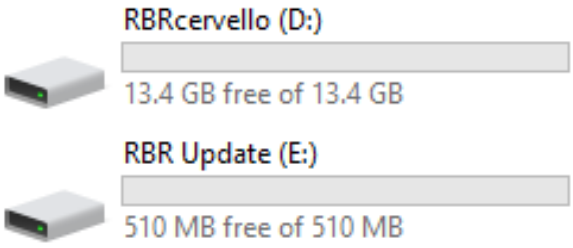


Fig. 8. Drives after using mount-storage

2 Software

The RBR*cervata* data controller can fully manage RBR loggers connected to it based on its stored settings. This includes downloading, clock syncing, dataset rollover, and configuration. All of the user-accessible controls are available through a command-line console on its USB port (using the USB communications device class protocol). See Using the USB port for more information.

> The RBR*cervata* incorporates software components in common with the RBR*cervello* buoy controller. As a result, this documentation and the the RBR*cervata* diagnostic commands may occasionally reference the RBR*cervello*.

2.1 The diagnostic menu

This menu can be accessed via the RBR*cervata* USB port. This menu is a fully-featured (Restricted) Bash terminal, supporting line editing and **tab** completion. Type `help` and press **Enter** to display a list of the the commands you will need to set up a deployment using the RBR*cervata* and RBR loggers. Each command can print a short manual by executing them with the `-h` or the `--help` argument.

```
RBRcervello> help

RBRcervello Menu.

These are the available commands to manage RBRcervello

   disable               Disable deployment.
   status                Provide a general overview or detailed information on
                         controller and instruments.
   mount-storage         Mount the data to the host computer.
   unmount-storage       Unmount the data to the host computer.
   diagnose              Diagnostics for MLM and telemetry.
   transparent-link      Establish a transparent link to an instrument
                         serial port.
   config-check          Print the configuration of a specific RBR subsystem.
   edit-config           Edit the configuration of a specific RBR subsystem.
   set-cervello-clock    RBRcervello clock setter.
   cervello-clock        RBRcervello clock checker.
   scrub-controller      Clear all data stored in the SD card.
   scrub-instrument      RBR instrument scrubber.
   enable                Enable deployment according to internal configuration
                         files.
   reboot-controller     Reboot the controller.
   help                  Show this menu.
```

See Diagnostic menu for more information.

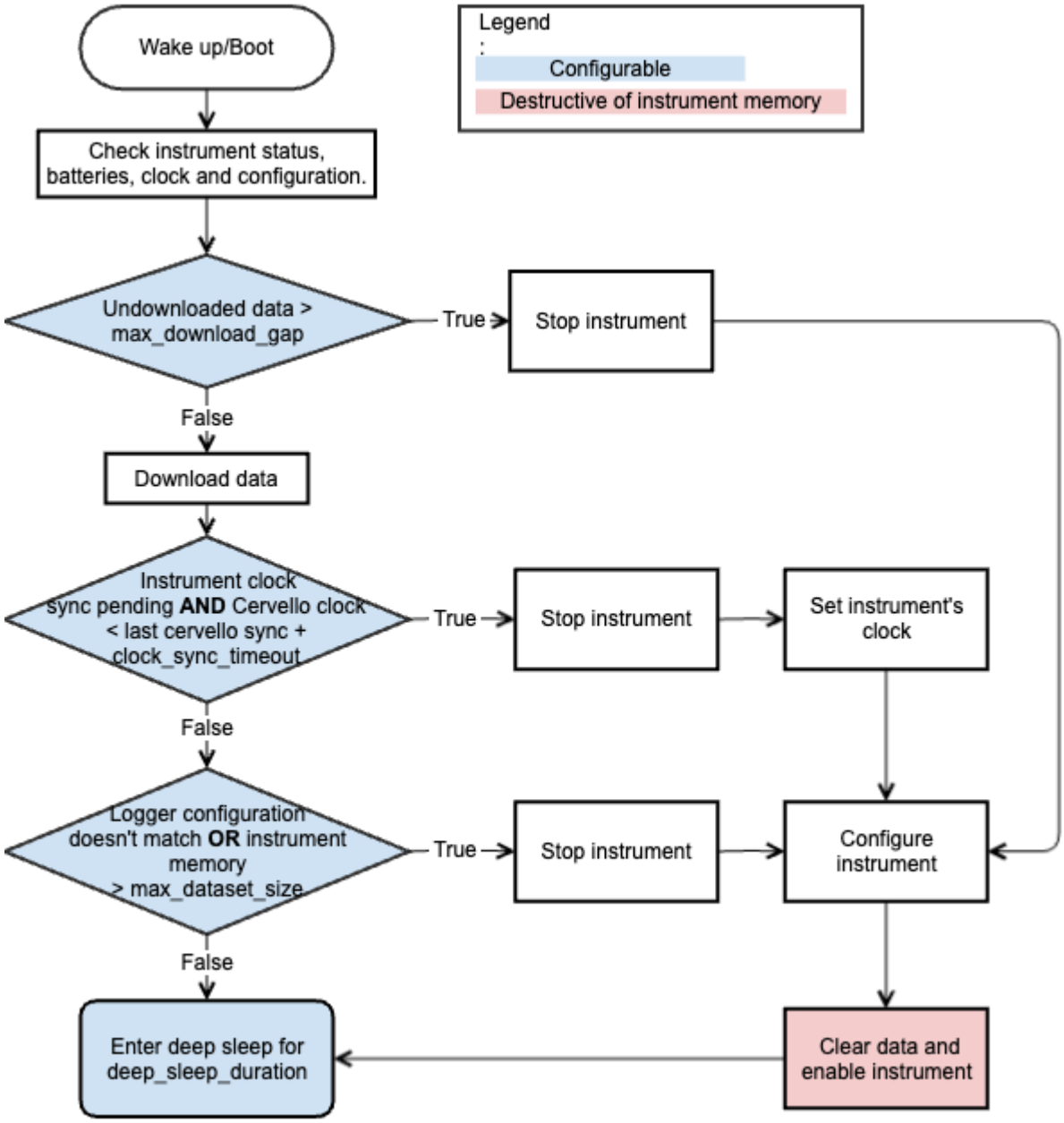
2.2 Configuration files

The RBR*cervata* behaviour can be managed using the configuration files. There is one RBR*cervello* configuration file (`user.json`) and an instrument configuration file (`deployment01.json` , `deployment02.json` , `deployment03.json` , etc.) for each instrument. The instrument files are listed in the order they appear in the RBR*cervello* file. The RBR*cervello* configuration file defines the RBR*cervata* behaviour, while the instrument configuration files define the configuration to be loaded into each instrument for each deployment.

See RBR*cervello* configuration file reference, Logger configuration file reference, and Updating the instrument for more information.

2.3 Behaviour in an active deployment

If the RBR*cervata* establishes communications with an instrument within 24h of the RBR*cervata* clock being set, it will set the instrument clock to its own clock.
This allows the RBR*cervata* to update the clocks of instruments left logging while the RBR*cervata* is retrieved along with their data.



2.4 Updating the instrument

The data controller has a USB interface for retrieving data from the unit, and for installing firmware and configuration updates.

> If you connect the USB cable while the data controller is in a deep-sleep state, it can take up to 15s to register on your computer.

> **Tools needed**
> - USB to MCIL-6-FS cable (provided)
> - Computer with a USB port
> - Serial terminal

Procedure

Use the diagnostic menu to retrieve data from the data controller via the external USB port.

1. Power the unit.
2. Close Ruskin.

> Always close Ruskin before connecting the controller to your computer.

3. Connect the provided USB patch cable to the instrument and computer.
4. Open the serial terminal as described in Using the USB port.
5. Disable the instrument using the `disable` command. See disable for more information.
6. Check that all the instruments are disabled using the `status` command. See status for more information.
7. Use the `mount-storage` command to expose the data to the host computer. See mount-storage for more information.

```
RBRcervello> mount-storage
Checking deployment status...
Stopping services...
Mounting data partition as read-only
```

8. In **RBR Update**, create the **rbrcervello** folder.

> The RBR*cervello* does not currently support simultaneously updating the configuration and the firmware.

9. Copy the relevant update files into the correct subfolder.
    a. If you are updating the RBR*cervata* firmware, create the **update** folder in **rbrcervello** and extract the provided update bundle (extracted, the file name will end with the extension .rbrbundle) into it.

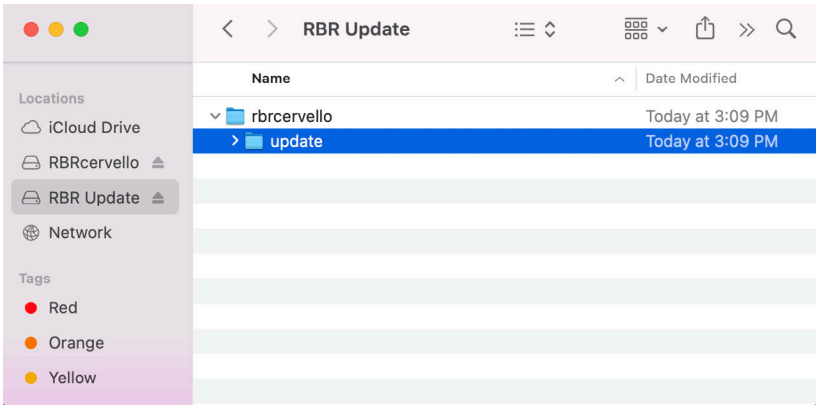


Fig. 9. Extract firmware .rbrbundle files into rbrcervello/update

b. If you are updating the RBR*cervata* configuration, create the **config** folder in **rbrcervello** and extract the provided configuration bundle (extracted, the file name(s) will end with the extension .json) into it.

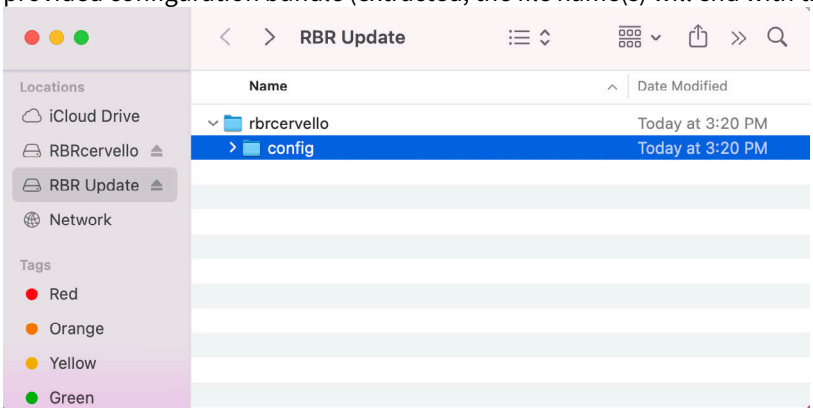


Fig. 10. Extract configuration .json files into rbrcervello/config

> The user may modify the configuration files according to the cervello configuration file reference and logger configuration file reference.

