RBR*cervello* (Antarctica)

INSTRUMENT GUIDE

rbr-global.com

Table of Contents

1 Introduction . . . **3**

2 Hardware . . . **4**

2.1 Opening and closing the RBRcervello . . . 4

2.2 RBRcervello Antarctica crate . . . 5

3 General maintenance . . . **6**

3.1 Support kit . . . 6

3.2 Replacing batteries . . . 6

3.3 Replacing desiccant . . . 7

3.4 Cables and connectors . . . 8

4 Getting started . . . **9**

4.1 Using the diagnostic port . . . 9

4.2 Telemetry . . . 12

4.3 Deployment preparation . . . 18

4.4 Deploying the RBRcervello Antarctica . . . 23

5 Features . . . **28**

5.1 USB interface . . . 28

5.2 Data hosting . . . 29

6 Troubleshooting . . . **31**

7 Reference . . . **34**

7.1 Deployment behaviour . . . 34

7.2 Diagnostic menu . . . 35

7.2.1 status . . . 35

7.2.2 disable . . . 37

7.2.3 diagnose . . . 38

7.2.4 config-check . . . 43

7.2.5 edit-config . . . 44

7.2.6 set-cervello-clock . . . 45

7.2.7 cervello-clock . . . 46

7.2.8 reboot-controller . . . 47

7.2.9 scrub-controller . . . 47

7.2.10 scrub-instrument . . . 48

7.2.11 enable . . . 49

7.2.12 help . . . 50

8 Repairs . . . **51**

1 Introduction

The RBR*cervello* is a general-purpose controller designed for use in surface expressions of various kinds. 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.

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 RBRcervello Antarctica crate

The RBR*cervello* Antarctica crate contains all the elements needed to deploy the RBR*cervello* controller in dry environments with a mooring line feed. It features cable feed-throughs for a mooring line, an external GPS antenna, and an external Iridium antenna, as well as staples, screws, screwdriver, hex keys, zip ties, castors, and clamps.

Fig. 1. RBRcervello Antarctica crate, open

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.

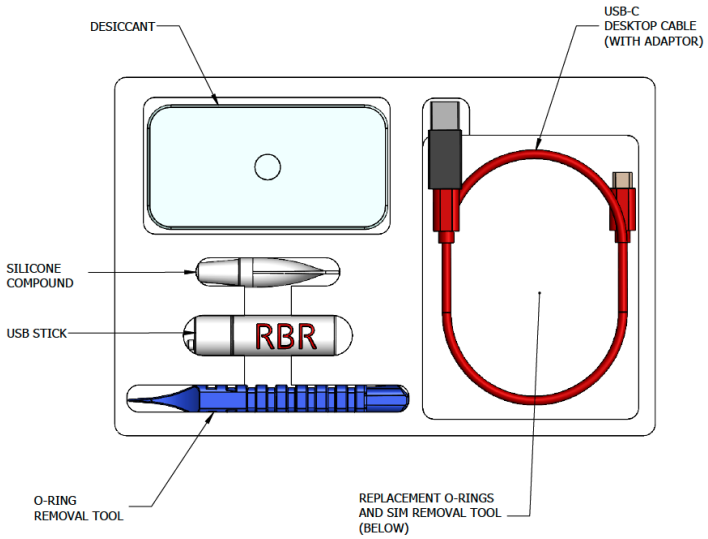


Fig. 2. RBRcervello support kit diagram

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. 3. MLM test loop cable

Fig. 4. 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 work 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. 5. 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.

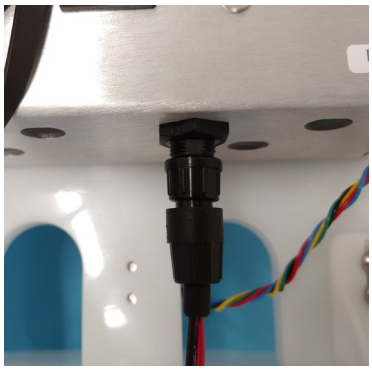


Fig. 6. Lock connector

Fig. 7. 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.

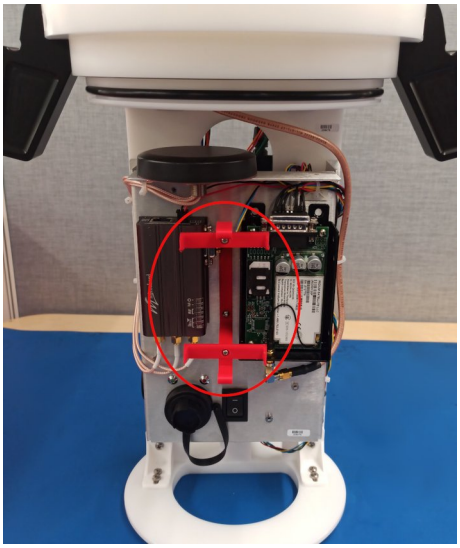


Fig. 8. Desiccant canister holder

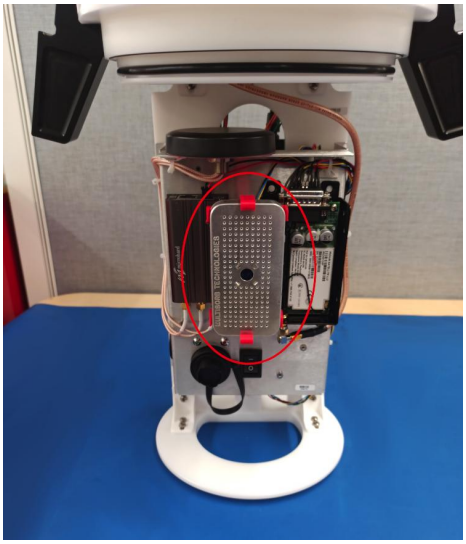


Fig. 9. Desiccant canister fully inserted

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

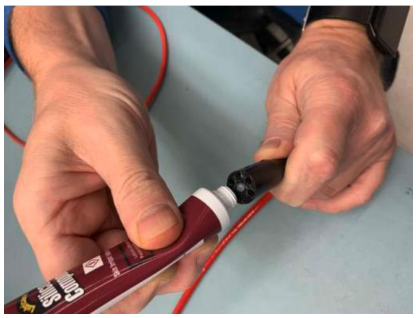


Fig. 10. Lubricating a connector

Reducing mechanical stress

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

4 Getting started

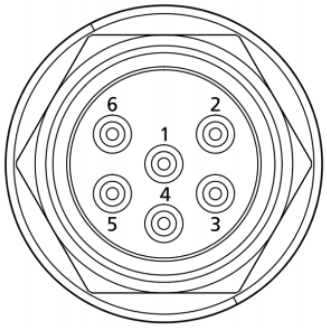
4.1 Using the diagnostic port

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

Cable connection

The RBR*cervello* diagnostic port consists of a 6-pin bulkhead connector with power and RS-232 connections. It is meant to be used in conjunction with an RBR patch cable with RS-232 electrical connections, like the one provided in the RBR*cervello* shipping crate.

External MCBH-6-MP diagnostic connector pinout



Pin No.	RS-232
1	Ground
2	Power +9 to +22 V
3	Data output from the instrument (Tx)
4	Data input into the instrument (Rx)
5	N/C
6	N/C

Serial terminal

- 115200baud
- 8bit, no parity-bit, 1 stop-bit: 8N1
- No flow control

Use a terminal emulator to communicate with the *RBRcervello* through its diagnostic port.

