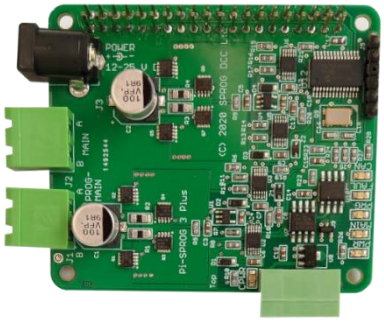
SPROG DCC

SPROG 3 Plus
Pi-SPROG 3 v2
Pi-SPROG 3 plus

DCC Decoder Programmer and Command Station

User Guide



Revision History 6
Acknowledgements 6
Introduction 7
Features 7
Requirements 7
Specification – Layout Power Districts 8
Specification – Service Mode Programming 8
Hardware Details 10
SPROG 3 Plus 10
SPROG 3 Plus Front Panel 10
SPROG 3 Plus Rear Panel 10
USB Interface 10
Layout Interface 10
CBUS Interface 10
Pi-SPROG 3 v2 11
PI-SPROG 3 v2 Status LEDs 11
Pi-SPROG 3 Plus 12
PI-SPROG 3 Plus Status LEDs 12
CBUS Interface 12
GPIO Pass-Through Connector (Pi-SPROGs Only) 13
Software Installation 14
Pi-SPROG 3 Plus and Pi-SPROG 3 v2 14
Connecting to the Raspberry Pi Access Point 14
Accessing the Raspberry Pi Desktop 14
SPROG 3 Plus 15
Install FTDI drivers (Windows/SPROG 3 Plus) 15
Install DecoderPro 15
Hardware Setup 16
Connect the Power Supply 16
Connect the Programming Track or Layout 16
Connect the CBUS (Optional) – Plus Models Only 16
Initial Setup Using Terminal Interface (Optional) 17
The SPROG DCC Generation 5 Interface 17
Checking the Firmware Version 17

Setting the CAN ID 17
Setting the Node Number 18
Changing Node Variables (NVs) 18
Reset SPROG 3 Plus to Factory Defaults 19
Getting Started with JMRI (DecoderPro and PanelPro) 20
Setting the JMRI Connection Preferences 20
SPROG DCC Generation 5 JMRI Tools 22
Pi-SPROG 3 v2 JMRI Tools 22
Pi-SPROG 3 Plus and SPROG 3 Plus JMRI Tools 22
Console 22
CBUS Send Event Frame 23
Node Manager 23
Command Station Monitor 25
Mode Switching with SPROG 3 Plus and Pi-SPROG 3 Plus 26
Programming Track Off When Not Programming 26
Programming Track Follows Main When Not Programming 27
Programming Track is Auto-reverse District 27
Mode Switching with Pi-SPROG 3 v2 28
Programmer (Service mode programming) 28
Command Station (Ops mode programming) 28
Voltage/Current Meter 28
The JMRI System console 29
The JMRI Power Manager 30
A Brief Introduction to CBUS 31
Operation with JMRI and CBUS Cabs (Throttles) 33
Release/Dispatch 33
Steal/Share 33
Connecting Throttle Devices to the Raspberry Pi Access Point 33
CBUS Events and DCC Accessory Control 34
Turnout Control – in Brief 34
Pi-SPROG 3 v2 34
SPROG 3 Plus and Pi-SPROG 3 Plus 34
Short Events 34
Long Events 34

Mapping from CBUS Event Numbers to DCC Accessory Address 35
Firmware version prior to 2e Beta 2 35
Firmware version 2e Beta 2 and later 35
Detailed Explanation of DCC Accessory Addressing 36
CBUS Node Variables 37
NV1 Command Station Number 37
NV2 User Flags 37
NV3 Operation Flags 38
NV4 Debug Flags 38
NV5 Programming Track Power Mode 38
SPROG 3 Plus and Pi-SPROG 3 Plus 38
Pi-SPROG 3 v2 38
NV6 Programming Track Current Limit, and 39
NV13 Main Track Current Limit 39
NV7 Input Voltage 39
NV8 Main Track Current, and 39
NV14 Programming Track Current 39
NV9 DCC Accessory Packet Repeat Count 39
NV10 Multimeter Mode 39
NV11, NV12 Node Number to Map to DCC 39
NV15 Main Track Current High-Water Mark, and 39
NV16 Programming Track Current High-Water Mark 39
NV17 Setup Mode 39
NV18 CAN ID 40
NV19, NV20 Node Number 40
NV21 DCC Preamble Bits 40
NV22 CAN Disable 40
CBUS Opcodes 41
OPCs Consumed 41
OPCs Output 42
CBUS Events 43
Events Consumed 43
Events Produced 43
Firmware Updates 44

Differences to MERG CANCMD 46
Known Issues 47
Glossary 49
Troubleshooting 50
Useful Links 51

Revision History

Date	Version	Firmware	JMRI	Comment
Jun-20	Draft 0.1	1.a.3	4.19.7	First version
Dec-20	Draft 0.2	2.a.1	4.21.3	V2 SPROG 3 Plus, Pi-SPROG 3 Plus
Feb-21	Draft 0.3	2.a.2	4.21.3	First draft release
Mar-21	1.0	2.b.1	4.21.3	Support for SPROG 3 v2
Apr-21	1.1	2.b.4	4.23.3	First Production release
Apr-21	1.3	2.b.5	4.23.3	Pi-SPROG 3 v2 Photographs, update Known Issues
May-21	1.4	2.b.6	4.23.4	Reformat. Throttle support. Differences to MERG CANCMD. Complete TODO items. Fix-typos
Sep-21	1.5	2.c.1	4.24	See known issues list
Sep-21	1.6	2.c.2	4.24	See known issues list
Sep-21	1.6.1	2.c.3	4.24	See known issues list
Jan-23	1.6.2	2.e.2	5.00	See known issues list Added connection and password details for WiFi access point Option for Event to accessory mapping offset
Jan-23	1.6.3	N/A	N/A	Added details of GPIO usage and wifi password

Firmware and JMRI columns refer to the earliest versions to which a User Guide version applies.

Acknowledgements

CBUS® is a registered trademark of Dr Michael Bolton.

Introduction

The SPROG 3 Plus, PI-SPROG 3 v2 and Pi-SPROG 3 Plus are DCC decoder programmers designed to support CBUS, a CAN based network developed by members of MERG for model railway control.

The SPROG 3 Plus connects to the USB port of a personal computer or similar device.

The Pi-SPROG 3 v2 and Pi-SPROG 3 Plus attach directly to the GPIO connector of a Raspberry-Pi computer. A Raspberry Pi 4 Model B is recommended for the best performance.

All are supported by the free JMRI software (http://www.jmri.org/).

For the remainder of this document references to SPROG 3 Plus apply equally to the Pi-SPROG 3 v2 and Pi-SPROG 3 Plus unless stated otherwise.

The SPROG 3 Plus and Pi-SPROG 3 Plus have two track outputs. With the supplied 5A power supply each output can supply up to 2.5A to the layout

The Pi-SPROG 3 v2 is the replacement for the Pi-SPROG 3 which was, in turn, a replacement (via firmware upgrade) for the Pi-SPROG One, and has a single track output that can be used in programmer or command station modes supplying up to 2.5A to the layout. It is not possible to upgrade from Pi-SPROG One or Pi-SPROG 3 to Pi-SPROG 3 v2, other than by purchasing the new hardware.

Features

- The SPROG 3 plus has an isolated USB interface to protect the host computer
- The Pi-SPROG 3 Plus has an isolated CBUS interface
- Flexible operating modes (Plus models only)
  - One layout power district and one service mode programmer output
  - Two layout power districts, one with auto switching to service mode programming track
  - Two layout power districts, one with auto-reverse
- Flexible operating modes (v2)
  - Programmer mode supporting service mode programming and test running of locos on a programming track
  - Command station mode for full layout control
- Programs virtually ***all*** NMRA compliant DCC decoders
- No extra hardware required for programming sound decoders (e.g. programming booster) – but does not allow loading of sound projects.
- Easy to use graphical interface with DecoderPro

Requirements

- JMRI from http://www.jmri.org/

- PI-SPROG 3 Plus and Pi-SPROG 3 v2 requirements:
  - Raspberry-Pi (R-Pi 4 model B recommended)
  - Raspbian OS image for the Pi-SPROG 3 Plus or v2
    - See the image creation instructions on our website
    - A suitable image may be purchased on micro SD card at the same time as the Pi-SPROG 3 Plus/v2 or downloaded from our website
    - No drivers are required
- SPROG 3 Plus requirements:
  - FTDI VCP Windows USB drivers from https://ftdichip.com/drivers/vcp-drivers/ for the SPROG 3 Plus
  - No drivers are required for popular Linux distributions

Specification – Layout Power Districts

Parameter	Minimum	Nominal	Maximum	Units	Note
DC Input supply voltage	12V		25V	V	1
Vin supply current – idle		50		mA	
Vin supply current – Operating Layout			5 2.5	A	2 3
Operating Temperature Range		25		°C	
Output Load – Operating Layout			2.5	A	4

Table 1 Specification/Operating Conditions

Notes:

1. The track voltage will be fractionally lower than the power supply input voltage.
2. Total for both outputs SPROG 3 Plus and Pi-SPROG 3 Plus
3. Pi-SPROG 3 v2
4. Per output. The combined output current limits must always be set lower than the power supply capability.

Specification – Service Mode Programming

3	Minimum	Nominal	Maximum	Units	Note
Vin supply current – programming		**300**		**mA**	**5**
Output Load - programming			**250**	**mA**	**5**

Table 2 Specification/Operating Conditions

Notes:

5. SPROG 3 Plus will remove track power if output current exceeds 250mA as measured 100ms after applying power. Surge current during decoder power-up may be considerably greater than this, but is allowed for.

Hardware Details

SPROG 3 Plus

SPROG 3 Plus Front Panel

SPROG 3 Plus Rear Panel

USB Interface

USB reception (Rx) is indicated by a brief flash of the USB Rx LED.

USB transmission (Tx) is indicated by a brief flash of the USB Tx LED

Layout Interface

Input power is indicated by the red PWR LED.

The track activity is indicated by the MAIN and PROG red LEDs, which will illuminate steadily when their respective outputs are powered up.

When the PROG output is being used for a programming track, the PROG LED will flash slowly when power is applied during a programming operation.

A fast flash on the MAIN or PROG LEDs indicates an overload condition.

CBUS Interface

The CBUS interface status is indicated with three red LEDs.

The CBUS RX LED flashes when a CBUS frame is received.

The CBUS TX LED flashes when a CBUS frame is transmitted.

The CBUS CFLT LED indicates a fault on the CBUS interface.

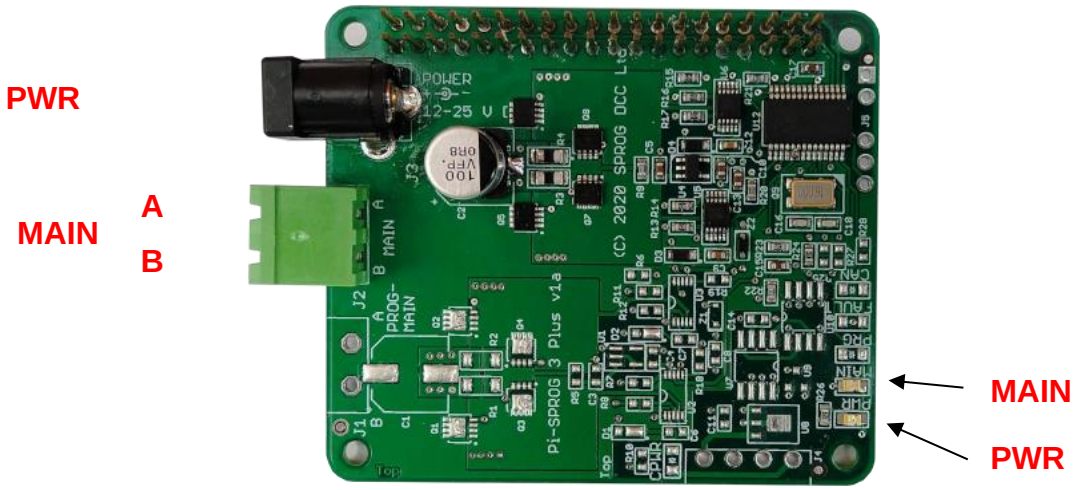
All three LEDs are illuminated when the module starts up. The CBUS RX and TX LEDS remain illuminated until data is received or transmitted on CBUS.

