

Low Cost DCC (LC-DCC) 2.2

Software Users Guide

24<sup>th</sup> April 2020

## Summary

Provided in this document is a software user's guide for the LC-DCC Windows and Android software and M3/M4 processor programming details.

Software that resides in the ARM processors that control the DCC track lines and associated Windows and Android software is purchased from eBay.

## Disclaimer

The designers accept no responsibility for any damage to any train or accessory decoder connected to this DCC system through incorrect assembly or use of the hardware design.

Please read s-9.1\_electrical\_standards\_2006.pdf NMRA standard before purchasing and using a power supply. Also note some cheap power supplies can give over voltage output.

## License / Usage Terms

All the software components are protected by license. When you buy the software from eBay, you are entitled to one license which will be provided by Email, contact [support@swws.co.uk](mailto:support@swws.co.uk) with processor identification code. Extra licenses may be purchased via PayPal, go to <http://www.swws.co.uk/lc-dcc.html> and complete the PayPal order.

## Existing users of the 1.X design

Existing users will need to obtain a license key, this will be provided free if you have purchased via my eBay account. Users creating their own software who want to upgrade will also need to contact us to obtain an "unlock code" for their M3 or M4 processor. Also, the M3/M4 pins A6/A7/C0/C1/B0/B1 functions have changed so this document should be consulted.

## LC-DCC Forum

A low cost DCC forum can be found at <http://low-cost-dcc.freeforums.net>, this was created in March 2019.

## Buying On EBay

Please note that there are a number of false adverts on EBay that cannot provide support or valid updates for this project and may not even provide anything for your money. The idea of Low Cost DCC is to make available a quality product at a price that everyone can afford. Only buy from the designer and developer of this project. To ensure you are buying from the only official EBay listing check the seller information is as displayed below (seller: johncaffyn, location: Bristol). Any other listing is false.

Low Cost DCC Controller 2amp or 10amp with CV programming and BlueTooth

★ ★ ★ ★ ★ 3 product ratings

Condition: **New**

Quantity:  20 available **86 sold** / [See Feedback](#)

**£14.99**

[Buy it now](#)

[Add to basket](#)

[Make offer](#)

[Add to watch list](#)

**100% buyer satisfaction** Click & Collect 162 watchers

Collection: Click & Collect - Select store at checkout.

Postage: **Free** Standard Delivery | [See details](#)  
Item location: Bristol, United Kingdom  
Posts to: Worldwide

Delivery: Estimated between **Fri, 12 Apr.** and **Mon, 15 Apr.**

Payments: [PayPal](#) [MasterCard](#) [VISA](#) [American Express](#) [Debit Card](#) Processed by PayPal | [See payment information](#)

Returns: No returns accepted | [See details](#)

Shop with confidence

**eBay Money Back Guarantee**  
Get the item you ordered or your money back. [Learn more](#)

**Seller information**  
**johncaffyn** (253 ★)  
100% Positive Feedback  
.....  
[Contact seller](#)  
[See other items](#)

Low Cost DCC 2.0

Configuration Track Control Check Updates  
Service Mode Time Table Exit LC-DCC

Have one to sell? [Sell it yourself](#)

## Contents

Summary .....	2
Disclaimer.....	2
License / Usage Terms .....	2
Existing users of the 1.X design.....	2
LC-DCC Forum .....	2
Buying On EBay .....	3
Introduction .....	6
M3/M4 Programming File.....	6
Programming STM32F411RE Nucleo Boards .....	7
Programming SMT32F103 Arduino Boards .....	9
Programming Verification.....	11
Windows & Linux Applications .....	13
Start Screen.....	13
Configuration Window.....	14
Serial Port Connection .....	14
WIFI Connection.....	14
Controller Settings .....	15
Current Overload Settings.....	16
Track Control Window .....	17
Popup Menu.....	18
Adding Engine Buttons.....	20
Engine Control.....	23
Function Buttons.....	23
Engine Window – Windows Only.....	24
DCC Packet Configuration Window.....	26
Service Mode Window .....	28
Time Table Window .....	29
LC-DCC Controller Communications Window .....	31
Android Bluetooth Application .....	32
Installing Android Application.....	32
Start-up Screen .....	33
Service Mode Screen.....	34
DCC Control Screen.....	35
Multi Train DCC Control Screen .....	36
Configuration Screen .....	37
Analogue Configuration Screen .....	38

Accessory Button Configuration .....	39
Engine & Accessory Address .....	40
Accessory Control Screen.....	41
License Management Screen .....	42
Android Voice Application .....	43
Start-up Screen .....	44
Configuration Options Screen.....	45
Update Custom Commands Screen .....	46
Google Voice Input Screen.....	47
Manage Licenses Screen .....	48
Version Change History.....	49

## Introduction

This book describes installation and use of the various software items that make up the LC-DCC system.

This includes programming the M3/M4 micro controllers, installing and using the Android application and installing and using Windows and Linux applications.

All software components can be downloaded free of charge from [www.swws.co.uk/dcc\\_files.html](http://www.swws.co.uk/dcc_files.html). The software requires a license to use which is purchased on EBay.

## M3/M4 Programming File

The M3 and M4 binary programming files are contained in .zip files that can be downloaded from [www.swws.co.uk/lc-dcc-files.html](http://www.swws.co.uk/lc-dcc-files.html). Within the .zip file there are four programming files, these files are for different current configurations. The INA219 can be configured to measure 3.2 amps maximum (default) or 6.4 amps, 9.6 amps or 12.8 amps by adding extra shunt resistors. The programming files are setup for each of these current limits and the correct one should be chosen for your configuration. If no INA219 is connected then you can program any binary file into the M3 or M4 processor.

## Programming STM32F411RE Nucleo Boards

To program STM32F411RE boards you must download from the STM website the **STM32 ST-Link Utility** Windows application and associated USB drivers.

The board is then programmed using the STM32 ST-LINK Utility as follows:

Install the STM V2 Link software, this can be downloaded for free from: <http://www.st.com/en/embedded-software/stsw-link004.html>.

Connect the F411RE board to a PC using a USB cable.

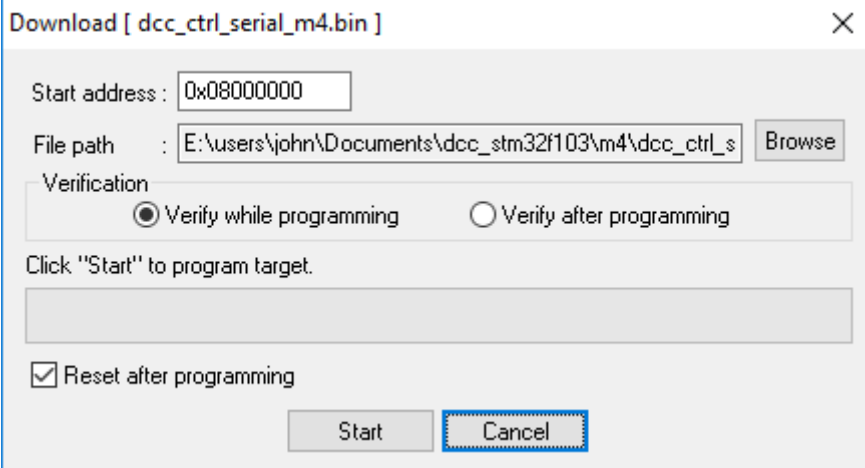
Use the file menu and open the binary file you require to load. This may be **bt\_terminal\_m4.bin** if you need to configure a Bluetooth module or any LC-DCC controller file such as **lc\_dcc\_serial\_m4\_3p2.bin**, **lc\_dcc\_serial\_m4\_6p4\_wadc.bin** etc.

The screenshot shows the STM32 ST-LINK Utility window. The 'Memory display' section is active, showing a hex dump of the file 'dcc\_ctrl\_serial\_m4.bin'. The address range is from 0x00000000 to 0x00000080. The data width is set to 32 bits. The hex dump shows the following data:

Address	0	4	8	C	ASCII
0x00000000	2000B360	08004E69	08004EC8	08004EC8	^3. i N.. È N.. È N..
0x00000010	08004EC8	4FF0E92D	4338F8DC	E891B083	È N.. - é ð O Ü ø 8 C f " ' è
0x00000020	93014100	F04F4422	F0040901	2C000307	. A . " " D O ð ... ð ... ,
0x00000030	F103FA09	BF884623	EB0C1DE3	1C6703E3	. ú . ñ # F , ð ä ... è ä . g .
0x00000040	6BE4F893	EA414297	463D0106	F883463C	" ø ä k — B A é . . = F < F f ø
0x00000050	D1EA1BE4	F8CC45F0	F3007338	F10E8099	ä . é Ñ ð È ð 8 s . ó " € . ñ
0x00000060	EBC80E01	EB00080E	93000308	F04F46BE	.. È ë ... ë ... " ¼ F O ð
0x00000070	F04F0B00	F1BE0901	46720F00	F10EBFB8	.. O ð ... ¾ ñ ... r F , ð . ñ
0x00000080	EB0C0207	F00E02E2	F8920407	F8901BE4	... è ä ... ð ... ' ø ä . ø

The status bar at the bottom shows 'Disconnected', 'Device ID : -----', and 'Core State : No Memory Grid Selected'.

Use the target menu then program and verify (CTRL+P) to program to F411RE flash memory:



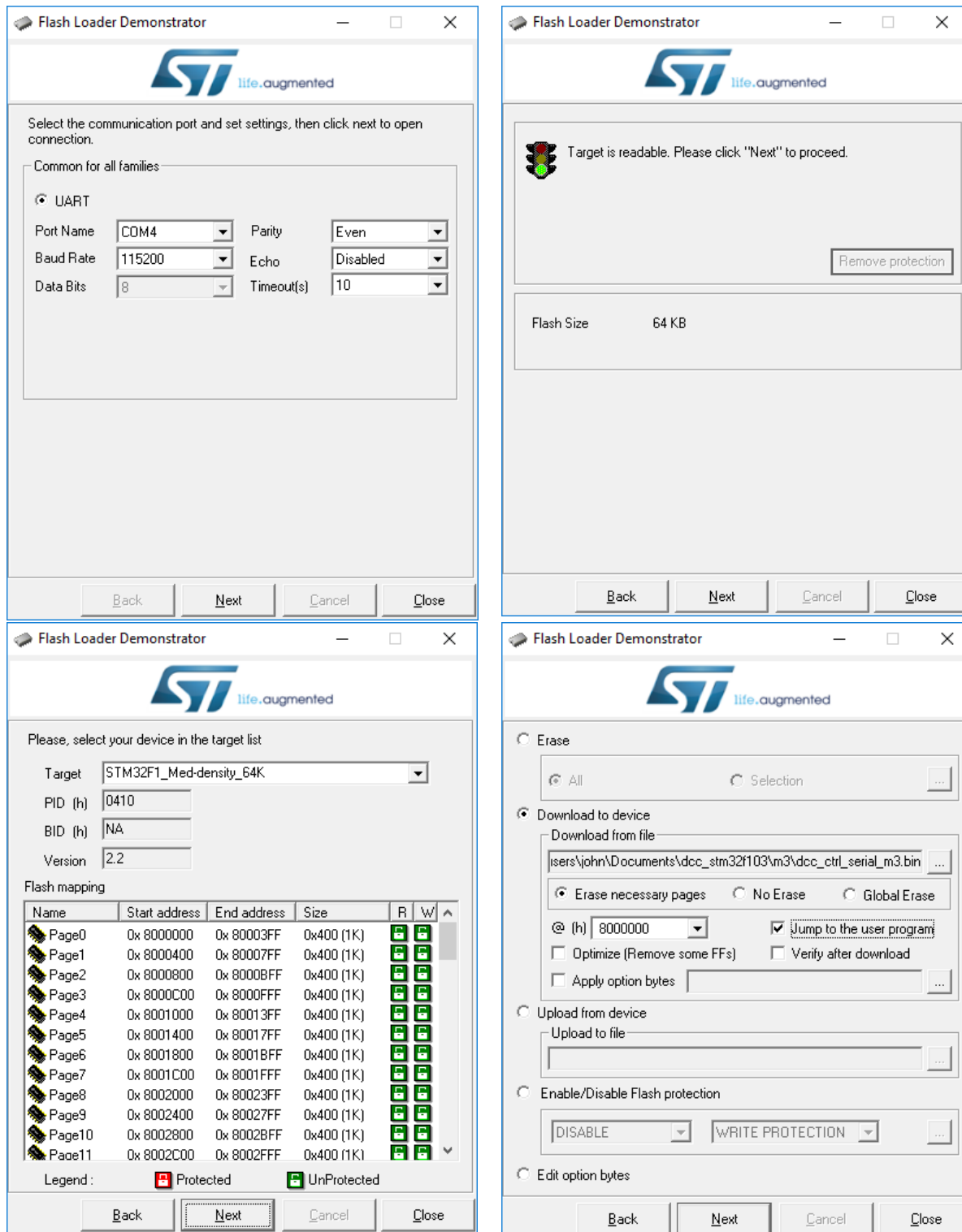
A screenshot of a 'Download' dialog box titled 'Download [ dcc\_ctrl\_serial\_m4.bin ]'. The dialog has a close button (X) in the top right corner. It contains the following fields and controls:

- 'Start address' field with the value '0x08000000'.
- 'File path' field with the value 'E:\users\john\Documents\dcc\_stm32f103\m4\dcc\_ctrl\_s' and a 'Browse' button to its right.
- 'Verification' section with two radio buttons: 'Verify while programming' (selected) and 'Verify after programming'.
- Text instruction: 'Click "Start" to program target.' followed by a large empty rectangular box.
- A checked checkbox labeled 'Reset after programming'.
- 'Start' and 'Cancel' buttons at the bottom right.



## Programming SMT32F103 Arduino Boards

To program STM32F103 boards there are two options, either via the serial interface of USART1 or by using the debug connector and an STM ST-LINK V2 USB adapter. If using the STM ST-LINK V2 USB adapter follow the instructions in the previous section for the STM32F411RE processor. For the serial USART1 option you must download from the STM website the **Flash Loader Demonstrator** Windows application. The programming screens appear as follows:



To program the STM32F103 make sure the two boot jumpers near the USB connector are set as shown below:



Connect a FTDI 232 USB serial adapter as shown above on the right to A9 and A10 pins on the STM32F103. Power the STM32F103 from either the FTDI module or from a USB cable connected between a PC and the STM32F103 USB connector. **Note if using the FTDI to power the STM32F103 ensure the correct voltage (jumper on FTDI) is selected and connected to the correct pin on the STM32F103.**

Run the flash loader software on the PC, select the correct COM port for the FTDI USB module. Select the next button three times assuming all is ok (as per the previous page screens).

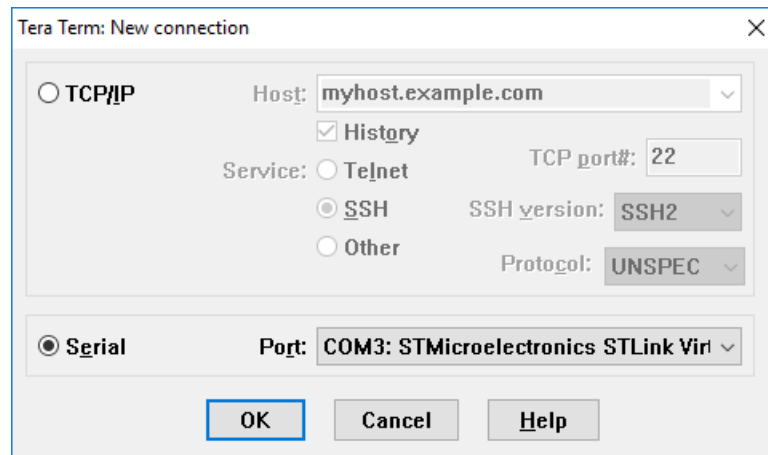
Select **Download to device** and choose which binary file to program. The binary files are different for the blue and black pill and have blue or black in their names: **lc\_dcc\_serial\_m3\_black\_3p2.bin**, **lc\_dcc\_serial\_m3\_blue\_3p2\_wadc.bin** for example. Once the programming is complete change the boot jumpers so both jumpers are near the USB connector.

Cycle power on the STM32F103 board and then follow the programming verification section.