Windows

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

Tera Term

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

Teraterm_RBRcervello.INI

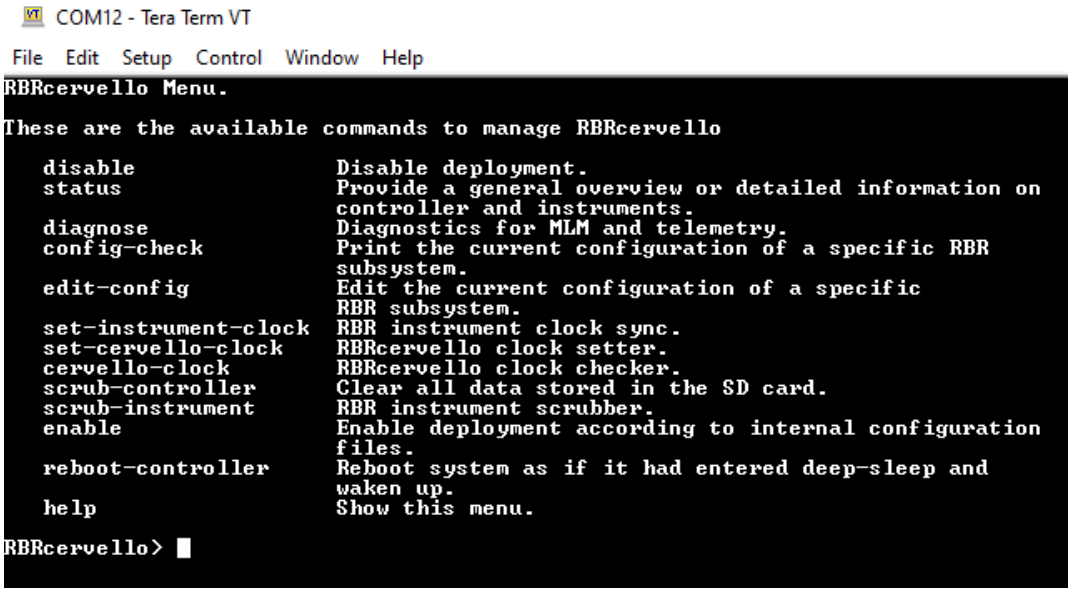
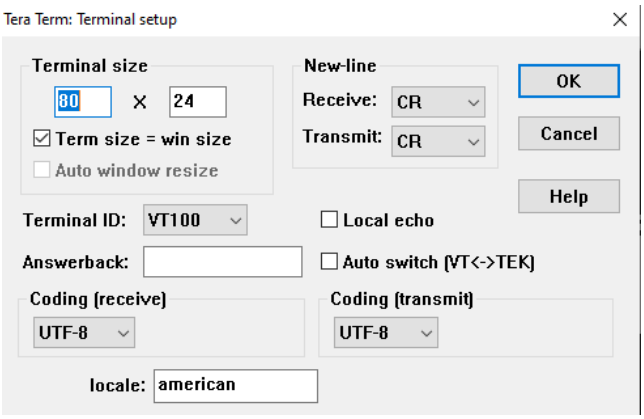


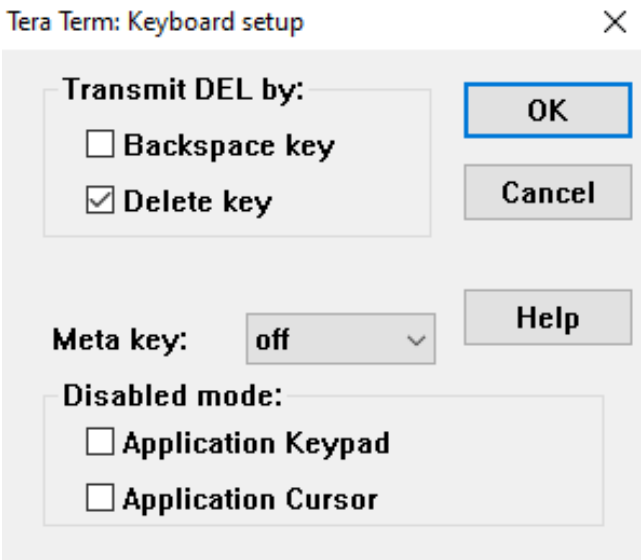
Fig. 11. Tera Term

Alternatively, configure the settings manually as follows:

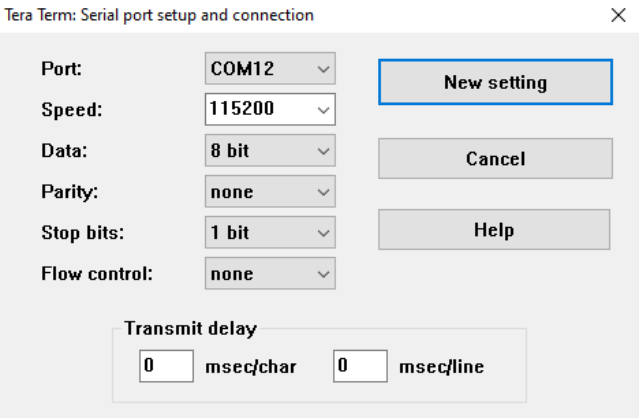
- In "Setup→Terminal setup" menu, enter the terminal size:



- In "Setup→Keyboard setup" menu, select "Delete key":



- In "Setup→Serial port setup and connection" menu, select the correct COM port:



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

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 -b 115200 -D /dev/ttyUSB0
```

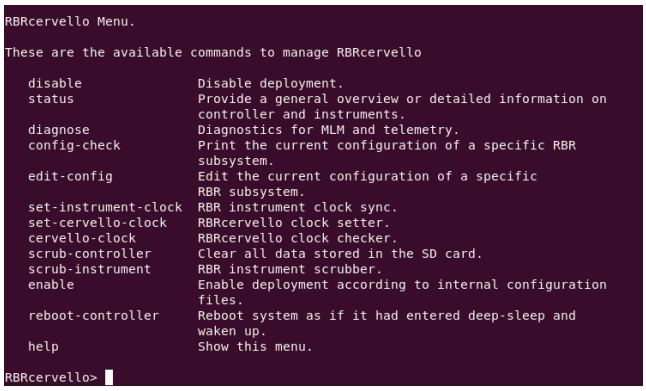


Fig. 12. Minicom

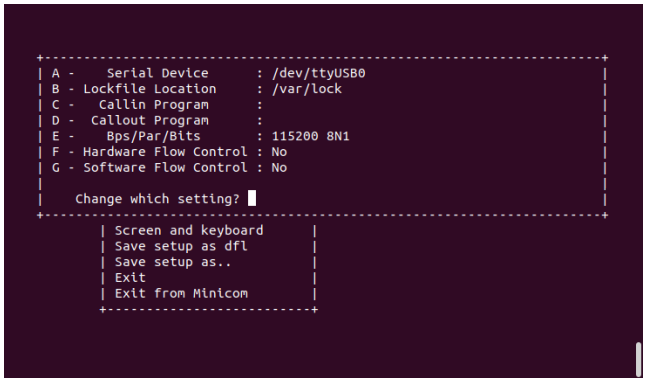


Fig. 13. Minicom serial settings

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

Diagnostic menu overview

Access the RBR*cervello* menu via the diagnostic port using an RS-232 interface (115200 8N1). 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.

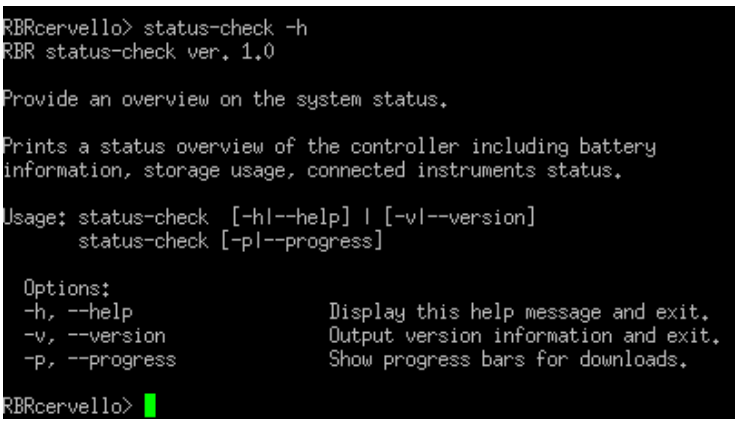


Fig. 14. "Status" help