9. Eject the **RBR Update** and **RBRcervello** drives from the computer before proceeding.

> Failing to eject the drive before disconnecting the cable or executing the `unmount-storage` command could result in an unsuccessful update.

10. Use the `unmount-storage` command to cleanly disconnect the drive. See unmount-storage for more information.

> The update may take up to 5mins to complete.

11. Once the update process is finished, the instrument will reconnect to the computer. If it doesn't, try reconnecting the USB cable.

Light indicator

While applying updates, the red light on the data controller will flash steadily. Once all updates are installed, the red light will cease flashing and the green light will turn on. The data controller will reboot itself within 5s and the green light will turn off.

If an error occurs during the process, the red light will repeatedly flash the Morse code representation of “SOS”: three short flashes, three long flashes, and three short flashes. If this occurs, contact us immediately at support@rbr-global.com or via the RBR website.

Once the red light has turned off and stays off for several seconds, reconnect the instrument.

Update and revert feature

The data controller runs the same checks described in config-check before applying a new configuration. If the checks fail, the data controller will flash “SOS” and will not update the configuration.

3 Hardware

Required tools
- 5mm Hex/Allen key.
- Nitrile gloves.

Recommended tools
- 5mm Hex/Allen torque wrench/driver

3.1 Opening the RBRcervata

The RBR*cervata* is serviced by opening the battery end-cap, allowing access to the controller, desiccant, and batteries, which are mounted to an electronics assembly atttached to the end-cap.

The battery end-cap is the one with the MCBH connectors.

1. Unscrew the retaining bolts from the battery end cap using a 5mm Hex/Allen key.
2. Pull the end-cap out of the RBR*cervata* pressure housing.

Do not twist the battery end-cap.

3. Remove and save the retaining bolts.

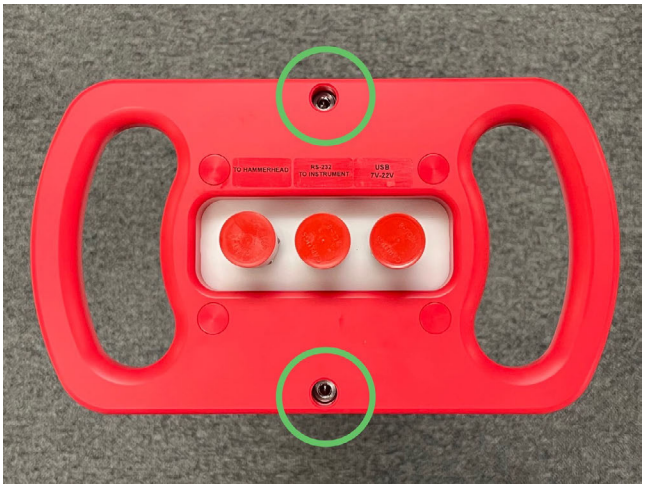


Fig. 11. Retaining bolts, tightened to 1.7N-m

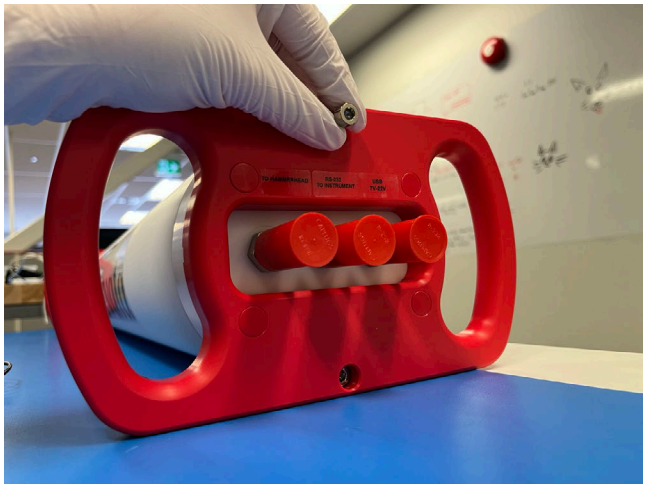


Fig. 12. Remove retaining screws with the RBRcervata lying on one side

3.2 Closing the RBRcervata

1. Place the RBR*cervata* electronics and connector end-cap assembly back into the pressure housing.

The battery end-cap has a small metal alignment pin which slots into a hole on the mating face of the pressure housing. If the alignment pin is correctly aligned, only a small push should be required to insert the end-cap assembly.

Do not attempt to force the end-cap and do not begin torquing the retaining bolts until the end-cap is fully seated into the pressure housing.

2. Re-insert the retaining bolts.
3. Torque the retaining bolts to 1.7N-m (15 in-lbs) using a 5mm hex torque wrench/driver.

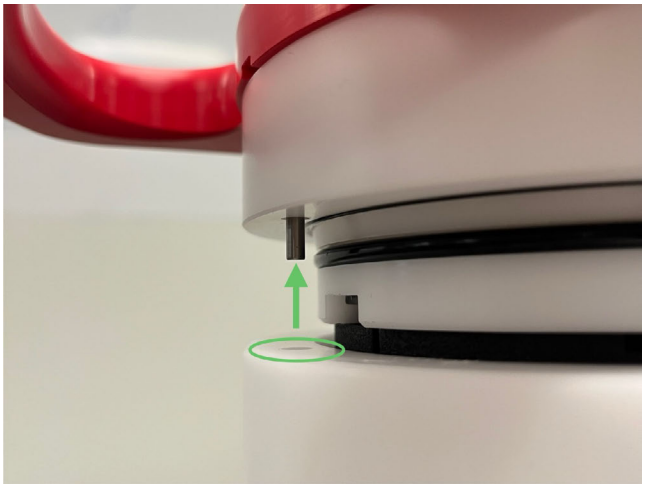


Fig. 13. Note the position of the alignment pin

Fig. 14. The alignment pin partially seated in its hole

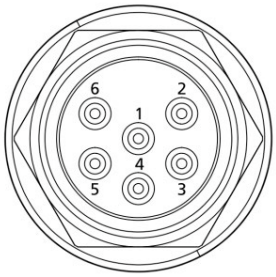
3.3 Using the USB port

Tools needed

- USB to MCIL-6-FS cable (provided)
- Computer with a USB port
- Serial terminal

Fig. 15. MCBH-6-MP USB port position

External MCBH-6-MP USB connector pinout



Pin No.	USB
1	Ground
2	Power +7 to +22 V
3	N/C
4	Power: 5V
5	D-
6	D+

USB Patch cable

Part No.	2-metre patch cable	Notes
0011088	USB - MCIL-6-FS to USB-C	For instruments with USB output. Includes a power terminal block. No extension cable available for USB.

Windows

On a Windows-based workstation, use any terminal emulator console that supports a serial port, such as Tera Term.

Tera Term

Tera Term is an open-source terminal emulator that lets you connect to hosts via serial port or the network. The default settings should work with the *RBRcervello* interface. In the unlikely event when they do not, load the configuration file provided below using the **Setup** > **Restore setup** menu.

Teraterm_RBRcervello.INI

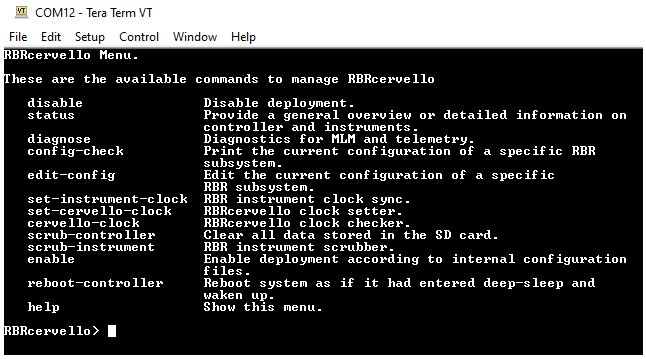
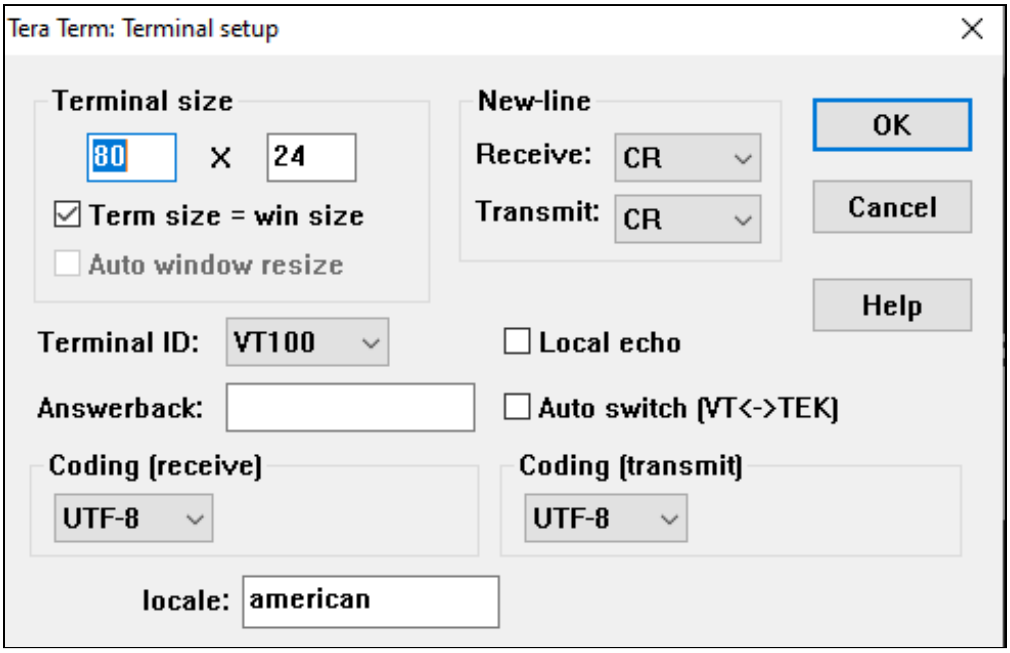


Fig. 16. Tera Term

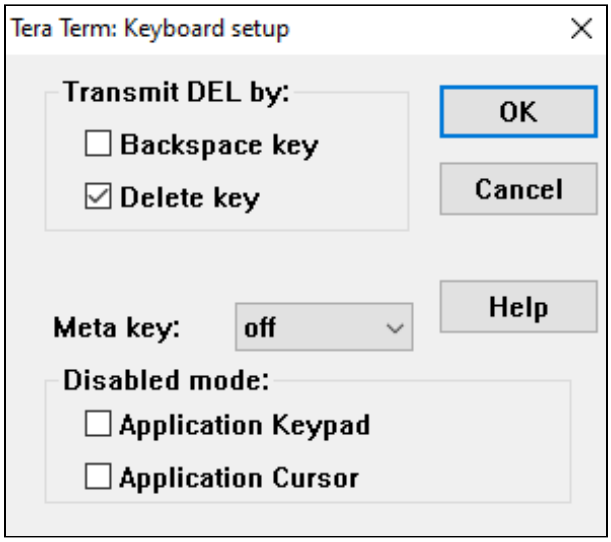
> Highlighting text on the terminal automatically copies it to the clipboard. Right-clicking on the terminal will paste the saved text on it.

