

電子連動ソフトウェア

電子連動アプリ概要

仕様

アプリは、下記機能を有する。

- DCC コマンドステーション DSAir2 と無線接続し、ポイントなどを制御する
 - 初期 Ver はポイントのみを制御する。在線情報は取得できないので、在線情報の基づく進路開通制御機能は OFF にする。
 - 将来的には信号機やレールへの電力供給リレーも制御可能とする。
 - 在線検知情報を取得、常時状態監視しそれに基づく制御を行えるようにする。
- 線路配線をユーザーが自由に描き、連動論理を構築できる。
- 機器を接続せずに、作成した連動論理データを検証できる。

アプリはブラウザ上で動作する Web アプリとする。Javascript 言語を用い、html ファイルからコントロールする。一部機能はタブレットなどのタッチパネルでの利用も想定した GUI とする。

アプリ基本画面

アプリの画面はレイアウトの線路配線を模した配線略図で情報を表示し、進路開通操作を行う配線略図画面と、アプリ内で定義されているデータの一覧とその状態を表示する補助画面の大きく 2 種類の画面とする。

配線略図画面では画面に描かれる配線略図に色によって線路の状態を示し、現在開通中の進路や車両の位置を表示する。本装置における進路開通の操作方法は、レイアウトの規模によらず、常に進路選別式による進路開通操作とする。配線略図画面にはレイアウトの線路配線のほか、ユーザーが進路開通操作を行うための”発点てこ”と”着点ボタン”を配置する。進路選別式のため、発点てこは進路の始点となる線路上に配置し、着点ボタンは進路の終点となる位置に配置する。

補助画面では、運用する電子連動装置で登録されているデータが一覧で表示される。補助画面には進路一覧情報のほか、各軌道回路の状態表示一覧や現場信号機の詳細情報一覧表示など、設備ごとに詳細情報を表示する画面がある。進路一覧では、進路の開通状態などのステータスが表示され、進路開通中は赤色にし強調して表示する。

アプリの操作

アプリの操作はマウス操作を基本とする。進路の開通操作を行う際は発点てこと着点ボタンの選択をマウス操作で行う。配線略図画面はマウスホイール操作とドラッグ操作で画面の移動をすることができ、大規模なレイアウトに対しても不自由なく直感的に操作できる。一部のショートカットキーに対応し、ファイルを開く際は「O（オー）」、レールパーツの回転には「R」といったキー操作を実装し、マウス操作と併用して操作性を向上させる。配線略図画面の画面スクロールや発点てこと着点ボタンの選択はタッチ操作で行えるようにする。

基本動作

アプリの基本動作は、現場通信処理、データベース更新、連動処理、画面更新の4つのステップを繰り返し動作するようにする。状態監視機能の実装後は、鉄道模型の車両はレイアウト上を走り続けるため、現場の装置から定期的に軌道回路の在線情報等を取得してアプリのデータ更新を行う必要がある。データ更新を行わないと、車両の在線位置が更新されないため、レイアウトの車両位置とデータが不一致して正常に動作しなくなる。

アプリケーションは、現場の装置から定期的に情報を取得してデータベース更新を行い、取得した情報をもとに進路の自動復位（設定した進路の自動解除動作）を行うことや、閉そく運転に関する編成の処理、鎖錠中の進路の解錠（進路区分鎖錠）などの連動処理を行う。連動処理後は、画面上の配線略図に表示する。一連の流れを繰り返し処理して運用する。

繰り返し処理の処理時間間隔は0.1秒間隔で現場装置の情報更新を行う動作とする。繰り返し処理の更新周期が長すぎる（例えば1.0秒）と、レイアウト上の車両位置とアプリの画面表示内容や連動処理にズレが生じ、編成番号の制御タイミングが遅れるなどの問題が生じる。逆に更新周期が短い（0.01秒）と、現場装置間との通信頻度が多くなり、情報の輻輳が生じて現場装置の処理が追い付かなくなる。

ユーザー補助機能

アプリは、誰が見ても容易に取り扱える画面の表示・操作方法とする。レイアウトの配線略図を画面に表示し、直感的に操作できるだけでなく、運用するために必要なシステム設定やシステムの初期化機能などの補助機能も操作性が悪くならないようにデザインする。

アプリ起動時のシステム初期化

アプリ起動直後はデータ定義のない白紙の配線略図画面、空欄の進路一覧ウィンドウを表示する。ユーザーは、連動情報ファイルを開いて運転を開始する。システムの初期化機能はボタンにより実行し、読み込んだファイルの内容と現場装置との設定内容の同期を行う。具体的には、ポイント開通位置の初期化（すべて定位位置に切り替え）、アプリで設定した信号機の設定構成と現場の機器構成の一致確認などを行う。

初期化中はプログレスバーを表示した初期化中ウィンドウを表示し、初期化処理中の間、どの項目を処理しているかリアルタイムに確認できるようにする。

連動装置の設定項目

連動装置の動作を決定するシステム設定はアプリ上でシステムプロパティとして設定項目を集約し、連動装置の動作や現場装置の設定を行う。タブにより機能ごとに各種設定を行えるようにする。