```
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.
   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.
RBRcervello>
```

See the Diagnostic menu section for more information.

4.2 Telemetry

All new RBR *cervello* Antarctica data controllers are equipped with a dual-mode GSM/RUDICS modem. Some legacy data controllers may only have the GSM-only modem integrated. See below for details on setting up your telemetry options, depending on your modem type.

GSM

Before 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.

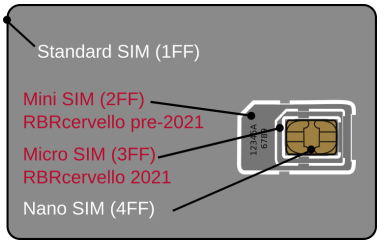


Fig. 15. SIM-card comparison (Wikipedia)

Fig. 16. Legacy GSM modem, isolated, with the SIM-card slot highlighted

Fig. 17. GSM modem, dual-mode, pre-April 2021

Fig. 18. GSM modem, dual-mode, post-April 2021

Supported bands

Legacy GSM-only	UMTS/HSPA FDD Bands (Bands 1,2,4,5,6,8)
Dual-mode GSM pre-2021	LTE FDD (Bands 2,4,5,12,13,14,66,71) WCDMA (Bands 2,4,5)
Dual-mode GSM 2021	LTE FDD (Bands 1,2,3,4,5,7,8,12,13,18,19,20,26,28) LTE TDD (Bands 38,39,40,41) WCDMA (Bands 1,2,4,5,6,8,19) GSM (Bands 2,3,5,8)

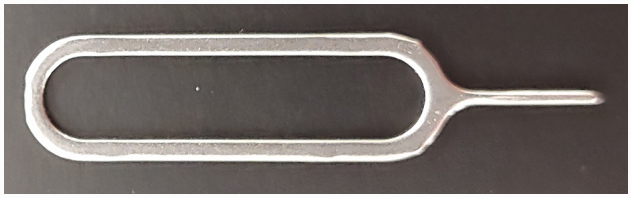
Accessing the GSM modem

For legacy, GSM-only data controllers (to open the main RBR*cervello* enclosure):

- 2.5mm hex key
- Philips #2 screwdriver

For dual-modems manufactured after April 2021:

- SIM removal tool (included in the support kit)



A GSM modem is integrated into the RBR*cervello*. Its location varies based on whether you have a legacy, GSM-only RBR*cervello* data controller variant or a dual-mode GSM/RUDICS data controller.

Legacy, GSM-only

1. Pull the handle straight up and remove the internal assembly from the housing.

   Mounted in the middle section of the internal frame are two black enclosures. The smaller of these is the head-end modem (HEM) of the mooring line modem (MLM) inductive modem system. The larger contains the RBR*cervello* processor and the GSM modem.

2. Disconnect the cables from the top and bottom of each enclosure.
3. Remove the four 2.5mm hex screws holding the HEM in place.
4. Remove the HEM.
5. Remove the four Philips #2 screws from the lid of the main RBR*cervello* enclosure.
6. Remove the lid.

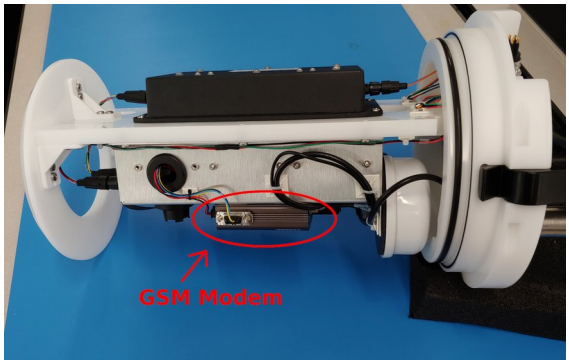
   The GSM modem is the topmost component in the main enclosure.

Dual-mode GSM/RUDICS

1. Flip the two tabs on the top handle outwards
2. Pull the handle straight up to remove the internal assembly.

3. Rest the internal assembly on the lip of the housing.
4. Disconnect the battery connector.