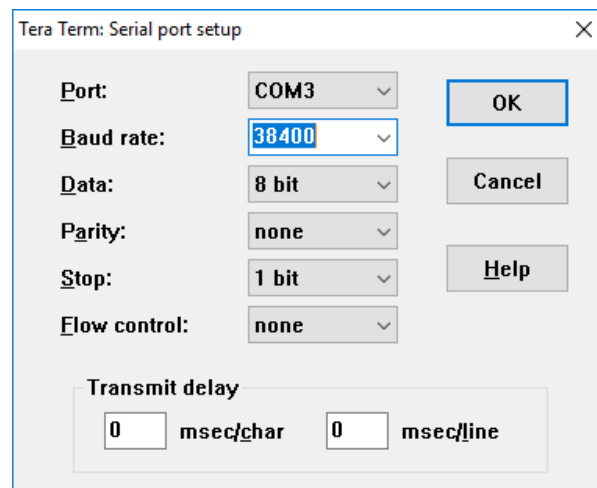
If you want to configure a Bluetooth module then program the **bt\_terminal\_m3.bin** file which works for both blue and black pill boards.

## Programming Verification

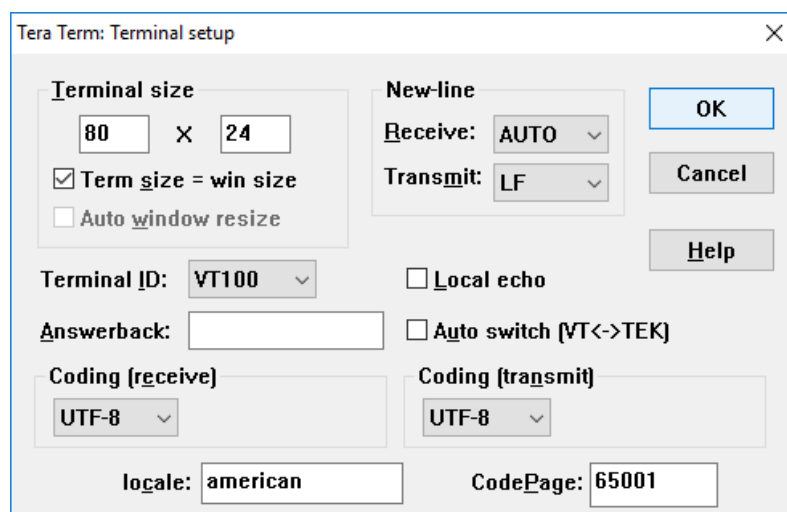
Once the board has been programmed and if required the boot jumpers changed (STM32F103) the board LED should flash once a second. To further verify the board has programmed use Tera Term or any other terminal emulator program and connect to the board USB connector using a USB cable. Set Tera Term up as follows:



The 'Tera Term: New connection' dialog box is shown. The 'Serial' radio button is selected. The 'Host' field is set to 'myhost.example.com'. The 'Service' is set to 'SSH'. The 'SSH version' is set to 'SSH2'. The 'Protocol' is set to 'UNSPEC'. The 'Port' is set to 'COM3: STMicroelectronics STLink Vir'. The 'OK' button is highlighted.



The 'Tera Term: Serial port setup' dialog box is shown. The 'Port' is set to 'COM3'. The 'Baud rate' is set to '38400'. The 'Data' is set to '8 bit'. The 'Parity' is set to 'none'. The 'Stop' is set to '1 bit'. The 'Flow control' is set to 'none'. The 'Transmit delay' is set to '0 msec/char' and '0 msec/line'. The 'OK' button is highlighted.



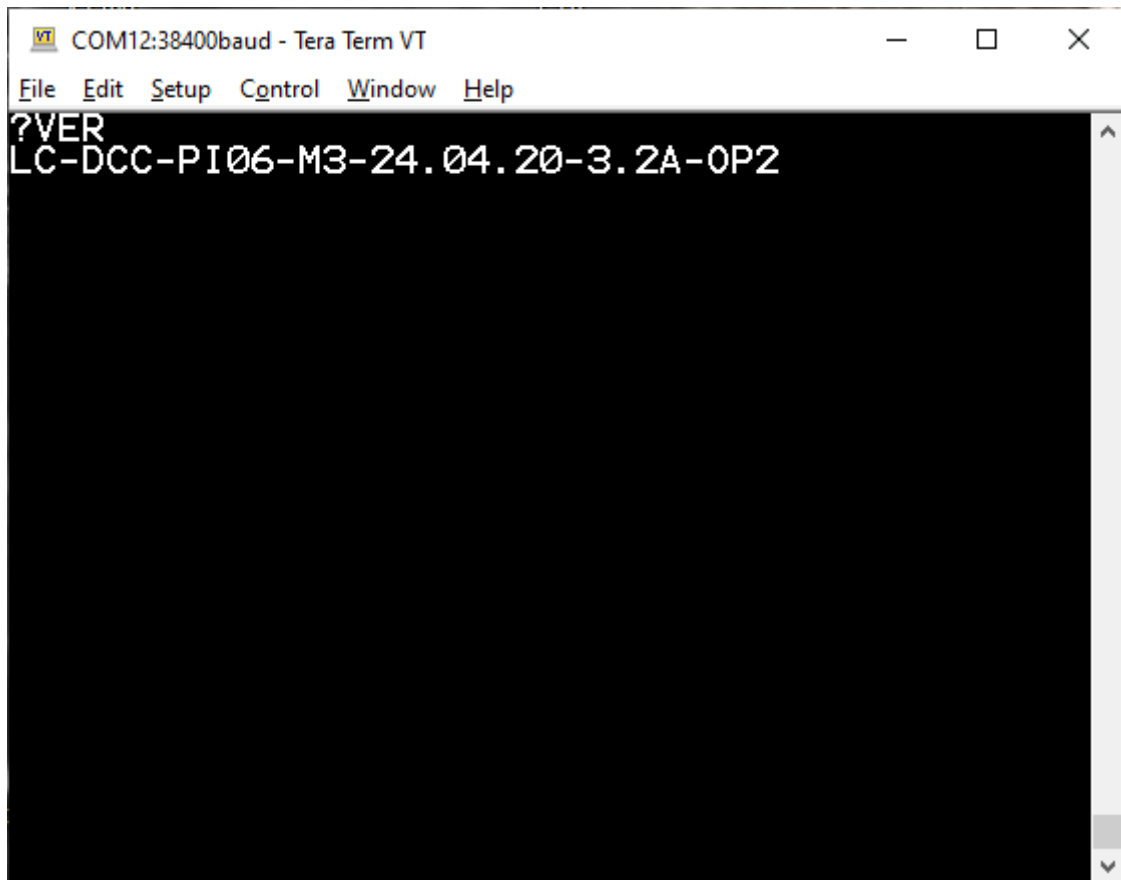
The 'Tera Term: Terminal setup' dialog box is shown. The 'Terminal size' is set to '80 x 24'. The 'Term size = win size' checkbox is checked. The 'Auto window resize' checkbox is unchecked. The 'Terminal ID' is set to 'VT100'. The 'Answerback' field is empty. The 'Coding (receive)' is set to 'UTF-8'. The 'Coding (transmit)' is set to 'UTF-8'. The 'locale' is set to 'american'. The 'CodePage' is set to '65001'. The 'New-line' settings are 'Receive: AUTO' and 'Transmit: LF'. The 'Local echo' and 'Auto switch [VT<->TEK]' checkboxes are unchecked. The 'OK' button is highlighted.

To verify correct operation of the software, use the following instructions on the terminal:

Press a lower case **z** key; this enables echoing of characters by the LC-DCC controller.

Next enter **?VER** to display the software version number

The output should appear as follows on the Tera Term display for an STM32F103 board with potentiometer control attached:



The screenshot shows a Tera Term window titled "COM12:38400baud - Tera Term VT". The menu bar includes "File", "Edit", "Setup", "Control", "Window", and "Help". The terminal display shows the command "?VER" entered, followed by the output "LC-DCC-PI06-M3-24.04.20-3.2A-0P2". The terminal has a black background with white text. A vertical scrollbar is visible on the right side of the terminal window.

Similar outputs will appear for the SM32F411RE boards. The software build date is indicated by 24.04.20 in the above screen shot, the protocol version by PI06, the INA219 current limit by 3.2 (amps) and OP2 indicates two track outputs.

## Windows & Linux Applications

The Windows/Linux application can be downloaded from the website specified at the start of this document. This application does not need to be installed it can be copied and placed on any of the PC drives. The application allows for CV programming (service mode), train/points control and a time table driven mode for trains and points.

The following sections explain the different screens, screen shots are for Windows and may differ slightly on Linux.

### Start Screen

When the application is executed the screen appears as follows:



This screen provides the user with buttons to configure the DCC control system, check for software updates on the internet, exit the application and run the different operating modes of the DCC control software.

The check updates button will interrogate the [www.swws.co.uk](http://www.swws.co.uk) website to establish if there is a newer version of the software available. If there is a newer version the user will be informed of the download URL link and the new version number and release date.

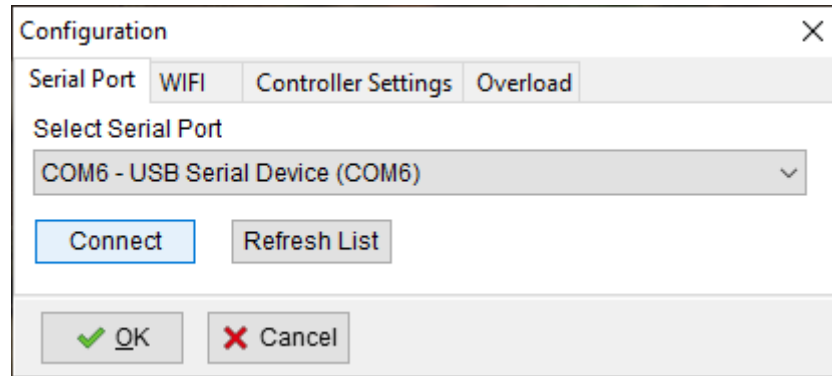
The exit button closes the Windows application.

The configuration button displays a configuration window which is explained in the next section.

## Configuration Window

### Serial Port Connection

The configuration screen has four pages. The first page allows the user to choose a serial port that is used to communicate commands to the LD-DCC controller:

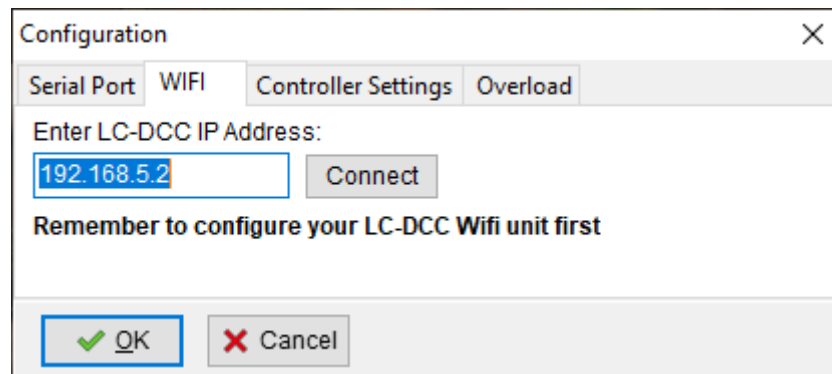


A serial port must be selected then the connect button must be clicked to connect to the controller. Note that when using Linux it may be necessary to change the access permissions for the serial device, for example:

```
sudo chmod ugo+rw /dev/ttyUSB0
```

### WIFI Connection

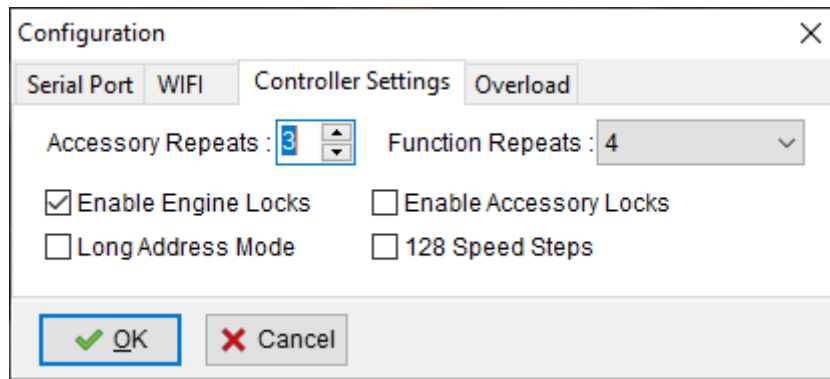
The second page allows the user to connect to an LC-DCC system that supports WIFI connection.



An IP address must be selected for the controller, then the connect button must be clicked to connect to the controller. The LC-DCC controller must be configured first.

## Controller Settings

The third page allows for a number of LC-DCC controller configuration options:



The screenshot shows a 'Configuration' dialog box with four tabs: 'Serial Port', 'WIFI', 'Controller Settings' (which is selected), and 'Overload'. In the 'Controller Settings' tab, there are two numeric input fields: 'Accessory Repeats' set to 3 and 'Function Repeats' set to 4. Below these are four checkboxes: 'Enable Engine Locks' (checked), 'Enable Accessory Locks' (unchecked), 'Long Address Mode' (unchecked), and '128 Speed Steps' (unchecked). At the bottom of the dialog are 'OK' and 'Cancel' buttons.

The controller settings page has the following configuration options:

### Accessory Repeats

As DCC is unreliable due mainly to the mechanical pickup on the track and dirt on the track the user can configure the number of times an accessory packet is sent to the decoder. The maximum number of accessory packets that can be sent is 8. The default value is shown above.

### Function Repeats

This controls the number of times a function packet is sent to a train decoder. A function repeat value can be 1 to 9 or continuous. The default value is shown above.

### Enable Engine Locks

This option enables engine locking. This means no two serial interface controls can manage the same engine. If there are multiple users controlling trains then this should be enabled. If you are using a second serial interface for Bluetooth walk about control then you probably want this not enabled.

### Enable Accessory Locks

This option enables accessory locking. This means no two serial interface controls can manage the same accessory address. If there are multiple users controlling points or accessories then this should be enabled. If you are using a second serial interface for Bluetooth walk about control then you probably want this not enabled.

### Long Address Mode

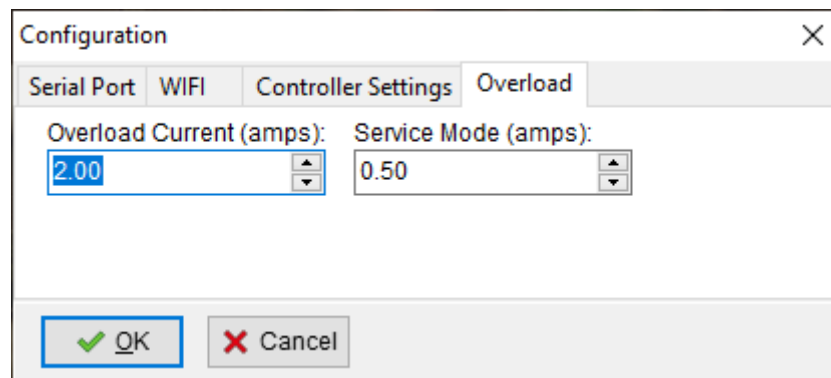
Long address mode enables 14-bit engine addresses in the LC-DCC controller and all addresses generated by the controller will be 14-bit addresses.

### 128 Speed Steps

This option sets the LC-DCC controller default speed steps to 128. The speed steps for any engine can be changed from the track control screen to either 28 or 128.

## Current Overload Settings

The fourth page allows the current overload detection configuration:



The image shows a 'Configuration' dialog box with a close button (X) in the top right corner. It has four tabs: 'Serial Port', 'WIFI', 'Controller Settings', and 'Overload'. The 'Overload' tab is selected. Inside the dialog, there are two settings: 'Overload Current (amps):' with a value of '2.00' and 'Service Mode (amps):' with a value of '0.50'. Both values are in input fields with up and down arrow buttons. At the bottom, there are two buttons: 'OK' (with a green checkmark icon) and 'Cancel' (with a red X icon).

