scannex

rt.buffer Getting Started Guide

2018-07-09
Firmware Version 2.00

rt.buffer Getting Started Guide

Date	*Author*	*Release*
2017-10-11	MP	For version 1.00
2017-12-15	MP	For version 1.03
2018-01-26	MP	For version 1.10
2018-07-09	MP	For version 2.00 (simplified)

Copyright © UK 2016-2018 Scannex Electronics Limited.
All rights reserved worldwide.

Scannex Electronics Ltd, UK
t: +44(0)1273 715460
f: +44(0)1273 715469

http://www.scannex.co.uk
info@scannex.co.uk

Table of Contents

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

1. Overview

The Scannex rt.buffer is a rugged, low power 3G (UMTS/HSPA+) remote telemetry unit providing a flexible solution in marine and harsh environmental conditions, with a powerful range of scripted collection, delivery and management options.

The rt.buffer is effectively a "platform" that provides fully scriptable operations for collecting, manipulating, calculating, and sending data from a variety of sensor types.

It is not "plug and play". It needs to be taught how to collect data, and told what to do with it. A number of Lua "App" scripts can be stored in the rt.buffer's flash file system, and one of those Apps can be configured to run - giving the rt.buffer its "personality".

The rt.buffer is powered by a sophisticated and efficient real-time-operating system (RTOS) that uses the Lua scripting core to provide user applications and extensions. Power consumption is kept to an absolute minimum, and can allow 5 year run-time from internal LSH20 style batteries.

1.1. Operating Modes

For battery operation, the rt.buffer stays in ultra low-power (40µA at 7.2V) sleep mode and wakes up on a variety of triggers or internal timers to collect data from external sensors. It will periodically deliver the data over the 3G network.

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

1.2. Data Collection

Data can be collected from sensors and devices such as:

- Digital pulse inputs (x2)
- Serial devices:
  - RS-232, RS-485, and SDI-12
  - ModBus RTU sensors
- Analog (24-bit ADC differential) suitable for pressure transducers etc

Complex second-resolution schedules for data collection can be easily setup using plain-text strings and the Lua scripting.

Alternatively, when connected to devices that "push" serial data, the rt.buffer can be set to wake up from ultra-low power mode to catch the data and store it.

Sample Lua Apps are available from Scannex for:

- ASCII Lines or Binary serial collection
- NMEA data handling
- Pulse + ADC sampling
- ModBus sampling over RS-485

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

1.3. Data Delivery

The rt.buffer can deliver to an IoT/Cloud server or directly to private servers in a fully transactional manner over 3G with optional gzip compression by:

- FTP and FTP+TLS Push
- HTTP or HTTPS POST[1]

Data that has been collected can also be retrieved through the USB connection manually.

By default the rt.buffer will push the data into the "/Send" folder of the IoT[2] server. However, the choice of server URLs, what data to send, and whether to use gzip compression is all completely programmable using Lua.

Data can be scheduled for delivery, or the Lua App can be programmed to push data when "interesting" data values or sequences appear.

1.4. Configuration and Setup

Initial configuration is normally performed through the USB connection, using Scannex's rtbTool application, or through the rtbAPI.dll (collectively referred to as the "USB-HID"). This provides full visibility of the internal file-structure of the rt.buffer's 128Mbyte flash storage[3].

Initially, one or more Lua apps must be loaded into the /Lua folder. Only one Lua App will be executed, but each rt.buffer can be deployed with multiple applications (so the App can be chosen immediately).

Individual configuration settings can be made over USB-HID, or a whole configuration file can be read or written with the '/Config/config.txt' file. The config.txt file is a compact human-readable ASCII format that allows for easy cloning or modification of settings.

Commands can be used over USB-HID to find out the internal state of the rt.buffer.

Additionally, the rt.buffer(s) can be remotely controlled by the Internet-Of-Things (IoT) server over the cellular Internet connection.

[1] HTTP and HTTPS to a web-server or node.js server will be available with a firmware update

[2] IoT = Internet Of Things server. e.g. FTP or HTTP server.

[3] Not replacable. The flash is manufactured with chip-on-board for maximum reliability

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

1.5. Remote Management and Updates

Nearly everything in the rt.buffer can be controlled remotely through the IoT server.

When the rt.buffer connects to the IoT server it will ask for updates. Those updates can include any of the following:

- Firmware upgrade
- Lua App upgrades
- Cellular survey requests
- Diagnostic Dump requests
- Configuration changes
- Executing arbitrary Lua code (e.g. for controlling connected devices)

1.6. Magnet Trigger

The rt.buffer includes a magnet sensor on its side. This starts the "User Process" that is designed to cover the needs of, say, an engineer that needs to get some live data from the rt.buffer in the field.