5. Remove the internal assembly fully and rest it on a flat work surface.
6. Orient the internal assembly with the HEM facing upwards.

   The GSM modem will be directly above the LEDs.



RBRcervello, dual-mode, pre-April 2021

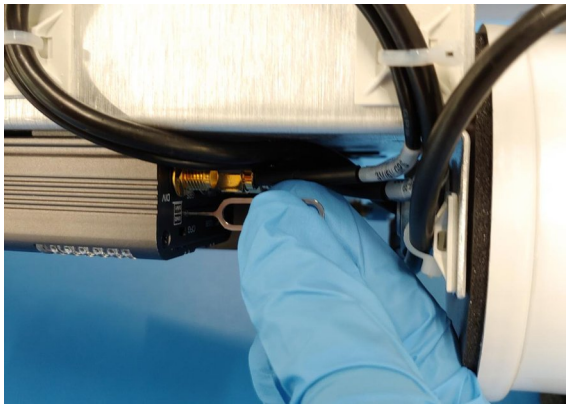
The GSM modem accepts a mini SIM-card (2FF). Once you have located the modem, insert the mini SIM-card into the SIM-card slot, oriented as indicated by the label on the modem.

RBRcervello, dual-mode, manufactured after April 2021

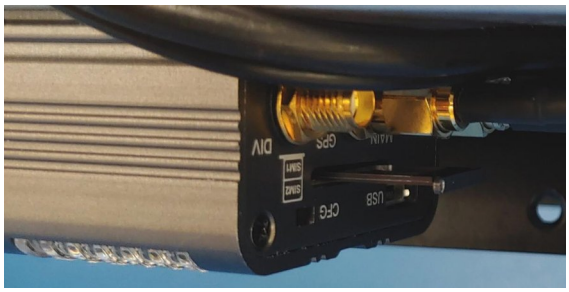
The GSM modem accepts a micro SIM-card (3FF).

1. Locate the pinhole in the SIM tray.

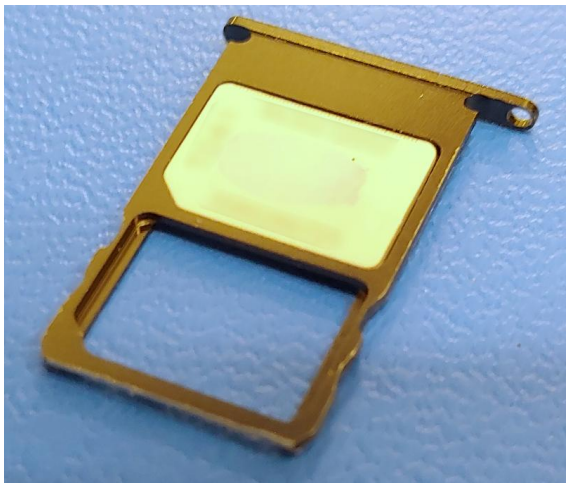
2. Insert the SIM removal tool as straight as possible into the pinhole and push. The tray will pop out.



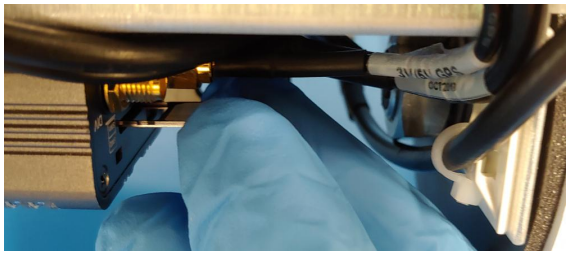
3. Pull the tray out of the modem.



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:

US-based data hosting

RUDICS type	standard (Standard or PPP)
Mobile-originated terminating IP	34.194.96.152
Mobile-originated terminating port	9131
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	9131
Mobile-terminated originating IP	54.223.67.137 (not currently used by RBR)

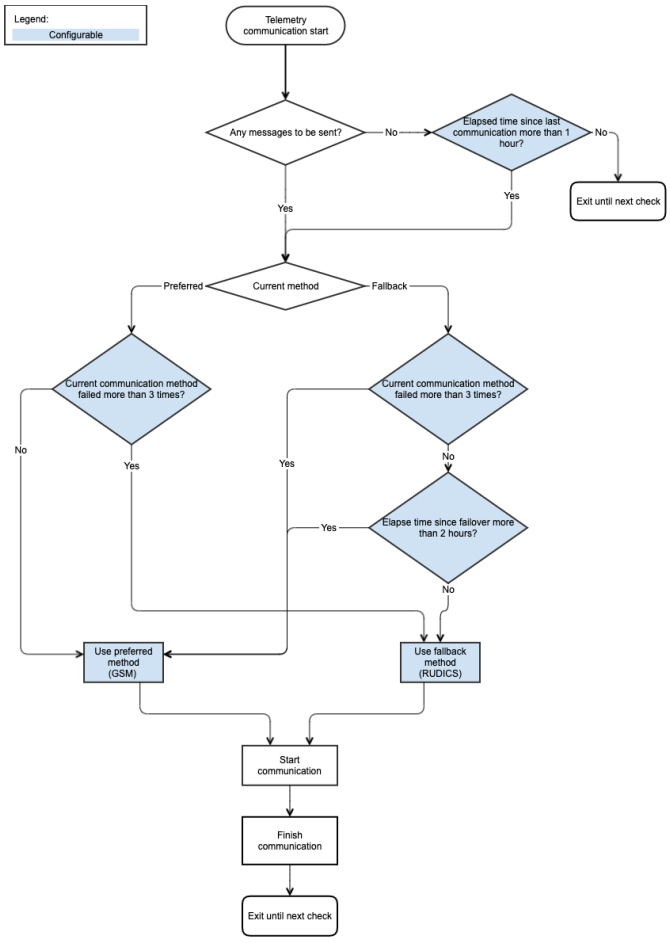
Upon setup, indicate to your airtime provider:

- 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

RBR*cervellos* equipped with a dual-mode 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.



4.3 Deployment preparation

Configuration

Each RBR*cervello* is configured by RBR to work with a specific instrument. For details particular to your instrument or to change the configuration, please contact RBR.

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

- A computer running Ruskin
- A USB cable as appropriate to connect your instrument (RBR*concerto*^{3} or RBR*maestro*^{3}) 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
- The mooring line modem (MLM) and head-end modem (HEM) can communicate with the sub-surface modems (SSM)
- The RBR*cervello* can communicate with the instrument via the MLM

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 test loop cable to the RBR*cervello.*
3. Connect the instruments to their respective ferrite holders.
4. Pass the test loop through the ferrite holders.
   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, RUDICS, and GSM.

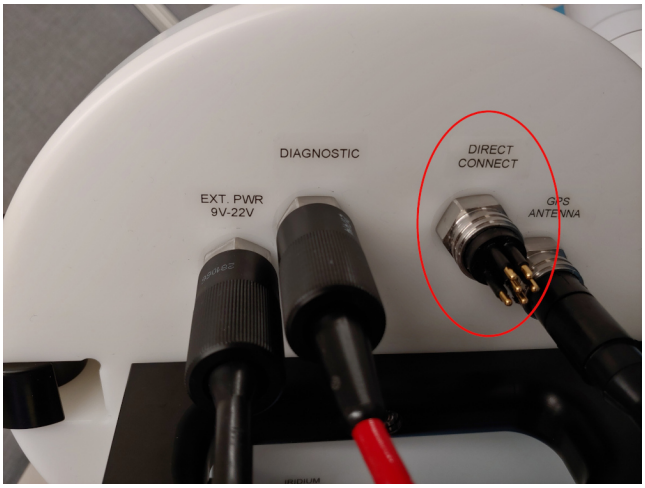


Fig. 19. DIRECT CONNECT port

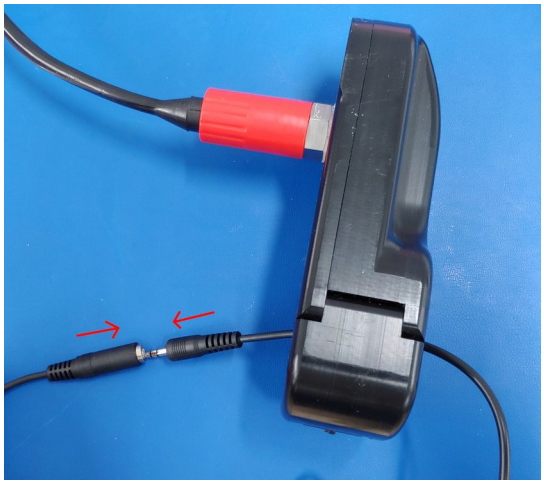


Fig. 20. MLM test loop through ferrite holder

Diagnosing MLM

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>
```

