



ニュースレター

Vol.29-No.6

2017年11月-12月

デジタルコンサルタント

Rick Sinclair(リック)

Curtis Jeung(カーティス)

最新ソフトウェア情報

Current Central Station 3 Version - 1.3.1 (0)

Current Central Station 2 Version - 4.2.1 (0)

Current Mobile Station 2 Version - 2.7

10月と11月上旬の数週間は忙しかったです。Märklinは、2つのショーに出展しました。コロラド州デンバーのロッキーマウンテンホビーエキスポ、そしてウィスコンシン州ミルウォーキーのトレインフェストで、CS3と新しい機関車のデモを行いました。次のショーは、2018年1月27～28日、マサチューセッツ州ウェストスプリングフィールドで開催されるAmherst Railway Society Railroad Hobby ShowとEuroEastです。

*Digital Clubのすべてのメンバーの皆様へ:安全で楽しいホリデーシーズンをお過ごし下さい..
鉄道模型で遊んで下さい!*

このニュースレターについて

最初の記事では、サウンドデコーダーのプログラミングに興味のある方のために、mDecoderツールを使用したmSD3デコーダーのプログラミングの説明をします。2番目の記事で、Curtisがセンサーの使用に関するいくつかのアイデアを提供し、使用上の限界についても説明します。

mDecoder Tool パート2

要約

前回の記事では、mDecoderツールを使用してmLD3(Märklin Locomotive Decoder 3)の設定およびプログラムする方法について説明しました。この記事では、mSD3(Märklin Sound Decoder 3)の設定とプログラミングの方法、およびサウンドファイルのセットアップ方法を説明します。デコーダー基本情報やモーター設定については説明しません。これらについては、前号(2017年9月-10月、Vol.29 No5)のニュースレターをご覧ください。

目的の機関車または近い機関車のサウンドプロファイルがある場合、Märklinサーバーから簡単にダウンロードできることを指摘しておきます。それが利用できない場合は、構築する必要があります。この記事では、ある機関車にダウンロードできるプロファイルがないため、そのサウンドファイルをゼロから作成しました。

さて、前回の記事の手順に従ったと仮定して、続けてプログラマーの“Sound”タブから始めます。

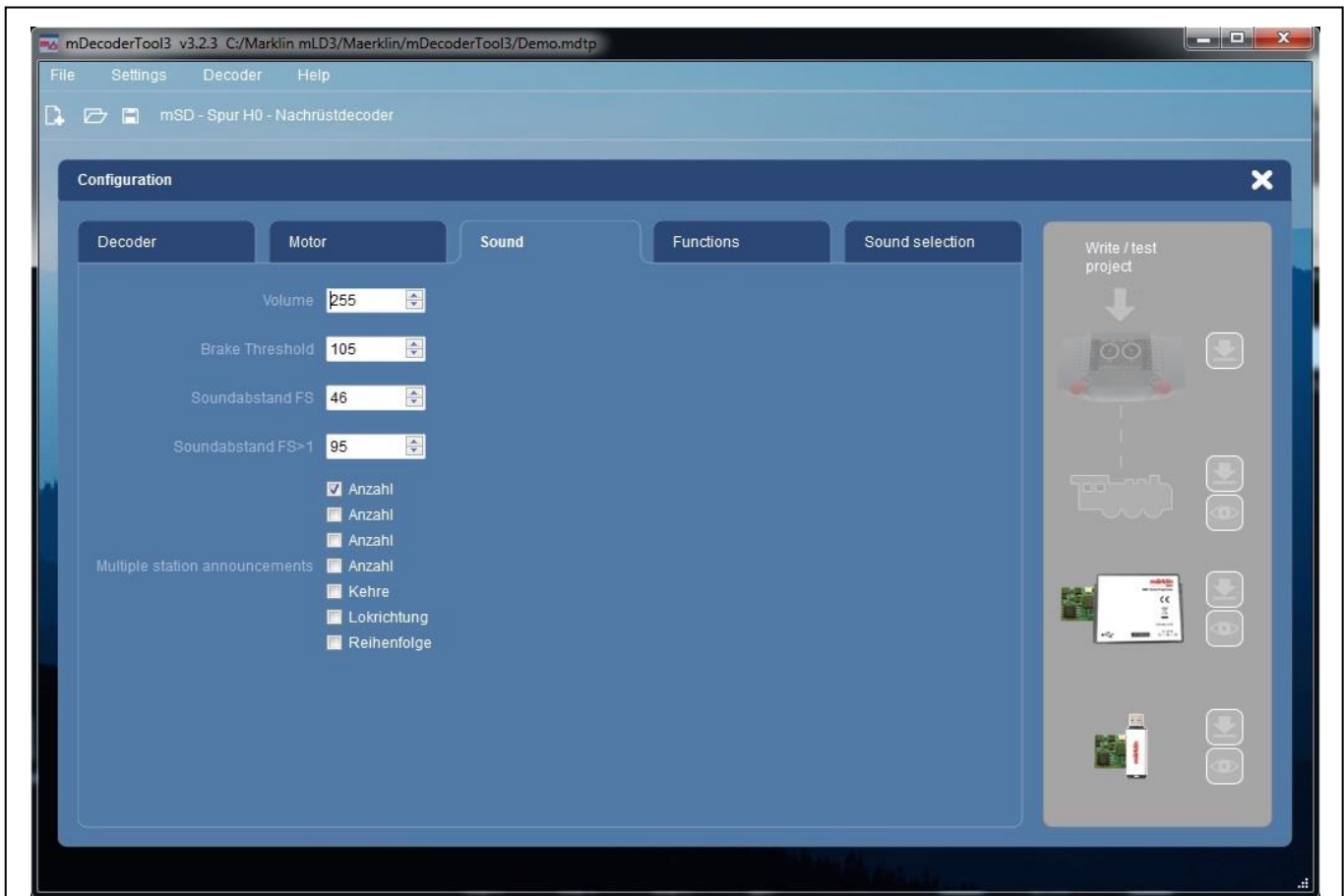


図1 Sound タブ

Sound タブ

「サウンド」タブでは、ブレーキのしきい値とランダムな音の間隔などの他、全体の音量を設定できます。他の変更と同様、変更が必要になるまでそれらをそのままにしておきます(図1)。

