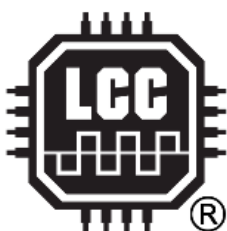




# SERVOIO-LCC Module for LCC®/OpenLCB™ SPROG DCC Ltd



Firmware v1.4



Copyright © March 2024 SPROG DCC Ltd.

This document may be distributed it under the terms of Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), version 3.0 or later.

**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](mailto:sprog@sprog-dcc.co.uk)

Date	Revision	Comments
February 2024	1.0	Created
March 2024	1.1	Fixed some known issues, see <a href="#">9 KNOWN ISSUES</a>
March 2024	1.2	Corrected Jumper numbering
March 2024	1.3	Added Test Mode
April 2024	1.3.1	Clarify associated firmware version Add new issues for 1.3 firmware
April 2024	1.4	Update for firmware version 1.4 Numerous issues fixed Added toggle mode for servo outputs

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.....	6
1.1	Features.....	6
1.2	Electrical Specifications .....	7
1.2.1	Simple I/O.....	7
1.2.2	Optional High Power Drivers.....	7
2	Installation.....	8
2.1	LCC Network.....	8
2.2	Blue/Gold Buttons and LEDs.....	8
2.3	Hardware Setup.....	8
2.3.1	I/O Bank A and Bank B Terminal Blocks .....	8
2.3.2	Servo Headers .....	8
2.3.3	I/O Drive.....	9
2.3.4	Power Distribution .....	11
2.4	Example Setups .....	14
2.4.1	Simple I/O.....	14
2.4.2	Servos.....	14
2.4.3	High current I/O .....	14
3	Basic Concepts.....	14
3.1	Active-low v. Active-high I/O .....	14
3.2	Connecting Inputs .....	15
3.3	Connecting Outputs.....	15
4	Configuration using the CDI (Configuration Description Information).....	17
4.1	Identification .....	17
4.2	Segment: Node Description .....	18
4.3	Segment: Bank A (J1) and Bank B (J19) .....	18
4.3.2	Consumers.....	21
4.3.3	Producers .....	22

4.4	Interaction of Events.....	23
4.4.1	Consuming Produced Events.....	23
4.4.2	Secondary Events .....	24
4.4.3	Output Events.....	24
4.4.4	Cascade Events (The Yard Throat Problem) .....	24
4.5	Segment: Servo on Bank A .....	24
4.5.1	Servo Startup action.....	26
4.5.2	Multiple Servo Control .....	26
4.5.3	Servo Power Down .....	26
4.5.4	Servo Events .....	26
4.6	Segment: Buttons .....	26
4.7	Segment: LED Configuration.....	27
4.8	Segment: Power Monitor .....	29
4.9	Segment: Reset Control.....	30
5	Using ButtonLed I/O .....	30
5.1	The ButtonLed Test Board .....	31
6	Servo Setup Mode .....	31
7	Test Mode .....	32
7.1	Operation in Test Mode.....	32
7.2	Reboot to Test Mode .....	32
7.3	Verify LED or Channel Output Operation .....	33
7.4	Verify Switch or Channel Input Operation.....	33
7.5	Exit Test Mode .....	33
8	Firmware Updates .....	33
9	Known Issues in This Release .....	33
9.1	Power Monitor .....	33
10	Issues fixed in This Release .....	33
10.1	SNII response if buffer full uses wrong error code.....	33
10.2	CDI XML fails with strict parser .....	33
10.3	Accessing Configuration Memory .....	33
10.4	Button Event Identification.....	33
10.5	ACDI Memory Space Layout .....	33
10.6	Memory Read Requests Flags .....	34

10.7 ButtonLED I/O.....34

10.8 Blue Button Only Produces Off (Button Up) Events ..... 34

10.9 Servo Position reporting Events .....34

11 Links to Further Information ..... 34

## 1 Introduction

To be successful in using the SERVOIO-LCC you should be familiar with basic LCC topics, e.g. see the Introduction to LCC/OpenLCB <https://www.sprog-dcc.co.uk/downloads/lcc/lcc-basics.pdf> on our website.

You will also need some familiarity with the JMRI software to connect to, and configure the SERVOIO-LCC. A computer connection is not required for operation.

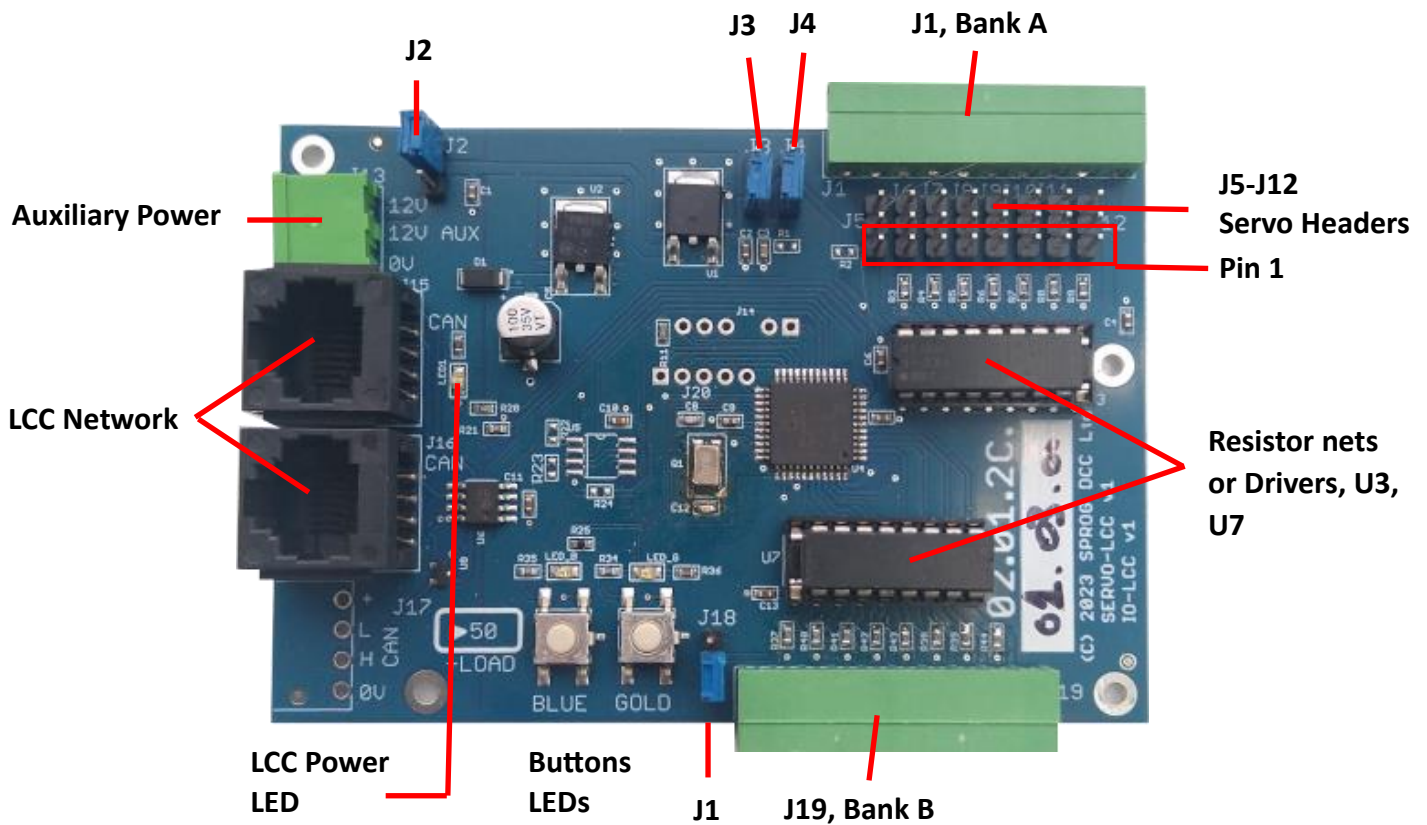
### 1.1 Features

The main features of the SERVOIO-LCC are

- Two I/O banks of eight channels for a total of sixteen channels
- Eight servo outputs on bank A only
  - Three-pin headers for servo outputs on bank A
  - Event generation for servo start-, mid- and end-points
  - Servo positions saved between power cycles
- Eight individual logic level I/O channels per bank
- Optional high voltage (50 V max)/high current (500 mA max per channel) output per bank
- Jumper selectable pull-up/down for I/O channels
- Flexible configuration through the CDI (Configuration Description Information)
- Two status LEDs configurable through the CDI
- Two pushbuttons configurable through the CDI
- Can consume own generated events for event chaining (secondary events)
- Dual RJ45 for network pass-through or terminator
- Powered from the LCC network for simple I/O
- Auxiliary power input for servos and high-power option
- Bootloader for firmware updates

When using servos on bank A, all I/O channels on bank A are servo control outputs. They may not be used for other output functions, nor as inputs.