The overload detection page settings are as follows:

### Overload Current

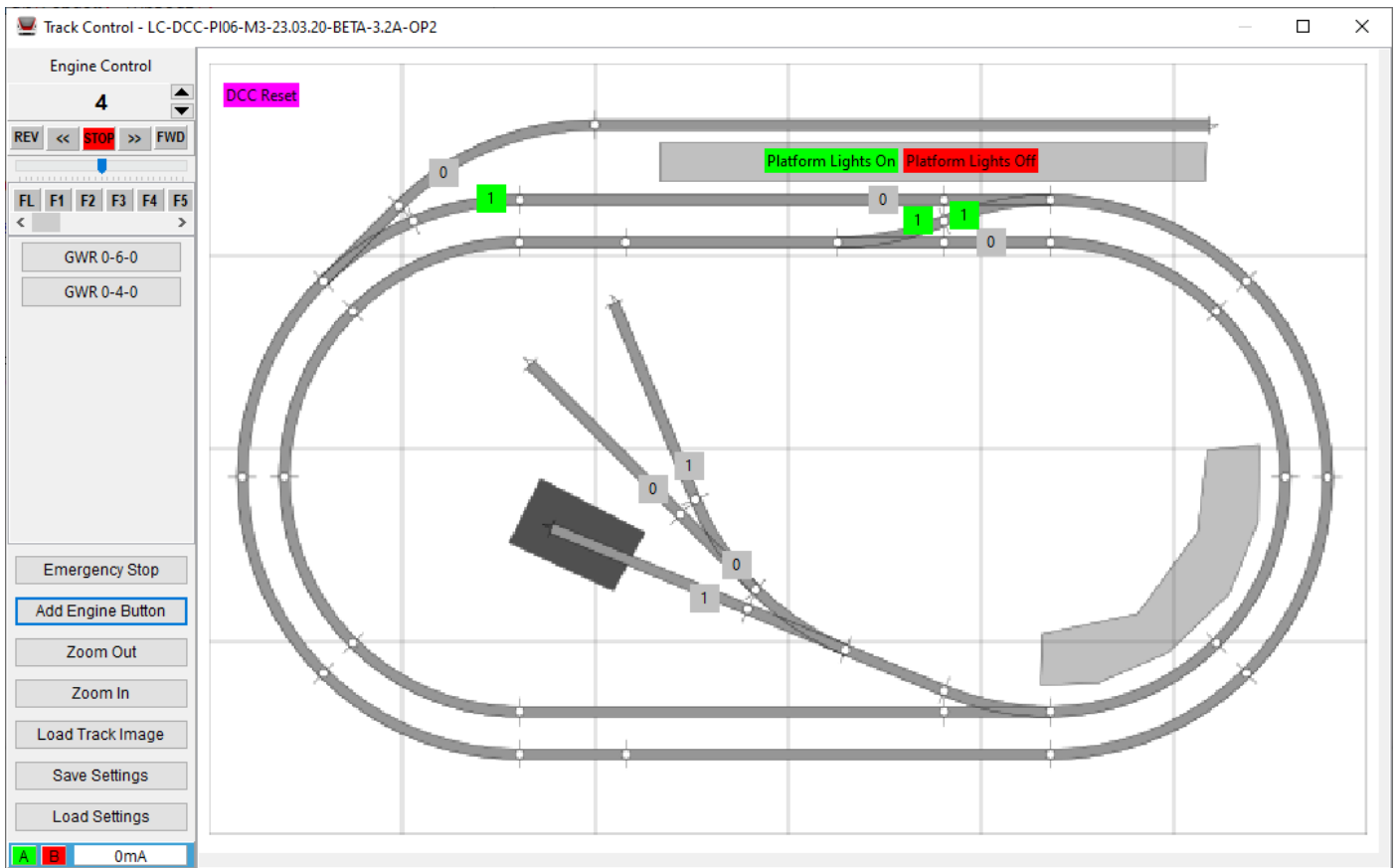
This is the overload detection maximum current allowed during normal train operation. If a current larger than this is detected the LC-DCC controller will remove power from both tracks A and B.

### Service Mode Current

This is the overload detection maximum current allowed during service mode operation. If a current larger than this is detected the LC-DCC controller will remove power from both tracks A and B.



## Track Control Window



The track control screen allows control of both trains and point/accessory decoders/dcc packets. Each point can have two buttons allocated to it to control point direction. DCC packet control can be added via buttons and the DCC packet configuration window. Engine buttons can be added to simplify engine control. Decoder function buttons can be labelled by the user for each engine address and the button on colour for each button can be defined. All configuration values can be saved and reloaded using the save/load settings buttons.

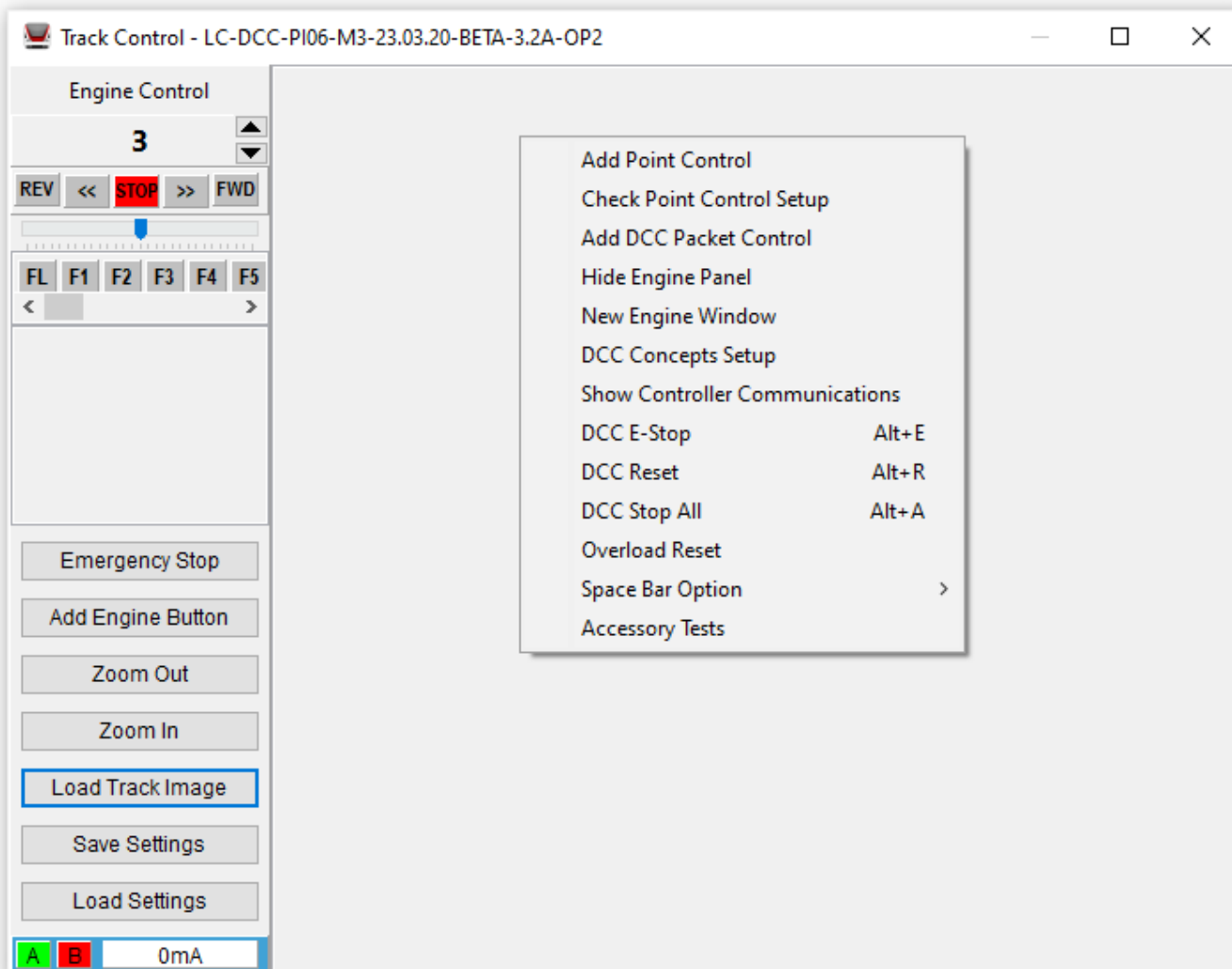
The status of track A and B power is monitored continuously along with the current being used by the layout.

To use this screen an image of the track layout must be loaded using the **Load Track Image** button. This can be in any of the common image formats like .png, .jpg etc.

In the example screen above can be seen the new DCC packet control buttons for: *depot*, *siding*, *all stop* and *platform lights*. These buttons allow user selected DCC packets to be sent to the controller and finally the track DCC signals.

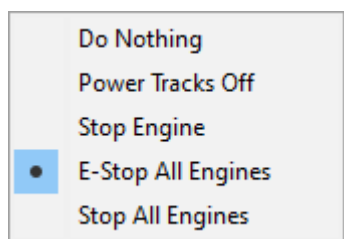
## Popup Menu

The popup menu on the layout display allows addition of point control, packet control, engine windows, overload reset, space bar configuration and other miscellaneous features such as: hide the left-hand engine panel to maximize the track display, check their point configuration and setup DCC Concepts learning accessories.



## Space Bar Option

The space bar menu option allows the following action to be performed when the space bar is pressed:



E-Stop All Engines generates a DCC emergency stop all whereas a Stop All Engines generates a broadcast stop DCC packet.

## Overload Reset

If the LC-DCC controller has detected an overload and disabled power to the tracks the overload can be reset by using this menu option. The cause of the overload must first be resolved.

### *Adding Engine Windows*

Engine windows can be added using popup menus from either the main track display or via engine buttons from the left-hand panel. See the [Engine Window](#) section for more information.

### *DCC Concepts Setup*

This menu option allows configuration of accessory decoders that have a program/learn switch. It allows an accessory address to be sent to the track via the LC-DCC controller so the accessory can learn the programming address. This menu option is only enabled before any train or accessory commands have been sent to the LC-DCC controller.

### *Adding Packet Control Buttons*

Use the Add DCC Packet Control menu option to add a button that controls the sending of DCC packets. See the [DCC Packet Configuration Window](#) for more information.

### *Adding Point Direction Buttons*

To add point direction buttons, use the right mouse button on the track image and then the popup menu **Add Point Control**, enter a label for the point and an address:

The image shows two sequential dialog boxes. The first, titled 'Point Label', has a text input field containing 'PT #13' and buttons for 'OK' and 'Cancel'. The second, titled 'Address', has a text input field containing '17' and buttons for 'OK' and 'Cancel'.

The point control setup can be checked by right clicking on the track image and selecting the **Check Point Control Setup** menu option. This will check addresses are not duplicated and direction values (0 or 1) are not duplicated. Any problems are highlighted red and yellow.

Once a point button has been added it can be managed by using the popup menu for the point direction button. This is accessed using a right mouse click on the point direction button.

The popup menu has the following options:

#### **Change Value (Direction)**

This option allows the accessory value sent to the DCC unit to be changed, valid values are 0 to 7 inclusive. This value along with the address is used to switch a DCC point decoder when the button is clicked.

#### **Change Address**

This allows the DCC address used for the point direction button to be changed. When the point direction button is clicked this address along with the direction value 0 to 7 will be sent to the DCC control unit to control the point decoder.

## Move

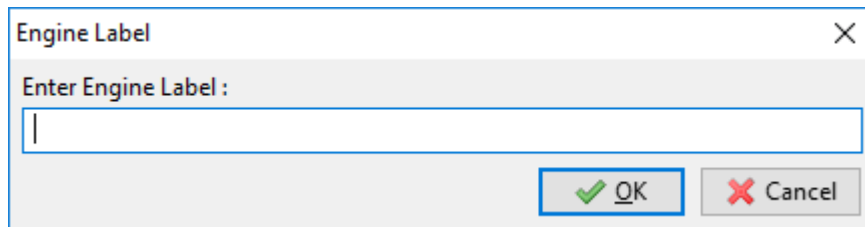
This option will move the point direction button with the mouse until the user clicks on the button or track image with the left mouse button.

## Delete

This option allows the point direction button to be deleted.

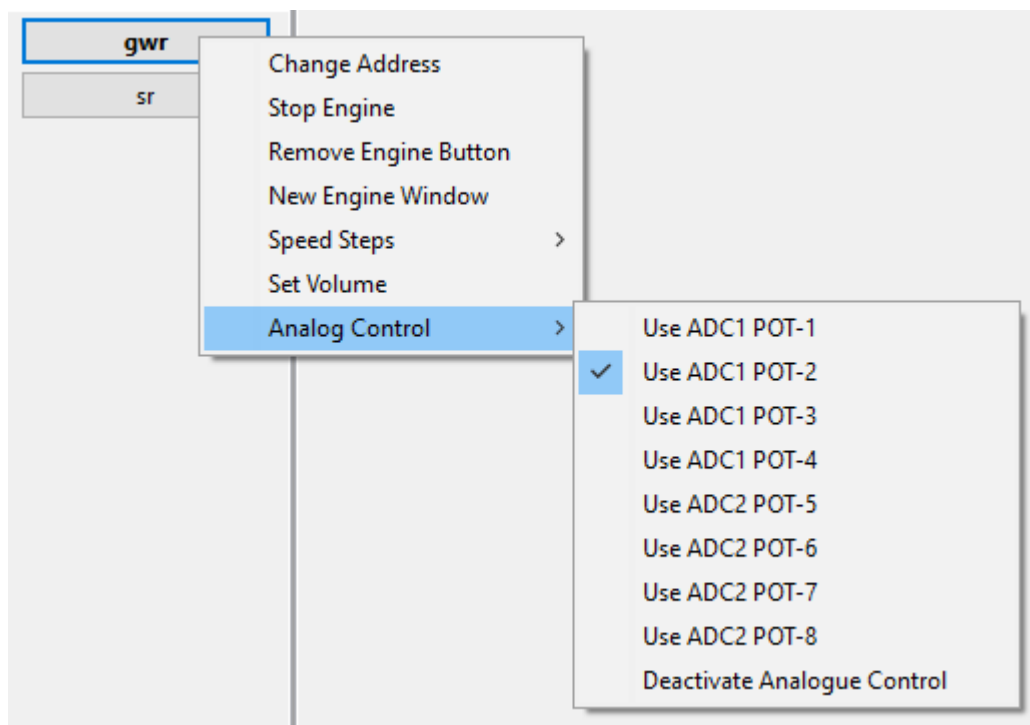
## Adding Engine Buttons

Engine buttons can be added using the **Add Engine** button. The button must have a label:



A dialog box titled "Engine Label" with a close button (X) in the top right corner. It contains a text input field with the placeholder text "Enter Engine Label :". Below the input field are two buttons: "OK" with a green checkmark icon and "Cancel" with a red X icon.

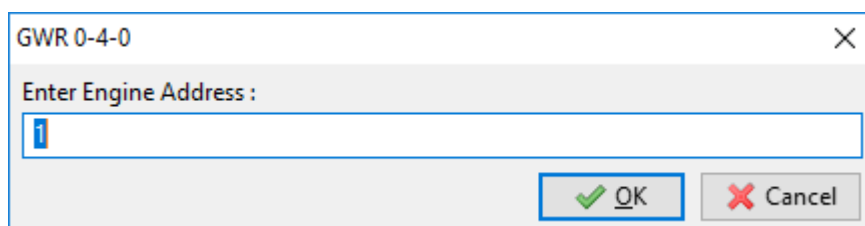
Each engine button has a popup menu accessed by using the right mouse click on the engine button:



The popup menu has the following options:

### Change Address

The engine address can be changed using the following dialogue window:



A dialog box titled "GWR 0-4-0" with a close button (X) in the top right corner. It contains a text input field with the placeholder text "Enter Engine Address :". Below the input field are two buttons: "OK" with a green checkmark icon and "Cancel" with a red X icon.

### Stop Engine

This option will send a DCC stop command for the engine address.

### **Remove Engine Button**

The engine button can be deleted with this option.

### **New Engine Window - Windows Only**

Creates an engine window with the same address as the engine button.

**Speed Steps**

Allows the speed steps to be changed between 28 and 128 steps for the engine decoder.

**Set Volume**

Allows the engine sound volume to be changed assuming the engine decoder has sound.

**Analog Control**

Allows configuration of the analogue potentiometer options assuming the LC-DCC controller has ADC potentiometers build into it.

## Engine Control

Apart from the engine buttons there is an engine address field that can be used to select engine address.

Below the engine address field are buttons to control engine speed + or -, engine direction >> or << for forward or reverse and an engine stop button.

Below these buttons is a scroll bar that can also be used to set engine speed.

