

LokProgrammer v4.0

Instruction manual

For software versions 4.4.13 and on

Article no.: Unofficial (See "About this manual" pg 42)

September 2014



Table of contents

Declaration of Conformity	3	<i>5.3. Tools</i>	9
WEEE-Declaration	3	6. Driver's cab	11
1. Important notes – Please read this first	4	7. Decoder Information, Read / Write CVs	11
2. Installation and start-up of the LokProgrammer	4	8. The “decoder settings” register	12
2.1 System requirements	4	8.1. Decoder address	12
2.2. Connecting the LokProgrammer	4	8.2. DCC / Analogue	12
2.3. Installing the software	4	8.3. Compatibility	12
2.4. Starting the program	4	8.4. DCC Settings	13
2.5. Software updates	5	8.5 Driving characteristics	13
2.6. Firmware updates	5	8.6. Function views	14
3. LokSound basics	5	8.7. Identification	17
3.1. Sound characteristics of locomotives	5	8.8. Manual CV Input	17
3.2. User defined sounds	6	8.9. Motor Settings	17
3.3. Automatic / Random sounds	7	8.10. Smoke Unit	19
3.4. Digital system / Protocols	7	8.11. Sound settings	19
3.5. CVs	7	8.12. Sound slot settings	20
3.6. Further information about LokSound decoders	7	8.13. Special options	21
4. Purpose of the LokProgrammer software	8	9. Information	22
4.1. Overview	8	9.1. Functions	22
4. 2. Assistant	8	9.2 General	22
5. Main screen	9	10. Sound modeling, adding sounds	23
5.1. View Panes	9	10.1. Sound section overview (opening view page)	23
5.2. Task bar	9	10.2. Sound files	26

11. Sound modeling, flow charts 29

11.1. Sound flow chart basics	29
11.3. Containers and container properties	31
11.4. Complex sound flow	32
11.5. Steam sound flow chart	35
11.6. Sound modeling terminology	37

12. Sound modeling, Examples and Tips 38

12.1. Flowchart drawing example	38
12.2. 74482 GE P42 AMD 103 project examples	39

13. Errors and troubleshooting 41

13.1. Last transition must be unconditioned	41
13.2. Dangling outgoing transition	41
13.3...exit Dangling incoming transition	41
13.4. Unhandled exception	41
13.5. Problems when reading the decoder	42
13.6. Troubleshooting problems	42
13.7. Customer service – Assistance and support	42
About this manual:	42

Declaration of Conformity

We, ESU electronic solutions ulm GmbH & Co KG, Industriestraße 5, D-89081 Ulm, declare herewith in sole responsibility compliance of the product "LokProgrammer" to which this declaration is related to, with the following standards:
EN 71 1-3 : 1988 / 6 : 1994 – EN 50088 : 1996 – EN 55014, part 1 + part 2 : 1993
EN 61000-3-2 : 1995 – EN 60742 : 1995 – EN 61558-2-7 : 1998
The „LokProgrammer“ bears the CE-mark according to the guidelines as per
88 / 378 / EWG – 89 / 336 / EWG – 73 / 23 / EWG

WEEE-Declaration

Disposal of old electrical and electronic devices (applicable in the European Union and other European countries with separate collection system). This mark on the product, the packaging or the relevant documentation indicates, that this product may not be treated as ordinary household garbage. Instead this product has to be delivered to a suitable disposal point for recycling of electrical or electronic equipment.



By disposing of this product in the appropriate manner you help to avoid negative impact on the environment and health that could be caused by inappropriate disposal. Recycling of materials contributes to conserve our natural environment.

For more information on recycling this product please contact your local administration, the rubbish disposal service or the shop where you have purchased this product. Batteries do not belong into household trash! Please do not dispose of discharged batteries in your household trash: take them to a collection point at your local town hall or dealer. Thus you assure an environmentally friendly way of disposal.

Copyright 1998 - 2009 by ESU electronic solutions ulm GmbH & Co KG. Irrtum, Änderungen die dem technischen Fortschritt dienen, Liefermöglichkeiten und alle sonstigen Rechte vorbehalten. Elektrische und mechanische Maßangaben sowie Abbildungen ohne Gewähr. Jede Haftung für Schäden und Folgeschäden durch nicht bestimmungsgemäßen Gebrauch, Nichtbeachtung dieser Anleitung, eigenmächtige Umbauten u. ä. ist ausgeschlossen. Nicht geeignet für Kinder unter 14 Jahren. Bei unsachgemäßem Gebrauch besteht Verletzungsgefahr.

Märklin® ist ein eingetragenes Warenzeichen der Firma Gebr. Märklin® und Cie. GmbH, Göppingen. RailCom® ist ein eingetragenes Warenzeichen der Firma Lenz Elektronik GmbH, Giessen. Alle anderen Warenzeichen sind Eigentum ihrer jeweiligen Rechteinhaber.

ESU electronic solutions ulm GmbH & Co. KG entwickelt entsprechend seiner Politik die Produkte ständig weiter. ESU behält sich deshalb das Recht vor, ohne vorherige Ankündigung an jedem der in der Dokumentation beschriebenen Produkte Änderungen und Verbesserungen vorzunehmen.

Vervielfältigungen und Reproduktionen dieser Dokumentation in jeglicher Form bedürfen der vorherigen schriftlichen Genehmigung durch ESU.

Installation and start of the LokProgrammer

1. Important notes – Please read this first

Thank you for purchasing the LokProgrammer set 53450/53451. With the LokProgrammer you can program ESU LokPilot- and LokSound decoders.

The LokProgrammer 53450 consists of two elements: An interface module that serves as the physical connection between the PC and the locomotive, and the software that can be run on any PC using MS Windows. The set 53451 has an additional USB adapter but is otherwise the same as 53450.

Never was it easier to program a digital decoder than with LokProgrammer. Thanks to the graphic interface of MS Windows you can achieve the optimal adaptation of LokSound decoders even if you have very little or no experience in programming digital decoders. This combination allows you to easily manipulate and adjust the many features and properties of LokSound decoders with your PC.

LokProgrammer also allows you to modify all sound fragments and sound effects stored on the decoder as often as you desire.

ESU provides over 100 different sound files on the ESU web site at www.esu.eu. You will certainly find the right sound for your locomotive.

Please also take note of the license agreement regarding downloading and using the sound files contained in the appendix.

This manual describes in detail how to modify sounds and which methods to use to achieve the desired results.

We wish you lots of fun in the world of LokSound.

ESU electronic solutions ulm GmbH & Co KG, November 2013

2. Installation and start-up of the LokProgrammer

Please note the remarks regarding installation to assure that your LokProgrammer software keeps working to your full satisfaction!

2.1 System requirements

In order to use this software you need a commercially available PC with the following requirements:

- Operating system: Microsoft Windows 98, 2000 or XP, Vista, Win7 32/64bit, Win8 and 8.1
- CD-ROM drive
- One serial port or an USB interface on your PC
- Audio card
- 20MB minimum available memory on your hard disc

For the utilization of the sound files with this software an Audio Card must be installed. All cards with a Windows driver are suitable.

2.2. Connecting the LokProgrammer

The LokProgrammer has to be connected as shown in Figure 1: Use the serial cable respectively the USB-adapter cable provided to connect the LokProgrammer to any available COM port (or USB-port) of your PC. Which port you select is immaterial.

Please make sure that the programming track is completely isolated from the rest of the layout to avoid possible damage of your LokProgrammer hardware!

Also make sure that there are no electrical connections between the individual wires.

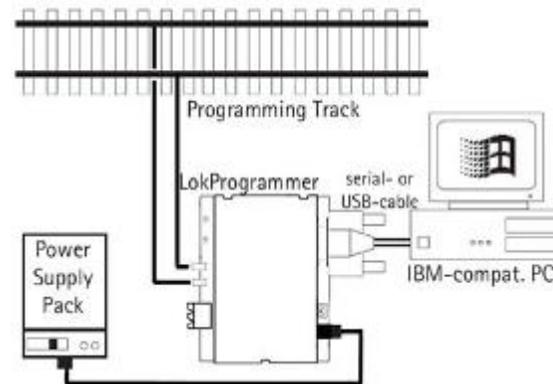


Fig.1.: Wiring the LokProgrammer



Fig.2.: Polarity of Power Supply Connector

There are two options for the power supply:

- Use the power pack with mains plug provided with the LokProgrammer. Connect the output of the power pack to the power supply terminals of the LokProgrammer as per Figure 2.
- Use the AC power output of a model train transformer and wire it to the screw terminals. We recommend this option for programming gauge 1 locomotives

Never connect both terminals at the same time. This could destroy the LokProgrammer!

After connecting the power supply the green LED on the LokProgrammer should light up.

The terminals „Track Out“ on the LokProgrammer are to be wired to the programming track. Polarity is irrelevant.

Make sure that the programming track is fully isolated from the layout!

The two LEDs on the LokProgrammer indicate the following:

Green LED:

- Is lit continuously when supply voltage is available.
- Blinks when the LokProgrammer receives data from the PC.

Yellow LED:

- Blinks quickly when voltage is applied to the programming track and data is transferred.
- Blinks slowly if the LokProgrammer detects a high current and is disconnecting the programming track.

2.3. Installing the software

Make sure that the LokProgrammer is connected as described above and is ready for use.

As soon as you insert the CD-ROM into the drive the installation program is starting automatically.

Should this not be the case select the CD-ROM drive in „Desk Top“ or in the „Windows Explorer“ and click onto „Set up“.

Alternatively you may click on the START button in the tool bar and select „Run“. Then type „x:\setup.exe“ and „OK“. Of course you must enter the name of the CD-ROM drive instead of the „x“ (usually „D“):

After a short while the program should start. Follow the instructions on the monitor and wait until the program is installed on the hard disk.

2.4. Starting the program

The installation program creates an entry in the start menu. Select „LokProgrammer vX“ in the Start menu under „Programs“; „X“ stands for the version number of the software. Select „LokProgrammer“. Then the program will start.

LokSound Basics

2.5. Software updates

ESU offers the latest version of the LokProgrammer software on the web page www.esu.eu. You will find it in the „Downloads“ menu under „Software“. Click onto the Download-symbol at the end of the line. A window opens. Click „Run“. Now the program will guide you through the installation procedure.



Fig.3.: Start Window for Internet Update

There is also an option for an automatic update provided the software is already installed on your computer:

- Go into the Start Menu and select Program „LokProgrammer vX“, (the X stands for the version number of your software).
- Click onto „Internet Update“. A window as per Fig. 3 opens.
- Click onto „Next“. The note „Downloading required Files. Please be patient“ appears. While this window is shown the files required for the update will be installed. Subsequently you can start the LokProgrammer software from the installation window.

Please note that software version 4.4.7+ runs and opens appropriate data (meant for decoder versions 3.5 and v4.0) thus this software incorporates software 2.6.6 and on for v3.5 decoders and versions 4.4.1 and on for v4.0 decoders. You cannot mix and match features of decoder versions, software and firmware is distinctly different for each decoder version, but this software will open sound projects for both decoder architectures. Specifics for version 3.5 decoders are available in the programmer manual for that decoder. This manual is specific to v4 decoders.

2.6. Firmware updates

The firmware is the operating system of the LokPilot- or LokSound decoders.

Please note: Certain new software options can only be activated with LokSound decoders with the latest firmware-update. Firmware installs as required when sound data is written to the decoder.



Privacy Protection:

ESU guarantees that no information will be downloaded from your PC to the ESU website. Data transmission is strictly limited to sending data from the ESU home page to your PC. Your personal data are protected at any time.

3. LokSound basics

In the following chapter it is explained how the LokSound decoder reproduces prototypical sounds, what options are available with digital command control for model trains and which protocols of digital systems are currently available in the market. Should you already have experience with digital systems and also be familiar with locomotive sounds you may skip this chapter and continue reading on page 16.

3.1. Sound characteristics of locomotives

With LokProgrammer and LokSound decoders you can reproduce sounds of steam locomotives, diesel-hydraulic and diesel-electric locomotives, electric locos or locos with manual transmission (e.g.: rail car). Of course the sound sequences are subject to the type of locomotive.

3.1.1. Steam locomotive



The dominant sounds of a steam locomotive are the hissing of the boiler and the exhaust chuffs when the locomotive is running. The chuffs are synchronized to the revolutions of the drivers and therefore accelerate or slow down whenever the locomotive runs faster or slower. We differentiate between locomotives with 2 or 4 cylinders and others with 3 cylinders. A steam locomotive with 3 cylinders generates either 3 or 6 exhaust chuffs per revolution of the drivers while a 2- or 4-cylinder locomotive generates 4 exhaust chuffs per revolution. The exhaust chuffs appear to be louder and harder during acceleration compared to normal running at constant speed. Whenever the valves are closed the only audible noise is the clank of the driving rods.

When the locomotive starts moving, the cylinder valves are open in order to push out any condensed steam and thus to avoid breakage of the driving rods.

This behavior can be simulated with LokSound decoders and with the aid of the LokProgrammer. The individual stages are

divided into separate Driving notches. The different sounds of the respective stages consist of individual recordings of the exhaust chuffs (also refer to Fig. 4 and chapter 11.5 for detailed explanations).

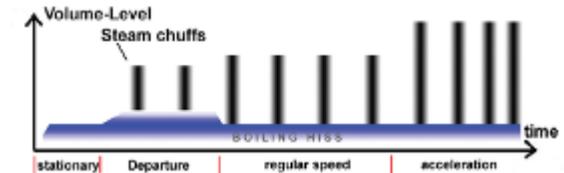


Fig.4.: Performance of a Steam Locomotive

3.1.2. Diesel locomotive (diesel-electric)



Diesel-electric locomotives are in principle electric locomotives with electrical generators that are powered by diesel engines. The diesel locomotive is generally driven at constant Driving notches subject to the speed of the locomotive. Therefore the noise generated changes (driving) step by (driving) step. The quiet electric motor can hardly be heard over the noise of the diesel powered plant. Most diesel-electric locomotives have 4 to 8 throttle notches.

Examples of diesel-electric locomotives are the DB class 232 („Ludmilla“), most American diesel locomotives by GE or ALCO or the MZ-locomotives by the Danish State Railways.

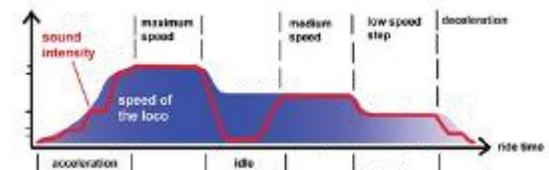


Fig.5.: Performance of a diesel-electric locomotive



3.1.3. Diesel locomotive (diesel-hydraulic)



The main item of equipment of a diesel-hydraulic locomotive is the torque-converter that uses fluids for power transmission. This energy flow is literally „fluent.“

That is the reason why diesel-hydraulic locomotives howl audibly once the throttle is opened and before the locomotive is actually moving. Since the revs of the motor sound depend on the speed, the noises generated during driving change without audible thresholds. Simply put, the sound is directly proportional to the speed.

Locomotives with LokSound decoders behave the same way; first the diesel engine revs up and once the revs are high enough the locomotive starts moving. The pitch of the sound can be adjusted subject to the speed. This is only possible in a combined unit (decoder plus sound module in one piece – for further info also refer to chapter 8.5.4). Examples for diesel-hydraulic locomotives are the DB class V200 (class 220) and the Regio-Shuttle or the DMU41 by the SNCB/NMBS.

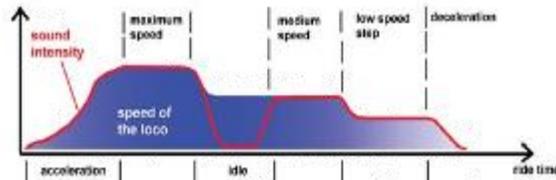


Fig.6.: Performance of a Diesel-hydraulic Locomotive

3.1.4. Diesel locomotive with manual transmission (manual gear gear box)



Diesel locomotives with manual transmission employ pinion gear for transmitting the power from the motor to the wheels similar to automobiles. The clutch is pressed during shifting from one gear to the next and thus the power transmission is interrupted for a short moment. The shifting of gears can clearly be heard in many a diesel locomotive with manual transmission. With the LokProgrammer software you can either store the original sound of gear shifting or you may choose the option „gear shift“ (User-Sound Slot 14) as described in chapter 9.6.2:

Examples of diesel locomotives with manual transmission are the German rail cars VT95 or some shunting locomotives, since manual transmissions are only practical in vehicles of relatively low weight and with low maximum speeds.

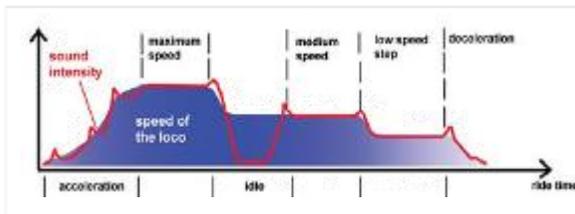


Fig.7.: Performance of a Diesel Locomotive with Manual Transmission

3.1.5. Electric locomotive



There are different sound types for electric locomotives. On the one hand the hum of the electric traction motor(s) is audible; it changes its pitch with the speed similar to diesel-hydraulic locomotives.

Other electric locomotives generate very dominant fan sounds. In some electric locomotives the sound of the fan is constant and therefore the sound does not change during driving.

By and large electric locomotives are not as noisy as other locomotive types and therefore they are ideal for applying User Sounds such as the whistle, horn, compressor, etc. (for more info please refer to chapter 9.5 and 9.6).

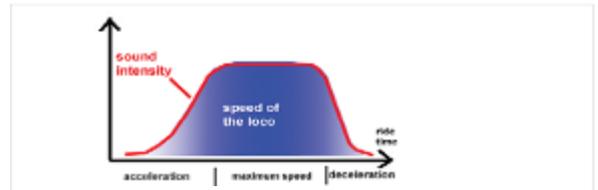


Fig.8.: Performance of an Electric Locomotive

3.2. User defined sounds

User-defined sounds („User-Sounds“) could be horns and whistles, coupler clank, sanding, etc. These sounds can be triggered by pressing a function button on your throttle once you have programmed them onto the decoder. Currently LokSound decoders support up to 28 functions such as head lights, smoke generator, etc. The latest versions of digital command stations such as the ESU ECoS can fully utilize this range.



3.3. Automatic / Random sounds

Random Sounds are triggered automatically and irregularly and can be used for safety valves, fans, compressors, etc.

With the LokProgrammer you can select the time between Random Sounds (details in chapter 8.5.3).

Other possibilities for triggering sounds automatically such as squealing brakes are contained in Decoder Settings and the appropriate flow chart (see chapter 9). Such sounds will be triggered at specific times based on those settings.

3.4. Digital system / Protocols

In this chapter we list all digital protocols for running model trains and setting signals and turnouts that are supported by the LokProgrammer.

3.4.1. DCC (NMRA)

DCC stands for „Digital Command Control“ and was formulated as standard by the NMRA (National Model Railroad Association).

In the early stages operation was limited to 14 speed steps and 80 addresses; today up to 10,000 addresses and 128 speed steps are available.

DCC is downward compatible in terms of control and decoders, e.g. older decoders can be controlled with up-to-date command stations / throttles and with certain limitations new decoders can be operated and programmed with older control devices.

3.4.2. Motorola®

The Motorola®-protocol goes back to 1984 is one of the oldest digital systems for model trains. Due to its age the operational options are limited.

The Motorola®-protocol can only handle 80 locomotive addresses with 14 speed steps and besides the headlight function only four additional function outputs can be controlled (functions 5-8 can be selected with the second Motorola®-address).

Since the Motorola®-protocol is still used in many digital systems ESU decoders are designed to work with this protocol as well.

3.4.3. M4

Since 2004 the MFX®-system is on the market. Theoretically this could run more than 16,000 model locomotives simultaneously with 128 speed steps.

The LokProgrammer software deals with certain settings somewhat differently to DCC.

For instance, instead of locomotive addresses the name of the locomotive has to be entered (e.g.: „class 01“ or „ICE“). The allocation of certain parameters to the CVs is also different to DCC.



Do not use the DCC-CVs mentioned from chapter 3.5 onwards for M4!



What does M4 mean?

At some points in this manual you will notice the term „M4“ for the first time and rightly wonder what this might mean.

This question can be answered quite simply: from 2009 forward, M4 is the name of a data protocol that was chosen by ESU to be implemented in their decoders. Decoders with the M4 protocol are one hundred percent compatible with command stations using mfx®. At such stations (e.g. Märklin® Central Station®) they will be recognized automatically and all playing functions are available just like when using mfx®. On the other hand, our ESU command stations using M4 will recognize all (Märklin® and ESU) mfx® decoders without any restrictions and will still work without any problems. As the (mutual) inventor of mfx® we can assure you of this.

In short: the technique stays the same, only the name has been changed.

3.4.4. Selectrix®

Selectrix® is another digital system. In contradiction to DCC the locomotive addresses are not transmitted individually but in groups. Thus it is limited to the driving sounds and Random Sounds but it is not possible to trigger any user defined sounds (e.g.: a whistle or bell). Selectrix® is almost exclusively used for N scale and Z scale; therefore it is also supported by the ESU LokSound micro decoder.



It is important not to confuse these systems when programming any sounds. For instance it is not possible to store any M4-project files on a DCC-decoder let alone to replay them.

3.5. CVs

3.5.1 Definition and application

CV stands for „Configuration Variable“. CVs can have values in bits or bytes. The CVs with bytes can have a range from 0 to 255 while the CVs programmed in bits function as on / off-switches.

Examples:

CV 63 (sound volume) is a CV that can be programmed byte-wise with a maximum value of 192. The value 0 means no sound while 192 stands for maximum sound volume. (150%)

In CV 49, bit 0 is a „switch“ for activating load compensation (as per 8.3.2). If this bit set to 0, load compensation is deactivated, is it set to 1, and then load compensation is active.

The NMRA (National Model Railroad Association) has allocated certain CVs to certain functions. For instance CV 1 is always used for the address, CV 5 for the maximum speed.

Advantages / Disadvantages

Digital decoders can be programmed without the need of comprehensive programming knowledge or equipment. Many digital command stations also offer internal programming menus.



Furthermore the programming with bits and bytes requires little memory space. Programming solely with CVs is not easy to remember and depending on the type of command station it can be quite cumbersome.

Furthermore CVs have only limited effect on sounds in LokSound decoders (e.g.: sound volume). The actual sounds cannot be adjusted with CVs but depend on the actual sound recording. In the LokProgrammer software CVs are shown in registers or as slide controls and can therefore easily be set to the desired values.

3.6. Further information about LokSound decoders

3.6.1. General

At the core of a LokSound decoder is a powerful processor. It is supported by an audio amplifier and a sound memory that can store up to 268.44 seconds of sound.



The eight channel mixer with active filter can replay eight different sounds simultaneously: One channel is reserved for the driving sounds while the other seven can be used for other sounds (such as bells, whistles, etc.), Random Sounds (e.g.: automatic safety valves or shoveling coal), and brake sounds. All eight channels will be mixed to one output in the decoder and transmitted to the speaker.

The memory of the LokSound decoder can be deleted at any time to make room for new sounds. Thus it is no problem whatsoever to modify a steam sound decoder into diesel sound. You can easily do that yourself with the aid of the ESU LokProgrammer whenever you want to!

Please note: this unimpeded change of sounds is limited to decoders sold for installation into locomotives by the user. LokSound decoders that are installed in locomotives by a model train manufacturer may not always offer this option!

A field at the lower edge of the screen shows the available memory space during programming (in seconds and bytes) as well as the total capacity of the particular decoder. Select the „Sound“ register and then one of the sound displays in order to see this (also refer to chapter 9).



If you wish to save some files but do not have enough memory space on the decoder you may have to delete some sound files from this project. Alternately you can shorten some of the sound fragments with your audio-program.

3.6.2. Connecting the speaker

The speaker is the end piece of the sound equipment. Of course we can only install small speakers into our model locomotives.

Therefore the speaker must meet a very demanding specification. ESU offers a range of speakers of different size and for different decoder types.

Please note that the audio output of the LokSound v3.5 decoder is designed for 100 Ohm whilst v4.0 and XL require 4,8, and other ohm ratings. Please see your decoder for specific speaker ratings.

3.6.3. Suitable sounds

ESU offers many different sound files for all sorts of locomotives on the website www.esu.eu. Please take note of the licensing conditions mentioned in the appendix regarding the download of sound files. Of course you can program your own sound projects on your LokSound decoder.

Generally you may use all files in Windows *.wav-format for LokSound decoders. WAV is the standard format for storing sounds of any kind on windows. If the recording is noise, music or speech makes no difference.

The files can originate from the CD-ROM supplied with the LokProgrammer, they could be downloaded from the internet or they could be created by you.

Wave-files can be stored in different levels of sound quality on the hard disc. The better the sound quality, the more memory space is required.

In order to achieve optimal sound quality you should record and edit wav files at CD quality (44100hz / 16bit). The program automatically converts the files to the suitable format matching the particular decoder.



Hint:

With the advent of v4 decoder architecture a new high capability conversion utility is included in the programmer software, therefore best audio quality for all v4.0 decoders is realized by recording and editing at CD quality (44100hz / 16bit, stereo or mono) and allowing the programmer software to convert your files as they are imported into the sound project. In this manual we cannot provide comprehensive instructions on how to edit or convert sound to digital files and how to save them on a hard disc. Please observe the manuals that were supplied with your PC or with your audio card, recording device, and the sound editing software you are using to capture and produce your user produced sounds.



3.6.4 Supported hardware

The LokProgrammer software as from version 2.5.0 supports only the LokProgrammer 53450 „LokProgrammer V3.0“.

The number of supported decoders varies subject to the LokProgrammer version.

The versions from 2.6.1. Support the following ESU-decoders:

- LokSound V3.5 with 8 and 16 MBit memory for 0 scale and H0 scale (DCC and Motorola®)
- LokSound micro for TT and N scale (DCC, Motorola® and Selectrix®)
- LokSoundXL V3.5 for G and I gauge (DCC and Motorola®)
- LokSound M4 for 0 and H0 scale for the users of Märklin® systems.

In addition the following (partly older product versions) are supported:

LokSound V3.0, LokSoundXL V3.0, LokSound2, LokSoundXL V2.0, LokPilot, LokPilotDCC, LokPilotXL, LokPilotXL DCC.

Software versions from 4.4.7 support v3.5 and v4.0 decoders.

The LokProgrammer software is subject to continuous development. In order to assure that you always work with the latest software version you should regularly call up the internet update facility. Whenever a new version with extended functionality and bug fixing is available it will be placed in the download section on our website.

The appearance on the screen may change subject to the features of a specific decoder. Therefore in certain cases only some of the features described here will be active or even more options may be available. Please always refer to the manual supplied with the decoder.

4. Purpose of the LokProgrammer software

In the following chapters the program functions of the LokProgrammer will be described. First the general functions and then the more special possibilities of adjusting ESU decoders (LokPilot and LokSound).

The appropriate CV in the DCC protocol for each option will be named as well as which setting is supported by which ESU decoder. LP stands for LokPilot, LS for LokSound.

Please bear in mind that you can only fully utilize the potential features of a decoder with the latest firmware.

4.1. Overview

- Setting / changing of all parameters of ESU decoders: all options can be set comfortably on the PC. Of course it is still possible to manually adjust any CV via digital command stations such as the ESU ECoS-command station.

- Modification of sound files, that are stored on an ESU LokSound module: it is possible to change all sound files on the LokSound module at any time, e.g. also at a later stage. Thus you can compose your own sounds using anything as source that can be saved on your PC: locomotive sounds, music, speech, etc. There are no limits to what you can do.

It is for instance easily possible to change the sounds from a steam locomotive to a diesel or electric locomotive - or vice versa.

- Test new ESU sounds: With the aid of the virtual cab (see chapter 6) you can test decoders on the programming track.

Limitations: Select decoders only allow full sound project installs, no individual sounds may be written. V4 project must not be protected in order to change individual files. Sound project must be available and open in LSP (LokSound Programmer Software) in order to change sound files. Decoder sound cannot be read into the software from the decoder. (See chapter 10 and on.)

4.2. Assistant

As soon as the software is started the assistant window pops up on the monitor. This enables you to call up the most important functions of the program. Subject to which function you select the appropriate window appears immediately. With the help of the assistant you can deal with important tasks easily and quickly.

The assistant helps you to carry out the following tasks:

- To read out decoder data for comfortable evaluation and modification.
- To completely modify the sound files of a decoder in order to easily change a steam sound decoder into one for a diesel locomotive.
- To generate a completely new project
- To open an already saved project.

In order to do this, select the desired option and follow the instructions in the small window

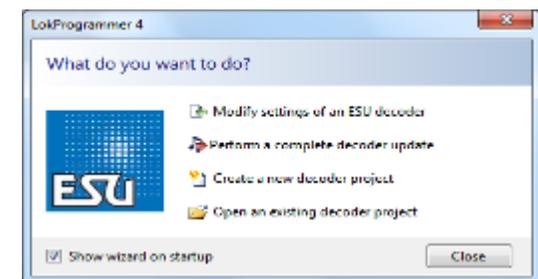


Fig.9. Assistant

Main Screen

5. Main screen

5.1. View Panes

According to the different tasks, the program is divided into different view panes and menus. Figure 10 shows the main screen of the LokProgrammer software and its main components:

- Drivers Cab: Here you can test decoders in an easy manner
- Read/Write CV's: individual adjustment of CVs provided the decoder supports DCC (NMRA).
- Decoder: for comfortable programming of ESU decoders with a graphic display
 - Information: for viewing information regarding functions and for general information about the file, such as type, country, etc.
- Sound: this serves for modifying sounds or to generate new sound compositions for LokSound decoders.

