RBR*cervello*

INSTRUMENT GUIDE

rbr-global.com

Table of Contents

1 Introduction . . . 4

2 Hardware . . . 5

2.1 Opening and closing the RBRcervello . . . 5

2.2 Block diagram . . . 6

3 General maintenance . . . 7

3.1 Support kit . . . 7

3.2 Replacing batteries . . . 7

3.3 Replacing desiccant . . . 8

3.4 Cables and connectors . . . 9

4 Getting started . . . 10

4.1 Using the USB port . . . 10

4.2 Telemetry . . . 14

4.3 Retrieving data . . . 19

4.4 Updating the instrument . . . 20

4.5 Deployment preparation . . . 23

5 Features . . . 28

5.1 Data hosting . . . 28

5.1.1 Remote Diagnostics . . . 30

5.2 Weather station (optional) . . . 31

5.3 Optical Modem (optional) . . . 32

6 Troubleshooting . . . 33

6.1 LTE/GSM modem troubleshooting . . . 33

6.2 RUDICS modem troubleshooting . . . 35

6.3 MLM troubleshooting . . . 37

6.4 Airmar weather station troubleshooting . . . 40

6.5 Optical modem troubleshooting . . . 42

7 Reference . . . 43

7.1 Deployment behaviour . . . 43

7.1.1 Behaviour with deep-sleep disabled . . . 43

7.1.2 Behaviour with deep-sleep enabled . . . 44

7.2 Diagnostic menu . . . 44

7.2.1 status . . . 45

7.2.2 disable . . . 47

7.2.3 diagnose . . . 47
7.2.4 config-check . . . 52
7.2.5 edit-config . . . 54
7.2.6 set-cervello-clock . . . 55
7.2.7 cervello-clock . . . 55
7.2.8 reboot-controller . . . 56
7.2.9 scrub-controller . . . 56
7.2.10 scrub-instrument . . . 57
7.2.11 enable . . . 58
7.2.12 help . . . 58
7.2.13 transparent-link . . . 59
7.2.14 mount-storage . . . 59
7.2.15 unmount-storage . . . 60
8 Repairs . . . 61
9 Revision history . . . 62

1 Introduction

The RBR*cervello* is a general-purpose controller designed for use in surface installations. Typically connected to one or more RBR instruments using an inductive modem, it is responsible for archiving logged data locally onto an SD card, as well as optionally uploading that data to the RBR data hosting service via Iridium Satellite or GSM cellular modems.

The most common implementation is with the Del Mar Oceanographic (DMO) Wirewalker™, a wave-powered wire-guided profiling platform manufactured by DMO. This platform can host a variety of instruments, including the RBR*concerto3* and RBR*maestro3*. Longer autonomy deployments can be obtained with the RBR*fermata* underwater battery canisters, which typically extend deployment lifetimes by a factor of forty. The inductive modem uses the jacketed steel mooring line of the Wirewalker to communicate to the surface.

The RBR*cervello* controllers have undergone several modifications over the years, progressing from single-modem versions to dual-modem, and eventually incorporating the deep-sleep feature.

- RBR*cervello*, single modem (GSM)
- RBR*cervello*, single modem (RUDICS)
- RBR*cervello*, dual modem (GSM/RUDICS), end-cap with two wired connectors, built-in antennas
- RBR*cervello* Antarctica, dual modem (GSM/RUDICS), deep-sleep feature, end-cap with four wired connectors, external RUDICS and GPS antennas
- RBR*cervello*, dual modem (GSM/RUDICS), deep-sleep feature, end-cap with two or four wired connectors, built-in antennas
- RBR*cervello*, dual modem (GSM/RUDICS), deep-sleep feature, end-cap with four wired connectors, built-in low-profile antennas

This documentation applies to the RBR*cervello* with firmware 1.18.0 and up.

The RBR*cervello,* dual modem, is a version that supports both LTE/GSM and RUDICS. It includes the deep-sleep feature, an end-cap with locking tabs, four wired connectors, and built-in RUDICS, LTE/GSM, and GPS antennas. The RBR*cervello* is perfect for mounting on a buoy.

Fig. 1. RBR cervello, dual modem, with four wired connectors

2 Hardware

2.1 Opening and closing the RBRcervello

Opening the controller

1. Position the RBR*cervello* upright.
2. Locate and grab the locking tabs on the cap.
3. Pull the tabs to their sides.
4. Pull the internal assembly from the external handle.

Closing the controller

1. Position the RBR*cervello* housing upright.
2. Insert the internal assembly into the housing.
3. Align the locking tabs with the openings on the top of the housing.
4. Press firmly.
5. Lock the tabs back in place.

2.2 Block diagram

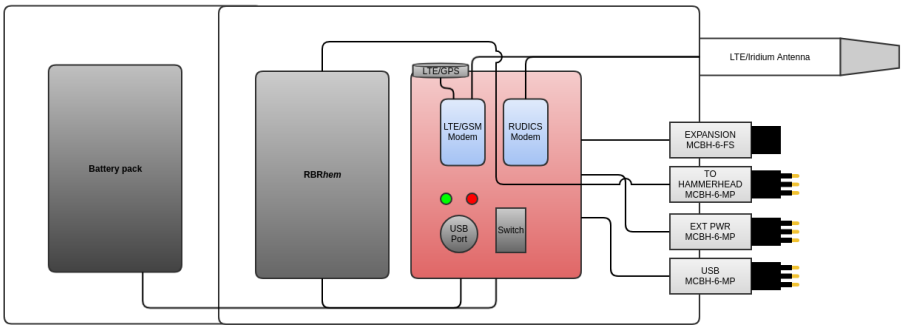


Fig. 2. Block diagram of the RBRcervello with an RBR mooring line modem

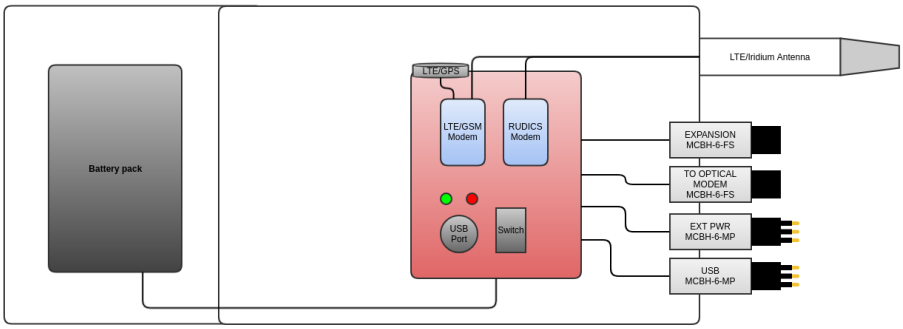


Fig. 3. Block diagram of the RBRcervello with an RBR optical modem

3 General maintenance

3.1 Support kit

RBR provides one support kit and one cable kit for every RBR*cervello* ordered.

The RBR*cervello* support kit contains an assortment of basic accessories and spare parts, as presented below. If you need more units, contact RBR.

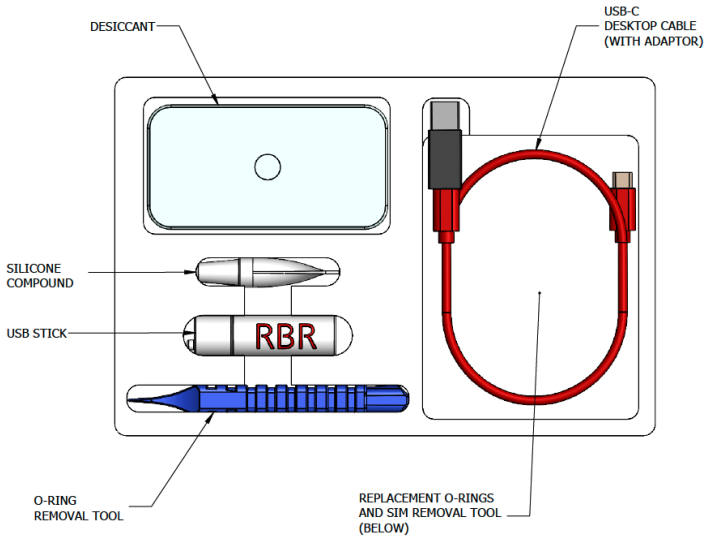


Fig. 4. RBR cervello support kit diagram

The RBR*cervello* cable kit contains the MLM inductive loop cable and MLM inductive test loop termination cable. If you need more units, contact RBR.

Fig. 5. MLM test loop cable

Fig. 6. Termination cable

3.2 Replacing batteries

RBR may ship new RBR*cervello* data controllers with or without lithium thionyl chloride batteries included, depending on your order.

When ordered without the batteries, open your RBR*cervello* and insert the batteries as shown below. Before each deployment, replace the batteries to maximise the operational time and prevent data loss.

To remove the battery pack:

1. Remove the internal assembly from the RBR*cervello*. See Opening the controller.
2. Remove the intermediate foam.
3. Press the switch to the **off** position.
4. Twist the lock counterclockwise.
5. Pull the internal battery pack out and remove it from the internal assembly.

Fig. 7. Battery connector

To install a new battery pack:

1. Insert the new battery pack into the RBR*cervello* internal assembly.
2. Insert the intermediate foam on top of the battery pack.
3. Push the battery pack down, into the connector, and twist the lock clockwise.
4. Press the switch to the "on" position.
5. Insert the internal assembly back in the RBR*cervello* housing. See Closing the controller.

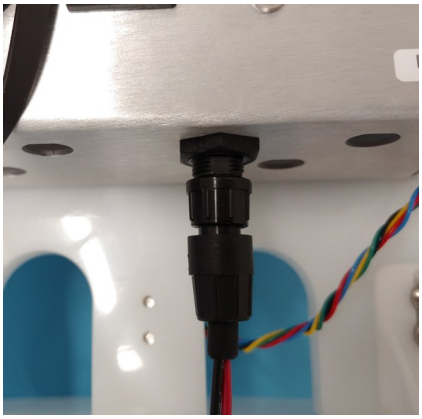


Fig. 8. Lock connector

Fig. 9. Inserted battery and foam

3.3 Replacing desiccant

Replace desiccant before each deployment. One spare desiccant canister is included in the support kit.

Fresh desiccant will keep the RBR*cervello* compartment dry and prevent malfunction. Water damage may occur if condensation forms inside the controller.

As a preventative measure, RBR recommends servicing the instrument in a cool, dry place (when possible).

Replacing the desiccant canister

1. Remove the battery end-cap and pull out the internal assembly. See Opening the controller.
2. Unclip the desiccant canister from the holder.
3. Insert the new desiccant canister into the holder.
4. Insert the internal assembly into the RBR*cervello* housing. See Closing the controller.

