

最新ソフトバージョン

Central Station 3	- 2.5.2 (6)
Central Station 2	- 4.3.0 (34)
Mobile Station	- 4.15
Wireless MS	- 4.27.0.13

デジタルコンサルタント

Rick Sinclair
Curtis Jeung

ミシガン州ノバイで開催されたNMRAナショナル・トレイン・ショーから帰ってきたばかりです。ETEグレートレイクス支部は、モジュールレイアウトで最優秀賞を受賞しました。カーティスと私は、このモジュールに多大な時間を費やしてきました。この栄誉と成果を共に分かち合えることを大変嬉しく思います。

最初の記事では、新しく発売された30601 D&RGW F-7 A機関車について、2番目の記事では、複雑なレイアウトをCS2からCS3へ移行する作業がほぼ完了したことをお伝えします。

新しい機関車が届きました

正直に言うと、私はメルクリンのUSモデルが好きで、できるだけたくさん手に入れようとしています。30601 D&RGW機関車が届いた時は、本当に興奮しました。なぜかという、これは私のお気に入りの車体色の一つだからです(図1)。レトロな外箱と、外箱を保護するプラスチック製のスリーブが付いているのが気に入っています。箱をきれいに保つのも簡単でしょう。

箱を開けて機関車を見てみましたが、期待を裏切らない出来栄でした。塗装は美しく鮮やかで、昔のオリジナルモデルとは4本のピンストライプと銀色の屋根が異なります。



図1 – メルクリン 30601

当然のことながら、まずは線路に乗せてモーターをテストし、それからライトを点灯させました。ヘッドライトは明るくて素晴らしいですが、ヘッドライトファンクションしかなく、音も出ないことに気がきました。これは、オリジナルのリオグランデ機関車(3062)がアナログで、1976年から1984年までの製造だったためヘッドライトしか点灯しなかったためだと思います。メルクリン社は旧製品の再現ということで、オリジナルモデルと同じ仕様にするを選んだようです。

ボディを外すと、最新の60972デコーダとヘッドライト用のLED回路基板があることに気づきました。奇妙に思ったのは、回路基板にヘッドライト上部と下部、そして車番灯用のLEDが取り付けられていたことです(図2)。しかも、それらはデコーダー取り付け基板にも配線されていました(図3)。このことから、デコーダーから点灯制御できるのではないかと考えました。