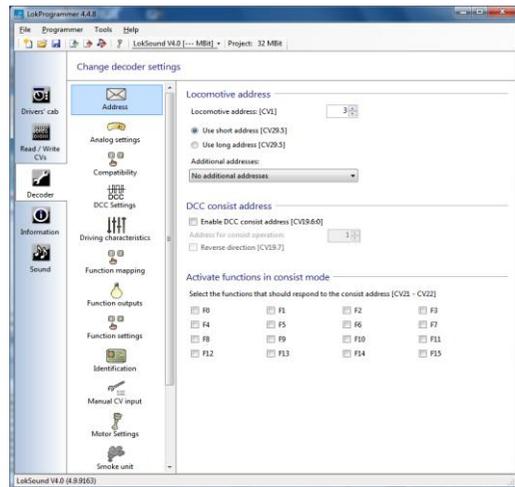


Fig.10.: Main screen

5.2. Task bar

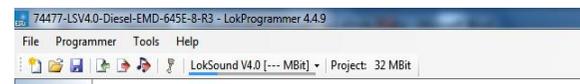


Fig.11. Task bar

- File: in this menu you can do the following with projects: New Project, Open, Import Project (converts v3.5 projects to v4.0 projects (Project to be converted must be first saved to version 2.7.9 or version 4.7.+ software as a v3.5 file)

You can also call up the internet update facility (refer to 2.5) and close the LokProgrammer software.

During file save, all data, settings and sound files will be written into the project file. Project files are saved with the ending ".esu" for v3.5 decoder and ".esx" for v4.0

- Programmer: here you can read and write decoder data and write sound files. Extended decoder data such as type of decoder and version number of the firmware can also be read here.

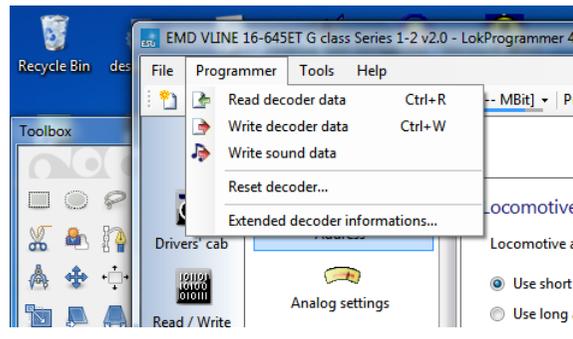


Fig.12.: Menu „Programmer“

“Read Decoder Data”: Prior to changing any data on the decoder it is advisable to read out all decoder data. Place the locomotive on the programming track and make sure the programming track is correctly connected.

Then click “Read decoder data” in the task bar at the top of the screen. The program starts to read the data immediately. Please be patient, this process may take one or two minutes.

The status is displayed in the progress bar.

“Write Decoder Data”: The CVs contained in the project file will be written onto the decoder connected to the LokProgrammer. Click „Continue“ in the window that opens first in order to write the CVs. All data on the decoder will be replaced by the new data.

“Write Sound Data”: The entire sound project will be written to the decoder using this command, replacing all sound data. Partial sound information cannot be written, therefore to use this command the project file for the decoder is required to have been opened first, using the file menu. A submenu dialog opens immediately after issuing this command: (next column)

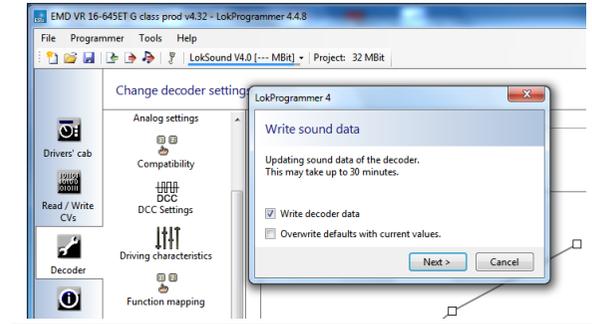


Fig.13.: Write Sound Data „Programmer“

If “Write decoder data” is checked, the decoder data in the project will be written first, followed by sound. If “Overwrite defaults with current values” is checked, then the current decoder data will become the defaults for when the command “reset decoder,,” is issued. When “Next” is clicked, the information write begins and will take up to 30 minutes to complete, depending on amount of sound data being written.

“Reset Decoder”: Resets decoder to either factory defaults or current defaults if “Overwrite defaults” has previously been used.

5.3. Tools

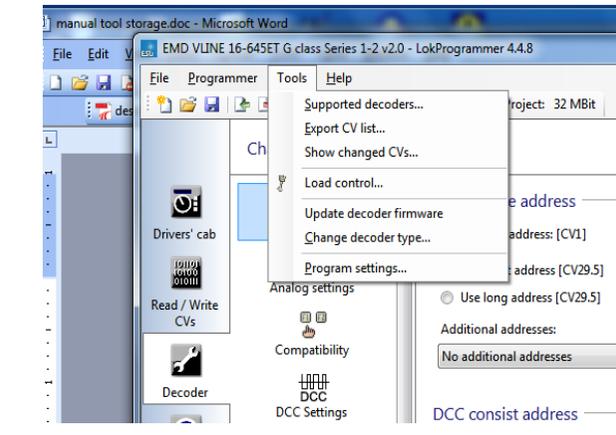


Fig.14.: Menu „Tools“

The Tools menu provides options that provide information about the current decoder and allow you to take certain actions in regards to the current decoder you are working with: (see next page)

Main Screen page Cont.

“Supported decoders,,“: This provides a listing of all the decoders currently supported by the programming software version you are currently using.

“Export CV List,,“: Creates a .txt file and opens a file save dialog, allows you to save a listing of the current CV’s for future use, such as comparisons, etc.

“Show changed CV’s,,“: Opens a sub-dialog box that displays a listing of only the CV’s that have been changed since the last time the project file was opened. This is extremely useful if you intend to perform an action such as manually programming a decoder. See figure 15 below:

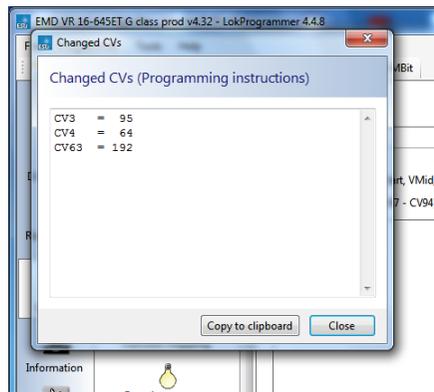


Fig.15.: Changed CV’s, sub dialog „Tools“

“Load Control,,“: Opens a sub-dialog box that lists common DCC motors that have performance templates built into the software based on each motor type. Selecting one from this list that matches or closely matches the motor you are using in your model and provides a starting point for setting up your model. See fig 16 at top of next column:

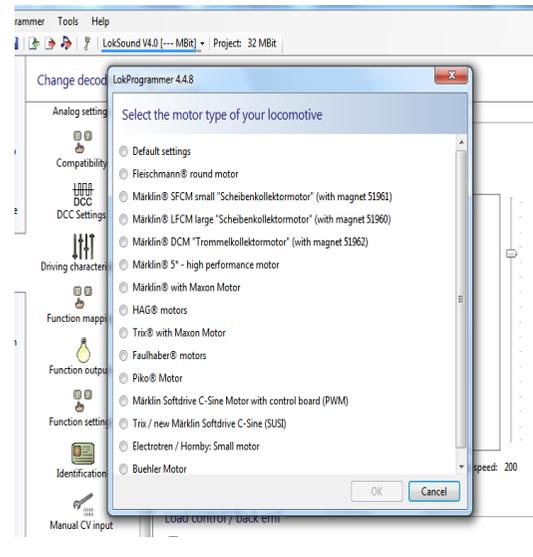


Fig.16.: Load Control, sub dialog „Tools“

“Update decoder firmware“: Commands an activity that will either confirm the current decoder is at the current firmware, or will write the current firmware to the decoder. Note, this action is also incorporated in the commands that write decoder data.

“Change decoder type“: This allows you to open a project such as one for a LokSound Micro or Standard, and change the decoder type to another supported decoder, such as an XL. Note; it is not possible to change a decoder type from a v4 project to a v3.5, this option is reserved for working within the same decoder architecture.

“Program Settings“: Opens a sub-dialog that allows you to set or change current programmer settings; such as file directories, languages, display CV’s, etc. This is also where com settings are controlled for communication with the LokPprogrammer hardware. See figures 17 and 18 next column:

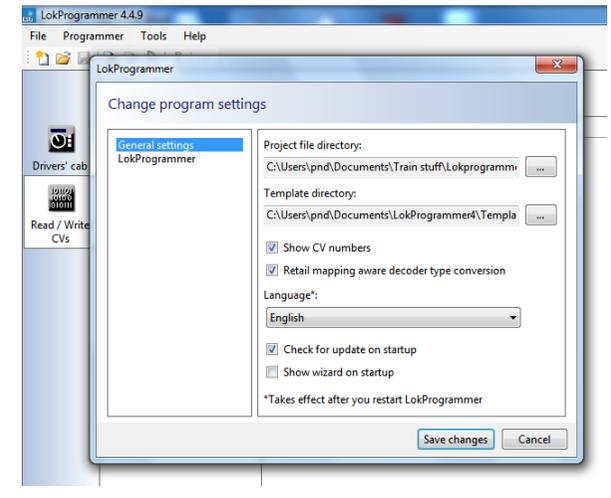


Fig.17.: Program Settings, sub-dialog„General“

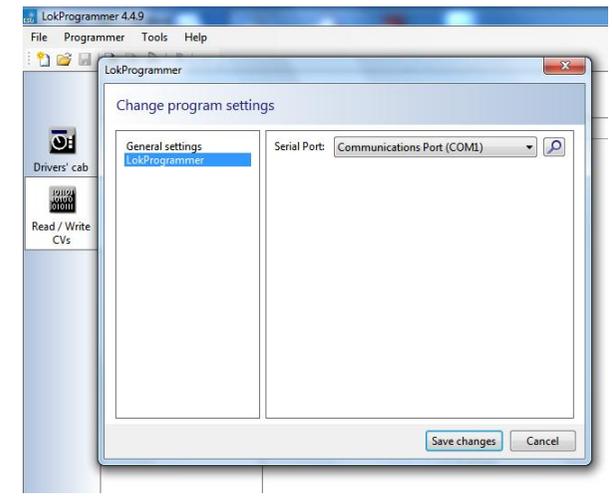


Fig.18.: Program Settings, sub-dialog „LokProgrammer“



6. Driver's cab

With the aid of the driver's cab you can test decoders and sound projects. You can run the locomotive and trigger all functions. Therefore you can test run your locomotive on the programming track with the LokProgrammer.

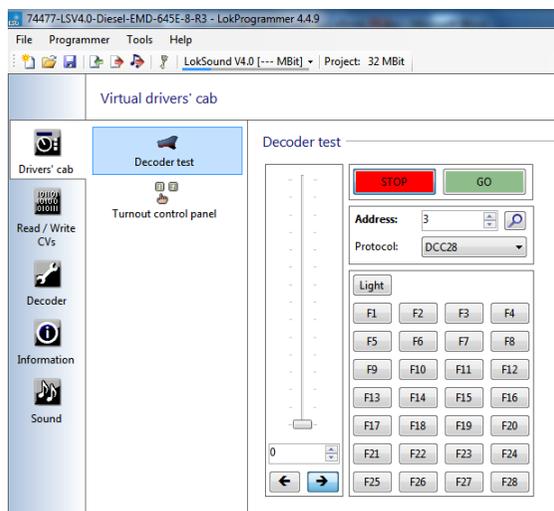


Fig.19.: Virtual Driver's Cab

Included in the Drivers Cab section is the Turnout Control Panel which allows you to also test the Switch Pilot decoder.

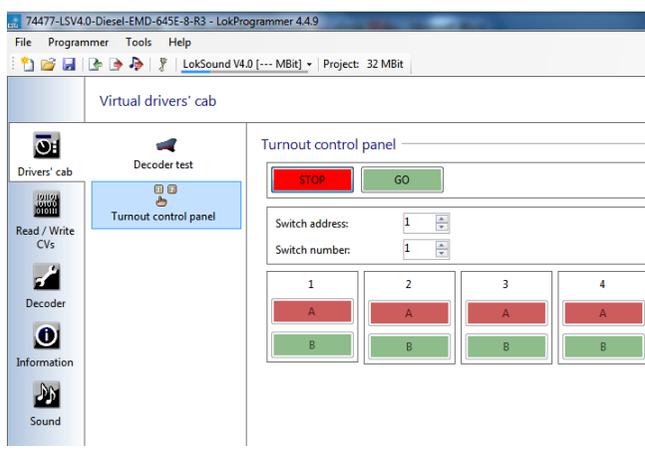


Fig.20.: Turnout Control Panel

There are some limitations though: the LokProgrammer power supply limits the permitted current to about 400 mA. Should the motor of the locomotive draw a higher current then the over current protection will be triggered and the power to the programming track will be shut off. This is indicated by the blinking yellow LED on the LokProgrammer. In this case deactivate the virtual cab and then turn it on again. All other functions in this register are self explanatory: You can enter the address and the number of speed steps. Please make sure that the speed steps matches the ones set on the LokProgrammer.

The LokProgrammer can run locomotives in DCC format, as from version 2.5 also in the Motorola® format. Due to the hardware the LokProgrammer cannot handle M4. Test your M4 projects in the Motorola® format.

Please check that your programming track is fully isolated from the mainline of your layout prior to turning on the virtual cab. Should there be any electrical bridge it could damage the LokProgrammer (also refer to 2.2.)!

Activate the locomotive for the test run by clicking the field „Activate Cab“.

Control the speed of the locomotive with the slide throttle. Clicking into the appropriate fields turns functions on and off. Up to function F12 you may also press the numbers on your computer keyboard.

Please bear in mind that running a locomotive with the LokProgrammer cannot and should not substitute a command station: due to the limited power of the power pack you will not be able to run more than one locomotive at any one time. The virtual cab simply gives you the opportunity to quickly test run your locomotive.

7. Decoder Information, Read / Write CVs

In the register „Read / Write CVs“ you can perform 2 actions, first in the list is an icon labeled “Decoder Information” (see figure 21), when the button titled “Read decoder information” is clicked, the decoder currently connected to the programmer will be read, and the information will be displayed on the decoder information dialog as shown in figure 21.

Also within this register you can read and write individual CV's on the currently active decoder attached to the programmer (see figure 22) , this is done as follows: Select the register „Read / Write CVs“. Then to Read a CV:

- Enter the number of the CV you want to read in the upper data entry field.
- Press the button „Read CVs“.
- The result will be shown in binary and decimal format.

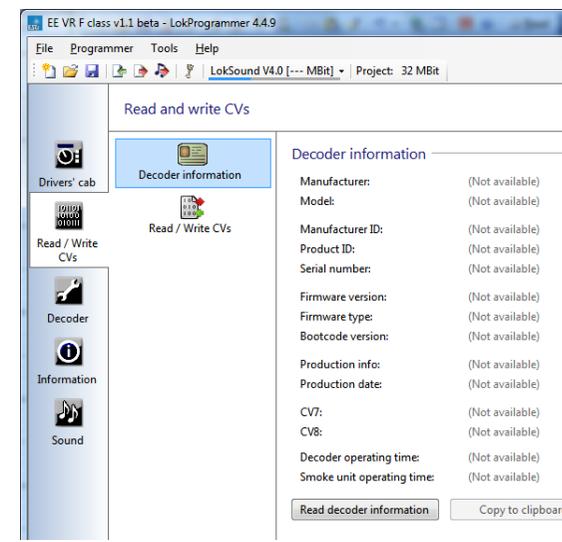


Fig. 21.: Decoder Information

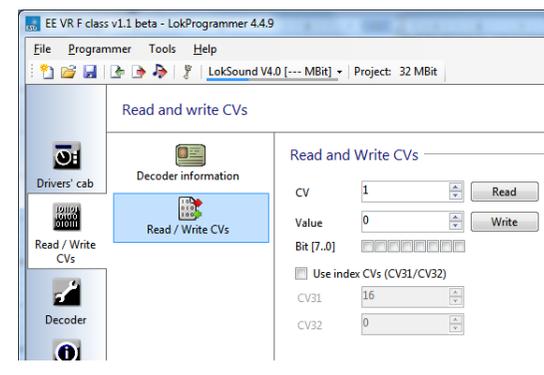


Fig.22: Read / Write CV's

Write a CV:

- Enter the number of the CV you want to write in the field at the top.
- Write the new value of the CV in the lower data entry field.
- Click onto the button „Write CVs“.
- The CV will be overwritten with the new value.
- The Index CV's are also shown, and should be set correctly in order to write values correctly.

Note: Similar to POM (Program on the Main), does not change sound project values.

8. The “decoder settings” register

All settings regarding the motor control and CV configuration (such as Function mapping, Sound settings, DCC settings, etc.) of the decoder are handled in the “Change decoder settings” register. Please note that this Register is initially empty when you start the program. Information will only be displayed in this field after you have generated a new project, opened an existing project or read out a decoder. Projects are an image of all data stored on a decoder.

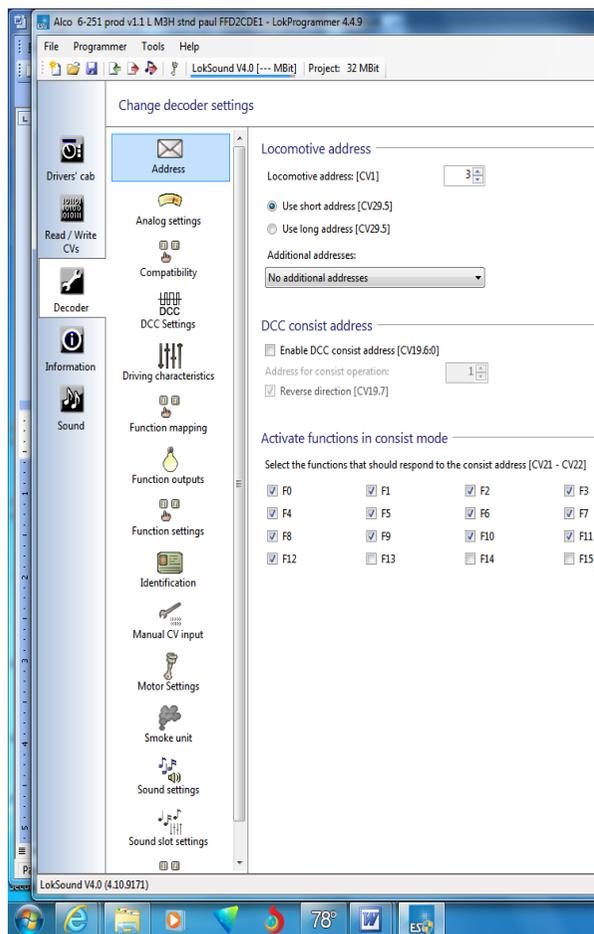


Fig. 23.: The „Change decode settings“ register

The buttons that let you go to the different options are on the left of the screen. Besides movement and sound behavior you

can adjust specific settings such as brake mode, address, etc. On the following pages we will explain the parameters and options.

8.1. Decoder address

8.1.1. Address (CV 1, CV 17, CV 18)

All modifications of the address are done in the window „Address“. Subject to the decoder type so called short (two digits, CV1) or long addresses (four digits, CV17 and CV18) can be used.

Please note that any settings in these CVs are only effective for operation with NMRA-DCC compliant command stations.

When operating decoders with the Märklin® / Motorola®-protocol a separate address, namely the Märklin®-address is valid.

You may enter a second address for M4-decoders in Motorola® mode in order to activate F5 to F8. Normally this would be the address of the decoder plus 1.

8.1.2. Consist settings (CV 19)

The DCC consist address is useful for multiple traction. It is also possible to activate function outputs for consists as well as function buttons for consist mode.

In some cases it is desirable to set certain functions in consist mode in such a way that the function is actually triggered by pressing one button in both (or all) locomotives (e.g.: lights).

Click onto the appropriate button of the function that should be activated in consist mode.

8.2. DCC / Analogue

Supported analogue modes and settings (CVs 13, 14, 50, 125, 126, 127, 128, 129, 130). In analogue mode load compensation is not active. Therefore by using the appropriate slide control you can adapt the start voltage and the maximum speed separately for AC or DC Analogue mode to match the characteristics of your motor or transformer. Furthermore you can select the functions that should be active in analogue mode (DC, AC or both; CV 50).

8.2.1. Active functions in analog mode (CV13, CV14)

Since most analogue layouts do not have input devices to trigger functions, these parameters allow you to pre-select which functions should be automatically active in analogue mode. It is recommended to turn on the sound (default value F1 for European projects, F8 for USA) and the smoke generator of steam locomotives (often F4). Also lighting should be active if desired. Checked functions cannot be controlled during analog operation, they are either on (checked), or off (unchecked).

8.2.2. AC Analogue mode (CV 29, CV50)

Activates the AC analogue mode, and allows setting of Start voltage (minimum speed) (CV127) and Maximum speed voltage (CV128).

8.2.3. DC Analogue mode (CV29, CV50)

Activates the DC analogue mode, and allows setting of Start voltage (minimum speed) (CV125) and Maximum speed voltage (CV126).

8.2.4. Analogue voltage hysteresis (CV 130, CV129)

The motor will stop when the voltage falls below start voltage minus motor hysteresis voltage. Functions will be activated when the voltage reaches the motor start voltage minus the function difference.

8.3. Compatibility

ESU v4 decoders have certain characteristics built into them to allow them to be configured to enhance operational compatibility with certain dcc command stations. Compatibility settings are enable by checking certain boxes in the programming software. These selection boxes are identified in the compatibility section. Options available are:

8.3.1 LGB MTS (CV49.5)

Checking this box enables serial function mode for f1 through f8 and improves compatibility with LGB Multi Train Systems command stations

8.3.2 Marklin Delta mode (CV49.2)

This option enables Marklin Delta mode in support of Marklin delta dcc systems.

8.3.3 Zimo Manual function (CV49.6)

Zimo manual function can be enabled by checking this option.

8.3.4 Serial user standard interface (CV 124.3)

Selecting this option will enable the decoder Serial User Standard Interface (SUSI), this will the decoder to communicate with up to 3 SUSI devices.

8.4. DCC Settings

There are 2 groups of items you can configure in this view window as shown in figure 24. These are RailCom settings and Speed step mode.

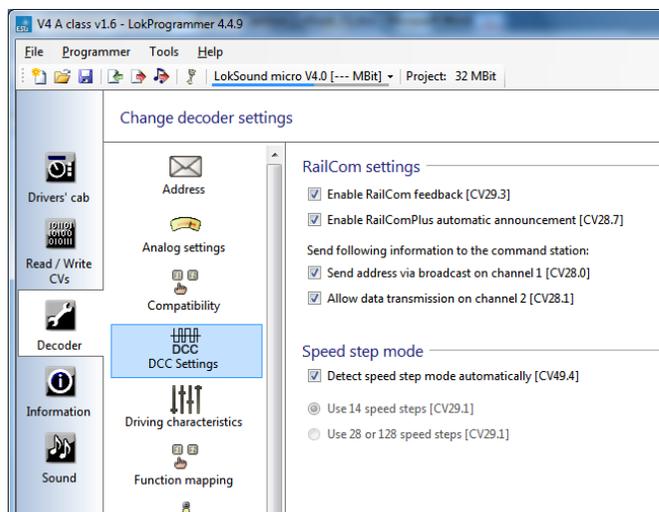


Fig. 24.: The „DCC Settings“ register

8.4.1 RailCom settings CV29, CV28)

These settings allow the enabling or disabling of RailCom information. Checking the first option enables RailCom feedback and allows setting the other 3 items as you desire. LokSound v4 decoders are RailCom enabled and thus offer you the features RailCom provides.

8.4.2 Speed step mode (CV49, CV29)

Here you can adjust more settings for running your locomotive. In DCC mode you have the option of setting the speed steps manually to 14, 28 or 128. Optionally you can check the first option box and allow the speed steps to be detected automatically.

8.5 Driving characteristics

Items within this view window allows the adjustment of several variables that affect overall driving features that are available, such as; acceleration and deceleration (momentum), brake options, speed trimming, and other power handling features.

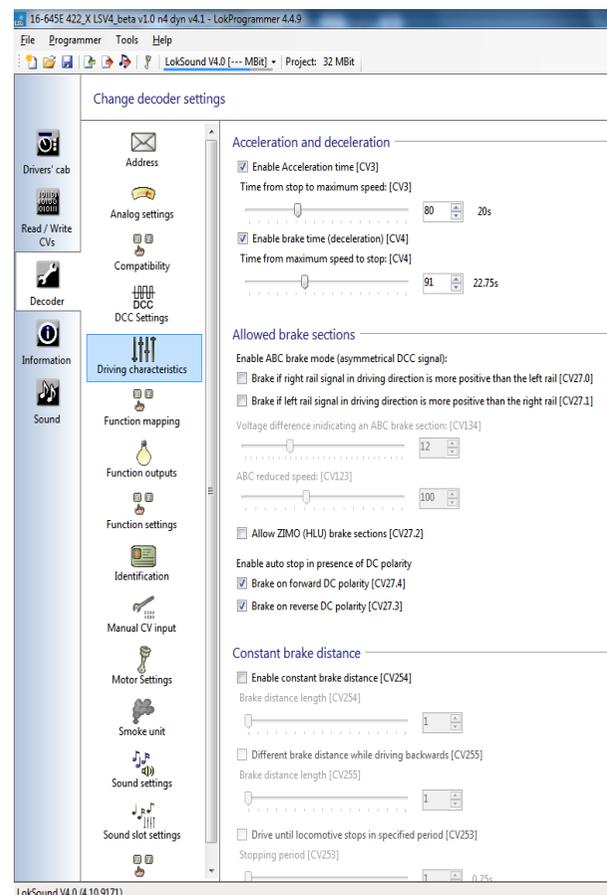


Fig. 25.: The „Driving characteristics“ register

8.5.1 Acceleration and deceleration (CV3, CV4)

Selecting the check boxes for acceleration and deceleration enables and sets momentum values for speeding up and slowing down. Setting momentum values allows the model being driven to act in a more realistic manner, and it allows certain sound features to be enjoyed.

Acceleration (CV3) can be adjusted using the variable slider within a decimal value range of 1 to 255 providing acceleration times of .1 seconds up to 63.75 seconds. Deceleration time (CV4) can be adjusted using the deceleration time across the same range, allowing you to set values for coasting. Higher Momentum settings also allow certain sound features to operate realistically, such as coasting sounds and acceleration sounds.

8.5.2 Allowed brake sections (CV27, CV134, CV123)

A variety of automatic brake sections can be enabled for the v4 decoder; this allows the layout on which the decoder is running to be configured to trigger locomotive braking when a section of track is reached, perhaps a curve, and then the locomotive returns to prior speed upon exit. The various types of brake section detection are enabled by checking the boxes that match your layout(s).

ABC brake mode can be set for either right rail (CV27.0), left rail (CV27.1), or both having a more positive voltage than the other. Voltage difference (CV134) to enable the trigger can be set with a variable slider with a range of 4 to 32, providing a significant flexibility in setting up layouts. The second slider (CV123) allows setting the amount of speed reduction over a range of values from 0 – 255, allowing automatic speed reduction from very slight to a complete stop.

Also available is compatibility with ZIMO (HLU) brake sections (CV27.2) by selecting the option to enable them.

You can also enable auto stop brake sections carrying a either a forward DC polarity (CV27.4) or reverse DC polarity (CV27.3) by selecting the appropriate check box.

8.5.3 Constant brake distance (CV254, CV255, CV253, CV27.7)

Constant brake distance allows you to precisely control where your trains will stop on your layout. This effect works in conjunction with brake sections, and it can be used without brake sections by setting only CV254 to determine a distance for braking, along with setting CV27 bit 7. With these settings made the v4 will generate a stop command when ever speed control is set to speed step 0. Stop distance will be based on the value set in CV254. See v4 decoder manual chapter 10.6 "Constant Brake Distance for detail.

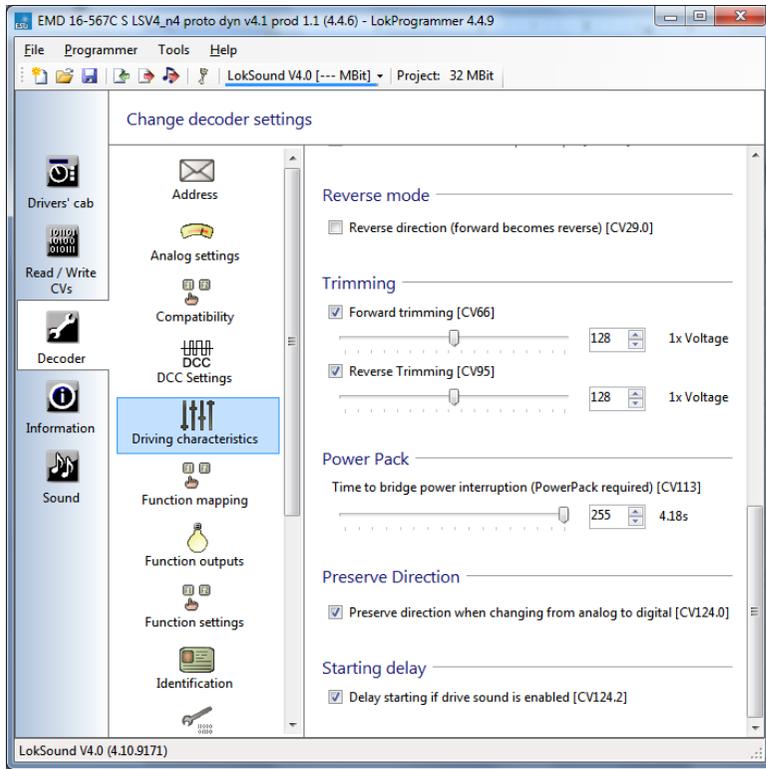


Fig. 26.: The „Driving characteristics“ register

8.5.4. Reverse mode (CV 29)

A tick at „Reverse mode“ changes the direction of travel and the directional characteristics of the headlights. This is useful in case the wiring has been done incorrectly (swapping of track leads or motor leads).

