

# SmileSound ユーザーズマニュアル

Rev 0.43 / 20231217

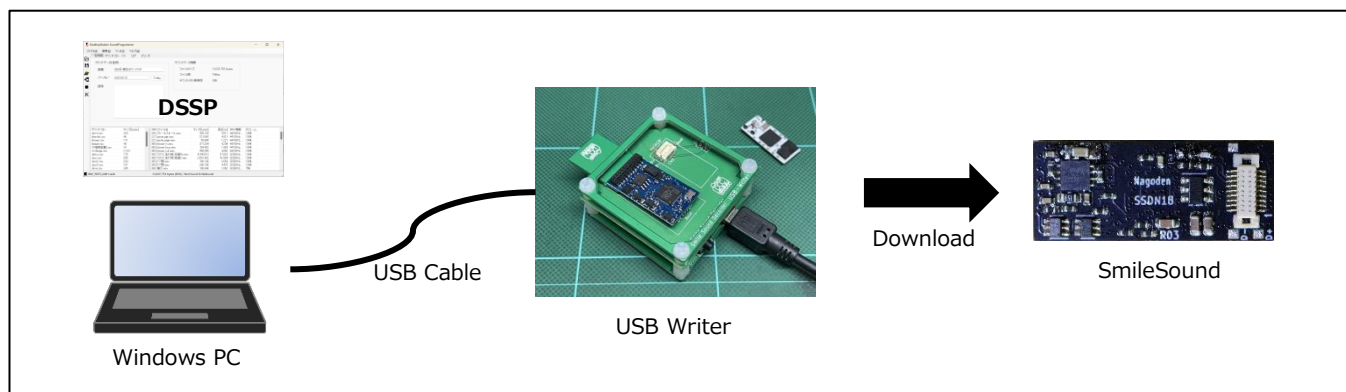
## 要旨

オープンな DCC サウンドの実現するため、デスクトップステーション株式会社と DCC 電子工作連合では共同で SmileSound と呼称する NMRA 規格準拠の DCC サウンドデコーダを開発し提供しています。このサウンドデコーダは日本で設計されており、日本の鉄道サウンドをユーザーが自由にプログラミングし、変更できるように考慮されたものです。海外メーカーのサウンドデコーダに比べて、日本のニーズに沿った機能・性能を取り込み、差別化を図っています。

開発途上のため、サポートに関する至らぬ点やバグや機能制限などございます。随時、改善活動を行ってまいります。

本書では、SmileSound を利用するために必要な、デコーダの取り扱い方法やソフトウェアの使用方法、サウンドデータの作成までを紹介しています。

ご不明点などありましたら、デジタル鉄道模型フォーラム(<https://desktopstation.net/bb/>)に投稿いただければ、本マニュアルの改訂時に内容を反映し、よりよいマニュアルにアップデートをしてまいります。



## 本マニュアルで対応する製品

- SmileSound Mini Next18 (開発元 Nagoden/DesktopStation)
- SmileSound Standard MTC21 (開発元 DesktopStation)
- DSSP (開発元 DesktopStation)
- USBライター(開発元 SmileWorks/DesktopStation)

## 目次

要旨.....	1
目次.....	2
1. はじめに.....	5
2. 注意事項・禁止事項 .....	6
3. 保証規定.....	7
4. 使用ハードウェア・ソフトウェア .....	9
5. SmileSound デコーダ .....	11
5.1. SmileSound デコーダの紹介 .....	11
5.2. 仕様.....	13
5.3. サポート CV .....	13
5.4. DCC コネクタ .....	14
5.4.1. DCC コネクタとは .....	14
5.4.2. Next18 .....	14
5.4.3. MTC21 .....	15
5.4.4. PluX .....	15
5.4.5. NEM651 6 ピン .....	16
5.4.6. NEM652 NMRA8 ピン .....	16
5.5. デコーダの説明 .....	17
5.5.1. SmileSound Standard MTC21 .....	17
5.5.2. SmileSound Mini Next18 .....	18
5.5.3. SmileSound Slim USB .....	18
5.5.4. SmileSound Sound Development Board .....	19
5.6. 集電対策・トマランコンデンサ .....	20
5.7. 車両への搭載 .....	21
5.8. デコーダの絶縁対策.....	22
5.9. スピーカーの絶縁対策.....	23
6. USB ライター.....	24
6.1. USB ライターの紹介.....	24
6.2. USB ライターの使い方 .....	24
6.3. DSSP を用いた SmileSound へのサウンドデータの書き込み .....	24
6.4. DSSP を用いた SmileSound へのファームウェアの書き込み .....	25
7. サウンドプログラマ DSSP .....	27
7.1. DSSP とは.....	27
7.2. DSSP のダウンロード、インストール .....	27
7.3. DSSP の起動と簡単な説明.....	29



7.4.	一般情報 .....	30
7.5.	サウンドフロー .....	31
7.6.	音データ .....	31
7.7.	ログ .....	32
7.8.	ファームウェア .....	32
8.	サウンドフロー .....	33
8.1.	はじめに .....	33
8.2.	サウンドフローで使用できるファイル形式 .....	33
8.3.	画面の説明 .....	33
8.4.	作成・実行ルール .....	34
8.5.	編集方法・動作原理 .....	35
8.6.	コマンドの追加・編集・削除 .....	35
8.7.	ラベルと条件分岐 .....	36
8.8.	フローの終わり方 .....	37
8.9.	変数の一覧 .....	37
8.10.	コマンドの一覧 .....	38
8.11.	主なコマンドの使用法 .....	39
8.11.1.	aux .....	39
8.11.2.	auxs .....	39
8.11.3.	call .....	40
8.11.4.	cxif .....	40
8.11.5.	date .....	40
8.11.6.	dirx .....	41
8.11.7.	echo .....	41
8.11.8.	emg .....	41
8.11.9.	exit .....	42
8.11.10.	if .....	42
8.11.11.	goto .....	42
8.11.12.	label .....	43
8.11.13.	let .....	43
8.11.14.	monf .....	44
8.11.15.	play .....	44
8.11.16.	plyx .....	45
8.11.17.	pit .....	45
8.11.18.	ret .....	46
8.11.19.	set .....	46
8.11.20.	spdx .....	47

8.11.21. sply .....	47
8.11.22. stm .....	48
8.11.23. stmc.....	48
8.11.24. stms.....	48
8.11.25. stop.....	49
8.11.26. slim .....	49
8.11.27. vol.....	50
8.11.28. volm.....	50
8.11.29. wait.....	50
8.11.30. wrnd .....	50
8.11.31. wspd .....	51
8.11.32. xif .....	51
9. CV 設定 .....	53
9.1. CV とは .....	53
9.2. モータタブ .....	53
9.3. 始動パルスアシスト機能 .....	54
9.4. 運転タブ .....	56
9.5. サウンドタブ .....	56
9.6. 速度カーブタブ .....	57
9.7. アドレス他タブ .....	58
9.8. AUX タブ .....	58
9.9. Steam タブ .....	58
10. サポート .....	60
10.1. 動作保証の環境条件.....	60
10.2. サポート体制について.....	61
10.3. 質問・回答 .....	62
10.3.1. CV 一覧.....	63
10.3.2. 参考文献 .....	64
10.3.3. 謝辞 .....	65

## 1.はじめに

スマイルサウンドデコーダは、日本国内で設計・開発され、全世界で共通で使用可能な NMRA DCC 規格に準拠した車載用の DCC サウンドデコーダです。

サウンドフローと呼ばれるスクリプトを用いることによって、車両の挙動やライト等の制御も含めたサウンドプログラミングの記述を行えます。同時並列にサウンドフローを実行可能な独自開発のインタプリタ・サウンドエンジンを備えることにより、様々な鉄道車両の挙動や表現を実現でき、表現豊かなサウンドモデル化に貢献します。

また、車両だけでなく、建物等のストラクチャーで LED や音の動きの表現も行えるように、DCC なしでもサウンドフローが動作する仕組みを導入しています。SmileSound は、鉄道模型車両だけでなく、レイアウト、ジオラマ等に幅広く、使用いただけます。

以下の特徴を有します。

- RP2040 と 16MB(128Mbit)の大容量・高速 FLASH メモリを活用したサウンドデコーダ
- 平易でシンプルかつ高機能なインタプリタエンジンを搭載し、最大16 点のユーザーカスタムプログラムを同時実行可能。状態遷移を表現可能なサウンドプログラミングを実現
- 最大同時発音数 10 チャンネル, 32kHz 16bit の再生に対応。省メモリ対応に貢献する 16kHz 及び 8kHz のサンプルレート及び 8bit音声もサポート。
- RailCom(BiDi)等、ワールドワイドで普及が進む DCC 関連技術の標準搭載
- 専用アダプタを介して USB 経由による高速ファームウェア・サウンドデータの更新
- オープンサウンドデータの膨大なサウンドライブラリを移植可能な機能・パフォーマンスの実現
- アナログ制御には未対応
- DesktopStation 製以外の他社コマンドステーションにも対応

オープンサウンドデータ(<https://desktopstation.net/sounds/>)と組み合わせて使用いただくことを前提としており、100 種類以上のサウンドデータをいつでも気軽に、SmileSound に書き込んで使用することができます。

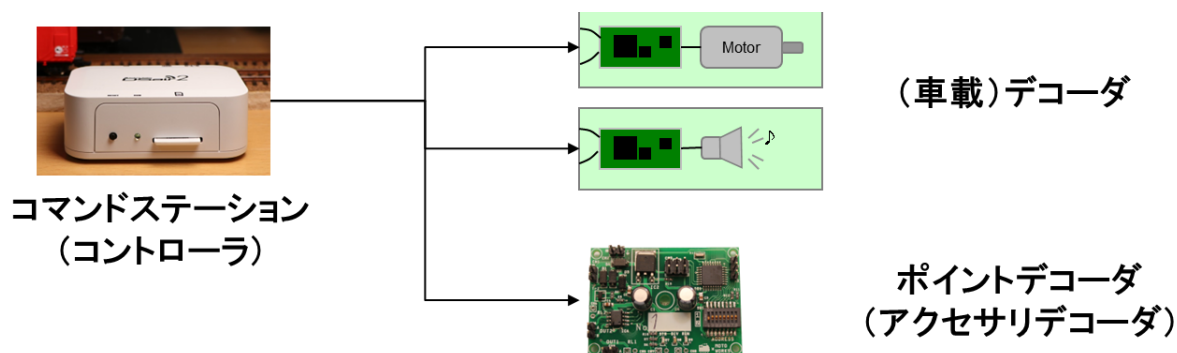


図 1.1 SmileSound の位置づけ

## 2.注意事項・禁止事項

SmileSound は、以下に示す注意事項・禁止事項を守り、適切に利用してください。以下の事項を守らない場合、ケガや健康を害するリスクがあります。使用についての責任は、利用者または機器の管理者が負います。

	屋外での使用、化学薬品類、液体類、多湿、油分、粉塵、密閉、引火性・可燃性の物質のある環境、高温・極低温環境では使用しない
	医療機器やその他の人体に影響を及ぼすリスクのある機器を使用する環境で使用しないこと。DCC に含まれる高調波ノイズによって誤動作の恐れあり
	搭載にあたっては、電線・導電部や露出部の絶縁を行うこと。不十分な絶縁作業により、車両の破壊・損傷・故障を誘発する場合があります。
	ショート、感電事故、発煙、コンデンサの破裂の恐れあり。通電中はレール、配線、車両の導電部・車輪等には触れないこと。
	Digital Command Control(DCC) 仕様準拠の製品のみと使用すること
	使用年齢 13 歳以上。幼児・児童の使用は、保護者の監督・責任のもとで使用すること
	異音、異臭、発煙時は、直ちに使用を中止し、メーカーへ修理を依頼すること。継続使用禁止。
	使用電圧・電流範囲を順守すること。DC12-18V の範囲で使用すること。日本国内での使用が認められている PSE マークの付いた正規の電源アダプタを使用すること。スケール・車両・デコーダの仕様に適合した電圧の電源アダプタを使用すること。
	電源投入中/使用中は、常に使用者が監視すること。電源の投入・未投入にかかわらず、AC アダプタを接続したままコマンドステーションを放置すること、無人運転、無人操作の禁止。 機器から離れる際には、AC アダプタをコンセントから外し、コマンドステーションの電源が入らない状態、車両への電力供給が行えない状態にすること。
	業務・産業などでの長時間・長期間の連続稼働・高耐久性や安定稼働が要求される用途、航空・宇宙・軍事などでの利用は禁止。

### 3.保証規定

【SmileSound デコーダに関する保証規定をここに記載しています】

#### 1. 保証範囲

当社は、この書面に記載された製品について保証します。

#### 2. 保証期間

ユーザーの購入日より、保証期間は1年間とします。また、有償修理・有償交換は、購入日より、3年間とします。

#### 3. 保証内容

この書面に記載された作業内容について、保証期間内に当社の責に帰すべき瑕疵により不具合が発生した場合は、代品との交換または補修を無料で行います。保証期間を経過した場合は、有償になります。

該当製品の販売が終了した場合には、代替製品に変えさせていただく場合がございます。

保証を受ける場合には、購入した店舗にお問い合わせください。デスクトップステーションオンラインストアで購入したものは、デスクトップステーションオンラインストアにて承ります。

#### 4. 有償修理・有償交換

保証期間内であっても、次のような事項に該当する場合は、有償修理または有償交換となります。

- ・ 購入場所及び購入日を証明する情報(注文メール,注文番号,納品書,領収書等)の提示がない場合
- ・ ユーザーまたは加工業者の取り付け作業に起因する不具合、故障（例えば、搭載中の絶縁不良等によるショート故障,誤った配線での故障,脱線や導電物の接触等）
- ・ 表示された商品の性能を超えた用途に使用された場合の不具合(例えば、HO 向け製品を G ゲージや 1 番ゲージなどに使用)
- ・ 商品または部品の経年変化（使用に伴う消耗、摩耗など）や経年劣化またはこれらに伴うその他の不具合
- ・ 保管場所・搭載場所の環境に起因する要因。埃、髪・ペットの毛、粉塵、高温多湿、結露、腐食またはその他の不具合
- ・ 商品または部品の材料特性に伴う仕様（基板端面の処理、コネクタのなど）
- ・ 天災その他の不可抗力（例えば、暴風、豪雨、高潮、地震、落雷、洪水、地盤沈下、火災など）による不具合またはこれらによって商品の性能を超える事態が発生した場合の不具合
- ・ 操作の誤り、調整不備または適切な維持管理を行わなかったことによる不具合（例えば、車輪やレールの清掃、コマンドステーションのメンテナンスなど）
- ・ ユーザー自身の取り付け、修理、改造（必要部品の取り付け・取外しを含む）に起因する不具合
- ・ 長時間または長期間・連続稼働・頻繁な停止・走行といったシビアコンディションでの稼働が要求される業務用途（例えば、博物館のジオラマ）または産業用途（工場、博物館、ミュージアム、イベント会場）への利用。



## 5. 修理・交換の対象

次のような事項に該当する場合は、保証・有償修理または有償交換が受けられません。

- ・ 購入店舗以外で保証・有償修理を受けようとする場合
- ・ 犯罪などの不法な行為に入手された場合
- ・ 弊社または弊社代理店以外の手段で購入・受領された場合
- ・ 競合企業または個人が、製品の分析のために購入・改造された場合
- ・ 日本国内向け製品を、海外で使用されている場合
- ・ 当社が提供するファームウェア以外を使用された場合
- ・ 弊社製品の模造品、弊社の許可なく改造された品
- ・ 弊社と間接・直接を問わず有償の業務サポート契約なく、業務用途または産業用途へ利用された場合。

## 6. 保証規定の改訂

本保証規定は、予告なく改訂する場合があります。



## 4.使用ハードウェア・ソフトウェア

本マニュアルで使用を想定しているハードウェア、ソフトウェアを以下に示します。

表 4.1 使用ハードウェア一覧

ハードウェア	
パソコン(Windows10 以降)  ※Windows ARM 版や、Mac のエミュレーション機能での動作保証は致しません。	
USB ケーブル(A-ミニ B)	
USB ライター(75018)	
デコーダテスター(LaisDcc860033, ESU53900)  ESU または LaisDcc のどちらか。自作品のサポートは致しません。自作品を使用される場合には、サポートが受けられませんので自己責任で使用ください。	
SmileSound デコーダ Mini Next18(75019)または Standard MTC21(75017)	 
DSair2(コマンドステーション)  ※動作チェック用などとして使用	

表 4.2 使用ソフトウェア一覧

ソフトウェア	備考
DSSP(7章)	SmileSound への書き込み、サウンドデータ編集に使用  【入手先】 <a href="https://desktopstation.net/smilesound/index.php?SoundProgrammer">https://desktopstation.net/smilesound/index.php?SoundProgrammer</a>
WAVE ファイル編集ツール	Audacity 等  【入手先】 <a href="https://www.audacityteam.org/">https://www.audacityteam.org/</a>
テキストエディタ	サクラエディタ, VisualStudioCode 等  【入手先】 <a href="https://sakura-editor.github.io/">https://sakura-editor.github.io/</a>

## 5. SmileSound デコーダ

### 5.1. SmileSound デコーダの紹介

SmileSound は、2019 年に発足した日本型サウンドデータ公開プロジェクトである「オープンサウンドデータ」に最適な DCC サウンドデコーダのあるべき姿として、開発が行われている日本設計のサウンドデコーダです。2021 年末から開発を開始し、2023 年に販売を開始しました。オープンサウンドデータに適合しやすい機能や仕様を 1 からソフトウェア・ハードウェアで開発を行い、海外のサウンドデコーダに頼らずに、日本国内だけで DCC サウンドを実現しています。

SmileSound に対応したデコーダは、現在、以下の種類を提供しています。これらの種類の大きな差異は、表 5.1.1 に示すように基板形状と AUX、モータ出力にあります。これは、スケールの大きさや用途に起因するものです。

- 75017 SmileSound Mini Next18  
N ゲージ向けに開発された小型のサウンドデコーダ
- 75019 SmileSound Standard MTC21  
HO 向けに開発された、汎用のサウンドデコーダ
- 75022 SmileSound Sound Development Board  
サウンドデータの開発用の基板。PC と直接接続が可能。
- 75026 SmileSound Slim USB  
室内灯一体型の N ゲージ（主に GreenMax 社製車両を念頭）・HO 向けのサウンドデコーダ。

