USB-LCC Interface Module
for LCC®/OpenLCB™
SPROG DCC Ltd

Firmware v1.4

Hardware v1.1

Contact:

SPROG DCC Ltd
Anvil House, Butt Lane
Harbury
Leamington Spa CV33 9JL
United Kingdom
https://www.sprog-dcc.co.uk

sprog@sprog-dcc.co.uk

Date	Revision	Comments
January 2024	1	Created
February 2024	1.1	Added known issues
April 2024	1.1.1	Added known issues Added USB latency timer setting
April 2024	1.2	Update for v1.4 firmware Numerous issues fixed
May 2024	1.3	Update to align with SERVOIO-LCC version 1.5
May 2024	1.3.1	Added firmware update, fixed typos
June 2024	1.4	Clarified USB latency timer
October 2024	1.4.1	Updated module pictures

Unless otherwise notes references in this document to LCC apply equally to OpenLCB, and vice-versa.

LCC® is a registered trademark of the NMRA

OpenLCB™ is a trademark of the OpenLCB Group

Contents

1 Introduction

The USB-LCC allows easy connection between a Raspberry-Pi computer and an LCC/OpenLCB network. The USB-LCC is supported by JMRI (Java model Railroad Interface).

NOTE: This document should be read in conjunction with the USB-LCC Errata, which may be downloaded from our website, for a list of known and fixed issues for this release.

1.1 Features

- USB interface for NMRA LCC and OpenLCB™
- Galvanic isolation between USB and LCC network
- Two bi-colour status LEDs
- Supported by JMRI
- Implements an LCC node
    - LCC bootloader for firmware upgrades
        - Via host or network
    - Two configurable LED functions via the CDI
- Dual RJ45 for network pass-through or terminator
- Uses standard FTDI drivers included with Windows, MacOS, Linux
- Host side is USB bus powered
- Network side is powered from LCC

Unlike generic USB-CAN interfaces, that may also be used for LCC, the USB-LCC implements a true LCC node with its own CDI (Configuration Description Information). The bootloader allows the node firmware be updated via the USB or LCC network connections.

1.2 Electrical Specification

The USB-LCC draws a small current (50 mA max.) from the LCC PWR_POS/PWR_NEG conductors in the LCC network cable and will operate properly with a supply voltage of 7.5 V to 15 V.

The 5 V supply from the USB connector is used to power the remaining active circuitry on the USB-LCC.

	Minimum (V)	Nominal (V)	Maximum (V)	
LCC Power	7.5	12	15	
USB		5 V		

2 Installation

There are no jumper links or other configuration required to use the USB-LCC.

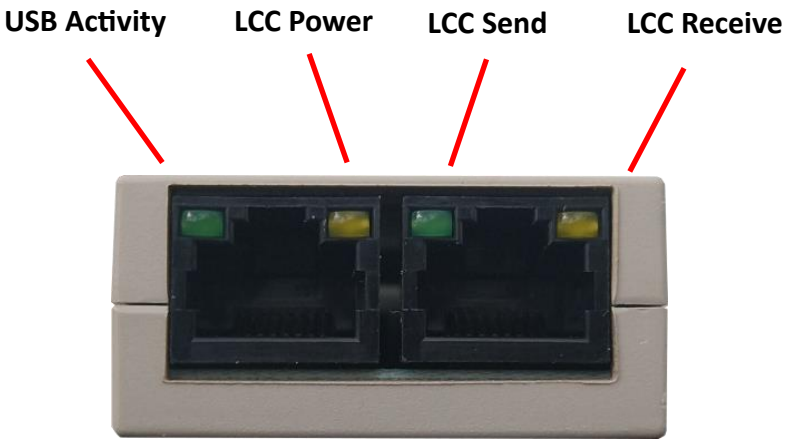
Power for the LCC interface must be available on the LCC network cable(s).

The USB-LCC can be connected anywhere along the LCC network, subject to the usual LCC cabling requirements (e.g., daisy chain connections, correctly terminated).

Four status LEDs are built in to the network connectors. One indicate USB activity, one indicates LCC power is present, and two configurable LEDs that default to LCC message send and LCC message receive.

The activity, send and receive LEDs blink briefly for each message.

The send and receive LEDS illuminate for a short period at startup.