- 描画設定
 - 連動装置の登録進路数・連動種別、制御信号数、軌道回路数の表示
 - 軌道回路名、線路名、ポイント番号、ポイント DCC-ID の表示有無
 - 編成番号の画面表示有無
 - 配線略図画面のサイズ（画面スクロールの範囲を決める）
- 接近鎖錠・保留鎖錠標準秒数の設定（進路設定時、個別設定しない場合に適用する秒数）
- 進路の自動復位有効・無効

データ別の詳細設定

アプリには様々なデータテーブルがある、線路データ、ポイントデータ、信号機データ、進路データなど、アプリ内で作成される各種データは、データごとに専用のプロパティウィンドウで設定できるようにする。専用のウィンドウで設定できるようにすることで設定漏れを防ぎ、各項目をまとめて設定することができるようにする。例えば線路プロパティでは、線路名称、参照する軌道回路データの設定を選択する。

配線略図画面でのユーザー操作機能

配線略図画面では、ユーザーが特定の設備を選択した場合は強調表示するようにする。レールを選択した場合は、レールが強調表示され、水色に表示される。同様に、進路において選択した進路が走行する線路を水色に、発点てこを選択した場合はてこを水色に着色する。

連動装置設定項目の自動検証

ユーザーが入力した項目が正しく設定されているか、誤って設定していないか自動検証する機能を盛り込む。検証機能は設備の分類ごとに以下の項目に分かれる。

- 配線略図検査
 - 作成した配線略図の各パーツ群が正しく接続されているかを検証
- ポイント検査
 - ポイントデータが現場のポイント制御番号と一致しているかを検証
- フィーダー割当検査
 - フィーダー切替での電源出力先が複数に設定されていないかを検証
- 信号設備検査
 - てこの配置状態や信号機の現示系統が崩れていないかを検証
- 軌道回路検査
 - 軌道回路の番号が配線略図内の離れた箇所でも複数設定されていないかを検証
- 連動検査

- 進路データと信号機データの連携など、各種設定に誤りがないか検証
- 現場設備検査
 - アプリ上での設定情報と現場装置（信号機など）の設定情報が合致しているか検証

上記すべての検査項目が完了しないと連動装置の運用ができないシステムとする。検査にて問題を発見した場合にはユーザーに通知し、データの修正を促す。検査項目が多い検査の場合は、検査の結果を専用のウィンドウで表示する。ウィンドウ内での表示は問題がなければ黒色、設定不備ではないが注意を促す内容を黄色、設定不備の場合は赤色で表示する。問題の項目が見つかった場合、ウィンドウ内の問題項目を確認して修正対応できるようにする。

また、すべての検査項目の状態は小ウィンドウにて表示し、検査未完了の項目を一目で確認できるようにする。赤色は検査未完了で、緑色で検査完了済みを表示し、アプリが現在のステータスにあるかを一目で確認できるようにする。

データ作成から運用開始までの流れ

アプリを起動し、配線略図作成から連動装置として運用を開始するまでの流れは、次の通りである。

1. 配線略図作成→配線略図検査
2. 信号設備配置→信号設備検査
3. データ紐づけ→ポイント検査、フィーダー検査、軌道回路検査
4. 進路作成→連動検査
5. 現場設備検証→現場設備検査
6. 運用開始

作成するデータの種類ごとにステップを分け、ステップ単位で検査を実行する。検査機能を段階的に実行することで、作成したデータの内容に問題がないか1つ1つ確認しながら運用開始まで進めていけるようにする。

連動装置として運用を開始するまでの流れは、まずユーザーが配線略図データを作図していく。作図完了後、配線略図検査を実施し配線略図を確定させる。配線略図確定後はて

こ・信号機等の信号関係設備を配置し、連動装置で制御するすべての設備を配置する。全設備の配置が完了したら、作成した配線略図・信号機に現場のポイント番号、軌道回路、信号機のデータ紐付けを行う。最後に進路データを作成し、進路データと制御する信号機を設定して連動検査を実行する。連動検査が完了すると、アプリ上のすべての設定が完了する。すべての設定が完了後は、現場設備検査にて現場の各装置と通信して設定情報を現場装置に書き込み、ポイント切替や進路開通確認など動作確認して運用開始となる。

配線略図作図機能



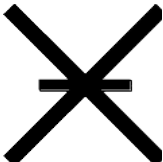




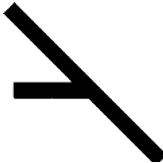
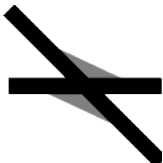

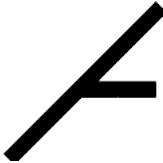

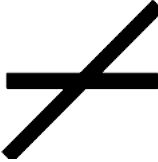

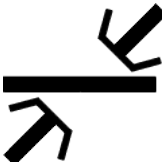
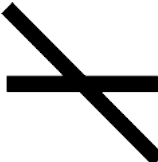
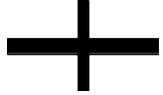
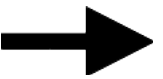
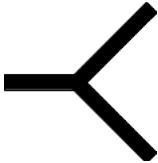