The CFLT LED will extinguish on the first CBUS frame and then illuminate if there is a fault on the CBUS interface.

When the module starts up (e.g., after applying power) the PROG, MAIN and CFLT LEDs will illuminate and then extinguish at 0.5 second intervals. CFLT will remain in the case of a fault on the CBUS interface (e.g., not connected).

The PWR LED will remain illuminated whilst power is applied.

Pi-SPROG 3 v2



PI-SPROG 3 v2 Status LEDs

Input power is indicated by the red PWR LED (See known Issues in this document).

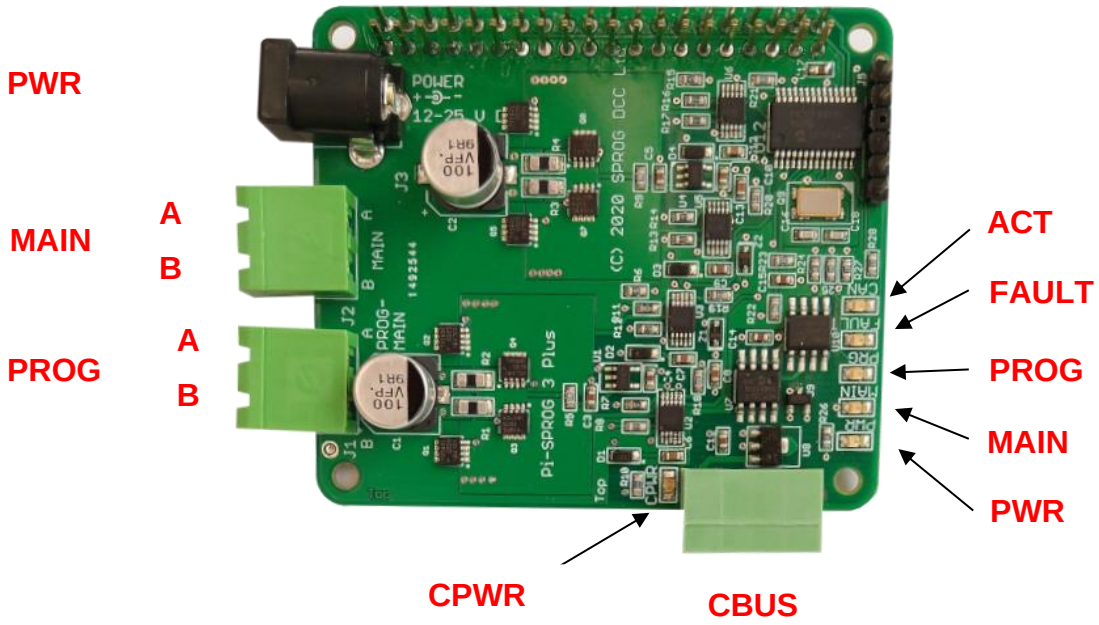
The track activity is indicated by the MAIN LED, which will illuminate steadily when the output is powered up.

The MAIN LED will flash slowly when power is applied during a service mode (programming track) programming operation.

A fast flash on the MAIN LED indicates an overload condition.

The MAIN LED will flash if Power is not connected.

Pi-SPROG 3 Plus



PROG output, CBUS interface, and associated components are not present on Pi-SPROG 3 v2.

PI-SPROG 3 Plus Status LEDs

Input power is indicated by the red PWR LED.

The track activity is indicated by the MAIN and PROG red LEDs, which will illuminate steadily when their respective outputs are powered up.

When the PROG output is being used for a programming track, the PROG LED will flash slowly when power is applied during a programming operation.

A fast flash on the MAIN or PROG LEDs indicates an overload condition on the respective output.

The MAIN and PROG/MAIN LEDs will flash together and the if power is not connected.

CBUS Interface

The CBUS interface status is indicated with three red LEDs.

The CBUS activity, ACT, LED flashes on CBUS frame transmission and reception.

The CBUS FAULT LED indicates a fault on the CBUS interface.

Both ACT and FAULT LEDs are illuminated when the module starts up.

The ACT LED will extinguish on the first CBUS frame and then flash for each subsequent CBUS frame.

The FAULT LED will extinguish on the first CBUS frame and then illuminate if there is a

fault on the CBUS interface.

The CBUS power, CPWR, LED indicates the presence of power on the CBUS connection.

When the module starts up (e.g., after applying power) the PROG, MAIN and CFLT LEDs will illuminate and then extinguish 0.5 second intervals. CFLT will remain in the case of a fault on the CBUS interface (e.g., not connected).

The PWR LED will remain illuminated whilst power is applied.

GPIO Pass-Through Connector (Pi-SPROGs Only)

The Pi-SPROG 3v2 and Pi-SPROG 3 Plus pass all Raspberry Pi GPIO pins through to a 40-pin header that allows further accessory boards (“hats”) to be connected. There are, however, some restrictions on using these pins.

The UART Rx, Tx, CTS and RTS pins (pins 8, 10, 36 and 11) are used by the Pi-SPROG 3v2 and Pi-SPROG 3 Plus. No further connection should be made to these pins.

All other pins may safely be used.

Software Installation

Pi-SPROG 3 Plus and Pi-SPROG 3 v2

The Pi-SPROG family require an SD card to be created with a suitable operating system (OS) and software image. Three possible routes to obtain the SD card image are:

- Purchase a ready built SD card from us
- Download the image from our website https://www.sprog-dcc.co.uk/download-page
- Create your own image following the instructions on our website https://www.sprog-dcc.co.uk/download-page

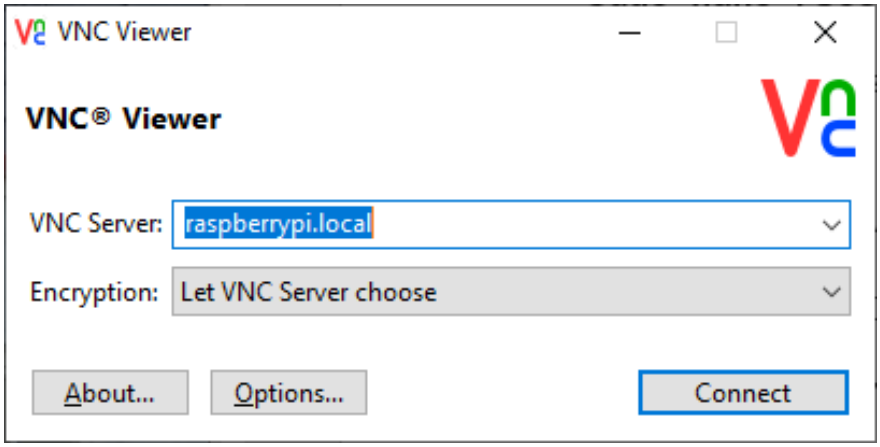
Connecting to the Raspberry Pi Access Point

If you are using an SD card purchased from SPROG DCC, or have followed our instructions to create one, with a Raspberry Pi 2 and WiFi dongle or a Raspberry Pi 3 or 4 with built in WiFi, then the default WiFi password is 'pi-sprog'.

Accessing the Raspberry Pi Desktop

We recommend using remote access software from a host PC or other suitable device. SPROG DCC software images are pre-configured to enable remote access through VNC server.

We recommend VNC viewer, or similar, on your PC.



If the server name is not resolved you will need to determine the IP address of the Raspberry Pi that is assigned by your router then connect to that IP address in VNC viewer.

The default vnc username and password for SPROG DCC software images are 'pi' and 'raspberry', respectively.

For wireless connection you must first connect to the access point using the wireless setup on your PC, just as you would to connect to a wireless router. The default IP address to connect to the SPROG DCC images is 192.168.6.1

Alternatively you can connect the Raspberry Pi to your router with an Ethernet cable and

access it remotely that way. Not that SPROG DCC images do not allow any “tunneling” between the WiFi access point and the wired connection.

If you prefer not to use remote access you can attach an LCD screen or HDMI monitor, along with a USB keyboard and mouse.

SPROG 3 Plus

The following steps are required to install the SPROG 3 Plus on your computer before you can use it for the first time:

- Install FTDI drivers (Windows only)
- Install DecoderPro 4.23.4 or later

Install FTDI drivers (Windows/SPROG 3 Plus)

Install the drivers from the optional USB stick (if purchased with the SPROG 3 Plus) or download from https://ftdichip.com/drivers/vcp-drivers/ for the SPROG 3 Plus on Windows.

Install DecoderPro

DecoderPro should be installed from the optional USB stick (if purchased with the SPROG 3 Plus) or a downloaded copy.

A newer version of DecoderPro than that supplied on the USB stick may be available from the JMRI download page http://www.jmri.org/download

This user guide assumes you are using version 4.23.4, or later.

To install from the USB key, browse to the directory specific to your operating system to find the JMRI installer. For example, if your USB key drive is D: on Windows, double click on the file D:\Windows\JMRI.4.23.4.exe.

Hardware Setup

Connect the Power Supply

The supplied power supply has a standard 2.1mm DC barrel connector (centre positive), that plugs directly in to the SPROG 3 Plus

Connect the Programming Track or Layout

Layout power districts may be connected to both MAIN and PROG track outputs. PROG is, optionally, the auto-reverse power district.

If using a service mode programming track with SPROG 3 Plus or Pi-SPROG 3 Plus, this must be connected to the PROG output.

! Care should be taken when using the service mode programmer that the programming track is isolated from the layout.

The MAIN output of the Pi-SPROG 3 v2 can function as either a service mode programming track output or main layout track output.

! Care should be taken when using the Pi-SPROG 3 v2 as a service mode programmer that the programming track is isolated, or all locos are removed from the layout, apart from the one to be programmed.

For all but the smallest layouts you should use a substantial power bus around the layout with short, thinner “droppers” to individual track segments. Avoid relying on rail joiners for conducting power around the layout.

Connect the CBUS (Optional) – Plus Models Only

CBUS requires at least a 3-way connection for proper operation. CAN HI, CAN LO and 0V. Many modules will also require an external 12V DC power supply that may be connected via the CBUS connector, or by alternate means.

The SPROG 3 Plus CBUS interface is internally powered and no 12V connection is required. The 4th pole of the CBUS connector may be safely connected to 12V (e.g., to standardize connector wiring) but is not connected internally, and will not draw any current.

The Pi-SPROG 3 plus CBUS interface is galvanically isolated and 12V power must be applied to the CBUS connector.

The CBUS must be correctly terminated for proper operation. For very small installations a single 60 ohm resistor is often sufficient. For larger installations a 120 ohm resistor should be connected across CAN HI and LO at each end of the bus.

Initial Setup Using Terminal Interface (Optional)

The SPROG DCC Generation 5 Interface

The SPROG 3 Plus is a SPROG Generation 5 products. The interface between the host and the SPROG 3 Plus is CBUS, using the GridConnect protocol The Plus models may also be connected to a CBUS network but this is not required to operate run trains and operate DCC accessories on a layout.

The SPROG 3 Plus also supports an additional command set that may be access through a terminal emulator program such as PuTTY https://www.chiark.greenend.org.uk/~sgtatham/putty/ This can be used for module setup and CBUS testing.

The SPROG 3 Plus includes a bootloader that allows firmware updates to be installed by the user.

JMRI must not be connected to the SPROG 3 Plus when using the terminal interface.

Use a terminal emulator program set to 460800 baud, 8 bits, no parity, 1 stop bit to connect to the SPROG.

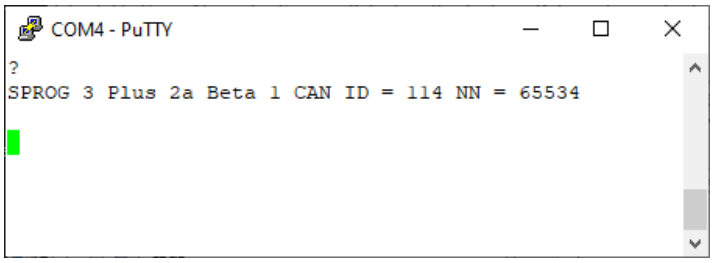
Commands may be typed in upper or lower case. All numeric entry is assumed to be decimal. All commands should be followed by a carriage return.

