

Sentinel 3 Remote Application Specification

*Sentinel 3 Remote Application Requirements.docx*

Issue: DRAFT

Date: 23rd August 2016

***Abstract:***

A brief summary of the report or proposal to help the reader quickly ascertain the paper's purpose.

|  |  |  |
| --- | --- | --- |
| **Prepared By:** |  | **Approved By:** |
|  |  |  |
|  |  |  |
| **M. Bend** |  | **B. Name** |

Shrivenham Hundred Business Park

Watchfield

Swindon

Wiltshire

SN6 8TY

UK



Tel: +44 (0) 1793 784389

Fax: +44 (0) 1793 784391

Email: [info@ppm.co.uk](mailto:info@ppm.co.uk)

Web: [www.ppm.co.uk](http://www.ppm.co.uk)

© Pulse Power & Measurement Ltd., 2016

No part of this document may be reproduced in any form without permission.

# Sign-off

|  |  |
| --- | --- |
| ***Job Title:***  C. Name | CR???? |
|  |  |
|  |  |

# Reference Documents

* Useful document 1: [Location of document 1](file:///\\ppmsrv01\engineering$\Project%20files) – issue: 01 (PPM doc)
* Useful document 2: [Location of document 2](file:///\\ppmsrv01\engineering$\Project%20files) – issue: 01 (PPM doc)

# Revision History

|  |  |
| --- | --- |
| Draft | Initial draft (03/11/15). |
|  |  |
|  |  |

# Contents

[Sign-off 2](#_Toc461714806)

[Reference Documents 2](#_Toc461714807)

[Revision History 2](#_Toc461714808)

[Contents 3](#_Toc461714809)

[1 Introduction 5](#_Toc461714810)

[a) Background 5](#_Toc461714811)

[b) Definitions 5](#_Toc461714812)

[c) Scope 6](#_Toc461714813)

[2 Detailed Requirements 7](#_Toc461714814)

[a) Application User Interface Requirements 7](#_Toc461714815)

[b) Script Engine Requirements 7](#_Toc461714816)

[c) Connection Requirements 8](#_Toc461714817)

[d) GUI Emulation Requirements 8](#_Toc461714818)

[e) Remote GUI Specific Command Set 8](#_Toc461714819)

[3 Design Notes 12](#_Toc461714820)

[a) Build Instructions 12](#_Toc461714821)

[b) Implementation Information 12](#_Toc461714822)

[b.01) Boost C++ libraries 12](#_Toc461714823)

[b.02) Sentinel 3 Controller GUI 12](#_Toc461714824)

[4 User Guide 13](#_Toc461714825)

[a) User Guide 13](#_Toc461714826)

[b) System Requirements 15](#_Toc461714827)

**List of Figures**

[Figure 1: Sentinel 3 desktop chassis with 2x Rx1, 2x Rx2, and 1x Rx6 fitted 5](#_Toc461714828)

[Figure 2 : Sentinel 3 Remote Agent with Manual Commands disabled (disconnected) 13](#_Toc461714829)

[Figure 3: Sentinel 3 Remote Agent with Manual Commands enabled (connected to a Sentinel 3 system) 14](#_Toc461714830)

[Figure 4: Options Menu 14](#_Toc461714831)

[Figure 5: File Menu 15](#_Toc461714832)

[Figure 6: Log File Copy Dialog 15](#_Toc461714833)

**List of Tables**

**No table of figures entries found.**

# Introduction

## Background

.

The Sentinel 3 system is designed to be highly configurable to suit different customer applications. In its most basic form a single input remote transmit head (Tx1) is connected via a fibre optic cable to a single input receiver module (Rx1) that is fitted into a chassis.

The system can be expanded into more complex formats using an eight input remote transmit head (Tx8), a dual input receiver module (Rx2), and a six input receiver module (Rx6). Three different types of fibre optic cable are required for each of the three receiver module variants.



Figure 1: Sentinel 3 desktop chassis with 2x Rx1, 2x Rx2, and 1x Rx6 fitted

The Sentinel 3 controller provides a touch screen interface to the Sentinel 3 system’s proprietary Graphical User Interface (GUI). Additionally, a remote application can be used to configure and use the Sentinel 3 through custom control scripts, or manual commands.

To allow users to remotely command the Sentinel 3 system without having to write custom control scripts, the remote GUI emulator can be used. This executable can be run on any PC or laptop connected to the Sentinel 3 system by either of the USB or LAN interfaces. Full emulation of local commands is supported. GPIB control can be supported with an external dongle.