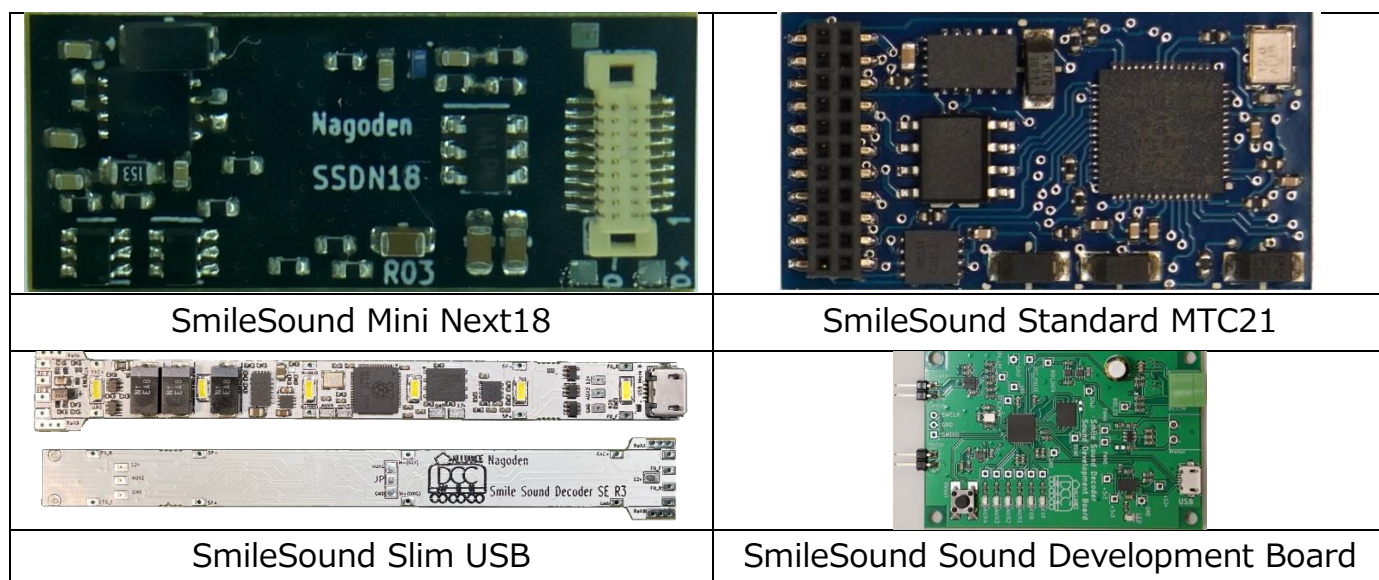


Figure 5.1.1 SmileSound decoder series

表 5.1.1 SmileSound デコーダ間の機能差異

項目	Mini Next18	Standard MTC21	Slim USB
対応コネクタ	Next18	MTC21	USB
モータ出力	瞬時 0.7Amax, 連続 0.5A	瞬時 1.5Amax, 連続 1.0A	瞬時 1.5Amax, 連続 1.0A
使用可能パッド	なし	あり	あり
サイズ	10.5x25mm	16x30mm	100x13x3.6mm
使用電圧	DC12V~16V, 最大 21V まで		
AUX 使用数	AUX4 まで	AUX6 まで	AUX2 まで
音量	小	大	小
コンデンサ搭載量	小	約 50uF	300uF
外付コンデンサ要求仕様	電解コンデンサまたは高分子コンデンサを指定。セラミックコンデンサは DC バイアス特性による容量低減があり、非推奨。		300uF の高分子コンデンサを基板上に搭載
外付コンデンサ要求容量	外付けで最低 200uF。推奨 400uF	外付けで 100-200uF を推奨。	原則不要

## 5.2.仕様

SmileSound がサポートする機能仕様を以下に示します。

表 5.2.1 SmileSound の仕様 (共通)

対応プロトコル	NMRA DCC (with RailCom Cutout)
SpeedStep	14, 28, 128
Function	F0-F32, 2022 年新仕様に準拠
アシンメトリーDCC	アシンメトリーDCC 未対応(自動ブレーキ機能未対応)
CV 方式	Direct, OPS/ POM(RailCom 連動機能は今後対応計画中)
モータ PWM	32kHz(設定変更可能)
線路電圧	12V~16V(推奨), 最大 21V まで
AUX 出力	<b>ヘッドライト・テールライト,AUX1-2:</b> Power, オープンコレクタ出力, 100mAmax <b>AUX3-6(Mini Next18 は AUX3-4):</b> Logic, 3.3V CMOS 出力。ライト等を繋げる場合はパワー出力に変換する必要があります。
サウンドメモリ	120Mbit(15MB)
サウンド出力仕様	32/16/8kHz 16bit/8bit.
スピーカー容量	3Wmax, 4-32Ω
サウンド同時発音数	12 音(内 2 音は蒸気サウンド専用)
サウンドのユーザー変更	専用ツールにてサポート。独自スクリプト方式。
ファームウェア容量	8Mbit(1MB)
アナログ運転	未サポート (対応予定なし)
サーボ機能	対応検討中
SUSI	未サポート(対応予定なし)
RailCom(BiDi)	サポート
安全・保護機能	<ul style="list-style-type: none"><li>・モータ出力保護(過熱,過電流,過電圧,低電圧)</li><li>・スピーカー出力保護(スピーカー配線間ショート,出力~GND 間ショートのみ)</li><li>・電源保護(過熱,低電圧,ソフトスタート)</li></ul>

## 5.3.サポート CV

63 ページを参照してください。

## 5.4.DCC コネクタ

### 5.4.1.DCC コネクタとは

SmileSound では、Next18 と MTC21 の2つの DCC コネクタに対応しています。国内の模型メーカーでは採用がほとんどされていませんが、海外では多く利用されています。

SmileSound Mini Next18 は、名前の通り、Next18 コネクタに対応しています。Standard MTC21 は、同様に MTC21 コネクタに対応しています。このほかに、NEM652(NMRA8 ピン)、PluX 等の規格があります。

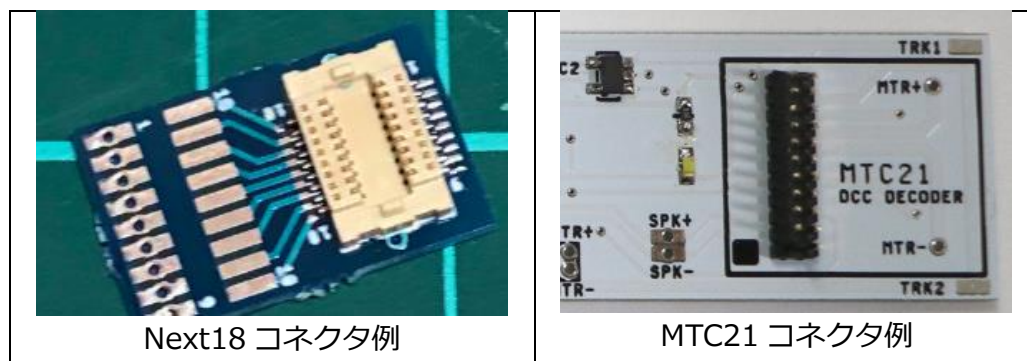


図 5.4.1.1 SmileSound 対応コネクタ例

### 5.4.2.Next18

Next18 コネクタは、主に N ゲージ向けに使用されている DCC コネクタです。基板対基板コネクタを採用しており、逆挿ししても、故障などが起きないようにピン配列は工夫されています。

日本の鉄道模型では、採用実績はありませんが、欧州の模型メーカーを中心に広く採用されています。TRAINO(<https://traino.jpn.org/>)から、日本の鉄道模型車両向けの搭載補助基板が販売されていますので、これらを利用することにより、N ゲージの DCC 化については比較的敷居は低くなっています。

以下に、Next18 コネクタのピン配列を示します。

表 5.4.2.1 Next18-S 規格のピン割り当て

割当端子	ピン番号		割当端子
線路 A	9	10	線路 A
ヘッドライト(P)	8	11	モータ出力-
スピーカー+	7	12	AUX2(P)
COM+	6	13	AUX4(L)
GND	5	14	GND
AUX3(L)	4	15	COM+
AUX1(P)	3	16	スピーカー-
モータ出力+	2	17	テールライト(P)
線路 B	1	18	線路 B

(P): パワー出力(オープンコレクタ), (L):ロジック出力 CMOS 3.3V

### 5.4.3.MTC21

MTC21 コネクタは、主に HO・16 番向けに利用されている DCC コネクタです。コネクタの特徴としては、汎用の 2x11 の 1.27mm ピッチピンヘッダーを使用していますが、INDEX ピンと呼ばれる 11 番目のピンが使用できない代わりに、目印として装着が可能となっていることが挙げられます。また、デコーダ側はピンフレームと基板が一体化したような構造を取っており、図 5.4.3.1 に示すように上から基板上の穴を通じて差し込む構成をしています。

Next18 コネクタよりも太いピンを使用しているため、HO の大電流に耐えることができます。

表 5.4.3.1 MTC21 規格のピン割り当て

割当端子	ピン番号		割当端子
センサ 1/AUX7(L)	1	22	線路 A
センサ 2/AUX8(L)	2	21	線路 B
AUX6(L)	3	20	GND
AUX4(L)	4	19	モータ出力+
—	5	18	モータ出力-
—	6	17	AUX5(L)
テールライト(P)	7	16	COM+
ヘッドライト(P)	8	15	AUX1(P)
スピーカー+	9	14	AUX2(P)
スピーカー-	10	13	AUX3(L)
ピンなし	INDEX	12	VCC(3.3V)

(P): パワー出力(オープンコレクタ), (L):ロジック出力 CMOS 3.3V

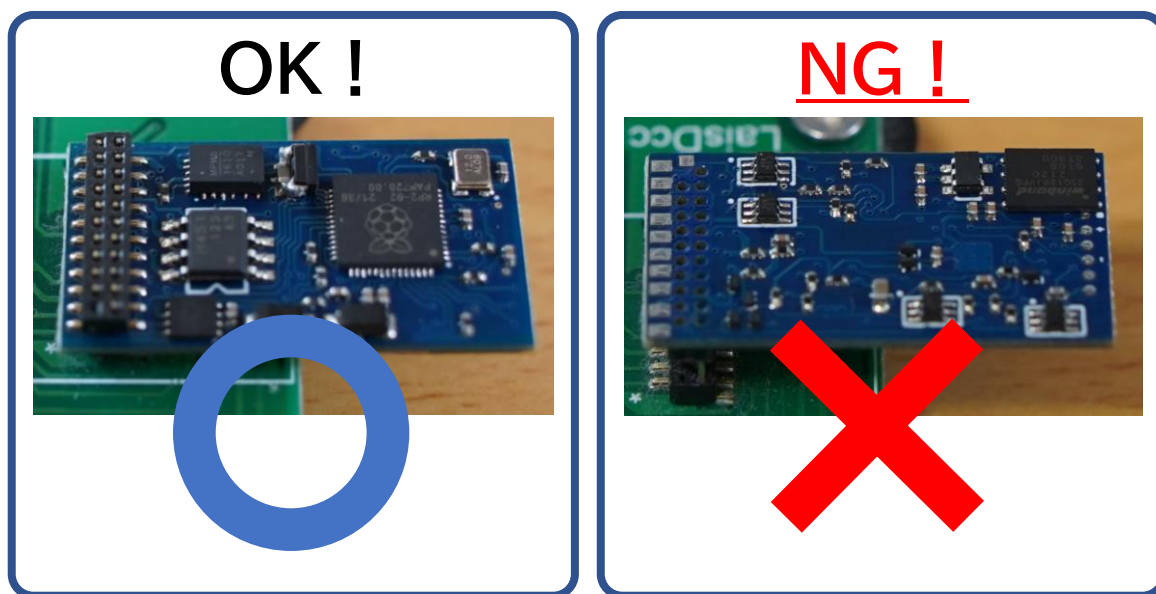


図 5.4.3.1 MTC21 コネクタの使い方

### 5.4.4.Plux

欧州の車両が対応している Plux 規格への変換は、構造や配線が複雑化するため非推奨とさせていただきます。N ゲージ向けの規格と、HO 向けの規格でコネクタが共有された設計となっています。

### 5.4.5.NEM651 6 ピン

欧州の N ゲージ車両の一部が、1.27mm ピッチ 6 ピンのコネクタを持っています。SmileSound では、未対応のため、ご自分で Next18 から 6 ピンに変換するアダプタを作成し、ご利用ください。

### 5.4.6.NEM652 NMRA8 ピン

KATO の HO プラ、天賞堂やトラムウェイの一部の車両には NEM652 (NMRA8 ピン) の DCC コネクタ・ソケットが搭載されております。基本的に HO(16 番)向けのコネクタとなりますので、Standard MTC21 を前提に変換方法を説明します。

#### ・ NEM652-MTC21 アダプタを使用する

860046 MTC21・NEM652 変換ケーブルを使用してください。容易に NEM652 化できます。ただし、体積を消費するので、搭載スペースは確保をお願いします。LaisDcc または、デスクトップステーションオンラインストアにて購入ができます。類似商品は、欧米のメーカー各社からも販売されています。また、自分で自作することもできます。



図 5.4.6.1 Lais860046 MTC21・NEM652 変換ケーブル

#### ・ 10047/10050 ExpBoard M21 SuperShort 幅狭版(実装済)を使用して自分で NEM652 化する

MTC21 引き出し基板を使い、ここから NEM652 の配線を引き出します。なお、NEM652 化することを想定していませんので、配線がごちゃごちゃしやすい欠点があります。

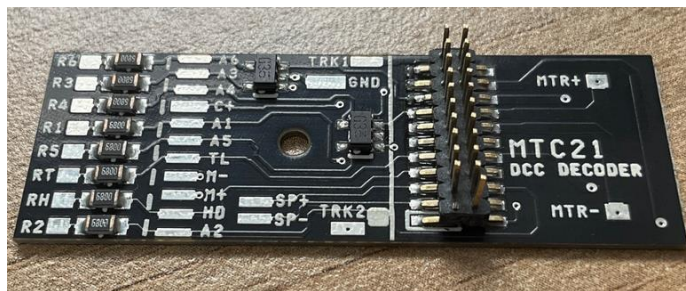


図 5.4.6.2 10047/10050 ExpBoard M21 Super Short 幅狭版

#### ・ MTC21 のコネクタを使わずに、デコーダから直接配線を引き出す

NEM652 への改造には、自分で AWG32 の電線やコネクタなどを用意し、配線することもできます。860047 NEM652 プラグ・ケーブル付 2 本や、860006 NEM652 プラグのみ 4 個を使うことで、入手が難しい部品集めや検討を省くこともできます。



## 5.5.デコーダの説明

### 5.5.1. SmileSound Standard MTC21

HO ゲージ向けを想定した、サウンドデコーダです。MTC21 規格に準拠しており、1.5Amax のモータ出力が可能です。大型のスピーカーを搭載できるスペースが確保できることから、N ゲージ向けのデコーダよりも、大音量が出せるようにアンプ回路を調整しています。

SmileSound Standard MTC21 は、ユーザーが使用可能なパッド（はんだ付け箇所）を用意しています。Mini Next18 には、ユーザーが使用可能なパッドはありません。図 5.5.1 に、ユーザーが使用可能なパッドを示します。

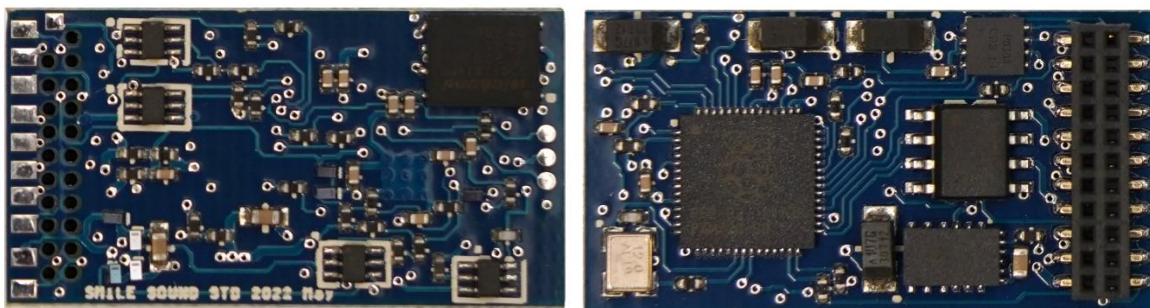


図 5.5.1.1 SmileSound Standard MTC21 の外観

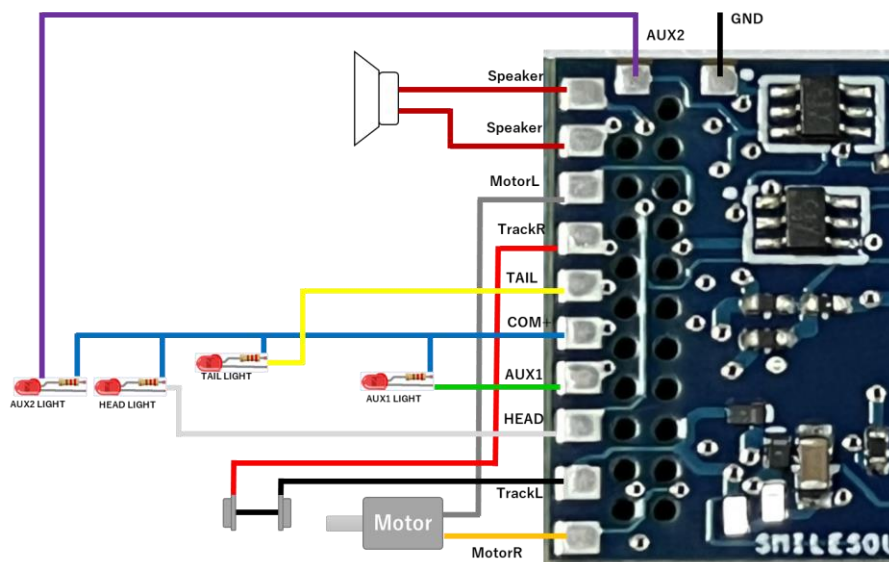


図 5.5.1.2 Standard MTC21 のパッドからの配線引き出し方法

表 5.5.1.1 Standard MTC21 のパッド

パッド名	パッドの説明
Speaker	スピーカー。線は 2 本あります。スピーカー 1 個で使用する場合、極性は関係ありません。2 個以上で使用する場合は音の位相に注意してください。
MotorR/L	モータへの配線。線は 2 本あり、極性を持ちます。逆配線にするとモータが逆回転します。
TAIL	テールライトへの配線(LED の場合はカソード側), (パワー出力)
HEAD	ヘッドライトへの配線汎用端子(パワー出力)
COM+	電源共通端子。ライトはアノード側に配線。
AUX1	汎用端子,主に室内灯(パワー出力)。オープンコレクタ出力
AUX2	汎用端子(パワー出力)。オープンコレクタ出力

TrackR/L	線路側の配線
GND	GND端子。トマランコンデンサなどに使用。

### 5.5.2. SmileSound Mini Next18

SmileSound Mini Next18 は、Nゲージ車両に使用することを想定とした汎用の小型サウンドデコーダです。Next18規格に適合しており、全世界のNext18規格対応車両に使用できます。

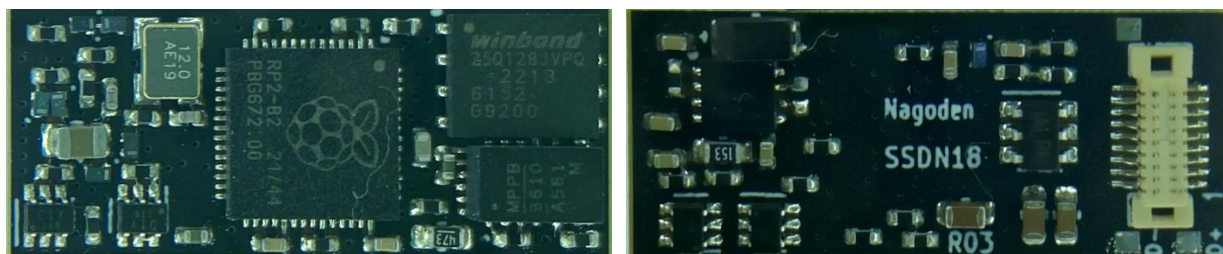


図 5.5.2.1 SmileSound mini Next18 の外観

### 5.5.3. SmileSound Slim USB

SmileSound Slim USB は、室内灯一体型となっており、GreenMax社のNゲージや、小型のHO・ナローに好適なサウンドデコーダです。基板上的パッドから、線を引き出して、車両のモータや線路からの配線に接続して使用します。単行車両向けに、ヘッドライト・テールライトのパッドも用意されています。

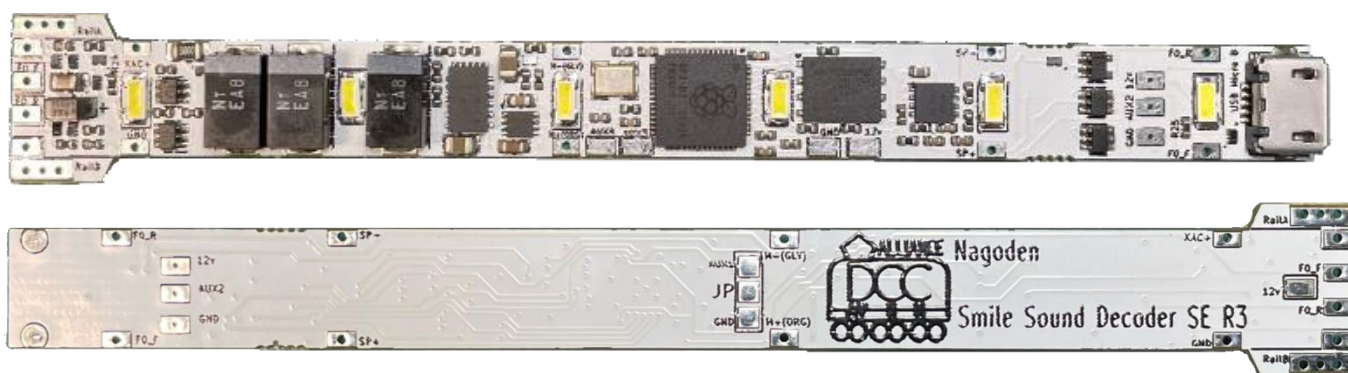


図 5.5.3.1 SmileSound Slim USB の外観

表 5.5.3.1 SmileSound Slim USB のパッド説明

パッド表記	信号の内容
RailA, RailB	線路からの配線
F0 F	ヘッドライト
F0 R	テールライト
12V	COM+またはC+と同様。共通電源。
KAC+	トマランコンデンサ接続パッド(+側)
GND	グラウンド
AUX2	
M+/M-	モータ出力