The SPROG 3 Plus does not echo characters back to the terminal. Local echo can be enabled in the PuTTY Terminal options to show what you have typed.

The SPROG 3 Plus does not allow backspace to be used.

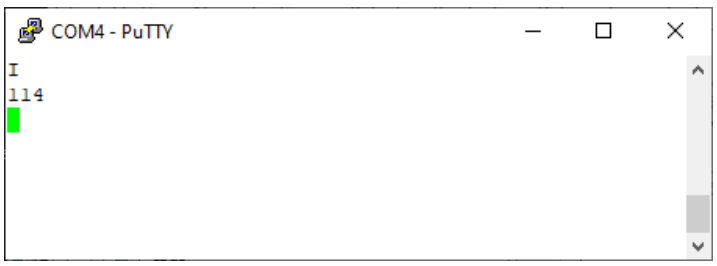
Checking the Firmware Version

Use the ? command to check the firmware version. This will also display the current CAN ID and CBUS Node Number

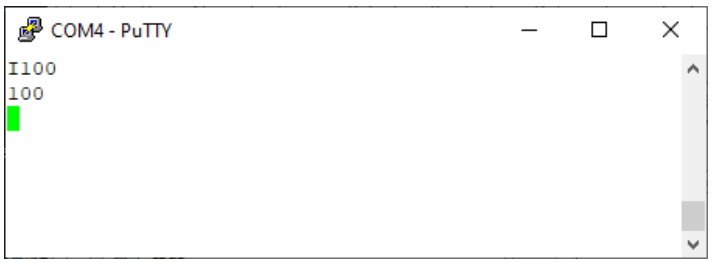


Setting the CAN ID

The I command will show the current CAN ID.



A new ID may be set by following the I command with a number in the range 1 – 127. The new ID will be displayed. The old ID will be displayed if the new value is invalid.



It is recommended that modules such as the SPROG 3 Plus, with a fixed CAN ID, have a CAN ID in the range 100 – 127.

Any CBUS traffic from JMRI will use the CAN ID assigned in the JMRI connection preferences.

CBUS traffic generated from the SPROG 3 Plus itself will use the SPROG 3's own CAN ID which is 114 in a new module.

All modules must have a unique CAN ID.

Setting the Node Number

The N command will show the current Node Number.

A new NN may be set by following the I command with a number in the range 1 – 65535. The new NN will be displayed. The old NN will be displayed if the new value is invalid.

The SPROG 3 Plus Node number is 65534 in a new module.

It is recommended, but not required for more advanced users, that all modules have a unique Node Number when using CBUS.

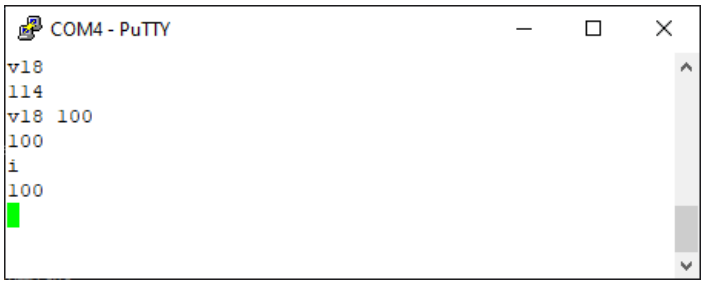
It is recommended that modules such as the SPROG 3 Plus, with a fixed NN, have an NN in the range 65520 - 65535.

Changing Node Variables (NVs)

The V command will show the current value of an NV.

A new value may be written to an NV by giving the value in the range 1 – 255. The new NV value will be displayed. The old NV value will be displayed if the new value is invalid.

The example shows setting the CAN ID through NV18.



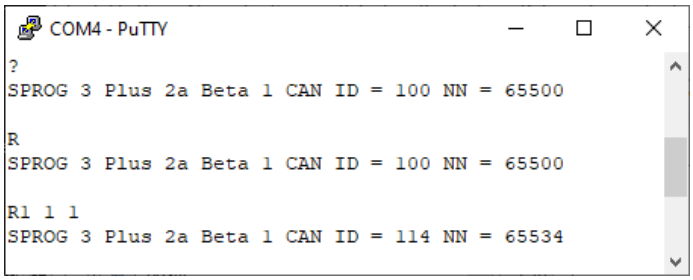
Take care to understand what you are doing when changing NVs.

See **CBUS Node Variables** for the NVs supported by the Pi-SPROG 3 Plus

Reset SPROG 3 Plus to Factory Defaults

The R command will reset the SPROG 3 Plus to the factory default, notably the CAN ID and Node Number. The output is similar to the ? command, showing the firmware version, ID and NN.

To prevent inadvertent resets, the R command must be followed by 3 parameters of any value between 0 and 65535. In the following example, R with no parameters has no effect.



The R command DOES NOT revert to the original firmware version, if updates have been applied.

Getting Started with JMRI (DecoderPro and PanelPro)

DecoderPro and PanelPro are different interfaces to the same underlying JMRI software. The split exists for historical reasons only. There are some differences in the functions that are available from the menus but a lot of features are available through both interfaces.

It has become customary to think of DecoderPro as the tool for programming decoders and PanelPro for controlling a layout.

DecoderPro will often be used with a dedicated programming track for 'service mode' programming. This allows full read and write access to all Configuration Variables (CVs) in a ***single*** decoder.

PanelPro will often be used with 'on the main' or 'ops mode' programming in conjunction with layout control. In this mode CVs may be written, but (without special hardware such as Railcom) values cannot be read back. The SPROG 3 Plus does not support reading from decoder in ops mode. The advantage of ops mode programming is that any loco, out of all the locos on a layout, may be programmed. Unlike service mode there is no single loco limit.

It is possible to access all the features of JMRI (Roster, Panels, Tables, Programmer, etc, …) through either interface, but the menu structure is different.

With the SPROG 3 Plus and the Pi-SPROG 3 Plus, use the **Mode Switch** tool to select either 'Programming track off when not programming' (recommended for new users) or 'Programming track follows main when not programming' to use a service mode programming track.

! Care should be taken when using the service mode programmer that the programming track is isolated from the layout.

With the Pi-SPROG 3 v2, use the **Mode Switch** tool to select either 'Programming (Service Mode Programming)' or 'Command Station (Ops mode programming)' to use a service mode programming track.

! Care should be taken when using the Pi-SPROG 3 v2 as a service mode programmer that the programming track is isolated, or all locos are removed from the layout, apart from the one to be programmed.

Setting the JMRI Connection Preferences

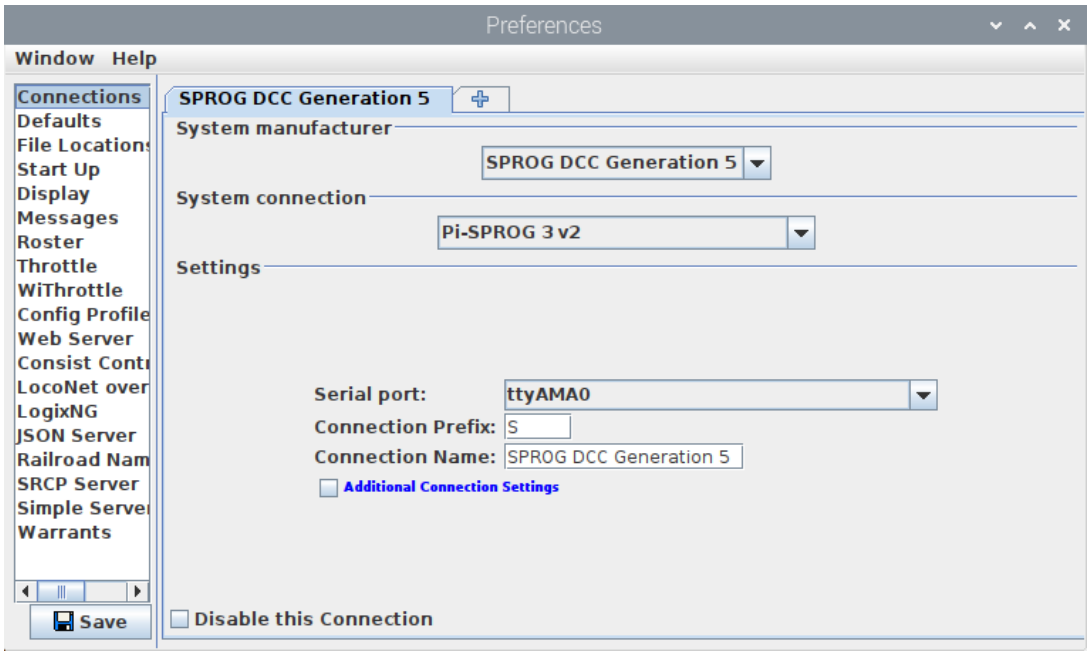
For the Pi-SPROG 3 Plus and Pi-SPROG 3 v2, if you downloaded or purchased an SD card image then this step can be skipped.

If setting up your own system then please be sure to select SPROG DCC Generation 5 (**NOT** simply SPROG DCC) as the System Manufacturer and the appropriate System Connection in the connection preferences for DecoderPro or PanelPro.

On Windows systems, the COM port required for the Serial port entry may be determined by looking for the SPROG 3 Plus in the Windows Device Manager under "Ports COM & LPT". The SPROG 3 Plus will appear as a USB Serial port.

On Linux the SPROG 3 Plus USB interface should be created as ttyUSBx when x will depend on what other hardware is connected to the system.

For the Pi-SPROG 3 Plus, Pi-SPROG 3 v2, if you have followed our recommended image creation, select the ttyAMA0 serial port.



Save the new settings and restart JMRI or PanelPro.

Further details of the general features of JMRI are beyond the scope of this document.

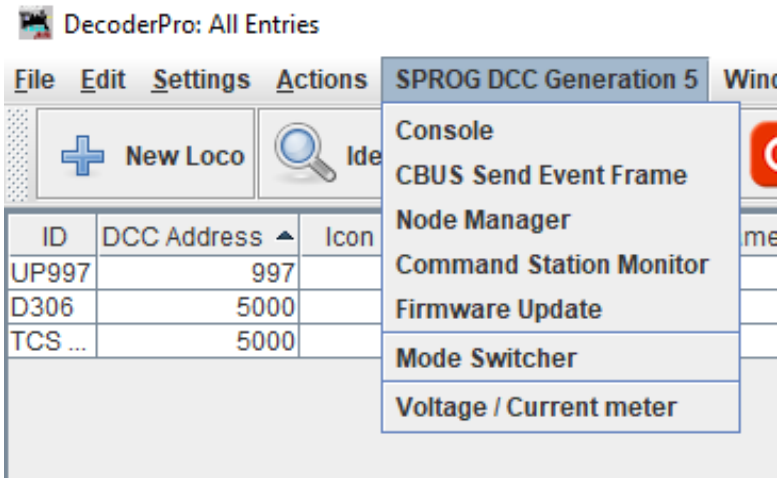
Please consult the on-line help in the software or on the JMRI help webpages.

SPROG specific features are described next.

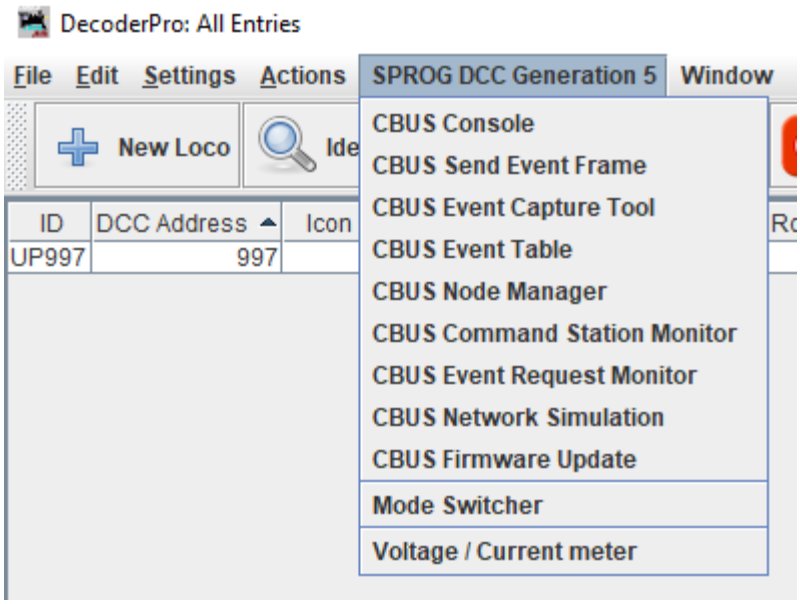
SPROG DCC Generation 5 JMRI Tools

A number of tools are available on the SPROG DCC Generation 5 menu.