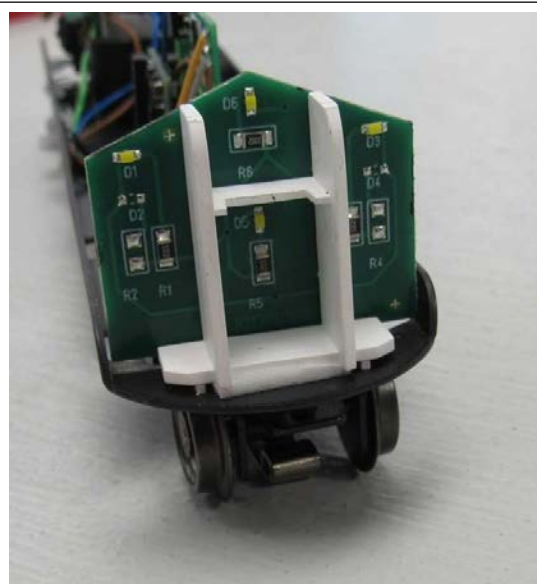


図2 - LEDライトボード

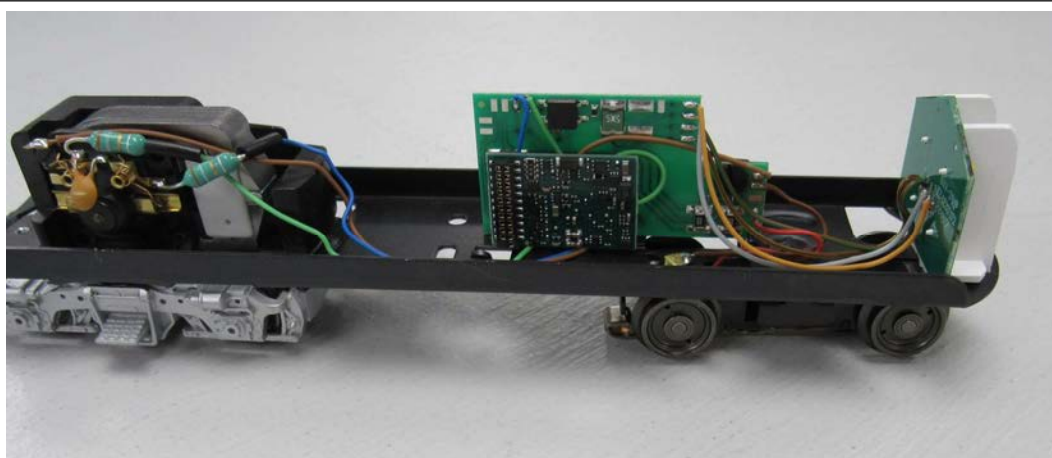


図3 - デコーダーベース基板とLED回路基板間の配線

私はすぐにテスターでベース基板の電極がデコーダーまで配線されていることを確認し、少しファンクションマッピングを行うことで、それらを制御できると確信しました。

再マッピング

デコーダーのどの出力がどのライトに接続されているかを調べました。その結果、Marsライト(上側ヘッドライト)は“Front Light(ライト前)”、車番灯は“Aux2(出力2)”、(下側)ヘッドライトは“Aux3(出力3)”であることがわかりました。

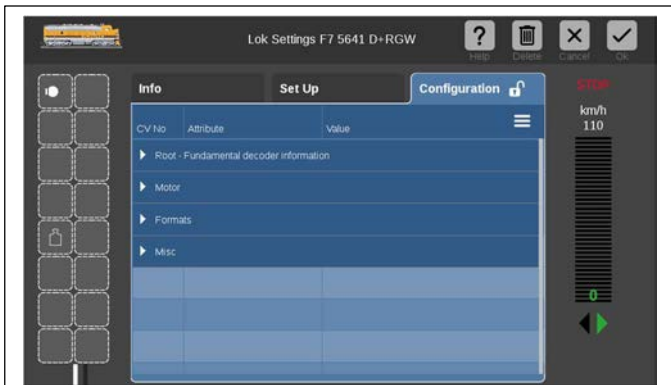


図4 - “Configuration(CV設定)” タブ

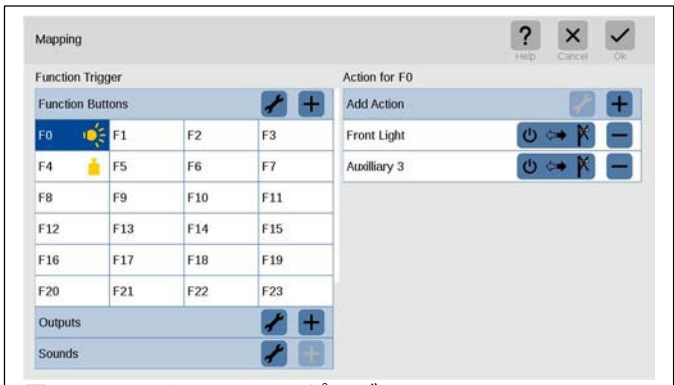


図5 - ファンクションマッピング画面

あとは、すべてのライトを制御および設定可能にするために、少しマッピングとアイコンの割り当てを変更するだけです。

必要なのは、機関車の編集モードに入り、“Configuration(CV設定)” タブをタッチします。そこでデコーダーのCVが読み込まれます(図4)。CS3がCVの読み取りを完了したら、任意のファンクションアイコンをタッチしてファンクションマッピングに入ります(図5)。