概要

汎用的な装置とするため、ユーザーがアプリ上で配線略図を作成し、連動論理のデータベースを構築できるようにする。この配線略図を描くための機能が配線略図作図機能である。作図機能では、ユーザーが線路線形を自由に描けるようにするため、基本的なレールをパーツ化し、パーツを組み合わせることで線路線形を描くものとする。画面上に表示する配線略図はレイアウト全体を横長の線路配線に簡略化して作図する。

レールパーツ化

配線略図を描くために必要な様々な線路の表記をパーツ化する。パーツは直線レール、曲線レール（折れ曲げ線）といった基本的なレールのほか、各種ポイントレール、クロッシング等をパーツとしてシンボル化する。

定義したシンボルを表に示す。

直線		斜め直線		絶縁クロス	
曲線		クロッシング		左ダブルスリップ	
左ポイント		左ポイント (45度)		右ダブルスリップ	
右ポイント		右ポイント (45度)		オーバークロス	
左クロッシング		エンドレール		アンダークロス	
右クロッシング		レール絶縁		ジャンプレール	
Y字ポイント		レール絶縁 (45度)		ダブルクロス	

ユーザーが作図する際にはこのレールパーツを組み合わせて作図し、レイアウトの線路配線を画面内に再現する。パーツを画面内に配置する際は、配置時にパーツの回転を行う機能を有し、例えば曲線パーツは図の角度から 0°、90°、180°、270°と回転して配置できるようにする。配置はマウス操作で行い、それぞれのパーツを繋げて配線略図を完成させていく。

レール絶縁は配線略図上で軌道回路を区分する際に用いる。使用するとアプリは電氣的に区切られていると認識する。アプリ上でレール絶縁を挿入する位置は、現実のレイアウト上に挿入した絶縁となるべく位置が近くなるように配置して配線略図を作成する。

ジャンプレールは画面上での線路配置を見やすくするために離れたレールパーツ同士を同一の線路としてみなすために設けたパーツである。一般に鉄道模型のレイアウトはエンドレス（周回するレイアウト）なため、画面左端と右端で行き止まりではなく、同一の線路とみなすために使用する。

レールパーツの線路化

前項で述べたレールのパーツを画面上に配置したのみでは、一つの連続した線路としてアプリ上では認識していない状態である。そこで、配置したレールパーツを連続した線路として認識するために、レールパーツ相互間をデータで結びつける。具体的には各レールパーツに ID 番号を振り、双方のパーツ ID を参照することで一体のデータとする。ID 番号 0 はデータなしを表し、未接続となる箇所には値に「0」を書き込む。ポイントレール等の複数の接続先を有するパーツはすべての接続先を記憶する。

データ化は互いのパーツが画面上で接続されていることを条件とし、接続していないレールパーツは別の線路として扱われる。

線路データの生成

前項で各レールパーツの線路化を行うと、次に閉そく区間や軌道回路区間などの制御しやすい区間に区切るために、電氣的に同一区間と分類される複数のレールパーツを 1 つの線路データとして別に付番する。これを線路データ化と称する。ポイントレールは異なる線路間の境界点とみなし、それぞれ異なる線路番号を付番する。この付番により大量に接続された直線パーツや曲線パーツを直接参照せずに、1 つのデータ塊として処理することで進路データベース化などの処理を簡略化する。線路データ間は相互に接続相手の線路番号

を参照できるようにし、アプリ上においても線路配線の接続関係を持たせる。本装置で進路を制御する際の線路の単位は、この線路データを基本とする。

ポイントレールの開通方向表示

ポイントレールは開通方向を画面上に表示し、ユーザーが現在どの方向にポイントレールが開通しているかを直感的に分かるようにデザインする。図はポイントレールが直進・分岐のどちらを向いているかを示した図である。この開通方向の表示切替は現場のポイント制御装置から受信した情報で判断して描画する。ポイント制御装置にてポイント切替を行うことに失敗した場合は、開通方向の表示は切替前の状態を保持するようにする。

線路の状態表示

作成した配線略図の各線路は、進路が開通している場合や軌道回路内に車両が在線している場合などの情報を画面に色で表示する。線路の色によって意味合いを分け、下図に示すような状態表示を定義する。在線時は赤色、進路が開通し線路が進路によって占有（鎖錠）されている時は黄色、現場機器から動力電源の通電を検知した場合は緑色、ユーザーが線路を選択し、何らかの操作をしようとしている場合は水色を着色する。

線路の状態表示が重複した場合は優先順位を設ける。例えば在線と鎖錠が同一線路内で生じた場合は鎖錠よりも車両の位置が優先して着色表示されるようにする。優先順位は選択の表示が最も強く、次に在線表示とする。