8.5.6 Trimming (CV66, CV95)

The trim function allows you to set the maximum speed separately for forward and reverse movement. The factor that is used to multiply the motor voltage, results from dividing the CV-value by 128 (forward CV 66 and reverse CV 95).

8.5.7 Power Pack (CV 113)

V4 decoders (HO and N) provide for the installation of capacitors or power packs (so called “keep alive” devices). CV113 allows control of the amount of time the device is active. Setting range is 0 to 255 via the adjustment slider; an estimated active time is shown to the right of the setting in seconds. Information for installing capacitors or power packs can be found in the decoder manual chapters 10.9 (configuring) and 6.11 (wiring).

8.5.8 Preserve Direction (CV124.0)

Checking this option keeps the direction constant when the v4 equipped loco transitions from DCC control to Analog controlled track sections.

8.5.9 Starting delay (CV124.2)

Usually, when the LokSound V4.0 sound is idling and you turn up the throttle, the locomotive begins to move only after the Diesel engine has reached notch 1. A steam loco will even release its brakes first and fill the cylinders. Although this behavior is very prototypical, one might not like it because it causes some delay. You can control this startup delay by simply not checking this option. This will cause the LokSound V4.0 decoder to immediately start moving when the throttle is turned up. However, the start up sound will not be synchronized with the motion anymore.

8.6. Function views

Both LokSound V4.0 and LokSound micro V4.0 decoders have identical function mapping. M4 and XL decoders have different screen displays. The display will shift when the decoder type for any given project is changed; therefore, the screen display depends on decoder type. Shown here is the display for V4 standard and micro decoders. A great deal of information is available in the v4 decoder manual concerning function mapping. Of course, which sound is assigned to which sound slot may vary depending on the decoder project. You will find a list with all available project files “Download/Sound files/LokSoundV4.0/” on our home page at www.esu.eu. You may also view and print a list with all functions and the sound slots employed.

The LokSound V4.0 decoder offers powerful and flexible function mapping options: Each function button can switch as many outputs as desired. Each output can be activated by several function buttons. Function buttons can be linked (e.g.: F3 AND F5 pressed simultaneously)

Function buttons can be inverted (e.g.: NOT when F8 is on). Besides the buttons F0 to F28 you can also incorporate the direction of travel or the speed (locomotive is moving / has stopped)

You may connect as many as 5 external sensors. While many model train enthusiasts need precisely these functions for optimal running of all their locomotives setting up function mapping represents so to speak the “free style”

version of decoder programming. Take your time to understand the concept behind it before you start changing any settings. For a complete review of all function mapping CV's, see the v4 decoder manual.

HINT: Even if you do not have LokProgrammer hardware, you can still use the software as an aid in making mapping changes in conjunction with the “Show changed CV's” capability.

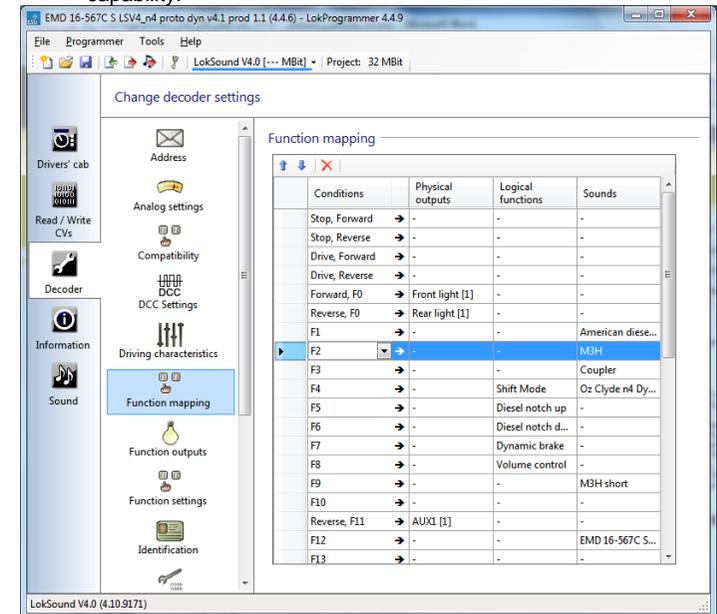


Fig. 27.: The „Function mapping“ register

The function mapping view is arranged as a matrix with vertical columns for conditions, physical outputs, logical functions and sounds. Horizontal rows list each condition and function key for use in mapping as desired. See figure 27.

8.6.1 Function mapping, “Conditions”

The input block (conditions block) shows which condition is required to achieve a certain output. Conditions are for instance “F3 On” or “Locomotive is stationary with direction set to forward, and F8 is switched on”. Configuring the outputs to behave as desired begins by deciding which function or function key you wish to configure to enable an output that will perform the desired action when the function is on, or off.

Change Decoder Settings

Function Mapping Cont.

The figure below; figure 28, displays an example of setting a condition against function key F11 in which the direction is set to reverse, and the key must be ON in order for the condition to equate to "true". This then becomes the condition that must be fulfilled in order for the desired output to be enabled.

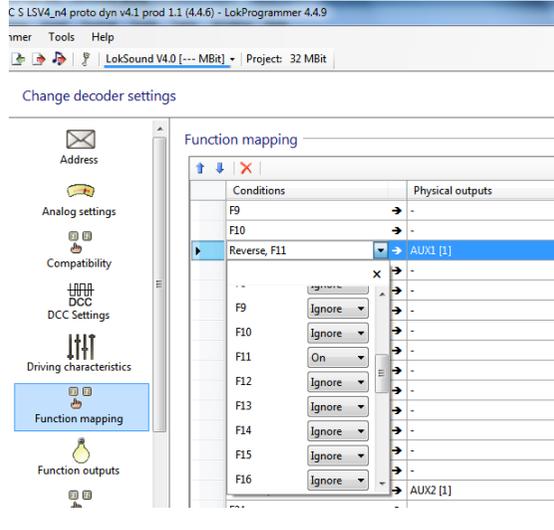


Fig. 28.: Establishing a condition for F11

In this manner all the functions available to the decoder can be configured to behave as desired. The next task is to set the desired output for the function.

8.6.2 Function mapping, "Physical outputs"

In the output block it is shown what action must be carried out when the condition(s) is/are met. This could be, for instance, switching a function output or a sound effect. LokSound decoders have up to 12 physical function outputs. "Headlights" and "Rear lights" are used for lighting, the remaining ones are freely available. Other functions include "Shunting Mode", "Acceleration / Deceleration On/Off" as well as virtual functions like "Sound On/Off". The function buttons ("F buttons") of your command station or throttle activate the function outputs. Generally, F0 is the lighting button, while we count the remaining buttons from F1 upwards.

See figure 29, next column, for an example of configuring the Physical output for F0 to enable the rear light to come on when the condition "F0 is on and direction is reversed" becomes true.

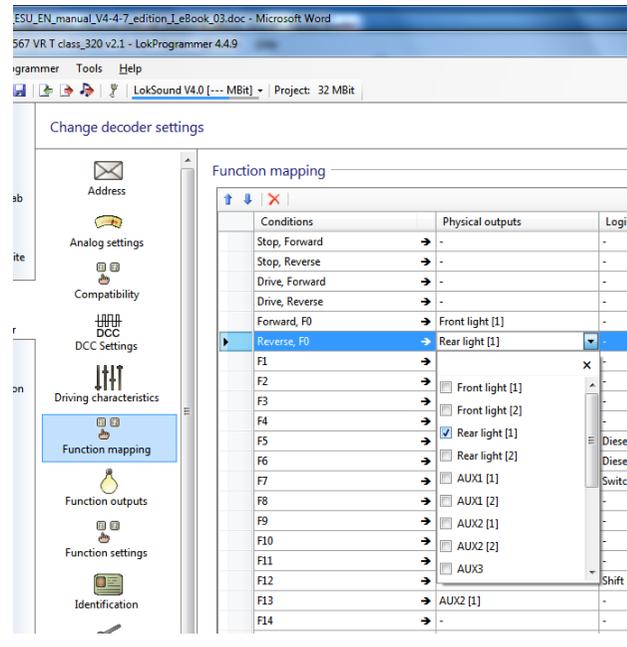


Fig. 29.: Setting a Physical output for Rear Light

Complicated lighting configurations can be set by selecting more than 1 output. For example, we wish to have the front light on bright when running forward, and dim when running in reverse. This can be done setting front light(1) and front light(2) on, and then setting the function outputs for front light(1) as on and front light(2) as dim in the function outputs section. (See figure 30 and 31)

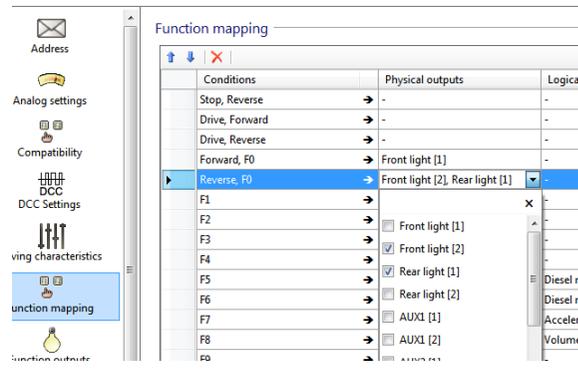


Fig. 30.: Setting a Physical output for Front / Rear

8.6.3 Function outputs, "Configuration"

Configuring the function outputs is done on a separate screen from the mapping portion, each output can be configured discretely, and some outputs such as Front and Rear lights, and Aux1 and Aux2 can each have 2 configurations, to satisfy certain needs, such as running with the front light on bright when going forward, and on dim when reversed. There are many configuration options available, as pictured in figures 31 and 32 dropdown menu options:

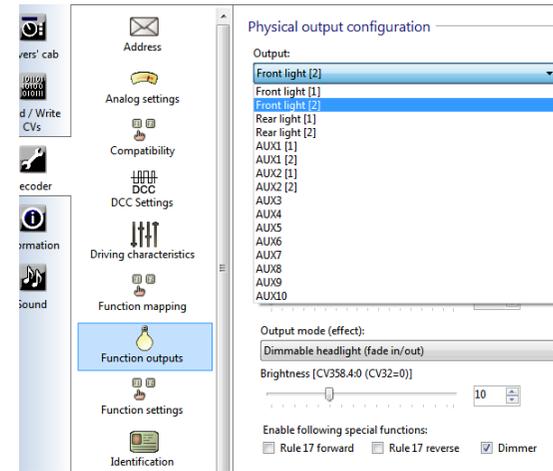


Fig. 31.: Physical output configuration for Front(2) dim

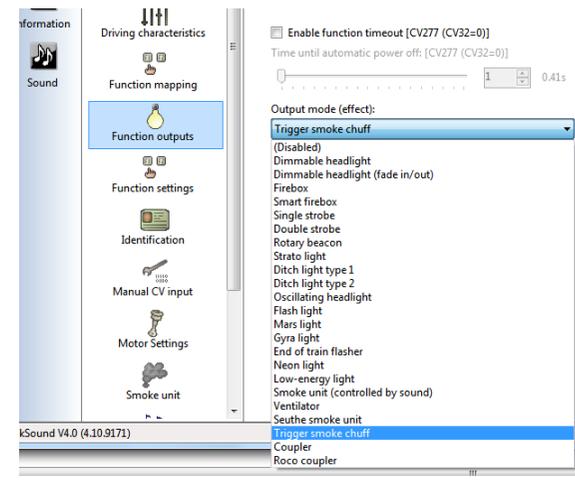


Fig. 32.: Setting a Physical output to trigger smoke chuff

8.6.6 Function settings, “General” and “Automatic uncoupling”

As mentioned in 8.6.5 above, when mapping logical functions to a function key, in some cases you should also use Function settings in order to enable the response you desire when the function key is engaged. General settings are for configuring lighting effects, and allow you to set the blink frequency for lighting situations that require blinking lights (CV112), and for setting grade crossing holding time (CV132). Holding time is the amount of time the grade crossing effect is enabled.

Automatic uncoupling is where automatic uncoupling is enabled (CV246), and where you can set the uncoupling speed across a range of 1 – 255 (same as the speed table range), default is 1 (CV246). Push time (CV248) is the time in seconds the automatic push is in effect; and Move time (CV247) is the time in seconds the locomotive moves away from the uncoupled car(s). Of course “time” relates directly to distance traveled at the speed set for the duration of the automatic uncoupling cycle. See figure 37.

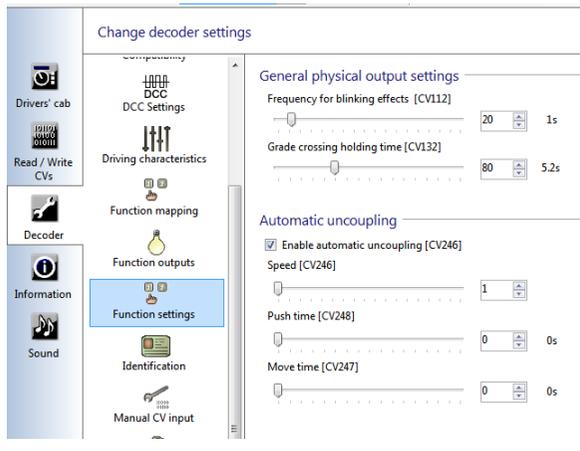


Fig. 37.: Configure Function settings

8.7. Identification

8.7.1 Data fields, “User identification”

The user identification fields are open fields that can be used for any purpose. These are designated as “User ID #1” and “User ID #2”. CV’s are CV105 and CV106 respectively. Value range for both is 0 – 255 and they can be set individually or together. Setting these CV’s does not change any decoder behavior, they are an open code section and can be used for any purpose, perhaps to track a certain version, or function key structure. Default value is 0 for both CV’s. See figure 38

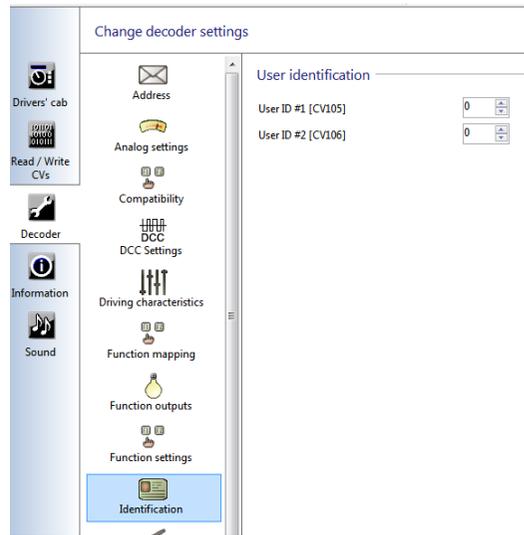


Fig. 38.: User Identification

8.8. Manual CV Input

8.8.1 Manual CV Input

This section of the change decoder data screens is made available should you wish to make manual CV changes without using the preformatted decoder set up screens and views. It is also very useful for researching CV’s. When you export the decoder CV listing (See “5.3 Tools”, Export CV list) the text file that is created during the export will consist of the same data displayed in this view.

8.8.2 CV changes

CV changes can be made directly by overwriting the data in any of the data fields; “Value”, “Binary”, or “Hex”, simply enter the value you wish and then tab out of the field and the data will be changed and the other two data columns will update. Of course it is much easier to enter direct numeric data in the “Value” column than it is to write Binary or Hex data.

After you are finished making changes, the new values may be written to the decoder using “Write decoder data” (See “5.2 Programmer”, Figure 12)

Note: The ability to make manual changes is very powerful and should be used with care. Consider saving the changed project file under a new name in order to provide a way to rewrite last known good data to the decoder to recover from mistakes. See figure 39.

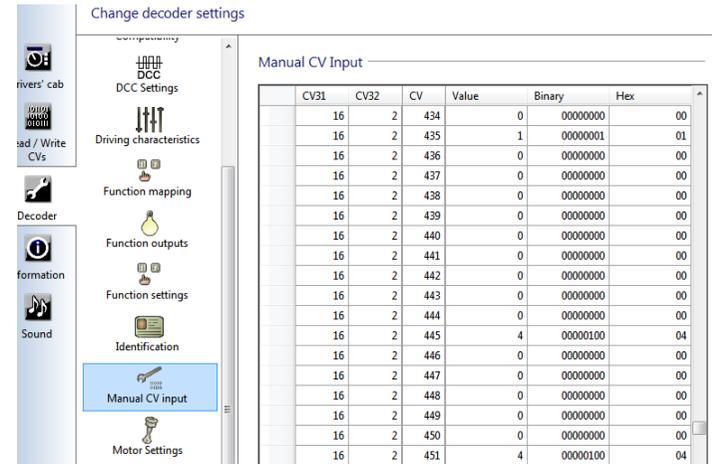


Fig. 39.: Manual CV Input

8.9. Motor Settings

Here is where you configure speed table, load control, PWM frequency, and overload protection settings on the v4 decoder series. Information presented here deals directly with making desired settings using the LokProgrammer tool to configure the decoder, therefore the level of detail presented is not at the same level as is found in the v4 decoder manual. Please use the detailed information found in the decoder manual, chapter 10, as the preferred source for decoder information.

8.9.1 Speed table selection and configuration (CV 2, CV 5, CV 6, CV 26, and CV 67-97)

Here you may select either a 3 point speed table (figure 40) or a custom speed curve (figure 41). If you use the 3 point table you should set minimum and maximum speed (CV2 and CV5) and then pick the mid range speed (CV 6) as desired. You may also use the slider on the right hand side to set both maximum and mid range speeds. In order to avoid possible rough or unusual running, please ensure

Change Decoder Settings Motor Settings

mid range speed is higher than start speed and maximum speed is higher than mid range speed (CV2<CV6<CV5).

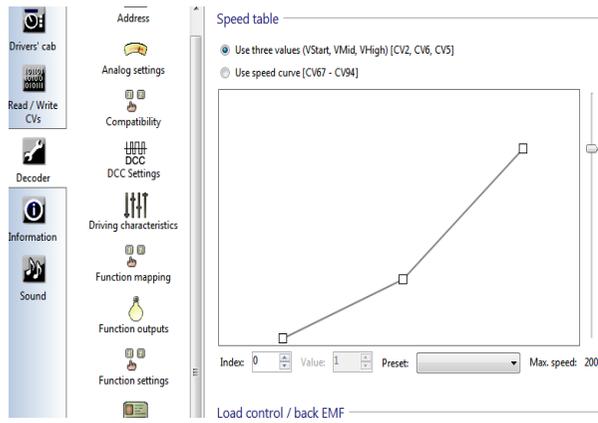


Fig. 40: 3 point speed table

You may also define your own speed curve: simply enter the desired values in the CVs 67 to 94 (also refer to Fig. 16). The decoder will superimpose these 28 values onto the real speed steps. Thus, you can adapt the driving performance optimally to your locomotive. Another option is available using the speed curve option, there are 5 predefined speed curves stored in the programmer software, you may select any of these by simply clicking on the one you desire to start with, and then you can modify them as desired. See figure 41.

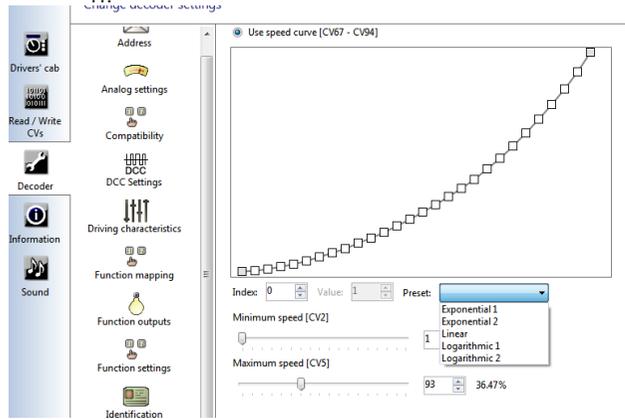


Fig. 41: Speed curve options and set up

Speed curve shape is set by adjusting the 28 point curve as desired, then minimum and maximum speed is set as it is with the speed table (CV's 2 and 5), the decoder then fits the

Speed curve between the minimum and maximum speed settings resulting in a smooth 28 point speed curve within the decoder. Please see the decoder manual, chapter 10 for more detail.

8.9.2 Load control / Back EMF (CV's 53, 52, 51, 55, and 56)
Load control compensation is enabled via the LokProgrammer by checking the box at the top of the load control section, following that, the CV's can be adjusted as desired. **HINT: If the decoder is installed in a model locomotive, there is an "auto set" feature you can use to set the initial parameters. See decoder manual, chapter 11 for details on how to use this feature.** Then if you need further adjustment you can "read decoder data" to load the automatic settings into the programmer for fine tuning. See figure 42.

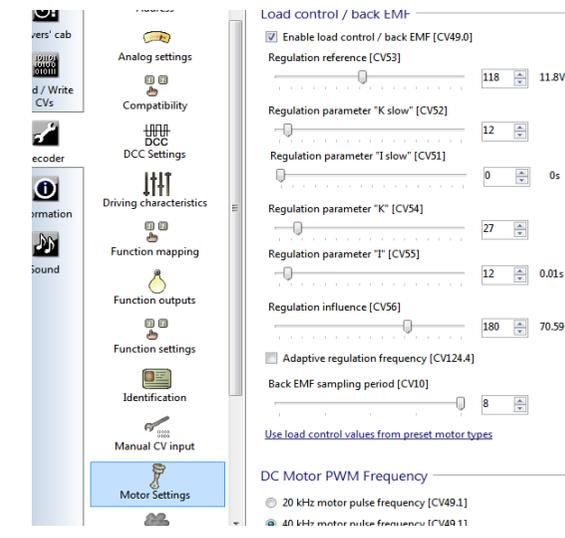


Fig. 42: Load control / back EMF setup

LokSound decoders enable you to adapt load compensation to the motor with CVs 53, 54 and 55. If the recommended values above do not lead to acceptable results, you can further optimize them. Especially for the slow driving sector (speed step 1) the LokSound V4.0 with CV 51 and CV 52 to change the gain control. This helps to avoid any jerking while driving extremely slowly.

Parameter K, stored in CV 54, influences how strongly load control will affect the driving performance. The higher the value, the more load control will respond to any changes and try to adjust the revs of the motor. Parameter K needs adjustment if the locomotive runs unevenly (jerks). Reduce the value of CV 54 by 5 and test-run the locomotive to see if there are any improvements. Repeat these steps until the locomotive runs smoothly at speed step 1.

Parameter I, stored in CV 55, provides important information to the decoder on how much inertia the motor has. Motors with large

flywheels naturally have more inertia than smaller ones or coreless motors. Adjust parameter I if the locomotive jerks somewhat just before it stops or jumps at lower speeds (lower third of the speed step range) or simply does not run smoothly. Increase the value by 5 starting with the initial value for motors with very small or no flywheels. Reduce the value by 5 starting with the initial value for motors with large flywheels. Test again and repeat this procedure until you arrive at the desired result.

In CV 53, you set the EMF reference voltage generated by the motor at maximum revs. This parameter may have to be adapted subject to the track voltage and the efficiency of the motor. If the locomotive reaches maximum speed when the throttle is set to about three-quarter and the top third of the throttle has no influence on the speed, then you should reduce the value of CV 53. Reduce the value by 5–8 and test the locomotive again. Repeat this process until the locomotive just reaches its maximum speed when the throttle is fully open. On the other hand, if the locomotive moves too slowly at full throttle then you should increase the value of CV 53 step by step until the maximum speed is reached.

Together with the LokSound V4.0 decoder an additional CV 52 has been introduced which separately determines the gain control considerably for the whole slow driving sector in speed step 1. If you are not satisfied with the driving behavior when the locomotive drives slowly or starts, while everything is fine with the medium and high speed steps, you should increase the value of CV 52 by 5–10 higher than the value set in CV 54. Here you can adjust the inertia of the motor separately for slow speeds and starting from a stop. The desired value is to be entered into CV 51. The parameters "K slow" and "I slow" jointly influence the behavior at speed steps 1 and 2 while the parameters CV 54 ("K") and CV 55 ("I") are responsible for the remaining speed steps. The decoder computes a speed curve in order to avoid any abrupt changes.

The decoder operates ex works with a mutable (adaptive) regulation frequency to drive the motor as precisely as possible. However, as a result some motors might show a nasty buzzing noise. For such motors you are able to set the regulation frequency on a constant value by checking the box to enable adaptive regulation frequency. You may also adjust the Back EMF sampling period using the slider, values range from 4 to 8 (CV10).

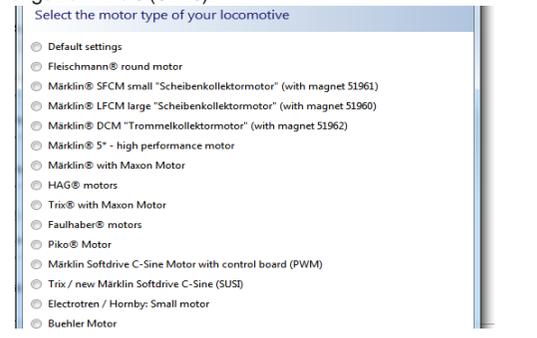


Fig. 43: Load control preset motors

You may apply presets that have been tested by ESU for many types of motors by clicking the option "use motor control values from preset motor types" See figures 42 and 43. See v4 decoder manual for mote detail.

8.9.3 DC Motor PWM Frequency (CV 49.1)

LokSound decoders load control works normally with 40 kHz. Sometimes it can be useful to reduce this frequency by half. For motors with little power due to high inductivity.

• If suppressors (such as capacitors, chokes, etc.) disturb load control but cannot be removed (e.g. some older Gützold® locos).

Check the desired setting in the Programmer software to set CV49 bit1 as desired for either 40 or 20 kHz.

8.9.4 Motor Overload Protection (CV 124.5)

ESU v4 decoders offer a setting to enable motor overload protection if needed. This is separate from the normal thermal protection for the decoder as described in decoder manual, chapter 6.10.1. Motor overload protection enables the decoder to monitor BEMF reactions of the motor to various operational loads; if the decoder detects motor overworking (possibly due to mechanical issues or stress) the decoder will stop the motor without affecting other decoder operation or sound.

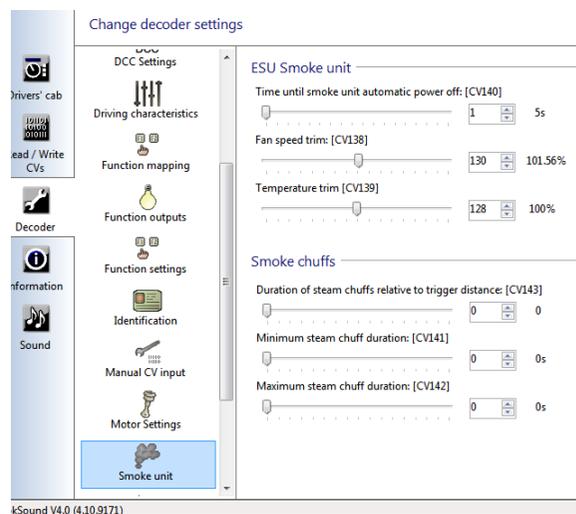


Fig. 44: Smoke Unit set up

8.10. Smoke Unit

ESU v4 DCC decoders enable a wide variety of output choices, one of which is smoke generators, such as ESU smoke generators or those from other vendors.

You can wire any kind of load such as LEDs (light emitting diodes), smoke generators or similar devices to the function outputs provided the maximum current draw is less than that of the decoder output. The permitted maximum current draw per function output is listed in chapter 20 of the decoder manual under „Technical Data.“, please confirm your intended smoke generator load is suitable for your decoder. It is important to stay within the current capabilities of the decoder you are installing and programming in order to avoid possible damage.

This section of the decoder settings works in conjunction with several function setting items listed in section 8.6 of this manual, such as function mapping, physical and logical outputs, and function outputs. Please set up all areas in order to control your smoke output as desired.

ESU v4 XL decoder can drive several types of smoke units directly, the standard v4 decoder and the micro v4 may not be able to fully drive a smoke unit, but you can still provide “smart” or sound driven smoke unit operation if you consider options such as SUSI or relays. Placing magnets or foil systems or other complicated trigger devices is not required if you use the smoke unit capability of ESU decoders. See figures 44, and the function mapping figures in section 8.6 for detail.

8.10.1 ESU Smoke unit (CV's 140,138,139)

Here you can set variables to control ESU and other smoke units. You can set time of operation until smoke unit power off across a range of 1 to 120 (1 to 600 seconds) (CV140). You can trim both fan speed (CV138) and temperature (CV139) to provide smoke density to your choosing using the appropriate slider value. Trim settings default to 128 (100%) across a range of 0 to 255.