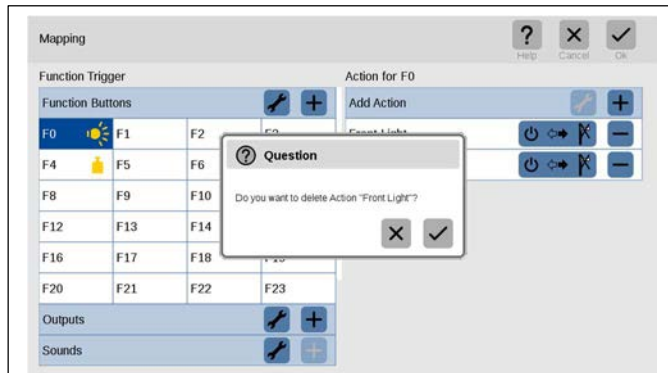


図6 - “Front Light” 出力の削除

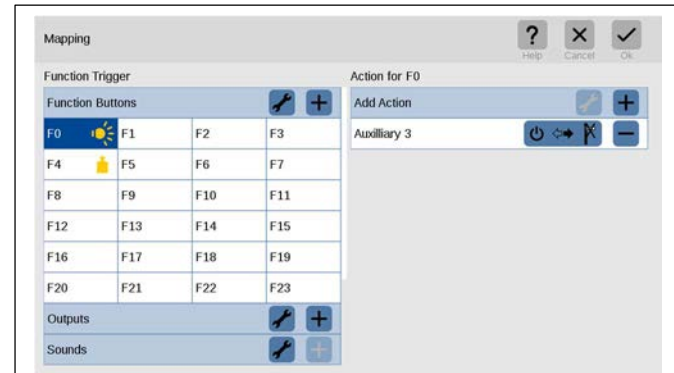


図7 - “Aux 3” 出力のみとしました

F0の機能マッピングを見ると、このボタンが“Front Light(ライト前)”と“Aux 3(出力3)”をアクティブにしていることがわかります。この2つの出力を分割するので、“Front Light(ライト前)”の右側にあるマイナス(-)をタップします(図6)。これにより、ファンクションボタンF0から上側ヘッドライトが削除され、出力3(下側)だけがヘッドライトとして機能します(図7)。

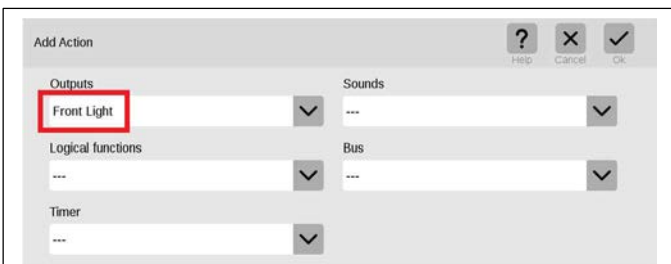


図8 F1に “Front Light(ライト前)” を選択

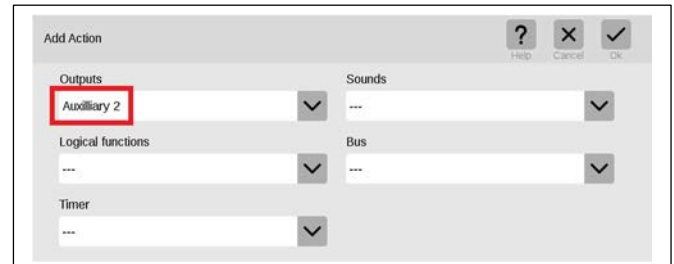


図9 F2に “Auxiliary 2(出力2)” を選択

その後、F1のボックスをタッチし、右側の“Add Action(アクションを追加)”の“+”をタッチして、“Outputs(出力)”プルダウンメニューで“Front Light(ライト前)”を選択します(図8)。次に“F2”ボタンをタッチし、再び“Add Action(アクションを追加)”を選択しますが、アクションは“AUX 2(出力2)”を選択します(図9)。すべて正しく設定できていれば、F0をタッチすると出力3(下側ヘッドライト)がオン・オフし、F1をタッチするとライト前(上側ヘッドライト=Marsライト)がオン・オフし、F2をタッチすると出力2(車番灯)がオン・オフします。

ファンクションアイコンの選択

次に、ファンクションボタンに適切なアイコンを選択します。ファンクションボタン上部の“Function Buttons(ファンクションボタン)”の工具ボタンをタッチしてアイコンを選択します。この画面からすべてのファンクションのアイコンを設定できます(図10)。

F1にはマーズライト用の“Front Light(車外灯前)”アイコン、F2には“Number Board(車番灯)”アイコンを選択しました(図11)。どちらも“Switching Function(オン・オフ型)”のままにしました。



図10 - ファンクションアイコンの選択



図11 - 選択されたアイコン

出力設定

次に、ファンクションの出力を設定する必要があります。ヘッドライトとナンバープレートのライトは点灯・消灯時にフェードイン・フェードアウトし、マーズライトは点滅させる必要があります。

左下の“Outputs(出力)”行の工具ボタンをタッチすると、「編集モード」に入ることができます(図12)。

ここから、“Front Light(ライト前)”に“Mars Light (マーズライト)”を選択し、Auxiliary 2(出力2)と3(出力3)に“Fade in / Fade out Light(フェイドイン/アウト)”を選択し、“Period(周期)”をそれぞれ25と23に設定します。

図13は、“Front Light(ライト前)”に“Mars Light (マーズライト)”を選択し、Auxiliary 2(出力2)と3(出力3)に“Fade in / Fade out Light(フェイドイン/アウト)”を選択し、フェードタイムに若干の差を付けたいため“Period(周期)”をそれぞれ25と23に設定した結果を示しています。