Diagnosing GSM

This test requires the RBR*cervello* to have an activated SIM-card installed in the GSM modem as described in Telemetry. If 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 GSM test will wait for any ongoing communication to finish and will diagnose the connection to the internet via the 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 RBR*cervello* 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 RBR*cervello* 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 RBR*cervello* 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...
```

4.4 Deploying the RBRcervello Antarctica

Installing the RBR*cervello* Antarctica on site

Follow the steps below to set up the RBR*cervello* Antarctica on site.

- 5mm hex key
- Philips #2 screwdriver
- zip ties
- cutting tool (not included!)

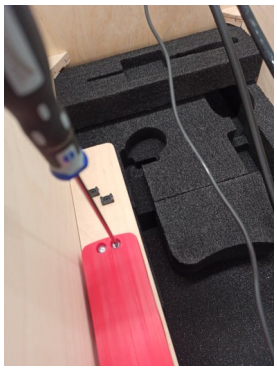
1. Open the crate.

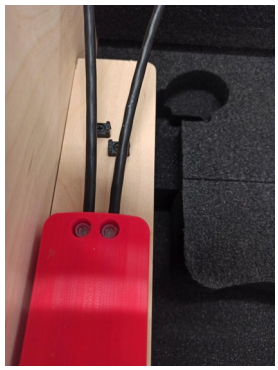
2. Install the mooring line modem (MLM) lines.

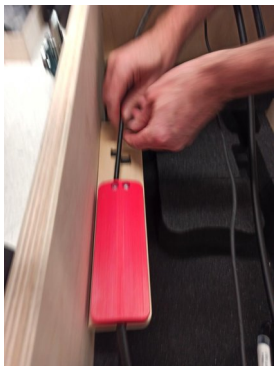
a. Cut the mooring line to the appropriate length to fit in the crate.

b. Strip both ends of the line by approximately 1cm.

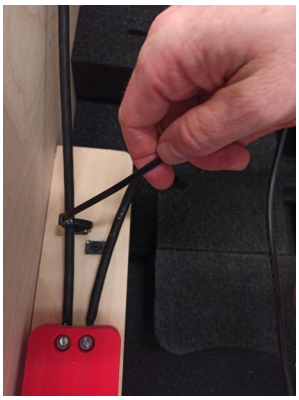
c. Attach each of the MLM lines to the MLM lines junction box. Use the screwdriver provided.

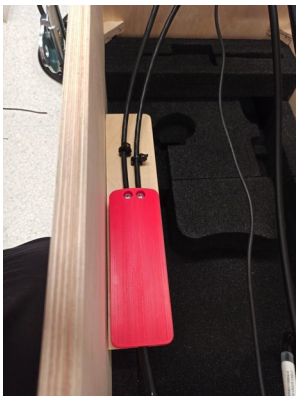






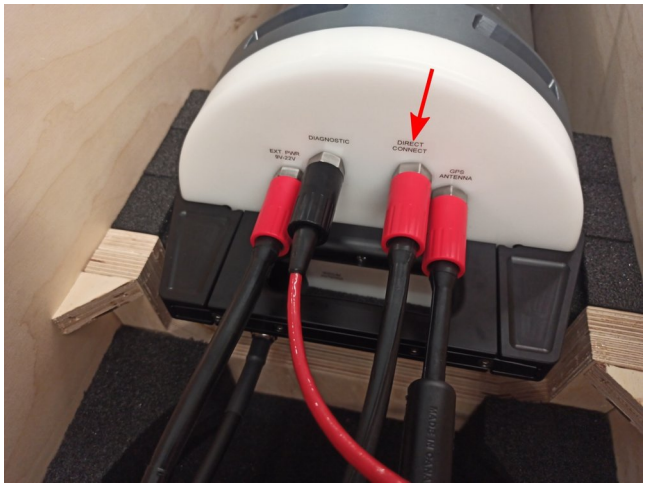
d. Secure each of the MLM lines to the crate. Use the zip ties provided.





e. Secure each of the MLM lines to the crate using their dedicated clamping slot. Use the 5mm hex key provided.

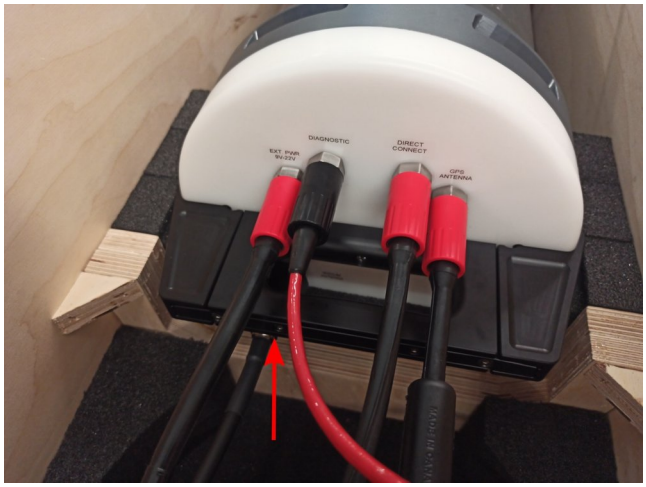
f. Attach the MLM lines junction box cable to the RBR*cervello* end-cap. Use the connector labeled "DIRECT CONNECT".



3. Install the Iridium antenna.

a. Attach the Iridium antenna to its cable.

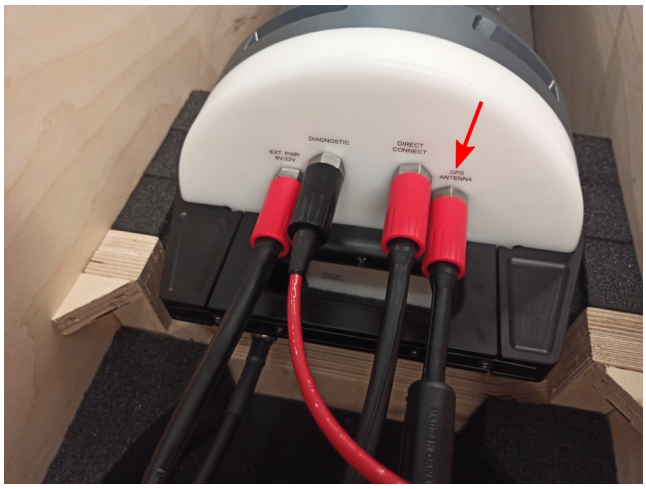
b. Attach the Iridium antenna cable to the RBR*cervello* end-cap. Use the connector labeled "IRIDIUM ANTENNA".



c. Secure the Iridium antenna cable to the crate via its dedicated clamping slot. Use the 5mm hex key provided.

4. Install the GPS antenna.

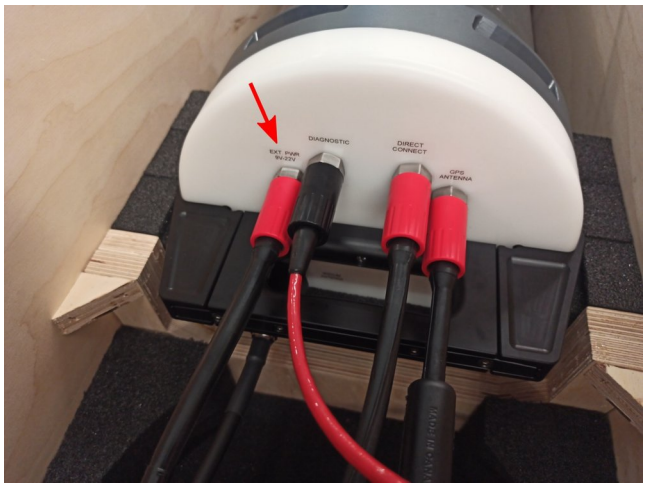
a. Attach the GPS antenna to the RBR*cervello* end-cap. Use the connector labeled "GPS ANTENNA".



b. Secure the GPS antenna cable to the crate via its dedicated clamping slot. Use the 5mm hex key provided.

5. Install the external power source (if applicable).

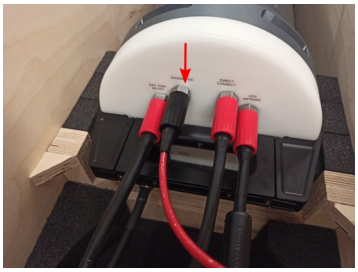
a. Attach the external power cable to the RBR*cervello* end-cap. Use the connector labeled "EXT. PWR".



b. Secure the external power cable to the crate via its dedicated clamping slot. The tab of the slot has two different sides to accommodate 6.3mm and 9.5mm cable diameters. Use the 5mm hex key provided.

Enabling the RBR*cervello*

1. Attach the provided patch cable to the RBR*cervello* end-cap. Use the connector labeled "DIAGNOSTIC".



2. Follow the steps in Using the diagnostic port to set up the terminal and access the RBR*cervello*'s menu.
3. Enable the RBR*cervello* by using the "enable" command.
4. Use the "status" command to check the status of all instruments attached to the mooring line. Verify that they are all logging.

> ⚠ The instrument status updates can take up to 20 minutes, depending on the number of instruments attached to the MLM line.