When using the high current drives on a bank, all output channels in the bank are outputs. They may not be used as inputs.



## 1.2 Electrical Specifications

The SERVOIO-LCC uses two sources of power: from the LCC network and from the auxiliary power connector.

For simple logic-level I/O, the power can be taken from the LCC network.

For higher powered I/O using the optional driver chips, or for servo operation, power must be supplied via the auxiliary power connector in addition to the LCC network.

The SERVOIO-LCC consumed less than 50 mA from the LCC network. Additional loads taking power from the network must be added to this when calculating the total power budget for the LCC network.

### 1.2.1 Simple I/O

Each I/O channel can source or sink 50 mA, absolute maximum. The total allowed across all I/O is 350 mA, absolute maximum. In practice, currents should be kept much less than these limits. LEDs, for example, are usually bright enough with just a few mA of current.

### 1.2.2 Optional High Power Drivers

When the optional Toshiba TDB62083APG are used the maximum current per I/O channel is 500 mA but this must be reduced when multiple output are driving simultaneously, to limit the overall power dissipation within the device. Please consult the TDB62083APG data sheet.

## 2 Installation

### 2.1 LCC Network

The SERVOIO-LCC has two LCC connectors, allowing for the required “daisy-chain” network connection between nodes.

### 2.2 Blue/Gold Buttons and LEDs

The LCC specification allows for very simple node configuration using the so-called Blue/Gold buttons and LEDs. The SERVOIO-LCC does not support this and all configuration is done through the CDI (Configuration Data interface) with a suitable software tool, such as JMRI (Java Model Railroad Interface).

The Blue/Gold buttons may, instead, be configured through the CDI to produce or consume events or display status. See [2.2 Blue/Gold Buttons and LEDs](#).

### 2.3 Hardware Setup

#### 2.3.1 I/O Bank A and Bank B Terminal Blocks

The two 10-way pluggable terminal blocks, J1 for Bank A and J19 for Bank B have the same pinout, as shown in the table. Pins are numbered from left to right when looking into the board mounted portion of the connector.

See [2.3.4 POWER DISTRIBUTION](#) for the derivation of the +Volts on the I/O bank connectors.

Pin	Function
1	+Volts
2	Channel 1
3	Channel 2
4	Channel 3
5	Channel 4
6	Channel 5
7	Channel 6
8	Channel 7
9	Channel 8
10	0V

#### 2.3.2 Servo Headers

The servo headers are connected to I/O bank A. Normal digital I/O and servos cannot be mixed within bank A.

Channel assignments are as shown in the table

Header	Servo channel
J12	1
J11	2
J10	3
J9	4
J8	5
J7	6
J6	7
J5	8

Each servo header has 3-pins. Pin 1 is the furthest from the board edge.

Pin	Function
1	Control pulse
2	5V
3	0V

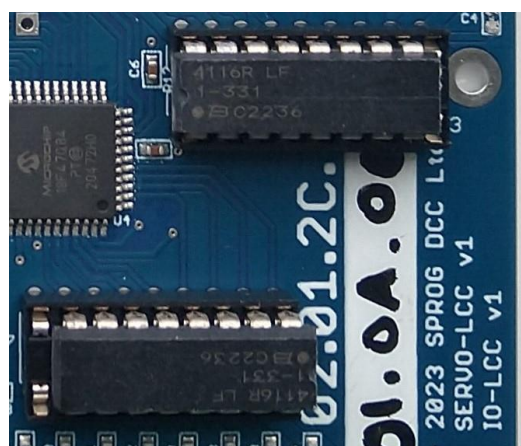
The following sections describe the functions in more detail.

### 2.3.3 I/O Drive

#### 2.3.3.1 Resistor Networks

SERVOIO-LCC is normally supplied with 330 ohm resistors installed in the two DIL (Dual-in-Line) sockets U3 and U7. These help to protect the I/O circuits of the on-board processor.

**NOTE:** The orientation of the resistor networks does not matter (i.e., they can safely be rotated 180 degrees) but they must be fitted in the correct position in the socket. U3 must be fitted in the 16 holes closest to the LCC network connectors. U7 must be fitted in the 16 holes furthest from the LCC network connectors.



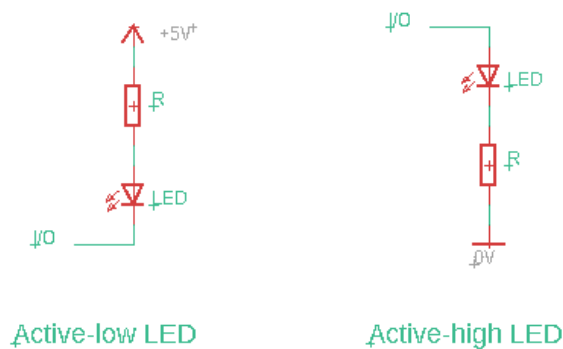
### 2.3.3.2 Optional High-power Drivers

Optionally, Toshiba TDB62083APG devices may be installed at either or both of U3 and U7. These are “low-side” or “open-drain” drivers with protection diodes and can be used when it is desired to driver higher power loads such as lamps or relays. The TDB62083APG can sink up to 400 mA on a single channel, or 50 mA if all channels are to be used at the same time.

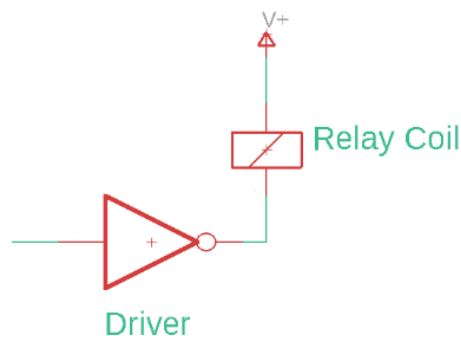
**NOTE:** TDB62083APG will invert the sense of the output from the SERVOIO-LCC, e.g. turning an output channel on will drive a low voltage on the output channel. This can be changed, if required, using the I/O polarity configuration, see [0](#)

Some examples are shown in the example schematics:

#### Driving LEDs



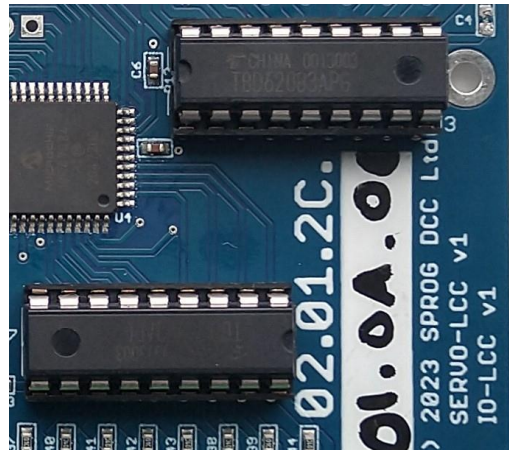
#### Relay with Optional driver



**NOTE:** The driver includes the necessary “freewheel” diode when driving inductive loads such as relays.

Configuration.

**NOTE:** The TDB62083APG must be fitted in the correct orientation. For U3, pin 1 is closest to the LCC network connectors. For U7, pin 1 is furthest from the LCC network connectors.

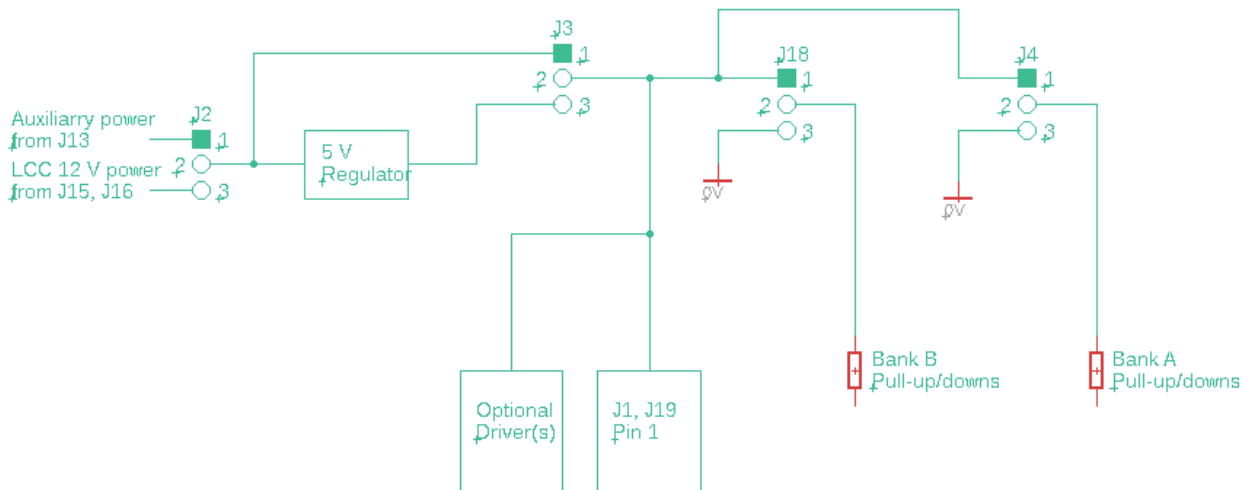


### 2.3.4 Power Distribution

Four 3-pin jumpers J1, J3, J4 and J18 are used to control the power distribution on the SERVOIO-LCC. These are shown below and on the PCB diagram in [1.1 FEATURES](#).