次に、“Sound Selection”(サウンドの選択)タブに進みます(図2)。

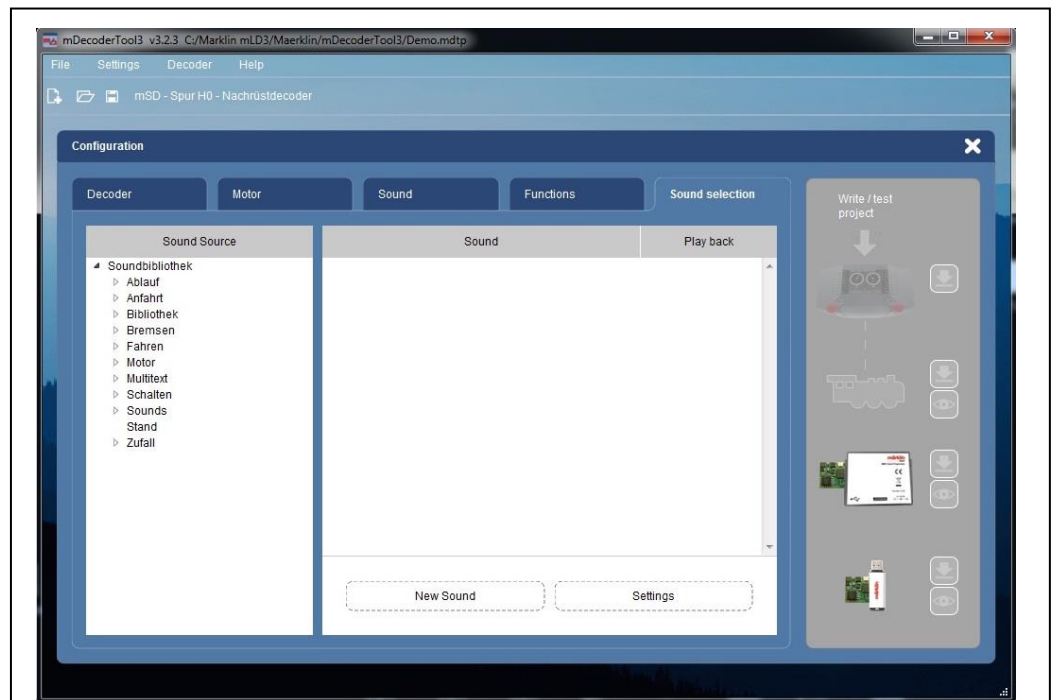


図2 Sound Selection タブ

サウンド設定

“New Sound”(新規サウンド)をクリックして、サウンドリストの作成を開始します。最初に“Fahrgeräusch”(機関音)を選択します。これとその後選択は、必要な数のサウンドを選択するまでプレースホルダー(テンプレート)として機能します(図3)。

次に、“New Sound”を使用して、“Sound without output”のような別のサウンドプレースホルダーを選択します。

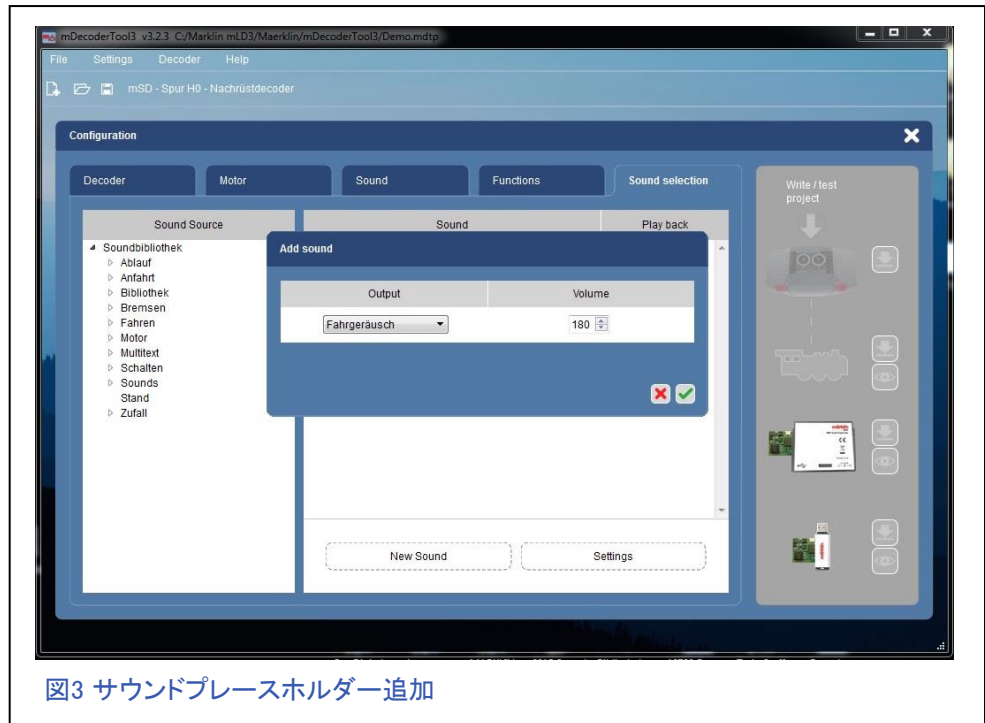


図3 サウンドプレースホルダー追加

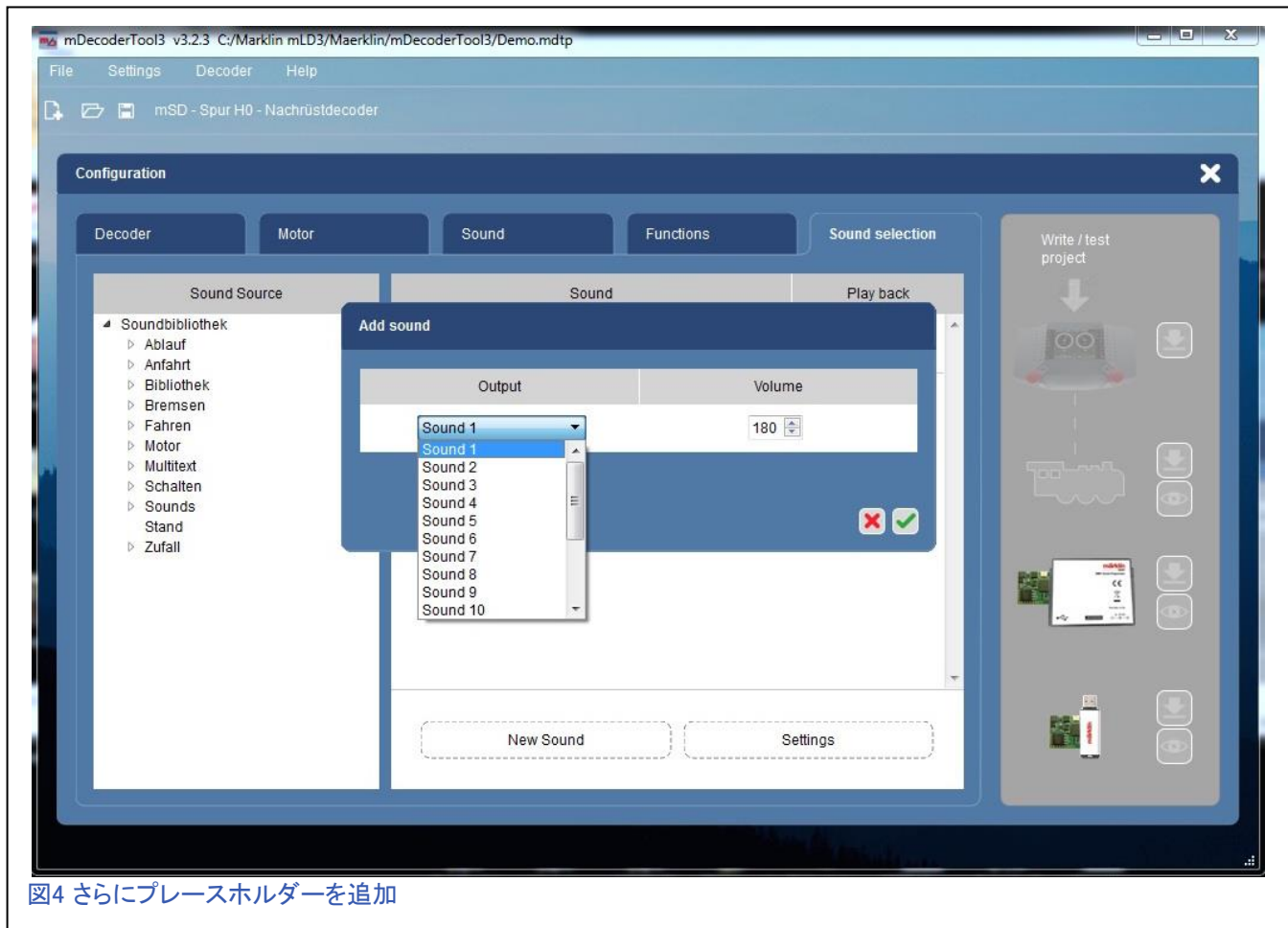


図4 さらにプレースホルダーを追加

“Sound without output”(出力なしのサウンド)は、“Driving”、“braking and shifting”という分類の、機関音に付随する音にとって重要です。さらに“New Sound”ボタンを使用して、サウンドを追加します。サウンド番号は、その後、ファンクションの設定とファンクションマッピングに使用されます(図4)。