```
RBRcervello> status

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

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

Cervello status: logging
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                 : logging
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                 : logging
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                 : logging
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                 : logging
Instrument 04 internal battery: 4.90 V
Instrument 04 external battery: 0.00 V

Download remaining: 0B
```

5 Features

5.1 USB interface

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 USB mass storage devices and can be used with flash drives. Bus-powered hard drives are not recommended as the RBR*cervello* may not provide sufficient power.

Retrieving data

> *Tools required*
>
> A blank, FAT32-formatted USB flash memory stick, at least 512MB in size

To retrieve data from the RBR*cervello*, connect a FAT32-formatted USB storage device to its USB port while it is powered on and awake. All data will be copied from the data controller to the storage device.

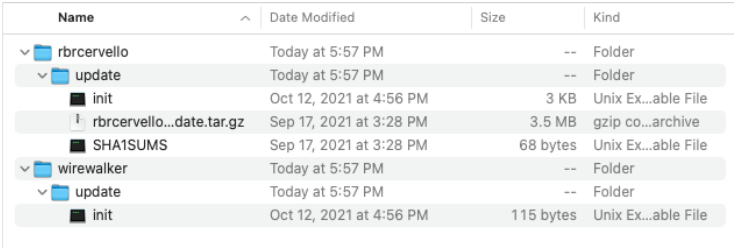
> An RBR*cervello* Antarctica could be in a deep sleep state when the USB stick is connected. Wake it up by connecting the diagnostic port and pressing "Enter" on the serial terminal. When the "RBRcervello>" prompt appears, plug in the USB stick.

> Connect only flash memory storage media (i.e., memory sticks or SSDs) to the USB port of the data controller. Attempted use of portable hard drives may damage the RBR*cervello*.

Applying updates

> *Tools required*
>
> A blank, FAT32-formatted USB flash memory stick, at least 512MB in size

Firmware and configuration updates for the data controller are delivered in the form of ZIP archives. To apply these updates, extract the archive into the root of an empty, FAT32-formatted USB storage device. The structure of all directories within the archive, including the top-level `rbrcervello` directory will be retained. After decompressing, the contents of the USB stick will resemble this structure:



Some archive extraction programs, including Safari and the macOS Finder, may modify directory structures during decompression. If this happens, either recreate directories as necessary to match the structure depicted above or use a program that does not modify contents during extraction (e.g., 7-Zip on Windows or The Unarchiver on macOS).

> Always retrieve the data from the RBR*cervello* and clear it completely before applying any updates.

As when retrieving data from the RBR*cervello*, connect the USB storage device to the USB port on the data controller.

Light indicator

During data transfer or applying updates, the red light on the RBR*cervello* will flash steadily. Once all data is transferred or updates installed, 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.

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, remove the storage device.

Once the red light has turned off and stays off for several seconds, reconnect the storage device and retry.

5.2 Data hosting

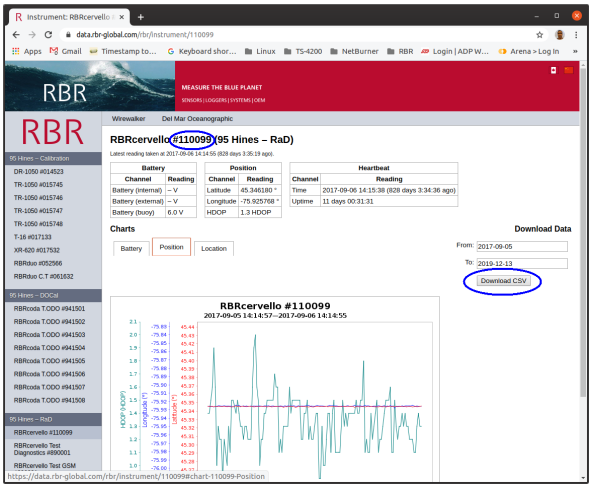
The RBR*cervello* is optionally supplied with real-time telemetry in the form of GSM cellular and Iridium Satellite modems. These send data to the RBR data hosting service, which is accessible at `https://data.rbr-global.com/*ClientReference*`. Each client site may be public or password protected. One 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*concerto*^{3} or RBR*maestro*^{3}), 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".



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.

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

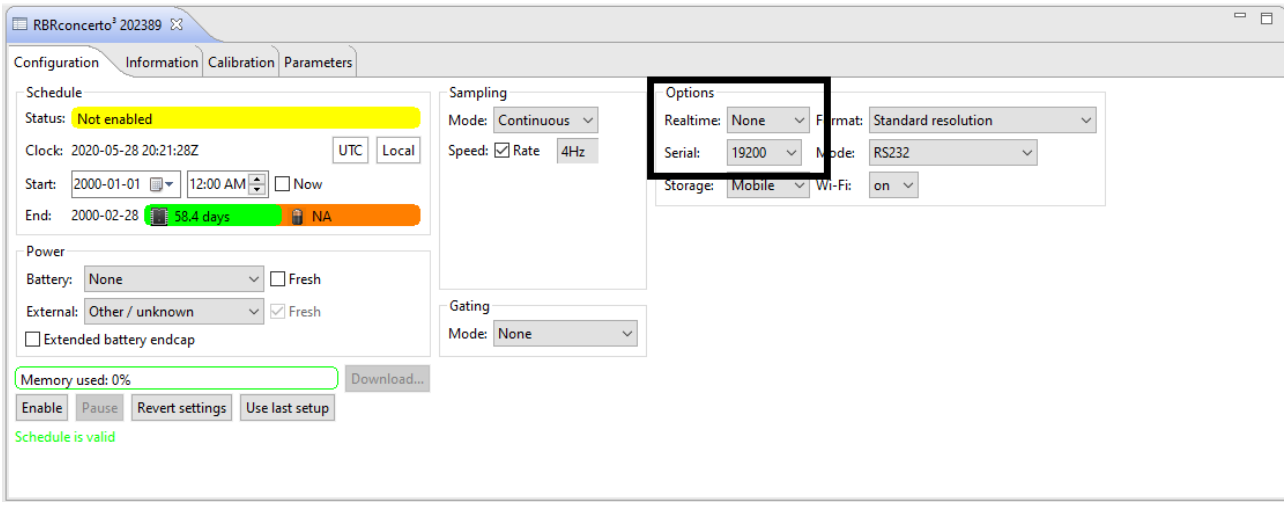
6 Troubleshooting

Because the RBR*cervello* system comprises multiple components, troubleshooting may be a multi-stage process. It involves separate procedures to verify correct functionality of each.

Follow the steps outlined in Using the diagnostic 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", run the MLM diagnostic task.
3. If the instrument status is "loggererror", ensure your logger is communicating at 19200baud (use Ruskin to check).



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

GSM

Run the GSM diagnostic task to troubleshoot common errors, such as a bad SIM-card or poor Internet connection.

Failure, bad SIM-card