8.10.2 Smoke chuffs (CV's 143,141,142)

This segment of the smoke unit set up allows you to control duration of the smoke “puff” relative to trigger distance (CV143), across a range of 0-255 providing up to 1 second duration per trigger event. Minimum (CV141) and maximum (CV142) duration of the smoke pulse is adjustable in a range from 0-255, this provides a maximum smoke pulse duration of 1 second, with minimum and maximum ranging from 0 – 1.04 seconds. These settings allow the smoke pulse to vary with locomotive speed, similar to the way the chuff sound varies with speed.

All smoke settings are direct values which lend themselves to POM programming for easy fine tuning. SUSI interface will provide “intelligent” smoke if your decoder functions cannot meet the current rating for the smoke unit. If using SUSI interface please remember to enable SUSI for the decoder, see section 8.3 “Compatibility” for details.

8.11. Sound settings

Sound settings section allows configuration of several areas that affect the reproduction of sounds stored on the decoder when the sound project is written to the decoder, thus allowing the user to select operation to match the model locomotive. Please refer to figures 45 and 46 as each section is covered.

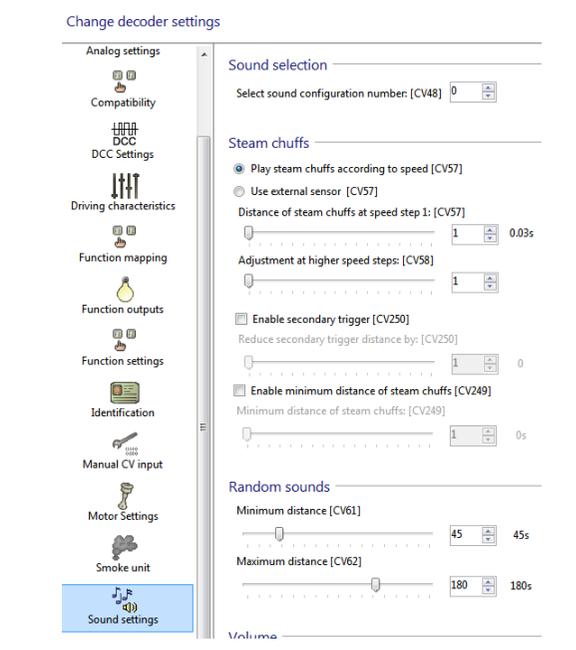


Fig. 45: Sound settings part 1

8.11.1 Sound selection (CV 48)

Version 4 decoders and software allow the use of pre-configured groups of sounds which are kept together in templates that can be placed in a sound project. Examples are whistle and horn packs. Within each group, each specific sound, such as a Nathan-P3 horn, are given an index number when the pack is created, the number range is 1 through 16, with a specific sound assigned to each number. Using the Sound selection configuration number you can select which horn or whistle you prefer to sound when the horn pack is assigned to a function key via function mapping. CV48 can also be written directly, allowing for changes while operating. (Power cycle may be required after change)

8.11.2 Steam chuffs (CV's 57, 58, 250, 249)

Synchronizing steam loco chuff sound to wheel revolution is performed in this section of the programmer. An abbreviated method is listed here, for full detail, please read the decoder manual, chapter 13.4. (Continued next page)

If using an external chuff trigger, such as magnets or a trigger switch, check the option titled “use external sensor”, then you have an option with the slider bar to enter the number of triggers required to make a chuff. E.G. if a magnet was mounted on a gear that required 4 revolutions to provide enough drive wheel progress to require a chuff, then you would put 4 in the slider box; this would mean it would take a total of 16 revolutions of the gear to equate to 4 chuffs per drive wheel revolution.

Before complicating your assembly by devising a trigger method, try the chuff option “Play steam chuffs according to speed”. Select that option, and then set the chuff sync sliders in order to produce the correct automatic chuff sounds. A significant number of modelers who once always used chuff triggers have found automatic chuff timing is just as accurate and significantly easier to set up.

CV57 is used to set the chuff sequence for 1 complete drive wheel revolution at speed step 1 (1 of 28). Time one drive wheel revolution in seconds; calculate the initial value for CV57 by dividing the time by .128 for 4 chuffs per revolution, or .103 for 3 chuffs per revolution. Enter the calculated value (rounded, no decimal) using the slider. Test and adjust until the correct chuff timing is found for speed step 1 of 28.

CV58 is used to set the higher speed chuff timing. Do this step only after you have found the correct timing for speed step 1. Increase loco speed to speed step 4 of 28, observe chuff rate, if chuff rate is now less than 4 chuffs per revolution, decrease the value of CV58 (default is 43 unless it has been overwritten). If chuff rate is too high, increase the value of CV58. Make incremental changes until you get the correct chuff rate at speed step 4. Note: use “read decoder data” to find the current values in the decoder. Note: in practice you will find CV’s 57 and 58 work together in a balance, you may find several settings that will work.

Minimum distance of steam chuffs (CV249) is used to fine tune high speed running of steam sounds, for instance if the chuff sound at high speed sounds like it is run together or “machine gun”, you can adjust the tendency for the sound to become over speed with this setting. To use this, check the option to enable, and put a trimmer value in with the slider. Hint: start at mid point and then adjust up or down as needed.

Hint: it may be quicker overall to use POM for the above, and then return to the programmer and “read decoder data” to capture the values into your project.

Please see the decoder manual, chapter 13.4 for full detail.

Enable secondary trigger check box (CV250) is useful in several instances. If you wish to mimic the sound of an articulated steam engine, such as a 2-8-8-0 or similar, you can check the box to enable secondary trigger, and put a value in with the slider bar to set a cycle time for the sound of the drivers running slightly out of phase with each other. It is also useful if you are modeling a dual engine diesel or a multi-channel steam sound (In this case, the sound project must be built using 2 channels, see the chapter for the sound modeling section in this manual.)

Please refer to figure 46 for the rest of Sound settings information:

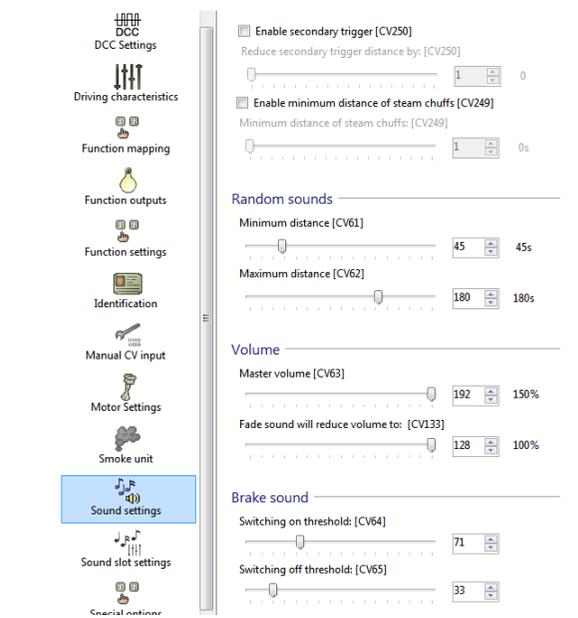


Fig. 46: Sound settings part 2

8.11.3 Random sounds (CV’s 61, 62)

Random sounds are cyclic sounds such as compressor, air pop off, pumps, etc. that are heard from locomotives when standing or running. These sounds do not involve brakes or prime mover sounds. The frequency and duration of these sounds is controlled by the settings established in this section of the programmer.

CV61 is used to set the minimum time between sounding of a random sound, in seconds, using the slider to set the value. CV62 is used to set maximum time between sounding. Random sounds will then occur between the two settings, on a random basis. If the sound project has five random sounds, then all five will be heard at various times, if only two random sounds are available, then of course it will be only the two that are heard.

8.11.4 Volume (CV’s 63, 133)

Master volume setting for the overall decoder sound level is set with this slider (CV63). The master volume can be set across a range of 0 to 192 with the slider, producing a master volume range of 0 to 150%. Individual sound slot volumes can be set also, see section 8.12 for those settings.

Fade sound (CV133) can be used to mimic the sound reduction experienced when entering tunnels, going behind obstacles or other times when you wish to fade the master volume. Magnitude of the sound fadeout is set with the slider across a range of 0 – 128 which produces a fade range of 0 to 100% volume when triggered.

8.11.5 Brake sound (CV’s 64, 65)

Brake sound on and off is set here, using the sliders that change values in CV’s 64 and 65. Brake sound on is triggered by reducing the speed step setting with the speed control (throttle). Therefore, if you wish the brake sound to be more sensitive to speed control change, set the “on” threshold to a higher value. After the brake sound is engaged, off sensitivity is set by the “off” threshold. Set these values as you wish. If using a good momentum setting along with the desired on / off thresholds you can “play” the brakes such that they will sound the same way a prototype locomotive does.

8.12. Sound slot settings

The sound slot configuration view allows you to set four items in regards to each sound slot; volume, minimum sound speed, maximum sound speed, and lock to drive sound. Each of the 27 sound slots can be named (from the Sound section) and have the above settings individually set here. See figure 47 and 48 for detail:

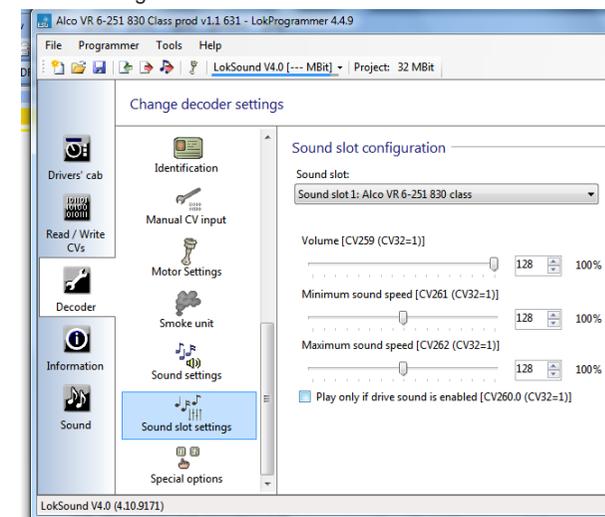


Fig. 47: Sound slot settings A

Change Decoder Settings

Sound slot settings / Special options

8.12.1 Sound slot selection

In order to configure a sound slot, first select the sound slot you wish to set up, refer to figure 48:

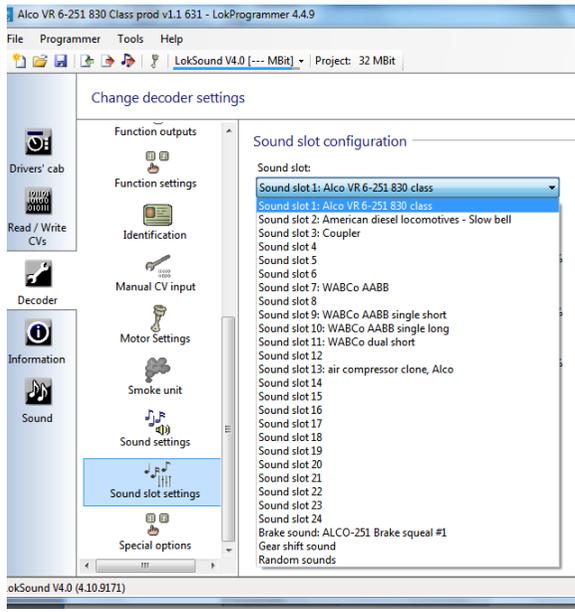


Fig. 48: Sound slot settings B

Click on the dialog under “Sound slot” and the sound slot list will display. Here you can view all the sound slots and the names associated with them. It is best practice when creating a sound project for v4 decoders to name the sound slots in use, if they are named the user can easily map them to function keys and also discern what type of sound each slot will play. Select the sound slot you wish to set up by clicking, and the display will provide the options as shown in figure 47.

8.12.2 Sound slot Configuration

First, set the sound slot volume using the volume slider, range is from 0-128, providing a volume setting of 0 to 100%. 100% represents the maximum volume for the sound slot, as established when the flow chart that plays the individual sounds was drawn. In the same way, set the volume for the other sound slots as desired.

Minimum and Maximum sound speed settings can be set to allow the sound to increase and decrease in perceived speed across the speed range of the locomotive. These two sliders default to a value of 128, providing a setting of 100%, which means if left alone the sound will be as modeled in the flow chart. Adjustment range of both is from 0-255, providing a possible pitch range from 0 to 200%. This option should be used with care as you can create some

sound effects that you did not intend. It could be used with sounds that may vary with prime mover rpm change, such as a compressor. It can also be used to slightly offset one prime mover from another that have the same prime mover sound, to make a consist sound more life like., such as a consist of EMD 567 diesels in which each engine idles slightly higher or lower rpm than the other, and in which top rpm is slightly different. This would be achieved with very minor settings of min / max difference.

The last adjustable item in regards to sound slot configuration is a check option which requires the drive sound (prime mover) be active before the sound slot is allowed to play. This adds realism in that items such as an engine driven compressor obviously cannot operate unless the engine has been started.

It should be noted that the CV's associated with each sound slot are different from the others, along with index values that allow selecting the correct CV's for programming under POM to change values during operation. See the decoder manual for detailed explanation of the CV structure.

8.13. Special options

This section is for setting decoder global settings that affect overall operation of the decoder. Refer to figure 49 and to the decoder manual chapter 9.5 for detail.

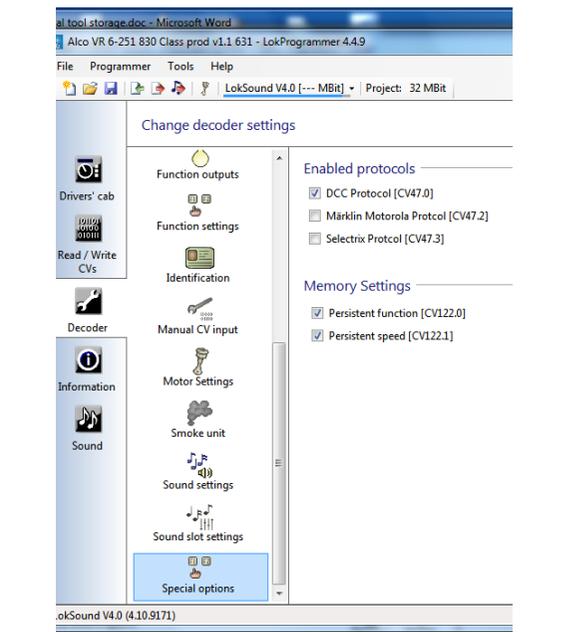


Fig. 49: Special options

8.13.1 Enabled protocols (CV 47)

ESU v4 decoders can operate under several different protocols from

various manufactures. Here we can select protocol based on the layout where you intend to run your locomotives. Place check marks in the appropriate selection boxes in order to match your running environment. Deselect those you do not intend to use.

8.13.2 Memory settings

This option allows you to choose how you wish the decoder to behave after a power interruption, such as a dirty track or system shut down. Checking these options determines if the decoder resumes operation with the previous function key settings and / or with the previous speed. In most cases smoothest operation is found by checking the boxes as in figure 49.

9. Information

The information section consists of two view panes that can provide information for some DCC command stations plus allow user information to be entered as desired. Both views in this section are optional, in that the user can choose to populate them with information or leave them in the default state. These are optional views.

9.1. Functions

Functions information view consists of text and graphics arranged in columns and rows similar to the function mapping section. The difference here is that the user can provide information here as he desires, and it is optional information. If the user employs an ECoS command station, the command station will use this information when the decoder is detected by the command station. Thus it becomes very valuable as this provides a graphical display of function mapping when in use on the layout. See figure 50 for example information.

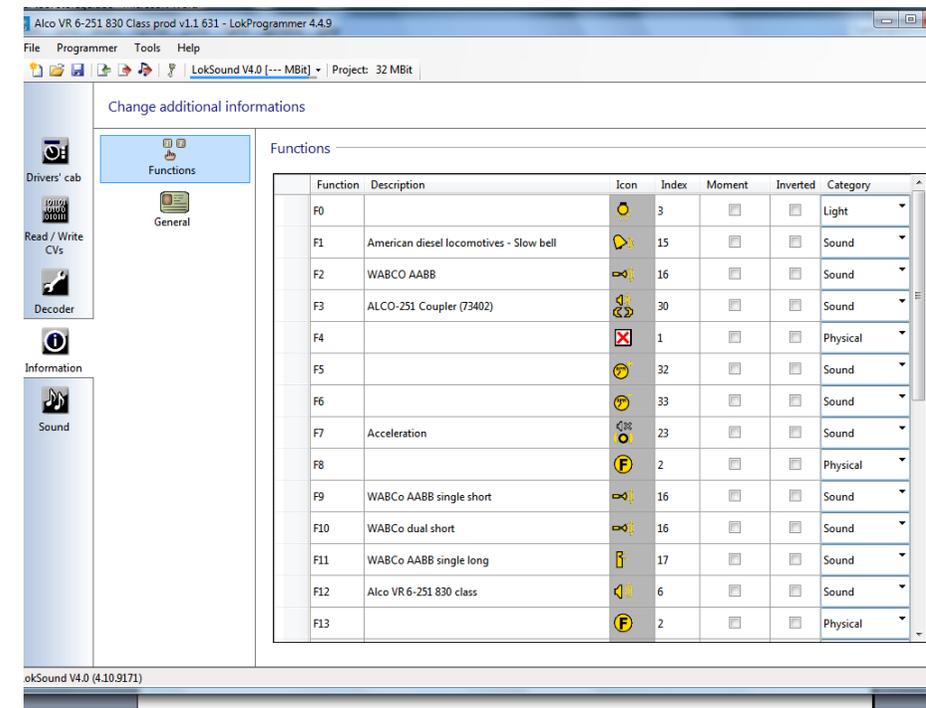


Fig. 50: Function information example

9.2 General

General information may be added on this view; many producers of sound projects also use this area to provide information about the sound project involved. In addition to information about the type project, country of origin, description and version, there is also an area to include a figure or photo. Images used in the view must be 190x40 pixels in order to be uploaded. The same image can also be loaded to some command stations, such as ECoS. See figure 51 for example information.

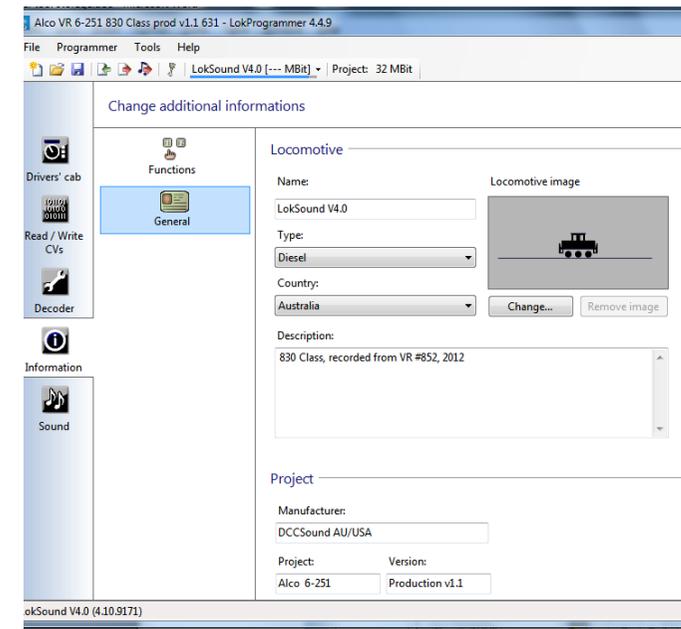


Fig. 51: General information example

10. Sound modeling, adding sounds

Sound modeling for the ESU v4 decoders is significantly different from all other sound decoders available. The core basis of the sound model is a sound flow chart, which is completely free form in nature. The model flow begins at the left hand of the chart, and progresses on from there. There are drawing objects that you can place on the chart, they are connected by lines. In the objects you link the object to the sound, and on the lines logical conditions are placed that tell the flow when to branch to another object and play the next sound. It is a very elegant modeling system and once you learn the objects and understand the conditions, you can truly model any sound you can imagine. Fortunately you do not have to know everything to begin, and you can learn as you go.

This section of the manual will explain the objects in the models and the methods to use them. Sound flow charts vary from very simple to very complex, there is a library of sound included in the software, and there are many sound projects available for download, so there is no need to have a recording studio or a recording device, you can start at any level you wish, mix and match, and learn as you go.

10.1. Sound section overview (opening view page)

After you click the "sound" icon the sound page opens a tab called "Sound project overview", it consists of 4 view panes, at the top there is a pane titled "Sound type" and below that "Available sound slots", below those on the left there is a pane that displays your computer file structure, and to the right of that there is a pane with "File name" at it's top. If you have a sound project open you will see the sound files that are in the sound project you have open. Above those panes you will see a grayed out menu bar with names that begins with "Sound slot simulation" and several other items. Each of the four view panes can be adjusted if you wish to maximize a section of the page, and sliders will be available when information is hidden so it can be scrolled to display.

If you have a pre-loaded decoder, (Select or Standard) the sound project that goes with the decoder is required to be open in order to write sound data to the decoder. However; you can only see the sound information the goes with your decoder when you have a v4.0 decoder with an unprotected sound project. Select decoders do not allow alteration of the base file, therefore the sound icon does not display. Protected v4 files also do not allow the display of the sound icon.

In the next few pages each view pane will be explored using a "drill down" approach such that every pane is covered and following that the modeling environment will be explored in detail.

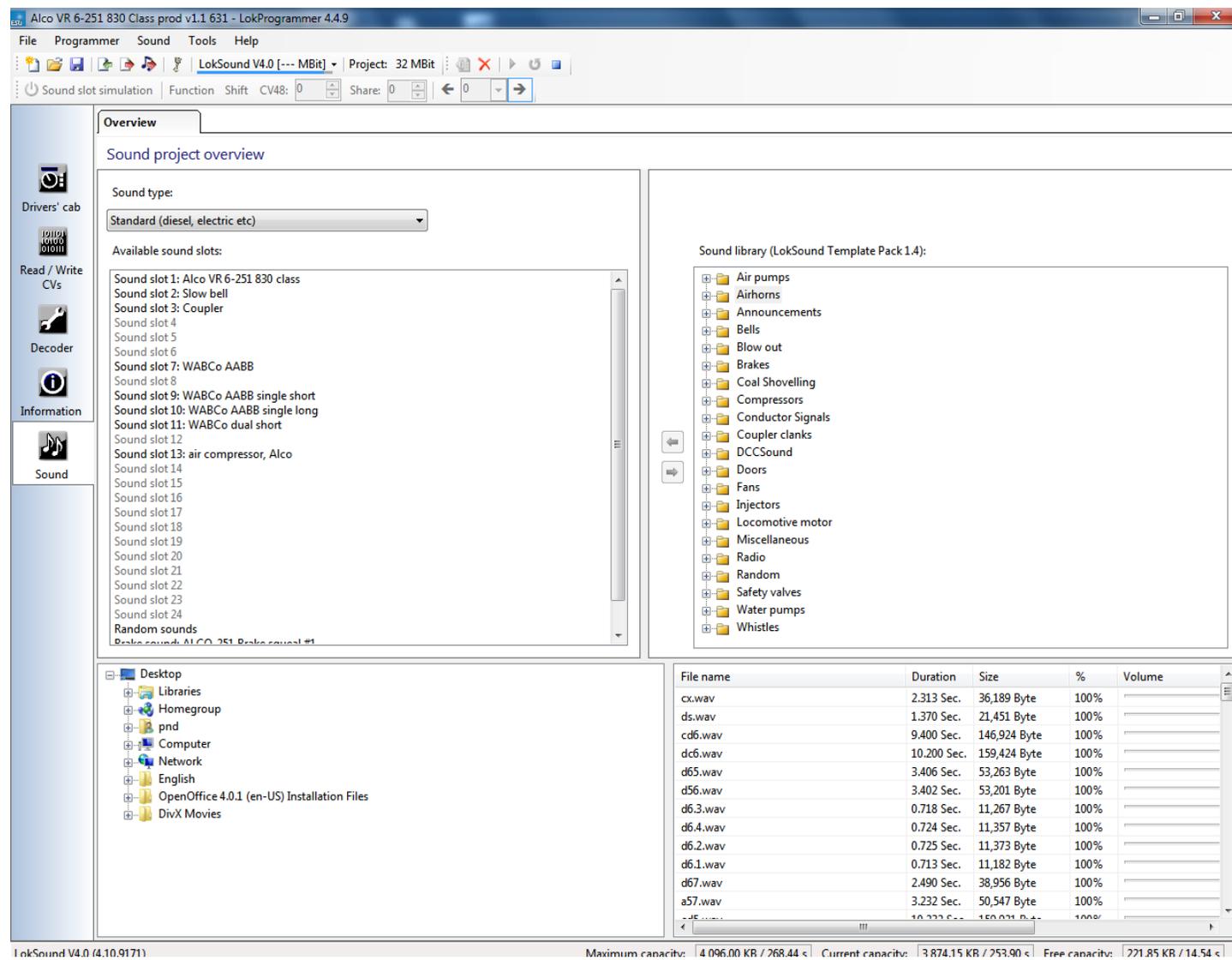


Fig. 51: Sound project overview, with a sound project open

10.1.1 Sound type

Here the type of sound project is designated by clicking the sound type dialog and choosing an option for the project. See figure 52:

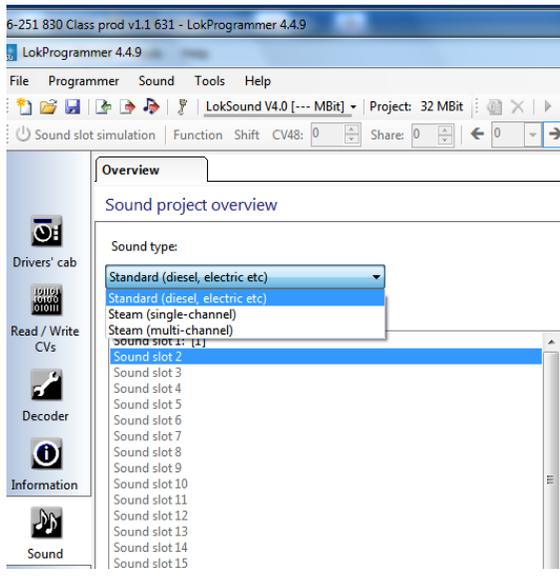


Fig. 52: Choose sound type for sound project

Sound type options are Standard (diesel, electric, etc.) or Steam. Within the Steam type there is an option for single channel or multi-channel. V4 decoders have the capability to support multi-channel prime mover sound, providing the option to model articulated steam locomotives and multi-engine diesel. Differences between sound type between “standard” and “steam” are few; the process of creating a prime mover sound flow is very similar for both types. When “steam” is selected an additional drawing object is available, the steam container. Containers and other differences between sound types will be covered in detail later. If you wish to model a dual engine diesel using multi-channel sound, then select “steam (multi channel)” and simply do not use the steam chuff container. A final option to create a new project is available from the file menu, to convert a v3.5 decoder project to v4.0. To do this, use the “import” option as covered in section 5.2.

10.1.2 Available sound slots

The rest of the upper left hand pane within the overview tab displays the available sound slots. The v4 decoder architecture supports 27 sound slots, most of them are numbered and near the bottom of the view they are named, the named sound slots are restricted to certain sound characteristics, as they are governed by decoder settings or they are assigned to specific

duties within the decoder sound flow chart. The named sound slots are “Random sounds”, “Brake sound:”, “Gear shift sound” and; if the sound type is steam, “Hiss and start up”. (Hiss and start up will display in place of sound slot #24). See figure 53 for example display.

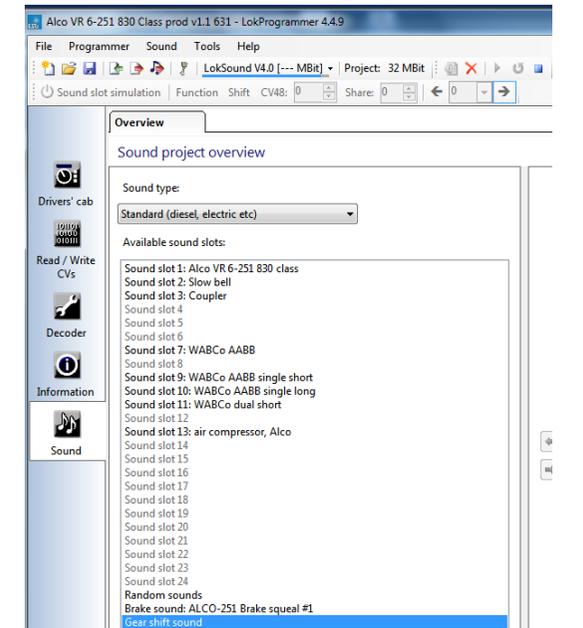


Fig. 53: Sample Available sound slots display

Note that in figure 53 you can see all the sound slots, some are grayed out and some are in dark text. The dark text indicates these sound slots have sounds associated with them, and if they are expanded the associated sound flow chart and sounds attached will be displayed. This will be covered in detail after the overview section is complete. The sound slots that are grayed out are available to have sound associated with them. The names displayed with the sound slots are optional, but it is a best practice to name the sound slots with the contents, such as prime mover type, or type of horn or whistle. This allows the user of the sounds to see the name when they are being mapped to functions or being configured in other ways in decoder settings as covered in the early sections of this manual. If the sound slots are not named, the only information that displays when performing function mapping or sound slot settings is the number of the sound slot, with no indication if sound is present or what is contained with the sound.