サウンド割り当て

必要な数のサウンドプレースホルダーを選択したら、個々のサウンドを割り当てることができます。左の“sound source”(音源)から選択します。たとえば、機関音の場合は“Motor”です。次に、希望の機関車を選択します。それを“Fahrgeräusch”のプレースホルダーにドラッグアンドドロップします。

完全なサウンドライブラリをダウンロードしてあったので、ここでは、“Bibliothek” -> BR 53を選択しています。BR96はないので、BR53を選択しました。これは、軸配置が類似しているためです(図5)。



図5 欲しい機関車サウンドをFahrgeräuschへドラッグ&ドロップ

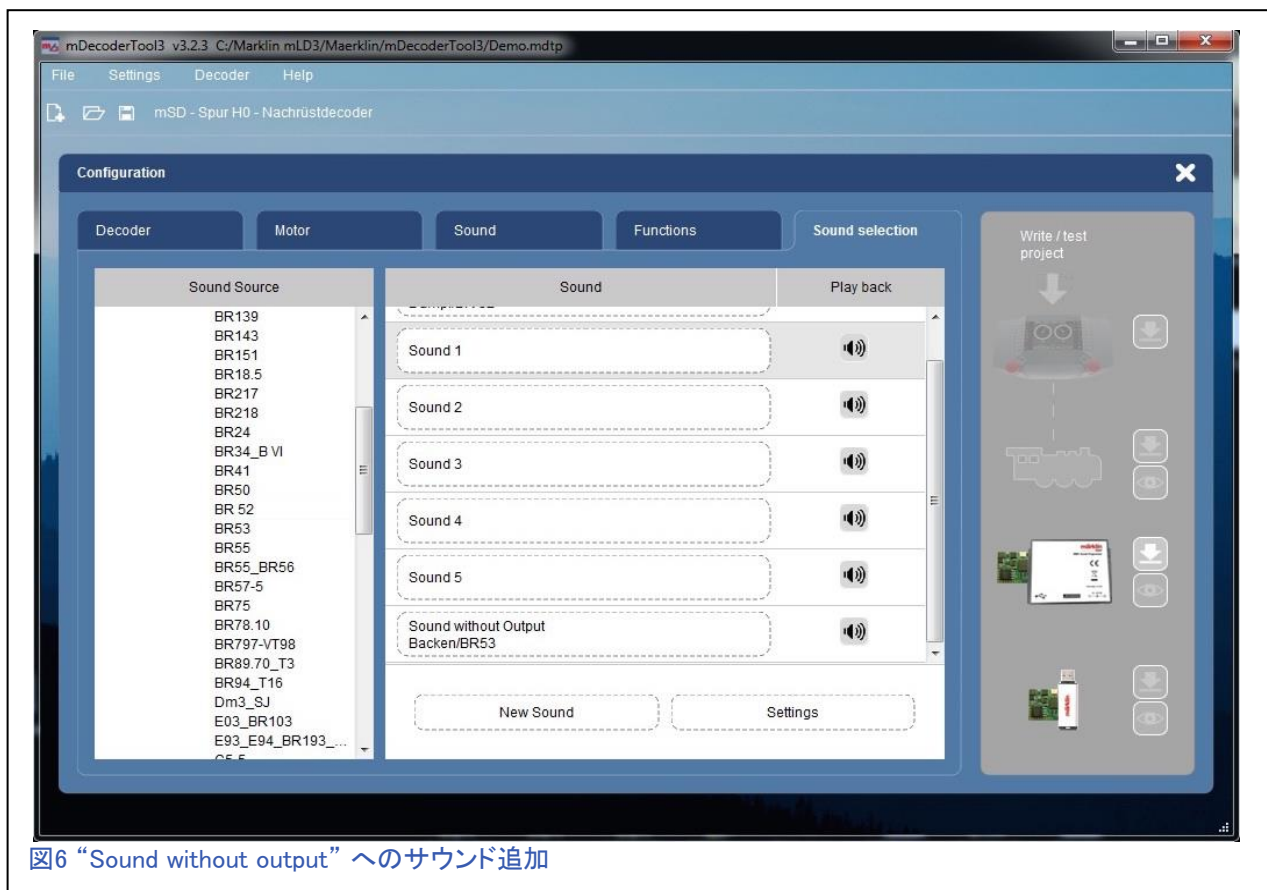


図6 “Sound without output” へのサウンド追加

“Sound without output” プレースホルダーには、approach(アプローチ)、brakes(ブレーキ、“braking/BR 53”など)またはswitching(入換)のカテゴリからサウンドを設定する必要があります。“Sound without output”のプレースホルダーをいくつか作成し、上記のカテゴリのサウンドで埋めることができます。“Sound without output”(出力なしのサウンド)は、ファンクションボタンに割り当てられないため、最大28個のファンクション数にはカウントされません(図6)。目的のサウンドがすべて配置されたら、“Functions”タブに切り替えます。

ファンクション設定

“Function”タブ(図7)から、サウンドをファンクションボタンに割り当てられます。

“Function Buttons”で目的のファンクションを選択します。選択したファンクションキーにアクションを追加するには、右側の列“Actions”各欄の“+”ボタンを使用します。工具ボタンは、それぞれのアクションを設定するために使用されます。ファンクションF0、F1、F2、およびF4には、設定例が事前に入力されています。これらはスキップまたは調整できます。ファンクションボックスをダブルクリックすると、シンボルと動作モード(オン/オフ型または瞬間型)を設定でき、アナログモードでオンにすることもできます。

ファンクションボックスを右クリックすると、フィールドをリセットできます。

“Actions”各欄に追加した設定を右クリックして、個別に削除できます。

ファンクションの割り当てとサウンドアクション

ファンクションボタンにサウンドを割り当てるには、目的のファンクションボックス(たとえばF3)をクリックし、右側パネル“Actions”下の“Sounds”欄の“+”を押します。次に、目的のサウンド(“Sound 2”など)を選択します。機関音(FahrgeräuschDampf / BR 53)を選択し、適当な音量に調整しました(図8)。