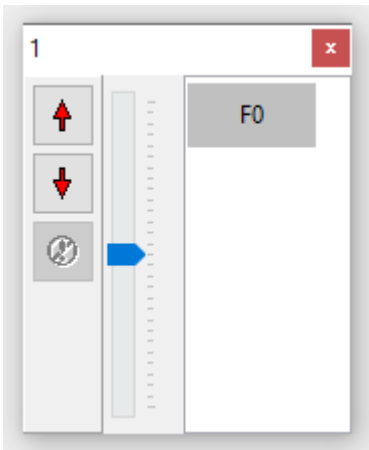
Below the speed scroll bar are buttons that allow engine functions to be turned on or off. The buttons currently support the NMRA DCC function group 1 functions F1..F4 and FL (light).

## Function Buttons

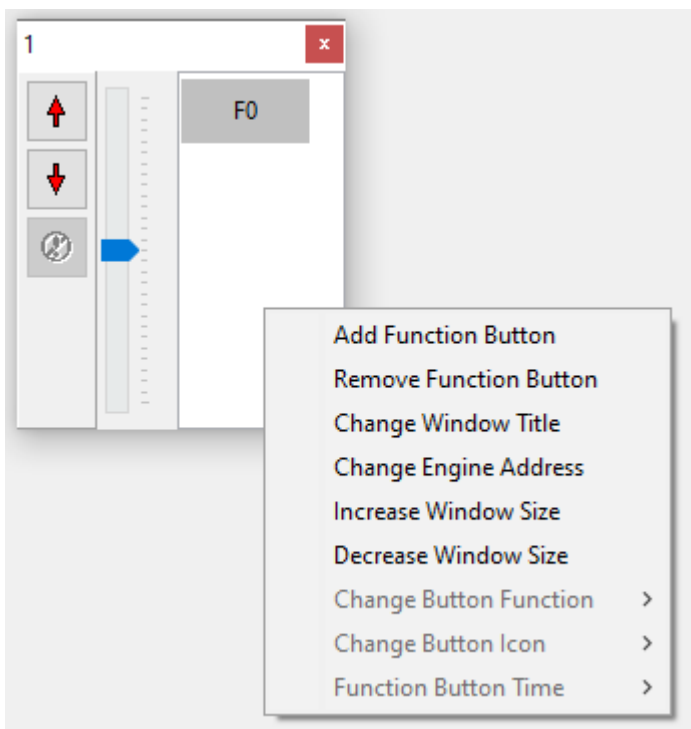
Each function button has a popup menu accessed by using the right-hand mouse button. The popup menu allows the function “on” colour to be changed and the button label to be changed for each engine address. All changes can be saved in the settings file.

## Engine Window – Windows Only

The user can add as many engine windows as required. The window allows control of one engine address. The user can add buttons to control engine functions. The window appears as below:



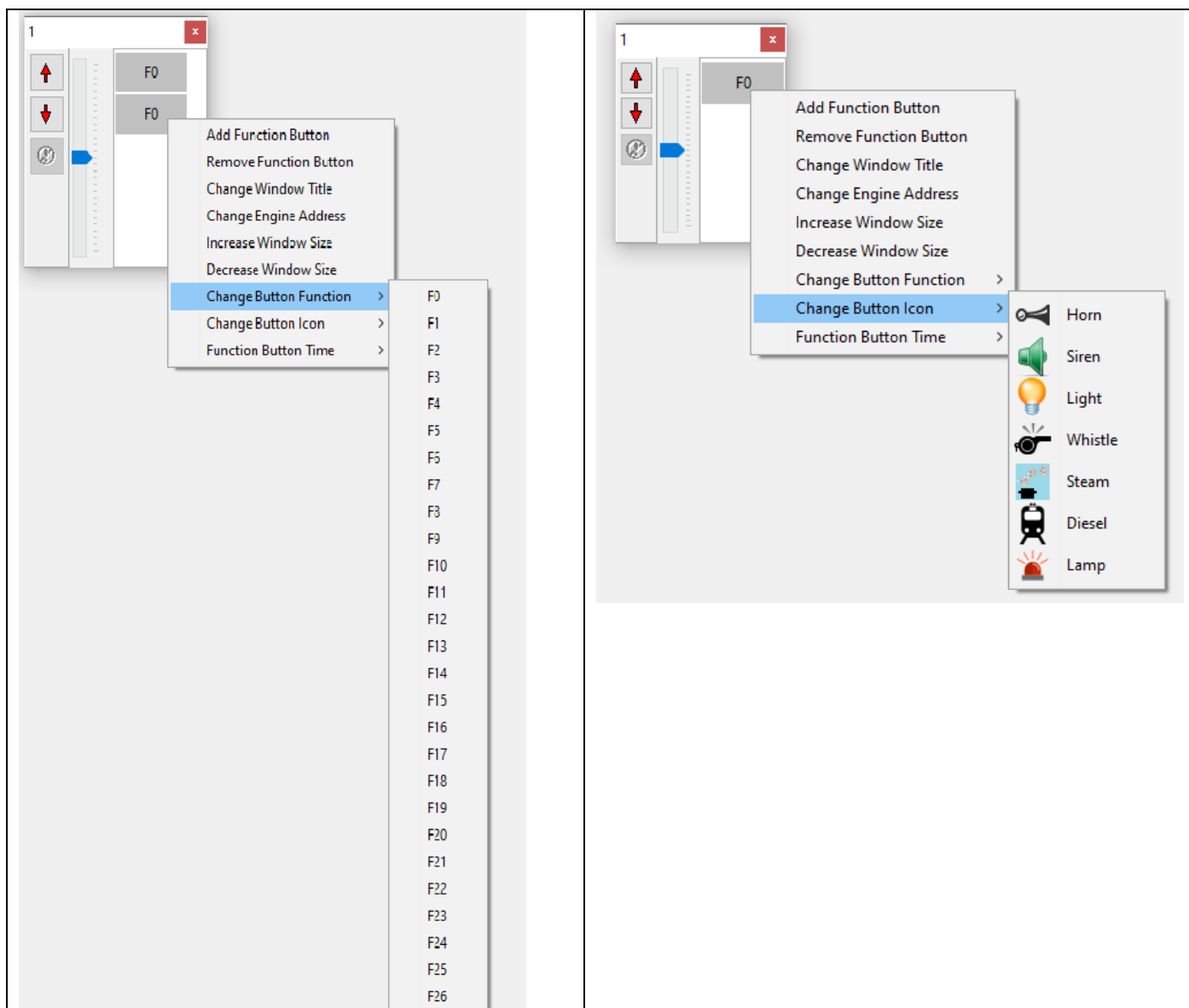
The user can configure the window by using the mouse right click button to access the following menu:



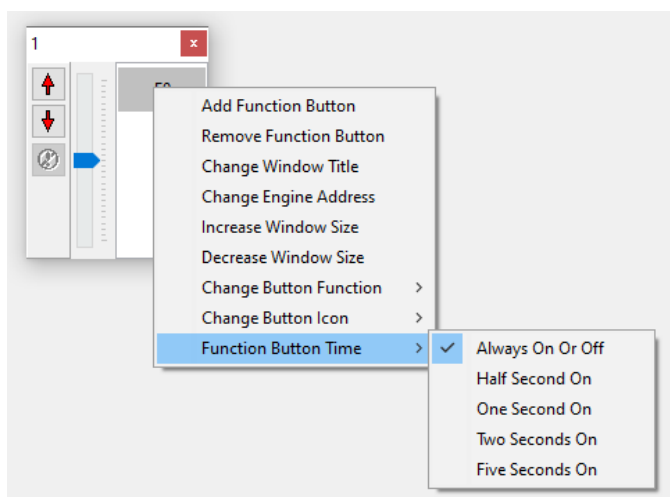
The user can add a new function button, delete (remove) a function button, change the window title, change the engine address or change the window size.



To change the function or icon used for a button, right click on the button and select the **Change Button Function** or **Change Button Icon** menu item then choose the new function or icon from those displayed:



The function buttons can be configured to automatically turn a function off after it has been turned on by using the **Function Button Time** menu option:



## DCC Packet Configuration Window

This window allows different DCC packets to be configured, on the track control window a button is provided to send the packet. The user can select the text for the button, the button background colour, the number of times the packet is sent and what type of packet is sent.

The window has four different pages to allow different DCC packets to be created:

The screenshot shows the 'Select & Format DCC Packet' window with the 'Basic Accessory' tab selected. The 'Address' is set to 1. The 'Accessory Value - DDD ...' section has radio buttons for 0 through 7, with 0 selected. The 'Active - C ...' section has radio buttons for 0 and 1, with 1 selected. The 'Use BroadcastAddress' checkbox is unchecked. The 'NMRAFORMAT' is displayed as: {preamble} 0 10AAAAA 0 1AAACDDD 0 EEEEEEEE 1. The 'Repeat Sends' is set to 1, 'Label' is 'Label your button', and 'Colour' is 'clSilver'. The 'OK' button is highlighted with a blue border.

The window above allows basic accessory packets to be configured, the user must provide the accessory address (1..512), value (0..7) to send to the decoder and a value for the activate bit (C). A broadcast can be created by selecting the **Use Broadcast Address** option.

The screenshot shows the 'Select & Format DCC Packet' window with the 'Extended Accessory' tab selected. The 'Address' is set to 1 and the 'Value' is set to 50. The 'Address Coding Method ...' section has radio buttons for 'Method 1 (JMRI Default)' and 'Method 2 (JMRI Alternate)', with 'Method 1' selected. The 'Use BroadcastAddress' checkbox is unchecked. A red warning message states: 'NMRA Standard maximum value is 31. However some manufacturers allow the full range 0 .. 255 to be used.' The 'NMRAFORMAT' is displayed as: {preamble} 0 10AAAAA 0 0AAA0AA1 0 000XXXXX 0 EEEEEEEE 1. The 'Repeat Sends' is set to 1, 'Label' is 'Label your button', and 'Colour' is 'clSilver'. The 'OK' button is highlighted with a blue border.

The window above allows extended accessory packets to be configured, the user must provide the accessory address (1..2044) and a value (0..255) to send to the decoder, values above 31 are not NMRA standard but some manufacturers support them. A broadcast can be created by selecting the **Use Broadcast Address** option.

Select & Format DCC Packet

BasicAccessory ExtendedAccessory Multi-Function F0-F28 Miscellaneous

Address:

☐ Use Broadcast Address

☐ Use Long Address

Function Group ...

☒ Group 1 F0-F4

☐ Group 2 F5-F8

☐ Group 2 F9-F12

☐ Expansion F13-F20

☐ Expansion F21-F28

BIT Settings ...

☐ F1 ☐ F2

☐ F3 ☐ F4

☐ FL

Repeat Sends:  Label:  Colour:

The window above allows some multi-function decoder packets to be configured, the user must provide an address and choose which function group to use. The bits in the function group must then be set or cleared. A broadcast can be created by selecting the **Use Broadcast Address** option.

Select & Format DCC Packet

BasicAccessory ExtendedAccessory Multi-Function F0-F28 Miscellaneous

Miscellaneous Packets ...

☒ Broadcast Stop Packet

☐ Broadcast Emergency Stop Packet

☐ Reset Packet

Repeat Sends:  Label:  Colour:

The window above allows for some miscellaneous packets to be configured; the repeat send option is disabled for these packets.

Service Mode - LC-DCC-PI06-M3-23.03.20-BETA-3.2A-OP2

File Decoder Reset

CV ...

CV Number: 1 CV Value: 2

Read Write

Long Address ...

1

Read Write

Set ACK Pulse Offset (APO)

Show Controller Communications

Write Generic Reset (CV8=8)

Track A/B ...

☒ Use Track B Output

Decoder Manufacturer ...

Manufacturer: LaisDCC

Model/Version No: 3

Read

CV29 ...

☐ Reverse Direction

☒ 28/128 Speed Steps

☐ DC Analog Enabled

☐ Railcom

☐ Complex Speed Curve

☐ Long Engine Address

Read Write

Speed Table ...

STEP	VALUE
1	90
2	92
3	94
4	96
5	98
6	100
7	102
8	104
9	106
10	108
11	110
12	112
13	114
14	116
15	54

Read Write Linear Fill Load Save

A B 76mA

This screen allows decoder service mode programming of all CVs from 1 to 1024. For special CVs such as CV29, 14-bit engine addresses (long address) or the speed table there are special controls to simplify programming. For all other CVs there is a simple control to read or write individual CV values.

Complete CV settings can be saved and restored using the **File** menu options **Save All CVs To File** and **Load All CVs From File** to store and recall CV settings to/from a file.

The DCC NMRA decoder ACK pulse is detected using the INA219 current monitor. The peak current before any service read is made and recorded and the peak current during the read is recorded. If the peak current increases this is considered to be a decoder acknowledge.

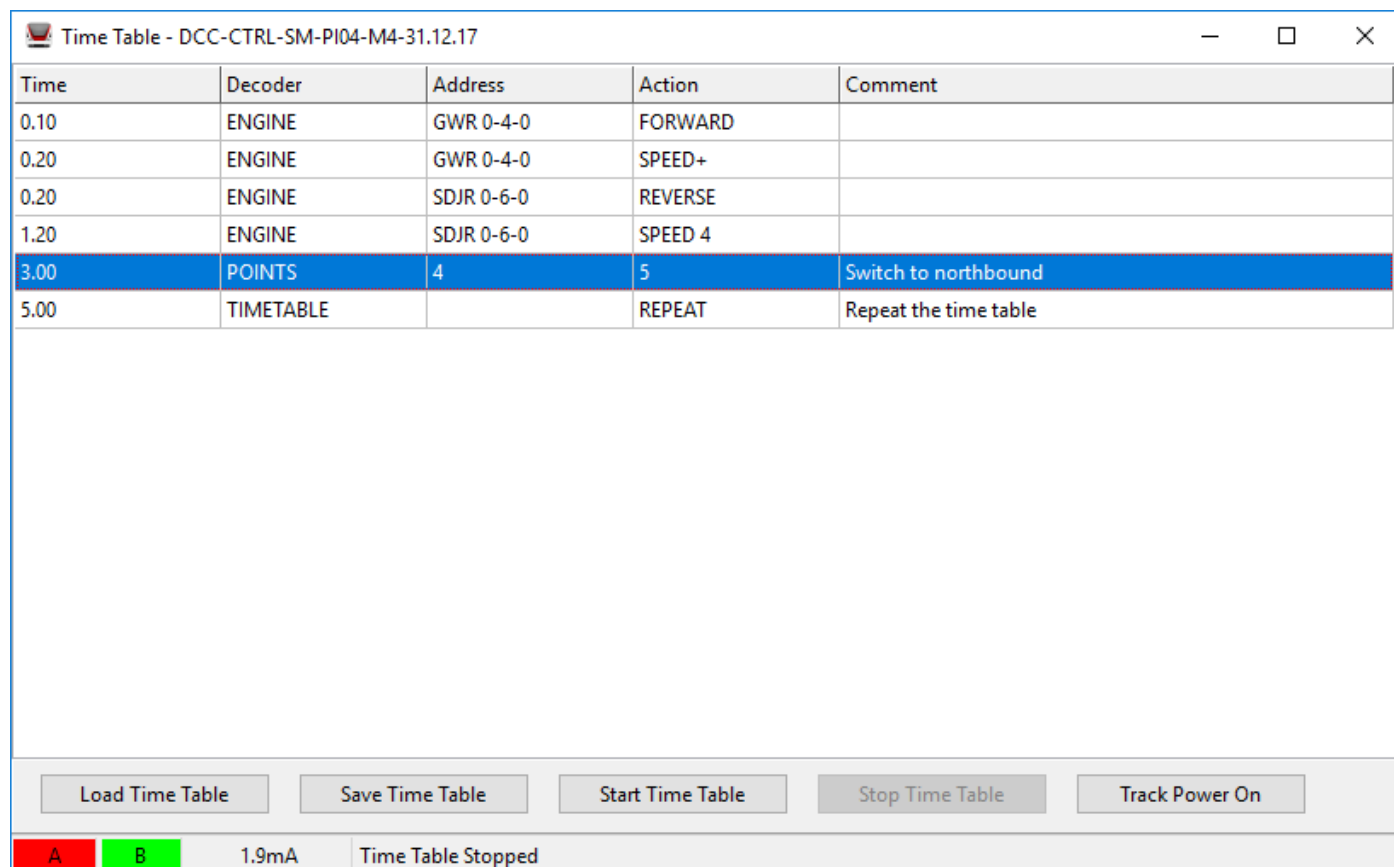
To overcome differences in decoders and noise on the current measurements the increase in current for an acknowledgment may be modified using the APO setting. This adjusts the acknowledgment pulse threshold from 20mA to 80mA (the default is 44mA).