Alternatively, configure the settings manually as follows

1. Go to **Setup** > **Terminal setup**, enter the terminal size

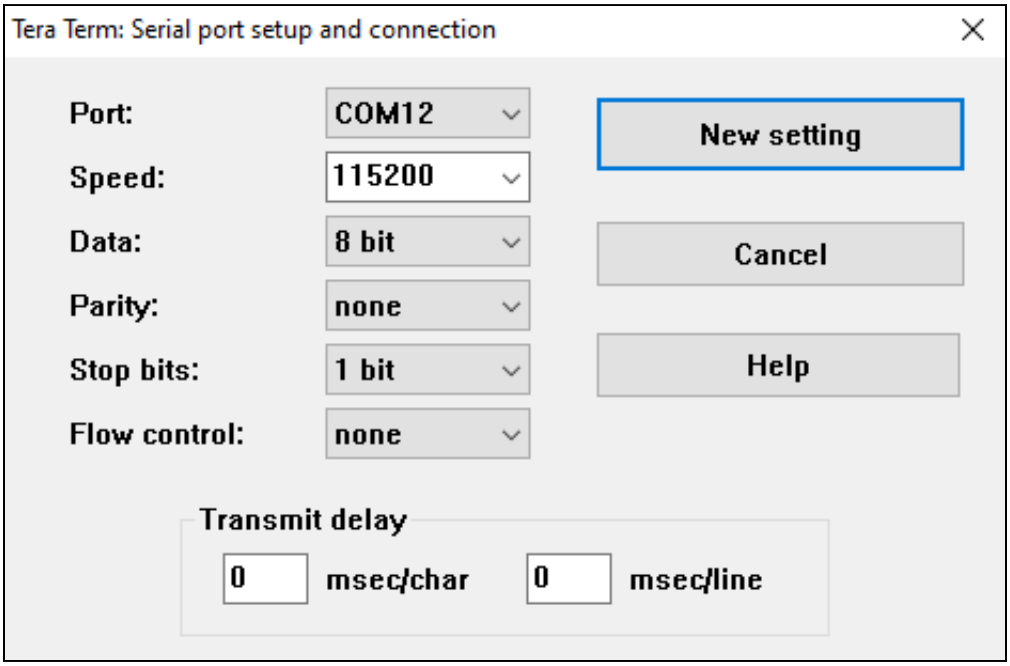


2. Go to **Setup** > **Keyboard setup**, select **Delete key**



3. Go to **Setup** > **Serial port setup and connection** , select the correct COM port:

Baud rate, the number of data/parity/stop bits, and flow control settings do not matter for USB communication devices.



Linux

On a Linux-based workstation, use any terminal emulator console that supports a serial port, such as Minicom.

Minicom

Minicom is a serial communication program that connects to devices through a GNU/Linux serial ports. Use the `-s` command to set up the terminal with the right port and settings, or execute it with explicit settings at runtime with the correct device:

```
minicom -D /dev/ttyACM0
```

To copy and paste text from and to a Linux terminal, use Ctrl+Shift+C and Ctrl+Shift+V.

Baud rate, the number of data/parity/stop bits, and flow control settings do not matter for USB communication devices.

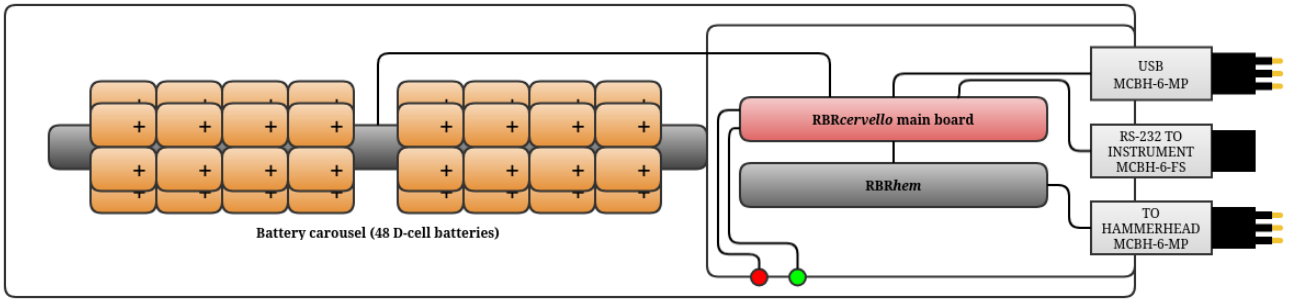
```
RBRcervello Menu.

These are the available commands to manage RBRcervello

   disable               Disable deployment.
   status                Provide a general overview or detailed information on
                         controller and instruments.
   mount-storage         Mount the data to the host computer.
   unmount-storage       Unmount the data to the host computer.
   diagnose              Diagnostics for MLM and telemetry.
   transparent-link      Establish a transparent link to an instrument
                         serial port.
   config-check          Print the configuration of a specific RBR subsystem.
   edit-config           Edit the configuration of a specific RBR subsystem.
   set-instrument-clock  RBR instrument clock sync.
   set-cervello-clock    RBRcervello clock setter.
   cervello-clock        RBRcervello clock checker.
   scrub-controller      Clear all data stored in the SD card.
   scrub-instrument      RBR instrument scrubber.
   enable                Enable deployment according to internal configuration
                         files.
   reboot-controller     Reboot the controller.
   help                  Show this menu.

RBRcervello>
```

Fig. 17. Minicom

```
+-----------------------------------------------------------------------+
| A -    Serial Device      : /dev/ttyACM0                              |
| B - Lockfile Location     : /var/lock                                 |
| C -   Callin Program      :                                           |
| D -  Callout Program      :                                           |
| E -    Bps/Par/Bits       : 115200 8N1                                |
| F - Hardware Flow Control : No                                        |
| G - Software Flow Control : No                                        |
|                                                                       |
|    Change which setting?                                              |
+-----------------------------------------------------------------------+
        | Screen and keyboard      |
        | Save setup as dfl        |
        | Save setup as..          |
        | Exit                     |
        | Exit from Minicom        |
        +--------------------------+
```

Fig. 18. Minicom serial settings

3.4 Block diagram



4 Maintenance

4.1 Replacing the O-rings

See Hardware overview for details on accessing the O-rings.

4.1.1 Location of the O-rings

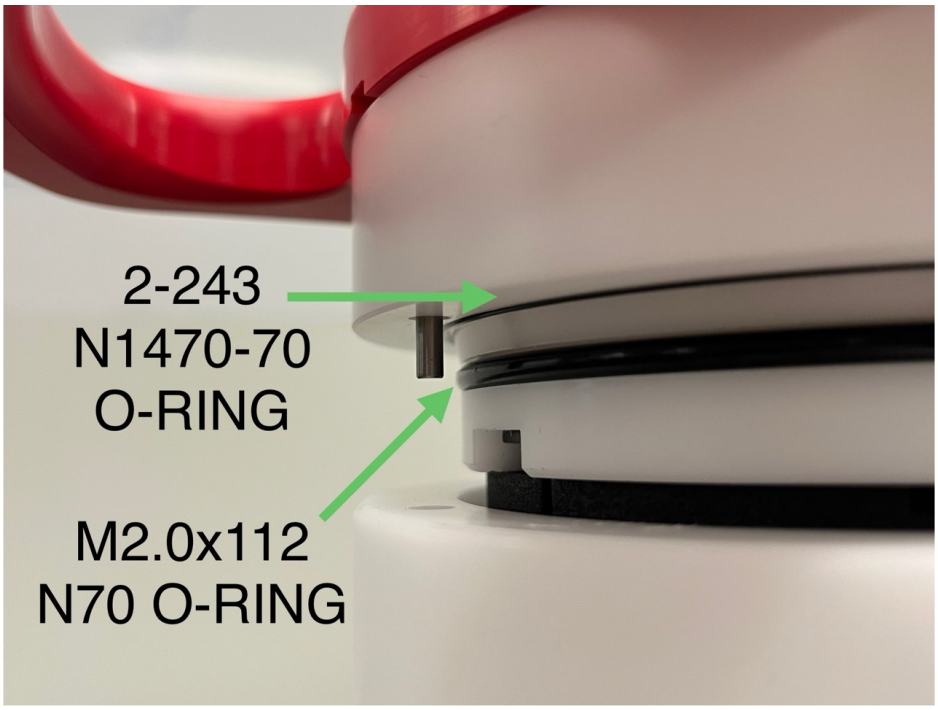


Fig. 19. Battery end-cap O-rings

4.1.2 Inspecting O-rings

Pay attention to the following areas

- The surface of the O-ring itself.
- The mating surface on the inside of the case between the threads and the open end.
- The inner surfaces of the groove in the end cap where the O-ring sits.

Any dirt should be removed by thoroughly wiping with a soft, lint-free cloth. Never use any material or tool which could scratch the O-ring or any of its mating surfaces. If dirt is present in the O-ring groove, remove the O-ring as described below to allow thorough cleaning of the groove. If an O-ring needs to be removed for any reason, it should be replaced.

If the O-ring is scratched, cut, distorted, perished or defective in any other way it must be replaced. If any of the surfaces of the O-ring groove are scratched, pitted or otherwise damaged, the logger may need to be returned to RBR for refurbishment: please contact us for advice.

4.1.3 Replacing an O-ring

1. Use the plastic O-ring tool tool provided in the RBR*cervata* support kit to lever the O-ring from its groove. This may require some physical effort.

> ⚠ Do not use a metal screwdriver or any other tool which may scratch the surfaces of the O-ring groove, doing so could render the end cap unusable.

2. Slide the O-ring out of its groove and off the logger. The O-ring may need to stretch quite a bit as it is pushed off; this requires some effort but can be done by hand.
3. Clean the groove thoroughly with a soft, lint-free cloth and compressed air, if necessary.
4. Inspect the O-ring groove carefully for scratches and other damage.

> ⚠ Take care to not scratch or damage the O-ring groove in any way. Contact support@rbr-global.com for advice if it appears damaged.

5. Apply a light film of silicone grease to all three inside surfaces of the groove.

> ⚠ Be careful to not trap any dirt, hairs or lint in the O-ring groove.