The LCC send and receive LEDs are configurable via the CDI and default to LCC network activity (see **3.1.1 LED CONFIGURATION**).

NOTE: The LCC send and receive functions indicate valid LCC messages on both the LCC network and the USB connection.

NOTE: The LED colours may vary in the final version of the USB-LCC.

2.1 USB Virtual COM Port Latency Timer

NOTE: Ignore this section for firmware version 1.4 and later.

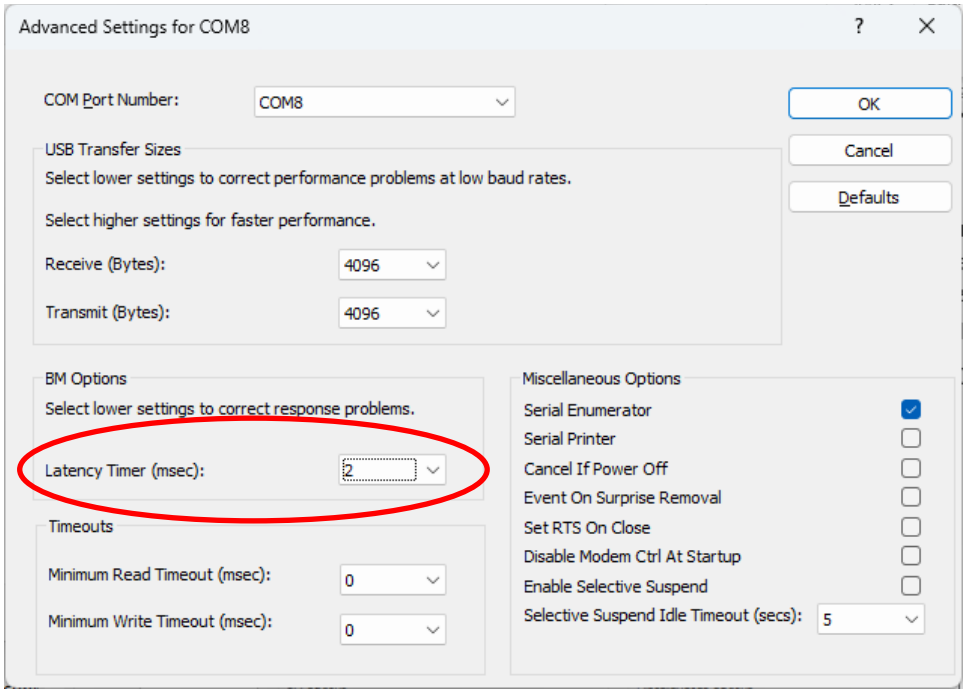
LCC messages vary in length may be shorter than the 64-byte buffer used for USB communications by the host computer. The USB interface will wait for an internal timer to expire before sending a short packet to the host. This can result in slower than expected performance when transferrin a lot of data such as reading the CDI from a module or downloading new firmware to a module.

We recommend any USB-LCC with firmware earlier than 1.4 be updated to the latest firmware.

With v1.4 firmware, further firmware updates to the USB-LCC (but not to other nodes on the network) will still be slow.

To ensure the most efficient data transfer between the LCC network and the host with firmware earlier than 1.4, the latency timer in the USB-LCC USB interface should be set to a lower value than the default 16 ms. A value of 2 ms seems to be a good compromise.

On Windows hosts, use the advanced port settings for the assigned COM port



On Linux hosts, issue the following command from a shell

```
sudo sh -c 'echo 1 > /sys/bus/usb-serial/devices/ttyUSB0/latency_timer'
```