Pi-SPROG 3 v2 JMRI Tools



Pi-SPROG 3 Plus and SPROG 3 Plus JMRI Tools

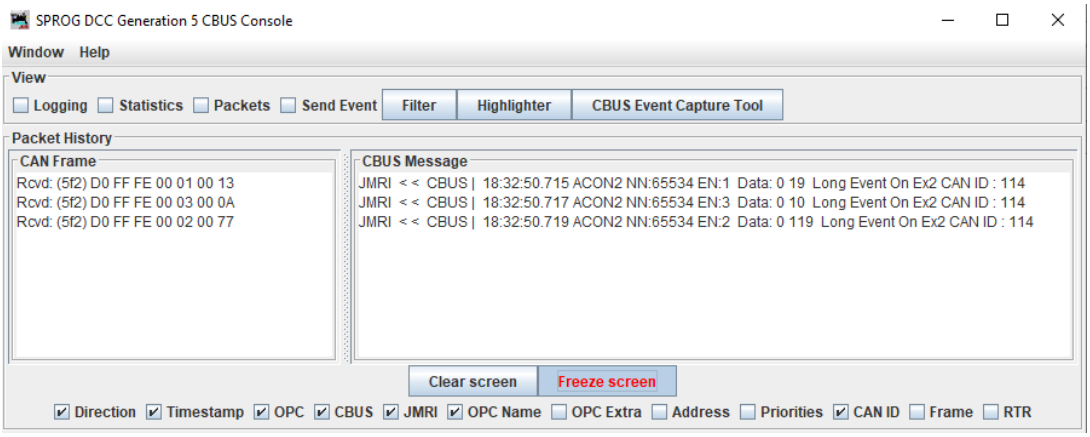


A selection of these are descibed in more detail below. The Firmwware Update Tool is described in a later section.

Console

The console shows the CBUS frames sent to/from SPROG 3 Plus and the host. It is useful for capturing diagnostic information if a problem occurs that is repeatable.

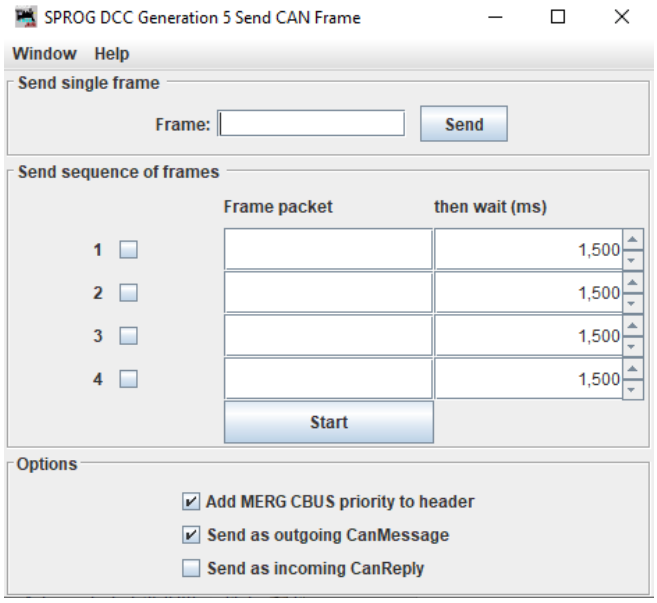
Various display options re selectable and the output may be logged to a file.



CBUS Send Event Frame

The send event frame tool allows CBUS events to be sent manually, e.g., for testing or setting up.

On the Pi-SPROG 3 v2 (that has no CBUS interface) events will always be translated into DCC accessory commands using the Event Number as the DCC accessory address.

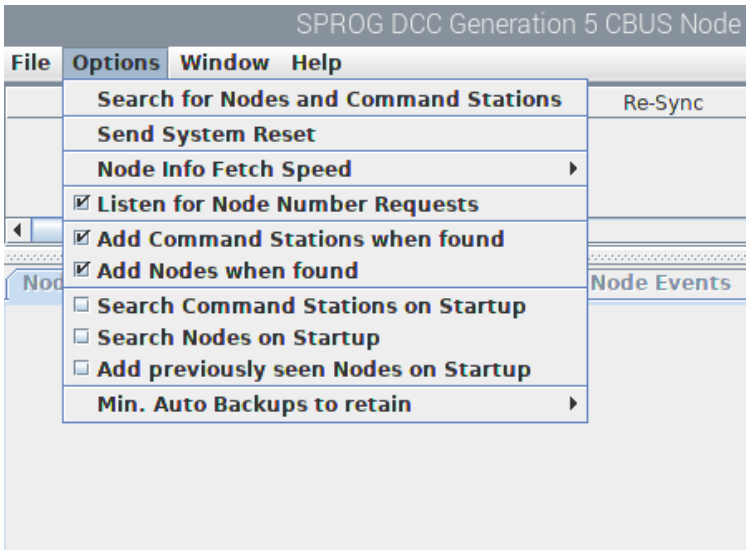


Node Manager

The node Manager allows access to internal settings in the SPROG 3 Plus. It is not required for normal, everyday, operation, unless you need to change some aspect of the SPROG 3 Plus operation (Older SPROG users may think of this as being similar to setting the SPROG mode word).

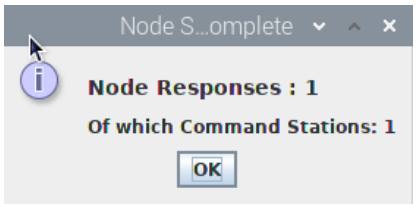
The node manager is started from the SPROG DCC Generation 5 > Node Manager menu item.

Select the option ‘Add Command Stations when found’ and ‘Add Nodes when found’ in the Options menu.



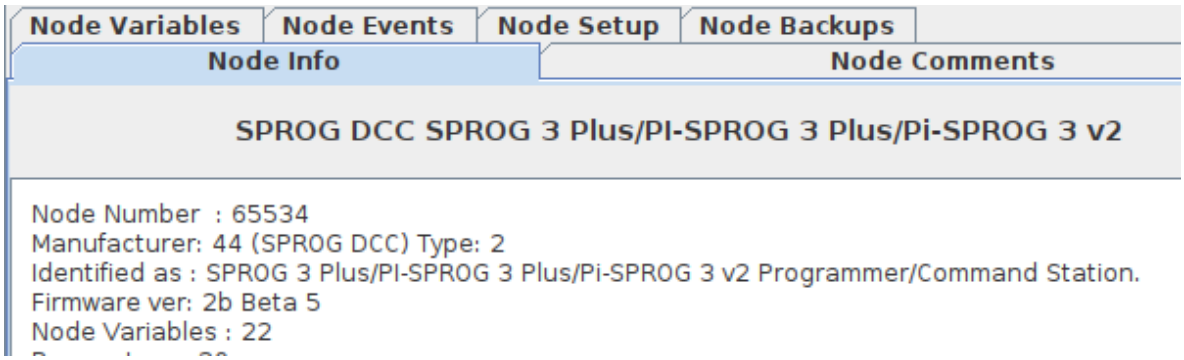
Click ‘Search for Nodes and Command Stations’ in the Options menu

The SPROG 3 Plus should respond as a Command Station

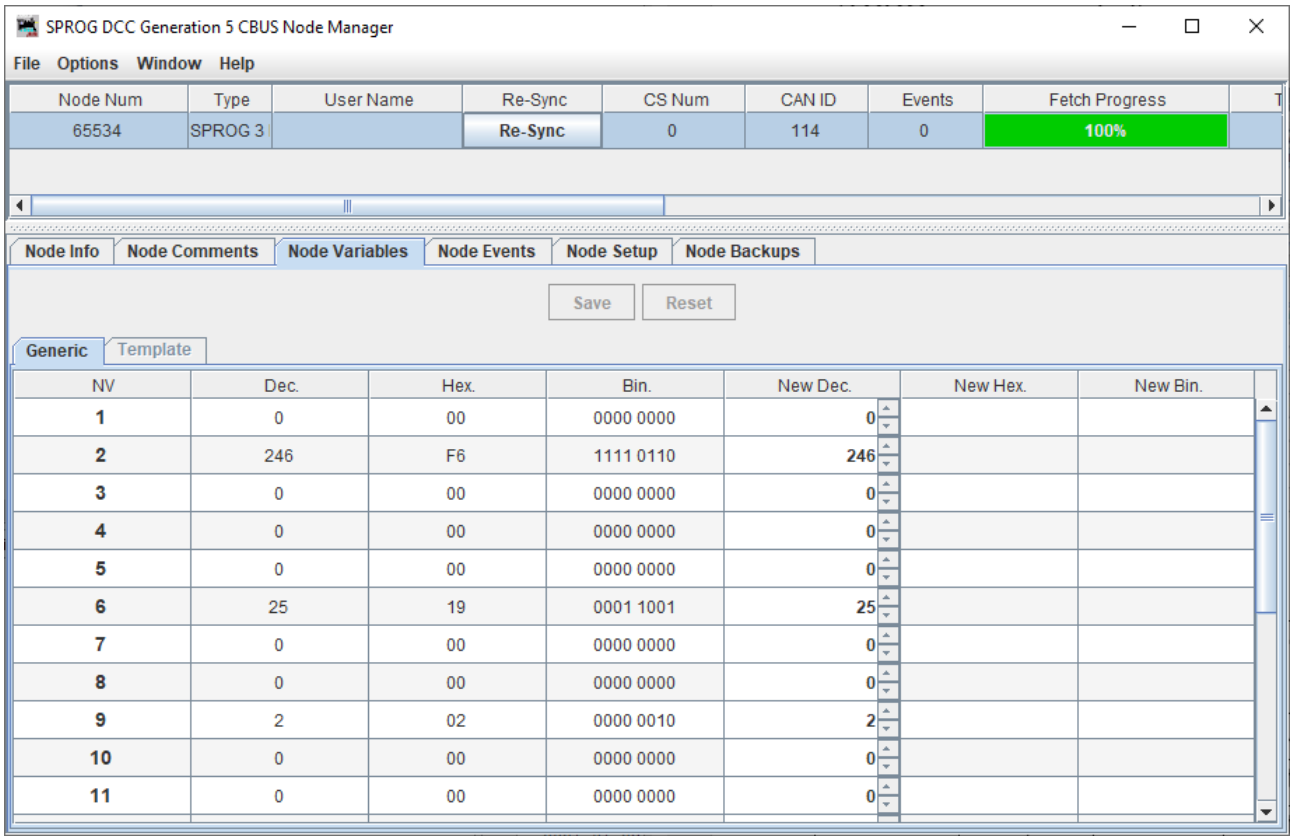


Click OK.

In the Node Info tab you can see details such as the firmware version of the SPROG 3 Plus



The Node Variables (NVs) control the operation and show the internal status of the SPROG 3 Plus, much like the CVs in a DCC decoder.



To change an NV use the spinner to select the new value, or type the new value directly, click the save button and then confirm the operation in the pop up dialog.