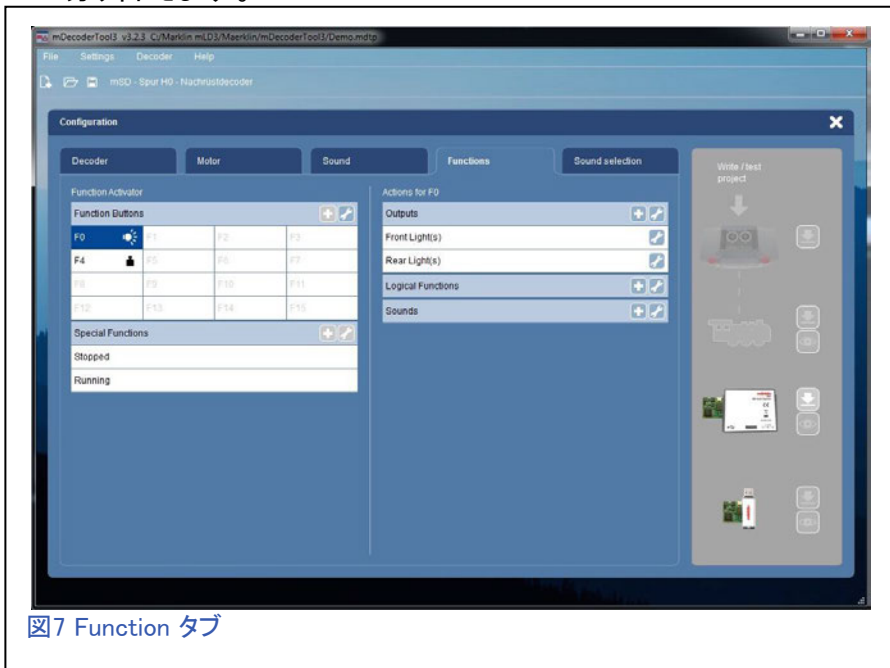


図7 Function タブ



図8 サウンド割り当て

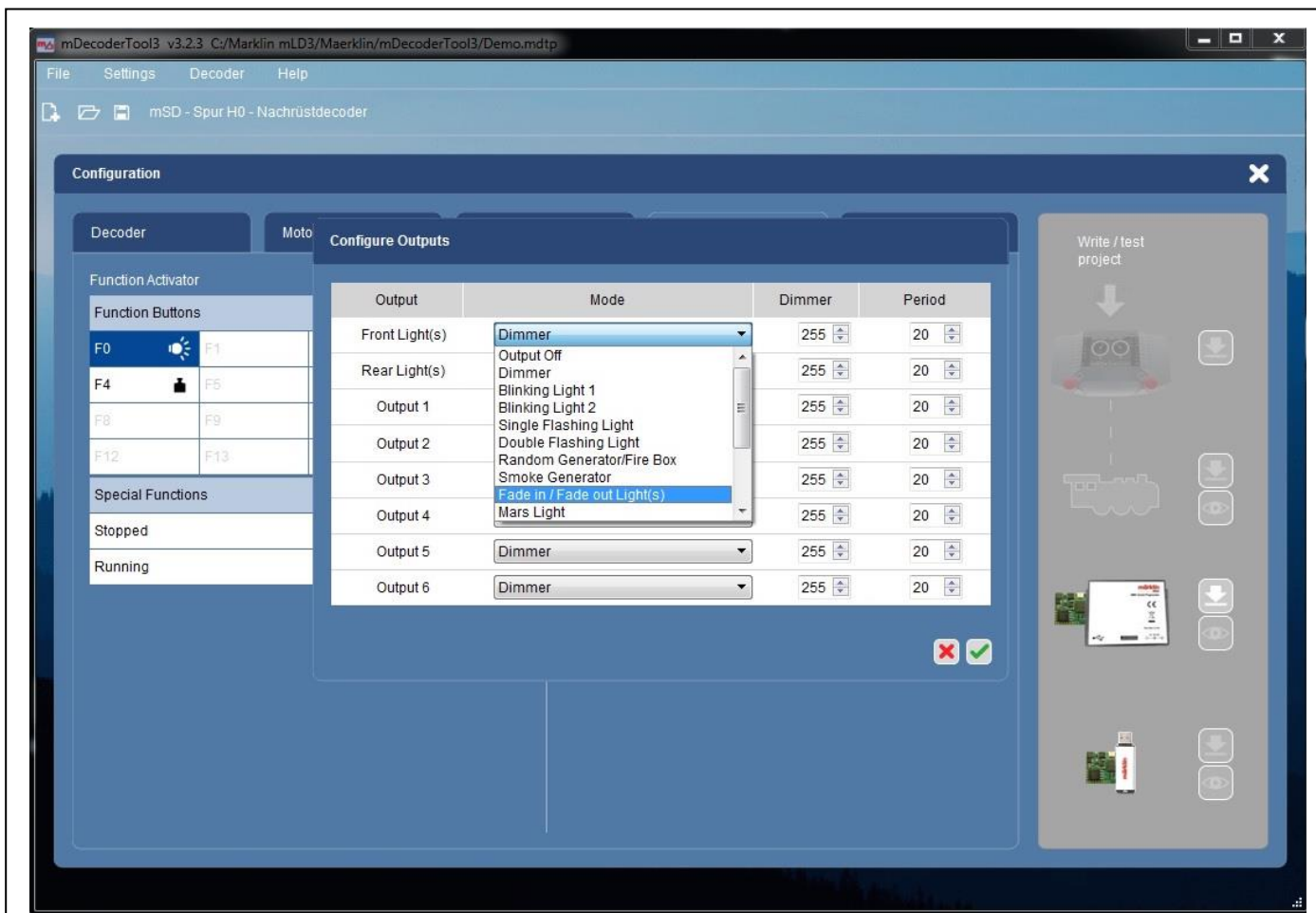


図9 物理的出力の編集

物理的出力の編集

右側パネル“Actions”下の“Outputs”欄の工具ボタンを押すと、物理出力の特性を調整できます。たとえば、ヘッドライトに“Fade in/Fade out Light(s)”(フェードイン/アウト ライト)を選択し、発煙器(Output1)に“Smoke Generator”を選択しました(図9)。

すべての設定を編集したら、プロファイルをデコーダーに書き込みできます。方法は画面右側のウィンドウにあります。このボタンを押して、選択したデバイスへの送信を開始します。デバイスが接続されていない場合、この矢印ボタンは灰色です。

(訳注:デコーダー書き込みには、60971DecoderProgrammer等ハードウェアが別途必要です。)

デコーダーの取り付け

このデコーダーのインストールを示すことも重要だと思います。先ほど言ったように、96型(37964)のデコーダー取り付けは失敗しました。デコーダーのプログラムを実行し、21ピンデコーダーを接続するだけの簡単な作業でしたが、難しい部分はスピーカーでした。

丸穴へ四角い釘

新しいMärklinのスピーカーは素晴らしいです。それらは小型で素晴らしい音を持っています。問題は、古いスピーカーで、丸く、ハウジングも同じだったことです。新しいスピーカーの形状は長方形です(図10)。古いスピーカーを新しいデコーダーで使用しましたが、音は満足できるものではありませんでした。よって、モディファイを加える必要があります。

スピーカーハウジングはデコーダマウントと前照灯ソケットも兼ねるため、変更できません。mSD3に付属のスピーカーは、小型なのでデコーダの下に収まるか見えましたが、収めるにはハウジングをカットする必要がありました。

モデラーズナイフを使って、角形スピーカー用にカットしました(図11)。おおよその大きさに削れたら、ぴったりと収まるように慎重に削りました(図12-13)。あまり削りすぎないことが重要でした。さもないと、曲線通過時にこのハウジングがシャーシの接続動作を邪魔してしまい、走行に問題が生じる可能性があります。

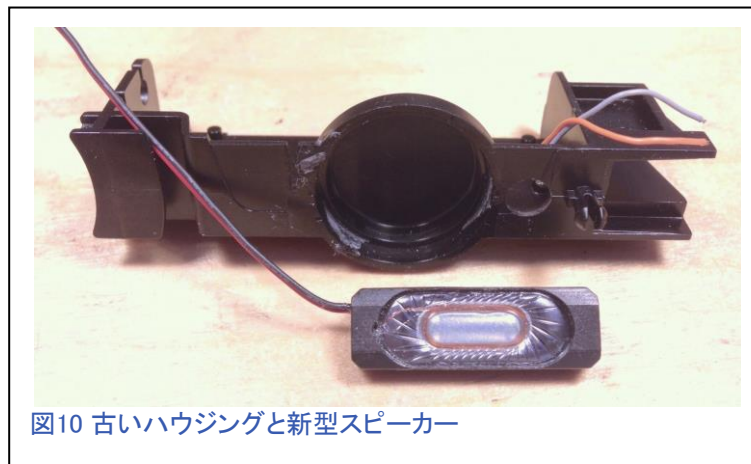


図10 古いハウジングと新型スピーカー



図11 ハウジングの切削

スピーカーが故障した場合、将来ハウジングをスペアパーツとして入手できるかどうか分からないため、スピーカーを固定する接着剤はほとんど使用していません。最終的な組み立てはスムーズに進み、機関車の音は素晴らしくなりました(図14)。

ベンチからもう1つのプロジェクトが！

Enjoy your hobbies!