図12 - F1出力の編集モード

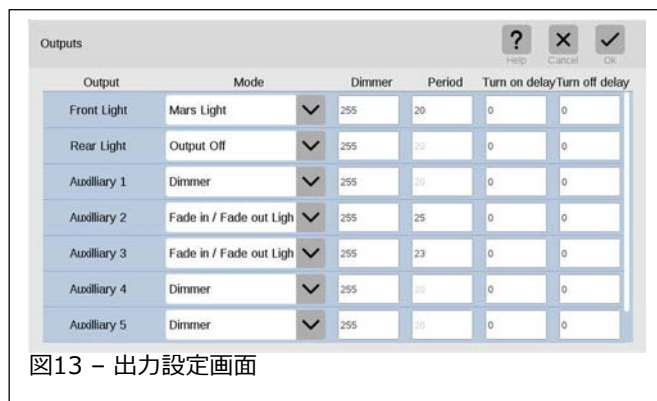


図13 - 出力設定画面

これらの変更を行った後、機関車は新しいライト設定で走行する準備が整います。

注意すべき点は、デコーダーの“reset(リセット)”を実行すると、これらの変更が失われ、再度実行する必要があることです。

考察

工場がデコーダーをなぜそのようにプログラムしたのか、しばらく考えた後、これは昔のモデルを現代風にアレンジしたものなので、顧客が望むライトをカスタマイズできるように工場が作ったのではないかと思います。これらのライトには様々な設定ができます。ですから、これが最良の選択肢であり、顧客が好きなものを選ぶようにしたのだと私は思います。

もう1つの優れた点は、21ピンのデコーダー取付け用ベース基板があるため、サウンドデコーダーの取り付けが非常に簡単なことです。スピーカーケーブルをデコーダーベース基板に半田付けするだけで、文字通りプラグアンドプレイで取り付けできます。

非常に多くのライトの可能性を備えた新しいモデルを楽しく試すことができました。皆さんが自分の好みに合わせてライトをカスタマイズすることを願っています。

Enjoy your hobbies!

Rick Sinclair

2列車単線シャトルの改修

この記事では、2列車単線シャトルの再プログラミング手順を紹介します。このシステムは元々CS2上で開発されましたが、CS3への移植にあたり技術サポートなしではユーザーが列車を交換できないため、システムの改修が必要になりました。これは、旧式のCS2アーキテクチャにおけるプログラミングの制限が一因でした。ここで使用する線路図を図1(下)に示します。

最初のステップは、在線センサー(コンタクト '24(18)')を削除し、シャトルのルーティング(進路制御)に関連するすべてのイベントを削除することでした。これは、古いコードとの互換性の問題を解決するよりも、白紙の状態から始める方が簡単だと考えたためです。'Lok...' イベントは単なる機関車コマンドであり、線路のルーティングとは関係がないため、そのまま残します。図2は、以前使用されていたイベントの大部分を示しています。