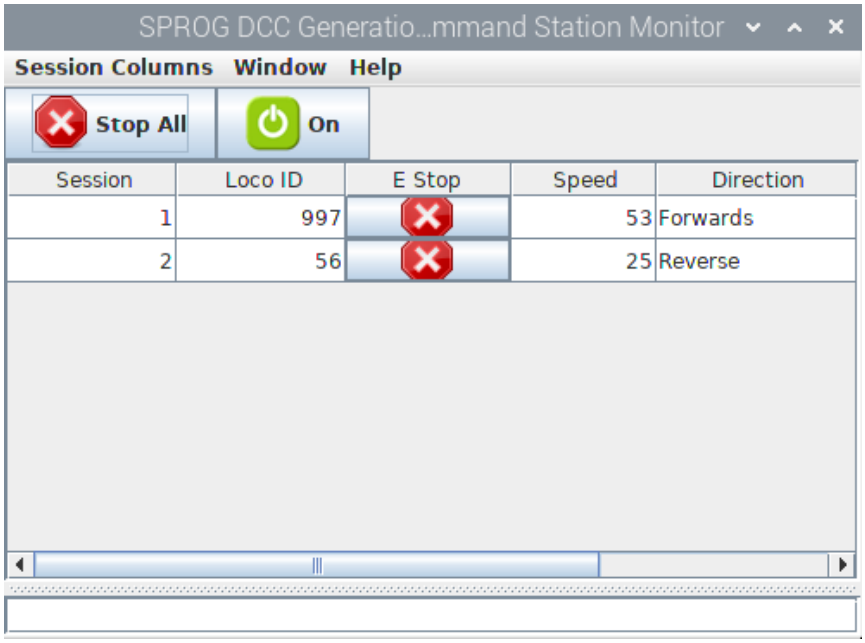
Take care to understand what you are doing when changing NVs.

Not all NVs are supported on all hardware variants. See **CBUS Node Variables**

Command Station Monitor

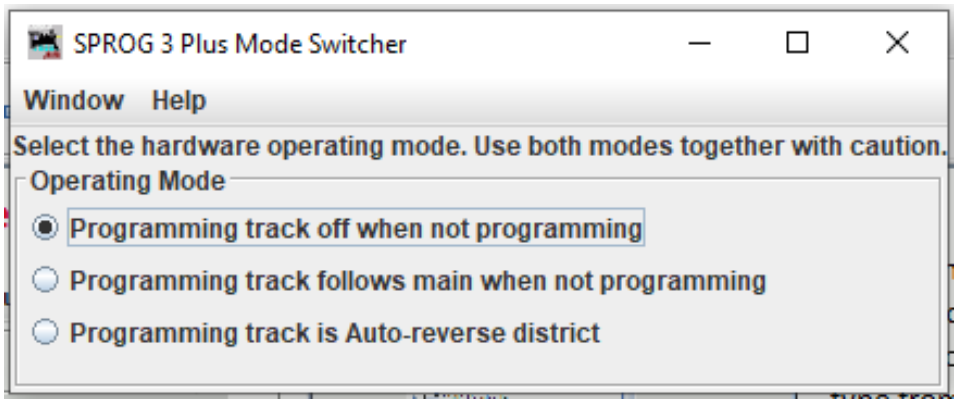
When using the SPROG 3 Plus as a command station controlling multiple locos, you can see the status of all locos in the command station monitor.

Emergency stop and power control buttons are also provided in the command station monitor window.



Mode Switching with SPROG 3 Plus and Pi-SPROG 3 Plus

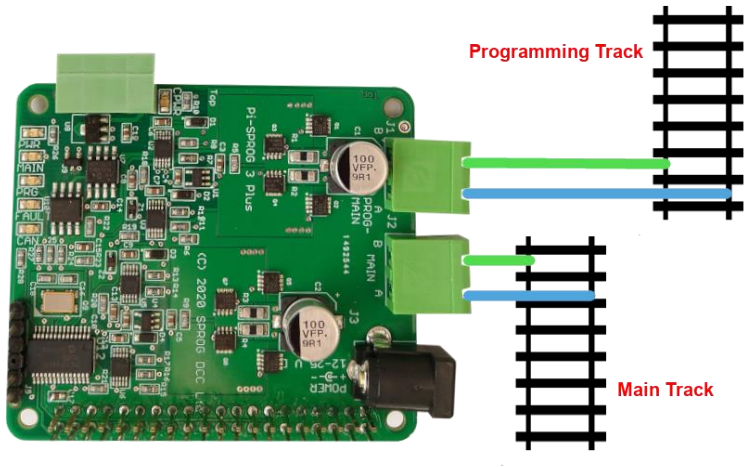
The two track outputs of the [Pi-]SPROG 3 Plus are nominally assigned to the main layout and the programming track, but may be assigned to different functions using the mode switcher tool to control the operation of the track outputs. Three operating modes are available.



Programming Track Off When Not Programming

In this mode the programming track output supports service mode programming and is turned off when no programming operation is taking place. This mode would typically be used where the programming track is close to, but not connected to, the main layout. Locos to be programmed would be lifted from the layout and placed on the programming track. It is not possible to run locos from the main track onto the programming track.

The programming track must be electrically isolated from the main layout and connected only to the programming track output. The loco must be entirely on the programming track with no wheels or bogies (trucks) bridging the gap when programming.



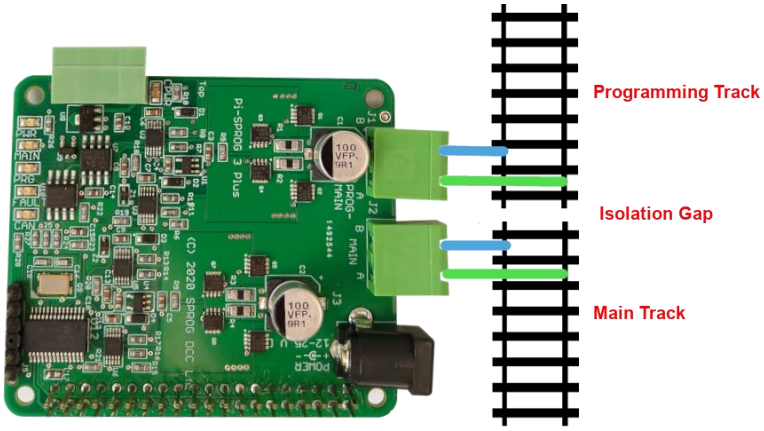
It is not possible to test run a loco on the programming track.

Layout operation may continue whilst programming.

Programming Track Follows Main When Not Programming

In this mode the programming track output supports service mode programming and is linked internally to the main track output. When no programming is taking place, the programming track output will follow the main track output, e.g., to allow locos to be driven from the main layout to a spur or siding being used as a programming track.

The programming track must be electrically isolated from the main layout and connected only to the programming track output. The loco must be entirely on the programming track with no wheels or bogies (trucks) bridging the gap when programming.



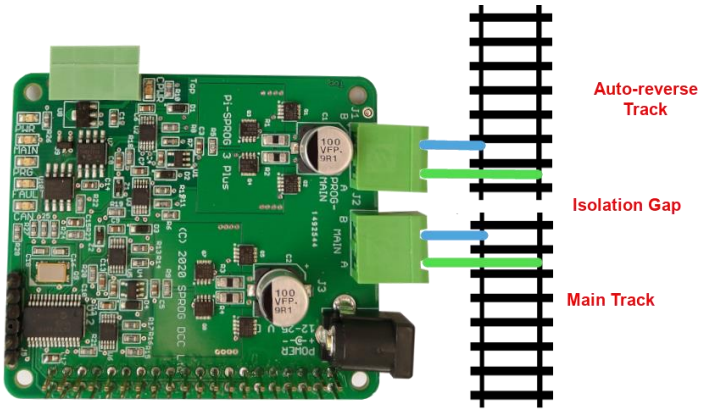
If the programming track is long enough then the loco can be test run on the programming track by selecting it in a throttle.

Layout operation may continue whilst programming.

Programming Track is Auto-reverse District

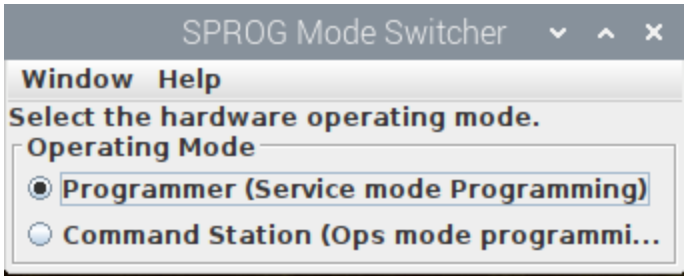
In auto reverse mode the programming track output does not support service mode programming. Instead, it can be used as a second layout power district with auto-reverse.

The two power districts must be electrically isolated from each other.



Mode Switching with Pi-SPROG 3 v2

With the Pi-SPROG 3 v2, only two operating modes are available, programmer and command station. Unlike earlier SPROG II and sprog 3, there is no need to close and restart JMRI with a different connection preference to change the operating mode.



Always close and reopen any programmer windows when switching modes to ensure they reflect the programming modes.

Programmer (Service mode programming)

In this mode the Pi-SPROG 3v2 supports service mode programming on a programming track, with full read and write access to the decoder. A single throttle can be used for test running a loco (if the programming track allows it, or a rolling road or running stand is used) after making changes to the decoder settings.

Care must be taken not to use this mode when connected to a layout as all locos on the layout will be reprogrammed.

Command Station (Ops mode programming)

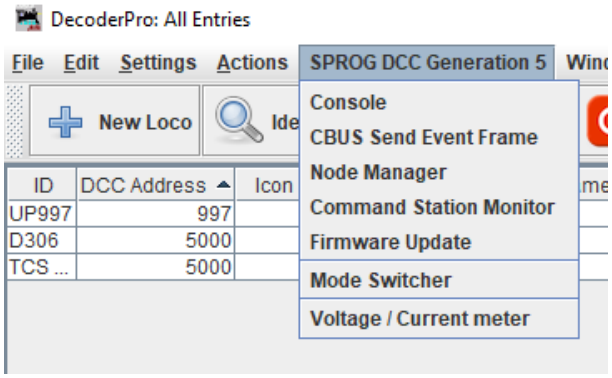
In command station mode the Pi-SPROG 3v2 supports Operations mode, or on the main, programming and multiple throttles for layout control. New values may be written to any decoder, but settings cannot be read from decoders.

Voltage/Current Meter

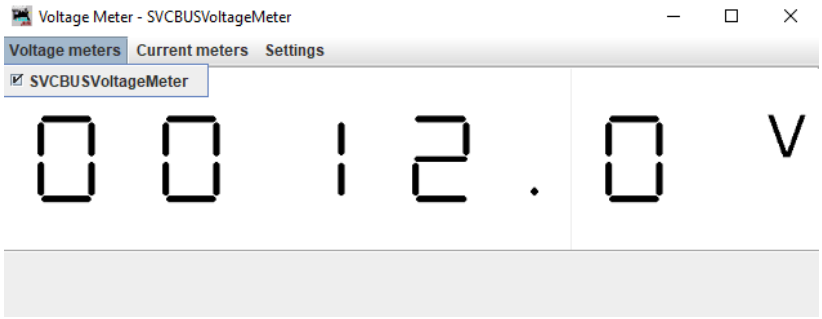
If Node Variable 10, Multimeter Mode, is set to 1 then the SPROG 3 Plus will send regular

voltage and current measurements (approx. every 2s) to the host and to CBUS. These may be displayed with JMRI Voltage/Current meters.

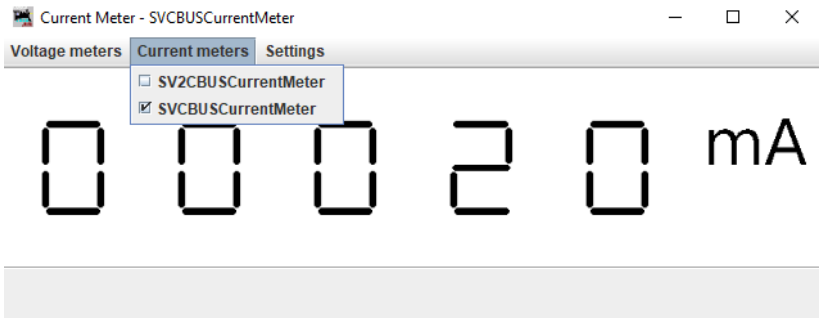
Open a meter from the SPROG DCC Generation 5 menu



Set the meter to be a voltage or current meter



The voltage meter monitors the input supply voltage. The DCC track voltage will be slight less than this.