Rick Sinclair (リック)



図12 スピーカーのおおよそのフィットリング(下から)

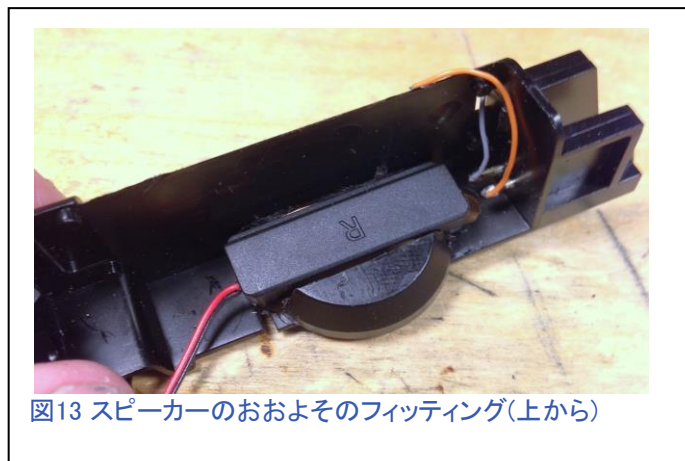


図13 スピーカーのおおよそのフィットリング(上から)

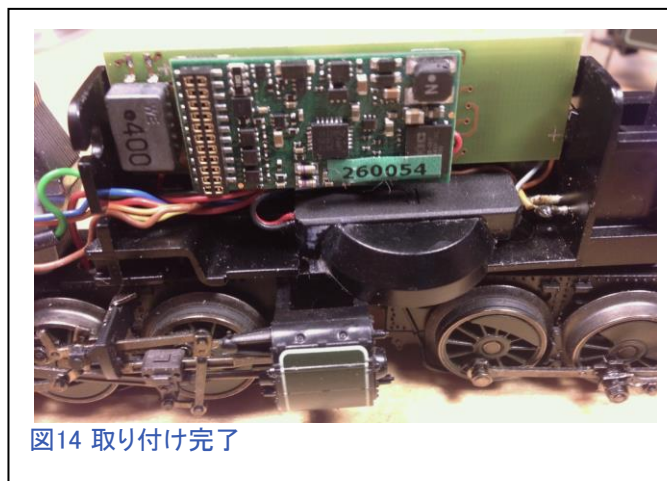


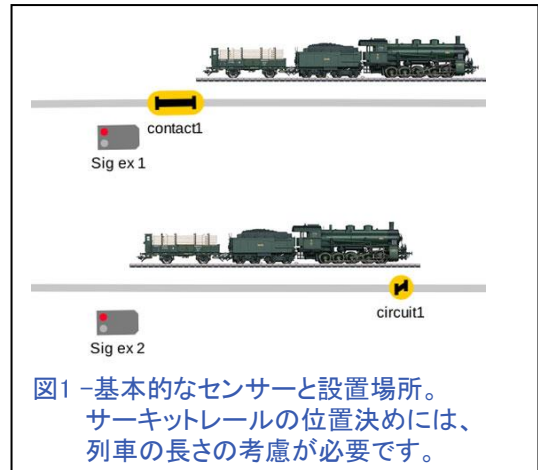
図14 取り付け完了

実践的センサー使用とイベント作成

私の記事の多くで、主にハードウェアとソフトウェアの技術的な観点から、レイアウトの自動化プロセスについて書いています。コンタクトレールまたはフィードバックセンサーがCSに在線情報を送信し、CSは、ユーザーが作成した設定に従って、自動化スクリプトを起動します。今月の記事では、センサーの使用に関するいくつかのアイデアを提供し、使用上の注意事項についても説明します。

基本的なセンサーの種類

一般的に使用されるセンサーには、コンタクトレール、サーキットレール、リードスイッチの3種類があります。リードスイッチは瞬間的な検出で、(LGBとTrixなど)2線式レールでの在線検出に使用されます。Märklinの3線式レールでは、集電シューによってトリガーされるサーキットレールがあり、その検出も瞬間的です。コンタクトレール(別名:レール占有センサー)は、持続的な検出を行います。これらは3線式レイアウトに簡単に設置できます。2線式ユーザーは、ほとんどリードスイッチ式ですが、この記事のサーキットレール式の部分を参考にできます。これは、どちらも瞬間的な検出という点で共通だからです。



持続型と瞬間型の違いは、レイアウト上でのセンサー位置決めに直接影響するため、注意することが重要です。図1は、単純な閉塞区間における配置の違いを示しています。‘contact1’は、信号機を「赤」に切り換える前に、‘Sig ex 1’を完全に列車編成が通過したことを検知するために配置されます。‘circuit1’も‘Sig ex 2’を完全に列車編成が通過したことを検知するために同様に配置されます。ただし、‘circuit1’の配置の1つの欠点は、許される列車の長さの制限があることです。たとえば、‘circuit1’の例の列車に車両を追加すると、‘Sig ex 2’を完全に列車編成が通過したことを保証できないことがわかります。

基本的な閉塞制御

閉塞区間は、この記事では、1編成の列車の占有のみを許可する線路区間として定義されています。つまり、1つの閉塞区間内で2編成の列車が存在することは許可されていません。これを制御するために、センサーを使用して、列車が背後の区間をクリアしたことを通知します(後続の列車に進行を許可します)。

図2では、空き区間(‘Sig ex 0’ と ‘Sig ex 1’ の間の線路)と対応するセンサーの位置(‘contact1’)を示しています。下の線路は、サーキットレール(‘circuit1’)を使用したセットアップです。列車が前の区間をクリアしたことを保証するために、‘circuit1’は‘Sig ex 2’のずっと先に配置されていることに注意して下さい。これは、通常、シューが先頭の機関車にあるためです。

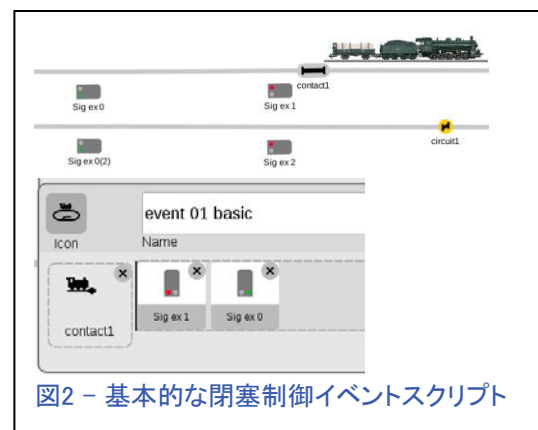


図2の下半分は、‘contact1’ のイベントスクリプトを示しています。‘circuit1’ のイベントスクリプトも同じです。スクリプトは、‘contact1’ がアクティブになったときに‘Sig ex 1’を赤に切り換え、次に‘Sig ex 0’を緑に切り換えます。イベントステップは、表示されている順序に実行されます。表示されていないイベントスクリプトの詳細の1つは、‘contact1’ が‘Departure(出発)’型トリガーに設定されていることです。リードスイッチやサーキットレールにはこのような設定はありません。このトリガー型については次の例で説明します。