定義	色	カラーコード	優先順位
選択	水色	#00FFFF	高い
在線	赤色	#FF0000	
鎖錠	黄色	#FFFF00	
通電	緑色	#00FF00	
在線なし	白色	#FFFFFF	低い

線路データと現場情報の紐づけ

軌道回路の在線情報等は、生成された線路データにデータ紐付けをすることで情報を反映する。軌道回路情報については現場情報データとして、線路データとは別に管理する。これは、線路略図上で生成される線路データの数と軌道回路の数が一致しないためである。例えば、ポイントを含む軌道回路の場合、アプリ上の線路データは1つではなく、複数に及んで定義されるが、軌道回路は1つのみとなる。複数の線路データに軌道回路情報を伝達するため、軌道回路の情報を線路2にデータ紐付け（リンク）を行い、軌道回路の情報を線路2に転送する。在線情報を取得した線路2データから隣接の線路データ3,4にはポイントの開通方向に従って情報伝達を行い、1つの軌道回路として動作するようにする。この情報伝達は配線略図上で絶縁レールがあるパーツまで情報を渡す。

信号関係設備のデザイン

信号関係設備は初期版ではその多くは実装しない。ただし発点てこ、着点ボタンは進路制御に必要なため実装する。

発点てこは、進路の種類ごとにこの色を変更して描画する。停車場間を走行する本線進路と停車場構内にて入換を行う進路の発点てこが同一色とするとユーザーにとって判別しづらくなるため、本線を走行するために使用するてこを赤色、入換信号機に使用するてこを白色、入換標識に使用するてこを緑色、開通テコを黄色とする。着点ボタンは進路の種類に依らず1種類とし、ボタンの色はWindowsのシステムカラーの灰色とする。

てこの種類	つまみ色	デザイン	カラーコード	用途
信号てこ	赤		#FF0000	本線用
入換信号てこ	白		#FFFFFF	構内入換用 (防護区間あり)
入換標識てこ	緑		#00FF00	構内入換用 (防護区間なし)
開通てこ	黄		#FFFF00	過走防護用
着点ボタン	灰色		#EBEBEB	全進路到着点

進路データ

進路データ概要

進路データは電子連動アプリの中で最も重要なデータである。進路データは作成した配線略図と発点てこ・着点ボタンの配置情報から進路データを生成し、生成した進路上に含まれる各線路番号のデータリスト化と、進路による切替対象のポイントレールの抽出と方向の決定、制御する信号機を設定する。進路データは進路1つにつき1データとして作成し、データベースを構築する。例えば100進路を持つ連動装置であれば、100個の進路データを生成する。

進路データには走行する進路の始点となる線路から到着点となる線路までのすべての線路番号をデータ群として保持し、線路データ間に挟まれたポイントレールを進路の方向に合うようにポイント切替を行う。

作成した進路データは発点てこ紐付けを行う。てこが進路データを参照することで、どの着点ボタンに対して進路設定可能かを表示できるようにする。

進路データの定義

進路データは線路のデータ群のほか、進路データの管理番号、進路の方向、進路の状態、進路の種類、制御する信号機番号、進路内を走行する線路のすべての番号、切替対象とするポイント番号、接近鎖錠・保留鎖錠の秒数、進行定位の有無を1つのデータ群として定義する。このデータをデータベースとして構築し、連動装置の進路データとして管理する。

アプリ内でのデータ管理は、進路データの進路番号が0以外の値を持つとき、進路データを有効なデータと扱う。これはアプリ内で進路データの追加作成、データ削除に対応するためである。有効な進路データを削除する場合、進路番号を0に書き換えることでアプリ内では削除された進路として処理する。進路番号の付番はデータベースのメモリアドレスが0番地ならば、番地0の値に1加算した番号1を進路データの番号とする。生成できる進路データ数には上限を設けない。

アプリはこの進路データを使用して、ポイントレールの切替制御のほか、進路相互の支障判定、進路開通により制御する信号機の指定、接近鎖錠・保留鎖錠の時素解錠動作に使用する。

進路で走行する線路データ生成のための経路探索

進路で走行する線路のデータリスト生成は、発点てこ・着点ボタンの位置関係から配線略図のデータを用いて経路探索して生成する。経路探索の順序は、ポイントレールの直進方向を優先して探索し、見つからなければ直進方向として探索したポイントレールまで戻り、探索する方向を分岐側へ切り替えて探索する。

発点てこが配置された線路データを起点とし、右方向または左方向に経路探索し、目的地となる着点ボタンが配置されている線路番号を探す。すべての線路データは隣接する線路データの番号を保持しているため、線路データ相互間で接続されている。隣接する線路データの番号を順次参照することにより経路探索が可能で、探索方法は目的となる線路番号が見つかるまで総当たり探索とする。探索の結果、エンドレール等の終点となる線路番号まで到達しても目的の線路番号に到達しなければ、ポイントがある線路番号まで探索を戻し、別の経路を探索する。目的の着点ボタンにまで到達したならば、進路確定として進路データを生成する。