## Definitions

ASCII – American Standard Code of Information Exchange

CPU – Central Processing Unit

FOL – Fibre Optic Link

GPIB – General Purpose Interface Bus

GUI – Graphical User Interface

IP – Internet Protocol

LAN – Local Area Network

MAC (address) – Media Access Control (address)

RAM – Random Access Memory

Rx - Receiver

Tx - Transmitter

UI – User Interface

USB – Universal Serial Bus

## Scope

The top level requirements for the Sentinel 3 Remote Application are as follows:

1. The Sentinel 3 remote application (hereafter referred to as ‘the Application’) shall be able to connect to a Sentinel 3 system over USB or LAN connection.
2. The Application shall allow the user to manually issue remote configuration commands to a Sentinel 3 system.
3. The Application shall allow the user to use custom control scripts to configure a Sentinel 3 system.
4. The Sentinel 3 remote command set shall include every function available to the local Sentinel 3 system GUI.
5. The Application shall emulate the Sentinel 3 system GUI, so that the connected Sentinel 3 system can be configured without requiring the user to write custom control scripts
6. The Application shall be compiled to be run on a PC running Windows XP, 7, 8, 8.1, or 10.
7. The Application shall be able to copy a log file from the Sentinel 3 system over the remote connection.

# Detailed Requirements

The top level requirements given in section 1c) can be analysed to give the following lower level, derived requirements.

## Application User Interface Requirements

1. The Application shall allow the user to specify the applicable connection settings (i.e. IP address & port for LAN connections) at runtime.
2. The Application shall connect to, and disconnect from the Sentinel 3 system on request of the user.
3. The Application shall provide a clear visual indication of whether a Sentinel 3 device is currently connected.
4. The Application shall allow the user to select and run a predefined configuration script from the filesystem of the host PC.
5. The Application shall allow the user to manually type configuration commands to be sent to the Sentinel 3 system.
6. The Application shall present the responses to any manual/scripted configuration commands in a clear, human readable form. This shall be arranged chronologically and have clear relation from the issued command to the received response.
7. The Application shall include an emulated copy of the Sentinel 3 system GUI.
8. The emulated Sentinel 3 GUI shall be capable of updating automatically, to reflect the true status of the Sentinel 3 system.
9. The Application shall present any error in a clear, human readable manner. Where appropriate, the application should endeavour to suggest possible resolution steps.

## Script Engine Requirements

1. The Application should be able to read and execute any predefined Sentinel 3 script.
2. The user shall be able to prematurely cease script execution via the application UI.
3. Sentinel 3 Scripts shall adhere to the following syntax rules:
4. Each Sentinel 3 command is to be delimited by a newline.
5. The Sentinel 3 command set is case-insensitive
6. Comments are denoted by a “#” symbol as the first character in the line. These shall be ignored by the script engine.
7. Commands to the script engine are denoted by a “$” symbol as the first character in the line. The following script engine commands shall be supported:
   1. “$PAUSE *n*” – Will pause script execution for *n* milliseconds. Omitting *n* shall present a dialog box which pauses script execution until dismissed by the user.
   2. “$REPEAT” – will continue to repeat all commands in the script prior to this line. Script execution shall be ceased by external action on behalf of the user (a “Stop Script” button).
8. Scripts should be saved as human readable text files, with the extension “.s3s”.
9. The full Sentinel 3 scripting command set is to be defined in the applicable User Handbook.
10. Script execution shall not block the main UI thread.
11. Script execution shall not disable the automatic updating of the emulated Sentinel 3 GUI.
12. Sentinel 3 commands may be manually sent to the Sentinel 3 during script execution.
13. Script execution shall gracefully cease if and when an error is encountered. Where possible, this error should be clearly stated to the user.

## Connection Requirements

1. The Application shall be able to connect to a Sentinel 3 system over a LAN connection
2. The Application shall be able to connect to a Sentinel 3 system over a USB connection.
3. Upon connection, the Application shall attempt to verify that the connected device is a Sentinel 3 system. If the connected system is not a Sentinel 3, the application shall disconnect and inform the user.
4. The Application shall gracefully handle connection errors, and shall present the user with sensible error messages.
5. Connection failures, including waiting for timeouts, shall not block the main UI thread

## GUI Emulation Requirements

1. The emulated GUI shall exactly replicate (where possible) all UI controls and actions present on the Sentinel 3 system GUI.
2. The emulated GUI shall be capable of periodically polling the connected Sentinel 3 for updated configuration data.
3. The application shall be capable of configuring the Sentinel 3 system based upon selections made by the user in the emulated Sentinel 3 GUI

## Remote GUI Specific Command Set

Given that the application must contain an emulated copy of the Sentinel 3 system GUI, the application must be able to request status information from the Sentinel 3 above and beyond what is available in the current user command set.