Jumper	Function	1-2	2-3	Comment
<b>J2</b>	VP select	J13 Aux	CAN	Voltage source for servo 5V regulator and I/O: Fit 1 – 2 for power from the auxiliary power connector, J13 Fit 2 – 3 for power from the LCC network
<b>J3</b>	I/O Voltage	VP	VServo	Voltage source for I/O drivers (if fitted) and pull-ups Fit 1 – 2 for power from the source selected by J2 Fit 2 – 3 for regulated 5 V power derived from the source selected by J2
<b>J4</b>	Bank A	Pull-up	Pull-down	Select I/O pull-up or pull-down for Bank A (J1 and J5 – J7 servos) Fit 1 -2 for pull-up resistors on bank A Fit 2 -3 for pull-down resistors on bank A
<b>J18</b>	Bank B	Pull-up	Pull-down	Select I/O pull-up or pull-down for Ban B (J19) Fit 1 -2 for pull-up resistors on bank B Fit 2 -3 for pull-down resistors on bank B

**NOTE:** Pin 1 of each jumper is annotated by the square pad on the underside of the circuit board.





2.3.4.1 I/O Power

Power for the SERVOIO-LCC I/O channels may be sourced from the LCC network or the auxiliary power connector.



The LCC network power connector must only be used when the node is driving light loads such as other electronic modules or LEDs. You must calculate the total power requirements of your LCC network and ensure sufficient power is available.

In all other cases you should supply auxiliary power to J13 from an independent power supply (i.e., not the same as the LCC network power).

J2	I/O Power Source	
Jumper 1 to 2	Power is taken from the auxiliary power input on J13	
Jumper 2 to 3	Power is taken from the LCC network	

2.3.4.2 I/O Voltage Selection

The voltage supplied to the I/O channels is selected by J3:



J3	I/O Voltage Selection	
Jumper 1 to 2	I/O voltage is the same as selected at J2	
Jumper 2 to 3	I/O voltage is regulated to 5 V	

**NOTE:** J3 must be set to supply 5V when using servos on bank A

**NOTE:** Bank A and bank B terminal blocks, pin 1, are connected to the voltage selected by J3.

#### 2.3.4.3 Bank A Pull-up/Pull-down



Bank A I/O may have 10k ohm pull-up or pull-down resistors on each channel, selected by J4:

J4	I/O Termination	
Jumper pins 1 to 2	Pull-up	
Jumper pins 2 to 3	Pull-down	

**NOTE:** Pull-ups will pull-up to the voltage selected by J3.

#### 2.3.4.4 Bank B Pull-up/Pull-down

Bank B I/O may have 10k ohm pull-up or pull-down resistors on each channel, selected by J18:

J18	I/O Termination	
Jumper pins 1 to 2	Pull-up	
Jumper pins 1 to 3	Pull-down	

**NOTE:** Pull-up will pull-up to the voltage selected at J3.

## 2.4 Example Setups

### 2.4.1 Simple I/O

For simple low current I/O such as driving or sensing other electronic modules or driving LEDs, power may be taken from the LCC network and resistor networks used at U3 and U7.

The choice of I/O voltage will be dictated by the external circuitry. For example, choosing 5 V will allow a single LED to be driven (with a suitable current limit resistor). Choosing LCC power will allow a number of LEDs to be connected in series (with a suitable current limit resistor).

Whether pull-up or pull-down resistors are appropriate will depend on the external circuitry.

### 2.4.2 Servos

To use servos on bank A the power must be drawn from the auxiliary supply, not the LCC network and resistor networks should be fitted at U3 and U7.

### 2.4.3 High current I/O

For high current I/O such as lamps, motors or relays TDB62083APG drivers must be fitted at U3 and/or U7 and power must be drawn from the Auxiliary supply.

The TDB62083APG are “open-drain” and the load should be connected between the Bank I/O and the positive supply. J4 and/or J18 should be set for pull-up (jumper pins 1 – 2).

The TDB62083APG include protection diodes for driving inductive loads such as relays or motors.

## 3 Basic Concepts

### 3.1 Active-low v. Active-high I/O

In the context of the I/O channels, the input or output can be described as being “low or “high”.

A signal that is low is at or close to 0 V when compared to the 0 V connection of the I/O bank.

A signal that is high is at a higher voltage, compared to the 0 V connection of the I/O bank, e.g., 5 V.

An active-low signal is defined to be active, or on, when low and inactive (off) when high.

An active-high signal is defined to be active, or on, when high and inactive (off) when low.

As an example, consider an input channel connected to one side of a pushbutton, or switch, with the other side of the pushbutton connected to 0 V. When the button is pushed, or the switch is closed, the input will be low. If this is defined to be the active or on state then the input is active-low.

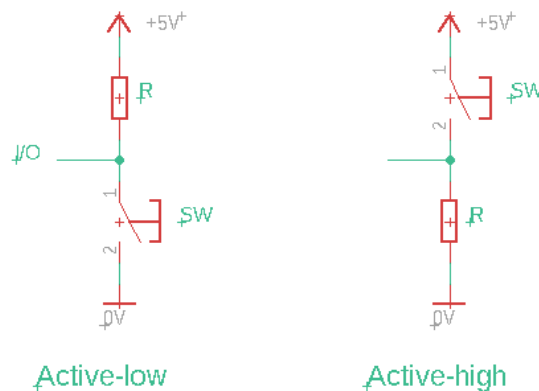
Internally, all I/O channels are treated as on or off (or toggling between on and off states), active high, as set by LCC events. The behaviour of the I/O connections can be modified using the CDI.

### 3.2 Connecting Inputs

Inputs may be inverted through the CDI polarity option to create active-low inputs.

Input	Input polarity set in CDI	Internal state
Low	Normal	Off
Low	Inverted	On
High	Normal	On
High	Inverted	Off

Example input connections are shown in the sample schematic:



The component marked SW is drawn as a switch but this could be, for example

- A physical switch or pushbutton
- Contacts operated by a relay or reed switch
- An electronic switch such as a transistor
- The output switch of an opto-coupler

### 3.3 Connecting Outputs

Similarly, outputs may be inverted through the CDI polarity option to create active-low outputs.

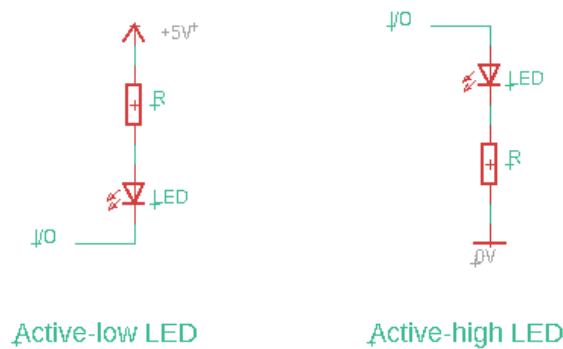
Internal state	Output polarity set in CDI	Input
Off	Normal	Low
Off	Inverted	High
On	Normal	High
On	Inverted	Low

**NOTE:** Using the TDB62083APG drivers for high current outputs inverts the polarity of the output. The CDI polarity setting can be used to correct for this, if required

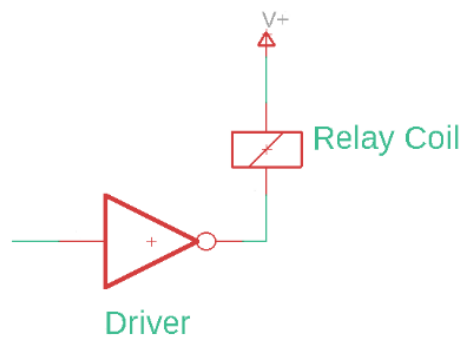
Internal state	Polarity	Output state with TDB62083APG
Off	Normal	High
Off	Inverted	Low
On	Normal	Low
On	Inverted	High

Some examples are shown in the example schematics:

Driving LEDs



Relay with Optional driver



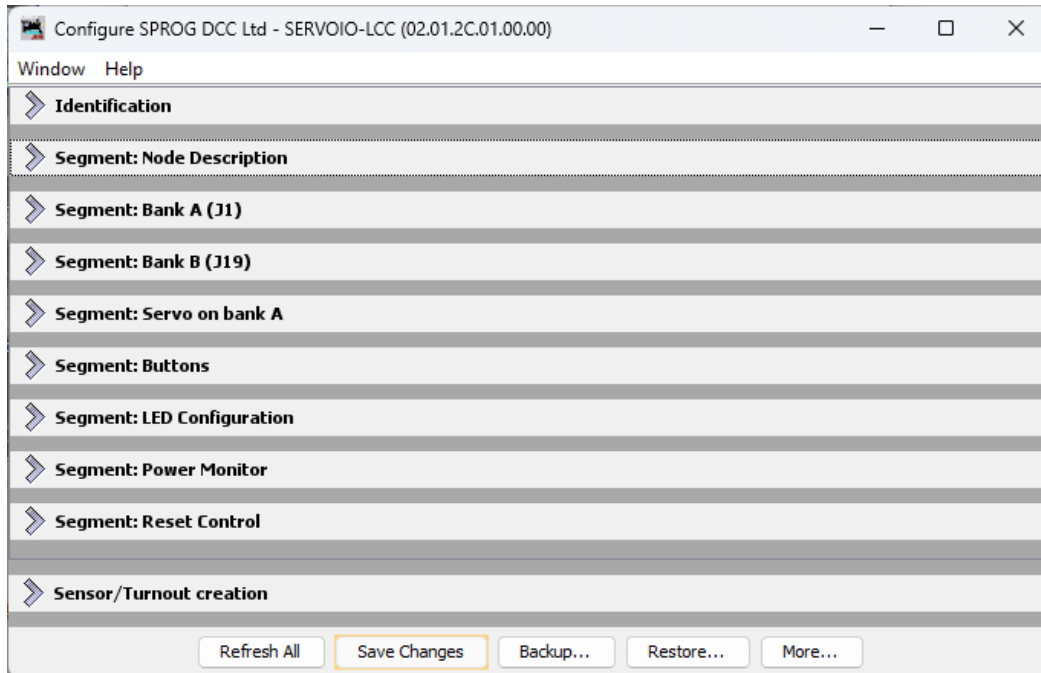
**NOTE:** The driver includes the necessary “freewheel” diode when driving inductive loads such as relays.

## 4 Configuration using the CDI (Configuration Description Information)

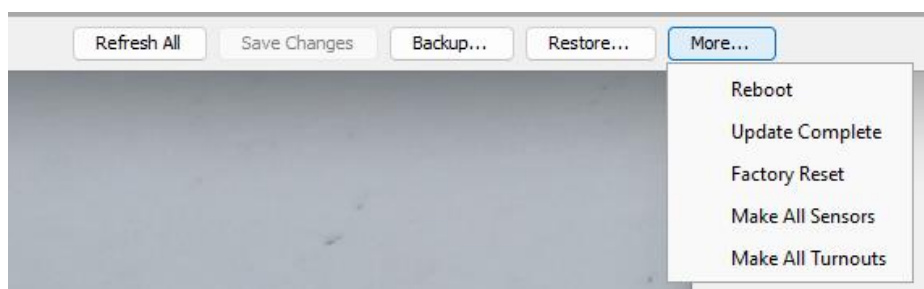
The CDI mechanism allows an LCC node to describe itself to a suitable software tool that can interact with the user in a human friendly way, e.g., through a GUI (Graphical user Interface).

The examples in this chapter show the use of Node Configuration tool in JMRI to display and edit the CDI to configure the operation of the SERVOIO-LCC.

The CDI is first shown here in its most collapsed form, followed by a description of each segment.



**NOTE:** After you have finished configuring the SERVOIO-LCC you must click Save Changes, if it is highlighted (it will not be highlighted if you written all changes individually), and then click More...Update Complete to apply your changes and reboot the SERVOIO-LCC



### 4.1 Identification

Displays node details that cannot be edited by the user.

Window Help

Identification	
Manufacturer:	SPROG DCC Ltd
Model:	SERVOIO-LCC
Hardware Version:	1.0
Software Version:	1.1

## 4.2 Segment: Node Description

In this segment you can enter a user-friendly name and description for the node.

Segment: Node Description	
User-provided description of the node	
Node Name	<input type="text"/> Refresh Write
Node Description	<input type="text"/> Refresh Write

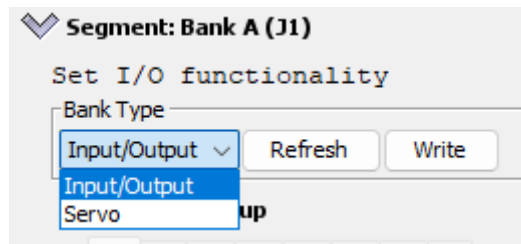
## 4.3 Segment: Bank A (J1) and Bank B (J19)

The Bank A and Bank B segments are very similar, apart from the choice of bank type.

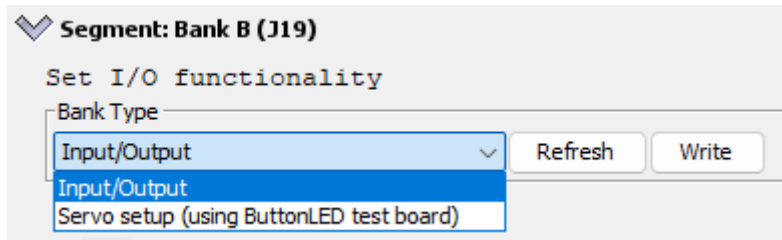
Segment: Bank A (J1)																																																																																																									
Set I/O functionality																																																																																																									
Bank Type	Input/Output <input type="button" value="Refresh"/> <input type="button" value="Write"/>																																																																																																								
<b>Channel setup</b>																																																																																																									
<table border="1"> <thead> <tr> <th>A1</th> <th>A2</th> <th>A3</th> <th>A4</th> <th>A5</th> <th>A6</th> <th>A7</th> <th>A8</th> </tr> </thead> <tbody> <tr> <td colspan="8">Description</td> </tr> <tr> <td colspan="8"><input type="text"/> Refresh Write</td> </tr> <tr> <td colspan="8">Function</td> </tr> <tr> <td colspan="8">Output <input type="button" value="Refresh"/> <input type="button" value="Write"/></td> </tr> <tr> <td colspan="8">Polarity</td> </tr> <tr> <td colspan="8">Normal <input type="button" value="Refresh"/> <input type="button" value="Write"/></td> </tr> <tr> <td colspan="8">Output On Period</td> </tr> <tr> <td colspan="8">10 <input type="button" value="Refresh"/> <input type="button" value="Write"/></td> </tr> <tr> <td colspan="8">Output Off Period</td> </tr> <tr> <td colspan="8">10 <input type="button" value="Refresh"/> <input type="button" value="Write"/></td> </tr> <tr> <td colspan="8">Units</td> </tr> <tr> <td colspan="8">Milliseconds (ms) <input type="button" value="Refresh"/> <input type="button" value="Write"/></td> </tr> </tbody> </table>		A1	A2	A3	A4	A5	A6	A7	A8	Description								<input type="text"/> Refresh Write								Function								Output <input type="button" value="Refresh"/> <input type="button" value="Write"/>								Polarity								Normal <input type="button" value="Refresh"/> <input type="button" value="Write"/>								Output On Period								10 <input type="button" value="Refresh"/> <input type="button" value="Write"/>								Output Off Period								10 <input type="button" value="Refresh"/> <input type="button" value="Write"/>								Units								Milliseconds (ms) <input type="button" value="Refresh"/> <input type="button" value="Write"/>							
A1	A2	A3	A4	A5	A6	A7	A8																																																																																																		
Description																																																																																																									
<input type="text"/> Refresh Write																																																																																																									
Function																																																																																																									
Output <input type="button" value="Refresh"/> <input type="button" value="Write"/>																																																																																																									
Polarity																																																																																																									
Normal <input type="button" value="Refresh"/> <input type="button" value="Write"/>																																																																																																									
Output On Period																																																																																																									
10 <input type="button" value="Refresh"/> <input type="button" value="Write"/>																																																																																																									
Output Off Period																																																																																																									
10 <input type="button" value="Refresh"/> <input type="button" value="Write"/>																																																																																																									
Units																																																																																																									
Milliseconds (ms) <input type="button" value="Refresh"/> <input type="button" value="Write"/>																																																																																																									

### 4.3.1.1 Bank Type

Bank A may be Servo control outputs on all channels or Individual I/O.