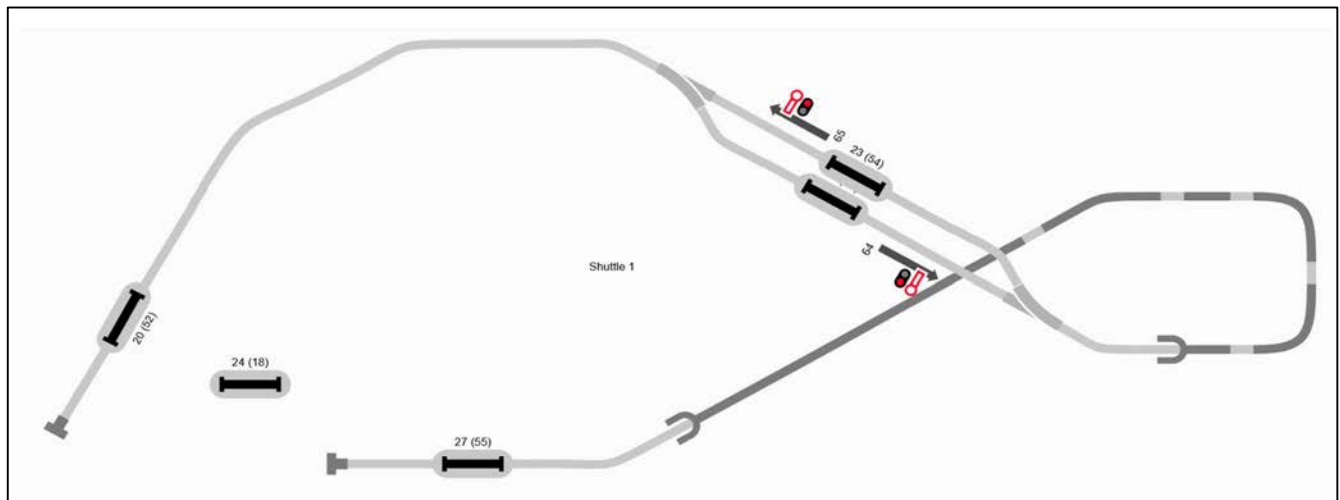


図1：2列車単線シャトル

'Lok x go' イベントと 'Lok x stop' イベントは、使用中の機関車の速度を設定するシンプルなものでした。図3は、ひとつの機関車に対する両方のイベントを示しています。シャトル制御でこれらのイベントをどのように使用するかは後ほど説明します。シャトルのような経路制御の一部である機関車の交換プロセスも簡素化します。

経路設定と2つの列車の難問

最初に、終端駅での制御について2つの方法を検討しました。1つは停止セクションを使用するもので、もう1つは複雑なイベントを必要とするものでした。(信号機やM84などによる)停止セクションの使用は誤りであることに気づいたので、その件の詳細は省略します。このアプローチがなぜ間違っていたかと、経路制御のポイントは中間駅であることを説明しておくことが重要だと思います。これは、イベントプログラミングを「やり直し」または再検討しなければならないことがあることを示しています。

終端駅に停止セクションを設けることが良い方法でない理由は、停止セクション自体にあります。列車が停止セクションに停車している間は、機関車デコーダーへのコマンドの送受信はできません。終端駅では機関車の進行方向を逆転する必要がありますが、いつ(信号機が「進行」になり)線路の電力が復旧し、いつコマンドを送信できるかという問題が生じます。後退コマンドを送信できる前に、列車が前進してしまう可能性があります。

中間駅(MLD)へのアプローチ

中間駅は2車線で、2つのポイントが右側通行に固定されている「デッド」状態です。ポイントがデッド状態でない場合は、ポイント切り替えステップを追加し、ポイントが右方向に設定されていることを確認してください。

中間駅に進入する列車は、対向列車が到着するまで待機する必要があります。どちらの列車が先に到着するかは設定できませんが、制御の前提条件として、両方とも中間駅に停車し、同時に発車することです。この前提条件については、以下に説明していきます。

中間駅での最初のイベント処理は、駅の両線とも在線かどうかを識別することです。2つの在線センサーは23(54)と25(53)で、両方が在線になって初めて、メインのイベントアクションが発生します。イベントの初めの条件判断部分を図4に示します。イベントが23(54)により起動されると、反対側の線路に列車が既に在線しているかどうかを確認しています。

CS2からのデータ移行のため、各コンタクトに2つの番号が必要です。最初の番号はCS3で自動で付けられたID、括弧内の番号はCS2でのIDです。両CSで番号付けのプロトコルは大きく異なります。私は両方の番号を併記することで、イベントの追跡を容易にしています。

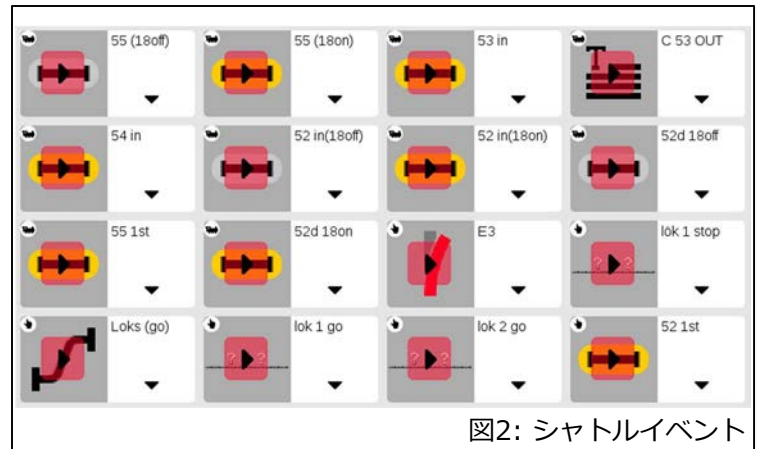


図2: シャトルイベント

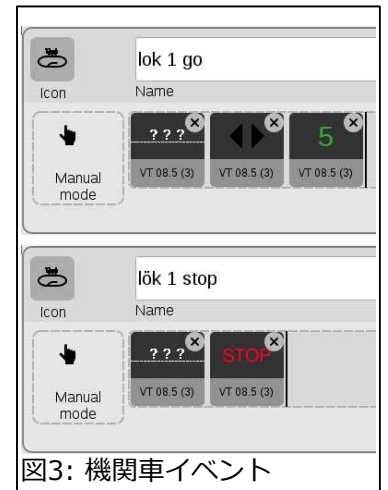


図3: 機関車イベント



図4: 23(54)の到着イベント

説明すると、1番線のイベントがコンタクト23(54)により起動されると、イベントアクションは2番線が占有され、コンタクト25(53)がアクティブになるまで実行されません。2番線のイベントとして、トリガーコンタクトと条件コンタクトが入れ替わった、逆のイベントがあります。両番線が在線になると、最後に起動されたイベント(コンタクト23またはコンタクト25)のみが処理されます。ただし、両方のアクションは実行されるので、ご安心ください。