There is also a **Decoder Reset** menu which contains resets for various DCC decoders. A reset can however be achieved by using the CV write facility and following the decoder manufacturers reset instructions.

Track power can be turned on and off using the **File** menu options **Track Power Off** and **Track Power On**. A green or red background in the **Track A/B** display shows track power status. To change from using track A or B simply click the **Use Track B Output** checkbox.

## Time Table Window

The time table window allows train function and accessory operations to be run from a time table. Engine speed and functions can be set as well as accessory on/off commands. The following is an example screen display of a running time table:

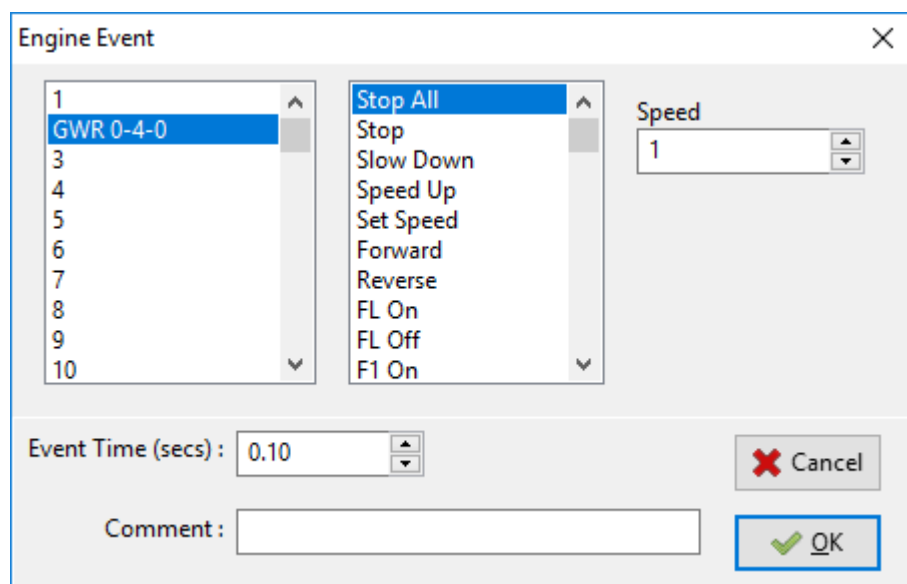


The screenshot shows a window titled "Time Table - DCC-CTRL-SM-PI04-M4-31.12.17". It contains a table with the following data:

Time	Decoder	Address	Action	Comment
0.10	ENGINE	GWR 0-4-0	FORWARD	
0.20	ENGINE	GWR 0-4-0	SPEED+	
0.20	ENGINE	SDJR 0-6-0	REVERSE	
1.20	ENGINE	SDJR 0-6-0	SPEED 4	
3.00	POINTS	4	5	Switch to northbound
5.00	TIMETABLE		REPEAT	Repeat the time table

Below the table are five buttons: "Load Time Table", "Save Time Table", "Start Time Table", "Stop Time Table", and "Track Power On". At the bottom, there are two status indicators: a red "A" and a green "B", followed by "1.9mA" and "Time Table Stopped".

Using the right mouse button, a menu can be accessed that allows adding, editing and deleting of time table events. The engine event form appears as below:



The screenshot shows a dialog box titled "Engine Event". It has two list boxes on the left. The first list box contains the following items: 1, GWR 0-4-0, 3, 4, 5, 6, 7, 8, 9, 10. The second list box contains the following items: Stop All, Stop, Slow Down, Speed Up, Set Speed, Forward, Reverse, FL On, FL Off, F1 On. To the right of these list boxes is a "Speed" label and a numeric input field with the value "1". At the bottom of the dialog, there is a label "Event Time (secs):" followed by a numeric input field with the value "0.10". Below this is a "Comment:" label followed by a text input field. At the bottom right, there are two buttons: "Cancel" and "OK".

This form can be used to add or edit engine time table events. Engine addresses can be changed to engine names using the right mouse button to access a popup menu.

The accessory event form appears as below:

Point Event

FIRST SIDING  
2  
3  
4  
SECOND SIDING  
6  
7  
8  
9  
10

Direction/Value ...

☒ 0 ☐ 1  
☐ 2 ☐ 3  
☐ 4 ☐ 5  
☐ 6 ☐ 7

Event Time (secs): 0.10

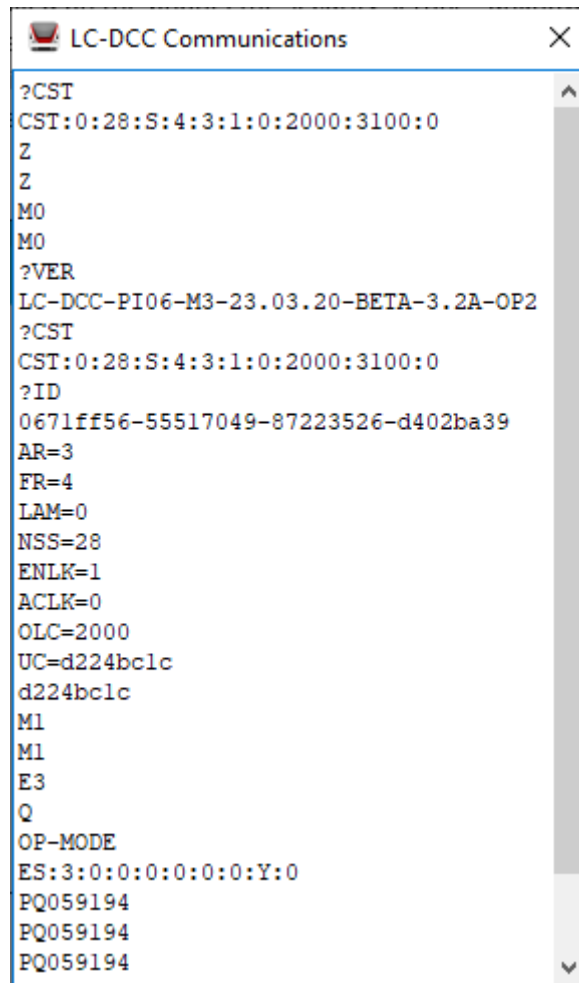
Comment:

Cancel OK

This form can be used to add or edit accessory (point) control events. Accessory addresses can be changed to meaningful names using the right mouse button to access a popup menu. Note any accessory value from 0 to 7 can be selected to be sent to the decoder. This supports all possible accessory modes available under NMRA DCC control.

## LC-DCC Controller Communications Window

On most screens there is a menu option or button to display the LC-DCC controller communication window. It appears as follows:



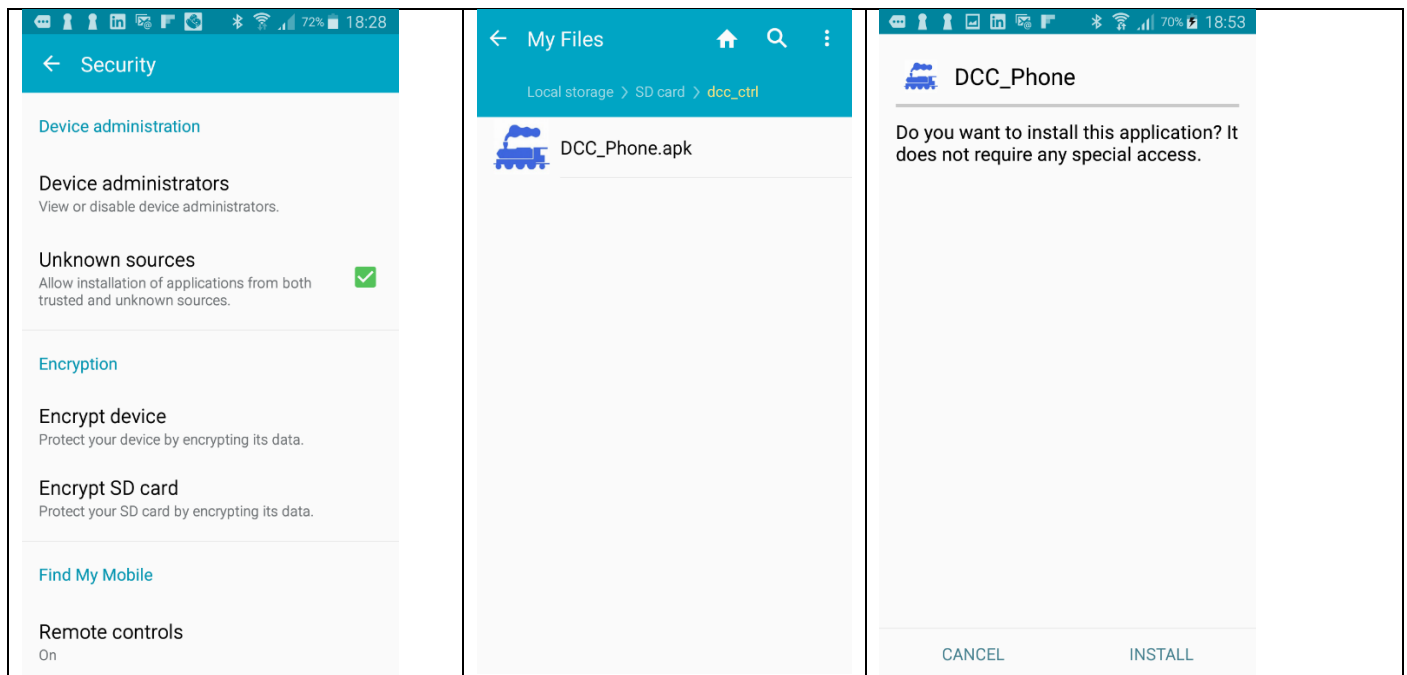
It contains the latest communication commands and responses to and from the LC-DCC controller. If you experience problems with your LC\_DCC controller we may ask you to copy this output and Email it to us so we can determine your problem. The output can be copied by using the popup menu accessed by clicking the right mouse button.

## Android Bluetooth Application

The Android application is stored in the .zip file as dcc\_phone.apk. The application is installed onto the Android device using the following instructions:

### Installing Android Application

To allow the software to be installed on the phone, the security setting “unknown sources” must be enabled, see below. The file dcc\_phone.apk is then copied to the Android device either by using USB or some other mechanism. The software is installed by running the “My Files” application, locating the dcc\_phone.apk file copied to the phone and selecting it then choosing install.



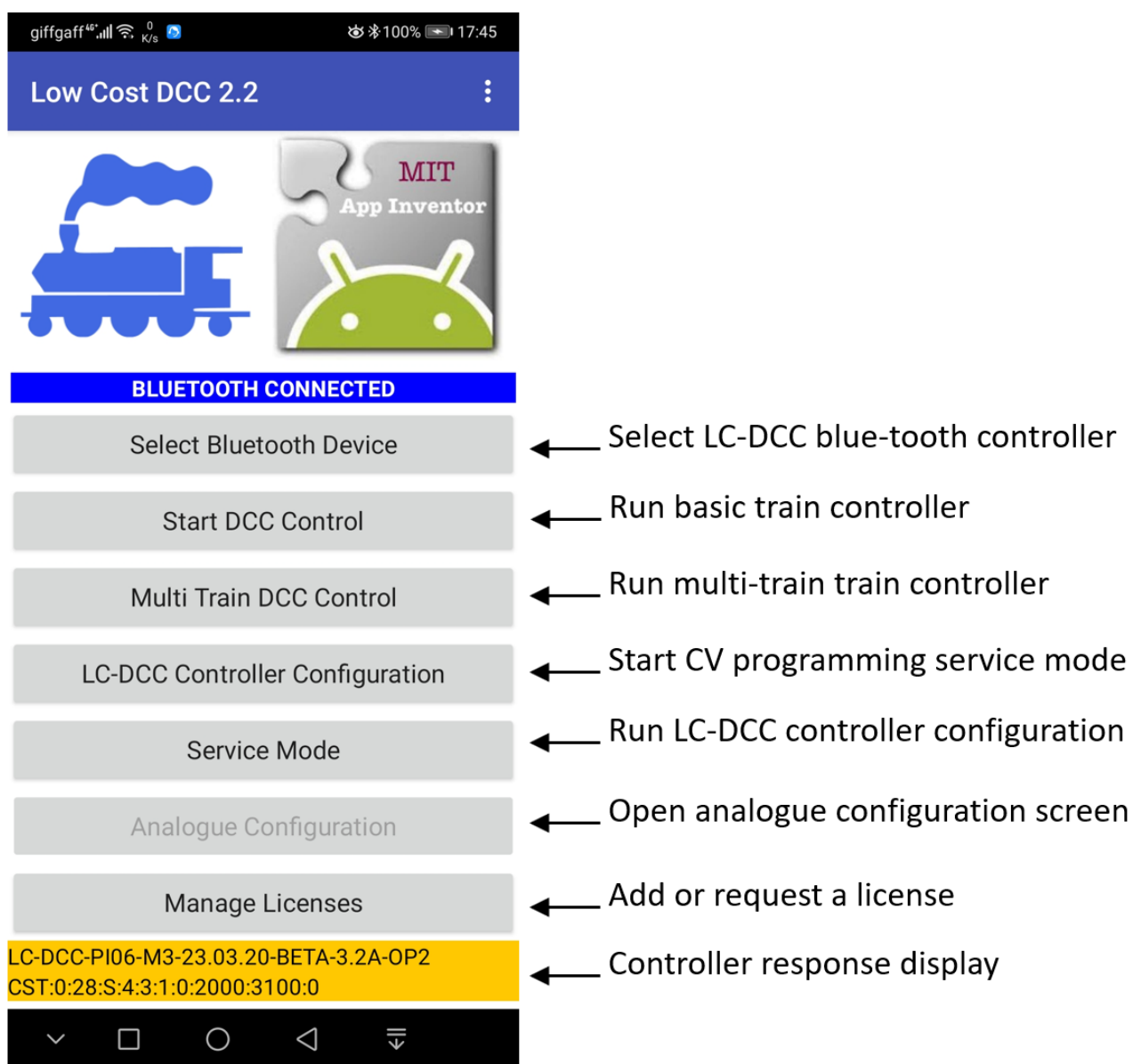


## Start-up Screen

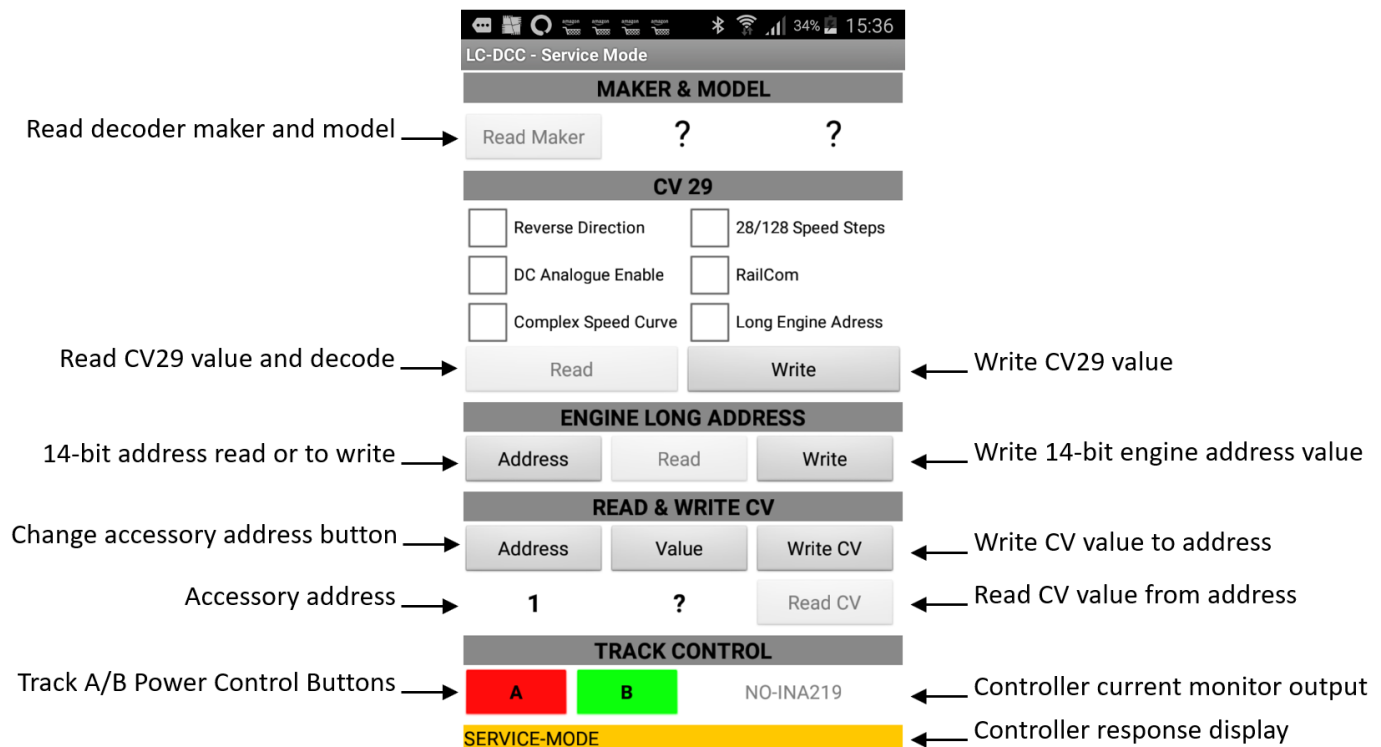
When the Android application is launched the screen shown below is displayed. The user must select a Bluetooth connection by clicking the **Select Bluetooth Device** button and choosing a Bluetooth connection. Once a connection has been chosen the other buttons become enabled and the user can select from the main options:

- Start DCC Control
- Multi Train DCC Control
- Service Mode
- DCC Controller Configuration
- Analogue Control Configuration
- License Configuration/Request

These different screens are explained in the following sections.