6. Select the proper new O-ring and make sure it is not damaged.
    a. The face O-ring (sealing between the flat end faces of the battery end-cap and the pressure housing) is a **2-243 N1470-70 O-ring**
    b. The bore O-ring (sealing between the “plug” of the battery end-cap and the inside of the pressure housing) is an **M2.0x112 N70 O-ring**
7. Lubricate the new O-ring with a very light film of silicone grease to ease installation.
8. Install the new O-ring by pushing it into place and popping it into its groove.
9. When the new O-ring is in place, inspect it once more for scratches and dirt.
10. Wipe away any silicone grease deposited on the end cap.

4.2 Replacing the batteries

See Hardware overview for details on accessing the battery carousel.

The RBR*cervata* is powered using 48 alkaline, NiMH, or $LiSOCl_2$ D-cell batteries. RBR recommends using fresh batteries before each deployment.

The batteries are all arranged with their positive terminal pointing towards the battery end cap.

Remove batteries from the positive terminal end first. Insert batteries with the negative terminal end first.

When using alkaline or nickel metal hydride batteries, simply put them in the carousel, ensuring correct polarity. These batteries are magnetic and hold in place on the carousel with no additional steps required.

Fig. 20. Batteries' positive terminals face the battery end-cap

Lithium batteries do not have enough magnetic material to hold them together in assembly. Use the red retention bands provided with your RBR*cervata*, as shown.

Fig. 21. Lithium batteries retained in carousel

4.3 Replacing the desiccant

Refer to Hardware overview for details on accessing the desiccant canisters.

4.3.1 Replacing desiccant canisters

RBR recommends replacing all desiccant canisters before each deployment. The use of fresh desiccants will keep the RBR*cervata* compartment dry and prevent the electronics from being damaged due to condensation inside the controller. As a preventative measure, RBR recommends always using fresh desiccant packs and, when possible, servicing the instrument in a cool, dry place.

⚠ Avoid using a metal screwdriver or any other tool which may scratch the electronics assembly.

To remove each desiccant canister, use a finger or a soft plastic or wood tool to lever the cap-side of the canister up and out of its holder.

To install each fresh desiccant canister, first press the side *opposite* the canister cap into the holder, then press the cap-side of the canister into the holder. Both sides of the canister should click into place.

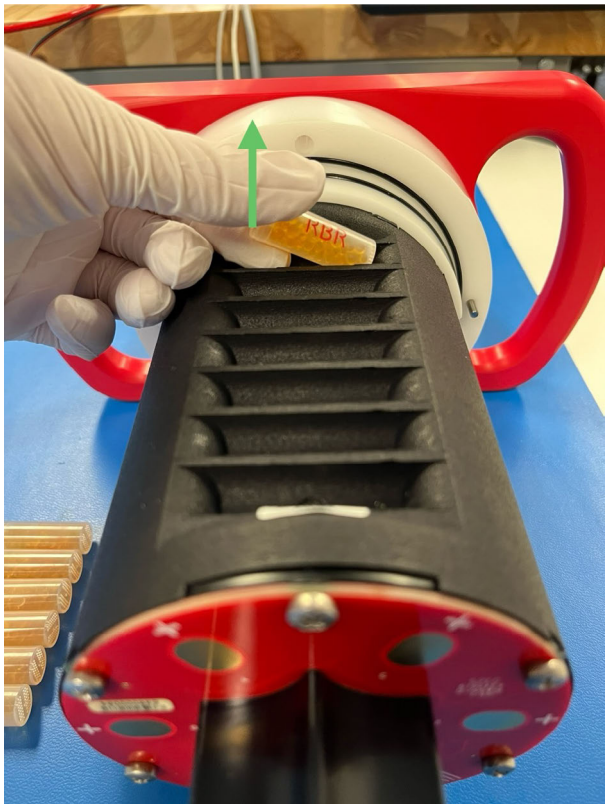


Fig. 22. Remove desiccant canisters cap-first

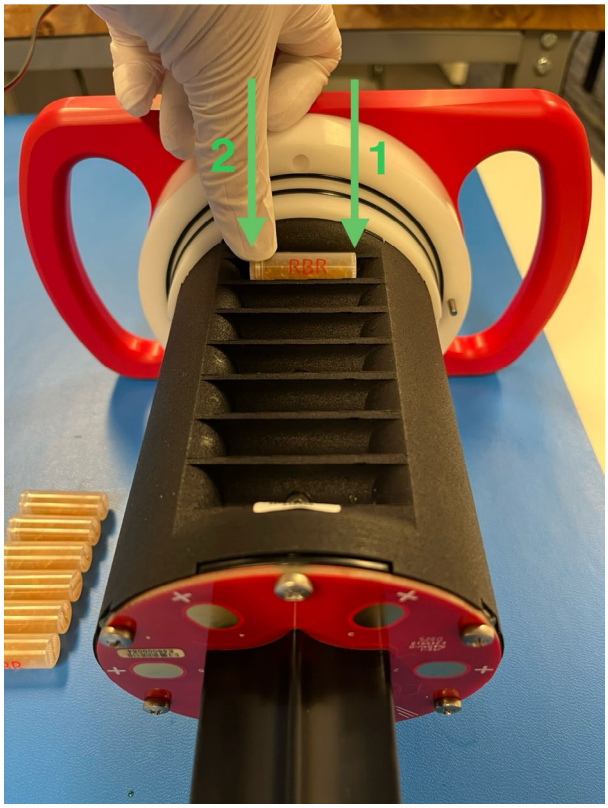


Fig. 23. Insert desiccant canisters end-first

4.3.2 Reusing the desiccant

All instruments ship with fresh reusable desiccant capsules. They use a cobalt-free colour changing indicator dye. Orange indicates fresh desiccant, while green indicates it is saturated (about 15% water by weight). Once exhausted, the capsules can be replaced with new ones (available from RBR), or refreshed.

Fig. 24. Desiccant capsules. (orange) Fresh. (green) Saturated

Refreshing the desiccant

Follow the steps below to refresh the desiccant.

1. Remove the saturated silica beads from their capsule.
2. Place them in the oven and heat at 120°C (250°F) for about two hours.

> Always remove the beads from their capsule before refreshing!
> The capsule will deform if heated to 120°C.

3. Take the refreshed beads out of the oven and return them to the capsule.

> Return the refreshed beads to the capsule immediately after reheating!
> If left outside the capsule, the desiccant will trap moisture and go back to green.

4. Wait until the silica beads cool down. Once cool, the desiccant is ready to be reused.

4.4 Cables and connectors

Cable bend radius

The smallest bend radius for RBR supplied cables is 15cm.

Lubricating the connectors

Lubrication improves watertight sealing, prevents corrosion, and reduces the force required to de-mate the connector. Use the silicone compound provided with your instrument.

- Apply the silicone compound to all female connectors before every mating
- Ensure each connector hole is filled with approximately 30% lubricant

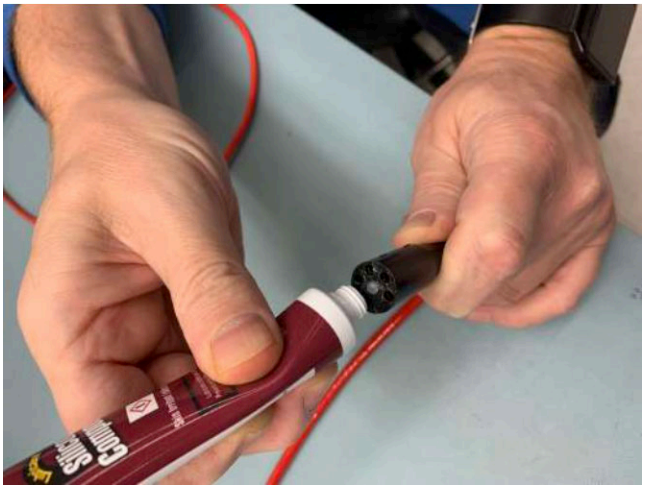


Fig. 25. Lubricating a connector

Reducing mechanical stress

- Do not pull on the cable
- Hold onto the connector to pull out the cable
- Disconnect by pulling straight out, not at an angle
- Avoid sharp bends at the point where the cable enters the connector
- Avoid angular loads on the connector

4.5 Repairs

RBR supports all our products. Contact us immediately at support@rbr-global.com or via the RBR website if there are any issues with your instrument. Please have the model and the serial number of the unit ready. Our support team will work to resolve the issue remotely. In some cases, you may have to return your instrument to RBR for further servicing.

There are no user-repairable parts of the instrument. Any attempt to repair without prior authorisation from RBR will void the warranty. Refer to the RBR warranty statement.

To return a product to RBR for an upgrade, repair, or calibration, please contact our support team to obtain a return merchandise authorisation code (RMA) and review the detailed shipping information on the RBR website.

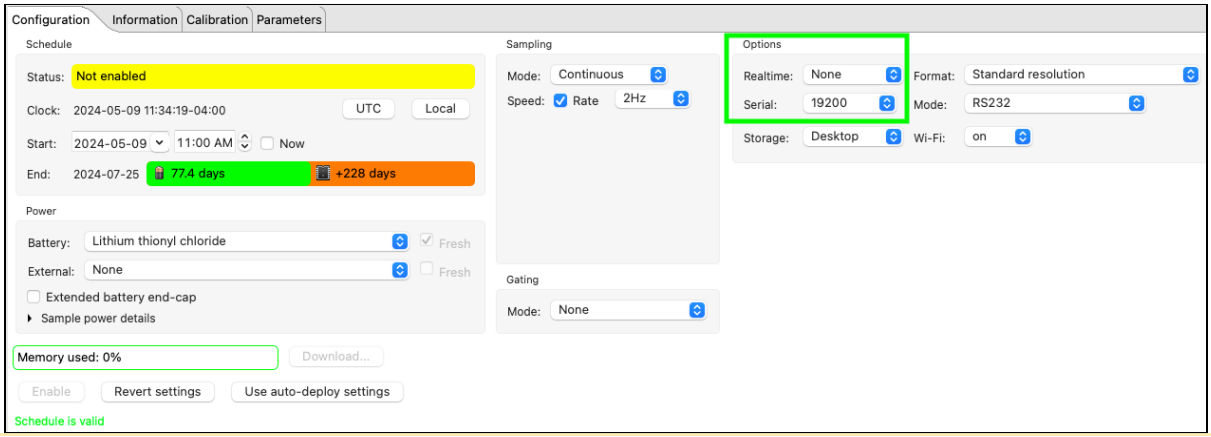
5 Troubleshooting

The RBR*cervata* system comprises multiple components, and thus troubleshooting may be a multi-stage process. There are separate procedures to verify correct functionality of each component.

Follow the steps outlined in Using the USB port to open the command menu.

5.1 Communication issues

1. Use the `status` command, to check the status of the instrument. See status for more information.
2. If the instrument status is `commchannelerror`, see MLM troubleshooting.
3. If the instrument status is `loggererror`, use Ruskin to check if your logger is communicating at 19200 baud. If the issue persists, contact support@rbr-global.com.