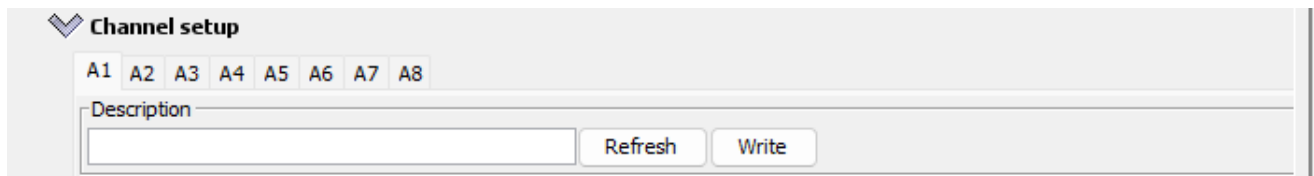


Bank B may be individual I/O or Servo setup I/O using a ButtonLed test board.

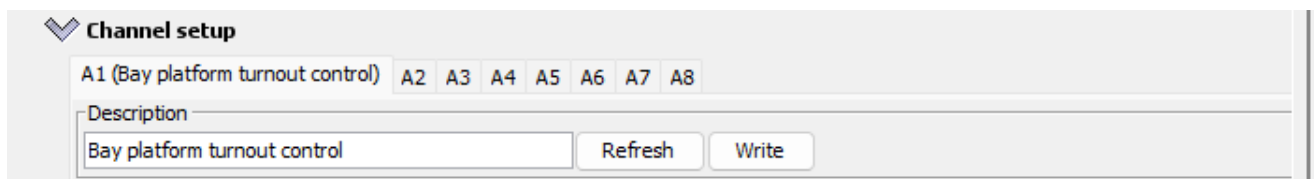


4.3.1.2 Description

The I/O channels are initially named A1 – A8 and B1 – B8. The node config tool uses a tabbed display with one tab for each of 8 channels in a bank. The Description field allows a user-friendly description to be applied to an I/O channel.

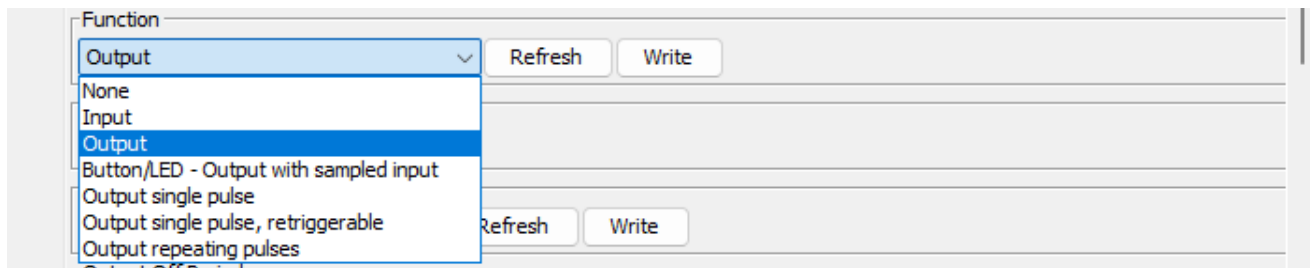


In this example A1 has been named “Bay platform turnout control”.



4.3.1.3 Function

The function selects simple input operation or a variety of output operations.

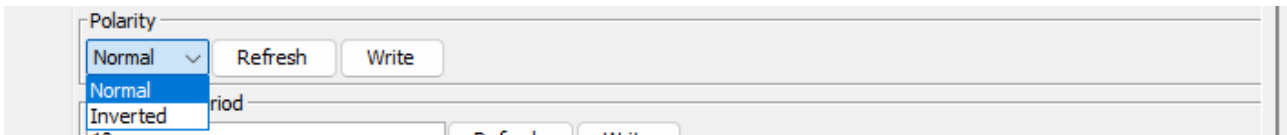


Function	Operation
None	No operation, internal state will be off
Input	Input that may produce an event
Output	Steady state output controlled by consumer events
Button/LED – Output with sampled input	Output with periodic input sampling – allows an LED and pushbutton to share a single I/O channel
Output single pulse	A consumed event can turn the channel on for a defined period. Further events during the on period will have no effect
Output single pulse, retriggerable	A consumed event can turn the channel on for a defined period. Further events during the on period will restart the timed period, extending the pulse
Output repeating pulses	A consumed on event can turn on a regular repeating on-off-on-... sequence. A different event can turn off the sequence

**NOTE:** If a bank is set to servo or servo setup the Function setting is ignored.

4.3.1.4 Polarity

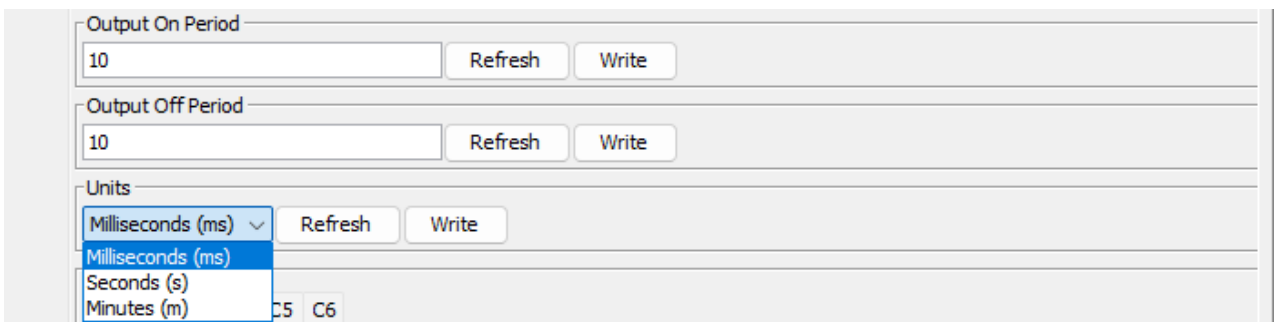
I/O channel polarity may be normal or inverted.



See 3 BASIC CONCEPTS for discussion of active-high and active-low I/O.

4.3.1.5 On/Off Periods

When an I/O channel output is set to one of the pulse modes, the pulse length is controlled by the following fields in the CDI



CDI Field	Usage
-----------	-------

Output On Period	Sets the on period for single and repeating pulses
Output Off Period	Sets the off period between repeating pulses
Units	Selects the units for on and off periods to be milliseconds, seconds or minutes

Pulse output times are accurate to approximately +/- 0.5 ms. The on and off periods default to 10 ms. It is recommended not to set periods shorter than this.

#### 4.3.2 Consumers

Each I/O channel has six consumer events available, initially name C1 – C6, presented in a tabbed display. Entering and writing an Event Name will display the name in the tab index, similar to the way channel names are displayed.

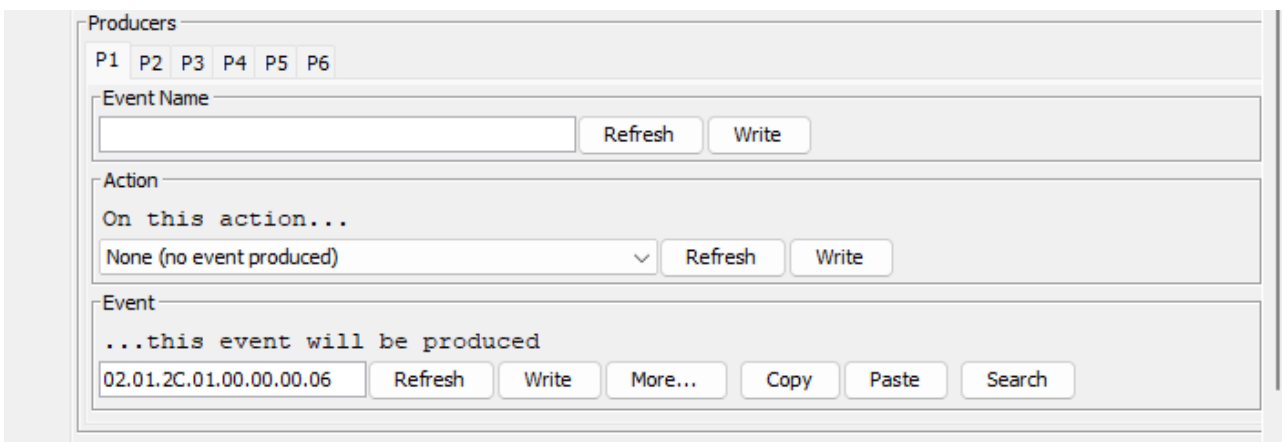
The action performed when a consumer event is consumed can be Unchanged, On, Off or Toggled.