鎖錠概念の導入

概要

進路相互間の支障判定や鎖錠機能などはすべてアプリのソフトウェア処理で実現する。連動装置には様々な鎖錠の機能があり、すべての機能を備えることで安全に運転できる装置となる。

てっ査鎖錠

てっ査鎖錠とは、「ポイントを含む軌道回路内に車両があるとき、その車両によってポイントを切替できないように鎖錠すること」である。線路データが在線の情報を取得した場合、線路データの端部に接続されるポイントレールは切替できないように鎖錠する。ポイントレールの同時切替が必要な渡り線等の場合は車両在線により鎖錠されるポイントが1箇所でも含まれる場合、すべてのポイントで鎖錠し、ポイント切替できないようにする。

進路鎖錠

進路鎖錠は車両が進路内を走行し、進路の終点へ到着するまで進路内の線路を鎖錠し、発点てこを復位（進路の解除操作）しても鎖錠を継続させる機能である。進路データで定義

された各線路は進路の開通操作により、進路によって走行する線路を鎖錠（占有）する。発点でこれを復位し、車両が進路内を走行して進路の終点まで到達した場合は進路鎖錠をすべて解く。

進路が複数の軌道回路で分割されている場合は、軌道回路単位で進路の鎖錠を解除する。分割して進路が解除される進路鎖錠のことを「進路区分鎖錠」という。

表示鎖錠

表示鎖錠は、信号機が進行信号から停止信号に切り替わらない場合、進路の解除を行わない機能である。進路の解除操作をしたならば、進路により制御していた信号機は直ちに停止信号へ切り替わらなければならない。進行信号を現示していると運転士から見れば進入して良いことになるため、進行現示から停止信号へ変化することに失敗すると、危険な動作になる。

今回の装置では、進路の解除操作を行った際、現場の信号機ユニットに対して停止信号制御を行う。停止信号制御に失敗した場合、進路の解除を行わないようにし、表示鎖錠としている。

接近鎖錠・保留鎖錠

進路の始点となる区間に車両が接近した直後で進路の解除操作を行った場合、高速で停車場へ接近する車両は急に停止できないため、信号機を超えてしまう恐れがある。信号機を超えた先にはポイントがあるため、進路の解除と同時に鎖錠を解くと、ポイントが切り替わる可能性があり、脱線の恐れがある。そこで、接近区間に車両が存在するならば、進路解除操作後は一定時間、進路の鎖錠を継続する。これを接近鎖錠という。

今回のアプリでは、接近鎖錠を行う区間は進路始点となる軌道回路の区間に在線した場合に、接近鎖錠を行うようにしている。鎖錠の時間は進路ごとに設定できるようにする。鉄道模型の場合、実際の鉄道よりも急制動が可能なため、設定秒数は10秒程度が妥当である。似た機能として、保留鎖錠というものがある。保留鎖錠の場合は、接近区間内に車両が存在するか否かに関わらず進路の解除操作後に一定時間、進路を鎖錠する。

アプリ上では、接近鎖錠・保留鎖錠により進路の解除に待ち時間が発生している間、「時素解錠中」のテロップを表示する。

片鎖錠

片鎖錠は2つの設備間で一方向にのみ鎖錠する仕組みである。片鎖錠の例として、開通てこによる過走防護と信号の現示アップがある。

過走防護とは進路終点となる線路に車両が進入する際、ブレーキ操作遅れにより進路の終点で止まりきれず、進路終点を超えてしまう過走に対して他車両と衝突しないようにする仕組みである。過走防護がない場合、隣接する別の線路を走行する車両と衝突する恐れがある。そこで、進入時に確認する信号機の信号を警戒信号とし、進入する車両の速度を低速度とする方法と過走の恐れがある区間を予め進路により鎖状する方法の2つを用いて防護を行う。過走防護を行うと衝突は避けられますが、前者は進入速度が遅いため、時間がかかること、後者は競合する進路の同時進出入ができないというデメリットがある。

そこで片鎖錠という概念を用いて、状況に応じて同時に進出可能とするか選択できる施策を行う。過走防護により過走の恐れがある区間の鎖錠を行ったならば、車両が高速度で進入するため、防護区間は車両が到着点となる線路で停止するまでは鎖錠を継続しなければならない。すなわち、過走の恐れがある区間の鎖錠は進路により鎖錠する関係となる。

過走防護の鎖錠は進路により鎖錠するが、過走防護により進路を鎖錠することはない。従って過走防護と進路の鎖錠関係は一方向な鎖錠関係になる。この鎖錠関係を片鎖錠と言う。

開通てこは進路の到着点となる線路に設ける黄色のてこであり、進路を延長する役割を持つ。今回のアプリでは、開通てこによる現示アップを再現する。開通てこにより過走区間の防護がされると、現場の信号機に現示アップ指示を行い、信号機は警戒信号から注意信号へ変化する。開通てこを操作せずに進路を開通し、警戒信号現示で進入する場合は他進路の同時進出入ができるため、過走防護のデメリットを回避できる。