中間駅イベント（階層イベント）

両方の列車が中間駅に到着したら、最も簡単なアクションは、適切な時間待機してから、両方の信号機を「停止」から「進行」に切り替えることです。ここでは、イベントを「ネスト（階層化）」します。ネストされたイベントは、頻繁に繰り返されるアクションの作成と編集の手間を削減・簡素化できます。

先ほど説明したイベントを作成します。両方の出発信号機を5秒後に「進行」にします。図5のイベントにはテキストアクションがあります。このアクションは5秒の遅延に設定されており、2つの腕木信号機を「進行」にする前にカウントダウンされます。これはトリガーコンタクトが割り当てられていないスタンドアロンイベントです(トリガーは手動のままです)。

このイベントをネストする(呼び出す)には、前項で説明した2つのイベント(MLD 25 と MLD 23)内にそれを配置するだけです。図6にそれを示します。

中間駅の出発イベント

レイアウトには、列車が中間駅を出た所にセンサーがないという欠点があります。信号機を「停止」にする手順を追加する必要があります。中間駅内のコンタクトの長さが不明だったため、駅が空くのに十分な時間を確保するために遅延時間を設定します。図7は、コンタクト25（53）で起動される出発イベントを示しています。コンタクト23（54）起動イベントにも同様のアクションとしましたが、遅延時間のみが異なります。

異なる長さの列車で有効に作用するように、これらを2つの別々のイベントにしました。

列車を区別して停止させる方法の検討

シャトル線上で2本の列車がそれぞれの終点に向かってしているとします。最大の問題は、どちらの列車を停止させるかということです。それぞれの列車がどちらの方向に向かっていているかを知る手段がないからです。例えば、列車「A」が駅「1」に向かっていている場合（そしてまだ到着していない場合）、駅「2」に列車「A」を停止させるように指示することはおかしいです。

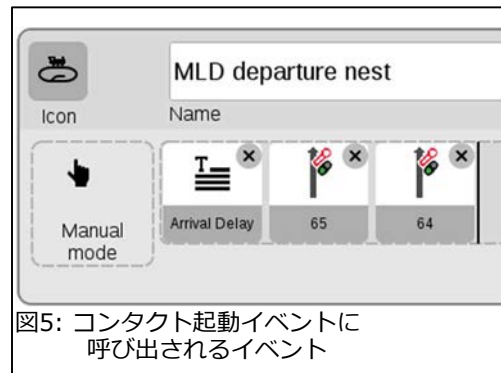


図5: コンタクト起動イベントに呼び出されるイベント

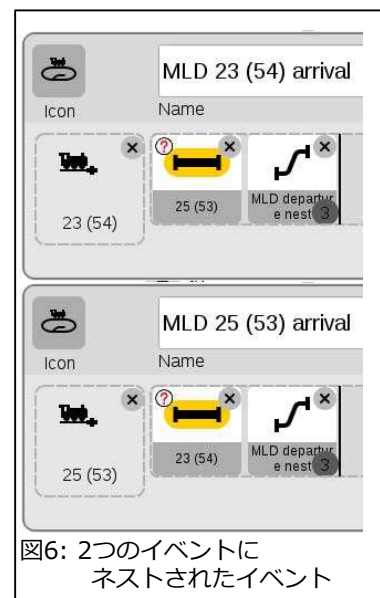


図6: 2つのイベントにネストされたイベント



図7: 中間駅の出発イベント

制御コンタクトにより機関車を「追跡」する

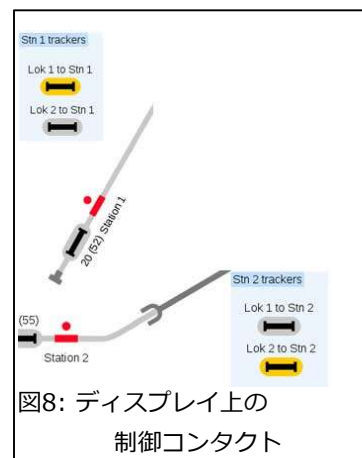
解決策は、シャトル線内の各機関車の行き先を示すマーカーとして「制御コンタクト」を使用することです。機関車と目的駅の組み合わせごとに、列車追跡用の4つの制御コンタクトを作成し、'Lok1 to Stn1'、'Lok2 to Stn1'、'Lok1 to Stn2'、'Lok2 to Stn2'という名前にしました。