実物と同じ現示変化に

図2の信号現示の変更手順を調べると、列車が前の閉塞区間をクリアしたときに、両方の信号機(‘Sig ex 1’ & ‘Sig ex 0’)が同時に変化することがわかります。一方実車では、機関車が通過し始めるとすぐに‘Sig ex 1’が赤に変わります。信号機を実物のように制御するために、図2のスクリプトを、トリガー設定の異なる2つのスクリプトに分割しました。これらの設定を図3に示します。‘Settings(設定) contact1’ ウィンドウを開くには、イベントシーケンスの左にある‘contact1’アイコンをクリックします(図3上半分)。



図3 - トリガー型を到着と出発に分割

図3の下半分は、それぞれのスクリプトのトリガー設定を示しています。選択肢は‘Arriving(到着)’と‘Departing(出発)’です。「到着」スクリプトは、列車の車輪がコンタクトレールと接触したときに起動されます。同じ列車の車輪がコンタクトレールから外れると、「出発」スクリプトが起動されます。

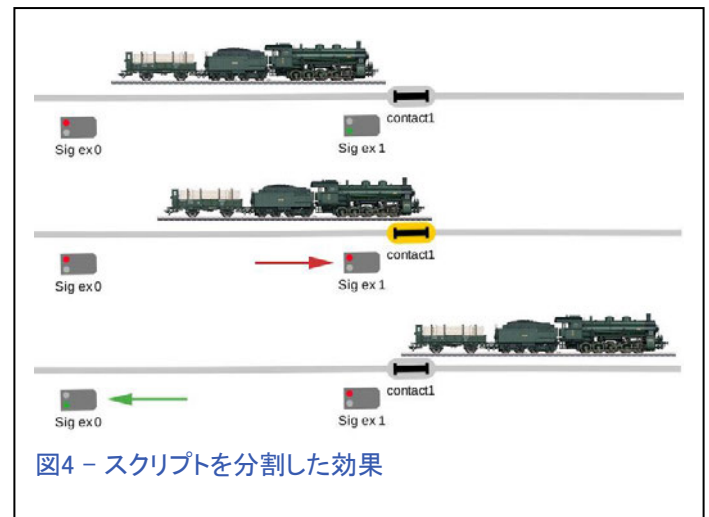


図4 - スクリプトを分割した効果

図3のスクリプト分割の効果を図4で説明します。上側には、‘contact1’ と接触する前の信号機が示されています。中央には、‘contact1’ の到着スクリプトの効果が示されています。下側には、‘contact1’ の出発スクリプトの効果が示されています。

この例のように、イベントを到着スクリプトと出発スクリプトに分離するのは基本的で簡単です。前述のように、この設定はリードスイッチまたはサーキットレールでは使用できません。1つのコンタクトレールでできることをサーキットレールで行うには2つが必要です。2つのトリガーを分離できるため、より高度で複雑なイベントスクリプトを作成できます。この記事の残りの部分では、トリガー設定とs88コンタクトの配置のより高度で技術的な例をいくつか示します。

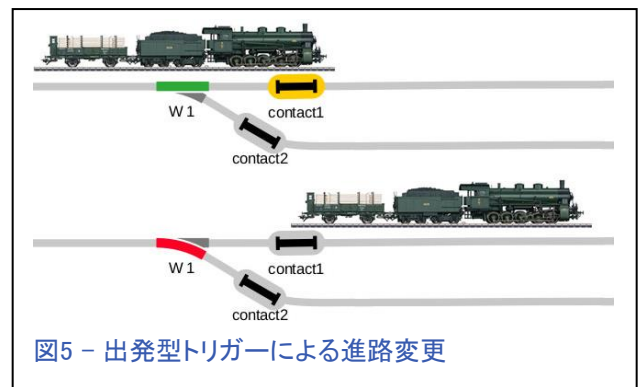


図5 - 出発型トリガーによる進路変更

代替ルート

次の例では、列車を別の線路に進路変更する方法を示します。この方法を使用して、列車を空いている線路に進路変更できます。覚えておくべきことは、「制御」列車はその背後の列車を進路変更するために使用されるということです。列車の進路をあなたが直接制御していないため、レイアウトの自動化にランダム性が生まれます。図5は、ポイントの後ろのコンタクトレールの配置を示しています。空き線路側にポイントを切り換える前に、(‘contact1’ の) 出発トリガーにより、列車がポイント‘W1’を十分にクリアしたことを保証しています。図6は、各コンタクトのスクリプトを示しています。どちらも「出発」トリガーに設定されています。両方のイベントは、ポイントを互いに反対の線路に切り換えるようにプログラムされています。

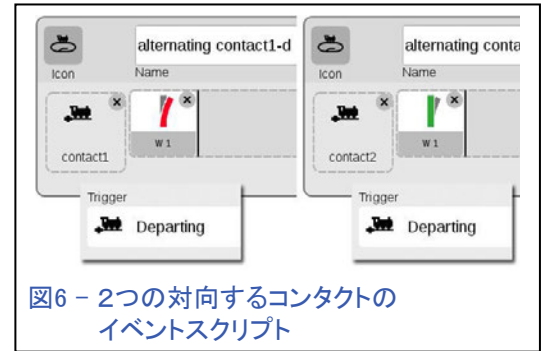


図6 - 2つの対向するコンタクトのイベントスクリプト

ダイヤモンドクロッシング

ダイヤモンドクロッシングでは、2つの異なるコンタクトで到着トリガーと出発トリガーを分離し、‘Sig ex 1’を制御します。前の例のように、‘contact2’は出発トリガーを使用して、交差する列車がクロッシングを完全に通過したことを保証します(図7および8)。

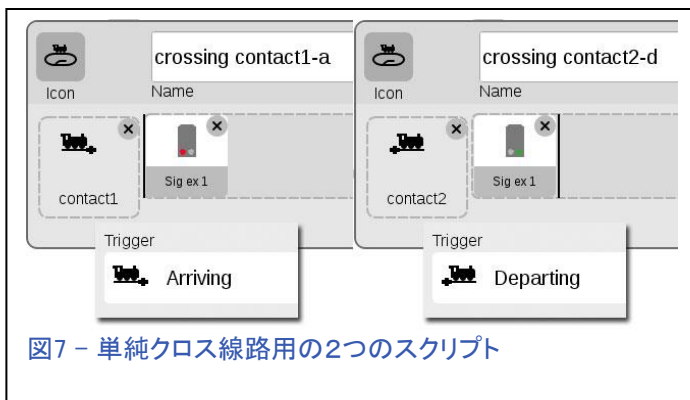


図7 - 単純クロス線路用の2つのスクリプト

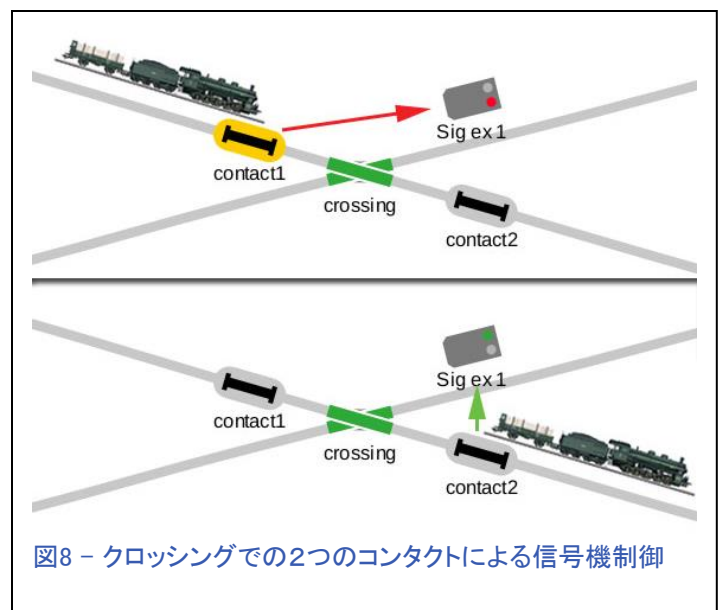


図8 - クロッシングでの2つのコンタクトによる信号機制御

駅の停車とバイパス

次の例では、1つの閉塞区間内でいくつかのイベントを組み合わせています。例は、本線から分岐した停車駅を作成し、貨物列車が停車駅をバイパスする方法を示します。複雑なため、わかりやすくするためにイベントを分割します。例自体は、効果に対して単一の制御アプローチであり、制御メカニズムを強調するために簡略化されています。