Fig. 10. Desiccant canister holder

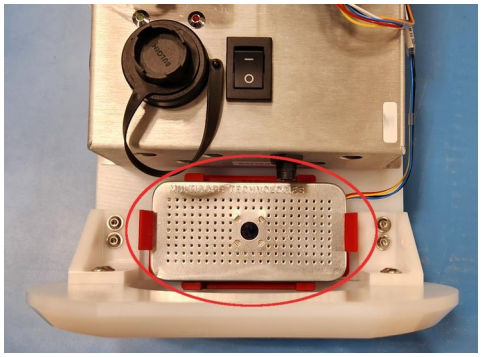


Fig. 11. Desiccant canister fully inserted

3.4 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

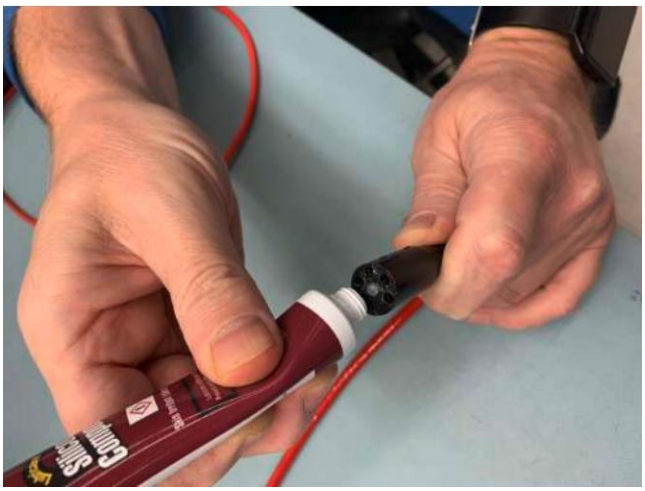


Fig. 12. Applying silicone compound to the female connectors.

Fig. 13. Cross-sectional view of a connector with 30% lubricant fill.

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 Getting started

Follow the steps below to set up your RBR*cervello* and prepare it for deployment.

4.1 Using the USB port

Tools needed

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

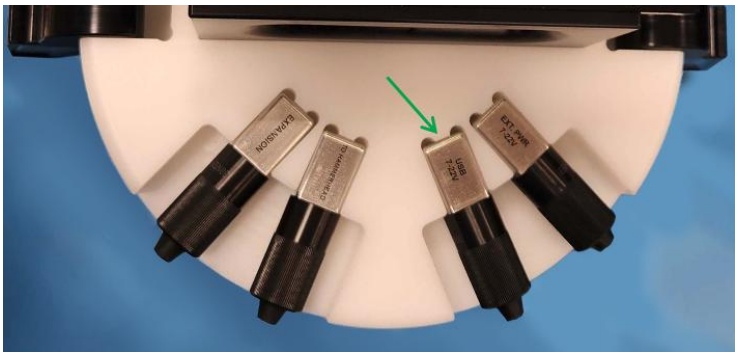
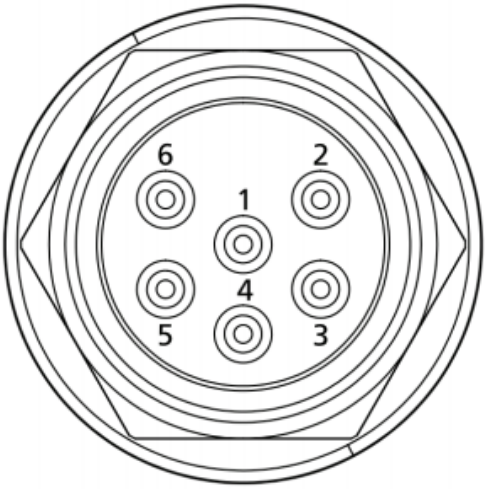


Fig. 14. Location of the USB port

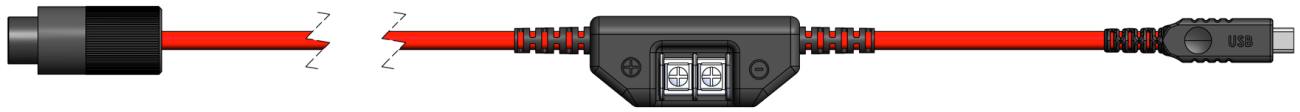
External MCBH-6-MP USB connector pinout



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

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

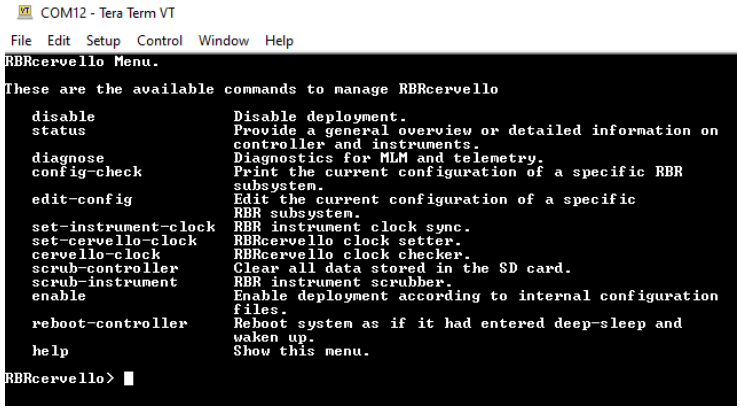
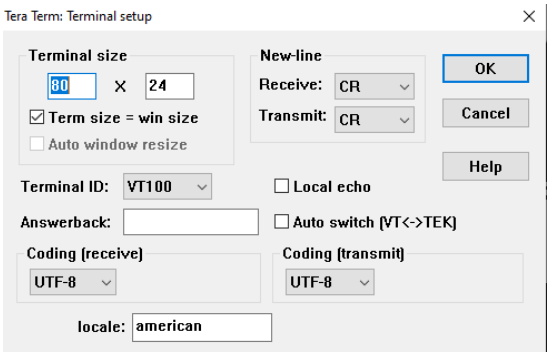


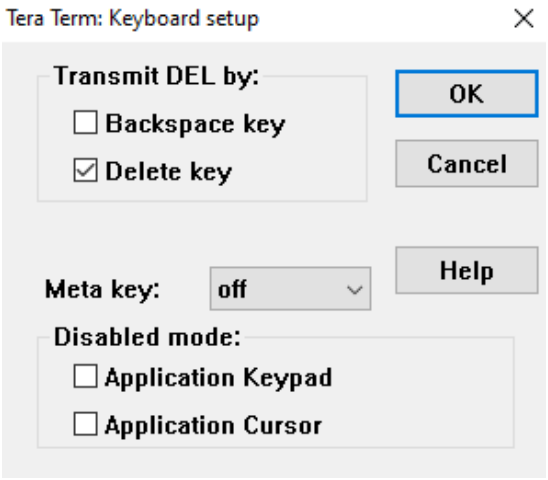
Fig. 15. Tera Term

Alternatively, configure the settings manually as follows

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

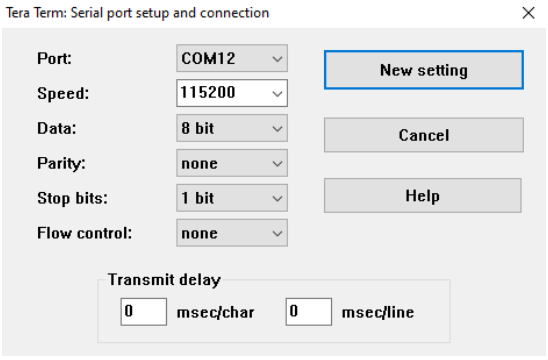


2. Go to **Setup** > **Keyboard setup**, select **Delete key**



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.

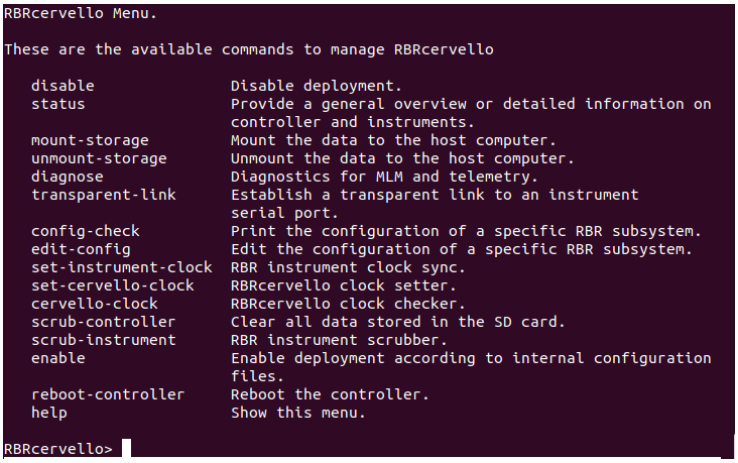


Fig. 16. Minicom

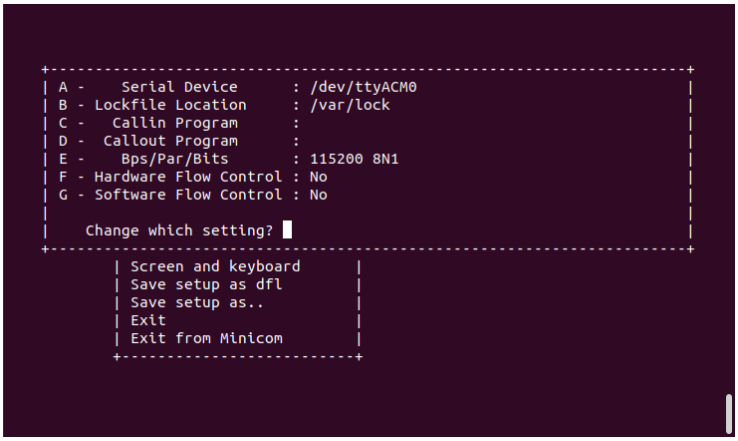


Fig. 17. Minicom serial settings

Diagnostic menu overview

Access the RBR*cervello* menu via the diagnostic port. It is a fully-featured Bash terminal (Restricted Bash) with **tab** completion. To see the list of available commands, execute the `help` command. You will find all you need to deploy the RBR*cervello* with the RBR instruments. As a quick in-menu reference, each command displays a short manual when executed with the `-h` or `--help` argument.

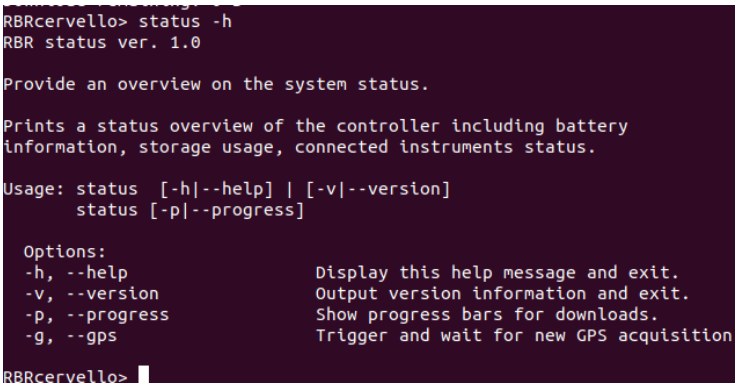


Fig. 18. "status" help

Diagnostic menu commands

```
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.
```

See the Diagnostic menu section for more information.