制御コンタクトはソフトウェアインジケータであり、レイアウト上に物理的なセンサーはありません。また、イベントを起動することもできません。イベント内で使用される「？」マークの付いた条件コンタクトと同様に使用されます。また、イベント内のアクションとして表示を変更することもできます。

イベントリストがあふれるのを防ぐため、ここでのイベントを別のイベントグループにしました(一度設定するだけです)。図8に示すレイアウトページには、列車追跡用の4つの制御コンタクト(CC)を表示しています。イベントでインジケータとして使用されます。

CCを使用するイベントは、構造はシンプルですが、内容は複雑です。図9は、CCを使用するイベントを示しています。このイベントは、追跡コンタクトのステータス(列車の行き先)を反転するためのもので、まず自身のステータスをチェックし、真の(「？」条件と一致する)場合、逆のCCのステータスをアクティブ化し、自身のステータスを非アクティブ化します。各機関車には、stn1またはstn2の2つの行き先しかありません。このプログラムでは、各機関車に対して、2つの追跡コンタクトのうち1つだけを点灯させます。

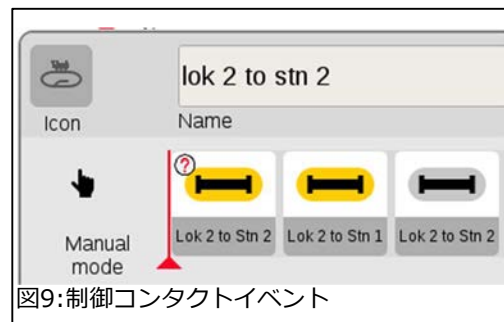
これらは手動モードのイベントであることにご注意ください。これらは別のイベントによって起動されるイベントであり、説明が難しいところです。



制御コンタクトの適用

図5のイベントに戻りましょう。このイベントでは、中間駅の機関車が終端駅に進むことを許可されています。各機関車の推定目的地を確認するためのIFマクロを追加します。ここでは、追跡コンタクトをガイドとして使用して各機関車の目的地を推定します。「推定」と述べたのは、CCには物理的コンタクトがないことを覚えておくことが重要であり、この方法が機能するには、推定条件と物理的条件を一致させる必要があります。たとえば、CCがLok1がStn1に向かっていることを示している場合、Lok1をStn1の方向に走行するように制御する必要があります。

IFマクロイベントには、独自のコンテナが必要です。これは、マクロを最初に作成したときに表示される空のコンテナです。図10は、IFマクロの設定前と、シャトルを有効にするために行われた設定後を示しています。マクロ内の各コンテナの内容と、それらがどのように構築されるかについて説明します。



最初のコンテナは、IFマクロが動作するために満たさなければならない条件を厳密に規定しています。条件が満たされた場合、THENコンテナのアクションが実行されます。条件が満たされなかった場合、ELSEコンテナのアクションが実行されます。

2列車シャトルでは、2つの異なる条件があります：

条件1：Lok1がStn1行き、Lok2がStn2行き

条件2：Lok1がStn2行き、Lok2がStn1行き

(各列車は反対方向へ向かうので)

ひとつのIFマクロには、これらの条件のうち1つだけを記述します。できるだけ簡単に説明します。

条件判定が「真」であれば、THENコンテナが実行されます。条件判定が「偽」であれば、ELSEコンテナが実行されます。ELSEコンテナを記述する必要がない場合もあります。

さて、IFコンテナを見ていきましょう。IFコンテナには1つの条件しか記述できませんが、ここでは、条件が2つあります (Lok1 to Stn1 と Lok2 to Stn2)。解決策は「ANDマクロ」です。

「ANDマクロ」には、複数の条件を配置でき、すべての条件が満たされた場合にのみ有効になります。条件1を表す「ANDマクロ」に、2つの追跡コンタクトを配置しました(図11)。そして、この「ANDマクロ」を「IFマクロ」内のIFコンテナに配置しました。