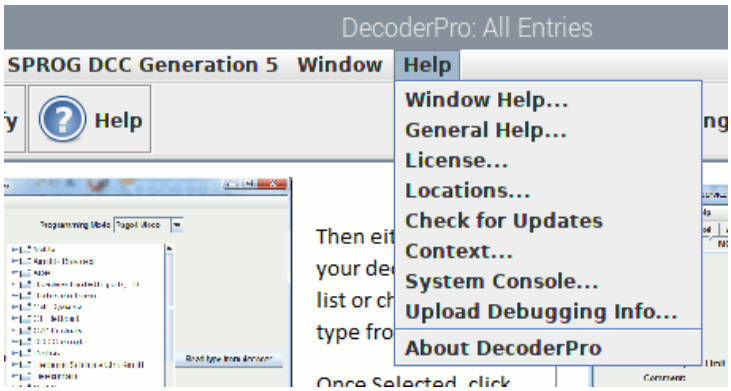
The SVCBUSCurrentMeter displays the main track current.

The SV2CBUSCurrentMeter displays the programming track current.

The voltage and current measurements are for information only and accurate to +/-5%. The current readings, especially, may fluctuate to a marked extent.

The JMRI System console

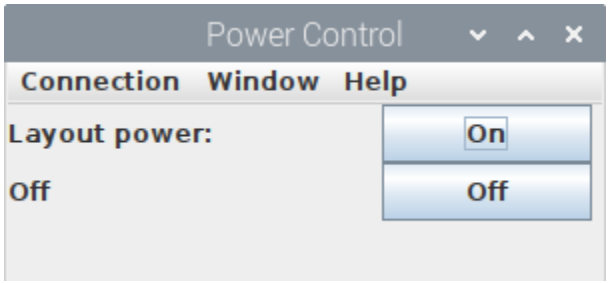
The JMRI system console can be opened from the Help > System console menu item.



The system console contents may be useful when something goes wrong. Look for ERROR or WARNING messages.

The JMRI Power Manager

The state of the track power can be controlled through the Power Control.



Or by clicking the icon that appears in various windows, e.g., throttles

During programming the prog track power will be controlled automatically but will not be reflected in the power manager.

By default, the power is always off at startup. This behaviour can be changed by setting the User Flags Auto power bit. The hardware will broadcast the power state during startup.

For all Pi-SPROGs, the initial power state will show as Unknown in JMRI as the hardware starts up before the software and the broadcast message is not seen.

For the SPROG 3 Plus the initial power state displayed by JMRI will depend whether the hardware or software starts up first. If JMRI is started before power is connected to the SPROG 3 Plus then the power state will reflect the Auto Power setting, once the hardware startup is complete.

A Brief Introduction to CBUS

The Pi-SPROG 3 v2 uses the CBUS protocol for communicating with the host only. It does not have an external CBUS interface.

Only a brief description of CBUS can be given here.

CBUS was developed by members of the Model Electronics Railway Group (MERG) and the protocol and other documents were hosted on the MERG website. ‘Ownership’ oe ‘control’ of CBUS vests in those individual, not in MERG.

Please refer to documentation on the MERG website for more in-depth information on CBUS.

CBUS is a Layout Control Bus (LCB), much like, e.g., Loconet® (trademark of Digitrax Inc.) or XpessNet, a communications network that allows electronic modules on a model railway to be connected, often to some form of personal computer (but not required). The modules that can be connected to CBUS include, but are not limited to, command stations (such as the SPROG 3 plus family), throttles, control panels, turnout and signal control, block detectors.

Using an LCB such as CBUS can save a lot of wiring on the layout by using just a few wires to connect all of the various elements together. For example, many switches on a control panel can control turnouts and signals in diverse locations on the layout with just a few wires.

CBUS, as used in the SPROG 3 plus family, is based on the Controller Area Network (CAN) bus, widely used, for many years, in the automotive and transport industries. Using CAN allows the choice of a wide range of well understood and cost-effective components to implement the interface to the connected modules.

As well as layout control messages, CBUS allows modules to be configured, e.g., to control how outputs are driven or how inputs are sensed. Each module is built with stored parameters that can be interrogated to determine its functionality. Operation is controlled by writing to Node Variables (NVs). This is comparable to reading and writing CVs in a DCC decoder, but uses a different process.

Whilst CBUS can be, and often is, used with a personal computer, it does not require a single “master” module. All modules co-exist on the network and can talk to each other at any time. Compare this to some other LCBs where each module may only send data when it is “polled” or asked for it. All SPROG 3 Plus family modules include a host interface and do not require a separate interface module.

A CBUS module typically forms a CBUS Node but it is possible that a single module could implement multiple CBUS nodes with similar or differing functions.

All CBUS messages (or events) include an Op-Code (OPC) which determines the nature of the message. For some OPCs, the sender will include its own Node Number (NN) in the message. Events will include an Event Number (EN). Layout control can thus be based on what is to happen (OPC) who sent the message (NN) and where it must happen (EV). CBUS modules are ‘taught’ to send certain events to CBUS or recognize and respond to certain EVs from certain NNs from CBUS. How the module generates the event or what it does with a received event is very much module dependent. A simple example would be a control panel switch connected to a module that sends event X when the switch is closed.

Another module is taught to do something (e.g., change the direction of a cross-over) when it sees that event.

CBUS can be used, indeed was originally intended to be used, in a strict ‘producer-consumer’ mode in which each event has a single producer that sends, or produces, an event and one or many consumers that listen for, or consume, that event. In this case the NN is an integral part of the event and consumers will also be taught the NN to be recognized when consuming an event. These events, including the NN and EN are known as ‘long events’.

To add flexibility CBUS can also operate in a ‘many-to-many’ mode in which multiple producers can produce the same event. These are known as ‘short events’. The NN is still included in the message (for diagnostic purposes) but is ignored by the consumer.

CBUS consumer modules may have Event Variables (EVs) associated with each event that they are taught. The meaning of any EV is module dependent and beyond the scope of this document. Typical useof an EV would be to modify the result of an event by, e.g., inverting the polarity of an output or setting a time for an output to be turned on or off.

CBUS modules may have Node Variables (NVs) that control the operation of the module as a whole, such as controlling initial behaviour at startup.

Operation with JMRI and CBUS Cabs (Throttles)

The Pi-SPROG 3 v2 support running trains with JMRI throttles, including WiThrottle connected devices. It does not directly support physical cabs as it has no CBUS interface.

The Pi-SPROG 3 Plus and SPROG 3 plus support running trains with JMRI throttles, including WiThrottle connected devices and physical throttles connected to CBUS.

At the time of writing, the only physical throttle that has been tested is the MERG CANCAB. This is available as a kit to MERG members. See MERG documentation for full details of how to use the CANCAB.

Release/Dispatch

When you release a loco whilst it is moving, it is considered to be “dispatched”, but the loco is not forgotten. The loco is driven at the current speed until it is reselected, by the same, or another, cab.

Steal/Share

It is possible for one cab to steal a loco from another cab, e.g., when handing a train over to another operator. Users will be notified by messages on the cab screen or computer sreen if using a JMRI throttle.

Similarly, two or more (up to 3) cabs may share a loco.

Connecting Throttle Devices to the Raspberry Pi Access Point

If you are using an SD card purchased from SPROG DCC, or have followed our instructions to create one and are using a Pi-SPROG 3v2 or Pi-SPROG 3 plus on a Raspberry Pi with built in WiFi, then you can connect a throttle with an app such as WiThrottle or EngineDriver.

The default WiFi password is ‘pi-sprog’, unless you changed it during or after creating the SD card image.

CBUS Events and DCC Accessory Control

If a layout is already equipped with DCC accessory decoders then there is no need to replace these with CBUS modules to control points, signals, etc.

CBUS events may be used to generate DCC accessory commands that will be sent on the track output(s).

! Note that accessory control from a CANCAB uses short events.

Turnout Control – in Brief

Turnout numbers assigned in JMRI are sent as CBUS events when a turnout is activated with either an ON or OFF event.

CBUS turnout numbers (and hence Event numbers) start at 1.

By default, the turnout number (event number) is decremented by 1 before using it as the DCC accessory address for turnout control.

The following sections explain this in more detail for each SPROG 3 variant.

Pi-SPROG 3 v2

The CBUS Node Number will be ignored and a short or long event will be mapped from the event number to a DCC accessory address by subtracting 1 from the event number. CBUS on events (ACON, ASON) will turn the DCC accessory on. CBUS off events (ACOF, ASOF) will turn the DCC accessory off.

SPROG 3 Plus and Pi-SPROG 3 Plus

At the time of writing, a very simple scheme exists to map CBUS events to DCC accessory commands.

It is intended that a more sophisticated method will be implemented in a future firmware release.

User flags bit 4, Map events (see screenshot below), must be set to map CBUS events to DCC accessory commands, otherwise CBUS control of DCC accessories is outside the scope of the SPROG 3 Plus and Pi-SPROG 3 Plus.

Short Events

The CBUS Node Number will be ignored and a short or long event will be mapped from the Event Number to a DCC accessory address as described below. CBUS ON events (ASON) will turn the DCC accessory on. CBUS OFF events (ASOF) will turn the DCC accessory off. This is similar to the behaviour of the Pi-SPROG 3 v2.

Long Events

Mapping between CBUS events and DCC accessory addresses is controlled by Node Variables. See **NV11, NV12 Node Number to Map to DCC**

When NV11 and NV12 are both zero, the CBUS Node Number will be ignored and a long event will be mapped from the Event Number to a DCC accessory address as described below. CBUS ON events (ACON) will turn the DCC accessory on. CBUS OFF events (ACOF) will turn the DCC accessory off. This is similar to the behaviour for short events and the Pi-SPROG 3 v2.

If the NV11, NV12 pairing is non-zero then the value formed with NV11 being the high byte and NV12 the low byte is used as a Node Number to match against the Node Number of a CBUS long event. If the Node Number in the event matches the Node Number in NV11/12 then the long event will be mapped from the Event Number to a DCC accessory address as described below. This allows accessory control to be restricted to a single node, e.g., a control panel. CBUS ON events (ACON) will turn the DCC accessory on. CBUS OFF events (ACOF) will turn the DCC accessory off.

Mapping from CBUS Event Numbers to DCC Accessory Address

Consideration must be given to the way in which various manufacturers have chosen to interpret DCC accessory addresses. The mapped CBUS event may not correspond directly to a DCC accessory address recognised by a DCC accessory decoder that has been programmed on a different DCC system.

Firmware version prior to 2e Beta 2

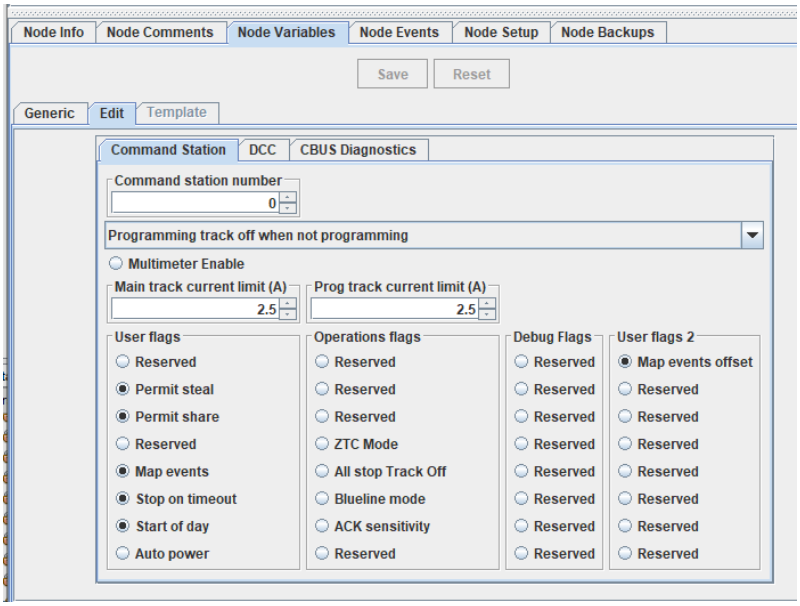
JMRI turnout numbers for CBUS, and hence the associated CBUS events, start from 1 whereas DCC accessory addresses start from 0.

CBUS events are mapped to DCC accessory addresses by subtracting one from the Event Number.

E.g., ACON event with EN=1234 will send DCC on command to accessory address 1233.