```
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: N/A
IMSI: N/A
ICCID: N/A
Waiting for the modem to connect to the GSM
```

```
   network.....................................................Hit the connection
   timeout! Giving up.
17 Test failure: failed to bring up connection!
18 /usr/bin/poff: No pppd is running.  None stopped.
19 messages-read: no process found
20 netpty: no process found
21 The serial link has been brought down.
22 Powering off the modem.
23 Restarting services...
24 Starting periodic command scheduler: cron.
25
26 RBRcervello>
```

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

RUDICS

Run the RUDICS diagnostic task to identify failures in the modem, signal strength, or activation issues of the SIM-card.

Failure, no satellite signal

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

```
 1 RBRcervello> diagnose rudics
 2 Checking deployment status...
 3 Stopping services...
 4 Waiting for publisher to terminate...
 5 Stopping periodic command scheduler: cron.
 6 Starting RUDICS test
 7 I think the RUDICS modem is on /dev/ttyS4 at 2400 baud.
 8 Waiting for the modem to come up...
 9 Teraform v1.1.2 (built 2021-10-12T20:17:02+0000)
10 2021-11-12 20:42:48   src/teraform.c:364 [INFO ]: Modem: "/dev/ttyS4"
11 2021-11-12 20:42:48   src/teraform.c:367 [INFO ]: Dial number: "0088160999999"
12 2021-11-12 20:42:48   src/teraform.c:388 [INFO ]: Configuring modem...
13 2021-11-12 20:42:48   src/teraform.c:432 [INFO ]: Waiting for modem to be
   available...
14 2021-11-12 20:42:48   src/teraform.c:451 [INFO ]: Checking IMEI...
15 2021-11-12 20:42:48   src/teraform.c:451 [INFO ]: IMEI: 3001250619999999
16 2021-11-12 20:42:48   src/teraform.c:478 [INFO ]: Checking ICCID...
17 2021-11-12 20:42:49   src/teraform.c:478 [INFO ]: ICCID: 8988169763060999999
18 2021-11-12 20:42:49   src/teraform.c:488 [INFO ]: Disabling flow control...
19 2021-11-12 20:42:49   src/teraform.c:520 [INFO ]: Waiting for satellite signal...
20 2021-11-12 20:42:52   src/teraform.c:520 [INFO ]: No signal found. Waiting to try
   again.
21 2021-11-12 20:43:03   src/teraform.c:520 [INFO ]: Waiting for satellite signal...
22 2021-11-12 20:43:03   src/teraform.c:520 [INFO ]: No signal found. Waiting to try
   again.
23 2021-11-12 20:43:14   src/teraform.c:520 [INFO ]: Waiting for satellite signal...
24 2021-11-12 20:43:18   src/teraform.c:520 [INFO ]: No signal found. Waiting to try
   again.
25 2021-11-12 20:43:29   src/teraform.c:520 [INFO ]: Waiting for satellite signal...
26 2021-11-12 20:43:29   src/teraform.c:520 [INFO ]: No signal found after the final
   attempt.
27 2021-11-12 20:43:29   src/teraform.c:520 [ERROR]: Failed after 4 attempts.
28 Failed to dial connection! Giving up.
29 Test failure: failed to bring up connection!
30 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

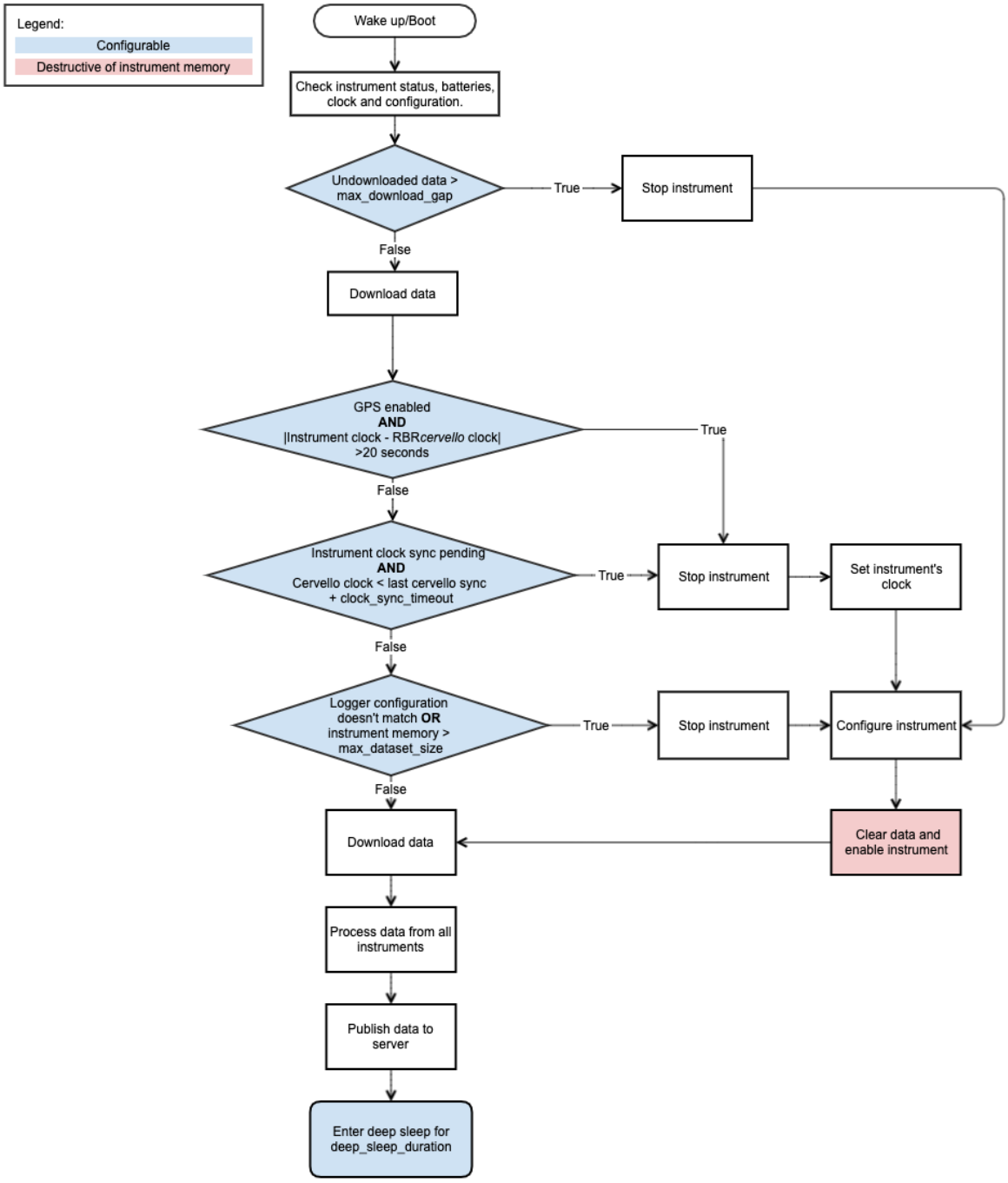
This failure usually occurs because of a bad connection with the network or when the SIM-card has not been successfully activated.

```
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>
```

7 Reference

7.1 Deployment behaviour

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



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 only 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

7.2.1.1 Name

status

7.2.1.2 Synopsis

Provide an overview of the system status.