### 5.5.4. SmileSound Sound Development Board

SmileSound Sound Development Board は、PC との接続インターフェースである USB コネクタを有しており、簡単に PC と接続して、ファームウェアの書き換えやサウンドデータの書き込みが行えます。サウンドデータの作成用途に好適です。

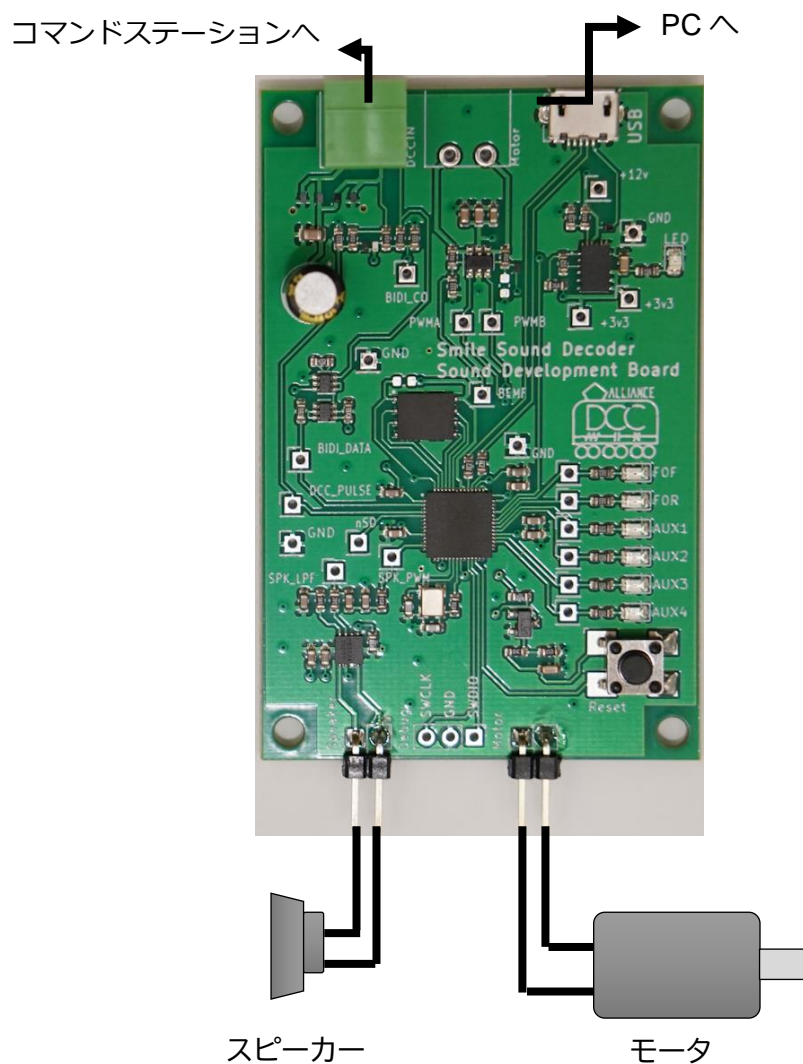


図 5.5.4.1 SmileSound Sound Development Board

## 5.6.集電対策・トマランコンデンサ

SmileSound デコーダで使用している電子部品は消費電力が大きいため、集電不良に対して若干劣る課題があります。このため、デコーダには外付けで電解コンデンサを取り付けることを推奨しています。コンデンサに加えて、全車集電（複数の車両で集電し、車両間に配線を設ける）ことも改善効果がより増大します。例えば、博物館のジオラマで走行する鉄道模型車両は、基本的に全車集電となっています。

コンデンサの要求としては、耐圧 25V の電解コンデンサを使用し、SmileSound mini Next18 で最低 200uF、SmileSound Standard MTC21 で最低 100uF を推奨しています。なお、搭載する車両の集電性能・集電軸数・ゴムタイヤの有無などによっても大きく変わります。この推奨値はあくまでも参考程度としてください。

たとえば、ゴムタイヤを付けている場合には、登板力は向上しますが、絶縁性のあるゴムが車輪に取り付けられているため、集電性能は落ちます。ゴムタイヤを付けた車両の場合には、コンデンサを多めにする対策が必要です。通電カプラ等で、複数の車両を使って集電を行える場合には、コンデンサを減らしても大きな問題にならない場合もあります。

なお、セラミックコンデンサをトマランコンデンサに使用する場合、注意が必要です。DC バイアスと呼ばれるコンデンサの容量が低下する現象が発生します。このため、記載された容量から 1/2~1/3 まで減少する場合があります。このため、DC バイアスが発生しない、電解コンデンサや高分子コンデンサを使用することをお勧めします。またセラミックコンデンサは内部抵抗が小さいため、突入電流が発生しやすく、トマランコンデンサ回路が無ければ、レールや車輪などが汚れやすくなる原因となります。

### トマランコンデンサ

<https://desktopstation.net/wiki/doku.php/%E3%83%88%E3%83%9E%E3%83%A9%E3%83%B3%E3%82%B3%E3%83%B3%E3%83%87%E3%83%B3%E3%82%B5>

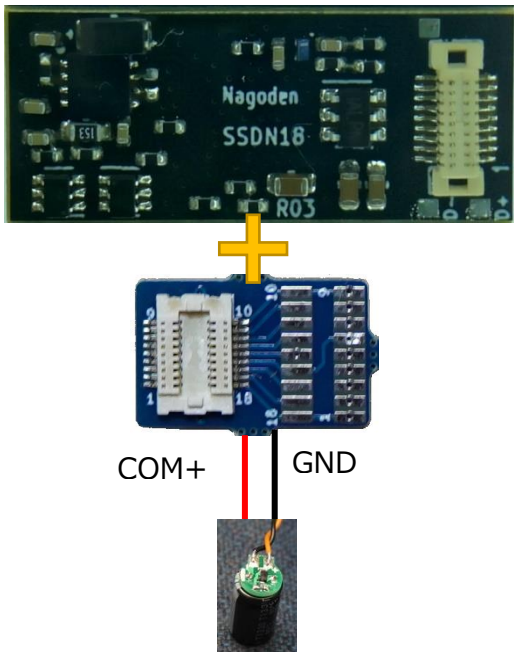
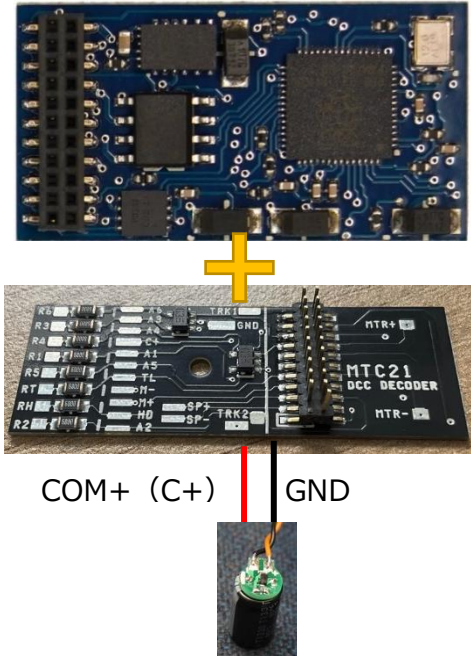
SmileSound mini Next18 の場合（例）	SmileSound Standard MTC21 の場合（例）
 <p>※200uF 程度まではトマランコンデンサ回路省略可</p>	 <p>※200uF 程度まではトマランコンデンサ回路省略可</p>

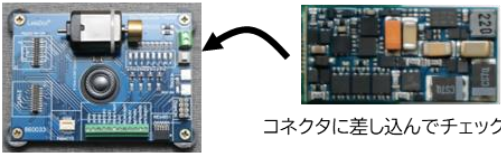
図 5.6.1 トマランコンデンサの接続例

## 5.7.車両への搭載

SmileSound を車両に搭載する方法については、別冊のマニュアルをご用意しております。搭載には、様々な注意事項がありますので、準備の上、丁寧に時間をかけて作業を行うようにしてください。

識者から、アドバイスをもらうことも大変有効です。

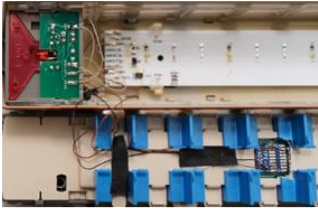
**①デコーダテストで動作チェック！**



コネクタに差し込んでチェック！


搭載前に必ずチェック！うまく動かない場合は初期不良交換してもらいましょう。

**②車両をチェック！絶縁対策！**



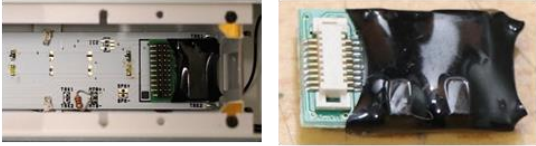
故障・焼損は絶縁不良が原因！剥出しの電線に、デコーダがズレて接触することがあります。固定、絶縁を入念に！

**③テスターで配線・回路チェック！**



電気回路テスターで、全ての配線が間違っていないかチェック！  
特に、線路からの配線を間違えて付けると、即故障です。

**④まず、安価なデコーダでチェック！**



LaisDccやLokPilotなどの安価なデコーダを搭載してチェック！CV読み書き、走行試運転を行って、搭載作業の最終確認を行いましょう。

図 5.7.1 車両搭載での注意事項

表 5.7.1 車両搭載のコツ・注意点

搭載状況を写真撮影しましょう。後で問題点が見つけれられて、次の作業に活かせます。サポートを受ける際にも、アドバイスを受けやすくなります。
常に謙虚に。どんなにチェックしても間違いは起きるという自覚を持ちましょう。
電線は細いものを選定しましょう。AWG32 の細さが目安です。 細い電線は、秋葉原のオヤイデ電気や、千石電商等で販売されています。
テスターは必ず用意しましょう。ショートテストは必ず実施！
絶縁テープ（ポリイミドテープやアセテートテープ）も用意しましょう
はんだごては、先に細いもの、太いものなど複数持ちましょう
熱収縮チューブも、いろいろな太さを用意しておきましょう。
スピーカーは固定しましょう。磁石が入っており、線路に接触してショートし、デコーダを故障させてしまう事例が多数報告されています。スピーカーの配線と線路が接触すると確実にデコーダが故障します。
デコーダテストは必ず用意しましょう。搭載前に、動作することを確認しましょう。
安価なデコーダで、搭載した車両が動くかテストしましょう。最初から高価なデコーダを使って故障させる方が多くいらっしゃいます。

## 5.8.デコーダの絶縁対策


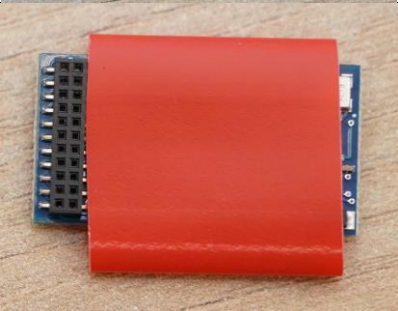
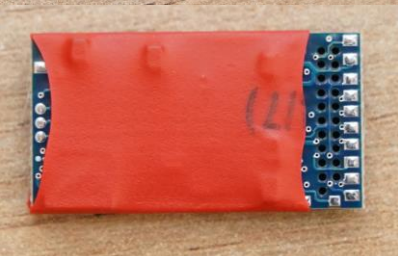
デコーダの絶縁対策は、非常に重要です。もっとも多いのは、デコーダに接続する配線を経由したショートですが、デコーダ自体に何らかの金属や導電性のものが当たって壊れることもあります。ここでは、デコーダ自体を絶縁性の部材でカバーする方法を紹介します。

デコーダ自体を絶縁部材でカバーするには、以下の方法があります。

- 熱収縮チューブを被せる
- ポリイミドテープを使って巻く
- アセテートテープを使って巻く
- セロテープを使って巻く ※見た目が悪いため非推奨
- ビニールテープを使って巻く ※ベタベタするため非推奨

もっとも効果的で見栄えが良いのは、「熱収縮チューブを被せる」方法です。以下に、電子部品店で販売されている熱収縮チューブを使った絶縁方法を紹介합니다。この方法には、工業用のドライヤーが必要です。お持ちでない場合は、はんだごてのこて先を温めて使う方法や、バーナーやライターで炙る方法もありますが、リスクもあるため非推奨です。

表 5.8.1 熱収縮チューブを被せる方法

	<p>熱収縮チューブを、デコーダのサイズにカットします。このとき、MTC21 コネクタや Next18 コネクタに被らないようにします。また、特に Standard MTC21 では、基板端にあるポゴピンを当てる丸いパッドが、USB ライターにあたるように長さを見極めてください。</p>
	<p>カットした熱収縮チューブを被せます</p>
	<p>可能であれば、ヒートガン（産業用ドライヤー）を少し遠くから熱を当てて収縮させます。 このとき、ピンセットなどでデコーダと熱収縮チューブを挟んで当てます。熱を当てた後は、非常に熱いので、火傷に気を付けましょう。</p>

## 5.9.スピーカーの絶縁対策

Nゲージなどで、よく使われているコンパクトな15x11mmのキューブスピーカーがあります。実はこのまま使うと、リスクが非常に高いのです。何が問題かと言いますと、以下の赤枠部分が、スピーカー信号が流れる導電部（電機の流れるところ）になっています。

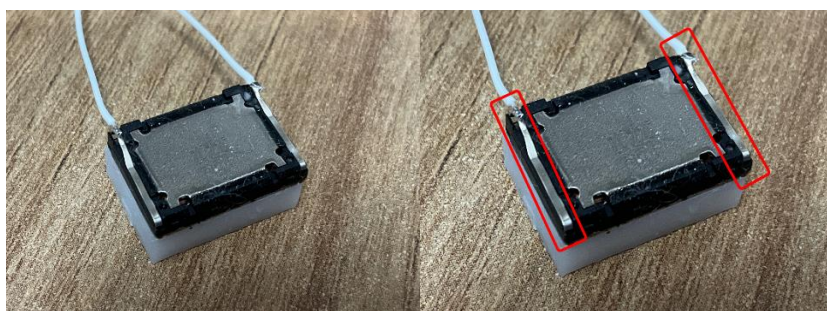


図 5.9.1 キューブスピーカーの導電部

ここは、デコーダのスピーカー端子に繋がっており、ここが線路からの配線に接触すると、デコーダに線路からの電圧が逆流するため、損傷します。これは、線路の高い電圧を入れて良い部分と、高い電圧がNGである低い電圧専用の部分が回路で決まっています。スピーカーの配線には高い電圧を出しません。また当然ですが、マイクのような機能はありませんので、何かを入力することもNGです。

線路電圧は12Vなどの高めの電圧ですので、高いところから低いところに流れる電気性質で、逆流が起きます。無理やり、スピーカーからデコーダに電圧をかけることになりますので、内部の回路が壊れます。

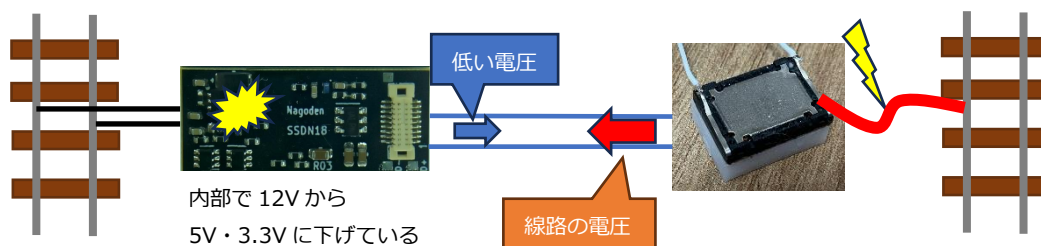


図 5.9.2 デコーダが故障するメカニズム

よって、スピーカーには、違う配線やダイキャスト・金属などが当たってショートしないように、対策が必要なのです。以下は、マスキングテープで覆った一例です。セロテープでもポリイミドテープでも構いません。カバーを付けてしまう。接着剤で固定して接触しないように動かなくするなど有効です。

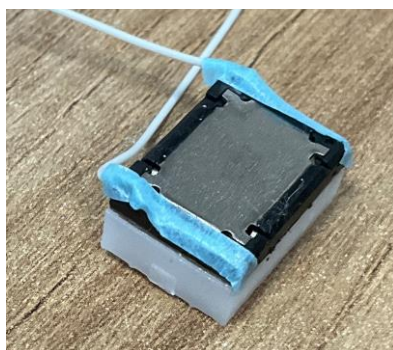


図 5.9.3 マスキングテープで絶縁対策したスピーカー

## 6.USBライター

### 6.1.USBライターの紹介

スマイルサウンドデコーダは、USB 経由での SmileSound にサウンドデータやファームウェアを書き込むためのアダプタ装置です。SmileSound 用のファームウェアとサウンドデータは、USBライターを介して書き込みを行います。

USB 配線は、ポゴピン(バネを内蔵した剣山のようなもの)を使って、USBライターとデコーダの間を接続しています。Next18のスマイルサウンドデコーダは指で簡単に取り外しできますが、MTC21のスマイルサウンドデコーダを取り外すときには、ツメやプラスチックのピン、ギター用のピック等を用いてください。この時、デコーダ基板上の部品に取り外しに使った治具の先などが直接当たらないように気を付けてください。



図 6.1.1 USBライター

### 6.2.USBライターの使い方

USBライターと、パソコンで動作するソフトウェアである DSSP を使うことで、ファームウェアの書き込みと、サウンドデータの書き込みが行えます。

SmileSound へのファームウェアやサウンドデータの書き込み方法には大きく 2 つの方法がありますが、通常は、DSSP を使った直接書き込み機能である「SmileSound デコーダにサウンドを書込む」と「ファームウェアのアップデート」の機能の利用を強く推奨します。

その他に、「サウンドデータを UF2 形式でエクスポート」を行って、サウンドデータを UF2 形式でエクスポートしたあと、手動の書き込み操作を行って書き込むこともできます。手動での書き込み手順については、SmileSound 故障時の緊急避難的な操作となり、弊社内で使用する手順となるため、本マニュアルでは紹介しません。

### 6.3.DSSP を用いた SmileSound へのサウンドデータの書き込み

後述する、7.2 を先に済ませてから、本項をお読みください。またあらかじめ、オープンサウンドデータのホームページから書き込みしたいサウンドデータをダウンロードするか、または何らかの方法でサウンドデータを入手してください。

DSSP では、スマイルサウンドデコーダを検出して、強制的にドライブを開く処理を行って、書き込みができるようにする機能が搭載されています。PC の環境や USB 機器の状況によっては、うまく行かないケースが稀に発生します。そのときは、いったん抜き差ししなおすなどの作業を行ってください。操作手順は以下の通りです。



表 6.3.1 サウンドデータの書き込み手順

手順	操作方法
1	USBライターにスマイルサウンドデコーダを装着する
2	USBライターのケーブルをパソコンに差し込む
3	DSSPを起動する。既にDSSPが起動している場合は起動する必要はありません。DSSPの一番下のバー上に「Found SmileSound on SerialPort」と表示されているのを確認する。
4	DSSPで書き込みたいサウンドデータを開く。
5	DSSPで書き込みボタンを押して書込む。
6	書き込みが終わるまで
7	書き込みが終わると、自動的にドライブが取り外しされるので、USBケーブルをパソコンから外す。
8	USBライターから、スマイルサウンドデコーダを取り外す
9	書き込み作業完了