If you change any settings on Ruskin, clear the data from the instrument and the data controller, as described in Quick start guide.

5.2 MLM troubleshooting

Tools needed

- USB to MCIL-6-FS cable (provided)
- MLM test loop (provided)
- Computer with a USB port
- Serial terminal

The RBR*cervata* uses the mooring line modem (MLM) inductive modem to communicate with the instrument mounted on the mooring line. The MLM system consists of a head-end modem (HEM) included within the RBR*cervata* housing, and a sub-surface modem (SSM) coupled onto the instrument. The instrument needs to be configured to communicate at the baud rate configured in the SSM which is typically 19200 bits-per-second.

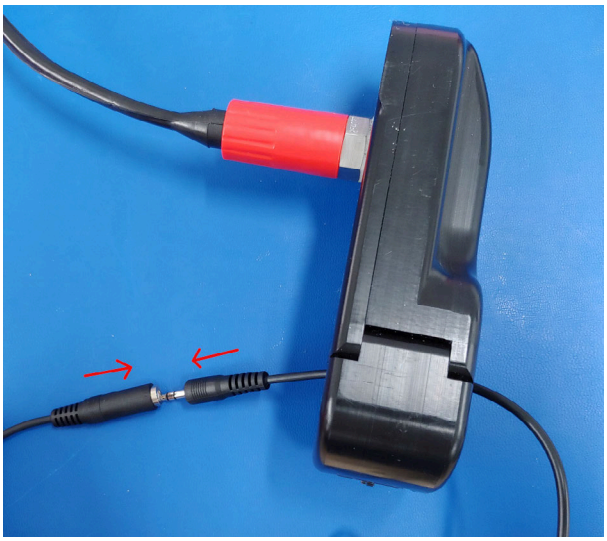
The MLM diagnostic task is a useful tool that checks the connection and settings of the HEM and the SSMs attached to the mooring line. When it finds an error, it will indicate the possible cause in the console.

5.2.1 MLM bench-test

To bench-test the system, connect all of its components together:

1. (*Optional*) Remove the internal assembly of the RBR*cervata* from the housing following the instructions in Hardware overview.
2. Connect the MLM test loop cable to the RBR*cervata* (*direct connect* or *TO HAMMERHEAD*, depending on your model).
3. Connect the instrument to its ferrite holder.
4. Pass the test loop through the ferrite holder and connect the audio jack connectors.
   The loop does not need to make contact with any part of the holder: passing through the middle of the ferrite is all that is required for functionality.
5. Connect the MCIL-6-FS end of the patch cable to the diagnostic port.
6. Connect the USB end of the patch cable to your computer.
7. Open a terminal on the corresponding communications port following the instructions in Using the USB port.
8. Press enter and wait until you see the `RBRcervello>` prompt.
9. Type `disable` and press **Enter**. See disable for more information.
10. Type `config-check -c cervello`. See config-check for more information.
11. Use the **arrow keys** to find the value of the `ssm` field for each instrument then press **Q** to exit.
12. If the configuration of any instrument does not match the serial number of the associated SSM, edit the configuration with the `nano` text editor using `edit-config -c cervello` and press **ctrl + X** to exit. See edit-config for more information.
13. Type `diagnose mlm` and press **Enter**. See diagnose for more information.

Fig. 26. "TO HAMMERHEAD" port



The MLM test routine will start running. This routine will assess communication with the HEM and SSM attached to the line, check their configuration, and perform a bit-error-rate test.

Follow the instructions on the screen as sometimes it prompts for user interaction.

Example MLM diagnostic task output

```
RBRcervello> diagnose mlm
Checking deployment status...
Stopping services...
Waiting for publisher to terminate...
Starting MLM test
I think the HEM is on /dev/ttymxc0 at 115200 baud.
Trying to wake the HEM...
        error unknown command
        error unknown command
        RBR HEM-1000 1.500 237884
Found HEM 237884.
HEM firmware version 1.500 >= 1.300.
        commands now unlocked
Checking HEM configuration...
Checking:       baud rate BAUD: 115200
Setting:        baud rate BAUD: 115200
Checking:       sleep timeout (10ms) ZTO: 1000
Setting:        sleep timeout (10ms) ZTO: 1000
Checking:       hardware flow control enable HWF: 0
Setting:        hardware flow control enable HWF: 0
Checking:       poll enable PEN: 0
Setting:        poll enable PEN: 0
Checking:       poll interval time (min) PINT: 5
Setting:        poll interval time (min) PINT: 5
Checking:       poll delay time (10ms) PDEL: 500
Setting:        poll delay time (10ms) PDEL: 500
Checking:       poll display enable PDISP: 0
Setting:        poll display enable PDISP: 0
Checking:       poll cmd string PCMD: Setting:  x46,x30,x30
        poll cmd string PCMD: Checking: x46,x30,x30
        poll wakeup char enable PWUPEN: 1
Setting:        poll wakeup char enable PWUPEN: 1
Checking:       poll wakeup char PWUPCHAR: x0F
Setting:        poll wakeup char PWUPCHAR: x0F
Checking:       poll sleep enable (zzz) PSLPEN: 1
Setting:        poll sleep enable (zzz) PSLPEN: 1
Checking:       poll address PADD: 00
Setting:        poll address PADD: 00
Checking:       modem open channel timeout (10ms) MDOCDUR: 1000
Setting:        modem open channel timeout (10ms) MDOCDUR: 1000
Checking:       modem open channel permanent MDOCPERM: 0
Setting:        modem open channel permanent MDOCPERM: 0
Checking:       modem maximum retry count MDRTRY: 1
Setting:        modem maximum retry count MDRTRY: 1
Checking:       transparent max chars before send TPMXC: 50
Setting:        transparent max chars before send TPMXC: 50
Checking:       transparent send char TPSNDC: x0D
Setting:        transparent send char TPSNDC: x0D
Checking:       transparent send char enable TPSNDEN: 1
```

```
Setting:        transparent send char enable TPSNDEN: 1
Checking:       transparent send char as well enable TPSNDC2: 1
Setting:        transparent send char as well enable TPSNDC2: 1
Checking:       transparent char timeout (10ms) TPCTO: 10
Setting:        transparent char timeout (10ms) TPCTO: 10
Checking:       transparent char timeout enable TPCTOEN: 1
Setting:        transparent char timeout enable TPCTOEN: 1
Checking:       transparent escape timeout (10ms) TPESCTO: 75
Setting:        transparent escape timeout (10ms) TPESCTO: 75
Checking:       transparent escape timeout enable TPESCTOEN: 1
Setting:        transparent escape timeout enable TPESCTOEN: 1
Checking:       transparent request ack enable TPACK: 0
Setting:        transparent request ack enable TPACK: 0
Checking:       transparent display ack enable TPACKDSP: 0
Setting:        transparent display ack enable TPACKDSP: 0
Checking:       transparent display nak enable TPNAKDSP: 0
Setting:        transparent display nak enable TPNAKDSP: 0
Checking:       ber pattern BERPAT: RAND
Setting:        ber pattern BERPAT: RAND
Checking:       ber length (bytes) BERLEN: x0800
Setting:        ber length (bytes) BERLEN: x0800

Ensuring there is no open SSM channel before attempting any channel activity...
No previously open channel.
Attempting SSM discovery:

Found SSM 238669

Discovery complete
Waiting for channel cooldown...

Opening a channel to the SSM 238669...
        channel requested (with ssm 238669)...GRANTED
.       RBR SSM-1000 1.500 238669
Opened channel to SSM 238669.
SSM firmware version 1.500 >= 1.300.
.       commands now unlocked
Checking SSM configuration...
Checking: .     baud rate BAUD: 19200
Setting:  .     baud rate BAUD: 19200
Checking: .     sleep timeout (10ms) ZTO: 1000
Setting:  .     sleep timeout (10ms) ZTO: 1000
Checking: .     RBR logger id retrieval enable LIDEN: 0
Setting:  .     RBR logger id retrieval enable LIDEN: 0
Checking: .     transparent max chars before send TPMXC: 512
Setting:  .     transparent max chars before send TPMXC: 512
Checking: .     transparent send char TPSNDC: x0A
Setting:  .     transparent send char TPSNDC: x0A
Checking: .     transparent send char enable TPSNDEN: 1
Setting:  .     transparent send char enable TPSNDEN: 0
Checking: .     transparent send char as well enable TPSNDC2: 1
Setting:  .     transparent send char as well enable TPSNDC2: 1
```

```
Checking: .     transparent char timeout (10ms) TPCTO: 150
Setting:  .     transparent char timeout (10ms) TPCTO: 10
Checking: .     transparent char timeout enable TPCTOEN: 0
Setting:  .     transparent char timeout enable TPCTOEN: 1
Checking: .     transparent escape timeout (10ms) TPESCTO: 75
Setting:  .     transparent escape timeout (10ms) TPESCTO: 75
Checking: .     transparent escape timeout enable TPESCTOEN: 1
Setting:  .     transparent escape timeout enable TPESCTOEN: 1
Checking: .     transparent request ack enable TPACK: 0
Setting:  .     transparent request ack enable TPACK: 0
Checking: .     transparent display ack enable TPACKDSP: 0
Setting:  .     transparent display ack enable TPACKDSP: 0
Checking: .     transparent display nak enable TPNAKDSP: 0
Setting:  .     transparent display nak enable TPNAKDSP: 0
Checking: .     ber pattern BERPAT: RAND
Setting:  .     ber pattern BERPAT: RAND
Checking: .     ber length (bytes) BERLEN: x0800
Setting:  .     ber length (bytes) BERLEN: x0800
Performing a bit error test to assess loop quality...
        (tx) sending ber msg...
.ber : x00000000 / 00004000
Bit error test passed.
.       commands now locked
        channel closed
Waiting for channel cooldown... commands now locked
MLM communication looks good.
Exiting diagnostic task. If you are done conducting all tasks, please reboot controller.
```

Transparent link

If you wish to communicate directly with the HEM you can run the following command

```
RBRcervello> diagnose transparent-link
```

MLM communication principles

Commands

Send each command to the modem on its own line. Lines must be terminated with a carriage return and/or a linefeed character. Most terminal emulators' default handling of the <Enter> key produces acceptable behaviour.

Sleep

If the HEM receives no commands for 10s, it will go to sleep. The HEM will wake up when it next receives a line termination character. This can lead to the modem “ignoring” commands upon first connection, or after having sat idle. To ensure the modem evaluates your command, it is good practice to send a line termination (hit the **Enter** key) once or twice before each command to both wakes up the modem and to cause it to stay awake while you're typing.

The instrument also has a 10-second sleep timeout. When communicating with the instrument over the inductive modem link, the same technique of sending a line termination character before each command should also be followed. Moreover, there is a 10-second inactivity timeout for the HEM connection to the instrument; if no data passes

across the inductive modem link for ten seconds, the connection is closed. To avoid having to frequently reopen the channel, periodically send a line termination character.