1. The following additional (Sentinel 3 Remote specific) command set shall be implemented in the Sentinel 3 system to allow the Application to accurately emulate the Sentinel 3 system GUI.

| Command | Arguments | Description |
| --- | --- | --- |
| S3GETBATT | None | Returns (for all 4 charging slots):  Battery is valid (1/0),  Battery charge (%),  Time to charged (minutes),  Voltage (V),  Current (mA),  Battery Type,  Serial Number,  Part Number,  Hardware version,  Firmware version,  Battery Temperature (°C). |
| S3GETINIT | None | Returns Default link configuration:  Gain,  Theta Compensation Mode,  Integrator Time Constant,  Input Impedance,  Low Noise Mode (true/false),  Power Units,  Maximum input level,  Window Tracking mode (true/false) |
| S3GETCONN | None | Returns Remote Connection information:  LAN IPv4 Address  LAN Gateway Mask,  LAN Port,  LAN MAC Address,  USB Port,  USB Driver Type,  Most Recent remote Message source,  Most Recent remote message received. |
| S3GETSYSI | None | Returns System information:  System Name  Serial Number,  Part Number,  Hardware version,  Software version,  Image date,  Build number,  Model ID,  Date & Time,  Remote or Local mode,  Config file name,  Config file version,  Config File Location,  Log file name,  Log file path. |
| S3GETRXMOD n | n = 1-6 | Returns configuration details from Rx module *n*:  FOL 1 light level (%),  FOL 1 link gain,  FOL 1 Active Tx Unit,  FOL 2 light level (%),  FOL 2 link gain.  FOL 2 Active Tx Unit. |
| S3GETTXMOD n m | n = 1-6  m = 1-6 | Returns configuration details from Tx module *m* connected to Rx module *n*:  Tx module Name,  Tx module type,  Serial Number,  Power Mode,  Battery level (%),  Theta,  Battery Serial Number,  Active Input. |
| S3GETINPUT n m i | n = 1-6  m = 1-6  i = 1-8 | Returns configuration details from input *i*, connected to Tx module *m* connected to Rx module *n*:  Input Name,  Gain,  Overdrive(true/false),  Max input level,  Integrator time constant,  Input Impedance,  Low Noise Mode (true/false). |
| S3GETALL | None | Returns the output from all other “S3GET\*”commands. |
| S3COPYLOG SIZE|n | SIZE  n = integer | S3COPYLOG SIZE returns the size of the log file, in response packets  S3COPYLOG n returns the nth packet of the log file |

1. Unless otherwise stated, each data item in the response string to the above commands shall be delimited by an ASCII Unit separator character (0x1F). For responses including data from multiple modules of the same type (i.e. a summary of all the charger slots), the ASCII Record separator character (0x1E) shall be used to delimit. Data from dissimilar modules (in, for example the S3GETALL response) will be delimited with the ASCII Group separator character (0x1D).
2. The following commands shall be added to the existing, user, command set to enable the application (and user) to completely emulate all user interface functions.

| Command | Arguments | Description |
| --- | --- | --- |
| NAME n m I name | n = 1-6  m = 1-6  i = 1-8  name = Name string | Sets the name of input *i* connected to Tx module *m* connected to Rx module in slot *n*.  Where i is omitted, the Tx name applies to the Tx module, and where m and i are omitted, the name applies to the indicated Rx module.  If n, m, and i are omitted, the name applies to the Sentinel 3 controller. |
| WAKEALL | None | Sets the power mode for all the connected Tx modules to ON. |
| SLEEPALL | None | Sets the power mode for all the connected Tx modules to SLEEP. |
| SHUTDOWN | None | Sets all the Tx modules to SLEEP mode & shuts the Sentinel 3 system down. |
| RESTART | None | Sets all the Tx modules to SLEEP and restarts the Sentinel 3 system. |
| LOGFCOPY | None | Requests the log file be copied to a connected USB drive. |
| SWUPDATE | None | Requests that the Sentinel 3 begin updating its OS from an image on a connected USB drive |

# Design Notes

## Build Instructions

The application has been built on a computer meeting the following system requirements:

* Microsoft Windows 8.1
* Intel Core i7-4790 CPU
* 8GB RAM
* Microsoft Visual Studio 2008 Professional Edition SP1
* Boost C++ libraries (<http://www.boost.org/>) Version 1.61.0, built for MSVC9.0, address-model = 64.