10.1.3 Sound library (LokSound Template Pack n.n)

ESU provides a library of sounds in a compressed format as part of the LokSound programmer software. Both the programmer software and the template packs are updated on a regular basis. It is good practice to keep your software and templates up to date. Template version number is indicated by the numbers at the end of the name, in place of

the “n.n”, version 1.4 is shown in figure 54 for as an example:

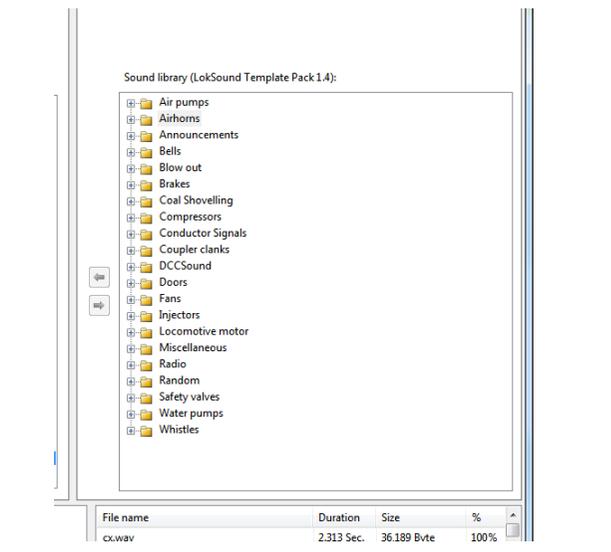


Fig. 54: Overview, template pack

Templates are easily used as a method for creating individualized sound projects or as a sound source in modifying projects you have downloaded from ESU download site. They also make excellent teaching or learning tools as you can see how different things you want to do can be accomplished in building custom sounds. You can also add your own creations to the template pack, in your own folders or in supplied folders. It is recommended to use your own folders, and then when the template pack is updated, it will not overwrite any of your own data. Templates are organized by folder as shown above, and folder titles are quite descriptive. The folder titled “Locomotive motor” is where full sound flow charts are stored, complete with associated sounds.

10.1.4 File name

The view in lower right hand holds the sound files that are part of the project file that is currently open. All the files displayed within this view are contained in the sound project that has been opened, but does not necessarily display the sound files that are loaded in the decoder, or will be loaded to the decoder when the sound is written to the decoder. If you see files in the view pane that are grayed out, then those files are not used in any of the sound project flow charts, but they do take up space in the available space calculation, so it is good practice to keep track of how many unused files you have with the project space. There are file commands and other tools to assist in sound file management that will be covered. See figure 55 for sound file overview:

Sound modeling

Sound project overview cont.

File name	Duration	Size	%	Volume
WABCo AAB8 single short.wav	1.057 Sec.	16,554 Byte	100%	
WABCo AAB8 single long.wav	2.317 Sec.	36,246 Byte	100%	
WABCo AAB8 dual short.wav	2.021 Sec.	31,626 Byte	100%	
comp init.wav	1.939 Sec.	30,343 Byte	100%	
Comp clone exit.wav	1.004 Sec.	15,733 Byte	100%	
Comp clone body.wav	2.428 Sec.	37,981 Byte	100%	
brake1_init.wav	1.312 Sec.	20,540 Byte	100%	
brake1_loop-part1.wav	0.225 Sec.	3,558 Byte	100%	
brake1_loop-part2.wav	0.258 Sec.	4,077 Byte	100%	
brake1_loop-part3.wav	0.095 Sec.	1,528 Byte	100%	
brake1_loop-part4.wav	0.366 Sec.	5,761 Byte	100%	
brake1_subexit1.wav	0.176 Sec.	2,794 Byte	100%	
brake1_subexit2.wav	0.204 Sec.	3,230 Byte	100%	
brake1_subexit3.wav	0.133 Sec.	2,123 Byte	100%	
brake1_subexit4.wav	0.239 Sec.	3,786 Byte	100%	
brake1_exit.wav	0.613 Sec.	9,623 Byte	100%	
Silence (2) (3) (4).wav	0.075 Sec.	1,222 Byte	100%	

Fig. 55: Sound project files, overview

The sound project files pane displays both files that are used in sound project flow charts, and also those not used (grayed out). It may be these have yet to be assigned, or are in place as options. Along with the file name the view pane also displays the duration of sound play for each file, the file size in Bytes, and the volume setting. Out of view in the figure is a volume slider that allows individual adjustment of each sound file. The volume defaults to 100%. When a sound file is placed in the sound project it will be converted to meet the requirements of the decoder architecture. The conversion utility used by the v4 decoder is by far the best way to convert recorded files from an outside source; it will give you the least amount of noise and the best fidelity. Recordings can be made and edited at CD quality, (44100 kHz / 16bit, mono or stereo) and the import utility will convert files to fit the decoder. The volume setting at the default value will play the sound at the volume (sound magnitude) that it was recorded at or the edited magnitude. Therefore, it is possible to drive the decoder sound amplifier / speaker into distortion when it is played if the individual sound file volume is set too high. Volume setting range is 0 to 200%. This is the first place within the decoder structure that sound volume can be set or changed, more will be covered later. This setting establishes the sound base for the file when played by the decoder.

At the bottom of the view pane are located various capacity values. Maximum capacity is the total amount of sound play time the decoder can support, 4096Kb / 268.44 seconds. Current capacity displays the amount of capacity used by all the files in the file list, both used in flowcharts and unused (grayed out) files. Free capacity displays how much space in Kb and seconds that is available for more files in the project.

10.1.5 File browser

The last section of the overview covers the file browser and methods to bring sound files into the sound project. The lower left hand pane of the overview tab displays the file browser similar to the display seen when opening a file browser within Windows, and it operates in the same way.

See figures 56, 57 and 58 as file browser examples:

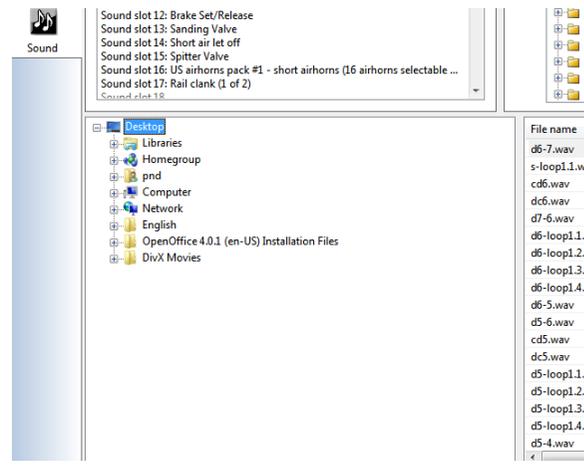


Fig. 56: Sound file browser, overview

Navigating the file browser to find sound files is straightforward, use it the same way you would use a windows file browser. The primary difference between this sound file browser and a Windows file browser is the sound file browser will only display sound files that meet specifications for import into the LokProgrammer software.

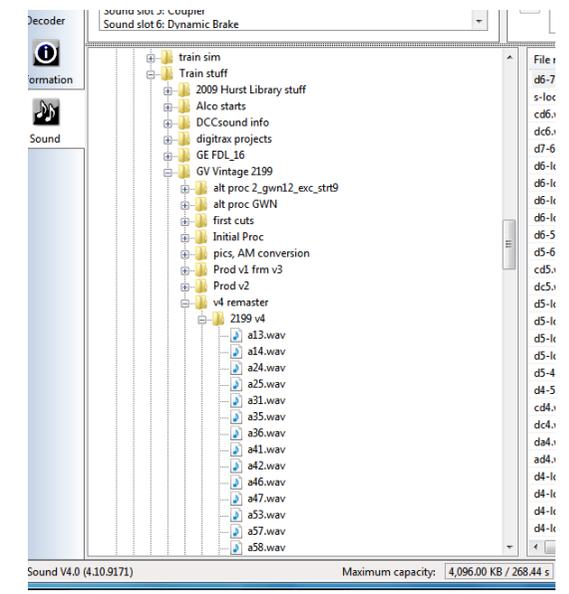


Fig. 57: Sound file drilldown, overview

Another option for browsing sound files for possible import into sound projects is to use existing sound project files, options are v4 sound projects and v3.5 sound projects produced with LokProgrammer software version 2.7.9 and higher. Software versions previous to 2.7.9 do not meet encoding requirements and will not display; those files must be opened first in 2.7.9 and then saved, after that they will display in the file browser.

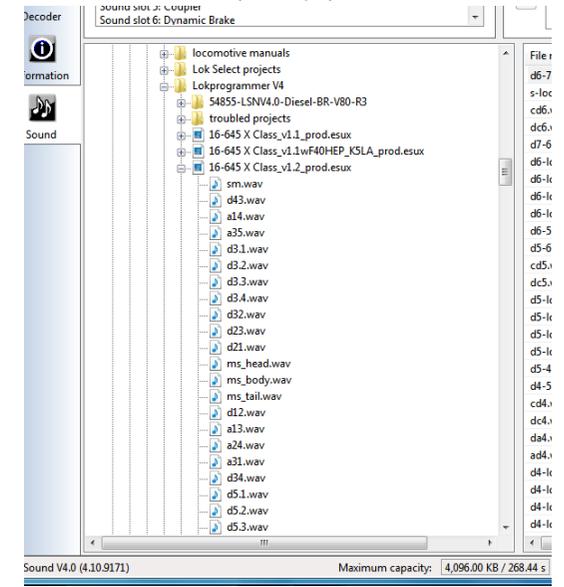


Fig. 58: Sound files in existing project, overview

Sound files must be in .wav format and be at CD quality or lower in order to display. Version 3.5 sound files from projects will be 15625Hz 8 bit and will not provide the same quality as a CD quality file when imported into a v4 sound project.

File name structure:

ESU has developed a file naming structure which is generally in use in current v4 projects. It is not a mandatory structure, but understanding it can be helpful in being able to read file content. The structure is alpha/numeric, alpha depicts a sound type, such as drive or coast, number depicts speed range, as in, d1 equates to drive 1, d8 is drive 8, 2 numbers together indicate an rpm range, such as a35, rpm notch change from n3 to n5. a – acceleration sounds (a1, a2, a35)
d – drive sound (d1, d2, d12)
c, cx, cs – coasting / coast to standing
s – standing (idle, hiss)
sd – standing to drive
dc – drive to coast
cd – coast to drive
ms – mute to standing
sm – standing to mute
Also seen are names such as “init” “loop” “exit” “head”, “body”, “tail” and numeric sub sounds such as d1.1, d1.2, d1.3, ad, da, etc. These will be covered in more detail in subsequent sections.

10.2. Sound files

10.2.1 Adding sound files

If you intend to modify sound information on an ESU sound decoder, it is required that you have access to the sound information. Sound information cannot be read from the decoder, it can only be written to the decoder. Performing "Read decoder data" will populate the LokProgrammer software with only CV information as covered under the sections of the manual previous to section 10. Therefore if you have a preprogrammed decoder and you wish to modify the sounds on that decoder, you must have access to the sound project that matches the sounds on the decoder. One of the first methods to add sound information to the programmer software is to open the sound project for the decoder in question, or open another sound project if you wish to replace the current sound within the decoder. Opening a sound project is the most often used way to add sound information to the programmer software, as shown in figure 59 and 60.

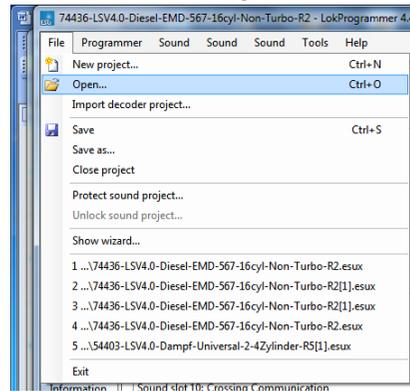


Fig. 58: File menu, Open

Figure 58 displays using the file menu, "Open" command which invokes the file navigation screen, a standard Windows file open command. The default file path is set during initial software install, and can be set to your liking using the top menu item "tools" > "Program settings" dialog. (See Chapter 5.3, and figure 17)

10.2.2 Open sound project

Opening a sound project will populate the sound section with all the sound files in the project, and also populate all decoder data settings as they were when the project was saved. If you already have the decoder set up as you wish it in regards to function mapping, address, momentum, etc. there is a way to avoid having to redo all the decoder set up. Simply open the project as above, and then perform "Read decoder data" from the main menu. You will be asked by a dialog if you want to accept decoder values for the current project. Say yes and your decoder data will be written instead of the default values in the project. Of course if you have

previously saved the project with all the values as you wish them you can just open the project. This case is only needed if you did not have the values saved and you do not wish to reconfigure the decoder. (Figure 60)

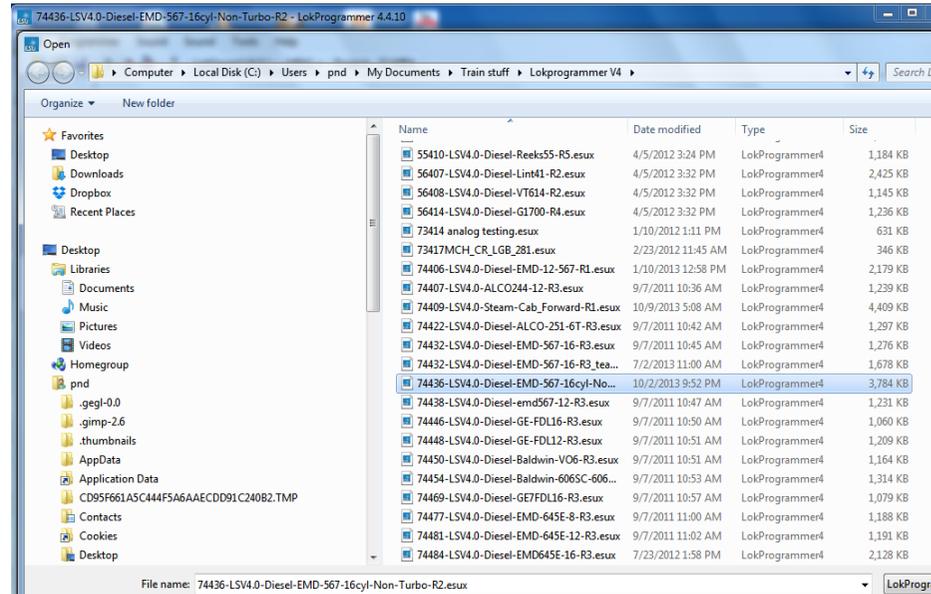


Fig. 59: File navigation

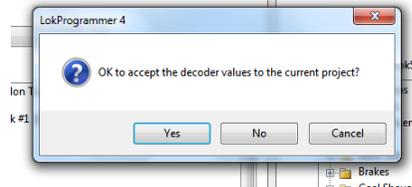


Fig. 60: Recover CV data

Other options for project opening is of course to select a file from the file history numbers as seen in the lower portion of figure 58, and the option to "Import decoder project", seen under the "Open" command. Import decoder project is used if you have a version 3.5 project file and you wish to convert it for use in v4 decoders. In order for a v3.5 project to be imported it must have been saved by software version 2.7.9 or later.

10.2.3 New project

Using this option will create a new sound project, with no sound in place. After selecting this option you will be presented a decoder list, and you should select the decoder type you intend to use for the project. (Figure 61) If you use this option you should be prepared to populate the sound information using other methods, which are: adding sound files by template, adding discrete sound files one by one, and adding sound files in groups. These three methods can also be used with an existing sound project, for instance if you wished to change a brake sound, or add a new horn or whistle.

You may also want to take on a task such as combining 2 existing projects to make a unique project, such as a turbo driven steam sound, or unique compound steam.

This is possible by using sound slots 1 and 2, with a different project as the base for each portion. Then by changing selected portions of each sound project you can make the 2 projects run together as a unique project. The manual cannot possibly demonstrate all the possibilities that are available; this is just a starting point for the imagination to build on.

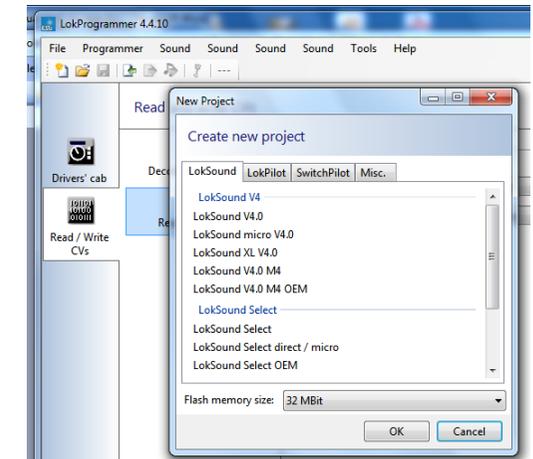


Fig. 61: New project

Sound modeling

Using sound templates

10.2.3 Adding sound by template (Sound library)

Sound can be added to existing projects or new projects via the templates that are provided with the LokProgrammer software, called the sound library. The templates are installed either by the update wizard or by downloading them and running the install utility. Templates are used by selecting them from the template list folders, and assigning them to a sound slot. Selecting a template and a sound slot can be done in either order. You will notice the sound slot and the template become highlighted as they are selected, and when you have both selected, a transfer arrow between the page views turns blue, you can now make the assignment by clicking the arrow.

Now figure 64 has the file name populated with files, and sound slots 1, 10, Random, and Brake all are no longer grayed out. The sound slots with sound attached are also named. There is a sound flow chart attached to each of the sound slots, the flow charts provide the sound model that the decoder plays when each sound slot is active.

Using the method displayed in the 3 figures here, a full sound project can be assembled using sound from the ESU sound library. This is not custom sound, but it is certainly customizing your sound project by assembling your project to your liking.

You can also make your own templates of custom or customized flowcharts and sounds, simply select a sound slot, and a folder destination and the right to left transfer arrow will highlight blue.

Hint: You can make your own folder to hold your templates, to make your folder in the templates directory, found by using main menu command "Tools > Program settings > General settings" the path to the template directory will be displayed there. To the right of the template directory click the button displaying 3 buttons, in the dialog that opens click "make new folder".

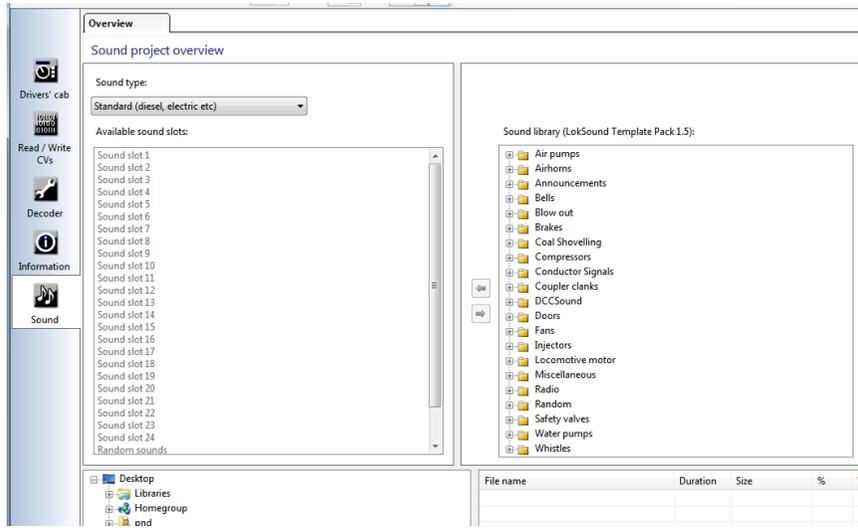


Fig. 62: Add sound template, initial view

Figure 62 displays the initial project overview tab just after using "New project", note there are no entries in the file name view, all sound slots are grayed out, the transfer arrows are gray, and template folders are visible. The next action is to select a sound slot for the template and the template desired. We need a drive sound, so that is what will be initially selected.

Note figure 63, the file name area is still unpopulated, sound slot 1 is selected as that is where drive sounds are placed, the EMD 16-567 is selected as the drive sound, and the right to left transfer arrow is now blue. Clicking the transfer arrow completes the sound slot assignment and populates the flowchart and the file name view. Using the same method brake sounds and random sounds will be added to the drive sound for display in figure 64.

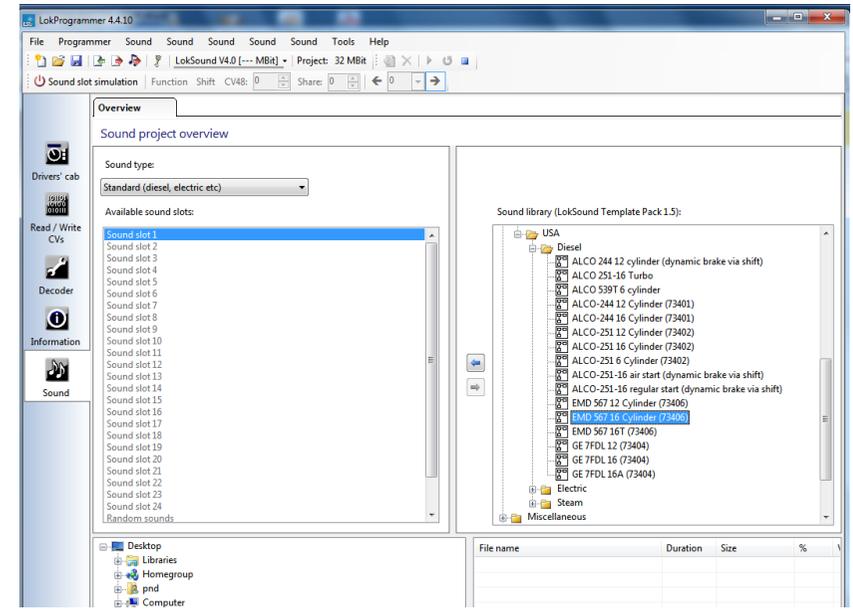


Fig. 63: sound template, sound slot selected

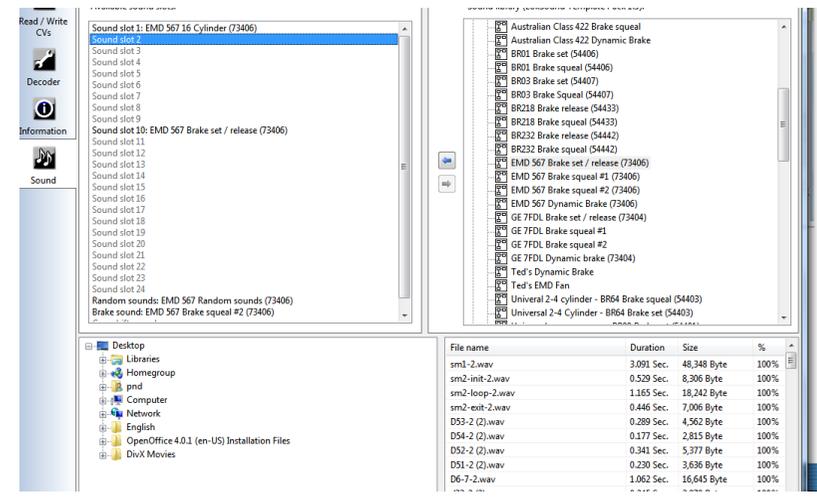


Fig. 64: New project populated with sound from templates

The next logical step may be adding your own custom sounds, or some sounds you have found in another sound library. That will be shown in the next section. This project will be used for all further overview screen shots so the sequences can be followed.

Sound modeling

Adding other sounds

10.2.4 Adding sound files (drag and drop)

Adding other sounds such as your own recordings or sounds from another sound library is very similar to using a sound library template, except in this case there is no flow chart to hold the sounds for play. The files that are added this way will display in the file name pane, lower right hand side, and they will be grayed out, indicating they are not yet in use in any flow chart. The only exception to this is if you add a file with the same name as a file already in use in the sound project, in this case a dialog will display asking if you wish to overwrite the existing file. If you respond with yes, the existing file will be written over and the new file will be linked in all flowcharts where the previous file was used. If you respond with no, the new file will be added at the bottom of the file list, grayed out, with a number added to the filename, such as "d1.1.wav (2)". This can very useful if you are replacing an entire flow chart, if you used the same filenames as before, and if you say yes to the prompt, then all the new files will overwrite the old and the flowchart will update with the new files automatically.

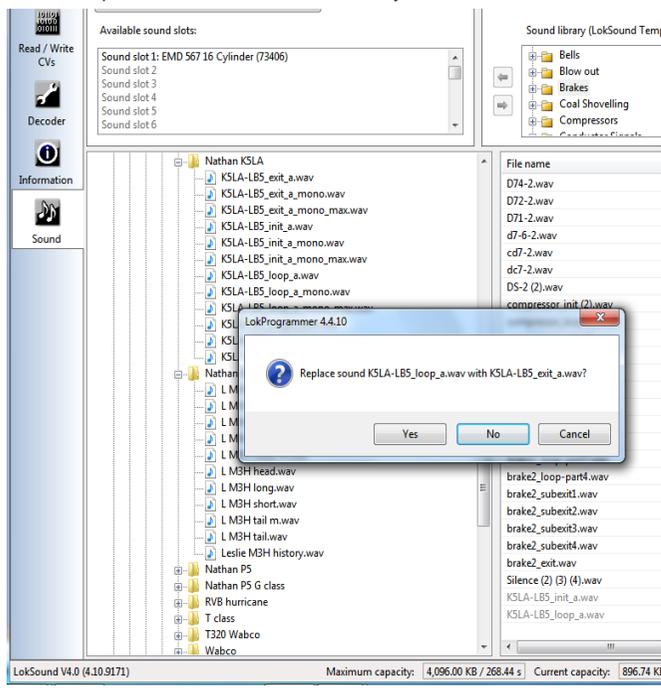


Fig. 65: Adding sound files, individually

Figure 65 displays file navigation to a specific horn type in lower left pane of the view, and the result of adding 3 files to the file name view in the lower right pane. Note there are 2 files grayed out, these have just been added, and the 3rd file is waiting on a dialog response. The dialog is triggered because the cursor was over an

existing file when dropped, so it is appropriate to ask if the file should be overwritten or not. In this case the response will be no, and the file being added will show up at the bottom of the list, grayed out because it is not linked to any flow chart.

The above method of adding files is on an individual file by file basis, discretely adding files one at a time. If you have many files to add at a single time, there is a quicker method, adding files by group. In this method you select a range of files using a different menu and then the files are added sequentially until they are all in the file. In each case, if the file being added is a duplicate file, the dialog shown in figure 65 will invoke each time, allowing you to choose to overwrite, or add to the file list by responding "no".

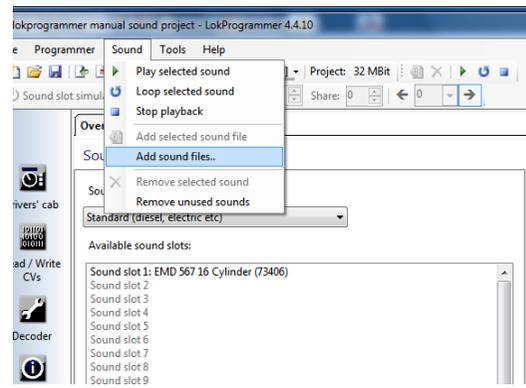


Fig. 66: Adding sound files, by group, part 1

Adding files by group is a dual step process; first you access the "Sound" menu as shown in figure 66. When you click the highlighted option "Add sound files..." the menu will call a file navigation dialog with which you can navigate to and select as many files as you wish. When you click "Open" at the bottom right of the dialog the files will be converted by the conversion utility and placed in the file name list. If the files already exist in the project, then the dialog shown in figure 65 will display once for each file selected that is a duplicate filename. See figure 67 for the file selection dialog. From software version 4.4.11 sound can be added using drag and drop from any windows file navigation view.

Up to this point in the LokProgrammer manual we have dealt with only general information and CV configuration in chapters 1 through 9; and with adding sound files in chapter 10. Chapter 11 will cover sound flow chart manipulation and creation; this is where true sound modeling begins.

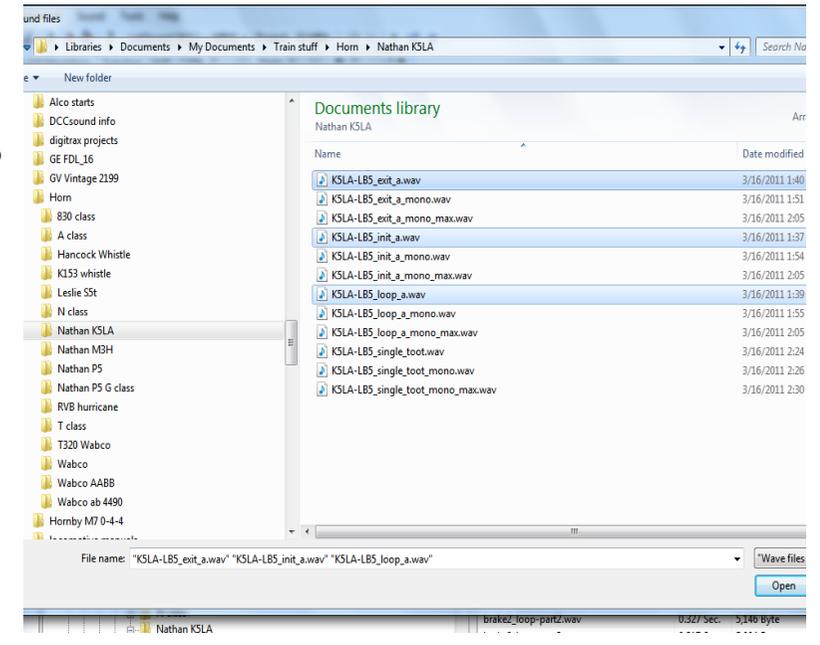


Fig. 67: Adding sound files, by group, part 2

Hint: All the flows, sounds slots, and other actions shown in Chapter 11 can be duplicated on your computer if you want to follow along, the software is free. You can teach yourself a lot in very little time, join in!