Action	Result
Unchanged	The internal state of the I/O channel is unchanged, i.e., the event is ignored
On	The internal state of the I/O channel is set to on <ul style="list-style-type: none"> <li>• A simple output or ButtonLed output will turn on</li> <li>• A single pulse, non-retriggerable output will turn on</li> <li>• A single pulse, retriggerable output will reset and stay on for the set period</li> <li>• A repeating pulse output will start pulsing</li> </ul>

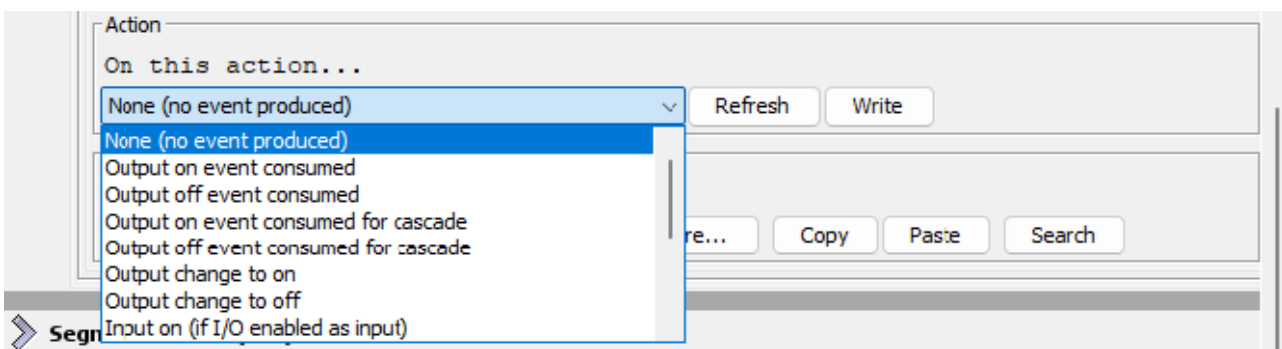
	<ul style="list-style-type: none"> <li>• A servo output will move towards the on position</li> </ul>
Off	<p>The internal state of the I/O channel is set to off.                  Outputs of all types will turn off                  A servo output will move towards the off position</p>
Toggled	<p>The internal state of the output, or pulse generation, is toggled from on to off or from off to on</p>

4.3.3 Producers

Each I/O channel has six producer events available, initially name P1 – P6, presented in a tabbed display. Entering and writing an Event Name will display the name in the tab index, similar to the way consumer event names are displayed.



An event is produced in response to the selected action. Some actions are shown in the screen shot below; the complete set are explained in the tables.



<b>Actions available on Bank A and Bank B</b>	
<b>Action</b>	<b>Event Production</b>
None (no event produced)	The event is never produced
Output on event consumed	Event produced if a consumed event turns the I/O output channel on.
Output off event consumed	Event produced if a consumed event turns the I/O output channel off.
Output toggle event consumed	Event produced if a consumed event toggles the I/O output channel.
Output on event consumed for cascade	See <a href="#">4.4.4 CASCADE EVENTS (THE YARD THROAT PROBLEM)</a>
Output off event consumed for cascade	See <a href="#">4.4.4 CASCADE EVENTS (THE YARD THROAT PROBLEM)</a>
Output toggle event consumed for cascade	See <a href="#">4.4.4 CASCADE EVENTS (THE YARD THROAT PROBLEM)</a>
Output change to on	Event produced if an output turns on (e.g., start of pulse for pulsed output). See <a href="#">4.4.3 OUTPUT EVENTS</a>
Output change to off	Event produced if an output turns off (e.g., end of pulse for pulsed output). See <a href="#">4.4.3 OUTPUT EVENTS</a>
Input on (if I/O enabled as input)	Event produced if the I/O channel turns on
Input on (if I/O enabled as input)	Event produced if the I/O channel turns off

<b>Additional Servo Actions Available on Bank A in Servo Mode Only</b>	
<b>Action</b>	<b>Event Production</b>
Servo at off end	Event produced when servo reaches off position
Servo leaves off end	Event produced when servo begins moving away from off position
Servo at on end	Event produced when servo reaches on position
Servo leaves on end	Event produced when servo begins moving away from on position
Servo at mid-point in off direction	Event produced when servo crosses mid-point whilst moving towards off end
Servo at mid-point in on direction	Event produced when servo crosses mid-point whilst moving towards on end

#### 4.4 Interaction of Events

The nomenclature use in this section is not standardised for LCC and may vary between manufacturers when describing similar features of an LCC node.

There are four cases where consumer and producer events or activity in the SERVOIO-LCC interact within the node.

##### 4.4.1 Consuming Produced Events

A basic principle of LCC is that all events appear on the network. There are no internal events within a node. This means that any event that is produced by a SERVOIO-LCC node can be specified

as a consumer event in the same node. For example, a switch input can produce an event on the network that can be consumed by the same SERVOIO-LCC to light an LED or perform some other action. The same event will also, of course, be available to all nodes on the network.

#### 4.4.2 Secondary Events

An event that is consumed by the SERVOIO-LCC may produce another event. An event that turns an output channel on might turn on a pulsed output on another channel.

#### 4.4.3 Output Events

A change in an output channel may produce an event. E.g. a channel configured to produce a repeating pulse might produce an event on every transition from off to on. It might produce a different event on every transition from on to off.

#### 4.4.4 Cascade Events (The Yard Throat Problem)

A low-level feature of the LCC protocol can cause problems when using output events to control a cascade of turnouts in a yard throat, for example.

The Output on/off event consumed for cascade action is used to mitigate this.

**TBD:** Full explanation and example to follow in a subsequent release of this document.

### 4.5 Segment: Servo on Bank A

If Bank A is configured for servo operation, the “Servo on Bank A” segment can be used to configure the operation of the servos. Otherwise, the CDI in this segment is ignored.

Eight servos in the tabbed display S1 – S8 correspond to I/O channels A1 – A8.

Servo on and off end-points correspond to I/O channel on and off states.

Starting with firmware version 1.4, output toggle events will toggle the servo position between on and off positions.

**Segment: Servo on bank A**

When Bank A is setup for servo control, these settings control the servo operation

Live Update

Off

**Servo settings**

Set Servo positions, speed, etc

S1 S2 S3 S4 S5 S6 S7 S8

Off position

0 (min, 1 ms) - 250 (max, 2 ms)

On position

0 (minimum 1 ms pulse) - 250 (maximum 2 ms pulse)

Off speed

Speed of movement towards off position, 1 (slowest) - 7 (fastest)

On speed

Speed of movement towards on position, 1 (slowest) - 7 (fastest)

Startup Action

Servo action at startup

None

Multiple Servo Control

Multiple Servos can move together or sequentially (recommended, to reduce power).

Move sequentially

Servo power-down

After moving some servos can be powered down (recommended to reduce power). Does not apply to digital servos.

Servo remain active

Servo on and off end-points are set with an integer value 0 – 250, corresponding to 1 ms and 2 ms servo control pulses, respectively.

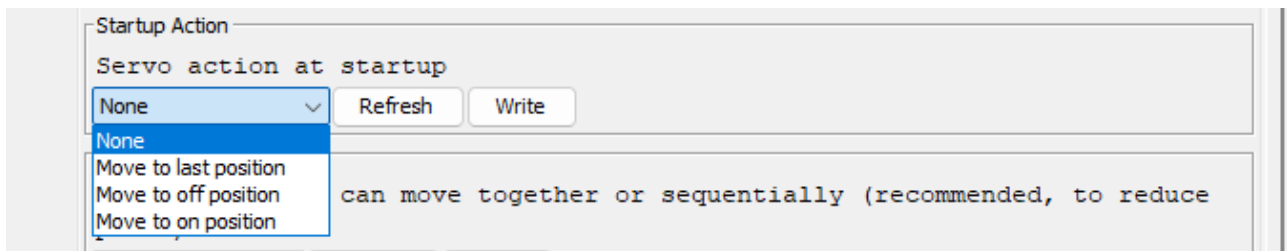
Consult your servos documentation to determine how the control pulse time corresponds to actual servo position.

There is no restriction on the relative values of the on and off positions. Swapping the values will reverse the operation of the servo.

On and off speed are set with an integer value 1 (slowest) – 7 (fastest).

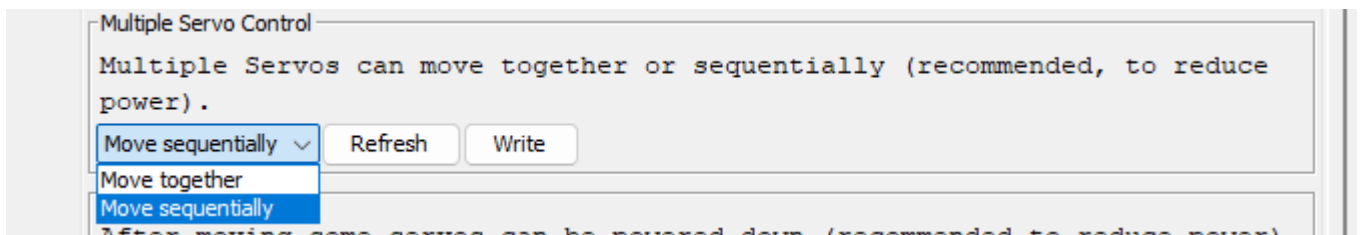
#### 4.5.1 Servo Startup action

When the SERVOIO-LCC starts up, the servos may individually do nothing, move to the last known position, move to the off position or move to the on position.