Initial state

If the loggercontroller service was running before the connection was opened, the HEM may still be in “transparent mode” – received commands will be passed directly to the attached instrument. To determine the state of the connection, send a line termination character once or twice to wake the HEM from sleep, then send the `a` command (by typing “a<Enter>”). The unit should respond

```
RBR HEM-1000 1.300 999999
```

If you instead receive something to the effect of

```
RBR RBRconcerto 1.410 999999
Ready:
```

then the modem is in “transparent mode” – it is passing everything you type through to the instrument on the other end of the inductive link. This happens when the logger controller has been interrupted. Wait up to 10 seconds for the transparent mode to time out and return you to the HEM command interface:

```
channel closed
***command mode***
```

SSM discovery

The HEM can automatically discover connected SSMs. This is a straightforward, rapid way to verify basic MLM functionality. To invoke it, send the `disc` command to the HEM:

```
disc
    discovery requested
    099999
    discovery complete OK
```

The command will take around 40 seconds to complete, and will list the serial numbers of any connected SSMs as they are discovered. If the SSM is discovered at this step, both the HEM and the SSM are functional. In this case:

- Type Ctrl+C to return to the diagnostic menu.
- Reboot the controller with reboot-controller to start all the services again.

Otherwise, continue on to checking the ferrites.

5.2.2 Ferrites

The ferrite holders (shown at right) hold the ferrite halves and clamp them to the mooring line. If the connection between a ferrite and the modem is faulty, the modem connection will be, at best, unreliable and, at worst, nonfunctional.

The ferrite holders are the first thing to check in the event that the HEM cannot see the SSM, starting with the surface ferrite holder.

Fig. 27. Open and closed ferrite holders

To perform a cursory check of the ferrite

1. Disconnect and reconnect the connectors at both ends of the cable to ensure that they are well-seated.
2. Open the ferrite holder. Check visible edges of the ferrite for damage (scratches, chips). Wiggle/push the ferrite halves to confirm that they are intact, not cracked/split.
3. Clean the ferrite's mating surfaces with a small amount of isopropyl alcohol.
4. Close the ferrite holder again, ensuring that the ends of both ferrite halves are flush and line up with each other.

If other tests point back to a ferrite failure, you can remove the ferrite halves from the enclosure to check them. Do this in a workshop or another safe area where no small parts will be lost during disassembly.

1. Open the ferrite holder.
2. Using a 3mm hex key, remove the retaining screws on the anti-strumming bar on the ferrite holder clasp.
3. Remove the bar.
4. Lift the ferrite half out of the recess, being careful to not lose the T-shaped spring from the bottom of the recess.
5. Check the ferrite half for cracks or fractures.
6. Check that the two O-rings are evenly spaced around the ferrite half, approximately at thirds.
7. Remove the spring from the recess. Make note of the orientation of the spring within the recess before removal: whether the top of the “T” is facing the left or the right side of the clasp.
8. Check the spring for cracks or surface scoring.
9. Insert the spring back into the recess.
10. Push the ferrite half back into the clasp.
11. Replace the anti-strumming bar and tighten its retaining screws.
12. Carefully slide the other ferrite half from the holder body and check it for cracks or fractures. Slide it back into its recess when satisfied.

The orientation of the spring determines the elevation it affords the ferrite half. If the ferrite halves do not compress tightly against each other when the clasp is closed, reorient the spring so the top of the “T” faces the clasp hinge.

Additionally, each ferrite holder contains a coil. With an ohmmeter, check the resistance across the outer two pins of the MCBH-F on a 3-pin version, or between pin 5 and 6 on a 6-pin version connector on the ferrite holder. It should measure 3-4Ohm. Significantly more (hundreds of Ohms or an open circuit) indicates a failure of the coil or the connector.

If everything seems OK, run the MLM diagnostic task again (see above) to verify that the issue went away.

6 Reference

6.1 Deployment calculator

The deployment calculator is available here

6.2 Command references

L3 command reference

RBR*cervata* command reference

6.3 RBRcervata configuration files

RBR*cervata* configuration file (user.json)

RBR instrument configuration file (deployment<n>.json)

6.4 Diagnostic menu

The diagnostic commands use the following formatting conventions:

1. Examples of literal input and output are shown in `bold` type.
2. When an item or group of items is optional, it is enclosed in `[square brackets]`.
3. Where an item can be one of several options, options are separated by vertical bars, `|` .
4. Placeholders for variable fields are in `<italics enclosed in angle brackets>`.

6.4.1 status

Usage

```
status [-h|--help] | [-v|--version]
```

```
status [-p|--progress]
```

```
status [-g|-gps]
```

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.
`-p, --progress`	Shows progress bars for downloads.
`-g, --gps`	Triggers and wait for new GPS satellite fix.

Description

Prints a status overview of the controller including battery information, storage usage, connected instruments status.

General overview

- `Serial number` : serial number of the RBR*cervello*.
- `Part number` : part number of the RBR*cervello*.
- `Cervello firmware version` : firmware version of the RBR*cervello*.
- `Clock` : current clock of the RBR*cervello*.

RBRcervello status

- `Cervello status` : current status of the RBR*cervello*, either **disabled** or **logging**.
- `Cervello internal battery` : RBR*cervello* internal battery voltage in Volts.
- `Cervello external battery` : RBR*cervello* external battery voltage in Volts.

Storage

- `Storage used` : RBR*cervello* memory used in MB.
- `Storage remaining` : RBR*cervello* memory still available in MB.
- `Storage size` : RBR*cervello* memory total size in MB.

Telemetry

- `Telemetry last update` : date/time of the last telemetry attempt, or **N/A**.
- `Telemetry last result` : the result of the last telemetry attempt, either **success**, **failed over X times** or **N/A**.
- `Telemetry last mode` : mode of last telemetry attempt, either **gsm**, **rudics** or **N/A**.

GPS

- `GPS last update` **:** date/time of the last GPS acquisition attempt.
- `GPS last result` **:** the result of the last GPS acquisition attempt, either **fixacquired**, **fixnotacquired** or **N/A**.
- `GPS last known position` **:** the position of the last GPS acquisition attempt, either the latitude and longitude or **N/A**.

Instrument

- `Instrument <XX> last update` : date/time of the last status update the RBR*cervello* performed on the instrument **XX** or **N/A**.
- `Instrument <XX>` : the last known status of the instrument, either:
  - one of the natural statuses of an RBR instrument: `disabled, pending, logging, gated, stopped, finished, unknown` .
  - one of the natural statuses of an Aquadopp instrument: `stopped, logging, unknown` .
  - **commchannelerror:** the RBR*cervello* is unable to open the communication channel.
  - **loggererror**: the instrument has encountered an error.

    - **misconfigured**: the RBR*cervello* is unable to configure the instrument.
    - **notdiscovered:** the RBR*cervello* did not find the instrument.
    - **error**
    - **N/A**.
- `Instrument <XX> internal battery` : the last known internal battery of the instrument **XX** in Volts or **N/A.**
- `Instrument <XX> external battery` : the last known external battery of the instrument **XX** in Volts or **N/A**.

Download

- `Download remaining` : the total number of bytes remaining to be downloaded over all the instruments.

If the RBR*cervello* is misconfigured, only the general overview will be displayed.

Examples

```
RBRcervello> status

RBRcervello
Serial number: 208685
Part number: 0011493
Cervello firmware version: 1.16.0

Clock: 2021-10-29 02:24:35

Cervello status: disabled
Cervello internal battery: 13.68 V
Cervello external battery: 1.31 V

Storage used     : 168 MB
Storage remaining: 14355 MB
Storage size     : 15299 MB

Telemetry last update: 2021-10-29 02:00:09
Telemetry last result: success
Telemetry last mode  : rudics

GPS last update        : 2021-10-29 02:17:59
GPS last result        : fixacquired
GPS last known position: 45.349128 -75.917473

Instrument 01 last update     : 2021-10-29 02:24:06
Instrument 01                 : disabled
Instrument 01 internal battery: 5.11 V
Instrument 01 external battery: 0.00 V

Instrument 02 last update     : 2021-10-29 02:23:37
Instrument 02                 : stopped
Instrument 02 internal battery: 4.74 V
Instrument 02 external battery: 0.00 V

Instrument 03 last update     : 2021-10-29 02:20:45
Instrument 03                 : stopped
Instrument 03 internal battery: 4.69 V
Instrument 03 external battery: 0.00 V

Instrument 04 last update     : 2021-10-29 02:23:02
Instrument 04                 : stopped
Instrument 04 internal battery: 4.90 V
Instrument 04 external battery: 0.00 V

Download remaining: 0B
```

6.4.2 disable

Usage

```
disable  [-h|--help] | [-v|--version]
```

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.

Description

Disable deployment, disable instruments and download their data. The RBR*cervello* will remain disabled upon power cycling.

Examples

```
RBRcervello> disable
Deployment disabled
```

6.4.3 diagnose

Usage

```
diagnose [-h|--help] | [-v|--version]
```

```
diagnose <command>
```

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.

Commands

`mlm`	Diagnoses MLM communication.
`gsm`	Diagnoses GSM telemetry.
`rudics`	Diagnoses Iridium RUDICS telemetry.
`calibrate-airmar-compass`	Runs the compass calibration.

Description

Performs in-depth tests for telemetry and MLM links.

`mlm` Ensures correct HEM and SSM configuration on every connected device.

`gsm` Ensures connectivity through GSM to the Internet.

`rudics` Ensures connectivity through Iridium RUDICS to the Internet.

`calibrate-airmar-compass` Ensures the compass is calibrated in the current environment.