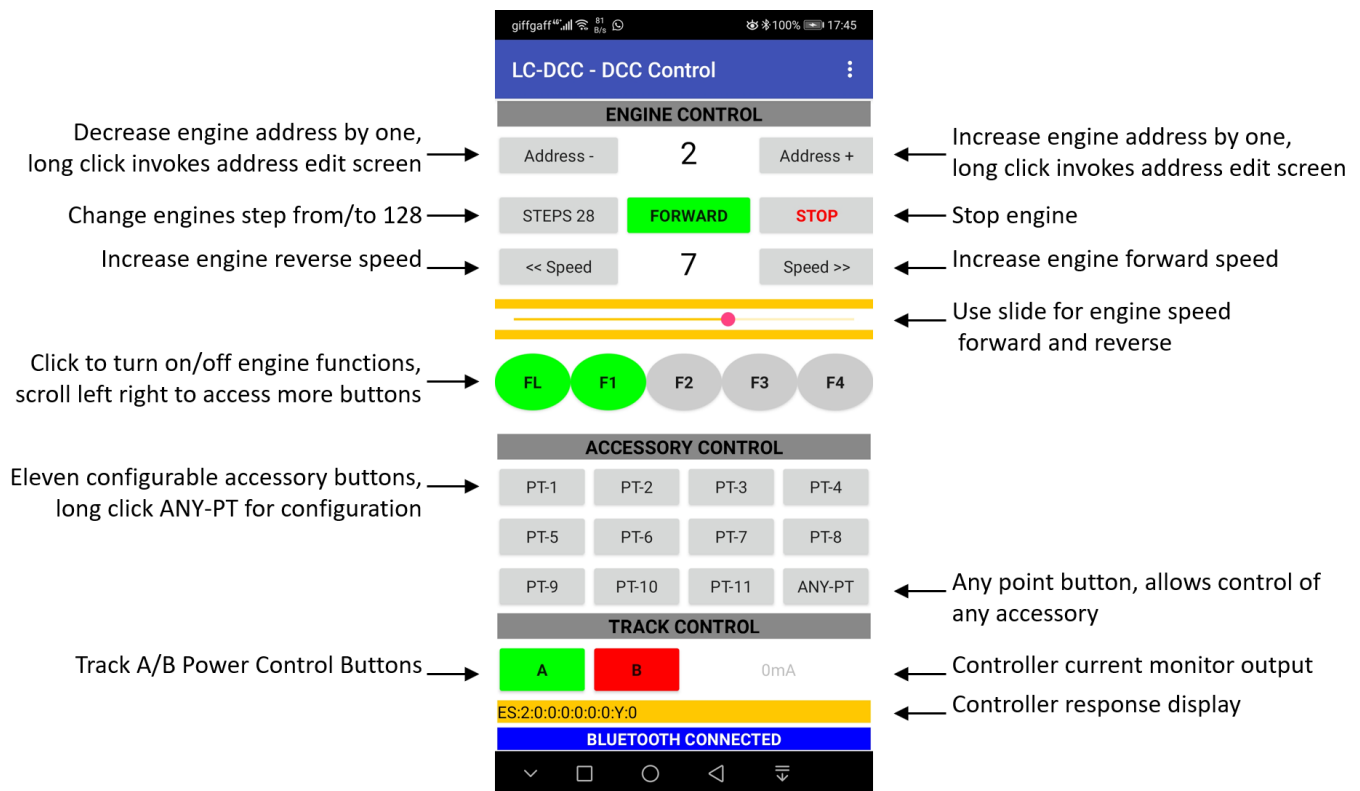
## Service Mode Screen



The service mode screen allows programming of decoder CV values. To read the decoder make and model click the **Read Maker** button. To program the CV29 value use the check boxes to set: reverse direction, 28/128 speed steps, analogue enable, railcom, complex speed curve and long engine addresses then click **Write**. To read CV29 click the **Read** button in the CV29 section of the screen. You can also use the **Read CV** and **Write CV** buttons for CV29 by entering the CV address as 29. To change between track A and track B perform a *long button press* on either button **A** or **B**. To read any CV address, select the address by clicking the **Address** button and then click **Read CV**. To Write any CV address, select the address in the address box, select the value by clicking the Value button and then click **Write**. A 14-bit engine address can be read or written by using the **Address** button in the **Engine Long Address** section to set the 14-bit address. The **Read** and **Write** buttons can be used to read or write the 14-bit address to or from CVs 17 and 18.

A *long button press* is when the button is pressed and “held down” for a short period.

## DCC Control Screen



The DCC control screen allows the control of engine decoders and accessory decoders. Use **Address +** and **Address -** buttons to select an engine address (long click for address edit screen). An **X** displayed in the engine speed display indicates that the engine is locked by another user. Once an unlocked engine address is chosen the engine speed will be displayed and the engine control buttons **Forward**, **Reverse**, **Stop**, **Speed >>** and **<< Speed** will become enabled, the engine speed slider will also become enabled. Use the FL (F0) to F28 buttons for the engine decoder functions. These buttons are highlighted light green when a decoder function is activated. To access different function buttons, scroll left or right over the function buttons.

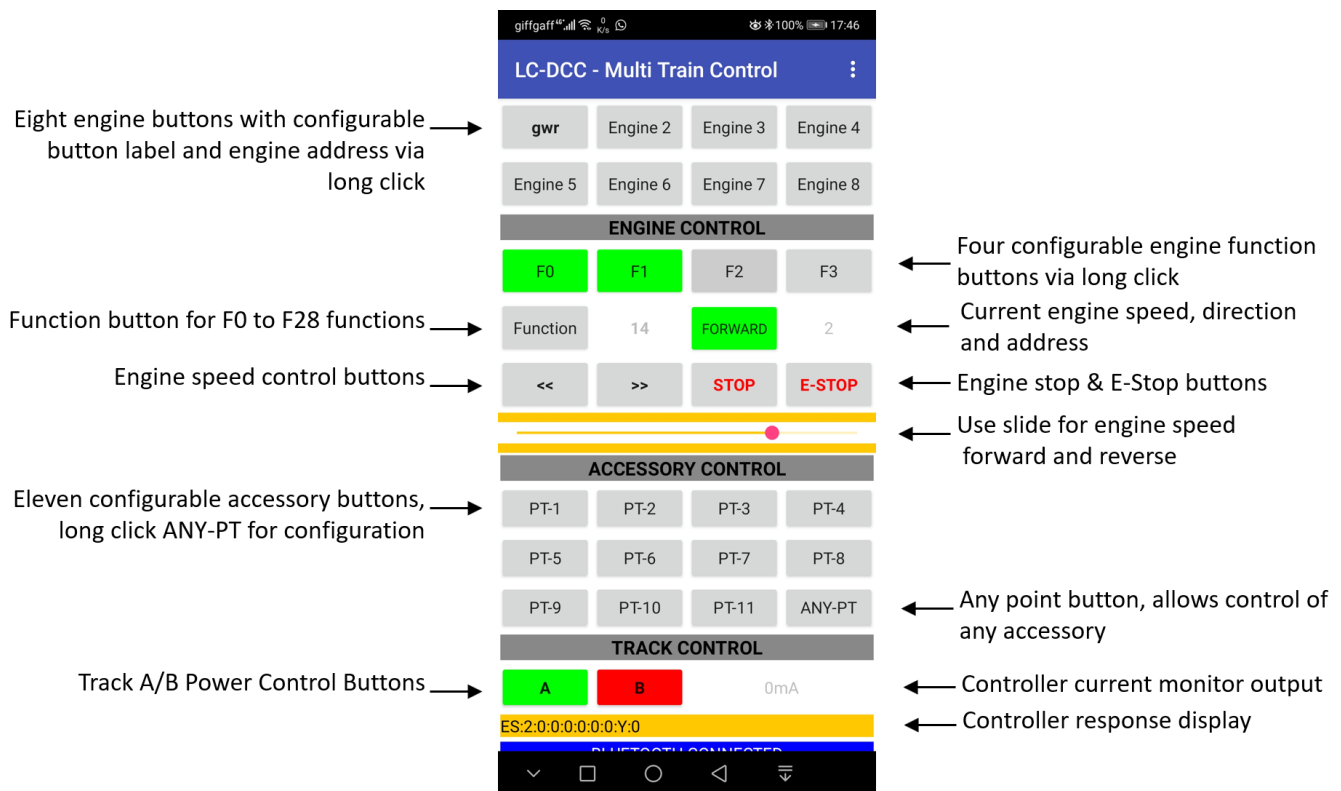
There are eleven accessory decoder buttons that can be used to control configured addresses and values. There is an accessory button labelled **ANY-PT** that can be used to send any accessory decoder value to any address. The **ANY-PT** button is also used with a long click to configure the eleven accessory buttons. The accessory button labels, decoder addresses and values can be configured. These eleven buttons can send decoder values 0/1, 2/3, 4/5 and 6/7 depending on whether the button indicates on or off.

Tracks A and B can be turned on or off by holding down either the **A** or **B** buttons.

If an INA219 is present in the LC-DCC controller then the current in milli-amps being used by the controller will be displayed next to the **B** button.

If the engine address selected is controlled by a potentiometer this will be indicated and any change in engine speed due to a change in the potentiometer will be displayed.

## Multi Train DCC Control Screen



The multi train control screen allows the user to control up to five engines using engine buttons on the screen. Each engine address and button text are set by long clicking the engine button. When the engine button is clicked the engine address for that button will be selected in the LC-DCC controller and the address will also be displayed on the screen.

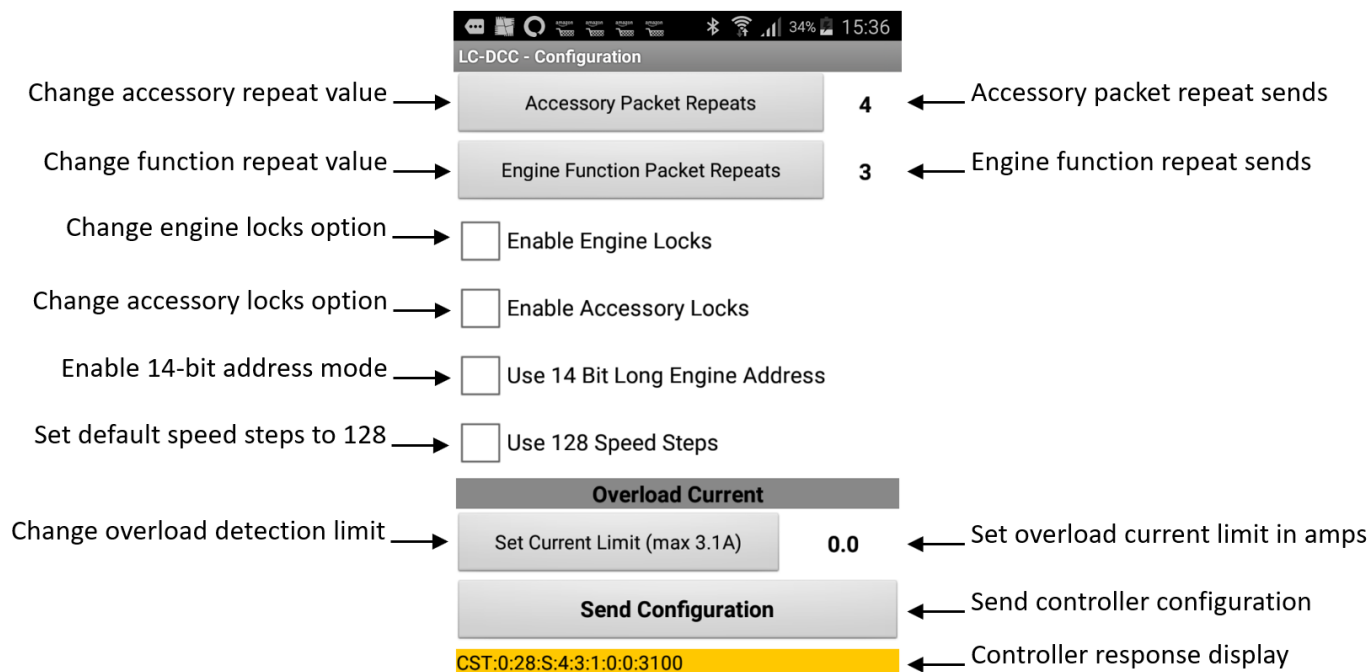
If an engine is already in use by another user then the engine speed display will show an **X** and no buttons will be enabled.

There are eleven accessory decoder buttons that can be used to control configured addresses and values. There is an accessory button labelled **Any** that can be used to send any accessory decoder value to any address. The **Any** button is also used with a long click to configure the eleven accessory buttons. The accessory button labels, decoder addresses and values can be configured. These eleven buttons can send decoder values 0/1, 2/3, 4/5 and 6/7 depending on whether the button indicates on or off.

The track power can be controlled by using long clicks on the track **A** and track **B** buttons. The current being used is also displayed under the track control section if an INA219 is present in the LC-DCC controller.

If the engine address selected is controlled by a potentiometer this will be indicated and any change in engine speed due to a change in the potentiometer will be displayed.

## Configuration Screen



The configuration screen is used to configure:

**Accessory Packet Repeats** – The number of times an accessory packet is sent to an accessory decoder

**Engine Function Packet Repeats** – The number of times a function packet is sent to an engine (255=continuous)

**Enable Engine Locks** – When there are multiple users, lock an engine to a user

**Enable Accessory Locks** – When there are multiple users, lock and accessory to a user

**Use 14 Bit Long Engine Address** – This enables use of long engine addresses from 1 to 9999

**Use 128 Speed Steps** – This sets the default engine speed steps to 128 if checked otherwise 28

**Set Current Limit** – Allows the user to change the default current overload detection limit

Once the settings have been configured use the **Send Configuration** button to send the settings to the LC-DCC controller.

## Analogue Configuration Screen

The screenshot shows the 'LC-DCC Analogue Configuration' screen. At the top, a status bar displays various icons and the time 17:29. Below the title bar, there are eight rows for potentiometer configuration. Each row has a checkbox, a label (e.g., 'POT1 ENABLE'), a 'Loco Address' button, and a numeric address field. Annotations with arrows point to these elements: 'Enable potentiometer engine control' points to the first checkbox; 'POT configuration will be disabled if associated ADC is not connected' points to the first row; 'Potentiometer 1 engine address' points to the first address field; and 'Cancel analogue configuration' points to the 'Cancel' button. A red box at the bottom contains instructions: 'Use address buttons to change engine address values for each potentiometer. Enable or disable each potentiometer via the check boxes.'