線路の占有化による進路支障判定処理

進路相互間の支障判定は線路データがある特定の進路によって鎖錠（占有）されている場合、同一の進路を共有する他の進路は設定不可とする排他処理を行い、進路支障と判定する手法で実現する。本物の電子連動装置ではこの方式を「軌道回路予約方式」といい、今回製作した電子連動装置においても進路相互間での支障判定は同様の論理にて動作するようプログラムする。

開通しようとしている進路の全線路データを参照し、線路データが占有していないかを一つ一つチェックしていく。進路内の全線路が占有でないと判定できたならば、進路開通処理を実行する。

進路支障判定処理の例外

進路支障判定処理には例外処理を追加する。それは、走行する進路が複数に分割されて制作され、それぞれ進路開通制御を行った場合である。具体的には停車場を通過する優等列車灯を運転するために、停車場外から進入する進路と、停車場内から進出する進路を同時に開通させようとする場合である。

この例外処理は進路の始点と終点相互間のみで適用し、進路の中間から分岐するような分割された進路を制御することはないようにする。

この例外処理は進路開通操作時の他、進路を解除する際も同様に適用する。進路の始点と終点が重複しており、どちらか一方の進路にて開通制御中であれば解除しない処理を行う。この例外処理により、停車場へ進入する場内進路と停車場から進出する出発進路相互間で干渉しないようにする。

進路の開通動作

進路の開通操作

進路の開通操作はアプリ画面上で発点てこをクリックして進路の始点を選び、次に着点ボタンをクリックする。発点てこを選ぶと着点ボタン選択モードへ遷移する。予め設定した進路データベースに登録されている着点ボタンが青く表示し、選択可能であることをユーザーに示す。問題なく進路が開通すると、画面に進路開通の状態が表示され、進路により鎖錠されている線路は黄色に着色される。

進路開通・解除処理のタイミング

進路の開通制御と解除制御のタイミングは、以下に示すようにアプリが現場装置と通信を行い、データベースの更新処理が完全に完了したタイミングで処理するようにする。処理中にデータ更新され、制御しようとしている進路内の軌道回路が処理中に在線状態に変化するなどが無いようプログラムを構築する。

現場通信処理→データベース更新→連動処理（進路の開通・解除動作はこの期間で行う）
→画面更新→（現場通信処理へ）

進路の開通動作フロー

開通させたい進路を選択したあとは、フローチャートに従って動作する。進路開通動作が始まると、まず発点でここに紐付けられている進路データが呼び出される。進路内の線路データリストを用いて支障進路の判定を行う。その後、進路内の軌道回路に車両在線がなく、ポイント切替とフィーダー電源切替がすべて進路データに従って動作したならば、進路内の線路を鎖錠（占有化）し、他進路と重複制御しないようにする。最後に信号機に進行を指示する信号の現示をするよう制御を行う。信号機の現示制御まで処理が完了すると進路開通の動作は完了となる。

アプリから現場の各装置へ制御する際は、装置へ送信した後の応答（返信）をチェックし、応答が無い、または受信した内容に問題がある場合は異常動作とみなし、新ロンボ開通失敗として処理するようにする。

進路の解除動作

進路の解除操作

進路の解除操作は進路開通中の発点でこを中立の位置に戻すことで進路の解除動作開始となる。解除を行う際は、右方向または左方向に倒れている発点でこをクリックする。クリックした後は次項で述べる進路の解除処理を行い、鎖状していた線路を順次解く。すべての進路鎖錠が解かれると、進路の解除完了となる。

進路の解除動作フロー

進路の解除動作は、車両の位置により動作が変化する。接近区間となる進路始点に車両が在線している場合は接近鎖錠動作、車両が進路内に進入し、到着点となる線路まで到達する前であれば、進路区分鎖錠動作とする。進路の解除動作に影響する範囲に車両の在線がない場合は、鎖錠していた線路をすぐに解除し、進路開通を解除とする。また、進路に解除秒数が設定されている場合は、鎖状している場合は、保留鎖錠動作とする。

接近鎖錠、保留鎖錠、進路区分鎖錠を行う場合は、進路データの状態を進路構成中から状態変化させ、進路解除待ちの状態とする。アプリはデータベースの情報更新と連動処理を

繰り返し処理しているため、システム全体での動作が1周した次の周期で再度進路解除処理を実行する。次周期の解除処理で、軌道回路の在線情報などのデータベースに変化がある場合は、変化内容に応じて進路の解除処理を進行させる。

接近鎖錠動作は、進路データで指定した時間が経過後、進路の解除とする。接近鎖錠動作とする場合は、進路データのステータスを”進路開通中”から”時素解錠中”へ変更し、解除待ちのモードへと変更する。待ち時間はアプリ内で時間のカウントを行い、指定した時間が経過したならば再度フローを実行して解除処理を行う。保留鎖錠動作は無条件で、”時素解錠中”動作へ変更し、一定秒数の後、進路解除とする。