11. Sound modeling, flow charts

This section covers sound flow chart basics, how to read a sound flow, how to manipulate existing sound flows and how to create your own, or add your own features to an existing flow chart. You must have sound files before you can begin to understand the flows, which is why this section comes near the end of the manual. The modeling system described below will demonstrate the extreme flexibility of this unique system, its elegance, and how it sets a new standard in DCC sound design.

11.1. Sound flow chart basics

Once the file name pane is populated with sound files, flow charts can be created or manipulated. Referencing figure 64, there is a sound project with 4 of the 27 sound slots populated with sound; these were placed there using templates from the library. The first sound slot examined will be a simple brake set release sound, which is assigned to sound slot 10 in figure 64, in the figure below the sound slot is highlighted, but not yet opened; it has been “right clicked” to reveal the underlying menu options.

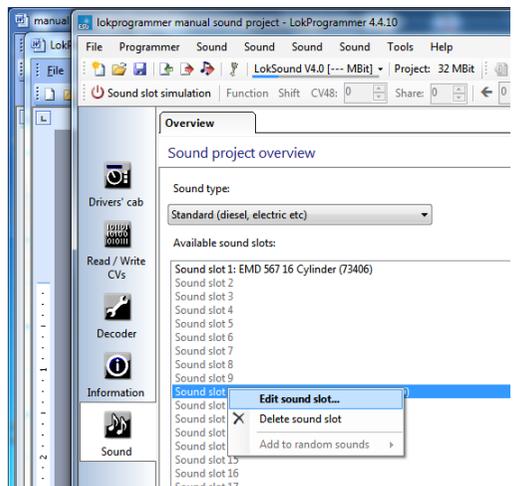


Fig. 68: Sound slot options

As shown in figure 68, there are three options displayed, Edit, Delete, and “Add to random sounds”. Clicking “Edit” opens the sound slot in a new view pane and displays the matching flow chart. Double clicking on the sound slot will perform the same action. Clicking “Delete” will remove the flow chart from the sound slot, the sounds associated with the flow chart remain in the file list, and the sound slot becomes grayed out. If “Add to random...” is displayed dark, clicking that option will move the flow chart to the

random sounds sound slot. In figure 69 the “Edit” action has been completed and the flow chart view window opens.

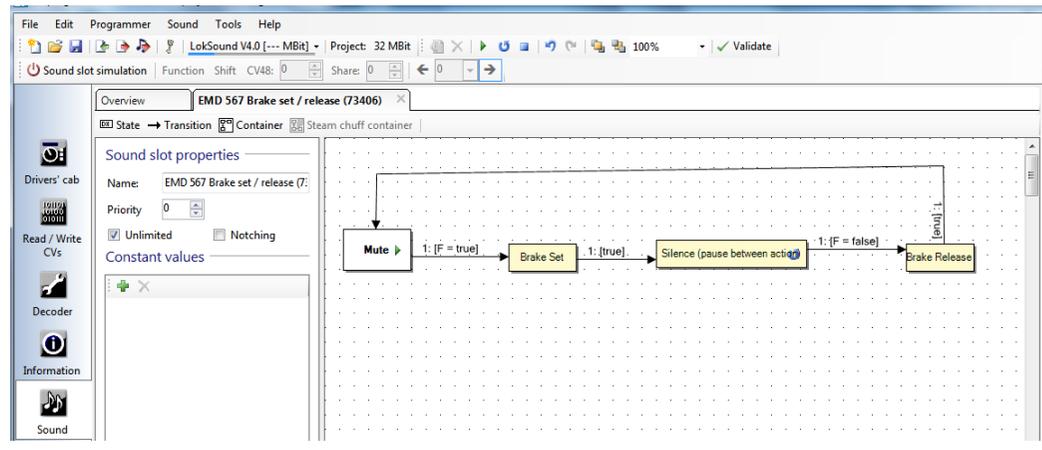


Fig. 69: Sound flow chart

Figure 69 displays the initial opening view of any flow chart. All flow charts are displayed in the same way.

Things to note in this view are; at the top you can see the tab for “Overview”, clicking this will return to the previous display, but the tab for this flow chart remains open, showing as a tab until closed. Below the tab title there are 4 items, “State”, “Transition” (with a short line in front of the word), “Container”, and “Steam Chuff Container” (grayed out). These are the drawing objects which can be placed on the flow chart drawing pane, covered later. Next seen is the sound slot name, here is where a name can be given to a sound slot, a descriptive name provides information when mapping function keys and setting sound slot values using the decoder settings icon. Below the name there is an option to provide a priority to the sound slot. The decoder has 8 sound channels and can play 8 sounds at once. Assigning priorities to the sound slots determines which slots have priority in case a channel conflict comes up during operation, if this happens, the sound slot with the highest priority will play and the other will be interrupted. Note the check mark under priority, “Unlimited”. This means that if the sound slot was playing and was interrupted, if unlimited is checked, the sound will resume when it has room in the priority list, if unlimited is not checked, then the function key the sound is mapped to will have to be cycled to invoke play again. This is a very important setting. “Notching” is checked if the flow chart has notches (e.g. most US diesels have 8 notches for run RPM settings). If Notching is checked, then functions “Notch up” and “Notch down” will work if they are mapped to function keys. “Constant values” is a highly important optional use item. If you have a complicated flow chart, such as a diesel engine sound flow with 8 notches, acceleration and coast, and perhaps new age auto-start and 2 stage idle, there will be a lot of drawing objects in the flow. If you have to make a change to something, perhaps the drive step schedule, you could be faced with making many changes at one time, with the possibility of error damaging a fine operating chart. “Constant values” allow you to define the items in the flow chart most prone to change, for example, the speed values for d1, d2, d3, etc. Acceleration values can be defined. In fact, you can define any value that tends to be changed with developing a new flow, or modifying an existing flow. Using this method, you can make value changes in the table, and the sound flow instantly responds to the new values, without the possibility of missing an item or making a mistake when working with drawing objects. The time saved using this method can literally be hours if you have to change a major sound flow chart. This will be clearly seen in subsequent flow charts. In the flow chart drawing area a simple flow is displayed above, there are 3 states, 4 transitions, and a mute box. All flows begin with a mute box, unless the flow is a drill down flow inside a container. The names of the objects make up a small language, if you want to “talk” sound, it is best to learn the language.

TRANSITION – looks like a line, allows the flow to branch along the transition, if the condition on the transition equates to true, the flow branches
STATE – looks like a box, is yellow when sound is attached, this sound plays when the transition that points to it becomes true
CONTAINER – looks like a box, it is light blue, it holds other containers and states, acts similar to a state, becomes active when the transition pointing to it becomes true, the sound flow “dives” into the container and plays until the end, and then it comes back out. (Drill down)
STEAM CONTAINER – similar in all ways to a container, it is dark gray, active only in steam projects, has special properties covered later

11.2. Sound slot and State properties

Figure 70 displays the same flow as figure 69 so we can read the flow.

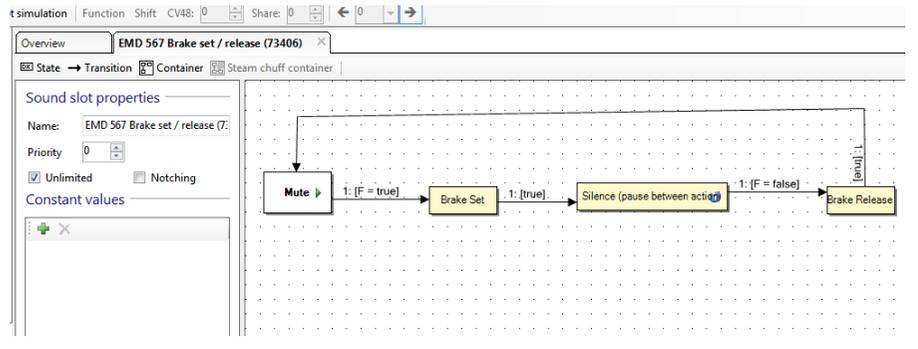


Fig. 70: Brake set / release flow chart

Reading the Brake set and release flow chart, figure 70, note that the flow begins at “Mute”, meaning there will be no sound until the condition on the transition is met. This is a simple “if Function = true” then the “Brake Set” sound is allowed to play. What does it mean by Function? It means that whatever function key has this sound slot mapped to it will play the brake set sound when it is enabled. We see that after the brake set is played, then the flow goes into a loop, and remains there until the function key is turned off. That event equates to “Function = false”, which is the condition to exit from the loop. When that occurs then the sound “Brake Release” is played and the sound flow moves back to Mute. Note that because all the states are descriptively named, we do not have to click on every state to see what sound is linked to it, we can infer that by simply reading the state name. This is a best practice. The conditions on the transition can be read easily, conditions will be covered in detail later on.

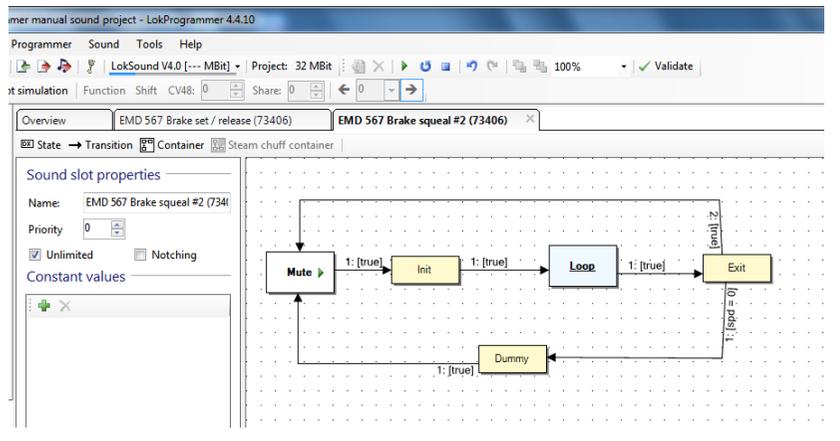


Fig. 71: Flow chart, Brake squeal

Figure 71 is a bit more complicated than the brake set flow chart, and it has a container in it the flow.

Reading this flow is similar to the previous, except notice there is nothing visible at the beginning to start the flow, therefore it will sound as soon as the decoder has power unless something else controls its activity. If we go back to the overview tab (figure 64) we find that this flow chart is attached to sound slot “Brake sound”. This is one of the pre-defined sound slots; it is controlled by the decoder sound settings that control when the brake sound is triggered. (Figure 46, Chapter 8.11.2) It is logical then that we do not want another trigger involved here, such as a function key, that is why there is no condition here. The first state in the brake squeal flow is named “Init”. This is old v3.5 terminology in which a single sound is broken down into 3 parts, init, loop, and exit. This allows a clean repeating sound to play in a loop such that the sound can play over a long period. Init plays once, loop could play infinitely or once, depending on the trigger event, and then when loop is finished, exit plays once. V4.0 terminology for the same sequence is “Head”, “Body”, “Tail”, either is acceptable, and depends on who draws the sequence.

Notice that “Exit” has 2 ways to quit the sound, this is an example of an OR gate, sound flow can travel either path depending on which becomes true first. One path reads simply “True”, meaning, if there is no trigger, it can exit. In this case the control is in the hands of the decoder brake settings (Figure 46, Chapter 8.11.2). If the locomotive stops but the brake trigger (based on the speed control) is still valid; in the pre v4.0 days the brake sound would still play, not realistic. By using the OR gate as in figure 71, the other way for the flow to leave the exit state is in the condition on the bottom transition, “Speed = 0”. That is the case if the locomotive stops, the decoder knows that, the condition becomes true, and the sound path can leave exit and pass on to state named “Dummy”. This allows the brake sound to be silenced either when the trigger event (speed control movement) ends, OR the locomotive stops. Dummy is not a very descriptive name; a check of the state reveals it will play silence until the Brake trigger is no longer valid. The figure below, 72, displays the state properties on state “Dummy”, this screen displays whenever you single click a state. It will be used to go through the state properties view:

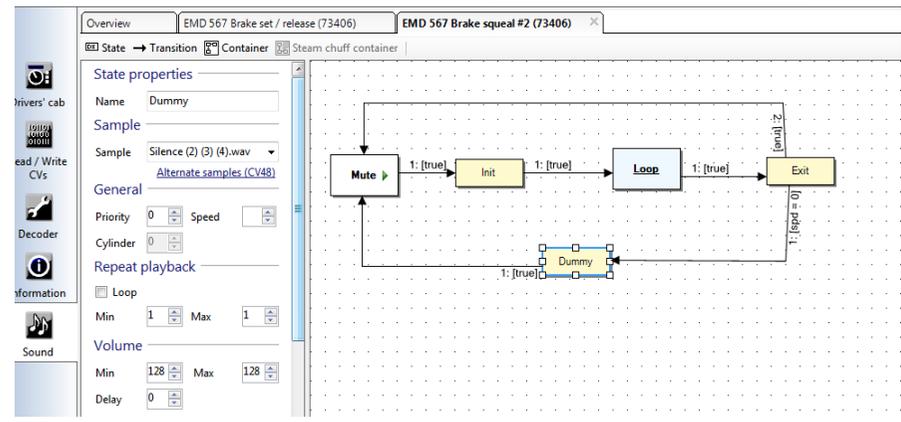


Fig. 72: Flow chart, Brake squeal, State properties 1

The first item of note in the State properties view, is that the state selected is named “Dummy”, just below “Sample” we can see that it plays silence.(2)(3)(4) wav, the numbers mean there are duplicate silence sound files in the file list, the numbers are appended by the file system, it will not allow duplicate filenames. There is an expansion arrow on the right of the Sample field which will display a drop down that shows all files in the file list, allowing selection of the sound you wish to play when the flow reaches the state you have selected. Another way to link a sound file to the state is to navigate to the file you want using the file list (lower right hand pane) and click/drag the file to the state desired, it will then become the file linked to the state, and either way has the same result. The next area of state properties is “General”: Priority can be set for the state, providing a “sub priority” if desired. 0 is the default, which means the sound slot priority from the Sound slot priorities carries down to the state, by setting a state

Sound modeling

State properties

priority you could prioritize state sound playing. If desired. Speed is an option setting located on the right of priority, the value range for speed is 0 to 255. If speed is set in this option it will not allow the state to play unless decoder speed is higher than the speed setting. If left blank (default) the setting does not affect state play in any way. Below priority is "Cylinder", this option allows the cylinder to be designated for steam projects. Note option grayed out because the project type is diesel. "Repeat playback" is the next set of state options, if loop is checked, then you set the number of playbacks in the loop using the min and max settings. Loop playback will be random between these 2 settings. In the "Volume" settings, you are allowed to set state volume in a range of 0 – 128, this is the 2nd place individual volumes can be set. In this case the volume can be at or lower than the base volume set in the file list, it cannot be higher than set in the file list. Using min / max settings you can alter a sound such that it is at one volume as it begins play and another volume when it ends. Useful if you want graduated sound volume, as in an rpm sweep from one notch to another. It is also useful if you want drive sounds to be at one level and acceleration sounds to be louder. In the same way, coast sounds could be lower than drive. You can also set a delay before play if required. Range is 0 – 255ms. Note that you can have the same sound from the file list play at different volume settings if it is used in multiple states.

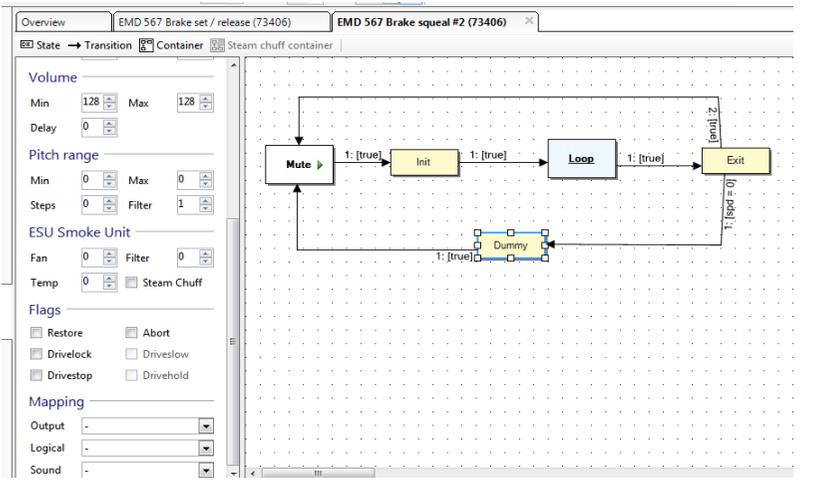


Fig. 73: Flow chart, Brake squeal, State properties 2

Pitch range properties allow the introduction of pitch change into a sound while it is playing. Using this you can simulate certain sound characteristics such as RPM change: Min / Max set amount of change (0 - 255), Steps control incremental change, 0 is smooth, and Filter controls smoothness of change, 7 is highest. **ESU Smoke Unit** allows you to modify Fan speed and Temperature (°C), Filter smoothes change between states at the state level. See Smoke unit, (Chapter 8.10) for global settings. Steam chuff causes smoke to puff with the sound as each chuff sound plays.

The "Flags" allow you to restrict and control decoder actions while a sound is playing in a state, all flags are set by checking a selection box to the left of the flag:

Restore – Sound play will start here if sound was interrupted previously by priority conflict This is useful when set on the init and loop state and not set on the exit state. When the schedule is preempted in the exit sound will not restart.

Drivelock- Decoder will not allow movement while playing (e.g. start sounds, idle)

Drivestop- Stop motor movement while state is playing

Abort- If function is turned off, stop sound play immediately

Driveslow- Do not allow motor stop while sound is playing

Drivehold- Do not allow motor speed to vary while state is playing

The last section of state properties involves mapping. Mapping at the state level allows linking of sound play to things occurring outside the decoder, such as physical and logical outputs, and even calling other sound slots into action. The small figures below display the options that can be linked to sound play:

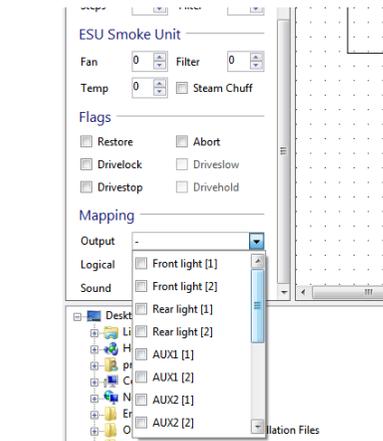


Fig. 74: Flow chart, output mapping

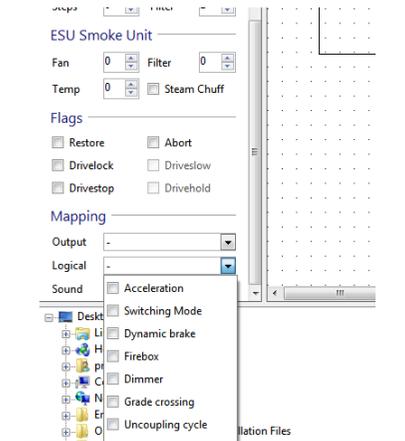


Fig. 75: Flow chart, Logical mapping

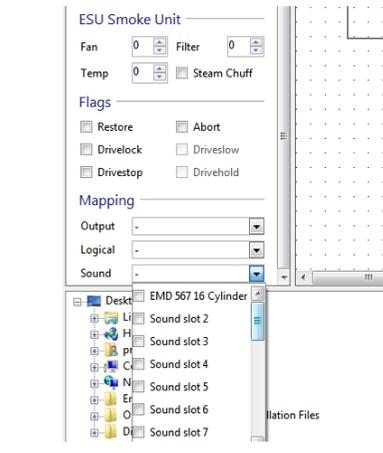


Fig. 76 Flow chart, Sound slot mapping

11.3. Containers and container properties

The next subject is containers and container properties. Containers are very similar to states, in that they carry sound, but containers carry sound in a different way, they carry sound in states placed inside the container. Containers allow you to segment and organize your sound. A loop is a perfect example for a reason to use a container. One frequent issue with sound decoders is heard when playing a loop. The decoder cannot simply quit playing a loop, if it happens to be at mid-point in a loop and receives a command to exit, such as increase speed, it cannot perform this command until the sound completes the loop. This can cause annoying response lags. One way to get around this is to have very short loops, but then they may sound false and "loopy". Now with the v4 decoder, there is a better way, provide a way to exit sound from within a loop.

Sound modeling

Container properties

The performance issue of having a long good sounding loop vs. a quick exit is no longer a problem, because the loop segments provide an exit option after each segment plays, thus providing a worst case time to exit which is the time it takes to play the current segment., in the example below this is 1/4 the time to play the full loop. This is what we are going to find when we examine the container that is in the brake squeal flow chart, figure 77 and 78:

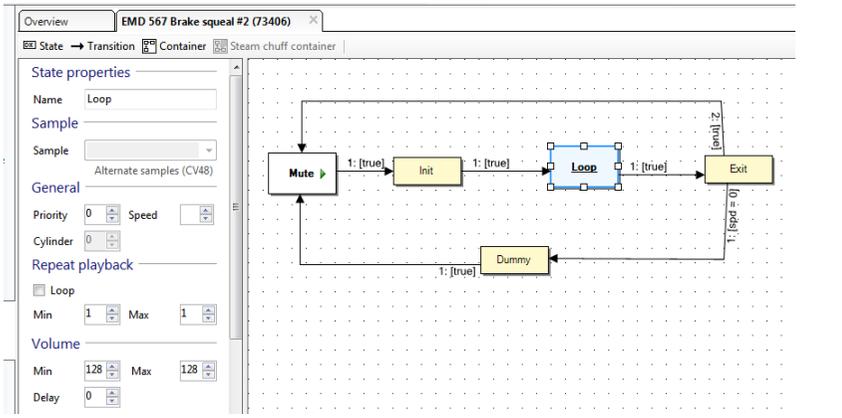


Fig. 77: Flow chart, Brake squeal, select container

Selecting the container with a single click will display the properties at this level, the same as those for states. Note the color of the container, light blue; this indicates there are states within the container. States without sound linked and empty containers display in white. Once data is associated with them States become yellow and containers become light blue. (Except for Steam containers, covered later) Double clicking on the container provides a “drill down” view of the container, allowing a view of what is in the container (figure 78).

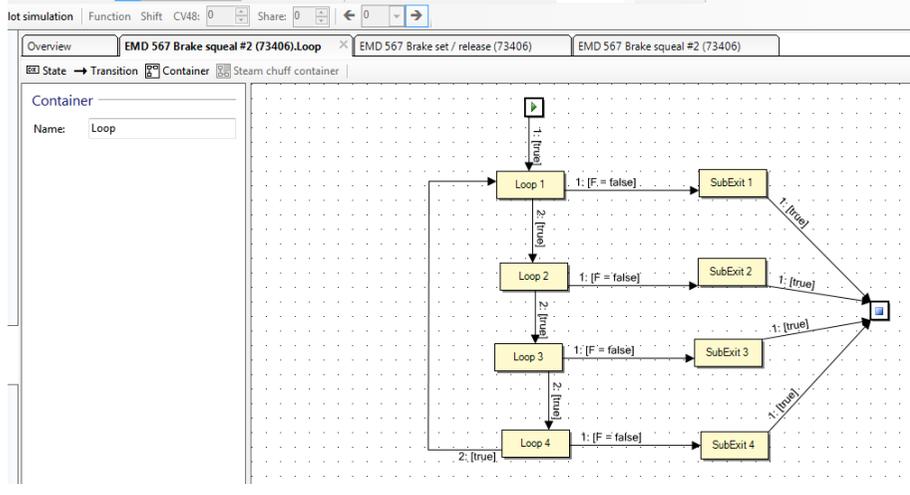


Fig. 78: Flow chart, Brake squeal, inside the container

This is the view inside the container immediately after double clicking the container, note that container

properties are few, only the name is a container property. The properties assigned at the higher level will inherit down inside the container, but they can be changed on individual states within the container if desired. The second thing to note is the green arrow at the top of the container; this is the sound flow entry path. To the right and near mid-point of the drawing is a blue square, this is the exit point of the flow. All the objects on the page can be arranged as desired, this one is top to bottom, it could just as easily be laid out left to right or bottom to top, the artistic part of flow design is up to the individual, the important part of the flow is the operation.

This flow is easy to read, and easy to lay out, it consists of four sets of the same two states. Once the first two have been drawn they can be copied and pasted three times to complete the basic flow, then all that is left is connecting the transitions. Each of the flow segments are the same, drawing wise, but each state in the four stages of the brake squeal loop are 1/4th the size of the original brake squeal loop. At each loop segment proceeding from top to bottom, there is an opportunity for the flow to branch and exit, in each case it is the condition “F=false”. If this equates to true, meaning the brake trigger event goes false, or off, then the flow can exit out of the container. This offers very quick sound response for the “driver”, and it makes the brake sound playable, in that based on the decoder settings for brake sound, the sound can be played by reducing the speed control enough to trigger the brake sound, and then increasing it enough for the brakes to release, enabling prototypical operation with the brakes, such as setting them for speed reduction and then releasing them when desired speed (perhaps yard speed) is reached.

Reading the flow chart shows that as the exit condition is evaluated in each loop segment, if the brake trigger is still enabled (meaning “F=true”) then the flow will proceed down through the next segment, and the next, and at the bottom it will branch back to the top and start over. If you are following using your computer, you can listen to each segment play, and hear how they sound individually. You can also simulate the brake sound and hear it as it would be played from the decoder by using the “Simulation” option above the tabs in the view of the brake flow.

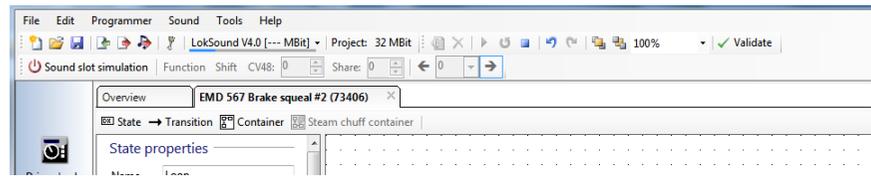


Fig. 79: Flow chart, Brake squeal, Sound slot simulation

Clicking the icon in red will turn it green and begin play of the brake squeal flow chart, or any sound slot flow chart selected, and you can simulate the brake trigger coming on or off by tapping “Function” to the right of the simulation icon. Using the Simulation options allow you to test your sound slots, including the drive sound. This will be covered in more detail later on.

As mentioned previously, individual states can have properties assigned, simply click on the desired state(s) and set the properties as desired. If no properties are assigned at this level, the properties set for the container will carry down to all the states in the container flow.

11.4. Complex sound flow

The next section looks at a complex flow, a drive sound, it may seem overwhelming perhaps, but in reality you can read it in the way you can read the above flows, just take it piece by piece. One way to view a complex flow is in chunks, for instance a drive sound usually has a drive section in the middle, this is the base flow, then it could have an acceleration chunk, to model what happens when the speed control is rapidly raised, and it may also have a coast chunk, modeling sounds when the speed control is reduced and momentum takes over while the train slows down. Looking at a complex flow provides a bit more practice in understanding a flow, and it sets the stage for going through the conditions, which provide the power behind the decoder.