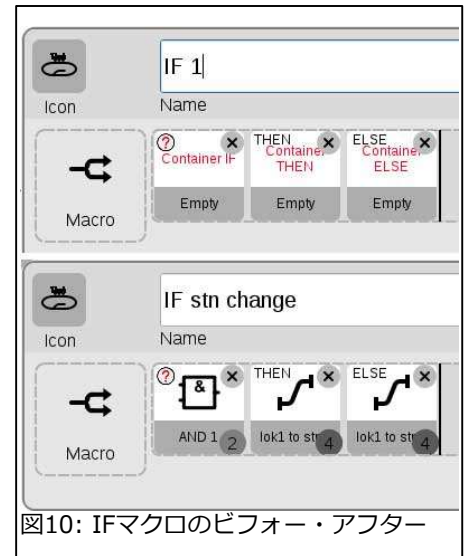


図10: IFマクロのビフォー・アフター

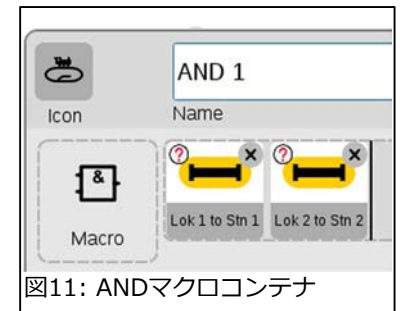


図11: ANDマクロコンテナ

IFマクロで何が起きているか

IFコンテナの条件が満たされた場合、THENコンテナ内のアクションが実行されます。ここで、“lok1 to stn2” というイベントを作成しました。図12は、2つの列車がIFコンテナで尋ねられた条件とは反対方向に移動していることを示すために、4つの追跡コンタクトを設定していることを示しています。このイベントをTHENコンテナに置きました。

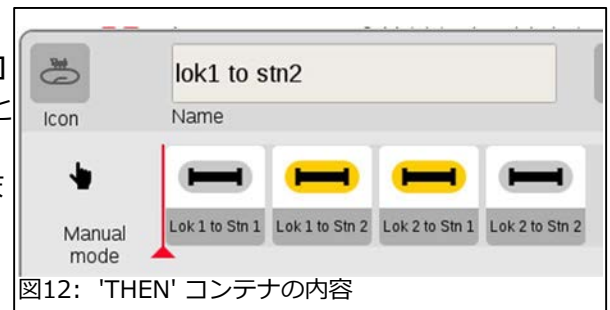


図12: 'THEN' コンテナの内容

IFコンテナの条件が満たされなかった場合、ELSEコンテナのアクションが実行されます。ここには、“lok1 to stn1” というイベントを作成しました。IFコンテナで尋ねられた条件に従って、追跡コンタクトを一致させるアクションを設定しました。IF条件が満たされなかったため、条件はIFコンテナ内の両方のコンタクトが「オフ」であることを示しています。したがって、ELSEコンテナは「オン」に戻す必要があります。アクションは図13に示します。

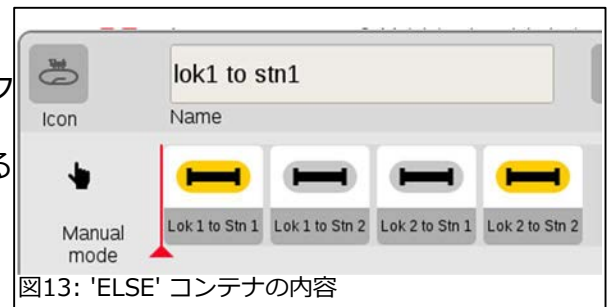


図13: 'ELSE' コンテナの内容

シャトル制御で何を行っているかですが、各列車が中間駅から出る時点を、それぞれの行き先を示す追跡コンタクトの設定に使用しています。中間駅へ入線するタイミングは両列車で同時ではありませんので。

終端駅で追跡コンタクトを設定することも検討しましたが、この記事の範囲を超える複雑さが生じました。中間駅は、両列車の正確な追跡のための共通ポイントとなり、行き先の正確な評価を可能にします。

ここまで、CS3のマクロイベント内でネスティングを使用する方法を示しました。これらは、機関車の擬似的な追跡をするために使用されます。次の記事では、2列車単線シャトルプログラミングを完成させ、両端駅での追跡コンタクトの使用方法に焦点を当てます。

Curtis Jeung

今後の予定：

- **Milwaukee Trainfest 2025**
November 1-2, 2025
- **ONLINE Webinar – Delayed one week due to vacation schedule.**
August 20, 2025

デジタル、技術、および製品関連について、リックとカーティスに質問するには：

Phone: 650-569-1318 Hours: 8:00am – 5:00pm CT. Monday through Friday
E-mail: digital@marklin.com