4.2 Telemetry

All new RBR*cervello* data controllers are equipped with a GSM/RUDICS dual modem.

LTE/GSM

Before LTE/GSM telemetry can function, a SIM-card equipped with a data plan must be installed into the RBR*cervello*.

> The end-user is responsible for provisioning a SIM-card and a data plan for use with the RBR*cervello*. To determine specific data quota requirements for your deployment scenario, please contact RBR.

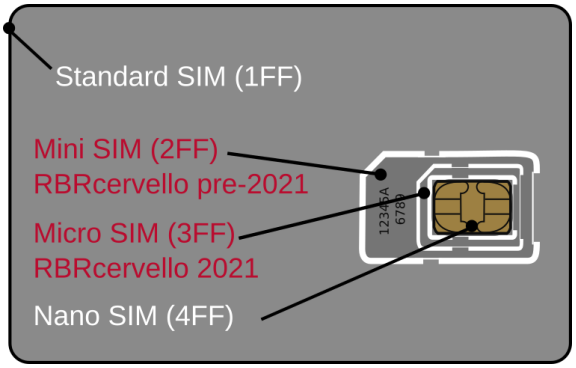


Fig. 19. SIM-card comparison (Wikipedia)

Fig. 20. GSM/RUDICS dual modem

Supported bands

LTE FDD (Bands 1,2,3,4,5,7,8,12,13,18,19,20,26,28)	WCDMA (Bands 1,2,4,5,6,8,19)
LTE TDD (Bands 38,39,40,41)	GSM (Bands 2,3,5,8)

Accessing the LTE/GSM modem

Tools needed

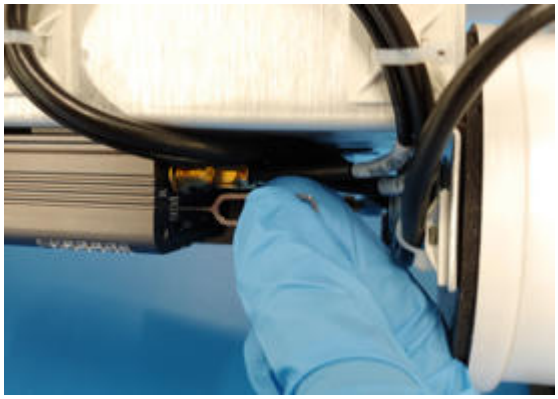
- SIM removal tool (included in the support kit)

Fig. 21. SIM removal tool

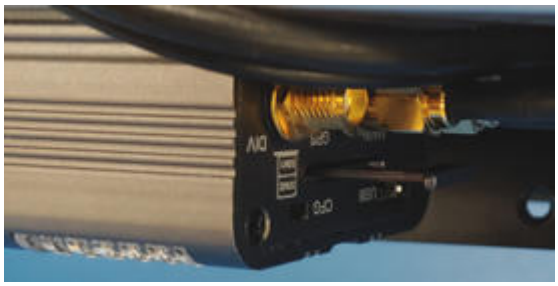
The LTE/GSM modem accepts a micro SIM-card (3FF).

1. Locate the pinhole in the SIM tray.

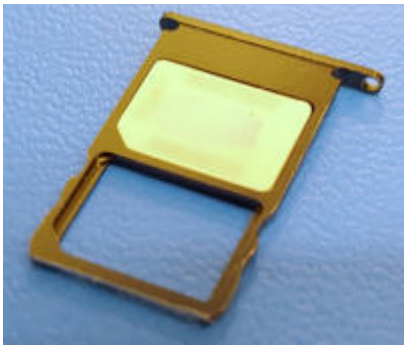
2. Insert the SIM removal tool as straight as possible into the pinhole and push. The tray will pop out.



3. Pull the tray out of the modem.



4. Put the micro SIM-card into the SIM1 position, with the electrical contacts facing down.



5. Insert the tray with the SIM-card installed and its electrical contacts facing downward. Handle it with care to prevent the SIM from falling out of the tray.

6. Push the tray in until flushed with the modem's chassis. Avoid moving the RBR*cervello* until the tray is fully inserted to prevent the SIM from falling out of place.

Most cellular providers require no further configuration. If telemetry difficulties arise, or if you know your provider requires additional APN configuration, please contact RBR.

Iridium RUDICS

For RBR*cervellos* equipped with Iridium RUDICS telemetry, a fixed SIM-card is already installed. This SIM-card can be used worldwide with airtime services provided by MetOcean Telematics. If you would like to use another airtime provider, please inform RBR before ordering.

Unlike Iridium SBD, where data recipients are configured on a per-device basis, RUDICS modems are associated with a shared configuration, known as a "RUDICS group". If you subscribe to airtime services through MetOcean Telematics and are using RBR-provided data hosting, you can use RBR's RUDICS group: `JOUBEH_RBRCERVELLO_STD_PROD` for US-based data hosting (https://data.rbr-global.com) or `JOUBEH_RBRCERVELLO_2_STD_PROD` for China-based data hosting (https://data.rbr.cn). Otherwise, you will need to set up a RUDICS group through your provider of choice. A setup fee for the group will apply. To use your data controller with RBR data hosting, the RUDICS group configuration is as follows:

United States-based data hosting

RUDICS type	standard (Standard or PPP)
Mobile-originated terminating IP	34.194.96.152
Mobile-originated terminating port	9140
Mobile-terminated originating IP	34.194.96.152 (not currently used by RBR)

China-based data hosting

RUDICS type	standard (Standard or PPP)
Mobile-originated terminating IP	54.223.67.137
Mobile-originated terminating port	9140
Mobile-terminated originating IP	54.223.67.137 (not currently used by RBR)

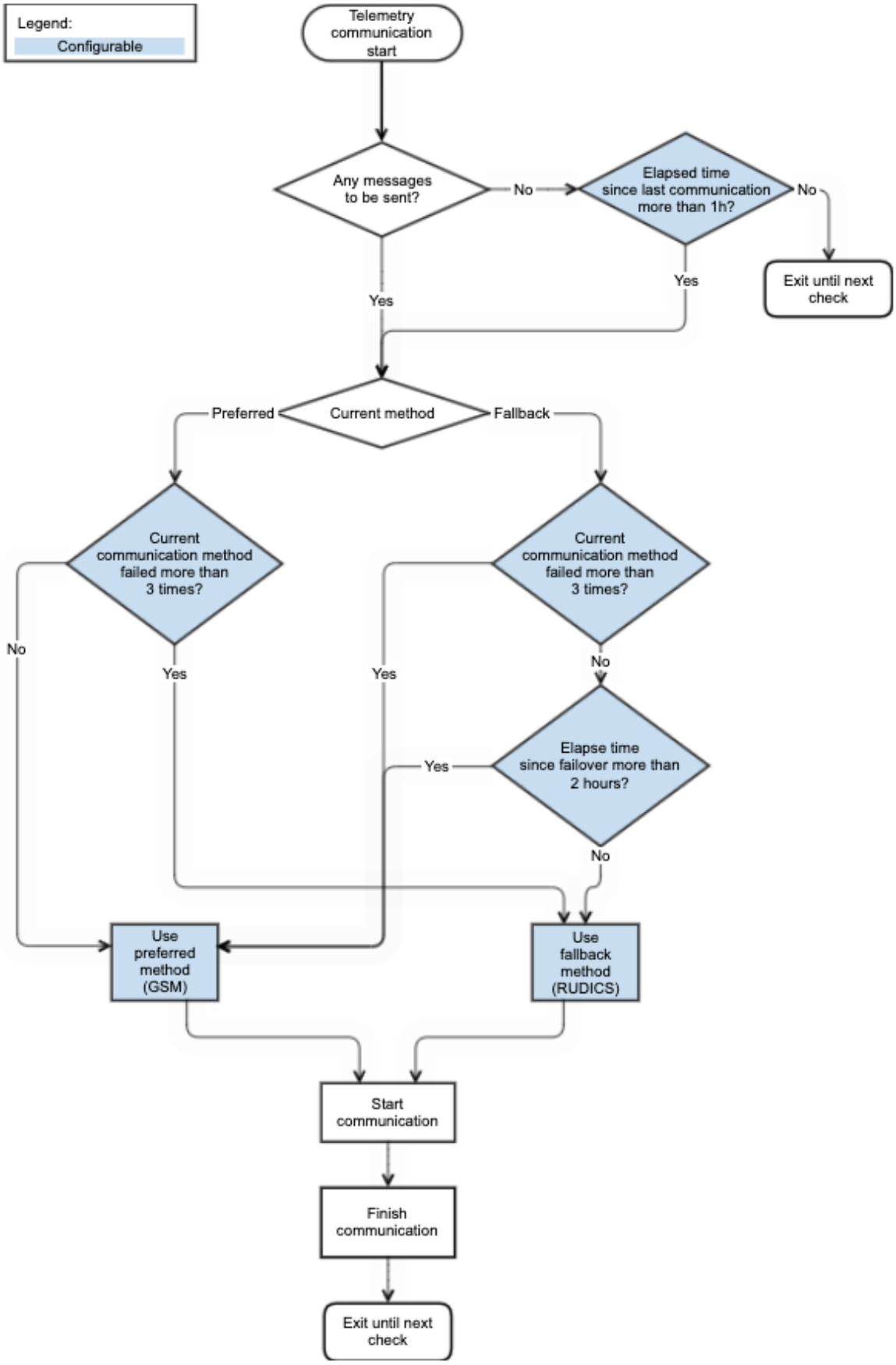
Upon setup, indicate to your airtime provider:

- the RUDICS group name to provision against
- the unique ID (ICCID) of the SIM-card installed in the modem

Contact RBR support to determine the ICCID of your SIM-card. The IMEI of the modem is not required, as that is only necessary for provisioning SBD service, which is not used by the RBR*cervello*.

Failover

The RBR*cervellos* equipped with a dual modem have a failover mechanism which determines its communication method. Usually the preferred method would be GSM, due to lower cost and faster speeds, and the fallback method would be Iridium RUDICS.


Legend:
Configurable
Telemetry communication start
Any messages to be sent?
No
Elapsed time since last communication more than 1h?
No
Exit until next check
Yes
Yes
Current method
Preferred
Fallback
Current communication method failed more than 3 times?
No
Yes
Current communication method failed more than 3 times?
Yes
No
Elapse time since failover more than 2 hours?
Yes
No
Use preferred method (GSM)
Use fallback method (RUDICS)
Start communication
Finish communication
Exit until next check

4.3 Retrieving data

Using the external MCBH USB port

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

Retrieving 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.
6. Check that all the instruments are disabled using the status command.
7. Use the mount-storage command to expose the data to the host computer.