#### 6.4.DSSP を用いた SmileSound へのファームウェアの書き込み

後述する、7.2を先に済ませてから、本項をお読みください。

DSSPを使って、お手持ちのSmileSoundデコーダの内部のソフトウェア（ファームウェア）のアップデートを行うことができます。過去の複数のバージョンをDSSP内に用意していますので、もしトラブルが発生した場合には、古いバージョンへ戻すことができます。なお、新機能が追加されている場合には、齟齬によってうまく動作しなくなる場合があります。ファームウェアは基本的に、最新版をご選択ください。

「ファームウェア」タブを選択すると、ファームウェアのアップデート操作が行えます。

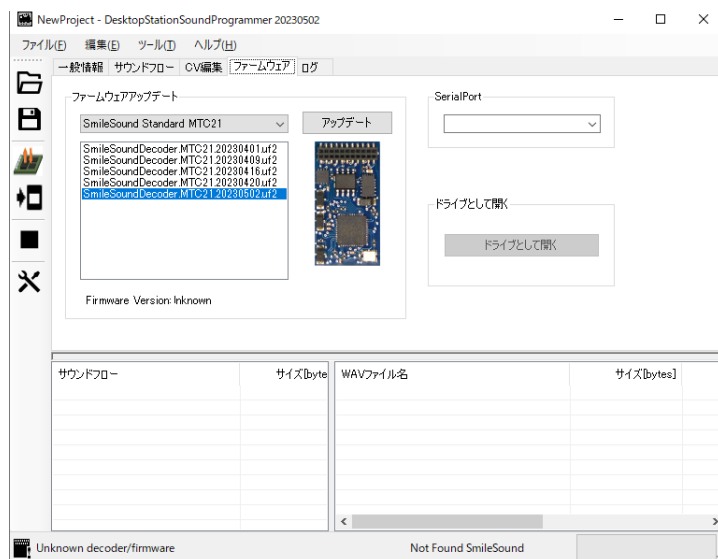


図 6.4.1 DSSP のファームウェアタブ

ファームウェアは、MTC21版とNext18版で異なります。通常、自動で選択されますが、間違

えて変更してファームウェアを書込むと、モータが動かないなどの不具合が出ますので、ご注意ください。間違えて書いてしまった場合には、手動操作でデコーダの種別をドロップダウンから変更し直して、「アップデート」ボタンを押してください。

書き込みたいファームウェアの日付（バージョン）を選択したら、「ダウンロード」ボタンを押して書き込みます。通常、最新版が自動的に選択されるようになっています。

この際、SmileSound が認識していることを確認してください。SmileSound が認識されていない場合は、うまく書き込みができない場合があります。

SmileSound を故障させた場合、多くのデコーダは DSSP で認識ができなくなります。また、PC からの電源供給でデコーダが異常発熱する場合があります。この際は、すぐに故障した SmileSound の利用を中止し、有償の故障交換サービス（購入から一定期間のみ、サービスが受けられます）を利用してください。

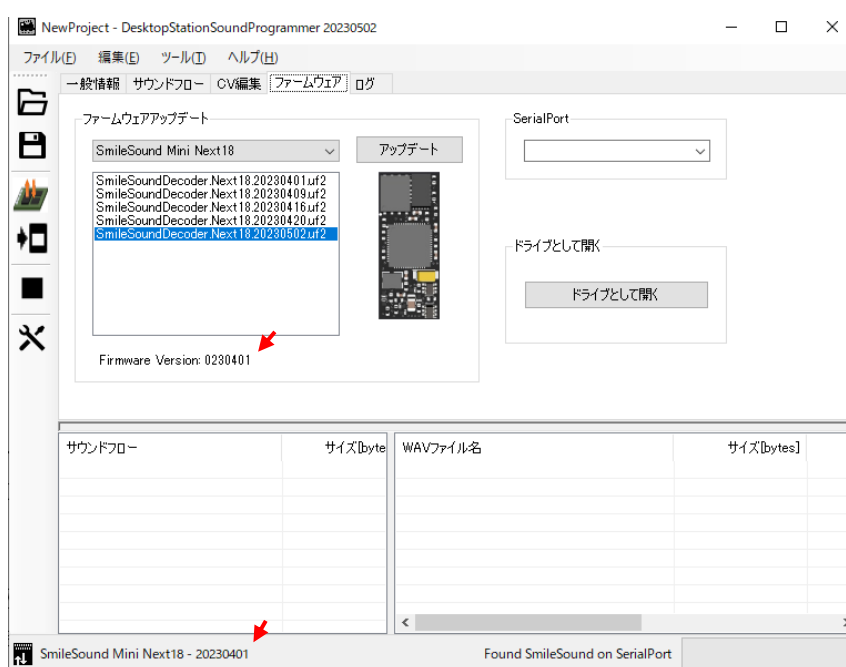


図 6.4.2 SmileSound の自動認識後の表示（バージョン情報が表示）

## 7.サウンドプログラマ DSSP

### 7.1.DSSP とは

DesktopStation SoundProgrammer(DSSP)は、SmileSound のサウンドデータを作成するための、Windows 向けのソフトウェアです。このツールを使う事で、ユーザーは SmileSound デコーダを自由にプログラミングできます。このツールで生成したサウンドデータ(ssdx ファイル)は、他のユーザーと共有することも可能です。サウンドの動きや AUX の操作、ファンクションボタンや走行スピード状態を組み合わせることで、SmileSound をサウンドデコーダとして機能させます。

主に、以下の機能を有しています。

- SmileSound 用のサウンドデータの確認、作成、編集
- SmileSound デコーダへのサウンドデータの書き込み
- SmileSound デコーダのファームウェアのアップデート
- SmileSound デコーダの CV 編集

DSSP はライセンス・規約を遵守いただくことで、どなたでも無償で使用できます。

### 7.2.DSSP のダウンロード、インストール

DSSP は、SmileSound のホームページから、DSSP の最新版がダウンロードできます。ダウンロードした DSSP は、ZIP 形式でパッケージされていますので、エクスプローラーか 7zip 等のアーカイバを使って解凍(展開)します。

#### DSSP の公開 URL

<https://desktopstation.net/smilesound/index.php?SoundProgrammer>

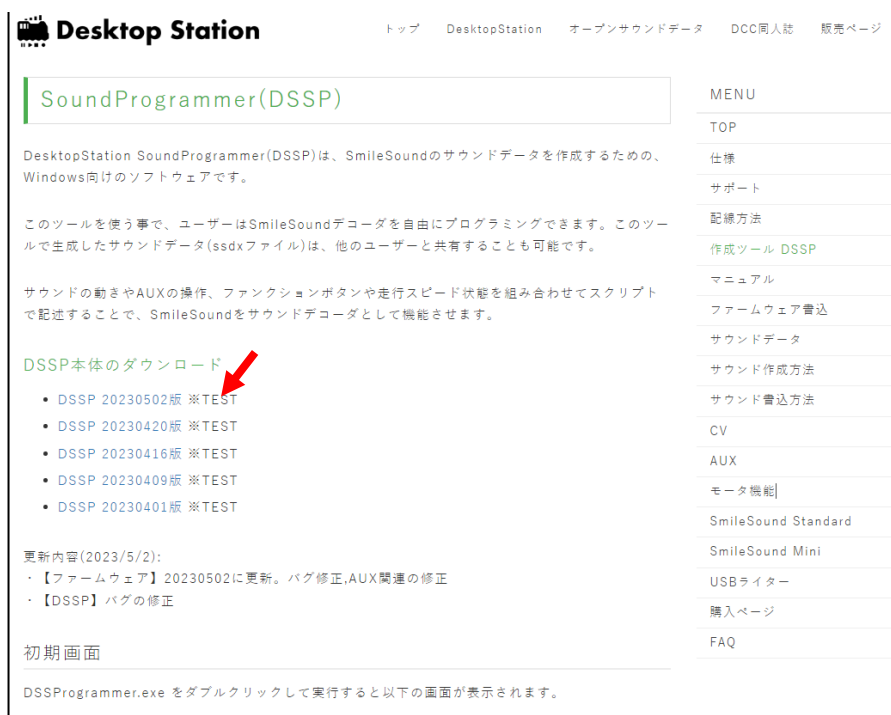


図 7.2.1 DSSP のダウンロードページ

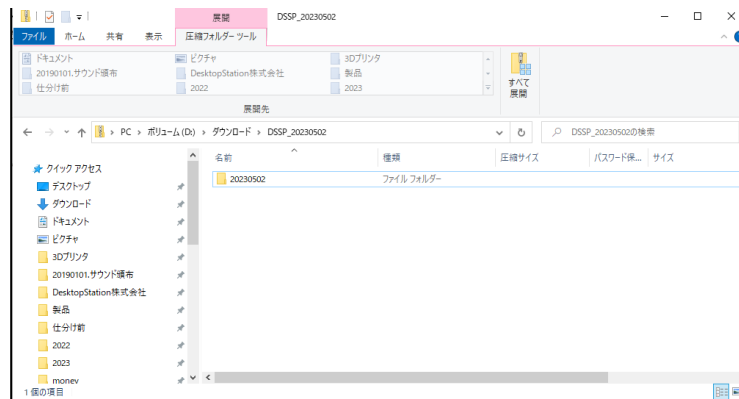


図 7.2.2 エクスプローラーで開いた DSSP の ZIP ファイル内の例

解凍すると、以下のようにフォルダが作成されます。解凍先やユーザーの PC 環境・設定によって、パス（フォルダの場所）は異なります。

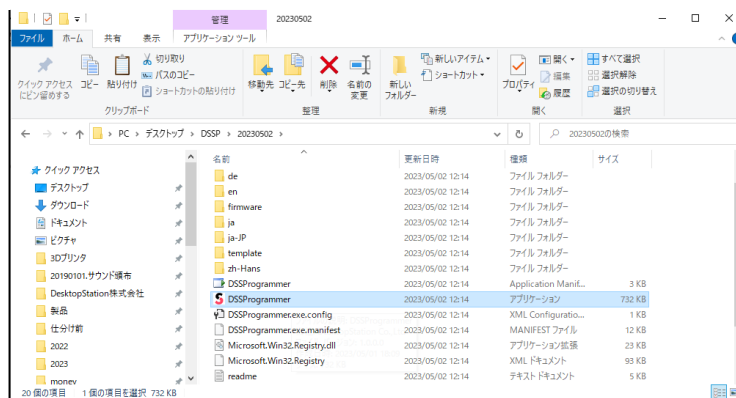


図 7.2.3 解凍した DSSP のフォルダ例

DSSProgrammer.EXE というファイルをダブルクリックすると、DSSP を起動できます。一番最初の起動時に、警告画面が表示されますが、

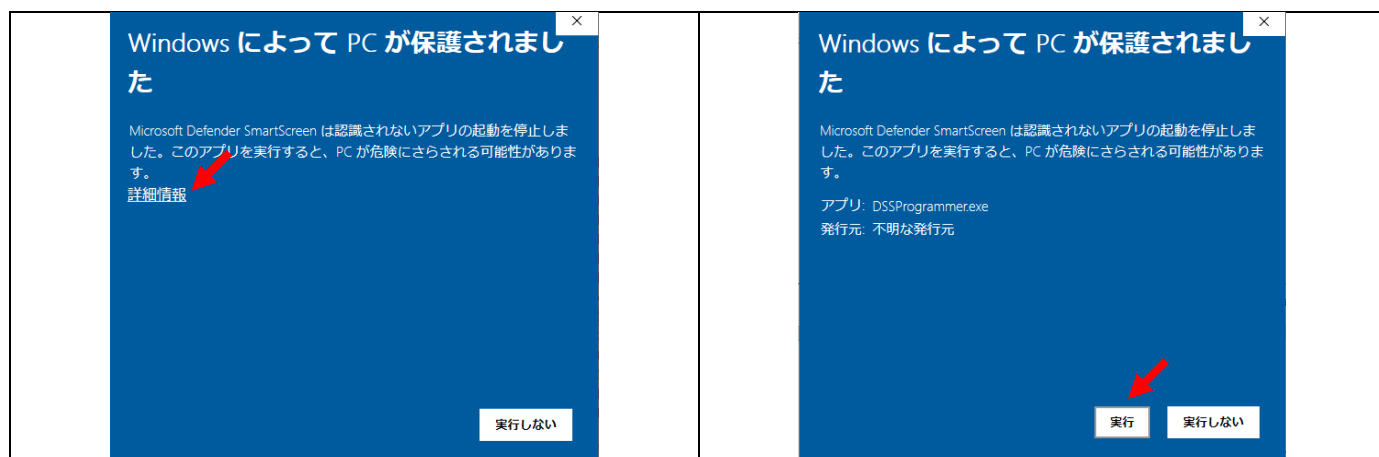


図 7.2.4 Defender SmartScreen 機能の回避方法

### 7.3.DSSP の起動と簡単な説明

DSSP を起動すると、図 7.3.1 に示すような画面が表示されます。DSSP が正常に起動しています。この画面から、サウンドデータの編集、SmileSound デコーダへの書き込み等を行っていきます。まずは、サウンドデータを開いて編集しましょう。

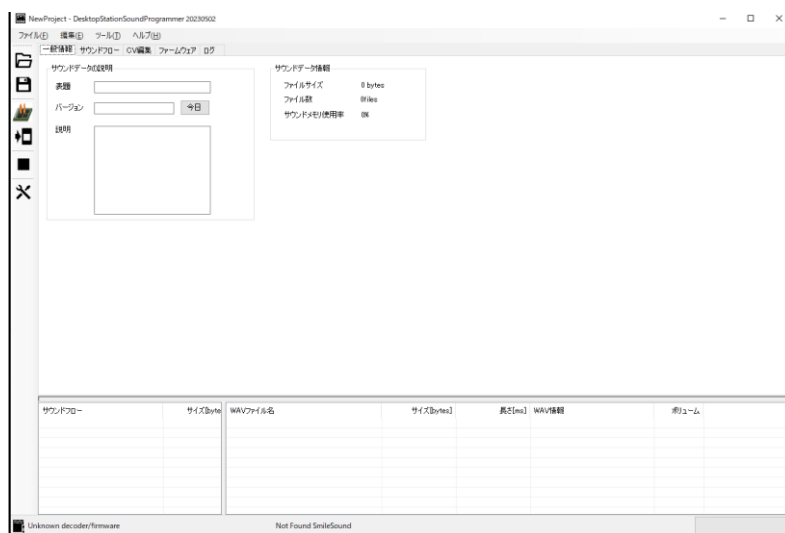


図 7.3.1 DSSP の起動直後の画面

サウンドデータは、オープンサウンドデータのホームページからダウンロードできます。ここでは、キハ40をダウンロードして、DSSP で開くまでの作業を説明します。

はじめに、パソコン上にインストールされている Chrome や EDGE、Safari 等のいずれかのブラウザで、オープンサウンドデータのホームページを開いてください。

#### オープンサウンドデータのホームページ

<https://desktopstation.net/sounds/>

オープンサウンドデータのホームページが表示されましたら、黒いバーで表示された上部のメニューから「サウンドの一覧」を選んでクリックします。すると、サウンドデータの一覧が表示されます。車両の車種名の横に **S** のマークがついているサウンドデータは、SmileSound デコーダへの移植が完了しています。**S** マークがついていないサウンドデータは、LokSound5 向けのみ対応となっていますので注意してください。

ここで、「ディーゼル・気動車」の項目から、「DMF15HSA キハ40系ディーゼル車」を選んでみましょう。すると、キハ40の写真と共に、サウンドデータの中身を説明する文章や、ファンクションキーの割り当て一覧が表示されます。



図 7.3.2 オープンサウンドデータのホームページ画面



図 7.3.3 サウンドデータの一覧

このページから、「サウンドデータファイル **SmileSound**」をダウンロードできます。DOWNLOAD ボタンを押して、ダウンロードしましょう。間違えて LokSound 向けのデータをダウンロードされまると、後述の DSSP で開くことができませんので、ご注意ください。SmileSound で使用できるサウンドデータファイルは必ず「ssdx」の拡張子のついたファイルとなります。ダウンロードが完了したら、このファイルを DSSP で開きます。画面上に D&D するか、ファイルメニューからプロジェクトを開く のどちらかで開きます。

図 7.3.5 に、キハ 40 のサウンドデータを開いた状態を示します。サウンドデータの中には、サウンドフロー(CSV)と音源データ(WAV)が含まれており、下側の2つのファイルビューワーに中身が表示されます。これらのファイルは、入替することもでき、ユーザーが自分でオリジナルの音源や CSV を外部からインポート(取り込み)することもできます。取り込み方は、編集メニューから行うか、ファイルビューワー上で右クリックし、ポップアップメニューを表示するか、D&D を行うか、のいずれかです。

この状態では、前述の SmileSound へのサウンドデータの書き込み手順を行うことで、SmileSound のデコーダにサウンドデータを書き込むことができます。また編集作業を行って、オリジナルのサウンドデータに改造することも可能です。



図 7.3.4 キハ 40 データ

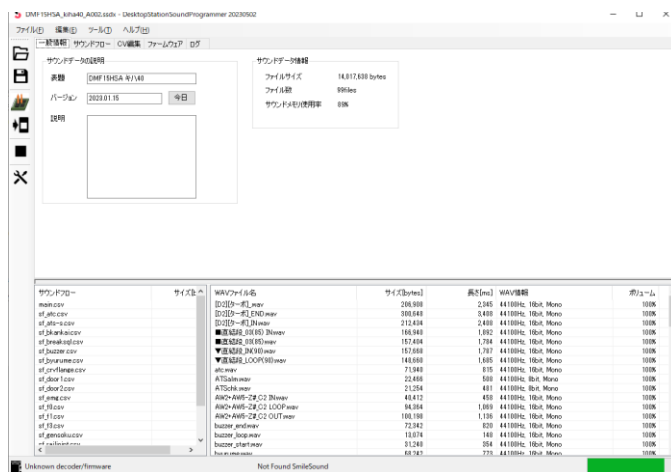


図 7.3.5 DSSP でキハ 40 データを開いた状態

## 7.4. 一般情報

一般情報タブでは、サウンドデータのタイトルや内容、日付などを登録する画面と、全体的な情報を表示する部分に分かれています。

表題、バージョン、説明の内容は、サウンドデータの内容をメモするための情報として使用されます。サウンドデータとして、何らかの処理が行われるわけではありません。記載内容は、デコーダの動作やサウンドデータの内容に影響を与えません。データ管理や、備忘録として使用ください。

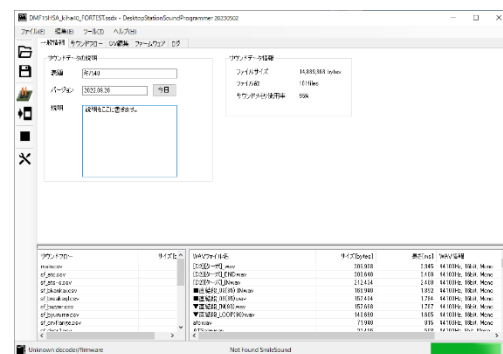


図 7.4.1 一般情報タブ

## 7.5.サウンドフロー

ファイルビューワーが2つあるうちの左側にある、サウンドフローの一覧からサウンドフローを選んで選択すると、サウンドフローが編集できる画面に自動で切り替わります。詳細は後述します。