Sound modeling

Complex sound flow

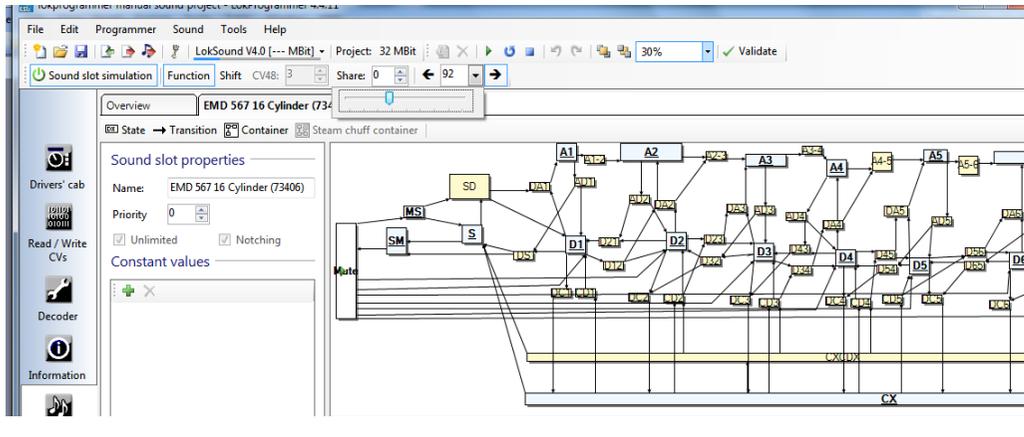


Fig. 80: Flow chart, EMD 567-16, 30% zoom and simulation

Figure 80 displays a diesel drive sound. This is the opening view of the page at 30% zoom. First please note the "Validate" button at the right of the top menu bar. This allows you to check a flow for errors. If any exist they will display as a small red "x" on the state or container that has the error. Common error codes will be covered at the end of the manual. If a flow chart has errors it will not allow saving, it must be fixed or abandoned. To the left of the validate button the zoom factor is displayed, flow charts open at 100%, and can be zoomed down in 10% increments, the above view is 30%, good for an overview and error checking. The second menu bar shows the Sound slot simulation button in green, also the Function button to the right is highlighted, meaning the function is on and simulating the drive activity. If you are following with the software active, you can hear the engine running. Next to Function is "Shift", "CV48" (grayed out), "Share" (set to 0), and a speed slider at 92. Shift and Share will be covered in the conditions section; the options above allow you to exercise these items if they are used in the flow chart as you simulate. In the flow displayed from this much zoom out we can see the overall layout of the flow. It is a diesel, there are no steam containers, it has a start section, then either beginning movement or idle, then a drive path of d1 through d6 visible but we can see it progresses to the right, between the drive steps there are states, these hold the rpm change between the drive steps, both going up to the right and down from right to left, there is also a state with the drive to standing sound attached, (sd) and then either idle (s) or shutdown sound (sm). That is the drive chunk. Above the drive piece is the acceleration section, these hold the sounds for the engine working harder, and the various rpm increase as the rpm builds and waits for the train to catch up due to momentum. Below the drive chunk we see a 2 stage coast chunk, going from drive to cx, which is coast, and then moving up to a state holding the coasting steady rpm sound as it waits for an rpm increase back to drive, or a stop. Lastly we can see the flow is made up of states and containers and we can infer there are quick exit loops in the containers, based on previous study. Also, note the Sound slot properties, it is a named flow, is set to the default priority, and is unlimited and notching. In the following figures close up views of different parts of the above flow will be displayed and we begin the study of the last primary drawing objects, the transitions, conditions, and actions

11.4.1 Transitions, Conditions and Actions

Figure 81 displays the beginning sequence of the EMD 567 drive flow chart. There are important features to consider when we concentrate on the transitions. Note coming from mute we can see 3 transitions out, and 1 returning to mute. The mute box actually has 8 transitions coming out, and each one has a priority assigned to it. The decoder evaluates each transition in turn, starting with 1 and then going to the next one, in sequence, until it hits the top priority, then it starts over. Each condition on each transition is evaluated in turn, as soon as a condition evaluates to "True", then the flow will branch and proceed to the next state or container. If all we do in this case is "Start" the prime mover by engaging f1, transition #8 will

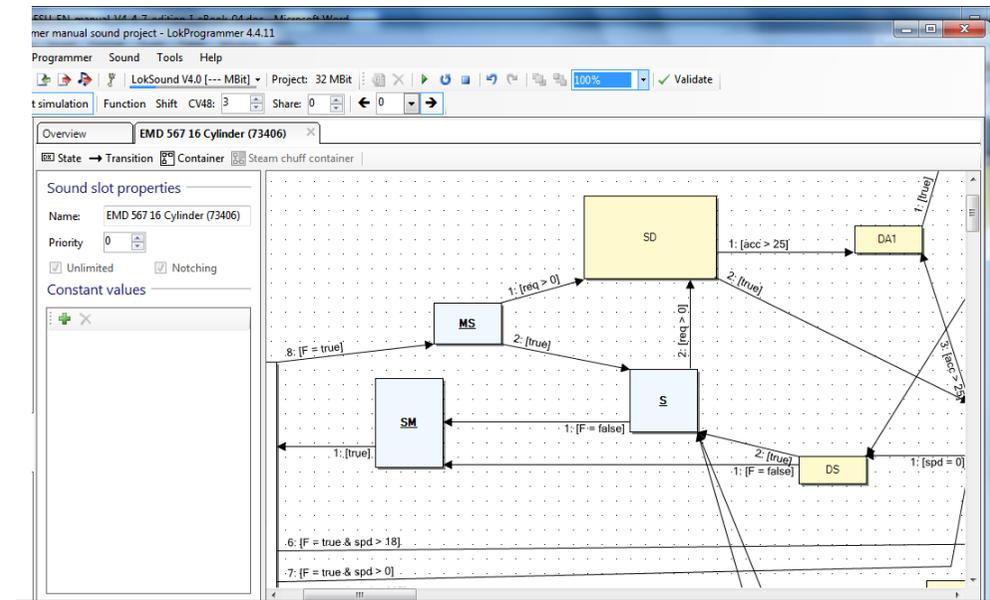


Fig. 81: Flow chart, EMD 567-16, 100% zoom

Decoder	DCC	Forward, F0	→	Front light [1]	-	-
	DCC Settings	Reverse, F0	→	Rear light [1]	-	-
Information	Driving characteristics	F1	→	-	-	EMD 567 16 Cylinder (73406)
Sound	Function mapping	F2	→	-	-	Sound slot 3
		F3	→	-	-	Sound slot 4
		F4	→	-	-	Sound slot 5
		F5	→	-	-	Sound slot 6
		F6	→	-	-	Acceleration, Switching Mode

Fig. 82: Decoder, Function mapping, function 1 mapped to drive sound

become true and the decoder will play the start sound and settle at idle "s". This is because, as shown in figure 82, function f1 is mapped to the EMD 567 drive sound. The sound flow will move to container ms (mute to start) and then when complete, the flow will branch to container s (standing), this is because transition #2 from ms is true, and transition #1 is false, because there is no speed request from the speed control. This is the key to how the v4 sound flow is charted. IF the transition is TRUE, THEN the flow will branch to the next object the transition is pointing to. This is how the flow charts work, all of them. Only one sound plays in a flow chart at a time, which sound that happens to be is governed by the flow and the conditions on the transitions.

In the above case, if f1 is engaged and if the speed control is set to 2, (based on 28 speed steps), then the "engine" would be heard to start, transition #1 would be true because requested speed is greater than 0 and the flow would move to state sd (standing to drive). SD happens to have the property "drivelock" set, so movement could not occur until sd finishes play, and then flow would branch to either DA1 or D1, depending on if there was an acceleration input greater than 25, else the flow would proceed to d1. In this way you can read the entire flow. It may be hard to understand right now because the meaning of all the conditions has not been explained, but the flow can be read and you can familiarize yourself with how it works.

Sound modeling

Transitions, Conditions and Actions

Focusing only on a small segment of drive sound shown in figure 81 displayed in figure 83 below, note that a transition has been selected by a single click, and the left hand pane now shows transition properties.

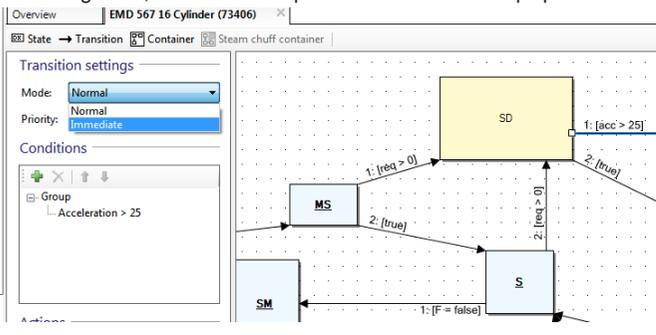


Fig. 83: Transitions settings

The selected transition above is coming from “SD” (standing to drive), the transition is highlighted blue, and it’s properties are displayed in the left hand pane. Mode has 4 options, **Normal**: The successor state will be played after the current sample has been finished. **Immediate**: The successor state will be played as soon as the condition is met (true) the current sample will be aborted immediately. **Exit**: This transition may only appear in containers: This transition works like a transition to the exit state with “true”, but can be placed anywhere in the container. E.g.: create an Init->Loop->Exit flow in a single container. In this case the exit transition would be between Loop and Exit, so the exit transition will be played after the loop and before exiting the container. **Cross**: This is automatically set on every trigger when using multi-channel steam: A cross transition will trigger the corresponding state on the other sound machine while playing the current state to the end. Multi channel steam plays the steam chuffs alternately on sound slot 1 and sound slot 2. Below this we see Conditions; here is where conditions are placed that control a flow branch.

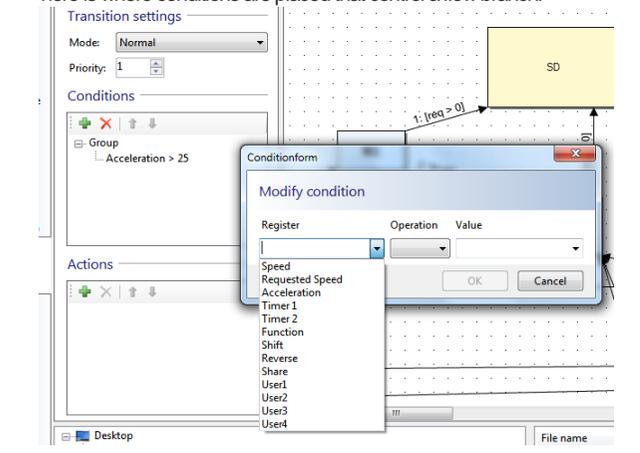


Fig. 84: Condition settings

Note in figure 83 the condition in effect is “Acceleration>25”. In figure 84 we see a dialog box, “Modify condition”. There are 3 ways to invoke this dialog, 1 is to click the “+” icon, this will call the dialog, and allow adding another condition to the transition. 2 is to double click the existing transition, allowing you to modify the values of the existing condition, and 3 is to select “Group” and then hit the “+” icon. The reasons for using these options are to allow the building of condition(s) that will control the actions of a sound flow branch as desired as follows:

- 1 – Using the plus icon will add a second group below the current one, the reason to use this option is to allow a flow branch in an OR condition, (also called an OR gate). An OR statement allows a branch if either the first condition OR the second condition evaluates as TRUE.
- 2 – If we want to change the value of branch condition then modifying the existing condition is easiest, just double click, and then change the value as desired, for instance, if we wanted the branch condition to be less sensitive to acceleration, we could change the value from the current 25 to 12, this would make the branch require about half the previous acceleration.
- 3 – Instead of an OR gate, we may wish to have an AND gate, which requires 2 or more conditions on the same transition, all of the conditions must equate to TRUE before the sound could branch. Single click the condition and hit “+” to add another condition to the group, this creates an AND gate.

11.4.1.1 Condition register options and definitions:

Speed – This is the value measured by the decoder, it is not a speed step value, it is within a range of 0 – 255, which matches the full speed range across the drawing field, which is also 0 to 255. If the drive step range from s to d8 (or the top drive step) is less than 0-255 then the speed conditions should match the flow chart range. Decoder settings will fit the sound spectrum into the driving characteristics.

Requested Speed – This is the speed value requested by the user when the DCC speed control is changed, but the values entered on the conditions should match the range (0-255) used on the drive flow, not speed step values.

Acceleration – Decoder measured acceleration rate based on magnitude of speed control change and rate of speed control change. Minus number indicate deceleration, which is negative acceleration.

Timer (1 and 2) – 2 timers are available to use for time based flow branching, values are from 0 – 255, in seconds, therefore a timer can be set for a maximum of 4 minutes 15 seconds.

Function – looks to the function key the sound slot that contains the flow chart is mapped to. If the function key is engaged, the function state is true, if disengaged the function is false. (True / False only)

Shift – Shift is a logical function that can be mapped to function key(s) using the logical function section (figure 35, chapter 8.6.4). Shift can be evaluated in a condition to be either true or false, if the condition evaluates as true then the flow will branch. (True / False only)

Reverse – evaluates the decoder direction setting, if reverse is true then the flow will branch on the reverse path. (True / False only)

Share – Share is a global register, meaning it can be active and evaluated across multiple flow charts. Share can be set in one flow chart and control flow branching in another flow chart. E.G. Share is set to 10 in flowchart A. Flowcharts B, C, D, and A all have branch conditions of “Share=10”. When all 4 flowcharts reach the condition “Share=10” then all 4 flow charts will take that flow path. Share can only have 1 value at any one time.

User(1,2,3,4) - 4 registers dedicated to variable conditions, such as random loops. E.G. “User1= 1”. Flow will branch when the value of User1 equals 1. Users (1,2,3,4) are all set by Actions, as are the values of Share, and Timer(1,2). If an action such as “User1=rand(3,8), then when the sound flow reaches that action, User1 will be set randomly between 3 and 8. Next a flow could enter a loop where there are 2 methods to exit, 1 condition for exit is “User1=0”. The other condition is a simple “true” condition, but it has an action that reduces the value of User1, such as “User1-1”, thus the value of User1 is decremented by 1 each time the loop plays, until User1=0, then the flow branches out of the loop.

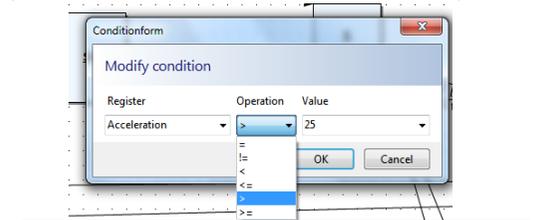


Fig. 85: Conditional Operators, Logical

11.4.1.2 Condition operators and values:

Each condition can be evaluated based on the logical operators as shown in figure 85. These are standard logical operators, and they can change somewhat depending on which register is being evaluated. For instance, Function, Shift, and Reverse only allow “=” (equal) as an operator.

The options for the “Value” field can be hand entered values, as in the drive flow chart example (figure 81), or they can be predefined values using the table of constant values. (Seen in figure 80 Using the Constant values table to predefine values that can be subject to change later can allow the use of a template flow, with values changed for different types of locomotives, such as the differences between a road diesel and a switcher that both have the same prime mover. The switcher would require different settings for the drive steps and acceleration, but the rest of the flow chart could be the same. Using the Constant values table such a change could be made in minutes as opposed to hours. An example of constant values will be shown later.

11.4.1.3 Actions:

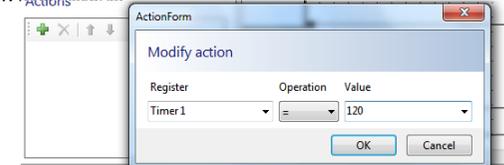


Fig. 86: Adding Action, Timer1=120

Figure 86 (Prev. page) displays adding an action to the transition being modified in figure 83, in this case setting a timer for 2 minutes to be used in a follow on condition. An example using the timer is shown below

Actions are used on transitions to set a value to be used in a follow on condition. The property field for Actions is directly below the Condition field. Adding, deleting, and modifying actions is done in same way as conditions.

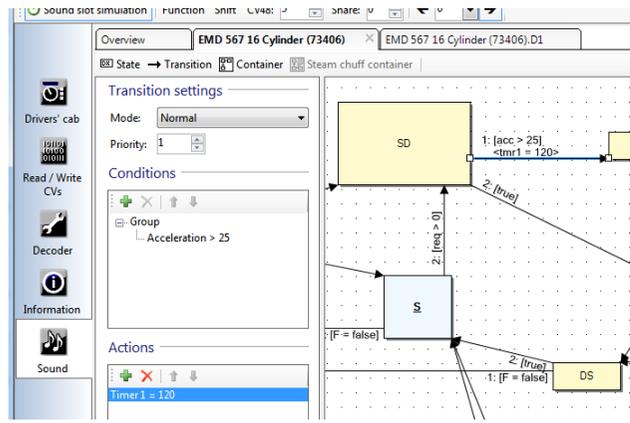


Fig. 87: Action added, Timer1=120

Figure 87 displays the view after an action setting Timer1 to 120 (2 minutes) has been accomplished. Figure 88 below shows the register items that can be set using actions. They are Timer(1,2), Share, and User(1,2,3,4).

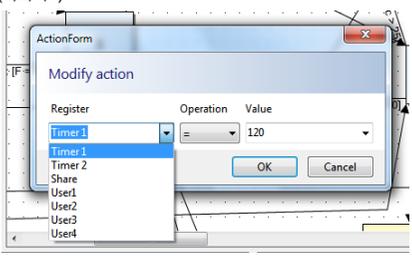


Fig. 88: Action register items

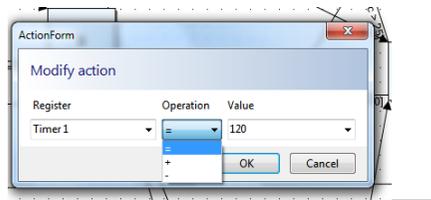


Fig. 89: Action logical operators

Figure 89 displays the logical operations that can be applied to actions, equal, plus, and minus. In the value field you can set the action values, or you can set a predefined variable, if it is available in the table of constant values.

11.5. Steam sound flow chart

Figures 90 and 91 displays a typical steam sound flow chart; note the use of the steam container and the project designation of "Steam single channel". "Steam multi channel" is also available. Selecting "multi channel" duplicates the flow from sound slot 1 to sound slot 2, then decoder settings for steam sounds (Chapter 8.11 figure 45) can be used to apply offsets to simulate an articulated steam locomotive such as a 4-8-4 Union Pacific Challenger.

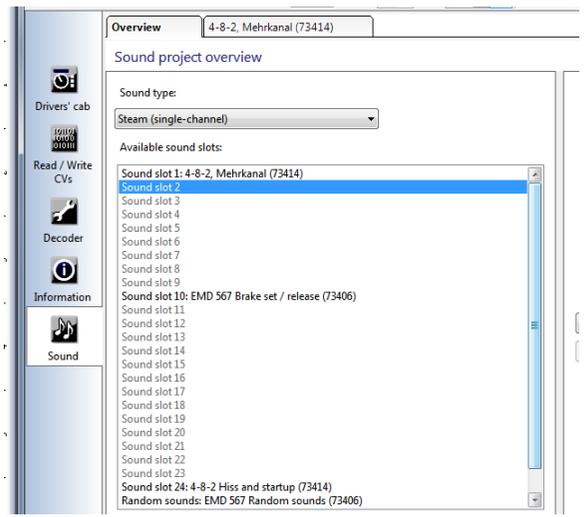


Fig. 90: Steam single channel project

Note that "Hiss and start up" sounds are placed on sound slot 24, if multi channel is selected then sound slot 24 is renamed "Hiss and start up". This file is loaded from a template; See figure 91.

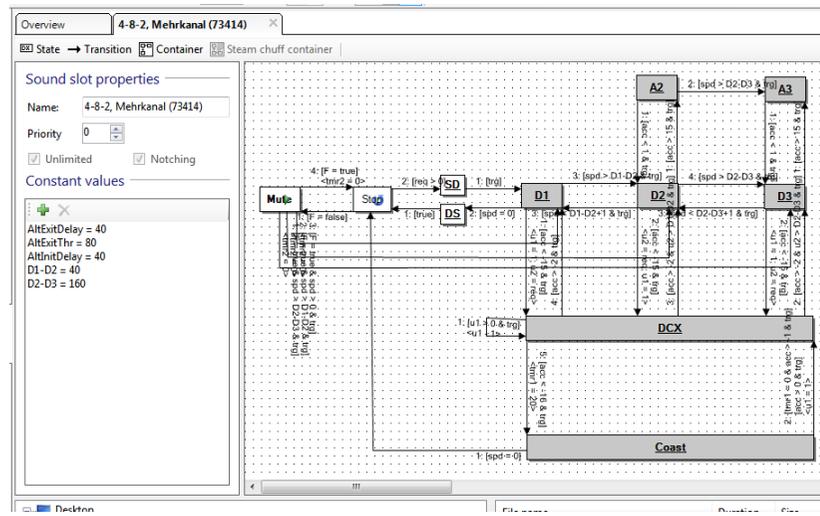


Fig. 91: Steam drive layout, Constant values

Figure 91 demonstrates a common steam layout in which the change in chuff sound as speed increases warrants multiple drive steps to capture the prototypes sound. In some steam flows on a single drive step is displays and the rate of chuff increases with speed is handled by decoder settings (Chapter 8.12, figure 47). In the examples section (Chapter 12.) a series of figures will display the steam layout of project 74414 Mikado.

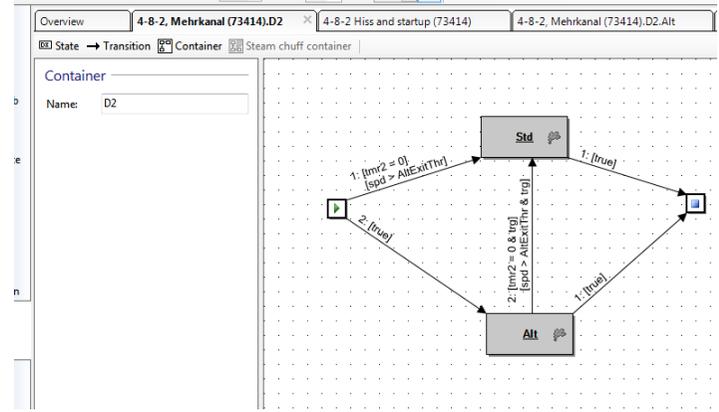


Fig. 92: Steam D2 drill down, Std and Alt containers

Note fig 91 Constant values, defining the Alt (alternate sounds) values which direct the sound into either the Std container or the Alt container.

Sound modeling

Steam

By double clicking either “Std” or “Alt” steam containers shown in figure 92 it is possible to “drill down” into the container and view the chuff layout as shown in figure 93 below:

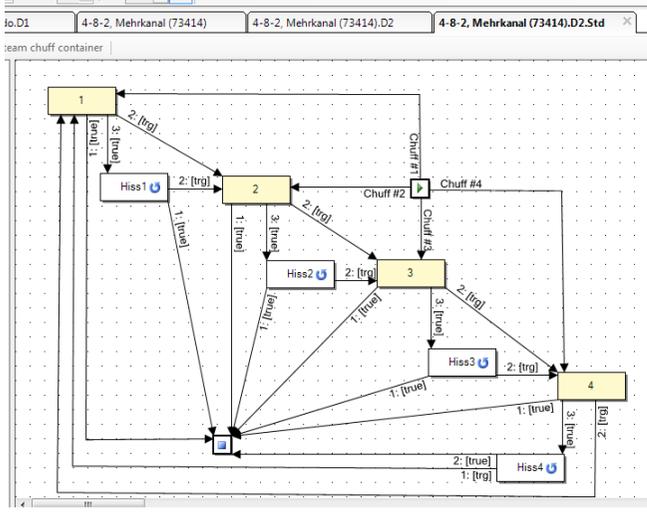


Fig. 93: Steam chuff layout

Note the layout of the chuff sequence; it is a 4 chuff per wheel revolution layout. The unique thing about the steam container is that as you layout your chuff sequence, each state has a transition going to it from the entry point, and the transitions are automatically named in sequence when the transitions are added. In addition, when the transitions are drawn between the states they automatically are named “trg” for “trigger”. It is the assignment of the trigger transitions that tell the chuffs how to be timed, based on decoder settings as explained in Chapter 8.11.2. Finally, note the absence of sound in the “hiss” states, this is because in this case the hiss sounds are played from sound slot 24, “Start up and hiss”. See figure 94.

Referring to figure 94 note the detail for the hiss layout, it has Constant values that are the same as the Steam chuff flow. Also it is a simple flow, in that the hiss will play appropriately when the chuff sounds from figure 93 play and are followed by a blank place holder, which is where the hiss will fill in. The reason for handling the hiss this way is that it makes overall development time much shorter, as the developer does not have to go into each chuff container and add the same sounds over and over. This is an example of the flexibility of the v4 architecture and it demonstrates you are not stuck with only one way of doing things.

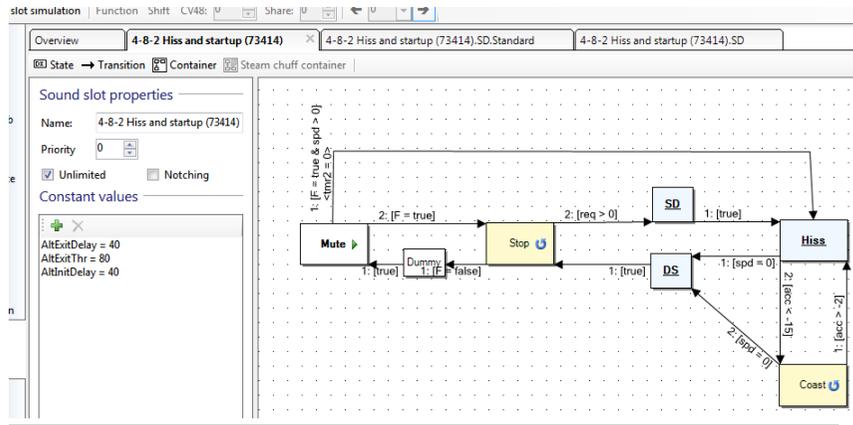


Fig. 94: Steam hiss layout

It can be seen from the flow that there are loops at states “Stop” and “Coast” that play appropriate hiss sounds for those events. In SD and DS containers, there are flows of sequential hiss appropriate for those events. Drilling down into SD we find both Alt and Std containers, and drilling down into Alt we can see where the items defined in Constant values are controlling the play.

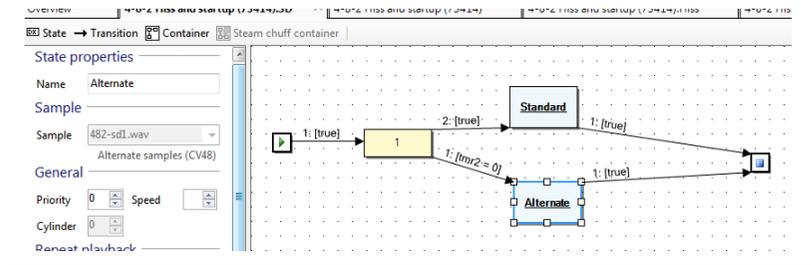


Fig. 95: SD drill down

Note in figure 96 below that the hiss play in state 2 is controlled by the value of “AltExitDelay” as defined in the Constant values table.

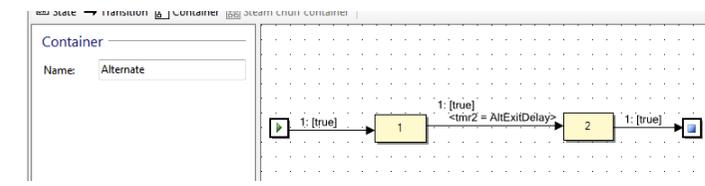


Fig. 96: Alternate container drill down

One final thing to note is that the containers in the “Start up and Hiss” flow is the lack of Steam containers. This is because only sound slots 1 and 2 can have steam containers, based on their relationship to decoder settings as mentioned earlier. However, normal containers work fine for the hiss flow, but it is important to note the Hiss flow can only go on sound slot 24 if play is to be automated and synchronized with sounds slots 1 and 2.

Sound modeling

Terminology definitions

11.6. Sound modeling terminology

The below listing provides definitions for the drawing items seen on various flow charts, many of these carry over from version 3.5 decoder practices.

Flow chart symbols:

State – a box, white initially, changes to yellow when linked to a sound file (.wav)

Transition – a line, connects to states and containers, holds conditions and actions

Container – a box, white, changes to light blue when states are added, provides an organization method and drill down capability to hold flow segments

Steam container - same as container, gray / dark gray, has special properties in that it organizes steam chuffs and chuff layout for various cylinder counts

MUTE – sound start point, begins / ends each flow chart, “home” for the sound flow, and contains no sound

S – (s), code for “standing” (idle)

SD – (sd) code for “standing to drive” (initial movement)

DS – (ds) code for “drive to standing” (final movement)

D – (d)(d1,d2,d3,...) code for drive step (notching, steam segment)

Dnn – (d12, d23, d32...) code for a speed transition, notch change

A – (a)(a1, a2, a3...) code for acceleration, rpm change, working harder, etc

Ann – (a12, a35, a53...) code for speed / rpm transition, notch skip

AD / DA - (ad2, da2...) code for drive to accelerate / accelerate to drive

C, CX, DX - (c, cx, dx) code for various coasting sounds

DC, CD, CS – (dc7, cd7, dc3, cd3,...) code for “drive to coast” / “coast to drive” / “coast to standing”

Numbers following letters signify the drive step number, as in, an 8 notch drive chunk would have D1 through D8 as drive steps, with various transition points between each step, such as d34, d54, etc marking notch (rpm) changes.

Alt – code for an alternate sound option that can play based on a time setting, or a speed setting. Typically used in steam projects to set the occurrence of start up with open cylinder cocks.

Std – code for normal sounds under typical running conditions if there is an Alt (Alternate) flow.

AltInitDelay - Defines how long (in seconds) the locomotive must be stopped before the next drive away uses the alternative path.

AltExitDelay - After drive away in alternate mode the flow will be exited after this time. The path may be exited before this time if the locomotive drives fast enough. You can change the speed setting using “AltExitThreshold”.

AltExitThreshold - After drive away in alternate mode the flow will exit if the locomotive drives faster than this value, the value depends on 255 speed steps which are calculated according to the decoder speed value. The alternate flow may also be exited by AltExitDelay.

Constant Values – A table reserved for user definition of variables used in a flow chart. E.G. “D12 = 18” defines the speed value on a scale of 0-255 when the sound path should branch from d1 to d2, d12 would hold the rpm transition sound playing the rpm change between notch 1 and notch 2. The table is optional; its use greatly simplifies making speed range changes in the flow chart, as there is no danger of missing a change item or causing a drawing error, since the drawing page is not touched during the change, only the table values.

Sound slot – a designation for an object designed to link to a flow chart, v4 has 27 sound slots, some are pre-designated for special tasks, “random sounds”, “brake sound”, “Steam hiss and start”, “Gear shift sound”. Sound slot 1 is reserved for drive sound flowcharts, if multi-channel, sound slot 2 is also reserved.

Sound library – a set of templates displayed on the sound overview page, upper right hand pane. Each template carries the sounds that match the template flow chart.

Sound flow chart – a group of drawing objects arranged on a drawing space, usually arranged left to right with acceleration going from center up and coast going from center down. The drawing page range / speed spectrum is 0-255. The flow chart links to all the sounds that define the purpose of the flow chart, such as a horn / whistle / drive sound. Flow charts are linked to sound slots; sound slots are mapped to function keys or decoder settings.