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

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

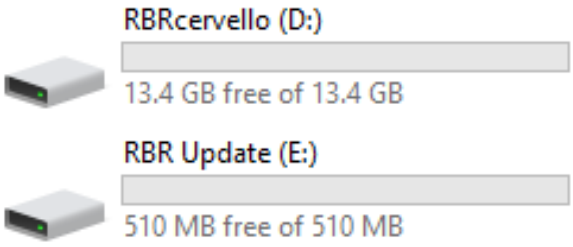


Fig. 22. Drives after using mount-storage

Using the internal USB-A port

> **Tools needed**
>
> - Computer with a USB port
> - USB flash drive formatted as FAT32 or exFAT

The RBR*cervello* has a USB interface for retrieving data from the unit and for installing firmware and configuration updates. It is compatible with FAT32-formatted or exFAT-formatted USB flash drives. Bus-powered hard drives are not recommended as the RBR*cervello* may not provide sufficient power.

To retrieve data from the RBR*cervello*, connect the USB storage device to its USB-A port while it is powered on and awake. All data will be copied from the data controller to the storage device. During data transfer, the red light on the RBR*cervello* will flash steadily. Once all data are transferred, the red light will cease flashing and the green light will turn on. At that point, the storage device may be disconnected safely. The green light will turn off after disconnecting the device.

4.4 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 might take it up to 15 seconds 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.
6. Check that all the instruments are disabled using the status command.
7. Use the mount-storage command to expose the data to the host computer.

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

8. Uncompress the update-package provided in the root directory of the "RBR Update". The folder structure should look like the screenshots below depending on what would be updated, the configuration files or the firmware.

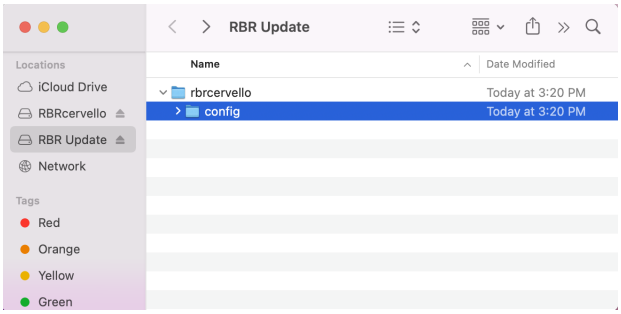


Fig. 23. Configuration update

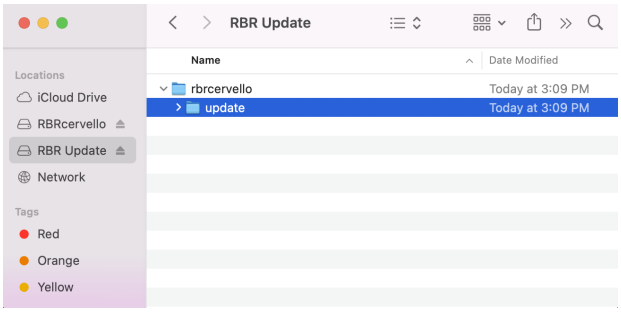


Fig. 24. Firmware update

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

9. Eject the "RBR Update" drive from the computer before proceeding.

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

10. Use the unmount-storage command to cleanly disconnect the drive.

> The update may take up to 5 minutes to finish.

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

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. At that point, the data controller will reboot itself within 5 seconds. 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 if there are any issues with your instrument.

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

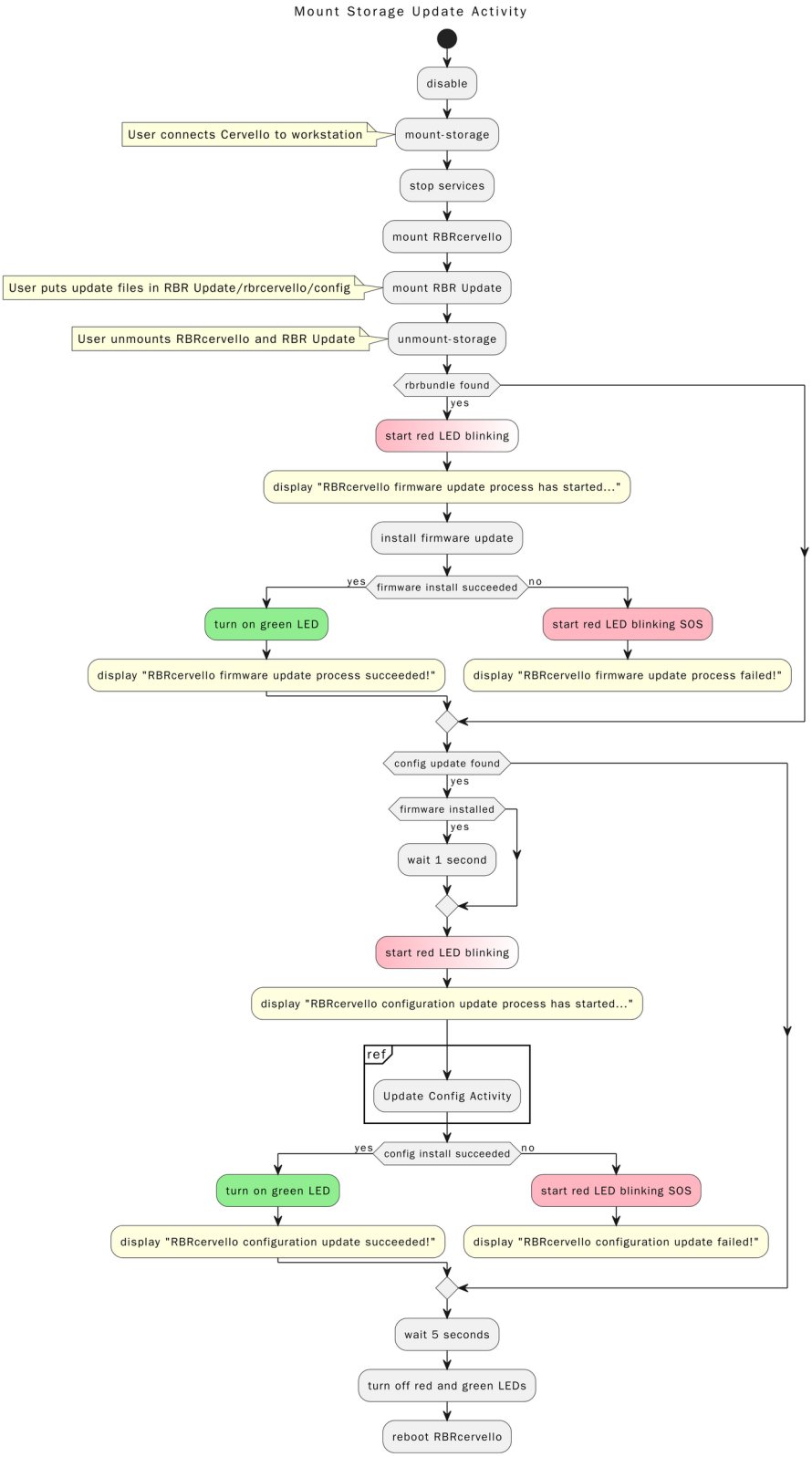
Update and revert feature

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.

After the configuration changes, and the RBR*cervello* has not yet communicated with the server, the RBR*cervello* will revert to the previous configuration if it fails to reach the server more than three times. This is intended to allow the RBRcervello to recover from configurations that prevent it from communicating with the server.

Workflow

The following diagram summarizes the workflow above.



4.5 Deployment preparation

Configuration

RBR configures each RBR*cervello* 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 RBRcervello Deployment Calculator.

Use your instruments or sub-surface modems (SSM) only with a properly configured data controller.

Failure to configure the RBR*cervello* with correct deployment parameters may compromise your mission. At best, invalid data will be reported; at worst, the deployment will be unsuccessful.

Bench testing the RBR*cervello*

Tools needed

- A computer running Ruskin
- A USB cable as appropriate to connect your instrument (RBR*concerto3* or RBR*maestro3*) to your computer
- A test loop for the MLM
- An RS-232 patch cable

An end-to-end bench test of the system will exercise all of its individual components and confirm that:

- The RBR*cervello* telemetry modems (GSM and Iridium RUDICS) are operational
- If installed, the mooring line modem (MLM) and head-end modem (HEM) can communicate with the sub-surface modems (SSM)
- If installed, the head-end optical modem (HEOM) can communicate with the sub-surface optical modem (SSOM)
- The RBR*cervello* can communicate with the instrument via the MLM or optical modem

If the RBR*cervello* has an inductive mooring line modem

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

1. Remove the internal assembly of the RBR*cervello* from the housing (optional).
2. Connect the MLM test loop cable to the RBR*cervello* (**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.
   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. Connect the batteries in the lower section of the internal frame of the RBR*cervello* to the connector labeled “Battery”.
8. Turn the switch on, if not done already.
   In case the RBR*cervello* was already powered, press any key while in the serial terminal to wake it up. The RBR*cervello* will boot up. This process will take up to two minutes. When booted, the data controller will search for and connect to an available instrument, and determine whether it needs to download any data from it.

9. Once in the terminal, run the three basic diagnostic tasks: MLM, GSM, and RUDICS.

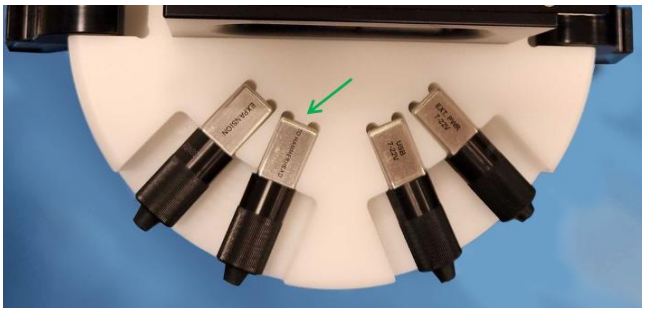


Fig. 25. TO HAMMERHEAD port

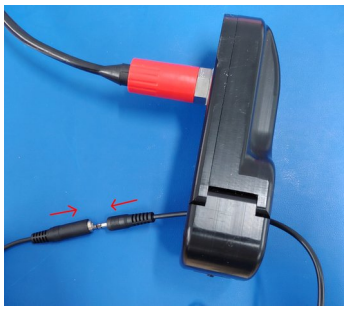


Fig. 26. MLM test loop through ferrite holder

Diagnosing MLM

The RBR*cervello* 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 buoy-mounted RBRcervello housing, and a sub-surface modem ("SSM") coupled onto the instrument. The instrument needs to be configured to communicate at the baud rate as 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 a human-readable form in the console.