## Implementation Information

The application is a multithreaded MFC dialog application. The main UI thread performs most UI operations, including refreshing the Sentinel 3 emulated GUI every 0.5s. Additional threads are spawned to perform blocking, or otherwise obstructive operations.

* Connecting to the Sentinel 3 (S3ConnectThread.cpp) – this thread opens a communication port and requests identity information from the connected Sentinel 3.
* Manual Message Handling (S3Comms.cpp) – when the user sends a manual message to the Sentinel 3, the sending of the message (and receiving of the response) is handled by this thread. This ensures that (even if the communication port is busy, and the manual message has to wait) the main UI thread is not blocked.
* Emulated Sentinel 3 GUI Data Update (S3DataGathering.cpp) – this thread updates the local copy of the Sentinel 3 configuration data roughly once every 1/3rd of a second (based upon a worst case loading of 6\*Rx6s, with 6\*Tx8s each). Each data gathering process takes up to 0.1s (measured excluding network latency).
* Script handling (S3ScriptHandlingThread.cpp) – this thread implements the script processing engine, and sends each command in the script to the Sentinel 3 in turn.
* Log file download (S3DownloadLogFile.cpp) – this thread copies the log file from the Sentinel 3 device and saves it to the user’s computer

### Boost C++ libraries

The application uses the Boost C++ library implementation of a Mutex (mutual exclusion object). The application defines a mutex to limit the access to the communication ports to a single thread at a time. This means that, while there may be multiple application threads communicating with the Sentinel 3 (such as the GUI update thread, the script running thread, etc), collisions in the communication can be minimised.

C++ standard library Mutexes are not supported by the Visual Studio 2008 compiler, as they were only added in C++11 (supported in Visual Studio 2012 and newer). Therefore a third party implementation was used.

### Sentinel 3 Controller GUI

The application is required to incorporate a copy of the Sentinel 3 GUI, for the user to manipulate. As the GUI may be changed in the course of the main Sentinel 3 software development, the application has endeavoured to keep the Sentinel 3 GUI source code as separate and as unmodified as possible. Most modifications are simply to resolve changed or unneeded includes. The Sentinel 3 GUI code is located in the “S3ControllerUI” in the root project folder. Detailed difference reports, including some justification information can be found in the Git VCS commit log.

# User Guide

## User Guide

The Sentinel 3 Remote Agent software application can be used to remotely monitor and configure a Sentinel 3 system over a LAN or USB connection. The application allows users to monitor and change the configuration of a Sentinel 3 system through:

1. An emulated copy of the Sentinel 3 user interface.
2. User defined control scripts.
3. Manual command entry.



Figure 2 : Sentinel 3 Remote Agent with Manual Commands disabled (disconnected)

The Sentinel 3 Remote Agent software implements a full reproduction of the Sentinel 3 user interface (shown in Figure 3). This interface allows users to configure or monitor a Sentinel 3 system from their computer using the same functions as on the integrated controller on the Sentinel 3. All normal UI controls are reproduced, and changes made are continually synchronised between the application and the Sentinel 3 system. More detail about the Sentinel 3 UI is found in the Sentinel 3 Handbook. The remote application cannot change the Sentinel 3 system’s connection settings (such as enabling or disabling USB, and controlling the LAN port). These must be configured on the Sentinel 3 device itself.