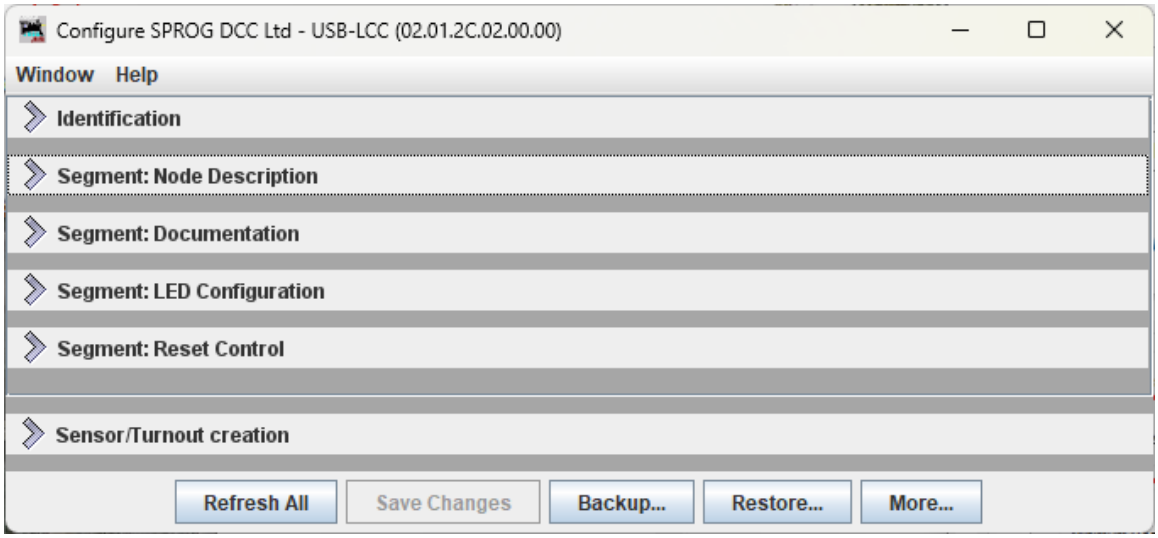
Maybe substituting ‘bash’ for ‘sh’ depending on your choice of shell.

On MacOS systems, delays of 200 ms have been observed which results in very poor performance.