Once triggered, it will connect to cellular Internet, contact the IoT server, and push a block of information at intervals.

Back-end processes can then analyse the data block and present it as HTML so the engineer can view on their smart phone. In fact, such data could be viewed anywhere in the world (assuming the web service is made available publicly).

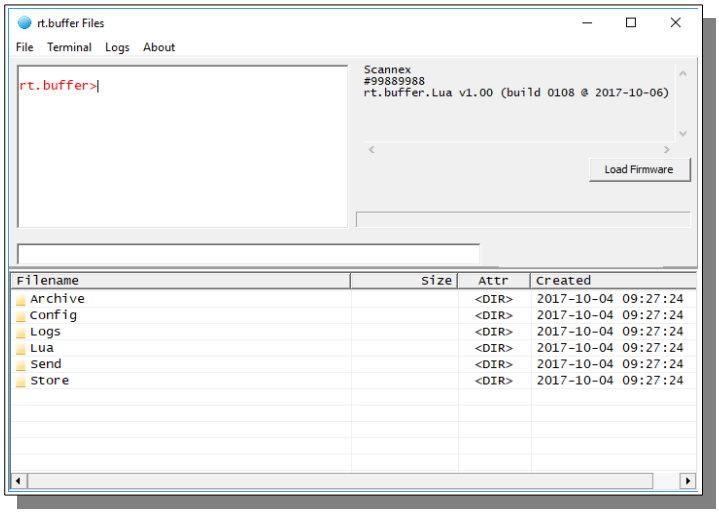
The timing, intervals, and specific action of the User Process can be overridden by writing Lua functions in the App.

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

2. Getting Started

2.1. Initial Steps

- You will need to install a valid SIM card
- Connect to the rt.buffer Engineer port with the USB cable.
  - No drivers are required, but Windows may show a pop-up from the system tray as it registers the USB-HID device.
- Run the rtbTool.exe application on Windows.
  - This application does not need to be installed, and does not require any DLLs or other libraries.
- Power up the rt.buffer (if not already powered)
- You should see the rt.buffer appear immediately - the serial number and name of the buffer will be shown in the top right window.
- Click on the terminal window and hit the ESC key. You should see a red prompt appear:
  - "`rt.buffer>`" shows the rt.buffer is running its main firmware. You should be able to see the folder structure, and can navigate to folders.
  - "`bootloader>`" shows that the rt.buffer is in the recovery boot loader mode. From this mode you can reboot, run the application, or load new firmware (either bootloader firmware or application firmware).
  - "`ate>`" shows the automated test firmware is running.



© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

- There are a set of configuration values that must be set before the rt.buffer can access the Internet and deliver data.
  - You can type the configuration values directly into the Terminal window
  - You can copy from another Windows application and use Ctrl-V to paste the changes into the Terminal window
  - You can edit the `/Config/config.txt` file

2.2. Required Configuration Values[4]

- `c.lua_app='name'`
  - Sets the Lua App name. The 'name' should be just the base-name (i.e. no extension).
  - e.g. c.lua_app='NMEA'
- `c.site_name='AcmeSite'`
  - The name of the site. This is not absolutely required, but is helpful for the scripting and naming of URLs.
- `c.cell_apn='Internet'`
  - The APN name required by the SIM contract
- `c.cell_user=''`
  - The username for the SIM Internet access (if required)
- `c.cell_pass=''`
  - The password for the SIM Internet access (if required)
- `c.cell_pin=''`
  - The PIN code for the SIM card (if set)

[4] Variable names are CASE SENSITIVE! Most are all in lower-case.

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

- `c.iot_url='ftp://user:pass@ftp.scannex.com/data'`
  - The base URL for accessing the IoT server
  - (Warning: don't use the one that's printed here!! Use your own.)

> - If username or password contain special characters (e.g. ':' or '@') then you must escape them with the Internet standard RFC-1738 section 2.2 format.

```
c.iot_url='ftp://user%40domain:pass%7e%3ard@ftp.scannex.com/data'
-- username = 'user@domain'
-- password = 'pass~:rd'
```

> - Double-quotes (") are **not** supported.

- `c.iot_job='@23:00'`
  - The job schedule for data and updates to the IoT server.

- `c.iot_upd='Update/{c.site_name}'`
  - Sets the relative path of the Update directory (i.e. the update.txt file).
  - Can be a full URL if you need a completely different server to the c.iot_url base.