Diagnosing the optical modem

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

1. Remove the internal assembly of the RBR*cervello* from the housing (optional).
2. Connect the MCIL-6-MP connector on the Head-End Optical Modem (labelled "Cervello") to the MCBH-6-FS connector on the RBR*cervello*'s endcap labelled "TO OPTICAL MODEM".
3. If not already installed, connect the Optical Modem BEC's Mini-DisplayPort connector to the instrument and screw it into place.
4. Connect the MCIL-6-FS connector on the Sub-Surface Optical Modem (labelled "LOGGER") to the MCBH-6F connector labelled ("OPTICAL MODEM") on the BEC.
5. Connect the MCBH-6-MP connector on the BEC to the RBR*fermata*. Once the instrument receives power, the SSOM will flash once.
6. Place the optical modems within 1m of each other, with their transmitter/receiver sides aligned to within a 240-degree cone. As the LEDs are very bright, it is recommended that the modems be placed with their transmitter/receivers facing down at the bench surface and that the pair should be shielded from observers with opaque material once communication is established.
7. Connect the MCIL-6-FS end of the patch cable to the RBR*cervello*'s USB port.
8. Connect the USB end of the patch cable to your computer.
9. Connect the batteries in the lower section of the internal frame of the RBR*cervello* to the connector labeled "Battery".
10. Turn on the RBR*cervello* faceplate switch. Once the RBR*cervello* receives power, and finishes booting up (which takes approximately 10 seconds), it will power on the HEOM, which will begin flashing at approximately 1s intervals until it wakes up the SSOM and establishes a communication link with the instrument, at which point further flashing indicates communication (discovery, configuration-check/configuration, and data download) between the RBR*cervello* and the instrument.

11. Open a serial terminal (e.g. TeraTerm, PuTTY) on the USB interface to the RBR*cervello*.
12. Press any key while in the serial terminal to wake up the RBR*cervello*.
13. Issue the `enable` command.
14. Wait 10s for the optical modems to establish a link.
15. Issue the `status` command. The RBR*cervello*'s status should read `logging`, the instrument's status should not read `loggererror` or `commchannelerror`, and the instrument's time-last-updated field should reflect the current UTC time.

Diagnosing LTE/GSM modem

This test requires the RBR*cervello* to have an activated SIM-card installed in the LTE/GSM modem as described in Telemetry. If LTE/GSM is not going to be used in this deployment, this test can be omitted.

After finishing the previous test, wait for the "RBRcervello>" prompt to appear again. Type “diagnose gsm” and press enter. The LTE/GSM test will wait for any ongoing communication to finish and will diagnose the connection to the internet via the LTE/GSM modem.

```
RBRcervello> diagnose gsm
Checking deployment status...
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: 867698099999999
IMSI: 302720599999999
ICCID: 89302720512345678987
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:
socket was closed
messages-read: no process found
netpty: no process found
The serial link has been brought down.
Powering off the modem.
Restarting services...
Starting periodic command scheduler: cron.
RBRcervello>
```

Diagnosing RUDICS

The RUDICS diagnostic task will test the entire communication pathway to the Iridium Satellite network. For this test to succeed, the RBR*cervello* needs to be positioned outdoors, with a clear view of the sky and with as few obstructions as possible.

After the "RBRcervello>" prompt appears again, type “diagnose rudics” and press enter. This routine will turn on the RUDICS modem and then will attempt to dial to the Iridium network as configured in Telemetry. It can identify failures in the modem, signal strength, or activation issues of the SIM-card.

```
RBRcervello> diagnose rudics
Checking deployment status...
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 2021-10-12T20:17:02+0000)
2021-11-12 20:26:24 src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-11-12 20:26:24 src/teraform.c:367 [INFO ]: Dial number: "0088169999999"
2021-11-12 20:26:24 src/teraform.c:388 [INFO ]: Configuring modem...
2021-11-12 20:26:24 src/teraform.c:432 [INFO ]: Waiting for modem to be available...
2021-11-12 20:26:24 src/teraform.c:451 [INFO ]: Checking IMEI...
2021-11-12 20:26:24 src/teraform.c:451 [INFO ]: IMEI: 300125099999999
2021-11-12 20:26:24 src/teraform.c:478 [INFO ]: Checking ICCID...
2021-11-12 20:26:24 src/teraform.c:478 [INFO ]: ICCID: 8988169999999999
2021-11-12 20:26:24 src/teraform.c:488 [INFO ]: Disabling flow control...
2021-11-12 20:26:25 src/teraform.c:520 [INFO ]: Waiting for satellite signal...
2021-11-12 20:26:31 src/teraform.c:520 [INFO ]: Found signal of strength 5/5.
2021-11-12 20:26:31 src/teraform.c:530 [INFO ]: Configuring data connection...
2021-11-12 20:26:31 src/teraform.c:564 [INFO ]: Dialing data connection...
2021-11-12 20:26:40 src/teraform.c:564 [INFO ]: Connection dialed.
The modem connection is up.
Brought the rudics connection up successfully.
Teraform v1.1.2 (built 2021-10-12T20:17:02+0000)
2021-11-12 20:26:40 src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-11-12 20:26:43 src/teraform.c:584 [INFO ]: Escaping from data call...
2021-11-12 20:26:44 src/teraform.c:594 [INFO ]: Disconnecting data call...
The serial link has been brought down.
Powering off the modem.
Restarting services...
Starting periodic command scheduler: cron.
RBRcervello>
```

If any of the tests result in failure, go to troubleshooting, or contact RBR support.

Clearing the controller and instruments

Before deployment, clear all data from the data controller and the instruments that would be attached to it.

Clearing the data is also known as "scrubbing". Commands are provided in the diagnostic menu for scrubbing the instrument and the controller. Both commands require the RBR*cervello*'s deployment to be disabled.

Follow the steps below.

1. Disable the data controller using the "disable" command.

   ```
   RBRcervello> disable
   Deployment stopped
   ```

2. Clear the memory in all instruments attached using the "scrub-instruments -a" command. This might take several

minutes, depending on the number of instruments.

```
RBRcervello> scrub-instrument -a
Clearing memory...
Instrument 01: success
Instrument 02: success
```

3. Re-check the status of the instruments using the "status" command.

```
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
Download remaining: 0B
```

4. Clear the data from the data controller using the "scrub-controller" command.

```
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...
```

If the instrument does not seem to start, go to troubleshooting, or contact RBR support.

5 Features

5.1 Data hosting

The RBR*cervello* is optionally supplied with real-time telemetry in the form of GSM cellular and Iridium Satellite modems. These send data to the RBR data hosting service, which is accessible at `https://data.rbr-global.com/ClientReference`. Each client site may be public or password protected. One year of data hosting service is included with each RBR*cervello.* After the first year, contact RBR to purchase additional years of service.

Your data is stored on the RBR data-hosting portal in the form of comma-separated value (CSV) files.

Data controllers report data under two different serial numbers: one for the attached instrument (RBR*concerto3* or RBR*maestro3*), and one for diagnostic information (battery levels and GPS coordinates). Remember to download from the appropriate instrument.

Obtaining data from the RBR data-hosting portal

Navigate to https://data.rbr-global.com/rbr (replacing RBR with your customer name) and click "Download CSV".

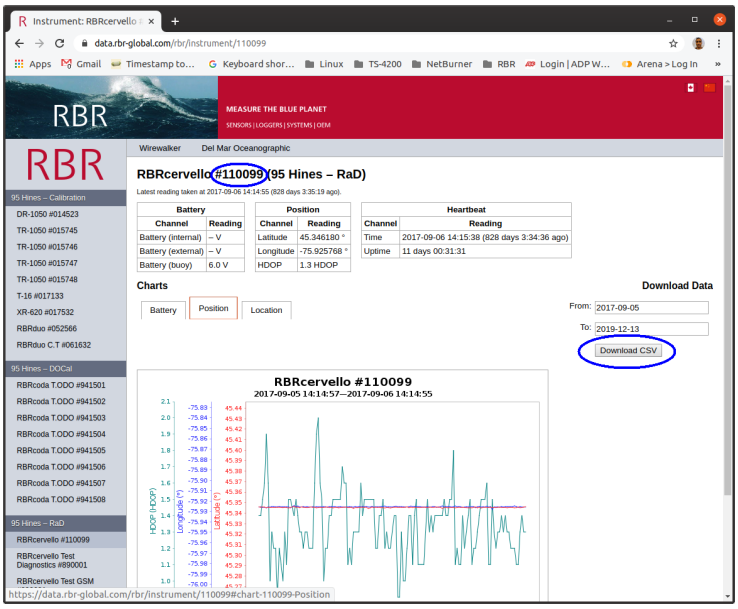


Fig. 27. Data-hosting portal interface.

Obtaining data using a shell script

Use the getcsvdata.sh script to poll and download data for a particular instrument.

After downloading the script, you will have to change the file permissions to enable execution. For example, to permit all users on the system to execute the script:

```
chmod +x getcsvdata.sh
```

Call the script using your customer URL slug, serial number, start date, and end date as arguments. For example, to download all data for the RBR's device "110099" from September 1, 2017 to December 31, 2020, use the script with this command:

```
./getcsvdata.sh rbr 110099 2017-09-01 2020-12-31
```

Manually downloading data files

Retrieve exports of instrument data by making HTTP GET requests via this URL pattern:

```
https://data.rbr-global.com/customer/download/serial?from=A&to=B
```

or

```
https://data.rbr-global.com/customer/download/serial?timespan=C
```

- `customer` is the customer slug; e.g. `rbr`, `csiro`
- `serial` is the serial number of the instrument for which data should be retrieved
- `A` and `B` are UTC dates in YYYY-MM-DD format and respect the instrument timezone. Alternatively, they can be YYYY-MM-DD**T**HH:MM:SS.
- `C` is an integer number of days

The first format will yield a CSV of all data from dates A to B, inclusive of both. The second format will yield a CSV of all data from the last C × 24 hours (*not* the most recent C × 24 hours of data available!).

For example, use the above URL format within a `curl` command to obtain the data from RBR's instrument with serial number 080296 from September 8, 2016 to September 9, 2016:

```
curl --location --remote-name --remote-header-name --compressed 'http://data.rbr-
global.com/rbr/download/080296?from=2016-09-08&to=2016-09-09'
```

It can also be more granular, from September 8, 2016, at 10:00 to 20:00:

```
curl --location --remote-name --remote-header-name --compressed 'http://data.rbr-
global.com/rbr/download/080296?from=2016-09-08T10:00:00&to=2016-09-08T20:00:00'
```

Remember to enclose the URL in quotation marks, lest your shell split the command around the ampersand (`&`) character.

In general, restrict yourself to downloading around one or two days of data at a time, and later concatenating the results. An instrument sampling at 6Hz will generate almost 100MB of CSV data per day. Large transfers may time out during the download process or your connection may be unexpectedly dropped.