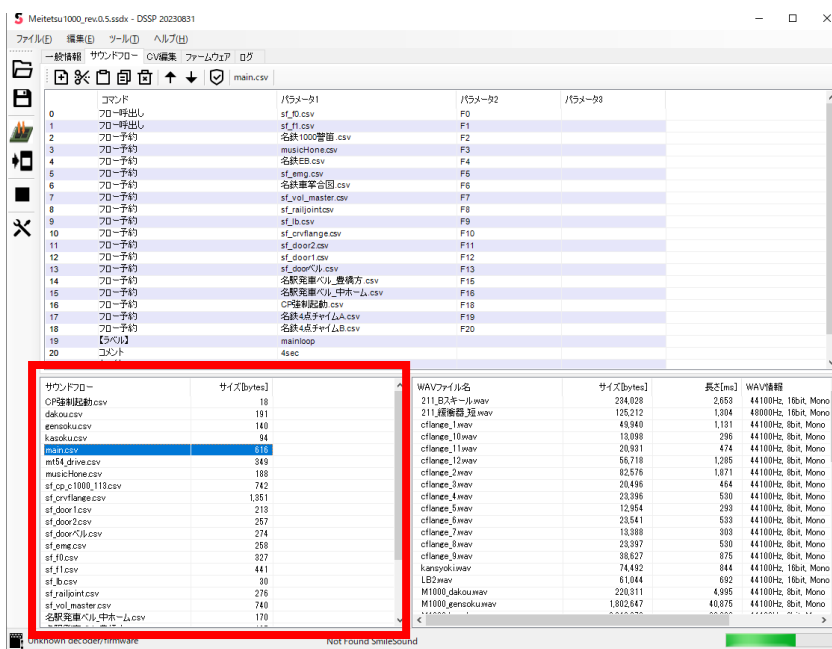


図 7.5.1 サウンドフローの一覧

## 7.6.音データ

下部のファイルビューワーが2つあるうちの右側にあるものは、サウンドデータ内に格納されている音データです。非圧縮 リニア PCM・整数型(UINT16,INT8)の RIFF WAV 形式となっています。D&D で新しい音データを登録できます。ダブルクリックすると、音が再生されます。音を停止したい場合は、再生が終わるまで待つか、左ツールバー上の■ボタンを押すと停止します。詳細は後述します。

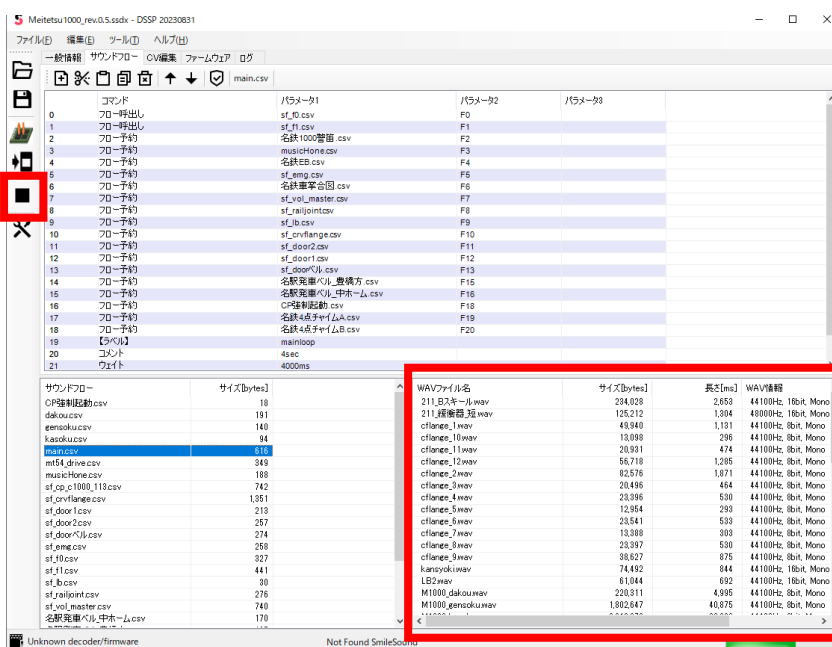


図 7.6.1 サウンドフローの一覧

## 7.7.ログ

サウンドフローにエラーが出ている場合、ログにエラー状態が表示されます。その他、動作状況が記録されることがあります。問題の発生時などにご活用いただけます。

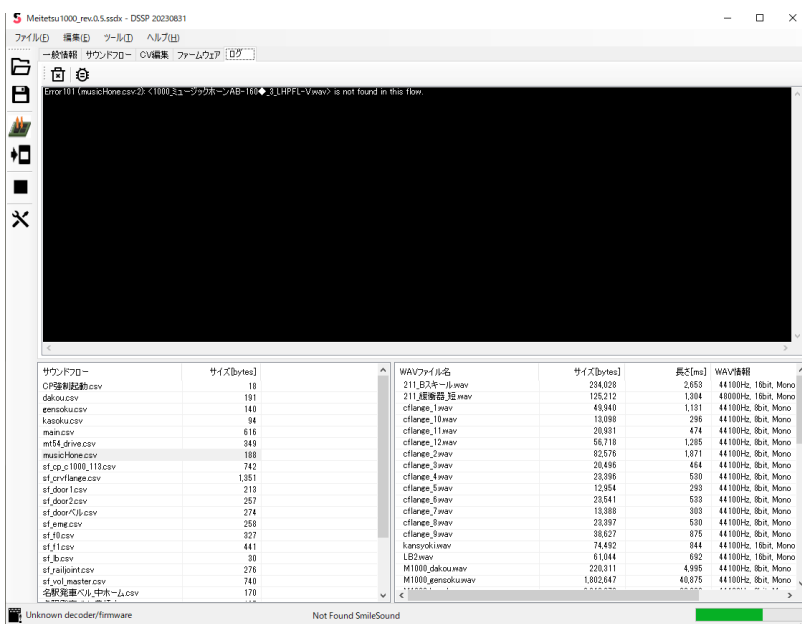


図 7.7.1 ログタブでエラーが出ている例

## 7.8.ファームウェア

接続された SmileSound のファームウェアをアップデートするための画面です。ファームウェアは、DSSPとセットで公開されますので、定期的にDSSPをアップデートし、最新のファームウェアが公開されている場合には、アップデートをお勧めします。左下に、USBライタに接続されている SmileSound のデコーダのファームウェアバージョンが表示されますので、容易に確認が可能です。

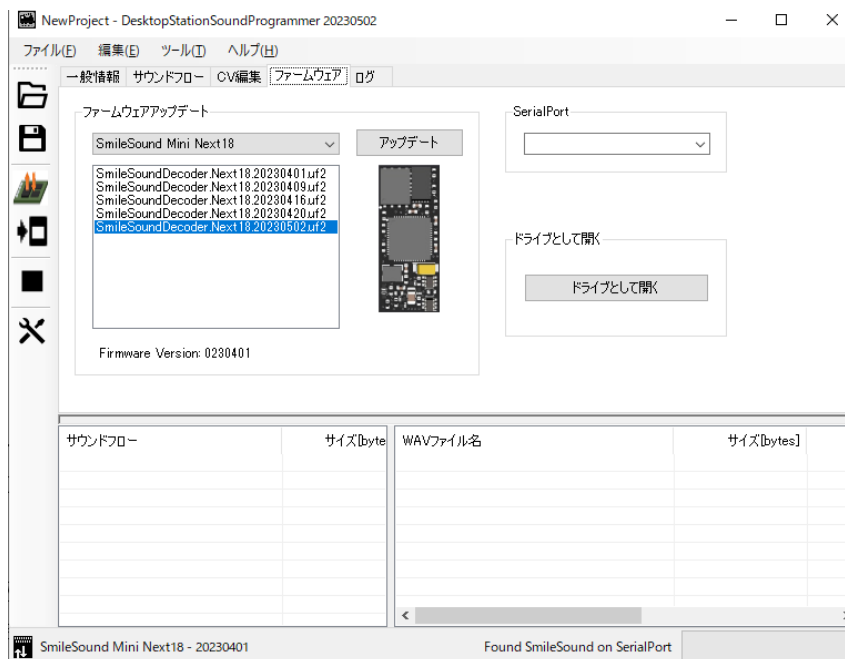


図 7.8.1 ファームウェア画面



## 8. サウンドフロー

### 8.1. はじめに

SmileSound は、デコーダ内部のフラッシュメモリにユーザーが作成したプログラムを格納し本章に記載された仕様・機能の範囲内で自由に動かすことができます。オープンサウンドデータで公開されている ssdx 形式のサウンドデータは、ここに記載されているコマンドを使って作られています。ssdx 形式は、SmileSound 専用のサウンドデータとなっています。競合他社のサウンドデータは、暗号化処理や特殊なデータとなっており、法律上で守られていることから DSSP では一切利用できませんので、ご注意ください。

SmileSound 上で動作するプログラムは「サウンドフロー」と呼びます。サウンドフローは、CSV 形式で記述が可能で、テキストエディタでも作成が可能です。

オープンサウンドデータで公開されているサウンドデータを参考にして、オリジナルのサウンドデータを作ることができます。

### 8.2. サウンドフローで使用できるファイル形式

サウンドフローのプログラミングで使用できるファイル形式を以下に示します。サウンドフローでは、基本的に CSV と WAV ファイルのみを内部で使用します。これをパックしたファイルが ssdx となります。このほかに、DSSP が独自に生成する管理ファイルが存在しますが、ユーザーが使用する必要はありません。

表 8.2.1 サウンドフローで使用できるファイルとその要求仕様

ファイルの種類	要求仕様	備考
CSV	UTF-8 形式, テキスト形式, カンマ区切り	ダブルクォーテーションの使用不可
WAV	RIFF-WAVE, 無圧縮(リニア PCM), サンプリングレート: 32kHz, 16kHz, 8kHz, ビット数: 16bit または 8bit モノラルのみ	タグ付きファイルは利用不可 Float 形式を除き DSSP で自動変換されます。
ssdx	デスクトップステーション株式会社独自開発のサウンドデータ形式。	インポート機能で使用可能

### 8.3. 画面の説明

サウンドフローの編集画面は、以下に示す通り、大きく3つの画面で構成されています。サウンドフローエディタ、サウンドフロー一覧、サウンド一覧です。

サウンドフロー一覧は、サウンドデータの中に登録されているサウンドフローをリストで表示するものです。このサウンドフロー一覧からサウンドフローを選び、サウンドフローエディタで実際の動きなどの編集を行います。

サウンドフローエディタは、サウンドフローの中に記述されているサウンドプログラムを編集する画面であり、コマンドが上から順番に並べて、条件分岐やサウンド再生コマンドを組みながら所望の動きを実現させていきます。

サウンド一覧は、サウンドデータの中に登録された WAV ファイル（音源ファイル）のことを示します。図 8.3.2 には、編集ツールボックスのアイコンの意味を示しています。このほかに、右クリックすることでポップアップメニューが表示されます。またキーボードのショートカットも用意されています。

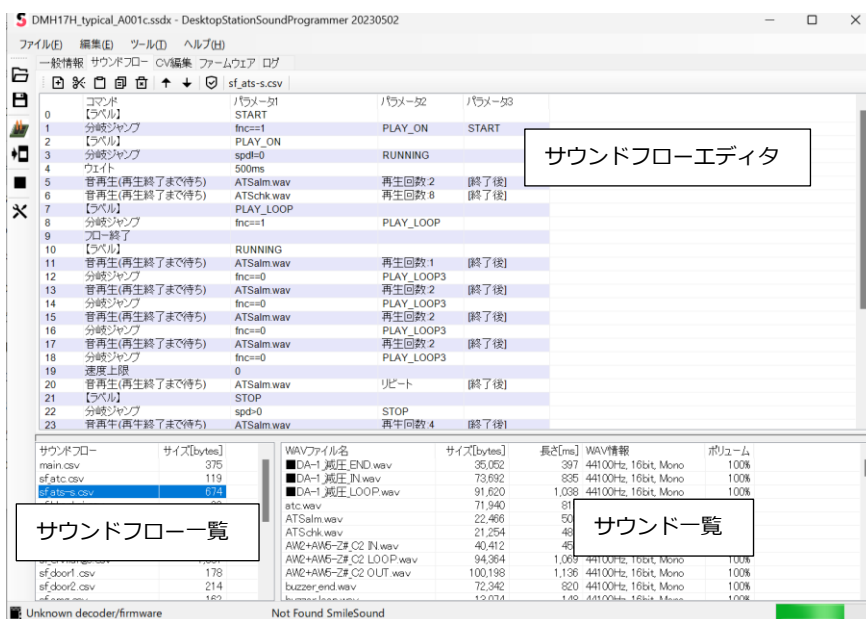


図 8.3.1 DSSP のサウンドフロー画面

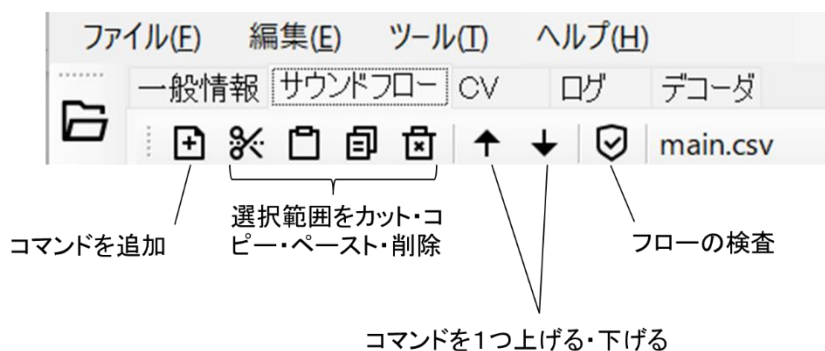


図 8.3.2 サウンドフローエディタの編集ツールボックス

## 8.4.作成・実行ルール

サウンドフローは、以下に挙げるルールに則って動作します。

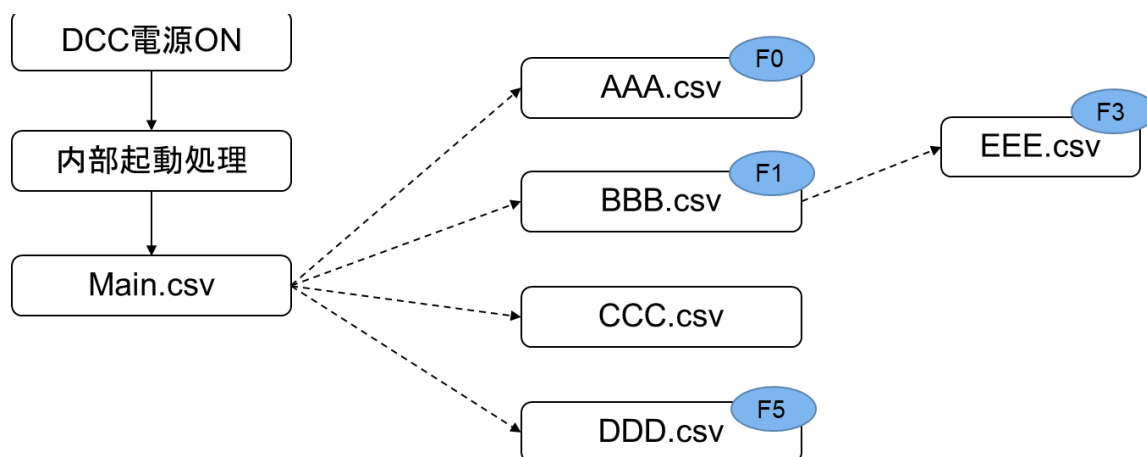
- スクリプトはロードされた後、常時実行されます。
- デコーダに電源が投入されると、起動完了処理後、DCC の信号に関係なく、main.csv が必ず最初にロードされ、実行が開始されます。main.csv に各 csv をコールする処理を記述する事で、様々な機能を実現できます。
- CSV に記述したサウンドプログラム（サウンドフロー）は同時に 16 スロット（16 個のサウンドフロー）を実行できます。
- コールされたサウンドフローは、自動的に空いているスロットに割り当てられます。
- 使っていないフロー（スロット）は自動的に終了する機能を有しており、意識する必要はありません。
- 割り当てられたフロー（スロット）を開放するコマンド（exit）も用意されています。
- 同時に 12 個のサウンドを再生可能です。ただし、内 2 個は蒸気サウンド機能に占有されています。

- 0.1 秒周期で常時、サウンドフローは上から下の行に実行されます。
- サウンドファイルは、8bit ならびに 16bit のモノラルの 32kHz, 16kHz, 8kHz の RIFF WAVE ファイル(LPCM)に対応しています。DSSP を使用することで、デコーダに書き込み時、自動で SmileSound のファームウェアが処理できるサンプリングレートに変換されます。Tag 等の情報は付加しないでください。
- 状態遷移はスクリプト上に記述します。if と goto で状態遷移の待つ条件を作ります。if の条件式が成立したら、遷移させる形です。
- 1つの CSV スクリプトは 128 行までです。
- サウンドだけでなく、AUX の操作や、速度のリミッタ、各種変数機能を搭載しています。
- DCC 信号が無くても、サウンドフローを動かすことができます。ただし、DCC 信号に関係する機能（ファンクションや速度、進行方向など）が使用できません。

## 8.5.編集方法・動作原理

サウンドフローは、C 言語の main 関数のような原理で、一番最初には main.csv という名前のサウンドフローが必ず呼び出されます。よって、サウンドデータ内には必ず、main.csv を配置してください。サウンドデータ内に main.csv が存在しない場合、デコーダは正常に起動しません。main.csv は、DCC 電源 ON またはアナログ電源が 7V 以上印加されたときから SmileSound デコーダの起動が行われ、呼び出しされます。よって、DCC でなくても電源供給がデコーダに行われれば自動で起動し、動作をします。つまり、SmileSound の起動過程において DCC コマンドステーションは必須ではありません。

サウンドフローを呼び出すとき、ファンクション番号を割り付けることができます。サウンドフローとファンクションは 1:1 の関係になります。



## 8.6.コマンドの追加・編集・削除

コマンドの追加方法は、右クリックから「フローアイテムの追加」を選ぶか、「コマンドを追加」のアイコンをクリックして、コマンドを増やします。

既にあるコマンドを変えたい場合には、コマンドをダブルクリックすると編集し、コマンドを変更したりパラメータを変更できます。

既にあるコマンドをサウンドフローから削除したい場合は、Delete キーを押すか、Ctrl+C でカットを行うと、削除することができます。

サウンドフローは、一つないし複数を選択し、クリップボードにコピーすることもできます。また、クリップボード上のサウンドフローのコマンドデータをペーストすることもできます。外部のテキストエディタと併用して使用する場合には、サウンドフローの専用識別子が1行目に入りますので、削除しないようにしてください。

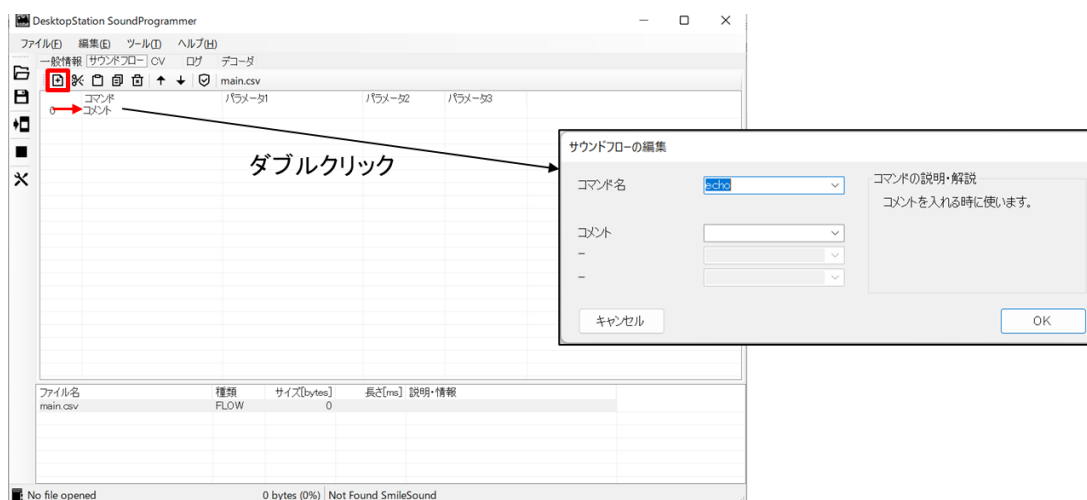


図 8.6.1 コマンドの追加方法

## 8.7.ラベルと条件分岐

ラベルは、if コマンドなどを用いてジャンプするときの目標位置を定義するもので、頻繁に使用します。

“ファンクションの操作を待つ”処理を書くことで、ファンクションに対応した動きを記述できます。以下の例は、ファンクション状態が ON か OFF かによって、グルグル待つ動きをしています。