Examples

```
RBRcervello> diagnose rudics
Stopping services...
Waiting for publisher to terminate...
Stopping periodic command scheduler: cron.
Starting RUDICS test
I think the RUDICS modem is on /dev/ttyS4 at 2400 baud.
Waiting for the modem to come up...
Teraform v1.1.2 (built 2021-10-27T13:30:04+0000)
2021-10-29 05:55:19   src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-10-29 05:55:19   src/teraform.c:367 [INFO ]: Dial number: "0088160000500"
2021-10-29 05:55:19   src/teraform.c:388 [INFO ]: Configuring modem...
2021-10-29 05:55:19   src/teraform.c:432 [INFO ]: Waiting for modem to be available...
2021-10-29 05:55:19   src/teraform.c:451 [INFO ]: Checking IMEI...
2021-10-29 05:55:20   src/teraform.c:451 [INFO ]: IMEI: 300125061274520
2021-10-29 05:55:20   src/teraform.c:478 [INFO ]: Checking ICCID...
2021-10-29 05:55:20   src/teraform.c:478 [INFO ]: ICCID: 8988169763060305628
2021-10-29 05:55:20   src/teraform.c:488 [INFO ]: Disabling flow control...
2021-10-29 05:55:20   src/teraform.c:520 [INFO ]: Waiting for satellite signal...
2021-10-29 05:55:26   src/teraform.c:520 [INFO ]: Found signal of strength 5/5.
2021-10-29 05:55:26   src/teraform.c:530 [INFO ]: Configuring data connection...
2021-10-29 05:55:26   src/teraform.c:564 [INFO ]: Dialing data connection...
2021-10-29 05:55:34   src/teraform.c:564 [INFO ]: Connection dialed.
The modem connection is up.
Brought the rudics connection up successfully.
Teraform v1.1.2 (built 2021-10-27T13:30:04+0000)
2021-10-29 05:55:34   src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-10-29 05:55:37   src/teraform.c:584 [INFO ]: Escaping from data call...
2021-10-29 05:55:38   src/teraform.c:594 [INFO ]: Disconnecting data call...
The serial link has been brought down.
Powering off the modem.
Starting periodic command scheduler: cron.
RBRcervello>
```

```
RBRcervello> diagnose gsm
Stopping services...
Waiting for publisher to terminate...
Stopping periodic command scheduler: cron.
Starting GSM test
I think the GSM modem is on /dev/ttyS4 at 230400 baud.
Waiting for the modem to come up...
Modem is up.
Waiting for PPP connection to the modem...
Established PPP connection to the modem. Modem identifiers:
IMEI: 867698040840616
IMSI: 302720511124309
ICCID: 89302720523086286725
Waiting for the modem to connect to the GSM network..
Reconnecting for Internet access (1/2)...
    Bringing PPP connection down...
    Bringing PPP connection back up...
    Waiting for the PPP connection to become the default route
Making sure we can access the web...
The modem connection is up.
Brought the gsm connection up successfully.
/usr/local/wirewalker-controller/diagnostics/../publisher/modes/gsm/netpty/bin/netpty:
error waiting for fds: Interruptl
messages-read: no process found
The serial link has been brought down.
Powering off the modem.
Starting periodic command scheduler: cron.
RBRcervello>
```

```
RBRcervello> diagnose mlm
Stopping services...
Waiting for publisher to terminate...
Stopping periodic command scheduler: cron.
Starting MLM test
I think the HEM is on /dev/ttyS3 at 115200 baud.
Trying to wake the HEM...
***command mode***
        RBR HEM-1000 1.300 208882
Found HEM 208882.
HEM firmware version 1.300 >= 1.300.
        commands now locked
Checking HEM configuration...
Checking:       commands now unlocked
        baud rate BAUD: 115200
Setting:        baud rate BAUD: 115200
Checking:       sleep timeout (10ms) ZTO: 1000
Setting:        sleep timeout (10ms) ZTO: 1000
Checking:       hardware flow control enable HWF: 0
Setting:        hardware flow control enable HWF: 0
```

```
Checking:       poll enable PEN: 0
Setting:        poll enable PEN: 0
Checking:       poll interval time (min) PINT: 5
Setting:        poll interval time (min) PINT: 5
Checking:       poll delay time (10ms) PDEL: 500
Setting:        poll delay time (10ms) PDEL: 500
Checking:       poll display enable PDISP: 0
Setting:        poll display enable PDISP: 0
Checking:       poll cmd string PCMD: x46,x30,x30
Setting:        poll cmd string PCMD: x46,x30,x30
Checking:       poll wakeup char enable PWUPEN: 1
Setting:        poll wakeup char enable PWUPEN: 1
Checking:       poll wakeup char PWUPCHAR: x0F
Setting:        poll wakeup char PWUPCHAR: x0F
Checking:       poll sleep enable (zzz) PSLPEN: 1
Setting:        poll sleep enable (zzz) PSLPEN: 1
Checking:       poll address PADD: 00
Setting:        poll address PADD: 00
Checking:       modem open channel timeout (10ms) MDOCDUR: 1000
Setting:        modem open channel timeout (10ms) MDOCDUR: 1000
Checking:       modem open channel permanent MDOCPERM: 0
Setting:        modem open channel permanent MDOCPERM: 0
Checking:       modem maximum retry count MDRTRY: 1
Setting:        modem maximum retry count MDRTRY: 1
Checking:       transparent max chars before send TPMXC: 256
Setting:        transparent max chars before send TPMXC: 256
Checking:       transparent send char TPSNDC: x0A
Setting:        transparent send char TPSNDC: x0A
Checking:       transparent send char enable TPSNDEN: 1
Setting:        transparent send char enable TPSNDEN: 1
Checking:       transparent send char as well enable TPSNDC2: 1
Setting:        transparent send char as well enable TPSNDC2: 1
Checking:       transparent char timeout (10ms) TPCTO: 20
Setting:        transparent char timeout (10ms) TPCTO: 20
Checking:       transparent char timeout enable TPCTOEN: 1
Setting:        transparent char timeout enable TPCTOEN: 1
Checking:       transparent escape timeout (10ms) TPESCTO: 75
Setting:        transparent escape timeout (10ms) TPESCTO: 75
Checking:       transparent escape timeout enable TPESCTOEN: 1
Setting:        transparent escape timeout enable TPESCTOEN: 1
Checking:       transparent request ack enable TPACK: 0
Setting:        transparent request ack enable TPACK: 0
Checking:       transparent display ack enable TPACKDSP: 0
Setting:        transparent display ack enable TPACKDSP: 0
Checking:       transparent display nak enable TPNAKDSP: 0
Setting:        transparent display nak enable TPNAKDSP: 0
Checking:       ber pattern BERPAT: RAND
Setting:        ber pattern BERPAT: RAND
Checking:       ber length (bytes) BERLEN: x0800
Setting:        ber length (bytes) BERLEN: x0800
Ensuring there is no open SSM channel before attempting any channel activity...
        channel closed
```

```
Channel closed successfully.
Attempting SSM discovery:
        discovery requested
        208088
Found SSM 208088.
        208090
Found SSM 208090.
        discovery complete OK
Waiting for channel cooldown...
Opening a channel to the SSM...
        channel requested (with ssm 208088)...GRANTED
.       RBR SSM-1000 1.300 208088
Opened channel to SSM 208088.
SSM firmware version 1.300 >= 1.300.
.       commands now unlocked
Checking SSM configuration...
Checking: .     baud rate BAUD: 19200
Setting:  .     baud rate BAUD: 19200
Checking: .     sleep timeout (10ms) ZTO: 1000
Setting:  .     sleep timeout (10ms) ZTO: 1000
Checking: .     RBR logger id retrieval enable LIDEN: 0
Setting:  .     RBR logger id retrieval enable LIDEN: 0
Checking: .     transparent max chars before send TPMXC: 512
Setting:  .     transparent max chars before send TPMXC: 512
Checking: .     transparent send char TPSNDC: x0A
Setting:  .     transparent send char TPSNDC: x0A
Checking: .     transparent send char enable TPSNDEN: 0
Setting:  .     transparent send char enable TPSNDEN: 0
Checking: .     transparent send char as well enable TPSNDC2: 1
Setting:  .     transparent send char as well enable TPSNDC2: 1
Checking: .     transparent char timeout (10ms) TPCTO: 20
Setting:  .     transparent char timeout (10ms) TPCTO: 20
Checking: .     transparent char timeout enable TPCTOEN: 1
Setting:  .     transparent char timeout enable TPCTOEN: 1
Checking: .     transparent escape timeout (10ms) TPESCTO: 75
Setting:  .     transparent escape timeout (10ms) TPESCTO: 75
Checking: .     transparent escape timeout enable TPESCTOEN: 1
Setting:  .     transparent escape timeout enable TPESCTOEN: 1
Checking: .     transparent request ack enable TPACK: 0
Setting:  .     transparent request ack enable TPACK: 0
Checking: .     transparent display ack enable TPACKDSP: 0
Setting:  .     transparent display ack enable TPACKDSP: 0
Checking: .     transparent display nak enable TPNAKDSP: 0
Setting:  .     transparent display nak enable TPNAKDSP: 0
Checking: .     ber pattern BERPAT: RAND
Setting:  .     ber pattern BERPAT: RAND
Checking: .     ber length (bytes) BERLEN: x0800
Setting:  .     ber length (bytes) BERLEN: x0800
Performing a bit error test to assess loop quality...
        (tx) sending ber msg...
.ber : x00000000 / 00004000
Bit error test passed.
```

```
.       commands now locked
        channel closed
Waiting for channel cooldown...
Opening a channel to the SSM...
        channel requested (with ssm 208090)...GRANTED
.       RBR SSM-1000 1.300 208090
Opened channel to SSM 208090.
SSM firmware version 1.300 >= 1.300.
.       commands now unlocked
Checking SSM configuration...
Checking: .     baud rate BAUD: 19200
Setting:  .     baud rate BAUD: 19200
Checking: .     sleep timeout (10ms) ZTO: 1000
Setting:  .     sleep timeout (10ms) ZTO: 1000
Checking: .     RBR logger id retrieval enable LIDEN: 0
Setting:  .     RBR logger id retrieval enable LIDEN: 0
Checking: .     transparent max chars before send TPMXC: 512
Setting:  .     transparent max chars before send TPMXC: 512
Checking: .     transparent send char TPSNDC: x0A
Setting:  .     transparent send char TPSNDC: x0A
Checking: .     transparent send char enable TPSNDEN: 0
Setting:  .     transparent send char enable TPSNDEN: 0
Checking: .     transparent send char as well enable TPSNDC2: 1
Setting:  .     transparent send char as well enable TPSNDC2: 1
Checking: .     transparent char timeout (10ms) TPCTO: 20
Setting:  .     transparent char timeout (10ms) TPCTO: 20
Checking: .     transparent char timeout enable TPCTOEN: 1
Setting:  .     transparent char timeout enable TPCTOEN: 1
Checking: .     transparent escape timeout (10ms) TPESCTO: 75
Setting:  .     transparent escape timeout (10ms) TPESCTO: 75
Checking: .     transparent escape timeout enable TPESCTOEN: 1
Setting:  .     transparent escape timeout enable TPESCTOEN: 1
Checking: .     transparent request ack enable TPACK: 0
Setting:  .     transparent request ack enable TPACK: 0
Checking: .     transparent display ack enable TPACKDSP: 0
Setting:  .     transparent display ack enable TPACKDSP: 0
Checking: .     transparent display nak enable TPNAKDSP: 0
Setting:  .     transparent display nak enable TPNAKDSP: 0
Checking: .     ber pattern BERPAT: RAND
Setting:  .     ber pattern BERPAT: RAND
Checking: .     ber length (bytes) BERLEN: x0800
Setting:  .     ber length (bytes) BERLEN: x0800
Performing a bit error test to assess loop quality...
        (tx) sending ber msg...
.ber : x00000000 / 00004000
Bit error test passed.
.       commands now locked
        channel closed
Waiting for channel cooldown... commands now locked
MLM communication looks good.
Starting periodic command scheduler: cron.
RBRcervello>
```

```
RBRcervello> diagnose calibrate-airmar-compass 2
Stopping services...
Waiting for publisher to terminate...
Stopping periodic command scheduler: cron.
Are you sure you want to begin a compass calibration sequence on port 2? (y/N) y
Powering on port 2...
Waiting for the station to boot up
Station ready!
Temporarily stopping station streaming...
Ready to start the calibration sequence!
=====================================================
The calibration sequence requires at least one full
slow circular turn of the station within 2-3 minutes
of starting the sequence.
=====================================================
Are you ready to start the compass calibration sequence? (y/N) y
Starting the calibration sequence!
=====================================================
Please complete at least one SLOW circular turn of
the station now.
=====================================================
Waiting for the calibration to complete
Calibration successful!
RBRcervello>
```

6.4.4 config-check

Usage

`config-check [-h|--help] | [-v|-version]`

`config-check [-c|--config-file <device>]`

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.
`-c, --config-file <device>`	Shows configuration file for a supported device.
`-t, --validate`	Validates configuration files for all supported device.