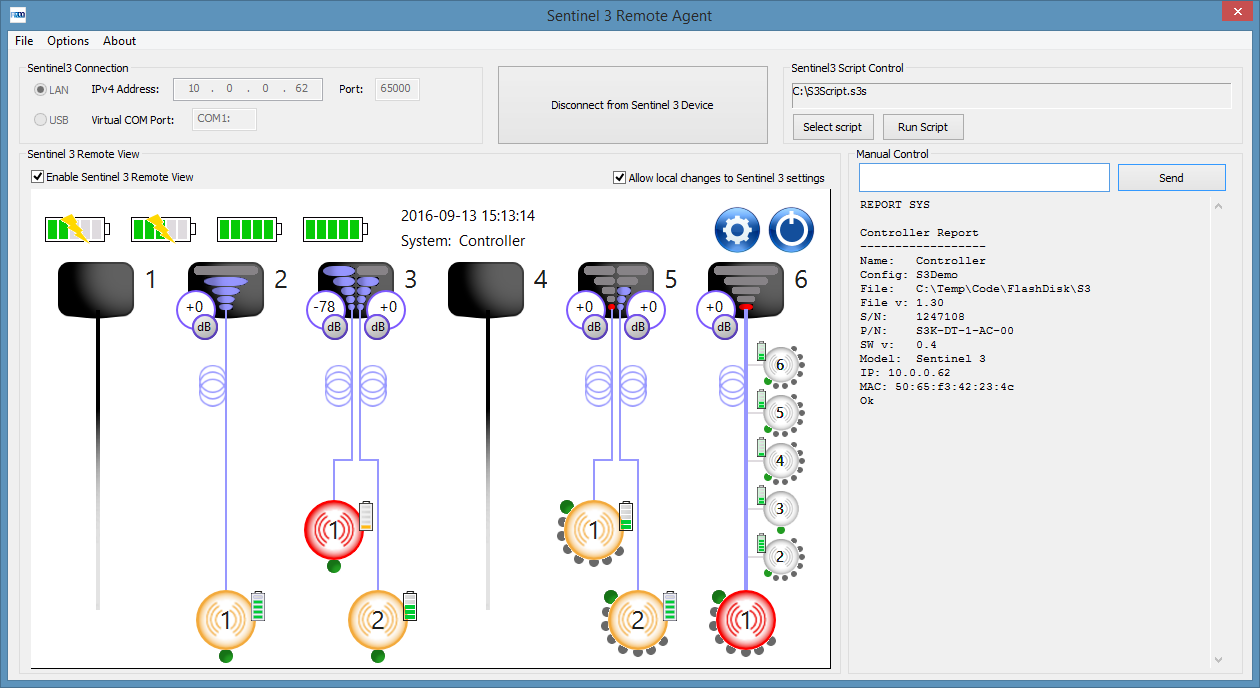


Figure 3: Sentinel 3 Remote Agent with Manual Commands enabled (connected to a Sentinel 3 system)

The application allows users to write their own command scripts (using the external control interface commands detailed in the Sentinel 3 handbook). These scripts can then be loaded and executed to quickly and repeatably configure the Sentinel 3 device. The command scripts are text files with the extension “.s3s”. Each line of the script file will be sent to the Sentinel 3 as a command, and any applicable responses will be displayed in the manual command area. Lines beginning with a “#” character will be treated as comments, and ignored by the script processing engine. In addition to the Sentinel 3 external control command set, the script engine defines two control directives (indicated with a “$” symbol at the start of the line):

* “$PAUSE *n*” – Will pause script execution for *n* milliseconds. Omitting *n* shall present a dialog box which pauses script execution until dismissed by the user.
* “$REPEAT” – will continue to repeat all commands in the script prior to this line. Script execution can then only be stopped by the user pressing the “Stop Script” button.

The application also allows advanced users to manually input commands to the Sentinel 3 system. This must be enabled in the Options menu (see Figure 4). Responses received from the Sentinel 3 system will be displayed below the command entry textbox.

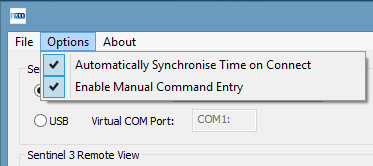


Figure 4: Options Menu

The Remote Agent software allows the user to copy the log file from the Sentinel 3 device over the USB/LAN connection (shown in Figure 5/Figure 6). This will, by default, be saved as a text file with the extension “.s3l”. The destination for the copied log file is configurable by the user. The log file will be a copy of the currently active log file on the Sentinel 3; therefore it may contain incomplete log entries at the end (these will have been in the process of writing to the log when the copy process was initiated).

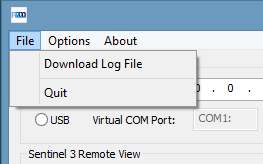


Figure 5: File Menu

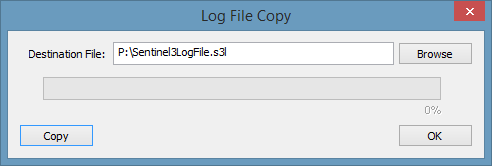


Figure 6: Log File Copy Dialog

## System Requirements

This application has been designed to run on most modern PCs.

The host PC should meet or surpass the following specifications:

* Display: Resolution greater than 1280x720 (16:9) or 1024x768(4:3).
* Operating System: Windows, XP or newer.
* USB or Ethernet connection to Sentinel 3 system.

PAGE INTENTIONALLY LEFT BLANK