Firmware version 2e Beta 2 and later

A new flags byte User Flags 2 is introduced. One bit of this, bit 0 Map events offset, is used to control the mapping of CBUS events to DCC accessory commands. The default is for this bit to be set and the offset to be applied as for prior firmware versions. This is shown in the screenshot:



Clearing the Map events offset bit will remove the offset so that CBUS events are mapped directly to DCC Accessory addresses.

E.g., ACON event with EN=123 will send DCC on command to accessory address 123.

DCC accessory address 0 is not available in this mode.

Detailed Explanation of DCC Accessory Addressing

The SPROG 3 Plus family generate DCC accessory decoder packets as described in section 2.4 of the NMRA's DCC standard S-9.2.1. The CBUS event is mapped to the bits in the accessory packet

10AAAAAA 1AAACDDD

The address bits "A" are generated from the CBUS event by dividing by 4. Bits 4-6 of byte 2 are inverted, as required by S-9.2.1.

Bit 3 of the second byte "C" is always set to activate the accessory.

Bits 1-2 of the second byte "DD" are generated from the remainder when the CBUS event is divided by 2. These select which if four pairs of outputs are being controlled.

Bit 0 of the second byte "D" is set for CBUS ON events, clear for CBUS OFF events. This bit selects which output of a pair is activated.

CBUS Node Variables

NVs control the operation and show the internal status of the SPROG 3 Plus, much like the CVs in a DCC decoder.

NVs may be changed through the terminal interface or through the JMRI node manager.

At the time of writing, the SPROG 3 Plus is not supported by the FCU (a configuration tool written and supported by MERG members).

A sub-set of the NVs are available on the Pi-SPROG 3 v2. These are indicated by an X in the v2? column in the following table.

Node Variable	Values	Default	v2?	Function
1	0	0	X	Command station number
2	0 – 255	118	X	User flags
3	0 – 255	0	X	Operation flags
4	0	0	X	Debug flags – not currently used
5	0,1,2	0	X	Programming track power mode
6	0 – 255	250	X	Programming track current limit, Amps x 100
7	0 – 255	-		Read only input voltage, Volts x 10, e.g. 118 represents 11.8 V
8	0 - 255	-		Read only main track current, Amps x 100
9	1 – 7	1		Accessory packet repeat count.
10	0,1	0		Multimeter mode. Set to 1 to enable voltage and current events.
11	0 – 255	0		NN to map to DCC hi byte
12	0 – 255	0		NN to map to DCC lo byte
13	0 – 255	250		Main track current limit, Amps x 100
14	0 – 255	-	X	Read only programming track current, Amps x 100
15	0 – 255	-		Read only main track current high-water mark, in Amps x 100
16	0 – 255	-	X	Read only programming track current high-water mark, Amps x 100
17	0,1	0		Setup mode – do not use
18	1 – 127	114		CAN ID
19	0 – 255	255	X	Node Number high byte
20	0 – 255	254	X	Node Number low byte
21	16 - 255	16	X	Number of DCC preamble bits transmitted
22	0, 1	0		CAN disable

NV1 Command Station Number

Only command station 0 is currently supported. Any other values will be ignored.

NV2 User Flags

The user flags NV contains 8 bits.

The default value is 01110110 or hex 76 or decimal 118.

Reserved bits should always be set to 0 and will read as zero.

Bit	Default	Function
0	0	Reserved
1	1	Permit Steal: Set to enable throttle steal option
2	1	Permit Share: Set to permit throttle share option
3	0	Reserved

Bit	Default	Function
4	1	Map Events: Set to map CBUS events to DCC accessory packets
5	1	Stop on Timeout: If set and a loco session times out, the train is brought to a stop. If clear the train is dispatched whilst moving
6	1	Start of Day: Set to send CBUS even 0 on startup which may be used as a start of day event
7	0	Auto Power: Set to turn track power on at startup

NV3 Operation Flags

The operation flags NV contains 8 bits.

The default value is 0.

Reserved bits should always be set to 0 and will read as zero.

Bit	Default	Function
0	0	Reserved
1	0	Reserved
2	0	Reserved
3	0	ZTC Mode: Set to modify bit timing when programming for ZTC decoders
4	0	All stop track off: Set to turn track power off if an all stop command is issued
5	0	Blueline Mode: Modify programming operation to suit blueline decoders
6	0	ACK sensitivity: Set to modify programming ACK pulse detection for certain large scale Zimo decoders
7	0	Reserved

NV4 Debug Flags

Debug flags NV is currently reserved. All bits will read as zero.

NV5 Programming Track Power Mode

The way the programming track works is controlled by Node Variable 5 and can have one of three values

SPROG 3 Plus and Pi-SPROG 3 Plus

NV5	Mode
0	Programming Track Off When Not Programming Programming track is independent of main track and is off when not programming
1	Programming Track Follows Main When Not Programming Programming track follows main track when not programming
2	Programming Track is Auto-reverse District Programming track follows main track Auto reverse on overload

Pi-SPROG 3 v2

NV5	Mode
0	Programmer Mode Service mode programming packets will be generated for use with a programming track
1	Command Station Mode Operations mode programming packets will be generated for “ops-mode” or “on the main” programming

NV6 Programming Track Current Limit, and

NV13 Main Track Current Limit

Current trip limits to be applied to the two track outputs in Amps x 100, e.g., a value of 200 will apply a current limit of 2.0A.

NV7 Input Voltage

Measured input supply voltage in Volts x 10, e.g., 125 represents 12.5V.

NV8 Main Track Current, and

NV14 Programming Track Current

Current measurement in Amps x 100, e.g., 70 represents 0.7A or 700mA.

NV9 DCC Accessory Packet Repeat Count

Range 1 – 7.

The number of times a DCC accessory packet is repeated. If DCC accessories do not operate reliably, try increasing the repeat count by 1.

NV10 Multimeter Mode

0 – Multimeter events disabled

1 – Multimeter current/voltage measurement events enabled

NV11, NV12 Node Number to Map to DCC

Set to zero to map all short events directly to DCC accessory commands where the event number becomes the DCC accessory number.

Set a non-zero value to match a specific node number and map all long events from that node to DCC accessory commands. NV11 is the high byte of the Node Number.

NV15 Main Track Current High-Water Mark, and

NV16 Programming Track Current High-Water Mark

The current high-water marks store the highest measured current as Amps x 100 (e.g., a value of 130 represents 1.3 Amps).

The high-water marks can be reset by writing 0 to the appropriate NV.

NV17 Setup Mode

Use with caution. Set bit 1 of NV17 to put the SPROG 3 Plus in setup mode.

NV18 CAN ID

It is recommended that modules such as the SPROG 3 Plus, with a fixed CAN ID, have a CAN ID in the range 100 – 127.

NV19, NV20 Node Number

Range: 0 – 255.

High and low byte, respectively of the CBUS node number. Overall range is 0 – 65535.

NV21 DCC Preamble Bits

The number of preamble bits sent before each DCC packet.

NV22 CAN Disable

0 – CAN bus is enabled (normal operation).

1 – CAN bus is disabled (for test purposes).

Voltage and current measurements are for information only and accurate to +/-5%.

CBUS Opcodes

OPCs Consumed

CBUS Opcode (OPC)	Action	Comment
ACOF	Send DCC accessory packet	
ACON	Send DCC accessory packet	
ARST	System reset	Software reset – not on Pi-SPROG 3 v2
ASOF	Send DCC accessory packet	
ASON	Send DCC accessory packet	
BOOT	Enter boot mode	
CANID	Set CAN ID	Not on Pi-SPROG 3 v2
DFNOF	Function update	By function
DFNON	Function update	By function
DFUN	Function update	By function group
DKEEP	Keep alive	
DSPD	Speed update	
GLOC	Request session	
KLOC	Release loco	
NNRSM	Factory reset	
NNRST	Reset	Not on Pi-SPROG 3 v2
NVRD	Read node variable	
NVSET	Set node variable	
PCON	Add to consist	Advanced consist only
QCVS	Read CV in service mode	
QLOC	Query loco	
QNN	Query node number	
RDCC3	Send DCC packet (3-bytes)	
RDCC4	Send DCC packet (4-bytes)	
RDCC5	Send DCC packet (5-bytes)	
RDCC6	Send DCC packet (6-bytes)	
RESTP	Emergency stop all	
RLOC	Request session	
RQEVN	Request number of events	
RQMN	Request name	Setup mode
RQNP	Request parameters	Setup mode
RQNPN	Request parameter	
RSTAT	Request status	
RTOF	Request track off	
RTON	Request track on	
SNN	Set node number	Setup mode
STMOD	Set throttle mode	Set speed steps
VCVS	Verify CV in service mode	
WCVB	Write CV bit in ops mode	By session
WCVO	Write CV byte in ops mode	By session
WCVOA	Write CV in ops mode	By address
WCVS	Write CV in service mode	
QCON	Query consist	Returns error – only advanced consist supported

OPCs Output

CBUS Opcode (OPC)	Meaning	Comment
ARST	System reset	Start up
CMDERR	Error during configuration	
ERR	Error	
ESTOP	Emergency stop	
NAME	Response to RQMN	
NNACK	Response to SNN	
NUMEV	Response to RQEVN	
NVANS	Response to NVRD	
PARAMS	Response to RQNP	
PARAN	Response to RQNPN	
PLOC	Engine report	Response to QLOC, RLOC or GLOC
PNN	Response to QNN	
TON	Track on	Startup and Response to RTON
TOF	Track off	Startup and Response to RTOF
STAT	Status	Startup and track overcurrent error

CBUS Events

Events Consumed

None, currently.

Events Produced

The Pi-SPROG 3 v2 generates a subset of SPROG 3 Plus events on its host interface, as indicated in the v2? Column in the table.

CBUS Opcode (OPC)	Event EV	v2?	Data
ACON2	1	X	Main track current sample, mA
ACON2	2		Supply voltage, V*10
ACON2	3	X	Programming track current sample, mA
	0	X	Start of Day

Firmware Updates

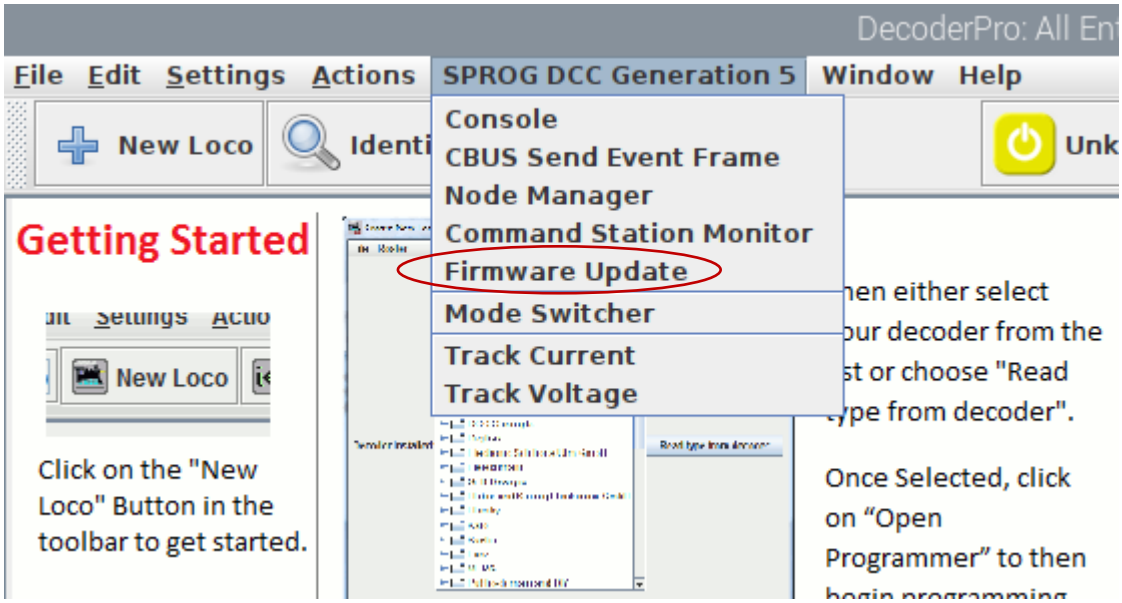
Firmware updates are performed using the Firmware Update tool in JMRI.

Firmware upgrade files (.hex files) will be available from SPROG DCC when an update is available.