> - The Lua App you have selected may also require additional configuration settings.

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

3. Using rtbTool

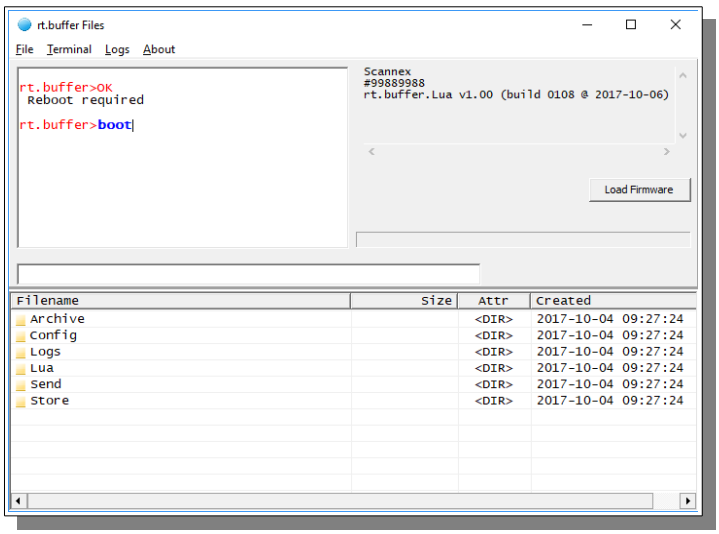
3.1. Upgrading Firmware

Firmware can be upgraded from either the BootLoader or from the main application.

> Changing firmware does not normally affect the NAND flash storage. You can upgrade bootloader or main firmware without affecting data.

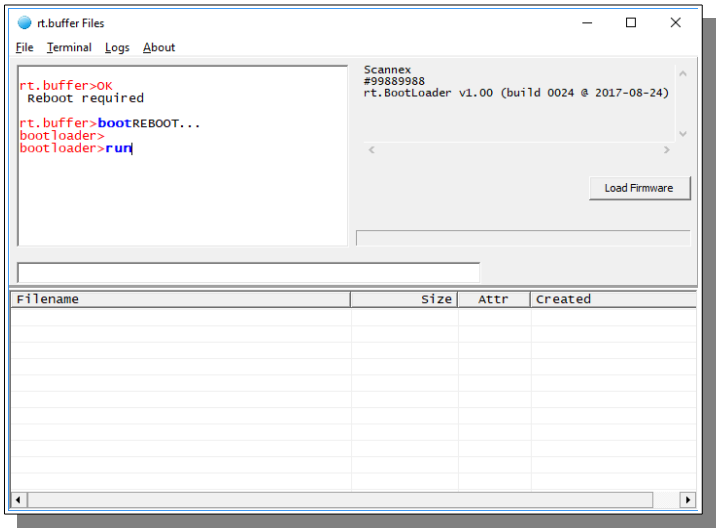
Using the rtbTool application, you can use the [Load Firmware] button.

- Verify that you have the right rt.buffer connected by checking its serial number in the top right panel.
- Click the [Load Firmware] button
- Navigate to a .BLF file (Binary Loader Format)
- Select the file, and click "Open"
- In a few seconds the display should show "Reboot required"...
- Type in "`boot`" [Return] into the terminal window...

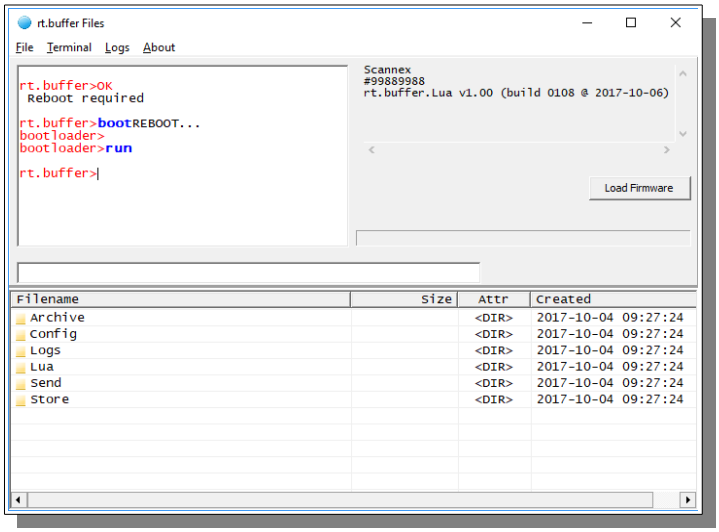


© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

- When the "bootloader>" prompt appears, type "run" [Return]



- The new firmware should be running.