#### 4.5.2 Multiple Servo Control

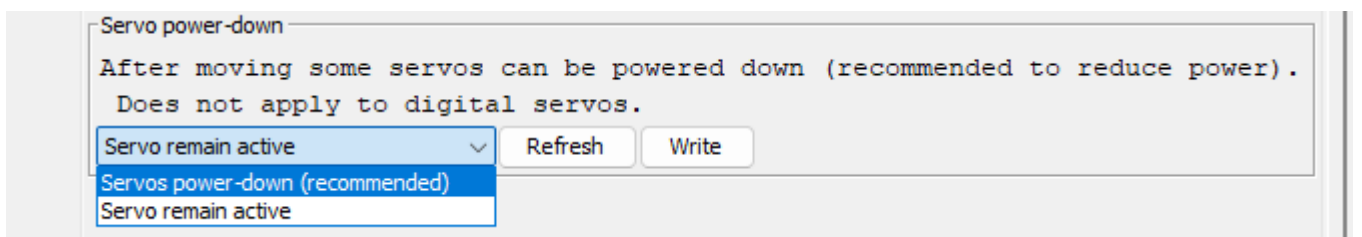
Servos can draw a surprising amount of power when moving, especially if they encounter mechanical resistance. If a number of servos are commanded to move simultaneously, the power drawn may be too great for the SERVOIO-LCC and cause erroneous operation. To mitigate this, the servos can be configured to move sequentially. i.e., a given servo will not move until lower numbered servos have finished moving.



#### 4.5.3 Servo Power Down

**NOTE:** This setting only applies to analog servos.

As a further mitigation for servo operating power, it is recommended that servos are powered off when movement is complete.



#### 4.5.4 Servo Events

Events may be produced in response to the movement of a servo. See [4.3.3 PRODUCERS](#) for the available events.

### 4.6 Segment: Buttons

The BLUE and GOLD buttons may be used to produce up to two events for button down (pressed) or button up (released) actions.

Segment: Buttons

Choose events produced by Blue/Gold buttons

B1 (Blue) B2 (Gold)

Button Name  
Blue Refresh Write

Button Events

P1 P2

Event Name  
Refresh Write

Action  
On this action...  
Button down Refresh Write

Event  
...this event will be produced  
02.01.2C.01.00.00.00.C0 Refresh Write More... Copy Paste Search

#### 4.7 Segment: LED Configuration

The BLUE and GOLD LEDs are configurable via the CDI. Each LED is controlled by up to four consumer events or internal status.

Segment: LED Configuration

Settings

Configure LEDs to respond to events or indicate network status

LED1 (Blue) LED2 (Gold)

LED Name  
Blue Refresh Write

LED Events

C1 C2 C3 C4

Event Name  
Refresh Write

Event  
Specify the event that (when consumed) will cause the LED action  
02.01.2C.01.00.00.00.C4 Refresh Write More... Copy Paste Search

Action  
Select the LED action to be performed in response to a consumed event, or the network or module status to be displayed by the LED:  
Turn LED on if event consumed Refresh Write

LED Pattern  
Select the LED blink pattern, repeating appx. every 2 s  
LED on Refresh Write

The CDI Action field controls the LED functions

Action  
Select the LED action to be performed in response to a consumed event, or the network or module status to be displayed by the LED:

Show combined LCC reception/transmission (LED events ignored) Refresh Write

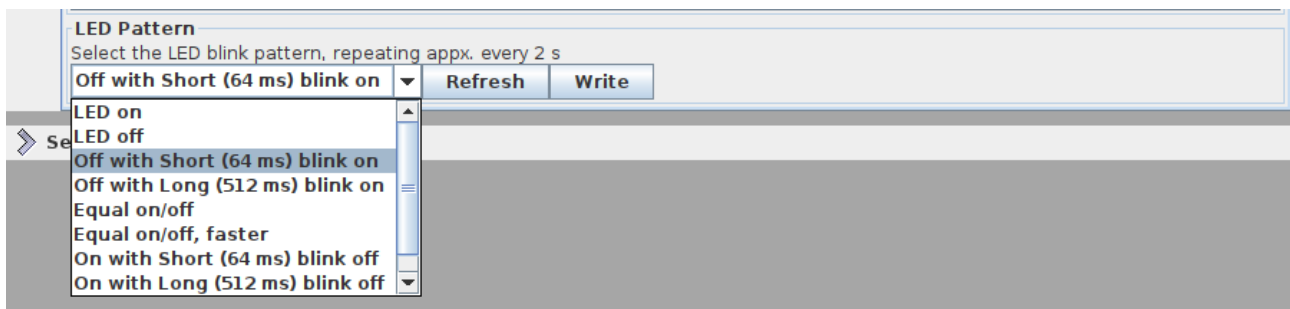
No Action

- Turn LED on if event consumed
- Turn LED off if event consumed
- Toggle LED if event consumed
- Show LCC packet reception (LED events ignored)
- Show LCC packet transmission (LED events ignored)
- Show combined LCC reception/transmission (LED events ignored)
- Indicate fault in the module (LED events ignored)

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 (LED events ignored)	The LED blinks briefly on each LCC message received
Show LCC packet transmission (LED events ignored)	The LED blinks briefly on each LCC message sent by the USB-LCC
Show combined LCC reception/transmission (LED events ignored)	The LED blinks briefly for each LCC message sent or received
Indicate fault in the module (LED events ignored)	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 pattern when turned on (or toggled from off to on) by a consumed event.



(One additional selection not shown – Use custom pattern)

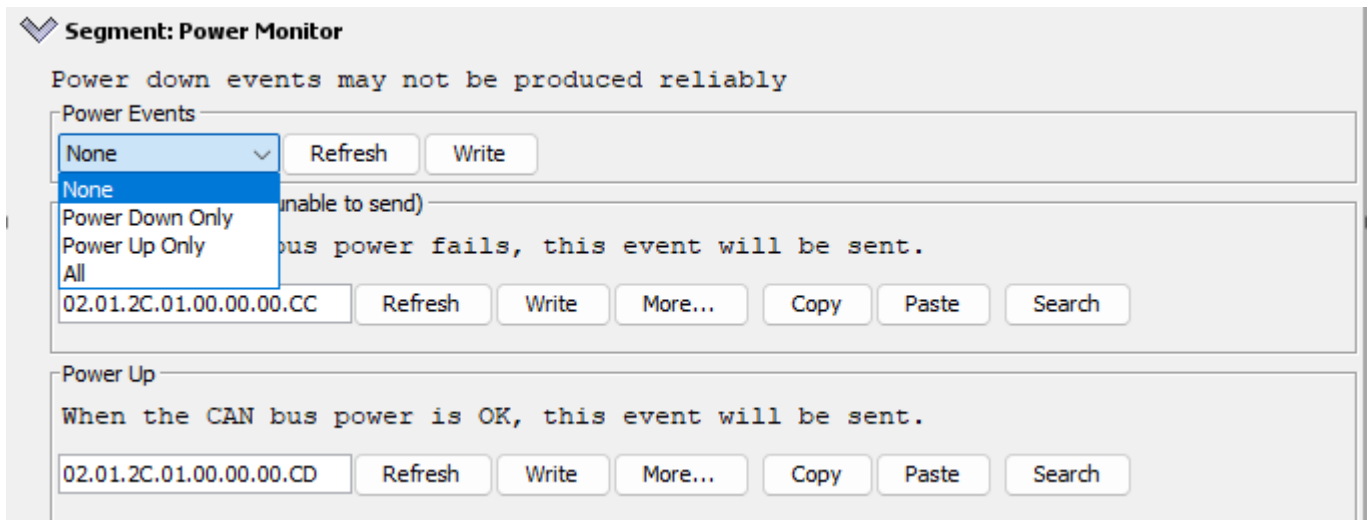
#### 4.8 Segment: Power Monitor

The SERVOIO-LCC has a power monitor that can be used to attempt to signal a problem with the LCC power supply. The power supply to the microcontroller is designed to continue to power the microcontroller and CAN interface for long enough to send an event.