File names – lower right hand pane of the sound overview page, displays all the sounds loaded in the sound project, dark text indicates file in use, grayed out files are present, but are not yet linked to any flow chart.

File navigation – (Desktop) Lower left hand pane of sound overview page, displays a file navigation view to assist in selecting individual sounds to “drag and drop” in the file list.

Sound project – A term used to designate a full set of data, made up of decoder settings (CV's) and sounds, which is created, opened, and edited by the LokProgrammer software.

LokProgrammer software (LSP) – The software used to create, open, edit sound projects. If used in conjunction with the LokProgrammer hardware, enables the reading of and writing to ESU DCC decoders. When written to, a decoder can be configured with both decoder settings (CV's) and decoder sound. Decoder sound cannot be read by the software, the sound project must be open in order to change any sound settings that are displayed via the sound icon. Can be used without LokProgrammer hardware.

LokProgrammer hardware – A communication interface intended to be used in conjunction with the LokProgrammer software and the ESU decoder.

12. Sound modeling, Examples and Tips

It is difficult to present the act of drawing a flow chart which is a dynamic process on a printed page which is static, but, the next few figures all represent a set of steps that take perhaps 2-5 minutes to accomplish. The result represents the basic process of drawing a sound flow chart, all flowcharts are done the same, and the only difference is the level of complexity.

12.1. Flowchart drawing example

Figure 97 shows the opening view when an empty sound slot is either double clicked, or right clicked followed by hitting "edit sound slot".

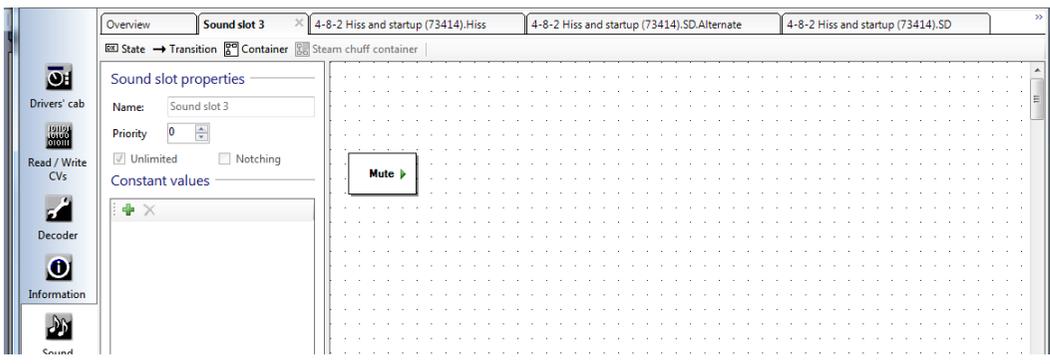


Fig. 97: New flow chart, opening view

The next step should be the naming of the sound slot and adding a few states or containers to begin the layout. States and Containers display in the upper left corner when added, and then they can be moved as desired. (Left click, drag and drop) After States / Containers are placed, they can be linked by adding transitions. The next view shows 3 states arranged on the drawing field, and one still in the upper left hand corner waiting to be moved.

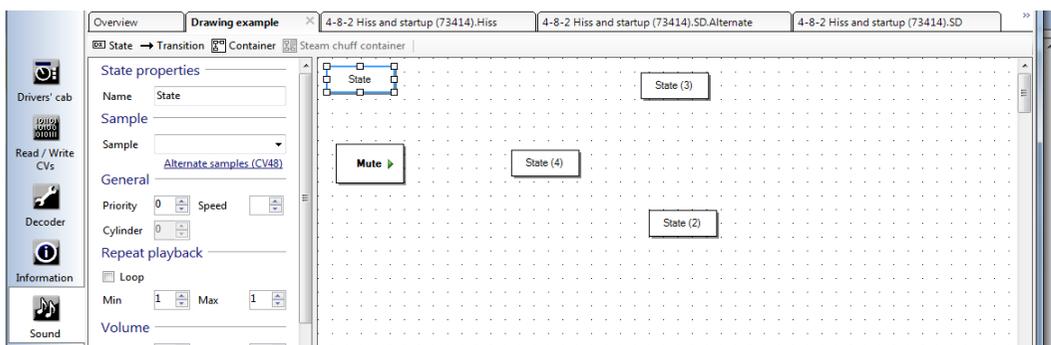


Fig. 98: 4 states added, 3 placed in position, 1 waiting to be moved

The goal of this flow is to play a beginning sound, have it loop in place 8 times, then proceed to the next sound, play that once, and exit. There is an exception however, should the loco be reversed, the loop should exit immediately and proceed to an alternate sound, play that once and exit the flow. Sound hard? No, it is actually very simple; the next figure will display a way to do exactly as stated. Along the way, the states will be named as they are set up so the flow can be followed easily.

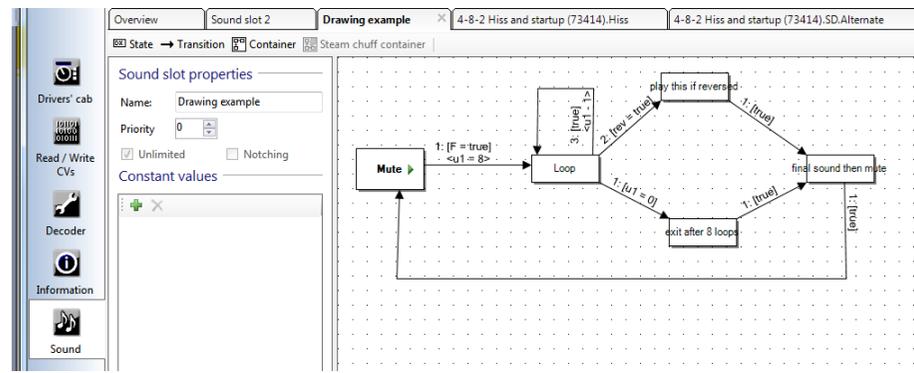


Fig. 99: Layout complete, all named, conditions and actions complete and validated

Use this example as an exercise and complete the drawing. You don't need sounds; you can draw and validate your flows without sounds linked. You do not need the LokProgrammer hardware to practice setting up sound flows. You can later add sounds to the flow and use the simulator to see how it works and sounds. The LokProgrammer hardware is only needed to write sound to the decoder, not for creating and testing sound flow. You only need the hardware when you are ready to write to a decoder.

If you examine the state called "loop" you will notice there is a state property that could have been set to loop 8 times (see figure 98 "Repeat playback") Why then is the flow in figure 99 drawn with a "manual" loop and an action to decrement "User1" by 1 each time it loops? This is because the set up statement requires the flow to exit the loop immediately upon reversing the locomotive. Had it been drawn using "Repeat playback" property, the flow would have to play all 8 loops regardless of direction prior to exiting the state.

Note the priorities on "Loop", transition 1 and 2 both have conditions; 1 cannot be taken until the value of "U1" (User1) equals 0, 2 cannot be taken unless "rev" (reverse) equals true (loco is in reverse), 3 has no condition and it has an action that decrements the value of User1 by 1, therefore after 8 loops User1 will equal 0 and the flow can branch out of the loop. Transition 1 coming from Mute has an action that sets the value of User1 to 8. The names of the states explain what is happening at any time the flow is active. If the transition properties were set in a way where the current number 3 was a different priority, an error would have been generated. This is the most common error you will see, and is almost always generated by making a change and having the transition priorities change.

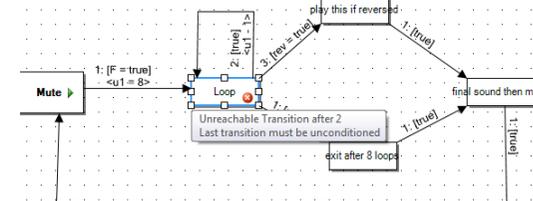


Fig. 100: Commonly seen error, last transition must be unconditional!

Modifying any flow chart is a combination of moving, adding, or deleting states, adding, moving, or deleting transitions, and changing values or changing conditions and actions, or adding or deleting them. There is an "undo" feature that allows you to step back to a previous change point if needed.

Sound modeling

Examples and Tips, Continued

12.2. 74482 GE P42 AMD 103 project examples

The next set of figures is taken from the GE P42 sound project. The intent is to demonstrate how a certain condition is used, or how to achieve a sound flow goal.

This project represents an example of making use of the v4 sound modeling capability to accomplish several prototypical sound characteristics that are rarely, if ever, seen or heard in any other decoder. First, a lo / hi idle triggered not by user input via a function key, instead it is triggered by events that occur on the prototype, such as temperature climb (simulated), compressor kick in, or beginning movement. There is an active multi speed HEP (Head end power) which can be influenced by time, a function key, locomotive road speed, etc. You are invited to download and view this project in the LokProgrammer software as it will be much easier to see and follow the modeling.

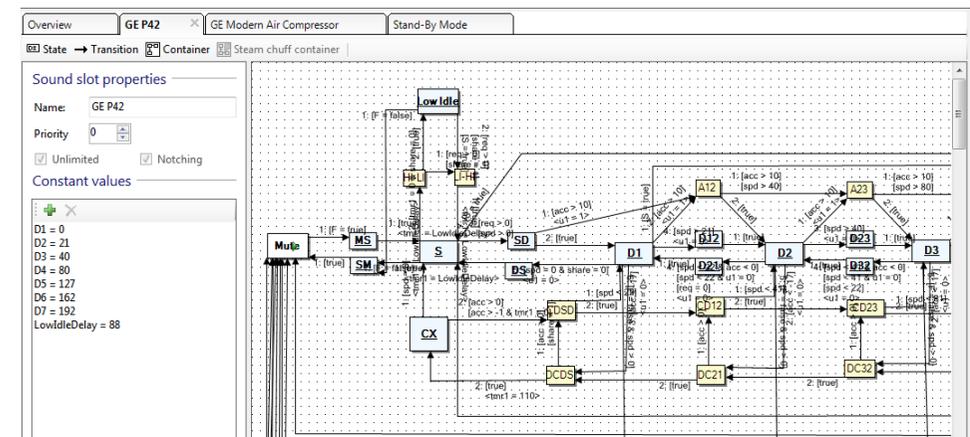


Fig. 101: P42 flow, low idle, constant values

P42 flow overview (above, partial), note the low idle flow, the next 2 figures show more detail:

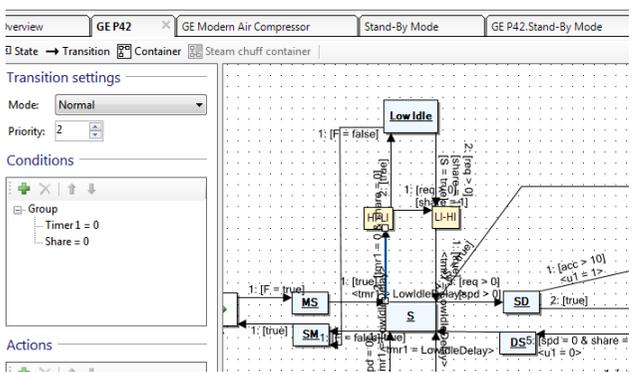


Fig. 102: P42 flow, low idle, AND gate conditions

Figure 102 (left) displays the conditions that allow the sound flow to go to low idle, demonstrating an AND gate; timer1 must = 0, AND share must = 0. Timer1 is simulating a temperature trigger. Timer 1 is set when the flow exit's MS, via the condition on the transition to S. It is set to 88 via the definition in constant values. It is also set when the flow returns to idle from driving. How does share get set to 0? It takes both conditions to allow low idle to occur. We look to figure 103 to see that event happen.

Notice that figure 103 is not from the P42 drive flow chart, it is from a different sound slot. Figure 103 displays the compressor flow chart. The compressor flow is placed on 2 sound slots, it is in the random sounds, such that it may come on randomly, and it is also on a sound slot mapped to a function key, such that it will sound when f20 is

engaged. The first thing that happens when the compressor flow begins is that share is set to 1 by an

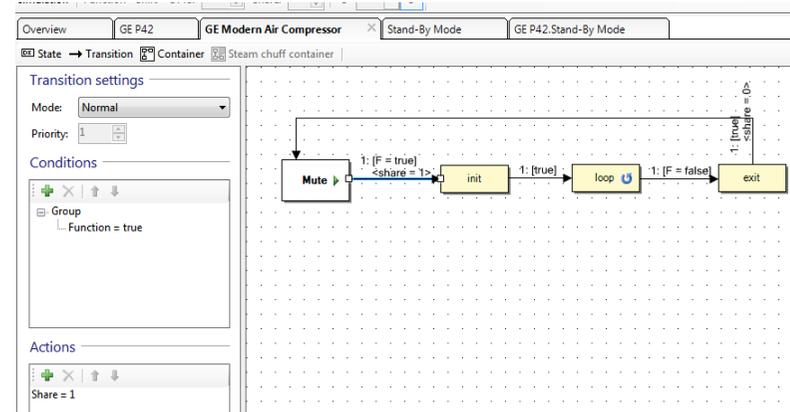


Fig. 103: P42 flow, compressor with transitions setting share

action on the transition out of mute. (Followed in figure 104.) After that the compressor sequence begins. It will loop until turned off, either by f20 or by the random events in the random sounds flow. The last thing that happens prior to mute is that share is set to 0 by an action on the exit transition. Now the AND gate in figure 102 evaluates to true, and the drive sound flow is allowed to progress to low idle.

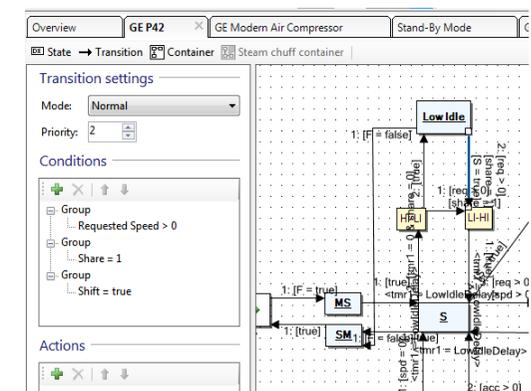


Fig. 104: P42 flow, low idle, OR gate conditions

In figure 104 we are looking at the flow from the standpoint of low idle. We know how the flow gets to low idle, now how does it get out? The OR gate provides 3 options for exit from low idle. Any of the 3 becoming true will allow the flow to exit and proceed to S (idle). These options are requested speed greater than 0, OR shift becoming true, OR share equal to 1. We know how share gets set to 1. That happens when the compressor kicks on. How does shift = true? Looking at function mapping shows that shift mode (a logical function) is assigned to f5, as well as the sound "short air let off". Therefore, allowing air to release will cause a compressor demand, requiring normal (Hi) idle while the compressor runs.

Sound modeling

Examples and Tips, Continued

Changing focus to the right hand side of the P42 drive flow, the HEP (Head end power) portion of the flow is displayed. This flow models the HEP quite faithfully, it is speed dependant and it has heavy load sound, normal load and stand by mode modeled. Various settings of share and shift are used to make the HEP prototypical and also respond to the user via a function. It also has a stand by mode for standing (idle), in which it can be silenced if the user desires.

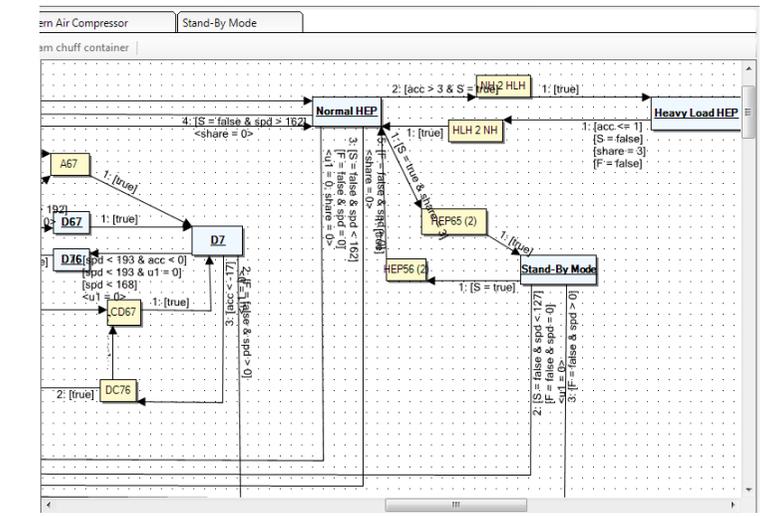


Fig. 105: P42 HEP flow

Figure 105 above displays the basic HEP which runs concurrently with the prime mover sound as the locomotive is underway. There are rpm transitions up and down to simulate the HEP going from normal load to both heavy load and stand by. Figure 106 below displays the drill down from the stand by container, showing the quick exit feature drawn in the loop portion. Note the setting of share upon exit.

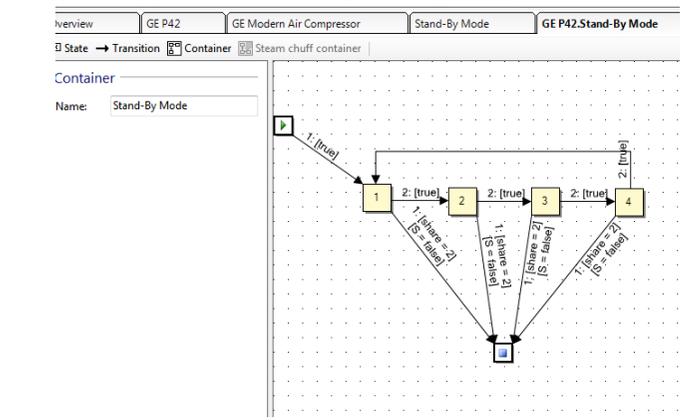


Fig. 106: P42 HEP, loop portions quick exit

The last example from the P42 project displays the stand by mode for HEP that allows the HEP to go silent. It's a simple flow set on sound slot 18. The flow sets share equal to 3 when it begins, and it requires both shift equal true and speed = 0 before the flow can become active. Upon exit it sets share to 2 and will exit when either speed is greater than 0 OR function= false, it is mapped to f13.

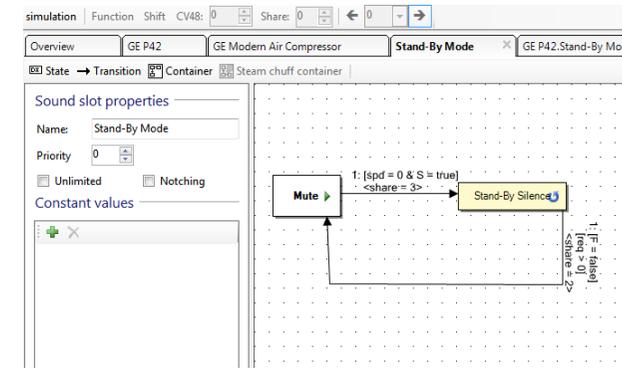


Fig. 107: P42 HEP, standby mode when standing

There is a tremendous amount of flow charting examples to be found with in the ESU sound library and in the projects available for download. Well worth looking over.

Tip: Use copy and paste, you can copy from one sound slot, paste the useful part into another, then hook up the transition lines. If you remove the sound links before copy, you can use the flow and link to other sounds later. Or if you want the sounds, you can copy the flow with sound links and when you paste it in you get the entire flow or portion, sounds and all.

Tip: Use containers for organization, and to create modules. For instance; the drive steps in the P42 flow are all using the same drive step flow (module), create it once, and use it many times. In the P42 flow, it is used 11 times for drive steps and HEP, same flow portion, just different sounds in each use.

Tip: Copy and paste works on transition lines as well as states and containers.

Tip: You can use drag and drop to link sounds to states, you can drag from the file list to the state, you can even drag from the file list to the drawing view and a state will be created with the sound linked "on the fly". You can also drag and drop from the file navigation view in the lower left hand pane and the sound file will be converted, a state created and the sound linked, on the fly.

Tip: The undo function can help a lot if you make a mistake or accidentally cause an error condition, don't forget it's there to help.

Tip: Use the "Constant values" table. The table is optional; but its use greatly simplifies making speed range and other value changes in the flow chart, as there is no danger of missing a change item or causing a drawing error. You can literally save yourself hours of work when updating or tweaking a complicated sound flow such as a drive sound.

Tip: You can have more than 1 instance of the LokProgrammer open at a time, makes copy and paste that much easier.

13. Errors and troubleshooting

As you begin exploring the drawing format, you will make mistakes and discover errors. The software must protect itself against errors, because the results of the flowchart drawing process is converted into computer code that tells the decoder how to sequence and play the sounds that are in the file list. Therefore, the software protects itself by issuing a somewhat terse error message, and placing a small red "X" in the offending state or container.

In almost all cases the software will not allow you to write sound data or save a sound project file if there is an error present.

As you start out learning how to draw the sound flows, you will become familiar with the small red x. Here are the most common errors you will see, and how to resolve them.

13.1. Last transition must be unconditional

This error will occur when adding transitions between states and when putting conditions on transitions. It also can occur when using copy and paste or moving a group of transitions and states using drag and drop. In most cases you will find a transition priority that reset during the action, or a priority mismatch that happened during transition adding or deleting.

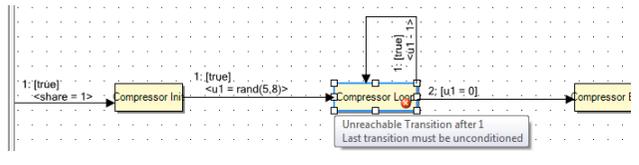


Fig. 108 "Last transition must be unconditional"

To repair this, examine the transition priorities, find one with no conditions and change the priority to make it the last transition in the cycle. For instance, in figure 108, change the priority of the manual loop to "2". This error will also inherit upward if the states are in a container. Figure 109 demonstrates how the condition will display with the states of figure 108 in the container.

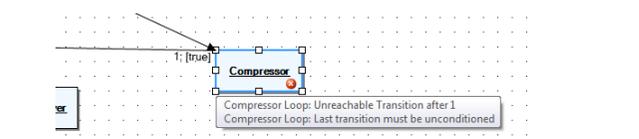


Fig. 109, Error inherited from state inside container

The error shown in figure 109 is self explanatory and does a good job of pointing you to the problem inside the container, look to transition 2 and examine the priority or the condition. Once you get used to seeing this error fixing becomes almost automatic, just change a priority and hit validate again.

13.2. Dangling outgoing transition

The "Dangling,...." error code appears most commonly when using copy and past with groups of objects, frequently a transition becomes unstuck and has to be properly connect to a state. Seen in either containers or states, the fix is to look for a red transition and connect the ends to the state or container they should connect to.

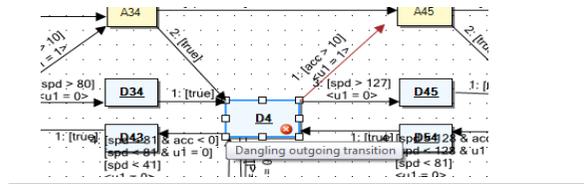


Fig. 110, Error from unhooked transition

As seemingly obvious as the above error is, you will find they are easy to miss if you are engaged in a major drawing effort. 2 other examples from disconnected transitions follow, from a state inside a container and the inherited error at the container level.

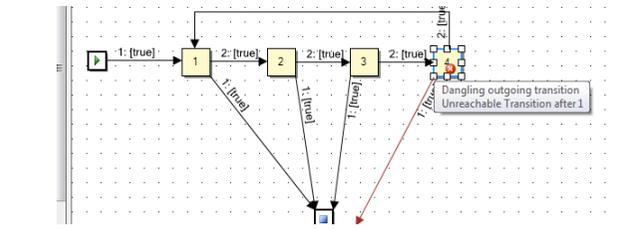


Fig. 111, Error from unhooked transition inside container

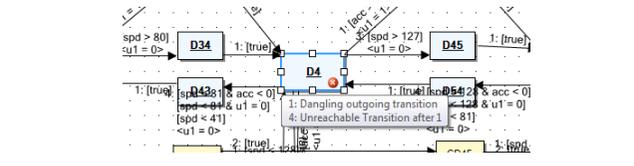


Fig. 112, inherited error from inside the container

13.3...exit Dangling incoming transition

This error, shown from a state level perspective within a container and also from the container perspective is more subtle than the previous. This is because the problem does not flag an error message at the state level, the flow in figure 113 does not display an error, because there is a path for the sound flow that passes the logic checking in the "validate" utility. But looking to figure 114 it is obvious it is seen as a problem at the higher level where the containers are.

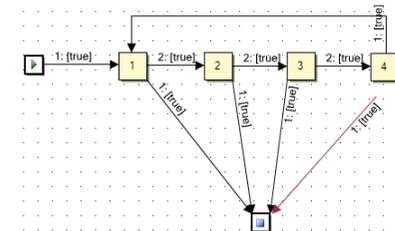


Fig. 113, unhooked transition from state end inside container

Note the transition error difference when it is unhooked at the originating state

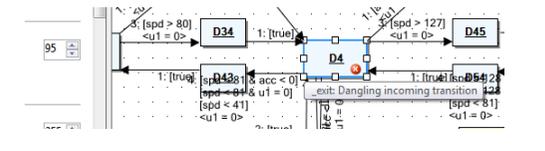


Fig. 114, inherited error from transition inside the container

13.4. Unhandled exception

The unhandled exception error is the most serious error that can occur. There are a few variations of the error; this is one that is caused by deleting a key portion of the flow, or of the table of Constant values, that is used at the core level of the project. The clue in this case is within the error statement "Index was out of range". The method used in this case to produce the error was deleting a portion of the table of constant values. Normally the program does not allow deletion if the item is in use, in this case it was deliberately forced in order to produce the error for demonstration. If this is experienced during the build of a new sound project, it is advised to exit the LokProgrammer without saving the project, restart and reload and re-accomplish any items not previously saved. The software is quite mature now, and this error is not often seen.

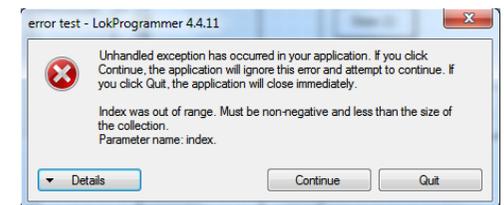


Fig. 115, unhandled exception error

This error may force a software re-install or a driver install to correct some sort of corruption, it depends on the circumstances of the occurrence. If it continues to occur, see troubleshooting section.

13.5. Problems when reading the decoder

Should the program not be able to read out decoder data then an error message will be displayed. The display of that message could have several reasons:

- The locomotive is not set properly on the programming track or the track is not properly connected to the LokProgrammer.
- The decoder is not wired correctly – particularly the motor leads - in the locomotive.
- The locomotive may have capacitor(s) in the motor circuit or on a circuit board.
- The decoder may be faulty.
- The track is dirty.

13.6. Troubleshooting Problems

Proceed carefully through the list above, even though you are convinced you know the solution, take your time and confirm each point:

- Test your programmer track attachment quickly by substituting another locomotive, this way you can confirm with a known good install.
- Trace your wiring connections carefully, isolate the decoder from the wiring if possible, then the wire connections can be traced more easily.
- Sometimes you may have to do a simple hook up, such as only attach track leads and motor leads, but be careful to protect for short circuits. If using a plug in decoder, substitute another decoder.
- If you have a decoder tester, wire the decoder to the tester and see if it works properly that way, this would confirm a bad decoder, or a bad locomotive.
- Try a decoder reset.
- If operation is intermittent, test carefully for mechanical issues, bad power pick ups, dirty track.
- If capacitor(s) is / are visible, disconnect one lead from the capacitor(s).
- The troubleshooting goal is to isolate the problem to a causative factor, decoder / locomotive / installation, once isolated you can then fix the problem.

13.7. Customer service – Assistance and support

Should you require assistance your first call should be to your dealer where you purchased your LokSound decoder. He is your competent partner for all questions around model trains. We recommend highly checking the FAQs on our website at www.esu.eu

Mail and telephone

Of course you can reach us via post, fax or telephone:

ESU electronic solutions ulm GmbH & Co. KG
Edisonallee 29
D-89231 Neu-Ulm

Tel.: +49 (0) 731 - 1 84 78 - 0

Fax: +49 (0) 731 - 1 84 78 - 299

(Office hours: Monday - Friday, 9am - 12am and 2pm - 5pm)

For the USA, Canada & Australia:

ESU LLC

477 Knopp Drive

Muncy Pennsylvania 17756

USA

Phone +1 570-649-5048 (Sales)

Fax +1 866-591-6440

(Office hours: Monday through Friday, 9AM to 5PM (EST))

Technical support

For Germany and Europe:

Fax: +49 (0) 731 - 1 84 78 - 299

For technical questions please also contact our support web forum!

Register here:<http://www.esu.eu/en/forum/>

For USA, Canada & Australia:

Monday, Wednesday & Friday 9am - 5pm (EST)

Phone: +1 570-649-5046

Fax: +1 866-591-6440

About this manual:

This is an Unofficial Manual, it has not been reviewed by ESU engineering. However, it has been given a thorough scrub by the writer and several volunteers that have a high level of experience with ESU v3.0, v3.5, and v4.0 series decoders.

I would like to thank the proofreaders for their input, and for their diligence and expertise. These folks are familiar contributors to the LokSound Community. Thank You!

Peter Ross

Mike Walters

Ian Bishop

Dave Heap

Jim Albanowski

Ted Najzer

Mark Roach

Special Thanks to Matthew (Matt) Herman, for his support and enthusiasm given to the ESU user community and the v4.0 decoder!!!

Phil Dunlop