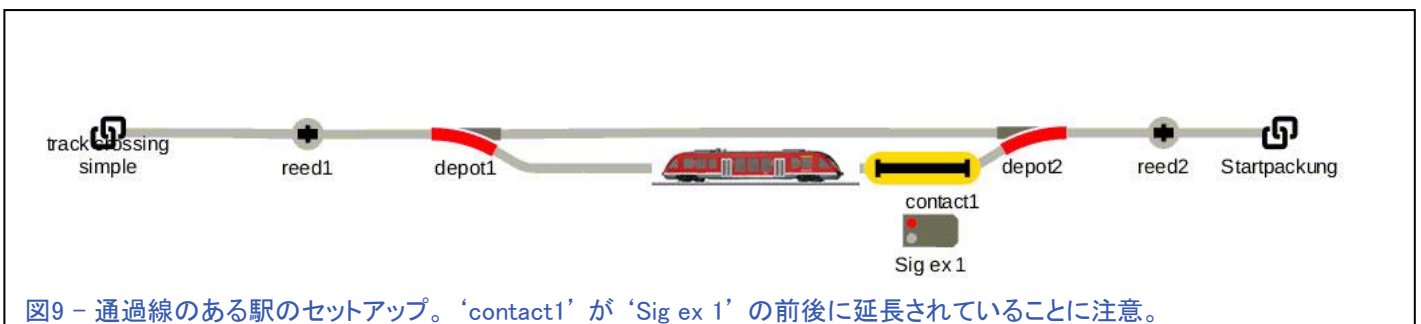


図9 - 通過線のある駅のセットアップ。‘contact1’が‘Sig ex 1’の前後に延長されていることに注意。

図9では、駅に‘Sig ex 1’を配置しました。レール関係では、コンタクトレールの長さを‘Sig ex 1’の前後に延長しました。通常の閉塞区間用コンタクトでは、信号の後ろにしかコンタクトレールありません。注:図は、コンタクトレールを長く描いています。これは記事用に編集したもので、CSIにはそのような長いコンタクトのアイコンは存在しません。

図10は、「到着」トリガーイベントスクリプトを示しています。スクリプトは、「列車が到着すると、10秒のタイマーを開始します。タイマーがカウントし終わると、信号機は緑にされ、列車は出発できます。」

‘contact1’ が信号機‘Sig ex 1’の手前側に延長された理由は、信号機が赤なので信号機に到達する前に列車が停止してしまうためです。もしコンタクトが信号機の後ろにしかない場合、列車はスクリプトを起動できず、遅延タイマーを開始できません。

なお、スクリプトに遅延ステップを追加するには、イベント編集パネルの‘Add’→‘Text(テキスト)’をクリックします。

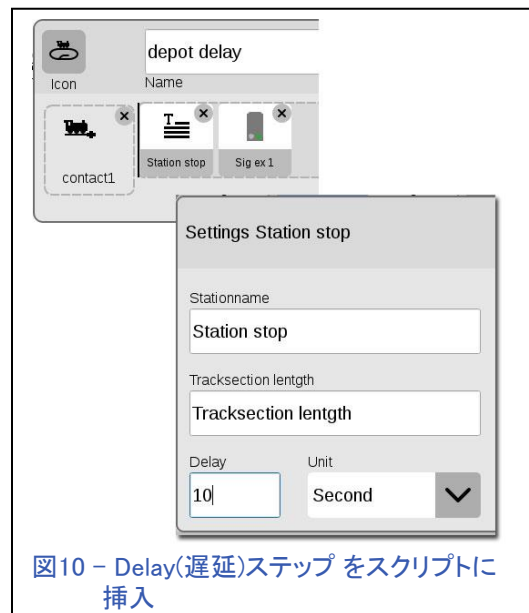


図10 - Delay(遅延)ステップをスクリプトに挿入

従来は、「到着」トリガーを使用して、信号機を赤にしていたが、今回はタイマーを開始し、‘Sig ex 1’を緑に切り換えるスクリプトが既に起動されているため、‘contact1’の「出発」トリガーを使用する別のスクリプトを作成しました。これにより、‘Sig ex 1’が赤にリセットされます(ここではスクリプト例の図はありません)。2つのコンタクトレールを使用して同じ事ができますが、ご覧のとおり、1つのコンタクトレールを延長してほぼ同じ効果を達成できました。図11は、代わりにサーキットトラックを設定する方法を示しています。この場合、‘circuit1’を信号停止区間の前に配置する必要があります。

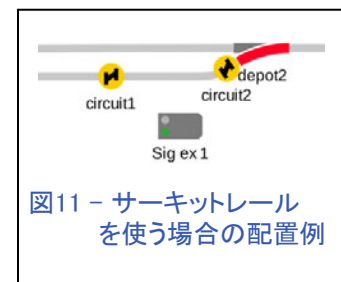


図11 - サーキットレールを使う場合の配置例

この例の2番目の着目点は、「通過」イベントです。図9で気づいたかもしれませんが、本線に2つのリードスイッチ(‘reed1’と‘reed2’)を配置しました。これにより、停止する必要のない列車に磁石を付けることで、駅を通過できます。図13(上)は、‘reed1’がイベントを起動し、ポイント‘depot1’と‘depot2’を直進に切り換えることを示しています。図13(下)は、‘reed2’を使用して駅側にポイントをリセットします。図12は、各リードセンサーのスクリプトを示しています。