進路区分鎖錠の場合は、車両は時間の経過とともに、開通している進路に従って走行していくため、軌道回路の鎖錠を解いて進路区分鎖錠を行う。車両が進路の終点となる線路に完全に進入したら進路データの状態を”進路区分鎖錠中”から”進路未開通”へ変化させ、解除完了とする。

DCC コマンドステーションとの通信

DCC コマンドステーションには DesktopStation の DSAir2 を用いる。DSAir2 に挿入した FlashAir が構築するネットワーク (<http://192.168.1.41/>) に接続し、アプリからコマンドを送信してポイントなどを制御する。DSAir2 との通信プロトコルは以下に示す。

DSair Wi-Fi Specification

This page describes Wi-Fi interface specification of DSair2. DSair2 supports http REST like API. You can control DCC and Marklin locomotives and turnouts via http.

本ページでは、FlashAir ベースで開発された DSair に、ユーザーの皆様が開発・製作した機器やソフトウェアを接続してコントロールするための仕様を公開します。世界で唯一のオープンな無線・Wi-Fi 対応の DCC コマンドステーションとして、活用いただければ幸いです。

ここに掲載した仕様・コマンドについて、ライセンスはございません。個人利用、クラブ利用、商用利用を含め、常識の範囲でご自由にお使い下さい。

接続方法

FlashAir カードに Wi-Fi で接続してください。SSID は、初期は FlashAir_(は任意の英数字)、パスワードは 12345678 です。FlashAir W-04 においては電源投入後、10-20 秒程度で、Wi-Fi 接続が可能になります。DSair のアプリの Config 画面で、SSID とパスワードは簡単に変更可能です。なお変更後は切断されます。SSID とパスワードを変更した物に再設定してください。

同時接続モード(FlashAir がアクセスポイントにもなり、無線 LAN ルータにも接続されるモード) の場合は、数分の時間が掛かる場合があります。

DSair へのコマンドの渡し方は、FlashAir の CGI コマンドの共有メモリアクセスを使用します。

注: DSairLite では RaspberryPi PicoW をベースとしているため、FlashAir コマンドを模した機能をサーバーに実装しています。

DSair2 の場合:

<http://flashair/command.cgi?op=131&ADDR=0&LEN=64&DATA=DSair> 内部コマンド

DSairLite の場合:

<http://192.168.42.1/command.cgi?op=131&ADDR=0&LEN=64&DATA=DSair> 内部コマンド

DSair 内部コマンドを変えることで、任意の命令を DSair に送って制御できます。

DSair への命令・コマンド

1 関数一覧/Function list

Function	Command	Example	Notes
線路電源,Power	PW(PowerFlag)	PW(1)	PW(0)で線路電源 OFF、PW(1)で ON です。
進行方向,Direction	DI(LocoAddr,Direction)	DI(49155,1)	
ファンクション,Function	FN(LocoAddr,FuncNo,FuncVal)	FN(59140,2,1)	
車両速度,LocSpeed	SP(LocoAddr,Speed,SpeedStep)	SP(59140,0,2)	
ポイント,Turnout	TO(AccAddr,AccDirection)	TO(14337,0)	
アナログ,AnalogPWM	DC(AnalogSpeed,AnalogDirection)	DC(300,2)	線路電源 OFF の時だけ使用可能。

Function	Command	Example	Notes
S88 在線センサ,S88 Sensor	gS8()	gS8(1)	S88 在線センサのデータ収集スタート
Parameters	Meanings	Notes	
PowerFlag	OFF=0, ON=1	線路電源の状態を切り替えます。	
Direction	FWD=1, REV=2	0 は FWD と見なします。	
FuncNo	0-28	F0～F28 に相当します	
FuncVal	OFF=0, ON=1	ファンクションの状態です。	
Speed	0-1023	1023 で最高速度です。512 で 50%です。	
Speedstep	0-2	DCC のとき 128Step は 2 です。MM2 モードの時は 0 を指定してください。	
AccDirection	分岐方向=0, 直進方向=1	ポイントの切り替え方向です。	
AnalogSpeed	0-1023	アナログ電圧(スピード)。0 が停止、1023 が最大電圧です。	
AnalogDirection	0 or 1:FWD, 2:REV	アナログの極性。0 または 1 は FWD, 2 は REV となります。	

Example of speed command is the follwoing.

DSair にコマンドを送る例としては、以下ようになります。

```
http://flashair/command.cgi?op=131&ADDR=0&LEN=64&DATA=SP(59140,0,2)
```

You can only send as HTTP GET command.

HTTP GET コマンドを送るだけです。

2 コマンドの送信間隔

0.5 秒(500ms)間隔程度で、コマンドを送信してください。他の Wi-Fi デバイスからコマンドを同時に打ち込んだ場合、後着優先となります。

3 Address meanings / アドレスの考え方

LocAddr and AccAddr define 16bit address space including DCC and Marklin Motorola2. This means you can control locootives and turnouts as 16bit address.