Manually downloading diagnostic data files

These files usually contain useful data for diagnostic purposes, such as the battery voltage of the instruments. The same HTTP GET requests can be used with the addition of the '*download-diagnostic*' attribute.

```
curl --location --remote-name --remote-header-name --compressed 'http://data.rbr-
global.com/rbr/download/999916?download-diagnostic&from=2022-01-09&to=2022-01-10'
```

Low bandwidth transfers

If your computer has relatively low bandwidth, use the getcsvdata.sh script described above to acquire data and save them to a server with high bandwidth, always-on connection. Subsequently, transfer only the difference using the `rsync` command.

```
rsync --verbose --archive --partial --progress --compress
myusername@my.server.address:/my/data/path/RBR-080296.csv .
```

Downloads

getcsvdata.sh

5.1.1 Remote Diagnostics

5.1.1.1 Overview

As of 1.18.9, the RBRCervello supports a set of basic remote diagnostics that can be sent from the remote web interface. These diagnostic tasks allow the user to retrieve the RBRcervello's status, retrieve RBRcervello's configuration, and send new configurations to the RBRcervello during deployments.

5.1.1.2 Remote diagnostic task status table

Remote Diagnostics

Remote diagnostic status

Last diagnostic task	Queued at	Received from RBRcervello at	Evaluated on RBRcervello at	Status	Download result	Cancel task
Retrieve Status (ID 1140)	2024-07-19 20:21:06			Pending transmission.	Download	Cancel
Retrieve Status (ID 1139)	2024-07-19 20:20:40			Pending result.	Download	Cancel
Retrieve Config (ID 1138)	2024-07-19 20:11:21			Cancelled.	Download	Cancel
Update Config (ID 1137)	2024-07-19 18:41:22	2024-07-19 18:41:40	2024-07-19 18:40:00	Failed.	Download	Cancel
Update Config (ID 1136)	2024-07-19 18:40:13	2024-07-19 18:40:46	2024-07-19 18:39:07	Succeeded.	Download	Cancel
Update Config (ID 1135)	2024-07-19 18:26:14	2024-07-19 18:27:00	2024-07-19 18:25:02	Failed.	Download	Cancel
Update Config (ID 1134)	2024-07-19 18:18:50	2024-07-19 18:19:31	2024-07-19 18:17:41	Failed.	Download	Cancel
Update Config (ID 1133)	2024-07-19 18:16:24	2024-07-19 18:17:32	2024-07-19 18:15:03	Succeeded.	Download	Cancel
Update Config (ID 1132)	2024-07-19 18:14:33	2024-07-19 18:15:32	2024-07-19 18:13:10	Failed.	Download	Cancel
Retrieve Config (ID 1131)	2024-07-19 18:08:08	2024-07-19 18:09:32	2024-07-19 18:07:22	Succeeded.	Download	Cancel

Screenshot of the remote diagnostic task status table

The diagnostic task status table shows the 10 most recent tasks queued on the server.

Diagnostic tasks transition between the following states:

1. "Pending transmission" indicates that the server has queued the task and is waiting for the RBR*cervello*'s next communication attempt.
    a. At this point, the user can click "Cancel" to cancel the task.
2. "Cancelled" indicates that the server will no longer attempt to send the task.
3. "Pending result" indicates that the RBR*cervello* has communicated with the server. The RBR*cervello* will attempt to communicate with the server as soon as possible after the task executes.
4. "Success" or "Failed" indicate that the server has received the task result from the RBR*cervello*.
    a. At this point, the user can click "Download" to download the result as a text file.

5.1.1.3 Tasks

The menu beneath the diagnostic task status table allows the user to queue diagnostic tasks.

Remote diagnostic tasks	
Retrieve Status	Queue task
Retrieve Configuration	Queue task
Send Configuration	Upload a zip file containing user.json and every deployment JSON referenced in it. Browse... No file selected. Queue task

Menu for queuing diagnostic tasks

Retrieve Status

Returns the result of the `status` diagnostic command.

Retrieve Configuration

Returns the Cervello and the instruments' configurations, one after another, in a text file, preceded by their names.

The configuration files start and end with curly braces ({}).

Send a new configuration to the RBRcervello

Requires a zipped archive containing the same rbrcervello/config/<config files> tree used to configure RBR*cervello* units before deployment. See “Using the USB port” and “Updating the instrument”.

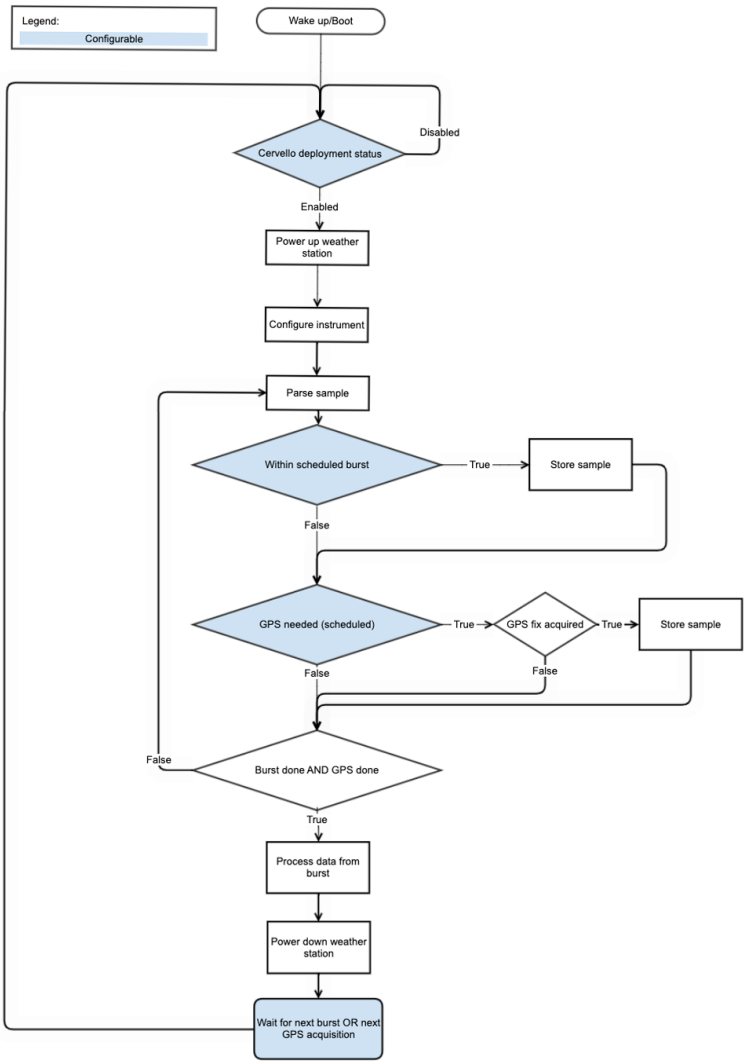
Before queueing this task, the web server runs the same configuration validation software used by `config-check`.

See “Updating the instrument” for more information on the update mechanism.

5.2 Weather station (optional)

The RBR*cervello* is optionally supplied with a weather station integration, particularly, with the Airmar 200WX WeatherStation®. It is connected to the system via the RBR*cervello* MCBH6-FS connector labelled "EXPANSION".

Deployment behaviour of the weather station depends on the configuration files of the RBR*cervello*. Contact us at support@rbr-global.com or via the RBR website to change these settings.



5.3 Optical Modem (optional)

The RBR*cervello* may be configured at the factory to communicate with sub-surface instruments using a pair of Hydromea LUMA250LP optical modems, permitting deployment in environments that are not sufficiently saline for the standard RBR MLM.

We provide an Optical Modem Kit (RBR P/N 0017848) consisting of a Sub-Surface Optical Modem (SSOM) and Head-End Optical Modem (HEOM) pair, which together form a transparent point-to-point serial connection between a modified RBR*cervello* and an RBR logger fitted with a specialized power/data battery endcap (BEC).

We advise against modifying the configuration of the modems as shipped from RBR. The shipped configuration should be optimal under most conditions. If you are experiencing problems with the optical modem link, please contact support@rbr-global.com for assistance.

Behavior

Beaconing

The Head-End Optical Modem is configured to transmit a beacon message every second.

To save power, the Sub-Surface Optical Modem is configured to go to sleep if 30 seconds pass without it receiving a beacon message.

This should not affect Cervello's data transmission.

No-attempt window

To save power while the Wirewalker profiling float is out of optical communications range, the Cervello will turn off the Head-End Optical Modem for a configurable "optical modem discovery no attempt window" if the optical communications link is broken. The optimal value of this setting will vary depending on the Wirewalker's maximum profiling depth and can be calculated as:

T = 2*(max_profiling_depth - 1.2*modem_max_range)/(0.5m/s)

This calculation does not include any safety factor on the profiling depth, and the profiling speed may change depending on the ballasting and wave conditions.

Maximum optical modem range

The optical modem's maximum range will be no more than 7m in clear water.

6 Troubleshooting

The RBR*cervello* 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. Refer to Deployment preparation for the detailed description of how to run diagnostic tests.

Communication issues

1. Using the status command, check the status of the instrument.
2. If the instrument status is `commchannelerror`, go to MLM troubleshooting.
3. If the instrument status is `loggererror`, use Ruskin to check if your logger is communicating at 19200baud.

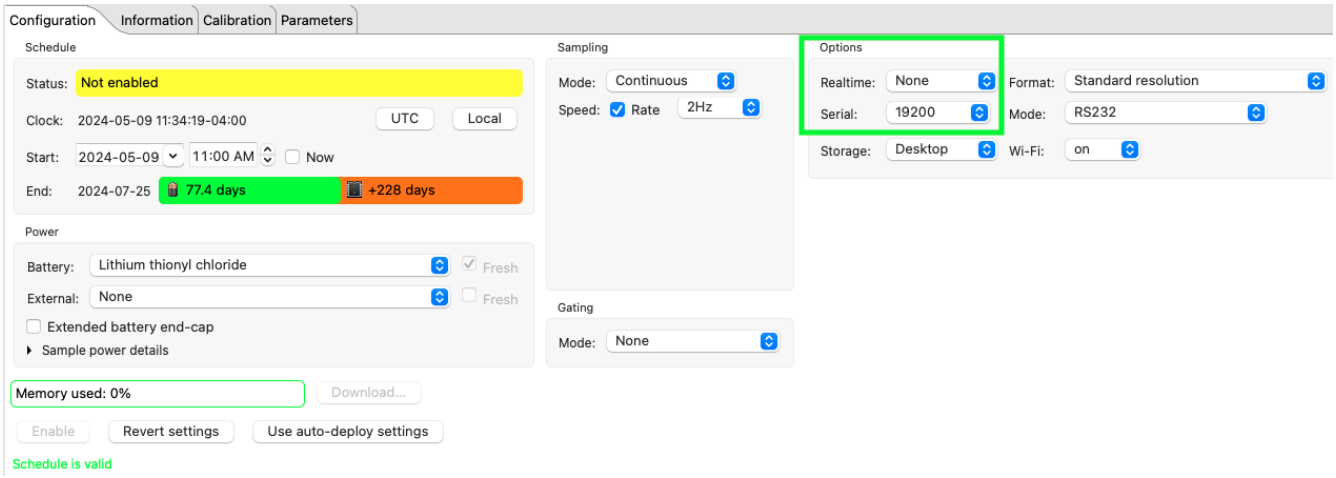


Fig. 28. Ruskin interface showing location of baud rate information

> If you change any settings on Ruskin, clear the data from the instrument and the data controller, as described in Deployment preparation.