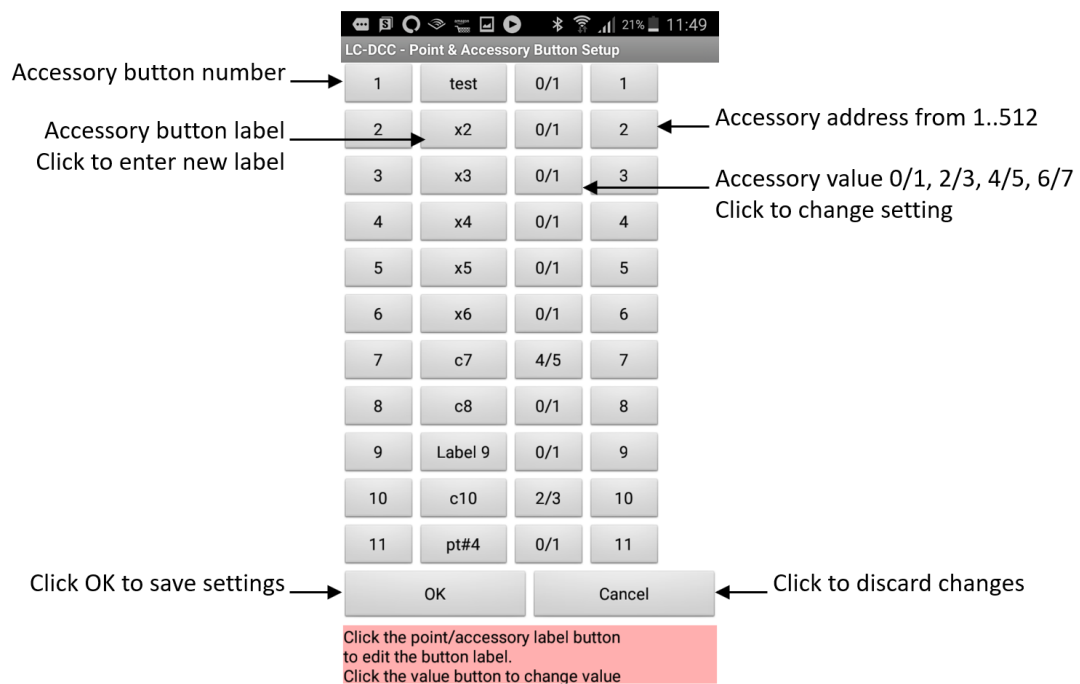
Potentiometer	ENABLE	Loco Address	Engine Address
POT1	<input checked="" type="checkbox"/>	Loco Address	1
POT2	<input checked="" type="checkbox"/>	Loco Address	2
POT3	<input checked="" type="checkbox"/>	Loco Address	3
POT4	<input checked="" type="checkbox"/>	Loco Address	4
POT5	<input type="checkbox"/>	Loco Address	5
POT6	<input type="checkbox"/>	Loco Address	6
POT7	<input type="checkbox"/>	Loco Address	7
POT8	<input type="checkbox"/>	Loco Address	9998

Send analogue configuration →   ← Cancel analogue configuration

Use address buttons to change engine address values for each potentiometer. Enable or disable each potentiometer via the check boxes.

This screen allows the user to configure up to eight potentiometers used for analogue control if ADC cards are fitted to the LC-DCC controller. Each potentiometer can be enabled or disabled using the **ENABLE** check box. The engine address controlled by each potentiometer can be assigned by clicking the **address** buttons. The analogue configuration is sent to the LC-DCC controller once the **OK** button is clicked. Check boxes will be disabled if the potentiometer is not fitted or enabled.

## Accessory Button Configuration



This screen allows the user to configure up to eleven accessory buttons that can be labelled, address configured and value configured. To change the button label simply click the label you want to change and enter a new label when prompted. To change the accessory address simply click the address you want to change, the address screen will appear and you can enter a new address. The values can be configured as 0/1, 2/3, 4/5, and 6/7 by clicking the button showing the value. The value used to control the accessory decoder is determined by whether the accessory button on the screen is indicating on or off. The settings for these eleven buttons are saved even when you close the android application.

## Engine & Accessory Address

LC-DCC Accessory Decoder Address

**Enter Accessory Address**

Maximum Address 512

+ + + +

0 3 0 0

- - - -

OK Cancel

300

LC-DCC Short Engine Address

**Enter Short Engine Address**

Maximum Address 127

+ + + +

0 0 0 1

- - - -

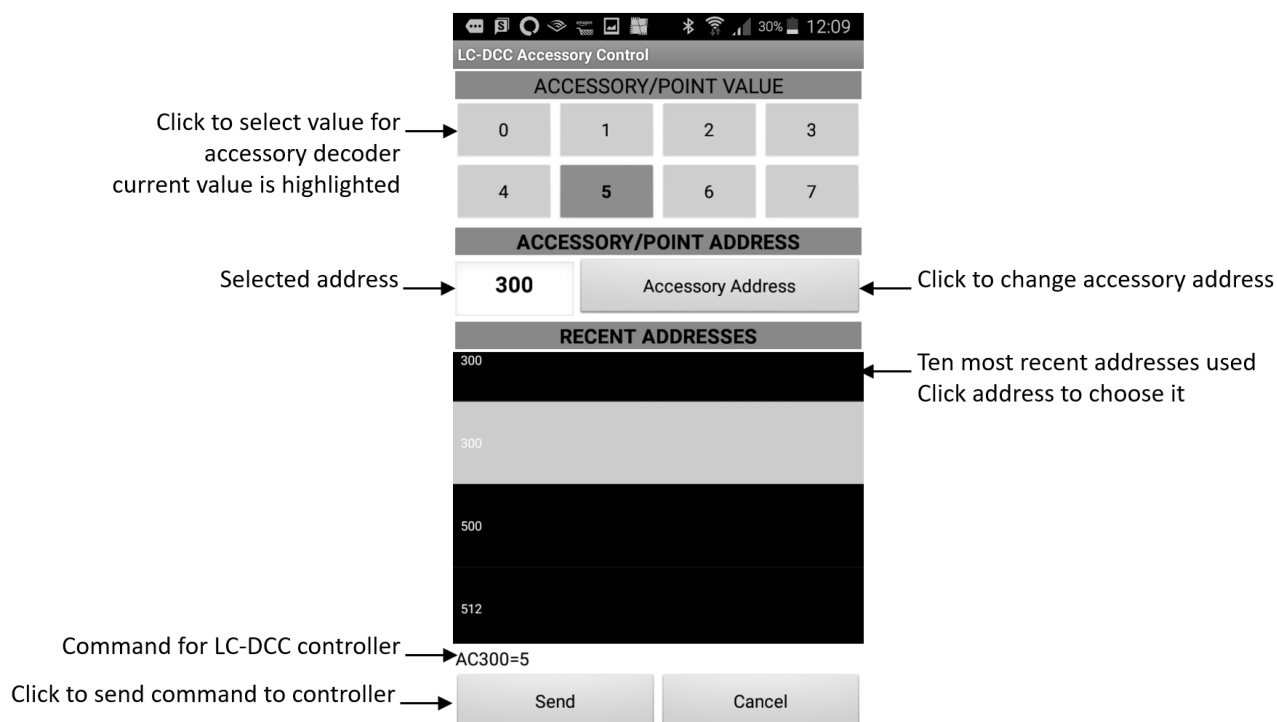
OK Cancel

1

The above screens are used to select either an engine address (short or long) or an accessory address. The plus and minus buttons are used to change the address digit, the digits automatically wrap at 9 and 0. The **OK** button is used to accept the address, the **Cancel** button is used to cancel the address edit.



## Accessory Control Screen

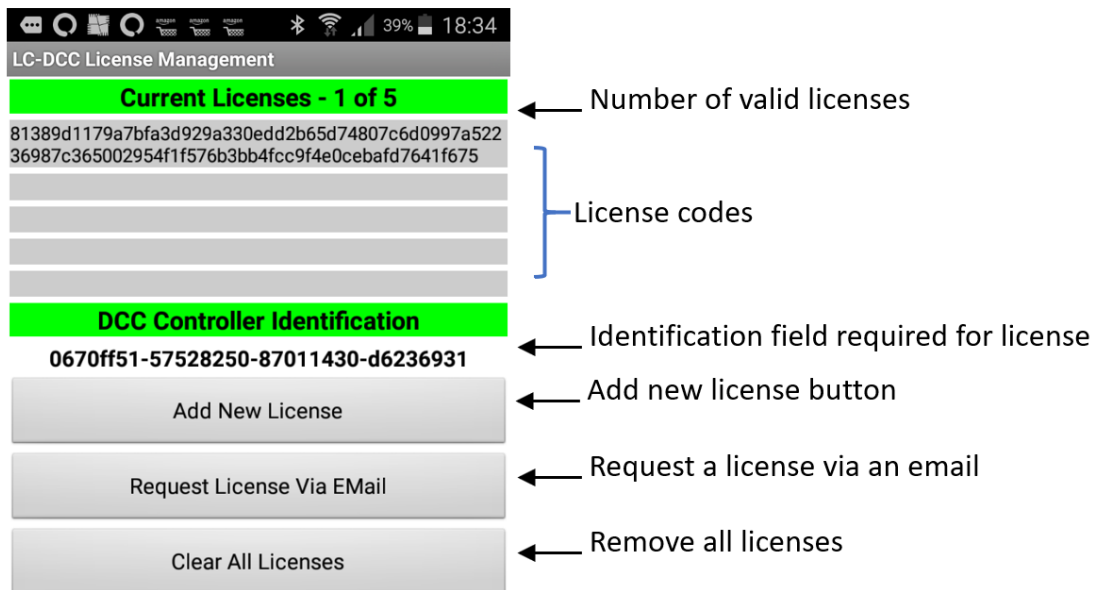


This screen is used to send an accessory control packet to the LC-DCC controller. The user must select the accessory address and the value to send. The values are displayed on eight buttons, the selected value is a highlighted button.

The address can be selected by either clicking the **Accessory Address** button or by selecting a previous address from the list. The controller command display is automatically updated when the address or value is changed. The **Send** button is used to send the command to the LC-DCC controller. The **Cancel** button is used to cancel the accessory command.

## License Management Screen

The license management screen below allows for the configuration of licenses to use with the LC-DCC controller. Each LC-DCC controller needs a different license. If you require multiple licenses contact [sales@swws.co.uk](mailto:sales@swws.co.uk) for bulk discount.



The LC-DCC controller identification is displayed in the middle of the screen. This must be supplied to [support@swws.co.uk](mailto:support@swws.co.uk) to obtain a license code. Up to five license codes are currently support in the Android application which allows connection to up to five different LC-DCC controllers. The license can be emailed to [support@swws.co.uk](mailto:support@swws.co.uk) by clicking the **Request License Via Email** button.

## Android Voice Application

The Android voice application is stored in the .zip file as lc\_dcc\_voice.apk. The application is installed onto the Android device using the same instructions as for the Android Bluetooth application in the previous section.

The application uses voice commands to control the LC-DCC controller, the predefined voice commands are:

Voice Command	LC-DCC Command	Notes
OPERATE	M1	Enter controller operational mode to run trains and accessories
STANDBY	M0	Enter standby mode, only generates DCC idle packets
FORWARD	>>	Command engine into forward direction
REVERSE	<<	Command engine into reverse direction
EMERGENCY STOP	!!	Emergency stop all engines
STOP	00	Stop currently selected engine
All STOP	HH	Stop all engines
POWER ON	PA1 PB1	Enable DCC power to both A & B tracks
POWER OFF	PA0 PB0	Disable DCC power to both A & B tracks
TRACK A ON	PA1	Enable DCC power to track A
TRACK A OFF	PA0	Disable DCC power to track A
TRACK B ON	PB1	Enable DCC power to track B
TRACK B OFF	PB0	Disable DCC power to track B
<b>ENGINE</b> <address>	E<address>	Set engine address for following commands
<b>SPEED</b> <value>	S<value>	Set speed value for currently selected engine address Example: "SPEED 28" sends controller command S28
<b>FUNCTION</b> <number> <ON or OFF>	F<number>=<1-0>	Set engine function number (0..28) either ON (1) or OFF (0) Example: "FUNCTION 2 ON" sends controller command F2=1
<b>POINT</b> <address> VALUE <value>	AC<address>=<value>	Set accessory address to value Example: "POINT 4 VALUE 1" sends controller command AC4=1
<b>ACCESSORY</b> <address> VALUE <value>	AC<address>=<value>	Set accessory address to value Example: "ACCESSORY 4 VALUE 1" sends to controller AC4=1

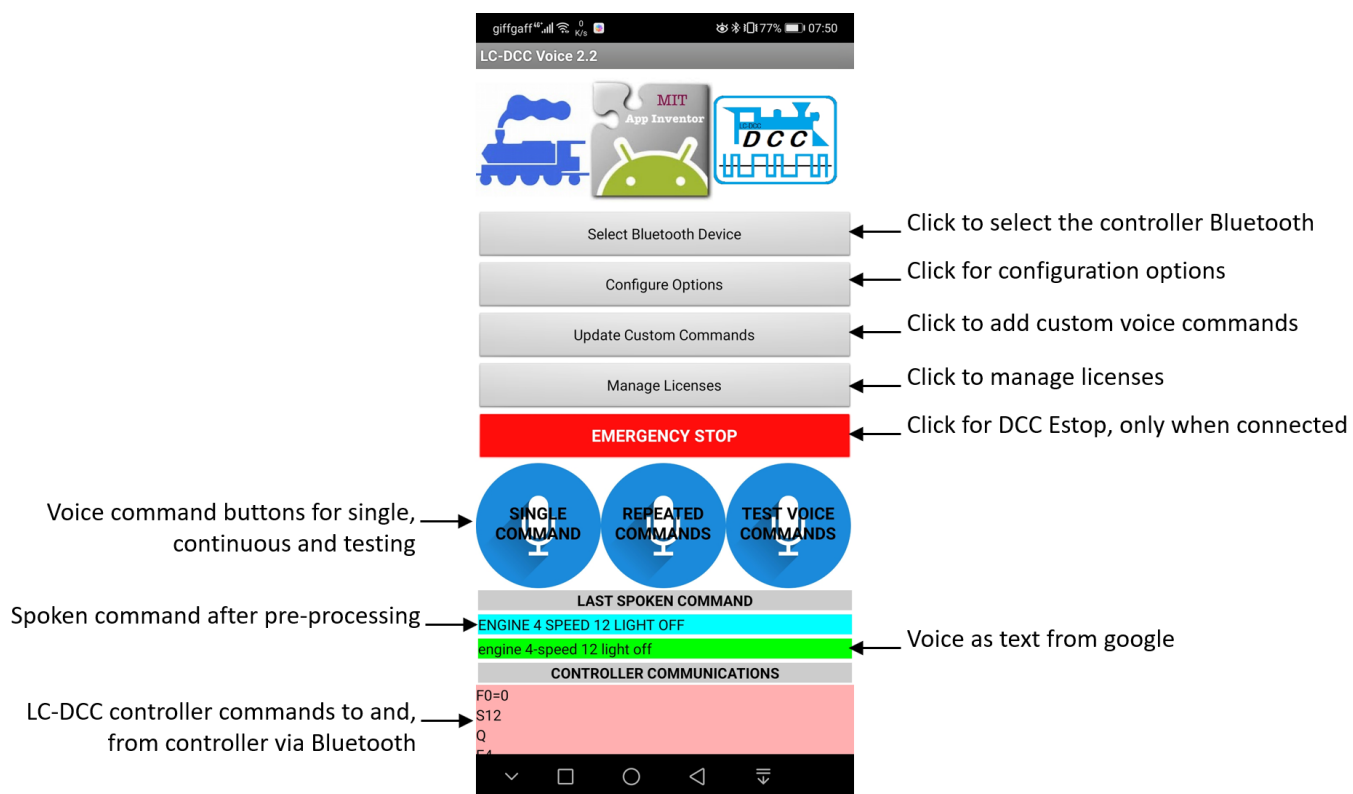
These commands can be overridden by the user custom commands. All words highlighted in bold in the table above can be redefined on the configuration screen.