LocoAddr, AccAddr のアドレスには、DCC とメルクリンの二つのプロトコルを包含し、さらに車両とアクセサリもカバーした 16bit のアドレス空間が定義されています。

Address type	range	Meaning	Notes
Marklin Motorola 2 Locomotives	0x0000-0x07FF	0(0x0000) - 255(0x00FF)	
DCC Locomtoives	0xC000-0x1FFF	0(0xC000) - 9999(0xE70F)	
Marklin Motorola 2 accessory	0x3000-0x37FF	1(0x3000) - 320(0x3140)	
DCC accessory	0x3800-0x3FFF	1(0x3800) - 2044(0x3FFC)	

実際にコマンドに使用する場合には、10 進数に直して使用してください。たとえば 0xC000→49152 となります。

DCC アドレス 3 を動かす場合は、LocoAddr には 49152+3=49155 を指定すると、DCC アドレス 3 の車両が動きます。MM2 アドレス 3 の場合は、LocoAddr には 0+3=3 を指定すると、MM2 アドレス 3 の車両が動きます。

ポイントの場合は、0x3800(16 進数)→14336(10 進数)をオフセット値にして、DCC ポイントアドレス 5 の場合は、14336-1+5=14340 を指定すると、ポイントが切り替わります。ポイントの場合は 1 はじまりなので、-1 しておくことが大事です。

4 JavaScriptでの操作

jQueryを使用した場合は、以下のようなコードでコマンドを送信できます。

```
function onChangeSpeed(inSpeed) {  
  
    var url = "/command.cgi?op=131&ADDR=0&LEN=64&DATA=SP(49155," + inSpeed + ",2)";  
    $.get(url, function (data) {});  
  
}
```

状態・ステータスの読み出し

5 取得方法

共有メモリの128バイト目以降からサイズ264バイト(ASCII TEXT)に、制御状態や車両・ポイントデータが含まれているので、これを解析することで、制御状態を把握することができます。

```
/command.cgi?op=130&ADDR=128&LEN=264  
N,0,0,0,155,00,2,22,000000000000;00000000000000;000000000000000000000000000000  
0000000000000000000000000000;0000,00,0,00000000/0000,00,0,00000000/0000,00,0,00000  
000/0000,00,0,00000000/0000,00,0,00000000/0000,00,0,00000000/0000,00,0,00000000/00  
0,00,0,00000000
```

6 パワーオン状態の確認

FlashAirの共有メモリのアドレス128にサイズ1のデータが格納されており、ここがYのときは線路電源ON、Nか0x00のときは線路電源OFFと判断します。

取得:
/command.cgi?op=130&ADDR=128&LEN=1

セット:
(setPower コマンドで自動的にセットされます)

7 ステータスデータの一覧

Status Data from DSair2(FlashAir shared memory) is ASCII text data.

データはすべてテキストである。S88データは、BASICからs88start命令が実行されると表示開始されます。実態は、gs8というコマンドをDSair2に送ると、S88機能が動作する流れになっています。

IndexByte	Size	Parameter	Definition	Notes
0	1	線路電源	ON="Y", OFF=0x00	線路の電源状態を示します
1	1	-	カンマ(,)	区切り文字
2	1	エラー番号		
3	1	-	カンマ(,)	区切り文字
4	1	FIRMWARE_VER		
5	1	-	カンマ(,)	区切り文字
6	1	制御車両数	(0-15, 16進数)	
7	1	-	カンマ(,)	区切り文字
8-10	3	線路電圧	120=12.0V	
11	1	-	カンマ(,)	区切り文字
12-13	2	出力電流	10=1.0A	
14	1	-	カンマ(,)	区切り文字

IndexByte	Size	Parameter	Definition	Notes
15	1	ハードウェア Ver		
16	1	-	カンマ(,)	区切り文字
17-18	2	送信回数		
19	1	-	カンマ(,)	区切り文字
20	1	S88 台数(1-2)		
21	1	S88 データ 1 バイト目 Low	S88 デコーダ 1 台目(16bit の場合)	
22	1	S88 データ 1 バイト目 High	S88 デコーダ 1 台目(16bit の場合)	
23	1	S88 データ 2 バイト目 Low	S88 デコーダ 1 台目(16bit の場合)	
24	1	S88 データ 2 バイト目 High	S88 デコーダ 1 台目(16bit の場合)	
25	1	S88 データ 3 バイト目 Low	S88 デコーダ 2 台目(16bit の場合)	
26	1	S88 データ 3 バイト目 High	S88 デコーダ 2 台目(16bit の場合)	
27	1	S88 データ 4 バイト目 Low	S88 デコーダ 2 台目(16bit の場合)	
28	1	S88 データ 4 バイト目 High	S88 デコーダ 2 台目(16bit の場合)	
29	1	S88 データ予約		
30	1	S88 データ予約		
31	1	区切り		
32-46	15	CV 応答データ(@CV,CVNo,Value,)		
47	1	区切り		
48-111	64	ポイント状態		
112	1	区切り		
113-	152	車両データ 8 台分	LocAddr,Spd,Dir,Func	

参考図