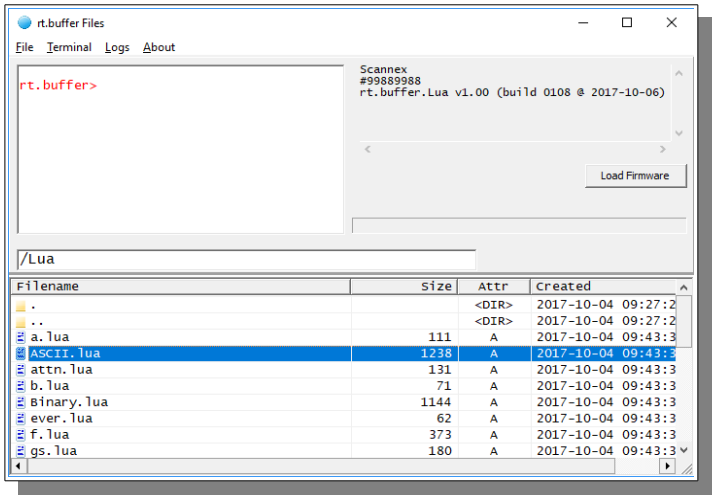


- You can check which version of firmware and bootloader are installed by typing "info" [Return] in the terminal window.

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

3.2. Loading and Viewing Applications

- Navigate to the /Lua folder
  - Double click on the yellow folder icons (".." will go up a directory level)
  - Or type in "/Lua" into the path bar and click [Return]
- You should see a list of .lua files



- Install new or updated Lua Apps:
  - Drag-and-drop from a file from a Windows Explorer window onto the file list.

> You cannot drop onto a yellow folder with the tool. You can only drop into the current directory.

- View or edit a Lua App:
  - You can just double click the file to get a convenient file editor/viewer.
- Save a file/files/directory to the PC:
  - Select the items you want to save to the PC
  - Right click and select "Save to PC"
  - Navigate to the required folder on the PC and click [OK].

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

3.3. Viewing, Saving, and Editing Files

Use the file list to navigate, and double click to allow viewing or editing of any file in the rt.buffer.

The right-mouse click context menu provides options for:

- Deleting files
- Renaming files
- Making directories
- Making empty files

> If the file has binary characters (e.g. NULL) and you hit the [Cancel] or the [Esc] key, the rtbTool may tell you that the file has changed - even if you did nothing to it. Just click to lose the changes.

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

4. Folder Structure

The following directories are used by the firmware (although other directories can be created and used within the Lua scripting):

/Archive

Files that are send to the IoT server from the /Send folder will be temporarily saved in the Archive folder.

By default, the last 32 files or 4MB are kept in this folder.

/Lua

Folder for all Lua scripts.

/Config

Configuration folder. Config.txt is the primary file.

/Logs

The log files. System.log is the main log file.

When system.log reaches 64k, it is renamed system.log.2. Log files .2, .3, and .4 are kept.

/Store

The folder where incoming data is built up in temporary files.

/Send

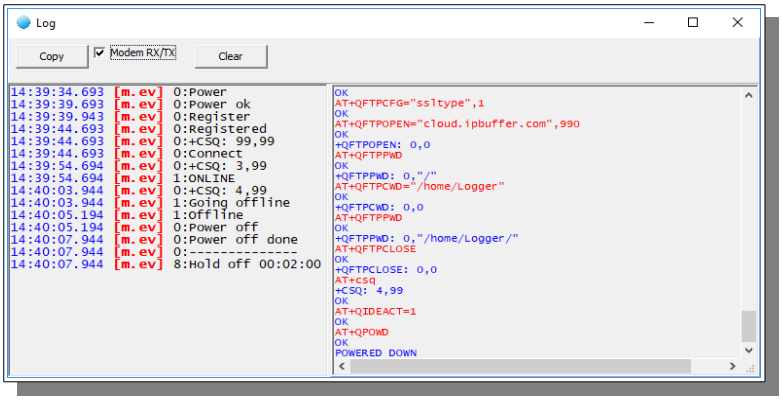
The folder where outgoing data is saved, ready to be sent over 3G. When it is time to transfer a file, or the /Store file is getting large, the file will be renamed and placed in the /Send folder.

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

5. Viewing real-time logs

The rtbTool can also show what is happening inside the rt.buffer.