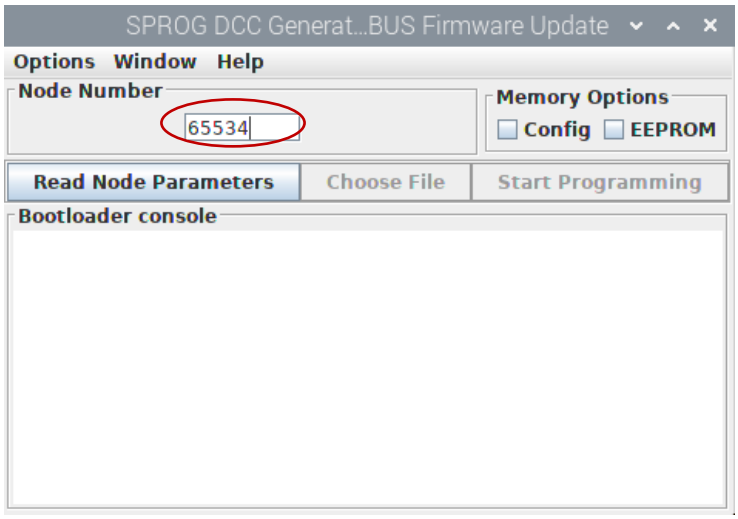
At the time of writing, it is not possible to apply firmware upgrades to the SPROG 3 Plus via CBUS. It is only possible to apply firmware upgrades via USB.

Shut down all throttles (Release or Despatch) and ensure there is no activity on CBUS before using the Firmware Update tool (e.g. disconnect it). Failure to do so may leave the module "bricked" requiring a return to the manufacturer.

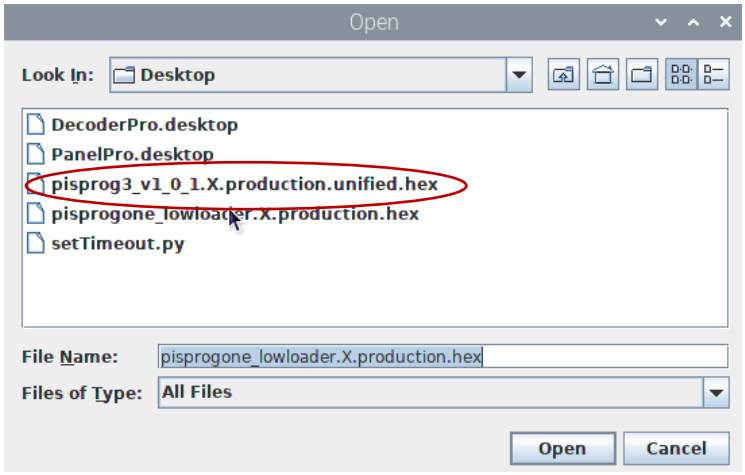
Start the Firmware Update Tool from the SPROG DCC Generation 5 menu in JMRI.



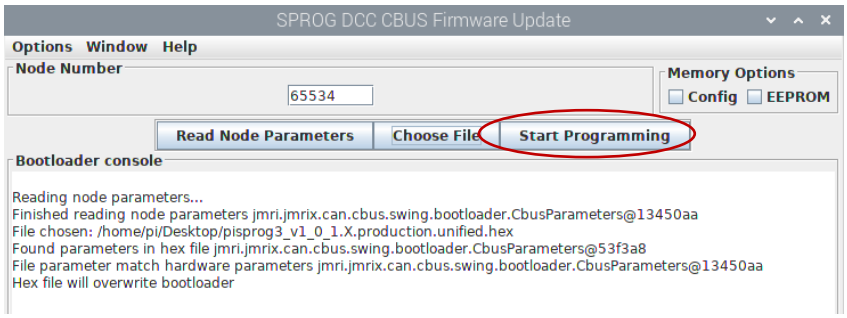
Enter the Node Number (the default is 65534 for the SPROG 3 Plus) and click Read Node Parameters.



If successful, click Choose File and browse to wherever you saved the hex file and click Open



Select EEPROM Memory Option to preserve EEPROM contents (Node Variables) and click Start Programming.



The new SPROG 3 Plus application will be written to the hardware.

Whilst the programming is in progress, the CBUS ACT and FAULT LEDs will flash together on a SPROG 3 plus or Pi-SPROG 3 Plus.

Wait for programming to complete (takes a little while).

Firmware Update window will show the completion message

The new firmware is now installed and running.

Differences to MERG CANCMD

The SPROG 3 Plus family shares a lot of features with the MERG CANCMD, which was originally based on SPROG firmware. There are, however, some differences.

Please also read the Known Issues list. Some apparent differences (not listed here) may be due to features that are not yet fully implemented in Pi-SPROG 3 Plus, or issues which are intended to be fixed in a future release..

	MERG CANCMD	SPROG 3 Plus	Pi-SPROG 3 plus	Pi-SPROG 3 v2
Programming Track	If J7 not fitted	Always [1]	Always [1]	Use Mode Selector
Node Variables	See MERG Docs	This document	This document	This document
Host interface	Via CBUS - Requires CANUSB or similar	Integrated, Isolated USB	Integrated USB, Isolated CBUS	Integrated Serial
CBUS interface	Yes	Yes	Yes – Isolated	No
Booster Interface	Yes	No	No	No
Integrated Booster	1 Amp	2 x 2 Amp minimum [2]	2 x 2 Amp minimum [2]	2.5 Amp
Power Input	AC 15 V DC 15 – 25 V	DC 12 – 25 V	DC 12 – 25 V	DC 12 – 25 V
CAB supply	Yes 12 V DC	No [3]	No [3]	N/A
Track Power at Startup	On	Controlled by NV2, bit 7	Controlled by NV2, bit 7	Controlled by NV2, bit 7

Notes:

[1] Programming track always available unless set to auto-reverse power district in the mode switcher.

[2] Can be safely increased to 2 x 2.5 Amp with a suitable PSU and Node Variable settings. The 2.1mm power connector is restricted to approximately 65W leading to a lower maximum current at inputs above 12 V. A short (< 5 cm) adapter cable or a 2.5 mm to 2.1mm DC barrel plug adapter can be used to suit larger power supplies.

[3] Use a separate 12 V DC supply for CBUS modules, including CABs.

Known Issues

SPROG DCC Generation 5 are new products and under active development.

Updates will be posted on the io group and company website.

Firmware/JMRI versions listed under Fixed? Column may still be in development or testing and not yet generally available.

Description	Affected Products			Fixed?
	S3P	PS3P	PS3v2	
At startup JMRI power icon may be out of sync with the hardware and will show power on, when it is actually off.	X	X	X	JMRI 4.23.4
Need to cycle track power off/on after changing operating mode with power on	X	X	X	Firmware 2.c.1
Changing pre-amble count is ineffective	X	X	X	Firmware 2.c.1
Operations Flag “All stop track off” does nothing	X	X	X	Firmware 2.c.1
Read only Node Variables display as zero in Node Manager	X	X	X	Firmware 2.c.1
Ticking EEPROM selection when Bootloading the SPROG 3 hangs errors and hangs in bootloader	X	X	X	Firmware 2.c.3 JMRI ???
Raspberry Pi will struggle to keep up if the CBUS interface is flooded with packets	N/A	X	X	
PROG LED does not follow PROG track power state when switching modes	X	X	N/A	Firmware 2.c.1
In “Prog track follows main when not programming” mode the loco will try to startup again between programming operations	X	X	N/A	Firmware 2.c.1
Cannot set CANID for messages from JMRI to CBUS	X	X	N/A	JMRI 4.23.5
Full CBUS tools menu not available	X	X	N/A	JMRI 4.23.5
MERG cabs (throttles) are not tested and may not operate as expected	X	X	N/A	Firmware 2.c.1
Bootloading a CBUS module does not work and will “brick” the module	X	X	N/A	Firmware 2.c.1
Voltage and current are incorrect in JMRI meters	X	X	N/A	Firmware 2.c.1
No start of day event is sent	X	X	N/A	
Only a single ARST is sent at startup.	X	X	N/A	
DCC accessory commands are not sent in response to events	X	X	X	Firmware 2.e.1
Support for DCC Functions above F28 in programmer mode	X`	X	X	Firmware 2.e.1 JMRI 5.1.2
Support for DCC Functions above F28 in command station mode	X`	X	X	Firmware 2.e.1 JMRI 5.1.3
MAIN LED stays on after programming	N/A	N/A	X	Firmware 2.b.5
PWR LED is not functional on v1a PCBs	N/A	N/A	X	V1b PCB
Pi-SPROG 3 v2 allows multi-throttles in programmer mode. This differs from previous SPROG behaviour.	N/A	N/A	X	
Single CV programmer allows service mode writes in	N/A	N/A	X	

Operations mode. *These will write to ALL locos on the track.*				
Editing User Flags 2 in Node Variable Edit tab writes to wrong NV – use Generic tab instead				JMRI 5.1.5 (when available)
Documentation				
Description of Current Limit NVs corrected				

Glossary

Item	Meaning
Accessory	An element attached to CBUS or DCC that has a unique address or number and can be controlled by suitable messages, e.g., a point motor connected to a DCC accessory decoder
Cab	See Throttle
Combi	A CBUS module that both produces and consumes events
Configuration Variable	Value written to a DCC decoder to customize its operation
Consumer	A CBUS module that receives events
CV	See Configuration Variable
EN	See Even Number
EngineDriver	Android app for controlling trains, connected via WiThrottle protocol to JMRI
Event Number	A unique number assigned to each event
EV	See Event Variable
Event	Something that happens on the layout, e.g., a switch closure – see also Event Number
Event Variable	Value written to a node to customize its handling of events
Fault	A short circuit – see also Overload
Long Event	An event comprised of both NN and EN
Module	A distinct electronic module. A CBUS module contains one or more CBUS nodes and usually contains associated circuitry to interface to the layout
NN	See Node Number
Node	A distinct CBUS function, part of a module
Node Number	A unique identification number assigned to every CBUS node
Node Variable	Value written to a node to customize its operation
NV	See Node Variable
On the Main	See Operations Mode
OPC	See op Code
Op Code	The first byte of every CBUS message that describes the message type.
Operations Mode	A DCC programming method which happens on the main layout. CVs can be written to a single, addressed, loco. Reading s in operations mode requires extra hardware such as RailCom.
Ops Mode	See operations mode
Overload	A current draw that is higher than the design maximum but not a short circuit – see also fault
Producer	A CBUS module that sends out events
Programming Track	Usually, a short piece of track isolated from (permanently, or switched) the main layout where service mode programming od DCC decoder tales place
Service Mode	A DCC decoder programming method which uses a programming track. CVs can be read from and written to.
Short Event	An event comprised of both NN and EN, but the NN is ignored by consumers
SOD	See Start of Day
Start of Day	Some action, e.g., sending a particular event or sequence of events to initialize the state of the layout, e.g., when it is turned on.
Throttle	Used to control speed, etc., for a loc. May be, e.g., a physical throttle module, a smart phone app or a JMRI software throttle. May have a wired or wireless connection to the command station or controlling computer.
WiThrottle App	iOS app for controlling trains, connected via WiThrottle protocol to JMRI
WiThrottle Protocol	A wireless communication protocol that allows, e.g., mobile phone apps to be used to control trains.

Troubleshooting

Before reporting any problems, please check the SPROG DCC website for any bug reports or updates.

If you are experiencing intermittent faults with your SPROG 3 Plus, please ensure that you are using a good quality DC, regulated power supply.

Useful Links

SPROG homepage https://www.sprog-dcc.co.uk for the latest information, updates, downloads, etc., for SPROG 3 Plus.

North American distributor for SPROG 3 Plus http://www.bbmgroup.com/sprog

SPROG DCC discussion group https://groups.io/g/sprog-dcc for latest news and discussion.

FTDI drivers https://ftdichip.com/drivers/vcp-drivers/

Java Model railroad Interface https://www.jmri.org/ for DecoderPro.

JMRI users group https://groups.io/g/jmriusers/topics for latest news and discussion.

Model Electronics Railway Group https://www.merg.org.uk/ for more information about CBUS

PuTTY (terminal emulator) https://www.chiark.greenend.org.uk/~sgtatham/putty/

National Model Railroad Association for DCC Standards https://www.nmra.org/index-nmra-standards-and-recommended-practices