条件分岐のパラメータ 1 に使用できる変数は、8.9 項に示す全てです。

	コマンド	パラメータ1	パラメータ2	パラメータ3
0	ラベル	START		
1	分岐ジャンプ	fnc==0 →ファンクションがOFFのとき	START	
2	音再生(再生終了まで待ち)	atc.wav	1	0
3	ラベル	PLAY_LOOP		
4	分岐ジャンプ	fnc==1 →ファンクションがONのとき	PLAY_LOOP	
5	サウンド停止			
6	フロー終了			

図 8.7.1 条件分岐の作成例

表 8.7.1 条件分岐の演算子の意味

演算子	演算子の意味	例
==	左と右が同じとき	spd==0
>=	左が右以上の時	acc>=1
<=	右が左以上の時	acc<=-1
>	左が右より大きい時	ref>0
<	右が左より大きい時	ref<128
!=	左と右が異なるとき	fnc!=1

## 8.8.フローの終わり方

サウンドフローを実行し、終わらせる方法が複数あります。処理によっては終わらせては困る動作（たとえば CP）があります。よって、フローの終わり方には以下に示すように、大きく3つの方法があり、サウンドフローに実装する機能に応じて、ご自分で選択してください。

表 8.8.1 フローの終わり方の例

フローの終わり方	対応するコマンド	説明
一番上に戻る	ret	繰り返し動かす処理の場合。
フローを終了する	exit または 記述しない	繰り返さない処理、date で呼び出されるサウンドフローの場合。
任意の場所に戻る (ジャンプする)	goto	繰り返し動かす処理の場合で、先頭に戻る場合には支障のある場合

## 8.9.変数の一覧

変数は、サウンドフローの中で変化する値を表現する識別子の一つです。たとえば、現在走行速度は spd というように英字で表現され、サウンドフロー内で条件に応じて動きを変えるときに使用します。サウンドフローで使用できる変数名は、以下に挙げた変数のみとなっており、ユーザーが自由に追加・変更することはできません。

表 8.9.1 サウンドフロー変数の一覧

定義済変数	呼び名	説明	値の範囲	使用例
local	ローカル変数	呼び出しているサウンドフロー内で使用できる変数	0-65535	local1,local2, … local8
share	共有変数	デコーダ全体のサウンドフローで使用できる変数	0-65535	share1,share2, … share8
spd	現在速度	車両が走行中の速度	0-255	spd
ref	指令速度	加減速中では現在速度とズレが生じます	0-255	ref
fnc	ファンクション	サウンドフローに割り付いているファンクション状態を 0(OFF)または 1(ON)で示します。特定のファンクション番号の状態は確認できません。	0-1	fnc,fnc0,fnc1,…fnc32
aux	AUX 状態	AUX 出力状態を 0(OFF)または 1(ON)で示します。	0-1	aux1,aux2…aux6
tmr	タイマ	呼び出しているサウンドフロー内で使用できるタイマ。1 以上の値をセットすると、1 秒ごとに 1 ずつ減ります。	0-65535	tmr1,tmr2…tmr4
acc	加減速度	加速しているとき正の値、減速しているとき負の値	0-255	acc
dir	進行方向	0 のとき直進、1 のとき後進	0-1	dir
cv	SoundCV の設定値	SoundCV は CV155-CV170 の範囲です。	0-255	cv1, …,cv16
rnd	ランダム変数	ランダムな値を返す変数です。rnd1-rnd100 が使用できます。	0-100	rnd1…rnd100
emg	非常制動状態変数	非常制動モードの時、1 になります。通常時は 0 です。	0-1	emg

## 8.10. コマンドの一覧

サウンドフローで使用できるコマンドを以下に示します。

表 8.10.1 コマンド一覧

<a href="#">aux</a>	AUX 出力
<a href="#">auxs</a>	AUX 出力電圧調整
<a href="#">call</a>	サウンドフローの即時読み出し
<a href="#">cxif</a>	常時監視条件分岐 if の全クリア
<a href="#">date</a>	サウンドフローのファンクション操作時読み出し
<a href="#">dirx</a>	進行方向制限
<a href="#">echo</a>	エコー
<a href="#">emg</a>	非常制動設定
<a href="#">exit</a>	サウンドフロー終了
<a href="#">if</a>	条件分岐 if
<a href="#">goto</a>	ラベルへ即移動
<a href="#">label</a>	ラベルの設定
<a href="#">let</a>	変数と数値の計算
<a href="#">monf</a>	監視ファンクション番号設定
<a href="#">play</a>	サウンドの再生(ウェイト有)
<a href="#">plyx</a>	サウンドの再生(ウェイト無)
<a href="#">pit</a>	再生ピッチの設定
<a href="#">ret</a>	先頭にジャンプ
<a href="#">set</a>	変数に値を設定
<a href="#">spdx</a>	速度下限制限(実質的な速度指定)
<a href="#">sply</a>	加減速音連動サウンド再生
<a href="#">stm</a>	蒸気サウンドの設定
<a href="#">stmc</a>	蒸気サウンドの設定クリア
<a href="#">stms</a>	蒸気サウンドの時間調整
<a href="#">stop</a>	サウンドの停止
<a href="#">slim</a>	速度制限
<a href="#">vol</a>	サウンドフロー内のサウンド音量設定
<a href="#">volm</a>	全体のサウンド音量設定
<a href="#">wait</a>	ウェイト
<a href="#">wrnd</a>	ランダムウェイト
<a href="#">xif</a>	常時監視条件分岐 if の登録

## 8.11. 主なコマンドの使用法

### 8.11.1.aux

ヘッドライト・テールライト、AUX の出力の ON/OFF を行います。AUX コマンドは1つずつ操作を行います。複数を操作する場合には続けて記述してください。

コマンド	引数 1	引数 2	引数 3
aux	AUX 番号(0,255,1-8)	AUX 操作(0 or 1)	-

AUX 番号は以下のように規定されています。

AUX 番号	定義
0	ヘッドライト
255	テールライト
1	AUX1
2	AUX2
3	AUX3
4	AUX4
5	AUX5
6	AUX6
7	AUX7
8	AUX8

#### 【記入例】

aux,0,1

aux,255,0

### 8.11.2.auxs

ヘッドライト・テールライト、AUX の出力が ON の時に、出力する電圧の量を調整します。AUXS コマンドは1つずつ操作を行います。複数を操作する場合には続けて記述してください。

コマンド	引数 1	引数 2	引数 3
aux	AUX 番号(0,255,1-8)	AUX 出力電圧(0-255)	-

AUX 出力電圧は以下のように定義しています。0 を設定すると、AUX コマンドで ON にしても、ライトは点灯しませんので、ご注意ください。また 0 に近い値を設定しても、視認しにくい明るさとなる場合があります。LED や回路によって変わりますので、詳細はライトユニットの取扱説明書を参照してください。

AUX 出力電圧	定義
0	消灯
1-254	中間電圧を PWM 方式で出力
255	常点灯

**【記入例】**

auxs,0,128

auxs,1,255

## 8.11.3.call

他のサウンドフローを呼び出して実行します。他のサウンドフローを呼び出しても、並列実行されるため、動作の影響を受けません。何らかの条件で、複数のサウンドを出す時に使用します。たとえば、発車時のブレーキ緩解音や警笛など。

コマンド	引数 1	引数 2	引数 3
call	サウンドフローファイル(csv)	割り付けるファンクション番号(0-28)	-

以下の記入例は、flow1.csv を呼び出す例です。ファンクション番号を特に使用しない場合は、0 としてください。サウンドフロー内で、if コマンドを使って fnc 変数を操作しなければ、特に支障はありません。

**【記入例】**

call,flow1.csv,2

## 8.11.4.cxif

xif で登録した"いつでも if"条件を、すべてクリアすることができます。

コマンド	引数 1	引数 2	引数 3
cxif	-	-	-

**【記入例】**

cxif

## 8.11.5.date

他のサウンドフローを呼び出して実行します。ただし、割り付けられたファンクションが OFF から ON になったときだけロードして動作します。警笛やドアなど、シンプルなファンクションに使用します。

他のサウンドフローを呼び出しても、並列実行されるため、動作の影響を受けません。何らかの条件で、複数のサウンドを出す時に使用します。たとえば、発車時のブレーキ緩解音や警笛など。

コマンド	引数 1	引数 2	引数 3
date	サウンドフローファイル(csv)	割り付けるファンクション番号(0-28)	-

flow1.csv を呼び出す例です。ファンクション番号を特に使用しない場合は、0 としてください。サウンドフロー内で、if コマンドを使って fnc 変数を操作しなければ、特に支障はありません。





**【記入例】**

```
date,flow1.csv,2
```

## 8.11.6.dirx

進行方向制限コマンドです。実質的に、進行方向を指定することもできます。

コマンド	引数 1	引数 2	引数 3
dirx	進行方向(0: 解除, 1: FWD,2: REV)	-	-

**【記入例】**

```
dirx,1
```

## 8.11.7.echo

コメント行を示します。特に処理は行いません。無視されます。

コマンド	引数 1	引数 2	引数 3
echo	コメント	-	-

コメントを記載した例。

**【記入例】**

```
echo,コメントです。
```

## 8.11.8.emg

非常制動(緊急停止)の ON/OFF ができます。速度上限コマンド(slim)、速度下限コマンド(spdx)とは別に動作しますので、非常制動を解除したら自動で復帰します。

非常制動を ON すると、CV4 で設定した速度に対して 1/5 の短時間で急減速・停止を行います。停止状態は、非常制動を OFF するまで保持されます。

コマンド	引数 1	引数 2	引数 3
emg	0 or 1	-	-

引数 1 に 1 を指定した場合、非常制動を ON します。0 を指定すると、非常制動を解除します。必ず、モーターが停止状態になってから非常制動は解除してください。

emg コマンドが非常制動時(ON)にすると、emg 変数が 1 になります。if 等を使って、非常制動中かどうかを判断できます。非常制動 OFF のときは emg 変数が 0 になります。

**【記入例】**

```
emg,1
```

## 8.11.9.exit

サウンドフローを終了し、使用していたスロットを開放します。再開はできません。なお、exit を記入しなくても、サウンドフローの最後に到達した場合にも自動で exit と同じ処理が動作します。終了させたくない場合は、goto 等をスクリプト末尾に置いてください。

コマンド	引数 1	引数 2	引数 3
exit	-	-	-

## 【記入例】

exit,

## 8.11.10.if

条件が整うと、指定のラベルにジャンプします。条件が偽の時の行き先ラベルを省略すると、if 文に続くコマンドを順序に従って実行します。たとえば、複数の if コマンドを用いて状態遷移を作りたいときに、偽の時の行き先ラベルを省略するなどして使用してみてください。

コマンド	引数 1	引数 2	引数 3
if	条件	真の時の行き先ラベル	偽の時の行き先ラベル(省略可)

条件には、様々な変数(表 8.9.1)が使用できます。ファンクション操作によってジャンプする例。ここでは、監視ファンクションが ON になった時の例です。

## 【記入例】

label,START

if,fnc==1,SOUND\_ON,START

~その他のコマンド~

label,SOUND\_ON

~その他のコマンド~

label,SOUNDLOOP

if,fnc==0,SOUND\_OFF,SOUNDLOOP

label,SOUND\_OFF

goto

## 8.11.11.goto

行き先ラベルにジャンプします。

コマンド	引数 1	引数 2	引数 3
goto	行き先ラベル名	—	—

記述例は、TEST ラベルにジャンプする例です。

**【記入例】**

goto,TEST

~その他のコマンド~

label,TEST

### 8.11.12.label

goto コマンドや if コマンドからジャンプしてくる位置を設定するものです。

コマンド	引数 1	引数 2	引数 3
label	ラベル名	-	-

TEST にジャンプする例

**【記入例】**

goto,TEST

~~~~

label,TEST

### 8.11.13.let

ユーザーが設定可能な変数に、値を代入するためのコマンドです。簡単な数式のみ対応します。

| コマンド | 引数 1    | 引数 2 | 引数 3 |
|------|---------|------|------|
| let  | 代入式(後述) |      |      |

代入式に使用できる左辺の変数は、以下となります。

| 変数の種類 | 変数の説明                                          | 使用可能範囲        | 値の範囲  |
|-------|------------------------------------------------|---------------|-------|
| local | フロー内のみで使用可能な変数。フローが終了するとクリアされる。                | local1~local4 | 0-255 |
| share | 共有変数デコード全体のサウンドフローで使用できる変数                     | share1~share4 | 0-255 |
| tmr   | タイマ変数。値をセットすると、1 秒ごとに値が減っていき、0 になると値は減らなくなります。 | tmr1~tmr4     | 0-255 |

代入式は、以下のような文を示します。

local2=5+1

カッコは使えません。

× let,share3=(1+9)\*2

右辺の数値を、変数にしても構いません。

let,share1=share1+10

右辺の変数には、代入可能変数に加えて、if コマンドでも利用可能な spd や acc といった変数が使えます。

let,share2=share2+acc

### 8.11.14.monf

サウンドフローの if コマンドで用いられる fnc 変数が監視するファンクション番号を変更できます。たとえば、サウンドフローごとにファンクション機能を作るのに用います。デフォルトはファンクション 0(F0)です。

| コマンド | 引数 1                | 引数 2 | 引数 3 |
|------|---------------------|------|------|
| monf | 監視するファンクション番号(0-32) | -    | -    |

ファンクション 8(F8)を、このサウンドフローで監視するように設定する例。fnc 変数は、自動的に F8 を監視するようになります。

#### 【記入例】

monf,8

### 8.11.15.play

サウンドを再生するコマンドです。内部に wait 相当の機能が自動設定されており、play コマンドの次の行には、再生終了直前まで遷移しません。

| コマンド | 引数 1      | 引数 2                       | 引数 3        |
|------|-----------|----------------------------|-------------|
| play | WAV ファイル名 | ループ回数(0 無限ループ,1-100 ループ回数) | 即再生(1 or 0) |

ループ再生させる場合は、ループ ON/OFF の引数を 0 にします。1 回だけ再生したい場合には 1 を入れます。2 を入れると 2 回再生します。最大で 100 回まで指定できます。

play コマンドで、すぐに WAV ファイルを再生したい場合は、即再生の引数に 1 を入れます。このサウンドフロー内で既に再生させているサウンドがある場合、停止してから再生することでよければ、即再生の引数に 0 を入れます。

他のサウンドがループ再生中に、このコマンドが呼ばれた場合、WAV ファイルの再生終了の瞬間で、この WAV ファイルに切り替わります。

#### 【記入例】



```
play,seibuaw_in.wav,1,0
echo,playwav
play,seibuaw_loop.wav,0,0
```

### 8.11.16.plyx

サウンドを再生するコマンドです。play コマンドのような再生待ちウェイト機能が無く、次の行に再生直後に遷移します。再生しながら複雑な停止制御や、動き方を実現したい場合にご利用ください。ウェイト以外は play コマンドと機能は変わりません。

| コマンド | 引数 1      | 引数 2                       | 引数 3        |
|------|-----------|----------------------------|-------------|
| plyx | WAV ファイル名 | ループ回数(0 無限ループ,1-100 ループ回数) | 即再生(1 or 0) |

ループ再生させる場合は、ループ ON/OFF の引数を 0 にします。1 回だけ再生したい場合には 1 を入れます。2 を入れると 2 回再生します。最大で 100 回まで指定できます。

play コマンドで、すぐに WAV ファイルを再生したい場合は、即再生の引数に 1 を入れます。このサウンドフロー内で既に再生させているサウンドがある場合、停止してから再生することでよければ、即再生の引数に 0 を入れます。

他のサウンドがループ再生中に、このコマンドが呼ばれた場合、WAV ファイルの再生終了の瞬間で、この WAV ファイルに切り替わります。

#### 【記入例】

```
plyx,seibuaw_in.wav,1,0
```

### 8.11.17.pit

サウンド再生ピッチ(再生速度)を変更するコマンドです。惰行音など、速度に応じたモータや機械軸の音を調整するために使用します。

| コマンド | 引数 1    | 引数 2   | 引数 3 |
|------|---------|--------|------|
| pit  | ピッチ(直値) | -      | -    |
| pit  | ピッチ下限値  | ピッチ上限値 | -    |

引数 1 だけを与えた場合には、速度に関係なく、その再生ピッチに変更します。

引数 1 と引数 2 に、ピッチの値を入れた場合、速度に応じてピッチが自動で切り替わります。ピッチ下限値は、必ずピッチ上限値よりも小さい値でなければなりません。

ピッチは、1024 を与えると標準の再生速度になります。最低値は 32(約 3%の再生スピード)、最大値は 2048(2 倍の再生スピード)となります。

ピッチ上下限値を設定した場合、停止状態のときに下限値の再生スピード、最高速度の時に上限値の最高スピード、50%速度の時に上下限値の半分の再生スピードで、該当するサウンドフローのスポットの音が再生されます。

サウンドファイルごとに再生ピッチは調整できませんのでご注意ください。



pit コマンドを実行すると、すぐに反映されます。速度に関係なくピッチ変更した場合、速度に応じた自動切換えは即座に OFF されます。

#### 【記入例】

```
pit,512,1024
play,dakou_loop.wav,1,0
```

#### 【記入例】

```
pit,684
play,beep.wav,0,0
```

### 8.11.18.ret

サウンドフローの最初に戻るコマンドです。記入されている位置に関係なく、サウンドフローの最初に戻ります。戻った後、そのまま実行は継続されます。サウンドフローを停止・終了したい場合は、exit コマンドを使用してください。

| コマンド | 引数 1 | 引数 2 | 引数 3 |
|------|------|------|------|
| ret  | -    | -    | -    |

どの場所に置いても、サウンドフローの最初に戻れます。

#### 【記入例】

```
ret
```

### 8.11.19.set

ユーザーが変更可能な変数に値を格納するコマンドです。

| コマンド | 引数 1 | 引数 2  | 引数 3 |
|------|------|-------|------|
| set  | 変数名  | 設定する値 |      |

変数に設定可能な値は変数によって異なります。

| 変数の種類 | 変数の説明                           | 使用可能範囲        | 値の範囲  |
|-------|---------------------------------|---------------|-------|
| local | フロー内のみで使用可能な変数。フローが終了するとクリアされる。 | local1～local4 | 0-255 |
| share | その他のサウンドフローと共有して使用可能なユーザー変数     | share1～share4 | 0-255 |
| tmr   | 値を設定すると、1 秒ごとに自動的に値が減っていくタイマ変数  | tmr1～tmr4     | 0-255 |
| cv    | CV の設定値                         | cv1-cv1024    | 0-255 |

**【記入例】**

```
set,local1,100
```

## 8.11.20.spdx

下限速度を設定するコマンドです。速度指令を受けずに指令代わりに使い、自動運転させることもできます。

| コマンド | 引数 1        | 引数 2 | 引数 3 |
|------|-------------|------|------|
| spdx | 下限速度(0-255) | -    | -    |

**【記入例】**

```
spdx,30
```

## 8.11.21.sply

速度に連動してサウンドを再生するコマンドです。再生する WAV ファイルの長さを 100%として、速度に応じて再生開始位置が自動で変更されます。なお、加速が終わるとサウンドは自動で停止します。

| コマンド | 引数 1      | 引数 2 | 引数 3 |
|------|-----------|------|------|
| sply | WAV ファイル名 | -    | -    |

オープンサウンドデータに使用されている VVVF サウンドは、LokSound の仕様上、加速や減速音を 6 分割や 8 分割して、速度に合わせるように実装しています。

しかし、分割する事で、加速が止まっているにもかかわらず音が出続けてしまう事や、ちょっとした加速を再現することが難しい現状があります。本コマンドを使用することで、速度や加減速状態に応じて、無段階に分割したかのように WAV ファイルを再生することができます。

本コマンドを使用する場合、速度と連動する関係上、WAV ファイルの長さ（再生時間）と加速時間(CV2)や減速時間(CV3)を同じ時間に設定する必要があります。ズレている場合、速度とうまく連動しません。

以下の記入例は、加速サウンドフローと減速サウンドフローを表したものです。両方は、モータ駆動直前にロードするように実装するか、サウンド ON と紐づいて call を行うとよいでしょう。

**【記入例（加速）】**