図12 - 図13の駅でのポイント切り換えスクリプト (左)通過列車検知 (右)ポイント戻し

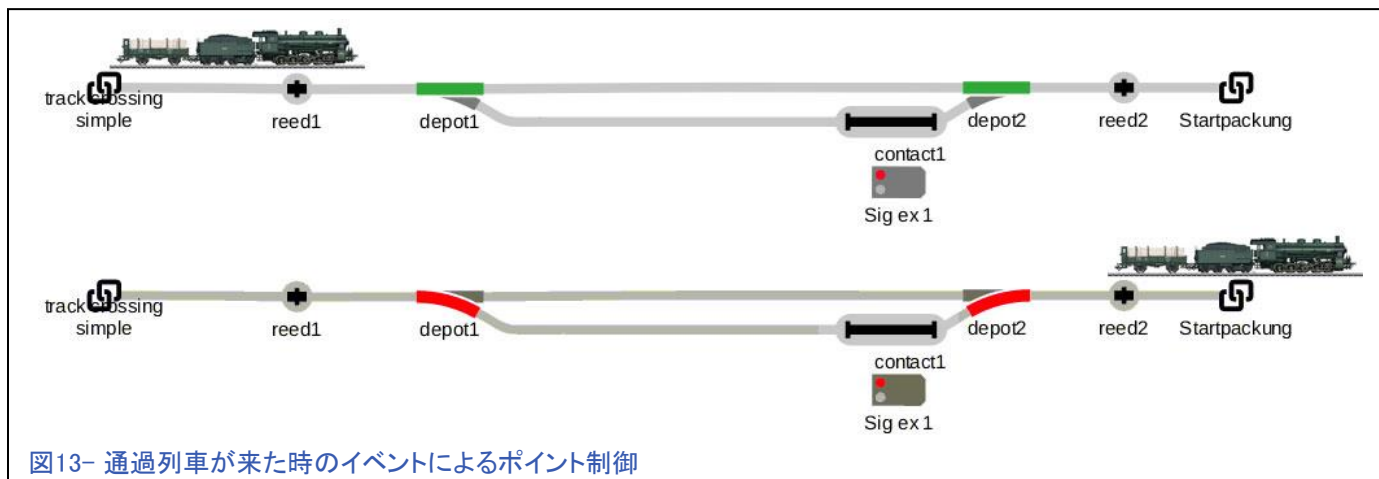


図13- 通過列車が来た時のイベントによるポイント制御

単純なシャトル運転

2つの終着駅間を行き来する単純なシャトルを運行するには、2つの単純なイベントスクリプトを作成します。CS2には、各機関車毎にシャトルを設定するボタンがあります。しかし、これはCS3からは削除されました。これはスペックダウンに思えるかもしれませんが、機能と使用感においてより堅牢でより良い方法があることを保証します。この方法をCS2に適用して、シャトル機能を廃止することもできます。最初に簡単なシャトル操作について説明します。

シャトル運転の場合、各終点到にコンタクトレールを設定する必要があります。イベントスクリプトは「到着」トリガーを使用します。スクリプト内容は：シャトル列車が終点到に到着し、停車、ポーズ(遅延)、方向転換、再度ポーズ、そして反対方向に加速して本線に戻ります。概略、センサーは図14のようなセットアップとなります。ここでは「到着」トリガーを使用することに注意して下さい。



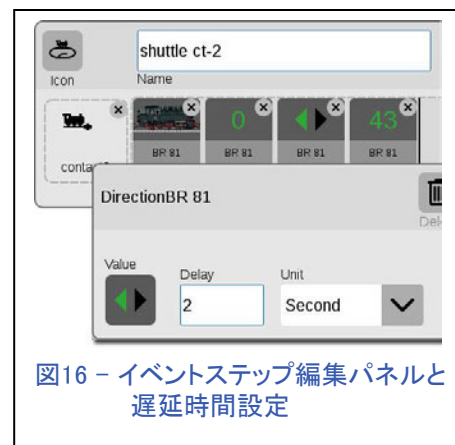
各コンタクトが起動するイベントスクリプトは基本的に同じです。基本的にと言うのは、例えばベルや警笛を追加することはできますが、(文字通り)基本だけを説明するからです。これらのスクリプトについて注意すべき2つの点があります。

まず、各コンタクトに割り当てた進行方向転換(各スクリプトの3番目のステップ、図15)です。機関車が転換しない場合があります。これを変更するには、ステップ自体をクリックしてステップ編集パネルに入ります(図16)。「Value(値)」ボタンをクリックして方向を変更できます。次の加速ステップの前に2秒の遅延(ポーズ)を追加しました。



これらのスクリプトについて注意する2番目の点は、スクリプトに機関車を割り当てたことです。警告というのは、これらは今のところ機関車を割り当てる必要があったスクリプトの唯一の例です。この警告については、記事の最後で説明します。

このタイプのシャトルスクリプトがCS2に比べて非常に優れている点は、機関車を簡単に変更できることです。別のシャトル列車を使用する場合は、スクリプトを再作成する必要はありません。機関車を変更するには、機関車のステップをクリックして、ステップ編集パネルを開きます。次に、「Locomotive(機関車)」プルダウンメニューをクリックすると、機関車リスト内のすべての機関車を表示しますので、シャトル運行する機関車を選択するだけです。反対側の終着駅のスクリプトも変更することを忘れないで下さい。



ここまでのすべての例は、単一のイベントが信号機やポイントを切り換え、機関車の進路を設定する簡単な手順を示しました。それらの例では言及しなかったいくつかの概念について明確にしたいと思います。1つ目は、私が使用する信号機用イベントは、信号機制御部分と停止制御部分に分けています。信号機制御から独立してブレーキモジュールをアクティブにできるスクリプトを作成することができます。第二に、シャトル列車の例を除き、どのイベントも特定の機関車に限定されないことに注意して下さい。「駅の停車と通過」の例のリードスイッチは、特定の機関車だけに紐付きませんが。前述したように、この重要な事実については記事の最後で説明します。

これらの例は、センサーを設置し、レイアウトを自動制御するスクリプトを作成する方法を示しています。このような制御を実際に使用して、これを超える新手法を発見して、レイアウトに新しい機能を導入するインスピレーションを皆さんが得られることを願っています。3つの異なるセンサー(コンタクト、サーキット、およびリードスイッチ)を使用すると、それらを混在できることがわかると思います。ただし、それぞれの長所と制限に注意する必要があります。この記事では、初心者レベルのセンサーとイベントスクリプトについて説明しています。次回の記事では、使用できる中級から上級レベルのスクリプトの例をいくつか紹介します。



図17 - ステップエディターでの機関車再選択

自動制御の神話

駅にいますと、列車が減速して近づき、警笛を鳴らしたり、ベルが鳴り止んだりする経験をしていると思います。そして、発車するときは、逆のプロセスを実行します。これらの一連の動作は、CSを使用して簡単にスクリプト化してイベントにできます。

図18は、駅への機関車の入場のスクリプトを示しています。線路上の列車の位置と対応するステップの一致に注意して下さい。センサーを使用してこのような動作を自動化しようとする、最終的に失敗します。表示された例そのままなら一応機能しますが、完全に機能すべきレイアウト用としては問題があります。

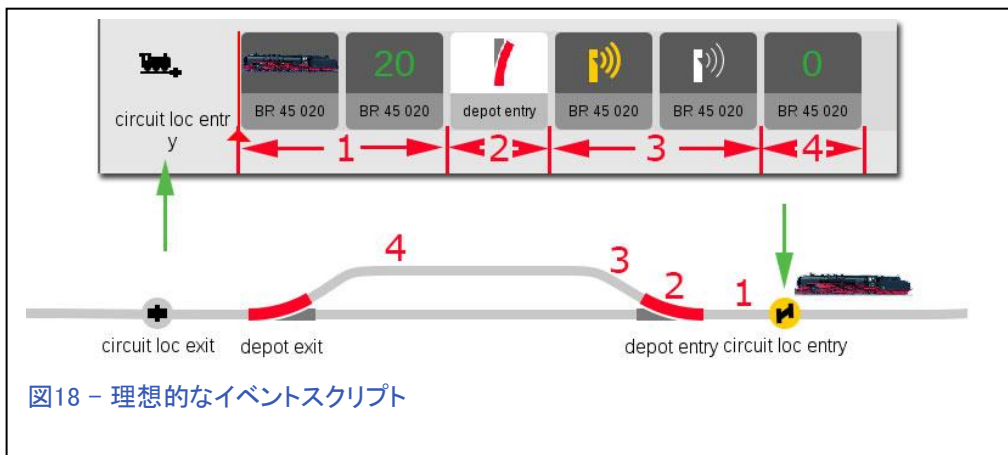


図18 - 理想的なイベントスクリプト

これが失敗するであろう主な理由は、CS3が現在の機関車の位置を認識していないことです。図18の例では、センサーの近くにいる唯一の機関車であるため、'circuit loc entry'のアクティブ化が正しいように見えることに注意して下さい。特定の機関車だけが、このイベントによって制御されるということです。