Modem issues

If any of the modem diagnostic tests fails unexpectedly, refer to the specific troubleshooting section:

- LTE/GSM modem troubleshooting
- RUDICS modem troubleshooting

Weather station issues

If the system seems to receive incorrect navigation readings, refer to Airmar weather station troubleshooting.

6.1 LTE/GSM modem troubleshooting

Diagnostic task

This test requires the RBR*cervello* to have an activated SIM-card installed in the LTE/GSM modem as described in Telemetry. If LTE/GSM is not going to be used in this deployment, this test can be omitted.

After finishing the previous test, wait for the "RBRcervello>" prompt to appear again. Type “diagnose gsm” and press enter. The LTE/GSM test will wait for any ongoing communication to finish and will diagnose the connection to the internet via the LTE/GSM modem.

```
RBRcervello> diagnose gsm
Checking deployment status...
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: 867698099999999
IMSI: 302720599999999
ICCID: 89302720512345678987
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:
socket was closed
messages-read: no process found
netpty: no process found
The serial link has been brought down.
Powering off the modem.
Restarting services...
Starting periodic command scheduler: cron.
RBRcervello>
```

Failure, bad SIM-card

```
RBRcervello> diagnose gsm
Checking deployment status...
Stopping services...
Waiting
Stopping periodic command scheduler: cron.
Starting GSM test
I think the GSM modem is on /dev/ttyS4 at
Waiting
Modem is up.
Waiting
Established PPP connection to the modem. Modem identifiers:
IMEI: N/A
IMSI: N/A
ICCID: N/A
Waiting
network.....................................................Hit the connection timeout!
Giving up.
Test failure: failed to bring up connection!
/usr/bin/poff: No pppd is running. None stopped.
messages-read: no process found
netpty: no process found
The serial link has been brought down.
Powering off the modem.
```

```
Restarting services...
Starting periodic command scheduler: cron.
RBRcervello>
```

If so, remove and re-insert the SIM-card in the tray in the SIM1 position, as explained in the Telemetry section.

6.2 RUDICS modem troubleshooting

Diagnostic task

The RUDICS diagnostic task will test the entire communication pathway to the Iridium Satellite network. For this test to succeed, the RBR*cervello* needs to be positioned outdoors, with a clear view of the sky and with as few obstructions as possible.

After the "RBRcervello>" prompt appears again, type “diagnose rudics” and press enter. This routine will turn on the RUDICS modem and then will attempt to dial to the Iridium network as configured in Telemetry. It can identify failures in the modem, signal strength, or activation issues of the SIM-card.

```
RBRcervello> diagnose rudics
Checking deployment status...
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 2021-10-12T20:17:02+0000)
2021-11-12 20:26:24 src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-11-12 20:26:24 src/teraform.c:367 [INFO ]: Dial number: "0088169999999"
2021-11-12 20:26:24 src/teraform.c:388 [INFO ]: Configuring modem...
2021-11-12 20:26:24 src/teraform.c:432 [INFO ]: Waiting for modem to be available...
2021-11-12 20:26:24 src/teraform.c:451 [INFO ]: Checking IMEI...
2021-11-12 20:26:24 src/teraform.c:451 [INFO ]: IMEI: 300125099999999
2021-11-12 20:26:24 src/teraform.c:478 [INFO ]: Checking ICCID...
2021-11-12 20:26:24 src/teraform.c:478 [INFO ]: ICCID: 8988169999999999
2021-11-12 20:26:24 src/teraform.c:488 [INFO ]: Disabling flow control...
2021-11-12 20:26:25 src/teraform.c:520 [INFO ]: Waiting for satellite signal...
2021-11-12 20:26:31 src/teraform.c:520 [INFO ]: Found signal of strength 5/5.
2021-11-12 20:26:31 src/teraform.c:530 [INFO ]: Configuring data connection...
2021-11-12 20:26:31 src/teraform.c:564 [INFO ]: Dialing data connection...
2021-11-12 20:26:40 src/teraform.c:564 [INFO ]: Connection dialed.
The modem connection is up.
Brought the rudics connection up successfully.
Teraform v1.1.2 (built 2021-10-12T20:17:02+0000)
2021-11-12 20:26:40 src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-11-12 20:26:43 src/teraform.c:584 [INFO ]: Escaping from data call...
2021-11-12 20:26:44 src/teraform.c:594 [INFO ]: Disconnecting data call...
The serial link has been brought down.
Powering off the modem.
Restarting services...
Starting periodic command scheduler: cron.
RBRcervello>
```

Failure, no satellite signal

This error indicates there is a connection problem with the antenna, or the antenna is obstructed.

```
RBRcervello> diagnose rudics
Checking deployment status...
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 2021-10-12T20:17:02+0000)
2021-11-12 20:42:48 src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-11-12 20:42:48 src/teraform.c:367 [INFO ]: Dial number: "0088160999999"
2021-11-12 20:42:48 src/teraform.c:388 [INFO ]: Configuring modem...
2021-11-12 20:42:48 src/teraform.c:432 [INFO ]: Waiting for modem to be available...
2021-11-12 20:42:48 src/teraform.c:451 [INFO ]: Checking IMEI...
2021-11-12 20:42:48 src/teraform.c:451 [INFO ]: IMEI: 3001250619999999
2021-11-12 20:42:48 src/teraform.c:478 [INFO ]: Checking ICCID...
2021-11-12 20:42:49 src/teraform.c:478 [INFO ]: ICCID: 8988169763060999999
2021-11-12 20:42:49 src/teraform.c:488 [INFO ]: Disabling flow control...
2021-11-12 20:42:49 src/teraform.c:520 [INFO ]: Waiting for satellite signal...
2021-11-12 20:42:52 src/teraform.c:520 [INFO ]: No signal found. Waiting to try again.
2021-11-12 20:43:03 src/teraform.c:520 [INFO ]: Waiting for satellite signal...
2021-11-12 20:43:03 src/teraform.c:520 [INFO ]: No signal found. Waiting to try again.
2021-11-12 20:43:14 src/teraform.c:520 [INFO ]: Waiting for satellite signal...
2021-11-12 20:43:18 src/teraform.c:520 [INFO ]: No signal found. Waiting to try again.
2021-11-12 20:43:29 src/teraform.c:520 [INFO ]: Waiting for satellite signal...
2021-11-12 20:43:29 src/teraform.c:520 [INFO ]: No signal found after the final
attempt.
2021-11-12 20:43:29 src/teraform.c:520 [ERROR]: Failed after 4 attempts.
Failed to dial connection! Giving up.
Test failure: failed to bring up connection!
Teraform v1.1.2 (built 2021-10-12T20:17:02+0000)
2021-11-12 20:43:29 src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-11-12 20:43:32 src/teraform.c:584 [INFO ]: Escaping from data call...
2021-11-12 20:43:35 src/teraform.c:584 [INFO ]: Escaping from data call...
2021-11-12 20:43:38 src/teraform.c:584 [ERROR]: Failed after 2 attempts.
The serial link has been brought down.
Powering off the modem.
Restarting services...
Starting periodic command scheduler: cron.
RBRcervello>
```

Failure, no carrier or activation issue

This failure usually occurs because of a bad connection with the network or when the SIM-card has not been successfully activated.

```
RBRcervello> diagnose rudics
Checking deployment status...
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 2021-10-12T20:17:02+0000)
2021-11-12 20:54:02 src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-11-12 20:54:02 src/teraform.c:367 [INFO ]: Dial number: "0088160999999"
2021-11-12 20:54:02 src/teraform.c:388 [INFO ]: Configuring modem...
2021-11-12 20:54:02 src/teraform.c:432 [INFO ]: Waiting for modem to be available...
2021-11-12 20:54:02 src/teraform.c:451 [INFO ]: Checking IMEI...
2021-11-12 20:54:03 src/teraform.c:451 [INFO ]: IMEI: 300125061999999
2021-11-12 20:54:03 src/teraform.c:478 [INFO ]: Checking ICCID...
2021-11-12 20:54:03 src/teraform.c:478 [INFO ]: ICCID: 8988169234002999999
2021-11-12 20:54:03 src/teraform.c:488 [INFO ]: Disabling flow control...
2021-11-12 20:54:03 src/teraform.c:520 [INFO ]: Waiting for satellite signal...
2021-11-12 20:54:12 src/teraform.c:520 [INFO ]: Found signal of strength 1/5.
2021-11-12 20:54:12 src/teraform.c:530 [INFO ]: Configuring data connection...
2021-11-12 20:54:12 src/teraform.c:564 [INFO ]: Dialing data connection...
2021-11-12 20:54:12 src/teraform.c:564 [INFO ]: Dial reported no carrier. Waiting 40s
to try again.
2021-11-12 20:54:52 src/teraform.c:564 [INFO ]: Dialing data connection...
2021-11-12 20:54:52 src/teraform.c:564 [INFO ]: Dial reported an error.
2021-11-12 20:54:52 src/teraform.c:564 [ERROR]: Failed after 2 attempts.
Failed to dial connection! Giving up.
Test failure: failed to bring up connection!
Teraform v1.1.2 (built 2021-10-12T20:17:02+0000)
2021-11-12 20:54:52 src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
2021-11-12 20:54:55 src/teraform.c:584 [INFO ]: Escaping from data call...
2021-11-12 20:54:58 src/teraform.c:584 [INFO ]: Escaping from data call...
2021-11-12 20:55:01 src/teraform.c:584 [ERROR]: Failed after 2 attempts.
The serial link has been brought down.
Powering off the modem.
Restarting services...
Starting periodic command scheduler: cron.
RBRcervello>
```

6.3 MLM troubleshooting

The RBR*cervello* 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 buoy-mounted RBRcervello housing, and a sub-surface modem (“SSM”) coupled onto the instrument. The instrument needs to be configured to communicate at the baud rate as 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 a human-readable form in the console.