```
label,START  
if,spd<1,START  
if,acc<=1,START  
sply,E233_kasoku.wav,
```

goto,START

#### 【記入例（減速）】

label,START

if,spd<2,START

if,acc>=-1,START

sply,E233\_gensoku.wav,

goto,START

### 8.11.22.stm

蒸気サウンド機能を設定するコマンドです。

| コマンド | 引数 1      | 引数 2        | 引数 3 |
|------|-----------|-------------|------|
| stm  | WAV ファイル名 | スロット番号(0-3) | -    |

スロット番号 0 から 3 まで 4 つの WAV ファイルを設定すると、stms コマンドで設定した時間間隔に合わせて、上記のドラフト音を速度に連動して再生します。

サウンドフロー動作中に変更することもできます。速度や状況 (ex.重い負荷) に応じてドラフト音を変化させることもできます。

#### 【記入例】

stm,A-01.wav,0,0

stm,A-02.wav,1,0

stm,A-03.wav,2,0

stm,A-04.wav,3,0

### 8.11.23.stmc

蒸気サウンド機能をクリアするコマンドです。

| コマンド | 引数 1 | 引数 2 | 引数 3 |
|------|------|------|------|
| stmc | -    | -    | -    |

スロット番号 0 から 3 まで 4 つの WAV ファイルを設定されたものをクリアします。

実質的に、蒸気サウンド機能を停止することができます。

#### 【記入例】

stmc,

### 8.11.24.stms

蒸気サウンド機能で、音が鳴るタイミングとなる時間間隔を調整するコマンドです。





| コマンド | 引数 1            | 引数 2      | 引数 3 |
|------|-----------------|-----------|------|
| stms | 1:走行開始時,2:最高速度時 | 調整時間(ミリ秒) |      |

走行開始時は、おおよそ 800~1200ms、最高速度時は 50-100ms 程度を指定します。  
車両に応じて調整が必要な場合があります。サウンドフローの本コマンドを使用しますが、この値を CV57(走行開始時)及び CV58(最高速度時)で微調整することもできます。

#### 【記入例】

```
stms,1,800,  
stms,2,50,
```

### 8.11.25.stop

play コマンドで再生したサウンドを停止するコマンドです。

| コマンド | 引数 1 | 引数 2 | 引数 3 |
|------|------|------|------|
| stop | -    | -    | -    |

ループ中等も問わず、サウンドは停止します。なお、同時に実行中の他のサウンドフローのサウンドは停止しません。

#### 【記入例】

```
stop
```

### 8.11.26.slim

モータの速度制限を行うコマンドです。このコマンドで、制限速度を設定すると、デコーダはその速度以下に制限を行います。コマンドステーションから速度指令が制限速度まで切り替わったかのようにふるまいます。制限速度以下の時には、変化はしません。また、コマンドステーションから制限速度以上の速度を設定されても、制限速度以上には速度は変わりません。

| コマンド | 引数 1        | 引数 2 | 引数 3 |
|------|-------------|------|------|
| slim | 制限速度(0-255) | -    | -    |

停止させたい場合は、制限速度を 0 にしてください。制限なし（最高速度まで可）とする場合は、制限速度を 255 と指定してください。

#### 【記入例】

```
slim,255
```

### 8.11.27.vol

サウンドのボリュームを調整するコマンドです。

| コマンド | 引数 1      | 引数 2 | 引数 3 |
|------|-----------|------|------|
| vol  | 音量(0-255) |      |      |

無音にする場合は、0 にしてください。音量を標準とする場合は、255 と指定してください。50%音量は127 です。

#### 【記入例】

```
vol,255
```

### 8.11.28.volm

サウンドのマスターボリューム（デコーダ全体のサウンド音量）を調整するコマンドです。サウンドフローのどのファイルからコマンドを実行しても、共通のマスターボリュームが調整されます。

| コマンド | 引数 1      | 引数 2 | 引数 3 |
|------|-----------|------|------|
| volm | 音量(0-255) | -    | -    |

無音にする場合は、0 にしてください。音量を標準とする場合は、255 と指定してください。50%音量は127 です。

#### 【記入例】

```
volm,127
```

### 8.11.29.wait

ms(ミリ秒)待ちます。並列して実行されているサウンドフローには影響を与えません。

| コマンド | 引数 1      | 引数 2 | 引数 3 |
|------|-----------|------|------|
| wait | 待ち時間(ミリ秒) | -    | -    |

5000 ミリ秒待つ例を以下に示します。

#### 【記入例】