3 Configuration

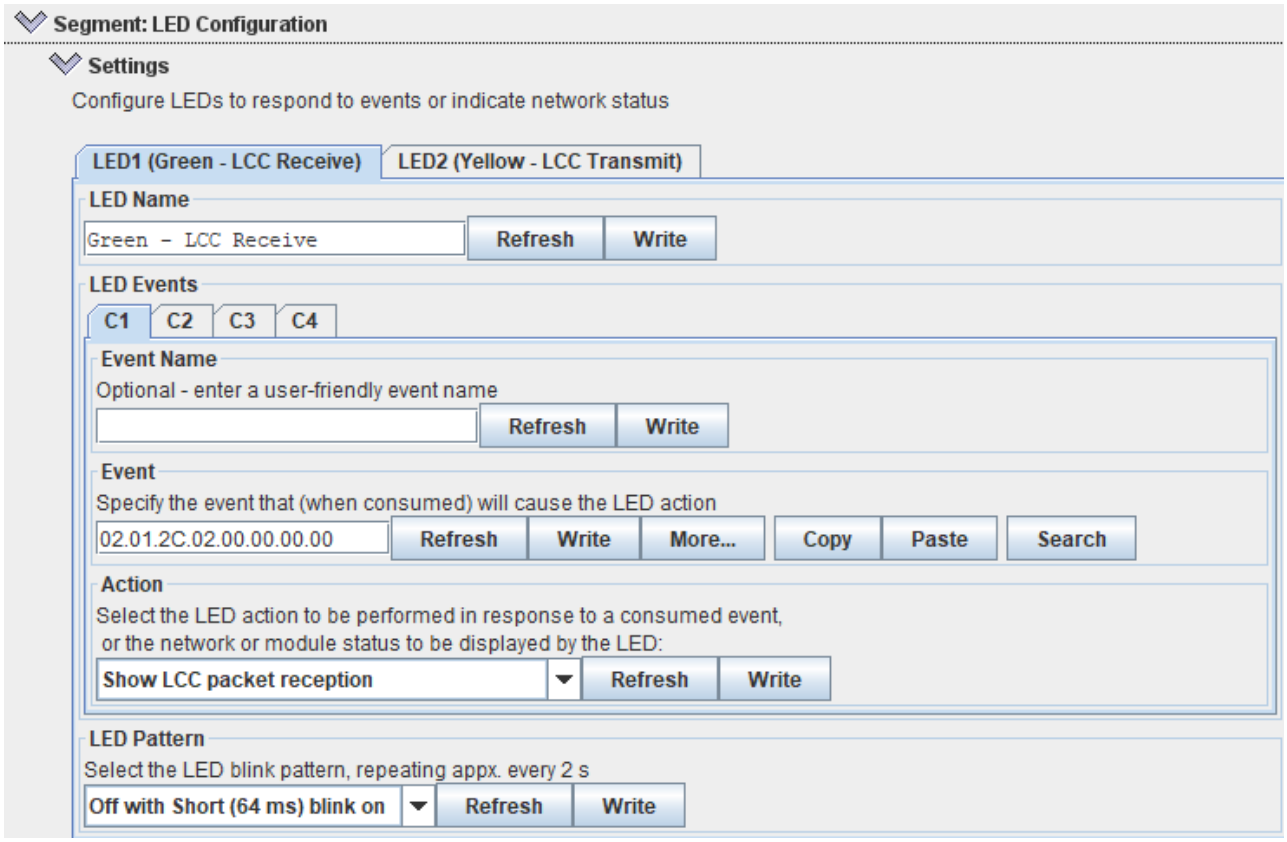
3.1 USB-LCC Configuration

The USB-LCC configuration is self-describing via the CDI (Configuration Description Information) and may be configured with suitable software tools such as the configuration dialog in JMRI.

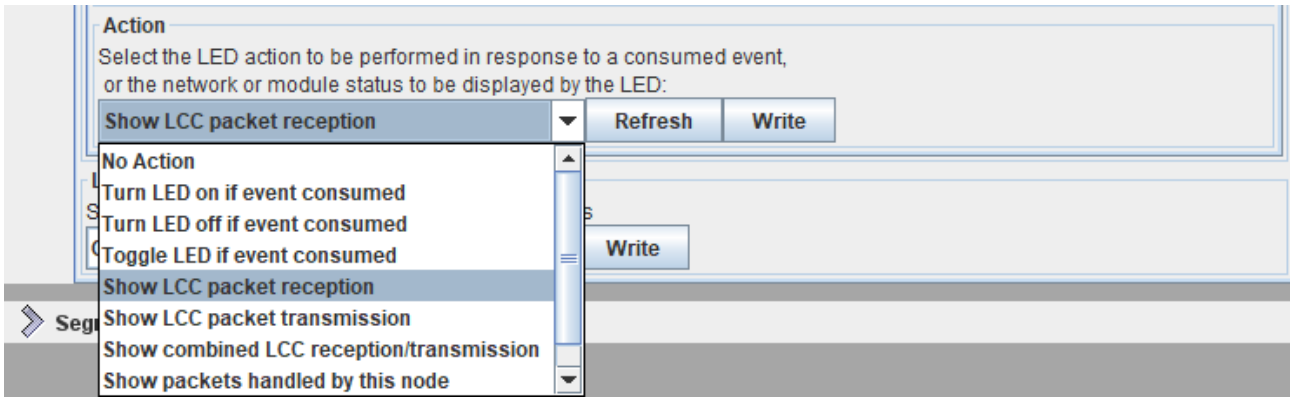


3.1.1 LED Configuration

Two of the status LEDs are configurable via the CDI. Each LED is controlled by up to four consumer events or internal status.



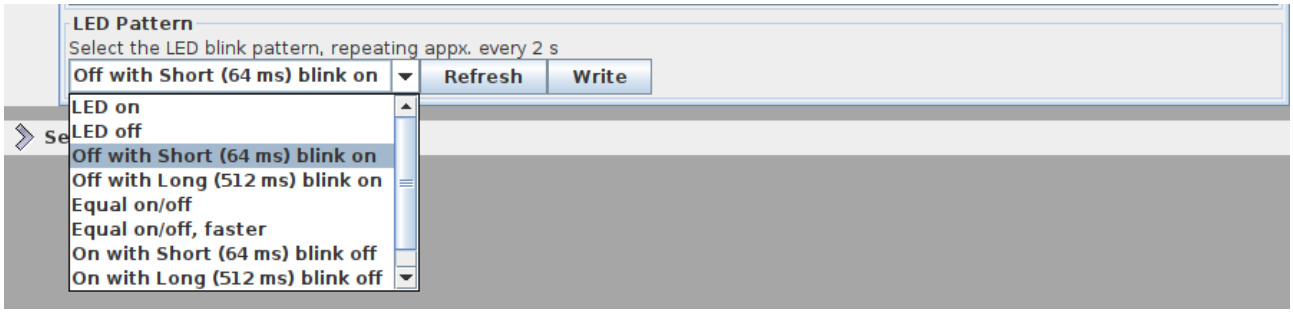
The CDI Action field controls the LED functions