7.2.1.3 Usage

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

status [-p|--progress]

status [-g|–gps]

7.2.1.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

-p, --progress

Show progress bars for downloads after displaying system overview. It will update indefinitely update the progress unless program is exited.

-g, --gps

Trigger and wait for a GPS acquisition before displaying the remaining of the system overview

7.2.1.5 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 RBRcervello, 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 RBRcervello 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.

7.2.1.6 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

7.2.2.1 Name

disable

7.2.2.2 Synopsis

Disable deployment.

7.2.2.3 Usage

disable [-h]

7.2.2.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

7.2.2.5 Description

Disable deployment, disable instruments, and generate a persistent flag to prevent reenabling instruments in future boot cycles.

7.2.2.6 Examples

```
RBRcervello> disable
Deployment disabled
```

7.2.3 diagnose

7.2.3.1 Name

diagnose

7.2.3.2 Synopsis

Perform in-depth tests for telemetry and MLM links

7.2.3.3 Usage

diagnose [-h|--help] | [-v|--version] [mlm|gsm|rudics]

7.2.3.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

mlm diagnose MLM telemetry

gsm diagnose GSM telemetry

rudics diagnose Iridium RUDICS telemetry

7.2.3.5 Description

mlm: Ensures correct HEM configuration and status. Discover all SSMs and configure them accordingly. Run various BER tests on the line.

gsm: Ensures connectivity through GSM to the Internet.

rudics: Ensures connectivity through Iridium RUDICS to the Internet.

7.2.3.6 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>
```

7.2.4 config-check

7.2.4.1 Name

config-check

7.2.4.2 Synopsis

RBR configuration checker.

7.2.4.3 Usage

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

config-check [-c|--config-file] [device]

7.2.4.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

-c, --config-file [device]

Show configuration file.

Where "device" is the device of which configuration will be shown.

The supported devices are the following:

instrumentXX displays the corresponding deployment configuration of instrument **XX**.

cervello displays the user configuration of the RBR*cervello*.

7.2.4.5 Description

Display in an interactive manner the current configuration of the RBR*cervello* or one of the configured instruments. Use arrow keys to scroll the configuration. Key **Q** to exit.

7.2.4.6 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"
        }

    }
}
```

7.2.5 edit-config

7.2.5.1 Name

edit-config

7.2.5.2 Synopsis

Edit the current configuration of an instrument or the RBRcervello.

7.2.5.3 Usage

edit-config [-n|–new-file] | [-c|--config-file] device

edit-config [-h]

7.2.5.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

-c, --config-file [device]

Where 'device' is the device of which configuration will be shown.

The supported devices are the following:

- **instrumentXX** allows editing the deployment configuration file of instrument **XX**
- **cervello** allows editing the RBRcervello user configuration file

-n, --new-file

Start editing from an empty file

7.2.5.5 Description

Edit interactively the configuration of a specific instrument or the RBRcervello configuration. This relies on the nano editor.

7.2.5.6 Examples

```
1 RBRcervello>edit-config -c instrument01
```

7.2.6 set-cervello-clock

7.2.6.1 Name

set-cervello-clock - sets internal clock.

7.2.6.2 Synopsis

RBR*cervello* clock setter.

7.2.6.3 Usage

set-cervello-clock [string-formatted-date]

set-cervello-clock [-h]

7.2.6.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

7.2.6.5 Description

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

> This command is intended to be used on units without a GPS, e.g. RBR*cervata*.

7.2.6.6 Examples

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

7.2.7 cervello-clock

7.2.7.1 Name

cervello-clock

7.2.7.2 Synopsis

RBR*cervello* clock checker

7.2.7.3 Usage

cervello-clock [-h|–help]

7.2.7.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

7.2.7.5 Description

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

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

7.2.7.6 Examples

```
1 RBRcervello> cervello-clock
2 Last time sync: '2021-06-28 15:31:00'
3 Current internal time: '2021-06-29 11:54:24'
```

7.2.8 reboot-controller

7.2.8.1 Name

reboot-controller - gracefully reboot the system.

7.2.8.2 Synopsis

Reboot system as if it had entered deep sleep and waken up.

7.2.8.3 Usage

reboot-controller [-h]

7.2.8.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

7.2.8.5 Description

Stop all internal services and reboot the RBR*cervello*. Useful for testing deployment behaviour.

7.2.8.6 Examples

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

7.2.9 scrub-controller

7.2.9.1 Name

scrub-controller - clears the system's storage.

7.2.9.2 Synopsis

Clear all data stored in the RBR*cervello* storage.

7.2.9.3 Usage

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

7.2.9.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

-f| --format

(Not available for RBR*cervata*)

Formats the SD card in the proper ext4 filesystem and initializes the folder structure.

7.2.9.5 Description

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

7.2.9.6 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

7.2.10.1 Name

scrub-instrument - memory eraser for instruments

7.2.10.2 Synopsis

RBR instrument scrubber.

7.2.10.3 Usage

scrub-instrument [-h|--help] | [-v|–version]
scrub-instrument [-a|–all]
scrub-instrument [instrumentXX]

7.2.10.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

instrumentXX

Erase instrument XX.

-a

Erase all instruments.

7.2.10.5 Description

Clear the internal memory of an instrument connected to the system. This command is unavailable if the RBR*cervello* status is logging.

7.2.10.6 Examples

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

7.2.11.1 Name

enable

7.2.11.2 Synopsis

Enable deployment according to internal configuration files.

7.2.11.3 Usage

enable [-h]

7.2.11.4 Options

-h, --help

Display this help message and exit.

-v, --version

Output version information and exit.

7.2.11.5 Description

Enable deployment, configure instruments, and enable them to start logging. Deletes any stopping flag for the deployment, whenever the system reboots, the deployment behaviour will continue.

7.2.11.6 Examples

```
RBRcervello> enable
Deployment enabled
```

7.2.12 help

7.2.12.1 Name

help

7.2.12.2 Synopsis

Display the RBR*cervello* diagnostic menu help.

7.2.12.3 Usage

help

7.2.12.4 Options

-h, --help

Description

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

7.2.12.5 Examples

```
RBRcervello> help
RBRcervello Menu.

These are the available commands to manage RBRcervello
   disable               Stop deployment.
   status                Provide a general overview or detailed information on
                         controller and instruments.
   diagnose              Diagnostics for MLM and telemetry.
   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                Start deployment according to internal configuration
                         files.
   reboot-controller     Reboot system as if it had entered deep-sleep and
                         waken up.
   help                  Show this menu.
```

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.