## Start-up Screen

When the Android application is launched the screen shown below is displayed. The user must select a Bluetooth connection by clicking the **Select Bluetooth Device** button and choosing a Bluetooth connection. Once a connection has been chosen the other buttons become enabled and the user can select from the following options:

- Configuration Options
- Update Custom Commands
- Manage Licenses
- Single Voice Command
- Continuous Voice Commands
- Test Voice Commands
- Emergency Stop

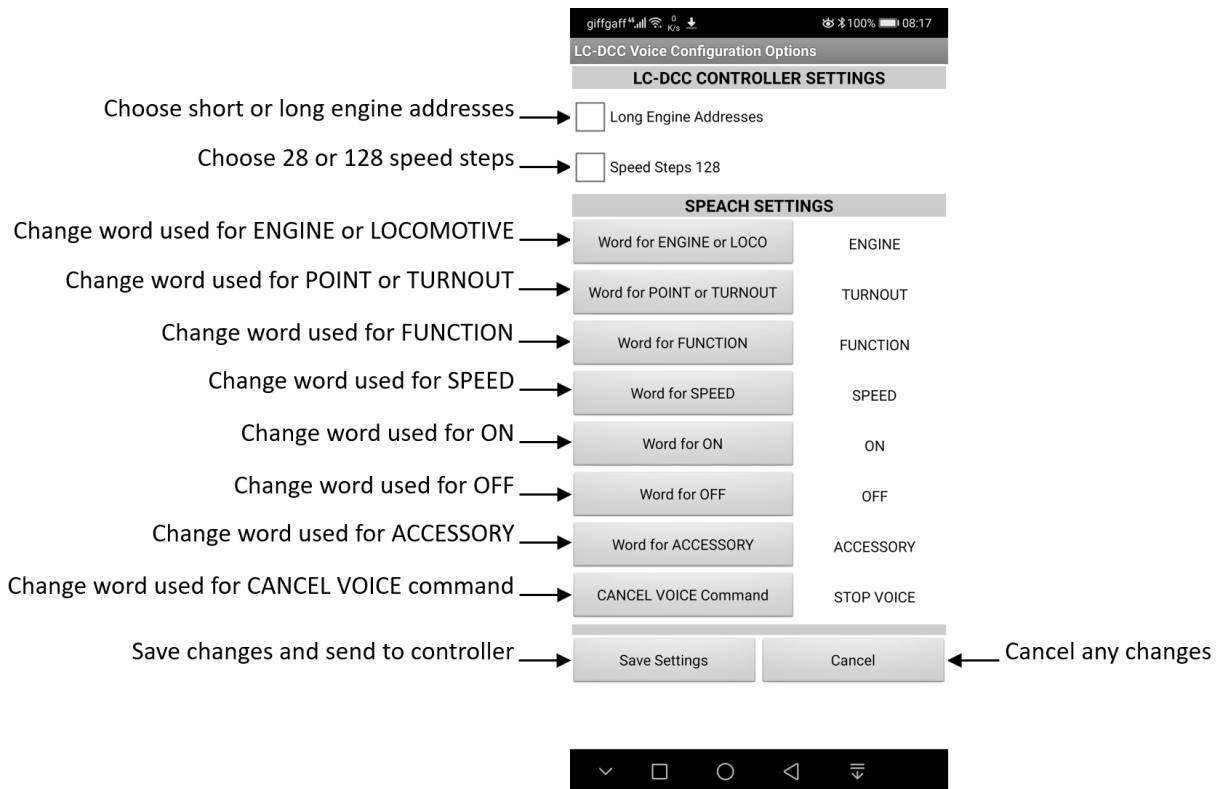
These different screens are explained in the following sections.



When using repeated commands, the application will continuously wait for voice commands. To cancel this use the voice command "CANCEL VOICE". This command can be redefined in the configuration screen.

## Configuration Options Screen

The configuration screen allows some of the words used by the application to be changed for locality use. The user may also configure some basic LC-DCC controller settings. Any settings not configured on the screen can be configured by creating a custom command. The configuration screen appears as follows:



At the top are the LC-DCC controller configuration settings, currently the use can change short/long engine address and speed steps. The second part of the screen contains all application words that can be changed by the user. To change a word: click the button and when prompted speak your new word. The new word will appear in the label opposite the button just clicked. Finally, at the bottom of the screen are two buttons: save to accept the changes made and cancel to discard any changes. All settings are saved even when the application is closed.

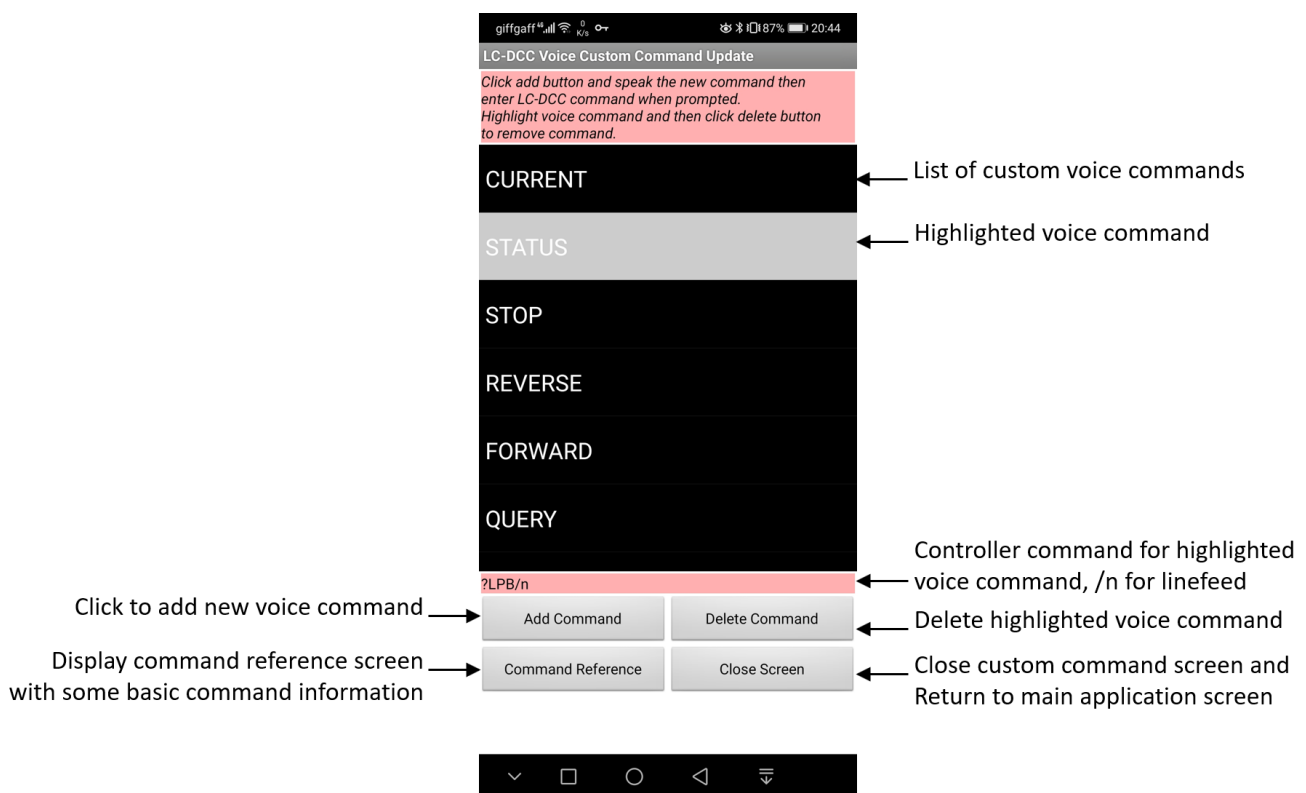
## Update Custom Commands Screen

The custom commands screen allows the user to add their own LC-DCC controller commands to the application. This extends the vocabulary that the application can support. As shown below a number of new commands have been added, the highlighted command **STATUS** generates a controller command **?LPB/n**. The /n indicates line-feed for the application, the new controller command may contain as many commands as required, any line-feeds need to be **/n**.

To add a new command, click the **Add Command** button, a google voice input screen will appear (see next section), speak your command, you will then be prompted to enter the LC-DCC controller command in a text edit screen. Once the process is complete the new command will appear on the list.

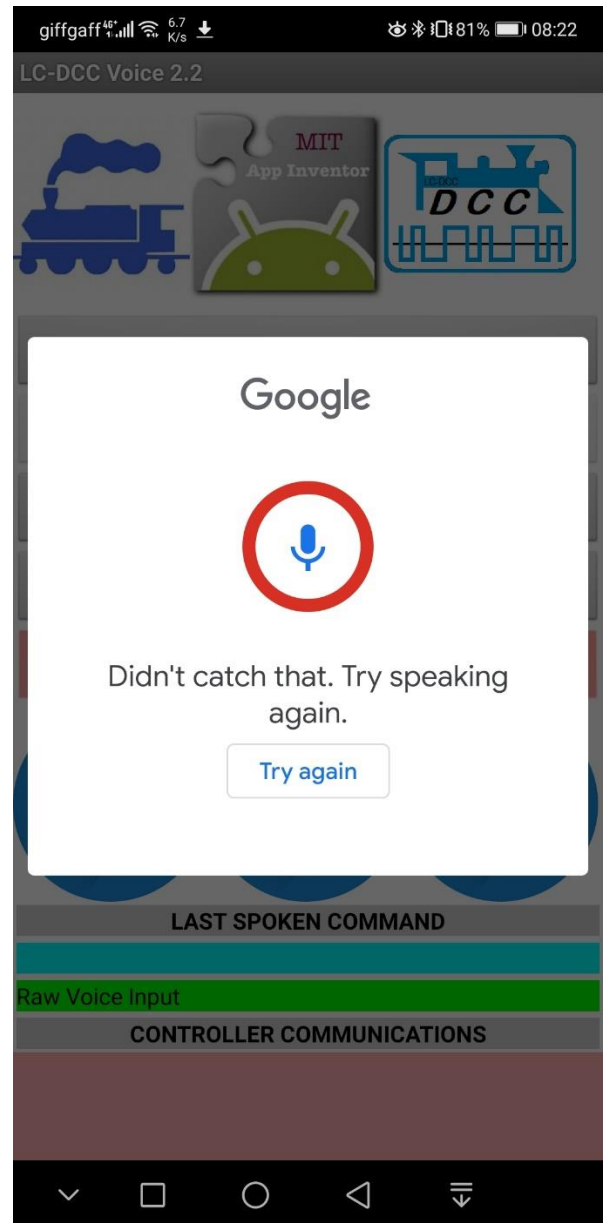
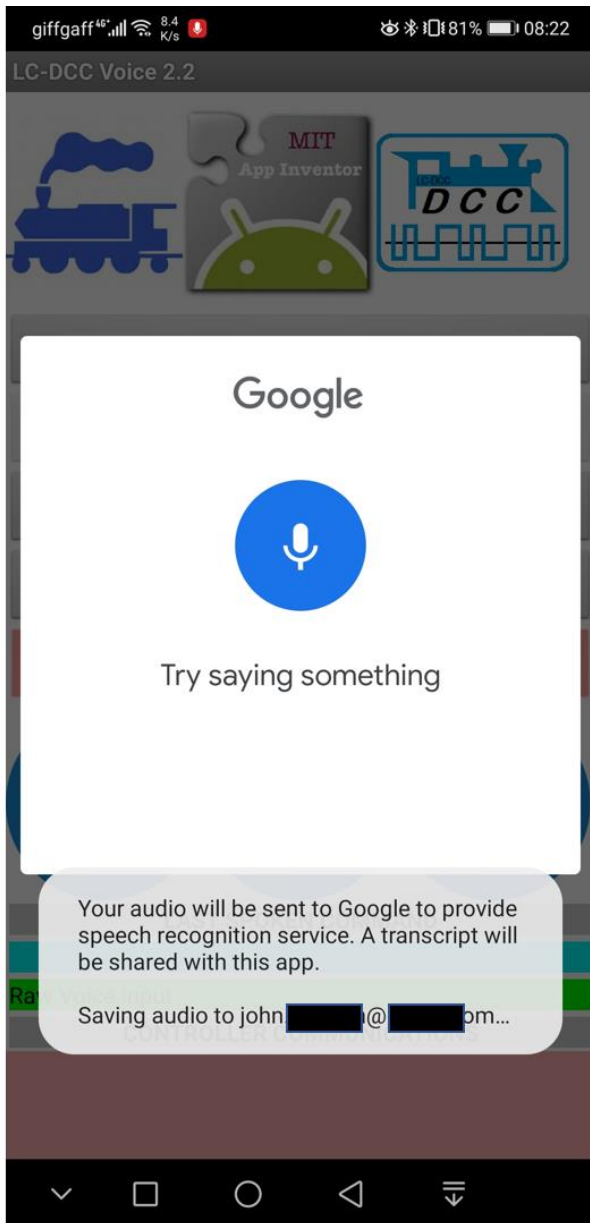
To delete a command simply highlight it and then click the **Delete Command** button

A list of common LC-DCC controller commands can be displayed by clicking the **Command Reference** button. For a more detailed list of commands consult the LC-DCC Serial/Bluetooth/WIFI Command Reference.



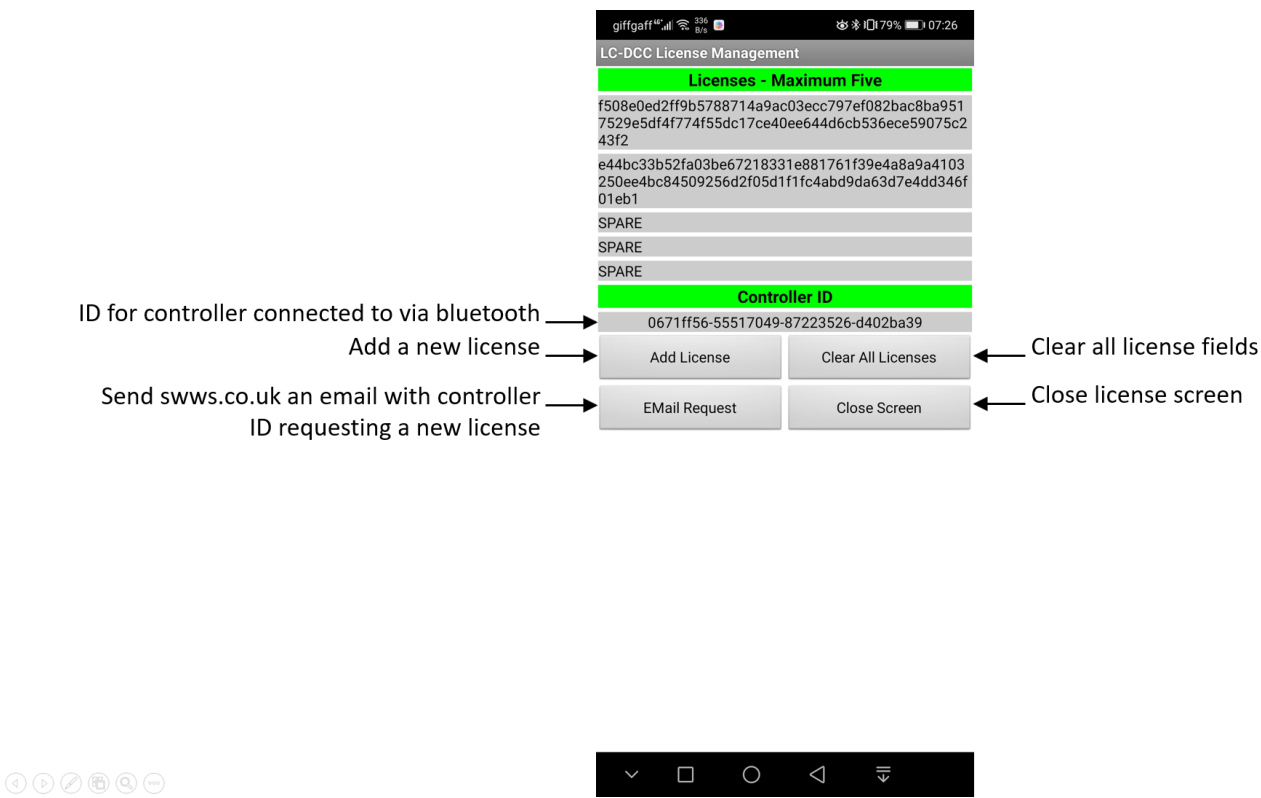
## Google Voice Input Screen

For the voice application to work your phone or tablet must be connected to the internet as the voice to text processing is performed in the “cloud”. When you choose to speak a command, the following screen will appear (left), if a problem recording your voice happens then the right-hand screen will appear:



Manage Licenses Screen

This screen allows the management of controller/software license activation codes. The screen appears as follows:



The screen displays current license activation codes along with the ID for the LC-DCC controller currently connected to the application via Bluetooth. To request a license, you can simply click the email request button and this should open a phone email application with an email to send to [support@swws.co.uk](mailto:support@swws.co.uk) for an activation code.



## Version Change History

### **April 2020**

Added Android voice application. Improved Android and Windows applications. Added packet queue.

Added eight potentiometers into controller and applications.

### **December 2019**

Android accessory button update. Added packet control to Windows application. Updated engine form.

### **March 2019**

Initial version.