```
wait,5000
```

### 8.11.30.wrnd

ms(ミリ秒)、ランダムに待ちます。並列して実行されているサウンドフローには影響を与えません。

| コマンド | 引数 1             | 引数 2             | 引数 3 |
|------|------------------|------------------|------|
| wait | ランダム待ち時間上限値(ミリ秒) | -                | -    |
| wait | ランダム待ち時間下限値(ミリ秒) | ランダム待ち時間上限値(ミリ秒) | -    |

0-5000ms の間でランダムに待つ例と、1000-2000ms の間でランダムに待つ例を以下に示します。

#### 【記入例(0-5000ms の間でランダムにウェイト)】

wrnd,5000

#### 【記入例(1000-2000ms の間でランダムにウェイト)】

wrnd,1000,2000

### 8.11.31.wspd

速度に連動して、ウェイト時間が設定できます。

$y=ax+b$  の一次方程式で表すと、 $a$  が速度連動ウェイト時間、 $x$  が速度、 $b$  がオフセット時間となります。

反比例モード(0)にすると、速度が上がるたびに、ウェイトが短くなります。最高速度の時、ウェイト時間だけ待ちます。蒸気機関車などでは、反比例モードで使用してください。

比例モード(1)にすると、速度が上がるたびに、ウェイトが長くなります。停止時は、オフセット時間だけ待ちます。

| コマンド | 引数 1       | 引数 2            | 引数 3    |
|------|------------|-----------------|---------|
| wspd | 速度連動ウェイト時間 | 反比例(0) or 比例(1) | オフセット時間 |

オフセット時間は、省略した場合、0ms と設定されます。

### 8.11.32.xif

どの状況にもかかわらず、監視を行う"いつでも if"を動かすための登録機能です。最大で、4 つの条件まで登録できます。4 つめ以降は登録した条件は無視されます。xif で登録した条件を、すべてクリアするには、cxif を使ってください。

| コマンド | 引数 1 | 引数 2     | 引数 3 |
|------|------|----------|------|
| xif  | 条件式  | ジャンプ先ラベル | -    |

たとえば、サウンド再生中に待つ処理を終わらせて、違う処理を動かし始めることができます。

加速中に、減速を開始したときのサウンド切り替えを行わせたり、すぐに反応させたいときに使用します。

#### 【記入例】

xif,f==0,END\_LOOP



## 9.CV 設定

### 9.1.CV とは

CV とは、Configuration Variable(設定値)の略で、DCC デコーダの設定を行う仕組みの総称を示します。SmileSound では、CV の規格に準拠した設定機能を提供しており、DSSP 上で簡単に行える仕組みを実装しています。

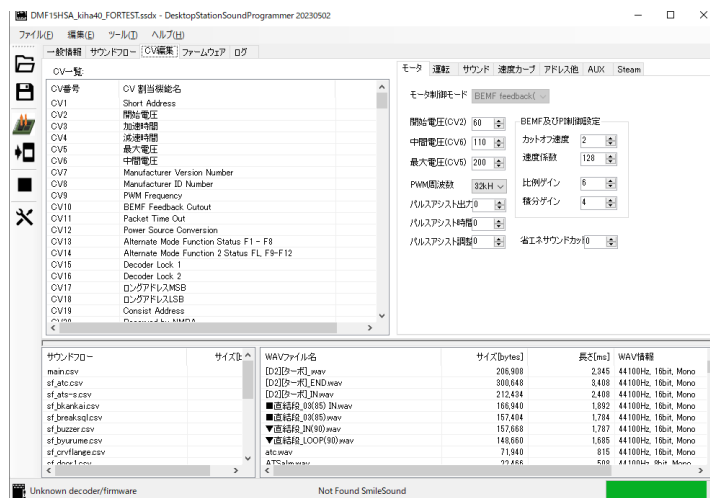


図 9.1.1 DSSP の CV 編集画面

### 9.2.モータタブ

モータの動きを調整する画面になります。

表 9.2.1 CV・モータタブでの設定方法

|                 | 調整方法                                                                         | 備考                       |
|-----------------|------------------------------------------------------------------------------|--------------------------|
| 開始電圧 CV2        | モータや台車、ギア比などで大きく変わります。概ね 16-64 の範囲で設定することが多いです。                              | トラムウェイ HO<br>キハ 40:48-64 |
| 中間電圧 CV6        | 255 の半分あたりの値、または CV5/2 くらいの値を目安にします。CV5/2 から CV5/3 程度の値にすることで、よりリアルな走りになります。 |                          |
| 最高電圧 CV5        | 最高速度を決める設定項目です。255 が最大です。200-255 程度で決めると良いでしょう。                              |                          |
| PWM 周波数         | 特に問題が無ければ 32kHz のままとしてください。                                                  |                          |
| BEMF 及び PI 制御設定 |                                                                              |                          |
| カットオフ速度         | BEMF で問題が無ければ特に変更は不要。                                                        |                          |
| 速度係数            | 以下のように設定してください。<br>旧国・ナロー等ゆっくり走るもの: 128<br>通勤列車、特急等: 64-96<br>新幹線: 64        |                          |

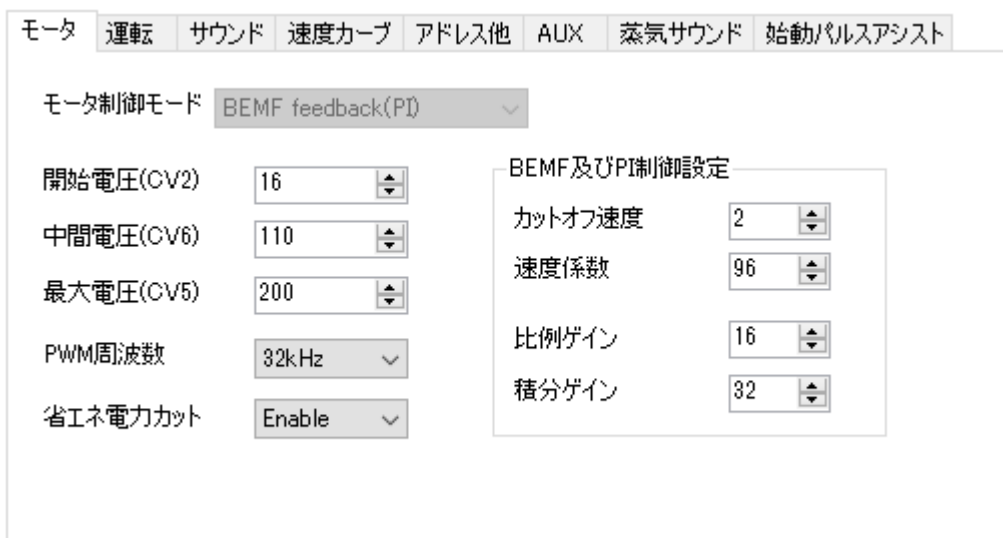


図 9.2.1 モータ CV 設定画面

### 9.3. 始動パルスアシスト機能

モノが動き出すときに静止摩擦という現象が発生するため、パルスアシスト機能は、この静止摩擦の影響を低減するための機能です。静止摩擦があるため、車両の走り始めが渋くなるケースが必ず発生します。パルスアシスト機能は、静止摩擦の影響を小さくするように、モータにかかるパワーを少しだけブーストする役目があります。モータや車両の重さに応じて、大きすぎず、小さすぎずの絶妙な値の設定が必要になります。

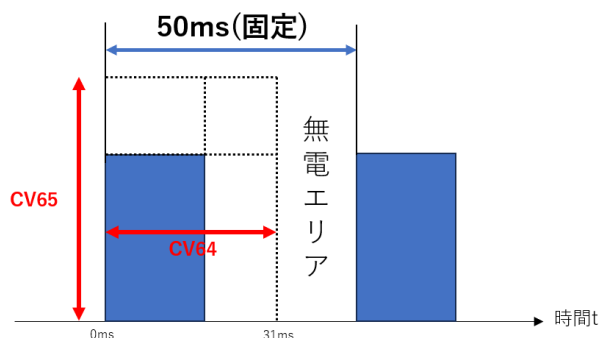


図 9.2.1 パルスアシストのイメージ

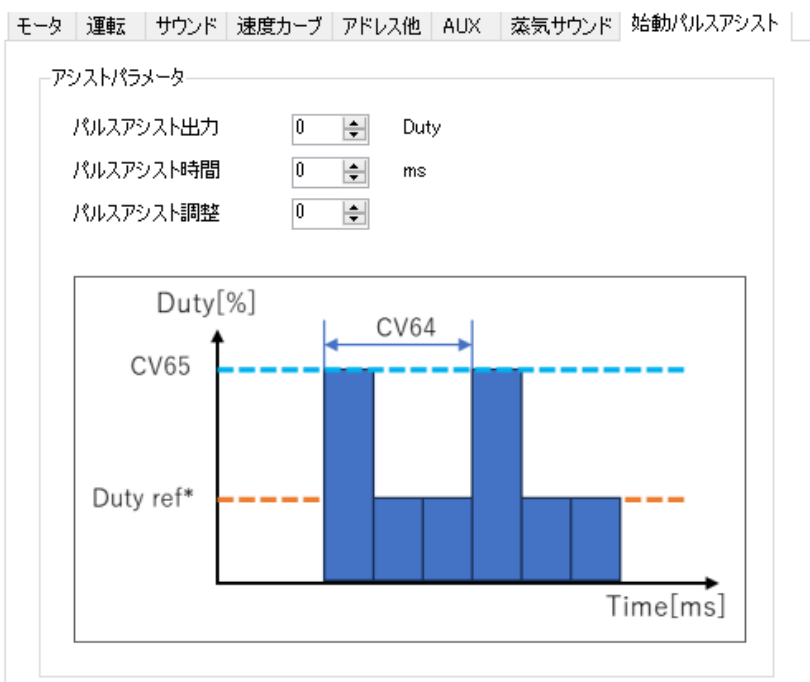


図 9.3.2 パルスアシスト設定

【パルスアシスト機能を使用しない場合】

CV64=0, CV65=0 としてください。パルスアシスト機能は動作しません。DSSP では、アシストパラメータを全て 0 にすると、パルスアシスト機能は無効となります。

#### 【パルスアシスト機能を適用する場合】

はじめに初期値として CV64=15, CV65=205 を設定します。DSSP では、パルスアシスト出力(CV65)を 40%、パルスアシスト時間(CV64)を 15ms、パルスアシスト調整を 0 に設定します。その後、CV64 を下げる(上げる)方向で、走らせながら試していただくと良いかと思えます。

目標は、SpeedStep1 (一番最小の走る速度)において、アシスト無し時で減速時に止まる直前の速度(最低速度)の 1/2~2/3 程度になる速度です。

下の図を説明します。一番左の時間(ms はミリ秒)は、パルスアシスト機能を動かす時間です。パルスアシスト機能は、一瞬だけ動かしますが、その一瞬を微調整できます。デフォルトは CV64=15 としていますので、パルスアシストの時間は 15ms で「通常」の設定となります。パルスアシストの時間を長くすると、飛び出している時間が長くなります。モータの仕様によって、長めにしないと走り始めない場合がありますので、後述の CV65 だけでは足りない場合は、CV64 の時間を少しずつ大きくしてみてください。

CV65 は、パルスアシストする力の度合いで、205 を標準としていますので、Duty が 40%となります。CV65 の値を小さくすると、パルスアシストする力が小さくなります。車両が飛び出す場合には、CV65 を小さくすると、飛び出しが収まりますので、ちょうどよい下限を見つけてください。

| CV64           | 通常 | REV時に短くアシスト |    |    | 通常  | REV時に長くアシスト |     |     |     |
|----------------|----|-------------|----|----|-----|-------------|-----|-----|-----|
|                |    | -3          | -2 | -1 |     | 0           | 1   | 2   | 3   |
|                |    | CV64設定値     |    |    |     |             |     |     |     |
| パルスアシスト時間 [ms] | 0  | 0           | -  | -  | -   | -           | -   | -   | -   |
|                | 1  | 1           | 33 | 65 | 97  | 129         | 161 | 193 | 225 |
|                | 2  | 2           | 34 | 66 | 98  | 130         | 162 | 194 | 226 |
|                | 3  | 3           | 35 | 67 | 99  | 131         | 163 | 195 | 227 |
|                | ~  | ~           | ~  | ~  | ~   | ~           | ~   | ~   | ~   |
|                | 15 | 15          | 47 | 79 | 111 | 143         | 175 | 207 | 239 |
|                | ~  | ~           | ~  | ~  | ~   | ~           | ~   | ~   | ~   |
|                | 28 | 28          | 60 | 92 | 124 | 156         | 188 | 220 | 252 |
|                | 29 | 29          | 61 | 93 | 125 | 157         | 189 | 221 | 253 |
|                | 30 | 30          | 62 | 94 | 126 | 158         | 190 | 222 | 254 |
|                | 31 | 31          | 63 | 95 | 127 | 159         | 191 | 223 | 255 |

| CV65                 | 通常      |     |
|----------------------|---------|-----|
|                      | CV65設定値 |     |
| パルスアシスト出力 (DUTY) [%] | 0       | 0   |
|                      | 1       | 5   |
|                      | 2       | 10  |
|                      | 3       | 15  |
|                      | ~       | ~   |
|                      | 40      | 205 |
|                      | ~       | ~   |
|                      | 47      | 241 |
|                      | 48      | 246 |
|                      | 49      | 251 |
|                      | 50      | 255 |

図 9.3.3 パルスアシスト調整表

パルスアシスト調整は、左側の表の横方向の-3~0~+3 の調整項目が該当します。

| CV64           | 通常 | REV時に短くアシスト |    |    | 通常  | REV時に長くアシスト |     |     |
|----------------|----|-------------|----|----|-----|-------------|-----|-----|
|                |    | -3          | -2 | -1 | 0   | 1           | 2   | 3   |
|                |    | CV64設定値     |    |    |     |             |     |     |
| パルスアシスト時間 [ms] | 0  | 0           | -  | -  | -   | -           | -   | -   |
|                | 1  | 1           | 33 | 65 | 97  | 129         | 161 | 193 |
|                | 2  | 2           | 34 | 66 | 98  | 130         | 162 | 194 |
|                | 3  | 3           | 35 | 67 | 99  | 131         | 163 | 195 |
|                | ~  | ~           | ~  | ~  | ~   | ~           | ~   | ~   |
|                | 15 | 15          | 47 | 79 | 111 | 143         | 175 | 207 |
|                | ~  | ~           | ~  | ~  | ~   | ~           | ~   | ~   |
|                | 28 | 28          | 60 | 92 | 124 | 156         | 188 | 220 |
|                | 29 | 29          | 61 | 93 | 125 | 157         | 189 | 221 |
|                | 30 | 30          | 62 | 94 | 126 | 158         | 190 | 222 |
|                | 31 | 31          | 63 | 95 | 127 | 159         | 191 | 223 |

| CV65                 | 通常      |
|----------------------|---------|
|                      | CV65設定値 |
| パルスアシスト出力 (curv) [%] | 0       |
|                      | 1       |
|                      | 2       |
|                      | 3       |
|                      | ~       |
|                      | 40      |
|                      | ~       |
|                      | 47      |
|                      | 48      |
|                      | 49      |
|                      | 50      |

図 9.3.4 パルスアシスト調整の範囲

## 9.4. 運転タブ

CV3, CV4 に関係する加速時間・減速時間を編集する画面です。一部のコマンドでは、ここで設定された加減速時間を元にしてサウンドを出力しますので、サウンドデータを新規で開発される場合には、サウンドデータごとに調整を行う必要があります。オープンサウンドデータのホームページで公開されているサウンドデータは既に調整済みですので、この画面の変更は不要です。

なお、調整されたサウンドデータに対して、本画面で設定を変更した場合、加速時・減速時のサウンドが異なる挙動を示す場合があります。調整済みのサウンドデータを使用される場合は、本画面の変更はしないことを強く推奨します。

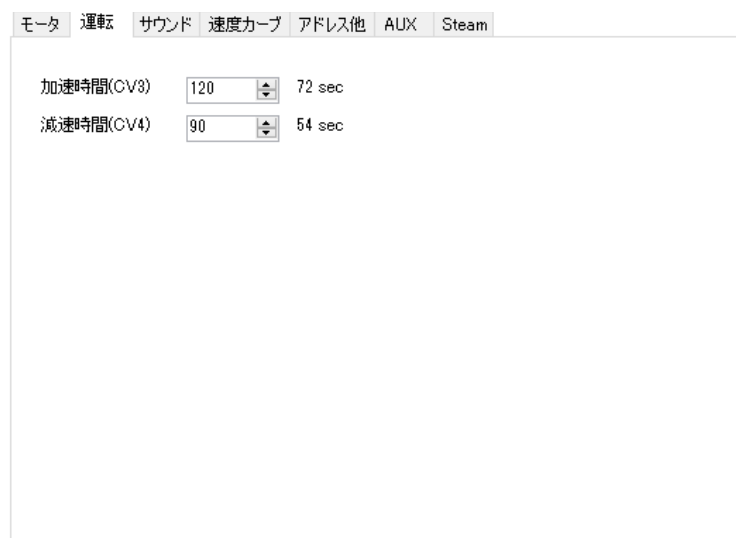


図 9.4.1 運転設定画面

## 9.5. サウンドタブ

CV63 マスター音量や、CV196, CV197 イコライザ機能の設定をする画面です。

マスター音量は、0-255 の範囲で、音量を設定できます。

イコライザは、16 が無効(効果なし)となっており、16 より大きい値を入れると低域(800Hz 以下)または高域(4kHz 以上)の音を強調します。逆に、16 よりも小さい値を入れると、低域または高域の音を弱めます。OPS/Program on Main(PoM)により、運転中に CV 値を変更することができます。



イコライザは、スピーカー自体の特性やエンクロージャ、スピーカーの固定方法・固定箇所、周囲の構造に大きく影響を受けます。特に小型のスピーカーや薄く小さなエンクロージャでは、期待する低音増強効果が得られない場合がありますので、予めご了承ください。



図 9.5.1 サウンド設定画面

## 9.6.速度カーブタブ

DCC デコーダには、速度テーブルと呼ばれる、加速の上昇カーブを調整する機能が搭載されています。標準では、CV2・CV5・CV6 を使った直線補完による速度カーブが使用されますが、CV67-94 に用意されている 28step の速度カーブを使用する事で、より滑らかな上昇カーブを設定できます。速度テーブルの切り替えは DSSP の速度カーブ設定画面で行えます。

速度カーブは、簡単な CSV ファイルで保存・読み込みができますので、車両の形式やモータ、台車の特性に合わせて作成し、自由に入れ替えることができます。グラフをクリックするマウス操作でも簡単に速度カーブパラメータは調整頂けます。

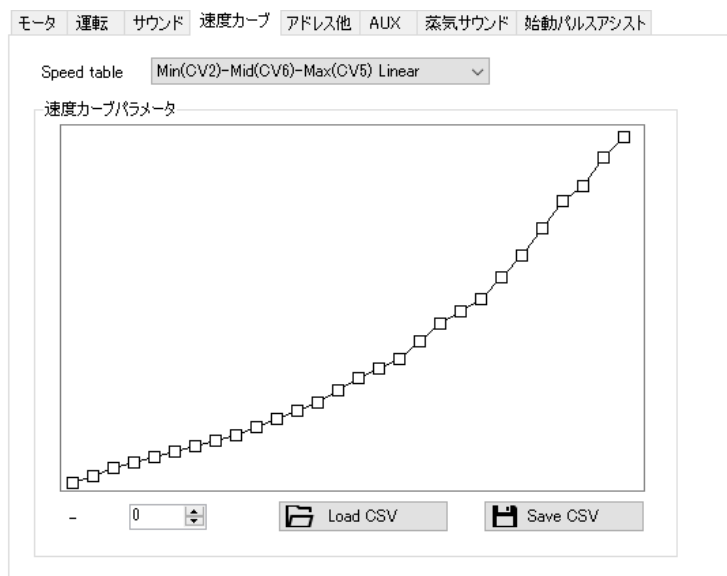


図 9.6.1 速度カーブ設定画面

## 9.7.アドレス他タブ

デコーダのアドレス設定を行う画面です。通常は、ショートアドレスの3としておくことをお勧めします。サウンドデータに3以外のアドレスを設定すると、動作チェックの時に故障と混同してしまう事例があるためです。DSSP ではサウンドデータを行い、アドレスの変更はお使いのコマンドステーションで行いましょう。



図 9.7.1 アドレス他設定画面

## 9.8.AUX タブ

デコーダの AUX 端子で出力することで制御が行える室内灯やヘッドライト・テールライトなどの LED を使ったライトのエフェクト設定を行う画面です。



図 9.8.1 AUX 設定画面

## 9.9.Steam タブ

蒸気コマンド(stm 等)で使用します。

低速時、音を出す間隔を長くし、高速時には音を出す間隔を短くすることで、上記のシュポシュポという音を再現しています。しかし、直線的に間隔を縮めると、低速時に急激に音の変わり方が激しくなります。そこで、下記のオレンジ色の線を描くように補間処理を行って、スムーズに音の間隔が短くなるように調

整しております。この調整具合も、今回のテストの対象となります。

表 9.9.1 蒸気サウンドの設定

| 設定値                         | 設定内容の説明                                                              |
|-----------------------------|----------------------------------------------------------------------|
| Low spd interval<br>(CV57)  | 走り始めの時の蒸気音の時間間隔を調整する CV です。設定した値の 10 倍の数値が足されます。1 を入れると 10ms が足されます。 |
| High spd interval<br>(CV58) | 最高速度時の時間間隔[ms]を調整する CV です。設定した値の 1 倍の数値が足されます。1 を入れると 1ms が足されます。    |

4つの音を、順番に鳴らしますが、2つのチャンネルを組み合わせ、交互に再生させています。この動きが特殊なため、ファームウェアで専用の機能を実装しています。0→1や1→2等に動く時間間隔は、上記の速度～蒸気音再生間隔[ms(ミリ秒)]で制御しています。ファイル名は後述する stm コマンドで指定しますので、わかりやすいように任意の名前を付けるとよいでしょう。また、音はサウンドフロー上で走行中に変更できるため、低速時のシュポシュポの音、高速時のシュポシュポの音を stm コマンドで切り替えることもできます。

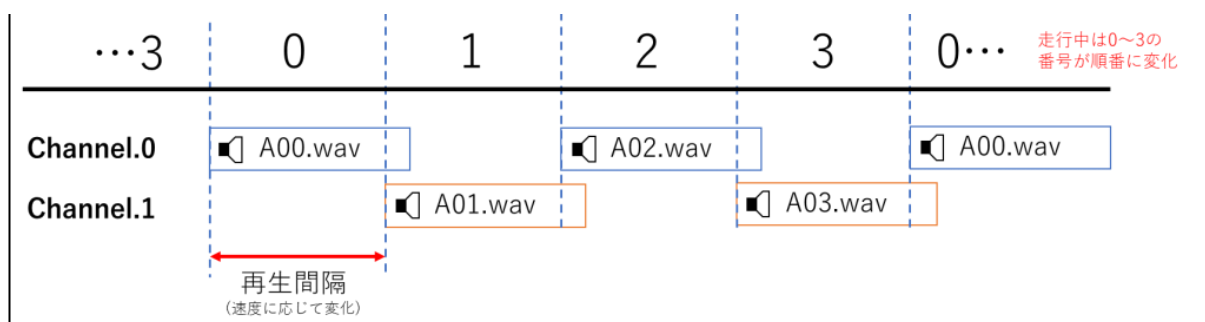


図 9.9.1 stm コマンドを用いた蒸気サウンドの再生タイミング制御

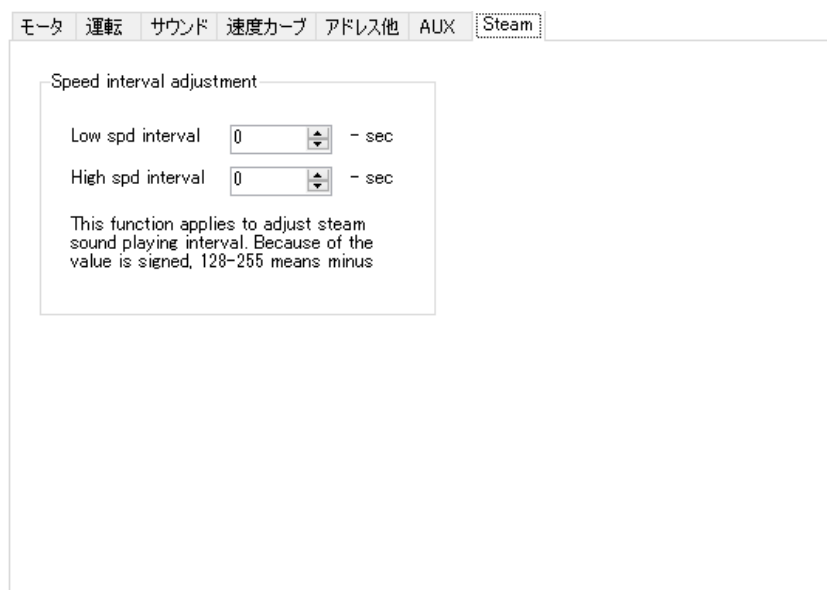


図 9.9.2 蒸気サウンド用 CV 設定画面

## 10. サポート

### 10.1. 動作保証の環境条件

デスクトップステーション株式会社がユーザーに提供するサポートでは、以下に示す環境をサポートの提供条件といたします。弊社で修理・調整を行い、動作を確認できた場合において、この条件に適合しない利用環境・条件・設定をお使いの場合、ユーザー環境で動作確認ができたものとみなします。あらかじめ、ご了承ください。

表 10.1.1 動作保証環境条件

|            | 保証対応機器                                                                  | その他条件                                     |
|------------|-------------------------------------------------------------------------|-------------------------------------------|
| コマンドステーション | DSair2, DSairLite                                                       | ファームウェアは最新版とします。                          |
| デコーダ       | SmileSound シリーズ                                                         | ファームウェアは最新版とします。                          |
| デコーダテスタ    | ESU 53900 Decoder Tester または<br>LaisDcc 860033 Decoder Tester Pro       | ユーザーが自作した同等機器であっても動作は保証しません。              |
| AC アダプタ    | 秋月電子で販売されている AC アダプタ DC12V または DC15V, DC16V                             | 購入から3年以内の AC アダプタのみ動作保証します。他機器流用品は保証しません。 |
| 線路・車両      | 線路は使用せず、コマンドステーションとデコーダテスタをフィーダ線で直結した状態で検証を行います。デコーダが車両搭載状態での動作保証はしません。 |                                           |

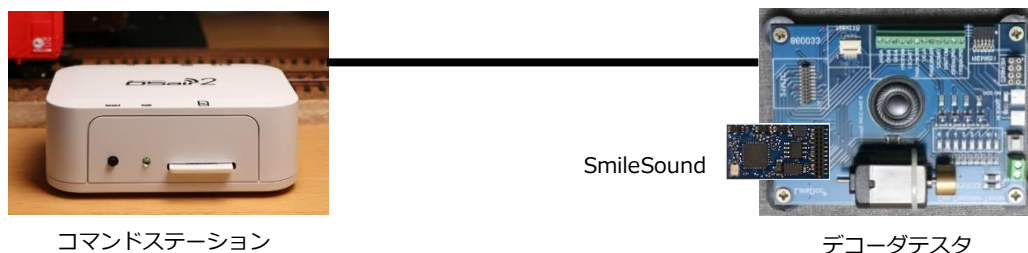


図 10.1.1 動作保証環境条件

## 10.2. サポート体制について

デスクトップステーション株式会社はインターネット専業体制となっており、実店舗や直接のサポート窓口を持っておりません。電話でのサポートは一切行っておりません。また、販売価格を抑えるために業務簡略化を強く進めております。このため、通常のお店とは異なり、非常に少ない人員で製品開発・販売・事業運営を行っているため、さまざまなサービスを省略しております。ご理解いただきますよう、よろしくお願い致します。

表 10.2.1 サポート問合せ内容と窓口

| サポート問合せ内容                                                                                                     | サポート窓口・連絡先                                                                                                                                                                                                                                                                  |
|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 購入前の相談、質問<br>購入後の質問、相談<br>機器やソフトの使用方法に関する相談<br>キットの組み立てに関する相談<br>搭載に関する相談、質問<br>他社機器との相性相談<br>不具合かどうか判別できないとき | デジタル鉄道模型フォーラム<br><a href="https://desktopstation.net/bb/">https://desktopstation.net/bb/</a><br><br>・登録、利用料は完全無料<br>・デスクトップステーション株式会社が運営                                                                                                                                    |
| 初期不良時<br>機器故障時（無料修理期間の範囲内外を問わず）<br>購入前の在庫・納期間合せ<br>不足部品の問合せ<br>注文時の支払い方法の変更<br>キット組立て用の部品分売・部品交換・代替部品提供の問合せ   | 購入店にご相談ください。デスクトップステーションオンラインストアは問い合わせフォームにて、注文番号を添えてご連絡ください。注文番号と購入履歴を照合してサポートいたします。<br><br>デスクトップステーションオンラインストア<br><a href="https://desktopstation.net/shop/contact">https://desktopstation.net/shop/contact</a><br><br>デスクトップステーションメール窓口<br>support @ desktopstation.net |
| 上記の内容以外のその他の問い合わせ                                                                                             | デスクトップステーションメール窓口<br>support @ desktopstation.net<br><br>※上記のサポート窓口にご相談すべき内容を、弊社のメール宛に個別相談した場合、返信を行わない場合があります。                                                                                                                                                              |

### 10.3. 質問・回答

質問としてよく挙げられる内容を、ここにご紹介します。その他で質問や相談がある場合には、デジタル鉄道模型フォーラムをご利用ください。弊社では、個別サポートは故障時・初期不良のみとなっており、通常の使用に関するユーザーへのサポートは行っておりません。

お問い合わせをいただいても、回答しない場合があります。予めご了承ください。

また、ユーザーに起因する故障については、保証規定（3章）に従って、有償交換となりますので、ご注意ください。

デジタル鉄道模型フォーラム：

<https://desktopstation.net/bb/>

#### Q. 突然音が出なくなり、故障しました。

スピーカー配線の処理に問題があり、線路か AUX、COM+関連の配線とショートすると、SmileSound の内部 IC が故障します。絶縁処理を正しく行うようにしてください。特に、長い配線や、はんだ付け部・導電部が露出している箇所や、デコーダやスピーカーの固定が甘い場合、走行時の振動や衝撃などで想定していない箇所と接触し、ショートを発生させるケースが多くあります。

#### Q. SmileSound mini Next18 の音量が小さいです。

設計者の判断により、SmileSound mini Next18 は、Standard MTC21 よりもサウンド音量を控えめに設計しています。

#### Q. CV の読み出しに失敗します。

モータの消費電流が小さい場合や、コマンドステーションとの相性により、CV の読み出しに失敗する場合があります。SmileSound デコーダは、Direct モードでの利用を推奨しています。

#### Q.線路で走らせると、すぐに止まります。

アナログで正常に走ることができる車両ですか？アナログで安定して走れない車両は、DCC では正常に動きません。集電箇所が少ない車両や、設計上、もともと集電に強くない車両もあります。全車集電がもっとも効果的ですが、コンデンサの増強・配線の見直しなど、集電不良対策をしてください。

車両ではなく、線路やフィーダ線、線路電圧に原因がある場合もあります。HO は、15-16V を使用するようになっています。

#### Q.競合他社のサウンドデータを書き込んで使用できますか。

SmileSound は、SmileSound 用に開発されたサウンドデータのみが使用可能です。競合他社のサウンドデータは、競合他社の仕様や権利に基づいて暗号化や特殊処理を施して作成されており、弊社がその権利を侵すことは法律上でも禁止されています。よって、競合他社のサウンドデータは一切使用できません。

### 10.3.1.CV 一覧

SmileSound で標準設定としている CV の一覧を以下に示します。ここで記載している初期値は、サウンドデータによって異なりますので、参考としてください。

表 10.3.1.1 CV 一覧

| CV 番号     | 機能説明                                | 初期値       |
|-----------|-------------------------------------|-----------|
| CV1       | ショートアドレス                            | 3         |
| CV2       | 始動電圧                                | 8         |
| CV3       | 加速時間(0.6 で割った時間が秒)                  | 120       |
| CV4       | 減速時間(0.6 で割った時間が秒)                  | 90        |
| CV5       | 最大電圧                                | 200       |
| CV6       | 中間電圧                                | 70        |
| CV9       | PWM キャリア                            | 0 (32kHz) |
| CV10      | BEMF カットアウト係数                       | 2         |
| CV17      | Long Address LSB                    | 0         |
| CV18      | Long Address MSB                    | 0         |
| CV28      | RailCom 有効設定                        | 1         |
| CV29      | デコード設定                              | 10        |
| CV54      | BEMF 係数(16 で割った値が 1.0 倍,48 で 3.0 倍) | 96        |
| CV55      | PI 制御器 P ゲイン                        | 16        |
| CV56      | PI 制御器 I ゲイン                        | 32        |
| CV62      | 集電不良対策・自動サウンド OFF                   | 0         |
| CV63      | マスターボリューム                           | 100       |
| CV64      | キックスタート切替速度                         | 50        |
| CV65      | キックスタート電圧                           | 0         |
| CV67-94   | スピードカーブ                             | -         |
| CV154     | ユーザーボリューム設定                         | 255       |
| CV155-170 | SoundCV 設定                          | -         |
| CV185     | ヘッドライト・AUX 出力設定                     | 0         |
| CV186     | テールライト・AUX 出力設定                     | 0         |
| CV187     | AUX1・AUX 出力設定                       | 0         |
| CV188     | AUX2・AUX 出力設定                       | 0         |
| CV189     | AUX3・AUX 出力設定                       | 0         |
| CV190     | AUX4・AUX 出力設定                       | 0         |
| CV191     | AUX5・AUX 出力設定                       | 0         |
| CV192     | AUX6・AUX 出力設定                       | 0         |
| CV193     | AUX7・AUX 出力設定                       | 0         |
| CV194     | AUX8・AUX 出力設定                       | 0         |
| CV195     | 予約                                  | 0         |
| CV196     | イコライザ・低域(-800Hz)                    | 16        |

|       |                 |    |
|-------|-----------------|----|
| CV197 | イコライザ・高域(4kHz-) | 16 |
| CV198 | 予約              | 0  |
| CV199 | 予約              | 0  |
| CV200 | 予約              | 0  |
| CV201 | 予約              | 0  |
| CV202 | 予約              | 0  |
| CV203 | 予約              | 0  |
| CV204 | 予約              | 0  |

### 10.3.2.参考文献

SmileSound は、開発にあたり以下の参考文献・OSS・ソフトウェア・仕様書等を参考としました。御礼を申し上げます。

- NMRA DCC Standard 公開規格書
- RailCommunity 公開規格書(RCN)
- DCCwiki (<https://dccwiki.com/>)
- Arduino, Arduino IDE, Arduino eco system  
<https://www.arduino.cc/>
- Pico-SDK  
<https://github.com/raspberrypi/pico-sdk>
- earlephilhower 版 Arduino Pico ライブラリ  
<https://github.com/earlephilhower/arduino-pico>
- NMRA Digital Command Control (DCC) Library  
<https://github.com/mrrwa/NmraDcc>
- mklittlefs  
<https://github.com/earlephilhower/mklittlefs>
- uf2  
<https://github.com/microsoft/uf2>



### 10.3.3.謝辞

SmileSound は、以下の企業・開発メンバーによって開発・運営されています。ここに記載のない、サウンドデータのクリエイターの皆様、ボランティアの開発協力者の方々にも感謝を申し上げます。

- デスクトップステーション株式会社
- Nagoden
- Fujigaya2
- TRAINO
- Maison de DCC
- SmileWorks
- HMX

# **SmileSound**

**FULL PROGRAMMABLE SOUND DECODER**



**Desktop Station**