LED Action	LED function
No Action	LED remains off
Turn LED on if event consumed	The LED displays the selected pattern
Turn LED off if event consumed	The LED turns off
Toggle LED if event consumed	The LED toggles between off and displaying the selected pattern
Show LCC packet reception	The LED blinks briefly on each LCC message received on the LCC Network or USB
Show LCC packet transmission	The LED blinks briefly on each LCC message sent by the USB-LCC on the LCC network or USB
Show combined LCC reception/transmission	The LED blinks briefly for each LCC message sent or received
Show packets handled by this node	The LED blinks briefly for a received message that is processed by the node (e.g. an event is consumed)
Indicate fault in the module	The LED indicates an error in the module (more details TBD)

The LED Pattern CDI field controls the way the LED is illuminated

There is one pattern setting available per LED. The LED displays the selected patter when turned on (or toggled from off to on) by a consumed event.

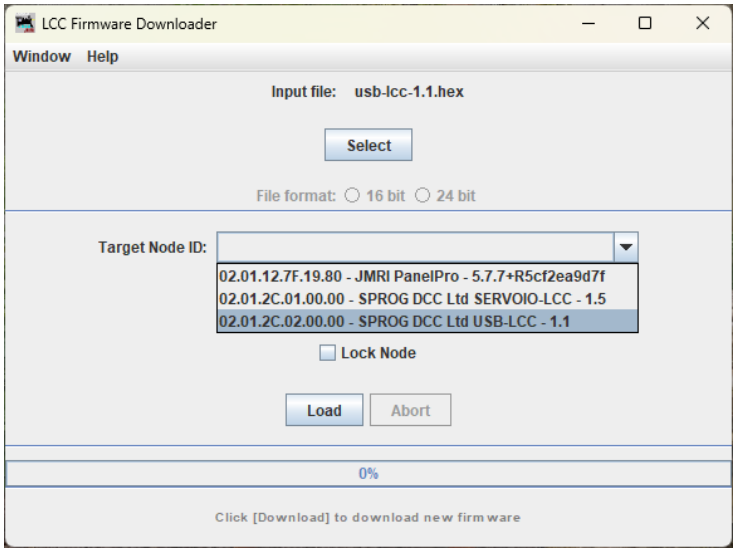


4 Firmware Updates

The USB-LCC supports firmware updates using the JMRI LCC Firmware Update tool.

Firmware updates will be announced on the SPROG DCC Ltd website and the new firmware file will be available for download.

Select the .hex file, select the target node ID and clisk the load button.



The USB-LCC will reboot when the firmware update is complete.

NOTE: It is recommended that you create a backup of the USB-LCC configuration using the node configuration tool. Occasionally it may be necessary to release firmware updates that "break" the configuration. If this is the case, it will be noted on the firmware download page. The old configuration can be restored, from the backup, after the firmware update is completed

5 Links to Further Information

SPROG DCC Ltd website https://www.sprog-dcc.co.uk For all our products and support.

SPROG DCC Ltd Official YouTube Channel https://www.youtube.com/@sprogdcc

OpenLCB group https://openlcb.org The group behind the OpenLCB/LCC standards.

NMRA LCC standards page https://www.nmra.org/lcc The LCC standards adopted by the NMRA.

OpenLCB discussion group https://groups.io/g/openlcb/topics Discussion of OpenLCB topics, more developer focussed.

The NMRA's LCC user group https://groups.io/g/layoutcommandcontrol/topics a good starting point for asking questions of other LCC users.

JMRI users https://groups.io/g/jmriusers/topics JMRI software topics.

JMRI website https://www.jmri.org Download the latest JMRI releases and access support pages.

Book: Introduction to Layout Command Control https://www.amazon.co.uk/Introduction-Layout-Command-Control-Practical/dp/0988825902 focussed on RR-Cirkits products but the concepts are applicable to any LCC hardware.