Diagnostic tasks

In the terminal window, press **Enter** a couple of times and wait for the `RBRcervello>` prompt. Type `diagnose mlm` and press **Enter**. Immediately, 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.

```
RBRcervello> diagnose mlm
Checking deployment status...
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...
RBR HEM-1000 1.300 208882
Found HEM 208882.
HEM firmware version 1.300 >= 1.300.
commands now unlocked
Checking HEM configuration...
Checking: baud rate BAUD: 115200
Setting: baud rate BAUD: 115200.
.
.
.
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 999888
Found SSM 989989
Discovery complete
Waiting for channel cooldown...
.
.
.
MLM communication looks good.
Restarting services...
Starting periodic command scheduler: cron.
RBRcervello>
```

Transparent link

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

```
RBRcervello> 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 10 seconds, 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 wake 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.

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. 29. 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 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.

6.4 Airmar weather station troubleshooting

Diagnostic task

This calibration sequence requires the RBR*cervello* to be mounted with the Airmar station in the buoy. If the system does not have an Airmar weather station, omit this step.

After finishing the previous test, wait for the "RBRcervello>" prompt to appear again. Type “diagnose calibrate-airmar-compass -p 4” and press "Enter". This script will request the user to do a slow circular turn of the buoy to compensate the compass for any nearby structures that might affect the compass readings.

```
RBRcervello> diagnose calibrate-airmar-compass -p 4
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 4? (y/N) y
Powering on port 4...
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>
```

Calibration failed

The Airmar station might reject a calibration sequence if it determines the data quality is not good enough. In such a case, repeat the process at different turning speeds to improve the quality of the calibration.

```
RBRcervello> diagnose calibrate-airmar-compass -p
Stopping services...
Waiting
Stopping periodic command scheduler: cron.
Are you sure you want to begin a compass calibration sequence on port
Powering on port
Waiting
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
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
Calibration failed! Please
RBRcervello>
```

6.5 Optical modem troubleshooting

> The optical modems' LEDs are very bright, flash rapidly, and may trigger photosensitive epilepsy. Avoid staring directly at the optical modem when it is in operation.

As an alternative to the RBR mooring line modem ("MLM") RBR offers a variant of the RBR*cervello* that uses an optical modem to communicate with the instrument mounted on the mooring line. This consists of a head-end optical modem ("HEOM") that is coupled to the buoy-mounted RBR*cervello*, and a sub-surface optical modem ("SSOM") coupled onto the instrument.

Power-on indication

The SSOM will flash once when powered on.

The HEOM will begin flashing at 1-second intervals when powered on.

Optical modem baud rate

The instrument needs to be configured to communicate at the baud rate as configured in the SSOM, which is typically 115200 bits-per-second. The RBR*cervello* and the HEOM are also pre-configured to communicate at 115200 bits-per-second.

Opening an interactive communication session

As with the RBR MLM, the optical modems can be troubleshooted using `transparent-link` command

With the HEOM connected to the RBR*cervello*'s optical modem port and the SSOM connected to an instrument's optical modem port:

1. Place the optical modems within 1m of each other, with their transmitter/receiver sides aligned within a 240-degree cone. As the LEDs are very bright, it is recommended that the modems be placed on the bench surface with their transmitter/receivers facing down and that the pair be covered.
2. Ensure that the instrument is receiving power. When the instrument begins receiving power, the SSOM will flash once.
3. Issue the `status` command.
4. If the RBR*cervello*'s status is not already `disabled`, issue the `disable` command.
5. Issue the `transparent-link` command.
6. If the HEOM is not already powered on, indicated by flashing at approximately 1s intervals, the text "`Onboard parameters initialised. ☒3☒set DEBUGLEVEL to6☒?`" should appear in the RBR*cervello*'s terminal after "`` `Connected. Press CTRL-C to exit when done.` ``", and the HEOM should begin flashing at approximately 1s intervals.
7. Tap the ENTER key. The HEOM should flash as it transmits the incoming data.
8. Issue the `id` command. The HEOM should flash as it transmits the incoming data and the SSOM should flash as it transmits the response. The instrument's type, version, serial number, and firmware type should appear in the RBR*cervello*'s terminal.

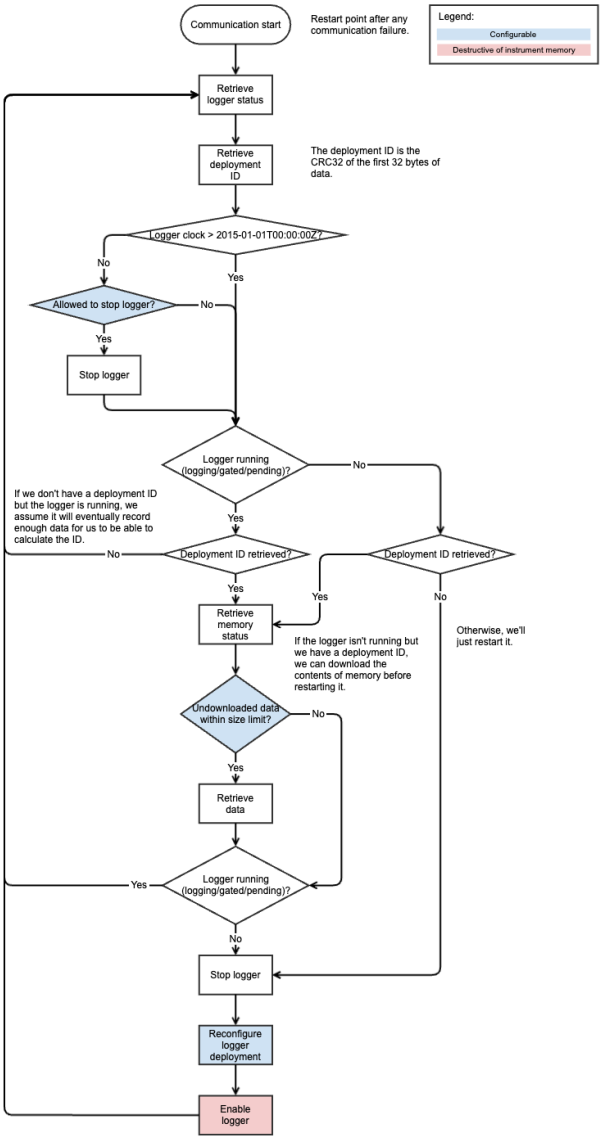
For further troubleshooting, if both the HEOM and the SSOM are receiving power but not exchanging data, an RBR RS-232 patch cable can be used to communicate with either optical modem from a PC running a terminal emulator at 115200 baud.

7 Reference

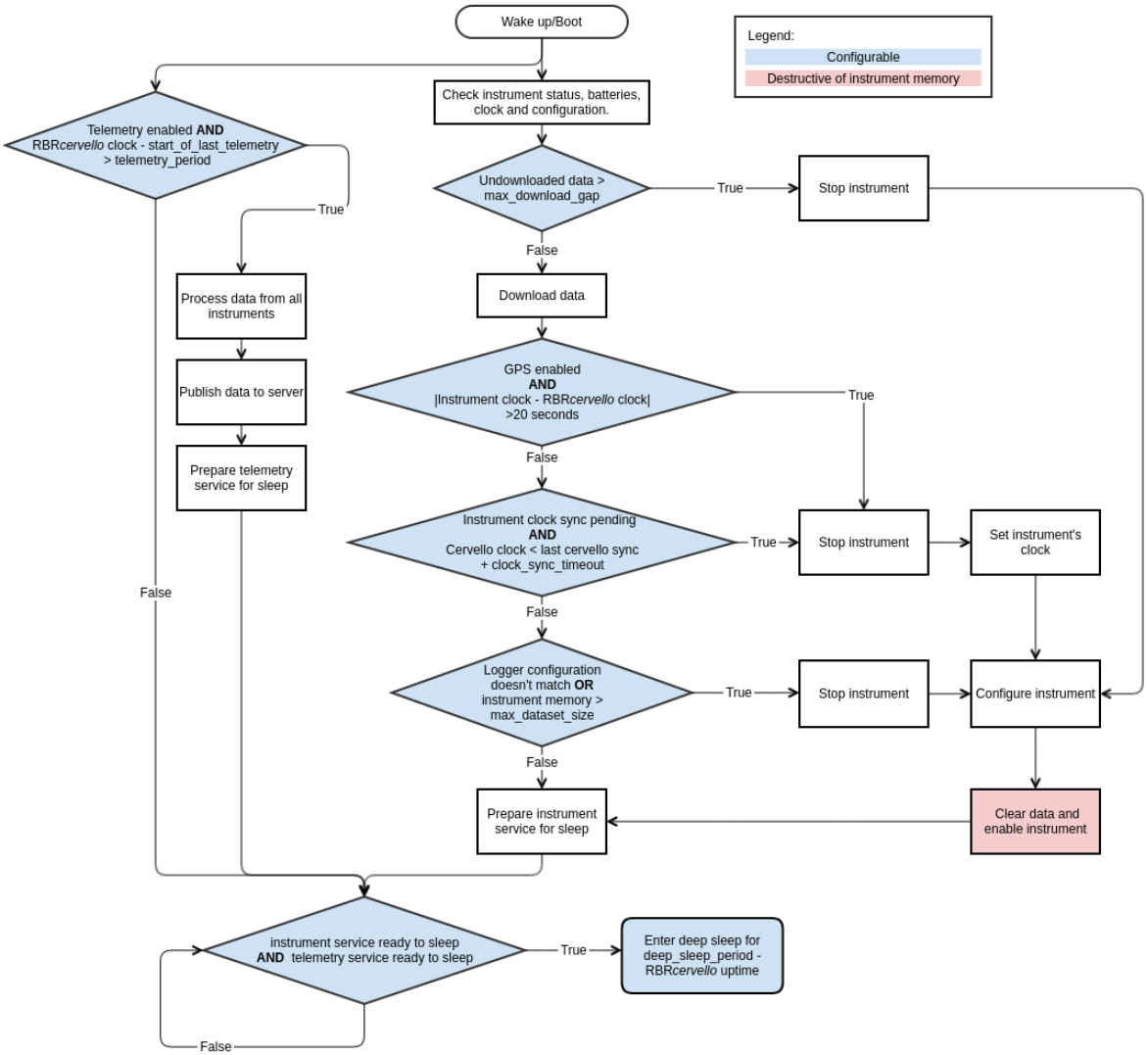
7.1 Deployment behaviour

The RBR*cervello* will oversee deployment of attached instruments, configure instrument settings, and start the logging schedule.

7.1.1 Behaviour with deep-sleep disabled



7.1.2 Behaviour with deep-sleep enabled



7.2 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>`.

7.2.1 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
```

7.2.2 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
```

7.2.3 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>
```

7.2.4 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
Validating subset
Validation succeeded!
```

7.2.5 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...
```

7.2.6 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'
```

7.2.7 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'
```

7.2.8 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!
...
```

7.2.9 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...
```

7.2.10 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
```

7.2.11 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
```

7.2.12 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.
```

7.2.13 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

```
RBRcervello> 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.
```

7.2.14 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

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

7.2.15 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

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

8 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.

9 Revision history

Revision No.	Release date	Notes
A	21-May-2024	Initial release.
B	14-Mar-2025	Added information for optical modem: • Block diagram • Feature description • Getting started instruction • Troubleshooting page
C	14-Sep-2026	Migrated. Reformatted and reviewed; no technical changes.