The supported devices are the following:

- `instrument<XX>` displays the corresponding deployment.json configuration file of an instrument, where **XX** is the instrument number.
- `cervello` displays the user configuration of the RBR*cervello*.

Description

Checks RBR configuration. Displays in an interactive manner the current configuration of the RBR*cervello* or one of the configured instruments. Uses arrow keys to scroll the configuration; **Q** to exit.

Examples

```
RBRcervello> config-check -c cervello
Showing configuration file of cervello.
{
    "file": {
        "date": "20210814000000",
        "version": "3.0.0",
        "schema": "0011713",
        "customer": "RBR"
     },
    "controller": {
        "battery_period": 600,
        "deep_sleep": {
            "enabled": false,
            "deep_sleep_duration": 3600
        },
        "clock_sync_timeout": 43200
    },
    "gps": {
        "enabled": true,
        "period": 600,
        "fix_timeout": 60,
        "hdop_threshold": 1.5
    ...
    "telemetry": {
        "enabled": true,
        "force_period": 3600,
        "mode": ["gsm", "rudics"],
        "gsm": {
            "failures_before_unavailable": 3,
            "retry_period": 7200,
            "apn": "auto",
            "destination": "dataincoming.rbr-global.com:9140"
        },
        "rudics": {
            "failures_before_unavailable": 3,
            "retry_period": 7200,
            "dial": "00881600005385"
        }
    }
}
```

```
RBRcervello> config-check -t
Validating /etc/rbrcervello/environment.json against /usr/local/rbrcervello/cervello-
config-tools/schemas/cervello_environment_schema.json
Validating /etc/rbrcervello/user.json against /usr/local/rbrcervello/cervello-config-
tools/schemas/cervello_user_schema.json
Validating /etc/rbrcervello/deployment01.json against /usr/local/rbrcervello/cervello-
config-tools/schemas/cervello_rbr_logger_schema.json
RBR logger has product code L3-M14-F15-BEC12-INT11-SCT13-SP11
Validating 'RBRconcerto³ C.T.D|fast32|deep' (type 'RBR') against '/etc/rbrcervello/
deployment01.json'
Validating subset 'raw'
Validation succeeded!
```

6.4.5 edit-config

Usage

`edit-config [-h|--help] | [-v|--version]`

`edit-config [-n|-new-file] | [-c|--config-file <device>]`

Options

`-h, --help`	Displays this help message and exit.
`-v, --version`	Outputs version information and exit.
`-n, --new-file`	Starts editing from an empty file.
`-c, --config-file <device>`	Shows configuration for a supported device.

The supported devices are the following:

- `instrument<XX>` allows editing the deployment configuration file of an instrument, where **XX** is the instrument number.
- `cervello` allows editing the RBR*cervello* user configuration file.

Description

Interactively edits the current configuration of a specific instrument or the RBR*cervello* with GNU nano.

Before applying the configuration, `edit-config` checks the configuration using the mechanism described in `config-check`.

See “Updating the instrument” for more information

Examples

```
RBRcervello> edit-config -c instrument01
Checking deployment status...
Modifying configuration file of instrument01.

Stopping services...

Are you sure you want to save? (y/N)y

RBRcervello configuration update process has started...


RBRcervello configuration update succeeded!


You should reboot to make sure everything is set properly.
Starting services...
```

6.4.6 set-cervello-clock

Usage

`set-cervello-clock [-h|--help] | [-v|--version]`

`set-cervello-clock <string-formatted-date>`

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.

Commands

`string-formatted-date`	Inputs new internal clock time, saves it, and exits. Format: *yyyy-mm-ddThh:mm:ss.*

Description

Sets the RBR*cervello* internal clock and saves the previous and new times in every RSK file involved, the RBR*cervello*, and the instrument.

This command is intended to be used on units with no GPS.

Examples

```
RBRcervello> set-cervello-clock '2023-12-25T12:34:56'
Previous internal time: '2014-12-25T10:12:00'
New internal time: '2023-12-25T12:34:56'
```

6.4.7 cervello-clock

Usage

`cervello-clock [-h|--help] | [-v|-version]`

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.

Description

Displays the current clock of the RBR*cervello* and its last manually triggered clock synchronization.

- `Last time sync:` displays the last manually triggered clock synchronization date/time. This does not reflect GPS clock resynchronization time.
- `Current internal time:` displays the current clock of the RBR*cervello*.

Examples

```
RBRcervello> cervello-clock
Last time sync: '2023-06-28 15:31:00'
Current internal time: '2023-06-29 11:54:24'
```

6.4.8 reboot-controller

Usage

`reboot-controller [-h|--help] | [-v|-version]`

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.

Description

Stops all internal services and reboots the RBR*cervello* as if it entered deep sleep and woke up. Useful for testing deployment behaviour.

Examples

```
RBRcervello> reboot-controller
...
The system is going down for reboot NOW!
...
```

6.4.9 scrub-controller

Usage

`scrub-controller [-h|--help] | [-v|-version]`

`scrub-controller [-f|-format]`

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.
`-f, --format`	Formats the SD card in the proper ext4 filesystem and initializes the folder structure.

Description

Clears all data stored in the RBR*cervello* storage. This command is unavailable if the RBR*cervello* status is "`logging`".

Examples

```
RBRcervello> scrub-controller
Are you sure you want to scrub the controller? (y/N)y
Removing files from
    /mnt/sdcard/data...
    /mnt/sdcard/logs...
    /mnt/sdcard/lost+found...
    /mnt/sdcard/messages...
    /mnt/sdcard/publisher...
    /mnt/sdcard/queues...
    /mnt/sdcard/responses...
    /mnt/sdcard/tasks...
```

6.4.10 scrub-instrument

Usage

```
scrub-instrument [-h|--help] | [-v|-version]
scrub-instrument [-a|-all]
scrub-instrument <command>
```

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.
`-a, -all`	Erases all instruments.

Commands

`instrument<XX>`	Erases instrument XX.

Description

Clears the internal memory of an instrument connected to the system. This command is unavailable if the RBR*cervello* status is logging. By default, it erases all instruments.

Examples

```
RBRcervello> scrub-instrument
Are you sure you want to scrub instrument01? (y/N) y
Checking deployment status...
Clearing memory...
Instrument 01: success
```

When the instrument number is not entered, the command defaults to instrument 01.

```
RBRcervello> scrub-instrument instrument02
Are you sure you want to scrub instrument02? (y/N) y
Checking deployment status...
Clearing memory...
Instrument 02: success
```

```
RBRcervello> scrub-instrument -a
Are you sure you want to scrub ALL instruments? (y/N) y
Checking deployment status...
Clearing memory...
Instrument 01: success
Instrument 02: success
```

6.4.11 enable

Usage

`enable [-h|--help] | [-v|-version]`

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.

Description

Enables deployments according to internal configuration files. Configures instruments and sets them up to start logging. Deletes any stopping flag for the deployment; whenever the system reboots, the deployment behaviour will continue.

Examples

```
RBRcervello> enable
Deployment enabled
```

6.4.12 help

Usage

`help`

Description

Displays the RBR*cervello* diagnostic `help` menu as shown on wake up.

Examples

```
RBRcervello> help
RBRcervello Menu.

These are the available commands to manage RBRcervello

   disable               Disable deployment.
   status                Provide a general overview or detailed information on
                         controller and instruments.
   diagnose              Diagnostics for MLM and telemetry.
   transparent-link      Establish a transparent link to an instrument
                         serial port.
   config-check          Print the current configuration of a specific RBR
                         subsystem.
   edit-config           Edit the current configuration of a specific
                         RBR subsystem.
   set-instrument-clock  RBR instrument clock sync.
   set-cervello-clock    RBRcervello clock setter.
   cervello-clock        RBRcervello clock checker.
   scrub-controller      Clear all data stored in the SD card.
   scrub-instrument      RBR instrument scrubber.
   enable                Enable deployment according to internal configuration
                         files.
   reboot-controller     Reboot system as if it had entered deep-sleep and
                         waken up.
   help                  Show this menu.
```

6.4.13 transparent-link

Usage

`transparent-link [-h|--help] | [-v|-version]`

`transparent-link [-p|--port] <port>`

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.
`-p, --port <port>`	Selects a specific port number to interact with. Defaults to 1.

Description

Stops all services permanently and open a transparent link with the chosen serial port.

Examples

```
BRcervello> transparent-link
Checking deployment status...
Stopping services...
Stopping periodic command scheduler: cron.
Waiting for publisher to terminate...
Connected. Press CTRL-C to exit when done.
```

When the port number is not entered, the command defaults to Port 1.

```
RBRcervello> transparent-link -p 2
Checking deployment status...
Stopping services...
Stopping periodic command scheduler: cron failed!
Waiting for publisher to terminate...
Connected. Press CTRL-C to exit when done.
```

6.4.14 mount-storage

Usage

`mount-storage [-h|--help] | [-v|-version]`

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.

Description

Mounts the data storage on the computer host. Stops all services and exposes the internal drives to the host computer.

Examples

```
BRcervello> RBRcervello> mount-storage
Checking deployment status...
Stopping services...
Mounting data partition as read-only
```

6.4.15 unmount-storage

Usage

```
unmount-storage [-h|--help] | [-v|-version]
```

Options

`-h, --help`	Displays this help message and exits.
`-v, --version`	Outputs version information and exits.

Description

Restarts internal services and unmounts the data storage from the computer. When an update package is detected, it will initialize the update process.

Examples

```
BRcervello> RBRcervello> unmount-storage
Mounting data partition as read-only
Restarting services...
Unmounting data from computer
```

7 Revision history

Revision No.	Release date	Notes
A	14-May-2025	Initial release