In practice, it may be impossible to send any meaningful event on a power failure for a number of reasons, including

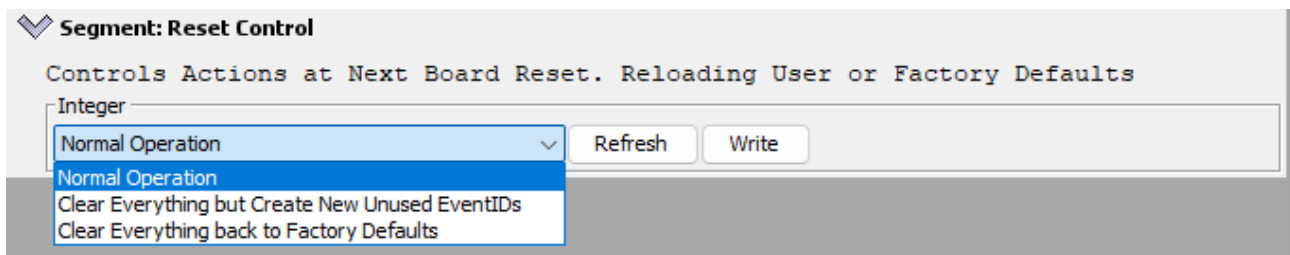
- The LCC transmit queue may be full
- The power failure may be causing error on the network
- Other nodes may be unable to process the event due to the power failure.

This feature has not yet been fully characterised.



#### 4.9 Segment: Reset Control

If you need to reset the SERVOIO-LCC back to factory settings, use the Reset Control.



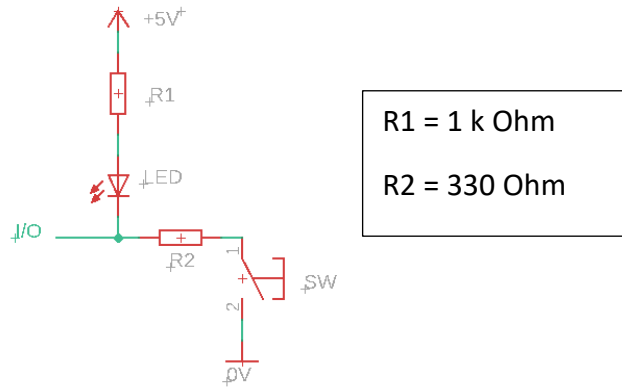
## 5 Using ButtonLed I/O

What we call ButtonLed I/O allows a single I/O channel to be used for both input and output, e.g., driving an LED and sampling a pushbutton. I/O channels may be selectively configured as ButtonLed I/O, see [4.3.1.3 FUNCTION](#)

The I/O channel is normally driven high or low to control the LED. The LED is connected between the I/O and the positive supply voltage of the I/O bank, with a suitable current limit resistor, as shown in the sample schematic, below.

The button is connected between the I/O and the zero volts or common connection of the I/O bank.

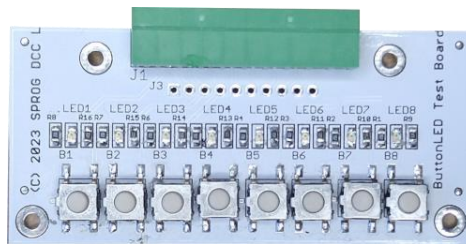
Periodically, the I/O is switched to being an input and the voltage is sampled. If the button contacts are open the LED and R1 act as a pull-up and a high voltage will be sampled. If the button contacts are closed then close to zero volts will be sampled. A further resistor, R2, is required as a current limit when the I/O is driving a high output and the button contacts are closed.



It can be seen that both the LED and button are active low, see [3.1 Active-low v. Active-high I/O](#) and the I/O polarity setting should be used to achieve the desired operation.

### 5.1 The ButtonLed Test Board

The ButtonLed test board, included in the LCC started kit or available separately, allows easy testing and configuration for ButtonLed I/O. The ButtonLed test board includes eight LEDs and eight pushbuttons and may be connected to either of the I/O banks.



## 6 Servo Setup Mode

To set the servo on and off positions, the SERVOIO-LCC can be placed into servo setup mode. This uses a ButtonLed test board, see [5.1 THE BUTTONLED TEST BOARD](#) (or the users own equivalent) connected to bank B to allow servo positions to be set and monitored.

To put the SERVOIO-LCC into servo setup mode, press and hold the BLUE button for approximately 3 seconds, until the BLUE LED starts flashing.

Whilst in servo setup mode the bank B buttons are used as shown in the table

Bank B button	Action
1	Select next servo channel
2	Select previous servo channel
3	Increase servo position (course adjustment)
4	Decrease servo position (course adjustment)
5	Increase servo position (fine adjustment)
6	Decrease servo position (fine adjustment)
7	Toggle between setting on and off positions
8	Save current servo setting

One of the bank B LEDs flashes to indicate which servo is being adjusted, e.g., bank B channel 1 flashing corresponds to the servo on Bank A channel 1 being set, etc.

A long on/short off flash indicates that the on position is being set.

A short on/long off flash indicates that the off position is being set.

**NOTE:** Save the current setting using button 8 before toggling the position or adjusting another servo.

To exit servo setup mode, press and hold the BLUE button for approximately 3 seconds, until the BLUE LED turns off. This will cause the SERVOIO-LCC to reboot and apply the new settings.

## 7 Test Mode

**NOTE:** Test mode is available from firmware version 1.3.

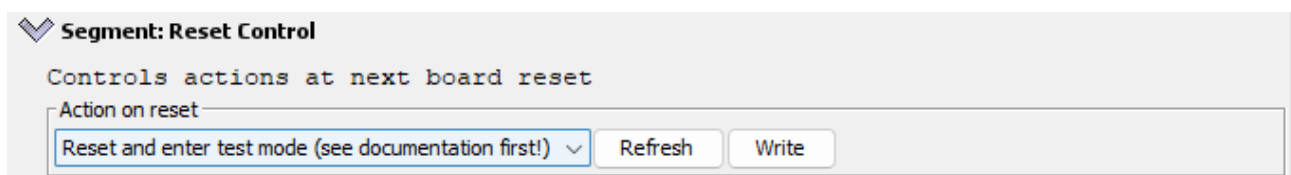
**NOTE:** Test mode exercises all output channels. You are recommended to disconnect the SERVOIO-LCC from whatever layout hardware is connected to banks A and B, unless you are sure no damage will result. In particular, test mode will not generate valid servo control pulses on Bank A which may result in unexpected servo movement.

### 7.1 Operation in Test Mode

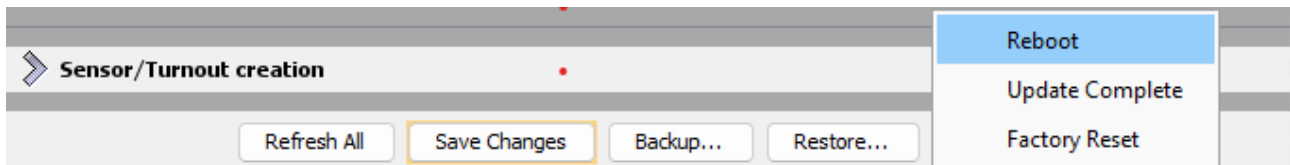
Test mode sets all I/O channels to ButtonLed mode and exercises each LED output and test each switch input. Test mode operation can be confirmed using ButtonLed test boards or user's own, similar, hardware connected to banks A and B.

### 7.2 Reboot to Test Mode

Select the "Reset and enter test mode..." action on reset in the Rese Control segment of the CDI.



Click "Write" and then click "More..." and Reboot.



The SERVOIO-LCC will reboot and enter test mode.

### 7.3 Verify LED or Channel Output Operation

Each output channel is activated for a brief period in turn, in the order: bank A1 – A8, bank B1 – B8, BLUE LED, GOLD LED after which the operation repeats. If ButtonLed test boards, or similar, are connected, individual LEDs should light up, one at a time, in sequence.

### 7.4 Verify Switch or Channel Input Operation

Pressing any switch on a ButtonLed in test mode will reset the LED pattern to start at the LED associated with that switch.

### 7.5 Exit Test Mode

To exit test mode, reboot the SERVOIO-LCC.

## 8 Firmware Updates

The SERVOIO-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.

## 9 Known Issues in This Release

### 9.1 Power Monitor

This feature has not yet been characterised.

## 10 Issues fixed in This Release

### 10.1 SNII response if buffer full uses wrong error code

Fixed.

### 10.2 CDI XML fails with strict parser

Fixed.

### 10.3 Accessing Configuration Memory

Memory space specified in the optional byte 6 of the memory read/write datagram is handled correctly.

### 10.4 Button Event Identification

Button events are identified as producer events.

### 10.5 ACDI Memory Space Layout

Format of the 251 (0xFB) and 252 (0xFC) memory spaces for ACDI are correct.

## 10.6 Memory Read Requests Flags

The Datagram Received OK message includes the reply pending flag.

## 10.7 ButtonLED I/O

I/O channels configured for Button/LED work as expected.

## 10.8 Blue Button Only Produces Off (Button Up) Events

The Blue button can produce on and off events.

## 10.9 Servo Position reporting Events

These are now reported in order.

# 11 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>

SPROG DCC Ltd Introduction to LCC/OpenLCB <https://www.sprog-dcc.co.uk/downloads/lcc/lcc-basics.pdf>

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.