- Click the "Logs" menu item
- A new window will appear that shows rt.buffer events in real-time over USB
- Checking the "Modem RX/TX" option will split the panel and also show real-time receive and transmit between the rt.buffer and internal 3G modem.
  - This is helpful if there are carrier or Internet issues.
- Pressing the [Copy] button will put onto the Clipboard:
  - The Terminal window
  - The Event log window
  - The Modem log window (if visible)
  - This data can be pasted into a word-processor or email to send back for support issues.



© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

6. Hardware

6.1. Main PCB

The rt.buffer uses a single-chip ARM Cortex-M4 processor which can run in ultra low-power sleep mode for extended battery life. It wakes from internal timer events or external triggers. Key functions:

- 128Mbyte high reliability flash storage
- RTC - updated through the terminal or over the Internet from NTP server
- 3G (HSPA+) global modem (mini PCI Express plug-in module)
  - SIM card holder fitted on the PCB
  - UMTS: 800/850/900/1900/2100MHz
  - GPRS & EDGE also supported
    GSM: 850/900/1800/1900MHz
  - PTCRB, GCF, ICASA, RCM approved
- 1 x Serial data collection port (non-isolated):
  - 300-115200 baud with optional flow control
  - Optional in-line RS485 module
  - Optional internal RS485 module
- 1 x high impedance ADC channel
  - 3V reference output
  - 24-bit resolution
  - Single-ended: 0V → 1.5V
  - Differential: ± 1.5V
- 2 x digital pulse inputs (max 64 Hz)
  - Can be used for pulse-counting and event changes
- Management port
  - USB: driverless USB-HID interface to Scannex Windows software
  - Serial port (non-isolated): 300-115200 baud with flow control
- Magnetic Switch
  - For instant local activation when in sleep mode
- LED status indicator

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

6.2. Power Supply

The rt.buffer can run from internal primary battery pack(s) and/or external DC supply and will automatically power from whichever is the highest voltage.

The options are:

Internal primary (non-rechargeable) batteries

A) 1 x lithium battery pack (2 x LSH20): 3-5 years battery life

B) 2 x lithium battery packs in parallel: 6-10 years battery life

C) 4 x alkaline LR20-cells (long life): 1 year battery life

- Above battery life figures assume:
  - Sampling every 30 secs for 1 sec
  - Transmitting once per day for approx 60 secs
    - *Note: dependent on network, and data volumes*
  - Ambient temperature 20ºC
  - The rt.buffer is not providing power to external devices

External Power: 9-28VDC (not internally isolated)

Power to the Device: Can supply 3.6V to the connected device, under Lua script control.

Battery Pack (Saft 2S1P LSH20)

- 2 x 3.6V primary LSH20 lithium-thionyl chloride D-size spiral cells, connected in series at 7.2V nominal voltage.
- Integral protection diode and thermal fuse

6.3. Construction

- Custom die-cast aluminium case
- Integral carry handle and connector protector
- IP68 to 2 metres for 72 hours
- Single compartment

6.4. Approvals

- FCC 47 SFR Part 15B-USA
- ICES 3 (B) / NMB-3 (B) - Canada
- CE RED (Radio Equipment Directive) 2014/53/EU
- RoHS

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

7. Connector Pin-Outs

7.1. Device

Mating Connector PX0410/08P or PX0400/08P

Bulgin PX0412/08S	rt.buffer Function
1	RS232: RxD input
2	RS232: CTS input
3	RS232: TxD output
4	RS232: RTS output
5	Digital Pulse Input 1
6	Digital Pulse Input 2
7	3.6V Power Output (to connected device)
8	Ground

7.2. Pressure

Mating Connector PX0410/06P or PX0400/06P

Bulgin PX0412/06S	rt.buffer Function
1	Analog Input +ve
2	Analog Input -ve
3	Reference Output (3.00V)
4	Ground
5	Screen
6	n/c

7.3. Power

Mating Connector PX0410/04S or PX0400/04S

Bulgin PX0412/04P	rt.buffer Function
1	0V
2	9-28VDC Supply (not isolated)
3	n/c
4	n/c

© UK 2016-2018 Scannex Electronics Ltd. All rights reserved worldwide.

7.4. Engineer

Mating Connector PX0410/08S or PX0400/08S

Bulgin PX0412/08P	rt.buffer Function
1	RS232: TxD output
2	RS232: RTS output
3	RS232: RxD input
4	RS232: CTS input
5	USB Device: Data-
6	USB Device: Data+
7	USB Device: Power
8	Ground

7.5. Antenna

Bulgin: PX0414	Mating Connector PX0416